Premo - An Unstructured Mesh Compressible Flow Solver

SAND - 2005-5781C

Thomas M. Smith, Curtis C. Ober
Alfred A. Lorber
and
Russell W. Hooper
Sandia National Laboratories, Albuquerque, NM

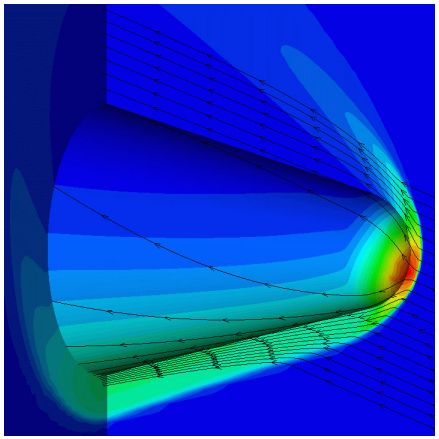
Sixth Biennial Tri-Laboratory Engineering Conference
Monterey Marriot Hotel
Monterey, CA
September 12-15, 2005

Supported by ASC Applications and ASC Algorithms

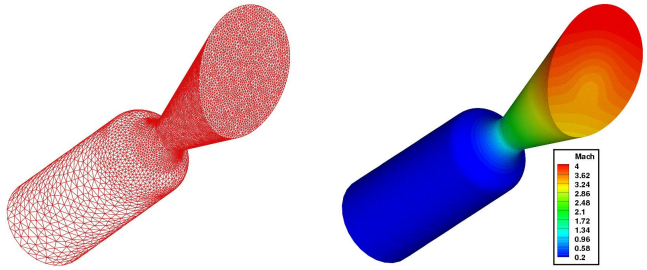
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04 94AL85000

Outline

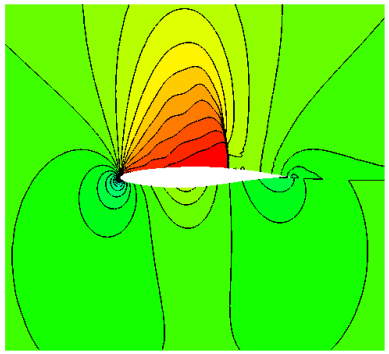
- Premo project
- Core discretizations and solvers
- Current capabilities - examples
- Emerging capabilities
- Summary



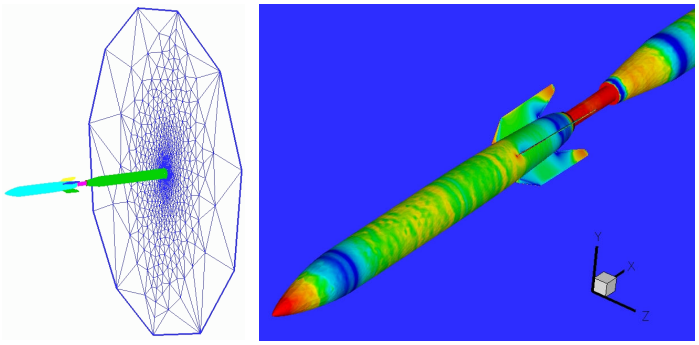
Pressure contours, Mach 4



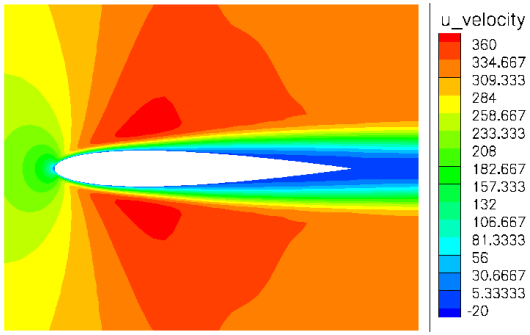
Mach no., Courtesy of M. Barone



RAE2822 Mach No. Contours



B61 (Euler): M=0.8, α=0
Mesh courtesy of J.L. Payne



NACA0012 u-velocity

Sandia National Laboratories

Primary Goal

Develop production capabilities to perform compressible flow simulations for the Nuclear Weapons Complex.

- Compressible subsonic through hypersonic
- Inviscid and viscous flows
- Laminar and turbulent flows
- Steady-state and transient

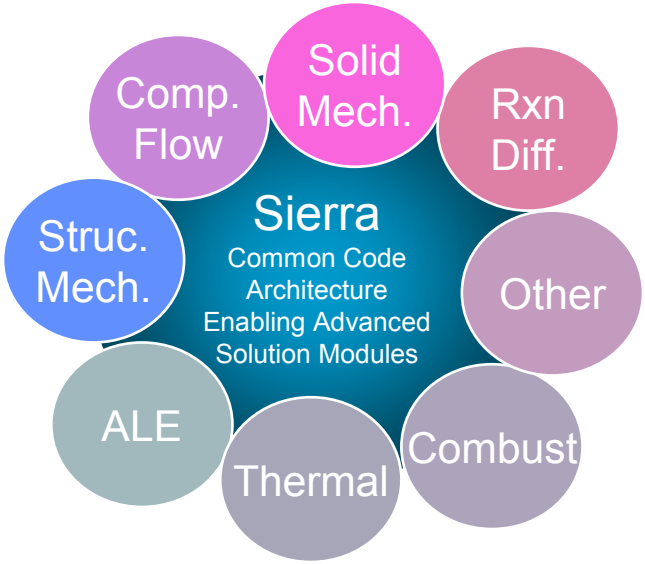
- Chemically reacting flow

- Multi-physics coupling
- Arbitrary body motion
- Relative body motion
- Adaptivity

Sandia National Laboratories

SIERRA Framework

- SIERRA Framework
  - Unstructured mesh finite-element infrastructure
  - Parallel services
  - H-Adaptivity
  - Load balancing
  - I/O services (restart)
  - Scalable linear/nonlinear equation solver interface
    - Trilinos/Nox, AztecOO, ML
  - Common interfaces and transfer operators for multi-physics code coupling
- SIERRA Code Management System (SCMS)
  - Configuration management (archiving, building, and testing software across multiple platforms)
  - Software Engineering infrastructure (requirements management, issue tracking, …)



Premo

(premō) [Latin] – to squeeze (compress)

Approach

- Develop Premo within the SIERRA framework
- Use an node-centered finite-volume method
- Use upwind advection schemes
- Use Trilinos and NOX for linear and nonlinear solves
- Five years in development

Supporting Staff

Principle Investigator – C.C. Ober, (1433)

Project Manager – S. E. Gianoulakis (1541)

Product Manager – J. L. Payne (1515)

Premo Developers – Ober (1433), Lorber (1541), Smith (1433)

Nonlinear Solvers – Hooper (1416), Smith (1433), Lorber (1541), Shadid (1437), Williams (1543)

Verification – Bond (1515), Knupp (1411), Ober (1433)

Time Integration – Ober (1433), Shadid (1437)

Linear Algebra ML – Tuminaro, Hu, (1414), Shadid, Lin (1437)

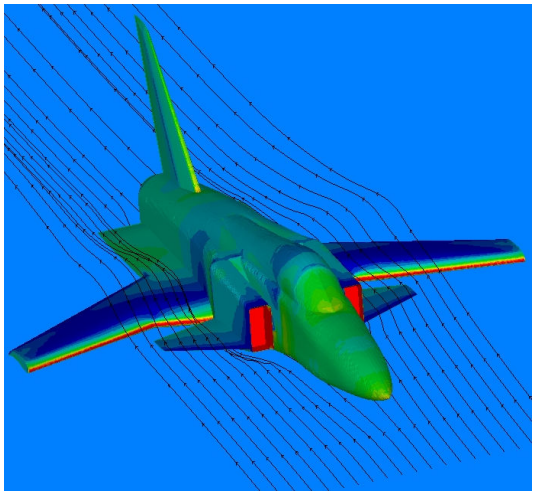
Optimization and Error Estimation – Van Bloemen Waanders (1411), Bartlett (1411)

External Contacts – McLay (U. T. Austin), Berggren (FOI), Haselbacher (U. Illinois)

Customers/Users

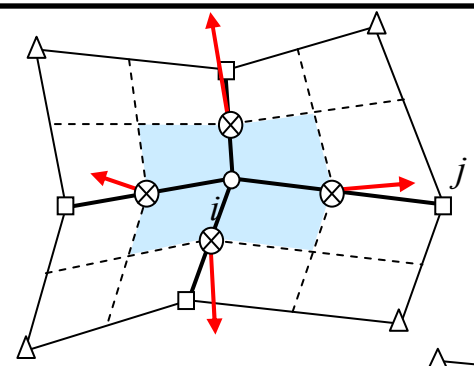
- **Aero. & CFM (1515, Basil Hassan)**
  - **Matt Barone (1515)**
  - **Ryan Bond (1515)**
  - **Jeff Payne (1515)**

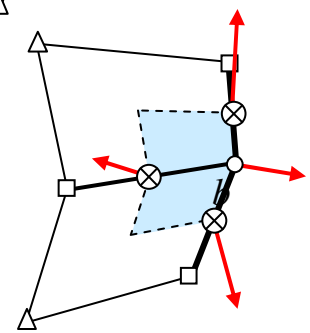
X29 – M=0.75, α=5 deg

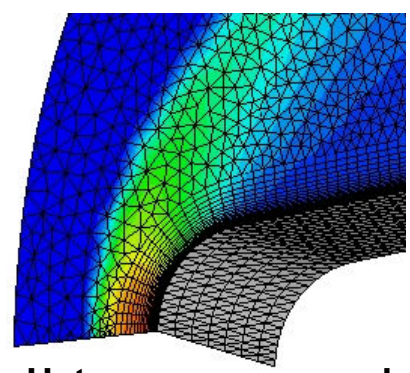


Node-Centered Finite-Volume Method

- **Compute on the dual of the finite element mesh**
  - Control surface area vectors are attached to edges
  - Control volumes are attached to nodes
  - External surfaces sub-divided into sub-surfaces
- **Edge based Assembly**
  - Interior flux quadrature achieved by looping over edges
  - BC flux quadrature achieved by looping over external faces
- **Mesh Transparency -** solve on hexs, tets, prisms, pyramids, and heterogeneous meshes
- **Governing Equations:** Euler, Navier-Stokes, RANS (Spalart-Allmaras) ideal gas; Euler equilibrium gas
- **Upwind advection**
  - Roe, S&W, HLLC, STVD schemes
  - MUSCL type reconstruction
  - Green-Gauss and weighted least-squares gradient construction
  - Barth-Jesperson, Vekatakrishnan, VanAlbada, VanLeer, MinMod limiters
- **Explicit and fully implicit time integration (2nd-order)**
  - Four stage low storage Runge-Kutta
  - Preconditioned Jacobian-Free Newton-Krylov
- **Implicit B.C.s -** for exterior and interior flows
- **Support auxiliary variables -** including force coeffs.
- **Support user defined:** B.C.s, I.C.s, source terms, auxiliary variables
- **Platforms:** Linux, Sun, SGI, IBM (Janus, Rogue, Liberty, Shasta, Spirit, Reddish, White, Renegade, Wizard) Apple G5 port in progress







Heterogeneous meshes

Sandia National Laboratories

Implicit Time Integration

(T.M. Smith and R.W. Hooper)

$$\mathbf{F}(\mathbf{W}) = \frac{\partial \mathbf{W}}{\partial t}\Delta V + \mathbf{R}(\mathbf{W}) = 0$$

Transient

$$\mathbf{F}(\mathbf{W}) = 0$$

For n=1,Nmax do

$$t^{n+1} = t^n + \Delta t$$

$$\left(\frac{\alpha}{\Delta t}\mathbf{I}\Delta V + \frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)\Delta\mathbf{W} = -\mathbf{F}(\mathbf{W})$$

$$\mathbf{J}\Delta\mathbf{W} = -\mathbf{F} \rightarrow \begin{cases} \mathbf{J}\mathbf{M}^{-1}\mathbf{y} = -\mathbf{F} \\ \mathbf{M}\Delta\mathbf{W} = \mathbf{y} \end{cases}$$

$$\left\|\mathbf{J}\Delta\mathbf{W} + \mathbf{F}(\mathbf{W})\right\|_2 < \eta\left\|\mathbf{F}(\mathbf{W})\right\|_2$$

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \Delta\mathbf{W}$$

$$\dot{\mathbf{W}}^{n+1} = \frac{\alpha}{\Delta t}\left(\mathbf{W}^{n+1} - \mathbf{W}^n\right) - (1-\alpha)\dot{\mathbf{W}}^n$$

$$\frac{\left\|\mathbf{F}(\mathbf{W}^n)\right\|}{\left\|\mathbf{F}(\mathbf{W}^0)\right\|} < tol$$

end do

Pseudo-Transient

$$\mathbf{R}(\mathbf{W}) = 0$$

$$\left(\frac{1}{\Delta t}\mathbf{I}\Delta V + \frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)\Delta\mathbf{W} = -\mathbf{R}(\mathbf{W})$$

$$\mathbf{J}\Delta\mathbf{W} = -\mathbf{R} \rightarrow \begin{cases} \mathbf{J}\mathbf{M}^{-1}\mathbf{y} = -\mathbf{R} \\ \mathbf{M}\Delta\mathbf{W} = \mathbf{y} \end{cases}$$

$$\left\|\mathbf{J}\Delta\mathbf{W} + \mathbf{R}(\mathbf{W})\right\|_2 < \eta\left\|\mathbf{R}(\mathbf{W})\right\|_2$$

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \Delta\mathbf{W}$$

$$\Delta t^{n+1} = f(\Delta t^n, \mathbf{R}^n, \mathbf{R}^{n-1})$$

$$\frac{\left\|\mathbf{R}(\mathbf{W}^n)\right\|}{\left\|\mathbf{R}(\mathbf{W}^0)\right\|} < tol$$

For $\Delta t \rightarrow \infty$ recover Newton's Method

Sandia National Laboratories

Preconditioned Matrix-Free Newton-Krylov

- GMRES iterative solver only requires the action of a matrix on a vector – use a Matrix-Free operator

$$\mathbf{JP} \approx \frac{\mathbf{F}(\mathbf{W}+\varepsilon\mathbf{P})-\mathbf{F}(\mathbf{W})}{\varepsilon}$$

- Preconditioned Matrix-Free operator in GMRES

$$\mathbf{JM}^{-1}\mathbf{P} \approx \frac{\mathbf{F}(\mathbf{W}+\varepsilon\mathbf{M}^{-1}\mathbf{P})-\mathbf{F}(\mathbf{W})}{\varepsilon}$$

- Preconditioning matrix should approximate $\mathbf{J}$ but be inexpensive to form in terms of cpu and storage costs and be well conditioned
  - Based on edge-graph – reduces the size, fast assembly
  - Based on 1st order upwind advection

$$J_{ij} = \frac{\partial \mathbf{F}}{\partial \mathbf{W}} \approx \frac{F_i\left(\mathbf{W}+\varepsilon e_j\right)-F_i\left(\mathbf{W}\right)}{\varepsilon}$$

- (1) Finite-Difference with coloring (FDC)
  - Only requires residual evaluations
  - Allows fast prototyping of new physics
  - Coloring minimizes the cost of forming $\mathbf{J}$ by perturbing several variables at once such that at most, one equation is affected by each perturbed variable
- (2) Automatic Differentiation – rewrite $\mathbf{R}$ templating on scalar type – TFAD (AD)
- (3) Approximate analytic - neglect sensitivity of the dissipation terms (ROE)
- (4) Hybrid - analytic for some terms, finite difference the rest (ROEFD)

- Once the preconditioning matrix is formed it is applied to a vector by NOX (e.g., iluk, ilut, block Jacobi, block symmetric Gauss-Seidel or ML)

Sandia National Laboratories

Operator Evaluation Cost

Operator	Hex mesh (NACA0012) (1 elem, GG, vanAlbada)	Hex mesh (Blunt cone) (3D, GG, vanAlbada)	Tet mesh (B61, no fins) (3D, LS, vanAlbada)
RHS	1	1	1
MF	1	1	1
Mvec (no storage)	5.82	6.60	----------
AD	7.34	6.98	4.78
ROEFD	5.54	4.08	3.29
ROE	5.30	3.91	2.93
FDC (colors)	32.98 (61)	40.28 (76)	81.24 (156)

Operator	Euler, Hex mesh (1 elem wide)	NS, Hex mesh (1 elem wide)	SARANS, Hex mesh (1 elem wide)
RHS	1	1	1
MF	1	1	1
AD	7.34	7.56	9.17
FDC (colors)	32.98 (61)	34.94 (61)	38.00 (67)

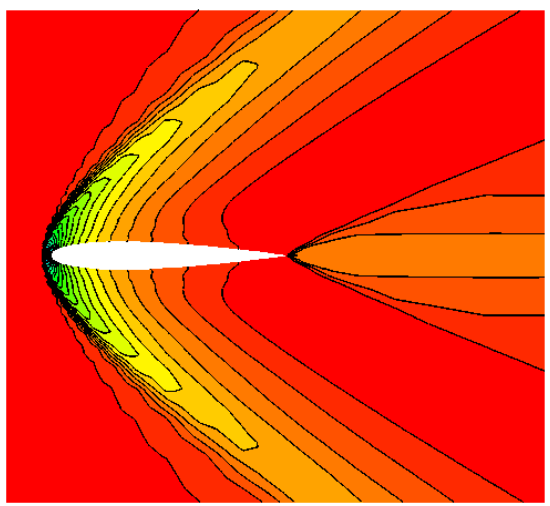
Timings do not include factorization cost
(e.g. For NACA0012 ilu(0) ~30)

Sandia National Laboratories

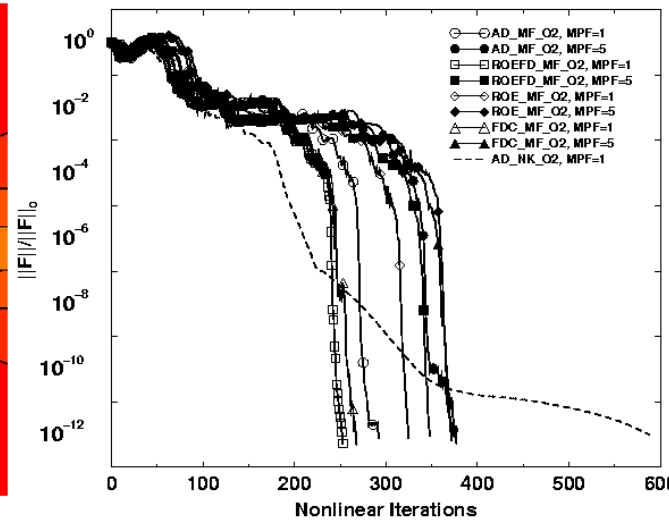
Supersonic Inviscid Flow

	MPF	CFL	f_A	NLIts	LIts	CPU
AD_MF_O2	1	1-5000	1.05	293	1,400	1,260
AD_MF_O2	5	1-5000	1.2	378	1,633	989
ROEFD_MF_O2	1	1-5000	1.05	254	1,262	1,082
ROEFD_MF_O2	5	1-5000	1.2	349	1,337	886
ROE_MF_O2	1	1-5000	1.05	326	1,453	1,299
ROE_MF_O2	5	1-5000	1.2	373	1,592	1,000
FDC_MF_O2	1	1-5000	1.05	269	1,909	2,874
FDC_MF_O2	5	1-5000	1.2	376	1,563	1,526
AD_NK_O2	1	1-50	1.05	589	1,918	1,578

Flow Conditions:
M=2
Alpha=0
Mesh:
NACA0012 O-grid
129x33x2
Boundaries 25 chords away from airfoil
Solver Settings:
$\eta=10^{-1}$
LTS
Adaptive CFL
Ilu(0), no RCM
Discretization:
Roe, GG, vanAlbada



Mach No. Contours



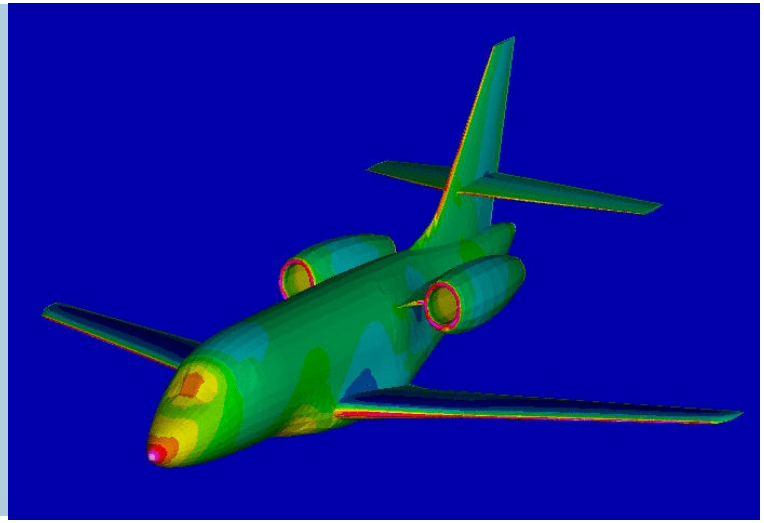
Sandia National Laboratories

Multi-Level Solver

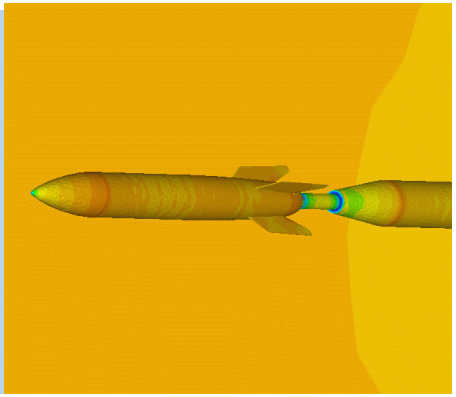
(R. Tuminaro and J. Hu)

- Using ML package in Trilinos through NOX nonlinear solver interface

- **Falcon problem (13^{+} Million dofs, 150 procs)**
- **Pseudo-transient + Newton**
- **Euler flow, Mach .75**
- **Linear solver: 109 solves, tolerance= 10^{-4}**
- **GMRES/ILU ➔ 7620 sec**
- **GMRES/MG ➔ 3787 sec**
- **10x improvement on final linear solve.**
- **> 5x gains on some problems over entire sequence**



- **B61 problem (6.5 million dofs, 64 procs)**
- **Pseudo-transient + Newton**
- **Euler flow, Mach .8**
- **Linear solver: 173 solves, tolerance= 10^{-4}**
- **GMRES/ILU ➔ 7494 sec**
- **GMRES/MG ➔ 3704 sec**



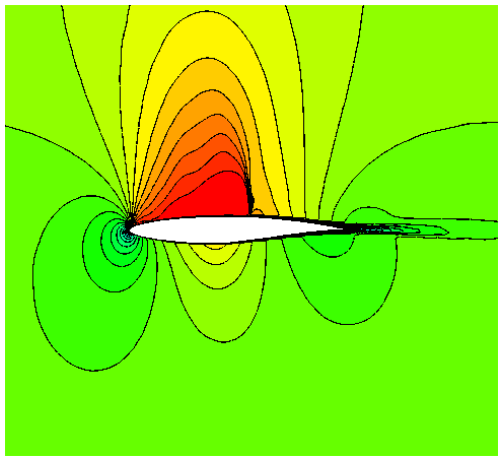
Transonic Turbulent Flow

Flow conditions:
M=0.729
Re=6.5E6
Alpha=2.31

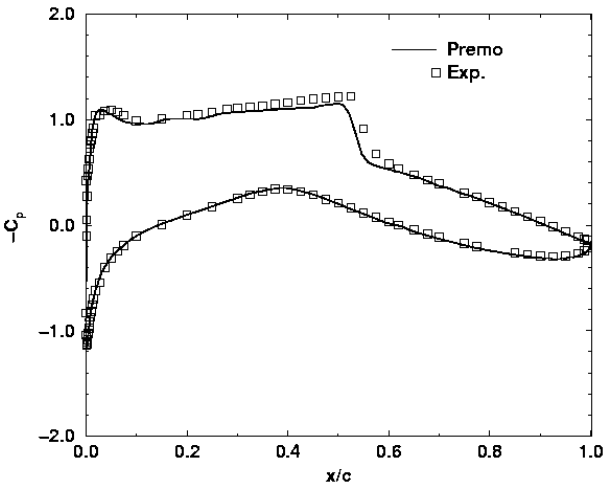
Mesh:
C-grid: 369x65x2
Boundaries ~20 chords
Away from airfoil

Solver Settings:
Ilu(1)/RCM

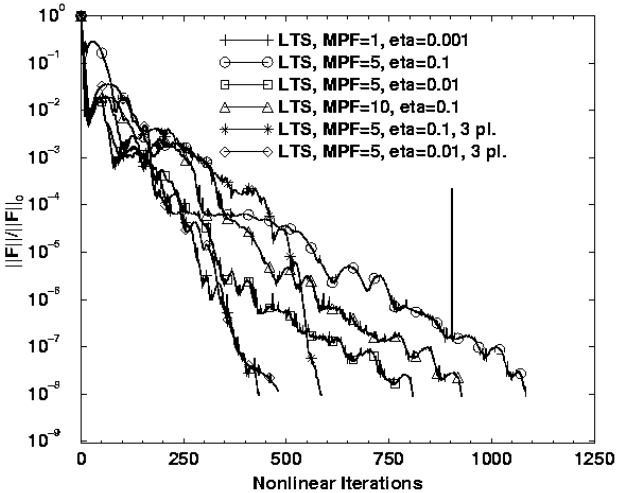
Discretization:
Spalart-Allmaras turbulence model
Roe, GG, vanAlbada



Pressure Coeff.



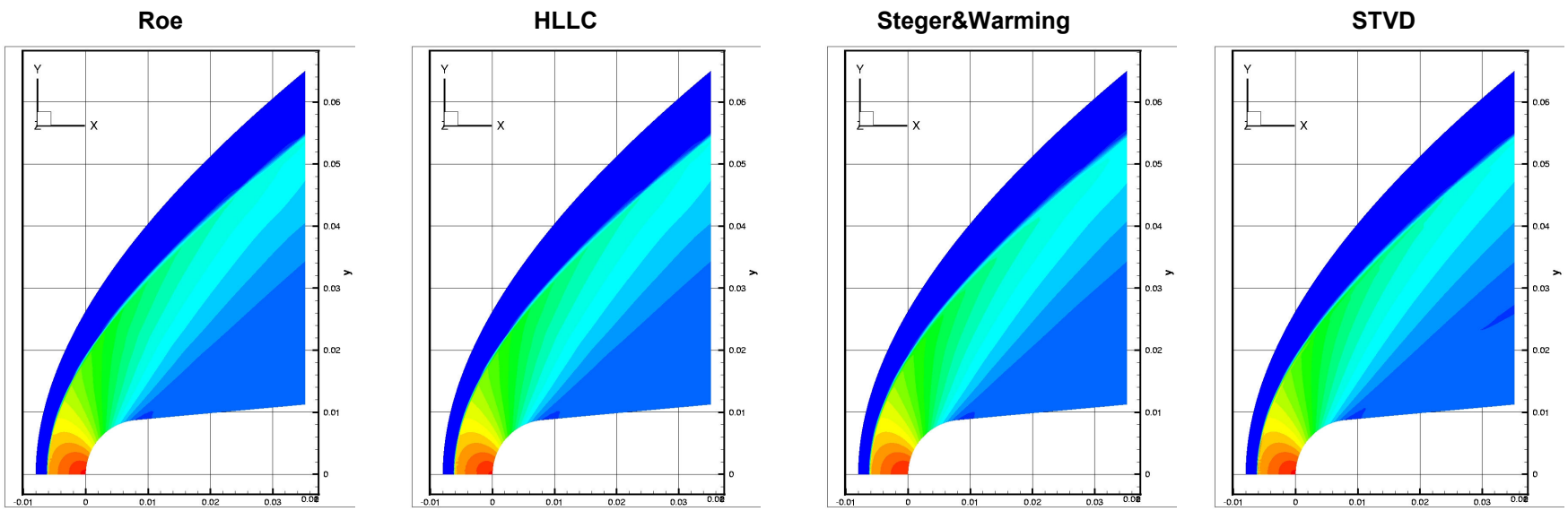
Mach No. Contours



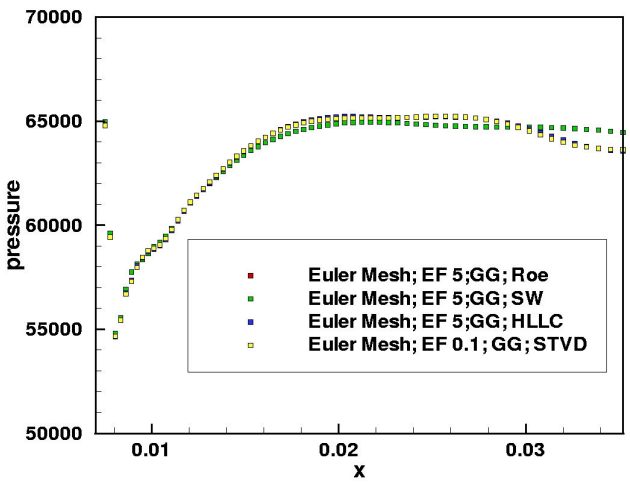
Cook, P.H., M.A. McDonald, M.C.P. Firmin, "Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements,"*Experimental Data Base for ComputerProgram Assessment*, AGARD Report AR 138,1979.

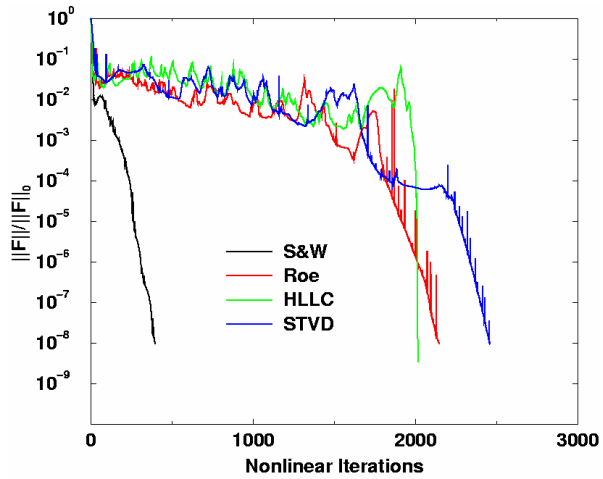
http://www.grc.nasa.gov/WWW/wind/valid/raetaf/raetaf01/raetaf01.html

Supersonic Blunt Wedge



Pressure Contours, Mach 3.0

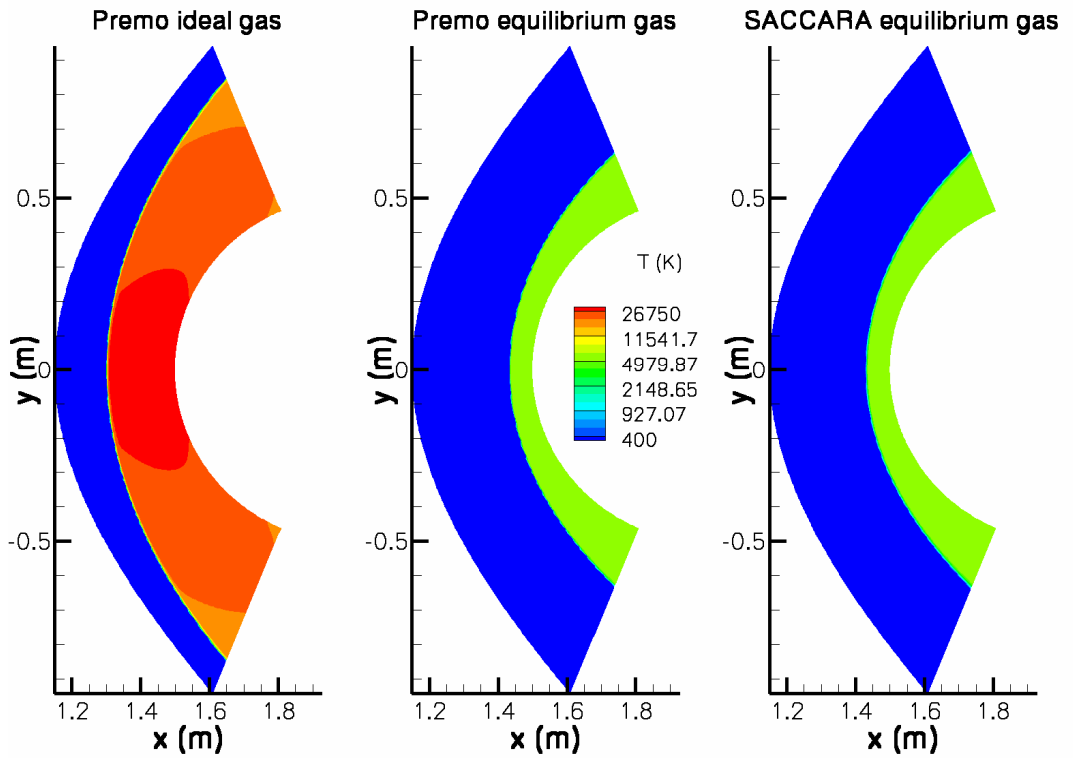




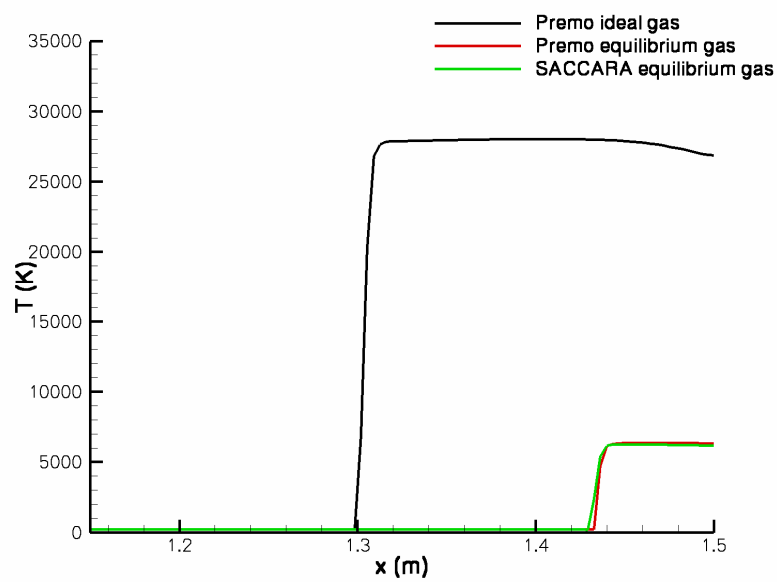
Sandia National Laboratories

Equilibrium Chemistry

(R.B. Bond and A.A. Lorber)



Temperature Contours, Mach 25



Stagnation T for Mach 25

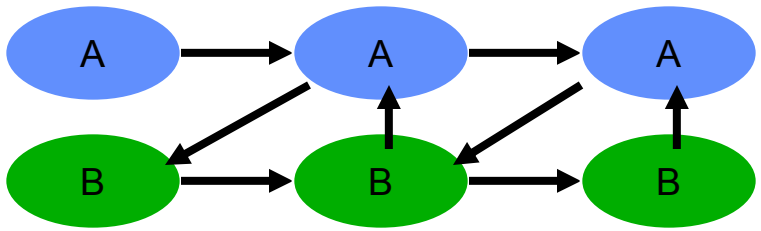
- Equilibrium gas properties determined from Liu and Vinokur
- Steger&Warming flux function $F(U) = \frac{\partial F}{\partial U} U + F'$
- 1st order upwind advection

Sandia National Laboratories

Multi-Physics Coupling

(R.W. Hooper)

Loose Coupling: Successive Substitution



Time,
NL Iters,
Real or Artificial parameter

Tight Coupling: Newton's Method

While not converged:

$$\mathbf{J}_n \Delta \mathbf{x} = -\mathbf{R}_n$$
$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta \mathbf{x}$$

$$\underbrace{\begin{bmatrix} J_{AA} & J_{AB} \\ J_{BA} & J_{BB} \end{bmatrix}}_{\tilde{\mathbf{J}} \quad \mathbf{M}} \begin{bmatrix} \Delta X_A \\ \Delta X_B \end{bmatrix} = \begin{bmatrix} -R_A \\ -R_B \end{bmatrix}$$

Intermediate Coupling: Jacobian-Free Newton-Krylov

1. Action of J on v: $\mathbf{J}\mathbf{v} \approx \dfrac{\mathbf{R}(\mathbf{x} + \delta \mathbf{v}) - \mathbf{R}(\mathbf{x})}{\delta}$

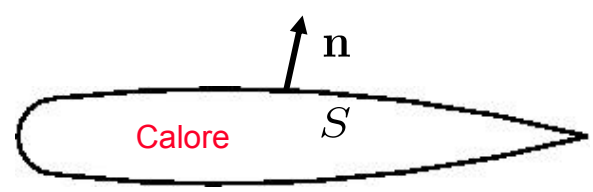
2. Convergence Acceleration (Preconditioning):

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{v} \approx \frac{\mathbf{R}(\mathbf{x} + \delta \mathbf{M}^{-1}\mathbf{v}) - \mathbf{R}(\mathbf{x})}{\delta}$$

Premo

$[T]_S = 0$

$[\mathbf{n} \cdot \nabla T]_S = 0$



- Co-spatial coupling
- Interfacial coupling
- Other

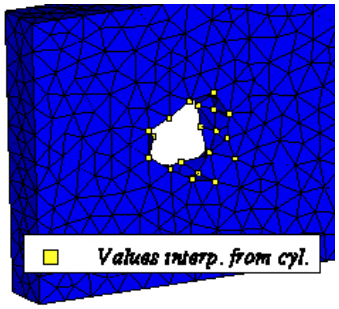
Sandia National Laboratories

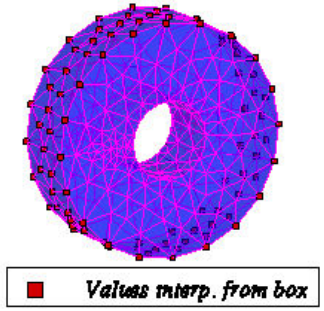
Towards Arbitrary Body Motion

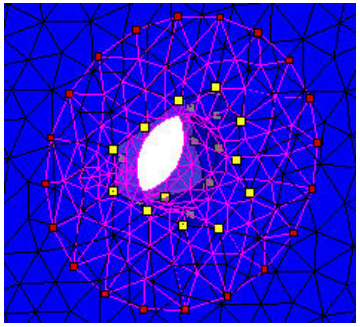
- **Rigid body motion** (C.C. Ober)
  - **NIRF with absolute velocities**
  - **Modified governing equations**
  - **Formulation supports 6 dof motion**
- **Euler Problem**
  - **26.565 degree cone**
  - **10 degree pitch**
  - **8 rev/second**
  - **Mach 2**

QuickTime™ and a
YUV420 codec decompressor
are needed to see this picture.

- **Relative Body Motion** (R. McLay & A.A. Lorber)
  - **Overset meshes**
  - **SUGGAR library determines holes and interpolation points**
  - **Framework element death defines new surfaces**
  - **Framework transfer objects compute interpolation values**







Summary

- Current Capabilities
  - Euler, NS, RANS solvers for subsonic to supersonic flows on unstructured meshes
  - Explicit and fully Implicit solvers
  - Active investigation/development of preconditioners (e.g., physics-based and ML)
- Emerging Capabilities
  - Heterogeneous mesh solver
  - Equilibrium chemistry for hypersonic high temperature inviscid flows
  - Multi-physics code coupling
    - Premo-Calore under development
    - Using Trilinos/NOX Library and Jacobian-Free approach
  - Arbitrary body motion
  - Relative body motion (overset meshes)
  - Mesh adaptivity
    - Uniform mesh refinement - interpolate I.C.s - solve
    - Demonstrated on hex mesh with hanging nodes and simple gradient based indicator
    - Non-hanging node currently available for tets (no special algorithms required)
    - Geometry fitting for resolving curved surfaces
  - K-ε RANS turbulence models
- Future plans
  - Semi-implicit solver
  - New Solver Strategies
    - Procedure sequencing (solve on a sequence of meshes)
    - Segregated physics solvers (RANS)
    - Adaptive solvers (semi-implicit/MFNK)
  - Non-equilibrium chemistry
  - Detached Eddy Simulation
  - Coupled fluid/structures interactions problems (aeroelasticity)

The End

Governing Equations

Conservative State Vector

$$\mathbf{W} = \left[\rho, \rho u, \rho v, \rho w, \rho E, \rho \hat{\upsilon}\right]^T$$

Primitive State Vector

$$\mathbf{U} = \left[\rho, u, v, w, p, \hat{\upsilon}\right]^T$$

Conservation Law form

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x_i} = \mathbf{H}$$

Apply Green's Theorem

$$\int_V \frac{\partial \mathbf{W}}{\partial t} dV + \oint_S \left(\mathbf{F}^c + \mathbf{F}^v\right) \bullet \mathbf{n} dS = \int_V \mathbf{H} dV$$

$$\mathbf{F}^c \mathbf{gn} = \begin{Bmatrix} (\rho u_i) n_i \\ (\rho u_1 u_i + p\delta_{i1}) n_i \\ (\rho u_2 u_i + p\delta_{i2}) n_i \\ (\rho u_3 u_i + p\delta_{i3}) n_i \\ ((\rho E + p) u_i) n_i \\ (\rho u_i \hat{\upsilon}) n_i \end{Bmatrix} \qquad \mathbf{F}^v \mathbf{gn} = \begin{Bmatrix} 0 \\ \tau_{i1} n_i \\ \tau_{i2} n_i \\ \tau_{i3} n_i \\ (-q_i + u_j \tau_{ij}) n_i \\ \mu_{eff} \dfrac{\partial \hat{\upsilon}}{\partial x_i} n_i \end{Bmatrix} \qquad \mathbf{H} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho\left(-C_{w1} f_w \left(\dfrac{\hat{\upsilon}}{d}\right)^2 + C_{b1} \hat{\upsilon} \hat{S} + \dfrac{C_{b2}}{\sigma}\left(\dfrac{\partial \hat{\upsilon}}{\partial x_j}\right)^2\right) \end{Bmatrix}$$

Sandia National Laboratories

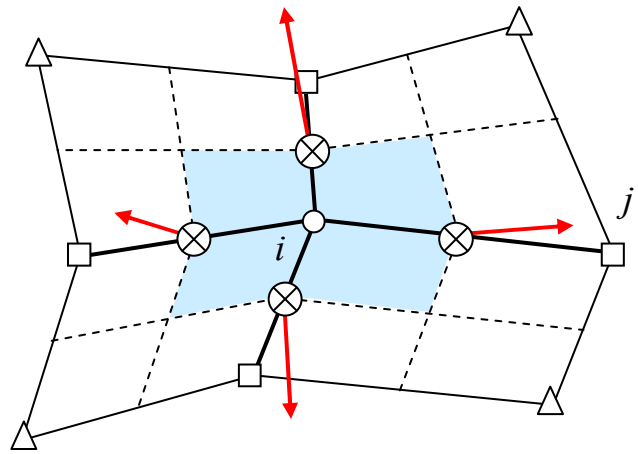
Node-Centered Finite-Volume Method

- **Define the dual of the finite element mesh**
  - Control sub-surface area summed to edges
  - Control sub-volumes summed to nodes
  - Surface normal is area weighted
  - External surfaces sub-divided into sub-surfaces

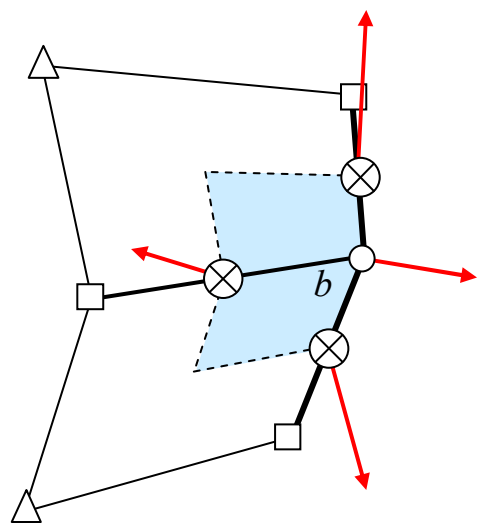
$$\Delta V_i = \sum_{k \in NSCV(i)} SCV_k$$

$$\mathbf{S}_{ij} = \sum_{k \in NSCS(edge(ij))} SCS_k^{\alpha}$$

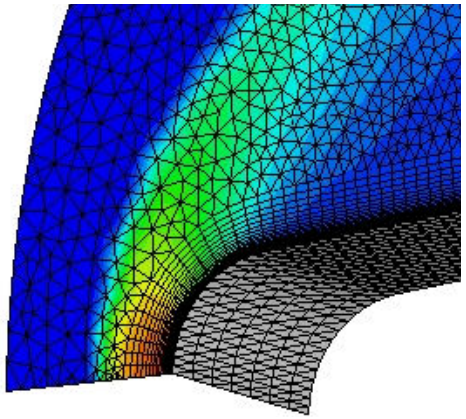
$$\mathbf{n}_{ij} = \mathbf{S}_{ij} \Big/ \left| \mathbf{S}_{ij} \right|$$



- **Edge based data structure**
- **Assemble by looping over edges and nodes and external faces**
- **Mesh transparency**



Heterogeneous meshes

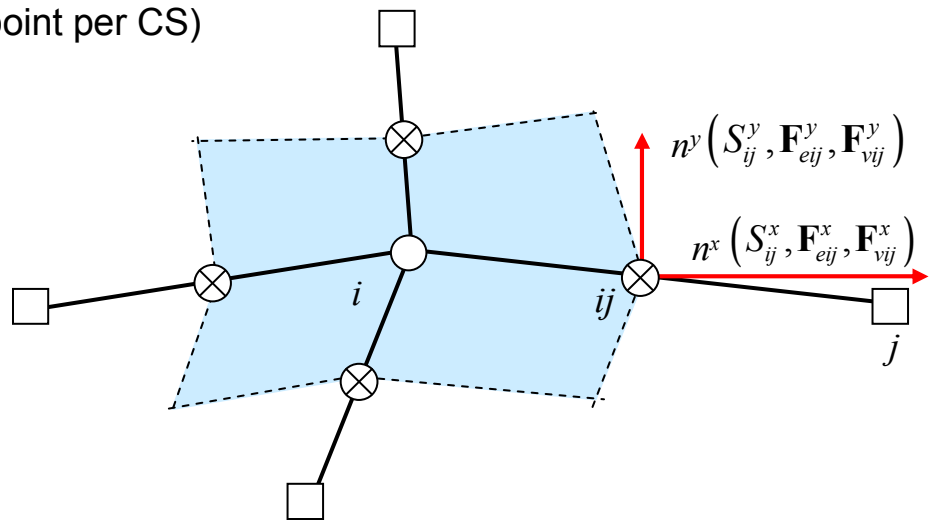


Sandia National Laboratories

Discrete Volume and Surface Integration

- **Surface integration**
  - Flux quadrature by midpoint rule (one integration point per CS)
  - Assume integration point lies on the edge midpoint

$$\int_S \mathbf{F}\mathbf{g}\mathbf{n}\,dS \approx \sum_{j\in NE(i)} \mathbf{F}\mathbf{g}\mathbf{n}_{ij}\left|\mathbf{S}_{ij}\right|$$



- **Volume integration**

$$\bar{\mathbf{W}}_i \Delta V_i = \int_{V_i} \mathbf{W}\,dV$$

- **Semi-discrete equation**

$$\frac{\partial \bar{\mathbf{W}}_i}{\partial t}\Delta V_i + \sum_{j\in NE(i)} \mathbf{F}_{ij}\,\mathbf{g}\mathbf{n}_{ij}\left|\mathbf{S}_{ij}\right| = \mathbf{H}_i \Delta V_i, \quad \forall i \in \Omega(V)$$

$$\frac{\partial \bar{\mathbf{W}}_i}{\partial t}\Delta V_i + \sum_{j\in NE(i)} \mathbf{F}_{ij}\,\mathbf{g}\mathbf{n}_{ij}\left|\mathbf{S}_{ij}\right| + \mathbf{F}_i\,\mathbf{g}\mathbf{n}_i\left|\mathbf{S}_i\right| = \mathbf{H}_i \Delta V_i, \quad \forall i \in \Gamma(S)$$

- **Local conservation (local flux balance)**
  - Lax-Wendroff Theorm: Given a consistent and conservative method, converging on a sequence of meshes will produce a weak solution

$$\mathbf{F}_{ij}\cdot\mathbf{n}_{ij}\left|\mathbf{S}_{ij}\right| = -\mathbf{F}_{ji}\cdot\mathbf{n}_{ji}\left|\mathbf{S}_{ji}\right|$$

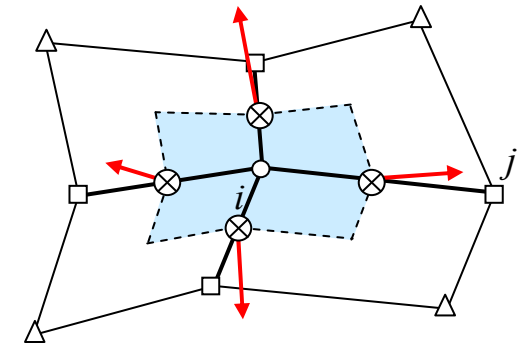
Sandia National Laboratories

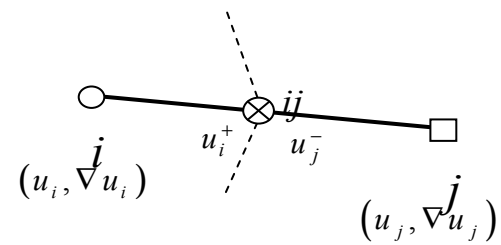
Edge-Based Assembly

- **Spatial Discretization**
  - Node centered finite-volume method
  - Median dual mesh: volumes stored on nodes, area vector stored on edges
  - 1st and 2nd order upwind advection
- **Second-order extrapolation (MUSCL, van Leer)**

$$u_i^+ = u_i + \frac{1}{2}\Phi(\Delta_i^+, \Delta_i^-)\nabla u_i \bullet \Delta\mathbf{r}_{ij} \qquad \Delta\mathbf{r}_{ij} = \left(\mathbf{r}_j - \mathbf{r}_i\right)$$

$$u_j^- = u_j - \frac{1}{2}\Phi(\Delta_j^+, \Delta_j^-)\nabla u_j \bullet \Delta\mathbf{r}_{ij} \qquad \Phi_i \in [0,1]$$





- **Residual and matrix assembly**
  - Gradients are pre-assembled and stored on nodes
  - Limiters are computed on edges

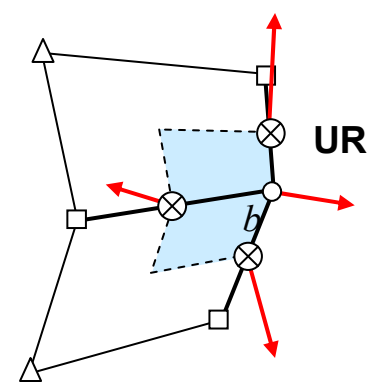
$$\nabla u_i = \frac{1}{\Delta V_i}\sum_{j \in NE(i)} \frac{1}{2}\left(u_i + u_j\right)\mathbf{n}_{ij}\left|\mathbf{S}_{ij}\right|$$

$$\Phi(a,b) = \frac{ab + |ab| + \varepsilon}{a^2 + b^2 + \varepsilon}$$

$$\Delta_i^+ = \Delta_j^- = u_j - u_i$$

$$\Delta_i^- = 2\nabla u_i \bullet \Delta\mathbf{r}_{ij} - (u_j - u_i)$$

$$\Delta_j^+ = 2\nabla u_j \bullet \Delta\mathbf{r}_{ij} - (u_j - u_i)$$



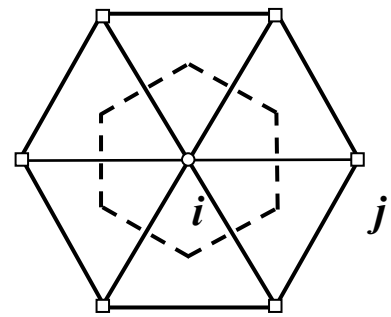
Sandia National Laboratories

Gradient Reconstruction

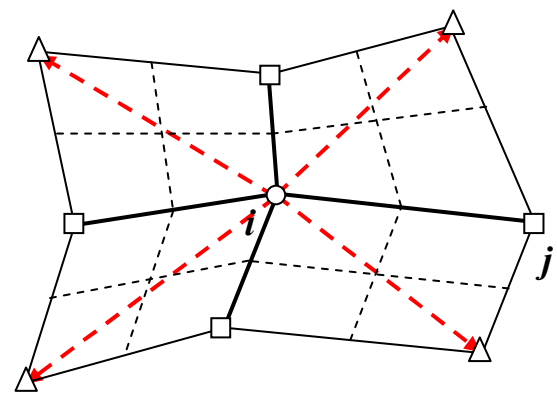
- **Green-Gauss Integral (Barth & Jesperson)**
  - Exact gradients for linear functions on tets (trapezoid rule)
  - Approx. on hexs (midpoint rule)

$$\nabla u_i = \frac{1}{\Delta V_i}\int_{S_i} u\mathbf{n}\,dS \approx \frac{1}{\Delta V_i}\sum_{j\in NE(i)}\frac{1}{2}\left(u_i+u_j\right)\mathbf{S}_{ij}$$

$$\nabla u_i = \frac{1}{\Delta V_i}\int_{S_i} u\mathbf{n}\,dS \approx \frac{1}{\Delta V_i}\sum_{j\in NE(i)}\frac{1}{2}\left(u_i+u_j\right)\mathbf{n}_{ij}\left|\mathbf{S}_{ij}\right| + u_i\mathbf{n}^b\left|\mathbf{S}_i\right|$$



- **Least-Squares (Haselbacher & Blazek)**
  - Exact derivatives for linear functions
  - Store six weights at each node
  - Requires no special treatment at boundaries
  - Requires virtual edges for hexs
  - Un-weighted form: $u_j = u_i + \nabla u_i \cdot \Delta\mathbf{r}_{ij}$
  - Inverse weighting $\xi_{ij} = \left|\Delta\mathbf{r}_{ij}\right|^{-p}$



$$\begin{bmatrix} \xi_1\Delta x_1 & \xi_1\Delta y_1 & \xi_1\Delta z_1 \\ \xi_2\Delta x_2 & \xi_2\Delta y_2 & \xi_2\Delta z_2 \\ \text{M} & \text{M} & \text{M} \\ \xi_{NE(i)-1}\Delta x_{NE(i)-1} & \xi_{NE(i)-1}\Delta y_{NE(i)-1} & \xi_{NE(i)-1}\Delta z_{NE(i)-1} \\ \xi_{NE(i)}\Delta x_{NE(i)} & \xi_{NE(i)}\Delta y_{NE(i)} & \xi_{NE(i)}\Delta z_{NE(i)} \end{bmatrix} \begin{Bmatrix} (u_x)_i \\ (u_y)_i \\ (u_z)_i \end{Bmatrix} = \begin{Bmatrix} \xi_1(u_1-u_i) \\ \xi_2(u_2-u_i) \\ \text{M} \\ \xi_{NE(i)-1}(u_{NE(i)-1}-u_i) \\ \xi_{NE(i)}(u_{NE(i)}-u_i) \end{Bmatrix}$$

$$(u_x)_i = \sum_{j\in NE(i)} W_{ij}^x\left(u_j-u_i\right)$$

$$(u_y)_i = \sum_{j\in NE(i)} W_{ij}^y\left(u_j-u_i\right)$$

$$(u_z)_i = \sum_{j\in NE(i)} W_{ij}^z\left(u_j-u_i\right)$$

Sandia National Laboratories

Explicit Time Integration

- **Explicit Time Integration**
  - **Steady-state and transient**
  - **High-order accuracy possible, time step limited by CFL**
  - **Good way to test residual calculations, time stepping, BC's**

$$\frac{\partial \bar{\mathbf{W}}_i}{\partial t} = -\frac{1}{\Delta V_i} \sum_{j \in NE(i)} \mathbf{F}_{ij} \cdot \mathbf{n}_{ij} \left| \mathbf{S}_{ij} \right| + \mathbf{H}_i, \quad \forall i \in \Omega(V)$$

$$\frac{\partial \bar{\mathbf{W}}_i}{\partial t} = -\frac{1}{\Delta V_i} \sum_{j \in NE(i)} \mathbf{F}_{ij} \cdot \mathbf{n}_{ij} \left| \mathbf{S}_{ij} \right| - \frac{1}{\Delta V_i} \mathbf{F}_i \cdot \mathbf{n}_i \left| \mathbf{S}_i \right| + \mathbf{H}_i, \quad \forall i \in \Gamma(S)$$

$$\frac{\partial \bar{\mathbf{W}}_i}{\partial t} = \mathbf{R}\left(\mathbf{W}^n\right)$$

- **CFL condition**

$$\Delta t_i = \frac{\Delta V_i}{\sum_{j \in NE(i)} \left(\left| \vec{u} \bullet \mathbf{n}_{ij} \right| + c \right) \left| \mathbf{S}_{ij} \right|}$$

$$\Delta t_0 = \min(\Delta t_i)$$

$$\mathbf{W}^{(0)} = \mathbf{W}^{(n)}$$

$$\mathbf{W}^{(1)} = \mathbf{W}^{(0)} + \alpha_1 \frac{\Delta t}{\Delta V} \mathbf{R}\left(\mathbf{W}^{(0)}\right)$$

$$\mathbf{W}^{(2)} = \mathbf{W}^{(0)} + \alpha_2 \frac{\Delta t}{\Delta V} \mathbf{R}\left(\mathbf{W}^{(1)}\right)$$

$$\mathbf{W}^{(3)} = \mathbf{W}^{(0)} + \alpha_3 \frac{\Delta t}{\Delta V} \mathbf{R}\left(\mathbf{W}^{(2)}\right)$$

$$\mathbf{W}^{(4)} = \mathbf{W}^{(0)} + \alpha_4 \frac{\Delta t}{\Delta V} \mathbf{R}\left(\mathbf{W}^{(3)}\right)$$

$$\mathbf{W}^{(n+1)} = \mathbf{W}^{(4)}$$

$$\alpha_1 = 1/4, \ \alpha_2 = 1/3, \ \alpha_3 = 1/2, \ \alpha_4 = 1.0$$

Sandia National Laboratories

Gradient Limiters

- **Barth-Jespersen (BJ)**
  - Ensures than reconstruction does not introduce new maxima or minima
  - Satisfies discrete maximum principle
  - Non-differentiable
  - Can cause convergence to stall
  - Enables good shock capturing

$$u_j^{\min} \le u_i(\mathbf{x}) \le u_j^{\max}, \ \forall j \in NE(i)$$

$$u_i^{\max} = \max_{j \in NE(i)} (u_i, u_j)$$

$$u_i^{\min} = \min_{j \in NE(i)} (u_i, u_j)$$

- **Venkatakrishnan (Venkat)**
  - Differentiable
  - Has better convergence properties than BJ limiter
  - One adjustable constant
  - Does not satisfy discrete maximum principle
  - Oscillations are admitted
  - Discontinuities are smeared
  - Replace *min(1,y)* with *ψ(y)*

$$\phi_{ij}^- = \begin{cases} \min\left(1, \dfrac{u_i^{\max} - u_i}{u_{ij}^- - u_i}\right), & \text{if } u_{ij}^- - u_i > 0 \\ \min\left(1, \dfrac{u_i^{\min} - u_i}{u_{ij}^- - u_i}\right), & \text{if } u_{ij}^- - u_i < 0 \\ 1, & \text{if } u_{ij}^- - u_i = 0 \end{cases}$$

$$\Phi_i = \min_{j \in NE(i)} \left(\phi_{ij}^-\right)$$

$$\psi(y) = \frac{y^2 + 2y}{y^2 + y + 2}$$

Sandia National Laboratories

Boundary Condition Enforcement

- **Weak enforcement** (Add flux from boundary surface)
  - **Performed during other flux calculations**
  - **Similar to interior flux contributions**
  - **Add contributions to residual equations**

$$\frac{\partial \mathbf{W}_b}{\partial t}\Delta V_b + \sum_{j \in NE(b)} \left[\mathbf{F}_{ij}^c - \mathbf{F}_{ij}^v\right] \bullet \mathbf{n}_{ij} \left|\mathbf{S}_{ij}\right| + \mathbf{H}_b \Delta V_b + \mathbf{F}_b \bullet \mathbf{n}_b \left|\mathbf{S}_b\right| = 0, \quad \forall i \in \Omega(\Gamma)$$

- **Inviscid wall flux**

$$\mathbf{u} \bullet \mathbf{n} = 0 \qquad \mathbf{F}^{slip} \mathbf{n} = \begin{Bmatrix} 0 \\ pn_x \\ pn_y \\ pn_z \\ 0 \\ 0 \end{Bmatrix}$$

- **Viscous wall treatment**

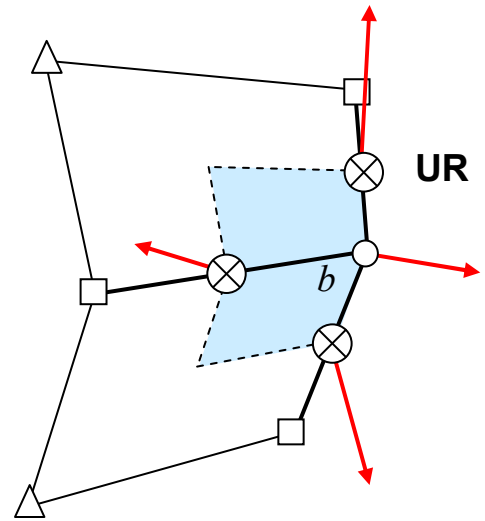
$$\mathbf{u} = 0, \; T = T_w \text{ or } \frac{\partial q}{\partial n} = \dot{Q}_w$$

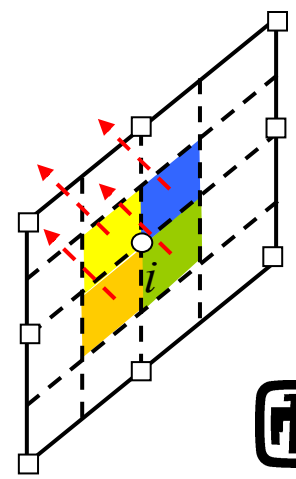
- **Residual enforcement**
  - **Performed after all other flux calculations**
  - **Replace flux balance with modified residual equation**

$$\mathbf{W}_{\rho u} - \mathbf{W}_{\rho} u^b = 0$$

- **All boundary conditions are contained in the residual equations**

- **UR – constant**
- **UR – U(U^b,P_0,T_0,α)**





Sandia National Laboratories

Edge-Based Matrix Assembly: Residual

- Conservation Law form of Governing Equations

$$\int_V \frac{\partial \mathbf{W}}{\partial t} dV + \int_S \left(\mathbf{F}^c + \mathbf{F}^v\right) \bullet \mathbf{n} dS = \int_V \mathbf{H} dV$$

- Semi-Discrete Equations (Residuals)

$$\mathbf{G}_i = \frac{\partial \mathbf{W}_i}{\partial t} \Delta V_i + \sum_{j \in NE(i)} \left[\mathbf{F}_{ij}^c - \mathbf{F}_{ij}^v\right] \left|\mathbf{S}_{ij}\right| + \mathbf{H}_i = 0, \quad \forall i \in \Omega(V)$$

$$\mathbf{W}_i \Delta V_i \approx \int_{V_i} \mathbf{W} dV$$

- Roe's Approximate Riemann flux function

$$\mathbf{F}_{ij}^c = \frac{1}{2}\left[\mathbf{F}\left(\mathbf{W}_i^+, \mathbf{n}_{ij}\right) + \mathbf{F}\left(\mathbf{W}_j^-, \mathbf{n}_{ij}\right) - \left|\tilde{\mathbf{A}}\left(\mathbf{W}_i^+, \mathbf{W}_j^-, \mathbf{n}_{ij}\right)\right| \left(\mathbf{W}_j^- - \mathbf{W}_i^+\right)\right]$$

- Viscous Fluxes require construction of gradients at edge midpoints
- Premo blends node gradients with the node difference, (Rie and Chow, O'Rourke)

$$\mathbf{F}_{ij}^v = \mathbf{F}\left(\mathbf{U}_i, \nabla \mathbf{U}_i, \mathbf{U}_j, \nabla \mathbf{U}_j, \mathbf{n}_{ij}\right) \quad \nabla u_{ij} \approx \frac{\left(\nabla u_i + \nabla u_j\right)}{2} + \left(u_j - u_i - \frac{\left(\nabla u_i + \nabla u_j\right)}{2} \bullet \Delta \mathbf{r}_{ij}\right) \frac{\Delta \mathbf{r}_{ij}}{\left|\Delta \mathbf{r}_{ij}\right|^2}$$

Sandia National Laboratories

Transonic Turbulent Flow

	MPF	η	Timestep	f_A	CFL	$\hat{\upsilon}_0$	NLIts	Lits	CPU
1	1	10^{-3}	LTS	1.05	1-100	$2.7e^{-5}$	434	11,279	112,591
2	5	10^{-1}	LTS	1.025	10-500	$2.7e^{-5}$	1,085	9,392	38,983
3	5	10^{-2}	LTS	1.2	1-500	$2.5e^{-5}$	809	16,893	51,028
4	10	10^{-1}	LTS	1.2	1-500	$2.5e^{-5}$	928	9,467	32,619
5*	5	10^{-1}	LTS	1.2	1-500	$2.7e^{-5}$	586	2044	11,785
6*	5	10^{-2}	LTS	1.1	1-500	$1.0e^{-5}$	481	7451	20,436

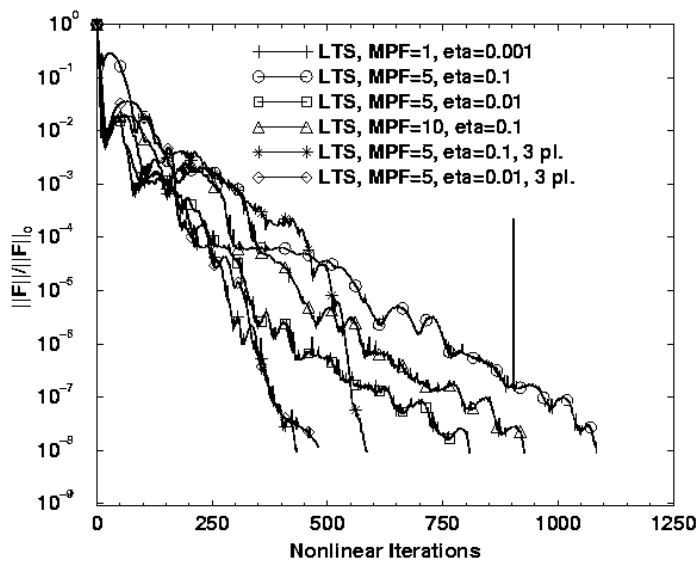
* 3 planes, 2 elements wide, Dirichlet symmetry planes, 3.6GHz machine, 6 uses Sutherlands Law

Flow conditions:
M=0.729
Re=6.5E6
Alpha=2.31

Mesh:
C-grid: 369x65x2
Boundaries ~20 chords
Away from airfoil

Solver Settings:
Ilu(1)/RCM

Spalart-Allmaras turbulence model



Previous Results
Flow Conditions:
Mach No. = 0.2
Re = 1.86E6
Alpha = 3.59

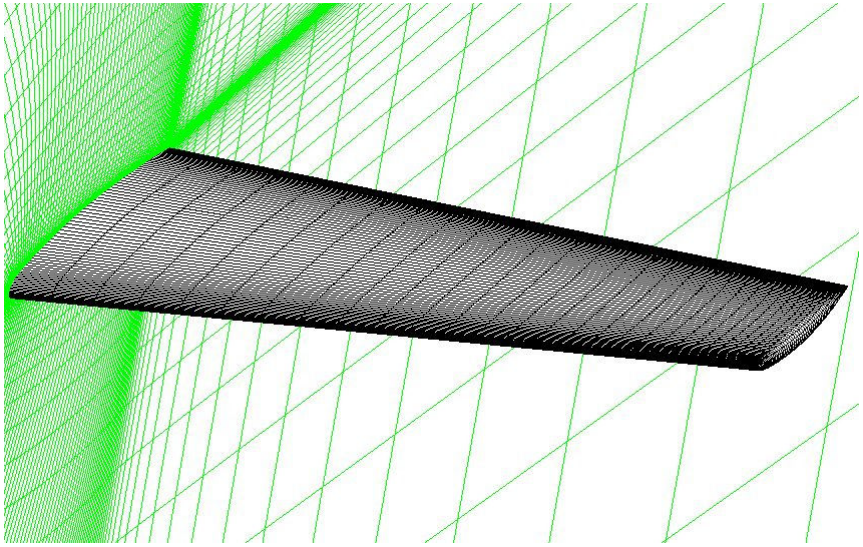
Mesh:
NACA0012 C-grid: 225x81x2

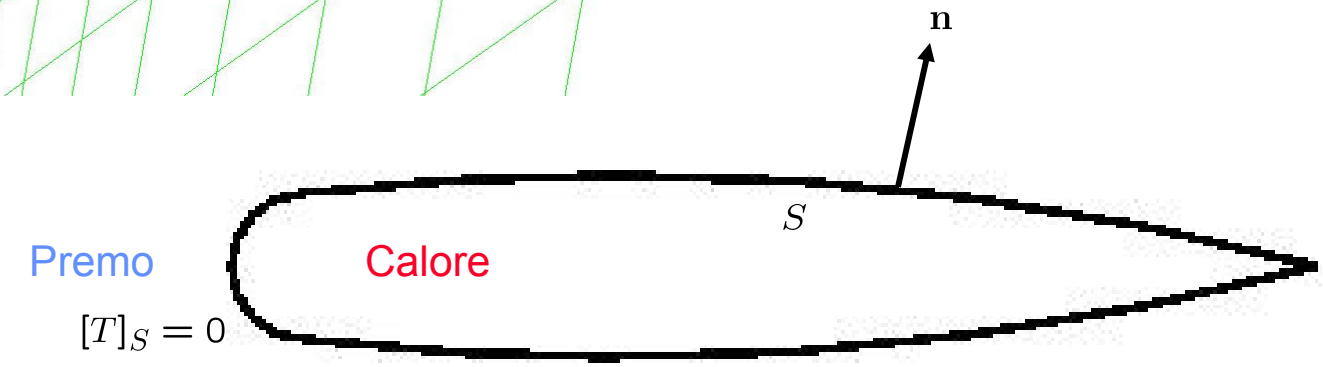
Discretization:
MF, FDC, MPF=10
Spalart-Allmaras

Convergence: 10^{-6}
CPU: 63,419

Sandia National Laboratories

Current Work : Premo-Calore Coupling





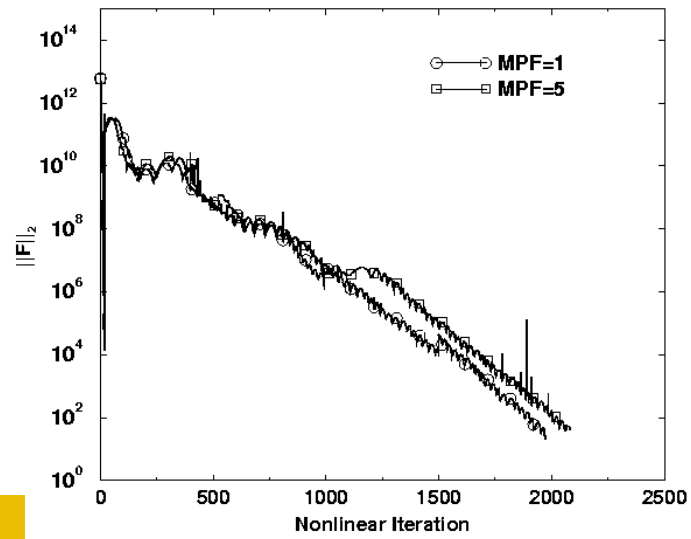
$[T]_S = 0$

$[\mathbf{n} \cdot \nabla T]_S = 0$

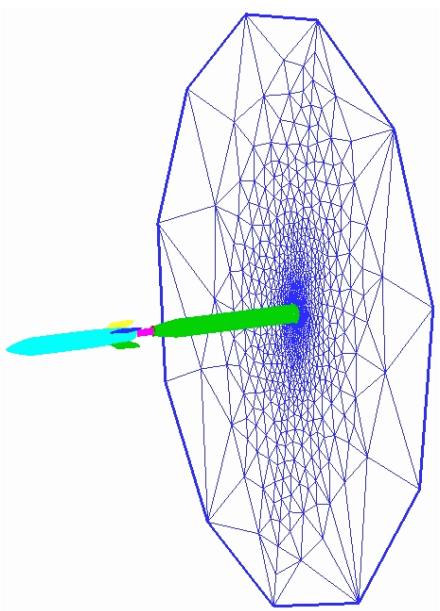
- Co-spatial coupling
- Interfacial coupling
- Other

Unstructured Mesh Parallel Solver

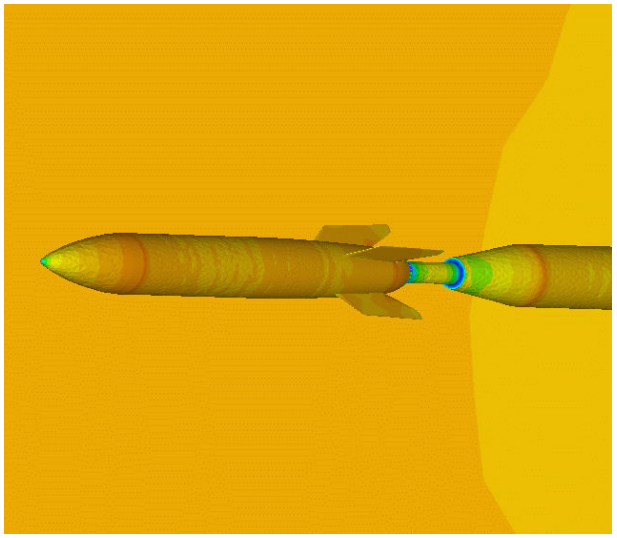
Flow Conditions:
M=0.8, alpha=0
Mesh: 174,383 nodes, 916,009 Tet elements
Solver Settings:
CFLmax=500
Ilu(1)/RCM, overlap=1, $\eta=10^{-1}$
Discretization: LS, van Albada, AD
Platform: 8 processors on Rogue



MPF	CPU (hrs.)
1	40
5	24



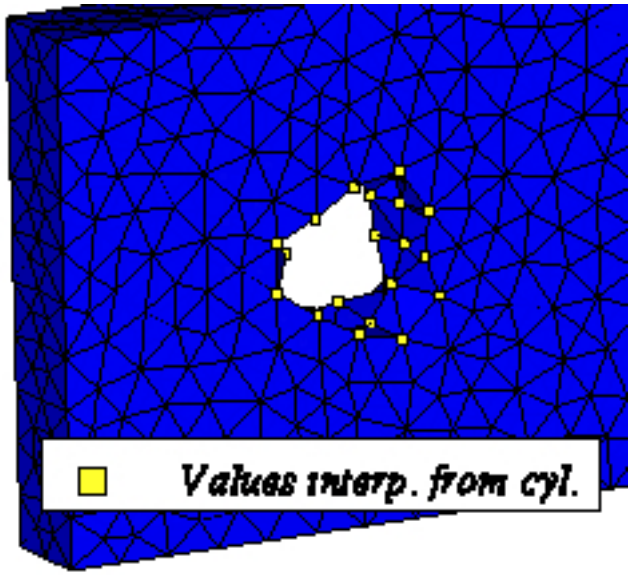
Mesh courtesy of J.L. Payne

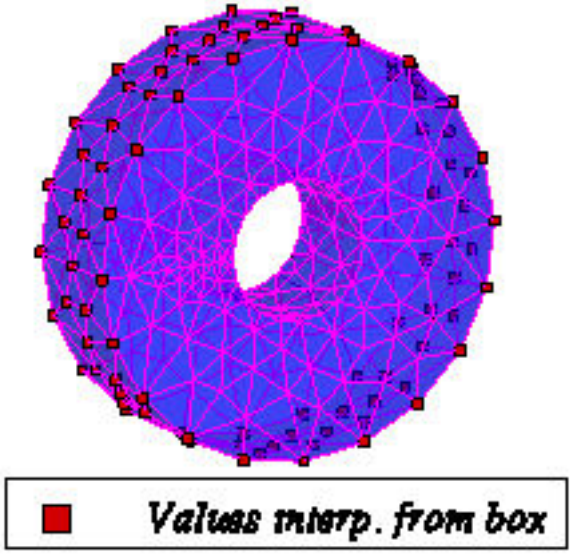
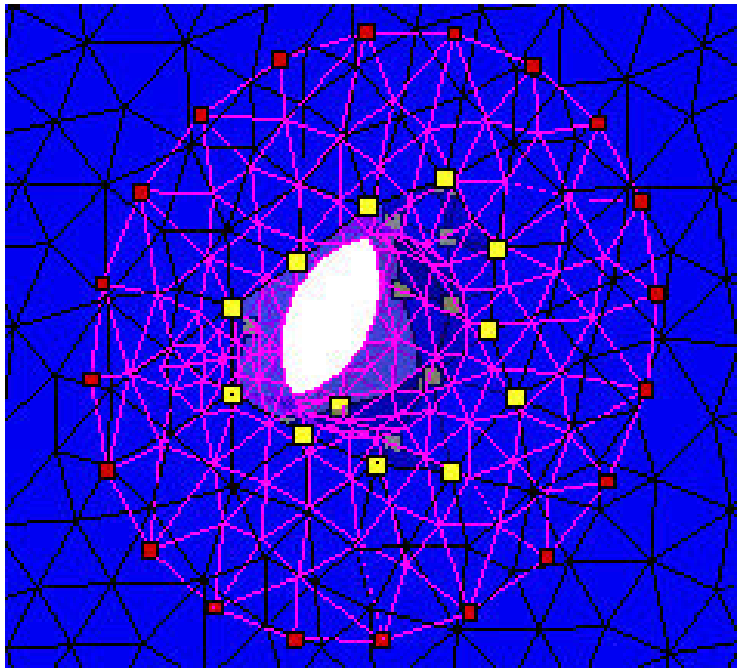


Mach No. Contours

Overset Meshes

(A.A. Lorber and R. McLay)





- Coarse Background Mesh
- Complex Geometry Mesh
- SUGGAR determines holes and interp. points.
- Element Death defines new surfaces
- Transfer Objects computes interp. values