Adaptive Multilevel Discretizations for Computational Fluid Dynamics

Ralf Deiterding

Center for Advanced Computing Research
California Institute of Technology
ralf@cacr.caltech.edu

1. Introduction: AMR methods for finite volume schemes
2. Blockstructured AMR for hyperbolic problems
   - The Berger-Colella method
   - Adaptive detonation simulations
3. Blockstructured AMR for elliptic problems
   - Geometric multigrid
   - Case study for the Poisson equation
4. Implementation of AMR algorithms in AMROC
   - Parallelization
5. Outlook: Embedded boundaries methods
   - Blockstructured AMR within the Virtual Test Facility

Conservation Law: $\partial_t \mathbf{q}(\mathbf{x},t) + \nabla \cdot \mathbf{f}(\mathbf{q}(\mathbf{x},t)) = \mathbf{s}(\mathbf{q}(\mathbf{x},t)) + \nabla \cdot \mathbf{g}(\mathbf{q}(\mathbf{x},t), \nabla \mathbf{q}(\mathbf{x},t))\,, \quad \mathbf{x} \in \Omega \subset I\!R^d\,, \quad t > 0$

Integral form (Gauss's theorem):

$$\int_\Omega \mathbf{q}(\mathbf{x},t+\Delta t)\,d\mathbf{x} - \int_\Omega \mathbf{q}(\mathbf{x},t)\,d\mathbf{x} + \int_t^{t+\Delta t} \int_{\partial\Omega} \mathbf{f}(\mathbf{q}(\mathbf{o},t))\,d\mathbf{o}\,dt = \int_t^{t+\Delta t} \int_\Omega \mathbf{s}(\mathbf{q}(\mathbf{x},t))\,d\mathbf{x}\,dt + \int_t^{t+\Delta t} \int_{\partial\Omega} \mathbf{g}(\mathbf{q}(\mathbf{o},t), \nabla\mathbf{q}(\mathbf{o},t))\,d\mathbf{o}\,dt$$

Finite Volume Discretization in 1D

Integrate over volume $I_j = [x_j - \frac{1}{2}\Delta x, x_j + \frac{1}{2}\Delta x[=: [x_{j-1/2}, x_{j+1/2}[$.

Approximation $\mathbf{Q}_j(t) \approx \dfrac{1}{|I_j|} \displaystyle\int_{I_j} \mathbf{q}(\mathbf{x},t)\,d\mathbf{x}$ $\qquad$ $\mathbf{s}(\mathbf{Q}_j(t)) \approx \dfrac{1}{|I_j|} \displaystyle\int_{I_j} \mathbf{s}(\mathbf{q}(\mathbf{x},t))\,d\mathbf{x}$, and numerical fluxes

$$\mathbf{F}\left(\mathbf{Q}_j(t), \mathbf{Q}_{j+1}(t)\right) \approx \mathbf{f}(\mathbf{q}(x_{j+1/2},t)), \quad \mathbf{G}\left(\mathbf{Q}_j(t), \mathbf{Q}_{j+1}(t)\right) \approx \mathbf{g}(\mathbf{q}(x_{j+1/2},t), \nabla\mathbf{q}(x_{j+1/2},t))$$

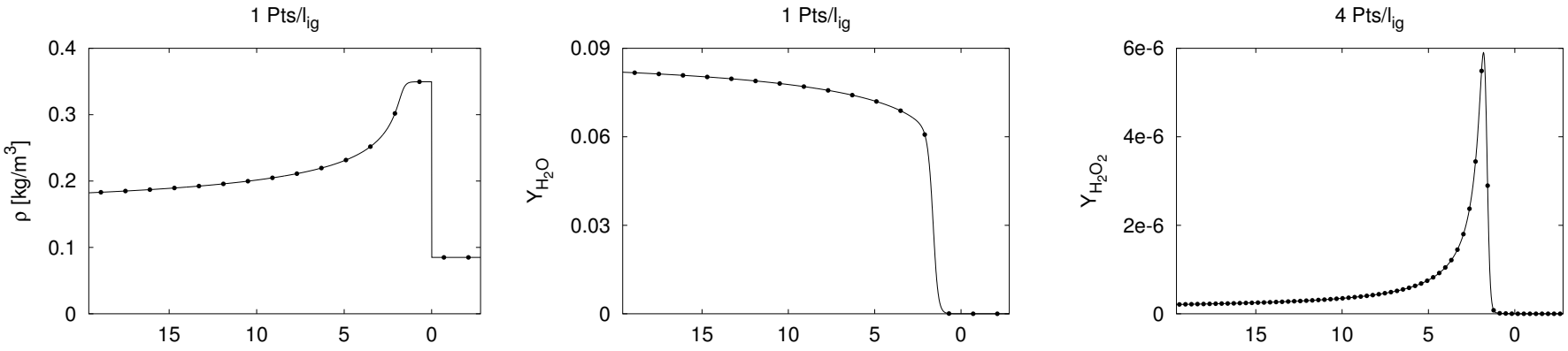
yield

$$\mathbf{Q}_j(t_{n+1}) = \mathbf{Q}_j(t_n) - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \left[\mathbf{F}\left(\mathbf{Q}_j(t), \mathbf{Q}_{j+1}(t)\right) - \mathbf{F}\left(\mathbf{Q}_{j-1}(t), \mathbf{Q}_j(t)\right)\right] dt -$$

$$\frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \left[\mathbf{G}\left(\mathbf{Q}_j(t), \mathbf{Q}_{j+1}(t)\right) - \mathbf{G}\left(\mathbf{Q}_{j-1}(t), \mathbf{Q}_j(t)\right)\right] dt + \int_{t_n}^{t_{n+1}} \mathbf{s}(\mathbf{Q}_j(t))\,dt$$

Motivation for Adaptive Mesh Refinement: Detonations Waves

1. Extremly high spatial resolution in reaction zone necessary. Discretization of an exact ZND detonation:



   Minimal spatial resolution: $7 - 8\,\mathrm{Pts}/l_{ig} \longrightarrow \Delta x \approx 0.2 - 0.175\,\mathrm{mm}$

   Uniform grids for typical geometries: $> 10^7\,\mathrm{Pts}$ in 2D, $> 10^9\,\mathrm{Pts}$ in 3D $\longrightarrow$ Self-adaptive finite volume method (AMR)

2. Problem size even with AMR in 3D enormous $\longrightarrow$ parallelization for massivly parallel systems with distributed memory

Elements of Parallel AMR Methods

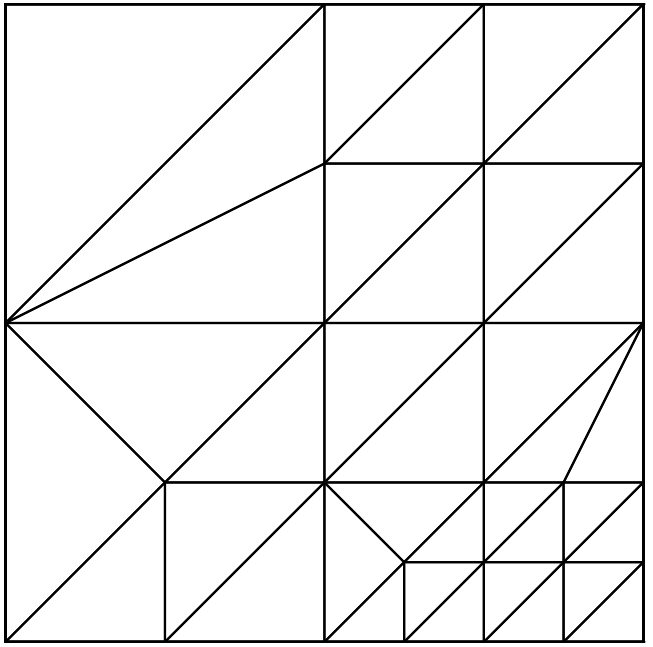
- Base grid
- Solver
- Error indicators
- Grid manipulation
- Interpolation (restriction and prolongation)
- Load-balancing

Unstructured Approach

- Coarse cells replaced by finer ones
- Global time-step
- Cell-based data structures
- Neighborhoods have to stored

+ Geometric flexible
+ No hanging nodes
+ Easy to implement

- Higher order difficult to achieve
- Cell aspect ratio must be considered
- Fragmented data
- Cache-reuse / vectorizaton nearly impossible
- Complex load-balancing
- Complex synchronization



Quad-tree-based Structured Approach

- Block-based data of equal size
- Time-step refinement
- Block stored in a quad-tree
- Global index coordinate system
- Neighborhoods need not be stored

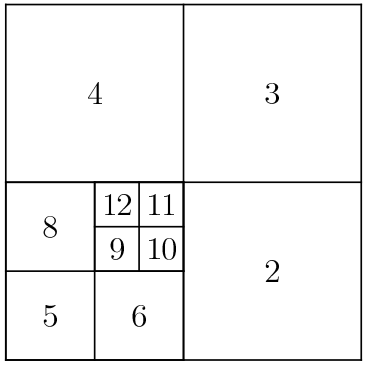
\+ Numerical scheme only for single regular block necessary

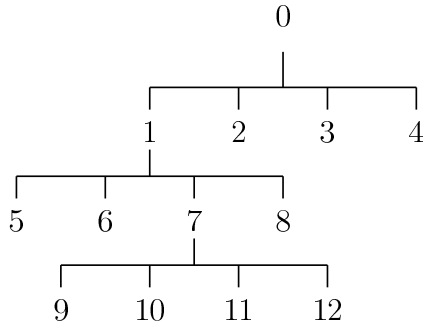
\+ Easy to implement

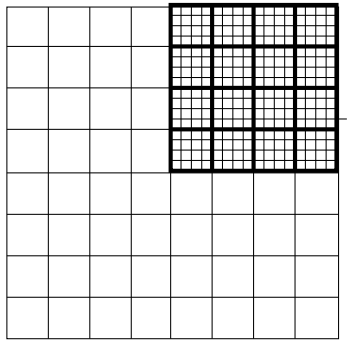
\+ Simple load-balancing

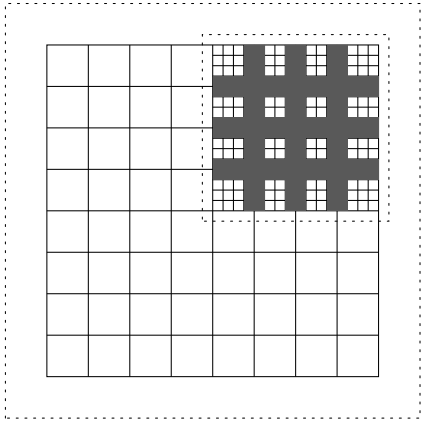
\+ Parent/Child relations according to tree

+/- Cache-reuse / vectorization only in data block









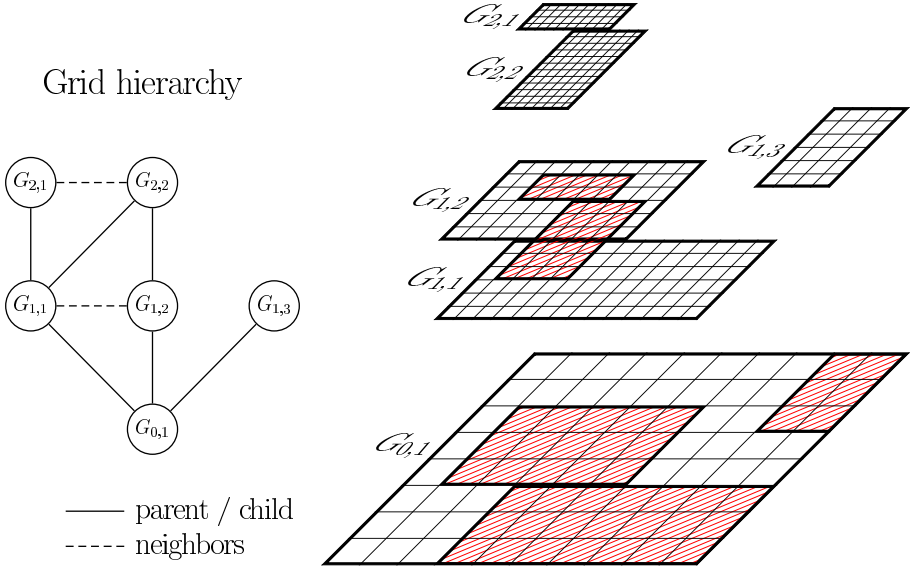
Wasted boundary space in a quad-tree.

Blockstructured Adaptive Mesh Refinement (SAMR)

- Refined block overlay coarser ones
- Time-step refinement
- Block-based data structures
- Global index coordinate system

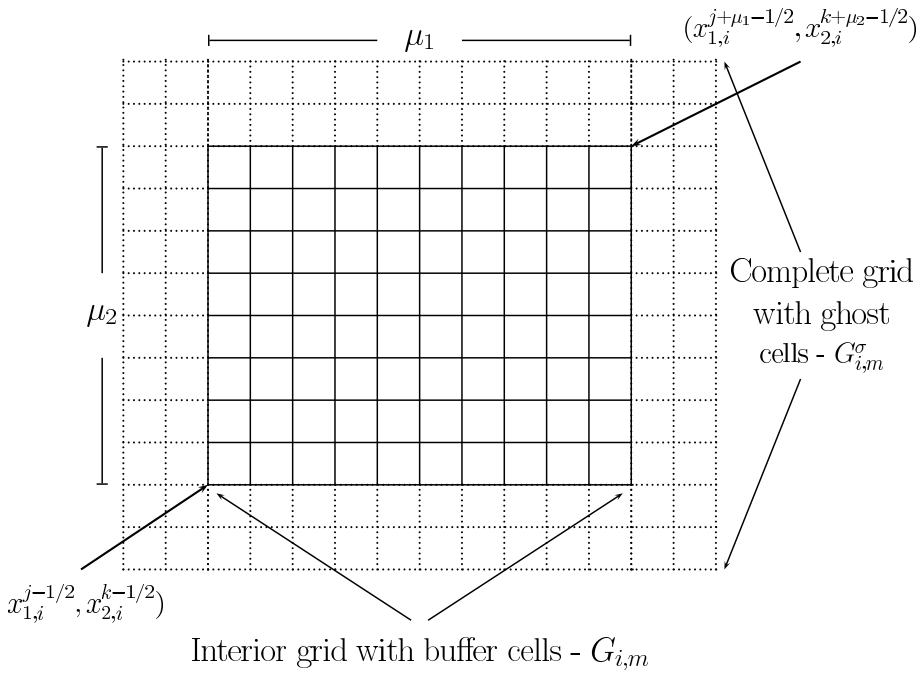
+ Numerical scheme only for single regular block necessary
+ Efficient cache-reuse / vectorization possible
+ Simple load-balancing
+ Minimal synchronization overhead

- Cells without mark are refined
- Cluster-algorithm necessary

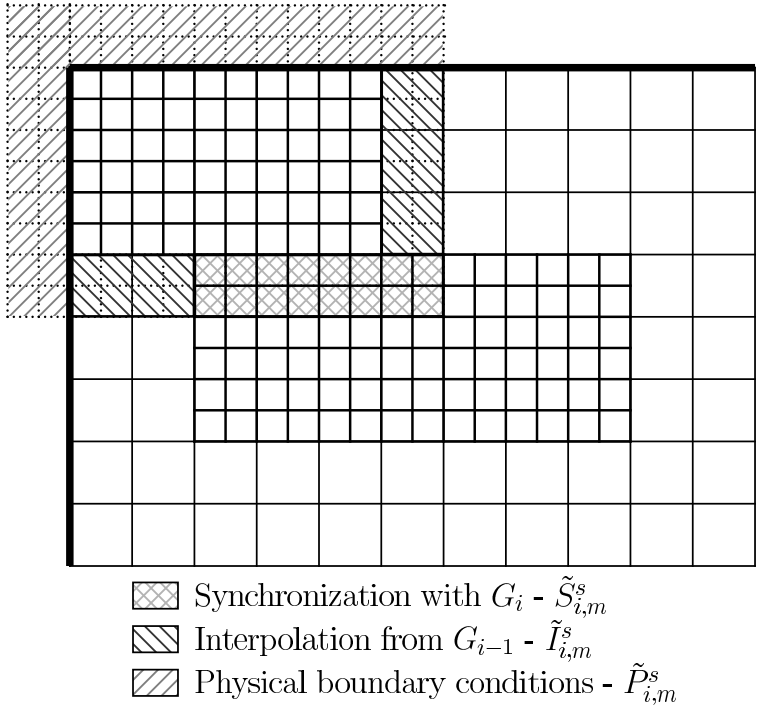


- Hanging nodes unavoidable
- Difficult to implement

A Refinement Grid



Parts of A Refinement Grid

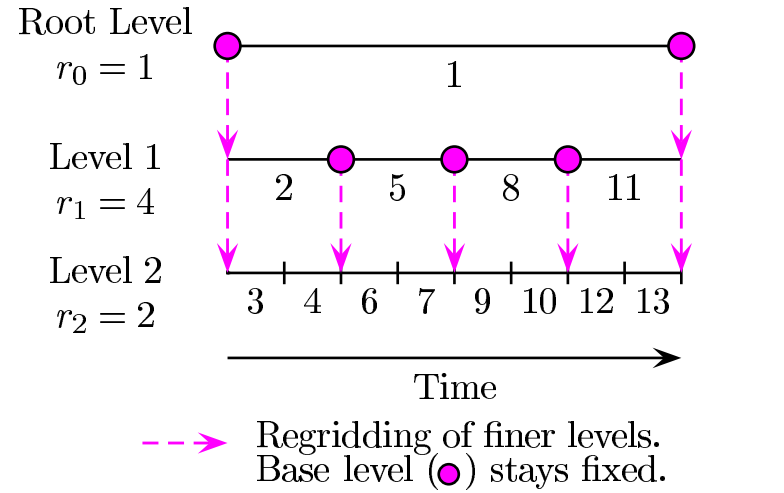


Domain of level l: $\mathcal{G}_l := \bigcup_{m=1}^{M_l} \mathcal{G}_{l,m}$ with $\mathcal{G}_{l,m} \cap \mathcal{G}_{l,n} = \emptyset$ für $m \neq n$

Refinements are properly nested: $\mathcal{G}_l^{r_l} \cap \mathcal{G}_{l-1} = \mathcal{G}_l^{r_l} \cap \mathcal{G}_0$

The Recursive Berger-Colella Algorithm *

```
AdvanceLevel(l)
   Repeat r_l times
        Set ghost cells of Q_l(t)
        If time to regrid?
             Regrid(l)
        UpdateLevel(l)
        If level l+1 exists?
             Set ghost cells of Q_l(t + Δt_l)
             AdvanceLevel(l + 1)
             Average Q_{l+1}(t + Δt_l) onto Q_l(t + Δt_l)
             Correct Q_l(t + Δt_l) with δF_{l+1}
        t := t + Δt_l
```




```
Regrid(l) - Regrid all levels ι > l
   For ι = l_c Downto l Do
        Flag N^ι according to Q_ι(t)
        If level ι + 1 exists?
             Flag N^ι below G̃_{ι+2}
        Flag buffer zone on N^ι
        Generate G̃_{ι+1} from N^ι
   EnsureNesting G̃_{l+1}, ..., G̃_{l_c+1}
   Recompose(l)
```

```
Start - Start integration on level 0
   l = 0,  r_0 = 1
   AdvanceLevel(l)
```

Refinement factor on level l: $r_l = \Delta t_{l-1}/\Delta t_l$

∗ M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1988.

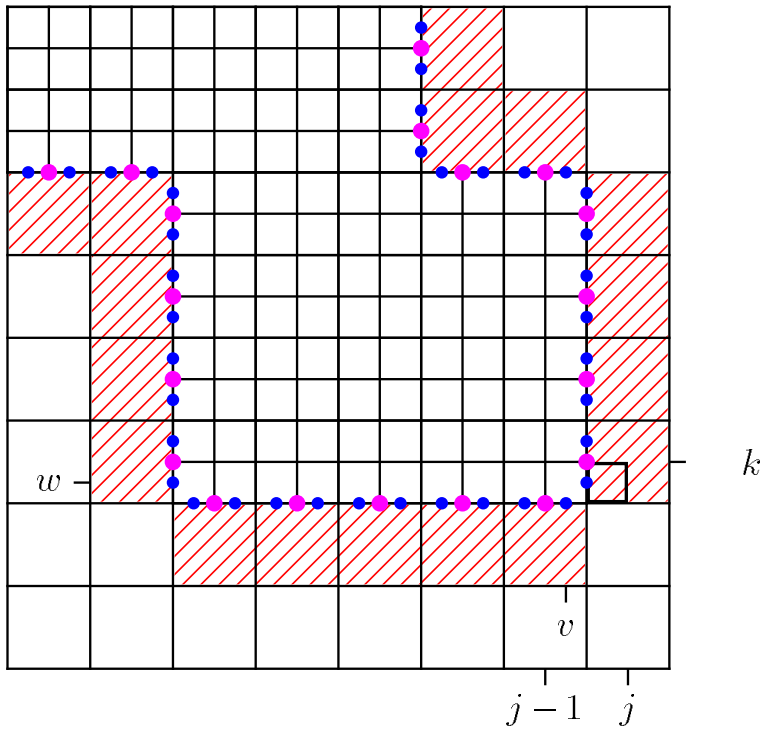
Conservative Correction

Example: Cell j, k

$$\check{\mathbf{Q}}^l_{jk}(t+\Delta t_l) = \mathbf{Q}^l_{jk}(t) - \frac{\Delta t_l}{\Delta x_{1,l}} \left(\mathbf{F}^{1,l}_{j+\frac{1}{2},k} - \frac{1}{r^2_{l+1}} \sum_{\kappa=0}^{r_{l+1}-1} \sum_{\iota=0}^{r_{l+1}-1} \mathbf{F}^{1,l+1}_{v+\frac{1}{2},w+\iota}(t+\kappa\Delta t_{l+1}) \right) - \frac{\Delta t_l}{\Delta x_{2,l}} \left(\mathbf{F}^{2,l}_{j,k+\frac{1}{2}} - \mathbf{F}^{2,l}_{j,k-\frac{1}{2}} \right)$$

Correction pass:

1. $\delta\mathbf{F}^{1,l+1}_{j-\frac{1}{2},k} := -\mathbf{F}^{1,l}_{j-\frac{1}{2},k}$
2. $\delta\mathbf{F}^{1,l+1}_{j-\frac{1}{2},k} := \delta\mathbf{F}^{1,l+1}_{j-\frac{1}{2},k} + \frac{1}{r_{l+1}} \sum_{\iota=0}^{r_{l+1}-1} \mathbf{F}^{1,i+1}_{v+\frac{1}{2},w+\iota}(t+\kappa\Delta t_{l+1})$
3. $\check{\mathbf{Q}}^l_{jk}(t+\Delta t_l) := \mathbf{Q}^l_{jk}(t+\Delta t_l) + \frac{\Delta t_l}{\Delta x_{1,l}} \delta\mathbf{F}^{1,l+1}_{j-\frac{1}{2},k}$



Euler Equations

$$\partial_t \rho_i + \nabla \cdot (\rho \mathbf{u}) = \dot{\omega}_i \text{ für } i = 1, \ldots, K$$

$$\begin{array}{rcrcl} \partial_t (\rho \mathbf{u}) & + & \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p & = & 0 \\ \partial_t (\rho E) & + & \nabla \cdot ((\rho E + p)\mathbf{u}) & = & 0 \end{array}$$

Ideal gas law and Dalton's law for gas-mixtures:

$$p(\rho_1, \ldots, \rho_K, T) = \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \rho_i \frac{\mathcal{R}}{W_i} T = \rho \frac{\mathcal{R}}{W} T \quad \text{with} \quad \sum_{i=1}^{K} \rho_i = \rho \quad \text{and} \quad Y_i = \frac{\rho_i}{\rho} .$$

Caloric equation: $h(Y_1, \ldots, Y_K, T) = \sum_{i=1}^{K} Y_i h_i(T)$ with $h_i(T) = h_i^0 + \int_0^T c_{pi}(s) ds$

Computation of $T = T(\rho_1, \ldots, \rho_K, e)$ from implicit equation $\sum_{i=1}^{K} \rho_i \, h_i(T) - \mathcal{R} T \sum_{i=1}^{K} \frac{\rho_i}{W_i} - \rho e = 0$

for *thermally perfect* gases with $\gamma_i(T) = c_{pi}(T)/c_{vi}(T)$.

Arrhenius-Kinetics: $\dot{\omega}_i = \sum_{j=1}^{M} (\nu_{ji}^r - \nu_{ji}^f) \left[k_j^f \prod_{n=1}^{K} \left(\frac{\rho_n}{W_n} \right)^{\nu_{jn}^f} - k_j^r \prod_{n=1}^{K} \left(\frac{\rho_n}{W_n} \right)^{\nu_{jn}^r} \right] \quad i = 1, \ldots, K$

- Parsing of mechanisms with Chemkin-II
- Evalutation of $\dot{\omega}_i$ with automatically generated optimized Fortran-77 functions in the line of Chemkin-II

R. Deiterding, Parallel adaptive simulation of multi-dimensional detonation structures, PhD thesis, Brandenburgische Technische Universität Cottbus, Sep 2003. dissertation.de - Verlag im Internet GmbH, Berlin, also available at http://www.dissertation.de

Fractional Step Methods

Solve homogeneous PDE and ODE successively!

$$\begin{aligned}
\mathcal{H}^{(\Delta t)}: &\quad \partial_t \mathbf{q} + \nabla \cdot \mathbf{f}(\mathbf{q}) = 0\,, \quad \text{IC: } \mathbf{Q}(t_m) \overset{\Delta t}{\Longrightarrow} \tilde{\mathbf{Q}} \\
\mathcal{S}^{(\Delta t)}: &\quad \partial_t \mathbf{q} = \mathbf{s}(\mathbf{q})\,, \quad \text{IC: } \tilde{\mathbf{Q}} \overset{\Delta t}{\Longrightarrow} \mathbf{Q}(t_m + \Delta t)
\end{aligned}$$

1st-order: $\mathbf{Q}(t_m + \Delta t) = \mathcal{S}^{(\Delta t)}\mathcal{H}^{(\Delta t)}(\mathbf{Q}(t_m))$, 2nd-order: $\mathbf{Q}(t_m + \Delta t) = \mathcal{S}^{(\frac{1}{2}\Delta t)}\mathcal{H}^{(\Delta t)}\mathcal{S}^{(\frac{1}{2}\Delta t)}(\mathbf{Q}(t_m))$

ODE integration in $\mathcal{S}^{(\cdot)}$ for Euler equations with chemical reaction

- Standard implicit or semi-implicit ODE-solver subcycles within each cell
- ρ, e, $\mathbf{u}$ remain unchanged!

$$\partial_t \rho_i = W_i \dot{\omega}_i(\rho_1, \ldots, \rho_K, T) \qquad i = 1, \ldots, K$$

Use Newton or bisection method to compute T iteratively.

Dimensional splitting for $\mathcal{H}^{(\cdot)}$:

$$\begin{aligned}
\mathcal{X}_1^{(\Delta t)}: &\quad \partial_t \mathbf{q} + \partial_{x_1} \mathbf{f}_1(\mathbf{q}) = 0\,, \quad \text{IC: } \mathbf{Q}(t_m) \overset{\Delta t}{\Longrightarrow} \tilde{\mathbf{Q}}^{1/2} \\
\mathcal{X}_2^{(\Delta t)}: &\quad \partial_t \mathbf{q} + \partial_{x_2} \mathbf{f}_2(\mathbf{q}) = 0\,, \quad \text{IC: } \tilde{\mathbf{Q}}^{1/2} \overset{\Delta t}{\Longrightarrow} \tilde{\mathbf{Q}}
\end{aligned}$$

Conservative quasi-1D finite volume methods:

$$\begin{aligned}
\mathcal{X}_1^{(\Delta t)}: &\quad \tilde{\mathbf{Q}}_{jk}^{m+\frac{1}{2}} = \mathbf{Q}_{jk}^m - \frac{\Delta t}{\Delta x_1}\left[\mathbf{F}^1(\mathbf{Q}_{j-s+1,k}^m, \ldots, \mathbf{Q}_{j+s,k}^m) - \mathbf{F}^1(\mathbf{Q}_{j-s,k}^m, \ldots, \mathbf{Q}_{j+s-1,k}^m)\right]\,, \\
\mathcal{X}_2^{(\Delta t)}: &\quad \tilde{\mathbf{Q}}_{jk}^{m+1} = \tilde{\mathbf{Q}}_{jk}^{m+\frac{1}{2}} - \frac{\Delta t}{\Delta x_2}\left[\mathbf{F}^2(\tilde{\mathbf{Q}}_{j,k-s+1}^{m+\frac{1}{2}}, \ldots, \tilde{\mathbf{Q}}_{j,k+s}^{m+\frac{1}{2}}) - \mathbf{F}^2(\tilde{\mathbf{Q}}_{j,k-s}^{m+\frac{1}{2}}, \ldots, \tilde{\mathbf{Q}}_{j,k+s-1}^{m+\frac{1}{2}})\right]
\end{aligned}$$

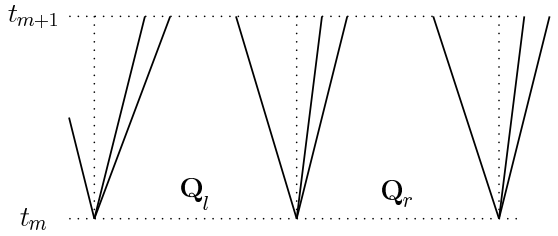
1st-order: $\mathcal{H}^{(\Delta t)} = \mathcal{X}_2^{(\Delta t)}\mathcal{X}_1^{(\Delta t)}(\mathbf{Q}(t_m))$, 2nd-order: $\mathcal{H}^{(\Delta t)} = \mathcal{X}_1^{(\frac{1}{2}\Delta t)}\mathcal{X}_2^{(\Delta t)}\mathcal{X}_1^{(\frac{1}{2}\Delta t)}(\mathbf{Q}(t_m))$

Roe's Approximate Riemann Solver

Appropriate matrix $\mathbf{A}(\hat{\mathbf{Q}}) = \mathbf{R}(\hat{\mathbf{Q}})\mathbf{\Lambda}(\hat{\mathbf{Q}})\mathbf{R}^{-1}(\hat{\mathbf{Q}})$

Wave decomposition: $\Delta\mathbf{Q} = \mathbf{Q}_r - \mathbf{Q}_l = \sum_m a_m \, \hat{\mathbf{r}}_m$

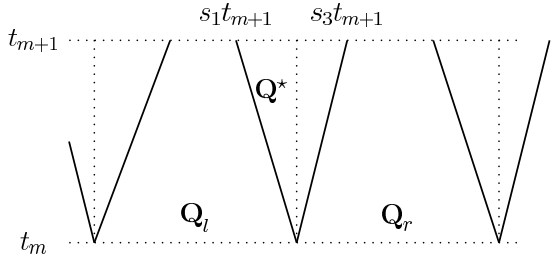
$$\begin{aligned}\mathbf{F}(\mathbf{Q}_l, \mathbf{Q}_r) &= \mathbf{f}(\mathbf{Q}_l) - \sum_{\hat{\lambda}_m > 0} \hat{\lambda}_m \, a_m \, \hat{\mathbf{r}}_m = \mathbf{f}(\mathbf{Q}_r) + \sum_{\hat{\lambda}_m < 0} \hat{\lambda}_m \, a_m \, \hat{\mathbf{r}}_m \\ &= \frac{1}{2}\left(\mathbf{f}(\mathbf{Q}_l) + \mathbf{f}(\mathbf{Q}_r) - \sum_m |\hat{\lambda}_m| \, a_m \, \hat{\mathbf{r}}_m \right) .\end{aligned}$$



Insert appropriate average state based on $\hat{x} = \dfrac{x_l\sqrt{\rho_l} + x_r\sqrt{\rho_r}}{\sqrt{\rho_l} + \sqrt{\rho_r}}$ into $\mathbf{A}$ as $\mathbf{A}(\hat{\mathbf{Q}})$.

Harten-Lax-Van Leer (HLL) Approixmate Riemann Solver

$$\mathbf{F}_{HLL}(\mathbf{Q}_l, \mathbf{Q}_r) = \begin{cases} \mathbf{f}(\mathbf{Q}_l) \, , & 0 < s_1 \, , \\ \dfrac{s_3\mathbf{f}(\mathbf{Q}_l) - s_1\mathbf{f}(\mathbf{Q}_R) + s_1 s_3(\mathbf{Q}_r - \mathbf{Q}_l)}{s_3 - s_1} \, , & s_1 \le 0 \le s_3 \, , \\ \mathbf{f}(\mathbf{Q}_r) \, , & 0 > s_3 \, , \end{cases}$$



$$s_1 = \min(u_{1,l} - c_l, u_{1,r} - c_r) \, , \quad s_3 = \max(u_{1,l} + c_l, u_{1,r} + c_r)$$

Switch from Roe to HLL scheme near vacuum state to avoid unphysical values.

$$\bar{\mathbf{Q}}(x,t) = \begin{cases} \mathbf{Q}_l \, , & x < s_1 \, t \\ \mathbf{Q}^\star \, , & s_1 \, t \le x \le s_3 \, t \\ \mathbf{Q}_r \, , & x > s_3 \, t \end{cases}$$

A Robust and Reliable Roe-type Scheme

(S1) Calculate standard Roe-averages $\hat{\rho}$, $\hat{u}_n$, $\hat{H}$, $\hat{Y}_i$, $\hat{T}$.

(S2) Compute $\hat{\gamma} := \hat{c}_p / \hat{c}_v$ with $\hat{c}_{\{p/v\}i} = \dfrac{1}{T_r - T_l} \displaystyle\int_{T_l}^{T_r} c_{\{p,v\}i}(\tau)\, d\tau$.

(S3) Calculate $\hat{\phi}_i := (\hat{\gamma} - 1) \left(\frac{\hat{\mathbf{u}}^2}{2} - \hat{h}_i \right) + \hat{\gamma}\, R_i\, \hat{T}$ with standard Roe-averages $\hat{e}_i$ or $\hat{h}_i$.

(S4) Calculate $\hat{c} := \left(\sum_{i=1}^{K} \hat{Y}_i\, \hat{\phi}_i - (\hat{\gamma} - 1)\hat{\mathbf{u}}^2 + (\hat{\gamma} - 1)\hat{H} \right)^{1/2}$.

(S5) Use $\Delta \mathbf{Q} = \mathbf{Q}_r - \mathbf{Q}_l$ and Δp to compute the wave strengths a_m.

(S6) Calculate $\mathbf{W}_1 = a_1 \hat{\mathbf{r}}_1$, $\mathbf{W}_2 = \displaystyle\sum_{\iota=2}^{K+d} a_\iota \hat{\mathbf{r}}_\iota$, $\mathbf{W}_3 = a_{K+d+1} \hat{\mathbf{r}}_{K+d+1}$.

(S7) Evaluate $s_1 = \hat{u}_1 - \hat{c}$, $s_2 = \hat{u}_1$, $s_3 = \hat{u}_1 + \hat{c}$.

(S8) Evaluate $\rho^\star_{l/r}$, $u^\star_{1,l/r}$, $e^\star_{l/r}$, $c^\star_{1,l/r}$ from $\mathbf{Q}^\star_l = \mathbf{Q}_l + \mathbf{W}_1$ and $\mathbf{Q}^\star_r = \mathbf{Q}_r - \mathbf{W}_3$.

(S9) If $\rho^\star_{l/r} \le 0$ or $e^\star_{l/r} \le 0$ use $\mathbf{F}_{HLL}(\mathbf{Q}_l, \mathbf{Q}_r)$ and go to (S12).

(S10) Entropy correction: Evaluate $|\tilde{s}_\iota|$.

$$\mathbf{F}_{Roe}(\mathbf{Q}_l, \mathbf{Q}_r) = \tfrac{1}{2} \left(\mathbf{f}(\mathbf{Q}_l) + \mathbf{f}(\mathbf{Q}_r) - \sum_{\iota=1}^{3} |\tilde{s}_\iota| \mathbf{W}_\iota \right)$$

(S11) Positivity correction: Replace $\mathbf{F}_i$ by

$$\mathbf{F}^\star_i = \mathbf{F}_\rho \cdot \begin{cases} Y^l_i\,, & \mathbf{F}_\rho \ge 0\,, \\ Y^r_i\,, & \mathbf{F}_\rho < 0\,. \end{cases}$$

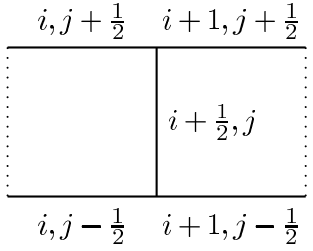
(S12) Evaluate maximal signal speed by $S = \max(|s_1|, |s_3|)$.

Possible Entropy corrections

1. Replace $|s_\iota|$ by $|\tilde{s}_\iota|$ only if $s_\iota(\mathbf{Q}_l) < 0 < s_\iota(\mathbf{Q}_r)$.
2. $|\tilde{s}_\iota| = \begin{cases} |s_\iota| & \text{if} |s_\iota| \ge 2\eta \\ \frac{|s_\iota^2|}{4\eta} + \eta & \text{otherwise} \end{cases}$

   $\eta = \frac{1}{2} \max_\iota \left\{ |s_\iota(\mathbf{Q}_r) - s_\iota(\mathbf{Q}_l)| \right\}$

2D modification of entropy correction:

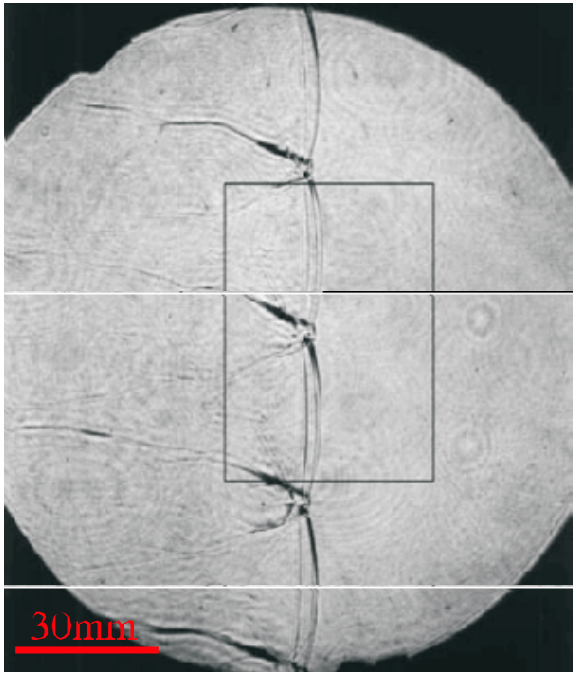


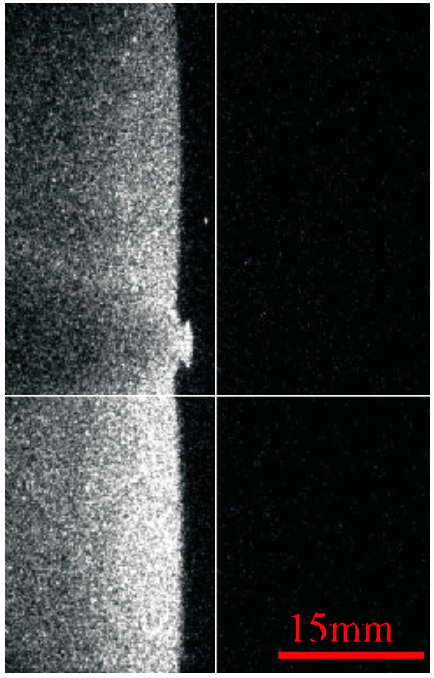
$$\tilde{\eta}_{i+1/2,j} = \max \left\{ \eta_{i+1/2,j}, \eta_{i,j-1/2}, \eta_{i,j+1/2}, \eta_{i+1,j-1/2}, \eta_{i+1,j+1/2} \right\}$$

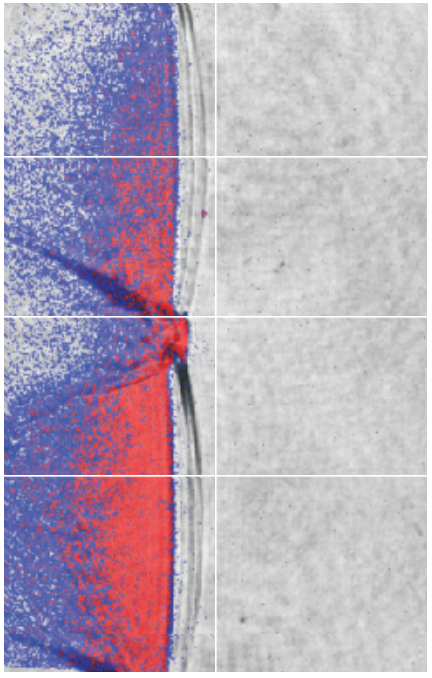
R. Sanders, E. Morano, and M.-C. Druguett. Multidimensional dissipation for upwind schemes: stability and applications to gas dynamics. *J. Comput. Phys.*, 145:511–537, 1998.

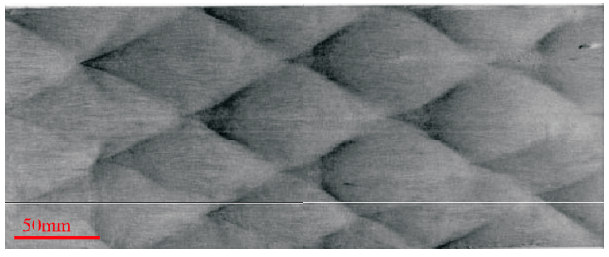
to avoid the carbuncle phenomenon at strong shocks.

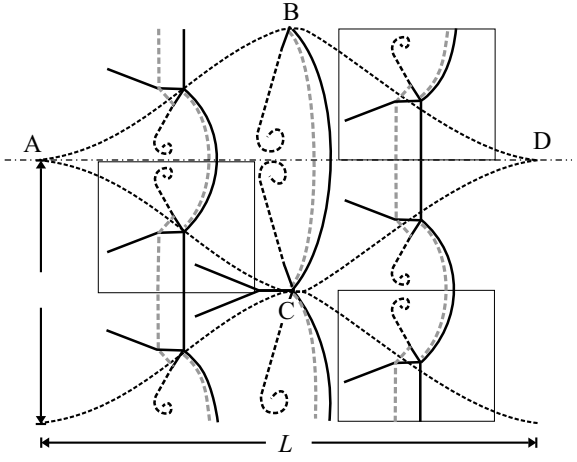
Transverse Detonation Structure - Regular Instability

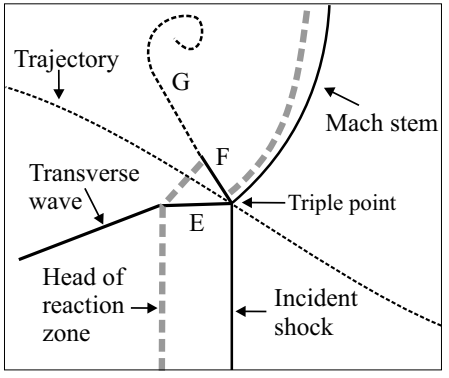






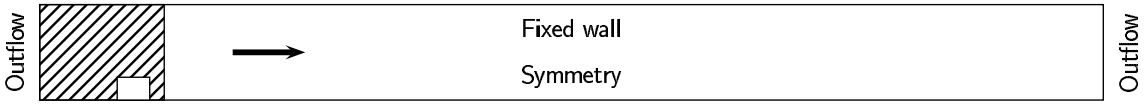






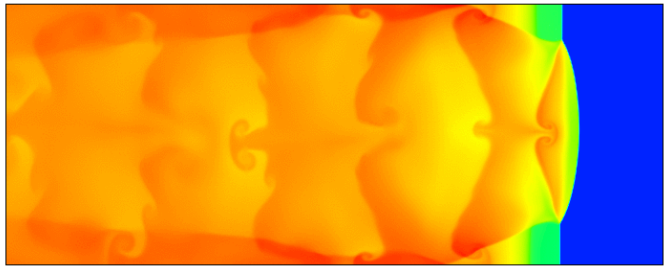
E: Reflected shock. F: Slip line. G: Diffusive extension of slip line.

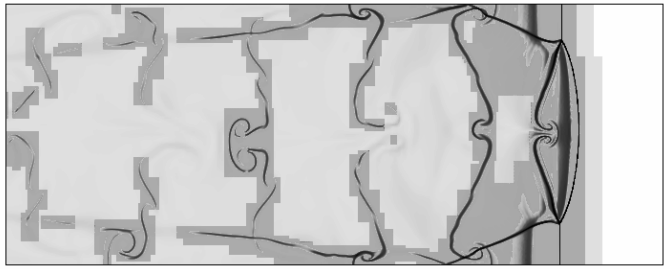
Cellular Structure Simulation - 2D



Domain $62\,\mathrm{cm} \times 3\,\mathrm{cm}$. CJ-detonation placed at $x = 4.6\,\mathrm{cm}$. Unreacted pocket $0.3\,\mathrm{cm}$ behind detonation front.

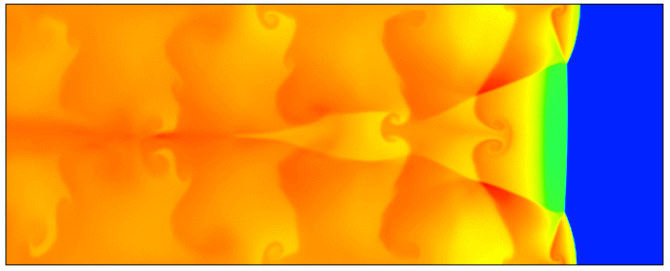
E. S. Oran, J. W. Weber, E. I. Stefaniw, M. H. Lefebvre, and J. D. Anderson. A numerical study of a two-dimensional h_2-o_2-ar detonation using a detailed chemical reaction model. *J. Combustion and Flame*, 113:147–163, 1998.

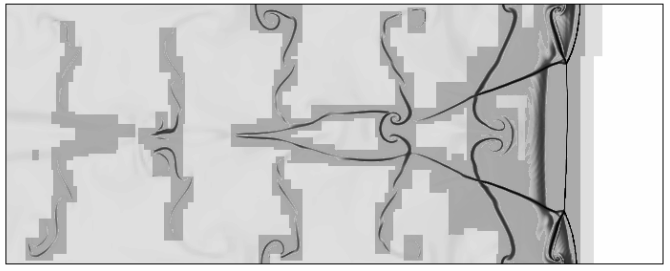




- Adaption criteria:
  1. Scaled gradients of ρ and p
  2. Error estimation in Y_i by Richardson extrapolation

$$\mathbf{Q}^n \overset{\Delta t, \Delta x}{\Longrightarrow} \tilde{\mathbf{Q}} \quad \text{and} \quad \mathbf{Q}^{n-1} \overset{2\Delta t, 2\Delta x}{\Longrightarrow} \hat{\mathbf{Q}} \qquad \tau = \frac{|\tilde{\mathbf{Q}} - \hat{\mathbf{Q}}|}{2^{d+1} - 2}$$



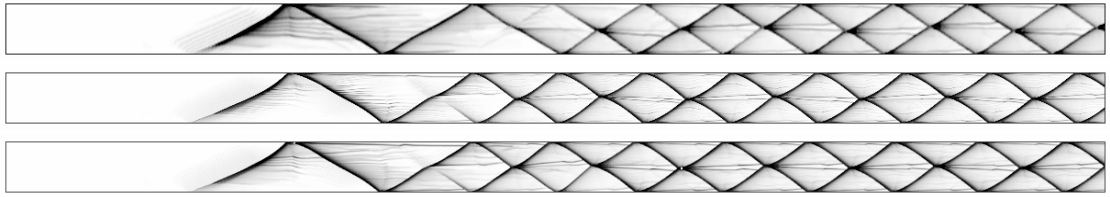


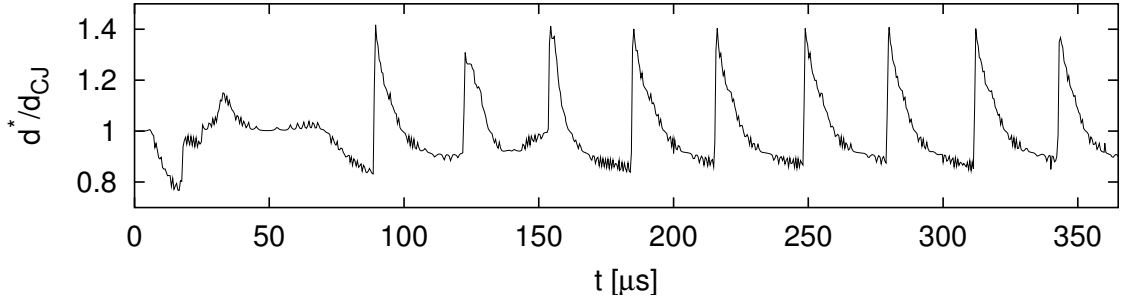
- Quasi-stationary. Unburned gas flows in with CJ velocity.
- Coarse grid shifted through domain with CJ velocity to obtain triple point tracks.

$$\max_t |\omega| \quad \text{with} \quad \omega = \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}$$

- 11.2 to $44.8\,\mathrm{Pts}/l_{ig}$. 2 refinement levels (2,4) or (4,4).
- $36\,\mathrm{k}$ instead $128\,\mathrm{k}$ $(11.2\,\mathrm{Pts}/l_{ig})$ to $450\,\mathrm{k}$ cells instead of $2.0\,\mathrm{M}$ $(44.8\,\mathrm{Pts}/l_{ig})$.

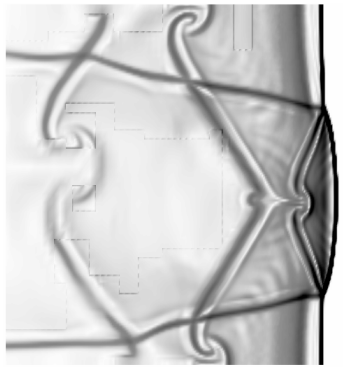
Grid Comparison



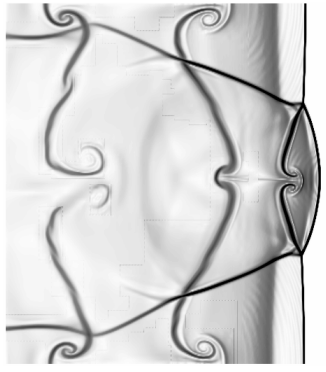


Task	Setup 1		Setup 2		Setup 3	
	s	%	s	%	s	%
Fluid dynamics	10049	31.5	72216	31.0	366496	33.5
Chemical kinetics	11347	35.5	77435	33.2	403006	36.8
Boundary setting	9776	30.6	78922	33.9	293798	26.8
Recomposition	595	1.9	3685	1.6	26392	2.4
Misc.	158	0.5	719	0.3	5558	0.5
Total	**31925**	8.9 h	**232977**	2.7 d	**1095250**	12.8 d

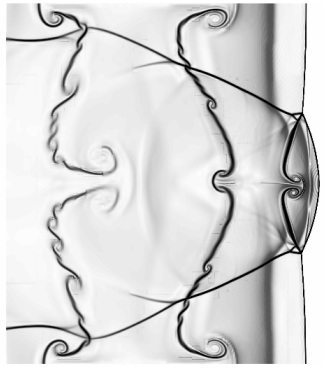
PC-Cluster of 7 Pentium III-850MHz-PCs connected with 1 GHz Myrinet.



$11.2\,\mathrm{Pts}/l_{ig}$

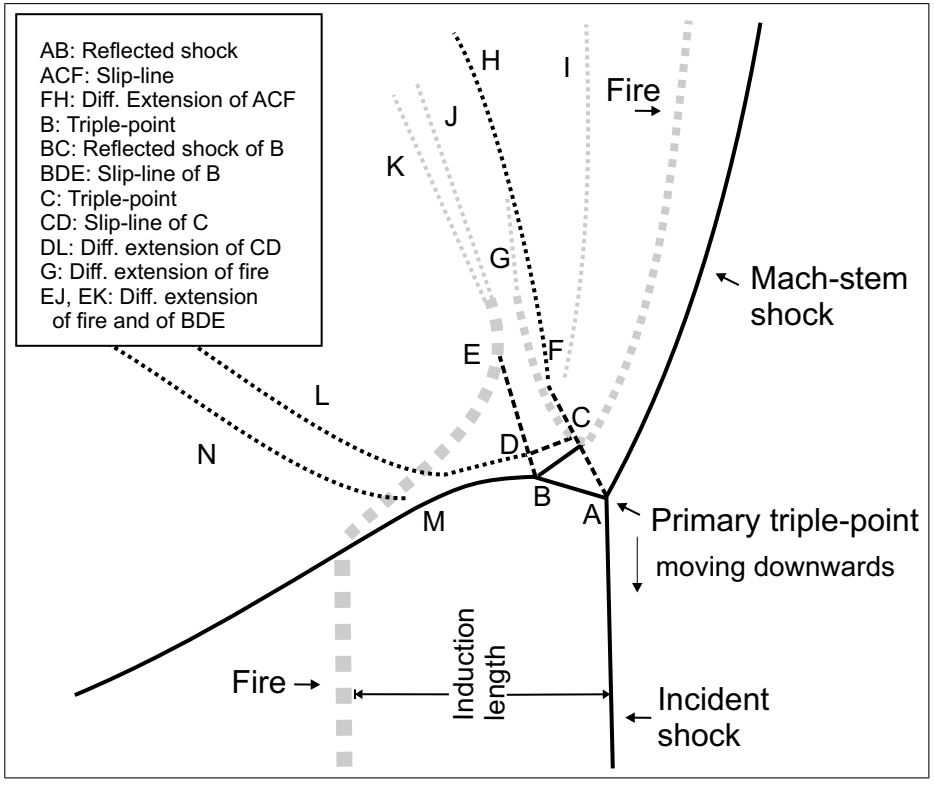


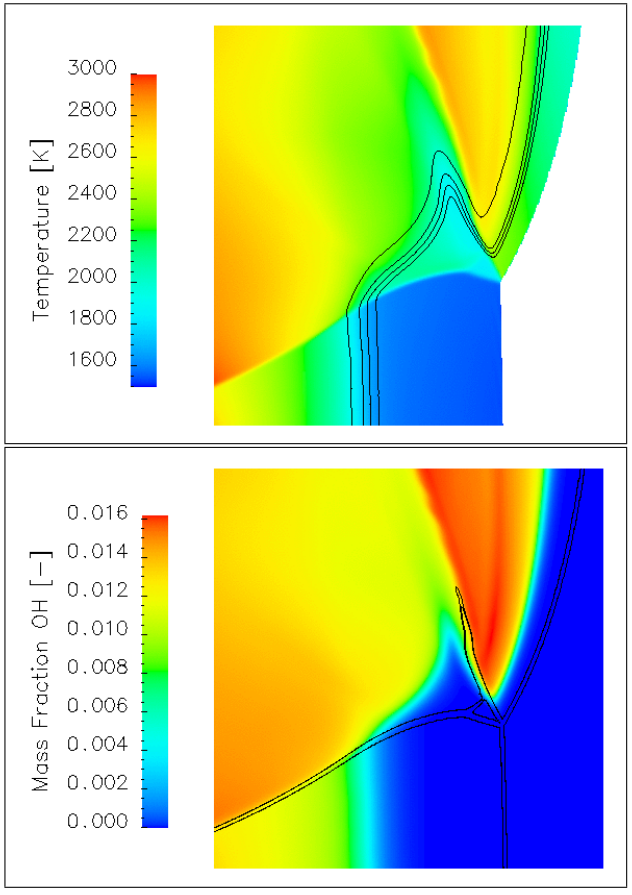
$22.4\,\mathrm{Pts}/l_{ig}$



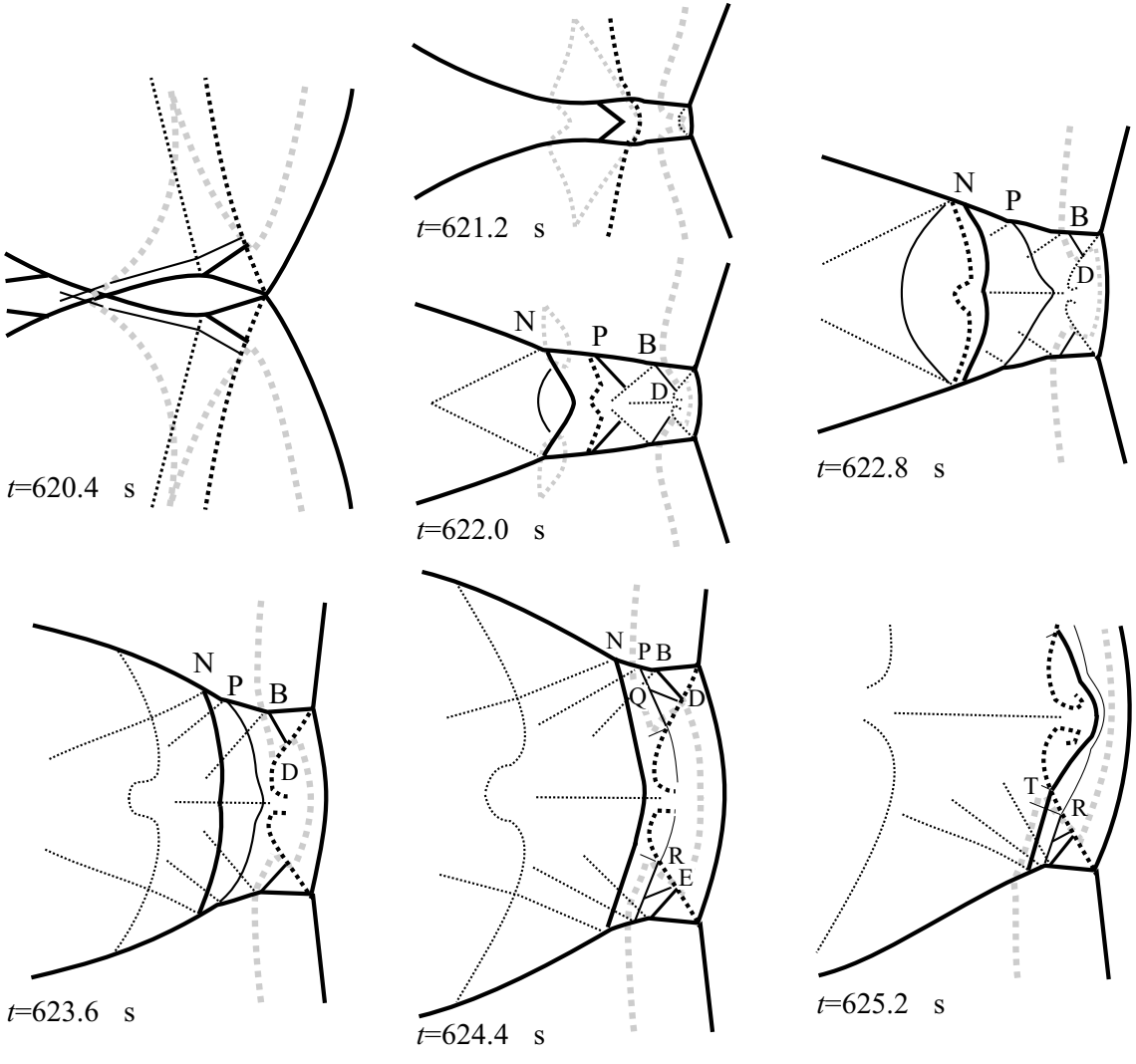
$44.8\,\mathrm{Pts}/l_{ig}$

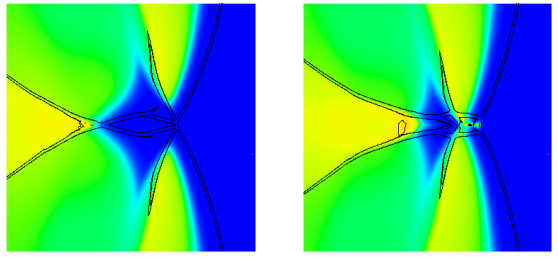
Flow Around a Triple-point



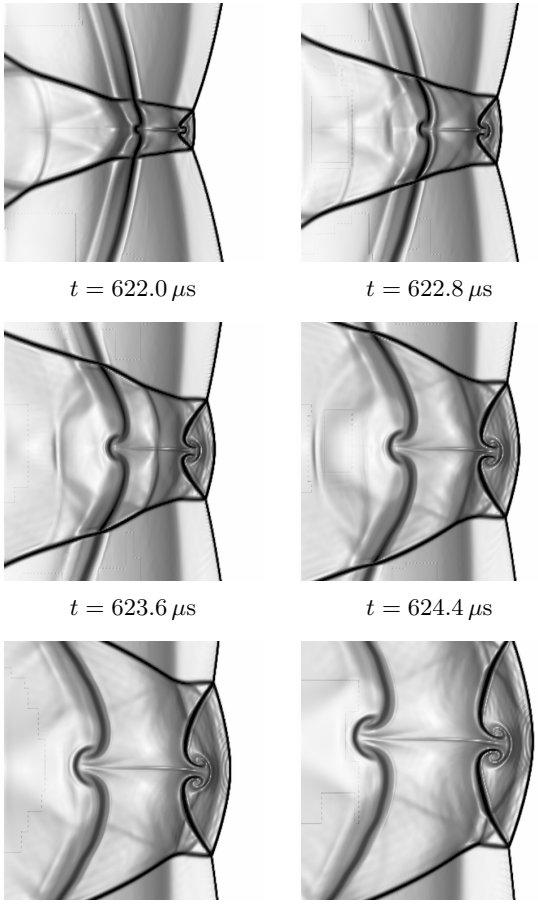


Collision of Triple Points and Development of the Double Mach Structure





$t = 620.4\,\mu s$ $t = 621.2\mu s$

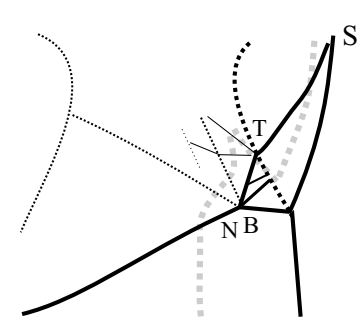


$t = 622.0\,\mu s$ $t = 622.8\,\mu s$

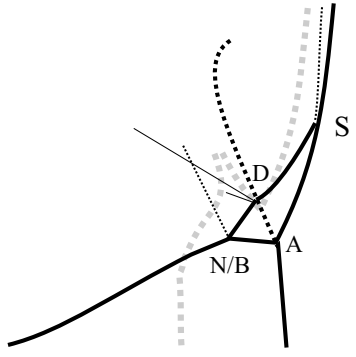
$t = 623.6\,\mu s$ $t = 624.4\,\mu s$

$t = 625.2\,\mu s$ $t = 626.0\,\mu s$

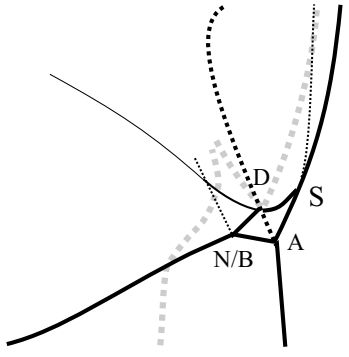
Development of the Triple Mach Structure Out of the Double Mach Structure Before the Next Collision



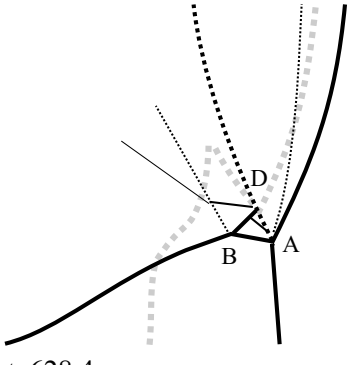
t=626.0 s



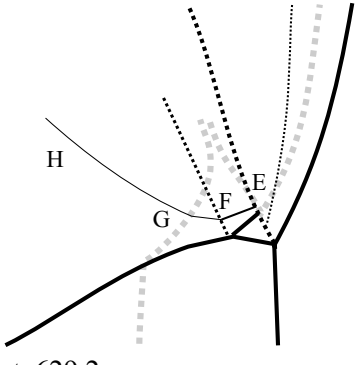
t=626.8 s



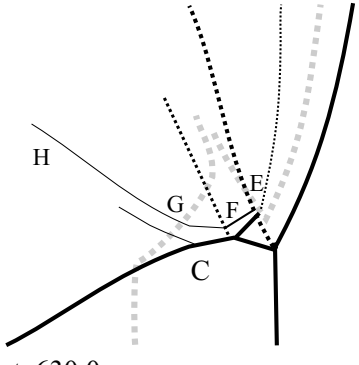
t=627.6 s



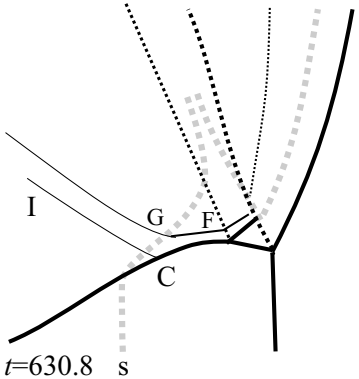
t=628.4 s



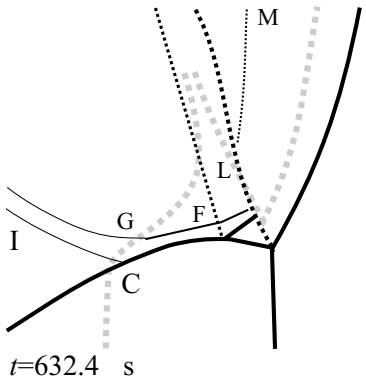
t=629.2 s



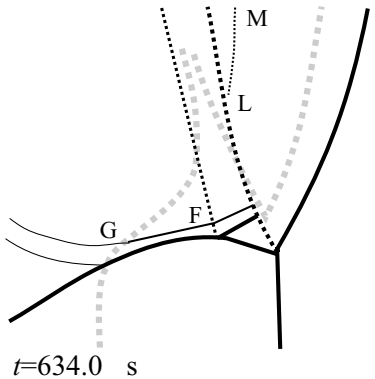
t=630.0 s



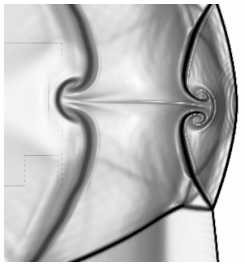
t=630.8 s



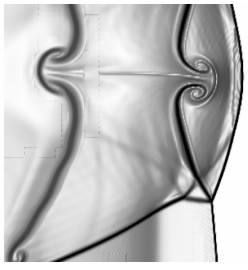
t=632.4 s



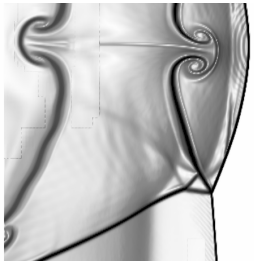
t=634.0 s



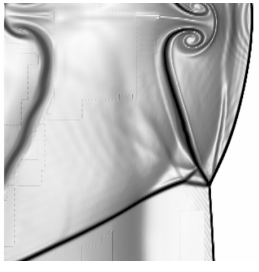
$t = 626.0\,\mu s$



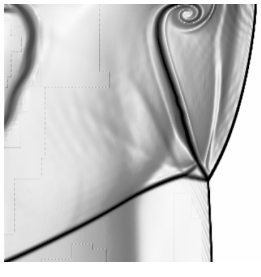
$t = 626.8\,\mu s$



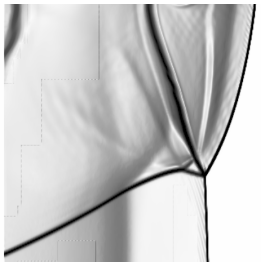
$t = 627.6\,\mu s$



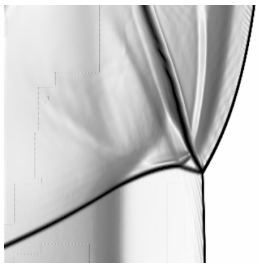
$t = 628.4\,\mu s$



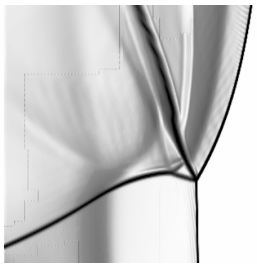
$t = 629.2\,\mu s$



$t = 630.0\,\mu s$



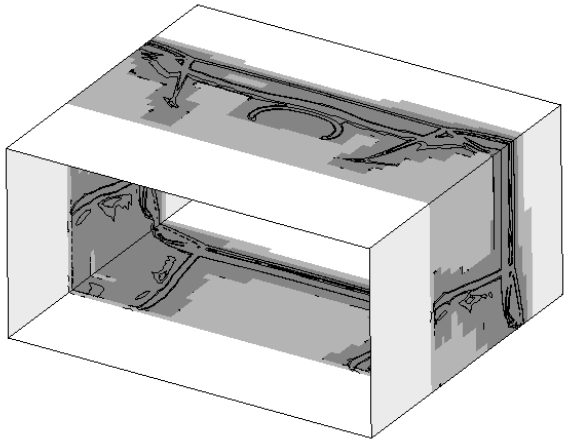
$t = 630.8\,\mu s$



$t = 632.4\,\mu s$

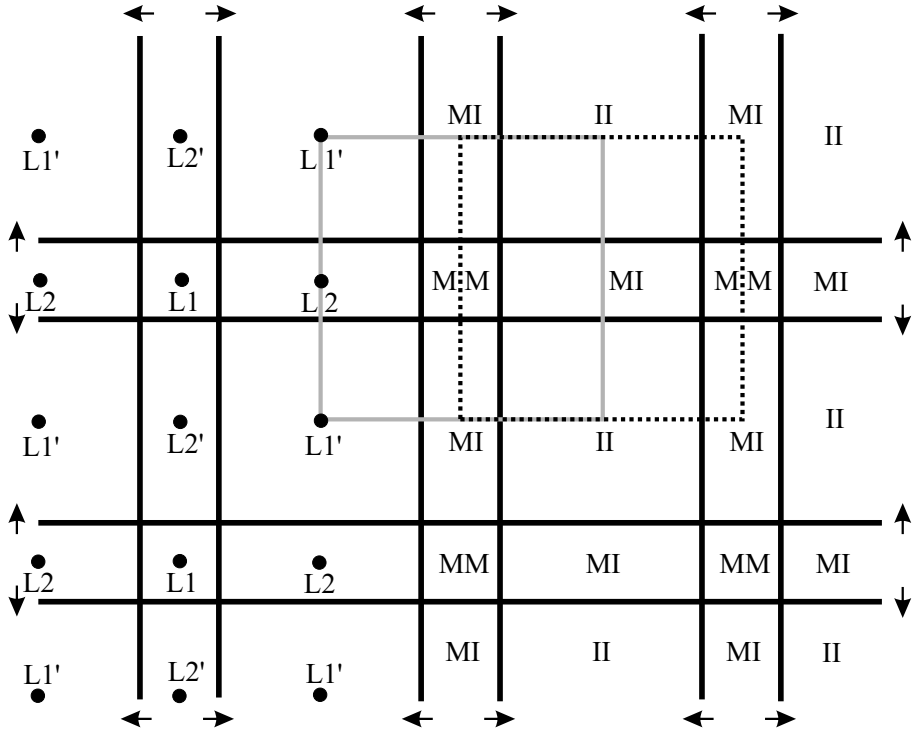
Cellular Structure Simulation - 3D

- Adaption criteria:
  1. Scaled gradients of ρ and p
  2. Error estimation in Y_i by Richardson extrapolation
- Quasi-stationary. Unburned gas flows in with CJ velocity.
- Coarse grid shifted through domain with CJ velocity to obtain triple point tracks.
- Symmetry boundary conditions in x_2-direction.
- $16.8\,\mathrm{Pts}/l_{ig}$. 2 refinement levels (2,3).
- Adaptive computation uses 2.0M-2.5M cells instead of 8.7M cells (uniform grid).
- 74h real time on 48 nodes Athlon 1.4GHz.

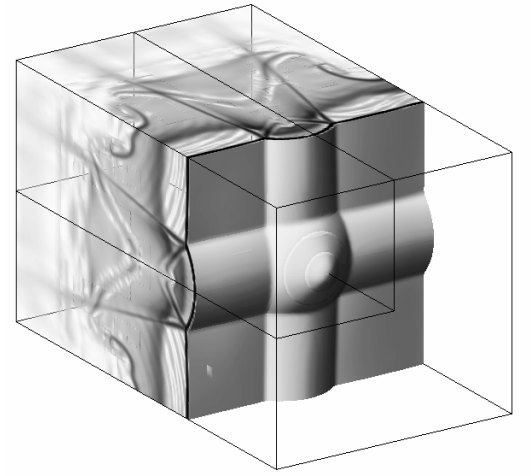


Refinement grids at
$t = 660\,\mu\mathrm{s} + 620\,\mu\mathrm{s}$.

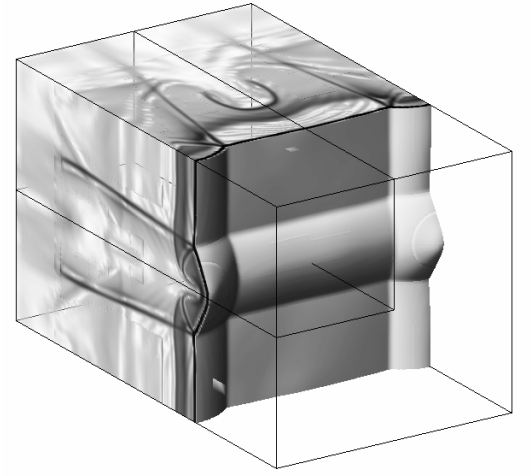
Periodicity of the Solution



Front view of the periodic solution

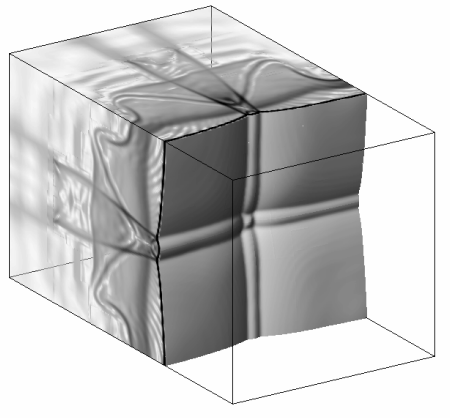


$t = 680\,\mu\mathrm{s} + 600\,\mu\mathrm{s}$ (Computation 1)

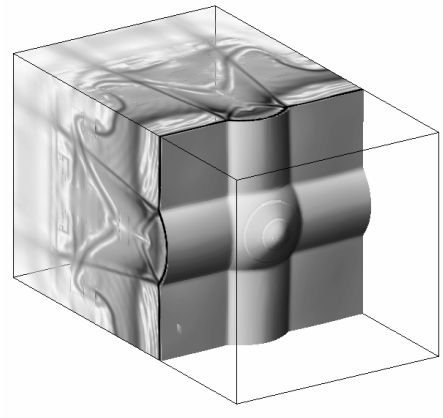


$t = 660\,\mu\mathrm{s} + 620\,\mu\mathrm{s}$ (Computation 2)

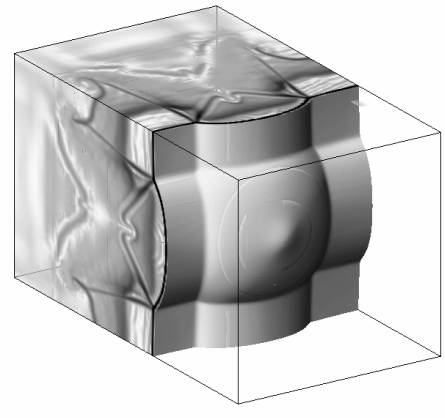
Schlieren Plot of Density



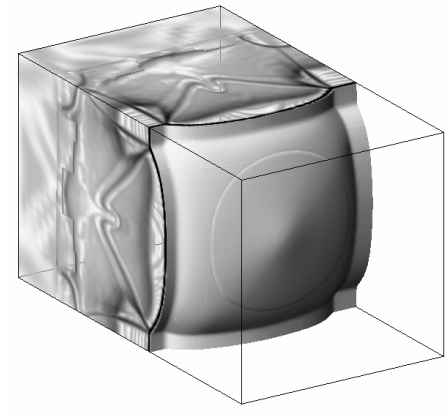
$t = 676\,\mu\text{s} + 600\,\mu\text{s}$



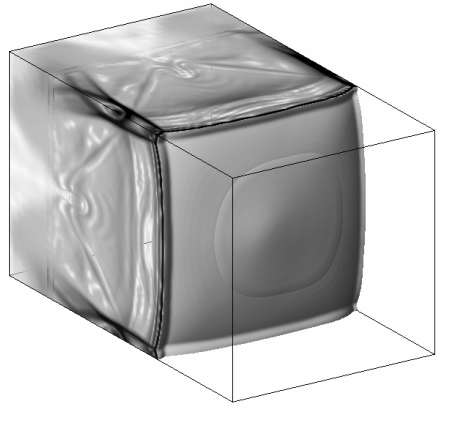
$t = 680\,\mu\text{s} + 600\,\mu\text{s}$



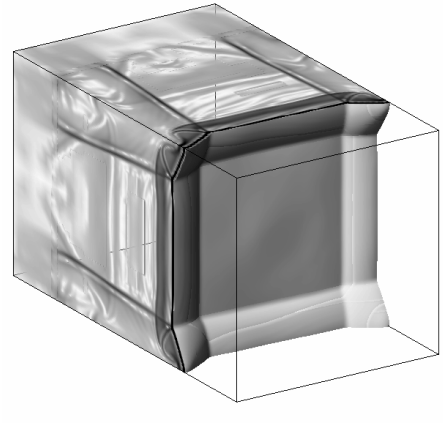
$t = 684\,\mu\text{s} + 600\,\mu\text{s}$



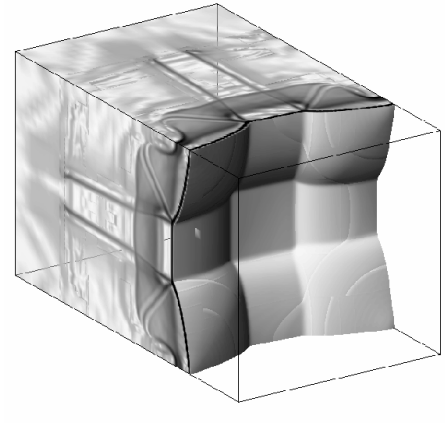
$t = 688\,\mu\text{s} + 600\,\mu\text{s}$



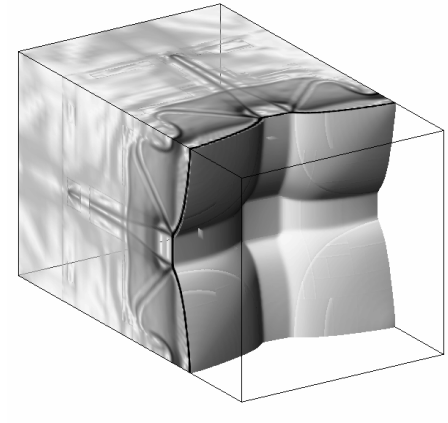
$t = 692\,\mu\text{s} + 600\,\mu\text{s}$



$t = 696\,\mu\text{s} + 600\,\mu\text{s}$



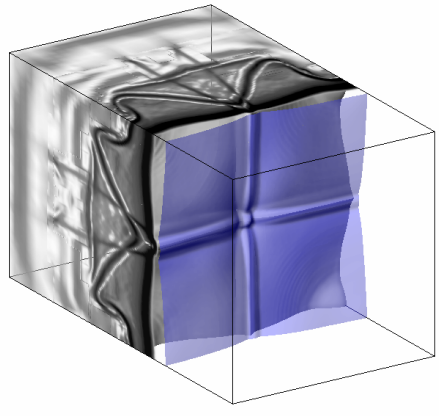
$t = 700\,\mu\text{s} + 600\,\mu\text{s}$



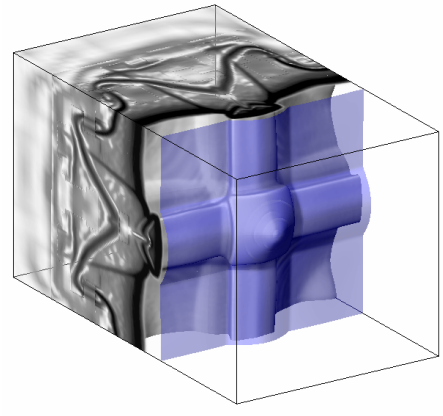
$t = 704\,\mu\text{s} + 600\,\mu\text{s}$

Animation

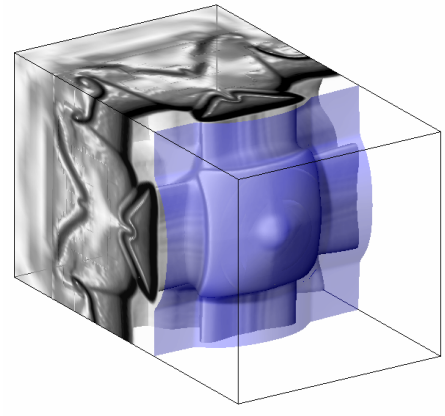
Schlieren Plot of Mass Fraction OH



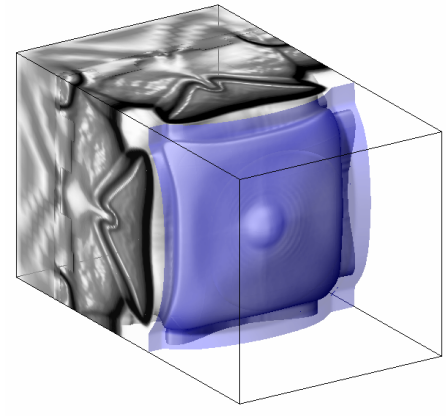
$t = 676\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$



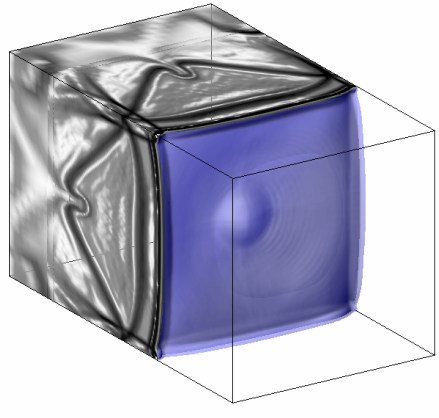
$t = 680\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$



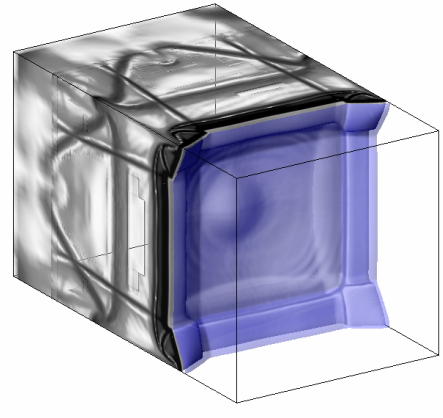
$t = 684\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$



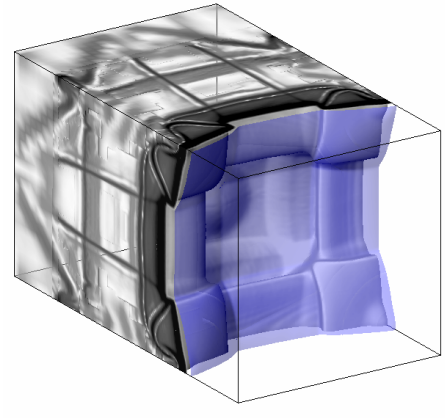
$t = 688\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$



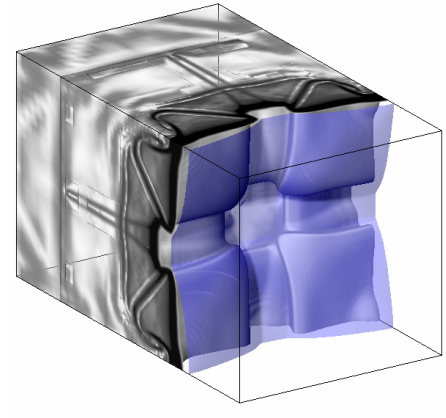
$t = 692\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$



$t = 696\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$



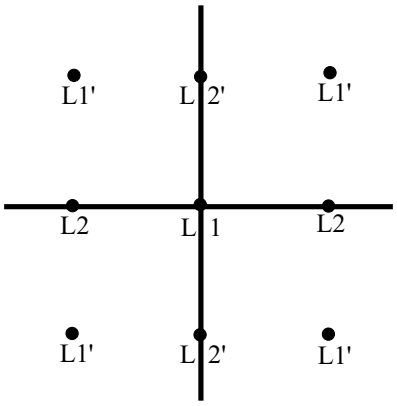
$t = 700\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$

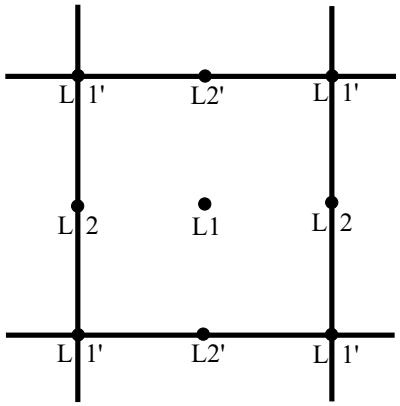


$t = 704\,\mu\mathrm{s}\ +600\,\mu\mathrm{s}$

Animation

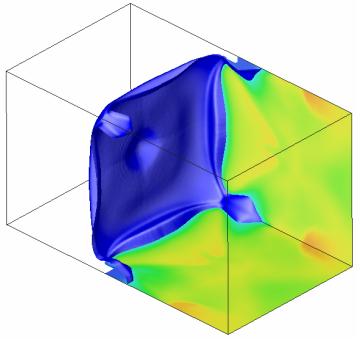
Temporal Development of Detonation Velocity



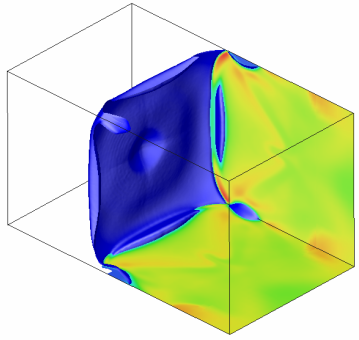


Point-wise reinitiation along L1 (left) and L1' (right)

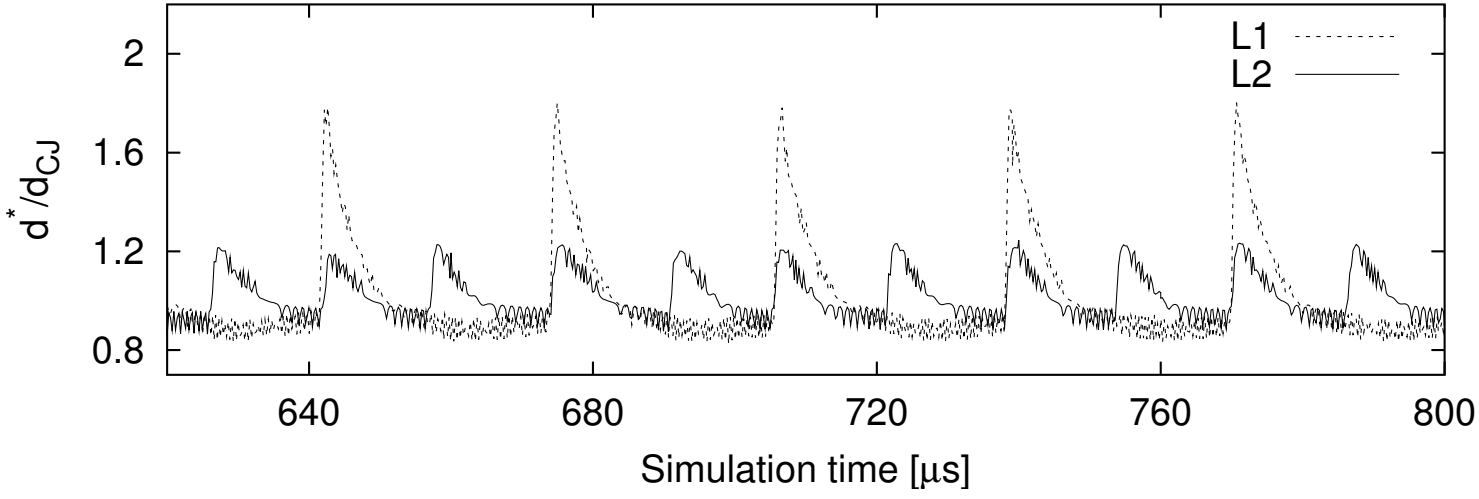
Y_{OH} during reinitiation



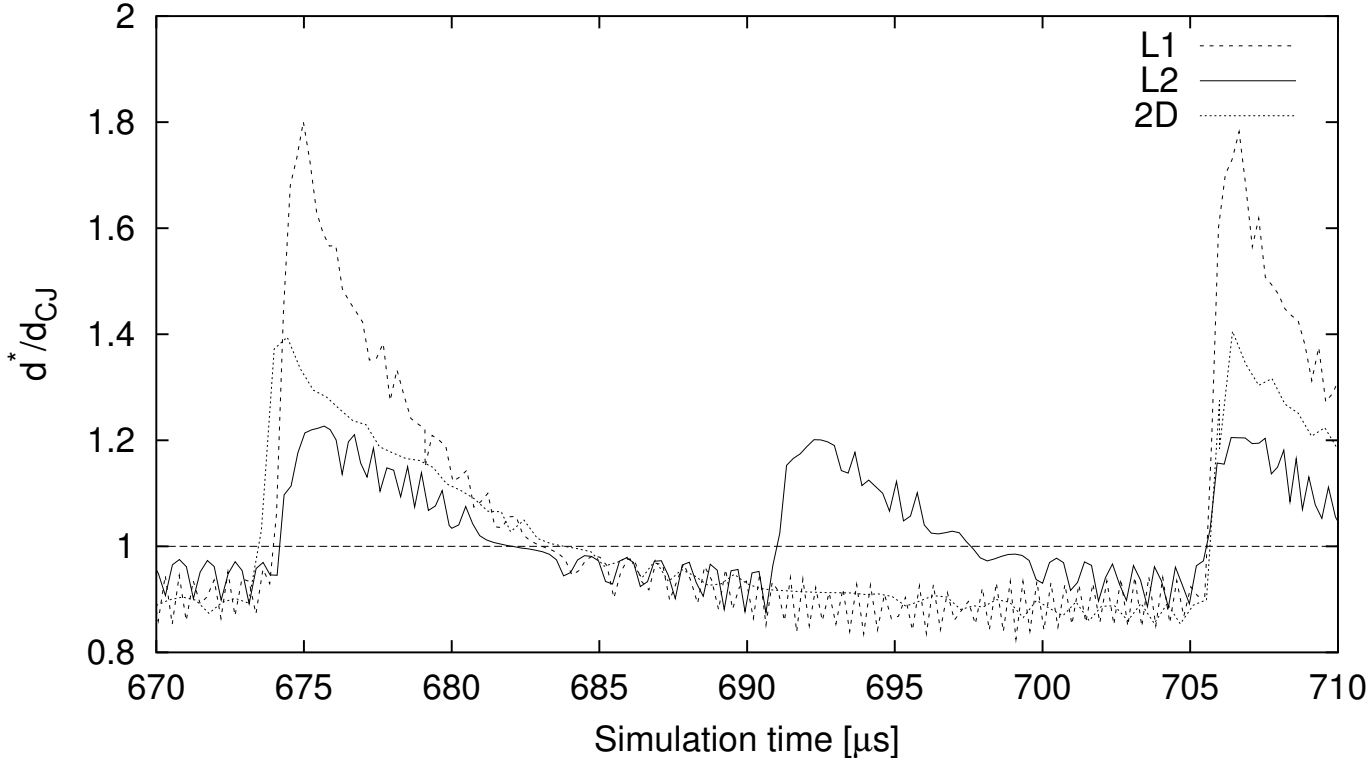
$t = 689.6\,\mu\text{s} + 600\,\mu\text{s}$



$t = 690.4\,\mu\text{s} + 600\,\mu\text{s}$



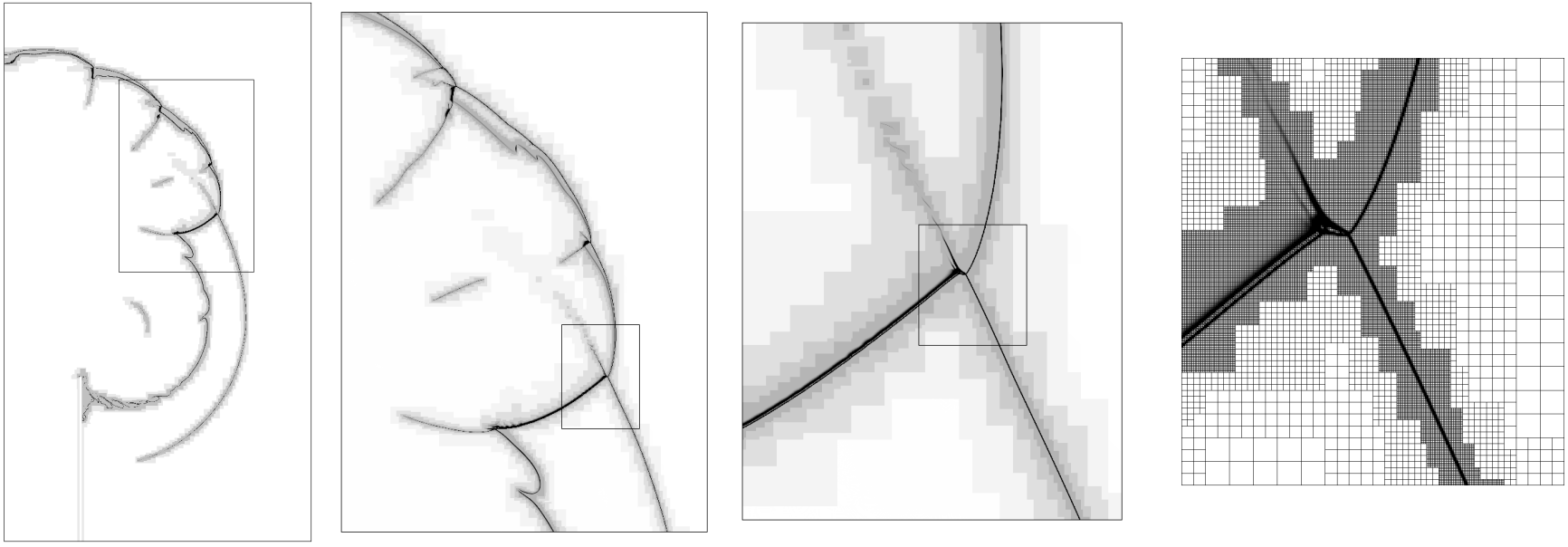
Comparison with 2D Simulation



The flow field is different in 2D and 3D, but the oscillation period is identical!

Simulation - Detonation Diffraction

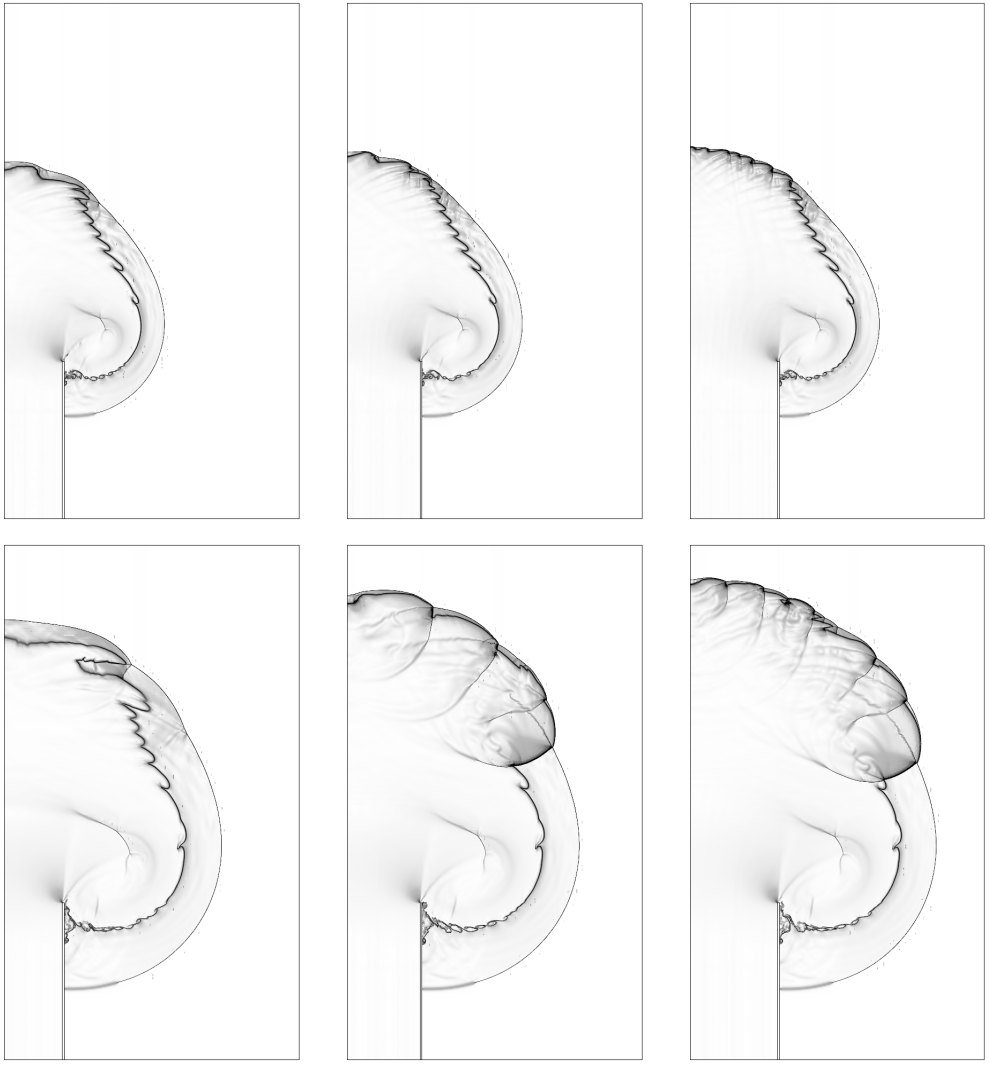
- CJ detonation for $\mathrm{H}_2 : \mathrm{O}_2 : \mathrm{Ar}/2 : 1 : 7$ at $T_0 = 298\,\mathrm{K}$ and $p_0 = 10\,\mathrm{kPa}$. $\lambda_c = 1.6\,\mathrm{cm}$.
- Adaption criteria:
  1. Scaled gradients of ρ and p
  2. Error estimation in Y_i by Richardson extrapolation
- $25\,\mathrm{Pts}/l_{ig}$. 5 refinement levels (2,2,2,4).
- Adaptive computations use up to 2.2M ($w = 6.4\,\mathrm{cm} \equiv 4\lambda_c$) and 2.4M cells ($w = 9.6\,\mathrm{cm} \equiv 6\lambda_c$) instead of 145.6M-150.8M cells (uniform grid).
- 80h ($w = 4\lambda_c$) to 84h ($w = 6\lambda_c$) real time on 48 nodes Athlon 1.4GHz.



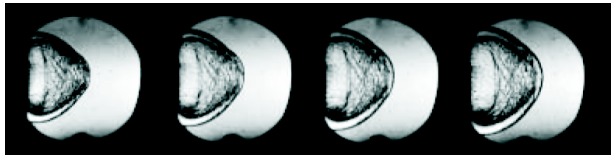
$t = 240$. Refinement on 5 levels. $w = 5\lambda_c$

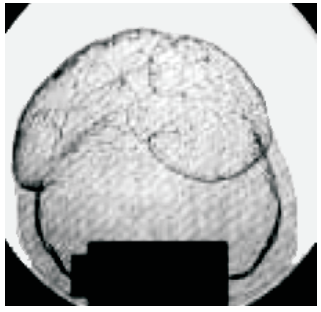
Diffraction of a $H_2 : O_2 : Ar$-Detonation

Upper row: $t = 160\,\mu s$, lower row: $t = 240\,\mu s$



$w = 4\lambda_c$ $w = 5\lambda_c$ $w = 6\lambda_c$





E. Schultz. *Detonation diffraction through an abrupt area expansion*. PhD thesis, California Institute of Technology, Pasadena, California, April 2000.

Blockstructured AMR for the Poisson Equation *

$$\begin{aligned} \Delta q(\mathbf{x}) &= \psi(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset I\!R^d \\ q &= \psi^{\Gamma}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega \end{aligned}$$

Discrete Poisson equation in 2D: $\mathcal{A}(Q_{\Delta x_1, \Delta x_2}) = \begin{bmatrix} & \frac{1}{\Delta x_2^2} & \\ \frac{1}{\Delta x_1^2} & -\left(\frac{1}{\Delta x_1^2} + \frac{1}{\Delta x_2^2}\right) & \frac{1}{\Delta x_2^2} \\ & \frac{1}{\Delta x_2^2} & \end{bmatrix} Q(x_{1,j}, x_{1,k}) = \psi_{jk}$

Patch-wise application of iterative method! Optimal choice: Red-Black-Gauss-Seidel iteration.

1. $Q_{jk}^{m+1} = \psi_{jk} - \left(\frac{1}{\Delta x_1^2} + \frac{1}{\Delta x_2^2}\right)^{-1} \left(\frac{Q_{j+1,k}^m}{\Delta x_1^2} + \frac{Q_{j-1,k}^m}{\Delta x_1^2} + \frac{Q_{j,k+1}^m}{\Delta x_2^2} + \frac{Q_{j,k-1}^m}{\Delta x_2^2}\right)$ for $j + k \bmod 2 = 0$
2. Synchronization
3. $Q_{jk}^{m+1} = \psi_{jk} - \left(\frac{1}{\Delta x_1^2} + \frac{1}{\Delta x_2^2}\right)^{-1} \left(\frac{Q_{j+1,k}^{m+1}}{\Delta x_1^2} + \frac{Q_{j-1,k}^{m+1}}{\Delta x_1^2} + \frac{Q_{j,k+1}^{m+1}}{\Delta x_2^2} + \frac{Q_{j,k-1}^{m+1}}{\Delta x_2^2}\right)$ for $j + k \bmod 2 = 1$

In a finite volume discretization the boundary conditions can only be prescribed iteratively.

Direct method: $Q^{m+1} := \mathcal{S}(Q^m, \psi)$ by iterating on $\mathcal{A}Q = \psi$

Smoothing (requires patch-wise defect calculation and patch-wise smoother):
$d^m := \psi - \mathcal{A}(Q^m)$, $v^m := \mathcal{S}(0, d^m)$ by iterating on $\mathcal{A}(v) = d^m$, $Q^{m+1} := Q^m + v^m = Q^m + \mathcal{C}(d^m)$

Idea of geometric multigrid: $Q_l^{m+1} := Q_l^m + \mathcal{P}_{l-1}^l \mathcal{S}_{l-1} \mathcal{R}_l^{l-1}(d_l^m)$

Smoothing of fine grid problem on coarser mesh to achieve faster damping (convergence!) of low frequency oscillations.

* D. Martin, K. Cartwright. Solving Poisson's equation using adaptive mesh refinement. UC Berkley, 1996.

Additive Geometric Multiplicative Multigrid Algorithm

AdvanceLevelMG(l) - Correction Scheme

Set ghost cells of Q_l	
Calculate defect d_l from Q_l,ψ_l	$d_l := \psi_l - \mathcal{A}(Q_l)$
If ($l < l_{max}$)	
Restrict d_{l+1} onto d_l	$d_l := \mathcal{R}^l_{l+1}(d_{l+1})$
Do ν_1 smoothing steps to get correction v_l	$v_l := \mathcal{S}(0, d_l, \nu_1)$
If ($l > l_{min}$)	
Do $\gamma > 1$ times	
AdvanceLevelMG($l-1$)	
Set ghost cells of v_{l-1}	
Prolongate and add v_{l-1} to v_l	$v_l := v_l + \mathcal{P}^l_{l-1}(v_{l-1})$
If ($\nu_2 > 0$)	
Set ghost cells of v_l	
Update defect d_l according to v_l	$d_l := d_l - \mathcal{A}(v_l)$
Do ν_2 post-smoothing steps to get r_l	$r_l := \mathcal{S}(v_l, d_l, \nu_2)$
Add addional correction r_l to v_l	$v_l := v_l + r_l$
Add correction v_l to Q_l	$Q_l := Q_l + v_l$

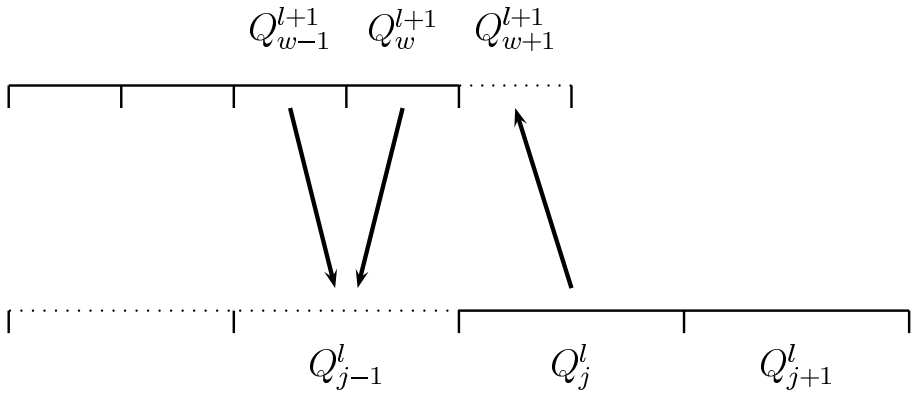
Start - Start iteration on level l_{max}

For $l = l_{max}$ Downto $l_{min}+1$ Do	
Restrict Q_l onto Q_{l-1}	$Q_{l-1} := \mathcal{R}^{l-1}_l(Q_l)$
Regrid(0)	
AdvanceLevelMG(l_{max})	

Modification of Stencil at Coarse-Fine Boundaries

1D Example: Cell j, $\psi - \nabla \cdot \nabla q = 0$

$$d_j^l = \psi_j - \frac{1}{\Delta x_l}\left(\frac{1}{\Delta x_l}(Q_{j+1}^l - Q_j^l) - \frac{1}{\Delta x_l}(Q_j^l - Q_{j-1}^l)\right) = \psi_j - \frac{1}{\Delta x_l}\left(G_{j+\frac{1}{2}}^l - G_{j-\frac{1}{2}}^l\right)$$

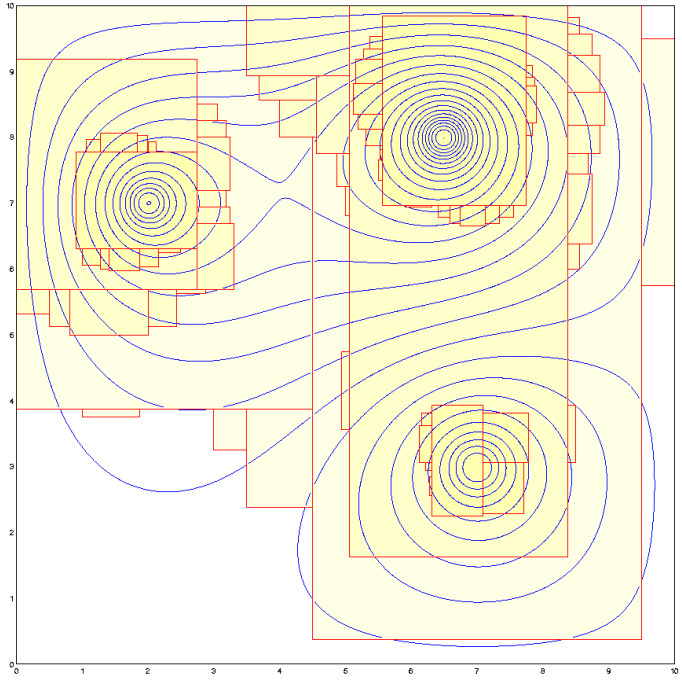


Correction pass:

1. $\delta G_{j-\frac{1}{2}}^{l+1} := -G_{j-\frac{1}{2}}^l$
2. $\delta G_{j-\frac{1}{2}}^{l+1} := \delta G_{j-\frac{1}{2}}^{l+1} + G_{w+\frac{1}{2}}^{l+1} = \delta G_{j-\frac{1}{2}}^{l+1} + (Q_j^l - Q_w^{l+1})/\Delta x_{l+1}$
3. $\check{d}_j^l := d_j^l + \frac{1}{\Delta x_l}\delta G_{j-\frac{1}{2}}^{l+1}$

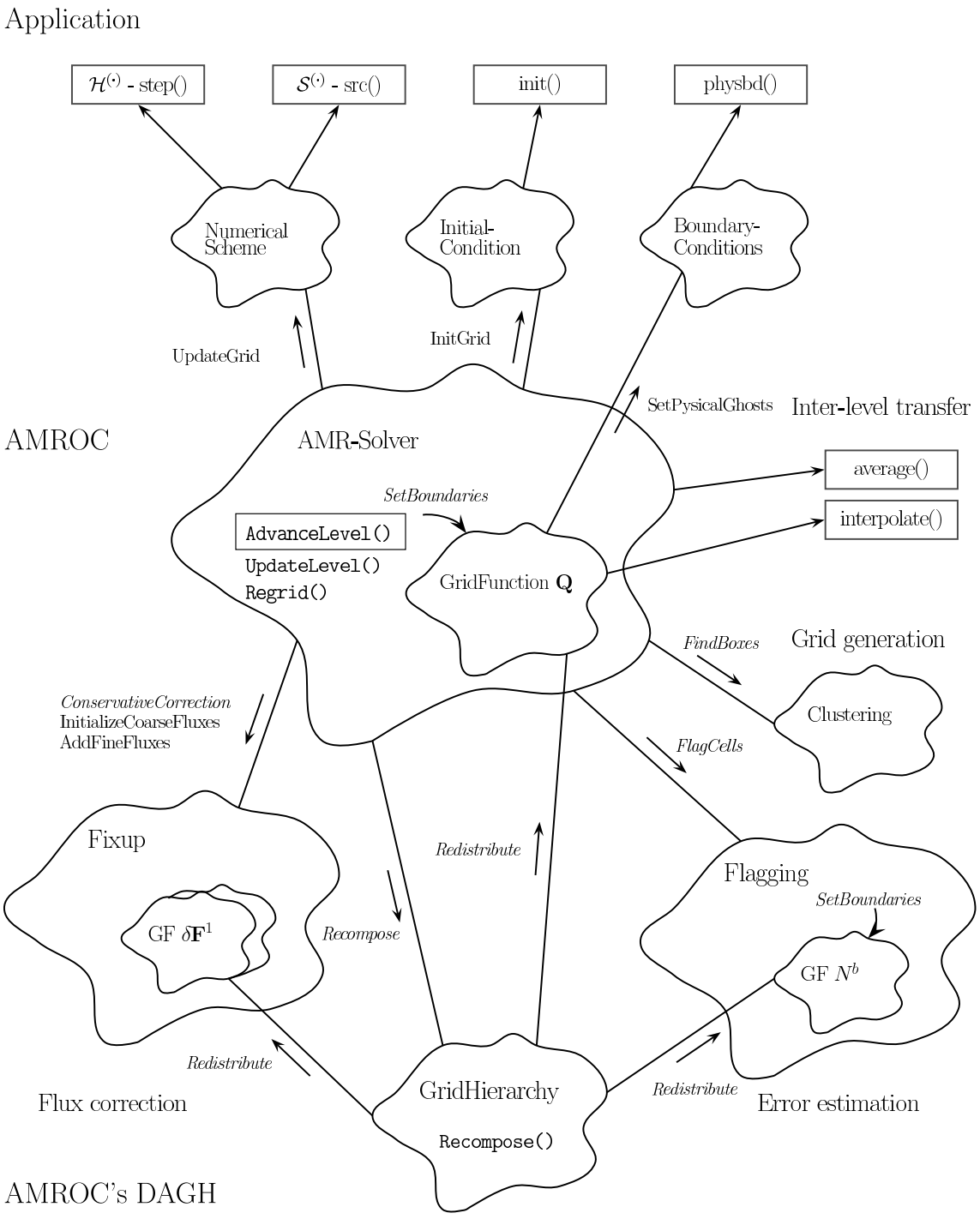
$$\check{d}_j^l = \psi_j - \frac{1}{\Delta x_l}\left(\frac{1}{\Delta x_l}(Q_{j+1}^l - Q_j^l) - \frac{1}{\Delta x_{l+1}}(Q_j^l - Q_w^{l+1})\right)$$

Example

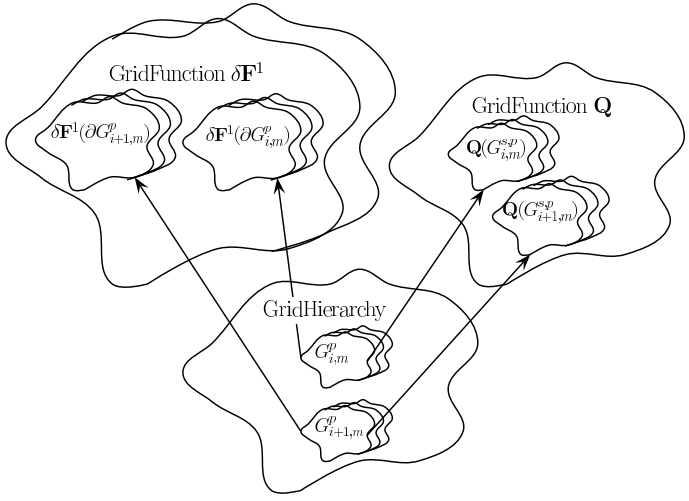


	80×80	640×640	640×640	640×640
l_{max}	3	0	0	0
l_{min}	-3	-6	-3	0
ν_1	5	5	5	1
ν_2	5	5	5	0
V-Cycles	16	15	504	
Iterations	1040	975	17640	
Time [sec]	14.8	18.6	640	

Stop at $\|d_l\|_{max} < 10^{-7}$ for $l \geq 0$, $\gamma = 1$, $r_l = 2$

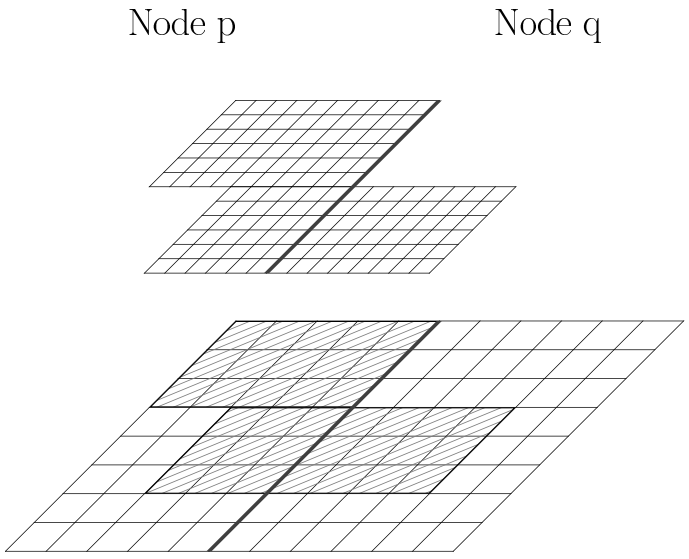


Design of AMROC



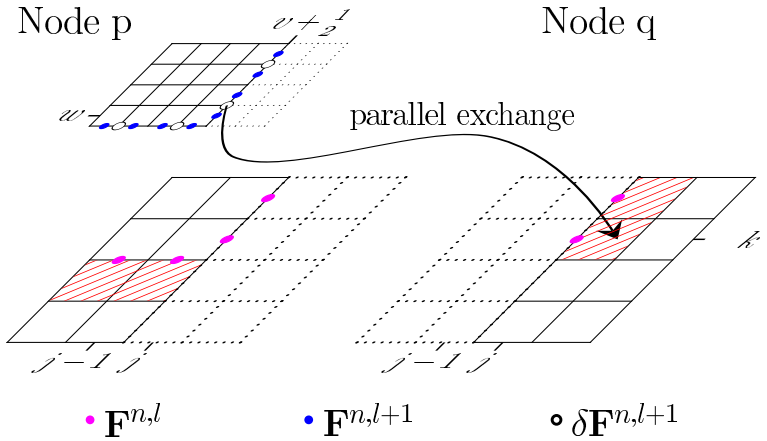
The concept of GridFunctions

AMROC with Clawpack:
http://amroc.sourceforge.net



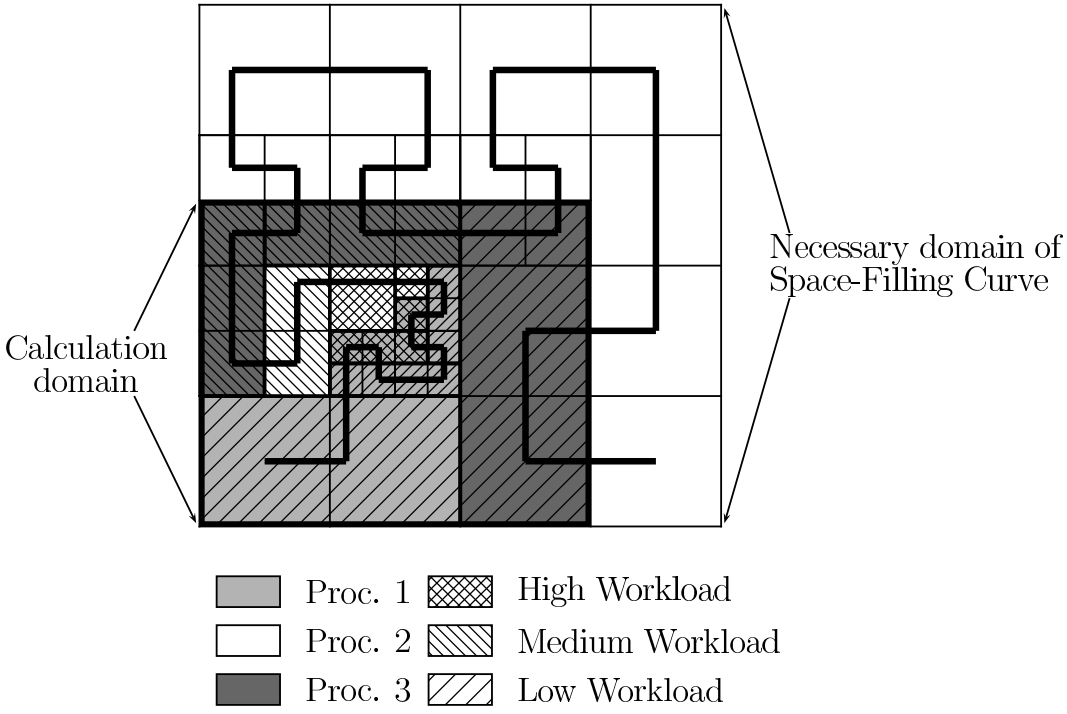
- Rigorous domain decomposition approach
- Data of all levels resides on same node
- Grid hierarchy defines unique "floor-plan"
- Redistribution of data blocks during reorganization
  of hierarchical data
- Synchronization when setting ghost cells

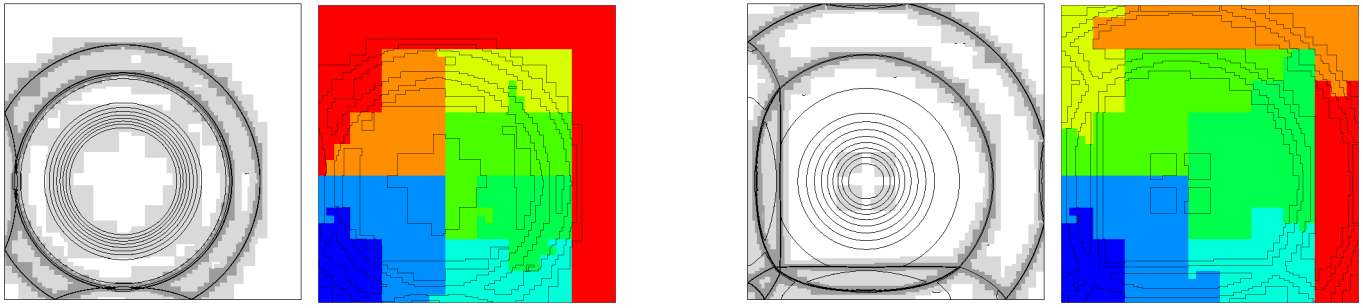
Parallelization Strategy



- Strictly local calculation of flux correction terms
- Only application requires communication

Partioning

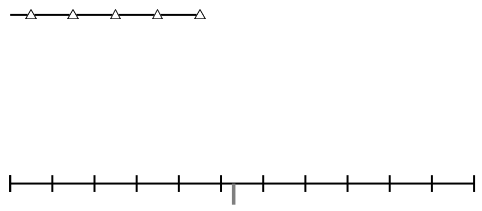
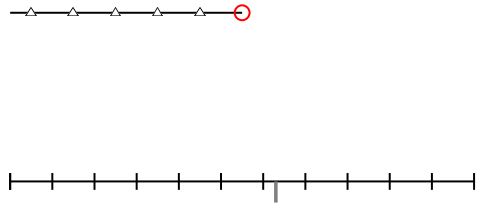
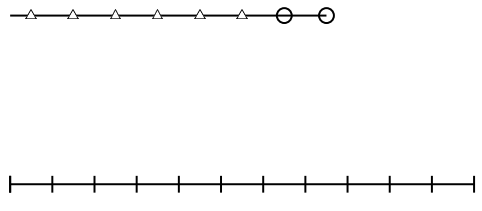

Calculation domain
Necessary domain of Space-Filling Curve
Proc. 1
Proc. 2
Proc. 3
High Workload
Medium Workload
Low Workload



Embedded Boundary Methods

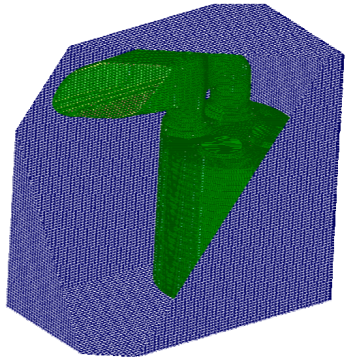
Representation of a moving boundary in a Cartesian method.

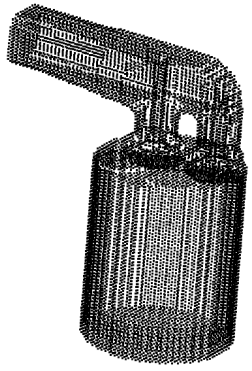
1. Methods that diffuse the boundary in one cell:
   - E.g. internal ghost cell values
   - Not conservative by construction
   - Usually combined with implicit geometry representation
2. Methods that represent the boundary sharply:
   - Cut-cell approach constructs appropriate finite volumes.
   - Conservative by construction. Correct boundary flux.
   - Key question: How to avoid small-cell time step restriction?
   - Usually explicit geometry representation used

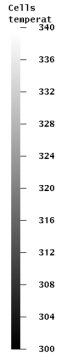
Propagate boundary	
1. Calculate new volumes 2. Reinitialize cell with $V_i^n < \frac{1}{2}, V_i^{n+1} = 1$	
1. Construct ghost cell values 2. Solve	

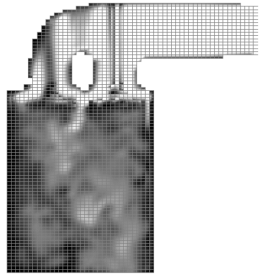
Diffused method using internal ghost cell values.

K. J. Richards et al., On the use of the immersed boundary method for engine modeling





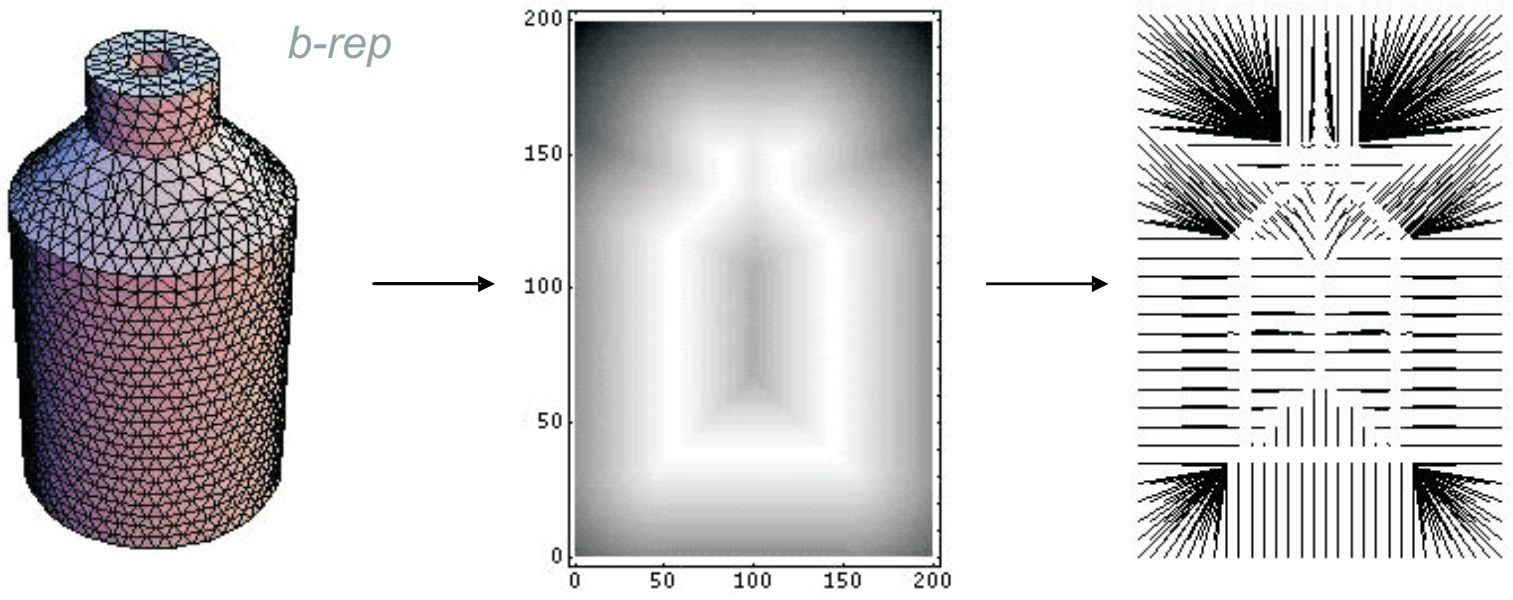




Projective initiated detonation by Patrick Hung

SAMR within the Virtual Test Facility *

- Construct AMR framework for Cartesian finite volume methods that supports the combination of embedded boundary and ghost-fluid methods.
- Implicit geometry representation based on level sets.
- Transformation of moving surface meshes into level set with exact and optimal Closest-point-transform algorithm by Sean Mauch.



Surface mesh | Distance $\varphi(x_1, x_2)$ | Normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$ to closest point

- Implement diffused boundary method first, but consider accurate embedded boundary method based on

$$V_j^{n+1}\mathbf{Q}_j^{n+1} = V_j^n\mathbf{Q}_j^n - \Delta t\left(A_{j+1/2}^{n+1/2}\,\mathbf{F}(\mathbf{Q}, j) - A_{j-1/2}^{n-1/2}\,\mathbf{F}(\mathbf{Q}, j-1)\right)$$

via subsequent correction step after Cartesian finite-volume update step later.

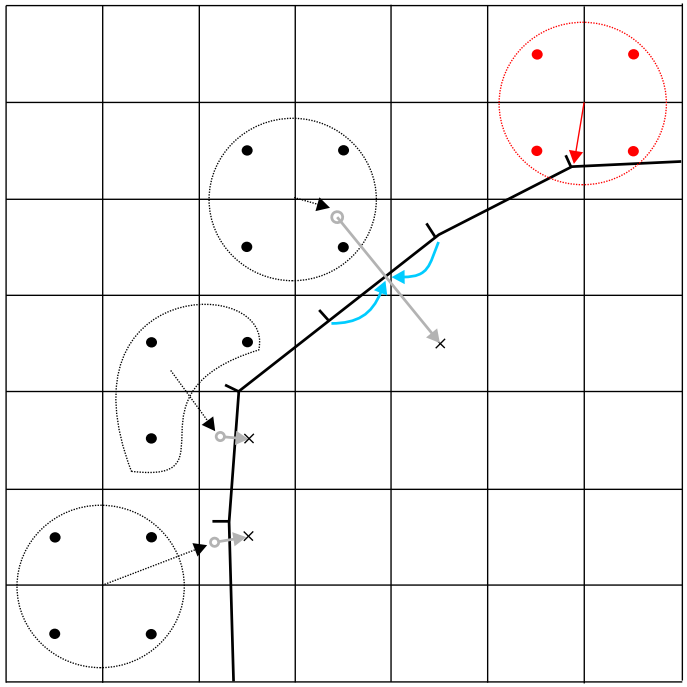
* J. Cummings, M. Aivazis, R. Samtaney, R. Radovitzky, S. Mauch, D. Meiron, A virtual test facility for the simulation of dynamic response in materials, *J. Supercomputing*, 23 (1): 39-50, Aug 2002.

Construction of Boundary Values for Cartesian Fluid Scheme

Moving boundary/interface is treated as a moving contact discontinuity.

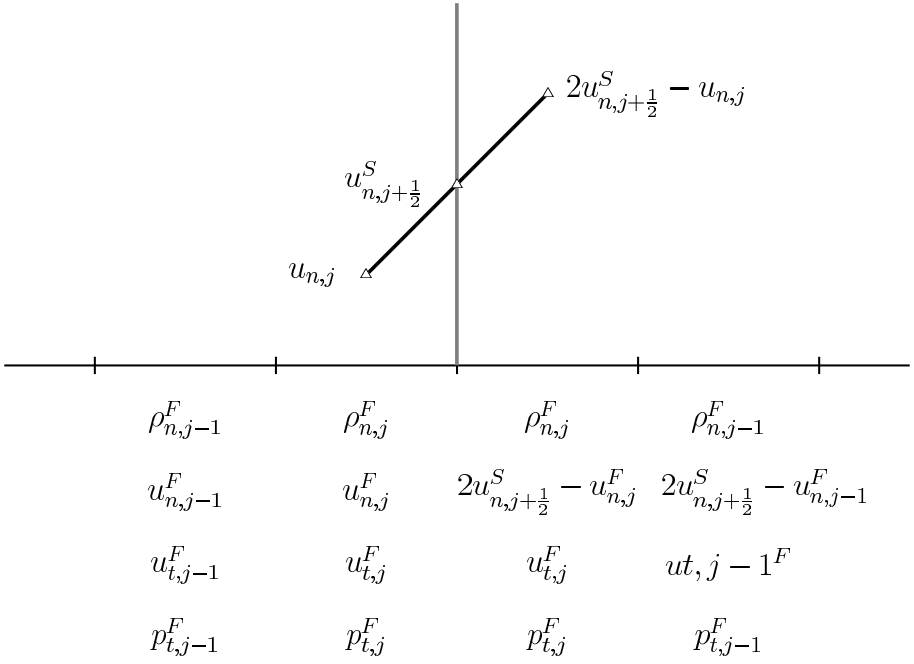
Solid-fluid coupling for FEM and FV scheme with mirroring:

Linear interpolation on fluid nodes



- One-sided construction of mirrored ghost cell and new FEM nodal point values
- FEM ansatz-function interpolation to obtain intermediate surface values

$$
\begin{aligned}
u_n^F &= u_n^S, \\
-p^F &= \sigma_n^S, \\
0 &= \sigma_t^S
\end{aligned}
$$



Find values ρ_M, p_M, $\mathbf{u}_M^F$ by interpolation at

$$\tilde{\mathbf{x}} = \mathbf{x} + 2\varphi\mathbf{n}$$

Velocity in ghost cells

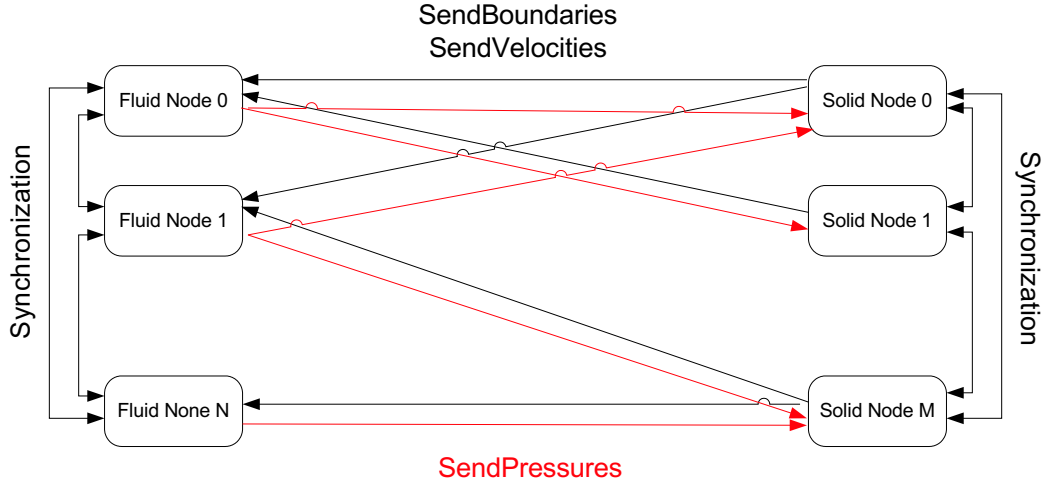
$$\mathbf{u}_{Gh}^F = \left((2\mathbf{u}^S - \mathbf{u}_M^F)\cdot\mathbf{n}\right)\mathbf{n} + \mathbf{u}_M^F - (\mathbf{u}_M^F\cdot\mathbf{n})\mathbf{n}$$

M. Arienti, P. Hung, E. Morano, and J. E. Shepherd. A level set approach to Eulerian-Lagrangian copuling. *J. Comput. Phys.*, 185:213–251, 2003.

Parallelization Strategy for Coupled Simulations

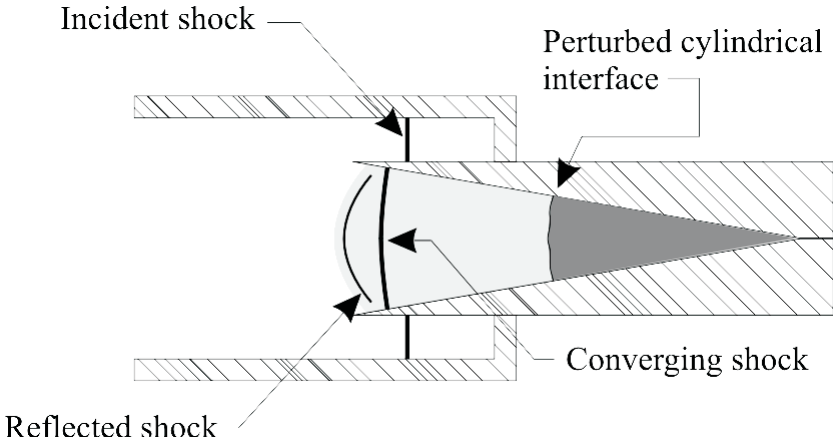
Coupling of an Eulerian Finite Volume Fluid Solver and a Lagrangian Finite Shell Element Solver:

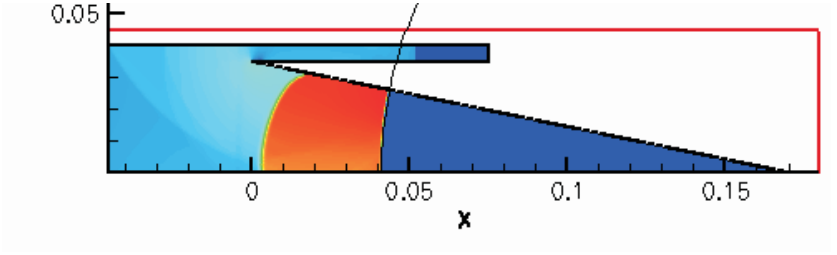
- Fluid solver operates on volume data, while solid solver uses surface meshes.
- For compressible gas dynamics the size of the fluid problems exceeds the solid problem by magnitudes.
- Effective parallelization approach in distributed memory environment:
  - Distribute both meshes seperately and copy necessary nodal values and geometry data to fluid nodes.
  - Setting of ghost cell values becomes strictly local operation.
  - Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes.
  - Only surface data is transfered.
  - Asynchronous communication ensures scalability
  - Generic encapsulated implementation guarantees reusability



Shock Focusing and Fluid Instability in Converging Geometry

- Experiment by Paul Dimotakis with Amy Lang, Garret Katzenstein, Daniel Lang.
- Focusing of stable shock wave (without Mach reflection).
- Richtmyer-Meshkov instability between two gases in converging geometry.
- Shock interaction with complex geometry and fluid interface.
- Simulation of compressible turbulence requires requires very accurate higher-order shock capturing method $\longrightarrow$ Hybrid Tuned Centered-Difference with Weighted Essentially Non-Oscillatory (WENO) stencil (David Hill).
- Subgrid-scale turbulence modeling for Large-Eddy Simulation (LES) for Euler equations (Dale Pullin, David Hill).
- Validation of simulation (Ravi Samtaney, David Hill) via
  - Shock evolution
  - Mixing layer thickness





Preliminary simulation (Julian Cummings, Michael Aivazis)

Detonation Driven Fracture

- Experiment by Joe Shepherd, T.-W. Chao, J. Austin
- Interaction of detonation, ductile deformation, fracture $\longrightarrow$ solid-fluid coupling of Eulerian AMR to Lagrangian FEM code (Michael Ortiz and his group, Fehmi Cirak) using shells elements with fracture capability
- Validation of simulation (Fehmi Cirak, Patrick Hung) via
  - Stress history of cylinder
  - Crack propagation history
  - Species concentration and detonation fine structure