ANNA KUCABA-PIĘTAL, MARCIN PAPRZYCKI

PARALLEL COMPUTING FOR COMPUTATIONAL FLUID DYNAMICS
RÓWNOLEGŁE OBLICZENIA DLA OBLICZENIOWEJ DYNAMIKI PŁYNÓW

INTRODUCTION
The main goal of this paper is to summarize the possibilities brought by parallel computers to the area of fluid mechanics. However, many of the issues raised here can be also applied to a broadly understood area of scientific computing. Fluid mechanics is one of the domains of physics substantially benefiting from the development of new numerical algorithms and a rapid increase of computational power of modern computers. The equations of fluid dynamics (Navier-Stokes equations) are partial differential equations containing both, first and second derivatives in spatial coordinates and first derivative in time. The time derivative appears as a linear term while the spatial derivatives quite often appear nonlinearly. Also, except for very limited special cases, the analytical solutions of these complicated systems of equations are not available. Only application of numerical approximations and computer generated solutions allowed solution of many real-life problems involving: forces on bodies (skin friction, drag, lift), efficiencies (turbine, diffuser), heat transfer, weather prediction and so on. These solutions lead to the development of practical applications (for more details see [2], [7]). Nowadays, problems concerned with computation of the boundary layer, turbulence, separation, vorticity and general unsteady flows for complex geometries of the flow fields are of particular interest to the researchers in the field. All these problems involve the solution of highly nonlinear Navier-Stokes equations. Progress in finding solutions is slow and involves adapting the well-known meth-
ods to modern high-performance computers or construction of new algorithms targeting the new computer architectures directly.

FEATURES OF NUMERICAL APPROXIMATIONS

The invention of the digital computer helped to develop a new meaning of the concept of approximation. This new understanding concerns the numerical approximation to the set of equations representing a mathematical model of a physical system. Approximate numerical solutions are sought for real-world problems, for which no analytical solutions exist. Figure 1. shows a diagram of the interaction between the real world, the mathematical model and the numerical approximation.

Figure 1. Numerical approximation.

All discretization techniques convert a given system of equations into a system of algebraic equations that can be solved by computer. Discretization introduces an error, which is connected with number of the grid points. Due to the fact, that to solve the Navier-Stokes equations we need approximations of derivatives in both space and time, each of the standard numerical approaches applied to it requires the very large number of grid points. This number depends on the dimensionality of the problem, its geometric complexity and the severity of the gradients of dependent variables. For example: the model of the pressure contour of the plane of symmetry of a car needs about 50,000 vertices in space [13] while when a flow over a periodic array of cylinders was calculated 40,000 vertices were needed [13]. Moreover, at each grid point, each dependent vari-

able and a number of auxiliary variables must be stored in the computer memory. Therefore it is easy to see that the computational requirements for the high-resolution solution of Navier-Stokes equations are: very fast computers (capable of delivering Teraflop performance) with very large memories (of order of 20 Gigabytes or more). In addition, to be able to store the intermediate results (for instance for the purpose of visualization), a very large I/O bandwidth is required as well as extremely large data storage capability (of order of thousands Terabytes). Let us look into the history and the current state of the art of high-performance computers.

PARALLEL COMPUTER BASICS

Hardware

The technological development of computers during the last thirty years produced machines faster by many orders of magnitude and substantially increased the size of available memory. The late 1970s and the 1980s saw the emergence of machines with vector-processing units and, later, multiple execution pipelines inside of a single processor. Currently, the most powerful single-processor (Cray T-9x series vector processor) is capable of delivering approximately 2 Gigaflips of sustained performance on dense matrix applications. Since its clock is running at 1.5 nano-seconds it can be expected that we are approaching the limits of what can be achieved on a single processor. It is thus clear that only multiprocessor computers will be able to deliver enough computation power to solve the equations of fluid dynamics.

The development of parallel computers can be traced to the early 1960s when Illiac IV was constructed at the University of Illinois. Since then the technology evolved considerably and became mainstream. Nowadays, it is quite typical to buy a PC with multiple (usually up to four) processing units (especially when such a PC is used as a network server). In addition, each processor has multiple execution path for both integer and floating point arithmetic. These changes occurred also on the cutting edge of parallel computer architecture. Below we discuss three important areas of change: model of computer memory arrangements and changes among the parallel computer manufacturers (for more information see [9,10]).

The computational model. Initially two basic types of architectures have been tried: single instruction multiple data (SIMD) and multiple instruction multiple data (MIMD). In the SIMD model a very large number of processors (usually relatively weak -- even one-bit processors) works synchronously. When the program is executed, each processor performs the same operation on a different data element (or no operation at all). Typically such machines consisted
Programming Environments

The software support for parallel computing also has evolved over time. The libraries like PVM (Parallel Virtual Machine) and MPI (Message Passing Interface) have been developed for distributed computing. These libraries enable developers to write parallel programs that can run on distributed systems. For example, the PVM library includes a set of functions and utilities for communication between processes running on different computers. Similarly, MPI provides a standard interface for message passing between processes. These libraries have been widely adopted by the scientific community for their ease of use and effectiveness in parallel programming.

Another trend in parallel computing is the use of hardware acceleration. This involves using special-purpose hardware to perform parallel computations more efficiently. For instance, GPUs (graphics processing units) are now frequently used in scientific computing due to their high computational power and efficiency in parallel processing.

The increasing interest in developing scientific computing-based applications has driven the development of new programming environments and software libraries. These tools are designed to make it easier for scientists and researchers to harness the power of parallel computing for their work.
Practical flow problems often require the calculation of the computational solution in complicated three-dimensional domains involving thousands of nodal unknowns. For such large-scale problems, the assessment of the relative efficiency of competing numerical schemes is an essential pre-requisite to the main body of the computational investigation. It is important to observe that the efficiency of the numerical method and the overall performance of the computational system are both very dependent on the choice of the parallel implementation. The original method developed by the Navier-Stokes equations has been extended to include the effects of the S-N e.g. Navier-Stokes equations are often used as a first approximation for many practical problems. When the fluid flow is large-scale and there are significant changes in the flow, the FEM method is used, whereas the Navier-Stokes equations are used to represent the fluid flow. Many codes used in the aerospace industry for solution of the Navier-Stokes equations rely on finite elements to solve Navier-Stokes equations. This data structure is suitable for parallel computations, because the boundary methods lead to the solution of sparse systems, which are well suited for large-scale problems.

STRESZCZENIE

**SUMMARY**

Nowe algorytmy numeryczne oraz programowanie równoległe są narzędziami umożliwiającymi rozwiązania złożonych obliczeniowo zagadnień mechaniki pływów. Przegląd ostatnich osiągnięć w obu dziedzinach jak i nadal istniejących problemów został przedstawiony w artykule.

New numerical algorithms and parallel computing provide tools for studying previously unsolvable problems in fluid dynamics. A summary of recent development in both areas and the perspective on the future research is presented.