Adaptive Mesh Refinement with Boundary Non-conforming Grids

Xun Huang and Xin Zhang

Aeronautics and Astronautics
School of Engineering Sciences
University of Southampton
Southampton, UK
Outline

- Introduction
- AMR
- Immersed boundary method
- Parallel implementation
- Summary
Background

- To improve efficiency
- To provide a potential means of modelling various, co-existing physics
Adaptive mesh refinement (AMR)

- Represent computational domain as hierarchal refinement levels
- Increase grid resolution only in areas of interest
Examples of block-based AMR computations

- Three refinement levels
- Refinement ratio of two
- 8 to 12 cells in every block
CAA solutions

For spatial derivatives:

• 4\(^{th}\)-order prefactored compact scheme or 4\(^{th}\)-order Dispersion-Relation-Preserving (DRP) scheme

• Artificial selective damping and/or explicit filter

For time integration:

• 4\(^{th}\)-order 4/6 stage explicit Runge-Kutta scheme (for simplicity the same time steps applied to all refinement levels)
A previous case with a thin flat boundary (AIAA-2005-2873)
A previous case with a thin flat boundary (AIAA-2005-2873)
**Efficiency (AIAA-2005-2873)**

Cells number:
- Coarsest level contains 2304 cells;
- Finest level contains 36864 cells.

Computing time: (for previous case, 1.3G PIII 512MB)
- 2191 seconds on finest mesh;
- 830 seconds by AMR.

AMR does increase efficiency and without sacrificing exactness
The cost of AMR on single CPU
**Immersed boundary method**

- Finite volume
- Applying a force
- Degenerate to stair step mesh
- Cut-cell Finite Difference
Two solutions

- Stair-step mesh: backward stencil, and only set pressure in ghost points.
- Cut-cell method: Set pressure, velocity, and density in ghost points, and use backward stencil.
Compare stair-step and cut-cell methods in 1D case
Scattering from cylinder with 0th-interpolation (Stair step mesh)
Compare exactness
Another case of IMM+AMR: Duct Radiation with thick boundary

Special zone for computing differentials in x direction

Thick=0.55
Length=1.65
Contour of pressure:
m=4, n=1, k=10, time=0~27.335, CFL=0.35
Parallel AMR: Dynamic load balancing

Space filling curve → Parallel depth-first tree traversals
The cost of AMR on parallel machine

Up to 8 processors
Details of code

- Tree data structures used for 2D AMR
- Codes written in Fortran90 and MPI
- Around 30 files for AMR, 10 files for CAA, 20,000+ lines code
- CAA Applications are developed above AMR & CAA Lib
Summary

• The parallel AMR code has been developed. The code is still need to be tuned to gain higher efficiency.
• Several methods of immersed boundary are tested.
• IMM plus AMR can be applied as an engineering tool. Body-fitted AMR is more suitable for academic purpose.
• Future work:
  (1) Make the code more efficient and general;
  (2) Run cases governed by Euler and/or NS equations.
Appendix 1: AMR Operations (1)

Regridding

if not need finer patches, delete connection

Generate finer area and superimpose on base mesh

Need refine

Restrict & Prolong

Restrict solution to update base mesh

Prolong to initialize new patches
Appendix 2: AMR Operations (2)

Ghost area construction
(AIAA 2005-2873)
Appendix 3: Parallel restriction, prolong and ghost construction

These parallel operations lead to a search in a parallel tree.
Appendix 4: Parallel AMR

- Algorithm
  - Cell-based AMR
  - Block-based AMR

- Software
  - OpenMP
  - Global Arrays toolkit
  - Extended Java & C
  - MPI

- Hardware
  - Shared-memory machines
  - Distributed-memory machines