Towards Automated Mesh Adaptation Using Simplex Cut Cells

Krzysztof J. Fidkowski

Aerospace Computational Design Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

November 2, 2007

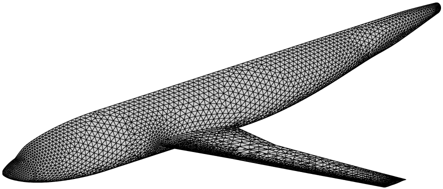
Outline

Outline

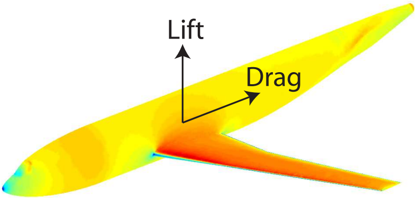
Computational Fluid Dynamics (CFD)

CFD in aerospace engineering

- Actively used in design and analysis
- Supplements/replaces expensive wind-tunnel tests
- Reduces design cycle time and allows for innovative designs

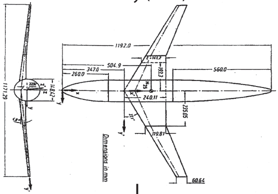


Geometry

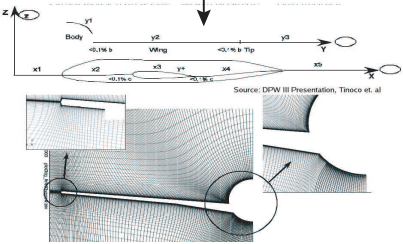


Flow solution

Geometry (CAD)



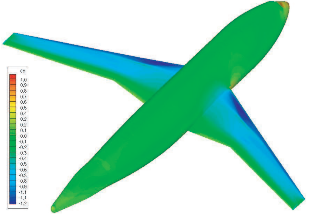
Mesh Generation (days - weeks)



Source: DPW III Presentation, Tinoco et. al

Flow Solution (hours)

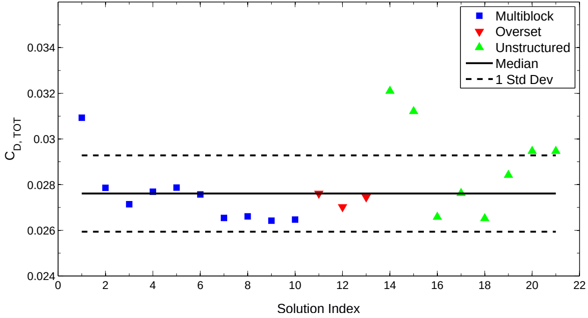
Error Estimation and
Mesh Adaptation



- Not representative of all CFD methods (e.g. Cartesian or boundary-potential methods)
- Typical use of finite volume in industry

AIAA Drag Prediction Workshop III – 2006

- Wing-body geometry, $M = 0.75, C_L = 0.5, Re = 5 \times 10^6$
- Run on today's state-of-the-art CFD codes



Differences in:

- Physical models
- Discretization
- **Mesh size distribution**

1 drag count ($.0001C_D$) $\approx$ 4-8 passengers (Boeing 747-400)

Identification of Problem

Current CFD Practices:

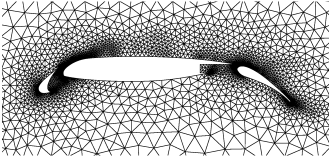
- Risk of unacceptably large errors is high
- Heavy “person-in-the-loop” involvement is required, especially during mesh generation
- Difficult to apply solution-based adaptation and optimization

Key Problems:

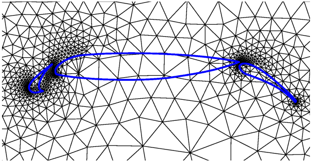
- Insufficient robustness
- Insufficient automation

Key Ideas

1. *Simplex* cut-cell meshing for *high-order* solutions

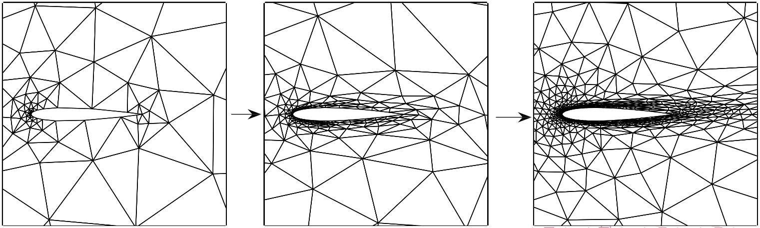


Boundary-conforming mesh

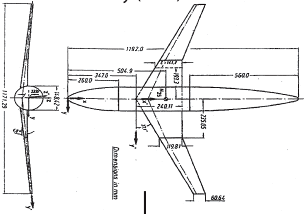


Simplex cut-cell mesh

2. Output-based *anisotropic* mesh adaptation for *high-order* solutions

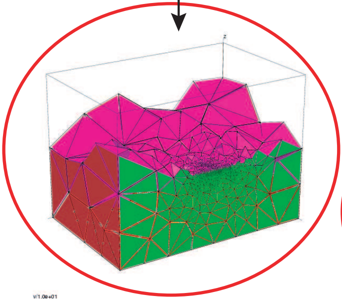


Geometry (CAD)



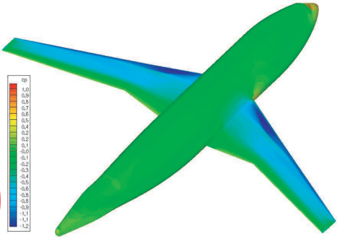
- Mesh generation and error estimation performed automatically
- Each cycle is an *adaptation iteration*

Mesh Generation (minutes)



Flow Solution (hours)

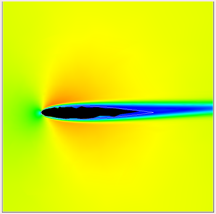
Error Estimation and Mesh Adaptation



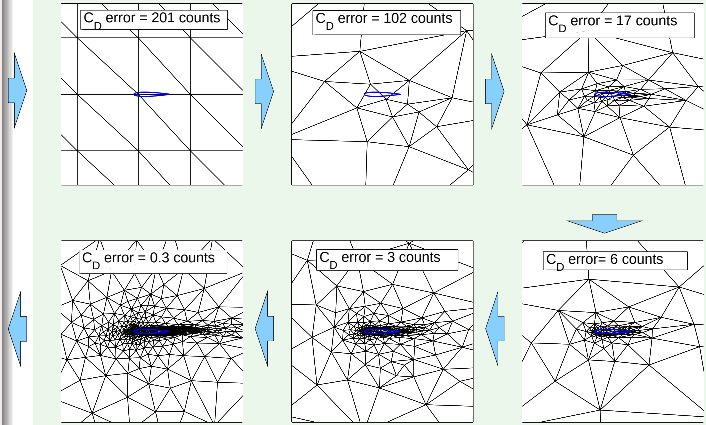
Example: Adaptation + Cut Cells

User

- $M_\infty = 0.5$, $Re = 5000$
- C_D to within 1 count



Automated



Outline

Compressible Navier-Stokes Equations

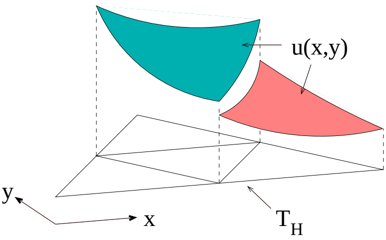
$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \underbrace{\mathcal{F}_i(\mathbf{u})}_{\text{Inviscid Flux}} - \nabla \cdot \underbrace{\mathcal{F}_v(\mathbf{u}, \nabla \mathbf{u})}_{\text{Viscous Flux}} = 0$$

- 5 conservative variables: $\mathbf{u} = [\rho, \rho u, \rho v, \rho w, \rho E]$
- 5 equations (conservation laws)
- Fluxes are nonlinear functions of $\mathbf{u}$
- Interested primarily in steady-state ($\partial \mathbf{u}/\partial t = 0$)

Discontinuous Galerkin Discretization

High-order finite-element method:

- Solution/test space: $\mathcal{V}_H = [V_H^p]^5$, $V_H^p = \{v \in L^2(\Omega) : v|_\kappa \in P^p(\kappa) : \forall\kappa \in T_H\}$
- Roe inviscid flux; 2nd form of Bassi and Rebay for elliptic term
- Discrete semi-linear form: $\mathcal{R}_H(\mathbf{u}_H, \mathbf{v}_H) = 0, \quad \forall \mathbf{v}_H \in \mathcal{V}_H$



Solution

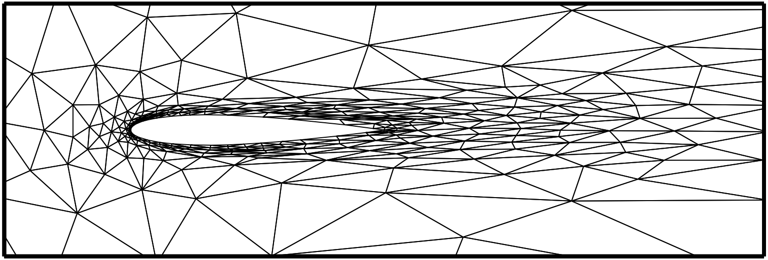
- Newton GMRES
- Store full linearization
- Initial approximate time stepping

Motivating features:

- High-order accuracy
- Element-wise compact stencil
- Ease of implementation

Outline

Output-Based Adaptation



$C_D = 565.7$ counts

- How accurate is this value?
- Where is more resolution necessary to improve the accuracy?
- How should that resolution be added?

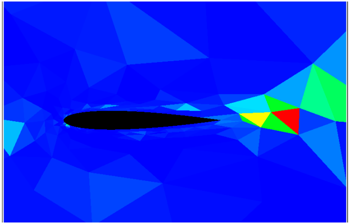
Output-Based Adaptation (ctd.)

Implementation

1. **Output error estimation and localization**
   - $\mathcal{J}(\mathbf{u})$ = output of interest (lift, drag, etc.)
   - $\mathbf{u}_H \in \mathcal{V}_H$ = approximate solution
   - $\mathcal{J}(\mathbf{u}_H) - \mathcal{J}(\mathbf{u})$ = output error
   - Solve for *adjoint*, ψ to estimate and localize the output error
2. **Automated anisotropic h-adaptation**
   - Anisotropy detection via extension of Hessian analysis to $p > 1$
   - Goal-oriented mesh optimization
   - Re-meshing at every adaptation iteration

Output Error Estimation: Local Error Indicator

Extensive previous work:
Pierce+Giles+Suli (2000),
Becker+Rannacher (2001),
Hartmann+Houston (2002),
Barth+Larson (2002)
Minor implementation differences



Error indicator for viscous case

$$\mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_H) \approx \underbrace{\mathcal{R}_H(\mathbf{u}_H, \psi - \psi_H)}_{\text{Primal Residual}} \approx \underbrace{\mathcal{R}_H^{\psi}(\mathbf{u}_H; \mathbf{u} - \mathbf{u}_H, \psi_H)}_{\text{Adjoint Residual}}$$

$\mathbf{u} - \mathbf{u}_H$ and $\psi - \psi_H$ estimated via reconstruction on enriched space.

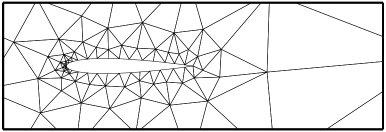
Elemental Error Indicator:

$$\epsilon_\kappa = \frac{1}{2}\left(\left|\mathcal{R}_h(\mathbf{u}_H, (\psi_h - \psi_H)|_\kappa)\right| + \left|\mathcal{R}_h^{\psi}(\mathbf{u}_H; (\mathbf{u}_h - \mathbf{u}_H)|_\kappa, \psi_H)\right| \right)$$

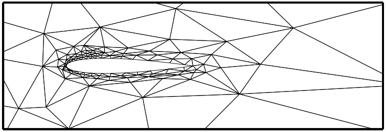
Anisotropic Adaptation

Idea: refine elements with high error; coarsen elements with low error

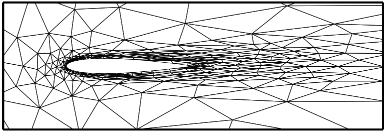
Iteration 0



Iteration 2



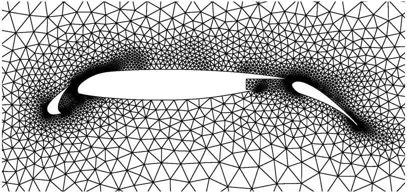
Iteration 4



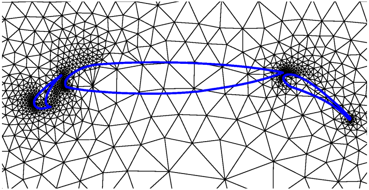
- Use *a priori* output error estimate to relate element error to size request: $\epsilon_\kappa \sim h_\kappa^r$
- Detect anisotropy by measuring $p+1$st order derivatives of a scalar quantity (Mach number)
- Optimize mesh size to meet requested tolerance and to satisfy error equidistribution
- Meshing: BAMG in 2D, TetGen in 3D
- *Left:* NACA 0012, $M = 0.5$, $Re = 5000$, $p = 2$ adapted on drag

Outline

What Are Cut Cells?



Boundary-conforming mesh



Simplex cut-cell mesh

Features

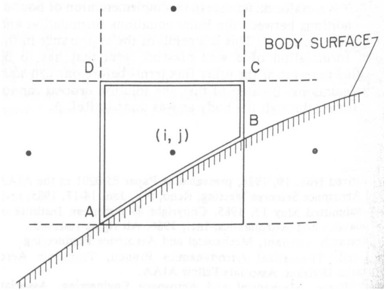
- Cut-cell meshes do not conform to geometry boundary
- Solution only exists inside the computational domain
- Premise: metric-driven meshing of a simple convex volume (e.g. box) is straightforward

History: First Cut Cells

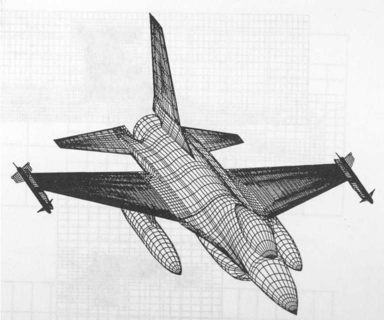
- 1979 – Purvis and Burkhalter: FV for 2D Full Potential Equations
- 1986 – Clarke, Salas, and Hassan: FV for 2D Euler
- 1987 – Gaffney, Salas, and Hassan: FV for 3D Euler

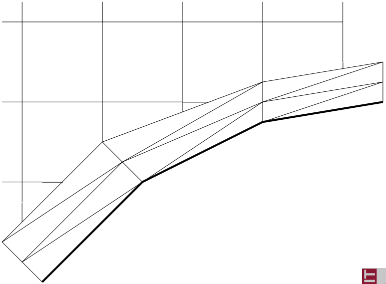
Features

- Linear cut cells
- Agglomeration to remove small cells
- Uniform grids



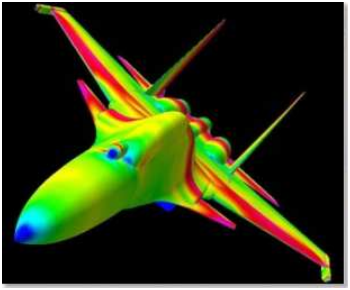
- 1986 – Boeing's TRANAIR: FEM for 3D Full Potential Equations; adaptation on geometry, user input, and solution; integration via Stoke's theorem. Still in use today.
- 1995 – Karman's SPLITFLOW (Lockheed): 3D RANS; required prismatic boundary layer mesh; outer flow via Cartesian cut cells.


TRANAIR

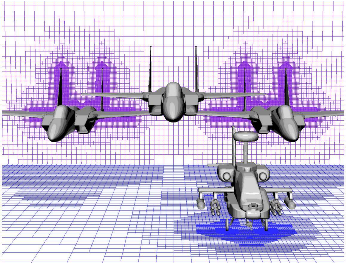

SPLITFLOW

History: Recent Work

- 1991 to present – MGAERO by Analytical Methods, Inc: finite difference for 3D Euler; multigrid, uniform grids.
- 1993+ – Application of adaptive refinement to Cartesian method for Euler; DeZeeuw, Powell, Coirier.
- 1999 to present – Cart3D: Mike Aftosmis *et al* , NASA; finite volume for 3D Euler; adaptively refined grids.



MGAERO



Cart3d

Why Simplex Cut Cells?

Objective: A robust, automated mesher and efficient meshes

Cartesian cut-cell method

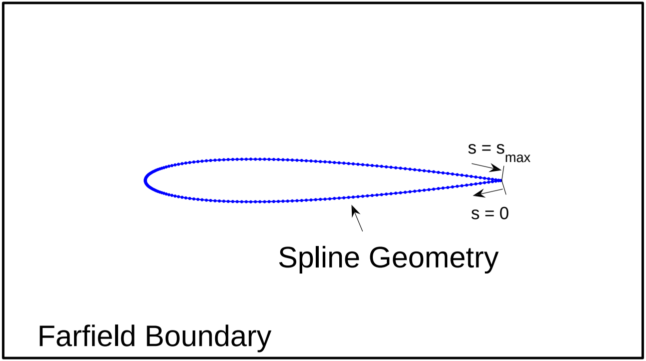
- Robust and automated grid generation
- Inability to adapt anisotropically

Simplex (triangles, tetrahedra) cut-cell method

- Robust and automated grid generation
- Ability to adapt anisotropically in any direction
- Not as lean as Cartesian method

Cubic splines

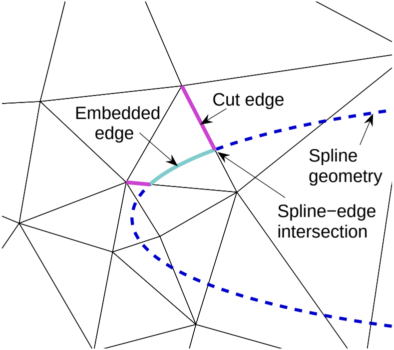
- Efficient treatment of curved boundaries
- Slope and curvature continuity at spline knots

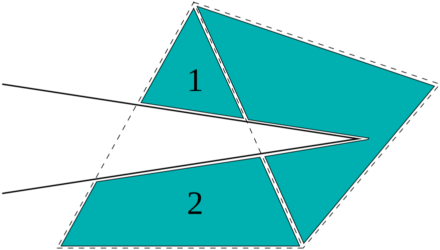


Intersection Problem

Implementation

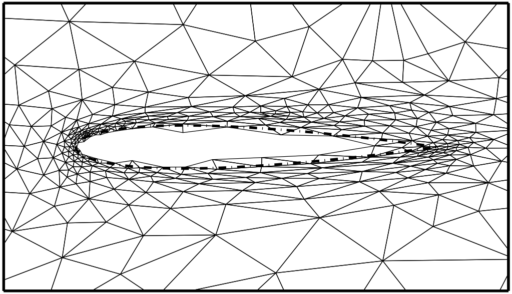
- Analytic intersections between splines and edges: cubic equation
- Multiply-cut triangles treated as a separate cut cells

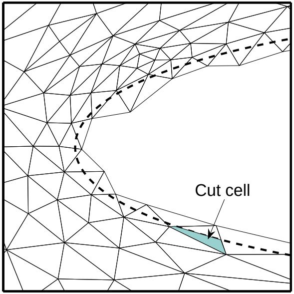




Intersection Problem (ctd.)

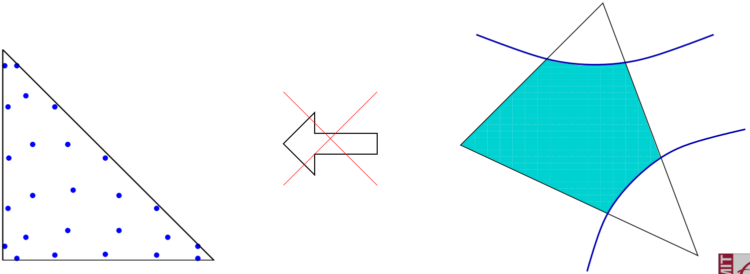
- Triangles completely contained inside geometry removed from mesh structure
- Integration rules on cut cells/edges calculated in preprocessing





Integration

- High-order finite element method requires integration over:
  - **Element boundaries** (edges in 2D, faces in 3D)
  - **Element interiors** (areas in 2D, volumes in 3D)
- Regular triangles and tetrahedra can be mapped to reference elements, where optimal integration rules exist
- These rules do not (in general) apply to cut cells, where areas and volumes are of irregular shape

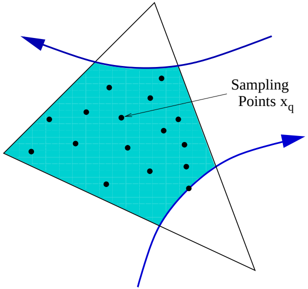


Area Integration

Goal

Sampling points, $\mathbf{x}_q$, and weights, w_q for integrating arbitrary $f(\mathbf{x})$ to a desired order:

$$\int_{\kappa} f(\mathbf{x})d\mathbf{x} \approx \sum_q w_q f(\mathbf{x}_q)$$



Key Idea

Project $f(\mathbf{x})$ onto space of high-order basis functions, $\zeta_{\mathbf{i}}(\mathbf{x})$:

$$f(\mathbf{x}) \approx \sum_{\mathbf{i}} F_{\mathbf{i}} \zeta_{\mathbf{i}}(\mathbf{x})$$

Choose $\zeta_{\mathbf{i}}(\mathbf{x})$ to allow for simple computation of $\int_{\kappa} \zeta_{\mathbf{i}}(\mathbf{x})d\mathbf{x}$.

Area Integration (ctd.)

Set $\zeta_{\mathbf{i}} \equiv \nabla \cdot \mathbf{G}_{\mathbf{i}}$ and use the divergence theorem:

$$\int_\kappa \zeta_{\mathbf{i}} d\mathbf{x} = \int_\kappa \nabla \cdot \mathbf{G}_{\mathbf{i}} d\mathbf{x} = \int_{\partial\kappa} \mathbf{G}_{\mathbf{i}} \cdot \mathbf{n} ds$$

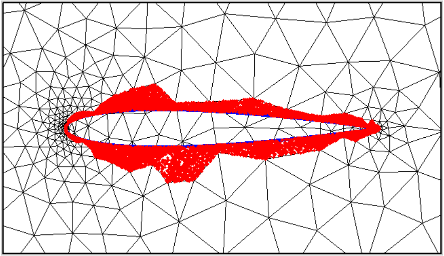
- $\mathbf{G}_{\mathbf{i}}$ = a standard high-order basis (e.g. tensor product)
- Line integrals over $\partial\kappa$ using 1D edge formulas

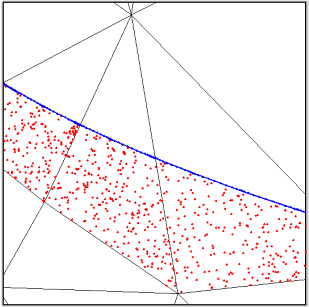
- Projection $f(\mathbf{x}) \approx \sum_{\mathbf{i}} F_{\mathbf{i}} \zeta_{\mathbf{i}}(\mathbf{x})$ minimizes the least-squares error at randomly-chosen sampling points, $\mathbf{x}_q$, inside the cut cell
- QR factorization and integration over κ leads to an expression for the quadrature weights:

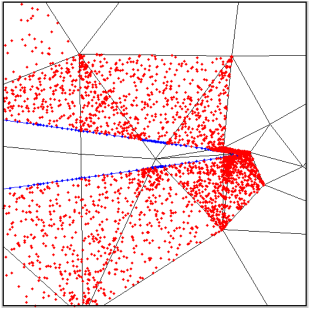
$$\int_\kappa f(\mathbf{x}) d\mathbf{x} \approx \sum_{\mathbf{i}} F_{\mathbf{i}} \int_\kappa \zeta_{\mathbf{i}}(\mathbf{x}) d\mathbf{x} = \sum_q f(\mathbf{x}_q) \underbrace{Q_{q\mathbf{j}} (R^{-T})_{\mathbf{j}\mathbf{i}} \int_\kappa \zeta_{\mathbf{i}}(\mathbf{x}) d\mathbf{x}}_{w_q}$$

Example: Quadrature Points

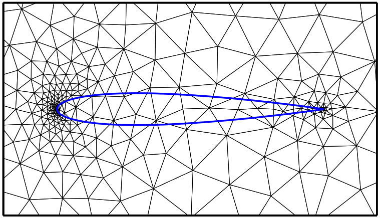
- NACA 0012
- 12 Gauss points per cut edge and spline segment
- Over 200 interior sampling points per element



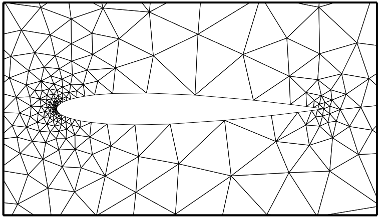




Example: Flow Solution



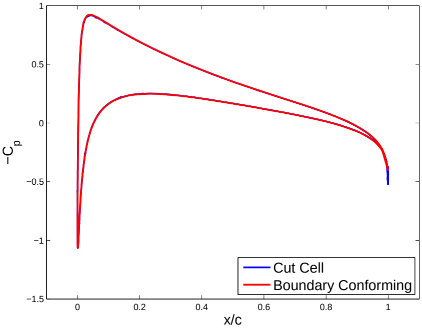
Cut-cell mesh



Boundary-conforming mesh

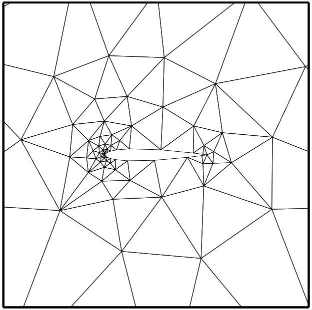
- $M = 0.5$, $\alpha = 3^o$
- $p = 2$ interpolation

C_p comparison

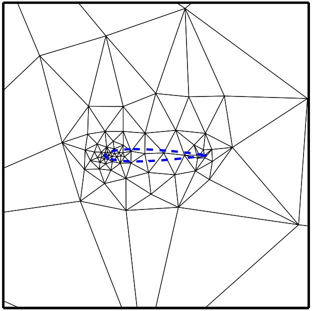


Drag Adaptation in a Viscous Case

NACA 0012, $M = 0.5$, $Re = 5000$, $\alpha = 2^o$

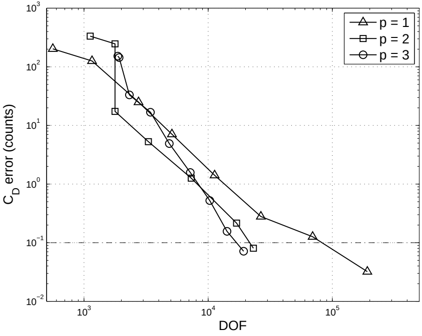


Initial boundary-conforming mesh

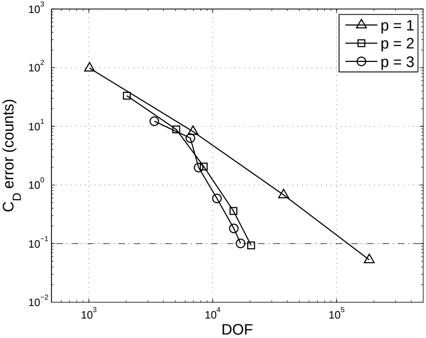


Initial cut-cell mesh

- Degree of freedom (DOF) vs. drag output error for $p = 1, 2, 3$.
- Requested tolerance is 0.1 drag counts (horizontal line).
- Cut-cell and boundary-conforming results are similar.

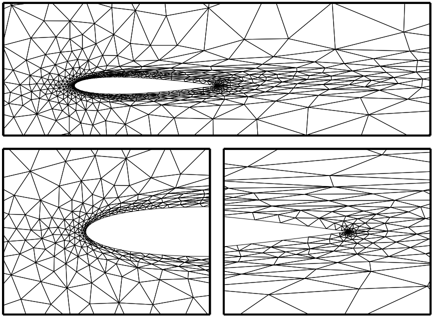


Boundary-conforming meshes

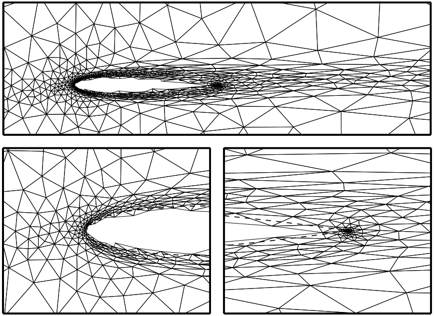


Cut-cell meshes

Viscous Case: Final Meshes



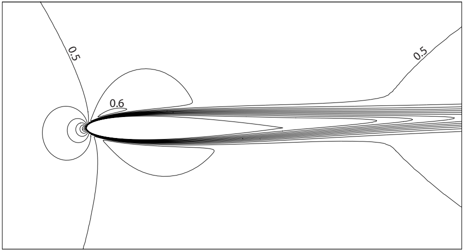
$p = 3$ adapted boundary-conforming mesh



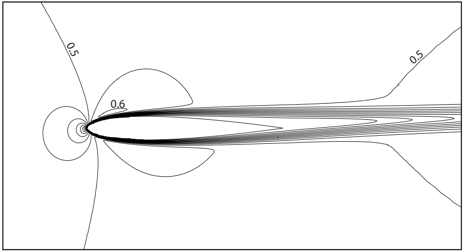
$p = 3$ adapted cut-cell mesh

$p = 1$ meshes have approximately 50 times more elements

Viscous Case: Mach Number Contours



Boundary-conforming, $p = 3$



Cut-cell, $p = 3$

$p = 1$ and $p = 2$ contour lines are very similar

Outline

Extension to Three Dimensions

- Cut-cell mesh generation becomes more difficult:
    - Geometry representation is not as straightforward as in 2D
    - Harder intersection problem: volume-surface instead of area-line
    - Integration rules needed on geometry surface, cut faces, and cut elements
- However, generating 3D boundary-conforming meshes is much more difficult compared to 2D:
    - Meshing around intricate 3D geometries is not trivial
    - No robust automated technique for curved geometries

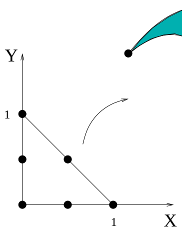
Geometry Definition

- Quadratic patches, 6 nodes per patch
- Patch surface ($\mathbf{x}$) is given analytically:

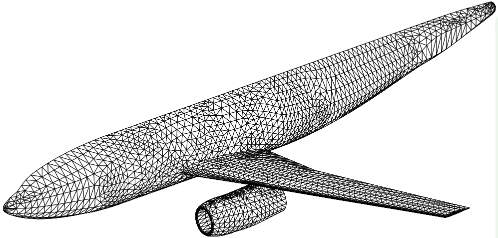
$$\mathbf{x} = \sum_j \phi(\mathbf{X})_j \mathbf{x}_j,$$

$\mathbf{X} = [X, Y]$: patch ref coords

- Water-tight representation (no holes)

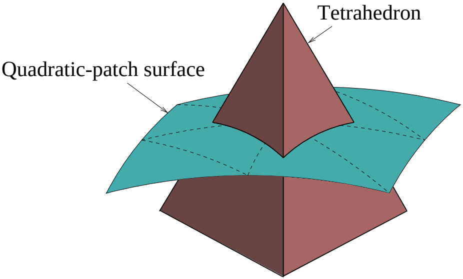


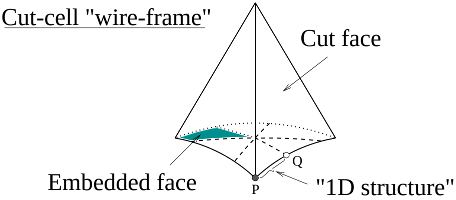
Patch reference space



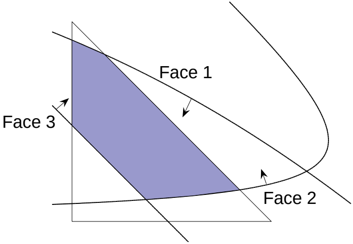
- Not exact; intermediate surface representation
- Surface tesselation and geometry interrogation from CAD via CAPRI
- Efficient resolution of curved surfaces

Intersection with Background Mesh





- Analytical intersection possible
- **Enabling feature**: intersection between a plane and a quadratic patch is a conic section (ellipse, hyperbola, etc.) in (X, Y)
- Robustness of cutting algorithm relies on robustness of conic-conic intersections



Conics in patch reference space

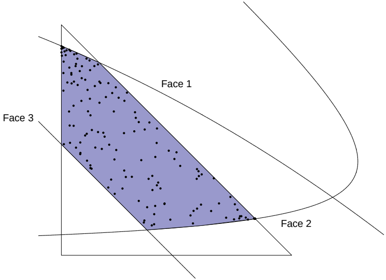
Integration

Requirements

- 2D integration on embedded boundary faces (on patches)
- 2D integration on cut faces (from background tetrahedra)
- 3D integration on cut-cell interiors

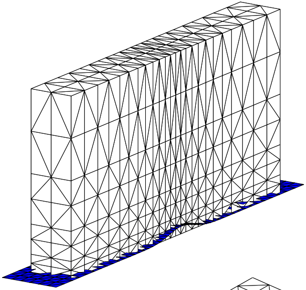
Methodology

- Gauss points on 1D edges of 2D embedded and cut faces
- Sampling point speckling for face integration (as in 2D)
- 3D extension of point speckling for cut elements

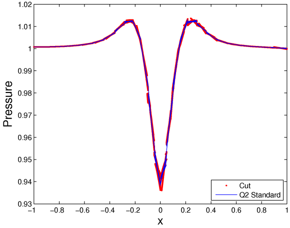


Example: Flow Solution

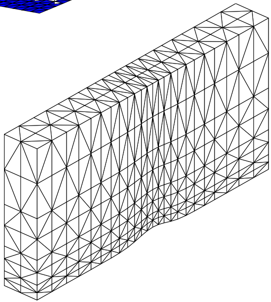
Cut-cell mesh:



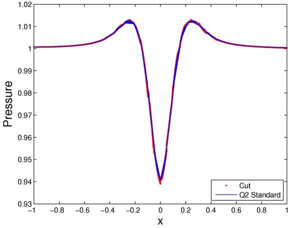
$p = 1$



Boundary-conforming mesh:

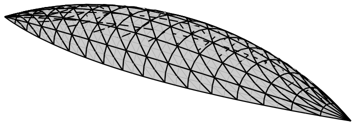


$p = 2$

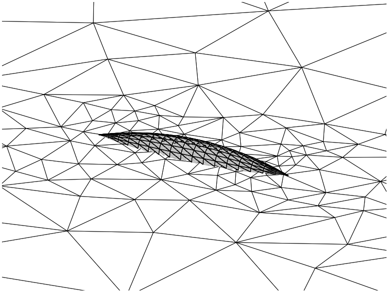


Flow Around a Football

- Inviscid, $M_\infty = 0.3$ flow around a body of revolution
- Model half the geometry



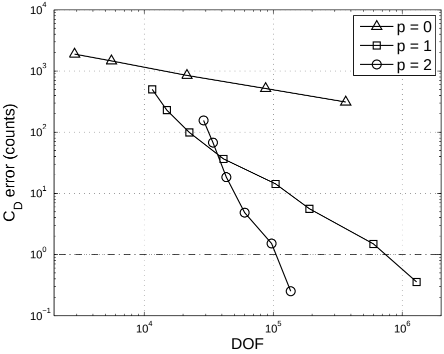
Surface representation: 256 quadratic patches



Initial background mesh: 2883 elements

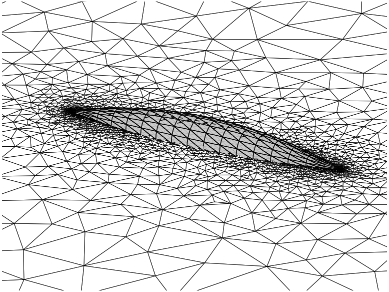
Football: Error Convergence

- Adapted on drag, with error tolerance of 1 drag count
- C_D measured using frontal cross-sectional area

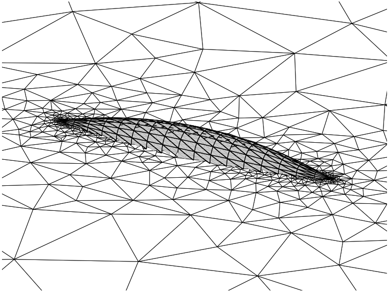


- $p = 0$ is not practical for accurate computation
- $p = 2$ converges much faster than $p = 1$

Football: Adapted Meshes

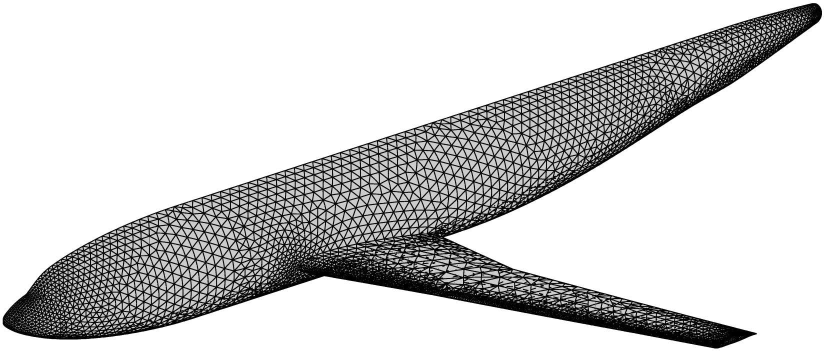


$p = 1$: 66304 elements: error = 4.0 drag counts



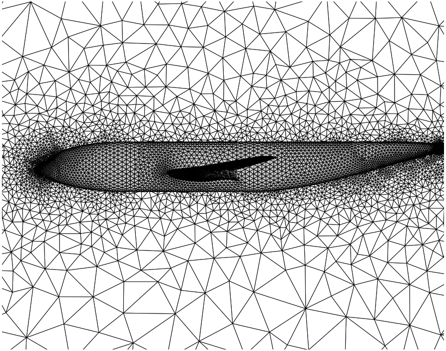
$p = 2$: 11354 elements: error = 0.6 drag counts

Wing-Body: Geometry

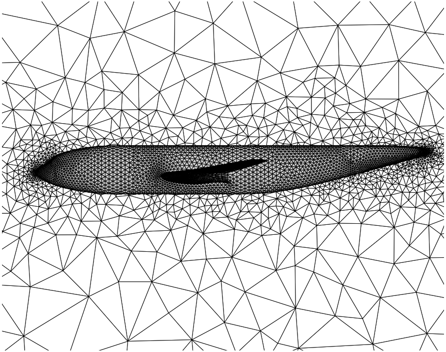


- Geometry from Drag Prediction Workshop
- 10,000 quadratic surface patches

Wing-Body: Adapted Meshes

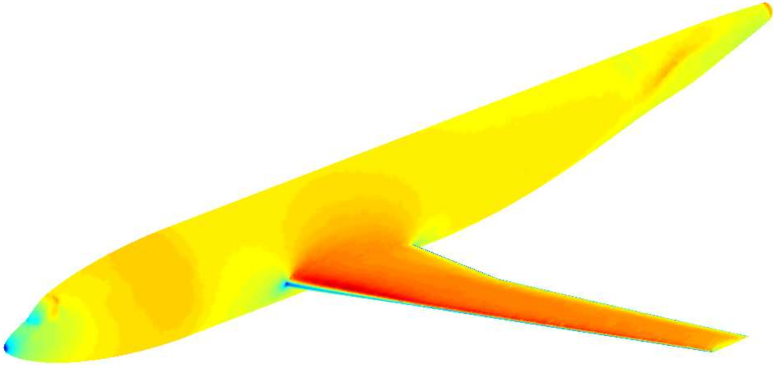


$p = 1$: 300,000 elements



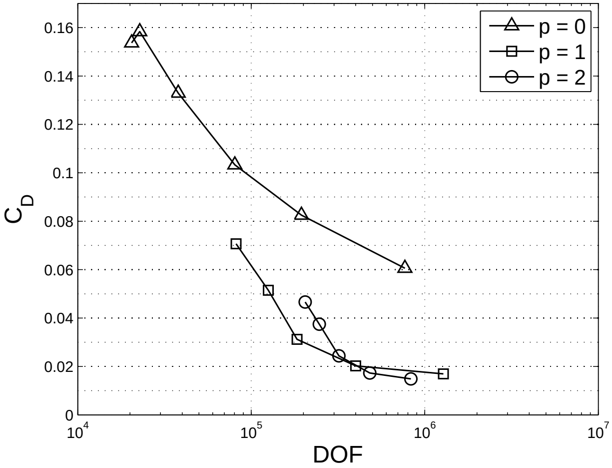
$p = 2$: 85,000 elements

Wing-Body: Solution



- Inviscid $M_\infty = 0.1$ flow
- Mach number contours shown for a $p = 2$ solution

Wing-Body Drag Comparison

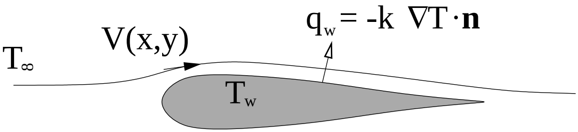


Outline

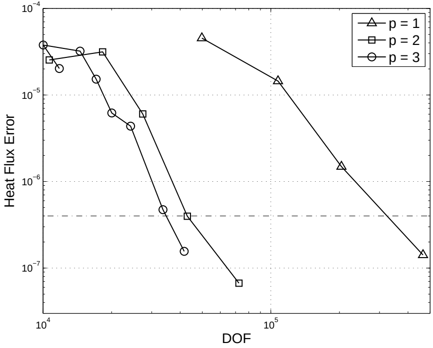
High Peclet Number Convection-Diffusion

- Test effectiveness of cut-cells + adaptation for highly-anisotropic boundary layer meshes
- Thickness of boundary layer governed by Peclet number, *Pe*

$$\nabla \cdot (\mathbf{V}T) - \nabla \cdot (k\nabla T) = 0, \qquad Pe = \frac{V_\infty L}{k}$$

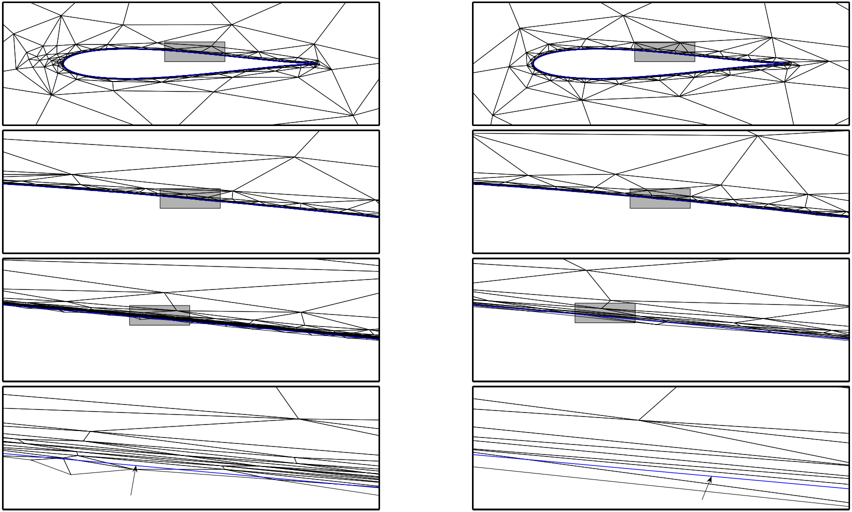


- $Pe = 4 \times 10^8$ simulates turbulent inner layer at $Re \sim 10^6$
- Heat flux output: $\mathcal{J} = \int_{\text{airfoil}} q_w ds$ + dual consistent terms
- Error tolerance is 1% of true heat flux



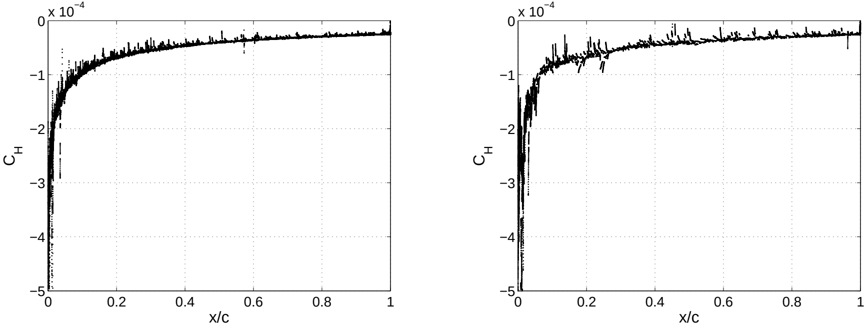
- $p = 1$ requires a factor of 10 more degrees of freedom than $p = 3$
- $p = 2$ performance is similar to $p = 3$

$Pe = 4 \times 10^8$: Adapted Meshes



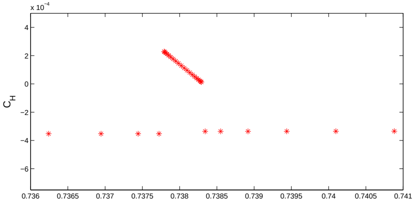
$p = 2$ $p = 3$

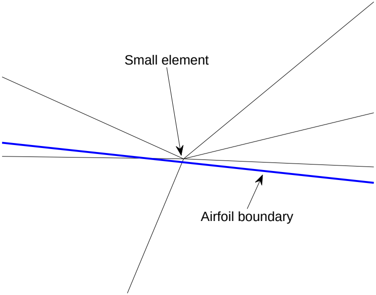
$C_H = q_w/(V_\infty \Delta T)$ along airfoil surface



$p = 2$ $\qquad$ $p = 3$

Diffusion Discretization on Small Elements

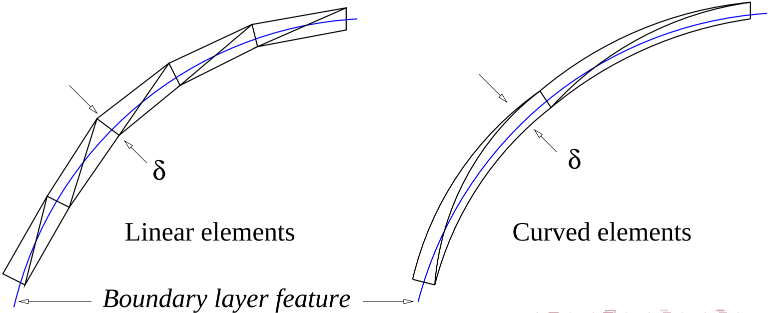




- Noise in derivative quantities observed on small elements adjacent to large elements
- Not specific to cut-cells, but cut-cell meshes are likely to contain small elements
- Problem due to viscous discretization + under-resolution
- Possible solutions:
  - Merge very small elements with neighbors
  - Seek a more robust discretization

Resolution of Curved Features

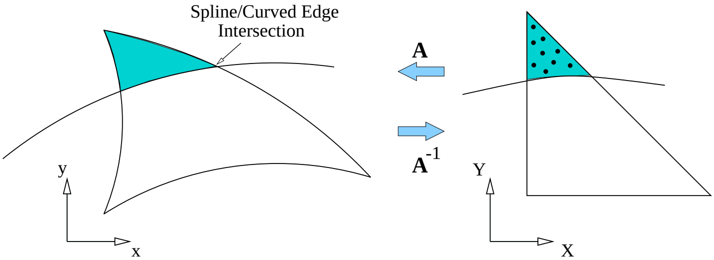
- Anisotropic features are efficiently resolved with anisotropic elements
- Feature curvature limits maximum element anisotropy when linear elements are used
- To take advantage of curved elements, solution representation must be in mapped (curved) space



Curved Elements

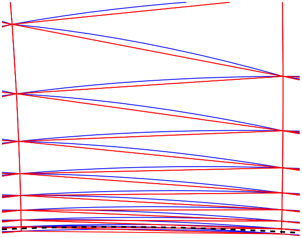
- DG boundary-conforming meshes employ curved elements to adequately represent curved boundaries
- Curved features may exist away from boundaries:
  - Unsteady shear layers
  - Curved shocks
- Ideally, elements should be curved based on the solution, not necessarily/just on the geometry
- Globally curving mesh elements while respecting geometry boundaries is a challenging task
- Cut cells with curved background meshes offer an alternative approach, more suitable for automation

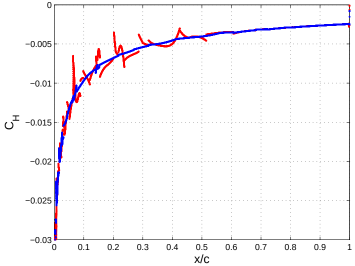
2D Cut Cells with Curved Elements



- Spline/curved-edge intersections obtained by applying Newton-Raphson method to the system of nonlinear equations
- Area integration rule derived in the element ref. space (X, Y)
  - Solution is polynomial in (X, Y)
  - Inverse of nonlinear mapping ($\mathbf{A}^{-1}$) is required to transform spline quadrature/intersection points into (X, Y) space
- Aside from cutting/integration, no fundamental code changes are required to incorporate curved cut cells

$Q = 1$ versus **$Q = 2$** boundary-layer cut-cell meshes





$p = 2$ solutions

- Convection diffusion for a Joukowski airfoil: output from cut $Q = 2$ mesh shows marked improvement over cut $Q = 1$ mesh
- Meshes were created manually - automated generation and adaptation of curved-element background meshes and extension to 3D is an ongoing research topic

Outline

Conclusions

- Cut-cells offer an automated alternative to boundary-conforming mesh generation, which is very often the bottleneck in CFD analysis and design
- Simplex cut cells allow for anisotropic meshes, which are necessary for practical viscous computations
- Adaptation with an output error estimator removes user guesswork from geometry-to-solution analysis process
- Curved background elements are more efficient at resolving curved anisotropic features; a robust adaptive scheme needs to be developed to take advantage of this efficiency
- Further work is required in making the viscous discretization more well-behaved on small elements

Acknowledgements

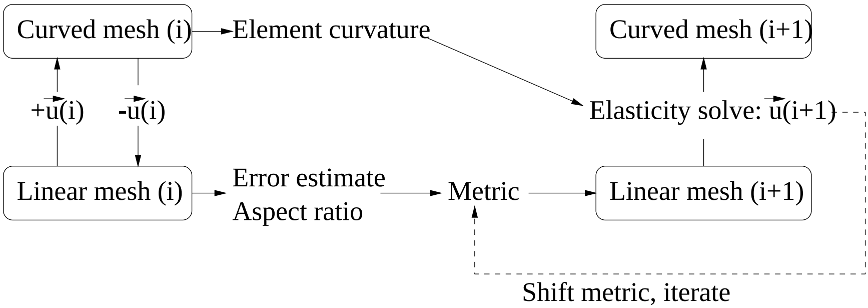
- Ph.D. Thesis advisor: David Darmofal
- Project-X team
- Department of Energy Computational Science Graduate Fellowship (DOE CSGF)

Questions?

Additional Research Directions

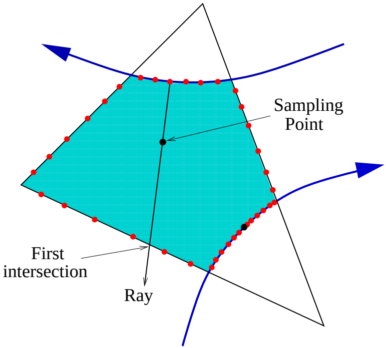
- **Decreasing computational cost of cut cells**
  - Reduce number of sampling points by more sophisticated (not random) selection; e.g. electric charge analogy
  - Allow for geometry approximation when background mesh is coarse
- **Improving conditioning of 3D volume integration rules at high orders**
  - Seek better support for integrand basis functions
  - Other polyhedra as alternatives to fitted bounding-box
- **Smoother geometry representation in 3D**
  - Investigate feasibility of intersection with more continuous geometry representations

Curved Mesh Adaptation



Sampling Point Selection

- $\mathbf{x}_q$ must lie inside the cut cell to keep the integrand evaluations physical for non-linear problems
- Currently choosing $\mathbf{x}_q$ randomly via ray-casting:



- Clusters of sampling points are undesirable in terms of QR conditioning $\Rightarrow$ use *oversampling*

Role of Anisotropy

Hessian Matrix

- Based on measuring degree and direction of quadratic variation.
- Standard practice in finite volume and linear FEM.
- Not reliably applicable to high-order solutions.

Example: $u = 1.0 + (x^2 + 16y^2) + \epsilon(64x^3 + y^3), \quad \epsilon << 1$

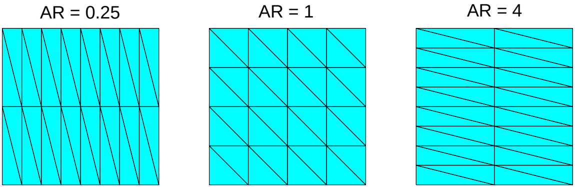
The Hessian matrix is

$$H = \begin{bmatrix} u_{xx} & u_{xy} \\ u_{yx} & u_{yy} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 32 \end{bmatrix} + O(\epsilon)$$

Ignoring $O(\epsilon)$ terms, Hessian analysis predicts

$$AR \equiv \frac{\Delta x}{\Delta y} = \sqrt{\frac{32}{2}} = 4.0.$$

Role of Anisotropy (continued)

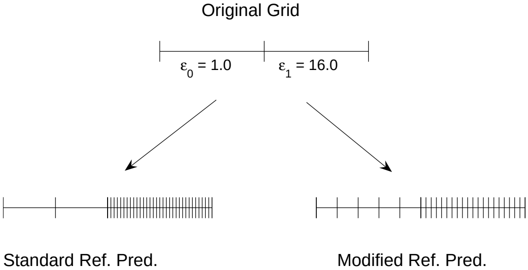


L_2 Interpolation Error

AR	$p = 1$	$p = 2$
0.25	2.31E-1	2.19E-7
1.0	5.91E-2	1.42E-6
4.0	2.37E-2	1.14E-5

For high-order ($p > 1$) anisotropy measures

- Use direction and magnitudes of $(p + 1)^{\mathrm{st}}$ derivatives.
- Directions of min/max. H.O. derivatives no longer guaranteed to be orthogonal.
- Currently employ a brute-force search for max H.O. derivative.



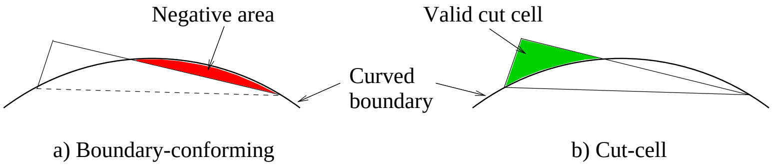
Assumed $\epsilon_\kappa \sim h_\kappa^1$

Standard Refinement Prediction	:	34	elements
Modified Refinement Prediction	:	25	elements

The Cut-Cell Advantage

Boundary-conforming mesh generation

- Common bottleneck in geometry-to-solution process
- Difficult (not robust) for complex 3D geometries
- Prone to failure on curved boundaries



Cut-cells

- Naturally handle curved boundaries and complex geometries
- Burden of robustness transferred to computational geometry
- Fully-automated mesh generation is possible