High-End Computing for Incompressible Flows

Dochan Kwak and Cetin Kiris
NASA Ames Research Center
Moffett Field, CA
dkwak@mail.arc.nasa.gov

First MIT Conference on Computational Fluid and Solid Mechanics
MIT, Cambridge MA02139
June 12-14, 2001

Outline of Talk

- INTRODUCTION
  - Status
  - Major Drivers of the Current Work
- OBJECTIVE
- SOLUTION METHODS
  - Formulation / Approach
  - Summary of Solver Development
  - Current Challenges
- HEC APPLICATIONS
  - Parallel Implementation
  - Application to SSME Turbopump
- DISCUSSION

Status from Applications Point of View

- Applications to Real-World Problems
  - N-S solution of full configuration was a big goal in the 80s
  - Numerical procedures and computing hardware have been advanced enabling simulation of complex configurations
- Some Examples of Successful Applications
  - Components of liquid rocket engine
  - Hydrodynamics (Submarines, propellers, ...)
  - Ground vehicles (automobile aerodynamics, internal flows...)
  - Biofluid problems (artificial heart, lung, ...)
  - Some Earth Science problems
- Current Challenges
  - For integrated systems analysis, computing requirement is very large
    ⇒ Analysis part is still limited to low fidelity approach
  - For high-fidelity analysis, especially involving unsteady flow, long turn-around time is often a bottle neck ⇒ Acceleration of solution time is required

Major Drivers of Current Work

- To provide computational tools as an economical option for developing future space transportation systems (i.e. RLV subsystems development)

Impact on component design ⇒ Rapid turn-around of high-fidelity analysis
Increase durability/safety ⇒ Accurate quantification of flow (i.e. prediction of low-induced vibration)

Impact on system performance ⇒ More complete systems analysis using high-fidelity tools

- Target

Turbo-pump component analysis ⇒ Entire sub-systems simulation

Computing requirement is large:
⇒ The goal is to achieve 1000 times speed up over what was possible in 1992

Objectives

- Mission of Conference
  To bring together Industry and Academia & (Government)
  To nurture the next generation in computational mechanics

- Objective of the Current Talk
  To discuss some current issues in large scale computing for mission-oriented tasks

Viscous Incompressible Flow

- Formulation
  Can be viewed as a limiting case of compressible flow where the flow speed is insignificant compared to the speed of sound (Preconditioned compressible N-S eq.)
  ⇒ Artificial compressibility approach
  - Artificial Compressibility Method (Chorin, 1967; Temam, 1977)
    ADI Scheme, FD (Central+diss) (Beam& Warming, 1978; Briley-McDonald, 1977)
    LU-SGS, FV (Central+diss) (Yoon and Jameson, 1987...)
    Line Relaxation, FD (Upwind) (... MacCormack, 1985)
    GMRES, FD (Upwind)

  Or truly incompressible
  ⇒ Pressure projection approach
  - MAC (Harlow and Welch, 1965)
  - Fractional Step Method (Chorin, 1968; Yanencko, 1971; Marchuck, 1975....)
  - SIMPLE type Pressure Iteration (Caretto et al., 1972; Patanka & spalding, 1972...)

  ⇒ Use derived variables
  - Vorticity-Velocity (Fasel, 1976; Dennis et al., 1979; Hafez et al., 1988
  - Stream function-vorticity

Artificial Compressibility Method

- Formulation

  $$\frac{1}{\beta}\frac{\partial p}{\partial t}+\frac{\partial u_i}{\partial x_i}=0$$

  - Introduces hyperbolic behavior into pressure field.

    Speed of pressure wave depends on the artificial compressibility parameter, β.
  - The equations are to be marched in a time like fashion until the divergence of velocity converges to zero.
    ⇒ Relaxes incompressibility requirement.

    Time variable during this process does not represent physical time step.

  For time-accurate solutions
  - Iterate the equations in pseudo-time level for each time step until incompressibility condition is satisfied.

    ⇒ Efficient sub-iteration is the key for success

Artificial Compressibility Method (INS3D-UP)

- Time accuracy is achieved by subiteration
  - Discretize the time term in momentum equations using second-order three-point backward-difference formula

    $$\frac{3q^{n+1}-4q^{n}+q^{n-1}}{2\Delta t}=-(rhs)^{n+1}$$
  - Introduce a pseudo-time level and artificial compressibility,
  - Iterate the equations in pseudo-time for each time step until incompressibility condition is satisfied.

    $$\frac{1}{\Delta\tau}(p^{n+1,m+1}-p^{n+1,m})=-\beta q^{n+1,m+1}$$

    $$\frac{1.5}{\Delta\tau}(q^{n+1,m+1}-q^{n+1,m})=-(rhs)^{n+1,m+1}-\frac{3q^{n+1,m}-4q^{n}+q^{n-1}}{2\Delta t}$$
- Code performance
  - Computing time : 50-120 ms/grid point/iteration
  - Memory usage: **Line-relaxation** 45 words/grid point
    GMRES-ILU(0) 220 words/grid point

Pressure Projection Method(INS3D-FS)

- Approach in generalized coordinates
  - Finite volume discretization
  - Accurate treatment of geometric quantities
  - Dependent variables - pressure and volume fluxes
  - Implicit time integration
  - Fractional step procedure
    Solve auxiliary velocity field first,
    then enforce incompressibility condition by solving a Poisson equation for pressure.
- Code performance
  - Computing time : 80 ms/grid point/iteration
  - Memory usage: 70 words/grid point

Pressure Projection Method

- Fractional-step
  - Solve for the auxiliary velocity field, using implicit predictor step:

$$\frac{1}{\Delta t}(u_i^* - u_i^n) = -\nabla p^n + h(u_i^n)$$

  - The velocity field at time level (n+1) is obtained by using a correction step:

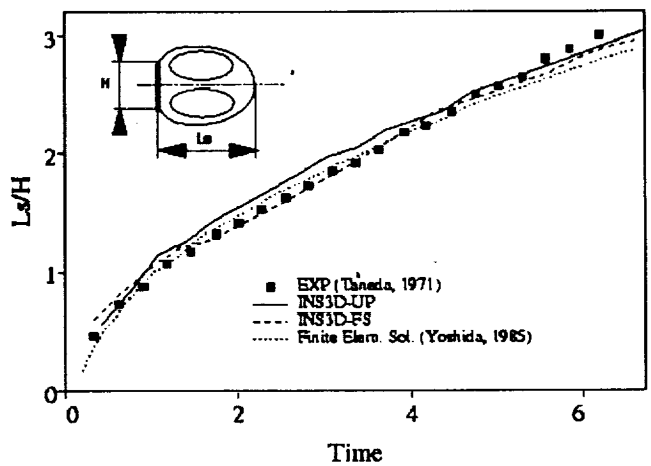
$$\frac{2}{\Delta t}(u_i^{n+1} - u_i^*) = -\nabla p^{n+1} + h(u_i^{n+1}) - \nabla p^* + h(u_i^*)$$

  - The incompressibility condition is enforced by using a Poisson equation for pressure ($p' = p^{n+1} - p^*$)

$$\nabla^2 p' = \frac{2}{\Delta t}\nabla \cdot u^*$$

Impulsively Started Flat Plate at 90°

- Time History of Stagnation Point



History of INS3D Development

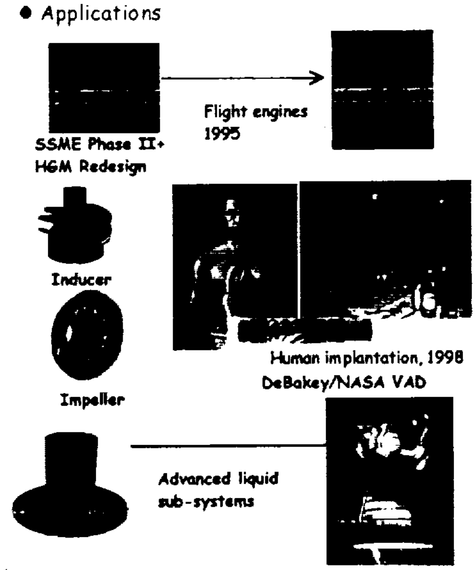
- Code

1982-1987 Original version
(Kwak, Chang)

1988-1997 INS3D-UP
(Rogers, Kiris, Kwak)
INS3D-LU
(Yoon, Kwak)
INS3D-FS
(Rosenfeld, Kiris, Kwak)

1998-Present Combined version
(Kiris, Kwak)

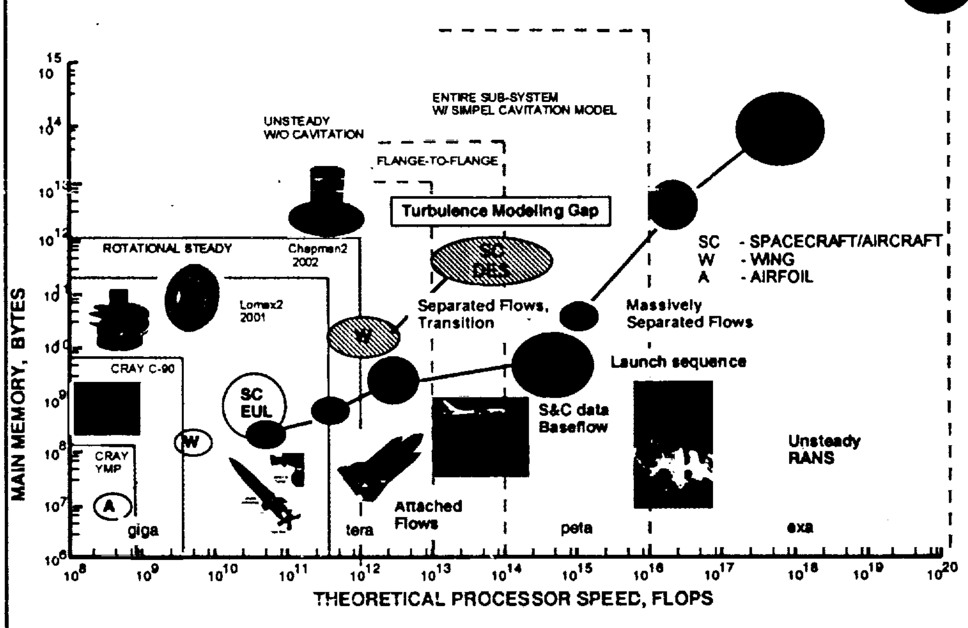
- Applications



Current Challenges

- Challenges where improvements are needed
  - Time-integration scheme, convergence
  - Moving grid system, zonal connectivity
  - Parallel coding and scalability
- As the computing resources changed to parallel and distributed platforms, computer science aspects become important such as
  - Scalability (algorithmic & implementation)
  - Portability, transparent coding etc.
- Computing resources
  - "Grid" computing will provide new computing resources for problem solving environment
  - High-fidelity flow analysis is likely to be performed using "super node" which is largely based on parallel architecture

Supercomputing Memory and CPU Requirements

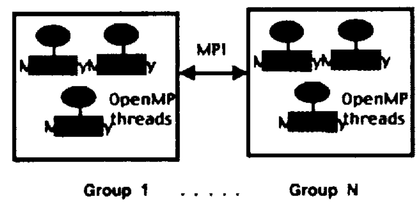


HEC TURB FIDELITY

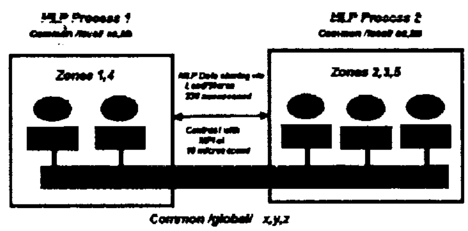
Parallel Implementation of INS3D

- INS3D-MPI
  (coarse grain)
  T. Faulkner & J. Dacles

- INS3D-MPI / Open MP
  MPI (coarse grain) + OpenMP (fine grain)
  Implemented using CAPO/CAPT tools
  H. Jin & C. Kiris

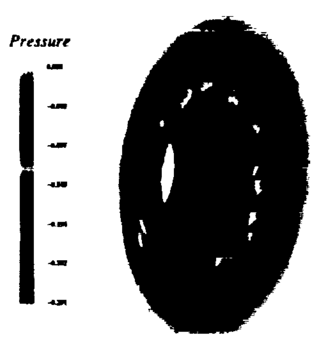


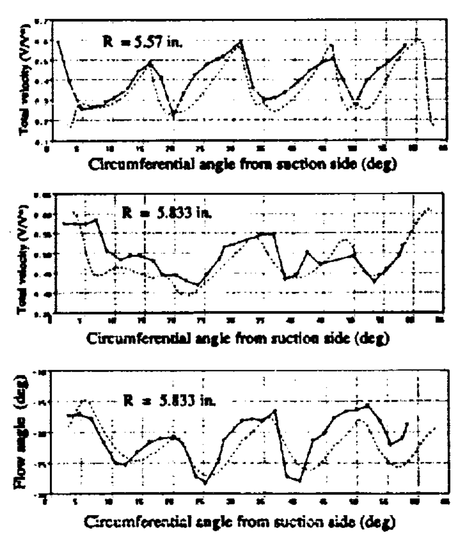
- INS3D-MLP
  C. Kiris



Parallel Implementation of INS3D

- Previous Work (SSME Impeller)





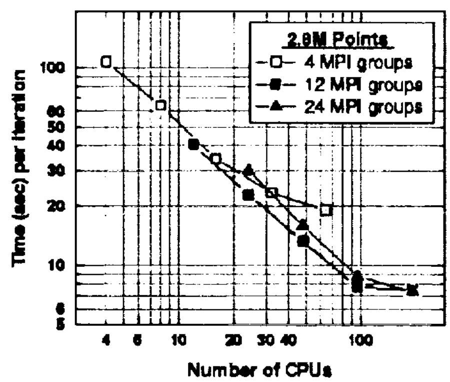
Parallel Implementation of INS3D

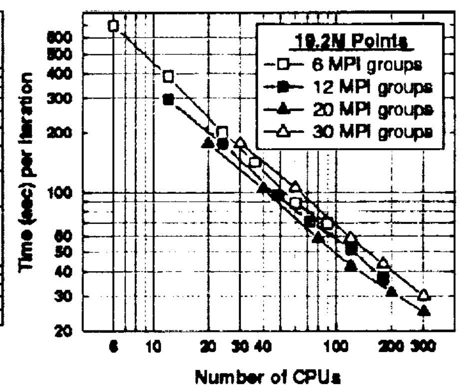
MPI coarse grain + OpenMP fine grain

TEST CASE : SSME Impeller

24 zones / 2.8 Million points

60 zones / 19.2 Million points



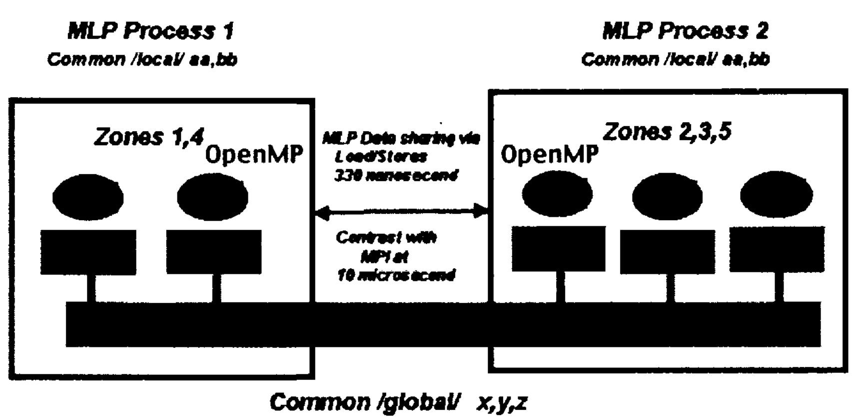


Parallel Implementation of INS3D

Multi-Level Parallelism (MLP)

INS3D-MLP : MLP routines + OpenMP

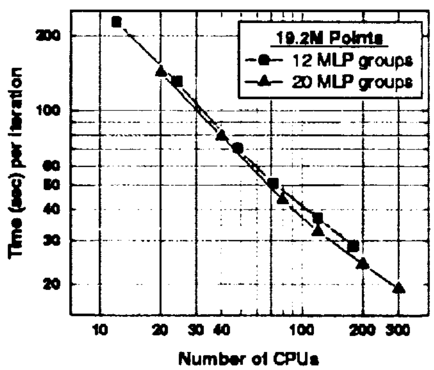
Shared Memory MLP Organization for Origin 2000

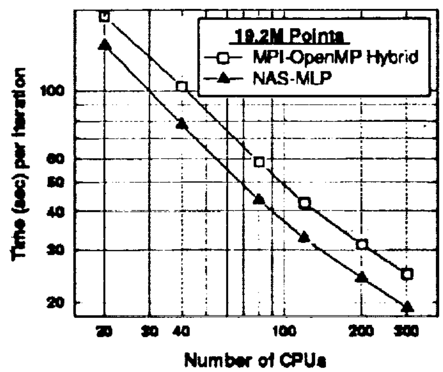


Parallel Implementation of INS3D

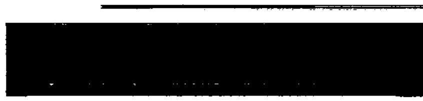
INS3D-MLP (NAS MLP no pin-to-node) / OpenMP

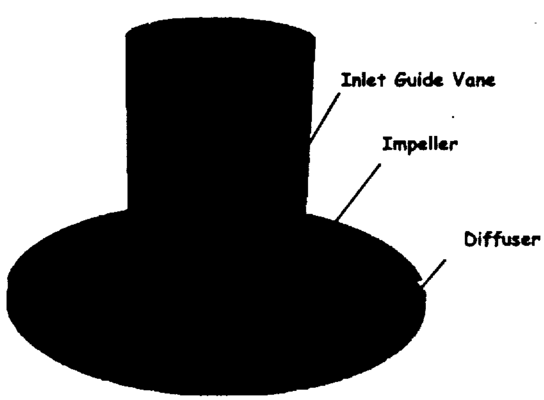
TEST CASE : SSME Impeller
60 zones / 19.2 Million points

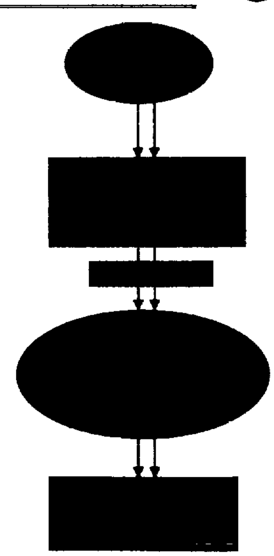




Space Shuttle Main Engine Turbopump

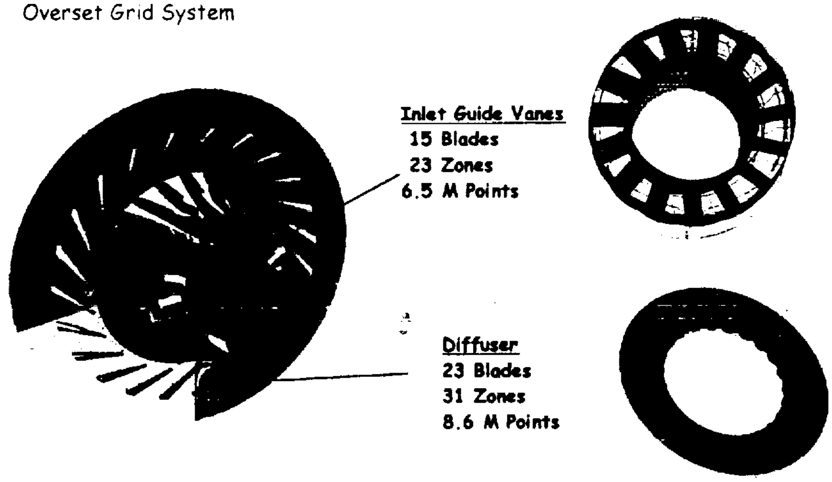




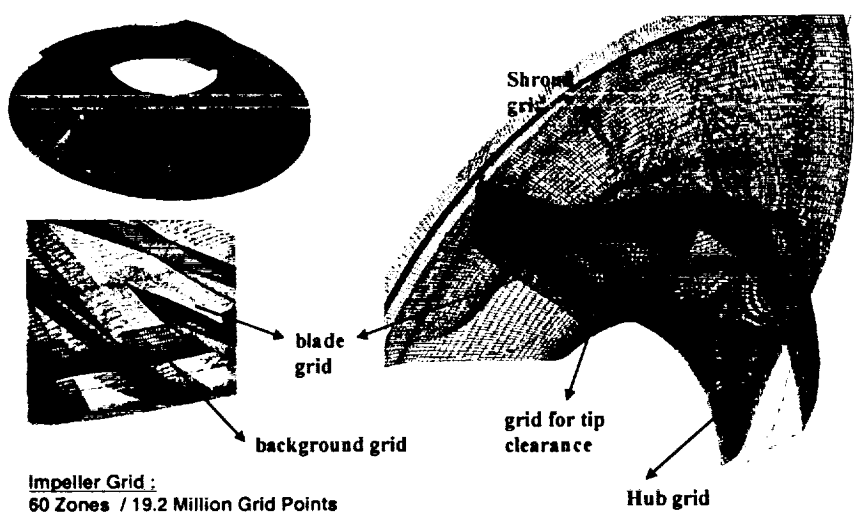


Space Shuttle Main Engine Turbopump

Overset Grid System



Shuttle Upgrade SSME-rig1

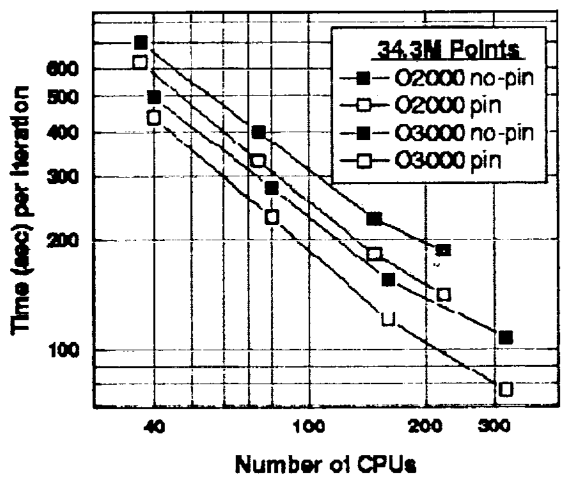


Impeller Grid :
60 Zones / 19.2 Million Grid Points
Smallest zone : 75K /Largest zone : 996K
Less than 192 orphan points.

Parallel Implementation of INS3D

INS3D-MLP / 40 Groups

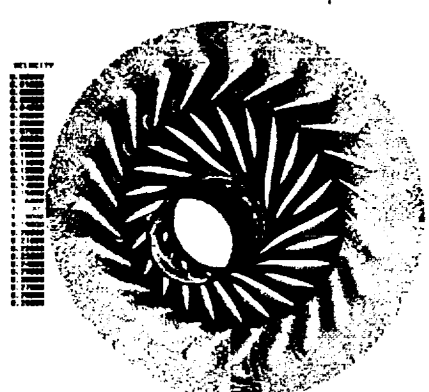
RLV 2nd Gen Turbo pump
114 Zones / 34.3 M grid points



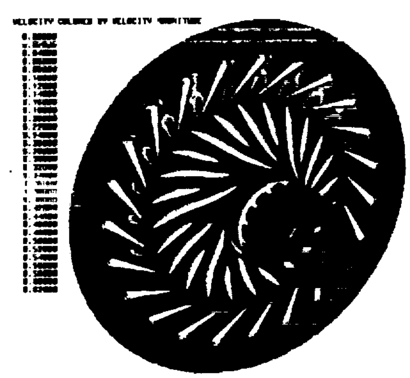
Per processor Mflop is between 60-70. Code optimization for cache based platforms is currently underway. Target Mflops is to reach 120 per processor. Increasing number of OpenMP threads is also the main objective for this effort.

SSME-rig1/ Overset Grid System

Initial Start: first time step



Time Step 5



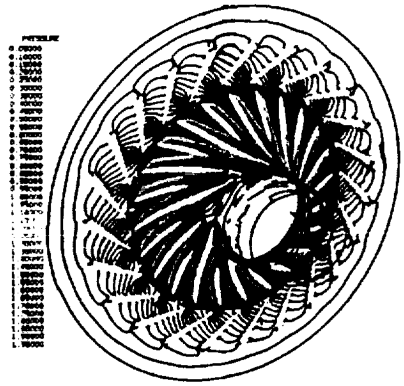
Velocity colored by magnitude

Impeller started at 10% of design speed

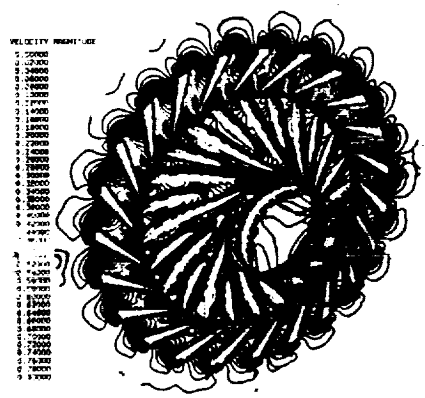
Impeller rotated 2.25 degrees

SSME-rig1 / Initial start

Time Step 7: Impeller rotated 3-degrees at 30% of design speed



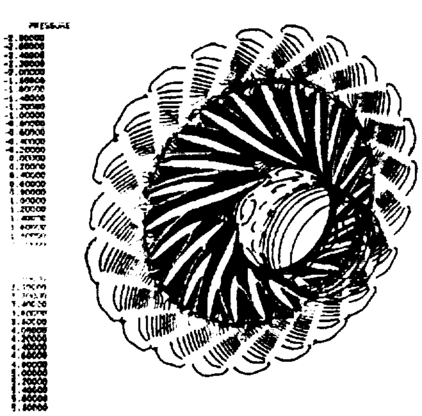
PRESSURE



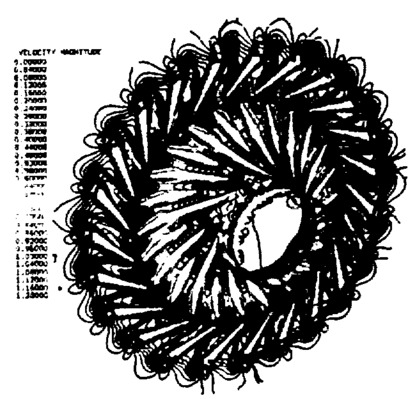
VELOCITY MAGNITUDE

SSME-rig1 / Initial start

Time Step 18: Impeller rotated 8-degrees at 100% of design speed

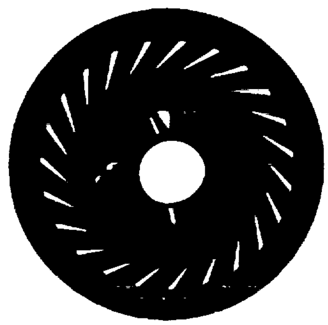


PRESSURE



VELOCITY MAGNITUDE

Summary of SSME Turbopump Simulation



Problem size:
- 34.3 Million Points
- 800 physical time steps in one rotation

CPU requirement:
- One physical time-step requires less than 20 minutes wall time with 80 CPU's on Origin 2000.
- One complete rotation requires one-week wall time with 80 CPUs.

I/O:
- Currently I/O is through one processor. Timing will be improved with parallel I/O since time-accurate computations are I/O intensive

Parallel Efficiency:
MLP/OpenMP version requires 19-25% less computer time than MPI/OpenMP version.
Pin-to-node for MLP version reduces computer time by 40%

Discussion on Numerical Procedures

- Finite Difference
  - Based on Taylor series expansion ⇒ Requires smooth grid
  - Need special care for grid singularity
  - Generally easier to use fine grids near wall at high Reynolds number
- Finite Volume
  - Formulation is more physical (conservation of properties)
  - Viscous flux calculation is not as straightforward
  - Difficult to implement higher order schemes
- In actual implementation, however,
  - These differences become unclear
    i.e. - FV in curvilinear coordinates requires lots of averaging depending on definition of variables such as staggered vs cell vertex arrangement
    - Both FD and FV implementations are very similar near grid singularities
  - Major differences come from time integration scheme which also affects the computational efficiencies, especially, for unsteady flow computations

Discussion on Applications

- Rapid turn around can be accomplished through the use of
  - Algorithm : convergence acceleration such as multi-grid, and GMRES
  - Parallel implementation
- Total process time can be reduced by
  - Automatic solution process including CAD to grid procedure
- Need further development of methodology as well as physics modeling for
  - Deep understanding of flow physics such as unsteady flow characterization for better aeroacoustics modeling,and flow induced vibration
    ⇒ Is LES method mature enough for this?
  - Need to matrix IT tools to flow simulation for smart flow control and optimization
- Efficient extraction of information is still a challenge

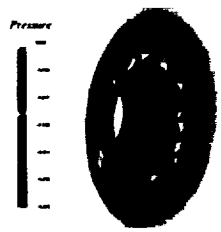
On top of all these we still need trained CFDers to solve many unsolved real world problems, for development of flow devices and for better understanding of flow physics

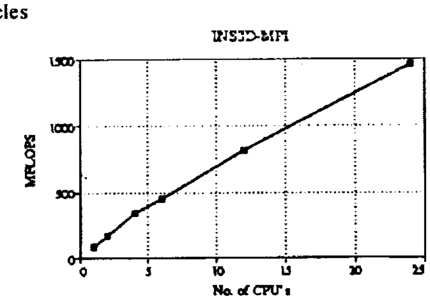
Back up charts

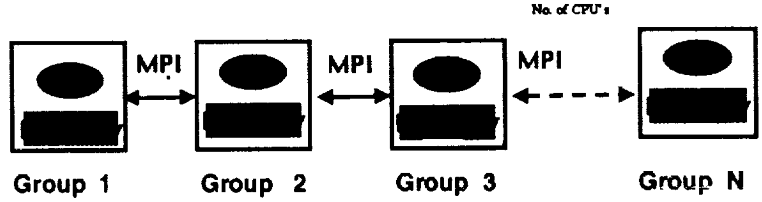
Parallel Implementation of INS3D

INS3D-MPI - coarse grain

Implemented by T. Faulkner & J. Dacles





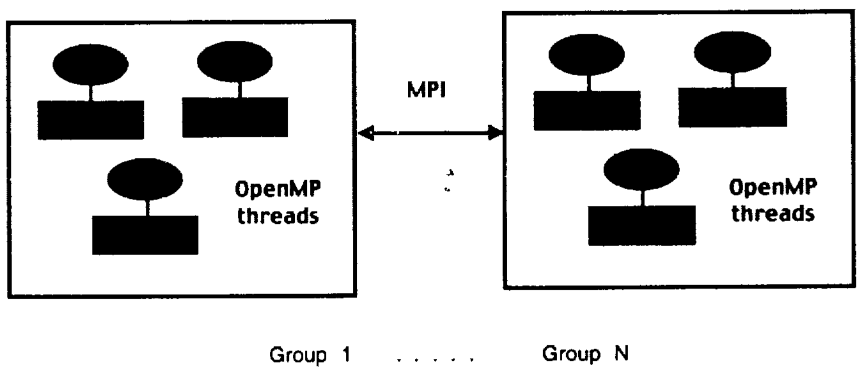


Parallel Implementation of INS3D

INS3D-MPI /OpenMP

MPI (coarse grain) + OpenMP (fine grain)

Implemented by using CAPO/CAPT tools

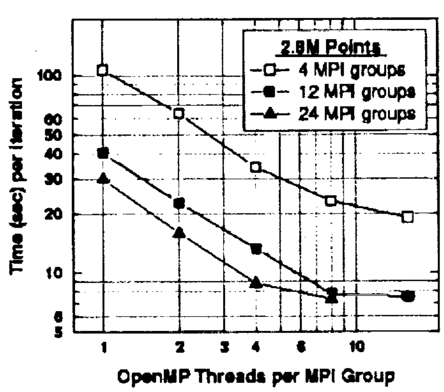


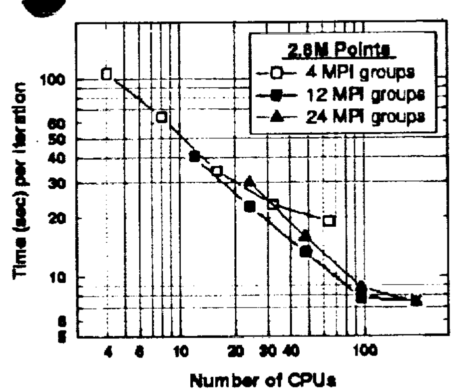
Parallel Implementation of INS3D

MPI coarse grain + OpenMP fine grain

TEST CASE : SSME Impeller

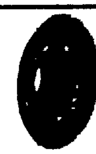
24 zones / 2.8 Million points



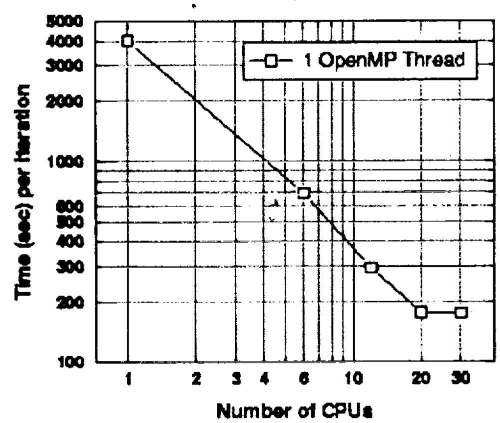


Parallel Implementation of INS3D

MPI coarse grain + OpenMP fine grain



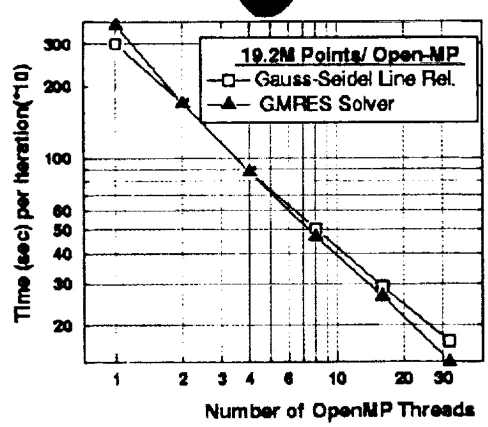
TEST CASE : SSME Impeller
60 zones / 19.2 Million points



Parallel Implementation of INS3D

OpenMP with two different solver

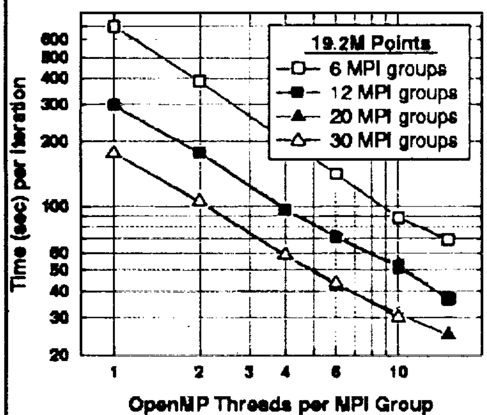
TEST CASE : SSME Impeller
60 zones / 19.2 Million points

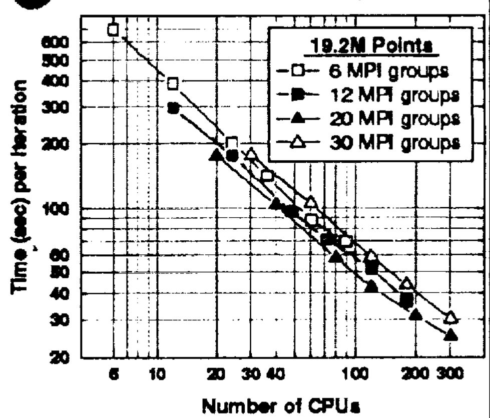


Parallel Implementation of INS3D

MPI coarse grain + OpenMP fine grain

TEST CASE : SSME Impeller
60 zones / 19.2 Million points

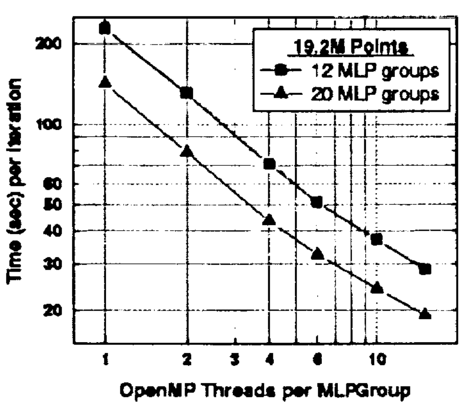


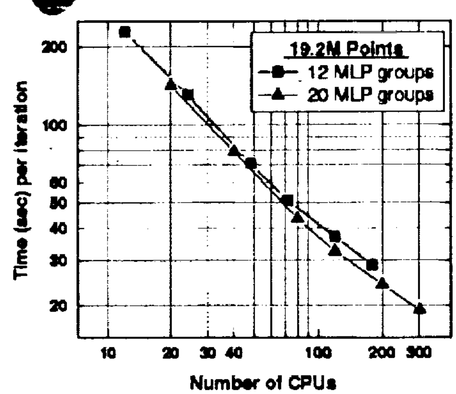


Parallel Implementation of INS3D

INS3D-MLP (NAS MLP no pin-to-node) / OpenMP

TEST CASE : SSME Impeller
60 zones / 19.2 Million points

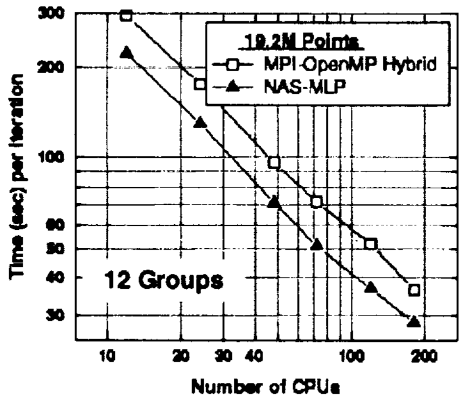


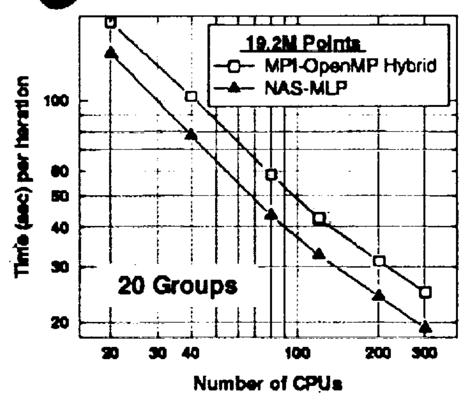


Parallel Implementation of INS3D

INS3D-MLP/OpenMP vs. -MPI/OpenMP

TEST CASE : SSME Impeller
60 zones / 19.2 Million points





Parallel Implementation of INS3D

INS3D-MLP
GMRES-ILU(0) solver – 20 Groups

TEST CASE : SSME Impeller
60 zones / 19.2 Million points