Detonation and hypersonics simulation with AMROC – Part II

Ralf Deiterding

Aerodynamics and Flight Mechanics Research Group
University of Southampton
Highfield Campus
Southampton SO17 1BJ, UK
Email: r.deiterding@soton.ac.uk

Xiamen
23rd July, 2019

Outline

Thermodynamic Model

The two temperature thermodynamic model is used to model the thermodynamic nonequilibrium,

Thermodynamic Model

The two temperature thermodynamic model is used to model the thermodynamic nonequilibrium,

$$e_s(T_{tr}, T_{ve}) = e_s^t(T_{tr}) + e_s^r(T_{tr}) + e_s^v(T_{ve}) + e_s^{el}(T_{ve}) + e_s^0$$

Thermodynamic Model

The two temperature thermodynamic model is used to model the thermodynamic nonequilibrium,

$$e_s(T_{tr}, T_{ve}) = e_s^t(T_{tr}) + e_s^r(T_{tr}) + e_s^v(T_{ve}) + e_s^{el}(T_{ve}) + e_s^0$$

- Computationally efficient,
- Widely used,
- Integrated into the open source library Mutation++ [Scoggins and Magin, 2014].

Thermodynamic Model

The two temperature thermodynamic model is used to model the thermodynamic nonequilibrium,

$$e_s(T_{tr}, T_{ve}) = e_s^t(T_{tr}) + e_s^r(T_{tr}) + e_s^v(T_{ve}) + e_s^{el}(T_{ve}) + e_s^0$$

- Computationally efficient,
- Widely used,
- Integrated into the open source library Mutation++ [Scoggins and Magin, 2014].

The internal energies are calculated within the Mutation++ library using the Rigid-Rotator Harmonic-Oscillator (RRHO) model.

Governing Equations

The two temperature thermodynamic model has been implemented using the equations,

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{W}$$

Governing Equations

The two temperature thermodynamic model has been implemented using the equations,

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{W}$$

where,

$$\mathbf{Q} = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_{N_s} \\ \rho u \\ \rho v \\ \rho e^{ve} \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho_1 u \\ \vdots \\ \rho_{N_s} u \\ \rho u^2 + p \\ \rho v u \\ \rho e^{ve} u \\ (\rho E + p) u \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho_1 v \\ \vdots \\ \rho_{N_s} v \\ \rho u v \\ \rho v^2 + p \\ \rho e^{ve} v \\ (\rho E + p) v \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \dot{w}_1 \\ \vdots \\ \dot{w}_{N_s} \\ 0 \\ 0 \\ Q_{ve} \\ 0 \end{bmatrix}.$$

Source Terms

The net species production rates,

$$\dot{w}_s = M_s \sum_{r=1}^{N_r} (\beta_{sr} - \alpha_{sr}) \left[k_{f,r} \prod_{i=1}^{N_s} \left(\frac{\rho_i}{M_i} \right)^{\alpha_{ir}} - k_{b,r} \prod_{i=1}^{N_s} \left(\frac{\rho_i}{M_i} \right)^{\beta_{ir}} \right] ,$$

$$k_{f,r}(T_c) = A_{f,r} T_c^{\eta_{f,r}} \exp\left[-\theta_r / T_c \right] ,$$

Source Terms

The net species production rates,

$$\dot{w}_s = M_s \sum_{r=1}^{N_r} (\beta_{sr} - \alpha_{sr}) \left[k_{f,r} \prod_{i=1}^{N_s} \left(\frac{\rho_i}{M_i} \right)^{\alpha_{ir}} - k_{b,r} \prod_{i=1}^{N_s} \left(\frac{\rho_i}{M_i} \right)^{\beta_{ir}} \right] ,$$

$$k_{f,r}(T_c) = A_{f,r} T_c^{\eta_{f,r}} \exp\left[-\theta_r / T_c \right] ,$$

and the energy transfer rate (neutral mixture),

$$Q_{ve} = \sum_s Q_s^{T-V} + Q_s^{C-V} + Q_s^{C-el} ,$$

$$Q_s^{T-V} = \rho_s \frac{e_s^v(T_{tr}) - e_s^v}{\tau_{v,s}^{T-V}} ,$$

$$Q_s^{C-V} = c_1 \dot{w}_s e_s^v, \quad Q_s^{C-el} = c_1 \dot{w}_s e_s^{el} ,$$

are both calculated using the Mutation++ library.

Numerical Integration

Finite volume method with two flux schemes implemented,

Numerical Integration

Finite volume method with two flux schemes implemented,

- Van Leer's flux vector splitting method [van Leer, 1982],
- The AUSM scheme [Liou and Steffen Jr, 1993].

Numerical Integration

Finite volume method with two flux schemes implemented,

- Van Leer's flux vector splitting method [van Leer, 1982],
- The AUSM scheme [Liou and Steffen Jr, 1993].

Second order in space and time,

- The MUSCL-Hancock scheme is used for the fluxes.
- Strang splitting is used to integrate the source term.

Double Wedge

Simulation of a double wedge in a high enthalpy flow of air [Pezzella et al., 2015].

T_∞	p_∞	U_∞	M_∞	L_1	θ_1	L_2	θ_2
$710\,\mathrm{K}$	$0.78\,\mathrm{kPa}$	$3812\,\mathrm{m/s}$	7.14	$50.8\,\mathrm{mm}$	30°	$25.4\,\mathrm{mm}$	55°

Table: Double wedge geometry and experimental conditions.

Double Wedge

Simulation of a double wedge in a high enthalpy flow of air [Pezzella et al., 2015].

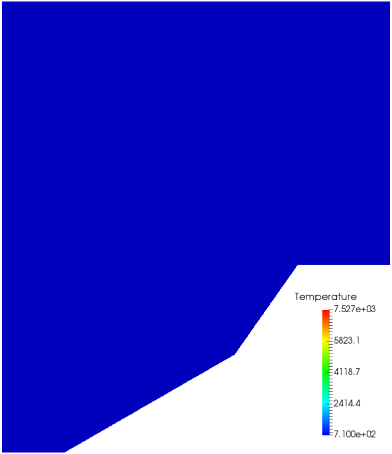
T_∞	p_∞	U_∞	M_∞	L_1	θ_1	L_2	θ_2
$710\,\mathrm{K}$	$0.78\,\mathrm{kPa}$	$3812\,\mathrm{m/s}$	7.14	$50.8\,\mathrm{mm}$	30°	$25.4\,\mathrm{mm}$	55°

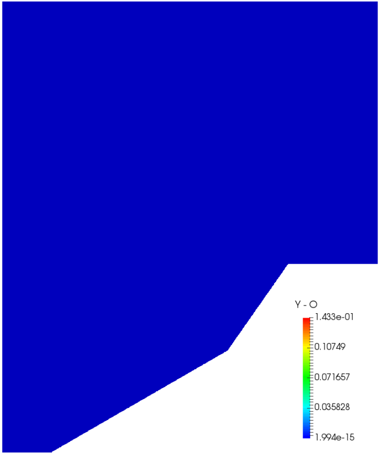
Table: Double wedge geometry and experimental conditions.

- Five species mixture of air.
- Initial 200×200 cell mesh, with 3 levels of refinement.
- Embedded boundary used to define geometry.
- Van Leer flux scheme.
- Physical time of $242\,\mu\mathrm{s}$.

Double Wedge

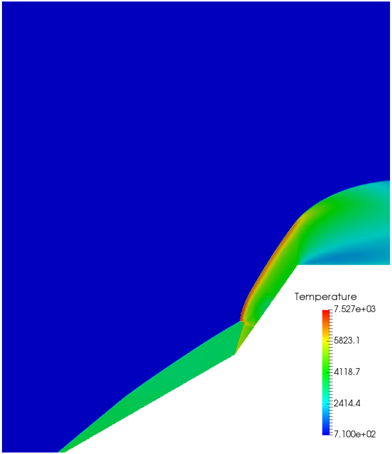
The temperature and mass fraction of atomic oxygen.

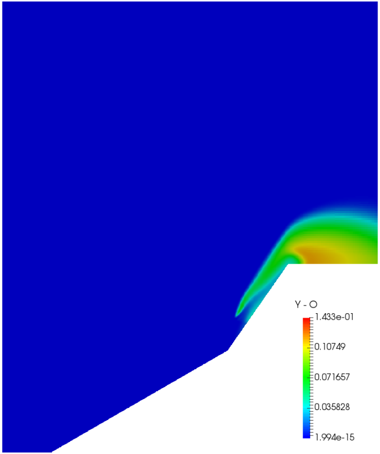




Double Wedge

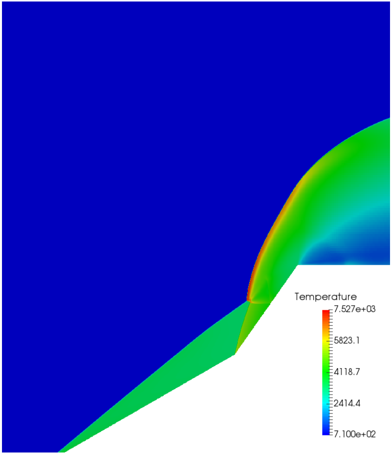
The temperature and mass fraction of atomic oxygen.

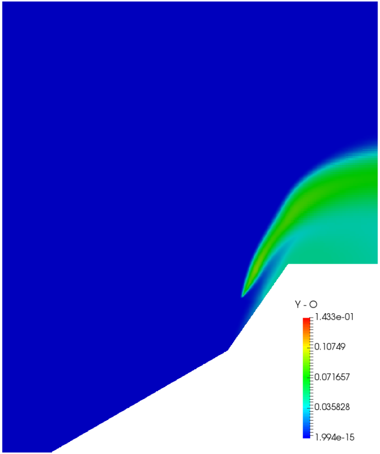




Double Wedge

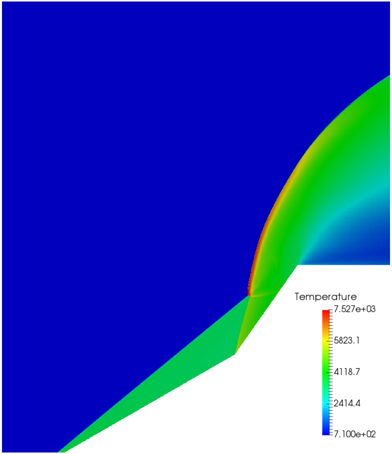
The temperature and mass fraction of atomic oxygen.

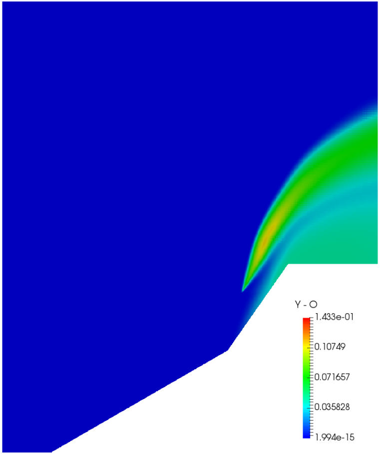




Double Wedge

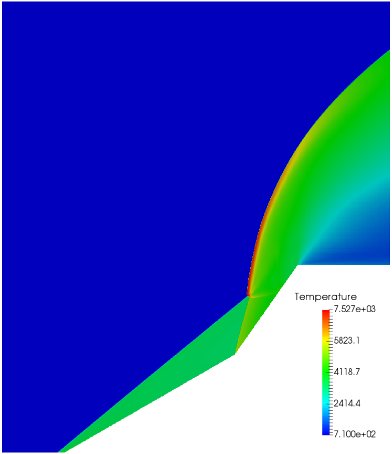
The temperature and mass fraction of atomic oxygen.

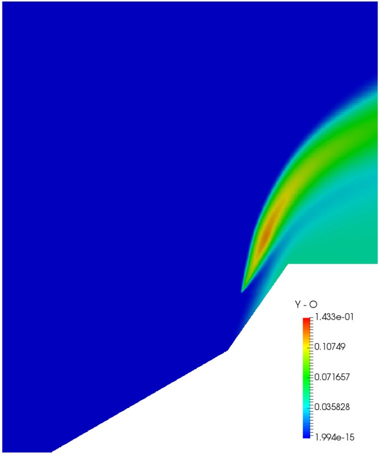




Double Wedge

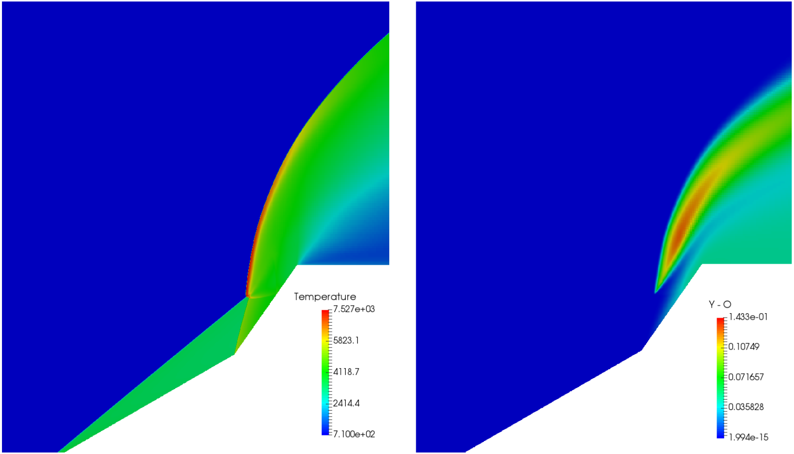
The temperature and mass fraction of atomic oxygen.





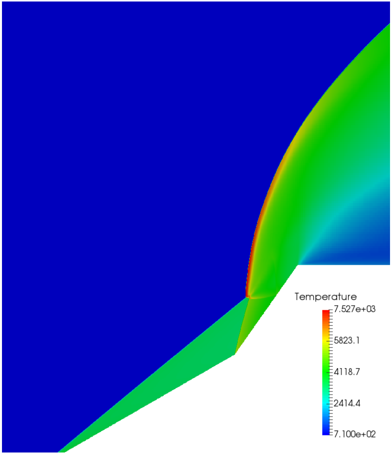
Double Wedge

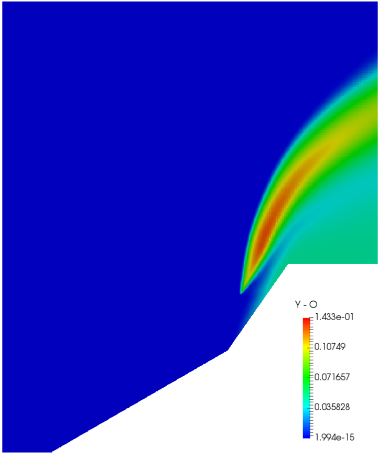
The temperature and mass fraction of atomic oxygen.



Double Wedge

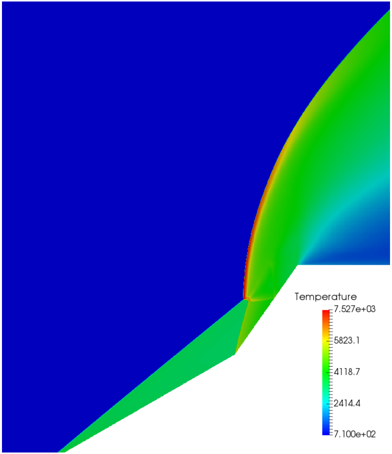
The temperature and mass fraction of atomic oxygen.

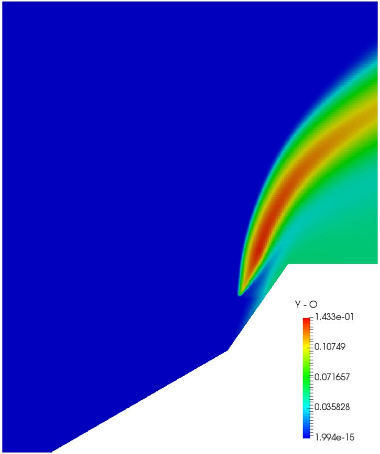




Double Wedge

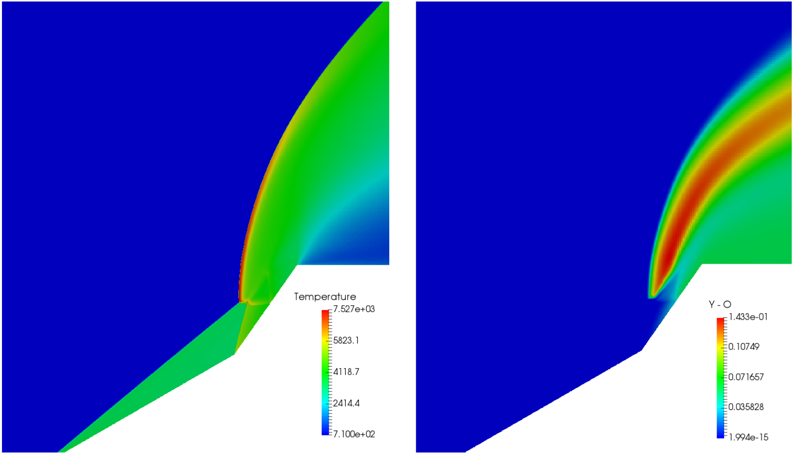
The temperature and mass fraction of atomic oxygen.





Double Wedge

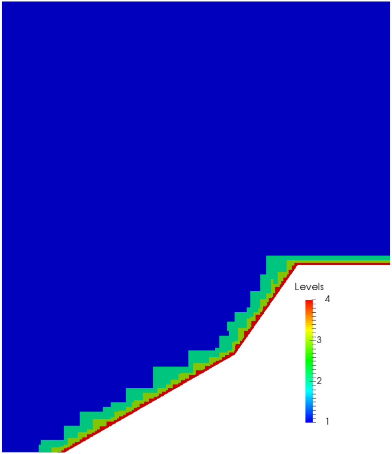
The temperature and mass fraction of atomic oxygen.

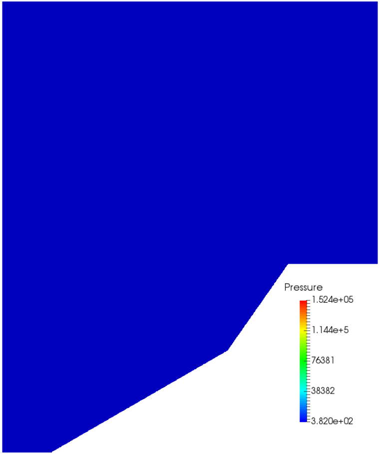


$t = 242\mu\mathrm{secs}.$

Double Wedge

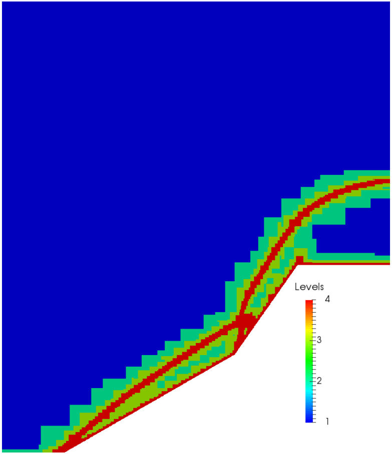
The mesh was refined using pressure and density gradients.

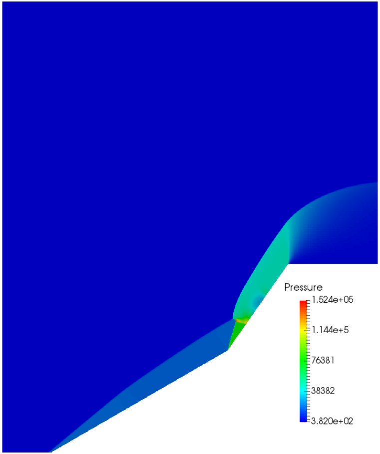




Double Wedge

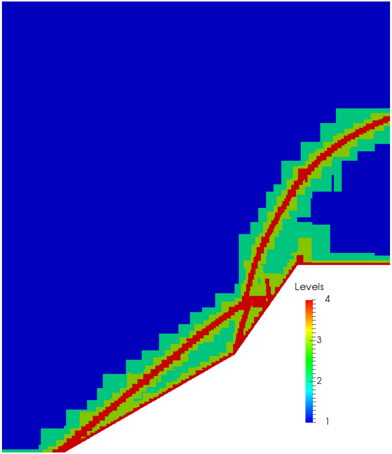
The mesh was refined using pressure and density gradients.

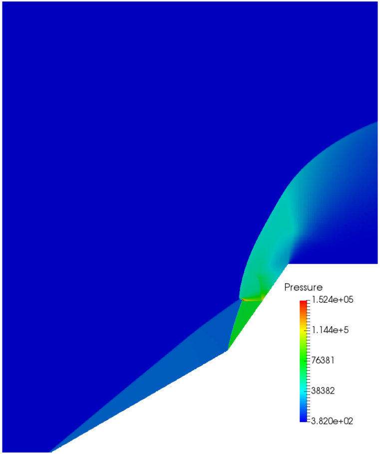




Double Wedge

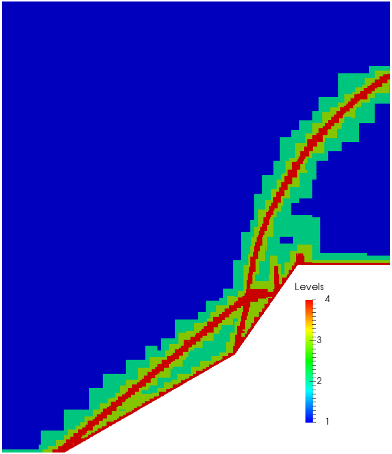
The mesh was refined using pressure and density gradients.

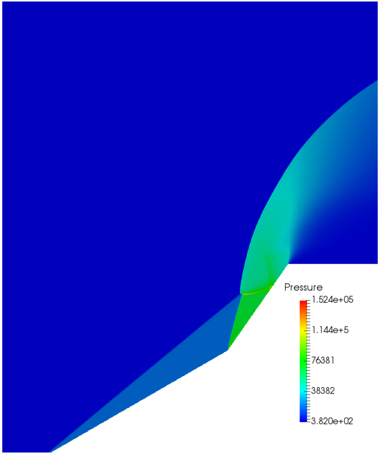




Double Wedge

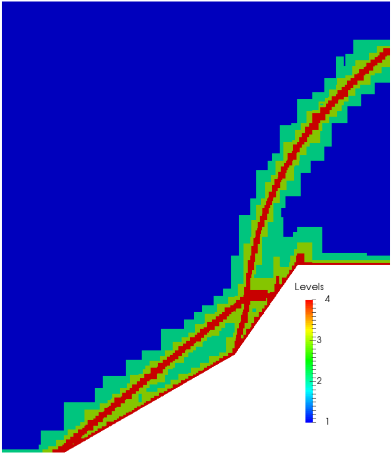
The mesh was refined using pressure and density gradients.

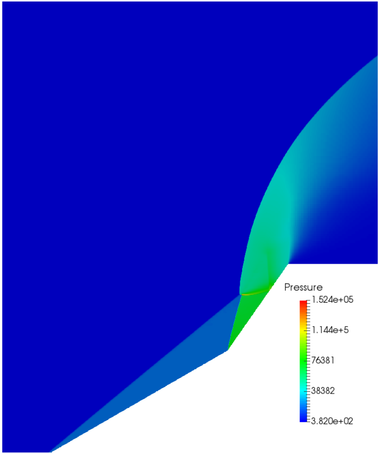




Double Wedge

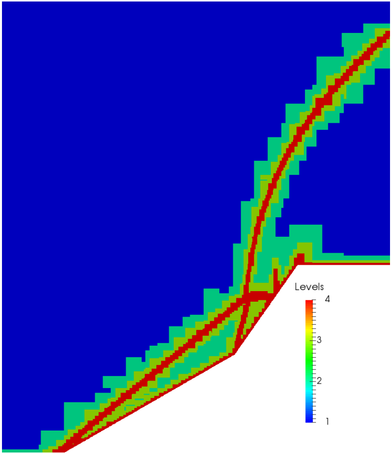
The mesh was refined using pressure and density gradients.

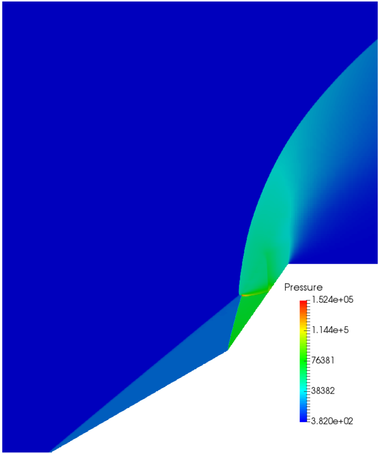




Double Wedge

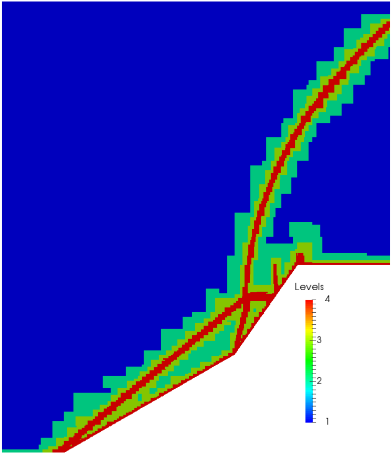
The mesh was refined using pressure and density gradients.

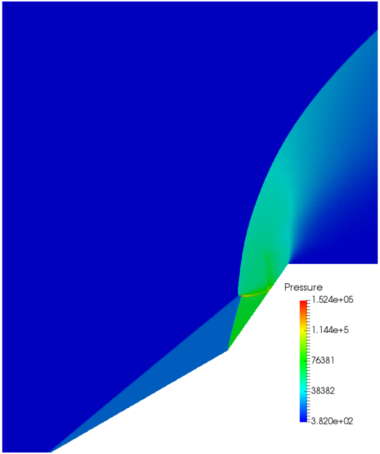




Double Wedge

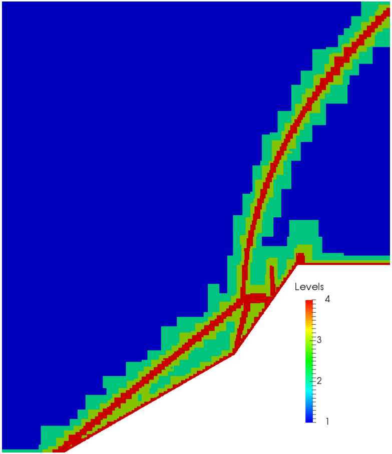
The mesh was refined using pressure and density gradients.

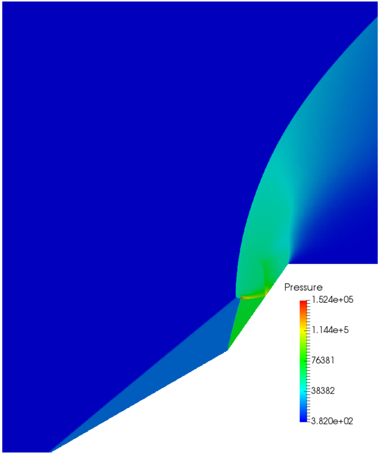




Double Wedge

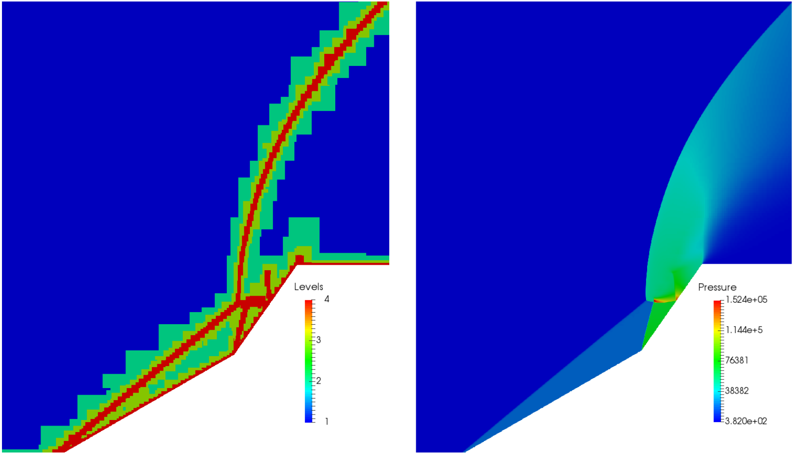
The mesh was refined using pressure and density gradients.





Double Wedge

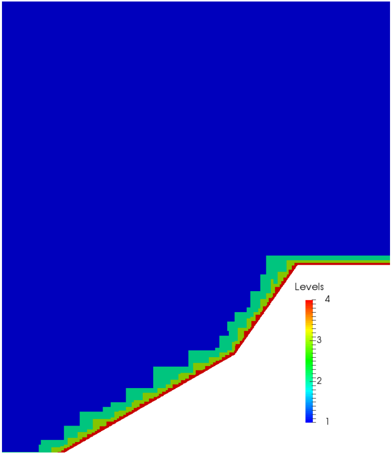
The mesh was refined using pressure and density gradients.

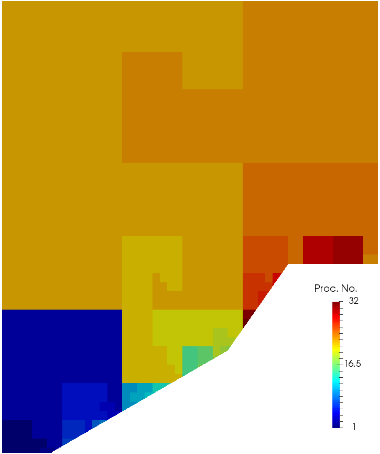


$t = 242\,\mu\text{secs}.$

Double Wedge

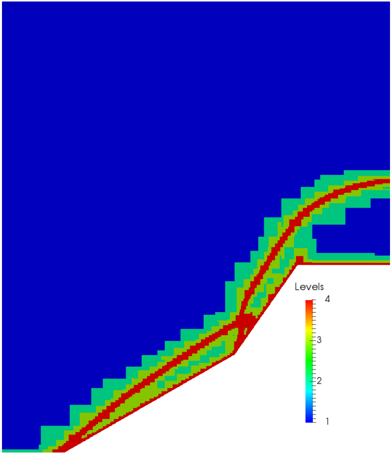
Dynamic load balancing distributes the cells across the processors.

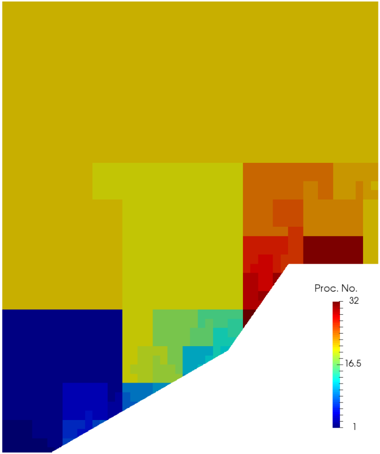




Double Wedge

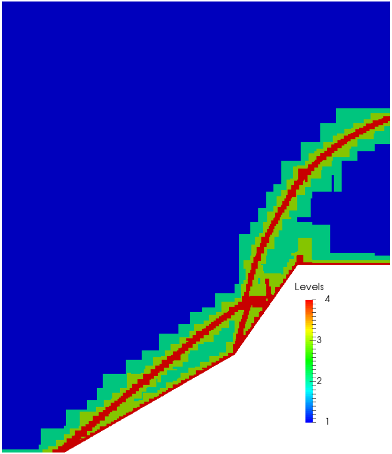
Dynamic load balancing distributes the cells across the processors.

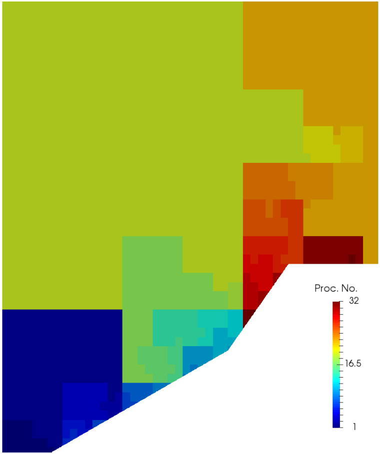




Double Wedge

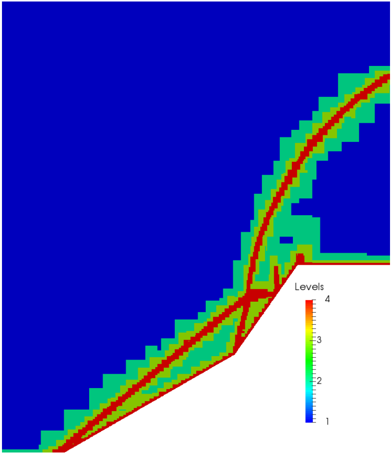
Dynamic load balancing distributes the cells across the processors.

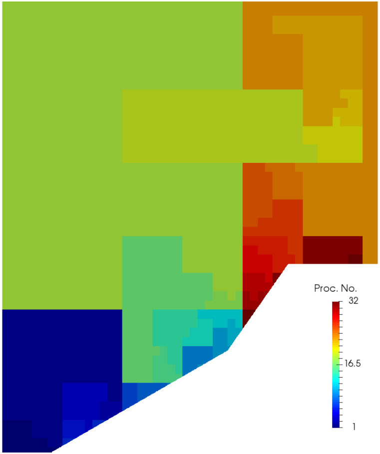




Double Wedge

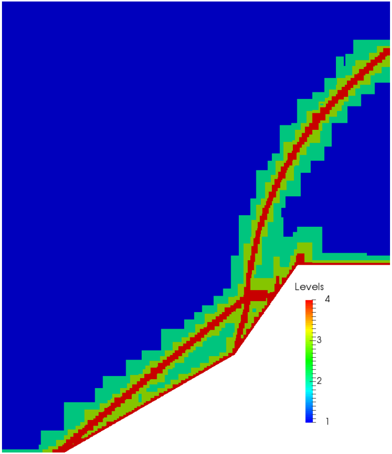
Dynamic load balancing distributes the cells across the processors.

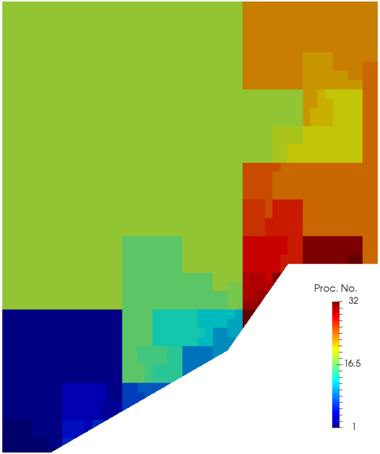




Double Wedge

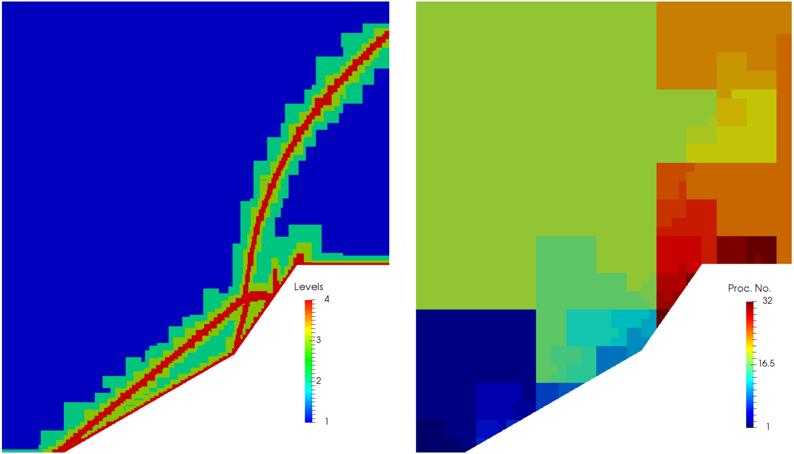
Dynamic load balancing distributes the cells across the processors.





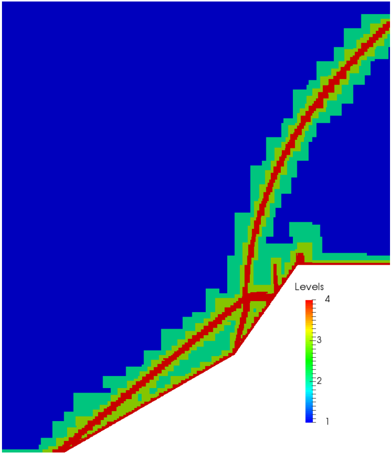
Double Wedge

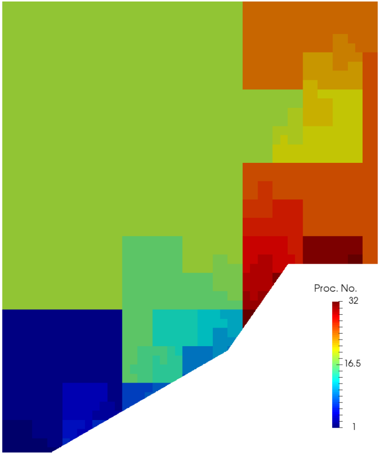
Dynamic load balancing distributes the cells across the processors.



Double Wedge

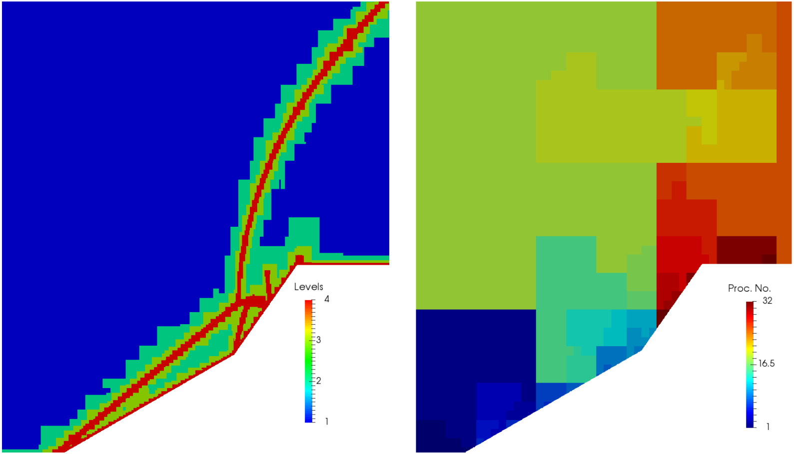
Dynamic load balancing distributes the cells across the processors.





Double Wedge

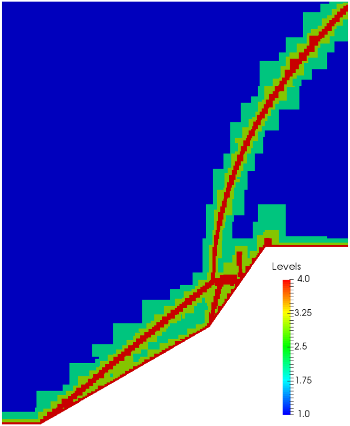
Dynamic load balancing distributes the cells across the processors.

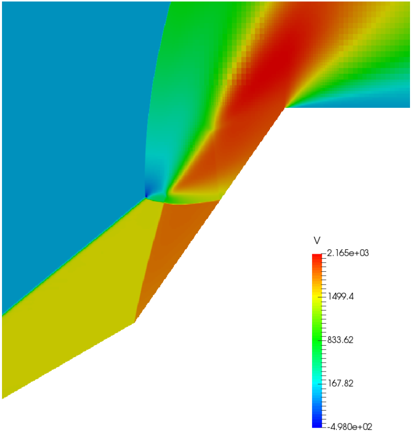


$t = 242\,\mu\text{secs.}$

Double Wedge

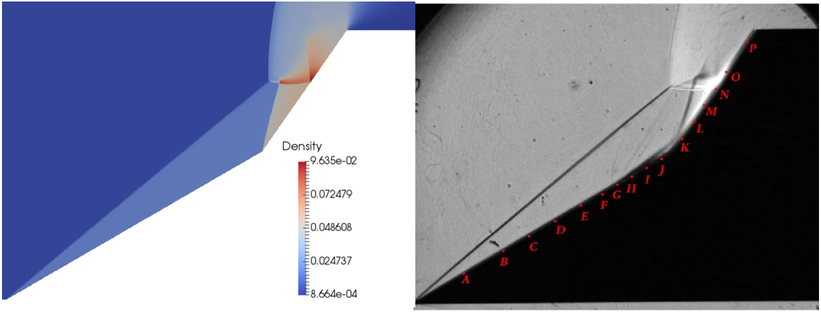
The AMR enables the flow features to be captured in detail.





Double Wedge

The AMR enables the flow features to be captured in detail.



The schlieren image is taken from [Pezzella et al., 2015].

Mapped Solution Update

Within the AMROC-Clawpack framework, the solution is stored in physical (x, y) space and the fluxes are mapped from computational (ξ, η) space.

Mapped Solution Update

Within the AMROC-Clawpack framework, the solution is stored in physical (x, y) space and the fluxes are mapped from computational (ξ, η) space.
Using dimensional splitting the solution update is given by:

Mapped Solution Update

Within the AMROC-Clawpack framework, the solution is stored in physical (x, y) space and the fluxes are mapped from computational (ξ, η) space.
Using dimensional splitting the solution update is given by:

$$\mathbf{Q}^{*}_{i,j} = \mathbf{Q}^{n}_{i,j} - \frac{\Delta t}{\Delta \xi}\left[\left(\hat{\mathbf{F}} - \hat{\mathbf{F}}^{v}\right)_{i+1,j} - \left(\hat{\mathbf{F}} - \hat{\mathbf{F}}^{v}\right)_{i,j}\right]\frac{\Delta\eta\Delta\xi}{V_{i,j}}\,,$$

Mapped Solution Update

Within the AMROC-Clawpack framework, the solution is stored in physical (x, y) space and the fluxes are mapped from computational (ξ, η) space.

Using dimensional splitting the solution update is given by:

$$\mathbf{Q}^{*}_{i,j} = \mathbf{Q}^{n}_{i,j} - \frac{\Delta t}{\Delta \xi}\left[\left(\hat{\mathbf{F}} - \hat{\mathbf{F}}^{v}\right)_{i+1,j} - \left(\hat{\mathbf{F}} - \hat{\mathbf{F}}^{v}\right)_{i,j}\right]\frac{\Delta\eta\Delta\xi}{V_{i,j}},$$

$$\mathbf{Q}^{n+1}_{i,j} = \mathbf{Q}^{*}_{i,j} - \frac{\Delta t}{\Delta \eta}\left[\left(\hat{\mathbf{G}} - \hat{\mathbf{G}}^{v}\right)_{i,j+1} - \left(\hat{\mathbf{G}} - \hat{\mathbf{G}}^{v}\right)_{i,j}\right]\frac{\Delta\eta\Delta\xi}{V_{i,j}}.$$

Mapped Solution Update

Within the AMROC-Clawpack framework, the solution is stored in physical (x, y) space and the fluxes are mapped from computational (ξ, η) space.
Using dimensional splitting the solution update is given by:

$$\mathbf{Q}^{*}_{i,j} = \mathbf{Q}^{n}_{i,j} - \frac{\Delta t}{\Delta \xi}\left[\left(\hat{\mathbf{F}} - \hat{\mathbf{F}}^{v}\right)_{i+1,j} - \left(\hat{\mathbf{F}} - \hat{\mathbf{F}}^{v}\right)_{i,j}\right]\frac{\Delta\eta\Delta\xi}{V_{i,j}}\,,$$

$$\mathbf{Q}^{n+1}_{i,j} = \mathbf{Q}^{*}_{i,j} - \frac{\Delta t}{\Delta \eta}\left[\left(\hat{\mathbf{G}} - \hat{\mathbf{G}}^{v}\right)_{i,j+1} - \left(\hat{\mathbf{G}} - \hat{\mathbf{G}}^{v}\right)_{i,j}\right]\frac{\Delta\eta\Delta\xi}{V_{i,j}}\,.$$

where $V_{i,j}$ is the volume of cell i, j in physical space. $\hat{\mathbf{F}}$, $\hat{\mathbf{F}}^{v}$, $\hat{\mathbf{G}}$, $\hat{\mathbf{G}}^{v}$ are the physical fluxes **per computational unit length**.

Mapped Mesh Computation

In the mapped mesh computations, the flux is transformed to align with the cell face,

$$\hat{\mathbf{F}} = T^{-1}\mathbf{F}_n(T\,\mathbf{Q}_l, T\,\mathbf{Q}_r)\,,$$

Mapped Mesh Computation

In the mapped mesh computations, the flux is transformed to align with the cell face,

$$\hat{\mathbf{F}} = T^{-1}\mathbf{F}_n(T\,\mathbf{Q}_l, T\,\mathbf{Q}_r)\,,$$

where T is the transformation matrix,

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{n}^x & \hat{n}^y & 0 & 0 \\ 0 & 0 & 0 & -\hat{n}^y & \hat{n}^x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Mapped Inviscid Fluxes

The inviscid fluxes per computational unit length are found by:

Mapped Inviscid Fluxes

The inviscid fluxes per computational unit length are found by:

- Rotating the momentum components to be normal to the face,
- Calculating the flux with the rotated solution vectors,
- Rotating the solution vector back,
- Scaling the flux using the ratio of the computational face to the mapped face

Mapped Inviscid Fluxes

The inviscid fluxes per computational unit length are found by:

- Rotating the momentum components to be normal to the face,
- Calculating the flux with the rotated solution vectors,
- Rotating the solution vector back,
- Scaling the flux using the ratio of the computational face to the mapped face

In the ξ directional sweep, this gives

$$\mathbf{F}_{i-1/2,j} = T^{-1}_{i-1/2,j}\mathbf{F}_n\left(T_{i-1/2,j}\mathbf{Q}_{i-1,j},\, T_{i-1/2,j}\mathbf{Q}_{i,j}\right).$$

where T is the rotation matrix used to rotate the momentum components, and $\mathbf{F}_n$ is the normal flux through the face.

Mapped Inviscid Fluxes

The inviscid fluxes per computational unit length are found by:

- Rotating the momentum components to be normal to the face,
- Calculating the flux with the rotated solution vectors,
- Rotating the solution vector back,
- Scaling the flux using the ratio of the computational face to the mapped face

In the ξ directional sweep, this gives

$$\mathbf{F}_{i-1/2,j} = T^{-1}_{i-1/2,j}\mathbf{F}_n\big(T_{i-1/2,j}\mathbf{Q}_{i-1,j},\, T_{i-1/2,j}\mathbf{Q}_{i,j}\big)\,.$$

where T is the rotation matrix used to rotate the momentum components, and $\mathbf{F}_n$ is the normal flux through the face.
The scaling is given by:

$$\hat{\mathbf{F}}_{i,j} = \frac{|\mathbf{n}_{i-1/2,j}|}{\Delta\eta}\,\mathbf{F}_{i-1/2,j}\,,$$

Mapped Viscous Fluxes

The physical viscous flux per computational unit length in the ξ directional sweep is given by,

$$\hat{\mathbf{F}}^{v}_{i-1/2,j} = \frac{\left|\mathbf{n}_{i-1/2,j}\right|}{\Delta\eta} \left[\left(\mathbf{F}^{v}\hat{n}^{x}\right)_{i-1/2,j} + \left(\mathbf{G}^{v}\hat{n}^{y}\right)_{i-1/2,j} \right] ,$$

Mapped Viscous Fluxes

The physical viscous flux per computational unit length in the ξ directional sweep is given by,

$$\hat{\mathbf{F}}^{v}_{i-1/2,j} = \frac{\left|\mathbf{n}_{i-1/2,j}\right|}{\Delta\eta} \left[(\mathbf{F}^{v}\hat{n}^{x})_{i-1/2,j} + (\mathbf{G}^{v}\hat{n}^{y})_{i-1/2,j} \right] ,$$

To calculate the derivatives needed for $\mathbf{F}^v$ and $\mathbf{G}^v$, one must use

$$\frac{\partial\phi}{\partial x} = \left(\frac{\partial\phi}{\partial\xi}\right)\left(\frac{\partial\xi}{\partial x}\right) + \left(\frac{\partial\phi}{\partial\eta}\right)\left(\frac{\partial\eta}{\partial x}\right) ,$$

and,

$$\frac{\partial\phi}{\partial y} = \left(\frac{\partial\phi}{\partial\xi}\right)\left(\frac{\partial\xi}{\partial y}\right) + \left(\frac{\partial\phi}{\partial\eta}\right)\left(\frac{\partial\eta}{\partial y}\right) .$$

Boundary Conditions

For wall boundary conditions the ghost cell values are set by first transforming the domain variables,

$$\hat{\mathbf{Q}} = T_w \mathbf{Q}_{\mathrm{dom.}} \, .$$

Boundary Conditions

For wall boundary conditions the ghost cell values are set by first transforming the domain variables,

$$\hat{\mathbf{Q}} = T_w \mathbf{Q}_{\mathrm{dom.}} \,.$$

Boundary Conditions

For wall boundary conditions the ghost cell values are set by first transforming the domain variables,

$$\hat{\mathbf{Q}} = T_w \mathbf{Q}_{\mathrm{dom.}} \,.$$

Then setting the ghost cell variables using interpolation,

$$\hat{\mathbf{Q}}^{\rho u}_{\mathrm{gc}} = \frac{-\frac{d_{gw}}{d_{gd}}\hat{\mathbf{Q}}^{\rho u}}{1-\frac{d_{gw}}{d_{gd}}} \,,$$

and

$$\hat{\mathbf{Q}}^{\rho v}_{\mathrm{gc}} = \hat{\mathbf{Q}}^{\rho v} \ \mathrm{slip}, \quad \hat{\mathbf{Q}}^{\rho v}_{\mathrm{gc}} = \frac{-\frac{d_{gw}}{d_{gd}}\hat{\mathbf{Q}}^{\rho v}}{1-\frac{d_{gw}}{d_{gd}}} \ \mathrm{no-slip},$$

Boundary Conditions

For wall boundary conditions the ghost cell values are set by first transforming the domain variables,

$$\hat{\mathbf{Q}} = T_w \mathbf{Q}_{\mathrm{dom.}} \,.$$

Then setting the ghost cell variables using interpolation,

$$\hat{\mathbf{Q}}^{\rho u}_{\mathrm{gc}} = \frac{-\frac{d_{gw}}{d_{gd}} \hat{\mathbf{Q}}^{\rho u}}{1 - \frac{d_{gw}}{d_{gd}}} \,,$$

and

$$\hat{\mathbf{Q}}^{\rho v}_{\mathrm{gc}} = \hat{\mathbf{Q}}^{\rho v} \ \ \mathrm{slip}, \quad \hat{\mathbf{Q}}^{\rho v}_{\mathrm{gc}} = \frac{-\frac{d_{gw}}{d_{gd}} \hat{\mathbf{Q}}^{\rho v}}{1 - \frac{d_{gw}}{d_{gd}}} \ \ \mathrm{no-slip},$$

Then rotating the ghost cell values using the inverse transformation,

$$\mathbf{Q}_{\mathrm{gc}} = T_w^{-1} \hat{\mathbf{Q}}_{\mathrm{gc}} \,.$$

CFL condition

The time step must be adjusted to account for the changes in mesh size.

CFL condition

The time step must be adjusted to account for the changes in mesh size. The Courant-Friedrichs-Lewy (CFL) condition can be written as [Moukalled et al., 2015],

$$\sum_f \left[\frac{\lambda_f^v |\mathbf{n}|_f}{d_f} + \lambda_f^c |\mathbf{n}|_f \right] - \frac{V_c}{\Delta t} \leq 0 \, ,$$

where λ_f^v and λ_f^c are the viscous and convective spectral radii, respectively, and d_f is the distance between the cell centres either side of the face.

CFL condition

The time step must be adjusted to account for the changes in mesh size. The Courant-Friedrichs-Lewy (CFL) condition can be written as [Moukalled et al., 2015],

$$\sum_f \left[\frac{\lambda_f^v |\mathbf{n}|_f}{d_f} + \lambda_f^c |\mathbf{n}|_f \right] - \frac{V_c}{\Delta t} \leq 0\,,$$

where λ_f^v and λ_f^c are the viscous and convective spectral radii, respectively, and d_f is the distance between the cell centres either side of the face.

Rearranging the above equation gives,

$$\frac{\Delta t}{V_c} \sum_f \left[\frac{\lambda_f^v}{d_f} + \lambda_f^c \right] |\mathbf{n}|_f \leq 1\,.$$

CFL Condition

With dimensional splitting, the CFL condition must be evaluated in each dimension separately, giving,

$$\max\left(\left[\frac{\lambda^{v}_{i-1/2,j}}{d_{i-1/2,j}} + \lambda^{c}_{i-1/2,j}\right] |\mathbf{n}|_{i-1/2,j} + \left[\frac{\lambda^{v}_{i+1/2,j}}{d_{i+1/2,j}} + \lambda^{c}_{i+1/2,j}\right] |\mathbf{n}|_{i+1/2,j}\,, \right.$$
$$\left. \left[\frac{\lambda^{v}_{i,j-1/2}}{d_{i,j-1/2}} + \lambda^{c}_{i,j-1/2}\right] |\mathbf{n}|_{i,j-1/2} + \left[\frac{\lambda^{v}_{i,j+1/2}}{d_{i,j+1/2}} + \lambda^{c}_{i,j+1/2}\right] |\mathbf{n}|_{i,j+1/2} \right) \frac{\Delta t}{V_c} \leq 1\,.$$

Hypersonic Sphere

Simulations of a half inch sphere travelling at hypersonic speeds in air [Lobb, 1964].

Mach number range between 8.4 and 16.1, with $p_\infty = 1333\,\mathrm{Pa}$ and $T_\infty = 293\,\mathrm{K}$.

The shock standoff distance was measured at each condition.

Hypersonic Sphere

Simulations of a half inch sphere travelling at hypersonic speeds in air [Lobb, 1964].

Mach number range between 8.4 and 16.1, with $p_\infty = 1333\,\mathrm{Pa}$ and $T_\infty = 293\,\mathrm{K}$.

The shock standoff distance was measured at each condition.

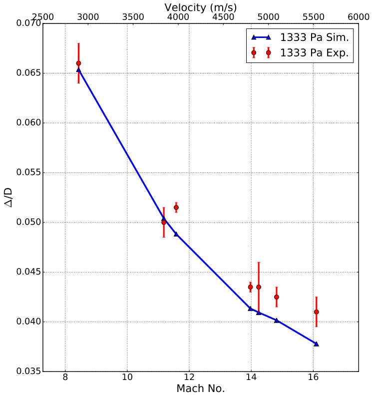
The shock standoff distance is used to validate the non-equilibrium model.

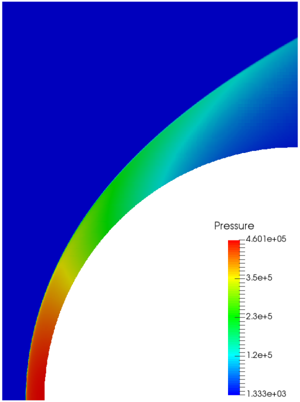
Validation of the axi-symmetric source term.

$$\mathbf{W}_{\mathrm{axi}} = -\frac{1}{y}\begin{bmatrix} \rho_1 v \\ \vdots \\ \rho_N v \\ \rho u v \\ \rho v^2 \\ (\rho E + p) v \end{bmatrix}$$

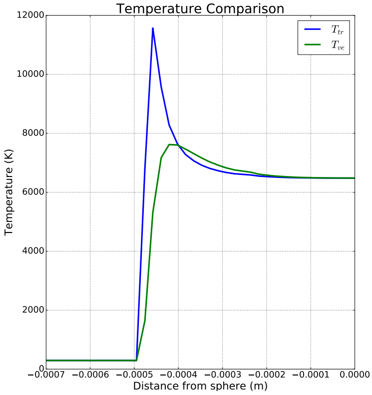
Hypersonic Sphere

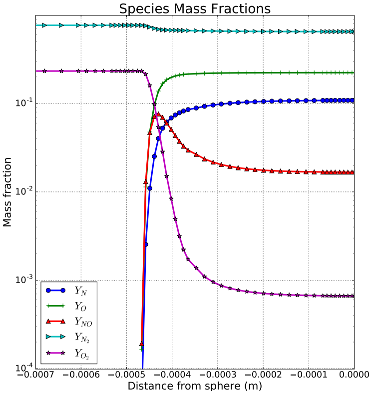
Computed shock standoff distances compared with experimental data.





Hypersonic Sphere





Mapped Mesh Computation

Experiments of a cylinder in hypersonic flow [Hornung, 1972] were simulated with the mapping and initial conditions given by,

$$x = \xi \cos(\eta), \quad y = -\xi \sin(\eta).$$

Radius	Y_{N_2}	Y_N	T_∞	p_∞	U_∞	M_∞
$0.0127\,\mathrm{m}$	0.927	0.073	$1833\,\mathrm{K}$	$2.91\,\mathrm{kPa}$	$5590\,\mathrm{m/s}$	6.14

Table: Cylinder geometry and freestream conditions

Mapped Mesh Computation

Experiments of a cylinder in hypersonic flow [Hornung, 1972] were simulated with the mapping and initial conditions given by,

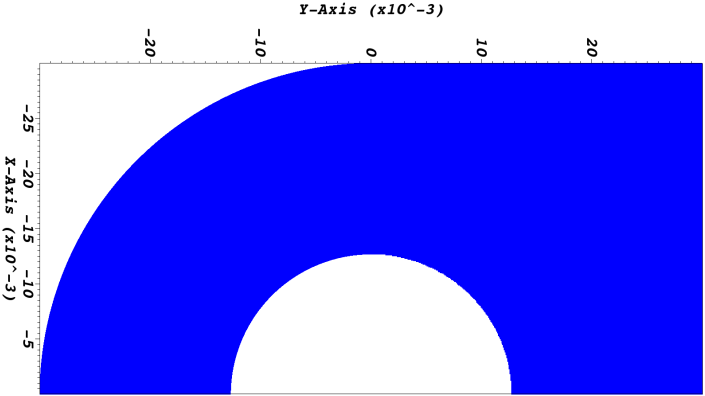
$$x = \xi \cos(\eta), \quad y = -\xi \sin(\eta).$$

Radius	Y_{N_2}	Y_N	T_∞	p_∞	U_∞	M_∞
0.0127 m	0.927	0.073	1833 K	2.91 kPa	5590 m/s	6.14

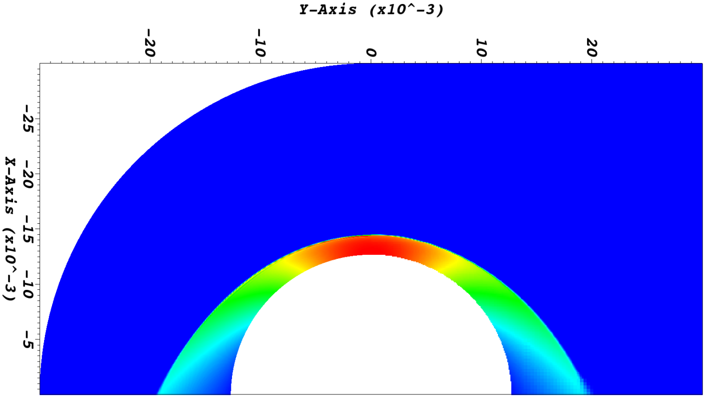
Table: Cylinder geometry and freestream conditions

The implementation was verified by comparing a mapped computation with a embedded boundary computation.

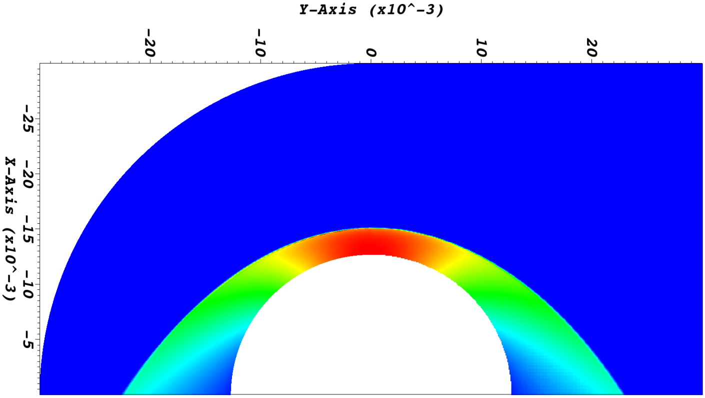
Mapped Mesh Computation



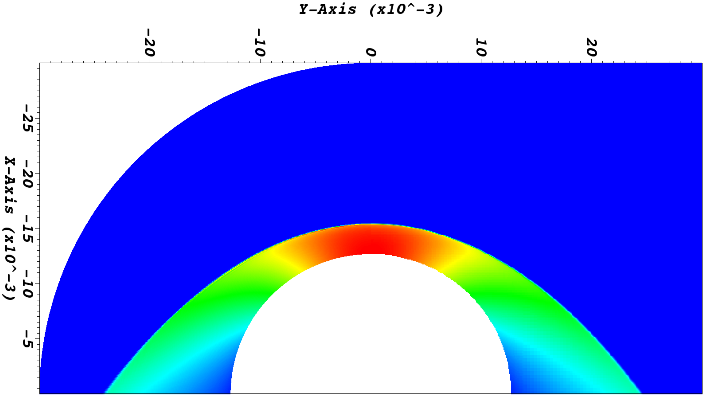
Mapped Mesh Computation



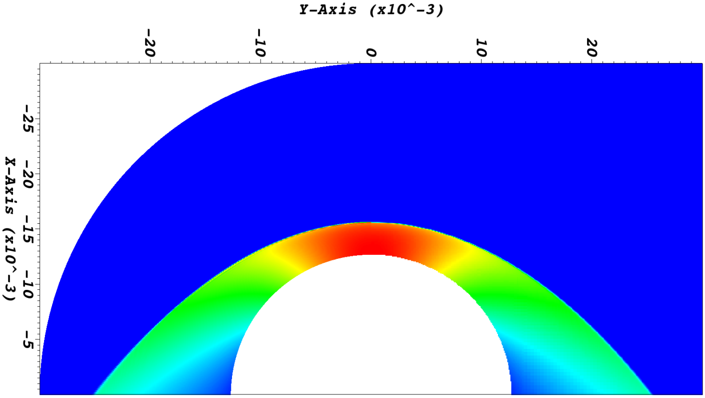
Mapped Mesh Computation



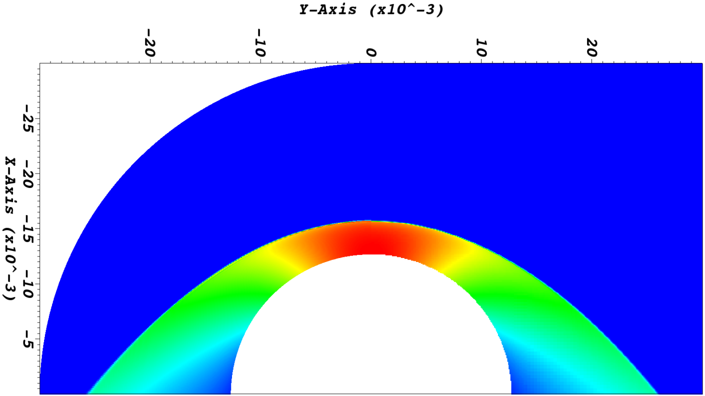
Mapped Mesh Computation



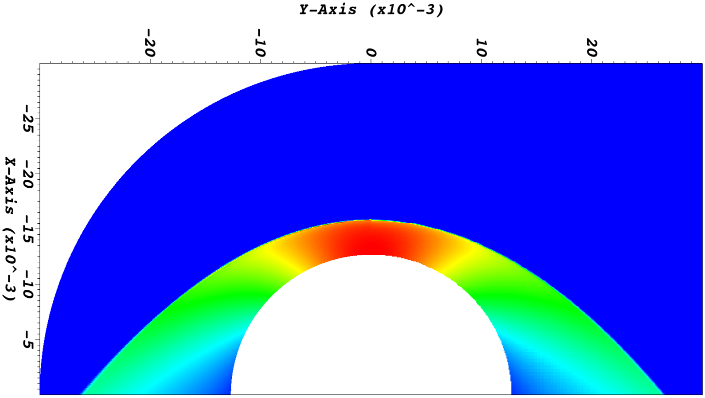
Mapped Mesh Computation



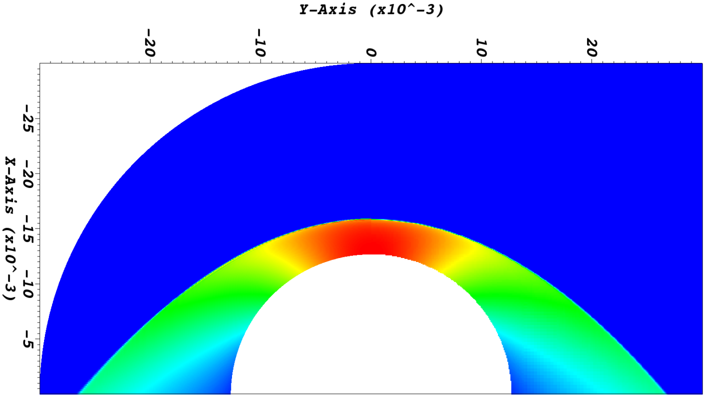
Mapped Mesh Computation



Mapped Mesh Computation

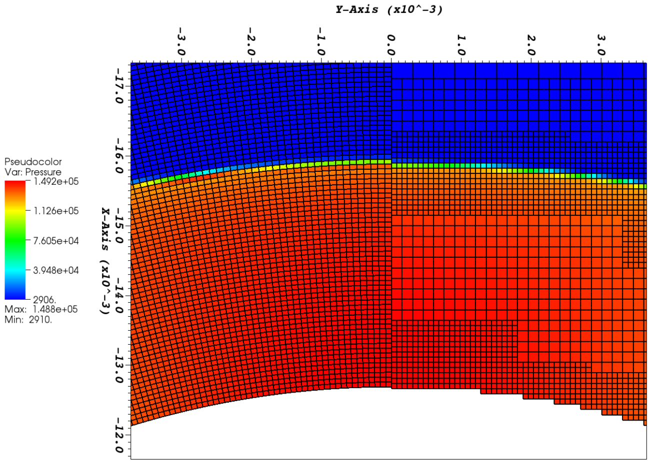


Mapped Mesh Computation



$t = 100\,\mu\text{sec}$

Mapped Mesh Computation



Viscous Computations

Preliminary results have been obtained for computations including the viscous flux vectors,

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \left(\mathbf{F} - \mathbf{F}^v\right)}{\partial x} + \frac{\partial \left(\mathbf{G} - \mathbf{G}^v\right)}{\partial y} = \mathbf{W}$$

Viscous Computations

Preliminary results have been obtained for computations including the viscous flux vectors,

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \left(\mathbf{F} - \mathbf{F}^v\right)}{\partial x} + \frac{\partial \left(\mathbf{G} - \mathbf{G}^v\right)}{\partial y} = \mathbf{W}$$

where,

$$\mathbf{F}^v = \begin{bmatrix} -J_{x,1} \\ \vdots \\ -J_{x,N_s} \\ \tau_{x,x} \\ \tau_{y,x} \\ \kappa_{ve}\frac{\partial T_{ve}}{\partial x} - \sum_{s=1}^{N_s} J_{x,s}e_{ve} \\ \kappa_{tr}\frac{\partial T_{tr}}{\partial x} + \kappa_{ve}\frac{\partial T_{ve}}{\partial x} + u\tau_{x,x} + v\tau_{y,x} - \sum_{s=1}^{N_s} J_{x,s}h_s \end{bmatrix}.$$

and a similar expression is obtained for $\mathbf{G}^v$.

Viscous Computations

The species diffusion uses a modified version of Fick's diffusion law [Sutton and Gnoffo, 1998],

$$J_{x,s} = -\rho D_s \frac{\partial Y_s}{\partial x} - Y_s \sum_{r=1}^{N_s} (-\rho D_r \frac{\partial Y_r}{\partial x}) \,.$$

Viscous Computations

The species diffusion uses a modified version of Fick's diffusion law [Sutton and Gnoffo, 1998],

$$J_{x,s} = -\rho D_s \frac{\partial Y_s}{\partial x} - Y_s \sum_{r=1}^{N_s} (-\rho D_r \frac{\partial Y_r}{\partial x}) \,.$$

The viscous stress tensor, $\tau_{i,j}$ is given by,

$$\tau_{i,j} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \delta_{i,j} \, \frac{2}{3} \mu \nabla \cdot \mathbf{u} \,,$$

where $\delta_{i,j}$ is the Kronecker delta.

Viscous Computations

The species diffusion uses a modified version of Fick's diffusion law [Sutton and Gnoffo, 1998],

$$J_{x,s} = -\rho D_s \frac{\partial Y_s}{\partial x} - Y_s \sum_{r=1}^{N_s} (-\rho D_r \frac{\partial Y_r}{\partial x}) \, .$$

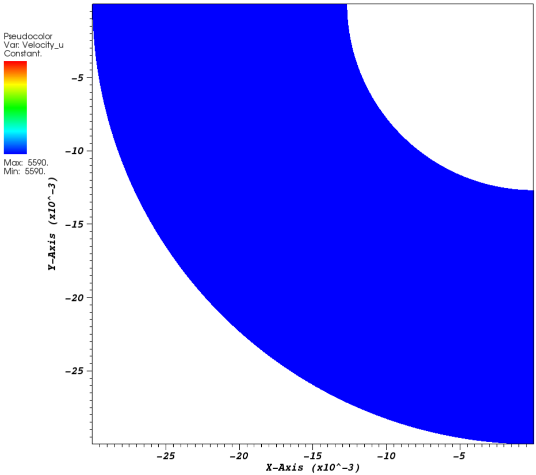
The viscous stress tensor, $\tau_{i,j}$ is given by,

$$\tau_{i,j} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \delta_{i,j} \, \frac{2}{3} \mu \nabla \cdot \mathbf{u} \, ,$$

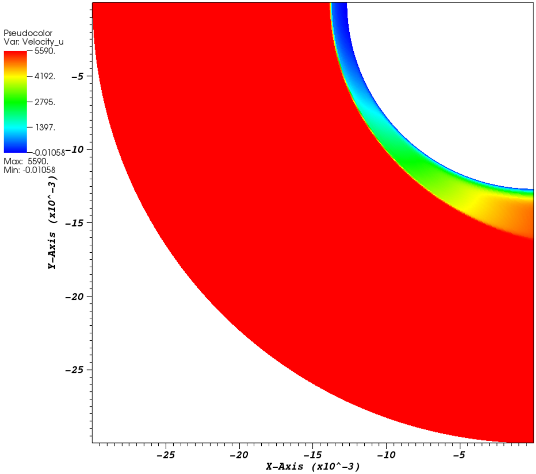
where $\delta_{i,j}$ is the Kronecker delta.

The diffusion coefficients, the viscosity and the thermal conductivities are all calculated within the Mutation++ library.

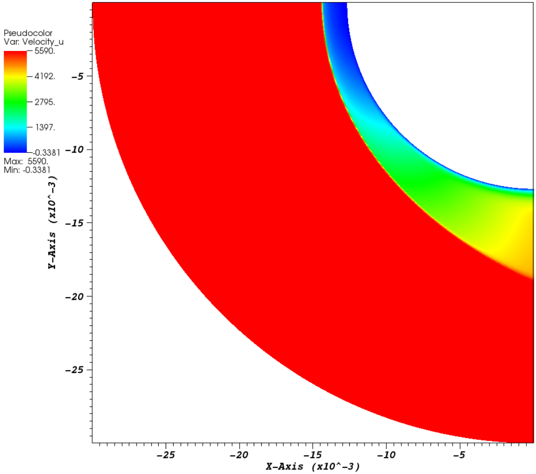
Viscous Computations



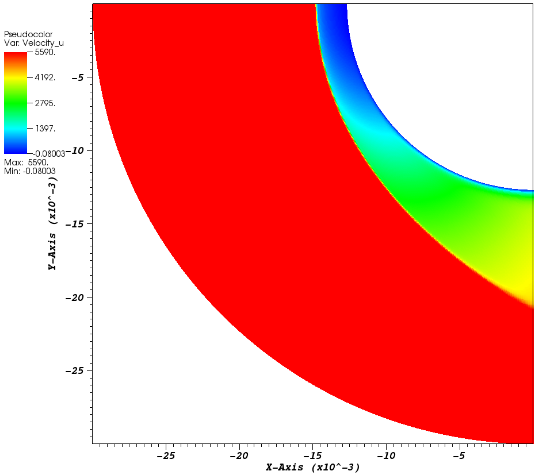
Viscous Computations



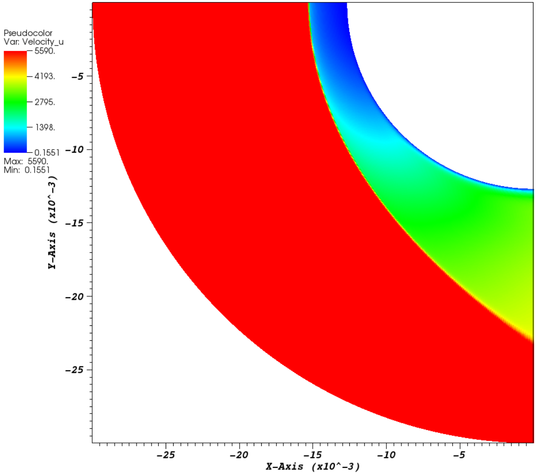
Viscous Computations



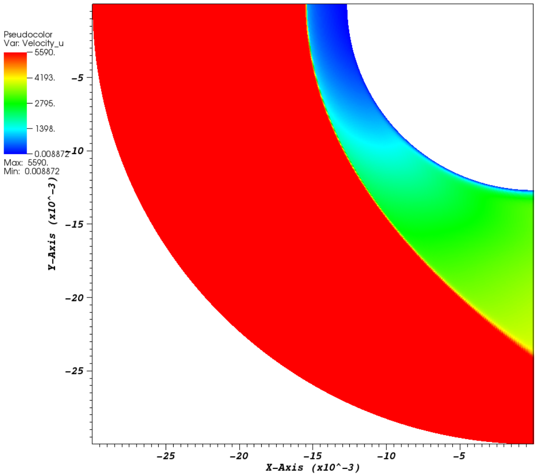
Viscous Computations



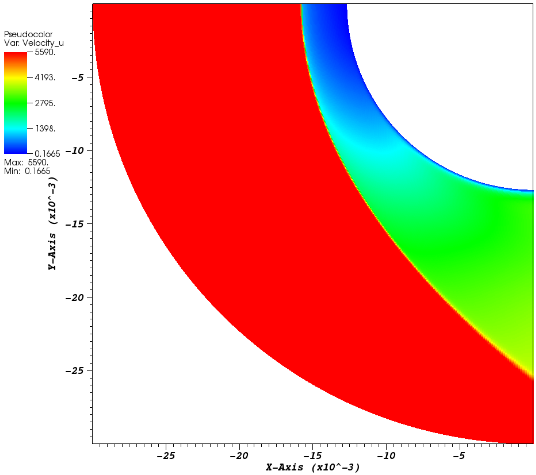
Viscous Computations



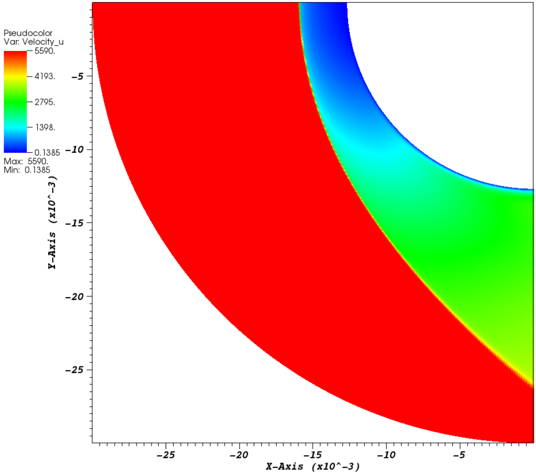
Viscous Computations



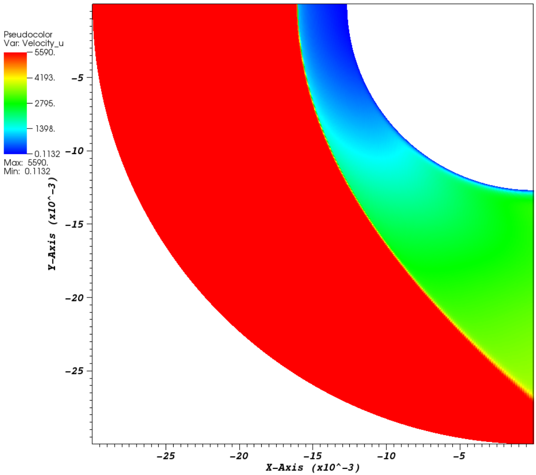
Viscous Computations



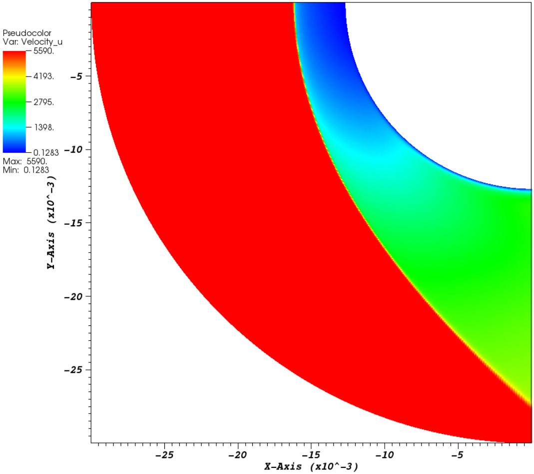
Viscous Computations



Viscous Computations



Viscous Computations



$t = 60\,\mu\text{secs.}$

Flat Plate Comparison

To test the implementation of the viscous fluxes a comparison between the mapped AMROC solver and the SU2 solver was completed.

Flat Plate Comparison

To test the implementation of the viscous fluxes a comparison between the mapped AMROC solver and the SU2 solver was completed.
A hyperbolic tangent mapping to stretch the grid away from the wall, with an initial spacing of 1e-5 m.

Flat Plate Comparison

To test the implementation of the viscous fluxes a comparison between the mapped AMROC solver and the SU2 solver was completed.
A hyperbolic tangent mapping to stretch the grid away from the wall, with an initial spacing of 1e-5 m.
A Mach 3 flow over a 0.3 m flat plate was simulated using both an isothermal and adiabatic wall using the same mesh in each solver.

Flat Plate Comparison

A comparison between the two boundary layers at $0.2\,\mathrm{m}$ is shown below,

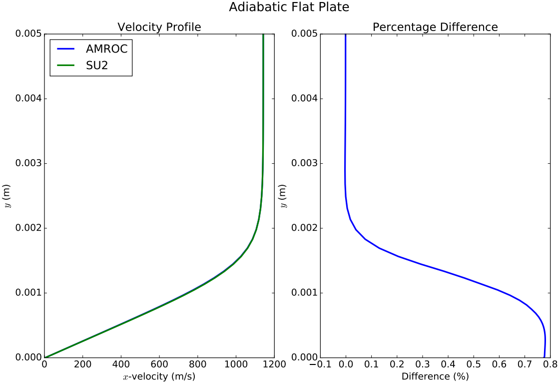


Figure: A comparison of the velocity boundary layers over an adiabatic flat plate, where $M_\infty = 3.0$.

Flat Plate Comparison

A comparison between the two boundary layers at $0.2\,\mathrm{m}$ is shown below,

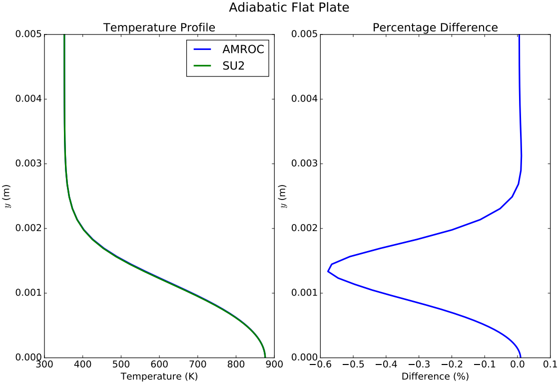


Figure: A comparison of the thermal boundary layers over an adiabatic flat plate, where $M_\infty = 3.0$.

Flat Plate Comparison

A comparison between the two boundary layers at $0.2\,\mathrm{m}$ is shown below,

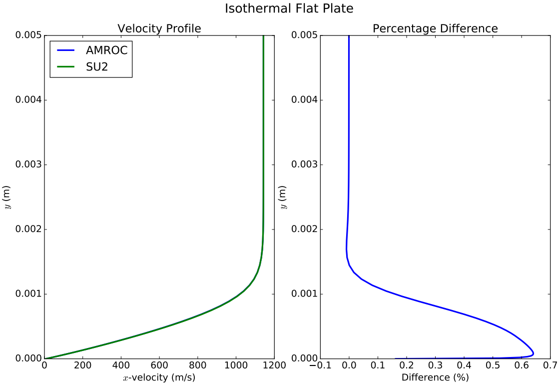


Figure: A comparison of the velocity boundary layers over an isothermal flat plate, where $M_\infty = 3.0$.

Flat Plate Comparison

A comparison between the two boundary layers at 0.2 m is shown below,

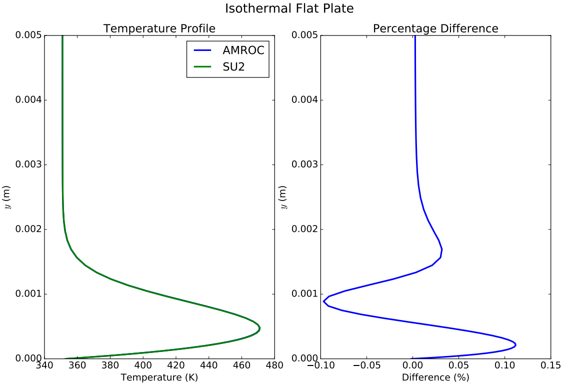


Figure: A comparison of the thermal boundary layers over an isothermal flat plate, where $M_\infty = 3.0$.

Cylinder Heat Flux Computation

The mapped mesh solver has been validated by simulating a cylinder in a nonequilibrium, high enthalpy flow.

Cylinder Heat Flux Computation

The mapped mesh solver has been validated by simulating a cylinder in a nonequilibrium, high enthalpy flow.
The inflow conditions and results were taken from [Degrez et al., 2009].

T_∞	ρ_∞	U_∞	Y_{N_2}	Y_N	Y_{O_2}	Y_O	Y_{NO}
694 K	3.26 g/m^3	4776 m/s	0.7356	0.0	0.1340	0.07955	0.0509

Table: Freestream conditions for the HEG cylinder simulation.

Cylinder Heat Flux Computation

The mapped mesh solver has been validated by simulating a cylinder in a nonequilibrium, high enthalpy flow.
The inflow conditions and results were taken from [Degrez et al., 2009].

T_∞	ρ_∞	U_∞	Y_{N_2}	Y_N	Y_{O_2}	Y_O	Y_{NO}
694 K	3.26 g/m^3	4776 m/s	0.7356	0.0	0.1340	0.07955	0.0509

Table: Freestream conditions for the HEG cylinder simulation.

Cylinder Heat Flux Computation

The mapped mesh solver has been validated by simulating a cylinder in a nonequilibrium, high enthalpy flow.
The inflow conditions and results were taken from [Degrez et al., 2009].

T_∞	ρ_∞	U_∞	Y_{N_2}	Y_N	Y_{O_2}	Y_O	Y_{NO}
694 K	3.26 g/m^3	4776 m/s	0.7356	0.0	0.1340	0.07955	0.0509

Table: Freestream conditions for the HEG cylinder simulation.

A cylinder mesh was generated with hyperbolic tangent stretching away from the wall using a 1e-6 initial spacing.

Cylinder Heat Flux Comparison

The simulated results show good agreement with the experimental results:

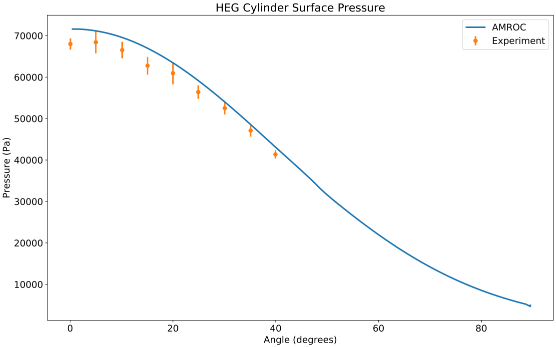


Figure: A comparison of the experimental and simulated surface pressures in the HEG cylinder experiment.

Cylinder Heat Flux Comparison

The simulated results show good agreement with the experimental results:

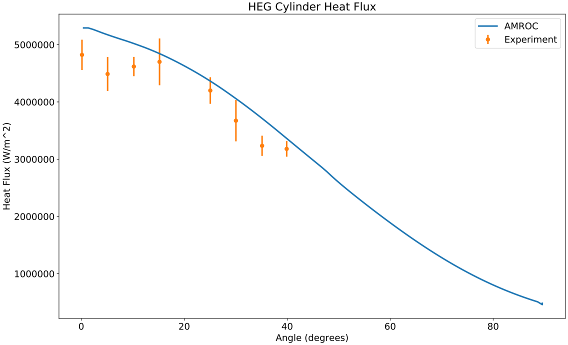


Figure: A comparison of the experimental and simulated surface heat fluxes in the HEG cylinder experiment.

Hybrid method

Convective numerical flux is defined as

$$\mathbf{F}^n_{inv} = \begin{cases} \mathbf{F}^n_{inv-WENO}, & \text{in } \mathcal{C} \\ \mathbf{F}^n_{inv-CD}, & \text{in } \overline{\mathcal{C}}, \end{cases}$$

Hybrid method

Convective numerical flux is defined as

$$\mathbf{F}^n_{inv} = \begin{cases} \mathbf{F}^n_{inv-WENO}, & \text{in } \mathcal{C} \\ \mathbf{F}^n_{inv-CD}, & \text{in } \overline{\mathcal{C}}, \end{cases}$$

- For LES: 3rd order WENO method, 2nd order TCD [Hill and Pullin, 2004]

Hybrid method

Convective numerical flux is defined as

$$\mathbf{F}^n_{inv} = \begin{cases} \mathbf{F}^n_{inv-WENO}, & \text{in } \mathcal{C} \\ \mathbf{F}^n_{inv-CD}, & \text{in } \overline{\mathcal{C}}, \end{cases}$$

- For LES: 3rd order WENO method, 2nd order TCD [Hill and Pullin, 2004]
- For DNS: Symmetric 6th order WENO, 6th-order CD scheme

J. Ziegler, RD, J. Shepherd, D. Pullin, *J. Comput. Phys.* 230(20):7598-7630, 2011.

Hybrid method

Convective numerical flux is defined as

$$\mathbf{F}^n_{inv} = \begin{cases} \mathbf{F}^n_{inv-WENO}, & \text{in } \mathcal{C} \\ \mathbf{F}^n_{inv-CD}, & \text{in } \overline{\mathcal{C}}, \end{cases}$$

- For LES: 3rd order WENO method, 2nd order TCD [Hill and Pullin, 2004]
- For DNS: Symmetric 6th order WENO, 6th-order CD scheme

  J. Ziegler, RD, J. Shepherd, D. Pullin, *J. Comput. Phys.* 230(20):7598-7630, 2011.

Use WENO scheme to only capture shock waves but resolve interface between species.

Hybrid method

Convective numerical flux is defined as

$$\mathbf{F}^n_{inv} = \begin{cases} \mathbf{F}^n_{inv-WENO}, & \text{in } \mathcal{C} \\ \mathbf{F}^n_{inv-CD}, & \text{in } \overline{\mathcal{C}}, \end{cases}$$

- For LES: 3rd order WENO method, 2nd order TCD [Hill and Pullin, 2004]
- For DNS: Symmetric 6th order WENO, 6th-order CD scheme

  J. Ziegler, RD, J. Shepherd, D. Pullin, *J. Comput. Phys.* 230(20):7598-7630, 2011.

Use WENO scheme to only capture shock waves but resolve interface between species.

Shock detection based on using two criteria together:

1. Lax-Liu entropy condition $|u_R \pm a_R| < |u_* \pm a_*| < |u_L \pm a_L|$ tested with a threshold to eliminate weak acoustic waves. Used intermediate states at cell interfaces:

$$u_* = \frac{\sqrt{\rho_L u_L} + \sqrt{\rho_R u_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \quad a_* = \sqrt{(\gamma_* - 1)(h_* - \frac{1}{2}u_*^2)}, \; \ldots$$

Hybrid method

Convective numerical flux is defined as

$$\mathbf{F}^n_{inv} = \begin{cases} \mathbf{F}^n_{inv-WENO}, & \text{in } \mathcal{C} \\ \mathbf{F}^n_{inv-CD}, & \text{in } \overline{\mathcal{C}}, \end{cases}$$

- For LES: 3rd order WENO method, 2nd order TCD [Hill and Pullin, 2004]
- For DNS: Symmetric 6th order WENO, 6th-order CD scheme

J. Ziegler, RD, J. Shepherd, D. Pullin, *J. Comput. Phys.* 230(20):7598-7630, 2011.

Use WENO scheme to only capture shock waves but resolve interface between species.

Shock detection based on using two criteria together:

1. Lax-Liu entropy condition $|u_R \pm a_R| < |u_* \pm a_*| < |u_L \pm a_L|$ tested with a threshold to eliminate weak acoustic waves. Used intermediate states at cell interfaces:

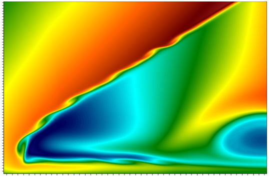
$$u_* = \frac{\sqrt{\rho_L u_L} + \sqrt{\rho_R u_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \quad a_* = \sqrt{(\gamma_* - 1)(h_* - \frac{1}{2} u_*^2)}, \ \dots$$

2. Limiter-inspired discontinuity test based on mapped normalized pressure gradient θ_j

$$\phi(\theta_j) = \frac{2\theta_j}{(1+\theta_j)^2} \quad \text{with} \quad \theta_j = \frac{|p_{j+1} - p_j|}{|p_{j+1} + p_j|}, \quad \phi(\theta_j) > \alpha_{Map}$$

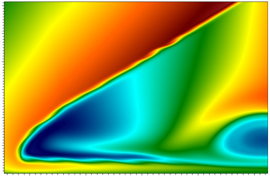
Results for shear layer in Mach reflection pattern

WENO/CD - 6 levels



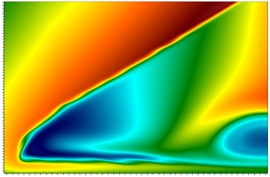
$\Delta x_{\text{min}} = 3.91 \cdot 10^{-6}\,\mathrm{m}$

WENO/CD - 7 levels



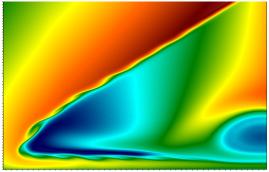
$\Delta x_{\text{min}} = 1.95 \cdot 10^{-6}\,\mathrm{m}$

WENO/CD - 8 levels



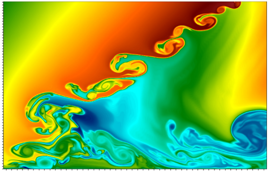
$\Delta x_{\text{min}} = 9.77 \cdot 10^{-7}\,\mathrm{m}$

MUSCL - 7 levels



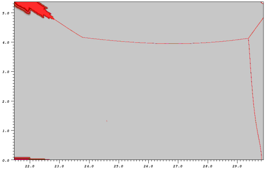
$\Delta x_{\text{min}} = 1.05 \cdot 10^{-6}\,\mathrm{m}$

MUSCL - 7 levels - Euler



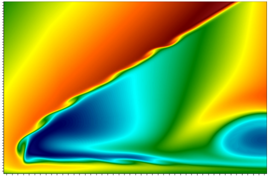
$\Delta x_{\text{min}} = 1.05 \cdot 10^{-6}\,\mathrm{m}$

Usage of WENO for WENO/CD - 8 levels



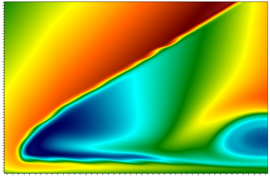
Results for shear layer in Mach reflection pattern

WENO/CD - 6 levels



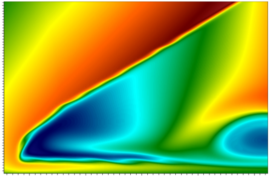
$\Delta x_{\min} = 3.91 \cdot 10^{-6}\,\mathrm{m}$

WENO/CD - 7 levels



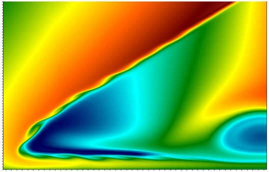
$\Delta x_{\min} = 1.95 \cdot 10^{-6}\,\mathrm{m}$

WENO/CD - 8 levels



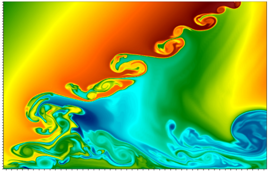
$\Delta x_{\min} = 9.77 \cdot 10^{-7}\,\mathrm{m}$

MUSCL - 7 levels



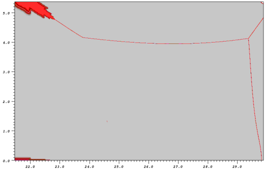
$\Delta x_{\min} = 1.05 \cdot 10^{-6}\,\mathrm{m}$

MUSCL - 7 levels - Euler



$\Delta x_{\min} = 1.05 \cdot 10^{-6}\,\mathrm{m}$

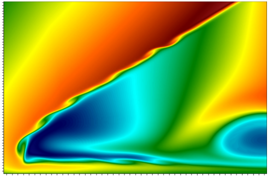
Usage of WENO for WENO/CD - 8 levels



- WENO/CD/RK3 gives results comparable to 4x finer resolved optimal 2nd-order scheme, but CPU times with SAMR 2-3x larger

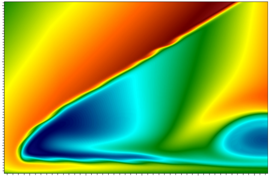
Results for shear layer in Mach reflection pattern

WENO/CD - 6 levels



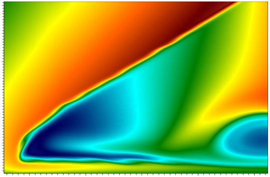
$\Delta x_{\min} = 3.91 \cdot 10^{-6}\,\mathrm{m}$

WENO/CD - 7 levels



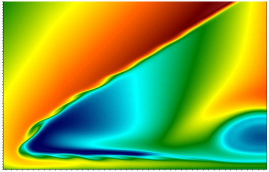
$\Delta x_{\min} = 1.95 \cdot 10^{-6}\,\mathrm{m}$

WENO/CD - 8 levels



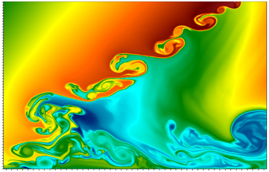
$\Delta x_{\min} = 9.77 \cdot 10^{-7}\,\mathrm{m}$

MUSCL - 7 levels



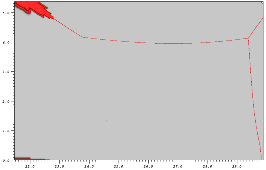
$\Delta x_{\min} = 1.05 \cdot 10^{-6}\,\mathrm{m}$

MUSCL - 7 levels - Euler



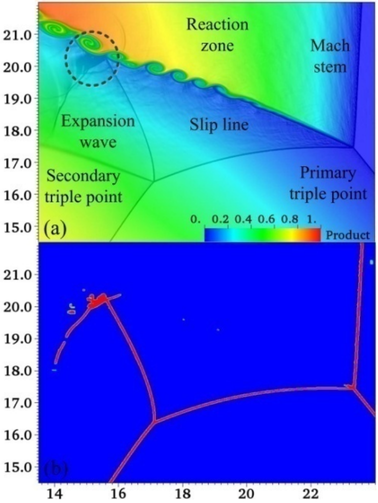
$\Delta x_{\min} = 1.05 \cdot 10^{-6}\,\mathrm{m}$

Usage of WENO for WENO/CD - 8 levels



- WENO/CD/RK3 gives results comparable to 4x finer resolved optimal 2nd-order scheme, but CPU times with SAMR 2-3x larger
- Gain in CPU time from higher-order scheme roughly one order

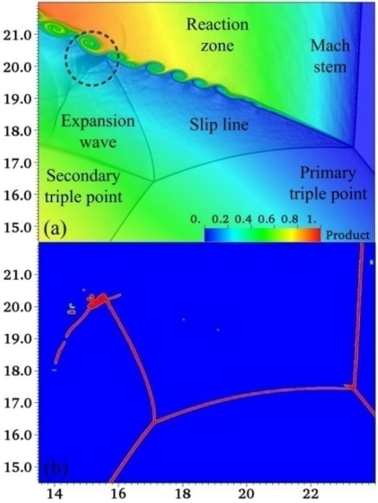
Detonation ignition by hot jet in 2d



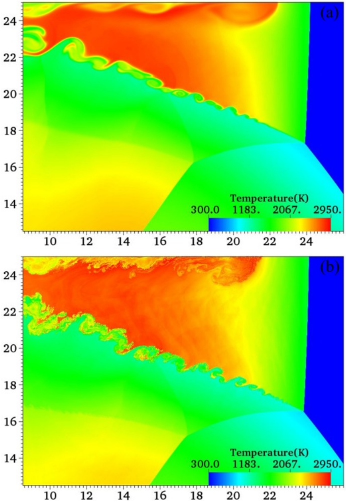
(a) Detailed structure, (b) WENO usage

X. Cai, RD, J. Liang, Y. Mahmoudi, *Proc. Combust. Institute* 36(2): 2725–2733, 2017

Detonation ignition by hot jet in 2d



(a) Detailed structure, (b) WENO usage



(a) Navier-Stokes, (b) Euler

X. Cai, RD, J. Liang, Y. Mahmoudi, *Proc. Combust. Institute* 36(2): 2725–2733, 2017

Conclusions – Hypersonics

- We have developed a first 2D prototype of two-temperature model solver that is suitable for very high temperatures, i.e., high enthalpy re-entry flows
- The Cartesian version is fully integrated into SAMR AMROC-Clawpack; structured non-Cartesian version runs also within AMROC-Clawpack but only on non-adaptive meshes so far
- SAMR framework can remain basically unchanged; however mapping needs to be considered in prolongation and restriction, flux correction, visualization (work in progress)

Conclusions – Hypersonics

- We have developed a first 2D prototype of two-temperature model solver that is suitable for very high temperatures, i.e., high enthalpy re-entry flows
- The Cartesian version is fully integrated into SAMR AMROC-Clawpack; structured non-Cartesian version runs also within AMROC-Clawpack but only on non-adaptive meshes so far
- SAMR framework can remain basically unchanged; however mapping needs to be considered in prolongation and restriction, flux correction, visualization (work in progress)
- For moving geometries, the goal is a Chimera-type approach that constructs non-Cartesian boundary layer meshes near the body and uses SAMR in the far field
- Incorporation of the methodology into the hybrid WENO/CD scheme for high enthalpy DNS in 3D is proposed within the next two years

References I

[Degrez et al., 2009] Degrez, G., Lani, A., Panesi, M., Chazot, O., and Deconinck, H. (2009). Modelling of High Enthalpy, High Mach Number Flows. *Journal of Physics D: Applied Physics*, 42.

[Hill and Pullin, 2004] Hill, D. J. and Pullin, D. I. (2004). Hybrid tuned center difference - WENO method for large eddy simulations in the presence of strong shocks. *J. Comput. Phys.*, 194(2):435–450.

[Hornung, 1972] Hornung, H. G. (1972). Non-equilibrium dissociating nitrogen flow over spheres and circular cylinders. *Journal of Fluid Mechanics*, 53:149–176.

[Liou and Steffen Jr, 1993] Liou, M.-S. and Steffen Jr, C. J. (1993). A new flux splitting scheme. *Journal of Computational Physics*, 107:23–39.

[Lobb, 1964] Lobb, K. R. (1964). Experimental Measurement of Shock Detachment Distance on Spheres Fired in Air at Hypervelocities. *High Temperature Aspects of Hypersonic Flows*, 14(5):519–527.

[Moukalled et al., 2015] Moukalled, F., Mangani, L., and Darwish, M. (2015). *The finite volume method in computational fluid dynamics: An Advanced Introduction with OpenFOAM and Matlab*. Springer.

[Pezzella et al., 2015] Pezzella, G., de Rosa, D., and Donelli, R. (2015). Computational Analysis of Shock Wave Boundary Layer Interactions in Non-equilibrium Hypersonic Flow. *20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, (July):1–12.

[Scoggins and Magin, 2014] Scoggins, J. B. and Magin, T. E. (2014). Development of Mutation++ : MUlticomponent Thermodynamics And Transport properties for IONized gases library in C++. In *11th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*.

[Sutton and Gnoffo, 1998] Sutton, K. and Gnoffo, P. A. (1998). Multi-Component Diffusion with Application To Computational Aerothermodynamics. *7th AIAA / ASME Joint Thermophysics and Heat Transfer Conference*.

[van Leer, 1982] van Leer, B. (1982). Flux Vector Splitting for the Euler Equations. In *Eighth international conference on numerical methods in fluid dynamics*.