CgWind: A Composite Grid Simulation Tool for Wind Energy Applications

Kyle K. Chand William D. Henshaw

Lawrence Livermore National Laboratory
Livermore, CA, USA

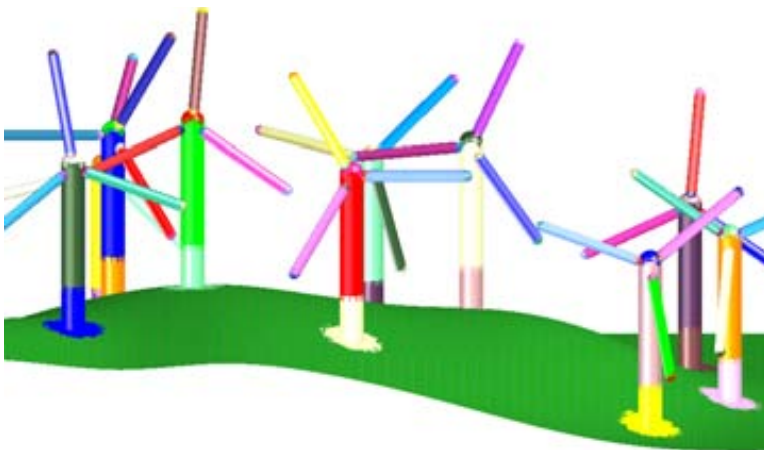
The 10th Symposium on Overset Composite Grids and Solution Technology
NASA Ames, Moffet Field, California, USA
September 20-23, 2010

LLNL-PRES-455051

Open source, high fidelity modeling for wind energy

Wind energy needs high-fidelity simulations

- design of larger, lighter, higher performance turbines
- accurate estimates of energy production with realistic geometry and terrain
- large scale wind park performance prediction coupled to meso-scale models



CgWind exploits DOE investments in high-fidelity modeling

- Overture framework: high-order accurate discretization and grid generation technology for complex geometries
- CgIns: a high-order accurate, high-efficiency incompressible (Boussinesq) flow solver
- WRF: meso-scale LES modeling

Our Goal:

To provide the wind energy community with a publicly available, accurate, flexible and efficient modeling tool for turbines and parks

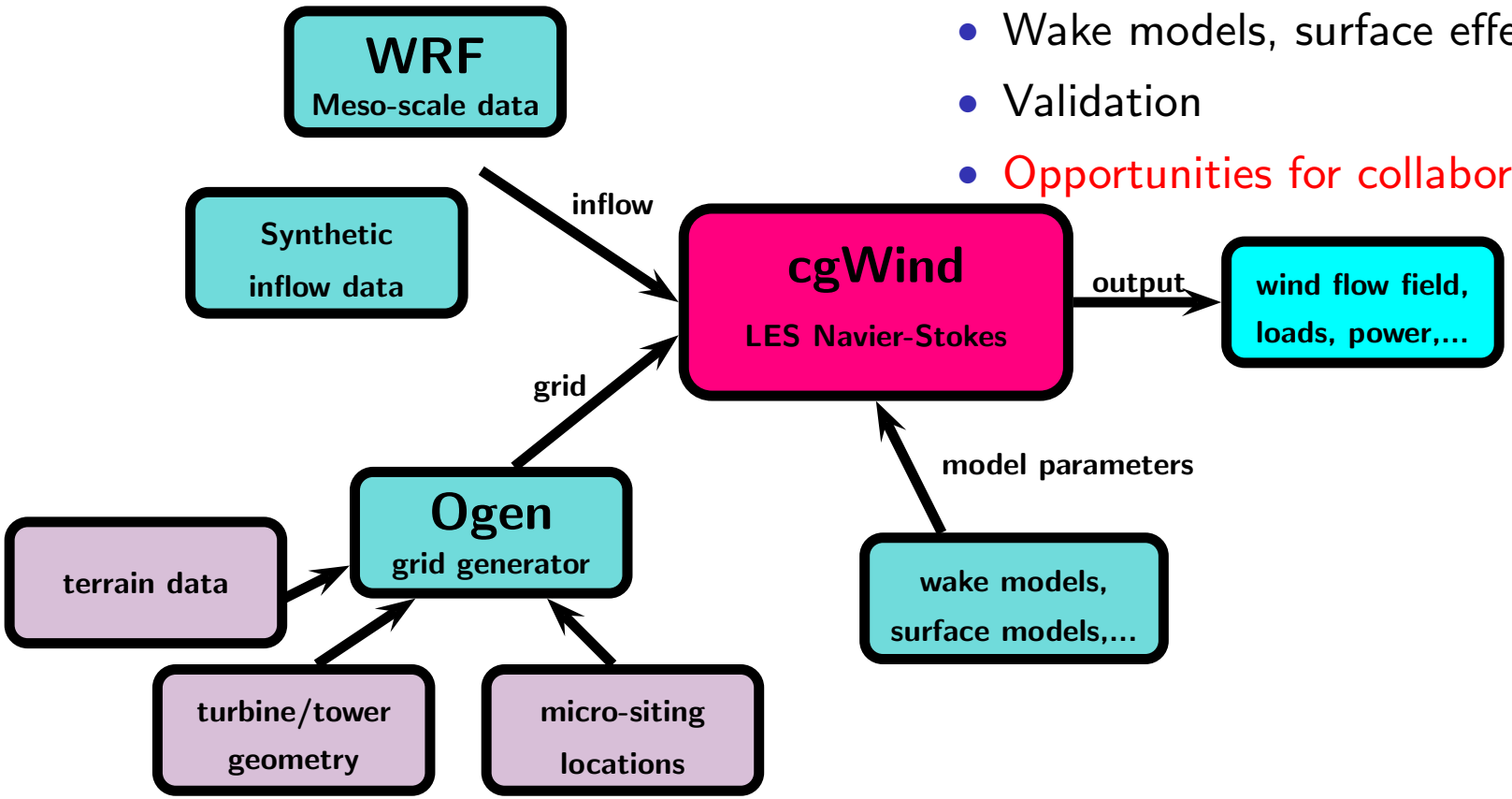
CgWind will integrate many sources of information

Initial development:

- Enhanced INS solver and LES
- Grid generation & multigrid
- Software framework, verification

Integration of new features:

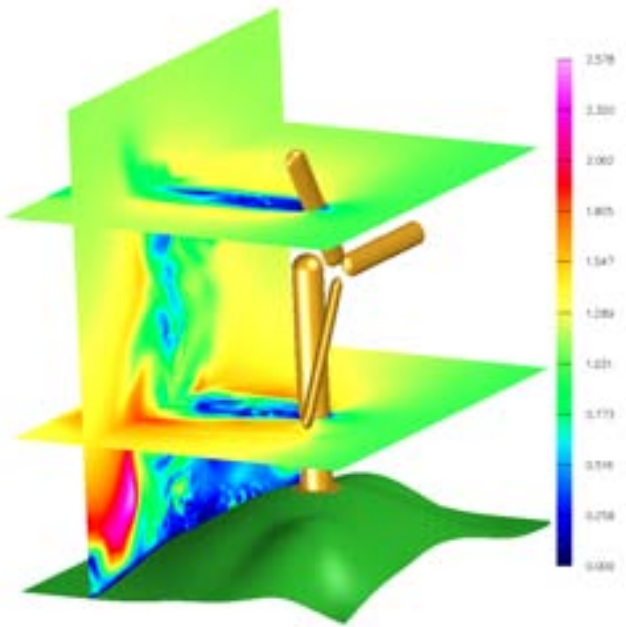
- Coupling to meso-scale models
- Terrain and siting data integration and grid generation
- Wake models, surface effects,...
- Validation
- Opportunities for collaboration



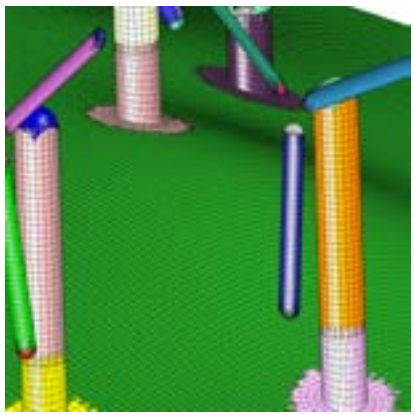
CgWind is built upon Overture solvers and grid generation tools

Demo simulation with existing solver

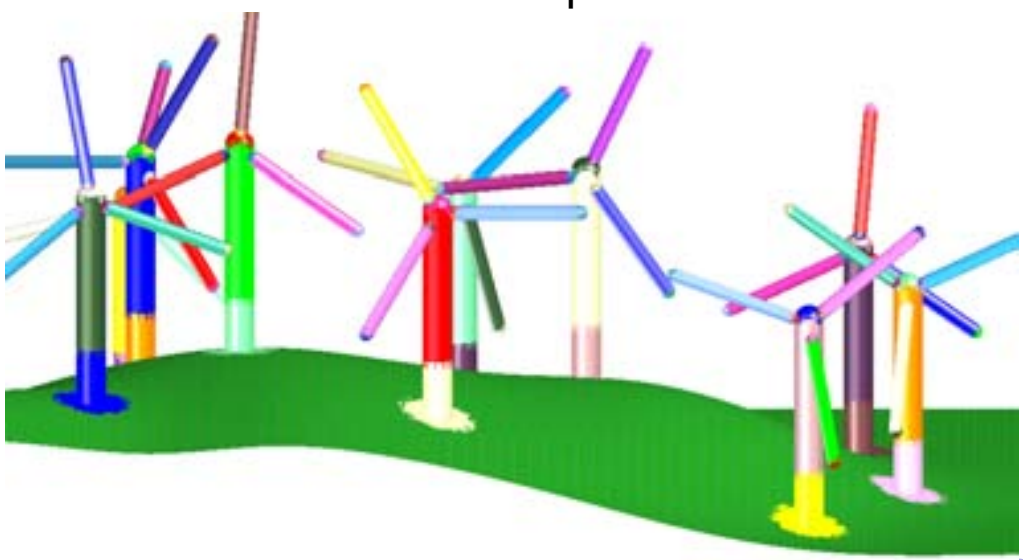
- 35M grid points, static grid
- Learned that more efficient high-order solvers are needed



Terrain & turbine mock-up



- 5.6M grid points, 121 grids
- 455s serial grid generation
- synthetic terrain
- parametrized turbine sites for automated placement



CgIns forms the core of CgWind

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

$$
\begin{aligned}
\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \nu\Delta\mathbf{u} + \alpha\mathbf{g}(T - T_{ref}) - \mathbf{f} &= 0, \\
\Delta p + \nabla\mathbf{u} : \nabla\mathbf{u} + \alpha(\mathbf{g}\cdot\nabla)T - \nabla\cdot\mathbf{f} &= 0, \\
T_t + (\mathbf{u}\cdot\nabla)T - \frac{1}{\rho C}\nabla\cdot(k\nabla T) - f_T &= 0, \\
(\mathbf{u}(\mathbf{x},0), T(\mathbf{x},0)) = (\mathbf{u}_I(\mathbf{x}), T_I(\mathbf{x})), \quad t = 0, \quad \mathbf{x}\in\Omega, \\
B(\mathbf{u},T) = 0, \quad t > 0, \quad \mathbf{x}\in\partial\Omega.
\end{aligned}
$$

- Explicit and implicit time stepping
  → CgWind: efficient approximate factorization schemes
- Second and fourth order central finite differences
  → CgWind : higher-order compact discretizations
- 2^{nd} order accurate multigrid or Krylov methods for the pressure
  → CgWind: parallel high-order accurate multigrid
- Structured, dynamic, overlapping grids provide accuracy
  → CgWind: parallel adaptive mesh refinement
- Smagorinsky-type turbulence models
  → CgWind: advanced nonlinear LES models

Approximate factorization & Compact discretization

- 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

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

- Compact schemes can be integrated into the AF solves
- Special "combined" compact schemes have been developed:
  $\rightarrow$ reduce the number of factors

$$(a\partial_r + b\partial^2_{rr})u \rightarrow P^{-1}(D_r a + D_{rr} b)u + \text{corrections}$$

  $\rightarrow$ preserve accuracy at boundaries
  $\rightarrow$ 4^{th} and 6^{th} order accuracy with a 5 point stencil
  $\rightarrow$ special penta-diagonal solvers that handle wider boundary stencils

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:

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

Factored schemes on curvilinear overlapping grids

If this mess is rewritten the factors become apparent:

$$\frac{\partial u_i}{\partial t} + \sum_{k=1}^{N_d} \left[\frac{\partial}{\partial r_k} \left(A_{ik} u_i \right) + \frac{\partial^2}{\partial r_k^2} \left(B_{ik} u_i \right) \right] = f_i^c + f_i$$

$$A_{ik} = \sum_{j=1}^{N_d} \left(u_j \frac{\partial r_k}{\partial x_j} - \nu_i \frac{\partial^2 r_k}{\partial x_j^2} + 4\nu_i \frac{\partial r_k}{\partial x_j} \frac{\partial^2 r_k}{\partial x_j r_k} \right)$$

$$B_{ik} = -\sum_{j=1}^{N_d} \nu_i \left(\frac{\partial r_k}{\partial x_j} \right)^2$$

- 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)

Incorporating compact schemes requires adding line solves

Compact schemes approximate derivatives implicitly

$$P_r U_r = D_r U + \mathcal{O}\left(h^p\right)$$
$$P_{rr} U_{rr} = D_{rr} U + \mathcal{O}\left(h^p\right)$$

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 bandwidth, 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 $2N_dN_{eqs}$ 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.

Boundary conditions and stencils for near-wall accuracy

No-slip walls are treated as a Dirichlet condition

We discretize the time derivative of the boundary condition:

$$\frac{\partial \mathbf{u}_{NS}}{\partial t} = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial t}, \mathbf{x} \in \partial\Omega_{NS}$$

which can be "factored" into a sequence of identity operators. P is decoupled from the interior by using a one-sided P for the interior points next to the boundary.

Other boundary conditions

- periodic boundaries are easily incorporated into a modified banded solver
- outflow conditions currently use first order extrapolation
- interpolation point P coefficients are decoupled by explicit finite-differences near the fringe

Augmented banded solvers allow wider stencils at boundaries

Details: interpolation points, linearization and iteration

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

At interpolation points:

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

where interpolation occurs in each iteration.

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.

Timestep algorithm

Variables

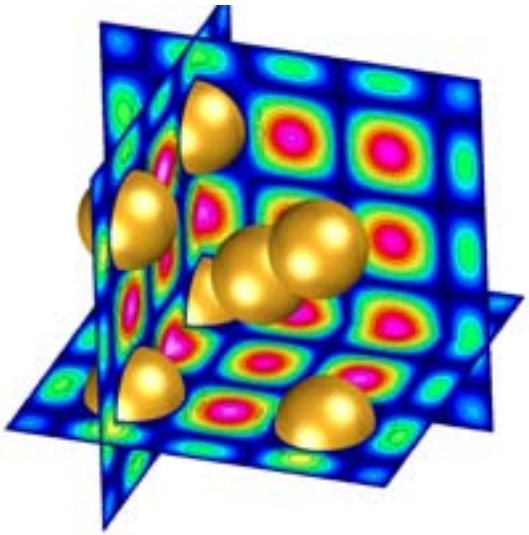
Δt : timestep
U^n : solution at step n
N_f : number of factors
A_f^n : operator for factor f computed using U^n
N_I : number of U^{n+1} update iterations
forcing : explicitly integrated forcing

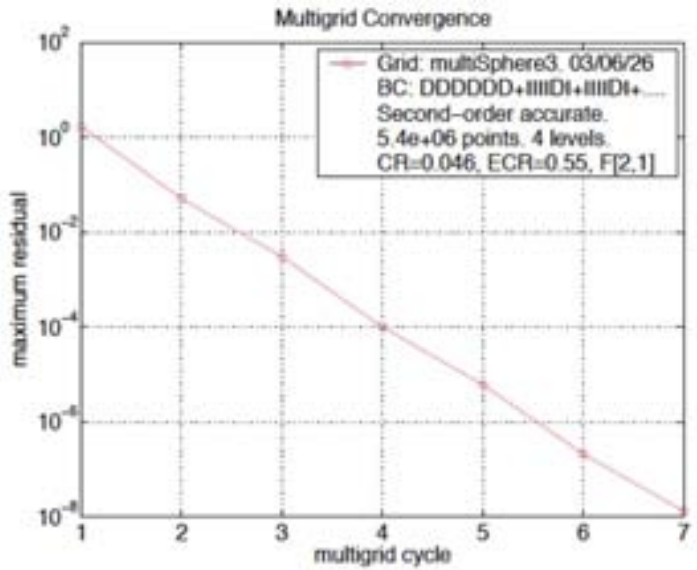
Notes:

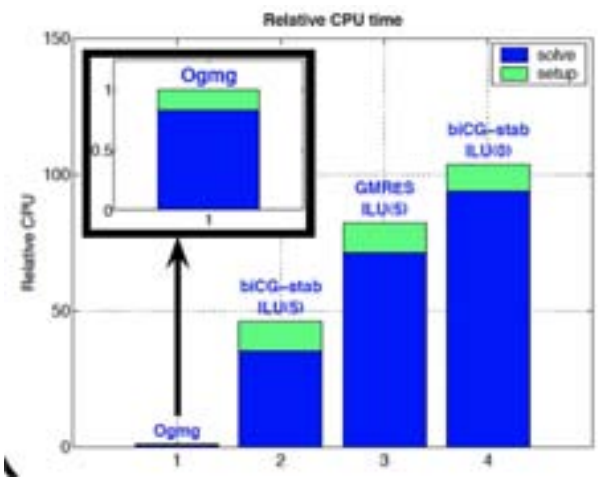
- N_I can also be set via a tolerance
- The solves are performed independently on each grid
- The code is implemented to minimize the number of temporary arrays

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

Matrix-free multigrid provides fast pressure solves on dynamic overlapping grids







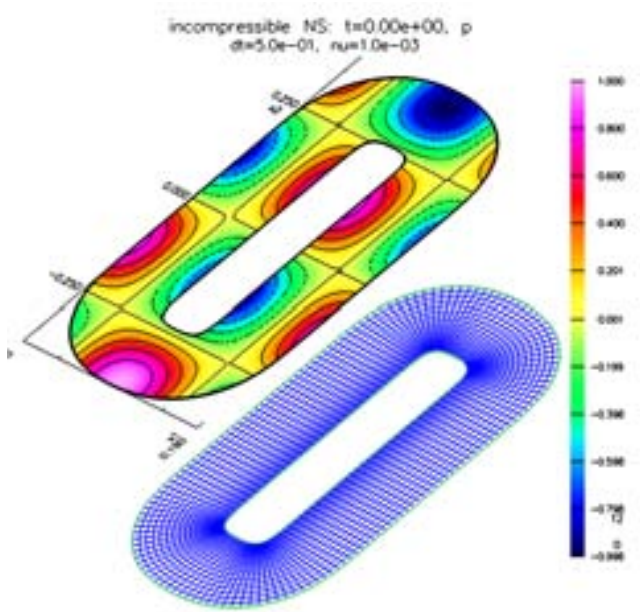
Multigrid exploits the grid & solver

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

Code Verification: an integral part of our development

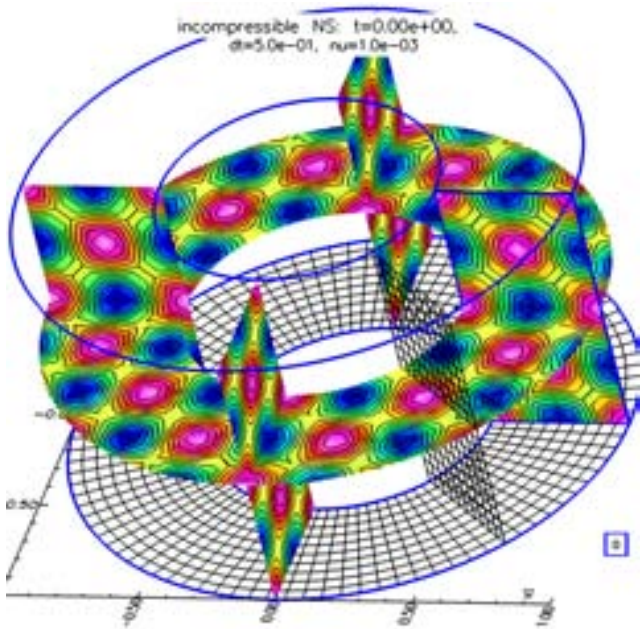
Twilight zone solutions (a.k.a. manufactured solutions) are extensively supported by the Overture framework. These tools are useful for code development as well as verification.

2D Twilight Zone



h_{max}	$\|e_p\|_\infty$	$\|e_u\|_\infty$	$\|e_v\|_\infty$
3.11e-02	1.59e-03	6.27e-03	7.34e-03
1.56e-02	1.12e-04	4.84e-04	5.17e-04
7.80e-03	6.13e-06	3.67e-05	3.02e-05
rate	4.01	3.71	3.97

3D Twilight Zone

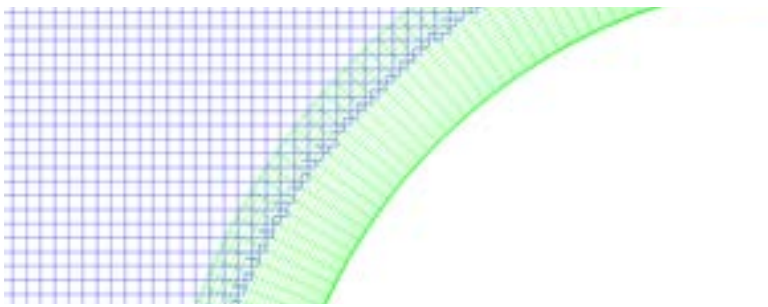


h_{max}	$\|e_p\|_\infty$	$\|e_u\|_\infty$	$\|e_v\|_\infty$	$\|e_w\|_\infty$
6.68e-02	4.81e-02	1.21e-01	5.40e-02	3.38e-02
3.34e-02	1.90e-03	5.75e-03	3.64e-03	1.45e-03
1.67e-02	5.96e-05	1.33e-04	6.73e-05	3.57e-05
rate	4.83	4.91	4.82	4.94

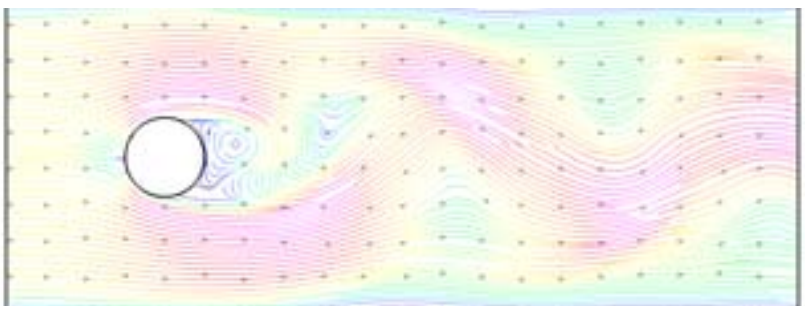
Demonstration: Re 200 flow past a cylinder in a channel

- 1e6 grid points, 300Mb
- computed to time 100 with cfl 4
- 17317 steps,
- 27hrs cpu time (desktop)

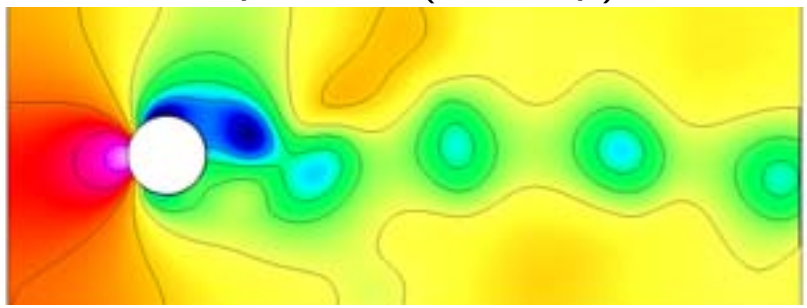
(a) domain



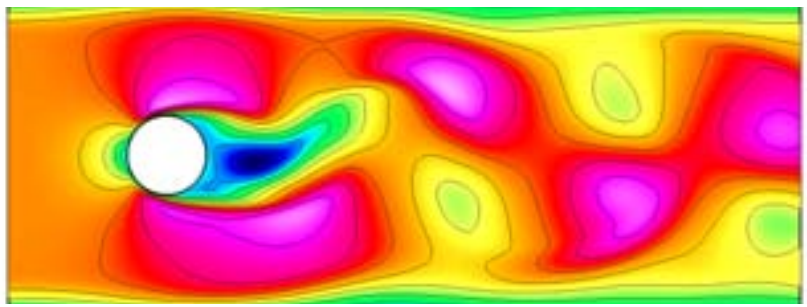
(b) grid closup



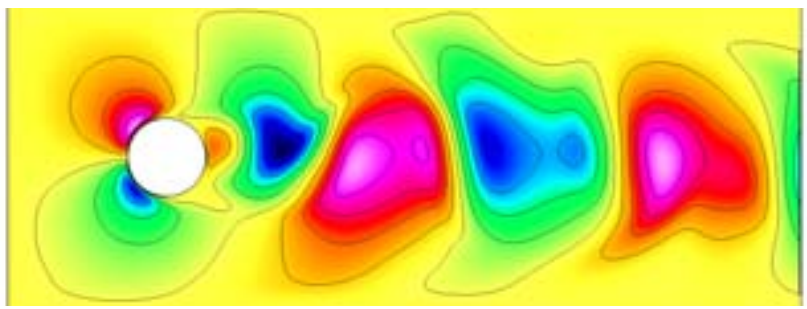
(c) streamlines



(d) pressure



(e) x-velocity



(f) y-velocity

Timing and memory comparison

The new algorithm has similar memory requirements to the existing explicit code, but can use a much larger timestep

	Δt	steps	memory	cpu-time
PC	$1.43e-3$	701	246Mb	3873s
AF	$6.25e-3$	161	305Mb	1010s

Table: Comparison of the explicit predictor-corrector (PC) and the compact approximate factorization (AF) scheme on the 2D flow past a cylinder problem with 1.02e6 grid points.

Currently profiling and optimizing the implementation

- Profiling suggests the AF scheme can be significantly optimized
- Multigrid pressure solve accounts for only 8% of the time in each case

Status and future work

Current status

- Initial high-order compact/AF scheme implemented
- Parallel moving grid generation and 4^{th} order accurate multigrid

This Year

- Parallel and moving grid AF scheme
- Implementation of LES models
- Release to collaborators

Subsequent years

- Linking to WRF for time dependent boundary conditions
- Integration of advanced wake models for large scale park simulations
- Highly-automated grid generation for terrain and turbines
- Public releases starting 2011

http://www.llnl.gov/casc/Overture