Parallel Computation of Industrial Flows on the Cray T3D

Olivier Byrde, *David Cobut* and *Mark L. Sawley*, Institut de Machines Hydrauliques et de Mécanique des Fluides, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

ABSTRACT: *Numerical flow simulation for industrial applications often necessitates considerable computational resources available only on high-performance parallel computer systems. The present study, undertaken within the framework of the Cray Research–EPFL Parallel Application Technology Program, has involved the porting of a 3D incompressible Navier-Stokes solver onto the 256-processor Cray T3D system installed at the EPFL. The parallelization technique employed and performance results obtained are discussed in the present paper. Large-scale numerical simulations of automotive aerodynamics and flow in hydraulic turbines and chemical mixers are presented.*

Introduction

Not so long ago, massively parallel processors (MPP) were considered as exotic systems since there was no clear indication that such machines could be used to solve real-world problems. In addition, vector computers were sufficient for the existing problem sizes. With the development of advanced MPP systems, such as the Cray T3D, and the need to perform much larger numerical simulations related to real-world problems, the situation is now significantly different. The present study will show that MPP systems can be effectively used to solve a variety of real-world problems, and that new breakthroughs in large-scale applications of industrial interest – as for example in computational fluid dynamics – will be possible only with the extensive use of such systems.

Flow solver

The parallel code used in this study is based on a multi-block code developed within IMHEF-EPFL for the numerical simulation of unsteady, turbulent, incompressible flows. This code, originally designed for vector supercomputers like the Cray Y-MP, solves the Reynolds-averaged Navier-Stokes equations on 2D and 3D structured and block-structured computational meshes.

Numerical scheme

The numerical method employs a cell-centered finite volume discretization with an artificial compressibility method to couple the pressure and velocity fields. A second-order upwind spatial discretization scheme based on the approximate Riemann solver of Roe is employed for the advection terms, while the diffusion terms are discretized using a central approximation. The time integration of the unsteady Navier-Stokes equations is performed using an implicit two-stage Runge-Kutta scheme. At each time step, a non-linear system resembling the equations for stationary flow is solved using an ADI method.

Parallelization

To date, only a subset of the original code has been parallelized; the parallel code can be used to compute steady, laminar, 3D incompressible flows. Two-dimensional flows are computed by replication in the degenerate third dimension. Parallelism is achieved by dividing the computational domain into a number of blocks (sub-domains), with the flow equations being resolved in all blocks in parallel by assigning one block to each processor. Communication between processors is necessary to exchange data at the edges of neighbouring blocks. The communication overhead is minimized by data localization using two layers of “ghost cells” surrounding each block. Data are exchanged between blocks using message passing via the PVM library, since a preliminary study [5] has shown that this provides a high level of both performance and portability.

Performance analysis

Studies have been undertaken to investigate the performance of the parallel code on the Cray T3D system. The present code uses an implicit numerical scheme to resolve the flow equations within each block, combined with an explicit updating of the boundary values. It is therefore relevant to question if, due to convergence degradation, such a scheme is scalable to the large number of processors available on the Cray T3D.

As a test case, inviscid flow between the blades of the "Durham low speed turbine cascade" [4] has been considered. Computations have been performed for 3D flows using a mesh containing 120x52x64 cells.

Figure 1 presents the convergence history as a function of the number of blocks employed. These results show that the convergence degradation for the present test case is not significant, resulting in a nearly linear speedup with increasing number of processors. The observation that the slowest convergence is obtained using 16 blocks is presumably due to the fact that the convergence rate is determined not only by the number of blocks, but also by the transient solutions within each block.

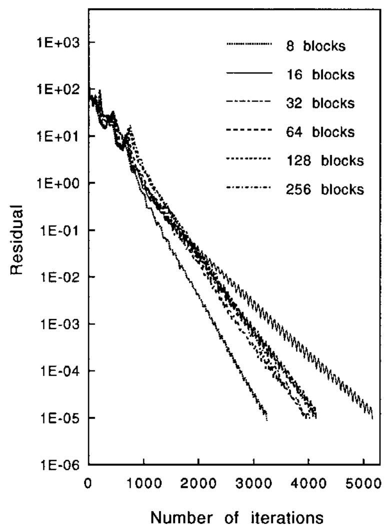

Figure 1: Convergence history.

The time to solution (i.e. the wall clock time required to obtain a residual of 10^{-5}) and the relative importance of the connectivity overhead (i.e. the time spent in communication and synchronization between blocks relative to the total time) are presented in Figure 2. These results show that due to the very fast communication network of the Cray T3D, approximately linear speedup is obtained even when a large number of processors is employed. Such a conclusion can not always be drawn: 2D computations using a cluster of up to 24 workstations [2] have shown that the poor performance of an Ethernet interconnection network can result in a greater time to solution as the number of processors is increased!

Pre- and post-processing

It is well recognized that for the numerical simulation of complex 3D industrial flows that involve an enormous quantity of data, the pre- and post-processing phases of the simulation procedure can necessitate a time (for data file manipulation and processing, I/O, etc.) often substantially longer than the cpu time required by the flow solver. This potential problem is further

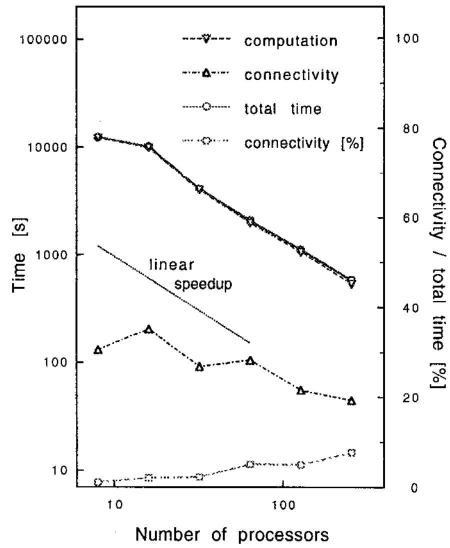

Figure 2: Time to solution on the Cray T3D.

exacerbated if the flow solution is obtained using an efficient solver on a high-performance parallel system. For this reason, it is essential to incorporate the pre- and post-processing phases into the parallel environment

Mesh generation

Block structured meshes, while being structured within each block, are generally irregular at the block level. The task of generating the block boundaries is therefore equivalent to determining an appropriate unstructured mesh. While the block boundary mesh is currently determined manually, it is desirable that its construction be performed using "automatic" unstructured hexahedral mesh techniques; such an approach is currently not possible.

Once the block structure is determined, the mesh generation within each block is undertaken in parallel, with the use of communication between processors to perform any necessary mesh smoothing across block boundaries. The use of parallel mesh generation – even in this limited form – has been shown to provide a substantial reduction in mesh generation time [3]. More importantly, since only the block boundary information is imported into the parallel system a significant reduction in I/O time is also obtained.

Visualization

Due to the enormous quantity of data generated by an unsteady 3D flow solver each time step, it is not practical to store these data on disk for later "off-line" processing and visualization. To overcome this problem, TPview, an "on-line" visualization tool, has been developed that uses the interprocess communication (IPC) library to interface to the commercial TECPLOT visualization package. The TPview software has been described in a separate presentation at the Barcelona CUG conference and is the subject of another article in the present proceedings [6].

Industrial applications

The parallel flow solver described above has been employed to compute a number of different large-scale 3D flows. Three specific examples, of interest to the turbomachinery, automotive and process industries, are briefly presented in the following sections.

Francis water turbine

Water turbines of the Francis type are widely used for hydro-electric power generation. The numerical simulation of the flow in a Francis turbine is conventionally undertaken by computing the flow separately in each component of the turbine: spiral casing; distributor; runner and draft tube. The coupling of the flow in these components is complicated both by the rotation of the runner relative to the other components and by the different periodicities of the distributor (24 blades for the present case) and the runner (13 blades).

Most numerical simulations of the flow in the runner assume that the flow is periodic; this allows only one inter-blade channel to be computed rather than the whole runner. Using the computational power of the Cray T3D, the flow in the entire runner can be computed in a reasonable time, allowing an initial assessment of the non-periodic flow behaviour due to the coupling between the runner and the distributor. Figure 3 presents the results of a simulation using 13 blocks of identical size (with a total of 104,832 cells) computed in 3.3 hours on 13 processors of the Cray T3D. For this computation, the periodicity of the flow at the exit of the distributor has been imposed as an input condition for the runner, resulting in non-periodic flow in the runner. The results of this simulation have shown that, for optimal operating conditions, the initial periodicity is quickly “lost” within the runner, with the flow into the diffuser section being essentially axisymmetric.

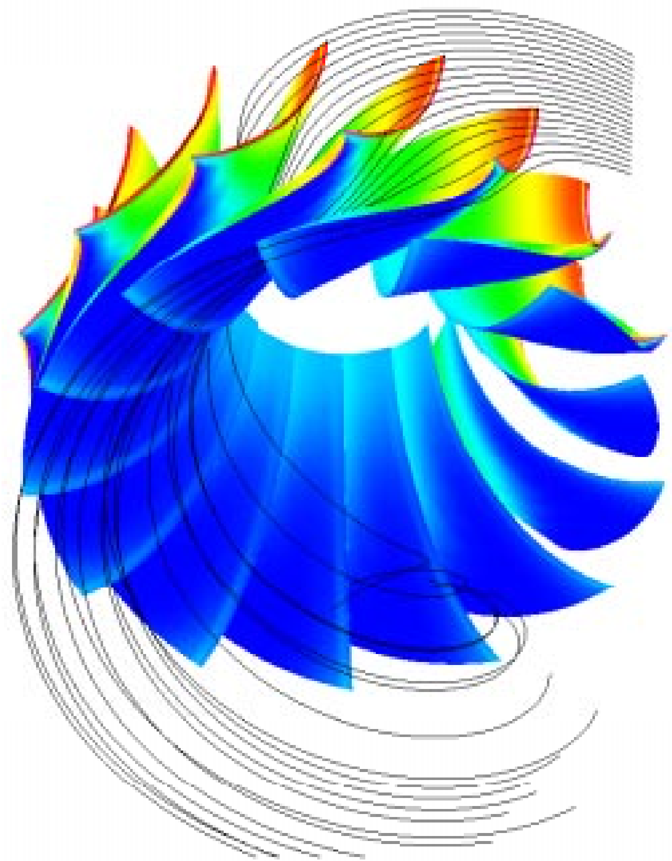

Figure 3: Surface pressure and streamlines for flow in the runner of a Francis turbine.

Formula 1 racing car

Due to the complex geometry and flowfield, it is currently not possible to simulate the flow around an entire Formula 1 racing car. The aerodynamic properties of the car are however principally determined by certain critical regions, such as the air inlet, undertray, wheels and front and rear wings. While the flow in these regions is coupled, it is nevertheless useful to study each region separately to obtain a detailed understanding of the flow. The present study has concentrated on a rear wing consisting of a series of multi-element airfoils (see Fig. 4). Four different configurations have been computed, corresponding to the three angles of the central element (denoted by c1-c3) plus the removal of this element (denoted by c0).

The primary role of the rear wing is to produce downforce. Although significant drag also results, this is smaller than that produced by, for example, the wheels. Assuming that the flow is attached on the rear surface of each of the wing elements, inviscid computations should provide reasonable estimates of the downforce. It is also noted that even at maximum car speeds (up to 330 km/h), the flow can be considered to be incompressible. The numerical flow simulations performed to date have assumed zero angle of yaw; symmetry along the central vertical plane enables only half of the rear wing to be considered.

Three-dimensional flow computations have been undertaken using a mesh comprised of 121 blocks with a total of 3.8 million cells being employed. The flow solution obtained for one of the rear wing configurations is presented in Figure 5.

These results clearly show the presence of outboard trailing vortices near the top of the side plates, as is observed on the race track. Computation of the pressure coefficient has allowed the contribution to the total wing downforce due to each element to be determined. The computed lift coefficients (normalized to the planform area of the largest element) are presented in Figure 6 for each of the four configurations considered.

Figure 6 shows that a modification of the central element results in a change of the lift coefficient for each of the other elements, indicating that the flow around each component element is strongly coupled. For each configuration, approximately half of the total downforce is provided by the largest element.

The above-mentioned computations required approximately 4 hours on 121 processors of the Cray T3D. It should be noted that since there is not the same number of mesh cells in each block, load imbalance occurs, as shown in Figure 7. This problem can be alleviated by computing more than one block per processor, which is possible using PVM on a parallel computer

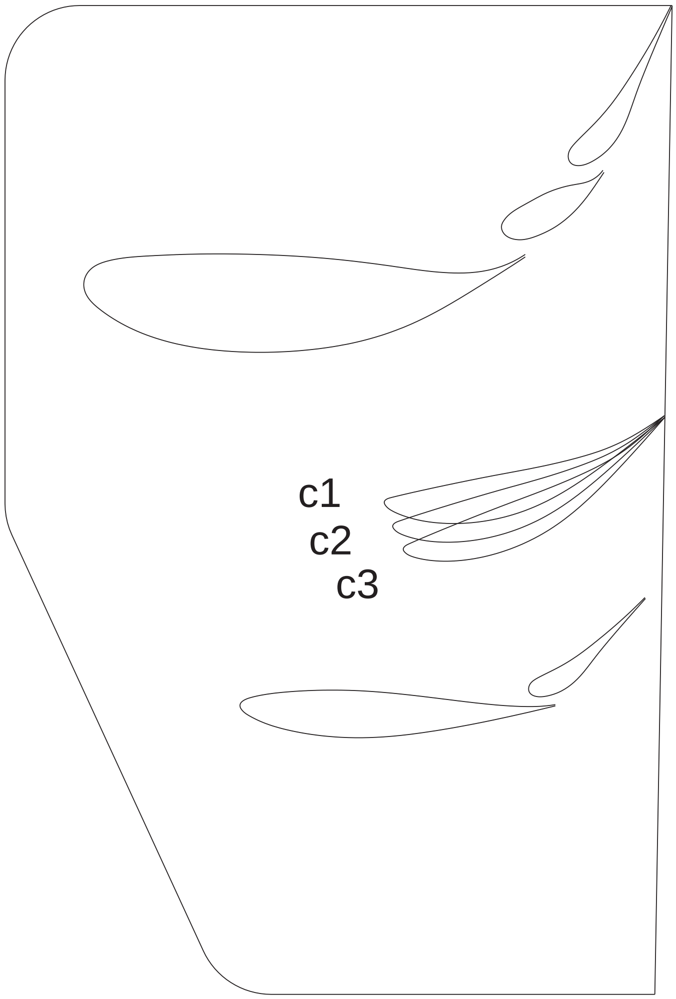

Figure 4: Multi-element rear wing.

system that allows multiple processes to run on each physical processor. Unfortunately the Cray T3D does not permit multiple processes.

Kenics static mixer

Static mixers are in-line mixing devices which consist of mixing elements inserted in a length of pipe. Mixers of this type

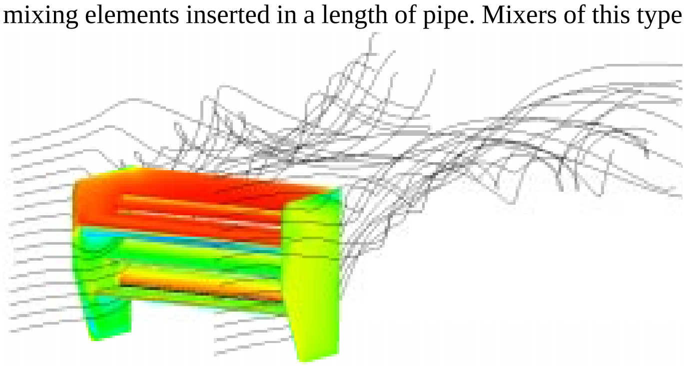

Figure 5: Surface pressure and streamlines for flow around the rear wing (configuration c2).

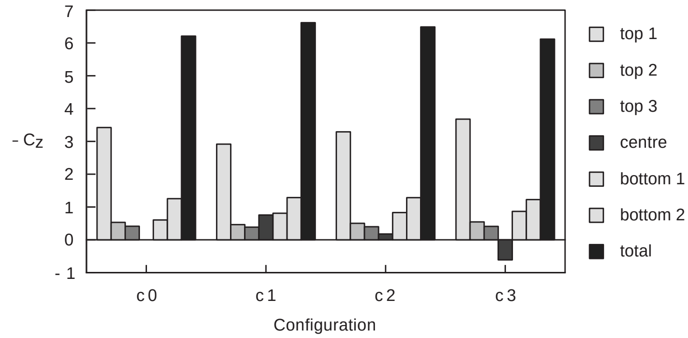

Figure 6: Component and total lift coefficient for the four different configurations considered.

are used in continuous operation, with the energy for mixing being derived from the pressure loss incurred as the process fluids flow through the mixing elements. A variety of element designs are available from the various manufacturers, with the number of elements required for a particular application being dependent on the difficulty of the mixing task (more elements are necessary for difficult tasks). At the present time, many installations are optimized by trial and error with much depending on previous experience and wide safety margins. In

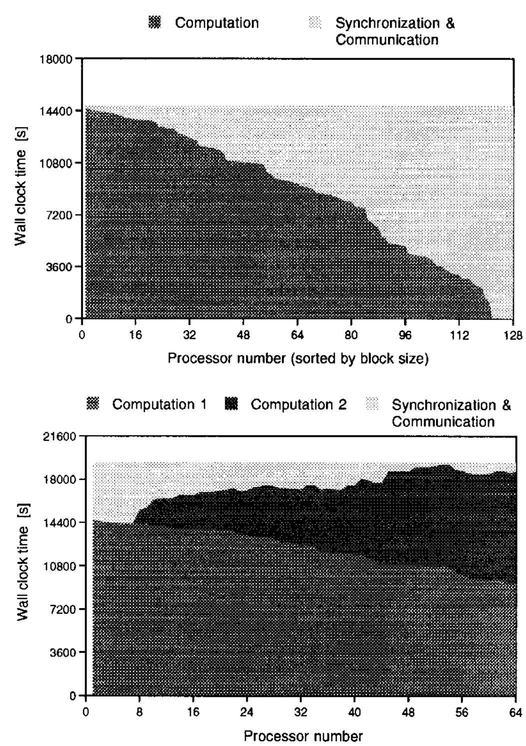

Figure 7: Wall clock time for 3D flow simulations computed with one block per processor (top) and estimated for two blocks per processor (bottom).

this context, the ability to simulate the flow through a particular mixer can greatly help the understanding of the mixing process and provide for better, faster and cheaper design optimization.

The geometry considered here corresponds to the Kenics static mixer manufactured by Chemineer Inc. [1]. Its design consists of a series of mixing elements aligned at 90°, each element being a short helix of one and a half tube diameters in length. Each helix has a twist of 180° with right-hand and left-hand elements being arranged alternately in the tube, as illustrated in Figure 8.

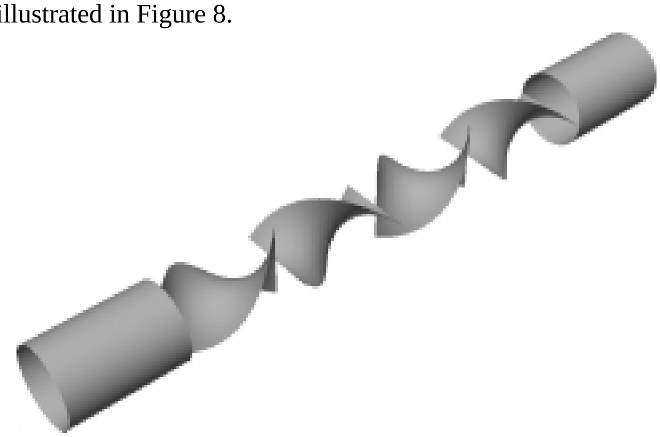

Figure 8: Perspective view of a 4-element Kenics static mixer.

Four standard configurations have been considered for the Kenics mixer, consisting of 4, 6, 8 and 10 mixing elements. A fifth configuration, where all elements are of the same kind (right- or left-handed) has also been investigated in order to illustrate the importance of flow reversal in the mixing process.

For symmetry reasons, only half elements are used for the computations. Each half-element is made of four blocks of identical size: two in the core and two in the outer edge. Three meshes have been considered in this study. The “coarse” mesh contains 9216 cells per half element, the “medium” mesh 28,224 and the “fine” mesh 112,896. For a 6-element mixer (i.e. 6 mixing elements plus inlet and outlet tubes), this corresponds to a 32-block mesh with a total number of cells of 55,296, 225,792 and 903,168, respectively. Cross-sections of both medium and fine meshes are shown in Figure 9. It is noted that the coarse mesh is of similar resolution to that currently used by industry.

Computations have been performed on the Cray T3D for each of the meshes mentioned above and for Reynolds numbers comprised between 10 and 400. Results presented here were obtained for a 4-element mixer using the medium mesh (with 24 blocks), these results being essentially the same as those computed using the fine mesh. On the contrary, differences observed with the results obtained using the coarse mesh indicate that this mesh may not provide sufficient resolution of the flow within the mixer.

At the inlet of the pipe, uniform flow is imposed; an approximately parabolic profile of the streamwise velocity develops by the beginning of the first mixing element. The first element

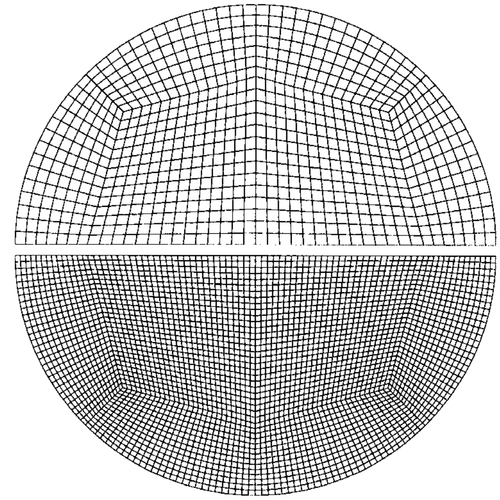

Figure 9: Cross-section of the computational domain showing the medium and fine meshes.

divides the flow into two parts, each having its separate core region of maximum velocity. Subsequent rotation, stretching and folding of this core region produce a series of layers of fluid with different streamwise velocity. In the presence of highly viscous fluids, these layers tend to disappear and a new high-velocity core region develops. The same process is then repeated in the next mixing element. It has been observed that, after the first element, the velocity profile is almost identical in all elements, resulting in a periodic flow pattern.

To gain more insight into the mixing process, the passage of fluid particles through the mixer has been traced. Figure 10 shows the distribution of an initially circular set of particles for solutions computed with Reynolds number Re=10 (high viscosity) and Re=100 (moderate viscosity). This demonstrates the importance of the fluid viscosity in the stretching and folding process and in the global mixing quality.

The convergence of the solver – and thus the time to solution – depends strongly on the Reynolds number, as is shown in Figure 11. This defines a region $10 < Re < 400$ where a solution can presently be obtained. The upper bound has its origin in the physics (flow becomes unsteady) and cannot be avoided, whereas the lower bound is due to the numerical method employed. A new solver is being implemented in the code which should allow the computation of very viscous flows with $Re < 10$.

Conclusion

The present study has shown that an existing 3D multi-block flow solver can be adapted to employ the enhanced capabilities of high-performance parallel computer systems. The use of the PVM message passing library has been seen to provide low communication overhead on the Cray T3D, resulting in good code performance.

The applications presented here show that parallel computation can effectively be used not only for research or academic

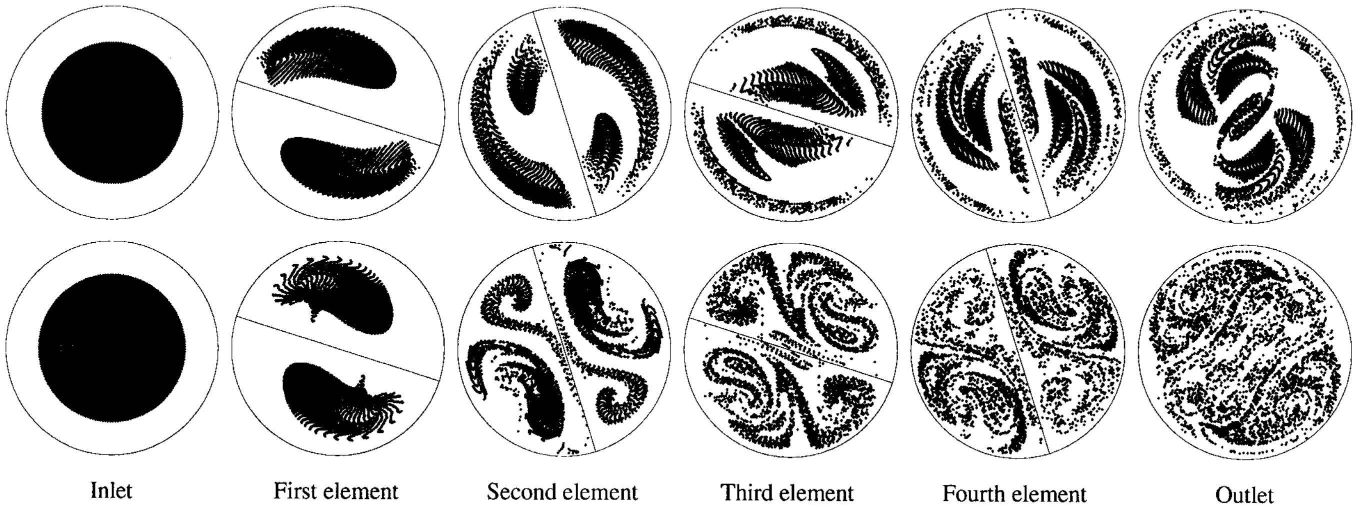

Figure 10: Particles at different locations in a four-element mixer for flows computed with *Re*=10 (top) and *Re*=100 (bottom).

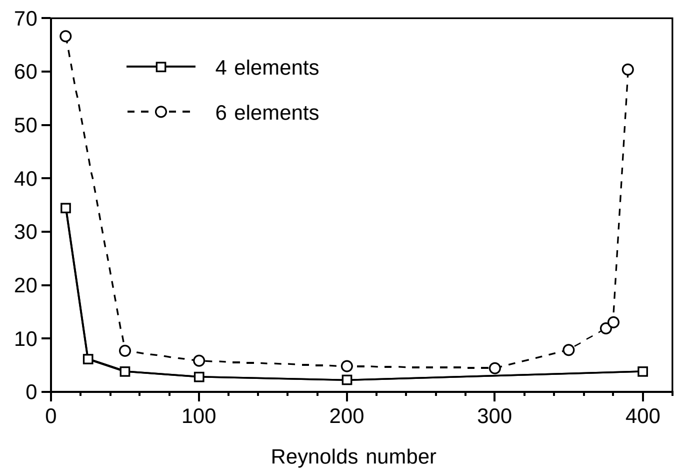

Figure 11: Total computation time as a function of the number of mixing elements and Reynolds number.

purposes, but also for real-world applications of interest to (but not limited to) the turbomachinery, automotive and process industries.

The use of the Cray T3D has enabled flow computations to be performed that are inaccessible to mono-processor computer systems. While workstation clusters and shared-memory computer systems can be employed for small- and medium-size simulations, the present study has demonstrated that large-scale applications can only be performed on massively parallel systems with the necessary memory requirements and computational power.

Acknowledgements

The geometry of the rear wing of the Formula 1 racing car was provided by PP Sauber AG. Financial support for the present study was provided by the joint Cray Research–EPFL Parallel Application Technology Program.

On-line resources

A series of abstracts (in HTML format) and the corresponding full-length articles (in PostScript format) on the subjects discussed in the present paper can be found on the EPFL–IMHEF World Wide Web server at the following URL:

http://imhefwww.epfl.ch/lmf/publications

References

[1] Bakker, A., LaRoche, R. and Marshall, E.M., 1995, “Laminar flow in static mixers with helical elements”, private communication.

[2] Byrde, O., Cobut, D., Reymond, J.-D., and Sawley, M.L., 1995, “Parallel multi-block computation of incompressible flow for industrial applications”, in “Parallel computational fluid dynamics: Implementations and results using parallel computers”, Proceedings of Parallel CFD '95 (Pasadena), North-Holland, Amsterdam, in press.

[3] Cobut, D., 1995, “Application du calcul scientifique parallèle à l'étude de l'écoulement autour de l'aileron arrière d'une voiture de Formule 1”, IMHEF Report D–95–10.

[4] Gregory-Smith, D.G., 1994, “Test case 3: Durham low speed turbine cascade”, ERCOFTAC seminar and workshop on 3D turbomachinery flow prediction II (Val d’Isère) Part III, pp. 96-109.

[5] Sawley, M.L., and Tegnér, J.K., 1995, “A comparison of parallel programming models for multi-block flow computations”, Journal of Computational Physics, 122, 280–290.

[6] Williams, S.A., 1996, “On-line visualization for scientific applications on the Cray T3D”, published in the present proceedings.