Unsteady CFD Simulations of Flow-Control Valves by an Unstructured Overset-Grid Method

Ph.D. dissertation presentation

Jingxin Liu

Advisory Committee:
Dr. Akin Ecer (co-chair)
Dr. Steve Frankel (co-chair)
Dr. Hasan U. Akay
Dr. Jayathi Y. Murthy
Dr. Gregory A. Blaisdell

School of Mechanical Engineering
Purdue University

November 06, 2008
Unsteady CFD Simulations of Flow-Control Valves by an Unstructured Overset-Grid Method

Outline:

• Research Motivation and Approach
• Methodology
• CFD Solver Equations and Boundary Conditions
• Testing Cases of Internal/External Flows:
  1. A Supersonic Ramp with oblique shock;
  2. A Convergent-Divergent Duct with normal shock;
  3. Quasi-3D NACA0012 Airfoil with pitching oscillations
• Complex Valve Flow Cases Applied:
  1. A Butterfly Valve ranging from fully-opened status to nearly-closed position
  2. A Mono-Cylinder IC Engine with a fixed central valve and piston movement
• Conclusions
Motivation

- Danhan and Solliec have studied flow-control valves experimentally and stated the complexity of such flows (*ASME Journal of Fluid Engineering, Vol. 122, pp.337-344, 2000*)

- Silvester has developed a simple analytical model for predicting the aerodynamic torque of flow-control valves (*Proceedings Institute of Mechanical Engineering, Vol. 196, pp. 229-237, 1982*)

- Leutwyler and Dalton used the commercial CFD code Fluent to predict steady aerodynamic torque, lift and drag forces on a 2-D butterfly valve model (*ASME Journal of Fluid Engineering, Vol. 128, pp. 1074-1082, 2006*)

- The flow fields of compressible fluid in a flow-control valve with movements and unsteady flow is quite complex and can be rather challenging for any CFD algorithm (*R. Morita, ASME Journal of Fluid Engineering, Vol. 129, pp. 48-54, 2007*)
Three leading techniques for treatment of grids surrounding moving bodies exist in the literature:

- **Deforming grids**: Arbitrary Lagrangian-Euler (ALE) formulation are used. These guarantee conservation and a smooth variation of grid size. They need re-meshing and smoothing techniques.

- **Adaptive Cartesian Grids**: These do not require re-meshing or mesh movement (ALE is not used). They need to detect the location of boundaries with respect to the mesh to refine/coarsen the mesh as the bodies move.

- **Overset Grids**: Do not require any re-meshing, allow for independent grids for components and the body. ALE techniques are required for moving the bodies.
Approach: Dynamic Unstructured Overset Grids

- While structured overset grid methods have been well developed, unstructured overset grid algorithms have not been as popular.

- In conventional CFD unstructured overset grid methods,
  - Establishing inter-grid communication between sub-grids is not trivial,
  - Usually manual hole cutting and inefficient searching algorithms are used,
  - Considerable modifications flow solvers are required.

- Developed unstructured overset grid method provides,
  - A systematic and automated hole-cutting process,
  - An efficient searching algorithm,
  - One-cell overlapping width reducing the computational effort,
  - Ease in integration to the CFD solver with very few changes
  - Ease in modeling complex geometries and movements. Once an unstructured mesh around a moving body (minor grid) is generated, it is coupled with a background grid for the entire flow domain for any specified motion,
  - Capability in handling the extreme condition of wall-to-wall contact case, in which the wall boundary of moving body is touching a fixed wall.
Methodology: General Definition of an Overset Grid

Each body has its own attached grid independently generated. Due to the arbitrary position of the different bodies/grids, some nodes can be located in the solid regions, namely non-flow/non-computational domains. These nodes have to be removed from the flow calculations.
Methodology: Developed Hole-Cutting Process

Step 1. Hole Cutting for Minor Grid
1.1 Check if boundary of background grid overlap with that of minor grid by using a recursive searching algorithm by building a data tree of bounding box, (if not, go to Step 2).

Valve in 30-degree position
(no intersection between boundaries)

Valve in 60-degree position
(Intersection occurs between boundaries)
Methodology: Developed Hole-Cutting Process (cont’d)

Step 1. Hole Cutting for Minor Grid
1.2 Remove inactive cells which have inactive vertexes and form new boundaries by an improved vector intersection algorithm.

Minor grid before hole cutting
Minor grid after hole cutting
Methodology: Developed Hole-Cutting Process (cont’d)

Step 2. Hole Cutting for background grid
2.1 Remove inactive cells which have inactive vertexes and form new boundaries by an improved vector intersection algorithm.
Hole Cutting Demo: Flow-Control Amplifier with a Moving Disc

Moving Grid Animation
Step 3: Donor cell searching and interpolation stencil building: Get donor cell from background grid for fringe boundary points of minor grid.

3.1 Build Alternate-Digital Tree (ADT) for all background grid cells,
3.2 Build recursive searching algorithm,
3.3 Get donor cell and interpolation coefficients using volume rate linear interpolation.
Methodology: Searching and Interpolation Algorithms Employed (cont’d)

Illustration of the ADT Searching Algorithm

Search the tree (A, B, C, D, E, F, G, H):
1. Check if $a^i \leq x_A^i \leq b^i$ for $i = 1, 2$.
2. Since $d_A^i \geq a^i$ and $c_A^i \leq b^i$, search the tree (B, C, D, E):
   2.1. Check if $a^i \leq x_B^i \leq b^i$.
   2.2. Since $d_B^i \geq a^i$ and $c_B^i \leq b^i$, search the tree (C, E):
      2.2.1. Check if $a^i \leq x_C^i \leq b^i$.
      2.2.2. Skip (left link is zero).
      2.2.3. Skip (because $c_C^i > b^i$)
   2.3. Skip (because $c_E^i > b^i$)
3. Skip (because $c_A^i > b^i$)
Methodology: Searching and Interpolation Algorithms Employed (cont’d)

Step 4: Get donor cell from minor grid for fringe boundary points of background grid

4.1 Build Alternate-Digital Tree (ADT) for all background grid cells,
4.2 Build recursive searching algorithm,
4.3 Get donor cell and interpolation coefficients by using volume rate linear interpolation.

Once a donor cell is identified, the solution parameters’ value \( q \) at the nodes of the donor cell \( e_A \) to the target point \( B \) is linearly interpolated as follows:

\[
q_B = \sum_{i} q_{iA} \cdot [N_{iA}]_B \quad \text{with} \quad \sum_{i} N_{iA} = 1
\]

Here, \( N_{iA} \) is the interpolation coefficient, which is obtained from the ratio of each newly constructed cell’s volume to the donor cell’s volume.

\[
\text{Sum of Volume} = \text{Cell}_{\text{Pt}2.4.1} + \text{Cell}_{\text{Pt}3.2.4} \\
+ \text{Cell}_{\text{Pt}3.4.1} + \text{Cell}_{\text{Pt}2.3.1} = \text{Cell}_{1.2.3.4}
\]
Integration to the CFD Solver: SunFlo3D

The CFD solver, SunFlo3D, is based on an external flow code, (USM3D) which was developed at NASA, a cell-centered, finite volume Euler and Navier-Stokes flow solver for viscous/inviscid, compressible flows (Frink, 1990).

Modifications made to SunFlo3D for this research:

1. Internal flow boundary conditions based on Riemann invariants were implemented,
2. Overset gridding-algorithm system was developed,
3. A boundary type was added for data interpolations between overset grids,
4. Arbitrary Lagrangian-Euler (ALE) formulation was implemented

Grid Generator: All the meshes used in this research were generated by using VGRIDns, developed by S. Pirzadeh (AIAA Journal, Vol. 32, No. 8, pp.1735-1737, August 1994)
CFD Solver: Flow Equations

- **Flow** equations (Arbitrary Lagrangian-Euler (ALE) formulation)

\[
\frac{\partial (\bar{Q}V)}{\partial t} + \oint_{\partial \Omega} (\bar{F}_i - \bar{F}_\nu) \cdot \hat{n} \, dS = 0
\]

\[
\bar{Q} = \frac{\int \bar{q} \, dV}{V}
\]

\[
\bar{F}_i = \begin{bmatrix}
\rho(u-W_x) \\
\rho u(u-W_x) + p \\
\rho u(v-W_y) + p \\
\rho (u-W_x) \\
\rho u(u-W_x) \\
\rho u(v-W_y) + p \\
(E+p)(u-W_x) + W_x p
\end{bmatrix}
\]

\[
\bar{F}_\nu = \begin{bmatrix}
\rho \nu_1 - \frac{\partial \rho}{\partial t} \\
\rho \nu_2 - \frac{\partial \rho}{\partial t} \\
\rho \nu_3 - \frac{\partial \rho}{\partial t} \\
\rho v - \frac{\partial \rho}{\partial t} \\
\rho v - \frac{\partial \rho}{\partial t} \\
\rho v - \frac{\partial \rho}{\partial t} \\
\rho v - \frac{\partial \rho}{\partial t}
\end{bmatrix}
\]

\[
\bar{F}_\nu = 0
\]

\[
\bar{F}_\nu = \begin{bmatrix}
\tau_{xx} \\
\tau_{yx} \\
\tau_{zx} \\
\tau_{yy} \\
\tau_{zy} \\
\tau_{zz}
\end{bmatrix}
\]

\[
\bar{F}_\nu = \begin{bmatrix}
u_{1x} + \nu_{2y} + \nu_{3z} - \rho \frac{\partial v}{\partial x} \\

\nu_{2x} + \nu_{1y} + \nu_{3z} - \rho \frac{\partial v}{\partial y} \\

\nu_{3x} + \nu_{2y} + \nu_{1z} - \rho \frac{\partial v}{\partial z} \\

\nu_{1y} + \nu_{2y} + \nu_{3z} - \rho \frac{\partial v}{\partial y} \\

\nu_{2z} + \nu_{1z} + \nu_{3z} - \rho \frac{\partial v}{\partial z} \\

\nu_{3z} + \nu_{2z} + \nu_{1z} - \rho \frac{\partial v}{\partial z}
\end{bmatrix}
\]

\[
\bar{F}_\nu = \begin{bmatrix}
\rho \nu_1 - \frac{\partial \rho}{\partial t} \\
\rho \nu_2 - \frac{\partial \rho}{\partial t} \\
\rho \nu_3 - \frac{\partial \rho}{\partial t} \\
\rho v - \frac{\partial \rho}{\partial t} \\
\rho v - \frac{\partial \rho}{\partial t} \\
\rho v - \frac{\partial \rho}{\partial t}
\end{bmatrix}
\]

\[
\bar{F}_\nu = \begin{bmatrix}
\tau_{xx} \\
\tau_{yx} \\
\tau_{zx} \\
\tau_{yy} \\
\tau_{zy} \\
\tau_{zz}
\end{bmatrix}
\]

\[
\bar{F}_\nu = \begin{bmatrix}
u_{1x} + \nu_{2y} + \nu_{3z} - \rho \frac{\partial v}{\partial x} \\

\nu_{2x} + \nu_{1y} + \nu_{3z} - \rho \frac{\partial v}{\partial y} \\

\nu_{3x} + \nu_{2y} + \nu_{1z} - \rho \frac{\partial v}{\partial z} \\

\nu_{1y} + \nu_{2y} + \nu_{3z} - \rho \frac{\partial v}{\partial y} \\

\nu_{2z} + \nu_{1z} + \nu_{3z} - \rho \frac{\partial v}{\partial z} \\

\nu_{3z} + \nu_{2z} + \nu_{1z} - \rho \frac{\partial v}{\partial z}
\end{bmatrix}
\]
CFD Solver: Boundary Conditions

Moving boundary:

For moving boundaries, wall boundary conditions are modified by taking into account the mesh movements,

\[ V_{outer\_B.C.} = V_{fluid} - \hat{n} \cdot [(V_{fluid} - W) \cdot \hat{n}] \]

where, \( W \) is the grid moving velocity.

For no-slip wall boundary condition, it becomes, \( V_{wall} = W \).

The wall pressure is calculated from the normal momentum equation as,

\[ \frac{\partial p}{\partial n} = -\rho n \cdot a_w \]

where, \( a_w \) is the acceleration of the body wall.
CFD Solver: Turbulence Model

- Spalart-Allmaras (S-A)

The Reynolds stress tensor \( \langle \rho u_i u_j \rangle \) is related to the mean strain rate through an apparent turbulent viscosity called eddy viscosity:

\[
- \rho u_i u_j = \mu_t \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right), \quad \mu_t = \rho \bar{\nu} f_{v1}
\]

where

\[
f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\bar{\nu}}{\nu}
\]

The PDE, which is solved separately from the flow equations, is in the following form:

\[
\frac{\partial \bar{\nu}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\nu} u_j) = C_{b1} [1 - f_{i2}] \bar{S} \bar{\nu} + \frac{1}{\sigma} \left\{ \bar{\nu} \cdot ((\bar{v} + \bar{\nu}) \nabla \bar{\nu}) + C_{b2} |\nabla \nu|^2 \right\} -
\]

\[
\left[ C_{w1} f_w - \frac{C_{b1}}{k^2} f_{i2} \right] \left( \frac{\bar{\nu}}{d} \right)^2 + f_{i1} \nabla U^2
\]
Numerical Study Cases

The developed overset unstructured grid method has been validated for the following test cases:

**Test Cases for Internal/External Flows:**
1. A Supersonic Ramp with oblique shock
2. A Convergent-Divergent Duct with normal shock
3. Quasi-3D NACA0012 Airfoil with pitching oscillations

**Test Cases for Complex Valve Flows:**
4. A Butterfly Valve ranging from fully-opened status to nearly-closed position
5. A Mono-Cylinder IC Engine with piston movement

For cases 1 and 4 conservation studies, as well as mesh refinement & accuracy studies are performed. For all of the above cases, simulations results are validated with either single grid solutions or experimental data.
Case Studies: Supersonic 15° Ramp

Mach_inlet = 2.5
Case Studies:  Supersonic 15° Ramp

Conservation Study of Overset Grid:
Case Studies: Supersonic 15° Ramp

Conservation Study of Overset Grid:

Numerical scheme is locally conservative:

**Background Grid:**\[ \sum_{\Delta V \in B} \frac{\partial Q_B}{\partial t} \Delta V = - \sum_{f \in \Gamma_B} F_f \Delta S_f \]

**Minor Grid:**\[ \sum_{\Delta V \in M} \frac{\partial Q_M}{\partial t} \Delta V = - \sum_{f \in \Gamma_M} F_f \Delta S_f \]

**Necessary and sufficient condition for global conservation:**\[ \sum_{\Delta V \in O} \frac{\partial Q_o}{\partial t} \Delta V = - \sum_{f \in \Gamma_o} F_f \Delta S_f \]
Case Studies: Supersonic 15° Ramp

Conservation Study of Overset Grid:

Non-conservative Interpolation for Overset Grid:

\[ \sum_{\Delta V \in O} \frac{\partial Q}{\partial t} \Delta V \neq - \sum_{f \in \Gamma_o} F_f \Delta S_f \]

Overlapping Zone:

Differences between LHS and RHS should decrease with mesh refinement

For one-cell overlapping zone, as the mesh size is getting smaller, the left-hand side of the above equation is approaching to zero with the control volume. On the right-hand side, the flux summation across all interpolation boundary faces was numerically calculated and shown to be approaching to zero. The convergence of the solution scheme is demonstrated for the supersonic ramp case.
Case Studies: Supersonic 15° Ramp

Conservation Study of Overset Grid:

**Summary of Non-Conservative Interpolation for Overset Grid:**

Tang and Zhou [102] presented mathematical proof and numerical examples on this subject and concluded that,

1) the conservation error of a numerical solution caused by a non-conservative interpolation has an upper bound when the solution itself is bounded;

2) the conservation error due to a non-conservative interpolation decreases as the mesh size gets small. Consequently, inaccuracy for shock jumps and locations caused by such a treatment reduces.

Therefore, under these conditions, non-conservative interface schemes are applicable to practical calculations, even if discontinuities are present near grid interfaces.

Case Studies: Supersonic 15° Ramp

<table>
<thead>
<tr>
<th></th>
<th>Single grid</th>
<th>Overset grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>total number of cells (coarse mesh)</td>
<td>48,778</td>
<td>54,026</td>
</tr>
<tr>
<td>total number of cells (fine mesh)</td>
<td>634,992</td>
<td>660,487</td>
</tr>
</tbody>
</table>
Case Studies: Supersonic 15° Ramp

Single grid solution of pressure contour for coarse mesh

Overset grid solution of pressure contour for coarse mesh

Single grid solution of pressure contour for fine mesh

Overset grid solution of pressure contour for fine mesh
Case Studies: Supersonic 15° Ramp

Variation of residual on coarse grid (600 time steps)

Variation of residual on fine grid (600 time steps)
Case Studies: Supersonic $15^\circ$ Ramp

Variation of flux summations across all interpolation boundaries on coarse grid (600 iteration steps)

Variation of flux summations across all interpolation boundaries on fine grid (600 iteration steps)
Geometric configurations for a convergent-divergent duct
Case Studies: A Convergent-Divergent Duct

single-grid solution

overset-grid solution
Case Studies: A Convergent-Divergent Duct

Pressure values on **top wall** at middle-span plane

Pressure values on **bottom wall** at middle-span plane
Case Studies: Unsteady Flow around a NACA 0012 Airfoil

Comparison of Grids with Different Levels of Overlapping

Multi-cell overlapping width

One-cell overlapping width
Case Studies: Unsteady Flow around a NACA 0012 Airfoil

Comparison of Grids with Different Levels of Overlapping

Multi-cell overlapping width  One-cell overlapping width
Case Studies: Unsteady Flow around a NACA 0012 Airfoil

Comparison of a Single Grid with an Overlapping Grid

One-cell overlapping width

Single grid solution
Case Studies: Unsteady Flow around a Pitching Airfoil (NACA 0012)

Unsteady Flow Results:
Angle of attack varies as
\[ \alpha(t) = \alpha_m + \alpha_p \sin(\omega \alpha t) \]
\[ M_\infty = 0.8 \]

Sinusoidal pitching solution.

Lift coefficient varying with pitching angles
Case Studies: 3D, Compressible Flow around a Butterfly Valve

Experimental setup of Morris and Dutton
(Morris, Ph.D. Dissertation, University of Illinois, Urbana, IL. 1987.)

Butterfly valve testing section
(rectangular pipe tunnel
height/depth: 2.67
Section of the arc valve geometry)
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Viscous Grids:
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Sections of the Valve Inviscid Grids at Different Angles:

0 degree

5 degree

10 degree

15 degree

30 degree

60 degree
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Viscous Grids:

Before Hole Cutting

After Hole Cutting
Specified Boundary Conditions, (for all cases, total temperature, T0 = 300K):

<table>
<thead>
<tr>
<th></th>
<th>Boundary conditions for Circular valve disk model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha = 0^\circ$</td>
</tr>
<tr>
<td>$P_b / P_0$ (Given)</td>
<td>0.75</td>
</tr>
<tr>
<td>Inlet Mach number (Calculated)</td>
<td>0.50</td>
</tr>
</tbody>
</table>
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Flow results and experimental validation for fully-opened valve position (0 degree).
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Flow results and experimental validation for 5-degree open valve position.
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Flow results and experiment validation of 10-degree open valve position.
Flow results and experiment validation of 30-degree open valve position.
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Flow results and experiment validation of 60-degree open valve position.
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Mesh Refinement and Accuracy Studies:

- Coarse mesh
- Medium-size mesh
- Fine mesh
### Mesh Refinement and Accuracy Studies:

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Number of cells</th>
<th>Number of nodes</th>
<th>Smallest cell size (Radius of valve is 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Background</td>
<td>Minor</td>
<td>Background</td>
</tr>
<tr>
<td>Coarse</td>
<td>1,345,667</td>
<td>123,615</td>
<td>234,896</td>
</tr>
<tr>
<td>Medium</td>
<td>2,251,848</td>
<td>372,868</td>
<td>390,192</td>
</tr>
<tr>
<td>Fine</td>
<td>3,331,264</td>
<td>1,242,076</td>
<td>578,122</td>
</tr>
</tbody>
</table>
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Mesh Refinement and Accuracy Studies:

Mach contours at 0-degree valve-angle position (coarse mesh)

Mach contours at 0-degree valve-angle position (Fine mesh)
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Mesh Refinement and Accuracy Studies:

Mach contours at 30-degree valve-angle position (coarse mesh)

Mach contours at 30-degree valve-angle position (Fine mesh)
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Mesh Refinement and Accuracy Studies:

Mach contours at 60-degree valve-angle position (coarse mesh)

Mach contours at 60-degree valve-angle position (Fine mesh)
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Mesh Refinement and Accuracy Studies:

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Maximum relative error of pressure $\times 100%$</th>
<th>Maximum relative error of inlet Mach $\times 100%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>2%</td>
<td>0.01%</td>
</tr>
<tr>
<td>Medium</td>
<td>2%</td>
<td>0.01%</td>
</tr>
<tr>
<td>Fine</td>
<td>1%</td>
<td>0.01%</td>
</tr>
</tbody>
</table>
Mesh Refinement and Accuracy Studies:

Maximum relative error between numerical results and experiment data for valve at 30-degree angle position

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Maximum relative error of pressure ( \frac{P - P_{\text{experiment}}}{P_{\text{experiment}}} \times 100% )</th>
<th>Maximum relative error of inlet Mach ( \frac{M - M_{\text{experiment}}}{M_{\text{experiment}}} \times 100% )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>4%</td>
<td>0.02%</td>
</tr>
<tr>
<td>Medium</td>
<td>2%</td>
<td>0.02%</td>
</tr>
<tr>
<td>Fine</td>
<td>2%</td>
<td>0.02%</td>
</tr>
</tbody>
</table>
Case Studies: 3D, Compressible Flow around a Butterfly Valve (cont’d)

Mesh Refinement and Accuracy Studies:

Maximum relative error between numerical results and experiment data for valve at 60-degree angle position

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Maximum relative error of pressure ( \frac{P - P_{\text{experiment}}}{P_{\text{experiment}}} \times 100% )</th>
<th>Maximum relative error of inlet Mach ( \frac{M - M_{\text{experiment}}}{M_{\text{experiment}}} \times 100% )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>5%</td>
<td>0.03%</td>
</tr>
<tr>
<td>Medium</td>
<td>4%</td>
<td>0.02%</td>
</tr>
<tr>
<td>Fine</td>
<td>4%</td>
<td>0.02%</td>
</tr>
</tbody>
</table>
Unsteady Flow Case Study for an IC Engine with Piston Movement

(experiments by A. P. Morse et al, Report FS/78/24, Imperial College, Department of Mechanical Engineering 1978).

Fig. 1. Axisymmetric piston-cylinder assembly of Morse et al.
Unsteady Flow Case Study for an IC Engine with Piston Movement

Schematic for overset grid wall-to-wall contact of the cylinder-piston IC engine assembly

before hole cutting

after hole cutting

Only One Cell Overlap Between Background Grid and Minor Grid
Unsteady Flow Case Study for an IC Engine with Piston Movement

Overset grids for crank angle of 0 degree (Top Dead Center)

Overset grids for crank angle of 90 degrees

Overset grids for crank angle of 180 degrees (Bottom Dead Center)
Unsteady Flow Case Study for an IC Engine with Piston Movement

Piston movement: mesh animation
Unsteady Flow Case Study for an IC Engine with Piston Movement

piston motion:

\[ Z(t) = s / 2 + s / 2 \times (1 - \cos(\omega t)) \]

\[ \omega = 200 \text{rpm} \]

\[ \text{Re} = \frac{\omega \times s \times r_r}{\nu} = 2000 \]

Fig. 1. Axisymmetric piston-cylinder assembly of Morse et al.
Calculations were carried out in the following sequence:

(I) Position and velocity of the piston were determined from the following given equations:

\[ X_{pl}(t) = X_0 + \frac{S}{2} \left( 1 - \cos(\omega t) \right) \quad \text{Eq. 7.3} \]

\[ V_{pl}(t) = \frac{dx_{pl}(t)}{dt} = \frac{S}{2} \sin(\omega t) \quad \text{Eq. 7.4} \]

Here, \( X_0 \) is the top dead center (TDC) position, \( S \) is the engine stroke length, \( \omega \) is the revolution speed of engine crank shaft.

(II) By using equations 7.3 and 7.4, positions of the new grid lines and local grid velocities were obtained.

(III) Call overset grid subroutine to build information data transformation stencil between background- and minor-grid.

(IV) According to the instantaneous mass flow rate, time-dependent boundary conditions at intakes orifice were determined.

(V) Until the solution was converged, solving governing equations procedure was repeated, during which interpolation boundary conditions keep updating for iterations.
Unsteady Flow Case Study for an IC Engine with Piston Movement

Solution at 36 Crank Angle

Calculated velocity-vector contours and streamline

Measured streamline contours [Morse et al., 1978]
Unsteady Flow Case Study for an IC Engine with Piston Movement

Solution at 90 Crank Angle

Calculated velocity-vector contours

Measured streamline contours [Morse et al., 1978]
Unsteady Flow Case Study for an IC Engine with Piston Movement

Solution at 270 Crank Angle

Calculated velocity-vector contours

Measured streamline contours

[Morse et al., 1978]
Unsteady Flow Case Study for an IC Engine with Piston Movement

Axial velocities radial profiles at the crank angle 36 degrees after top dead center (TDC),

with profiles axially spaced 10 mm apart starting from the cylinder head

with profiles axially spaced 30 mm apart starting from the cylinder head

[Graphs showing velocity ratios with numerical solution and experimental data]
Unsteady Flow Case Study for an IC Engine with Piston Movement

Axial velocities radial profiles at the crank angle 144 degrees after top dead center (TDC)

with profiles axially spaced 10 mm apart starting from the cylinder head

with profiles axially spaced 30 mm apart starting from the cylinder head
Unsteady Flow Case Study for an IC Engine with Piston Movement

Overset-grid solutions of flow-jet velocities

Crank angle 36 degrees

Crank angle 90 degrees

Crank angle 144 degrees
Unsteady Flow Case Study for an IC Engine with Piston Movement

Piston movement: unsteady solution animation
Summary of the Research

1. An unstructured grid method has been developed with features of simplicity and automation of hole cutting and efficient searching process.

2. This method guarantees the one-cell overlapping grid width leading to less computational time and less numerical errors;

3. The developed overset unstructured grid method has been applied to cases such as:
   - a supersonic ramp with oblique shock capture,
   - a converging-diverging duct with normal shock capture,
   - a unsteady oscillating airfoil,
   - internal inviscid/viscous flow computations of a 3D circular-arc-shape butterfly valve at different fixed positions, and
   - IC engine assembly with piston movement.

   For these cases, simulations results are validated with single grid solutions or experimental data with satisfactory accuracy.
Highlights of the Developed Approach of the Dynamic Unstructured Overset Grid Method

Developed method provides:

- A systematic and automated hole-cutting process
- An efficient searching algorithm
- Ease in integration to the CFD solver with very few changes
- Ease in modeling complex geometries and movements. Once an unstructured mesh around a moving body (minor grid) is generated, it is coupled with a background grid for the entire flow domain for any specified motion
- Capable handling extreme condition for wall-wall touch case, in which the wall boundary of moving body is touching the wall of fixed one
- One-cell overlapping grid capability reduces computational time while maintaining accuracy
Acknowledgements

Advisory Committee:
Dr. Akin Ecer (co-chair)
Dr. Steve H. Frankel (co-chair)
Dr. Hasan U. Akay
Dr. Gregory A. Blaisdell
Dr. Jayathi Y. Murthy

CFD Lab Colleagues:
Dr. Erdal Yilmaz
Mr. Resat U. Payli
Ms. Cai Shen
Thank you all very much!