Mecánica Computacional Vol. XXIV
A. Larreteguy (Editor)
Buenos Aires, Argentina, Noviembre 2005

PARALLEL SIMULATIONS OF FLOW AROUND HIGH-ASPECT RATIO CYLINDERS EMPLOYING MPI

Ricardo B. Flatschart[*†], Julio R. Meneghini[†], Fábio Saltara[†], Alessandro Lima[†] and Rafael Gioria[†]

[*] Present address: Embraer S/A, São José dos Campos, SP, Brazil
[†]University of São Paulo, NDF – EPUSP, Dept. of Mechanical Engineering, São Paulo, SP, Brazil
email: jmeneg@usp.br

Key words: vortex-induced vibration; flow interference; bluff body flow, marine risers.

Abstract. In this paper, the dynamic response of a long circular cylinder due to vortex shedding is numerically investigated. The cylinder with high aspect ratio has the structural properties typical of marine risers. The cylinder is divided in two-dimensional sections along the riser length. The *Discrete Vortex Method* (DVM) is employed for the assessment of the hydrodynamic forces acting on these two-dimensional sections. The hydrodynamic sections are solved independently, and the coupling among the sections is taken into account by the solution of the structure in the time domain by the *Finite Element Method*. Parallel processing is employed to improve the performance of the method. The simulations are carried out in a cluster of *Pentium IV* computers running the Linux operating system. A master-slave approach via MPI – *Message Passing Interface* – is used to exploit the parallelism of the present code. The riser sections are equally divided among the nodes of the cluster. Each node solves the hydrodynamic sections assigned to it. The forces acting on the sections are then passed to the master processor, which is responsible for the calculation of the displacement of the whole structure. Scalability of the algorithm is shown and discussed.

1 INTRODUCTION

Vortex Induced Vibrations – VIV – has been a substantial challenge in the field of Ocean Engineering. The onset of cyclic forces due to vortex shedding on marine structures, such as risers employed in the petroleum exploration industry, can cause fatigue damage and collapse of these structures. For a better understanding of the phenomena involved, a good description of the complex flow field developed around the structures is of great importance. *Computational Fluid Dynamics* – CFD – is a tool of growing significance in the design phase of these structures. Several CFD methods, such as the *Finite Volume Method* and the *Finite Element Method*, have been used for this purpose. These methods are based on the solution of the partial differential equations that describe the flow field, the well-known Navier-Stokes equations. The discretization of the equations is carried out over a mesh. This can be very troublesome in terms of computational efficiency, memory requirements and complexity of the numerical method. These shortcomings lead to the development of lagrangean methods that do not rely on the use of a mesh. One of these methods of particular interest is the *Discrete Vortex Method* – DVM [4,5]. This method is based on the surface vorticity boundary integral approach for potential flow analysis.

The computation of the flow field around the structures could be very demanding in terms of computational resources. To overcome this problem, parallel processing is often employed. The main idea of parallel processing is to divide the workload among a certain number of processors. Parallel processing is accomplished by means of compiler directives or calls to sets of library routines. The most common standards used for parallel processing are MPI [2] and OpenMP [3]. OpenMP is used in shared memory architectures, often referred to as Symmetric Multi Processing (SMP) machines. MPI, on the other hand, can be used in both shared and distributed memory systems.

The main focus of this work is the parallelization of a computer code used in simulations of marine risers, developed at the University of São Paulo. The overall performance of the parallel code is compared with the original code. A speed up curve is obtained to demonstrate the scalability of the parallel version. The simulations are carried out in a cluster of *Pentium IV* computers running the Linux operating system. The structural and hydrodynamic models employed are briefly explained. A Finite Element structural model based on the Euler-Bernoulli beam theory [6] is used. The dynamic response of the riser is evaluated by solving a general equation of motion in the time domain. The riser is divided in two-dimensional sections along its length. The hydrodynamic forces are evaluated in these two-dimensional strips by the DVM. Viscous effects are modeled through a growing vortex core method [4]. In this way, a quasi three-dimensional analysis is achieved. A similar approach was used by Graham and Willden [12], using a mixed eulerian-lagrangian vortex method, by Fregonesi *et al.* [13] and Yeung *et al.* [14] employing a lagragian approach. A complete review of vortex methods can be found in Sarpkaya [15]. In the present method, the hydrodynamic forces are assessed through two-dimensional sections by the DVM. In this way, the three-dimensional characteristics of the flow around the riser are neglected. This is a limitation of the method employed in this work. However, according to Graham and Willden [12], assuming that the major component of the wake vorticity is still aligned with the cylinder, and spanwise gradients of all flow variables are assumed to be much less than gradients in the other directions, a two-dimensional simulation for the hydrodynamic part, as a first approach, is expected to provide reasonable results. A hypothetical vertical marine riser of 220 meters immersed in a uniform current is analyzed. Parallelization of the computational code is carried out via MPI. A master-slave approach is employed to exploit the parallel characteristics of the numerical scheme.

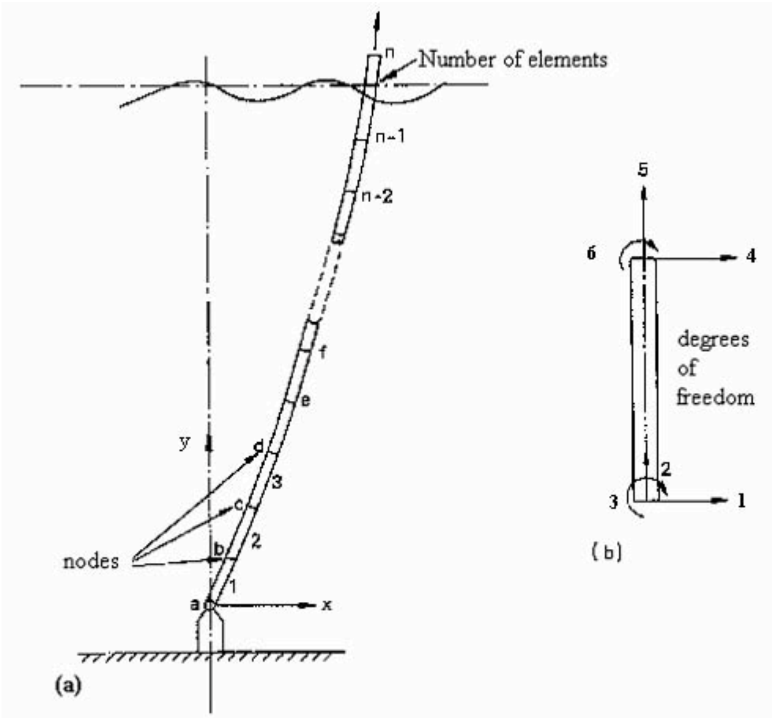

Figure 1. Configuration for the tandem arrangement.

2 STRUCTURAL AND HYDRODYNAMIC MODELLING

A finite element structural model based on the Euler-Bernoulli beam theory is employed to calculate the dynamic response of the cylinder. A general equation of motion is solved through a numerical integration scheme in the time domain. This numerical structural model is based on the studies of Patel and Witz [7] and Ferrari Jr. [6]. Firstly, we find a static solution for the riser. Then, in the dynamic analysis, the stiffness matrix obtained from the static analysis is used as an average approximation. A lumped approach is employed. A mass lumped matrix is constructed and the damping matrix is evaluated in a global manner. More details can be seen in the references mentioned above. The *Discrete Vortex Method* (DVM) is a Lagrangian numerical scheme technique for simulating two-dimensional, incompressible and viscous fluid flow. The method employs the stream function-based boundary integral method and incorporates the growing core size or core spread method in order to model the diffusion of vorticity. In the DVM the body is discretized in N_w panels, and N_w discrete vortices with circulation Γ_i are created from a certain distance of the body, one for each panel. These vortices are convected and their velocities are assessed through the sum of the free stream velocity and the induced velocity from the other vortices. The induced velocities are calculated through the Biot-Savart law. Forces on the body are calculated integrating the pressures and viscous stresses. Viscous stresses are evaluated from the velocities in the near-wall region, and the pressure distribution is calculated relating the vorticity flux on the wall to the generation of circulation. Details about the employed method can be found in Yamamoto *et al.* [16] .

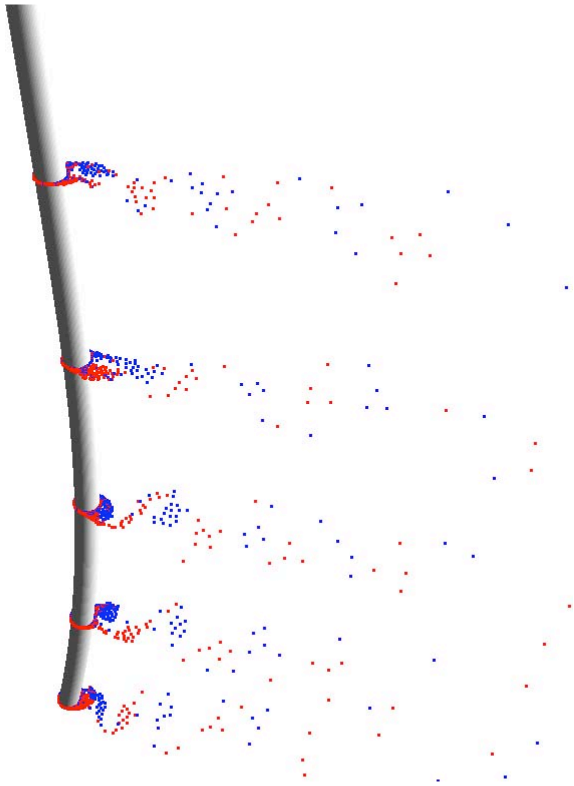

Figure 2. Cylinder and elastic base mounted on the channel test section structure.

3 PARALLEL PROCESSING

As one can see from the previous section, the hydrodynamic sections used in the DVM are solved independently one from another. There is no external information necessary to evaluate the hydrodynamic forces acting on the structure, except those given by the structure itself. This is the reason for the present method can be regarded as a quasi three-dimensional method. This characteristic of the method can be efficiently explored for parallel processing. Since the hydrodynamic sections are solved independently, one can divide the computational task of the DVM among a given number of processors. The force coefficients obtained from the sections are passed to a master processor, which is responsible for solving for the new displacement of the structure by the FEM. The new positions are then passed back to the other processors, and the cycle restarts.

The present code uses the Message-Passing paradigm for the parallelization of the computational code, in a master-slave approach. The code runs in a cluster of *Pentium* class computers. Details of the cluster are: 64 nodes: Pentium *IV* 2.4 GHz processors, 512MB memory, gigabit Ethernet; 1 server: Dual Xeon 2.4 GHz, 2GB memory, 2 gigabit ethernet,; RAID array with 7 SCSI hard disks of 72GB each; Operating system: Red Hat Linux 8.0 with kernel 2.4.20; Lam-MPI version 7.0; Intel Fortran Compiler 7.1; Gnu gcc version 3.2-7. Communication among the processors is done via *broadcast* and *scatter*/*gather* operations. A graphical explanation of these operations is given in Figures 3 and 4. In the broadcast operation, data in one of the processors – usually the master processor – is passed to all the others. This operation is employed for variables that all the processors must know to correctly execute the code. The scatter and gather operations are used to distribute and collect information to and from the other processes, respectively. The scatter operation is useful, as an example, for passing the new positions of the structure from the master processor to the other nodes of the cluster. Information about the hydrodynamic forces, necessary to solve the dynamic problem via FEM can be passed to the master processor through a gather operation.

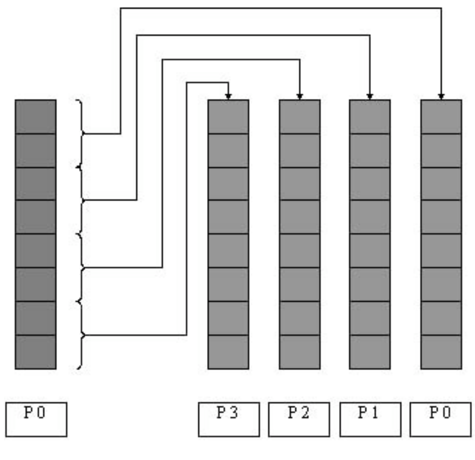

Figure 3. Broadcast operation.

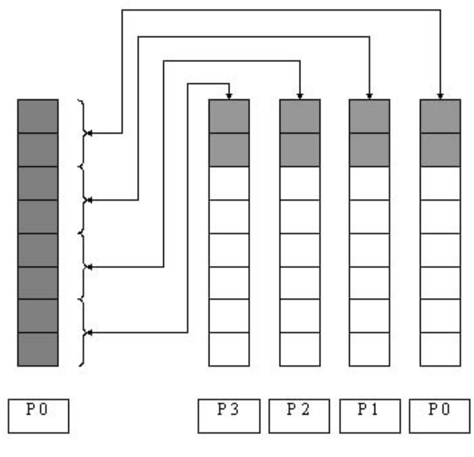

Figure 4. Scatter/Gather operations.

To ensure a balanced workload among the processors, the riser sections are equally divided among the nodes of the cluster, as one can see in Figure 5. Each node solves the hydrodynamic sections assigned to it. It must be noted that, with exception on the master node, all the nodes solves exactly the same problem, but with a different data set. This kind of computing paradigm is also referred as SIMD – *Single Instruction Multiple Data*. The master processor is responsible for the serial parts of the global computational code – data I/O and the solution of the structure, and also for some of the hydrodynamic sections of the riser. The slave nodes must wait for the completion of the FEM analysis by the master node to obtain the new position of the riser. This could be a potential bottleneck. In our case, the master node is also the disk server and the access gateway to the cluster. It has a different specification from the nodes, with a slightly superior performance compared to the computing nodes of the cluster. Some tests confirmed that the FEM is not the most computational intensive task of the method, and the master node can handle this overhead without great impact on the overall performance of the code. Usually, one desires to keep the amount of data to be communicated among the processors small. This is because communication latency has a heavy impact in the overall performance of a parallel code. With recent advances in computer hardware – faster network interfaces and switches – this limitation is getting lower, but still important.

In the present method, the amount of communication between the nodes is kept small, in order to guarantee small communication latency. There is no necessity of one hydrodynamic section to known information about the flow field properties of the other sections. Excluding broadcast operations necessary to initialize the solver, only information about the position of the riser sections and hydrodynamic forces are passed between the nodes and the master processor. This characteristic makes the present method very attractive for parallel processing.

4 RESULTS AND DISCUSSION

A hypothetical drilling riser is modeled in order to evaluate the scalability of the code. Geometric characteristics of the riser: Water depth: 200 *m*; Riser Length: 220 *m*; Riser pipe outside diameter: 0.533 *m*; Riser pipe inside diameter: 0.498 *m*; Linear weight in air of riser sections: 5.03 *kN*/*m*; Buoyancy upthrust: 2.66 *kN*/*m*; Riser top Tension: 1362.5 *kN*; Modulus of elasticity of riser pipe:2.07×10^9 N/m^2; Specific weight of the fluid surrounding the riser bore: 1025 kg/m^3; Specific weight of the fluid in the riser bore: 1137.75 kg/m^3; Specific weight of the riser wall material: 7850 kg/m^3. The riser is discretized into 120 elements equally spaced below the still water level, and 20 elements equally spaced above the still water level. A uniform steady current profile is imposed, with a velocity of 0.5m/s. Each hydrodynamic section is discretized with 64 panels. The simulation is carried out for 10,000 iterations, which corresponds to about 530 seconds of simulation. Figures 6 and 7 show the envelope of maximum and minimum displacements in the inline and transverse directions. The maximum amplitudes observed in the inline and transverse directions are 0.8 *m* and 0.18 *m*. For this configuration, one can observe that the third mode is the predominant vibration mode.

Figure 8 shows the time spent to complete the simulations, for different number of processors, compared to the time observed in the serial code. There is no significant difference with respect to the total time spent on the simulations between the serial code and the parallel code running in just one machine. As one can expect, the total time of the simulation seems to be approximately proportional to *1/N*, where *N* is the number of processes employed. In an ideal case, the total time of the simulation would be *1/N*. The deviation of the curve is due to communication latency between the nodes and the master process. For a better understanding of the performance of the parallel code, it is a common procedure to draw the actual speed up curve compared with a linear (ideal) speed up. This curve gives a perception about how much is the impact of communication latency for the overall performance of the method. It shows also if the code scales well with increasing the number of processors. In some cases, a parallel code can stagnate and show degradation in performance, due to heavy communication latency. The speed up curve obtained in the simulations is presented in Figure 9. As one can see, a good scalability is obtained for the number of processors tested. There is no sensible degradation of the method with increasing number of processors. The present analysis is limited to 60 nodes of the cluster, because the parallel code is designed in a way that all the nodes, with exception of the master, receive a similar amount of data to process. In this way, the code only runs in a number of nodes which is an integer multiple of the number of hydrodynamic sections.

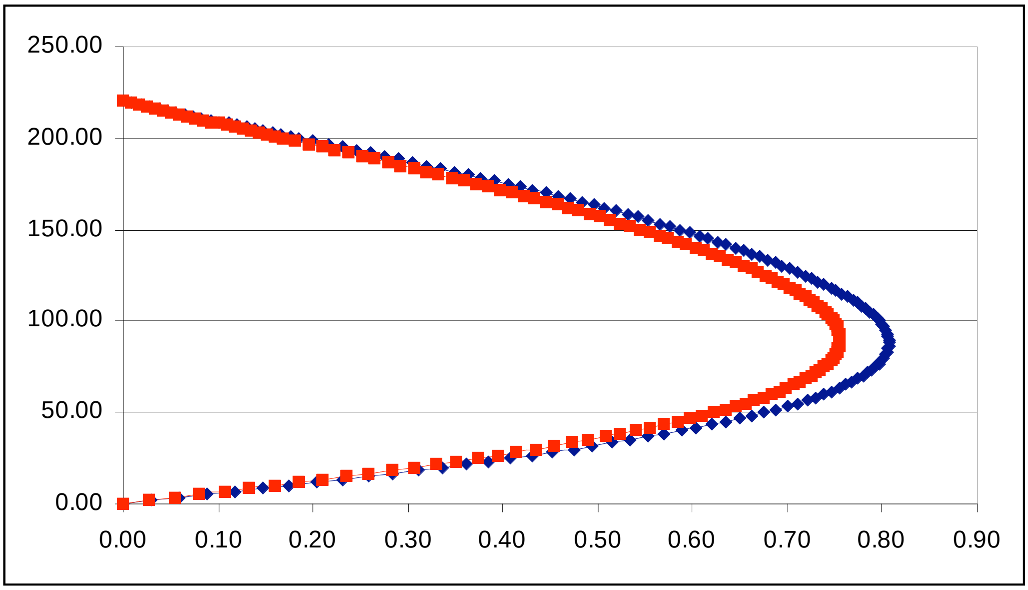

Figure 6. Envelope of inline displacements.

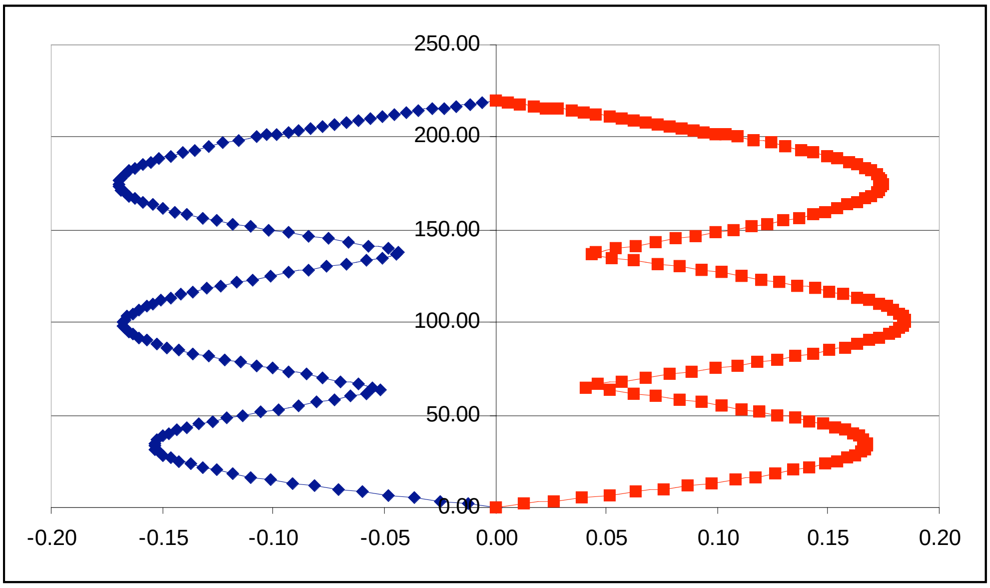

Figure 7. Envelope of transverse displacements.

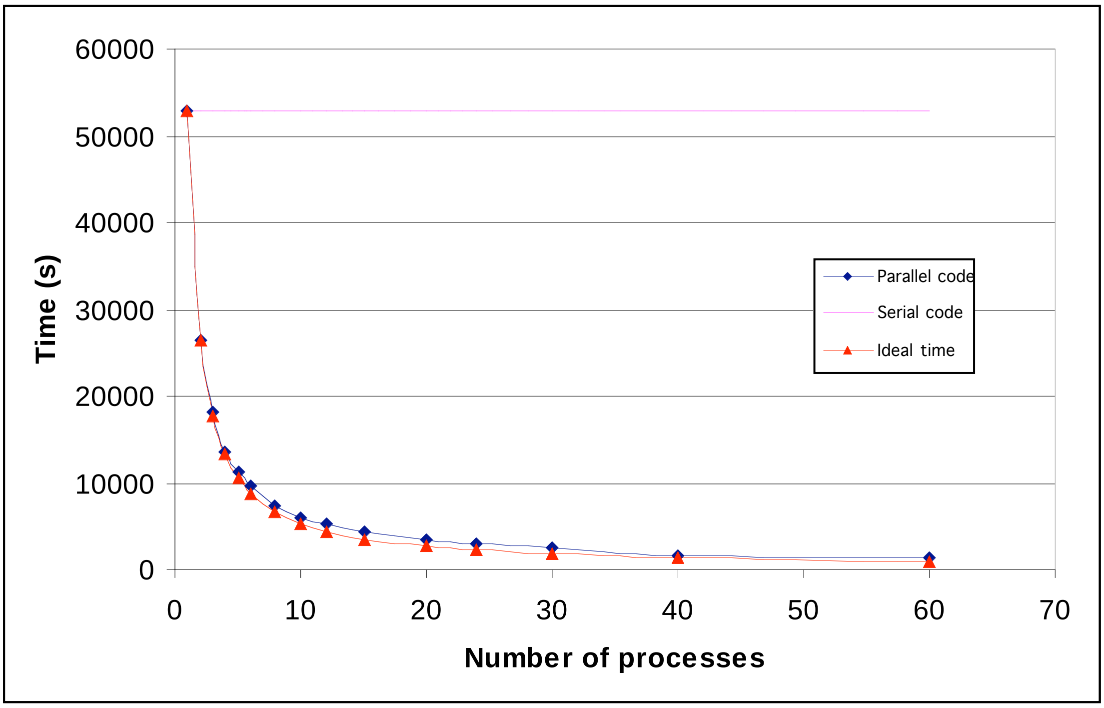

Figure 8. Comparison of computational time.

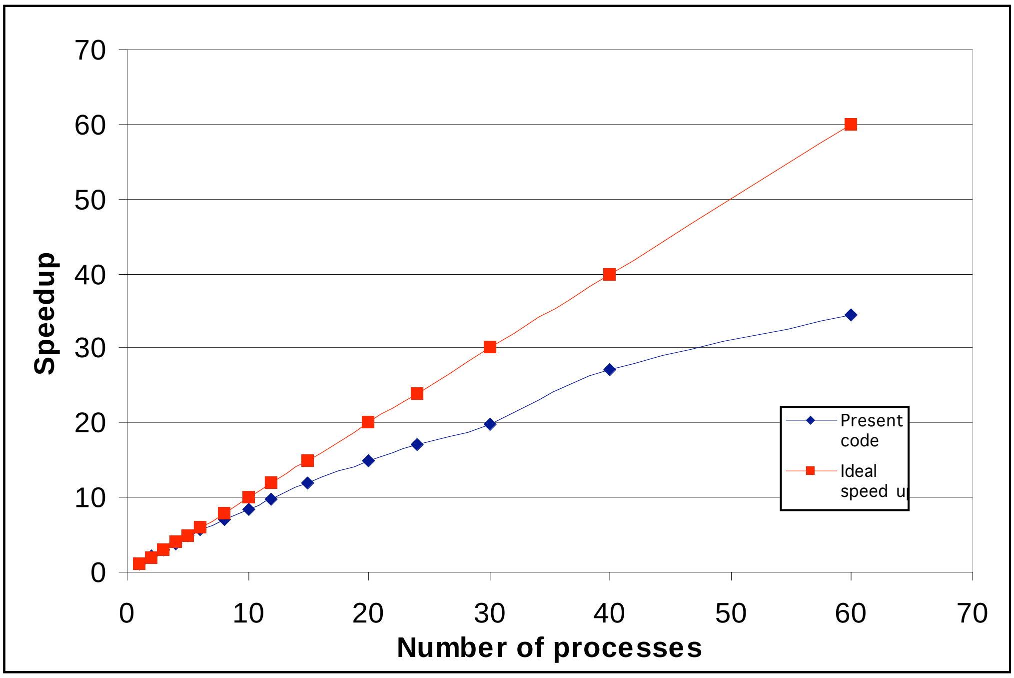

Figure 9. Speed up curve.

5 CONCLUSIONS

The results obtained shows a good speed up of the parallel code. The speed up curve shows a reasonable scalability, without stagnation in performance with increasing number of processes. However, a deviation of the ideal speed up is observed in Figure 9. This deviation is mainly caused by communication latency, but could also be caused by a higher number of subroutine calls in the parallel code. Communications between the master and slave nodes is carried out by means of scatter/gather operations. This approach offers better communication efficiency, higher portability and better code maintenance. One must bear in mind that the present method is very well suited for parallel processing, as the hydrodynamic strips could be solved independently one from another. An additional good feature for this parallel code is the small amount of data that must be passed between the master and slave processes, thus allowing small communication latency. It is important to note that the parallelization of the present code has been conceived in a modular way. The serial code can be obtained by just commenting out the calls to the parallel routines in the source code. Portability is also a matter of great importance for parallel codes. The present code can be compiled in any system with a standard Fortran compiler and MPI libraries. Extension to a larger number of sections is straightforward. The most striking conclusion that one can draw from the results, is the perspective opened by the use of parallel processing. For this case in particular, a simulation that would take about 16 hours, can be completed around 35 times faster. It should be pointed out, however, that this particular method is very well suitable for parallel processing. The amount of communication between the processors is kept low enough to prevent high latency.

ACKNOWLEDGMENTS

The authors are grateful to FINEP/CTPetro, FAPESP, CNPq and Petrobrás for providing a research grant for this investigation. The first author is also grateful to CNPq for his doctoral scholarship.

6 REFERENCES

[1] Zienkiewicz, O. C., Morgan, K.,1983, "*Finite Elements and Approximation*," Wiley, New York.

[2] Message Passing Interface Forum. MPI: A Message-Passing Interface Standard (version 1.1). Technical report, 1995. http://www.mpi-forum.org

[3] OpenMP Architecture Review Board. OpenMP Fortran Application Program Interface (version 2.0). Technical report, 2000. http://www.openmp.org

[4] Spalart, P. R., Leonard, A., Baganoff, D.,1983, "Numerical simulation of separated Flows," Ph.D. thesis, Stanford University, California (NASA).

[5] Lewis, R. I., 1991, "*Vortex Element Methods for Fluid Dynamic Analysis of Engineering Systems*", Cambridge University Press.

[6] Ferrari, J. A., 1998, "Hydrodynamic Loading and Response of Offshore Risers", Ph.D. thesis, University of London, UK.

[7] Patel, M. H., and Witz J. A., 1991, "*Compliant Offshore Structures*", Butterworth-Heinemann, England.

[8] Craig. R. R., 1981, "*Structural Dynamics – An Introduction to Computer Methods*", John Wiley & Sons.

[9] Fregonesi, R. A., 2002, "Estudo do escoamento tridimensional ao redor de um agrupamento de cilindros em tandem". MsC thesis, Universidade de São Paulo.

[10] Warburton, G.B., 1976, "*The Dynamical Behavior of Structures*", 2nd ed., Pergamon International Library.

[11] Park, W., and Higuchi, H.,1989, "Computation of Flow past Single and Multiple Bluff Bodies by a Vortex Tracing Method", University of Minnesota.

[12] Graham, J.M.R., Willden, R.H.J., 2001, "Application of mixed eulerian-lagrangian vortex methods to cross-flow past long flexible circular cylinders", *Proceedings of 2nd International Conference on Vortex Methods*, Istanbul, Turkey.

[13] Fregonesi, R.A., Meneghini, J.R., Saltara, F., Jabardo, P.J.S., Ferrari Jr., J.A., Casaprima, E., 2003, "The flow around a group of circular cylinders: a comparison between CFD calculations and experiments", *Proceedings of OMAE 2003*, Cancun, Mexico.

[14] Yeung, R. W., Sphaier, S. H., and Vaidhyanathan, M. (1993) "Unsteady Flow About Bluff Cylinders," *Int. J. Offshore and Polar Eng.*, vol. 3, no. 2, pp 81-91.

[15] Sarpkaya, T. (1989) "Computational methods with vortices – The 1988 freeman scholar lecture", *Journal of Fluids Engineering* 111, 5-52.

[16] Yamamoto, C. T., Meneghini, J. R., Fregonesi, R. A, Saltara, F. and Ferrari Jr., J. A. (2004) "Numerical Simulation of Vortex-Induced Vibration of Flexible Cylinders", *Journal of Fluids and Structures,* ISSN 0889-9746, pp. 467-489, vol 19.