Unsteady Separated Flow Simulations using a Cluster of Workstations

Anirudh Modi, Lyle N. Long and Robert P. Hansen
Aerospace Engineering Department
The Pennsylvania State University
University Park, PA, U.S.A.
http://cac.psu.edu/˜lnl/

Abstract *The possibility of predicting the full three-dimensional, unsteady, separated flow around complex ship and helicopter geometries is explored using unstructured grids in a parallel flow solver. The flow solver used is a modified version of the Parallel Unstructured Maritime Aerodynamics (PUMA) software. Since this requires immense computational resources, one has to often depend on expensive supercomputers to do the job. The COst effective COmputing Array (COCOA) is an inexpensive Beowulf-class supercomputer built and tested by the Aerospace Engineering Department at the Pennsylvania State University, as part of an effort to make this possible at a very economic cost. Various benchmarks were conducted on COCOA to study its performance at solving such problems.*

Keywords: CFD, parallel computing, Beowulf clusters, Navier-Stokes, MPI

1 Introduction

The prediction of unsteady separated, low Mach number flows over complex configurations (like ships and helicopter fuselages) is known to be a very difficult problem. The possibility of predicting these types of flows with the aid of inexpensive parallel computers is explored in this work. A parallel, finite volume flow solver was used and efforts were made to expedite the entire solution process.

The increasing use of helicopters in conjunction with ships poses major problems. In the presence of high winds and rough seas, excessive ship motions and turbulent separated flow from sharp-edged, box-like ship super-structures make landing a helicopter on ships a very hazardous operation. The strong unsteady flows can cause severe rotor blade deformations. Recent research on ship airwakes has been conducted from several different approaches. The most likely model of a ship, but rather crude, is a sharp edged blunt body called the General Ship Shape (GSS) [4, 3]. More geometrically precise studies have been carried out in wind tunnels [9, 2] and full scale tests have been conducted by the US Navy [12]. There have been several other attempts at numerically simulating ship airwakes, but no method to-date has been entirely satisfactory for predicting these flow fields.

Another class of problems that are very similar to those above are the flow over spheres and cylinders. Spheres and cylinders are considered as prototype examples from the class of flows past axisymmetric bluff bodies. Over the decades, a lot of work has gone into the study of unsteady separated flow over spheres and cylinders at various Reynolds numbers. Since these are simple geometric shapes and are easily reproducible and thus tested, they have enjoyed a lot of importance in the study and validation of numerical flow solvers designed to deal with such complex flows. Extensive experimental data is readily available for several different flow conditions for both the flow over a cylinder and over a sphere. Recently, Tomboulides [11] has carried on a complete *Direct Numerical Simulation* (DNS) and *Large Eddy Simulation* (LES) of flow over the sphere at various Reynolds numbers ranging from 50 to $20,000$.

2 Parallel Flow Solver

PUMA stands for **P**arallel **U**nstructured **M**aritime **A**erodynamics. It is a computer program for the analysis of internal and external, non-reacting compressible flows over arbitrarily complex 3D geometries. It is written entirely in ANSI C using MPI (Message Passing Interface) libraries for message passing and hence is highly portable giving good performance [1]. It is based on the Finite Volume Method (FVM) that solves the full three-dimensional Navier-Stokes equations, and supports mixed topology unstructured grids composed of tetrahedra, wedges, pyramids and hexahedra (bricks). PUMA may be run so as to preserve time accuracy or as a pseudo-unsteady formulation to enhance convergence to steady-state. It uses dynamic memory allocation, thus problem size is limited only by the amount of memory available on the machine. It needs 582 bytes/cell and 634 bytes/face using double precision variables (not including message passing overhead). PUMA implements a range of time-integration schemes like *Runge-Kutta*, *Jacobi* and various *Successive Overrelaxation Schemes (SOR)*, as well as both *Roe* and *Van Leer* numerical flux schemes. It also implements various monotone limiters used in second-order computations.

3 VGRID

Before any numerical solution can be computed, the physical domain must be filled with a computational grid. The grid must be constructed in a way to accurately preserve the geometry of interests while providing the proper resolution for the algorithm to be applied. The two major categories of grid construction are structured grids and unstructured grids. Each type of grid has its own particular advantages and disadvantages. Structured grids are easier to handle computationally because their connectivity information is stored block to block. Structured grids are however more difficult to construct and tend to waste memory with unnecessary cells in the far field. Unstructured grids are more difficult to handle computationally because their connectivity is stored for each node. Unstructured grids, however, tend to be easier to construct and do not waste memory in far field cell resolution.

The unstructured grids created around the geometries studied in this research, were generated using VGRID, a grid generator based on the advancing front method (AFM). VGRID [8] was developed by *ViGYAN, Inc.* in association with the NASA Langley Research Center, as a method of quickly and easily generating grids around complex objects. VGRID is a fully functional, user-oriented unstructured grid generator [10]. GridTool is the program that acts as a bridge between Computer Aided Design (CAD) packages and grid generation in VGRID. The typical process starts with generating a drawing for the geometry of interest in a CAD package (eg. ProEngineer, AutoCAD). This geometry is then exported from the CAD package to an IGES format. GridTool can then prepare the geometry for grid generation by providing VGRID with a complete and accurate definition of the geometry. This is accomplished by specifying curves along the geometry and then turning these curves into unique surface patches that define the geometry. Source terms are then added to the computational domain, which provide VGRID with the starting information for the grids in the AFM. The source terms can be freely placed anywhere in the domain and act as a control mechanism for clustering. An example of a grid generated using VGRID can be seen in Figure 1. This grid represents a part of the Apache helicopter geometry and was obtained from H.E. Jones of the U.S. Army.

4 COCOA

The COst effective COmputing Array (COCOA) is the Pennsylvania State University Department of Aerospace Engineering initiative to bring low cost parallel computing to the departmental level [6]. COCOA is a 50 processor cluster of off-the-shelf PCs connected via fast-ethernet (100 Mbit/sec). The PCs are running RedHat Linux with MPI for parallel programming and DQS for queueing the jobs. Each node of COCOA consists of Dual 400 MHz Intel Pentium II Processors in SMP configu-

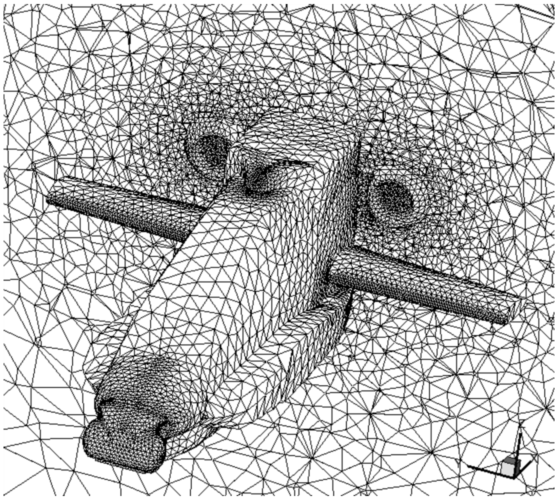

Figure 1: Volume grid generation over Apache helicopter geometry

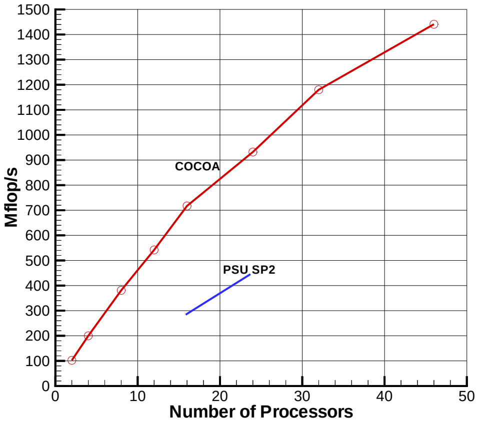

Figure 2: Total Mflops vs Number of Processors on COCOA for PUMA test case

ration and 512 MB of memory. A single Baynetworks 27-port fast-ethernet switch with a backplane bandwidth of 2.5 Gbps was used for the networking. The whole system cost approximately \$100,000 (1998 US dollars). Detailed information on how COCOA was built can be obtained from its web-site [5]. COCOA was built to enable the study of complex fluid dynamics problems using CFD (Computational Fluid Dynamics) techniques without depending on expensive external supercomputing resources.

4.1 Benchmarks

Since the cluster was primarily intended for fluid dynamics related applications, the flow solver PUMA was chosen as one of the first benchmarks. Figure 2 shows the Mflops obtained from an inviscid run on a general ship shape (GSS) geometry on different number of processors using PUMA. For this case, an unstructured tetrahedral grid with $483,565$ cells and $984,024$ faces was used and the run consumed 1.2 GB of RAM. The benchmark showed that COCOA was almost twice as fast as the Penn State IBM SP (older RS/6000-370 nodes) for our applications. Figure 5 shows the results obtained from the `netperf` test (done using: `netperf -t UDP_STREAM -l 60 -H <target-machine> -- -s 65535 -m <packet-size>`). This is indicative of the communication speed between any two nodes. It is seen that almost 96% of the peak communication speed of 100 Mbit/sec is achieved between any two nodes for packet sizes above $1,000$ bytes.

A set of well-known parallel benchmarks related to CFD were also conducted on COCOA using the publicly available *NAS*[1] *Parallel Benchmarks (NPB) suite v2.3* [13] written in Fortran 77. Of the eight tests contained in the suite, five were *kernel* benchmarks and the other three were *simulated CFD application* benchmarks. There were four different problem sizes for each test: Class“W”, “A”, “B” and “C”. While class “W” was the workstation-size test (smallest), size “C” was supercomputer-size test (largest). Figures 3 and 4 show the performance of COCOA for each of the problem sizes for the LU solver and Multigrid solver tests, respectively.

5 Results

One of the first cases which was used to validate PUMA was that of a General Ship Shape (GSS), which resembles a generic frigate. The strong unsteady flows can cause severe rotor blade deformations for helicopter landings on these ships, hence a numerical simulation was done to study this. The

---

[1] NAS stands for *Numerical Aerospace Simulation* facility, and is a part of NASA Ames Research Center

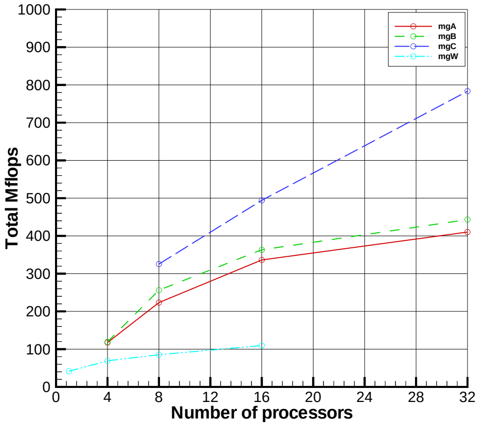

Figure 3: NAS Parallel Benchmark on COCOA: *Multigrid* (MG) test

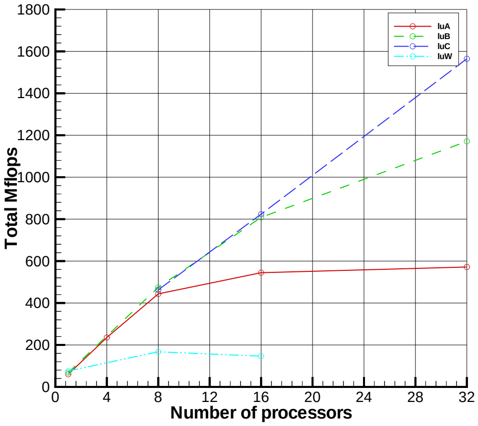

Figure 4: NAS Parallel Benchmark on COCOA: *LU solver* (LU) test

flow conditions were taken to be $M_\infty = 0.065$ and the yaw angle $\beta = 30^\circ$. Both inviscid and viscous ($Re = 1.15 \times 10^8$) cases were run [4]. For the viscous run, an unstructured tetrahedral grid with $483,565$ cells and $984,024$ faces was used and the run consumed 1.1 GB of RAM. The surface velocity contours on the GSS geometry as computed by PUMA can be seen in Figure 6. Qualitative comparisons were then made with the available experimental data which were found to be extremely good [4].

Figure 7 shows the surface total velocity contours for a Landing Helicopter Aide (LHA) geometry. This case was run to study a specific spot on the LHA where a lot of problems have been seen in the event of a helicopter landing. The flow conditions were 25 knots (12.7 m/s) at 5° yaw. The inviscid grid consisted of $1,216,709$ cells and $2,460,303$ faces and the run consumed 2.3 GB of RAM. The initial $2,000$ timesteps were traversed using SSOR and the remaining timesteps using the less expensive 2-stage Jameson-style Runge-Kutta scheme. The entire run took 39 hours on 32 processors of COCOA. Figure 8 shows surface Mach contours for an Apache helicopter geometry. The flow conditions were $U_\infty = 114$ knots. The inviscid grid consisted of $555,772$ cells and $1,125,596$ faces and the run consumed 1.9 GB of RAM.

A complete unsteady, separated flow solution over a sphere in uniform flow for Reynolds number of $1,000$ has also been simulated using PUMA. Since flow over a sphere is considered as a prototype example from the class of flows past axisymmetric bluff bodies, and because extensive experimental data is available for it, this was considered as an ideal case to use for the validation of PUMA. The unstructured grid for this geometry consisted of $306,596$ cells and $617,665$ faces, and the run consumed 600 MB of RAM on 32 nodes of COCOA (running 2-stage Runge-Kutta). The Mach contours for a specific timestep for horizontal and vertical slices are shown in Figure 9. In spite of running the case for just two cycles of vortex shedding, the time-averaged C_p results (Figure 10) compared quite well with the experimental data from Modi and Akutsu [7]. The correlation between the surface C_p values was almost perfect in the range $0^\circ \leq \theta < 110^\circ$, but seemed to vary by approximately a constant amount for the wake region ($110^\circ \leq \theta \leq 180^\circ$). Figure 11 depicts the flow along the slice $X = 0$ at different instants of time. Although the streamlines shown do not signify the actual unsteady streamlines, they do give the portray the general tendency tendency of the flow. Streaklines would be a better visualization tool, but they are very tedious and difficult to obtain. The figure clearly shows the alternating nature of the vortex shedding for $t = 0$ and $t = 2.75$ (the times used are mentioned in diameter units, i.e. the time required by the flow to convect one

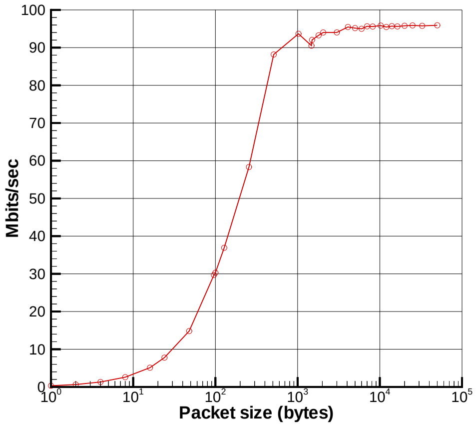

Figure 5: Mbits/sec vs Packet size on COCOA for `netperf` test

diameter in the flowfield). Detailed information on this simulation can be found in the Master's thesis of Anirudh Modi [6].

6 Conclusions

COCOA was found to be extremely suitable for our numerical simulations. One of the real benefits of inexpensive machines is that they do not have to be shared with hundreds of other users, and we do not have to wait days in a queueing system. We quite often have to wait several days at a supercomputer center just to use 16 processors. Also, while it is quite difficult to get $50,000$ CPU hours at a supercomputer center, our Beowulf cluster gives us more than $400,000$ CPU hours per year. COCOA was also found to have good scalability with most of the MPI applications used. Although Beowulf clusters have very high latency as compared to conventional supercomputers, this did not affect our applications as most of our codes had only a few large messages being communicated at every timestep. For those codes that have high communication to computation ratios, COCOA was not found to be an ideal platform because of the high latency. With several more enhancements planned for COCOA in the near future, including the addition of several more nodes and increasing the networking bandwidth by addition of more fast-ethernet cards to every node, the performance is expected to increase. Faster interconnect networks could also be used, such as Myrinet, Gigabit, or ATM; but these would increase the cost of the system by approximately 50%.

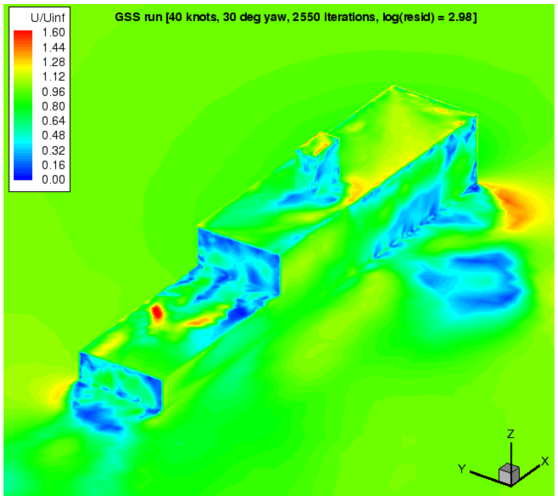

Figure 6: Surface U/U_∞ contours for GSS geometry for flow speed of 40 knots and yaw angle of 30°

References

[1] C. W. S. Bruner. Parallelization of the Euler equations on unstructured grids. PhD Dissertation, Virginia Polytechnic Institute and State University, May 1996.

[2] D. W. Hurst and S.J. Newman. Wind tunnel measurements of ship induced turbulence and the prediction of helicopter rotor blade response. *Vertica*, 12(3):267–278, 1988.

[3] Jingmei Liu and Lyle N. Long. Higher Order Accurate Ship Airwake Predictions for the Helicopter/Ship Interface Problem. *54th Annual American Helicopter Society Forum*, 1998.

[4] Lyle N. Long, Jingmei Liu, and Anirudh Modi. Higher Order Accurate Solutions of Ship Airwake Flow Fields Using Parallel Computers. *RTO AVT Symposium on Fluid Dynamics Problems of Vehicles Oper-*

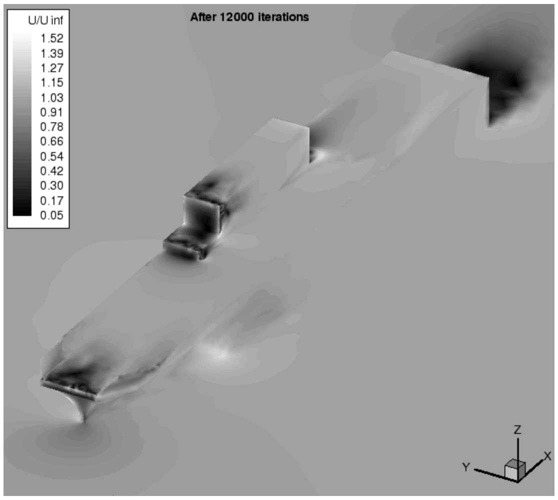

Figure 7: Surface U/U_∞ contours for LHA ship geometry (Yaw angle $\beta = 5°$, $U_\infty = 25$ knots)

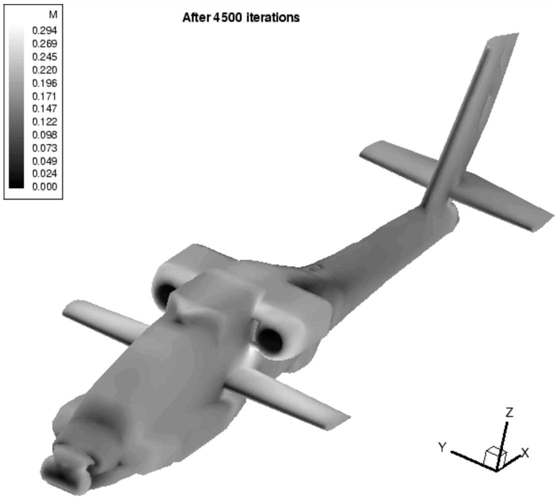

Figure 8: Surface Mach contours for Apache helicopter geometry ($U_\infty = 114$ knots)

ating near or in the Air-Sea Interface, October 1998.

[5] Anirudh Modi. COst effective COmputing Array. http://cocoa.ihpca.psu.edu, 1998.

[6] Anirudh Modi. Unsteady Separated Flow Simulations using a Cluster of Workstations. M.S. Thesis, The Pennsylvania State University, May 1999.

[7] V. J. Modi and T. Akutsu. Wall Confinement Effects for Spheres in the Reynolds Number Range of 30-2000. *Trans. of ASME I: Journal of Fluids Engg*, 106:66–73, March 1984.

[8] Shahyar Pirzadeh. Progress Toward a User-Oriented Unstructured Viscous Grid Generator. *AIAA Paper 96-0031*, January 1996.

[9] M. M. Rhoades and J. Val. Healey. Flight deck aerodynamics of a non-aviation ship. *Journal of Aircraft*, 29(4):619–626, 1992.

[10] Steven Schweitzer. Computational Simulation of Flow around Helicopter Fuselages. M.S. Thesis, The Pennsylvania State University, January 1999.

[11] Ananias Georgios Tomboulides. Direct and large-eddy simulation of wake flows: Flow past a sphere. PhD Dissertation, Princeton University, June 1993.

[12] S. Williams and K. R. Long. Dynamic interface testing and the pilots rating scale. *The 53rd AHS Forum*, April 1997.

[13] Alex Woo. NAS Parallel Benchmarks. http://science.nas.nasa.gov/Software/NPB/, 1997.

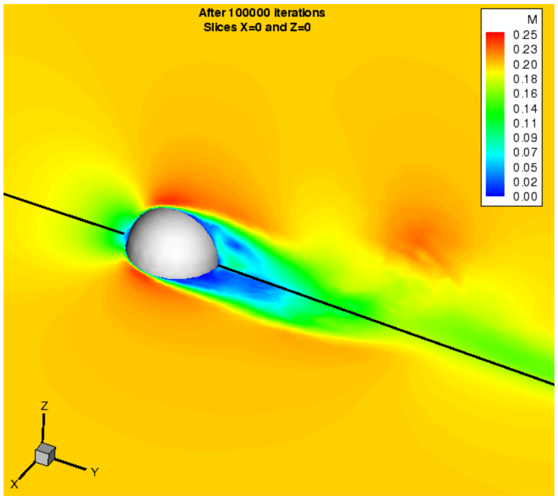

Figure 9: Mach contours for viscous sphere run ($M_\infty = 0.2$, $Re = 1000$)

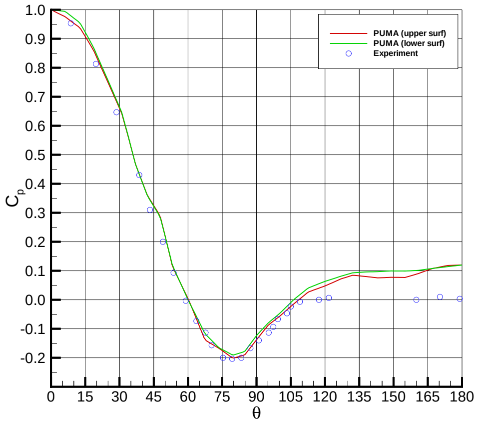

Figure 10: Time averaged plot for C_p vs θ for upper surface compared with experiments [7], for viscous sphere run ($M_\infty = 0.2$, $Re = 1000$) [Note: Here, C_p is defined as $(P_\theta - P_{60^\circ})/(P_{0^\circ} - P_{60^\circ})$]

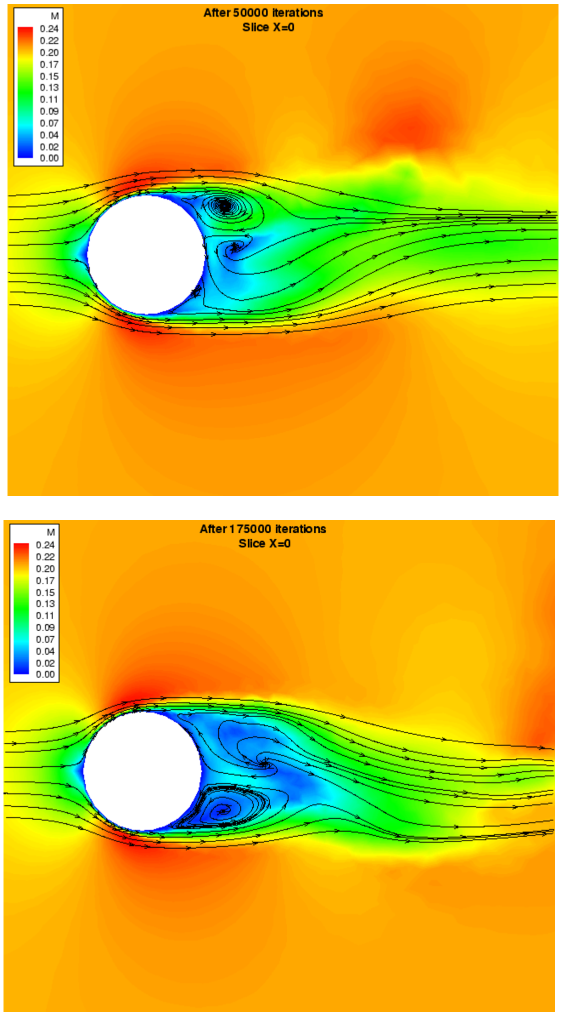

Figure 11: Mach contours and streamlines (at $t = 0.0, 2.75$) for viscous sphere run ($M_\infty = 0.2$, $Re = 1000$)