

## Adaptive Mesh Refinement Strategies for Immersed Boundary Methods (invited)

Elias Balaras & Marcos Vanella

Fischell Department Bioengineering University of Maryland, College Park MD 20742, USA

Supported by NSF, AFOSR and ONR



- Introduction and Motivation
- Adaptive mesh refinement methodology
- Embedded boundary method and fluid-structure interaction strategy.
- Examples
- Summary and Conclusions



Non-boundary conforming methods are popular in recent years, especially for fluid-structure interaction problems (multiple bodies, large displacements/deformations)

Immersed boundary approaches are usually tied to Cartesian grids that do not allow flexibility in grid refinement.

In general are limited to low Reynolds numbers (special cases of high Re applications have also been reported)

Some form of local refinement is required to extend this class of methods to practical applications

### Adaptive mesh refinement: topology











Divide the domain in sub-blocks. Each sub-grid block has a structured Cartesian topology, and is part of a tree data structure that covers the entire computational domain.

Local refinement of a sub-grid block is performed by bisection in each coordinate direction.

Number of nodes in each sub-block remains constant.



## Adaptive mesh refinement: overview



• We use a **projection method**, where advective and diffusive terms are advanced explicitly

• We use the **Paramesh toolkit** (developed by MacNeice and Olson) for the implementation of the AMR process. The package creates and maintains the hierarchy of sub-grid blocks, with each block containing a fixed number of grid points.

• A **single-block Cartesian** grid solver is employed in each sub-grid block:

 standard staggered grid in each subblock



second-order central finite-differences

• A hybrid **direct/multigrid** solver is used for the **Poisson** equation (developed in collaboration with the **FLASH-Group**)

• **Guard cells** are used to discretize equations *standard staggered grid* at the interior coarse-fine interfaces

coarse-fine interface





#### Step 1: coarse guard-cell filling



#### Step 2: coarse guard-cell filling

coarse-fine interface

Guard-cells must be filled at block edges in order to complete the differencing stencil

A sequence of 1D quadratic interpolations is used to fill the guard-cells





Multigrid is a natural choice considering the hierarchy of sub-grid blocks covering the computational domain.

The algorithm uses full multigrid cycles, in which all relaxation sweeps extend across the full computational domain.



• A relaxation sweep is performed over the full domain composed of sub-grid blocks at three different refinement levels (2 blocks at level 2, 6 blocks at level 3, and 8 blocks at level 4).

• When restricting to the next coarser refinement level, sub-grid blocks at the finest level undergo the restriction operation (the eight blocks at level 4 are restricted to two blocks at level 3, and the next relaxation sweep is performed)

• . . . . . . .



The multigrid algorithms have an inherent scaling limitation:

- as the grid gets coarser, there is less computational load to distribute among processors.
- as the number of blocks at a level approaches the number of processors, we begin to see the overhead cost of low computation/communication ratio.
- further reduction in the number of blocks, processors start to become idle and load balance deteriorates (at the coarsest level, very few processors are busy).

In the hybrid scheme we have developed:

- we do not complete a V cycle, and instead coarsening of the grid is stopped at a predetermined level.
- the coarse level may be any level that is fully refined (i.e. containing blocks that completely cover the computational domain).
- The solution at this level is computed using one of the parallel direct solvers

### Adaptive mesh refinement: Poisson Solver





Performance of Multigrid and Hybrid Poisson solvers on BG/P (A. Dubey)





 $= F(u_{f1}^{*}) + F(u_{f2}^{*}) - \Delta t F((\nabla_{i} \delta p)_{f1}) - \Delta t F((\nabla_{i} \delta p)_{f2}) = F(u_{f1}^{n+1}) + F(u_{f2}^{n+1})$ 

## Adaptive mesh refinement: accuracy



### Validation: Taylor Green Vortex

- Compare numerical solution to analytical solution of 2D Navier-Stokes equations
- Domain: [π/2, 5π/2]x [π/2, 5π/2]
- Homogeneous Dirichlet/Neumann velocity boundary conditions and Neumann pressure boundary condition







### Validation: Taylor Green Vortex





### Vortex Ring impinging on a wall, Re ≈ 570



Positions in the X=0 plane, for centers of X vorticity:

Q contour for vortex impinging normal to a wall,  $Re \approx 570$ 



### Vortex Ring impinging on a wall, Re ≈ 570

- Compare AMR solution to numerical solution using a Single Block, Cartesian solver.
- Velocity Dirichlet BCs in top and Bottom Boundaries, periodic on side walls. Pressure Neumann BCs.



vorticity isolines at a cross section,  $Re \approx 570$ 



#### Positions in the X=0 plane, for centers of X vorticity:



A 'direct-forcing' embedded-boundary method is used to imposed boundary conditions on arbitrary boundaries\*



\* Balaras Comp. & Fluids 2004, Balaras & Yang J. Comput. Phys. 2006, Vanella & Balaras, Submitted J. Comput. Phys. 2008



•Fluid-Structure Interaction: Strong coupling Scheme (Yang, Preidikman & Balaras, J. Fluids & Structures, 2007.)

$$\begin{split} p \mathbf{y}^{j} &= \mathbf{y}^{j-4} + \frac{4}{3} \Delta t \ \left( 2 \ \mathbf{F}^{j-1} - \mathbf{F}^{j-2} + 2 \ \mathbf{F}^{j-3} \right) \\ \mathbf{y}^{j} &= \ ^{k+1} \mathbf{y}^{j} - \mathbf{e}^{j} \\ \hline \\ \mathbf{y}^{j} &= \ ^{k+1} \mathbf{y}^{j} - \mathbf{e}^{j} \\ \hline \\ \mathbf{z}^{k} &= \ ^{k-1} \mathbf{z}^{k} \\ \frac{\partial \phi^{k}}{\partial x_{i} \partial x_{i}} &= \ ^{k} H S_{i}^{k} + f_{i}^{k} \\ \frac{\partial \phi^{k}}{\partial x_{i} \partial x_{i}} &= \ ^{k} \frac{\partial \phi^{k}}{\partial x_{i}}, \\ u_{i}^{k} &= \ ^{k} \frac{\partial \phi^{k}}{\partial x_{i}}, \\ p^{k} &= \ p^{k-1} + \phi^{k} \\ \hline \\ \hline \\ \mathbf{z}^{k+1} \mathbf{y}^{j} &= \ ^{k} \frac{1}{8} \left[ 9 \ \mathbf{y}^{j-1} - \mathbf{y}^{j-3} + 3 \Delta t \ \left( \ ^{k} \mathbf{F}^{j} + 2 \ \mathbf{F}^{j-1} - \mathbf{F}^{j-2} \right) \right] \end{split}$$



• Structure: First order coupled nonlinear ordinary differential equations

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & [\mathbf{M}(\mathbf{q}_1)] \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_2 \\ \mathbf{F}(\mathbf{q}_1, \mathbf{q}_2, t) \end{bmatrix}$$

For the case of slender rigid bodies in planar motion.

$$\begin{bmatrix} [\mathbf{I}]_{3x3} & [\mathbf{0}]_{3x3} & \\ \hline \mathbf{m} & \overline{\mathbf{0}} & \overline{\mathbf{0}} \\ [\mathbf{0}]_{3x3} & \mathbf{0} & \mathbf{m} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & I_{cm} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \\ \dot{\mathbf{q}}_2 \\ \end{bmatrix} = \begin{bmatrix} \mathbf{q}_2 \\ f_x(\mathbf{q}_1, \mathbf{q}_2) \\ M_{cm}(\mathbf{q}_1, \mathbf{q}_2) - \mathbf{m}^* g \\ M_{cm}(\mathbf{q}_1, \mathbf{q}_2) \end{bmatrix}$$

$$y(t)$$
Where  $\mathbf{q}_1 = [x(t) \ y(t) \ \theta(t)]^*$ .

### Falling plates: parameters



Fluid Structure interaction of Two falling plates

Two runs:

• The FSI algorithm with AMR every 5 time steps. 8x10 cell blocks and maximum of 7 levels of refinement, 59.000 points.

 Same FSI and IB strategies in a single Cartesian mesh and a direct Block Tridiagonal Poisson solver.
 1.250.000 points.

Grid size around bodies ≈ 0.008c in both cases.

#### **Properties:**

Chord length c = 1, a thickness of 10% c, m = 0.5, lo = 0.0416667, 3 DOF per Body, g = 0.12, v = 0.005. In both runs  $\Delta t = 1.22e-3$ , and 16000 timesteps.



## Falling plates: results



### Falling plates: results





Positions in the x-y plane of centers of mass of both bodies taking time as a parameter.





#### Computational setup and Eulerian/Largangian grid arangement









Dry restitution coefficient zero. Comparison with the experiment by Eames & Dalziel Dust, J. Fluid Mech. 403, 2000.





Dry restitution coefficient one

# Hovering Fly



### Hovering Fly at Re = 300:



- Re =  $U_{tmax} * L_r / v = 300$ .
- Boundary Conditions: Dirichlet in z, periodic in x, y.
- Level 0:  $32 \times 32 \times 32$  grid cells. Blocks  $16^3$  cells. Coarse calculation maximum 3 levels of refinement ( $\Delta x = 0.019$ ).
- Integration for 4 flapping cycles. Harmonic kinematics for beating angle and geometric angle of attack.
  - IB Treatment of the wings as membranes.





# Hovering Fly







• Proposed approach combines the computational efficiency of embedded boundary Cartesian solvers with the resolution capabilities of the boundary conforming approaches

- The accuracy of the single-block solver is unaffected
- Dramatic reduction in grid points can be achieved compared to single-block solvers

• Grid discontinuities introduce significant challenges to eddy resolving techniques (i.e. large-eddy simulations)