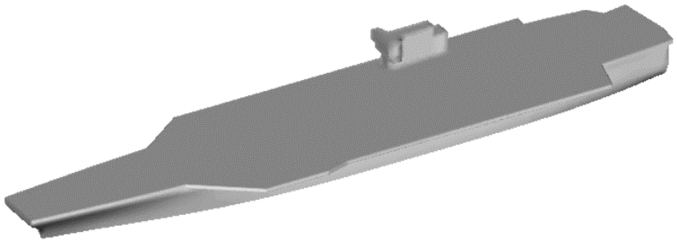
Aerospace Engineering

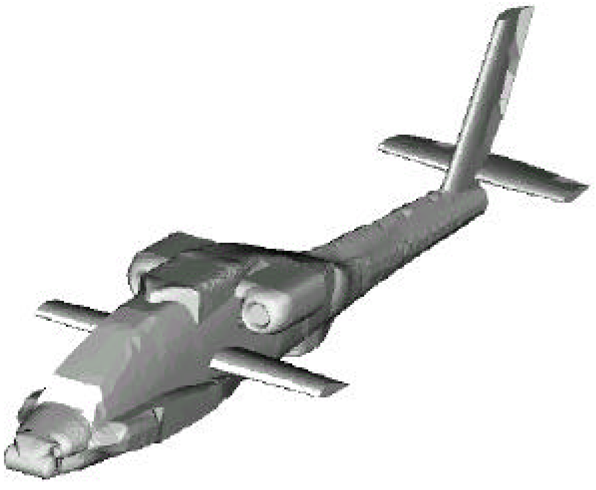
Unsteady Separated Flow Simulations using a Cluster of Workstations

Anirudh Modi

Advisor: Dr. Lyle N. Long

4/27/99





OUTLINE

Background

OUTLINE

Penn State Aerospace Engineering

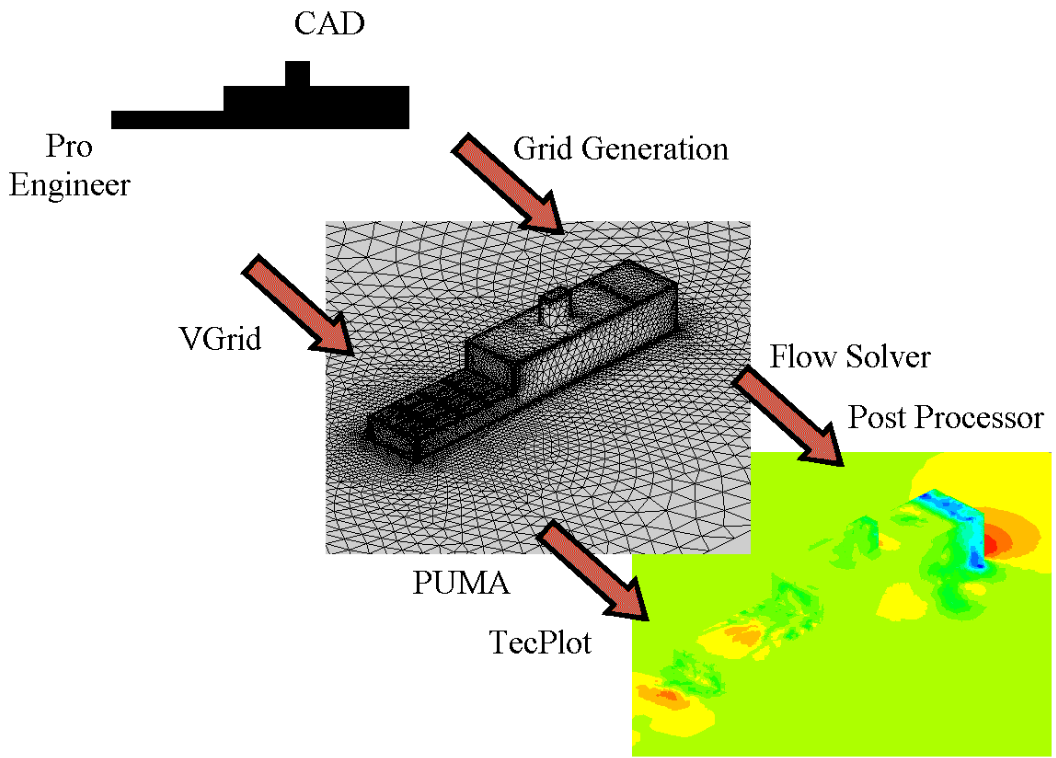
Background

OUTLINE

Aerospace Engineering

Background

CAD to Solution

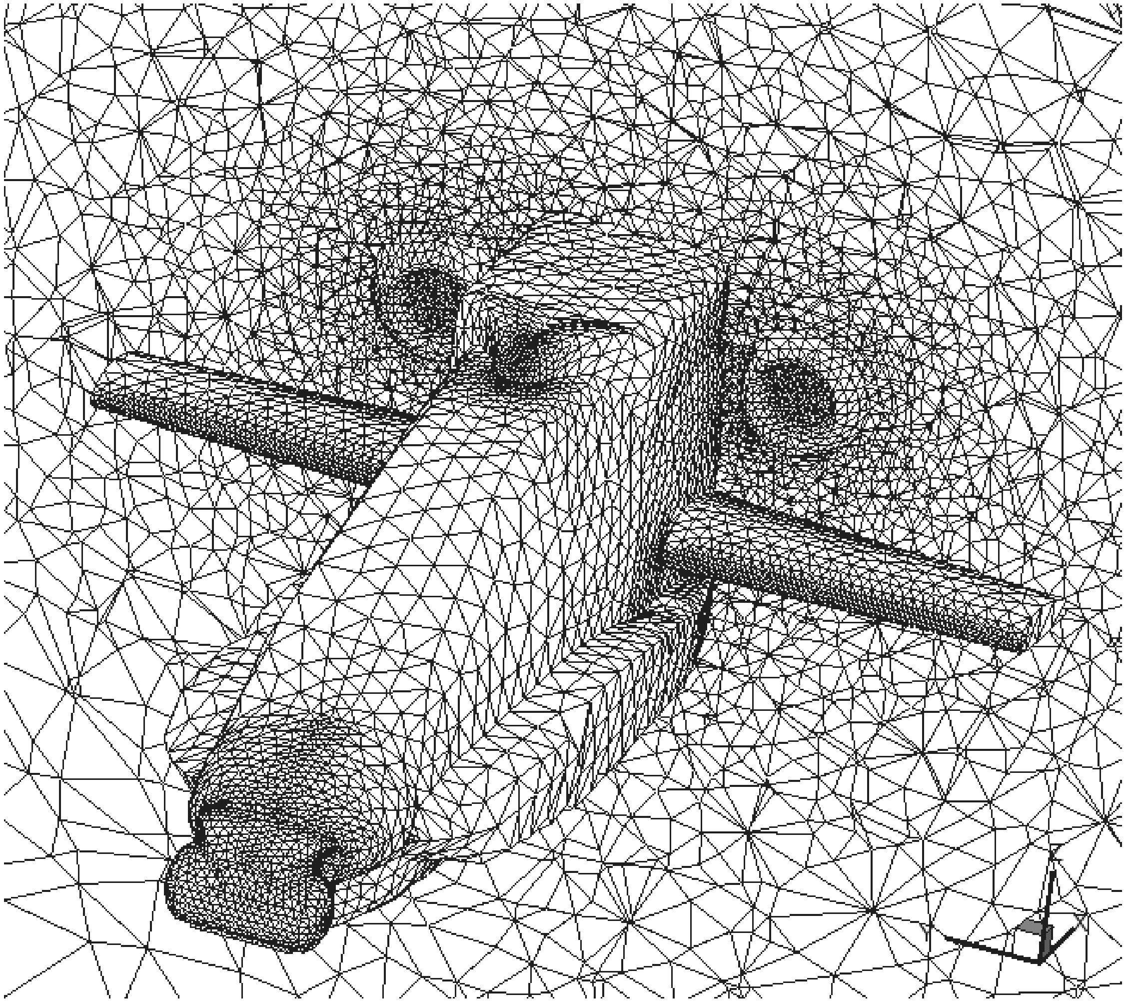


OUTLINE

Background

CAD to Solution

Grid Generation



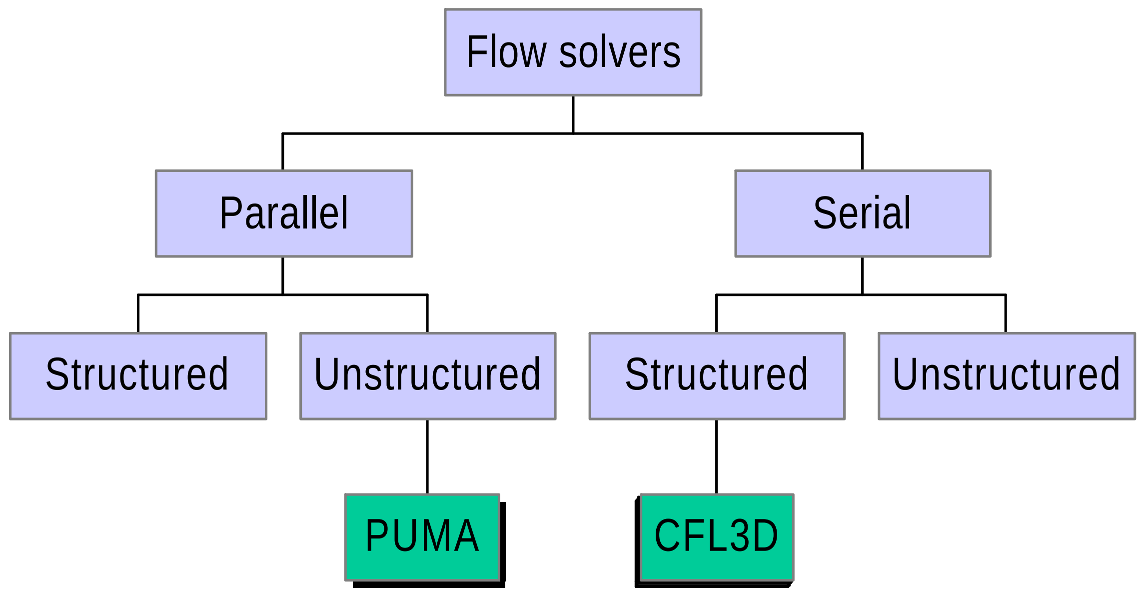
OUTLINE

Background

CAD to Solution

Grid Generation

Flow Solver



OUTLINE

Penn State Aerospace Engineering

Background

CAD to Solution

Grid Generation

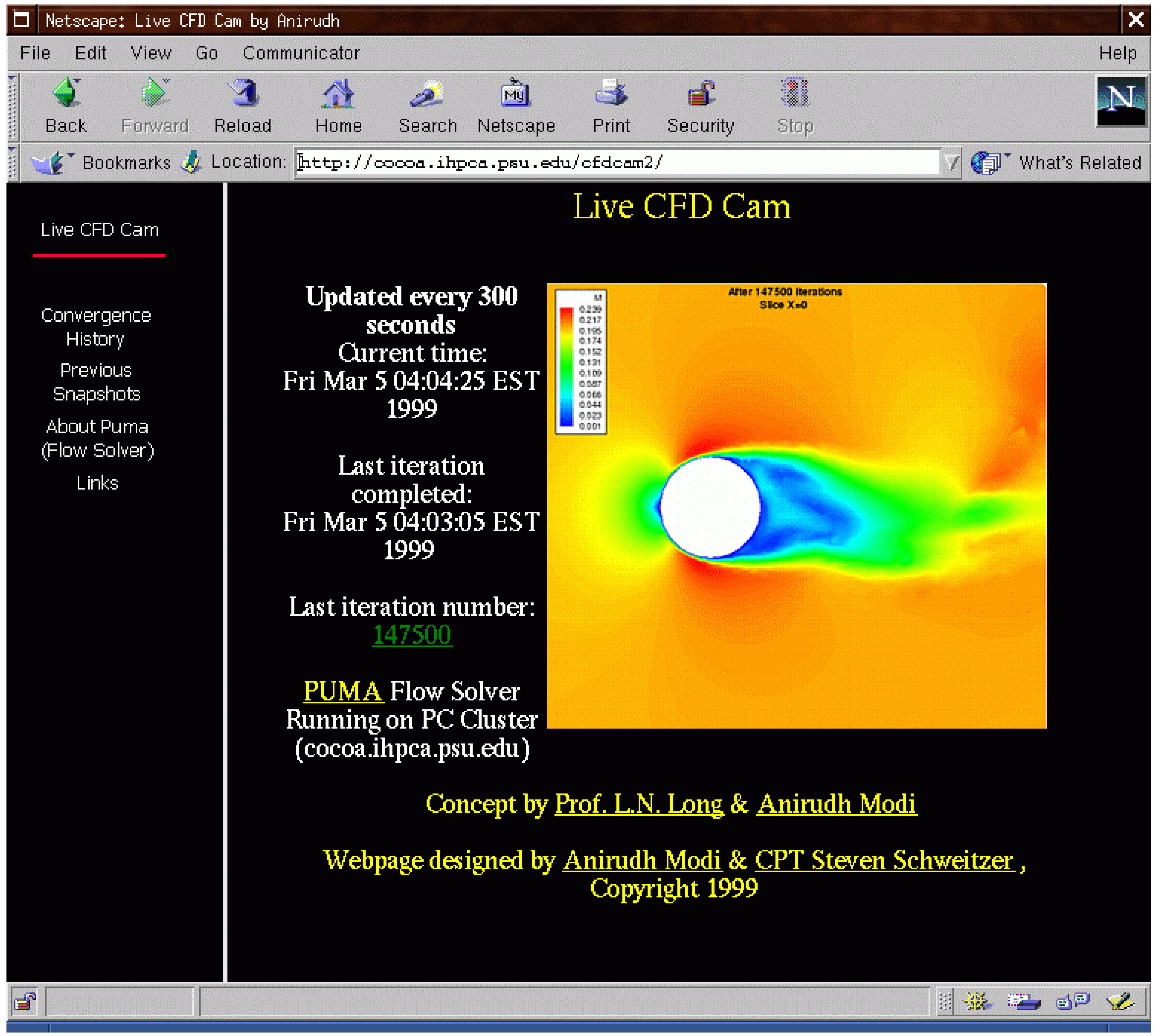
Flow Solver

Parallel Computers

OUTLINE

PennState Aerospace Engineering

- Background
- CAD to Solution
- Grid Generation
- Flow Solver
- Parallel Computers
- Post-processing



OUTLINE

Aerospace Engineering

Background

CAD to Solution

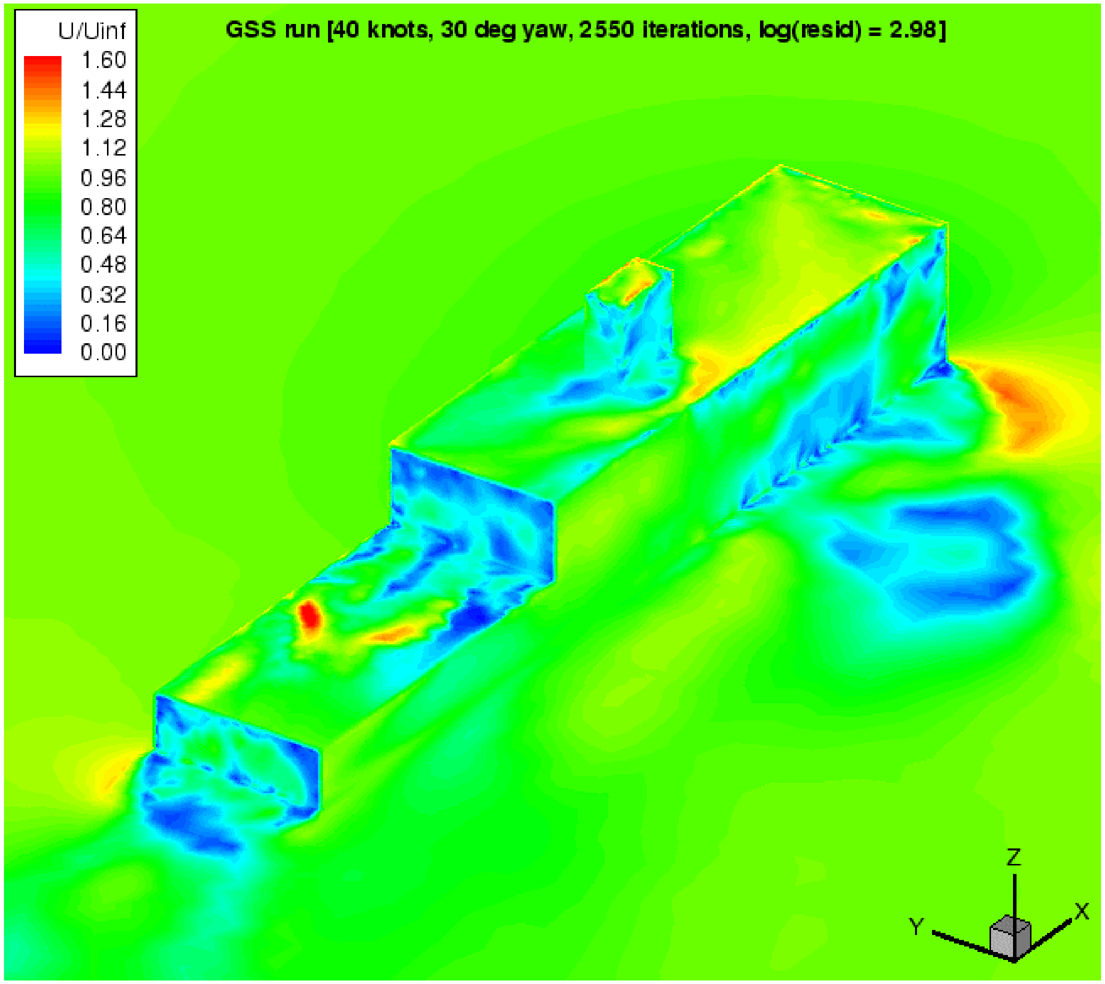
Grid Generation

Flow Solver

Parallel Computers

Post-processing

Results



OUTLINE

PENNSTATE
Aerospace Engineering

Background

CAD to Solution

Grid Generation

Flow Solver

Parallel Computers

Post-processing

Results

Future Work

NLDE

k-exact

Preconditioning

OUTLINE

Aerospace Engineering

- Background
- CAD to Solution
- Grid Generation
- Flow Solver
- Parallel Computers
- Post-processing
- Results
- Future Work
- Conclusions

Background

Aerospace Engineering

- 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.
  - » helicopter landing on ship is very hazardous.
  - » for helicopters, knowledge of separated flow in sufficient detail needed for a study of rotor-fuselage interactions.
- Previous approaches mainly used serial computers and those which utilized parallel computers demanded heavy supercomputing resources which were very expensive to obtain.

Background

Aerospace Engineering

- ➢ No standard test case exists for flows around such complex configurations.
- ➢ However, flow over spheres and cylinders are considered as prototype examples from the class of flows past axisymmetric bluff bodies.
  - » A lot of work has gone into the study of unsteady separated flow over spheres and cylinders at various Reynolds numbers.
  - » Tomboulides[1991]: *Direct Numerical Simulation* (DNS) and *Large Eddy Simulation* (LES) of flow over the sphere (Reynolds numbers ranging from to 500 to 20,000).

Past Work

Aerospace Engineering

- Recent research on ship airwakes has been conducted from several different approaches [J.Healy, '92].

- Chaffin and Berry [1990] utilized the well known CFL3D flow solver for their investigation into separated flow around helicopter fuselages.

- Duque et al [1995] have used the OVERFLOW flow solver to analyze the flow around the United States Army's RAH-66 Comanche helicopter.

CAD to Solution

Penn State Aerospace Engineering

CAD Package
Generate IGES file

GridTool
Patch geometry and set source parameters

VGRID
Generate unstructured volume mesh

PostGrid
Complete grid and adjust quality
Save as .bc, .mapbc, .grd, .int files

fromVgrid utility
Converts output from PostGrid to .sg files used by PUMA

fromVgrid utility
Converts output from PostGrid to .sg files used by PUMA

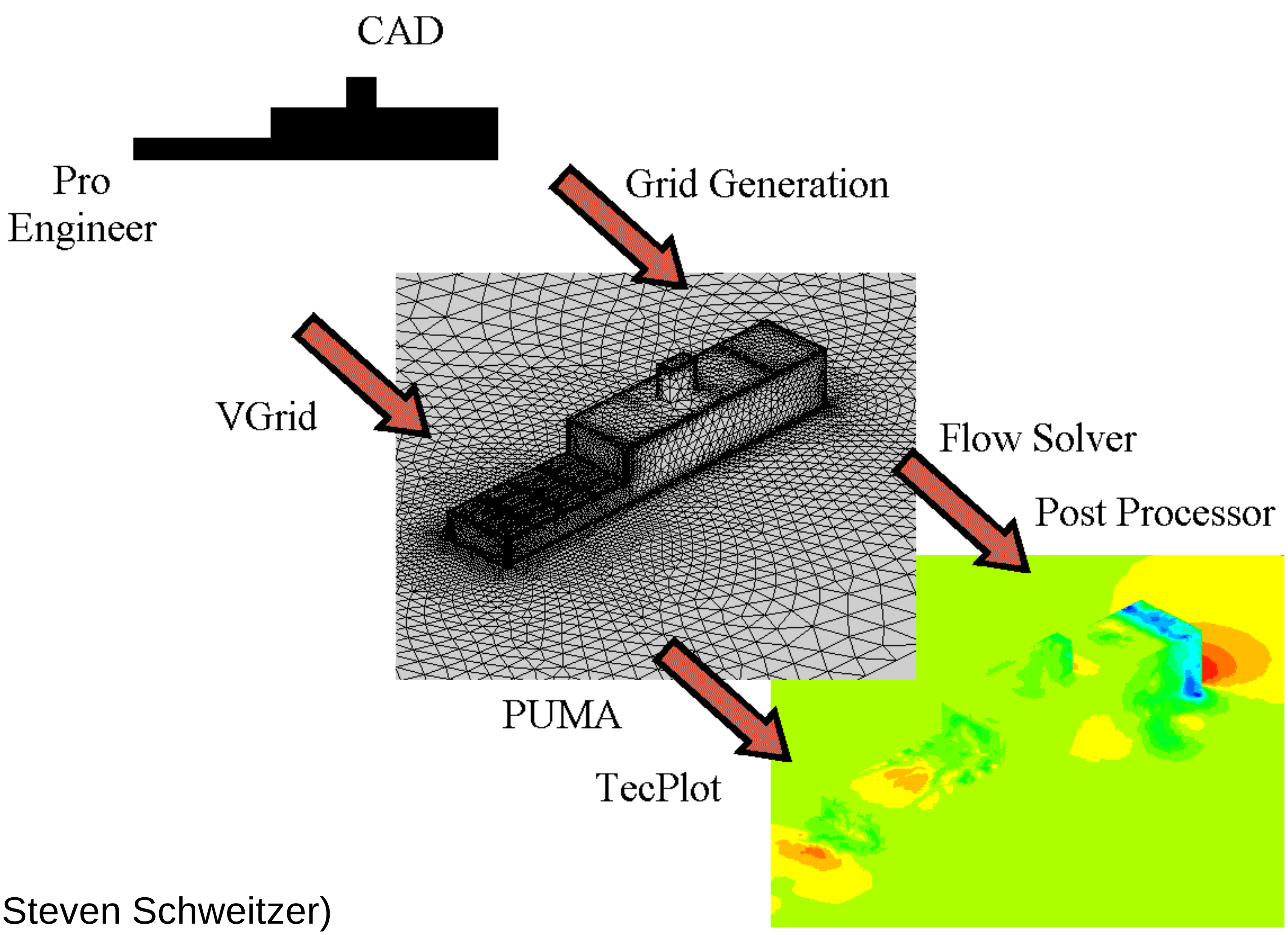
gps utility
Optimize grid for minimum communications for parallel runs (.sg.gps file)

PUMA
Generates flow solution over the given grid (.rst files)

toTecplot utility
Generates .dat files for TECPLOT

TECPLOT/Live CFD-cam
Display the solution

Example



(Courtesy Steven Schweitzer)

Grid Types

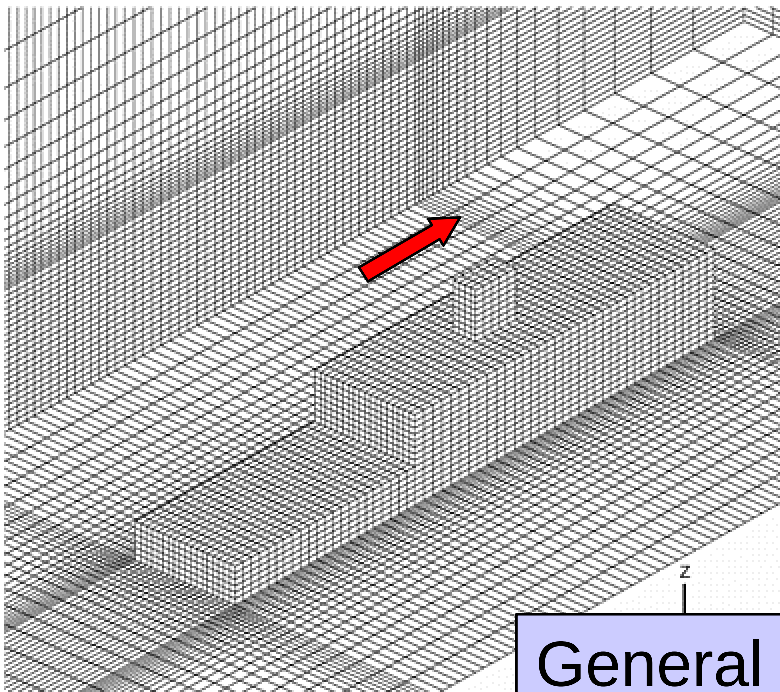
Aerospace Engineering

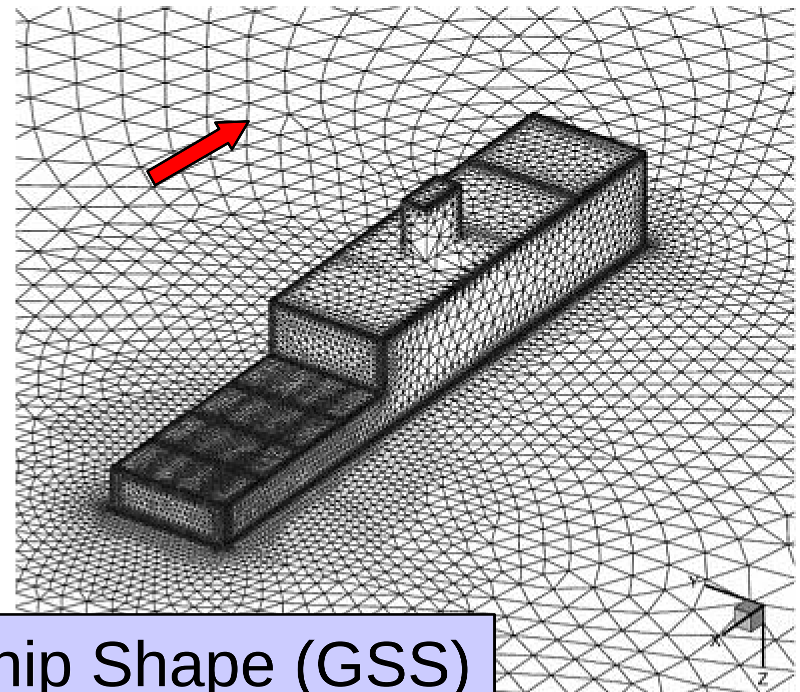
Structured

-Easier computationally
-Memory waster
-Difficult with complex shapes

Unstructured

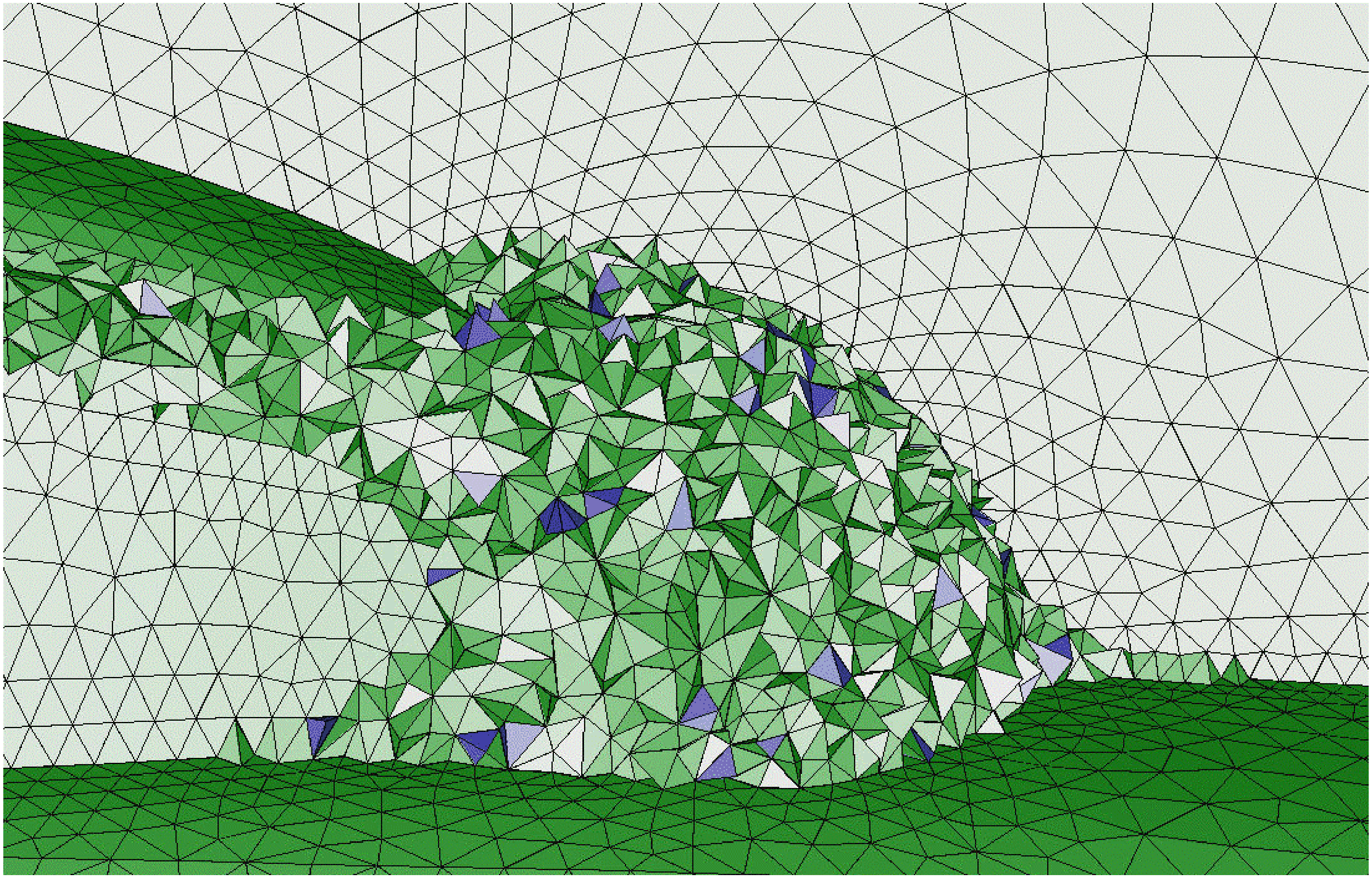
-Difficult computationally
-Cells easily concentrated
-Easy to construct around any shape





General Ship Shape (GSS)

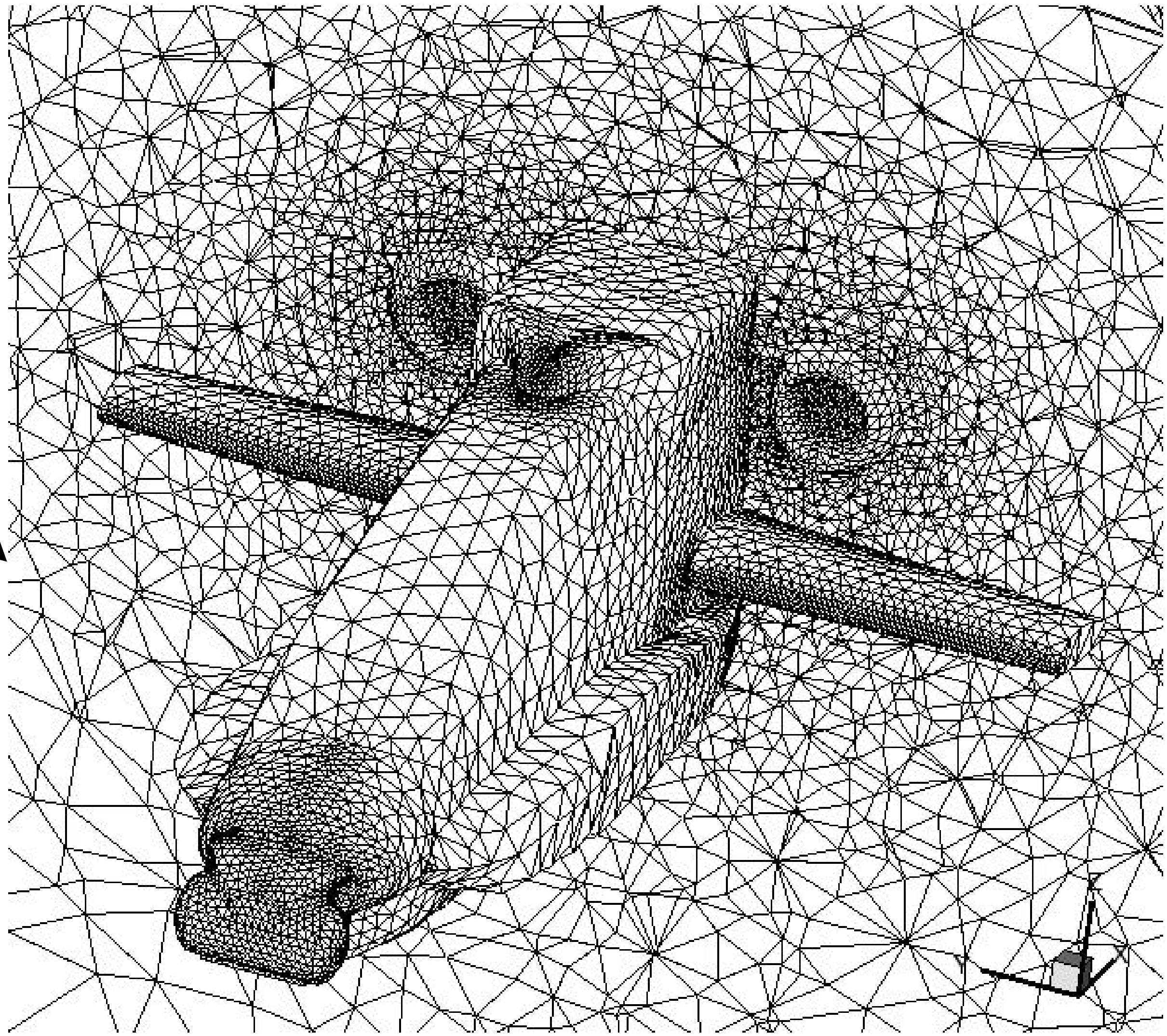
VGRID



VGRID

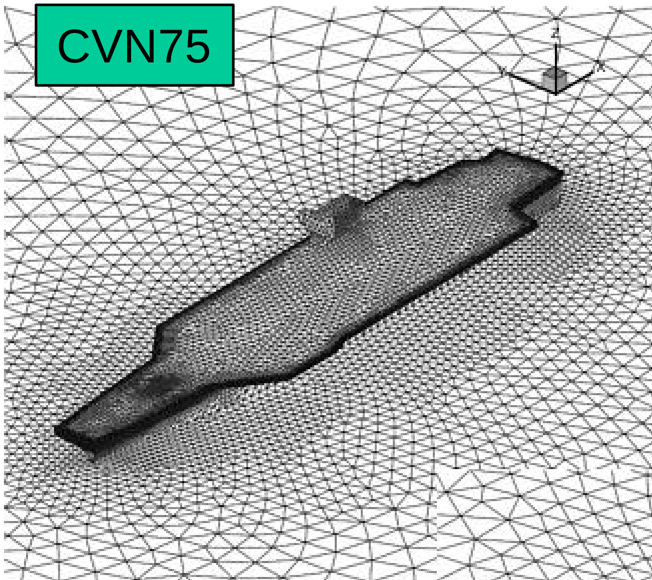
Aerospace Engineering

555,772 cells
1,125,596 faces



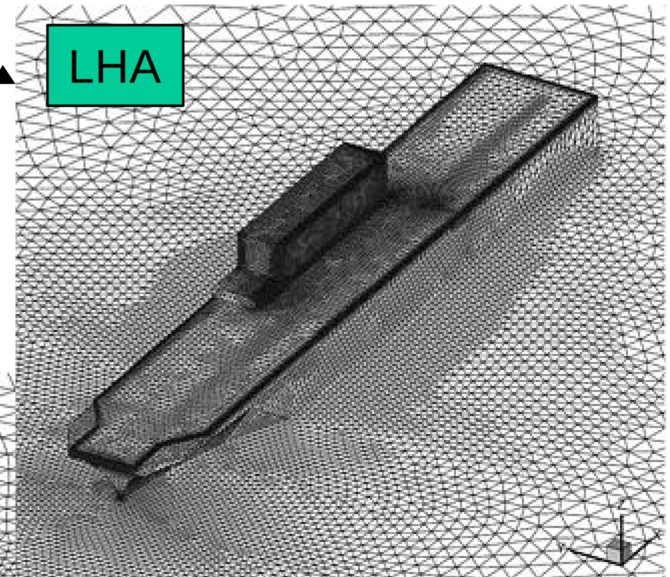
Unstructured Grid Samples

Aerospace Engineering

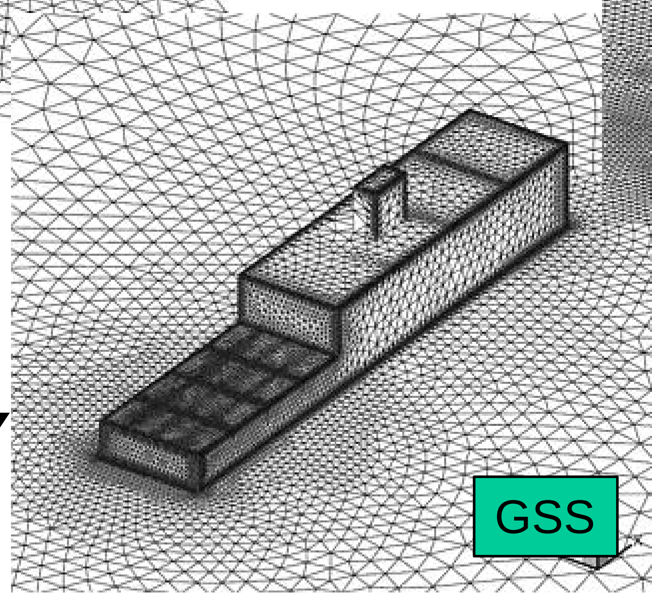


478,506 cells
974,150 faces

1,216,709 cells
2,460,303 faces

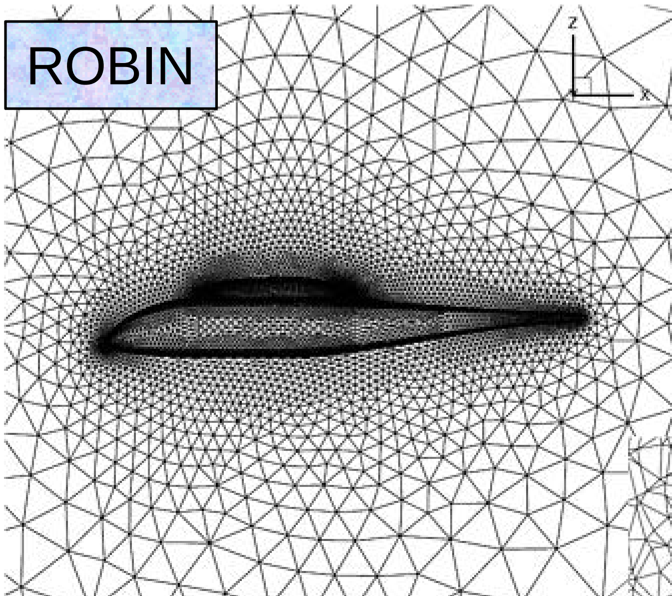


483,565 cells
984,024 faces

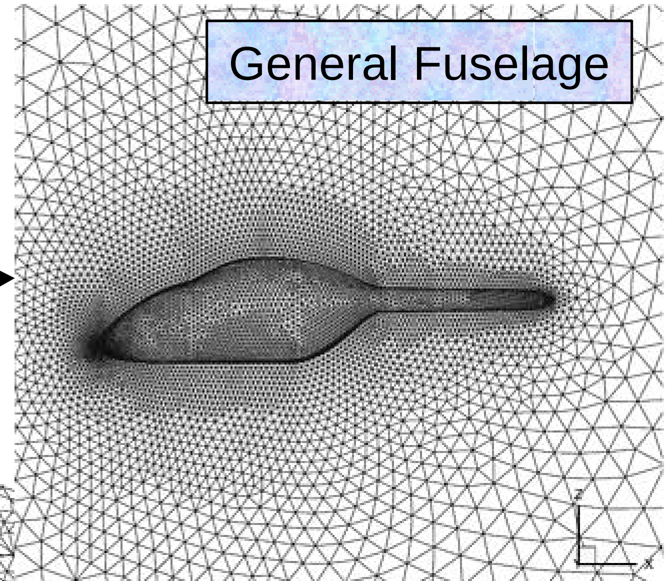


Ship Configurations

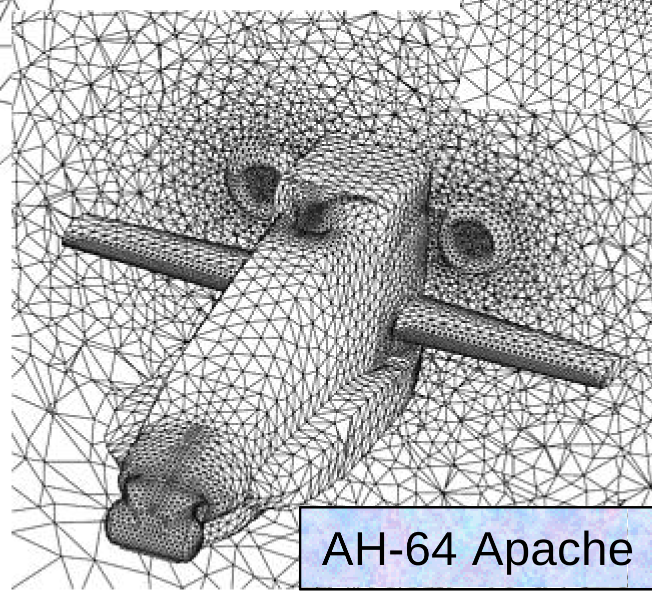
Unstructured Grid Samples



260,858 cells
532,492 faces



380,089 cells
769,240 faces

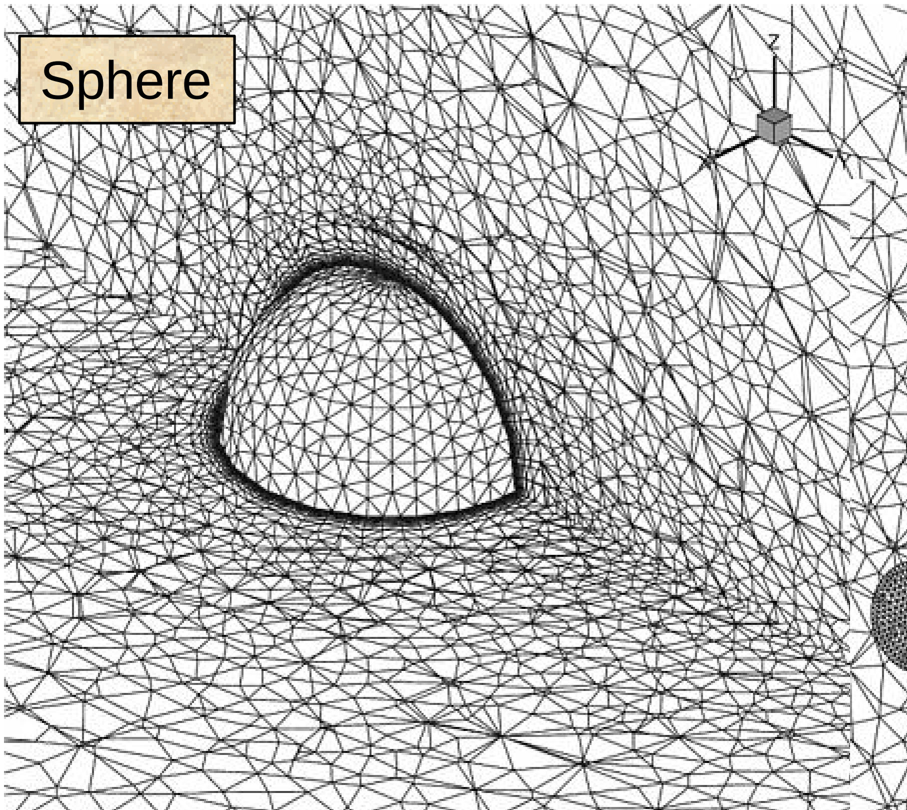


555,772 cells
1,125,596 faces

Helicopter Configurations

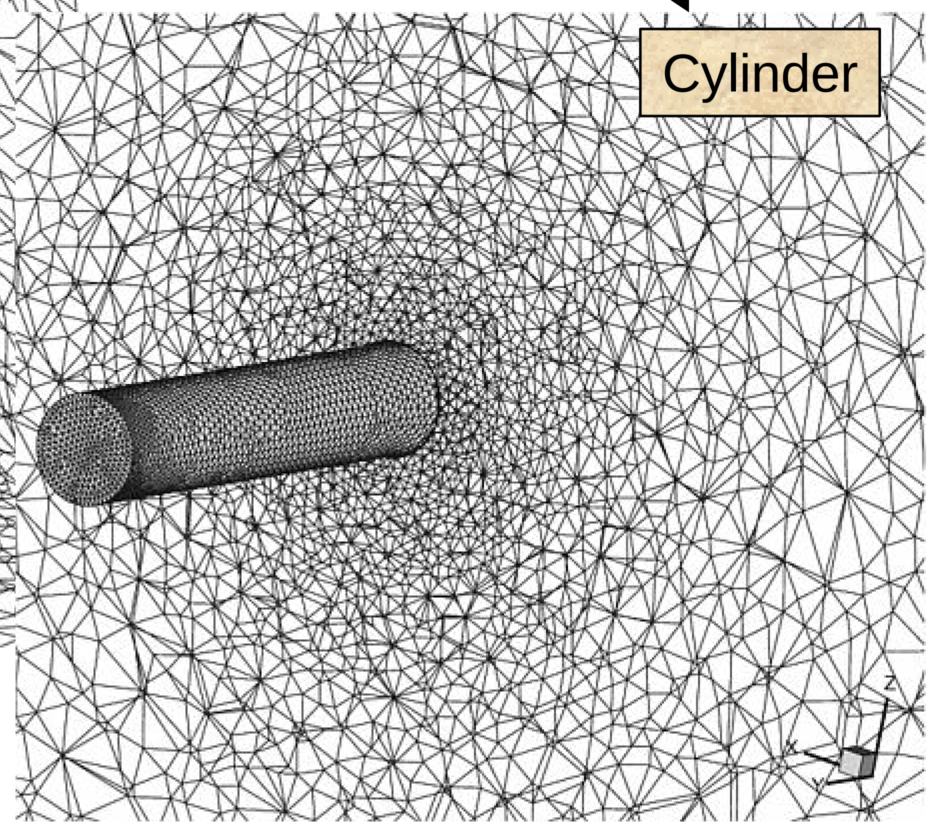
Unstructured Grid Samples

Aerospace Engineering



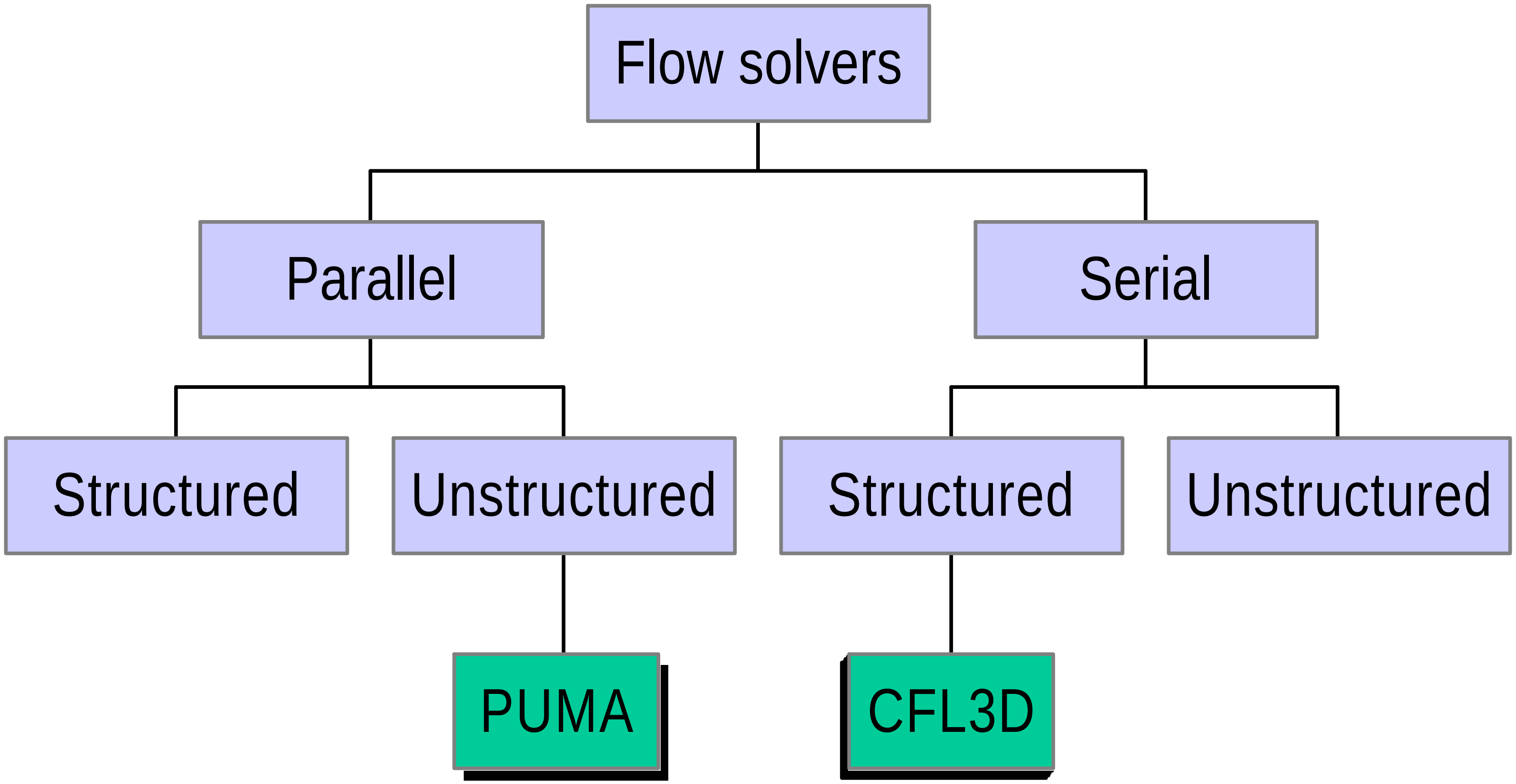
306,596 cells
617,665 faces

806,668 cells
1,620,576 faces



Viscous grids over axisymmetric bluff bodies

Flow Solvers



PUMA: Introduction

Aerospace Engineering

- Parallel Unstructured Maritime Aerodynamics. Written by Dr. Christopher W.S. Bruner (U.S. Navy, PAX River)
- Computer program for analysis of internal and external non-reacting compressible flows over arbitrarily complex 3D geometries (Navier-Stokes solver).
- Written entirely in ANSI C using MPI library for message passing and hence highly portable giving good performance.
- Based on Finite Volume method and supports mixed topology unstructured grids composed of tetrahedra, wedges, pyramids and hexahedra (bricks).

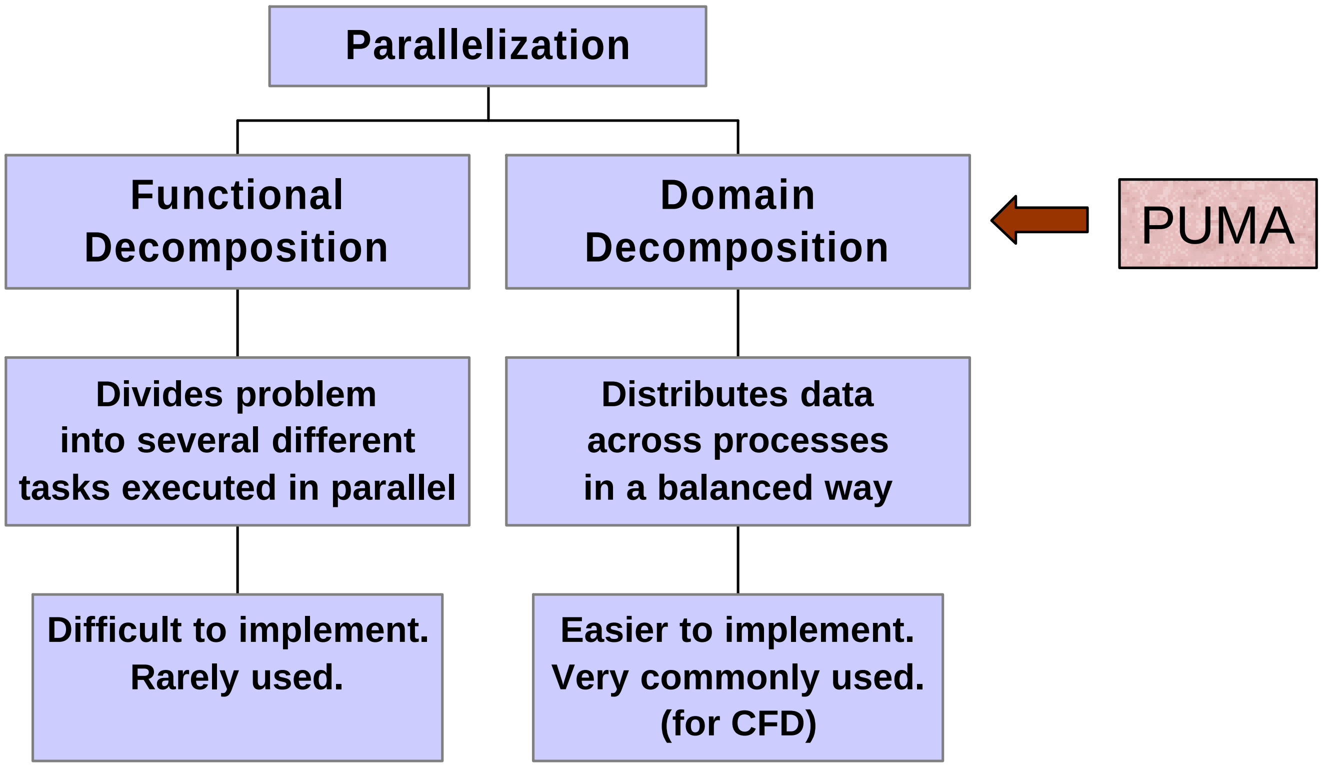
PUMA: Introduction

Aerospace Engineering

- May be run so as to preserve time accuracy or as a pseudo-unsteady formulation (different Δt for every cell) to enhance convergence to steady-state.
- Uses dynamic memory allocation, thus problem size is limited only by the amount of memory available on the machine. Needs 582 bytes/cell and 634 bytes/face using double precision variables (not including message passing overhead). Requires 25000-30000 flops/iter/cell.
- PUMA implements a range of time-integration schemes like *Runge-Kutta*, *Jacobi* and various *Successive Over-relaxation Schemes (SOR)*, as well as both *Roe* and *Van Leer* numerical flux schemes.

Parallelization in PUMA

Aerospace Engineering



PUMA uses *Single Program Multiple Data* (SPMD) parallelism, i.e., same code is replicated to each process.

Parallelization in PUMA

Aerospace Engineering

communication time = latency + (message size)/(bandwidth)

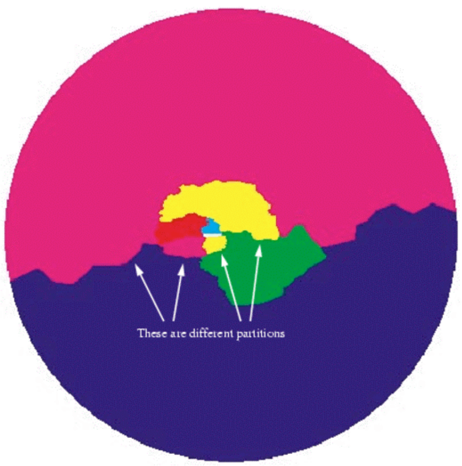
First term — Second term

PUMA

Domain Decomposition

Approach #1: Minimization of inter-domain boundaries	Approach #2: Minimization of communications
Length of all shared data is minimized	Number of messages exchanged is minimized
Neglects latency. Ideal for slow networks.	Neglects second term. Ideal for fast networks.
Popular techniques require eigenvalues of connectivity matrix. Expensive.	Large knowledge base exists. Implementation is easy. Expensive.
Requires a-priori knowledge of number of domains/processors	Independent of number of domains/processors

Grid around RAE 2822 a/f



8-way partitioning.
Using METIS s/w

8-way partitioning.
Using GPS reordering.

Parallelization in PUMA

Aerospace Engineering

- Each compute node reads its own portion of the grid file at startup.
- Cells are divided among the active compute nodes at runtime based on cell ID and only faces associated with local cells are read.
- Faces on the interface surface between adjacent computational domains are duplicated in both domains. Fluxes through these faces are computed in both domains.
- Solution variables are communicated between domains at every timestep which ensures that the computed solution is independent of the number of compute nodes.
- Communication of the solution across domains is all that is required for first-order spatial accuracy, since Q_L and Q_R are simply cell averages to the first order.
- If the left and right states are computed to higher-order, then Q_L and Q_R are shared explicitly with all adjacent domains. The fluxes through each face are then computed in each domain to obtain the residual for each local cell.

CFL3D vs PUMA

Penn State Aerospace Engineering

	CFL3D	PUMA
	Finite Difference solver	Finite Volume solver
Parallel	N	Y
Multigrid	Y	N
Unstructured Grids	N	Y
Structured Block Grids	Y	N
Order of Accuaracy	2^{nd}	2^{nd}
Time Integration Scheme		
Runge-Kutta	N	Y
Gauss Seidel	N	Y
Jacobi	N	Y
RSOR	N	Y
FSOR	N	Y
SSOR	N	Y
ADI	Y	N

CFL3D vs PUMA

Aerospace Engineering

Turbulence Models	CFL3D	PUMA
Baldwin-Lomax	Y	N
Baldwin-Barth	Y	N
Spalart-Allmaras	Y	Y
Wilcox k-Omega	Y	Y
Menter’s k-Omega	Y	N
Abid k-Epsilon	Y	N
Speziale-Gatski k-Omega	Y	N
Speziale-Gatski k-Epsilon	Y	N
Girimaji k-Epsilon	Y	N
Upwind	Y	Y
Compressible	Y	Y
Lines of Code	100K	10K
Programming Language	F-77	C/MPI
Portable	Y	Y

Parallel Computers

COst effective COmputing Array (COCOA)

25 Dual PII 400 MHz
512 MB RAM each (12+ GB!!)
54 GB Ultra2W-SCSI Disk on server
100 Mb/s Fast Ethernet cards
Baynetworks 450T 27-way switch
(backplane bandwidth of 2.5 Gbps)
Monitor/keyboard switches
RedHat Linux with MPI

http://cocoa.ihpca.psu.edu

Cost just $100,000!! (1998 dollars)

COCOA: Motivation

Aerospace Engineering

- To get even 50,000 hrs of CPU time in a supercomputing center is difficult. COCOA can offer more that 400,000 CPU hrs annually!
- One often has to wait for days in queues before the job can run.
- Commodity PCs are getting extremely cheap. Today, it just costs $3K to get a dual PII-400 computer with 512MB RAM from a reliable vendor like Dell!
- Advent of Fast Ethernet (100 Mbps) networking has made a reasonably large PC cluster feasible (at a very low cost; 100 Mbps ethernet adaptor ~ $70). Myrinet and Gigabit networking are soon getting popular.
- Price/performace (or $/Mflop) for these cheap clusters is way better than for a IBM SP/SGI/Cray supercomputer (atleast factor of 10 better!)
- Maintenance for such a PC cluster is less cumbersome than the big computers. A new node can be added to COCOA in just 10 minutes!

COCOA

Aerospace Engineering

- COCOA runs on commodity PCs using commodity software (RedHat Linux).
- Cost of software: negligible. The only commercial software installed are Portland Group Fortran 90 compiler and TECPLOT.
- Free version of MPI from ANL (MPICH) and Pentium GNU C compiler (generates highly optimized code for Pentium class chips) are installed.
- Distributed Queueing System (DQS) has been setup to submit the parallel/serial jobs. Several minor enhancements have been incorporated to make it extremely easy to use. Live status of the jobs and the nodes is available on the web:

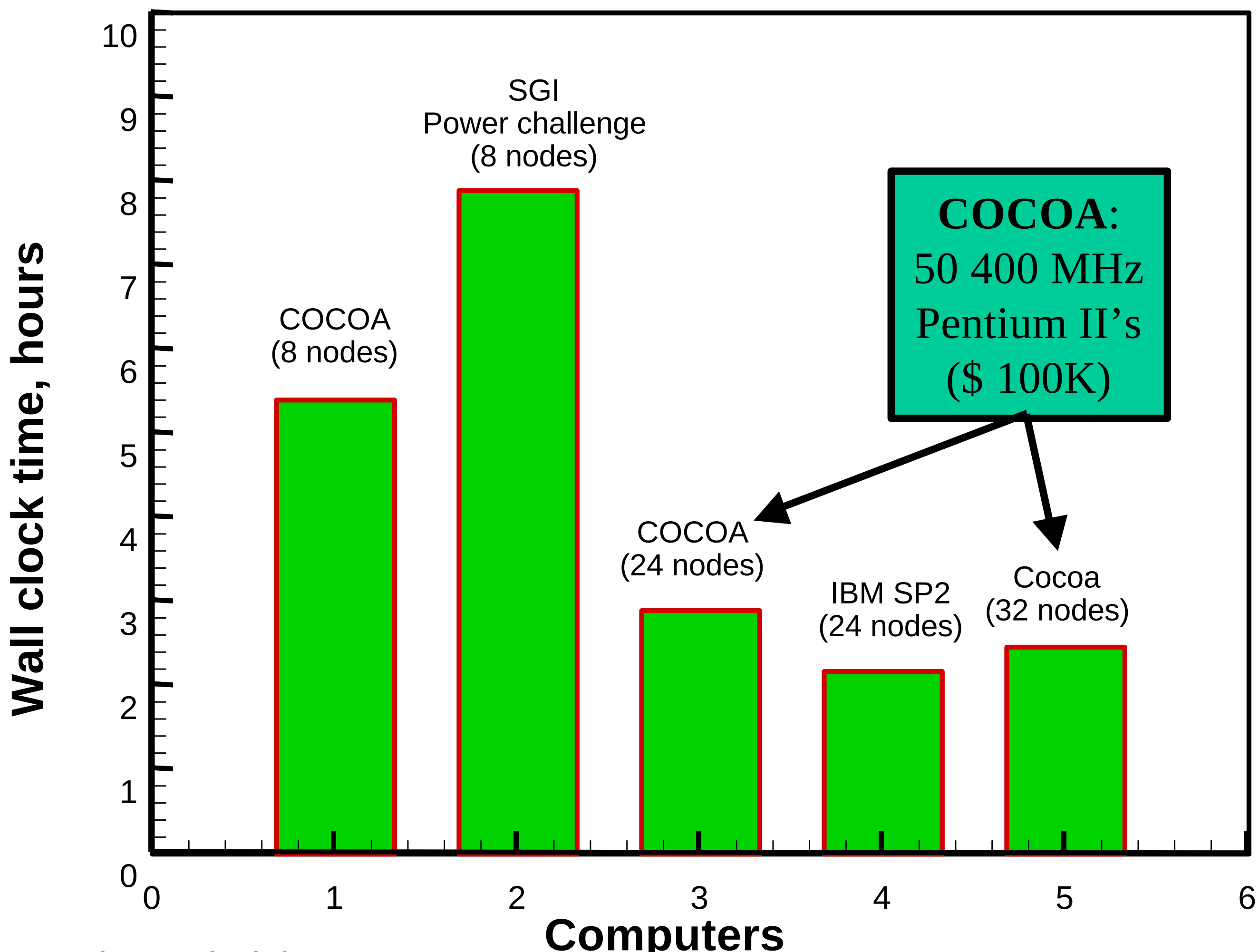
  http://cocoa.ihpca.psu.edu

- Details on how COCOA was built can be found in the COCOA HOWTO:

  http://bart.ihpca.psu.edu/cocoa/HOWTO/

Timings of NLDE on Various Computers

Aerospace Engineering

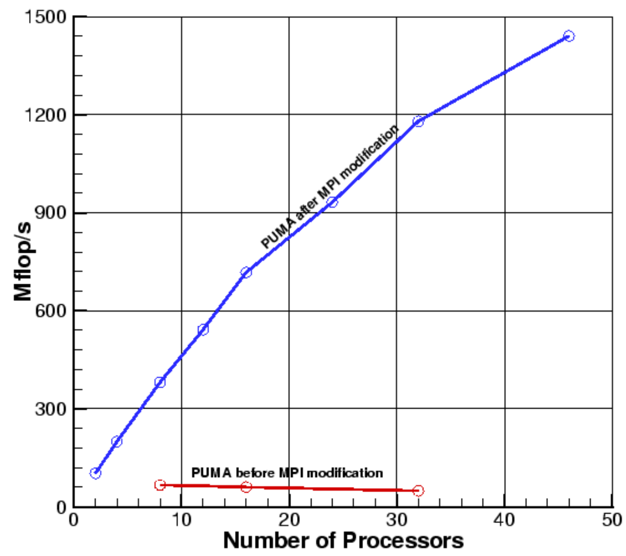


(Courtesy Dr. Jingmei Liu)

COCOA: Modifications to PUMA

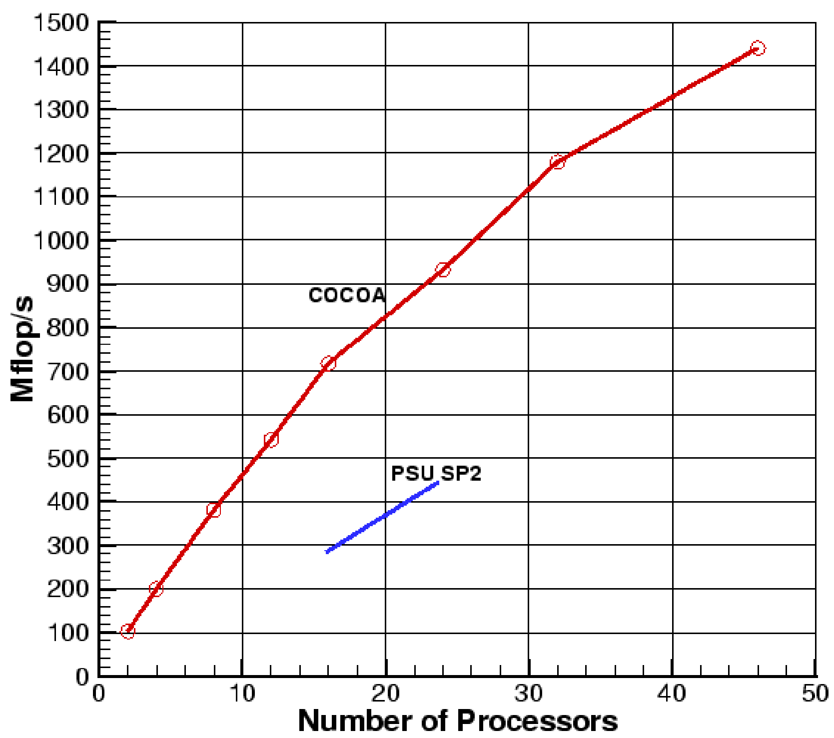
Aerospace Engineering

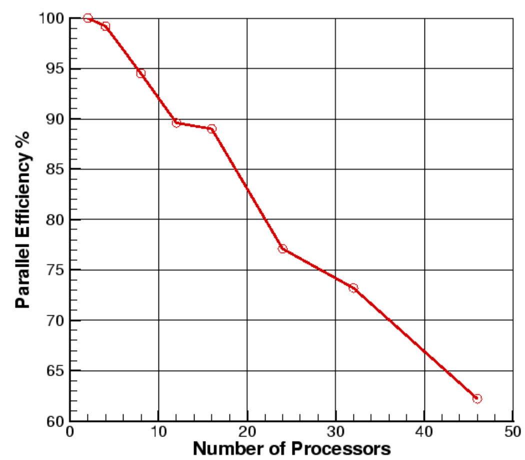
- Although PUMA is portable, it was aimed at very low-latency supercomputers. Running it on a high-latency cluster like COCOA posed several problems.
- PUMA often used several thousand very small messages (< 100 bytes) for communication which degraded its performance considerably (latency!!). These messages were non-trivially packed into larger messages (typically > 10 Kbyes) before they were exchanged.
- After modification, the initialization time was reduced by a factor of 5-10, and the overall performace was improved by a factor of 10-50!!

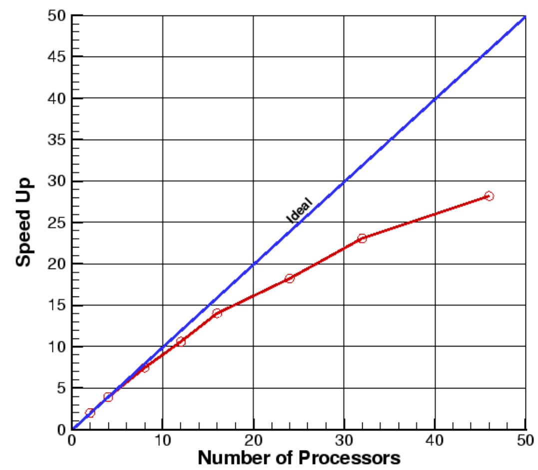


COCOA: Benchmarks

PENNSTATE
Aerospace Engineering





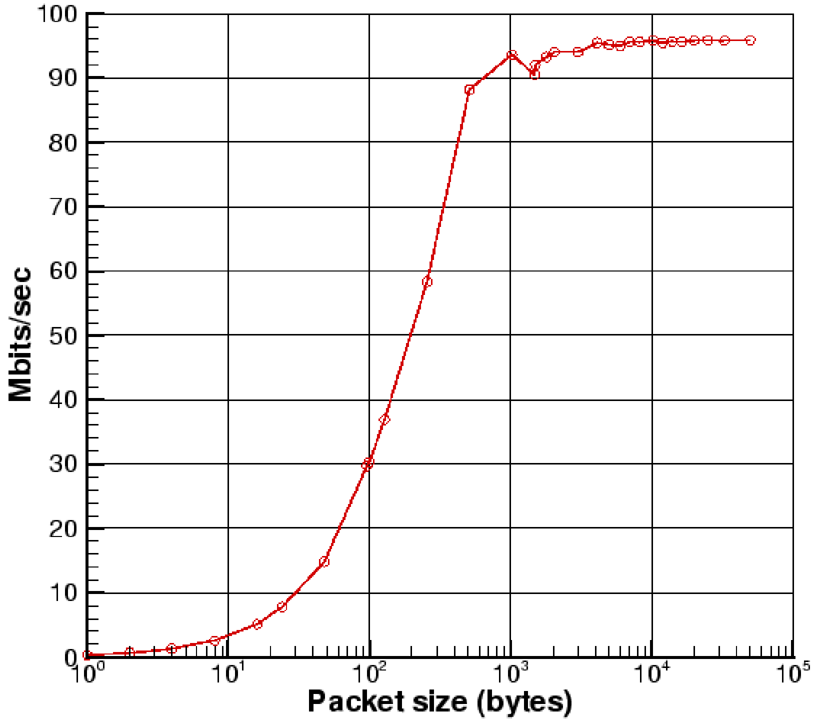


Performance of "*Modified PUMA*"

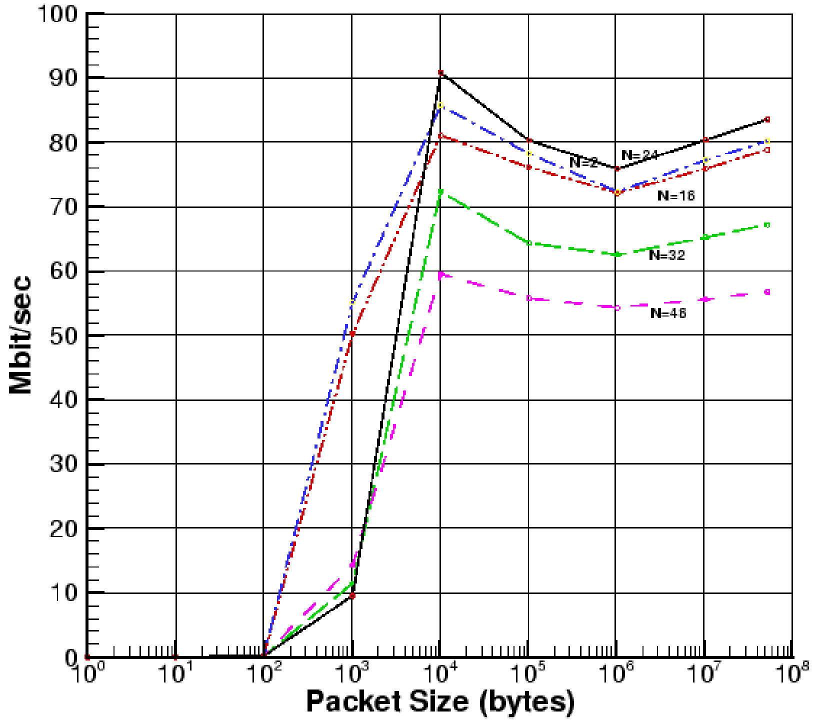
COCOA: Benchmarks

Aerospace Engineering

Network Performance



netperf test between any two nodes



MPI_Send/Recv test

➡ Ideal message size >= 10 Kbytes

COCOA: Benchmarks

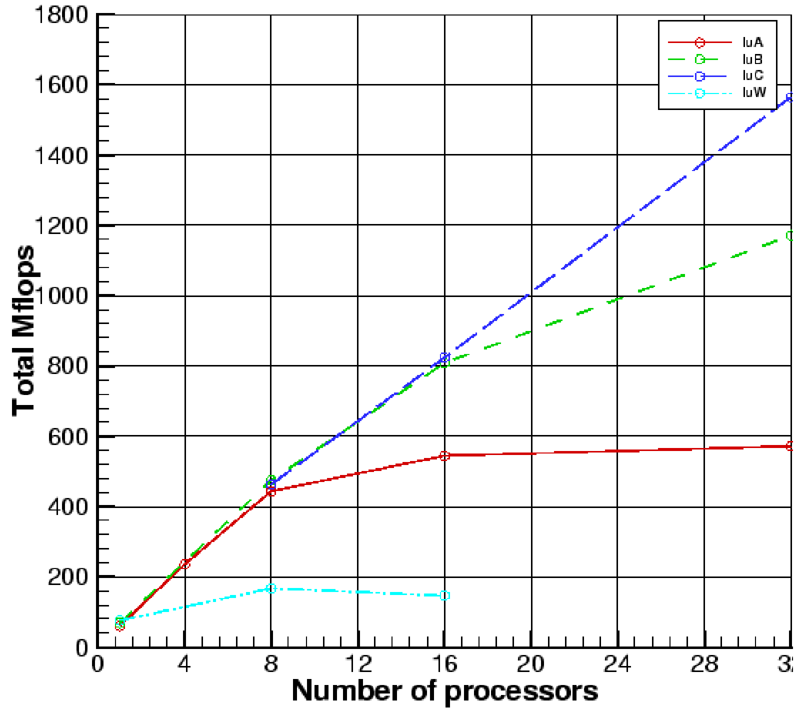
Aerospace Engineering

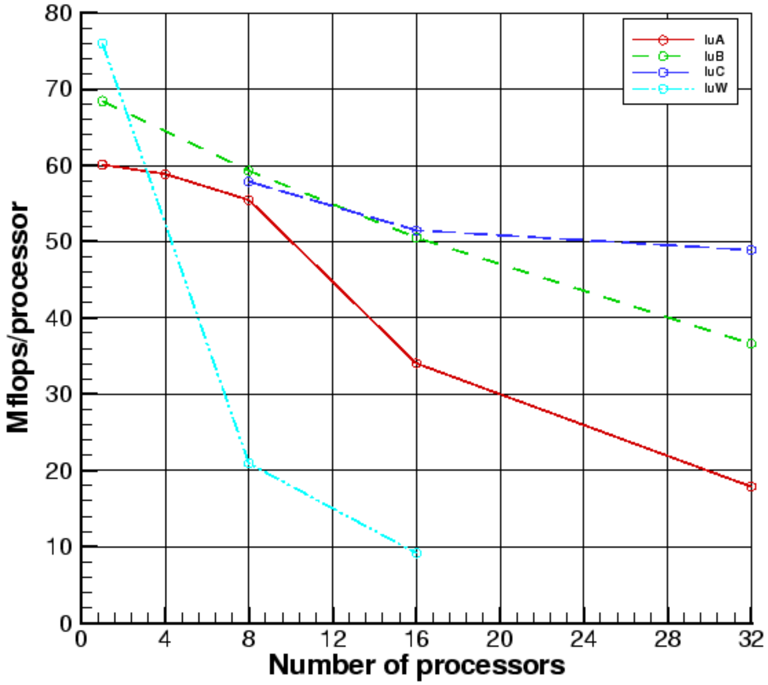
NAS Parallel Benchmarks (NPB v2.3):

- Standard benchmark for CFD applications on large computers.
- 4 sizes for each benchmark: Classes W, A, B and C.
  - » Class W: Workstation class (small in size)
  - » Class A, B, C: Supercomputer class (C being largest)

Benchmark Code	Class A	Class B	Class C
Embarrassingly parallel (EP)	2^{28}	2^{30}	2^{32}
Multigrid (MG)	256^3	256^3	512^3
Conjugate gradient (CG)	1.4×10^4	7.5×10^4	1.5×10^5
3D FFT PDE (FT)	$256^2 \times 128$	512×256^2	512^3
Integer sort (IS)	2^{23}	2^{25}	2^{27}
LU solver (LU)	64^3	102^3	162^3
Pentadiagonal solver (SP)	64^3	102^3	162^3
Block tridiagonal solver (BT)	64^3	102^3	162^3

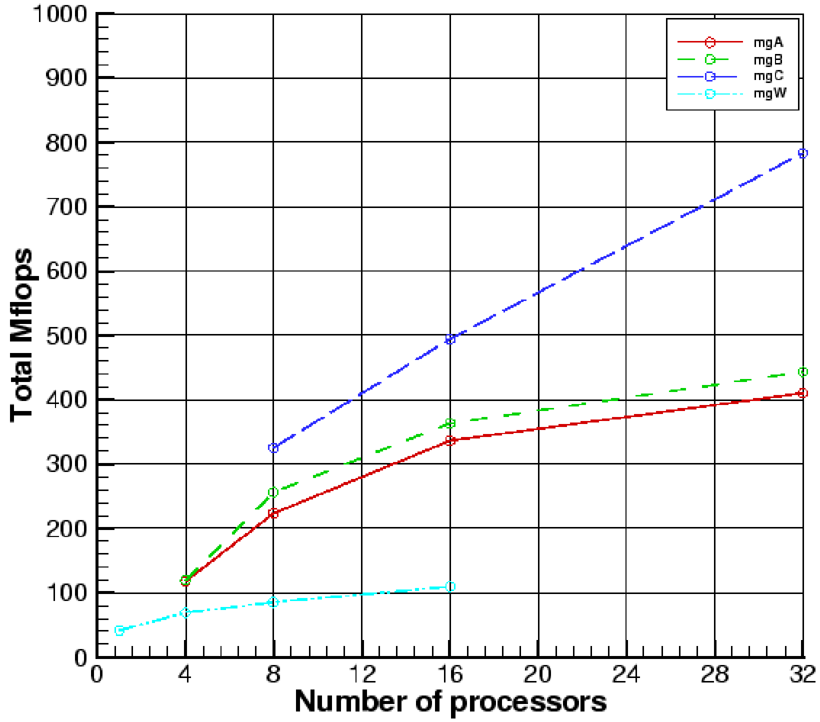
COCOA: Benchmarks

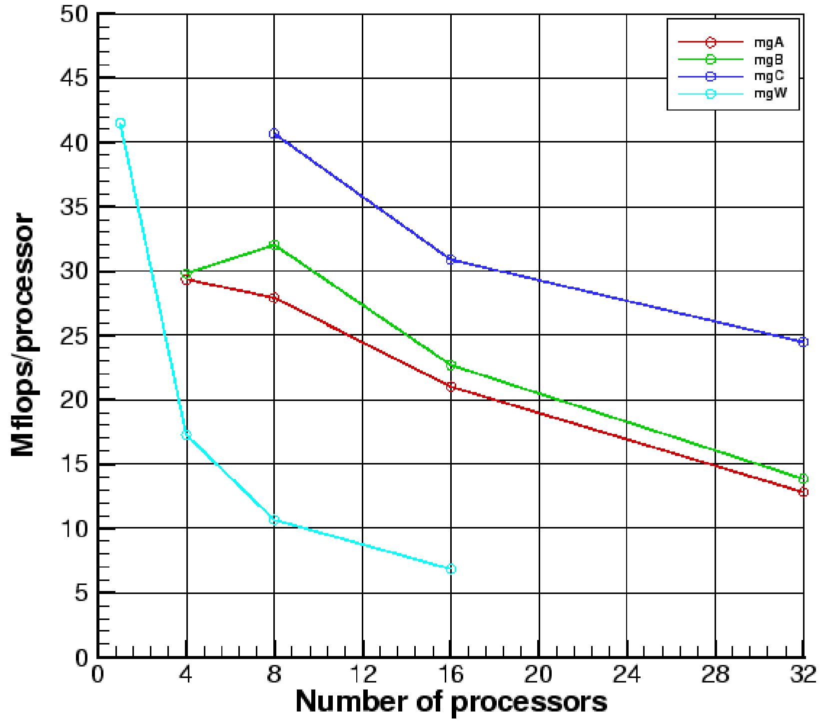




NAS Parallel Benchmark on COCOA*: LU solver* (LU) test

COCOA: Benchmarks

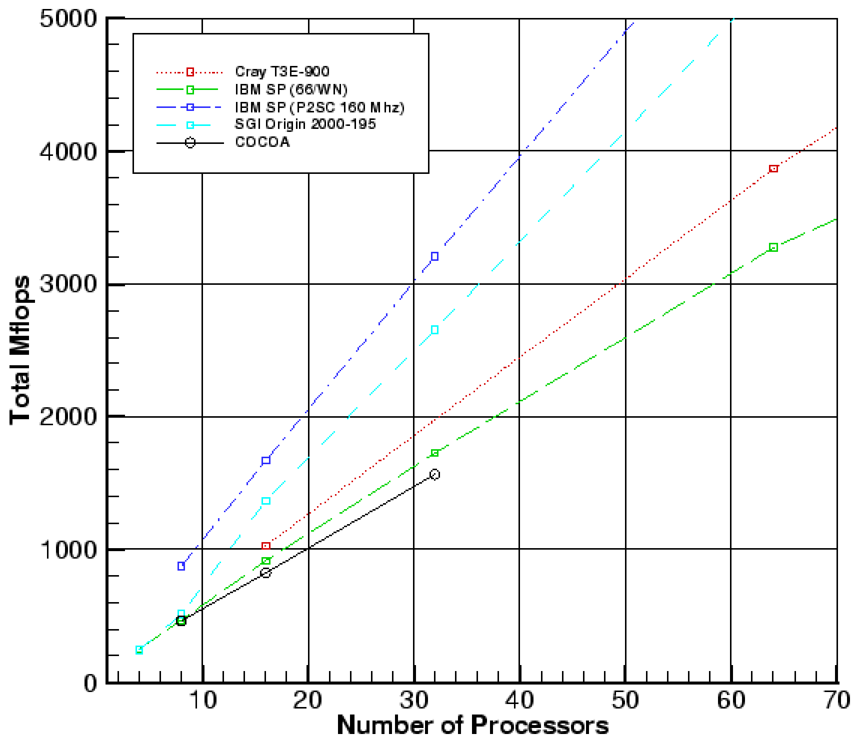


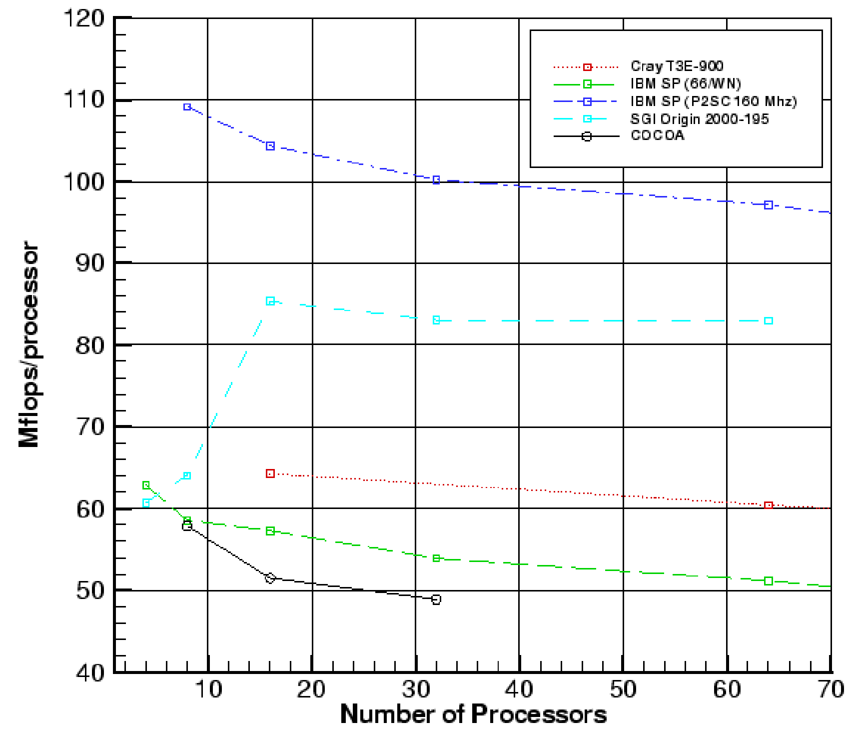


NAS Parallel Benchmark on COCOA*: Multigrid* (MG) test

COCOA: Benchmarks

Aerospace Engineering





LU solver (LU) test: Comparison with other machines

Post Processing and Visualization

Aerospace Engineering

- Since TECPLOT was the primary visualization software available at hand, a utility *toTecplot* was written in **C** to convert the restart data (.rst, in binary format) from PUMA to a TECPLOT output file.
- Necessary, as PUMA computes the solution data at the cell centers, whereas TECPLOT requires it at the nodes.
- Functions to calculate *vorticity* and *dilatation* were added to the utility to facilitate in the visualization of unsteady phenomena like vortex shedding and wake propagation. => Non-trivial for unstructured grids.

Live CFD-Cam

Aerospace Engineering

- Entire post-processing and visualization phase were automated. PUMA had to be slightly modified to facilitate this.
- Several utilities and TECPLOT macros were written (e.g., tec2gif). A client-server package was designed to post-process the solution and send it to the web-page (all done using UNIX shell scripts!!).
- Has several advantages: one can come to know in advance if the solution seems to be diverging and take corrective action without wasting a lot of expensive computational resources.
- Unsteady flow can be visualized as and when the solution is being computed.
- Useful as a computational steering tool.

Live CFD-Cam

Penn State Aerospace Engineering

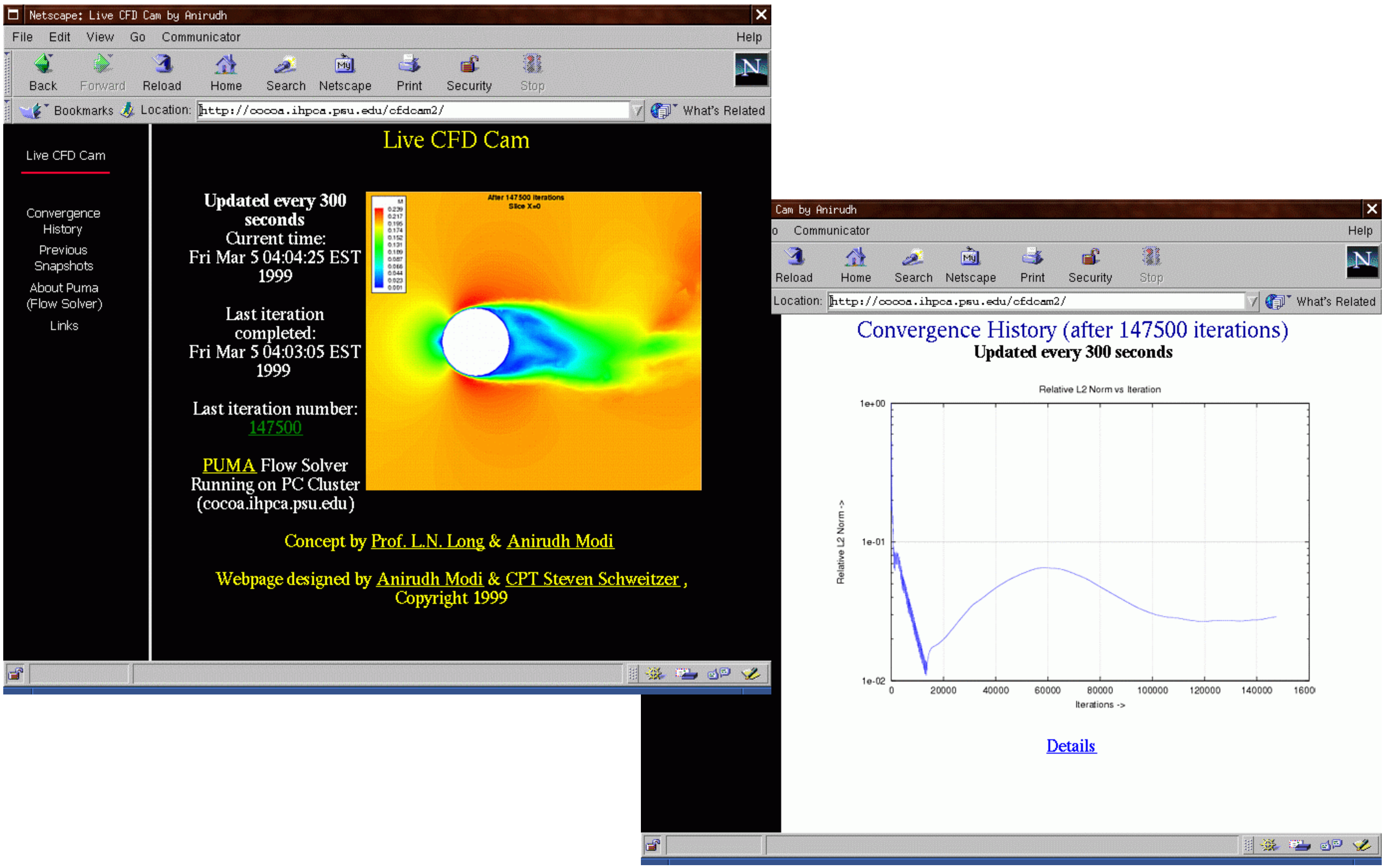
- Several CFD-Cams can run simultaneously! Live CFD-Cam is a fully configurable application.
- All the specific information for the run are read from an initialization file (*SERVER.conf*).

```
GridFile = grids/apache.sg.gps

ImageSize = 60

toTecplot_Options = 1 remove_surf.inp

Tecplot_Layout_Files = apache_M_nomesh.lay apache_CP_nomesh.lay

Destination_Machine = anirudh@cocoa.ihpca.psu.edu:~/public_html/cfdcam6

Destination_Directory = Apache

Destination_File_Name = ITER

Remote_Flag_File =anirudh@cocoa.ihpca.psu.edu:~/public_html/cfdcam6/CURRENT

Residual_File = apache.rsd
```

Sample *SERVER.conf* file

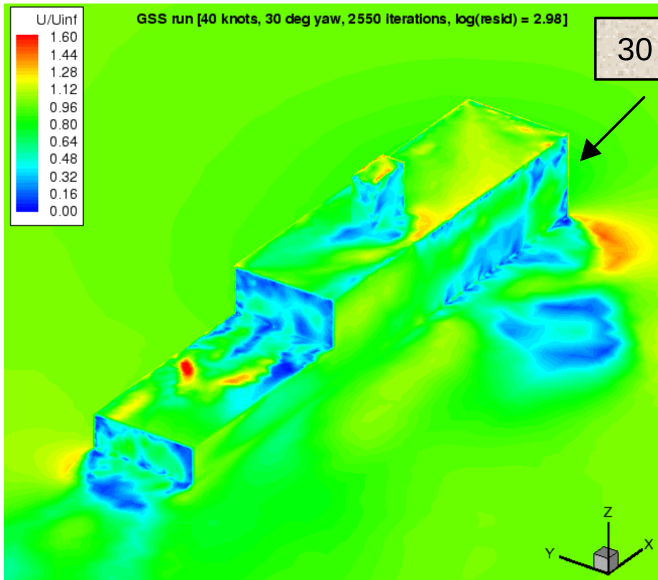
Live CFD-Cam

Aerospace Engineering

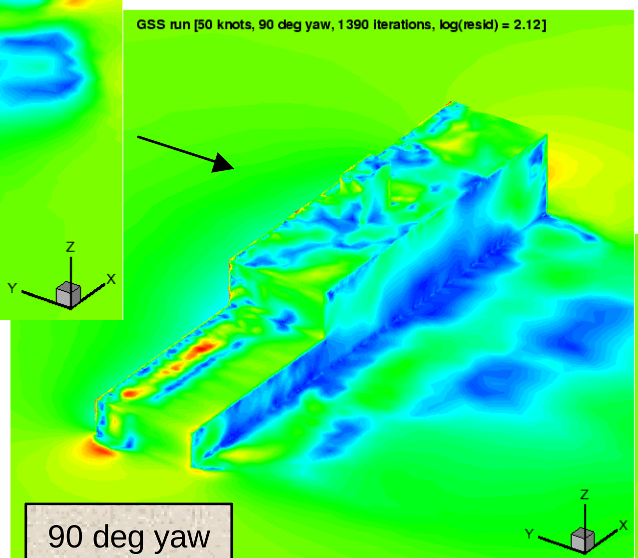


Results: Ship Configurations

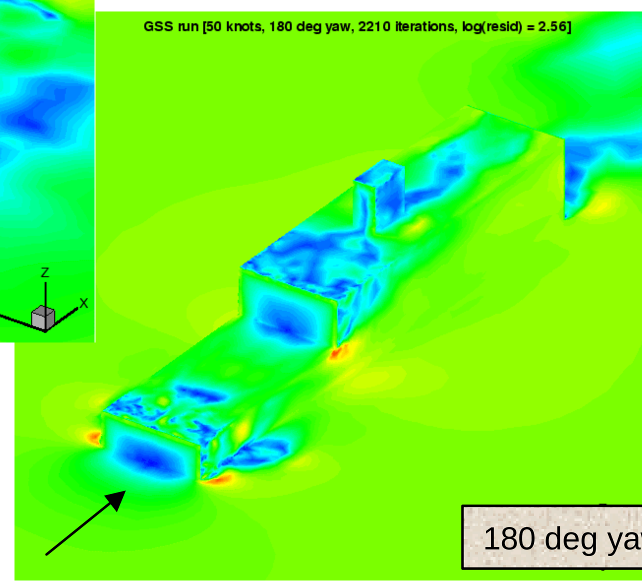
Aerospace Engineering



30 deg yaw



90 deg yaw



180 deg yaw

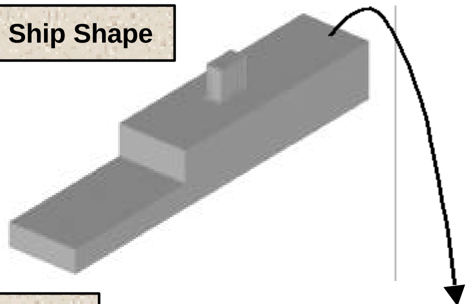
483,565 cells
984,024 faces
1.1 GB RAM

GSS inviscid runs

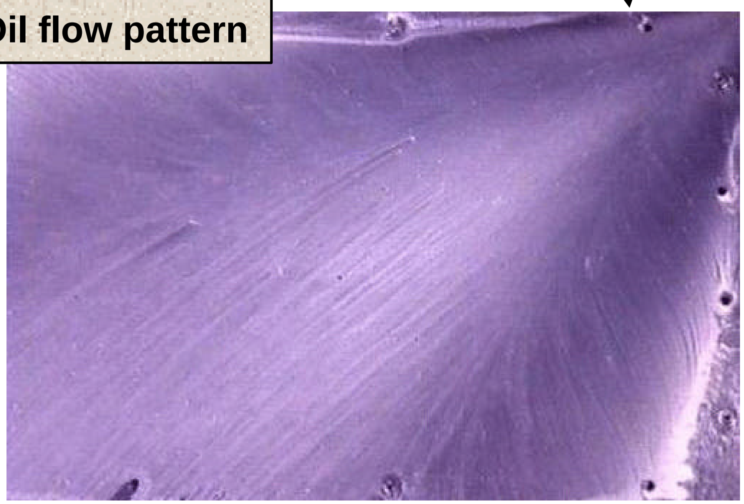
Results: Ship Configurations

PENNSTATE
Aerospace Engineering

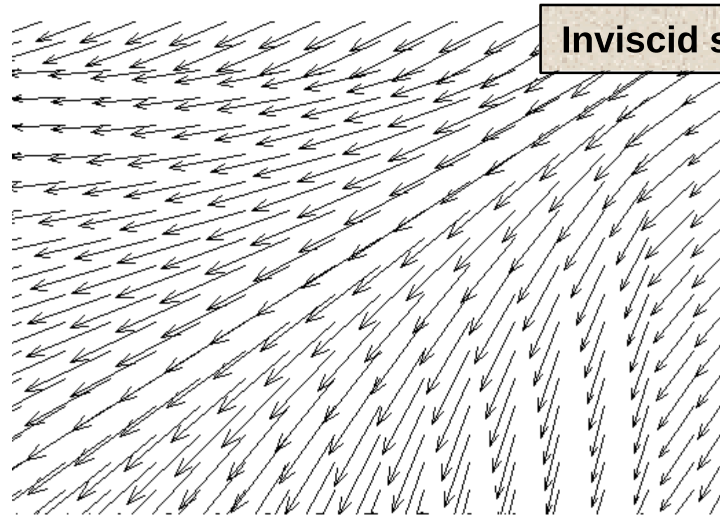
General Ship Shape



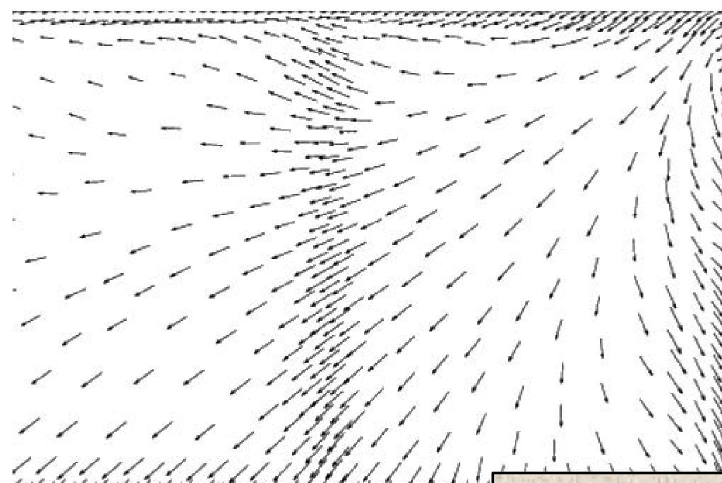
Oil flow pattern



Inviscid solution



Viscous solution



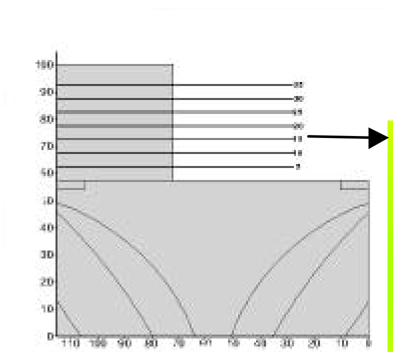
**Comparison with Experiments:
Front part of Bridge Deck
(30 deg yaw)**

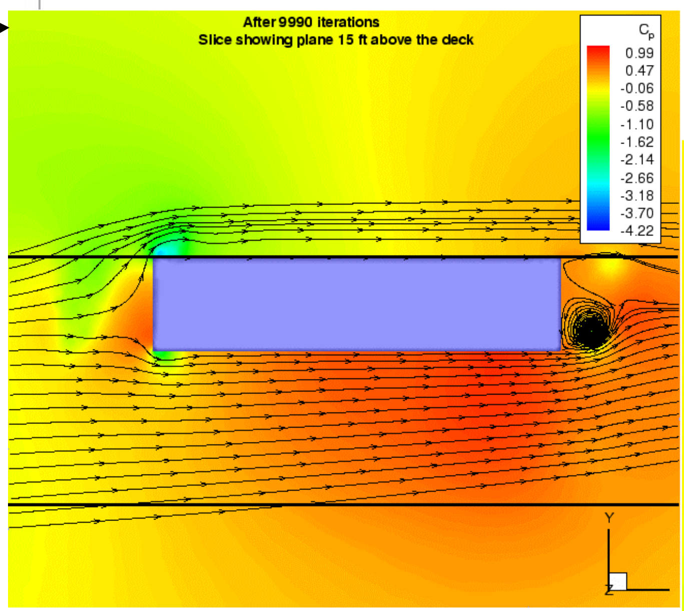
Results: Ship Configurations

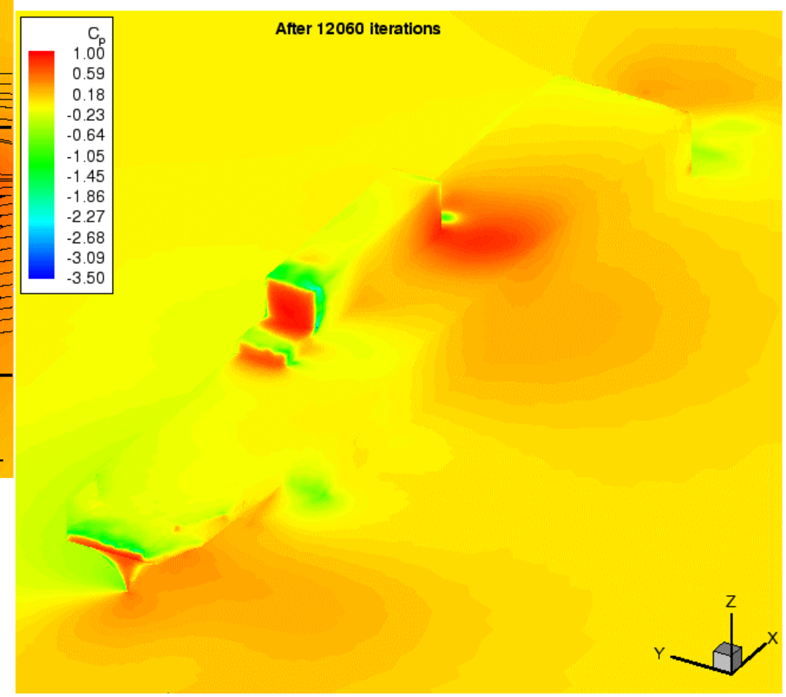
Aerospace Engineering

Flow conditions:
U=25 knots
β=5 deg

1,216,709 cells
2,460,303 faces
3.7 GB RAM



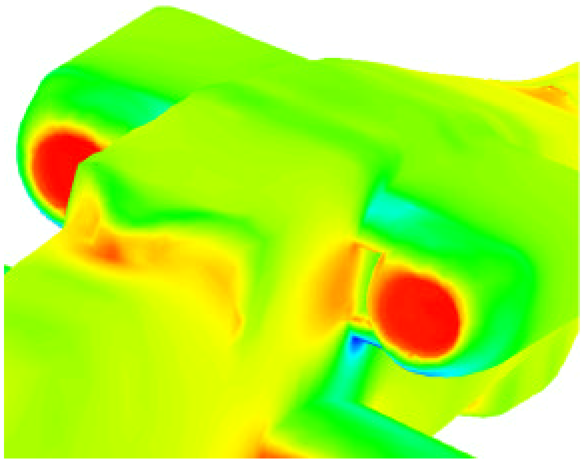




Landing Helicopter Aide (LHA)

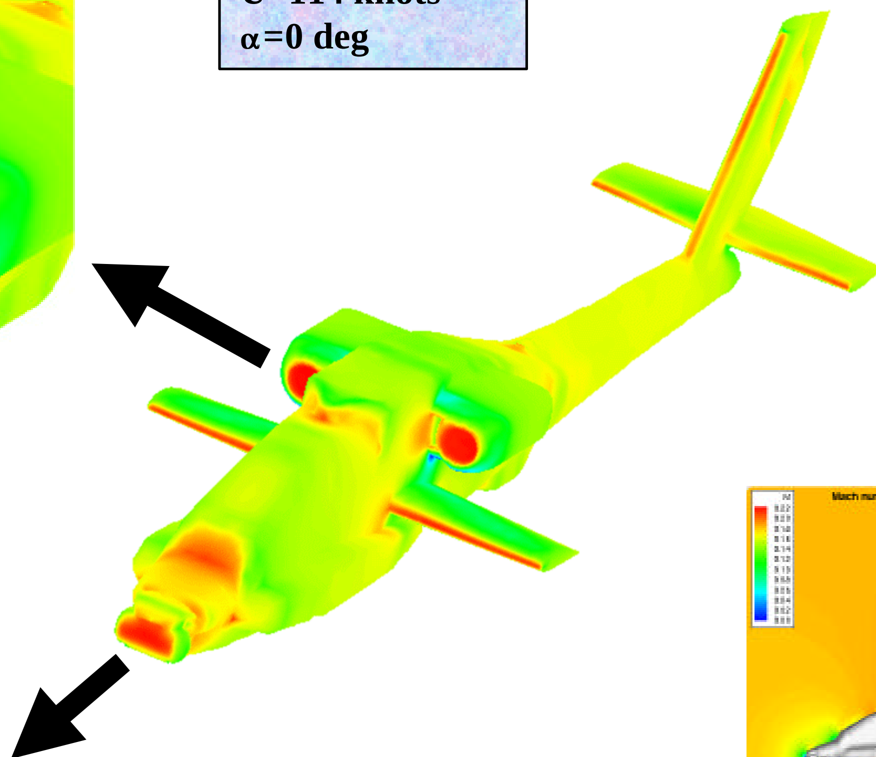
Results: Helicopter Configurations

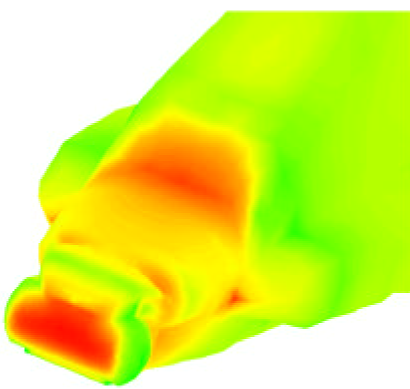
Aerospace Engineering



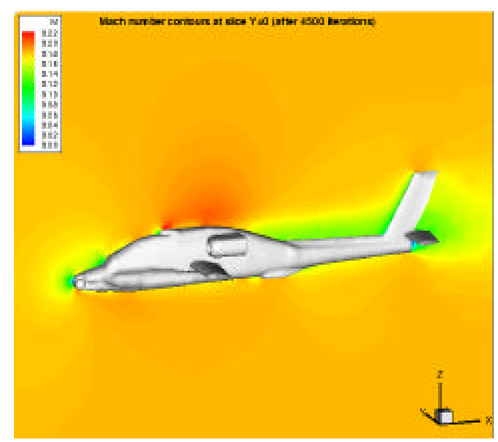
Flow conditions:
U=114 knots
α=0 deg

555,772 cells
1,125,596 faces
1.9 GB RAM





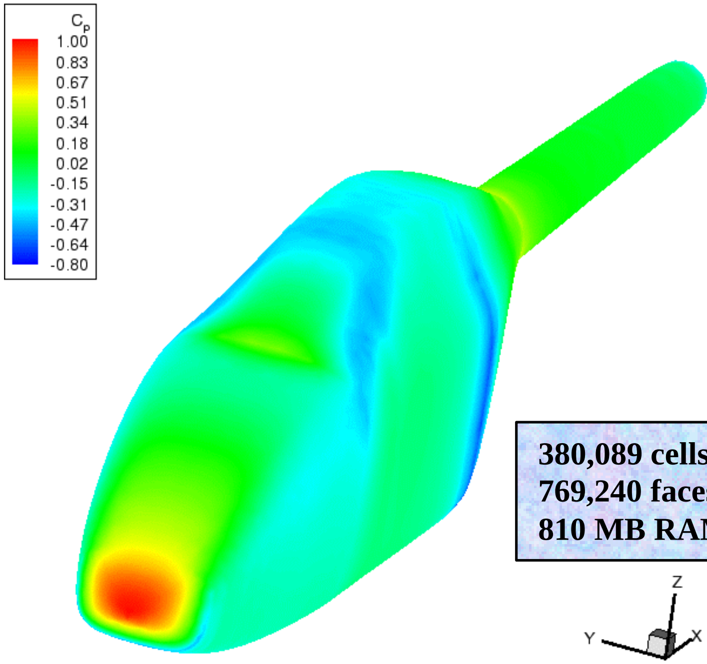
AH-64 Apache



Results: Helicopter Configurations

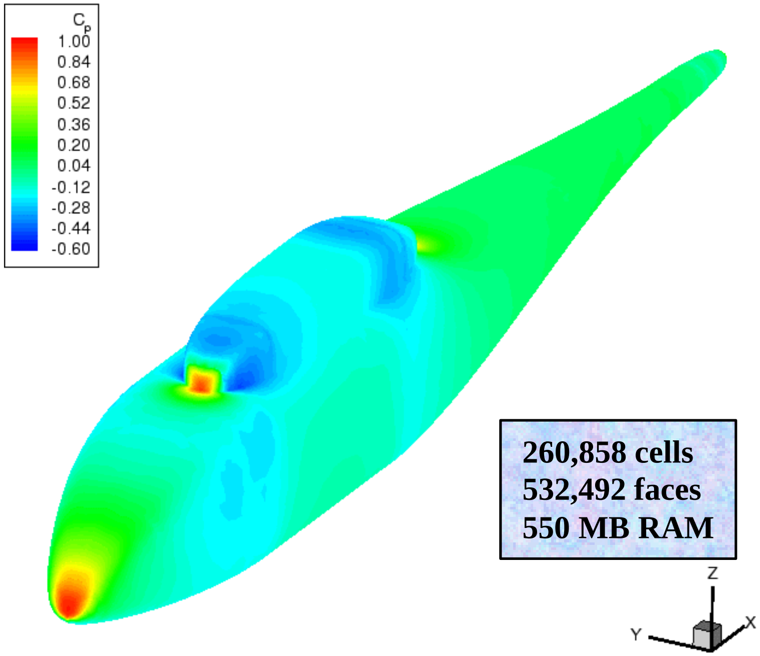
PENN STATE
Aerospace Engineering

Flow conditions:
U=114 knots
α=0 deg



380,089 cells
769,240 faces
810 MB RAM

Boeing General Fuselage



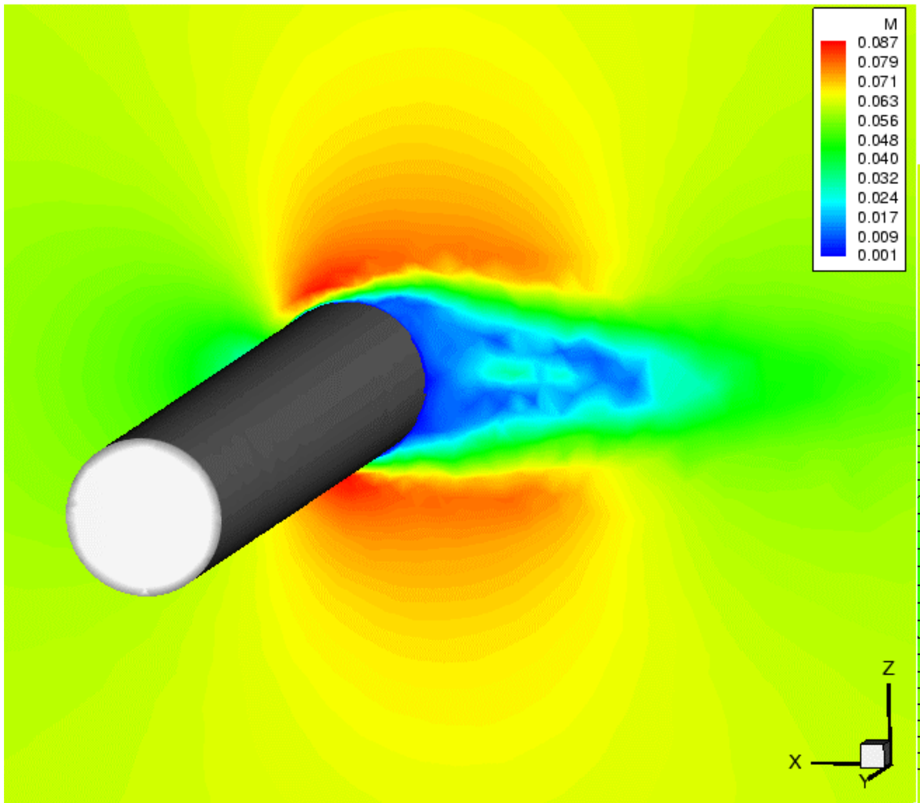
260,858 cells
532,492 faces
550 MB RAM

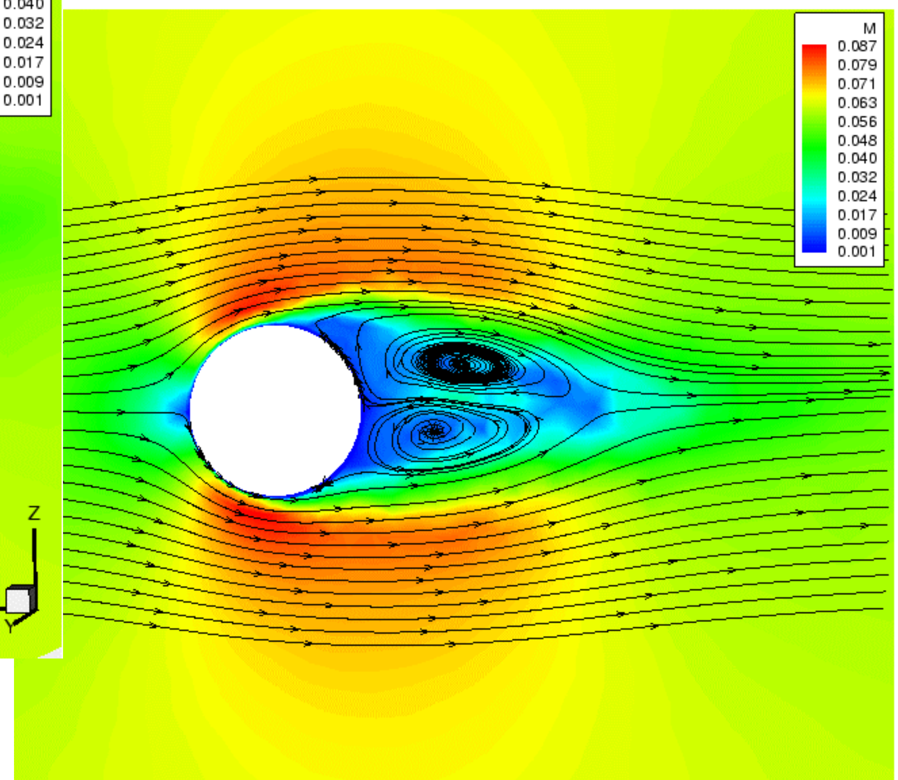
ROBIN fuselage

Results: Viscous Cylinder

Aerospace Engineering

Flow conditions:
U=41 knots (M=0.061)
α=0 deg
Re=1000

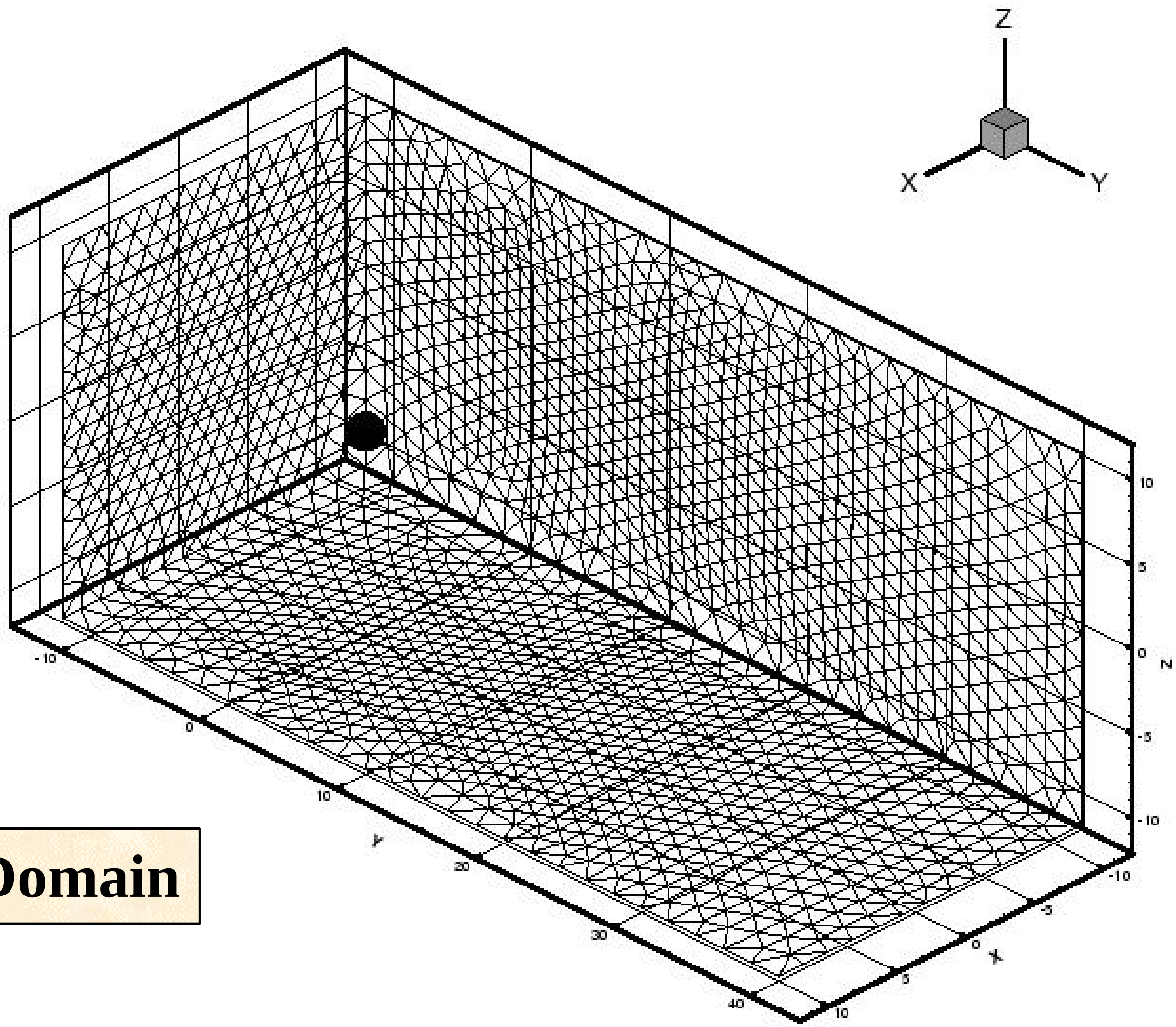




806,668 cells
1,620,576 faces
2.4 GB RAM

Results: Viscous Sphere

PENNSTATE
Aerospace Engineering

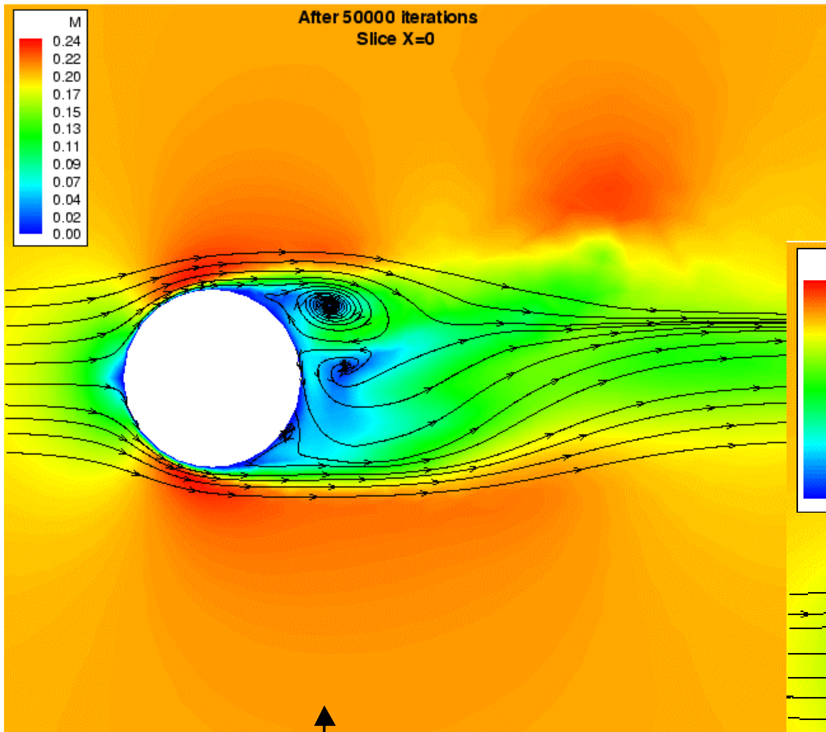


Domain

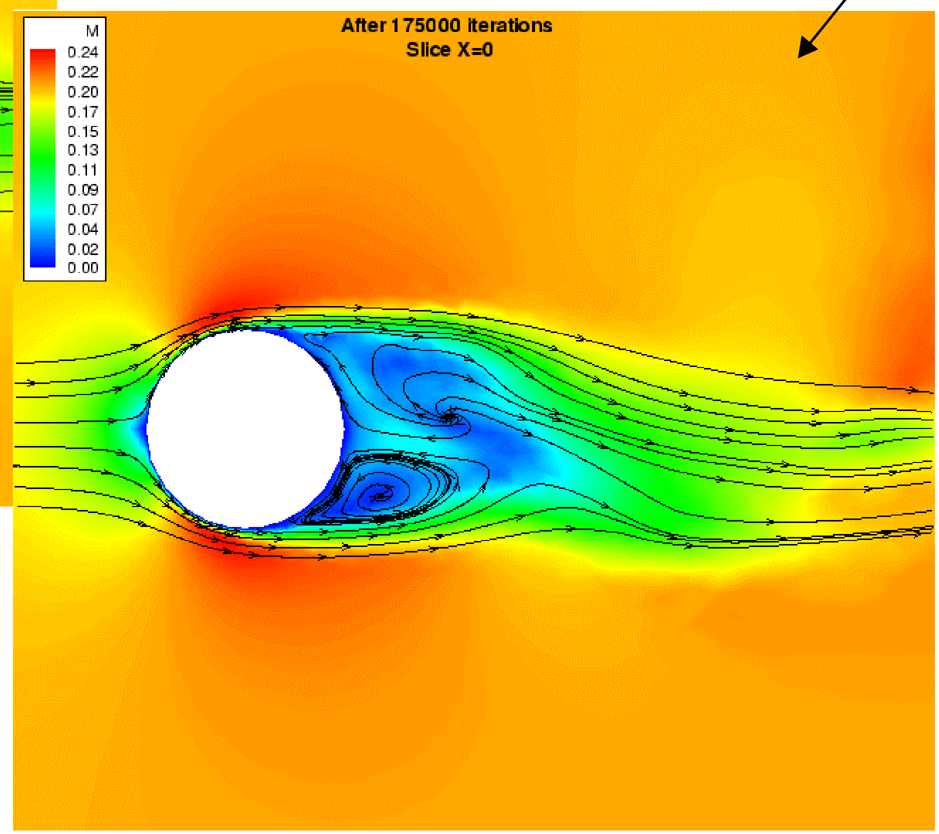
Results: Viscous Sphere

Aerospace Engineering

<u>Flow conditions:</u>
U=133 knots (M=0.20)
α=0 deg
Re=1000



t=0.0

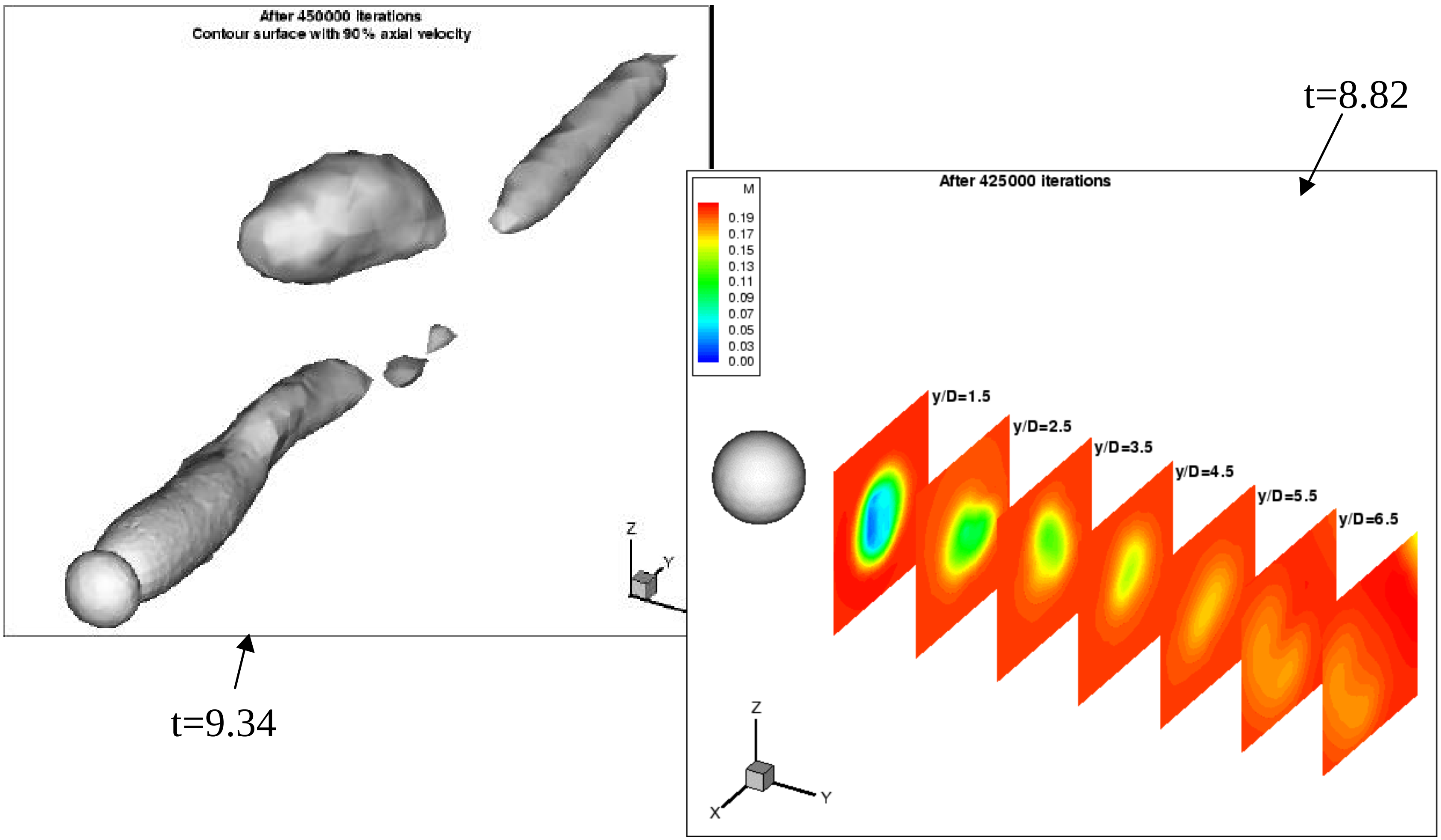


t=2.75

306,596 cells
617,665 faces
600 MB RAM

Results: Viscous Sphere

Aerospace Engineering

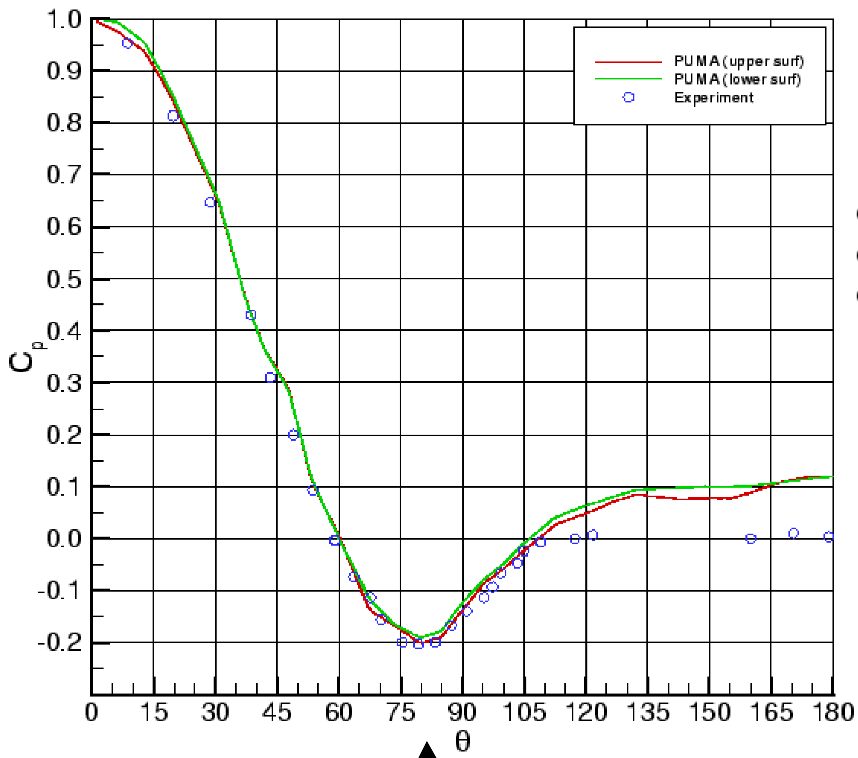


t=8.82

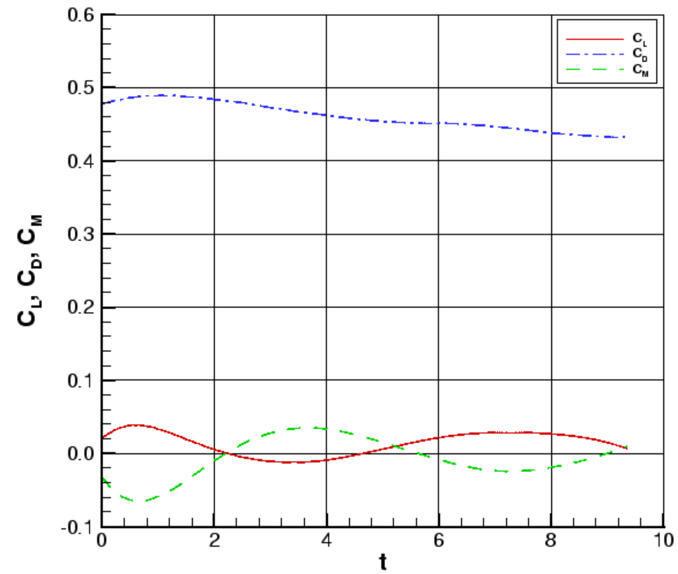
t=9.34

Results: Viscous Sphere

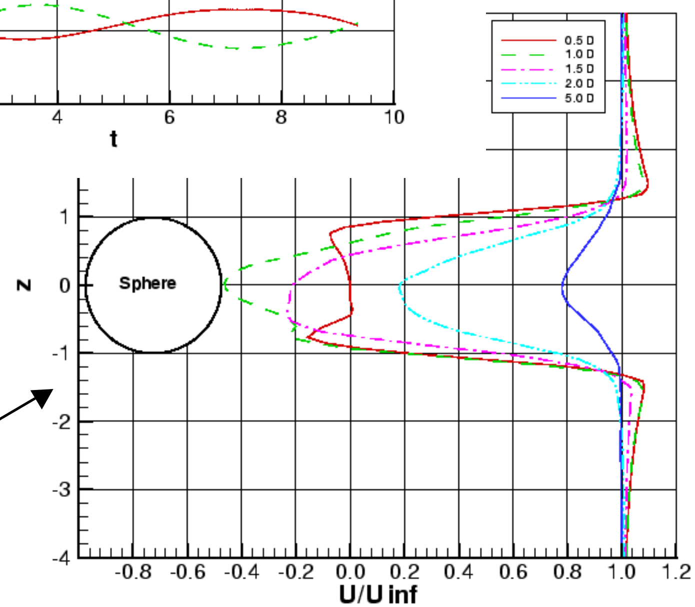
Aerospace Engineering



Time averaged

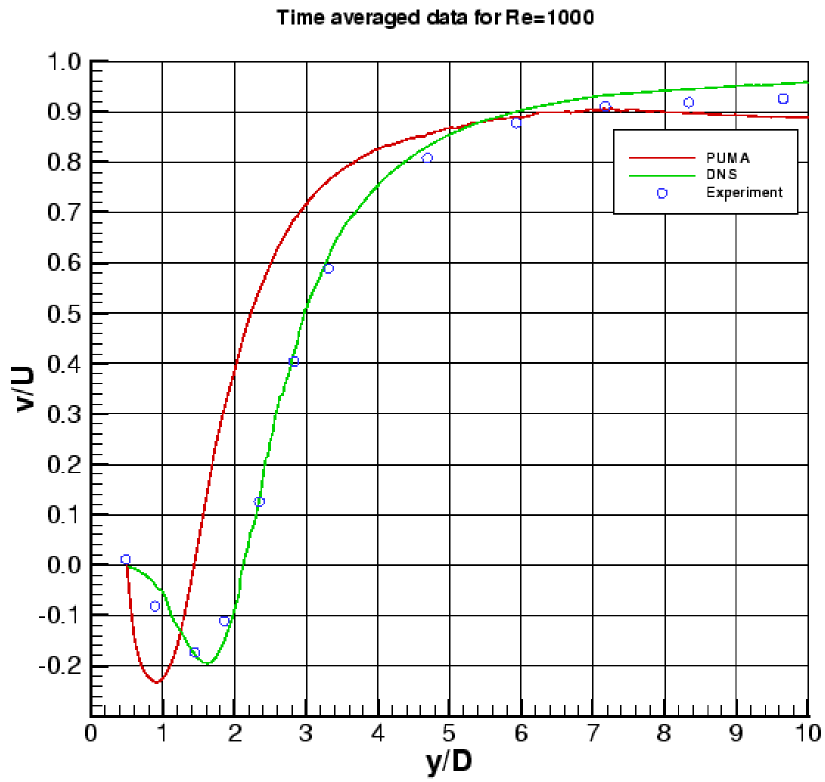


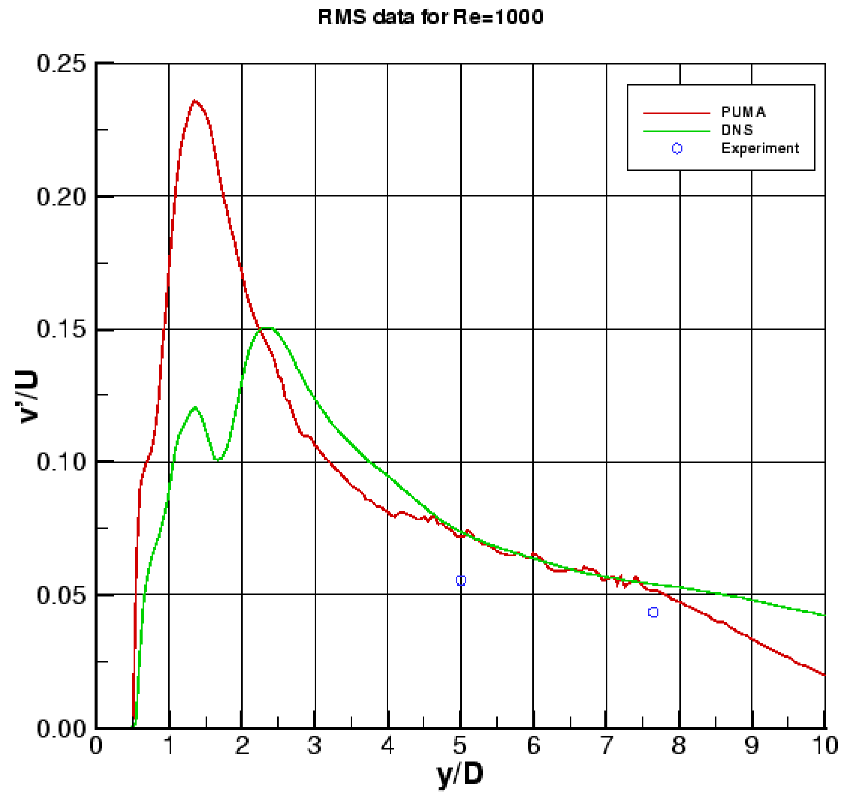
Time history



t=8.79

Results: Viscous Sphere

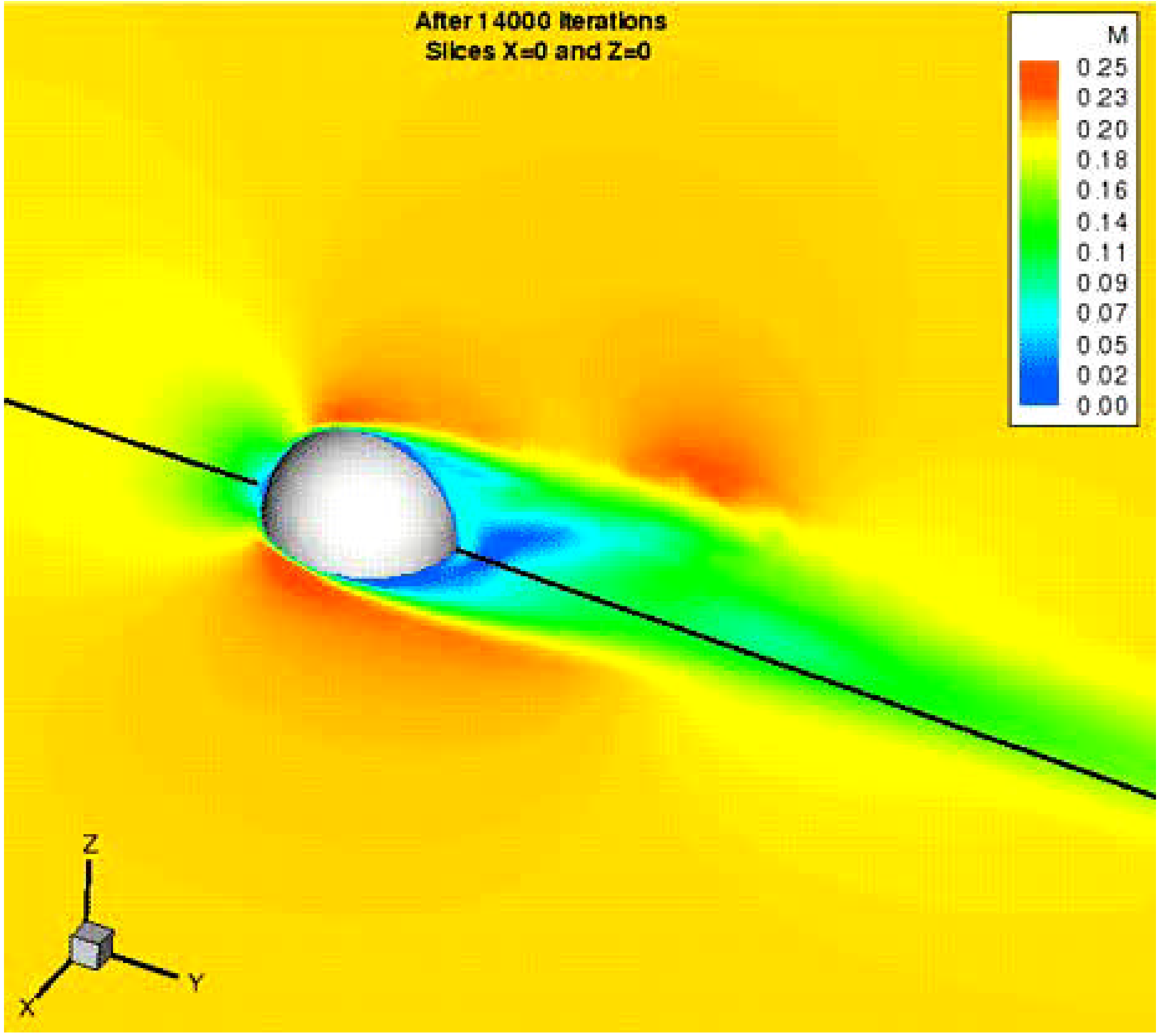




Time averaged

Results: Viscous Sphere

Aerospace Engineering



Sample Movie

Conclusions

- A complete, fast and efficient unstructured grid based flow solution around several complex geometries has been demonstrated.
- The objective to achieve this at a very affordable cost using inexpensive departmental level supercomputing resources like COCOA, has been fulfilled.
- GSS and sphere results compare well with experimental data.
- PUMA has proven capable of solving unsteady separated flow around complex geometries.

Conclusions

Aerospace Engineering

- Using VGRID, COCOA, PUMA, and Live CFD-cam, incredible turn-around times for several large problems involving complex geometries has been achieved.
- COCOA was also found to have good scalability with most of the MPI applications used, although it is not ideal for communication-intensive applications (high latency).

Future Work

- Pre-Conditioning
- k-exact
- NLDE