

# Solving Fluid Structure Interaction Problems on Overlapping Grids

Bill Henshaw

Center for Applied Scientific Computing,  
Lawrence Livermore National Laboratory, Livermore, CA, USA.

Massachusetts Institute of Technology,  
Cambridge Massachusetts, September 20, 2012.



# Acknowledgments.

## Primary collaborators for this effort:

Jeff Banks (LLNL), Kyle Chand (LLNL), Don Schwendeman (RPI), Björn Sjögreen (LLNL).

## Supported by:

**ASCR** Department of Energy, Office of Science, ASCR Applied Math Program .

**LDRD** LLNL: Laboratory Directed Research and Development (LDRD) program .

**NSF** National Science Foundation .

- 1 Background: overlapping grids, Overture and CG.
- 2 Incompressible flows and rigid bodies.
  - 1 high-order accurate factored scheme and boundary conditions.
  - 2 matrix-free multigrid.
  - 3 moving grid generation.
- 3 Compressible flow and light rigid bodies (partitioned algorithms).
  - 1 a one-dimensional model problem.
  - 2 the added-mass Newton Euler equations.
  - 3 overcoming the added-mass instability for light solids.
- 4 Compressible flow and deforming bodies (partitioned algorithms).
  - 1 Deforming Composite Grids (DCG).
  - 2 a one-dimensional model problem.
  - 3 an interface projection scheme (fluid-solid Riemann problem).

# The Overture project is developing PDE solvers for a wide class of continuum mechanics applications.

**Overture** is a toolkit for solving PDE's on overlapping grids and includes CAD, grid generation, numerical approximations, AMR and graphics.

The **CG** (Composite Grid) suite of PDE solvers (**cgns**, **cgins**, **cgmx**, **cgsm**, **cgad**, **cgmp**) provide algorithms for modeling gases, fluids, solids and E&M.

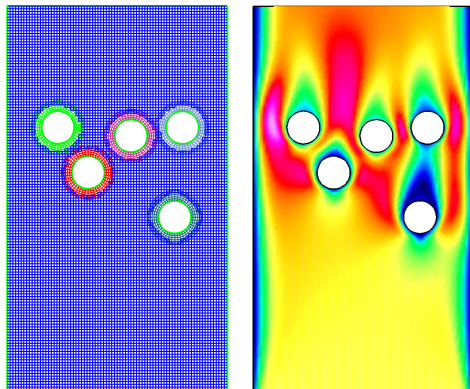
Overture and CG are available from [www.llnl.gov/CASC/Overture](http://www.llnl.gov/CASC/Overture).

We are looking at a variety of applications:

- wind turbines, building flows (**cgins**),
- explosives modeling (**cgns**),
- fluid-structure interactions (e.g. blast effects) (**cgmp+cgns+cgsm**),
- conjugate heat transfer (e.g. NIF holhraum) (**cgmp+cgins+cgad**),
- damage mitigation in NIF laser optics (**cgmx**).

# What are overlapping grids and why are they useful?

**Overlapping grid:** a set of structured grids that overlap.

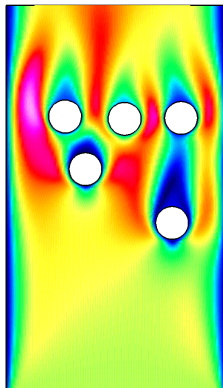
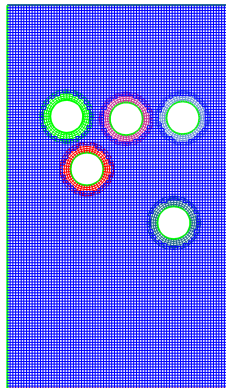


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Smooth grids for accuracy at boundaries.
- Efficient for high-order methods.



# What are overlapping grids and why are they useful?

**Overlapping grid:** a set of structured grids that overlap.

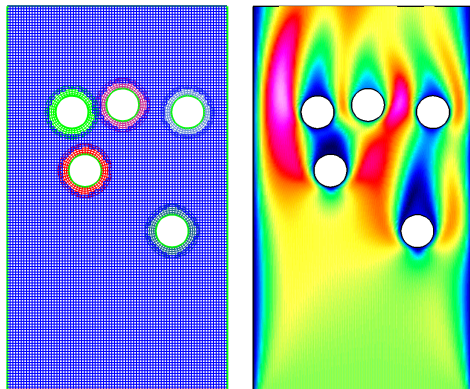


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Smooth grids for accuracy at boundaries.
- Efficient for high-order methods.



# What are overlapping grids and why are they useful?

**Overlapping grid:** a set of structured grids that overlap.

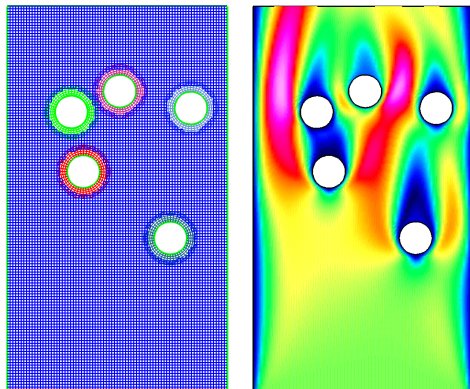


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Smooth grids for accuracy at boundaries.
- Efficient for high-order methods.



# What are overlapping grids and why are they useful?

**Overlapping grid:** a set of structured grids that overlap.



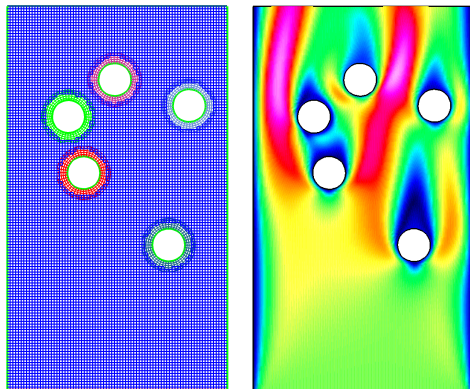
- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Smooth grids for accuracy at boundaries.
- Efficient for high-order methods.





# What are overlapping grids and why are they useful?

**Overlapping grid:** a set of structured grids that overlap.

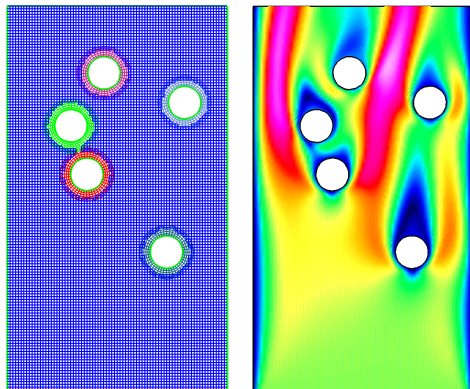


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Smooth grids for accuracy at boundaries.
- Efficient for high-order methods.



# What are overlapping grids and why are they useful?

**Overlapping grid:** a set of structured grids that overlap.

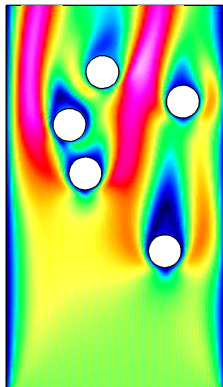
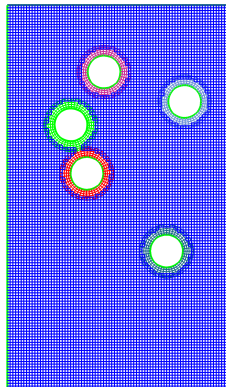


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Smooth grids for accuracy at boundaries.
- Efficient for high-order methods.



# What are overlapping grids and why are they useful?

**Overlapping grid:** a set of structured grids that overlap.

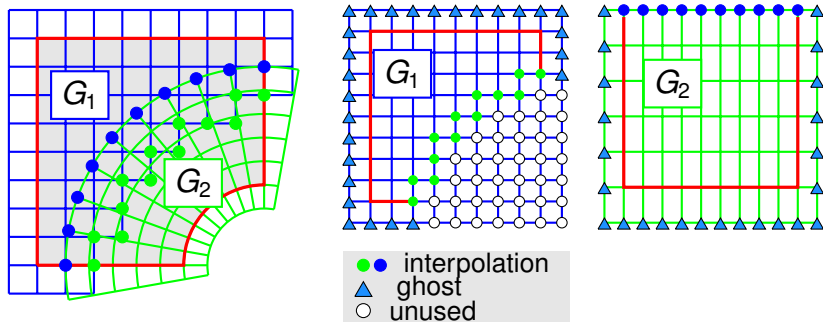


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Smooth grids for accuracy at boundaries.
- Efficient for high-order methods.

Asymptotic Performance Principle for overlapping grids

*As grids are refined, CPU/memory usage can approach that of a Cartesian grid.*

# Components of an overlapping grid



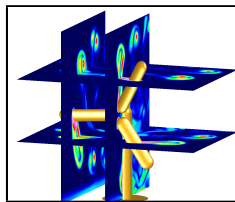
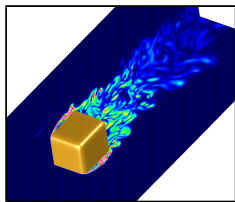
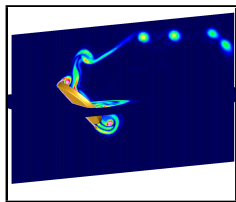
Left: an overlapping grid consisting of two structured curvilinear component grids,  $\mathbf{x} = G_1(\mathbf{r})$  and  $\mathbf{x} = G_2(\mathbf{r})$ . Middle and right: component grids for the square and annular grids in the unit square parameter space  $\mathbf{r}$ . Grid points are classified as discretization points, interpolation points or unused points. Ghost points are used to apply boundary conditions.


# Incompressible flows and rigid bodies.

# Cgins: an efficient solver for the incompressible Navier-Stokes (INS) equations

A parallel split-step solver is being developed based on:

- 1 Fourth-order accurate approximate-factored/compact time-stepping scheme for the momentum equations.
- 2 Fourth-order accurate multigrid solver for the pressure equation.
- 3 Fast overlapping grid generation for moving geometry.



- WDH., *A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids*, J. Comput. Phys, (1994).
- WDH, N.A. Petersson, *A Split-Step Scheme for the Incompressible Navier-Stokes Equations*, 2003.
- WDH., K. K. Chand, *A Composite Grid Solver for Conjugate Heat Transfer in Fluid-Structure Systems*, J. Comput. Phys, 2009. 

# Incompressible Navier-Stokes.

$$\begin{aligned}\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \nu \Delta \mathbf{u} - \mathbf{f} &= \mathbf{0}, & t > 0, \quad \mathbf{x} \in \Omega \\ \nabla \cdot \mathbf{u} &= 0 & t > 0, \quad \mathbf{x} \in \Omega\end{aligned}$$

Divergence damping term:  $\alpha \nabla \cdot \mathbf{u}$  is important.

Wall boundary conditions:

$$\mathbf{u} = \mathbf{0}, \quad \nabla \cdot \mathbf{u} = 0, \text{ (pressure BC)} \quad \mathbf{x} \in \partial\Omega,$$

with *numerical boundary condition*:

$$p_n = -\mathbf{n} \cdot (\nu \nabla \times \nabla \times \mathbf{u}) + \mathbf{n} \cdot \mathbf{f}.$$

Use  $-\nabla \times \nabla \times \mathbf{u}$  instead of  $\Delta \mathbf{u}$  for implicit time-stepping.

# Incompressible Navier-Stokes.

Split-step, velocity-pressure formulation:

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

Divergence damping term:  $\alpha \nabla \cdot \mathbf{u}$  is important.

Wall boundary conditions:

$$\mathbf{u} = 0, \quad \nabla \cdot \mathbf{u} = 0, \quad (\text{pressure BC}) \quad \mathbf{x} \in \partial\Omega,$$

with *numerical boundary condition*:

$$p_n = -\mathbf{n} \cdot (\nu \nabla \times \nabla \times \mathbf{u}) + \mathbf{n} \cdot \mathbf{f}.$$

Use  $-\nabla \times \nabla \times \mathbf{u}$  instead of  $\Delta \mathbf{u}$  for implicit time-stepping.



# Incompressible Navier-Stokes.

Split-step, velocity-pressure formulation:

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

Divergence damping term:  $\alpha \nabla \cdot \mathbf{u}$  is important.

Wall boundary conditions:

$$\mathbf{u} = 0, \quad \nabla \cdot \mathbf{u} = 0, \quad (\text{pressure BC}) \quad \mathbf{x} \in \partial\Omega,$$

with *numerical boundary condition*:

$$p_n = -\mathbf{n} \cdot (\nu \nabla \times \nabla \times \mathbf{u}) + \mathbf{n} \cdot \mathbf{f}.$$

Use  $-\nabla \times \nabla \times \mathbf{u}$  instead of  $\Delta \mathbf{u}$  for implicit time-stepping.

# Incompressible Navier-Stokes.

Split-step, velocity-pressure formulation:

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

Divergence damping term:  $\alpha \nabla \cdot \mathbf{u}$  is important.

Wall boundary conditions:

$$\mathbf{u} = 0, \quad \nabla \cdot \mathbf{u} = 0, \quad (\text{pressure BC}) \quad \mathbf{x} \in \partial\Omega,$$

with *numerical boundary condition*:

$$p_n = -\mathbf{n} \cdot (\nu \nabla \times \nabla \times \mathbf{u}) + \mathbf{n} \cdot \mathbf{f}.$$

Use  $-\nabla \times \nabla \times \mathbf{u}$  instead of  $\Delta \mathbf{u}$  for implicit time-stepping.

# Incompressible Navier-Stokes.

Split-step, velocity-pressure formulation:

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

Divergence damping term:  $\alpha \nabla \cdot \mathbf{u}$  is important.

Wall boundary conditions:

$$\mathbf{u} = 0, \quad \nabla \cdot \mathbf{u} = 0, \quad (\text{pressure BC}) \quad \mathbf{x} \in \partial\Omega,$$

with *numerical boundary condition*:

$$p_n = -\mathbf{n} \cdot (\nu \nabla \times \nabla \times \mathbf{u}) + \mathbf{n} \cdot \mathbf{f}.$$

Use  $-\nabla \times \nabla \times \mathbf{u}$  instead of  $\Delta \mathbf{u}$  for implicit time-stepping.

# PDE based numerical boundary conditions (NBCs)

- High-order FD schemes with wide stencils require additional NBCs.
- One-sided difference approximations can be less stable and accurate.

## NBC Principle

*Derive NBCs from the PDE and physical boundary conditions.*

Example (heat equation):

$$\begin{aligned}u_t &= \nu(u_{xx} + u_{yy}) + f(x, y, t), & 0 < x < 1, & -\infty < y < \infty, \\u(0, y, t) &= g(y, t), & \text{(Dirichlet boundary condition),} \\u_x(1, y, t) &= k(y, t). & \text{(Neumann boundary condition),}\end{aligned}$$

Compatibility conditions are used as NBCs:

$$\begin{aligned}g_t(y, t) &= \nu(u_{xx}(0, y, t) + g_{yy}(0, y, t)) + f(0, y, t), \\k_t(y, t) &= \nu(u_{xxx}(1, y, t) + k_{yy}(1, y, t)) + f_x(1, y, t).\end{aligned}$$

Normal mode theory shows why compatibility conditions can be applied to lower order accuracy.

# BCs and NBCs for the incompressible Navier-Stokes.

No-slip wall		
Condition	Type	Assigns
$\mathbf{u} = \mathbf{g}$	physical	bndry
$\nabla \cdot \mathbf{u} = 0$	physical	ghost
$\mathbf{t}_\mu \cdot \mathcal{L}(\mathbf{u}, p) = 0$	compatibility	ghost
$\partial_n(\nabla \cdot \mathbf{u}) = 0$	compatibility	ghost
Extrap $\mathbf{t}_\mu \cdot \mathbf{u}$	numerical	ghost
$p_n = \mathcal{P}(\mathbf{u})$	compatibility	ghost (p)
$\Delta p = G$	compatibility	bndry (p)
Extrap $p$	numerical	ghost (p)

$$\mathcal{L}(\mathbf{u}, p) := \mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p + \nu \nabla \times \nabla \times \mathbf{u} - \mathbf{f},$$

$$\mathcal{P}(\mathbf{u}) := \mathbf{n} \cdot (\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla \times \nabla \times \mathbf{u} - \mathbf{f}).$$

# Approximate factorization & compact discretizations - a key point is maintaining accuracy at boundaries.

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

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

becomes

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

- Compact schemes can be integrated into the AF solves
- Special “combined” compact schemes have been developed:  
→ reduce the number of factors

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

- preserve accuracy at boundaries
- 4<sup>th</sup> and 6<sup>th</sup> order accuracy with a 5 point stencil
- special penta-diagonal solvers that handle wider boundary stencils

# Smallest Scale LES (SSLES) model

- Our high-order accurate schemes all use central differences.
- Nonlinear dissipation is used to stabilize the schemes.

It can be proved [HKR] that the minimum scale of the INS locally satisfies

$$\lambda_{\min} \propto \sqrt{\frac{\nu}{\|\nabla \mathbf{u}\|_{\text{loc}} + c}}.$$

Setting  $h = \lambda_{\min}$  leads to the non-linear dissipation operators:

$$\mathcal{D}_2(\mathbf{u}_i) = (a_{21} + a_{22}\|\nabla_h \mathbf{u}_i\|)h^2 \Delta_h \mathbf{u}_i,$$

$$\mathcal{D}_4(\mathbf{u}_i) = (a_{41} + a_{42}\|\nabla_h \mathbf{u}_i\|)h^4 (-\Delta_h^2) \mathbf{u}_i.$$

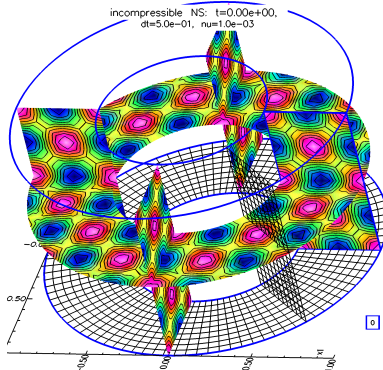
- 1 These serve as simple LES models (Smagorinsky type).
- 2 These terms stabilize the scheme even if  $\nu = 0$ .

[HKR1] *On the smallest scale for the incompressible Navier-Stokes equations*, WDH, H.-O. Kreiss, L.G.M. Reyna, Theoret. Comput. Fluid Dynamics, 1989.

[HKR2] *Smallest scale estimates for the incompressible Navier-Stokes equations*, WDH, H.-O. Kreiss, L.G.M. Reyna, Arch. Rational Mech. Anal., 1990.

# Code verification: an integral part of our development

Twilight zone solutions (a.k.a. manufactured solutions) are extensively supported by the Overture framework.



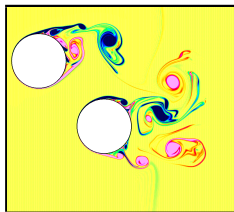
$h_{max}$	$ \epsilon_p _\infty$	$ \epsilon_u _\infty$	$ \epsilon_v _\infty$	$ \epsilon_w _\infty$	$ \nabla \cdot \mathbf{u} _\infty$
1.34e-01	1.91e-01	3.42e-01	2.14e-01	1.23e-01	2.80e+00
6.68e-02	1.20e-02 (15.9)	1.84e-02 (18.6)	1.01e-02 (21.2)	6.43e-03 (19.1)	2.71e-01 (10.3)
3.34e-02	1.02e-03 (11.8)	1.49e-03 (12.3)	7.50e-04 (13.5)	3.01e-04 (21.4)	3.10e-02 (8.7)
1.67e-02	7.90e-05 (12.9)	1.25e-04 (11.9)	6.55e-05 (11.5)	1.74e-05 (17.3)	4.00e-03 (7.8)
rate	3.7	3.8	3.9	4.3	3.1





# AFS performance: flow past two cylinders.

Grid tcilce64, $N_g = 22\text{M}$ grid-points					
Method	P-solver	$N_p$	TPSM	TTS	RPG
PC24	BICGS(5)	16	840	16000	205
PC24	MG	16	15	210	67
AFS24	MG	16	34	54	32



- 1 Speed-up from baseline PC24-BICGS(5) to AFS24-MG is  $16000/54 \approx 300$ .
- 2 Memory is reduced by a factor of  $205/32 \approx 6$ .

$N_p$  : number of processors.

$N_g$  : number of grid-points.

TPS : CPU (s) time-per-step.

TPSM : CPU time (s) per-step, per-million-grid-points, per-processor =  $N_p \times TPS / (N_g / 10^6)$ .

TTS : normalized time-to-solution = TPSM / CFL-number.

RPG : memory usage in real-per-grid-point.

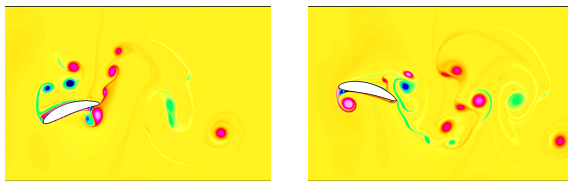
PC24: predictor-corrector, 2nd-order time, 4th-order space.

AFS24: approximate-factored scheme, 2nd-order time, 4th-order space.

BICGS(5) : bi-Conjugate-Gradient-Stabilized with ILU(5) preconditionner.



# AFS results - pitching plunging airfoil



A pitching and plunging airfoil, computed in parallel with Cgins. Contour plots of the vorticity.

Grid joukowsky32, $N_g = 7\text{M}$ grid-points					
Method	P-solver	$N_p$	TPSM	TTS	RPG
AFS24	BICGS(5)	16	1100	1800	235
AFS24	MG	16	52	76	66

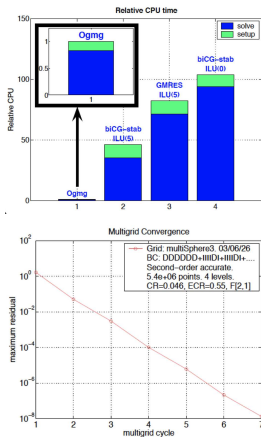
- 1 Speed-up from baseline AFS24-BICGS(5) to AFS24-MG is  $1800/76 \approx 24$ .
- 2 Memory is reduced by a factor of  $235/66 \approx 3.5$ .

# Fourth-order Accurate Parallel Multigrid Solver for the Pressure Equation.

# The Omgm overlapping grid multigrid solver has been extended to 4th-order accuracy and parallel.

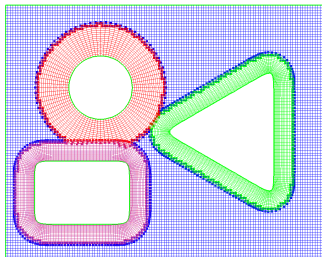
Omgm is many times faster than Krylov methods.

- matrix-free; optimized for Cartesian grids.
- automatic coarse grid generation.
- adaptive smoothing
  - variable sub-smooths per component grid.
  - interpolation-boundary smoothing (IBS).
- Galerkin coarse grid operators (operator averaging).
- *PDE-based* numerical boundary conditions for Dirichlet and Neumann problems.

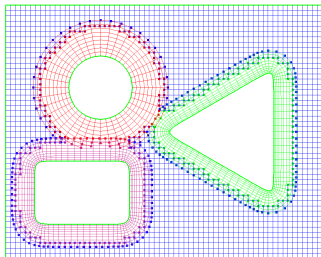
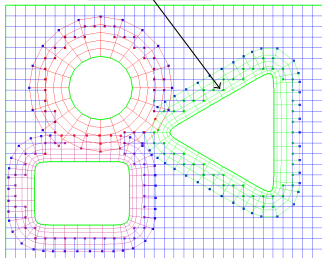


WDH., *On Multigrid For Overlapping Grids*, SIAM J. Sci. Comput. **26**, no. 5, (2005) 1547–1572.

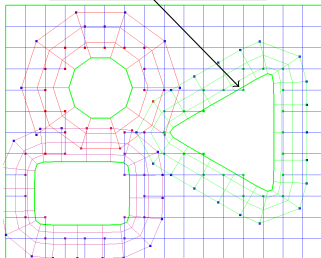
# Automatic coarse grid generation is a key feature.



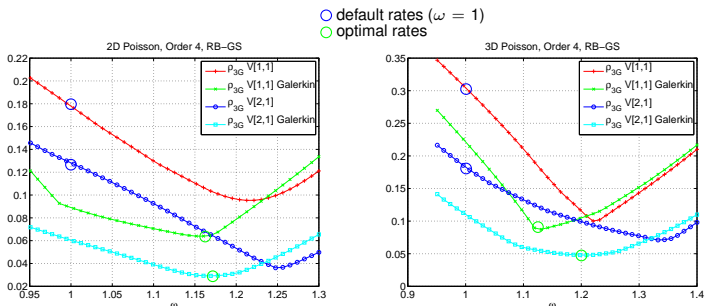
overlap increases



interpolation accuracy reduced



# Local Fourier analysis significantly improves convergence rates. Over-relaxed Red-Black smoothers and Galerkin coarse grid operators.



Three-grid multigrid convergence rates as a function  $\omega$ .

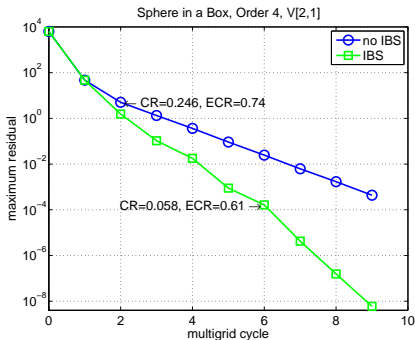
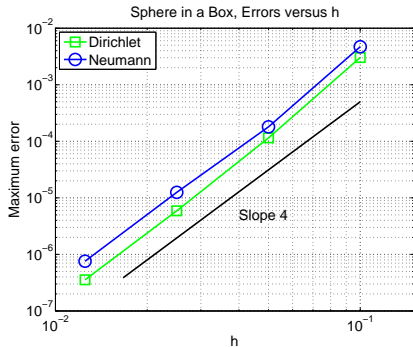
$\omega$  : relaxation parameter in Red-Black Gauss-Seidel smoother.

$\rho_{3G}$  : convergence rate per cycle for a 3 grid (i.e. 3 level) MG.

$V[m, n]$  : MG V-cycle,  $m$  pre-smooths and  $n$ -post smooths.

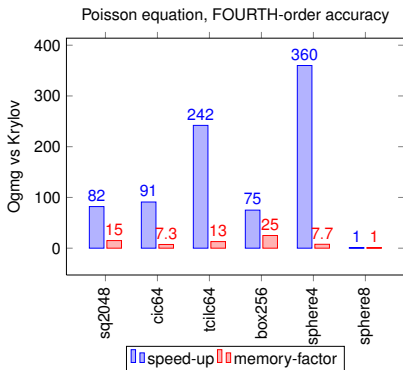
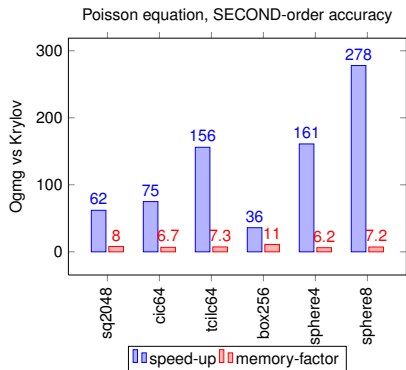
Galerkin : Galerkin coarse grid operators.

# Accuracy and convergence of the new fourth-order accurate parallel version of Ogmig.



# Multigrid is much faster than Krylov based methods.

And uses much less memory.



Performance of Ogm for solving Poisson's equation on various grids. Ogm is compared to a Krylov solver (bi-CG stab, ILU(1)) from PETSc.



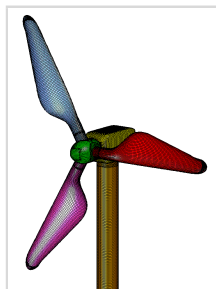


# Rapid grid generation for moving geometry is critical.

Ogen is the overlapping grid generator in Overture.

Overlapping grid generation consists of two major steps:

- 1 construct component grids (Mappings).
- 2 grid connectivity: cut holes and determine interpolation information using Ogen (this is the step that requires most of the CPU time).



In recent work changes have been made to support

- the generation of large (billion point +) grids.
- parallel moving-grid flow simulations (Ogen is called at each time step).

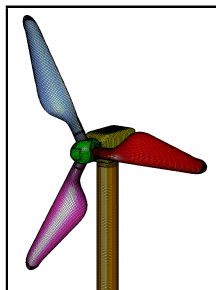
Grid	order of accuracy	grid points	processors (nodes $\times$ p/n)	cpu (s)
Sphere in a box	2	2.1 billion	16 (8 $\times$ 2)	136
Re-entry vehicle	4	215 million	128 (16 $\times$ 8)	1990

# Rapid grid generation for moving geometry is critical.

Ogen is the overlapping grid generator in Overture.

Overlapping grid generation consists of two major steps:

- 1 construct component grids (Mappings).
- 2 grid connectivity: cut holes and determine interpolation information using Ogen (this is the step that requires most of the CPU time).



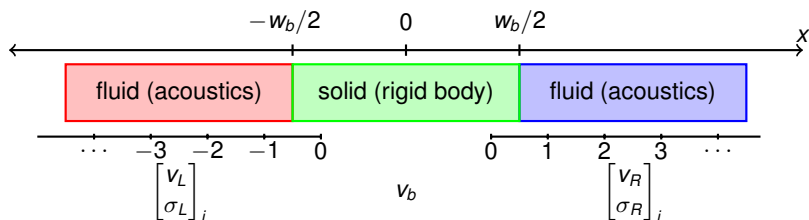
In recent work changes have been made to support

- the generation of large (billion point +) grids.
- parallel moving-grid flow simulations (Ogen is called at each time step).

Grid	order of accuracy	grid points	processors (nodes $\times$ p/n)	cpu (s)
Sphere in a box	2	2.1 billion	16 (8 $\times$ 2)	136
Re-entry vehicle	4	215 million	128 (16 $\times$ 8)	1990

# FSI for compressible fluids and light rigid bodies (partitioned algorithms).

# Light rigid body in a fluid - model problem



Fluid domains (acoustics):

$$\frac{\partial}{\partial t} \begin{bmatrix} v_k \\ \sigma_k \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ \rho c^2 & 0 \end{bmatrix} \frac{\partial}{\partial x} \begin{bmatrix} v_k \\ \sigma_k \end{bmatrix} = 0, \quad k=L,R,$$

Rigid body:

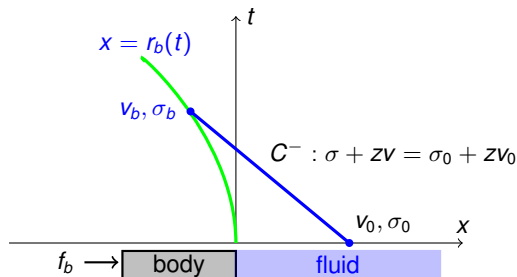
$$m_b \dot{v}_b = \mathcal{F}, \quad \mathcal{F} = \sigma_R|_{x=w_b/2} - \sigma_L|_{x=-w_b/2}$$

Interface conditions:

$$v_L|_{x=-w_b/2} = v_b, \quad v_R|_{x=w_b/2} = v_b.$$



# The fluid-solid Riemann problem for a rigid body.



From characteristics we obtain the fluid force on the body:

$$\sigma(r_b, t) = \sigma(r_b+, t-) + z(v(r_b+, t-) - v_b(t)). \quad (1)$$

where  $z = \rho c$  is the fluid impedance.

Equation (1) defines our stress projection.

Note the dependence of the stress on the velocity of the body  $v_b$ .

# Light rigid body in a fluid - algorithm

First order upwind scheme

$$\begin{bmatrix} v_k \\ \sigma_k \end{bmatrix}_i^{n+1} = \begin{bmatrix} v_k \\ \sigma_k \end{bmatrix}_i^n + \Delta t R_k \Lambda_k^- R_k^{-1} D_- \begin{bmatrix} v_k \\ \sigma_k \end{bmatrix}_i^n + \Delta t R_k \Lambda_k^+ R_k^{-1} D_+ \begin{bmatrix} v_k \\ \sigma_k \end{bmatrix}_i^n,$$

Partitioned time-stepping algorithm (new scheme:  $\alpha = z$ , traditional:  $\alpha = 0$ ),

- 1 Advance fluid domains:  $\begin{bmatrix} v_k \\ \sigma_k \end{bmatrix}_i^{n+1}$
- 2 Set  $\mathcal{F}^{n+1} = \sigma_{R,1}^{n+1} + \alpha(v_{R,1}^{n+1} - v_b^{n+1}) - \sigma_{L,-1}^{n+1} - \alpha(v_b^{n+1} - v_{L,-1}^{n+1})$ ,
- 3 Solve for rigid body:  $m_b v_b^{n+1} = m_b v_b^n + \Delta t \mathcal{F}^{n+1}$ ,

$$v_b^{n+1} = \left[ m_b + 2\Delta t \alpha \right]^{-1} \left[ m_b v_b^n + \Delta t \left( \sigma_{R,1}^{n+1} - \sigma_{L,-1}^{n+1} + \alpha(v_{R,1}^{n+1} + v_{L,-1}^{n+1}) \right) \right].$$

- 4 Set ghost values from projection.

Note the *added mass* term  $2\Delta t \alpha$ .

## Theorem

*The first-order upwind scheme with projection parameter  $\alpha = z$  is stable for  $\lambda = c \frac{\Delta t}{\Delta x} < 1$  and any  $m_b \geq 0$ .*

## Theorem

*The first-order upwind scheme with projection parameter  $\alpha = 0$  has the time-step restriction  $\lambda < 1$  and  $\Delta t < m_b(4 - \lambda)/(z\lambda)$ .*

**Note 1:** The projection scheme ( $\alpha = z$ ) remains stable with the usual time-step restriction for any  $m_b \geq 0$ .

**Note 2:** The traditional scheme ( $\alpha = 0$ ) is formally stable for  $m_b > 0$  but the time-step  $\Delta t$  goes to zero at  $m_b \rightarrow 0$ .

# The Newton-Euler equations for rigid body dynamics

The equations of motion for the rigid body are the Newton-Euler equations:

$$\begin{aligned}\dot{\mathbf{x}}_b &= \mathbf{v}_b, \\ m_b \dot{\mathbf{v}}_b &= \mathcal{F}, \\ A\dot{\boldsymbol{\omega}} &= -WA\boldsymbol{\omega} + \mathcal{T}, \\ \dot{E} &= WE.\end{aligned}$$

$m_b$  : mass of the body,

$\mathbf{x}_b(t) \in \mathbb{R}^3$  : position of the center of mass,

$\mathbf{v}_b(t) \in \mathbb{R}^3$  : is the velocity of the center of mass,

$\boldsymbol{\omega}(t) \in \mathbb{R}^3$  : is the angular velocity,

$A(t) \in \mathbb{R}^{3 \times 3}$  : moment of inertia matrix,

$E(t) \in \mathbb{R}^{3 \times 3}$  : matrix with columns being the principle axes of inertia,

$W(t) \in \mathbb{R}^{3 \times 3}$  : angular velocity matrix,  $W\mathbf{a} = \boldsymbol{\omega} \times \mathbf{a}$ ,

$\mathcal{F}(t) \in \mathbb{R}^3$  : resultant force on the body,

$\mathcal{T}(t) \in \mathbb{R}^3$  : resultant torque the body.



## Extension to 3D: added mass matrices

For a point  $\mathbf{r}$  on the rigid body in 3D, the interface projection is

$$-\rho(\mathbf{r}, t)\mathbf{n} = -\rho_f\mathbf{n} + z_f [\mathbf{n}^T(\mathbf{v}_f - \mathbf{v}(\mathbf{r}, t))] \mathbf{n}.$$

Force and torque:  $(\mathbf{v}(\mathbf{r}, t) = \mathbf{v}_b(t) - \mathbf{y}(\mathbf{r}, t) \times \boldsymbol{\omega}(t))$ ,

$$\mathcal{F} = \int_{\mathbf{r} \in \partial B} -\rho(\mathbf{r}, t)\mathbf{n} \, ds = -A^{vv}\mathbf{v}_b - A^{v\omega}\boldsymbol{\omega} + \tilde{\mathcal{F}},$$

$$\mathcal{T} = \int_{\mathbf{r} \in \partial B} \mathbf{y} \times (-\rho(\mathbf{r}, t)\mathbf{n}) \, ds = -A^{\omega v}\mathbf{v}_b - A^{\omega\omega}\boldsymbol{\omega} + \tilde{\mathcal{T}}$$

where  $\mathbf{y} = \mathbf{r} - \mathbf{x}_b(t)$  and  $A^{ij}$  are the **added-mass** matrices

$$A^{vv} = \int_{\partial B} z_f \mathbf{n} \mathbf{n}^T \, ds, \quad A^{v\omega} = \int_{\partial B} z_f \mathbf{n} (\mathbf{Y} \mathbf{n})^T \, ds, \quad (2)$$

$$A^{\omega v} = \int_{\partial B} z_f (\mathbf{Y} \mathbf{n}) \mathbf{n}^T \, ds, \quad A^{\omega\omega} = \int_{\partial B} z_f \mathbf{Y} \mathbf{n} (\mathbf{Y} \mathbf{n})^T \, ds, \quad (3)$$

and where  $\mathbf{Y}$  is the matrix corresponding to  $\mathbf{y} \times$ .



# Extension to 3D: added mass matrices

For a point  $\mathbf{r}$  on the rigid body in 3D, the interface projection is

$$-\rho(\mathbf{r}, t)\mathbf{n} = -\rho_f\mathbf{n} + z_f [\mathbf{n}^T (\mathbf{v}_f - \mathbf{v}(\mathbf{r}, t))] \mathbf{n}.$$

Force and torque:  $(\mathbf{v}(\mathbf{r}, t) = \mathbf{v}_b(t) - \mathbf{y}(\mathbf{r}, t) \times \boldsymbol{\omega}(t))$ ,

$$\mathcal{F} = \int_{\mathbf{r} \in \partial B} -\rho(\mathbf{r}, t)\mathbf{n} \, ds = -A^{VV}\mathbf{v}_b - A^{V\omega}\boldsymbol{\omega} + \tilde{\mathcal{F}},$$

$$\mathcal{T} = \int_{\mathbf{r} \in \partial B} \mathbf{y} \times (-\rho(\mathbf{r}, t)\mathbf{n}) \, ds = -A^{\omega V}\mathbf{v}_b - A^{\omega\omega}\boldsymbol{\omega} + \tilde{\mathcal{T}}$$

where  $\mathbf{y} = \mathbf{r} - \mathbf{x}_b(t)$  and  $A^{ij}$  are the **added-mass** matrices

$$A^{VV} = \int_{\partial B} z_f \mathbf{n} \mathbf{n}^T \, ds, \quad A^{V\omega} = \int_{\partial B} z_f \mathbf{n} (\mathbf{Y} \mathbf{n})^T \, ds, \quad (2)$$

$$A^{\omega V} = \int_{\partial B} z_f (\mathbf{Y} \mathbf{n}) \mathbf{n}^T \, ds \quad A^{\omega\omega} = \int_{\partial B} z_f \mathbf{Y} \mathbf{n} (\mathbf{Y} \mathbf{n})^T \, ds, \quad (3)$$

and where  $\mathbf{Y}$  is the matrix corresponding to  $\mathbf{y} \times$ .



# The added-mass Newton Euler Equations

The rigid body equations become:

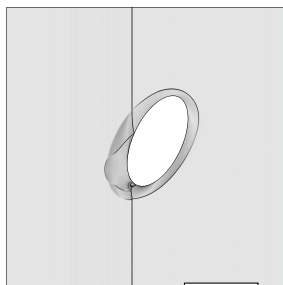
$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & m_b l & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_b \\ \dot{\mathbf{v}}_b \\ \dot{\omega} \\ \dot{E} \end{bmatrix} + \begin{bmatrix} 0 & -I & 0 & 0 \\ 0 & A^{v\omega} & A^{v\omega} & 0 \\ 0 & A^{\omega v} & A^{\omega\omega} + WA & 0 \\ 0 & 0 & 0 & -W \end{bmatrix} \begin{bmatrix} \mathbf{x}_b \\ \mathbf{v}_b \\ \omega \\ E \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathcal{F}} \\ \tilde{\mathcal{T}} \\ 0 \end{bmatrix}. \quad (4)$$

**Note:** These *added-mass* Newton-Euler equations can be solved using an implicit time-stepping scheme.

# Multi-dimensional algorithm

- 1 second-order accurate Godunov based scheme for the (reactive) Euler equations.
- 2 Implicit Runge-Kutta (DIRK) time-stepping for added-mass rigid-body equations.
- 3 Second-order accurate in the max norm for smooth solutions (with limiter turned off).
- 4 First-order accurate in the discrete  $L_1$ -norm for problems with shocks.
- 5 Adaptive mesh refinement.
- 6 Equations are solved in the moving grid coordinate system.

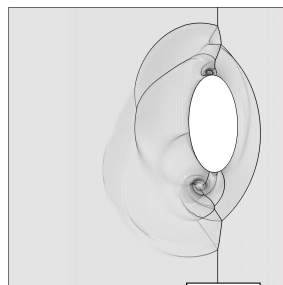
# Shock driven zero mass ellipse



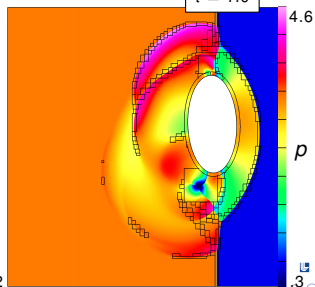
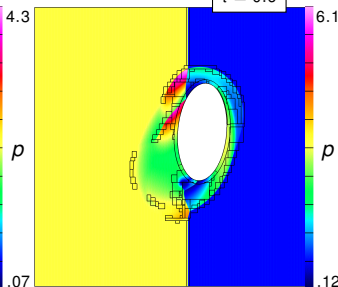
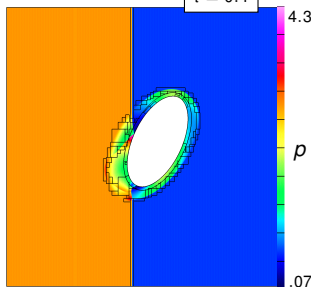
$t = 0.4$



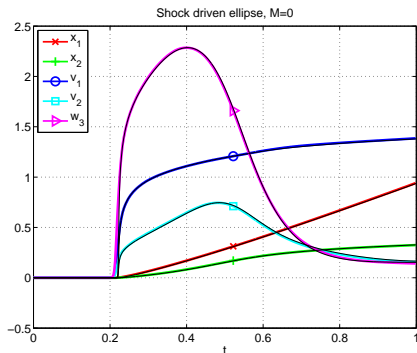
$t = 0.6$



$t = 1.0$



# Shock driven zero mass ellipse



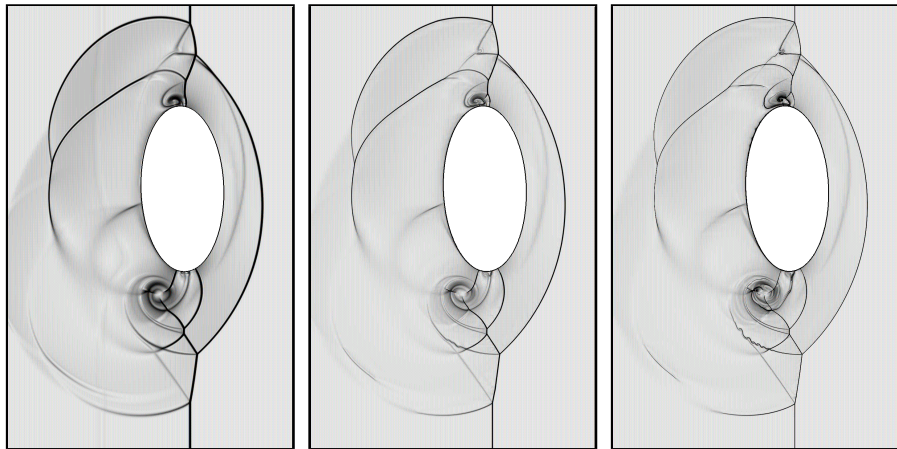
Time histories of rigid-body properties. Colour: grid  $\mathcal{G}_{re}^{(8)}$ , black: fine grid  $\mathcal{G}_{re}^{(32)}$

Grid $\mathcal{G}^{(j)}$	$h_j$	$e_\rho^{(j)}$	$r$	$e_u^{(j)}$	$r$	$e_v^{(j)}$	$r$	$e_p^{(j)}$	$r$
$\mathcal{G}_{re}^{(8)}$	1/40	2.1e-3		9.3e-4		9.6e-4		2.1e-3	
$\mathcal{G}_{re}^{(16)}$	1/80	9.9e-4	2.1	4.3e-4	2.1	4.6e-4	2.1	9.6e-4	2.2
$\mathcal{G}_{re}^{(32)}$	1/160	4.7e-4	2.1	2.0e-4	2.1	2.2e-4	2.1	4.5e-4	2.2
rate		1.08		1.09		1.07		1.11	

A posteriori estimated errors ( $L_1$ -norm) and convergence rates for the accelerated ellipse at  $t = 1.0$ .



# Shock driven zero mass ellipse



Coarse grid  $\mathcal{G}_{re}^{(32)}$  (left), medium grid  $\mathcal{G}_{re}^{(16 \times 4)}$  (middle), fine (AMR) grid  $\mathcal{G}_{re}^{(8 \times 4 \times 4)}$  (right).



# FSI for compressible fluids and deforming elastic solids (partitioned algorithms).



# Deforming composite grids (DCG) for Fluid-Structure Interactions (FSI)

**Goal:** To perform coupled simulations of compressible fluids and deforming solids.

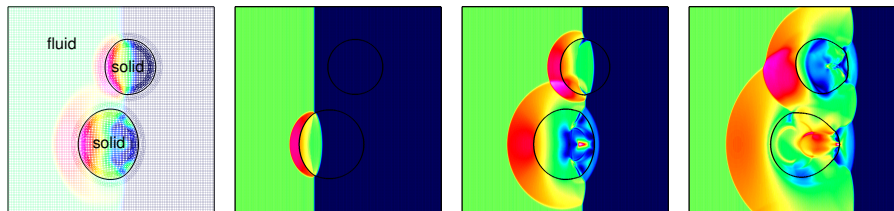
A mixed Eulerian-Lagrangian approach:

- Fluids: general moving coordinate system with overlapping grids.
- Solids : fixed reference frame with overlapping-grids (later: unstructured-grids, or beam/plate models).
- Boundary fitted deforming grids for fluid-solid interfaces.

Strengths of the approach:

- maintains high quality grids for large deformations/displacements.
- efficient structured grid methods (AMR) optimized for Cartesian grids.

# A sample FSI-DCG simulation



Mach 2 shock in a gas hitting two elastic cylinders.

- Solve Euler equations in the fluid domains on moving grids.
- Solve equations of linear elasticity in the solid domains.
- Fluid grids at the interface deform over time (hyperbolic grid generator).

**Fluid solver:** we solve the inviscid Euler equations with a second-order extension of Godunov's method (cgcn).

- WDH., D. W. Schwendeman, *Parallel Computation of Three-Dimensional Flows using Overlapping Grids with Adaptive Mesh Refinement*, J. Comp. Phys. **227** (2008).
- WDH., DWS, *Moving Overlapping Grids with Adaptive Mesh Refinement for High-Speed Reactive and Nonreactive Flow*, J. Comp. Phys. **216** (2005).
- WDH., DWS, *An adaptive numerical scheme for high-speed reactive flow on overlapping grids*, J. Comp. Phys. **191** (2003).

**Solid solver:** we solve the elastic wave equation as a first order system with a second-order upwind characteristic scheme (cgsm).

- Daniel Appelö, JWB, WDH, DWS, *Numerical Methods for Solid Mechanics on Overlapping Grids: Linear Elasticity*, J. Comp. Phys., (2012)

# The elastic piston - a 1D FSI model problem



$\bar{\rho}$  : density

$\bar{u}$  : displacement

$\bar{v}$  : velocity

$\bar{\sigma}$  : stress

$c_p$  : speed of sound

$\bar{z} = \bar{\rho}c_p$  : impedance

$\rho$  : density

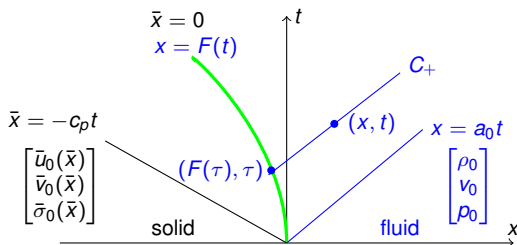
$v$  : velocity

$\sigma = -p$  : stress, pressure

$a$  : speed of sound

$z = \rho a$  : impedance

# The elastic piston



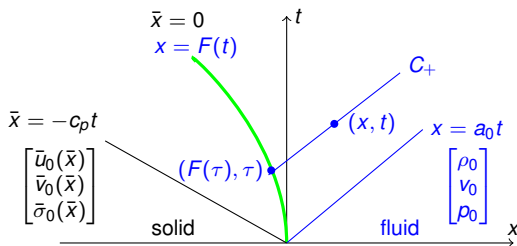
The governing equations for the solid and fluid are

$$\left\{ \begin{array}{l} \bar{u}_t - \bar{v} = 0 \\ \bar{v}_t - \bar{\sigma}_{\bar{x}} / \bar{\rho} = 0, \text{ for } \bar{x} < 0, \\ \bar{\sigma}_t - \bar{\rho} c_p^2 \bar{v}_{\bar{x}} = 0 \end{array} \right. \quad \left\{ \begin{array}{l} \rho_t + (\rho v)_x = 0 \\ (\rho v)_t + (\rho v^2 + p)_x = 0, \text{ for } x > F(t), \\ (\rho E)_t + (\rho E v + p v)_x = 0 \end{array} \right.$$

where  $\rho E = p/(\gamma - 1) + \rho v^2/2$ . The interface conditions are

$$\begin{aligned} \bar{v}(0, t) &= v(F(t), t), \\ \bar{\sigma}(0, t) &= \sigma(F(t), t) \equiv -p(F(t), t) + p_e. \end{aligned}$$

# The elastic piston



The governing equations for the solid and fluid are

$$\left\{ \begin{array}{l} \bar{u}_t - \bar{v} = 0 \\ \bar{v}_t - \bar{\sigma}_{\bar{x}} / \bar{\rho} = 0, \text{ for } \bar{x} < 0, \\ \bar{\sigma}_t - \bar{\rho} c_p^2 \bar{v}_{\bar{x}} = 0 \end{array} \right. \quad \left\{ \begin{array}{l} \rho_t + (\rho v)_x = 0 \\ (\rho v)_t + (\rho v^2 + p)_x = 0, \text{ for } x > F(t), \\ (\rho E)_t + (\rho E v + p v)_x = 0 \end{array} \right.$$

where  $\rho E = p / (\gamma - 1) + \rho v^2 / 2$ . The interface conditions are

$$\begin{aligned} \bar{v}(0, t) &= v(F(t), t), \\ \bar{\sigma}(0, t) &= \sigma(F(t), t) \equiv -p(F(t), t) + p_e. \end{aligned}$$

# An exact solution to the elastic piston problem

For a given  $x = F(t)$ , and constant  $\rho_0, p_0, v_0 = 0$ , the solution in the fluid region  $F(t) < x < a_0 t$  is (assuming no shocks)

$$v(x, t) = \dot{F}(\tau(x, t)), \quad \frac{a(x, t)}{a_0} = 1 + \frac{\gamma - 1}{2} \left( \frac{v(x, t)}{a_0} \right), \quad \frac{p(x, t)}{p_0} = \left( \frac{\rho(x, t)}{\rho_0} \right)^\gamma = \left( \frac{a(x, t)}{a_0} \right)^{2\gamma/(\gamma-1)},$$

$$x - F(\tau) = \left[ a_0 + \frac{\gamma + 1}{2} \dot{F}(\tau) \right] (t - \tau).$$

The general solution for the solid follows from the d'Alembert solution,

$$\bar{u}(\bar{x}, t) = f(\bar{x} - c_p t) + g(\bar{x} + c_p t),$$

$$f(\xi) = \frac{1}{2} [\bar{u}_0(\xi) - \frac{1}{c_p} \int_0^\xi \bar{v}_0(s) ds] \quad \text{for } \xi < 0,$$

$$g(\xi) = \begin{cases} \frac{1}{2} [\bar{u}_0(\xi) + \frac{1}{c_p} \int_0^\xi \bar{v}_0(s) ds] & \text{for } \xi < 0, \\ F(\xi/c_p) - f(-\xi) & \text{for } \xi > 0. \end{cases}$$

Applying the interface equations gives an ODE for  $F(t)$  in terms of the initial conditions,

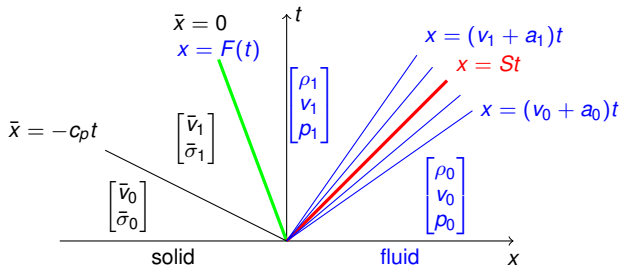
$$\frac{p_0}{\bar{\rho} c_p^2} \left[ 1 + \frac{\gamma - 1}{2a_0} \dot{F}(t) \right]^{2\gamma/(\gamma-1)} + \frac{\dot{F}(t)}{c_p} = -[\bar{u}'_0(-c_p t) - \frac{1}{c_p} \bar{v}_0(-c_p t)], \quad \text{for } t > 0.$$

Alternatively if we choose  $F(t) = -\frac{a}{q} t^q$ , we can choose initial conditions in the solid as

$$\bar{u}_0(\bar{x}) = -\frac{p_0}{\bar{\rho}_0 c_p^2} \int_0^{\bar{x}} \left[ 1 + \frac{\gamma - 1}{2a_0} \dot{F}(-s/c_p) \right]^{2\gamma/(\gamma-1)} ds, \quad \bar{v}_0(\bar{x}) = \dot{F}(-\bar{x}/c_p), \quad \text{for } \bar{x} < 0,$$

to give a smooth solution with the specified interface motion.

# The solution of the fluid-solid Riemann (FSR) problem defines our interface projection.



- special case of elastic piston problem - constant initial conditions in fluid and solid.
- the fluid may have a shock or expansion fan on the  $\mathcal{C}^+$  characteristic.



# Solution of the linearized fluid-solid Riemann problem

Characteristic relations:

$$\text{Solid: } \bar{z}\bar{v} \mp \bar{\sigma} = \bar{z}\bar{v}_0 \mp \bar{\sigma}_0, \text{ on } d\bar{x}/dt = \pm c_p,$$

$$\text{Fluid: } zv \mp \sigma = zv_0 \mp \sigma_0, \text{ on } dx/dt = v_0 \pm a_0,$$

where  $z = \bar{\rho}c_p$  and  $\bar{z} = \bar{\rho}a_0$  are the *acoustic impedances*.

The state next to the interface is an impedance weighted average of the fluid and solid states:

$$v_1 = \bar{v}_1 = \frac{\bar{z}\bar{v}_0 + zv_0}{\bar{z} + z} + \frac{\sigma_0 - \bar{\sigma}_0}{\bar{z} + z},$$
$$\sigma_1 = \bar{\sigma}_1 = \frac{\bar{z}^{-1}\bar{\sigma}_0 + z^{-1}\sigma_0}{\bar{z}^{-1} + z^{-1}} + \frac{v_0 - \bar{v}_0}{\bar{z}^{-1} + z^{-1}}.$$

The solution to the full nonlinear problem can also be determined.

# The FSI-DCG time-stepping algorithm

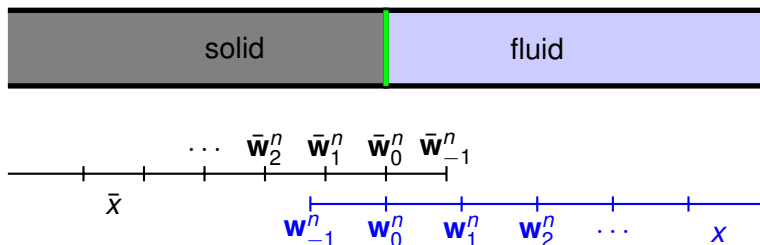
The FSI-DCG interface approximation is an extension of the scheme of Banks and Sjögreen:

J. W. Banks and B. Sjögreen, *A Normal Mode Stability Analysis of Numerical Interface Conditions for Fluid/Structure Interaction*, Commun. Comput. Phys., 2011.

The main steps are:

- 1 The fluid and solid domains are first advanced independently giving provisional interface values.
- 2 The provisional interface values are projected based on the solution to the fluid-solid Riemann problem.

# The FSI-DCG time-stepping algorithm



Define discrete approximations, ( $v_i^n \approx v(i\Delta x_i, n\Delta t)$ ,  $\bar{v}_i^n \approx \bar{v}(i\Delta \bar{x}_i, n\Delta t)$ )

$$\text{Solid: } \bar{\mathbf{w}}_i^n = [\bar{u}_i^n, \bar{v}_i^n, \bar{\sigma}_i^n] \quad i = -1, 0, 1, \dots$$

$$\text{Fluid: } \mathbf{w}_i^n = [\rho_i^n, v_i^n, p_i^n], \quad i = -1, 0, 1, \dots$$

# The FSI-DCG time-stepping algorithm

stage	condition	type	assigns
1a	Predict grid and grid velocity	extrapolation	$F^p, \dot{F}^p$
1b	Advance $\mathbf{w}_i^n, \bar{\mathbf{w}}_i^n, i = 0, 1, 2, \dots$	PDE	interior, interface
2a	Eval $v_l, \sigma_l, \rho_l$ from FSR	projection	$v_l, \sigma_l, \rho_l$
2b	$v_0^n, \bar{v}_0^n = v_l, -\rho_0^n, \bar{\sigma}_0^n = \sigma_l, \rho_0^n = \rho_l$	projection	$\mathbf{w}_0^n, \bar{\mathbf{w}}_0^n$
2c	$\mathbf{w}_{-1}^n = \mathcal{E}_{+1}^{(3)} \mathbf{w}_0^n, \bar{\mathbf{w}}_{-1}^n = \bar{\mathcal{E}}_{+1}^{(3)} \bar{\mathbf{w}}_0^n,$	extrapolation	$\mathbf{w}_{-1}^n, \bar{\mathbf{w}}_{-1}^n$
2d	Eval : $\dot{v}_f = -\frac{1}{\rho} \partial_x p, \dots$	PDE	$\dot{v}_f, \dot{v}_s, \dot{\sigma}_f, \dot{\sigma}_s$
2e	Eval $\dot{v}_l, \dot{\sigma}_l$ from FSR	projection	$\dot{v}_l, \dot{\sigma}_l$
3a	$-\frac{1}{\rho} \partial_x p = \dot{v}_l, \rho a^2 \partial_x v = \dot{\sigma}_l$	compatibility	$\rho_{-1}^n, v_{-1}^n$
3b	$\bar{\rho} c_p^2 \partial_x \bar{v} = \dot{\sigma}_l, \frac{1}{\bar{\rho}} \partial_x \bar{\sigma} = \dot{v}_l$	compatibility	$\bar{v}_{-1}^n, \bar{\sigma}_{-1}^n$
3c	Correct grid and grid velocity	projection	$F^n, \dot{F}^n$

# The interface projection step

Using the linearized FSR solution, the interface values are an impedance weighted average of the provisional fluid and solid values:

$$v_I = \frac{\bar{z}\bar{v}_0 + zv_0}{\bar{z} + z} + \frac{\sigma_0 - \bar{\sigma}_0}{\bar{z} + z},$$
$$\sigma_I = \frac{\bar{z}^{-1}\bar{\sigma}_0 + z^{-1}\sigma_0}{\bar{z}^{-1} + z^{-1}} + \frac{v_0 - \bar{v}_0}{\bar{z}^{-1} + z^{-1}}$$

**Compare:** the *standard* FSI scheme uses the *heavy solid* limit,  $\bar{z} \gg z$ , *velocity-from-solid, stress-from-fluid*:

$$v_I = \bar{v}_0$$
$$\sigma_I = \sigma_0 = -p + p_e$$

The standard scheme is unstable for *light* solids.

Note: for hard problems with shocks hitting the interface, there are advantages to using the full nonlinear solution to the FSR problem.

# Elastic piston numerical results

Computed solution for a smoothly receding piston.

		Fluid						Solid					
grid	N	$\rho$	r	$v$	r	$p$	r	$\bar{u}$	r	$\bar{v}$	r	$\bar{\sigma}$	r
G <sub>1</sub>	20	2.2e-3		2.7e-3		1.6e-3		4.9e-4		9.3e-5		1.2e-4	
G <sub>2</sub>	40	5.4e-4	4.1	6.2e-4	4.3	3.7e-4	4.3	1.2e-4	4.2	2.6e-5	3.6	2.3e-5	5.0
G <sub>3</sub>	80	1.4e-4	3.9	1.5e-4	4.0	9.3e-5	4.0	2.9e-5	4.1	6.2e-6	4.1	5.0e-6	4.7
G <sub>4</sub>	160	3.5e-5	3.9	3.9e-5	3.9	2.3e-5	4.0	7.3e-6	4.0	1.5e-6	4.1	1.1e-6	4.4
rate		1.99		2.03		2.03		2.03		1.99		2.24	

Max-norm errors for **very light solid**:  $\bar{\rho} = 10^{-5}$ ,  $z/\bar{z} = 5.8 \times 10^3$ .

		Fluid						Solid					
grid	N	$\rho$	r	$v$	r	$p$	r	$\bar{u}$	r	$\bar{v}$	r	$\bar{\sigma}$	r
G <sub>1</sub>	20	9.2e-4		9.6e-4		8.5e-4		3.0e-4		1.3e-5		1.1e-5	
G <sub>2</sub>	40	2.1e-4	4.3	2.3e-4	4.2	2.0e-4	4.2	7.3e-5	4.2	3.0e-6	4.4	2.6e-6	4.2
G <sub>3</sub>	80	5.6e-5	3.8	5.6e-5	4.1	4.9e-5	4.1	1.8e-5	4.0	6.9e-7	4.4	6.3e-7	4.2
G <sub>4</sub>	160	1.5e-5	3.9	1.4e-5	4.0	1.2e-5	4.0	4.5e-6	4.0	1.6e-7	4.2	1.5e-7	4.2
rate		1.99		2.04		2.04		2.02		2.12		2.07	

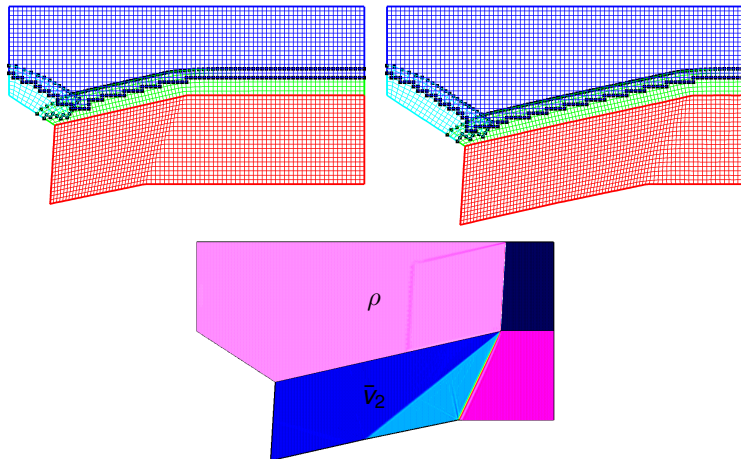
Max-norm errors for **very heavy solid**:  $\bar{\rho} = 10^5$ ,  $z/\bar{z} = 5.8 \times 10^{-7}$ .

- Max-norm errors are converging to second-order accuracy.
- Scheme is stable for large and small impedance ratios.





# Superseismic shock: grids and computed solution



- The red grid for the solid domain is shown adjusted for the displacement.

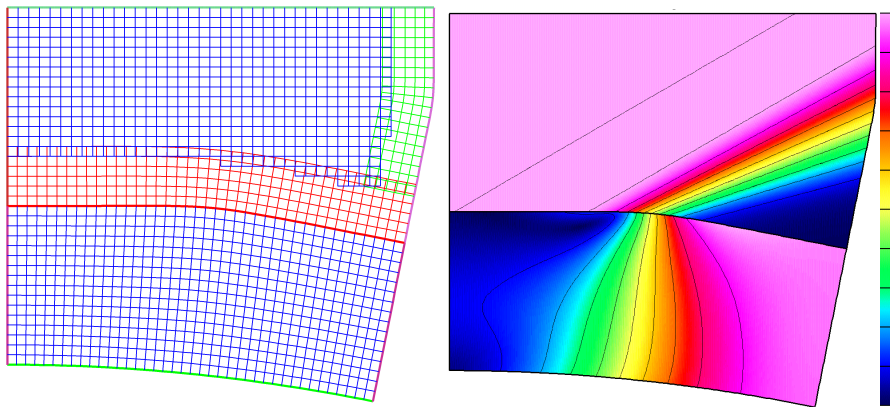


# $L_1$ -norm errors and convergence rates for the superseismic shock

Grid	Solid						Fluid					
	$ \bar{\mathbf{u}} $	$r$	$ \bar{\mathbf{v}} $	$r$	$ \bar{\boldsymbol{\sigma}} $	$r$	$\rho$	$r$	$ \mathbf{v} $	$r$	$T$	$r$
$\mathcal{G}_{SS}^{(4)}$	8.6e-5		2.8e-3		7.9e-3		4.6e-3		2.2e-2		1.1e-2	
$\mathcal{G}_{SS}^{(8)}$	3.6e-5	2.4	1.7e-3	1.6	5.0e-3	1.6	2.6e-3	1.8	1.3e-2	1.7	6.6e-3	1.7
$\mathcal{G}_{SS}^{(16)}$	1.5e-5	2.5	1.1e-3	1.6	3.1e-3	1.6	1.4e-3	1.9	6.7e-3	1.9	3.5e-3	1.9
$\mathcal{G}_{SS}^{(32)}$	5.6e-6	2.7	6.9e-4	1.6	2.0e-3	1.6	7.0e-4	2.0	3.4e-3	2.0	1.7e-3	2.0
rate	1.32		0.67		0.67		0.91		0.92		0.91	

- Due to the discontinuities, the  $L_1$ -norm errors do not converge at second-order.
- The solid variables  $\bar{\mathbf{v}}$  and  $\bar{\boldsymbol{\sigma}}$  converge at the expected rates of 2/3 (due to spreading of the linear discontinuities).
- In isolation the fluid domain should converge at first order.

# The deforming diffuser grid and solution



Contours of the fluid pressure,  $[\min, \max] = [.5, 1.]$  and norm of the solid stress tensor  $|\bar{\sigma}|$ ,  $[\min, \max] = [0, .054]$ .

# Max-norm errors and convergence rates for the deforming diffuser

Solid						
Grid	$ \bar{\mathbf{u}} $	$r$	$ \bar{\mathbf{v}} $	$r$	$ \bar{\sigma} $	$r$
$\mathcal{G}_{dd}^{(2)}$	2.3e-4		8.6e-4		4.0e-2	
$\mathcal{G}_{dd}^{(4)}$	5.8e-5	3.9	2.3e-4	3.8	1.2e-2	3.4
$\mathcal{G}_{dd}^{(8)}$	9.9e-6	5.9	4.6e-5	5.0	2.0e-3	5.8
$\mathcal{G}_{dd}^{(16)}$	1.6e-6	6.2	9.5e-6	4.8	3.4e-4	5.9
rate	2.40		2.18		2.31	

Fluid								
Grid	$\rho$	$r$	$v_1$	$r$	$v_2$	$r$	$T$	$r$
$\mathcal{G}_{dd}^{(2)}$	3.6e-2		1.7e-2		2.3e-2		1.2e-2	
$\mathcal{G}_{dd}^{(4)}$	8.8e-3	4.1	3.8e-3	4.5	7.1e-3	3.2	2.5e-3	4.7
$\mathcal{G}_{dd}^{(8)}$	2.1e-3	4.1	8.6e-4	4.4	2.3e-3	3.1	7.9e-4	3.2
$\mathcal{G}_{dd}^{(16)}$	5.0e-4	4.3	2.0e-4	4.3	5.2e-4	4.5	1.7e-4	4.8
rate	2.05		2.15		1.80		2.02	

- max-norm errors at  $t = 1$  with the Godunov slope-limiter off.
- solutions are converging to second-order in the max norm.

# Summary

We have been developing efficient algorithms for modeling

- 1 incompressible flows and rigid body motion,
- 2 compressible flows and rigid/deforming bodies.

The approach is based on

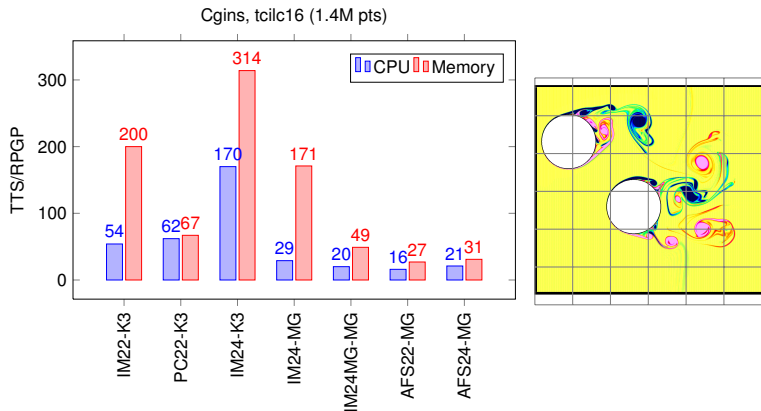
- 1 overlapping grids for flexible representation of geometry,
- 2 high-order accurate algorithms,
- 3 accurate treatment of interfaces and boundary conditions.

# Backup Slides

# Half-plane problem analysis

# Parallel overlapping grid generation

# AFS results



Performance of Cgins for different time stepping methods for flow past two cylinders in a channel. The CPU is a normalized time-to-solution (TTS). The memory usage is measured in reals-per-grid-point. K3 stands for the Krylov solver BiCGStab-ILU(3).



# Fluid Structure Interactions

Our approach to fluid structure interactions is based on

- Moving and deforming overlapping grids.
  - Adaptive mesh refinement.
  - Accurate interface approximations (e.g. incorporation of the fluid-solid Riemann problem to handle added-mass instabilities).
- 
- WDH., DWS, *Moving Overlapping Grids with Adaptive Mesh Refinement for High-Speed Reactive and Nonreactive Flow*, J. Comp. Phys. **216** (2005).
  - WDH., K. K. Chand, *A Composite Grid Solver for Conjugate Heat Transfer in Fluid-Structure Systems*, J. Comput. Phys, 2009.
  - J. W. Banks and B. Sjögreen, *A Normal Mode Stability Analysis of Numerical Interface Conditions for Fluid/Structure Interaction*, Commun. Comput. Phys., 2011.
  - J. W. Banks, WDH, D.W. Schwendeman, *Deforming Composite Grids for Solving Fluid Structure Problems*, J. Comput. Phys, 2012.
  - J. W. Banks, WDH, B. Sjögreen, *A stable FSI algorithm for light rigid bodies in compressible flow*, LLNL-JRNL-558232, submitted.

# Euler equations on a mapped moving grid

Todo – solid mechanics equations

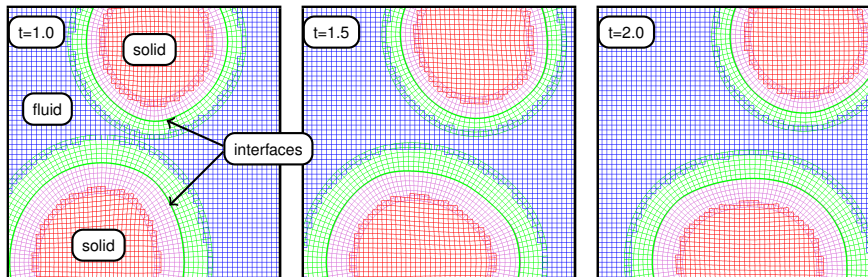
For the Euler equations, consider a mapping for a (possibly moving) component grid given by  $\mathbf{x} = \mathbf{G}(\mathbf{r}, t)$ . Following the work in [?], the mapped equations are

$$\partial_t \mathbf{w} + \frac{1}{J} \nabla_{\mathbf{r}} \cdot \mathbf{F} + \frac{\mathbf{w}}{J} \nabla_{\mathbf{r}} \cdot \mathbf{V} = 0, \quad (5)$$

where  $J = |\mathbf{x}_{\mathbf{r}}|$  is the Jacobian of the mapping,  $\mathbf{F}$  is the mapped flux tensor, and  $\mathbf{V}$  is related to the grid velocity  $\dot{\mathbf{G}}$ . In particular, the vector components of  $\mathbf{F}$  and the scalar components of  $\mathbf{V}$  are given by

$$\mathbf{F}_i = \left( J \sum_{j=1}^{n_d} \frac{\partial r_i}{\partial x_j} \mathbf{f}_j \right) - V_i \mathbf{w}, \quad V_i = J \sum_{j=1}^{n_d} \frac{\partial r_i}{\partial x_j} \dot{\mathbf{G}}_j, \quad i = 1, \dots, n_d. \quad (6)$$

# Deforming Composite Grids for FSI



Composite grids at different times from the FSI-DCG simulation a shock hitting two solid elastic cylinders. The green fluid grids deform over time to follow the fluid-solid interfaces. The blue background Cartesian grid for the fluid remains fixed.

# Summary

- the deforming composite grid (DCG) approach is being developed to model fluid-structure interactions (FSI) for compressible fluids and elastic solids.
- the solution of the fluid-solid Riemann problem can be used to define stable interface approximations for both *light* and *heavy* solids.
- our FSI-DCG approximation was verified to be second-order accurate in the max-norm on problems with smooth solutions.

## Future work:

- support for adaptive mesh refinement (AMR) at the interface.
- more general solid mechanics models.
- incompressible fluids.
- extend to three space dimensions.