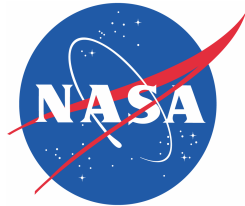


FUN3D Overview Part I: Inviscid Discretization

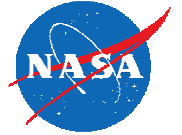


Eric J. Nielsen
FUN3D Development Team
NASA Langley Research Center

Technical Interchange with Sandia National Labs
March 7-8, 2011

<http://fun3d.larc.nasa.gov>

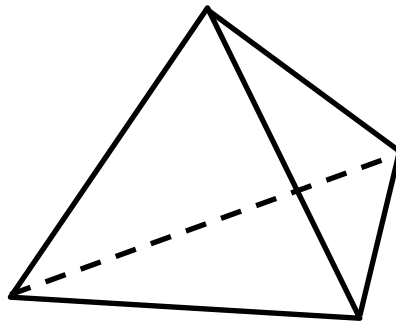
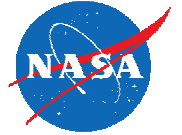
Introduction / Disclaimers



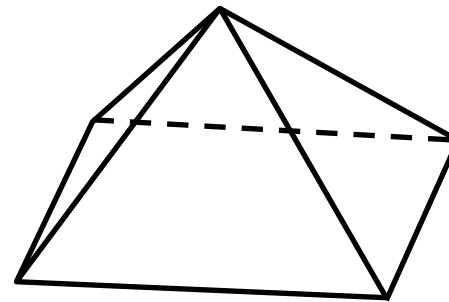
- Note there are hundreds of options/research paths available in FUN3D; we will only cover the “production” options here
 - Nothing is ever black and white
- We will scratch the surface on each topic, but reviewing the suggested publications in each area is recommended (all available on FUN3D website)
- The most basic overview of the discretization and algorithms is contained in

Anderson, W.K. and Bonhaus, D.L., “An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids,” *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1-21.
- Detailed course notes from a semester-long 1998 graduate course based on FUN3D are provided, as well as training manuals from a recent two-day FUN3D training course (streaming video on website)
- Many implementation issues and advanced capabilities not covered
 - Domain decomposition/parallel computing, computer science aspects, visualization...
- Let’s jot down today’s unanswered questions for tomorrow’s telecon
- Always feel free to send questions to fun3d-support@lists.nasa.gov

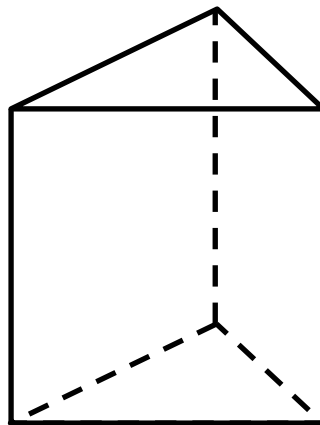
FUN3D Primal Element Types



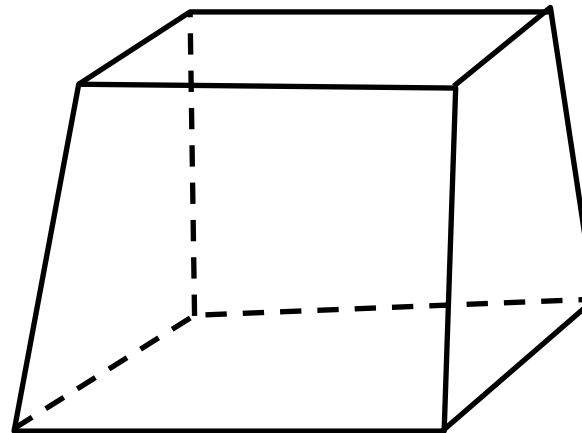
Tetrahedron



Pyramid



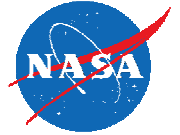
Prism



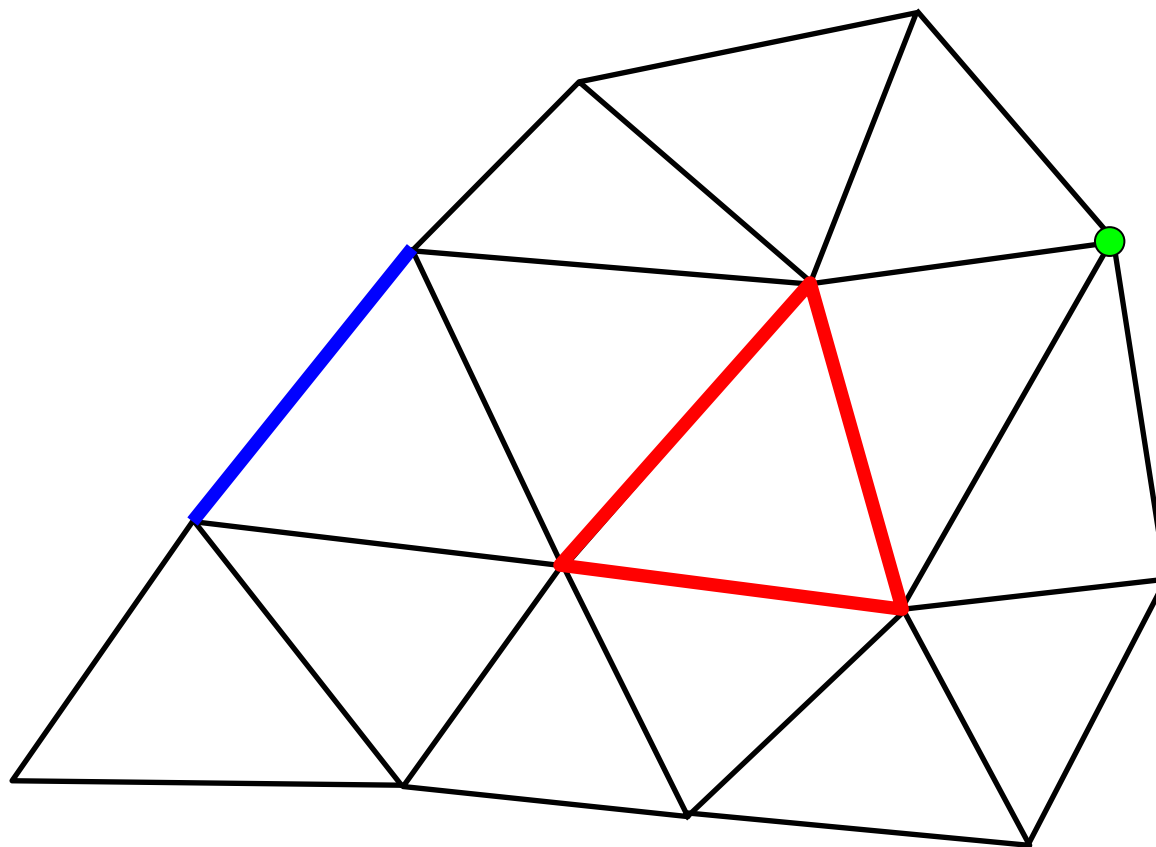
Hexahedron

- Meshes intended to be run as true 2-D cases must be composed of solely prisms and/or hexahedra; prisms must be oriented with triangles in x-z plane

Some Nomenclature / Terminology



For simplicity, we will work primarily in 2-D



Primal Cell
Edge
Node

Governing Equations



Here, we will focus on the discretization of the inviscid terms

$$A \frac{\partial \mathbf{Q}}{\partial t} + \oint_{\partial \Omega} \vec{\mathbf{F}}_i \cdot \hat{\mathbf{n}} dl - \oint_{\partial \Omega} \vec{\mathbf{F}}_v \cdot \hat{\mathbf{n}} dl = 0$$

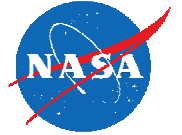
$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}$$

$$\vec{\mathbf{F}}_i = \begin{bmatrix} \rho \bar{U} \\ \rho u \bar{U} + \hat{n}_x p \\ \rho v \bar{U} + \hat{n}_y p \\ (E + p) \bar{U} \end{bmatrix}$$

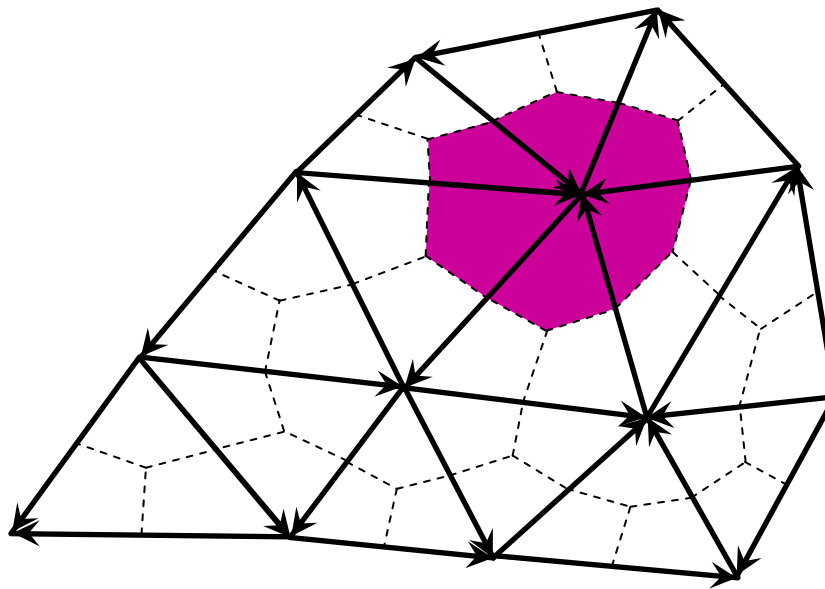
$$p = (\gamma - 1) \left[E - \rho \frac{(u^2 + v^2)}{2} \right]$$

- Nondimensionalization not covered here; see Tutorial slides on FUN3D website or CFL3D manual at <http://cfl3d.larc.nasa.gov>

Control Volumes



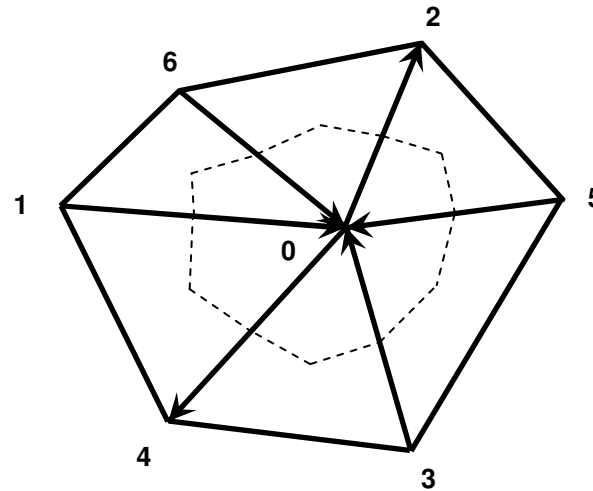
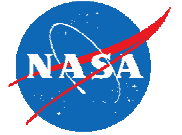
A node-based scheme is used where the equations are solved on a set of non-overlapping control volumes surrounding each node



The dashed lines form the *dual cells* of the mesh

- The dual mesh is constructed by connecting edge midpoints with primal cell centroids: “median dual”
- Each edge is arbitrarily assigned a direction
- The variables are stored at the nodes and the discrete flux balance will be evaluated on the dual cells of the mesh

Inviscid Flux Balance

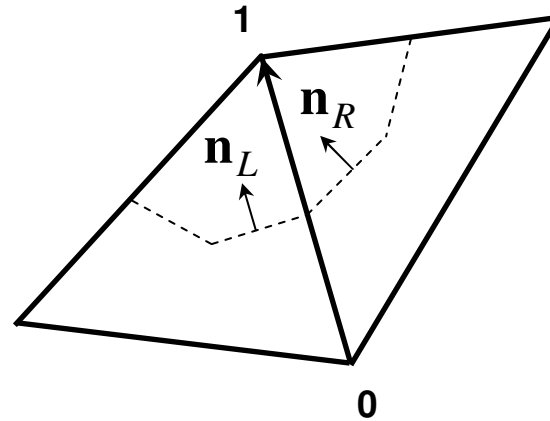
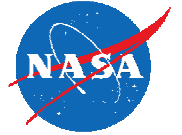


To form the flux balance at node 0:

$$\oint_{\partial\Omega} \vec{F}_i \cdot \hat{n} dl = \oint_{\partial\Omega} \vec{F}_i \cdot d\mathbf{n} \approx \sum_{i=1}^N \Phi(\mathbf{q}_L, \mathbf{q}_R, \hat{n}) \times l_i$$

- $\mathbf{q}$ is the primitive variable vector $\{\rho, u, v, p\}^T$
- The numerical flux Φ can be evaluated on each surface of the dual mesh based on $\mathbf{q}_L$ and $\mathbf{q}_R$
- Left and right sides are determined by the pre-assigned edge directions
- Since unit normal is defined as outward from the left CV, contributions are added to the CV on the left and subtracted from that on the right

Inviscid Flux Balance



The combined directed area is used:

$$\mathbf{n} = \mathbf{n}_L + \mathbf{n}_R$$

$$\oint_{\partial\Omega} \vec{\mathbf{F}}_i \cdot \hat{\mathbf{n}} dl = \oint_{\partial\Omega} \vec{\mathbf{F}}_i \cdot d\mathbf{n} \approx \sum_{i=1}^N \Phi(\mathbf{q}_L, \mathbf{q}_R, \hat{\mathbf{n}}) \times l_i$$

- Loop over the edges and evaluate the flux; send the result to the sums at the nodes

Flux Evaluation



- FUN3D supports a number of flux functions Φ ; Roe's scheme is commonly used:

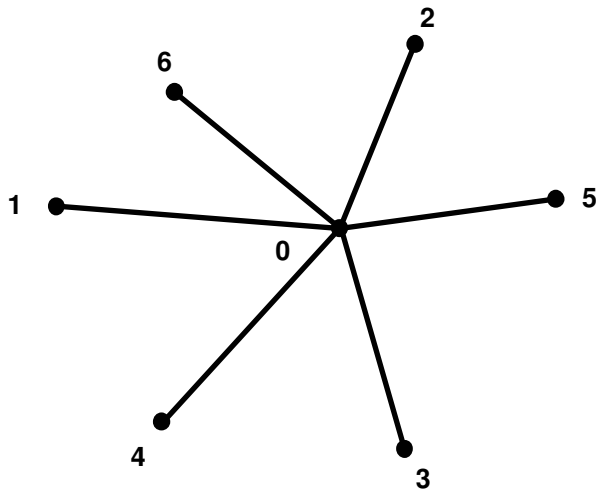
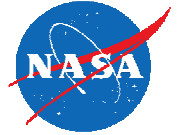
$$\Phi = \frac{1}{2}(\mathbf{F}(\mathbf{q}_L) + \mathbf{F}(\mathbf{q}_R)) \cdot \hat{\mathbf{n}} - \frac{1}{2}|\mathbf{A}(\mathbf{q}, \hat{\mathbf{n}})|(\mathbf{q}_R - \mathbf{q}_L)$$

- The matrix $\mathbf{A}$ is evaluated at the Roe-averaged state
- If $\mathbf{q}_L$ and $\mathbf{q}_R$ are taken as the values at the nodes and the matrix $\mathbf{A}$ is neglected, this yields a Galerkin discretization
- If $\mathbf{q}_L$ and $\mathbf{q}_R$ are taken as the values at the nodes, this yields a first-order accurate scheme
- To achieve higher-order accuracy, the values at the nodes are extrapolated to the cell face using an unstructured MUSCL scheme:

$$\mathbf{q}_L = \mathbf{q}_i + (1 - \kappa)\nabla\mathbf{q}_i \cdot \left(\frac{\mathbf{r}_j - \mathbf{r}_i}{2}\right) + \frac{\kappa}{2}(\mathbf{q}_j - \mathbf{q}_i)$$

- $\nabla\mathbf{q}_i$ is the gradient of the variables at node i and $\mathbf{r}$ is the vector from the node to the face; resulting stencil is nearest and next-nearest neighbors
- FUN3D takes $\kappa = 0$ for tetrahedral grids; $\kappa = 0.5$ otherwise
- Limiting can be applied in many forms; not covered here

Least-Squares Gradient Reconstruction



Data at each node i can be expressed as

$$\mathbf{q}_i = \mathbf{q}_0 + \mathbf{q}_{x_0}(x_i - x_0) + \mathbf{q}_{y_0}(y_i - y_0)$$

Rearrange to obtain over-determined system of linear equations $\mathbf{Ax}=\mathbf{b}$:

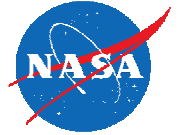
$$\begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ \vdots & \vdots \\ \Delta x_N & \Delta y_N \end{bmatrix} \begin{Bmatrix} \mathbf{q}_{x_0} \\ \mathbf{q}_{y_0} \end{Bmatrix} = \begin{Bmatrix} \mathbf{q}_1 - \mathbf{q}_0 \\ \mathbf{q}_2 - \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_N - \mathbf{q}_0 \end{Bmatrix}$$

Decompose $\mathbf{A}$ into $\mathbf{QR}$ so that $\mathbf{x}=\mathbf{R}^{-1}\mathbf{Q}^T\mathbf{b}$

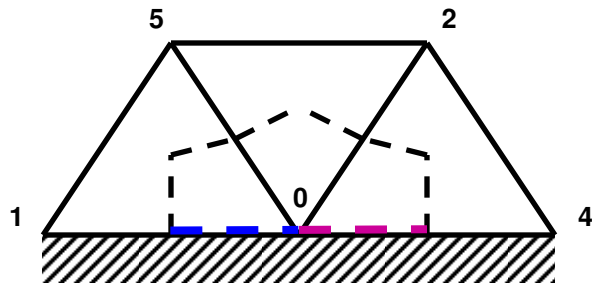
This approach yields edge formulas for the gradients based on precomputed weights depending solely on geometry:

$$\mathbf{q}_{x_0} = \sum_{i=1}^N W_i^x (\mathbf{q}_i - \mathbf{q}_0) \quad \mathbf{q}_{y_0} = \sum_{i=1}^N W_i^y (\mathbf{q}_i - \mathbf{q}_0)$$

Boundary Conditions: Tangency



After computing the fluxes on each edge, we must close off the boundaries

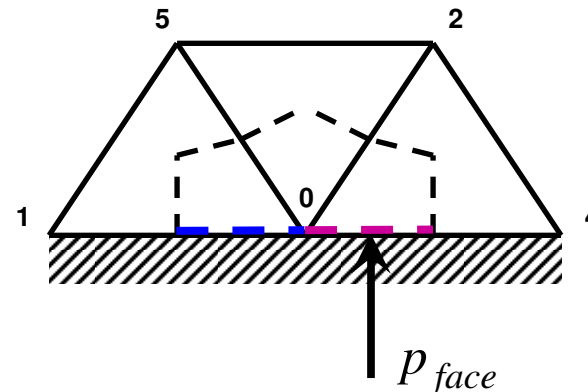


The velocity normal to the solid wall is zero, so we are left with only a pressure term:

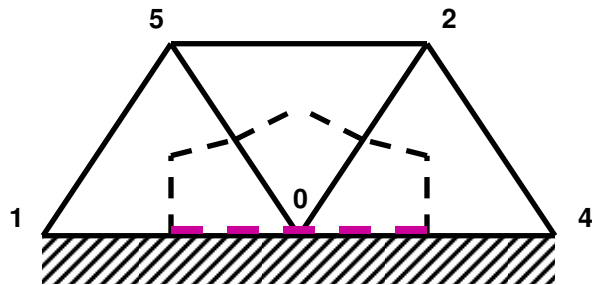
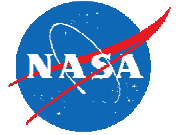
$$\vec{F}_{boundary} = \begin{bmatrix} \rho \bar{U} \\ \rho u \bar{U} + \hat{n}_x p \\ \rho v \bar{U} + \hat{n}_y p \\ (E + p) \bar{U} \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{n}_x p \\ \hat{n}_y p \\ 0 \end{bmatrix}$$

One can show that for triangles, to preserve a linear distribution of data:

$$p_{face} = \frac{5}{6} p_0 + \frac{1}{6} p_4$$



Boundary Conditions: Farfield



To get the data in the farfield, we use locally one-dimensional Riemann invariants:

$$R^{\pm} = \bar{U} \pm \frac{2c}{\gamma - 1}$$

Depending on the local flow conditions, $R^{\pm}$ are evaluated from inside/outside the domain.

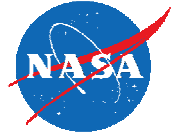
Given these values, we can then solve for the quantities on the boundary and form the flux.

Other Items of Interest



- Cell-centered/node-centered studies (early 90's and more recent)
 - Robustness
 - Accuracy
 - Cost
 - Implications for adjoint-based schemes
- Weighted vs unweighted least-squares schemes
- Least-squares issues on stretched grids near curved boundaries
- Edge-lumping on general elements
- Recent third-order scheme of Katz
- Aerothermodynamics: high-energy flows, flows with strong shocks
- Limiting strategies

Some Relevant Publications



Gnoffo, P.A., "Updates to Multi-Dimensional Flux Reconstruction for Hypersonic Simulations on Tetrahedral Grids," AIAA-2010-1271, 2010.

Diskin, B. and Thomas, J.L., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes," AIAA-2010-1079, 2010.

Vatsa, V.N. and White, J.A., "Calibration of a Unified Flux Limiter for Ares-Class Launch Vehicles from Subsonic to Supersonic Speeds," 56th JANNAF Propulsion Meeting, Las Vegas, NV, April 2009.

Gnoffo, P.A., "Multi-Dimensional, Inviscid Flux Reconstruction for Simulation of Hypersonic Heating on Tetrahedral Grids," AIAA-2009-0599, 2009.

Thomas, J.L., Diskin, B., and Rumsey, C.L., "Towards Verification of Unstructured-Grid Solvers," *AIAA Journal*, Vol. 46, No. 12, pp. 3070-3079, 2008.

Diskin, B. and Thomas, J.L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," NIA Report No. 2007-08.

Gnoffo, P.A., "Semi-Analytic Reconstruction of Flux in Finite Volume Formulations," AIAA-2006-1090, 2006.