A high-order accurate, high-efficiency incompressible Navier-Stokes solver on overlapping grids

Kyle K. Chand
special guest star: Bill Henshaw

Lawrence Livermore National Laboratory
Livermore, CA, USA

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

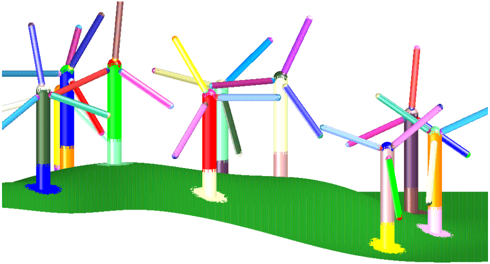
11th Symposium on Overset Composite Grids and Solution Technology
October 15-18, 2012
Dayton, Ohio USA

LLNL-PRES-590252

CgIns: a high-fidelity modeling tool for computational wind engineering

This project expands and enhances our prior work

- Overture framework: high-order accurate discretization and grid generation technology
- CgIns: a high-order accurate, high-efficiency Boussinesq flow solver
- New: efficient approximate factorization schemes
- New: higher-order compact discretizations
- New: parallel high-order accurate multigrid



Our Goal:

To provide a publicly available, high-order accurate, flexible and efficient incompressible LES tool

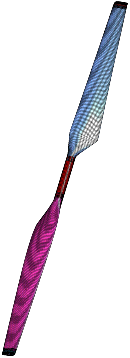
CgIns solves the incompressible Navier-Stokes equations with a split-step method on composite grids

$$
\begin{aligned}
\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \nu\Delta\mathbf{u} - \mathbf{f} &= 0, \\
\Delta p + \nabla\mathbf{u} : \nabla\mathbf{u} - \alpha\nabla\cdot\mathbf{u} - \nabla\cdot\mathbf{f} &= 0.
\end{aligned}
$$

$$
\begin{aligned}
(\mathbf{u}(\mathbf{x},0), T(\mathbf{x},0)) = (\mathbf{u}_I(\mathbf{x}), T_I(\mathbf{x})), &\quad t = 0, \quad \mathbf{x}\in\Omega, \\
B(\mathbf{u}, T) = 0, &\quad t > 0, \quad \mathbf{x}\in\partial\Omega.
\end{aligned}
$$

Naturally, we use structured overlapping grids:

- High efficiency due to regular data structure
  → Cartesian grids dominate the domain (optimized data & algorithms)
- Overture provides extensive grid generation and management tools
  → Automatic, high-order, parallel, composite grid generator
  → The framework supports high-order accurate, composite grid solvers
- Smooth grids are essential to high-order accurate algorithms



Efficiency and accuracy are achieved by combining approximate factorization methods with compact discretizations

- Approximate factorization (AF) schemes offer larger timesteps with second order accuracy in time:
  AF schemes discretizes
  $$\frac{\partial U}{\partial t} + AU + BU = 0$$
  by starting with Crank-Nicolson:
  $$(I + \frac{\Delta t}{2}(A+B))U^{n+1} = (I - \frac{\Delta t}{2}(A+B))U^n$$
  which is approximately factored to become
  $$(I + \frac{\Delta t}{2}A)(I + \frac{\Delta t}{2}B)U^{n+1} = (I - \frac{\Delta t}{2}A)(I - \frac{\Delta t}{2}B)U^n$$
- Deferred corrections may be incorporated to increase time accuracy

Efficiency and accuracy are achieved by combining approximate factorization methods with compact discretizations

- Approximate factorization (AF) schemes offer larger timesteps with second order accuracy in time:
  AF schemes discretizes

$$\frac{\partial U}{\partial t} + AU + BU = 0$$

by starting with Crank-Nicolson:

$$(I + \frac{\Delta t}{2}(A+B))U^{n+1} = (I - \frac{\Delta t}{2}(A+B))U^{n}$$

which is approximately factored to become

$$(I + \frac{\Delta t}{2}A)(I + \frac{\Delta t}{2}B)U^{n+1} = (I - \frac{\Delta t}{2}A)(I - \frac{\Delta t}{2}B)U^{n}$$

- Deferred corrections may be incorporated to increase time accuracy
- Compact spatial schemes can be integrated into the AF solves
- Special "combined" compact schemes have been developed:
  $\rightarrow$ reduce the number of factors
  $\rightarrow$ preserve accuracy at boundaries
  $\rightarrow$ 4^{th} and 6^{th} order accuracy

All this must work on composite, dynamic grids on a range of HPC systems while preserving stability and accuracy

Factored schemes on curvilinear overlapping grids

Begin with the conservative form of the momentum equation:

$$\frac{\partial u_i}{\partial t} + \sum_{j=1}^{N_d}\left[\frac{\partial (u_j u_i)}{\partial x_j} - \nu_i \frac{\partial^2 u_i}{\partial x_j^2}\right] = f_i$$

where f_i contains the pressure gradient, buoyancy terms and any forcing.

One way to write this equation on a curvilinear grid is:

$$\begin{aligned}\frac{\partial u_i}{\partial t} \quad &+ \quad \sum_{k=1}^{N_d}\sum_{j=1}^{N_d}\left\{\frac{\partial}{\partial r_k}\left[\left(u_j\frac{\partial r_k}{\partial x_j} - \nu_i\frac{\partial^2 r_k}{\partial x_j^2} + 4\nu_i\frac{\partial r_k}{\partial x_j}\frac{\partial^2 r_k}{\partial x_j r_k}\right)u_i\right]\right. \\ &- \quad \left.\frac{\partial^2}{\partial r_k^2}\left[\nu_i\left(\frac{\partial r_k}{\partial x_j}\right)^2 u_i\right]\right\} = \\ &\quad u_i\sum_{k=1}^{N_d}\sum_{j=1}^{N_d}\left[\frac{\partial}{\partial r_k}\left(\frac{\partial r_k}{\partial x_j}u_j + \nu_i\frac{\partial^2 r_k}{\partial x_j^2}\right) - \frac{\partial^2}{\partial r_k^2}\left(\frac{\partial^2 r_k}{\partial x_j}\right)^2\nu_i\right] \\ &+ \quad \nu_i\sum_{k=1}^{N_d}\sum_{j=1}^{N_d}\frac{\partial r_k}{\partial x_j}\sum_{l=1,l\neq k}^{N_d}\frac{\partial r_l}{\partial x_j}\frac{\partial^2 u_i}{\partial r_k r_l} + f_i\end{aligned}$$

Factored schemes on curvilinear overlapping grids

If the INS equations are rewritten in a curvilinear coordinate system the factors become apparent:

$$
\begin{aligned}
\frac{\partial u_i}{\partial t} &+ \sum_{k=1}^{N_d}\left[\frac{\partial}{\partial r_k}\left(A_{ik}u_i\right)+\frac{\partial^2}{\partial r_k^2}\left(B_{ik}u_i\right)\right] = f_i^c + f_i \\
A_{ik} &= \sum_{j=1}^{N_d}\left(u_j\frac{\partial r_k}{\partial x_j} - \nu_i\frac{\partial^2 r_k}{\partial x_j^2} + 4\nu_i\frac{\partial r_k}{\partial x_j}\frac{\partial^2 r_k}{\partial x_j r_k}\right) \\
B_{ik} &= -\sum_{j=1}^{N_d}\nu_i\left(\frac{\partial r_k}{\partial x_j}\right)^2
\end{aligned}
$$

- The LHS is approximated with a factored Crank-Nicolson (CN) discretization
  $\rightarrow$ Still have to deal with the nonlinearity...
- The RHS is integrated explicitly using Adams-Bashforth (AB).
  $\rightarrow$ AB integration of $f_i^c + f_i$ does not seem to cause a severe Δt limit.
- This combination maintains CN stability for parabolic equations (Beam-Warming, 1979)

Integrating part of the problem explicitly can still yield an (almost) unconditionally stable method

Beam and Warming showed how to integrate mixed-derivative parabolic terms explicitly and still maintain CN's A-Stability

Linear stability analysis on the semi- or fully discrete system demonstrates that integrating the pressure explicitly is ok

- using approximately factored CN with Adams-Bashforth for the pressure gradient results in a stability restriction:

  $$\alpha \Delta t \leq 1$$

  ...but we are free to choose α (divergence damping)
- using Crank-Nicolson for the pressure gradient results in unconditional stability

Incorporating compact schemes requires adding line solves

Compact schemes approximate derivatives implicitly

$$P_r \frac{\partial U}{\partial r} = D_r U + \mathcal{O}\left(h^p\right)$$
$$P_{rr} \frac{\partial^2 U}{\partial r^2} = D_{rr} U + \mathcal{O}\left(h^p\right)$$

where the P and D are matrices and U is the discrete solution

Incorporation into an approximate factorization, for example:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} - b\frac{\partial^2 u}{\partial x^2} = 0.$$

An approximately factored CN discretization, with compact approximations, would need four banded solves:

$$(I + P_r^{-1} D_r(\frac{\Delta t}{2} a))(I - P_{rr}^{-1} D_{rr}(\frac{\Delta t}{2} b))U^{n+1} =$$
$$(I - P_r^{-1} D_r(\frac{\Delta t}{2} a))(I + P_{rr}^{-1} D_{rr}(\frac{\Delta t}{2} b))U^{n}.$$

INS would need $2N_d N_{ops} N_{eqs}$ solves

Combined schemes reduce the number of line solves

Generally, $P_r \neq P_{rr}$ which leads to many factors

By adding stencil width, we can set $P_r = P_{rr} = P$ and compute the corresponding P, D_r, and D_{rr} operators.

The combined operator for advection-diffusion yields 1 factor

$$\left[P + \frac{\Delta t}{2}(aD_r - bD_{rr})\right] U^{n+1} = \left[P - \frac{\Delta t}{2}(aD_r - bD_{rr})\right] U^n$$

For INS, we have $2N_d N_{eqs}$ line solves.

The leading-order T.E. is lower but the stencil is wider

	FD4	CC4	OC4	CC6
$\frac{\partial u}{\partial x}$	$\frac{1}{30}u^{5\prime}$	$\frac{1}{180}u^{5\prime}$	$\frac{1}{180}u^{5\prime}$	$-\frac{1}{9450}u^{7\prime}$
$\frac{\partial^2 u}{\partial x^2}$	$\frac{1}{90}u^{6\prime}$	$\frac{1}{360}u^{6\prime}$	$\frac{1}{240}u^{6\prime}$	$\frac{19}{75600}u^{8\prime}$

Figure: Leading truncation error constants for stencil width-5 approximations. FD4 - 4^{th} order finite difference; CC4 - combined $4th$ order compact; OC4 - “optimal” 4^{th} order compact; CC6 - combined 6^{th} order compact.

Details: interpolation points, linearization and iteration

Extrapolation in time is used to estimate $\mathbf{u}^{n+1}$ for the interpolation and parallel ghost point equations as well as the initial linearization state (i.e. A_{ik}^{n+1}).

At interpolation and parallel ghost points:

$$
\begin{aligned}
\mathbf{u}_I^{n+1,0} &= \mathbf{u}_I^n + (\mathbf{u}_I^n - \mathbf{u}_I^{n-1}) \\
\mathbf{u}_I^{n+1,m*} &= \mathbf{u}_I^n + (\mathbf{u}_I^{n+1,m-1} - \mathbf{u}_I^n), m > 0 \\
\mathbf{u}_I^{n+1,m} &= \mathbf{g}(\mathbf{x}), \text{where } \mathbf{g} \text{ is either a b.c. or interpolation}
\end{aligned}
$$

At interior points, the LHS linearization state is:

$$
\begin{aligned}
\mathbf{u}^{n+1,0} &= \mathbf{u}^n + (\mathbf{u}^n - \mathbf{u}^{n-1}) \\
\mathbf{u}^{n+1,m} &= \mathbf{u}^{n+1,m-1}, m > 0
\end{aligned}
$$

where the LHS equations are solved, updating $\mathbf{u}$, in each iteration. This approach maintains 2^{nd} order accuracy without requiring block-banded solvers.

Even more details

Yes, there is an artificial viscosity...

Each directional factor gets a dissipative term:

$$\left\{-(a_{21}+a_{22}|\nabla\mathbf{u}|)\Delta_+\Delta_- + (a_{41}+a_{42}|\nabla\mathbf{u}|)(\Delta_+\Delta_-)^2\right\}\mathbf{u}_i^{n+1}$$

where

$$\Delta_+\Delta_- u_i = u_{i+1} - 2u_i + u_{i-1}$$

- the nonlinear a_{22} term is similar to a Smagorinsky Large Eddy Simulation viscosity
- the a_{42} term performs the same function but preserves more high frequency content than the lower order term
- typically $a_{21} = a_{22} = 0$ except near boundaries with insufficient resolution to capture the boundary layers

Timestep algorithm

Variables

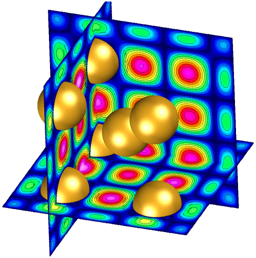
Δt : `timestep`
U^n : `solution at step` n
N_f : `number of factors`
A_f^n : `operator for factor`
f `computed using` U^n
N_I : `number of` U^{n+1}
`update iterations`
forcing : `explicitly integrated forcing`

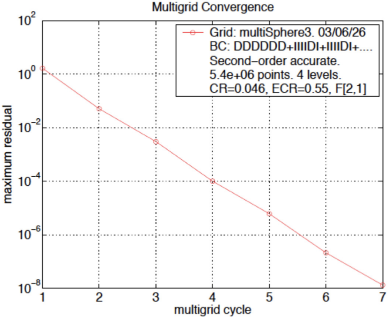
Notes:

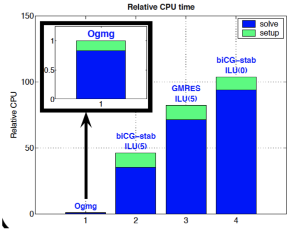
- N_I can also be set via a tolerance
- The solves are performed independently on each grid
- The code is implemented to minimize the number of temporary arrays

BEGIN:
forcing $= 0$
$U^{n+1} = 0$
$U^* = U^n$
for ($f = N_f - 1;\ f \geq 0;\ f++$)
  solve $P_f U^{**} = (P_f - \frac{\Delta t}{2} A_f^n) U^*$
  $U^* \leftarrow U^{**}$
  *addForcingForFactor(**forcing**,f)*
endfor
$R \leftarrow U^* +$ **forcing**
$U^{n+1,0} \leftarrow 2U^n - U^{n-1}$
for ($m = 1;\ m \leq N_I;\ m++$)
  for ($f = 0;\ f < N_f;\ f++$)
    solve$(P_f + \frac{\Delta t}{2} A_f^{n+1,m-1}) U^{**} = U^*$
    $U^* \leftarrow U^{**}$
  endfor
  $U^{n+1,m} \leftarrow U^*$
  $U^* \leftarrow R$
  interpolateAndApplyBC($U^{n+1,m}$)
  updateInterpolationPointForcing($U^{n+1,m}$,U^*)
endfor
$U^{n+1} \leftarrow U^{n+1,N_I}$
solvePressureEquation(U^{n+1})
END

Multigrid provides fast pressure solves on dynamic overlapping grids





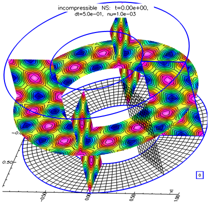


Matrix-free multigrid exploits the grid & solver

- relatively inexpensive setup and memory efficient
- efficient for high-order accurate methods
- mesh-independent convergence rates

Code Verification is an integral part of our development

Verification demonstrates correct implementation of numerical approximations and that the method possesses the required accuracy and stability



Verification is difficult for complex algorithms

- High-order accurate algorithms are sensitive to small errors and grid irregularities
- Weak instabilities may only be apparent after long computations and are hard and expensive to diagnose

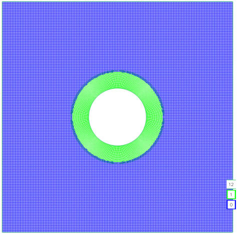
Twilight-zone (manufactured) solutions provide rigorous verification

- Exact multidimensional solutions are posed and used to force the PDE
- Errors are measured and convergence rates are estimated
- Can catch "low order" errors that are consistent but reduce accuracy

h_{max}	$\lvert e_p \rvert_\infty$	$\lvert e_u \rvert_\infty$	$\lvert e_v \rvert_\infty$	$\lvert e_w \rvert_\infty$	$\lvert \nabla \cdot \mathbf{u} \rvert_\infty$
1.34e-01	2.74e-02	1.13e-01	8.63e-02	7.96e-02	1.63e+00
6.68e-02	5.74e-03 (4.8)	5.81e-03 (19.4)	4.25e-03 (20.3)	4.06e-03 (19.6)	9.86e-02 (16.5)
3.34e-02	5.19e-04 (11.1)	3.16e-04 (18.4)	2.25e-04 (18.9)	2.10e-04 (19.3)	8.69e-03 (11.3)
1.67e-02	3.40e-05 (15.3)	1.87e-05 (16.9)	1.38e-05 (16.3)	1.28e-05 (16.4)	9.03e-04 (9.6)
rate	3.2	4.2	4.2	4.2	3.6

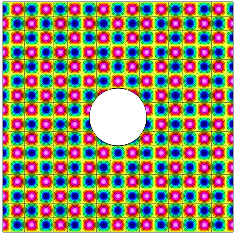
Both steady state and time dependent manufactured solutions are used to test spatial and temporal approximations

Verification of fourth-order spatial and second-order temporal accuracy



h_{max}	$\lvert e_p \rvert_\infty$	$\lvert e_u \rvert_\infty$	$\lvert e_v \rvert_\infty$	$\lvert \nabla \cdot \mathbf{u} \rvert_\infty$
6.13e-02	1.17e-02	4.35e-03	4.93e-03	9.36e-02
3.08e-02	7.16e-04 (16.3)	1.68e-04 (25.9)	1.72e-04 (28.7)	5.99e-03 (15.6)
1.54e-02	4.31e-05 (16.6)	1.14e-05 (14.7)	9.52e-06 (18.1)	4.45e-04 (13.5)
7.70e-03	2.91e-06 (14.8)	7.62e-07 (15.0)	6.67e-07 (14.3)	3.43e-05 (13.0)
rate	4.0	4.1	4.3	3.8

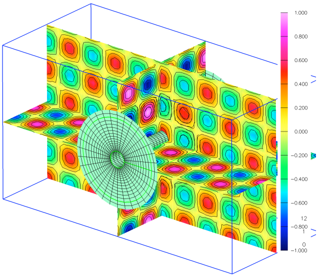
Table: Time dependent exact solution



h_{max}	$\lvert e_p \rvert_\infty$	$\lvert e_u \rvert_\infty$	$\lvert e_v \rvert_\infty$	$\lvert \nabla \cdot \mathbf{u} \rvert_\infty$
6.13e-02	1.97e-05	1.20e-05	1.04e-05	3.27e-04
3.08e-02	1.32e-06 (14.9)	8.97e-07 (13.4)	8.11e-07 (12.8)	2.06e-05 (15.9)
1.54e-02	7.58e-08 (17.4)	5.66e-08 (15.8)	4.85e-08 (16.7)	1.28e-06 (16.1)
7.70e-03	4.75e-09 (16.0)	3.09e-09 (18.3)	2.90e-09 (16.7)	8.23e-08 (15.6)
rate	4.0	4.0	4.0	4.0

Table: Steady state exact solution

Manufactured solutions test the algorithm with full dimensionality, nonlinearity and grid complexities



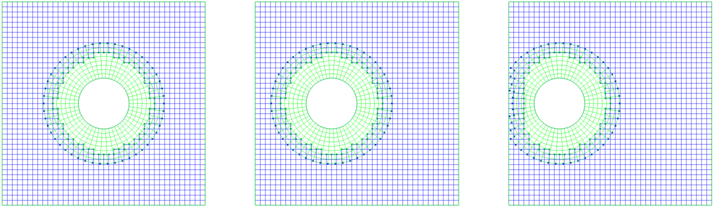
Full 3D test of the INS algorithm on an overlapping grid

- 4th order spatial, 2nd order temporal
- Tests overlapping grid algorithm and long time integration
- Demonstrates optimized Cartesian and curvilinear grid code

h_{max}	$\lvert e_p \rvert_\infty$	$\lvert e_u \rvert_\infty$	$\lvert e_v \rvert_\infty$	$\lvert e_w \rvert_\infty$	$\lvert \nabla \cdot \mathbf{u} \rvert_\infty$
2.94e+00	5.10e-02	6.71e-02	3.56e-02	3.44e-02	1.52e-01
1.47e+00	3.62e-03	4.40e-03	2.19e-03	2.21e-03	2.28e-02
7.36e-01	2.62e-04	3.16e-04	1.44e-04	2.34e-04	1.34e-03
rate	3.8	3.9	4.0	3.6	3.4

Moving grids are particularly challenging to develop and verify

Subtle bugs are often buried in complex code



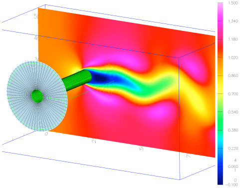
h_{max}	$\|e_p\|_\infty$	$\|e_u\|_\infty$	$\|e_v\|_\infty$	$\|\nabla \cdot \mathbf{u}\|_\infty$
6.13e-02	1.56e-02	2.84e-02	2.17e-02	2.03e-01
3.08e-02	2.56e-03 (6.1)	1.53e-03 (18.6)	1.25e-03 (17.4)	1.52e-02 (13.4)
1.54e-02	2.08e-04 (12.3)	7.63e-05 (20.1)	6.21e-05 (20.1)	1.38e-03 (11.0)
7.70e-03	1.37e-05 (15.2)	3.91e-06 (19.5)	2.85e-06 (21.8)	1.18e-04 (11.7)
rate	3.4	4.3	4.3	3.6

A verified code can be validated

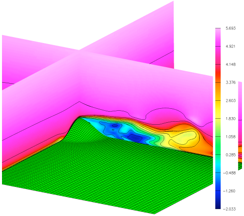
Validation tests the mathematical model's ability to represent the physical problems of interest

Validation is accomplished via comparison to experimental data

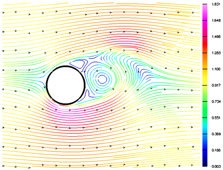
- Good experimental data are necessary
- Errors in the approximation of the mathematical model must be understood (e.g. verification)
- Like verification, validation is a continuous process



(a) flow past a cylinder



(b) flow past a steep hill



(c) moving cylinder

Computing the Strouhal number for 3D flow past a right circular cylinder

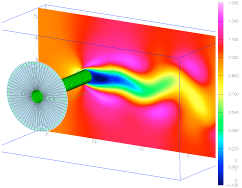
Schlichting, 1960

This case uses many parts of the model

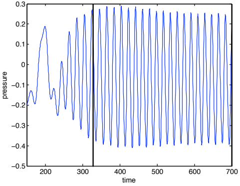
- the 3D compact AFS scheme
- parallel algorithms
- fourth-order spatial accuracy
- more code optimization still required

$Re = 1000$ Performance and Results

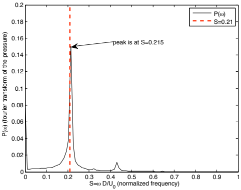
- 400k grid points, 6 2.2Gz Xeon cores
- under 3hrs of wallclock time
- 1Gb of RAM (600Mb for 1 cpu)
- $St = .215$ matches Schlichting



(a) velocity contours



(b) pressure history



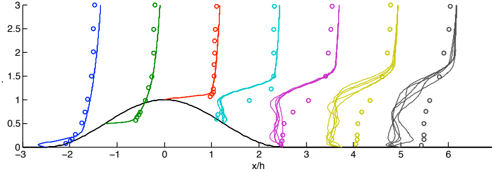
(c) normalized frequency

Flow over Ishihara, Hibi and Oikawa's wind tunnel hill demonstrates the need for code optimization and more modeling

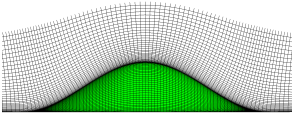
Ishihara, Hibi, Oikawa, 1999

This case is challenging

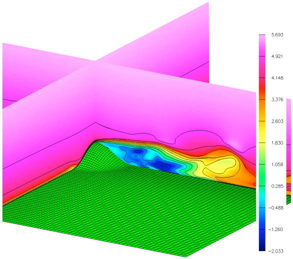
- $Re = 12000$, inflow given by log-law (coefficients from experiment)
- 4 million grid points
- 128 cpus, 56hrs of wallclock time
- Memory and cpu performance are worse than expected
- *The poor agreement with data is probably due to the lack of a wall model*



(a) instantaneous u/U_0 (curves) and experimental values



(b) grid near the hill



(c) x-velocity contours

Impulsively started rotating and translating cylinder

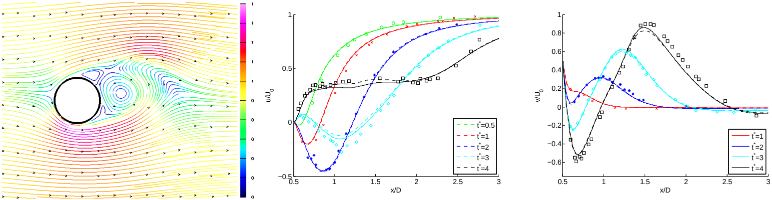
Coutanceau and Menard, 1985

This problem is easy to resolve

- $Re = 200$, cylinder impulsively started into translation and rotation
- 368k grid points, 177Mb RAM, less than one hour on 2.2Gz cpu

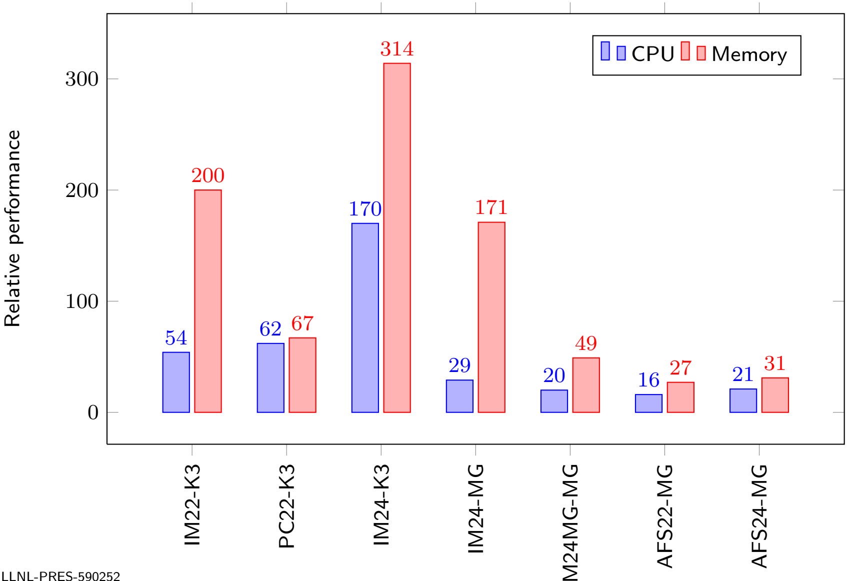
This problem is tested in two ways

- rotate the cylinder and impose translation via boundary and inflow conditions (dashed lines)
- rotate and translate the cylinder (solid lines)
- *the results are sensitive to how "impulsively" the cylinder is started!*



(a) streamlines at t^*=4 (b) x-velocity comparison (c) y-velocity comparison

The factored scheme performs well compared to the original predictor-corrector and implicit algorithms



Flow past a rotating ring

movie by Bill Henshaw

- 14×10^6 grid points
- $Re \approx 2000$
- about 36hrs on 64 processors

loading movie...

Turbine mock-up

movie by Bill Henshaw

loading movie...

Our new method has been verified and (partially) validated, but more modeling work is required

Much of the core functionality is implemented and tested

- High-order compact/AF scheme implemented and verified
- Parallel moving grid generation and 4^{th} order accurate multigrid implemented
- Parallel and moving grid AF scheme is verified

But there is still more work to do...

- Memory and cpu performance could be futher optimized
- Wall models have been implemented and are currently being tested

Future applications

- Wind park modeling with terrain
- Building flow modeling and control system design
- Urban scale flows

CgIns is available at: http://www.llnl.gov/casc/Overture