Verification and Application of the Unified Coordinate System in Preliminary Design

Nathan Woods   Ryan Starkey

Busemann Advanced Concepts Laboratory
Department of Aerospace Engineering Sciences
University of Colorado at Boulder

20th AIAA Computational Fluid Dynamics Conference

Outline

The Project Cycle

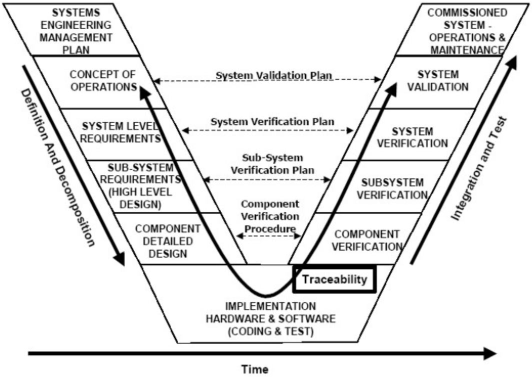


Figure: The life cycle of a project. (Wikimedia Commons)

Wind Tunnels and CFD

Wouldn't it be great if CFD actually worked like a wind tunnel?

- Quick tests for design shapes
- No grid generation

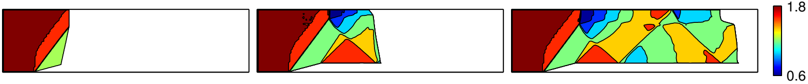


Figure: Mach contours for a transonic duct flow. See also http://www.youtube.com/watch?v=g-a9oGpKiEw

The Virtual Wind Tunnel

- Lagrangian Fluid dynamics
  - Natural analogue to wind tunnel testing
  - Fails due to severe grid distortion
- Unified Coordinates
  - Largely preserves benefits of Lagrangian systems
  - Distortion can be controlled

Original Development of UCS

Developed by W. H. Hui[1] from from Lagrangian coordinates

- The time-dependent Euler equations[2][3]
- Extension to viscous flows[4]
- External flows and oscillating airfoils[5],[6]

Extensions of UCS

Applications of UCS to other systems

- Reactive flows[7]
- Multimaterial flows[8]
- Plasma dynamics[9]
- Gas-kinetic (BGK) aerodynamics[10]

Outline

The Unified Coordinate Transformation

$$\begin{pmatrix} dt \\ dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ U & A & L & P \\ V & B & M & Q \\ W & C & N & R \end{pmatrix} \begin{pmatrix} d\lambda \\ d\xi \\ d\eta \\ d\zeta \end{pmatrix} \tag{1}$$

Conservation Equations

$$\frac{\partial \mathbf{E}}{\partial \lambda} + \frac{\partial \mathbf{F}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} = 0 \tag{2}$$

$$\mathbf{E} = \begin{pmatrix} \rho\Delta \\ \rho\Delta u \\ \rho\Delta v \\ \rho\Delta e \\ A \\ B \\ L \\ M \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \rho(1-h) I \\ \rho(1-h) Iu + pM \\ \rho(1-h) Iv - pL \\ \rho(1-h) Ie + pI \\ -hu \\ -hv \\ 0 \\ 0 \end{pmatrix}, \mathbf{G} = \begin{pmatrix} \rho(1-h) J \\ \rho(1-h) Ju - pB \\ \rho(1-h) Jv + pA \\ \rho(1-h) Je + pJ \\ 0 \\ 0 \\ -hu \\ -hv \end{pmatrix} \tag{3}$$

$$\Delta = AM - BL,\ I = uM - vL,\ J = Av - Bu \tag{4}$$

The UCS Algorithm I

The algorithm is given as follows. For each time step:

1. Compute optimal time step[4]. Predict the maximum coordinate advancement of the first column of nodes. If necessary, limit the time step such that the first column ends the time step no more than $\Delta\xi$ from the upstream boundary, to prevent nonuniformity in the node spacing.
2. Apply Strang splitting, as above:
    1. Identify the appropriate time-advancement and the active interfaces for the given Strang step. Active interfaces are left/right for the ξ steps of Strang splitting, and top/bottom for the η step.
    2. Step through nodes. For each node $n_{i,j}$:
        1. Identify adjoining states using neighboring nodes, boundary conditions, and MUSCL interpolation, as appropriate.

The UCS Algorithm II

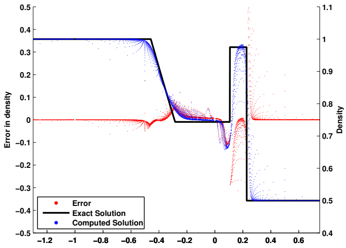
2. Solve the local, one-dimensional Riemann problem to obtain the values of flow variables at the interface between adjoining states.
3. Use interfacial flow values and the value of h that corresponds to the cell being updated to compute updated cell metric coefficients A, B, L, and M. Store all updated node values separately until after all nodes have been computed.
4. Compute physical flux into the cell using the values of flow variables at the interfaces and the updated geometric variables corresponding to the cell being updated.
5. Use flux to compute updated flow values.

3. Update all nodes.

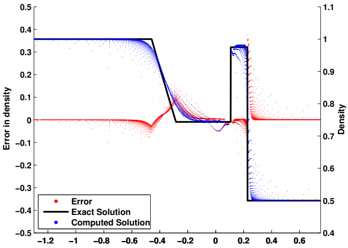
3. Solve for h and update coordinate positions using trapezoidal integration.
4. Add/remove columns of nodes as needed.

Outline

The Riemann Problem



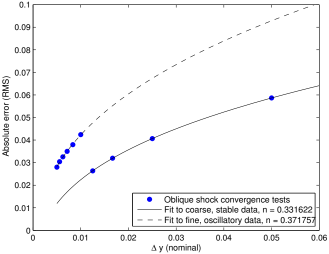
(a) $h = 0$



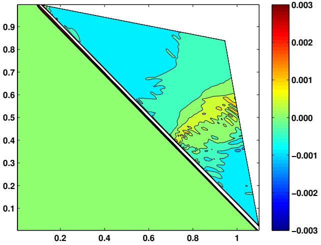
(b) $h = 0.999$

Figure: The similarity solution of the Riemann problem and the corresponding error in the numerical solution, computed throughout the simulation region.

The Oblique Shock



(a) Root-mean-squared error for the oblique shock problem. The appearance of grid instabilities leads to two distinct error curves with different rates of convergence *n*.



(b) A plot of normalized error in pressure, highlighting the oscillations which propogate downstream from the oblique shock.

Figure: The oblique shock wave

The Expansion Corner

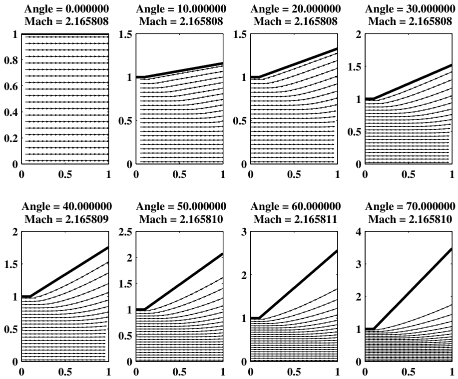


Figure: Computed streamlines for Prandtl-Meyer expansion at increasing expansion angles. Angles are given in degrees.

The Diamond Shock Train

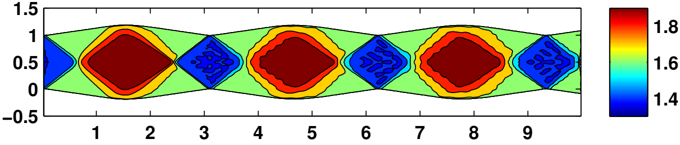


Figure: Computed Mach number for an under-expanded nozzle flow, showing the diamond-shock train. Note the presence of oscillations which grow as the flow progresses downstream.

The Isentropic Nozzle

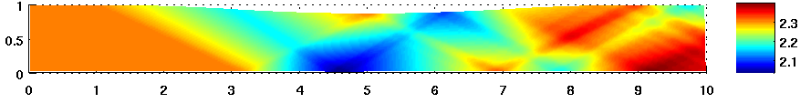
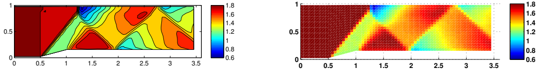
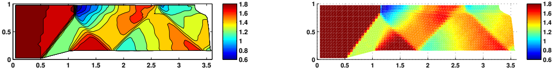


Figure: Computed Mach number for isentropic flow through a constricting, parabolic, supersonic channel

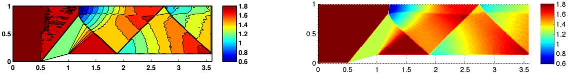
The Transonic Duct



(a) $h = 0$, nominal grid size: 72×20



(b) $h = 0.25$, nominal grid size: 72×20



(c) $h = 0$, nominal grid size: 360×100

Figure: Qualitative accuracy comparison between UCS and Eulerian simulations for a transonic duct flow. Notice the improved resolution of the slip line and the walls for the UCS solution.

Outline

Summary

- The major advantage of UCS is the automatic mesh
- UCS must be implemented in a way that improves stability and accuracy

- Future Work
  - Optimize for speed
  - Boundary layer solver
  - Shock-aligned grid

References and Further Reading I

- Hui, W. H., "The Unified Coordinate System in Computational Fluid Dynamics," *Communications in Computational Physics*, Vol. 2, No. 4, August 2007, pp. 577–610.
- Hui, W. H., Li, P. Y., and Li, Z. W., "A Unified Coordinate System for Solving the Two-Dimensional Euler Equations," *Journal of Computational Physics*, Vol. 153, 1999, pp. 596–673.
- Hui, W. H. and Kudriakov, S., "A Unified Coordinate System for Solving the Three-Dimensional Euler Equations," *Journal of Computational Physics*, Vol. 172, January 2001, pp. 235–260.

References and Further Reading II

- Hui, W., Wu, Z. N., and Gao, B., “Preliminary Extension of the Unified Coordinate System Approach to Computation of Viscous Flows,” *Journal of Scientific Computing*, Vol. 30, No. 2, February 2007, pp. 301–344.
- Hui, W. H., “A unified coordinates approach to computational fluid dynamics,” *Journal of Computational and Applied Mathematics*, Vol. 163, 2004, pp. 15–28.
- Hui, W., Hu, J., and Zhao, G., “Gridless Computation Using the Unified Coordinates,” *Computational Fluid Dynamics 2004*, edited by C. Groth and D. W. Zingg, Springer Berlin Heidelberg, 2006, pp. 503–508.

References and Further Reading III

- Azarenok, B. N. and Tang, T., “Second-order Godunov-type scheme for reactive flow calculations on moving meshes,” *Journal of Computational Physics*, Vol. 206, No. 1, 2005, pp. 48 – 80.
- Jia, P., Jiang, S., and Zhao, G., “Two-dimensional compressible multimaterial flow calculations in a unified coordinate system,” *Computers and Fluids*, Vol. 35, 2006, pp. 168–188.
- Zhilkin, A., “A Dynamic Mesh Adaptation Method for Magnetohydrodynamics Problems,” *Computational Mathematics and Mathematical Physics*, Vol. 47, No. 11, 2007, pp. 1819–1832.

References and Further Reading IV

Jin, C. and Xu, K., “A unified moving grid gas-kinetic method in Eulerian space for viscous flow computation,” *Journal of Computational Physics*, Vol. 2007, 2007, pp. 155–175.