Advanced Multi-Scale Methods for Hypersonic Propulsion

By

Z. J. Zikoski, C. M. Romick, J. M. Powers and S. Paolucci

Aerospace and Mechanical Engineering
University of Notre Dame, Indiana 46556

AFOSR/NASA Foundational Hypersonics Project Review
Dallas, Texas, November 17-21, 2008

Project Summary

- An adaptive method is applied to problems in hypersonic propulsion.
- Compressible reactive Navier-Stokes model includes detailed chemical kinetics, multi-species transport, momentum and energy diffusion.
- These problems are typically multidimensional and contain a wide range of spatial and temporal scales.
- Our adaptive wavelet method allows this range of scales to be resolved while greatly reducing the required computer time and automatically produces verified solutions.

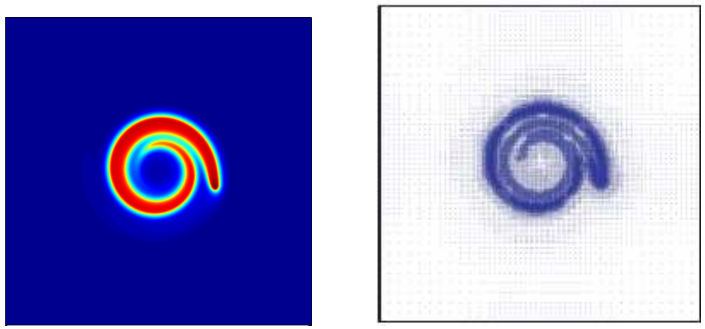


Figure: Flameball-vortex interaction– computed temperature field and adaptive grid.

Wavelet Approximation in Domain $[0,1]^d$

Approximation of $u(\mathbf{x})$ by the interpolating wavelet, a multiscale basis, on $\mathbf{x} \in [0,1]^d$ is given by

$$u(\mathbf{x}) \approx u^J(\mathbf{x}) = \sum_{\mathbf{k}} u_{j_0,\mathbf{k}} \Phi_{J_0,\mathbf{k}}(\mathbf{x}) + \sum_{j=J_0}^{J-1} \sum_{\lambda} d_{j,\lambda} \Psi_{j,\lambda}(\mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^d$, $\lambda = (\mathbf{e}, \mathbf{k})$ and $\Psi_{j,\lambda}(\mathbf{x}) \equiv \Psi^{\mathbf{e}}_{j,\mathbf{k}}(\mathbf{x})$.

- Scaling function:
$$\Phi_{j,\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{d} \phi_{j,\mathbf{k}}(x_i), \ k_i \in \kappa_j^0$$

- Wavelet function:
$$\Psi^{\mathbf{e}}_{j,\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{d} \psi^{e_i}_{j,\mathbf{k}}(x_i), \ k_i \in \kappa_j^{e_i}$$

where $\mathbf{e} \in \{0,1\}^d \setminus \mathbf{0}$, $\psi^0_{j,k}(x) \equiv \phi_{j,k}(x)$ and $\psi^1_{j,k}(x) \equiv \psi_{j,k}(x)$, and $\kappa_j^0 = \{0, \cdots, 2^j\}$ and $\kappa_j^1 = \{0, \cdots, 2^j - 1\}$.

1-D Interpolating Scaling Function and Wavelet

Some properties of $\phi_{j,k}$ and $\psi_{j,k}$ of order p ($p \in \mathbb{N}$, even):

- $\phi_{j,k}$ is defined through $\phi(2^j x - k)$ where $\phi(x) = \int \varphi_p(y)\varphi_p(y-x)dy$, the auto-correlation of the Daubechies wavelet $\varphi_p(x)$.
- The support of $\phi_{j,k}$ is compact, *i.e.* $\text{supp}\{\phi_{j,k}\} \sim |O(2^{-j})|$.
- $\phi_{j,k}(x_{j,n} = n2^{-j}) = \delta_{k,n}$, *i.e.* satisfies the *interpolation property.*
- $\psi_{j,k} = \phi_{j+1,2k+1}$.
- $\text{span}\{\phi_{j,k}\} = \text{span}\{\{\phi_{j-1,k}\}, \{\psi_{j-1,k}\}\}$.
- $\{1, x, \cdots, x^{p-1}\}$, for $x \in [0,1]$, can be written as a linear combination of $\{\phi_{j,k},\ k = 0, \cdots, 2^j\}$.
- $\{\{\phi_{J_0,k}\}, \{\psi_{j,k}\}_{j=J_0}^{\infty}\}$ forms a basis of a continuous 1-D function on the unit interval $[0,1]$.

Sparse Wavelet Representation (SWR) and Irregular Sparse grid

- For a given threshold parameter ε, the multiscale approximation of a function $u(\mathbf{x})$ can be written as

$$u^J(\mathbf{x}) = \sum_{\mathbf{k}} u_{J_0,\mathbf{k}} \Phi_{j_0,\mathbf{k}}(\mathbf{x}) + \sum_{j=j_0}^{J-1} \sum_{\{\boldsymbol{\lambda} \,:\, |d_{j,\boldsymbol{\lambda}}| \geq \varepsilon\}} d_{j,\boldsymbol{\lambda}} \Psi_{j,\boldsymbol{\lambda}}(\mathbf{x})$$

$$+ \underbrace{\sum_{j=j_0}^{J-1} \sum_{\{\boldsymbol{\lambda} \,:\, |d_{j,\boldsymbol{\lambda}}| < \varepsilon\}} d_{j,\boldsymbol{\lambda}} \Psi_{j,\boldsymbol{\lambda}}(\mathbf{x})}_{R_\varepsilon^J},$$

  and the SWR is obtained by discarding the term R_ε^J.

- For interpolating wavelets, each basis function is associated with one dyadic grid point, *i.e.*

$$\Phi_{j,\mathbf{k}}(\mathbf{x}) \quad \text{with} \quad \mathbf{x}_{j,\mathbf{k}} = (k_1 2^{-j}, \ldots, k_d 2^{-j})$$

$$\Psi_{j,\lambda}(\mathbf{x}) \quad \text{with} \quad \mathbf{x}_{j,\boldsymbol{\lambda}} = \mathbf{x}_{j+1,2\mathbf{k}+\mathbf{e}}$$

SWR and Irregular Sparse Grid (continued)

- For a given SWR, one has an associated grid composed of *essential* points, whose wavelet amplitudes are greater than the threshold parameter ε

$$\boldsymbol{\mathcal{V}}_e = \{\mathbf{x}_{j_0,\mathbf{k}}, \bigcup_{j \geq j_0} \mathbf{x}_{j,\boldsymbol{\lambda}} \ : \ \lambda \in \boldsymbol{\Lambda}_j\}, \quad \boldsymbol{\Lambda}_j = \{\lambda \ : \ |d_{j,\lambda}| \geq \varepsilon\}.$$

- To accommodate the possible advection and sharpening of solution features, we determine the *neighboring* grid points:

$$\boldsymbol{\mathcal{V}}_b = \bigcup_{\{j,\boldsymbol{\lambda} \in \boldsymbol{\Lambda}\}} \mathcal{N}_{j,\boldsymbol{\Lambda}},$$

where $\mathcal{N}_{j,\boldsymbol{\lambda}}$ is the set of neighboring points to $x_{j,\boldsymbol{\lambda}}$.

- The new sparse grid, $\mathcal{V}$, is then given by

$$\boldsymbol{\mathcal{V}} = \mathbf{x}_{j_0,k} \cup \boldsymbol{\mathcal{V}}_e \cup \boldsymbol{\mathcal{V}}_b.$$

SWR and Irregular Sparse grid (continued)

- There exists an adaptive fast wavelet transform ($AFWT$), with $O(N)$, $N = \dim\{\boldsymbol{\mathcal{V}}\}$ operations, mapping the function values on the irregular grid $\boldsymbol{\mathcal{V}}$ to the associated wavelet coefficients and *vice-versa*:

$$AFWT(\{u(\mathbf{x}) \,:\, \mathbf{x} \in \boldsymbol{\mathcal{V}}\}) \rightarrow \mathcal{D} = \{\{u_{j_0,\mathbf{k}}\}, \{d_{j,\boldsymbol{\lambda}},\ \boldsymbol{\lambda} \in \boldsymbol{\Lambda}_j\}_{j>j0}\}.$$

- Provided that the function $u(\mathbf{x})$ is continuous, the error in the SWR $u_\varepsilon^J(\mathbf{x})$ is bounded by

$$\|u - u_\varepsilon^J\|_\infty \leq C_1\ \varepsilon.$$

- Furthermore, for the function that is smooth enough, the number of basis functions $N = \dim\{u_\varepsilon^J\}$ required for a given ε satisfies

$$N \leq C_2\ \varepsilon^{-d/p}, \quad \text{and} \quad \|u - u_\varepsilon^J\|_\infty \leq C_2\ N^{-p/d}.$$

Derivative Approximation of SWR

- Direct differentiation of wavelets is costly (with $O(p(J-j_0)N)$ operations) because of different support sizes of wavelet basis on different levels.
- Alternatively, we use finite differences to approximate the derivative on a grid of irregular points. The procedure can be summarized as follows:
  1. For a given SWR of a function, perform the inverse interpolating wavelet transform to obtain the function values at the associated irregular points.
  2. Apply locally a finite difference scheme of order n to approximate the derivative at each grid point.
- Estimate shows that the pointwise error of the derivative approximation has the following bound:

$$\|\partial^i u/\partial x^i - D_x^{(i)} u_\varepsilon^J\|_{\boldsymbol{\nu},\infty} \le C N^{-\min((p-i),n)/2}, \quad \|f\|_{\mathcal{G},\infty} = \max_{\mathbf{x}\in\boldsymbol{\nu}} |f(x)|.$$

Dynamically Adaptive Algorithm for Solving Time-Dependent PDEs

Given the set of PDEs

$$\frac{\partial u}{\partial t} = F(t, u, u_x, u_{xx}, \ldots),$$

with initial conditions

$$u^0 = u(x, 0).$$

1. Obtain sparse grid, $\boldsymbol{\nu}^m$, based on thresholding of magnitudes of wavelet amplitudes of the approximate solution u^m.
2. Integrate in time using an explicit time integrator with error control to obtain the new solution u^{m+1}.
3. Assign $u^{m+1} \rightarrow u^m$ and return to step 1.

Compressible Reactive Flow

Code solves the n-D compressible reactive Navier-Stokes equations:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\frac{\partial}{\partial x_i}\left(\rho u_i\right) \\
\frac{\partial \rho u_i}{\partial t} &= -\frac{\partial}{\partial x_j}\left(\rho u_j u_i\right) - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \\
\frac{\partial \rho E}{\partial t} &= -\frac{\partial}{\partial x_j}\left(u_j(\rho E + p)\right) + \frac{\partial u_j \tau_{ji}}{\partial x_i} - \frac{\partial q_i}{\partial x_i} \\
\frac{\partial \rho Y_k}{\partial t} &= -\frac{\partial}{\partial x_i}\left(u_i \rho Y_k\right) + M_k \dot{\omega}_k - \frac{\partial j_{k,i}}{\partial x_i}, \qquad k = 1, \ldots, K
\end{aligned}$$

Where ρ-density, u_i-velocity vector, E-specific total energy, Y_k-mass fraction of species k, τ_{ij}-viscous stress tensor, q_i-heat flux, $j_{k,i}$-species mass flux, M_k- molecular weight of species k, and $\dot{\omega}_k$-reaction rate of species k.

Compressible Reactive Flow (cont.)

Where,

$$
\begin{aligned}
E &= e + \frac{1}{2} u_i u_i \\
\tau_{ij} &= -\frac{2}{3}\mu \frac{\partial u_l}{\partial x_l}\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \\
q_i &= -\mathsf{k}\frac{\partial T}{\partial x_i} + \sum_{k=1}^{K}\left(h_k j_{k,i} - \frac{RT}{m_k X_k} D_k^T d_{k,i}\right) \\
j_{k,i} &= \frac{\rho Y_k}{X_k \overline{M}} \sum_{j=1, j\neq k}^{K} M_j D_{kj} d_{j,i} - \frac{D_k^T}{T}\frac{\partial T}{\partial x_i} \\
d_{k,i} &= \frac{\partial X_k}{\partial x_i} + (X_k - Y_k)\frac{1}{p}\frac{\partial p}{\partial x_i}
\end{aligned}
$$

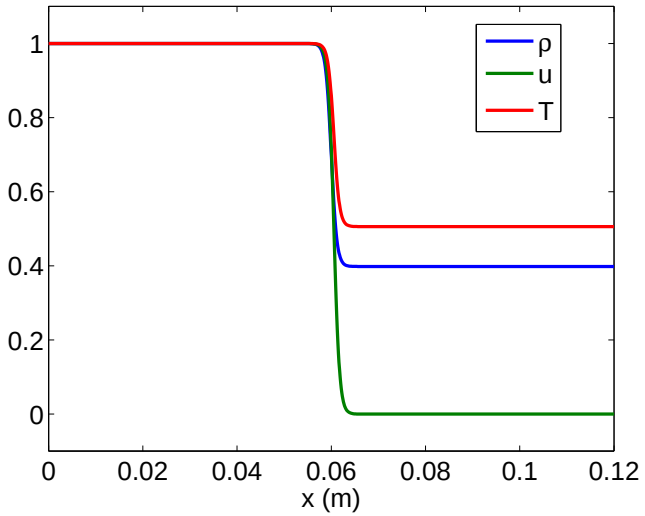
Compressible Reactive Flow (cont.)

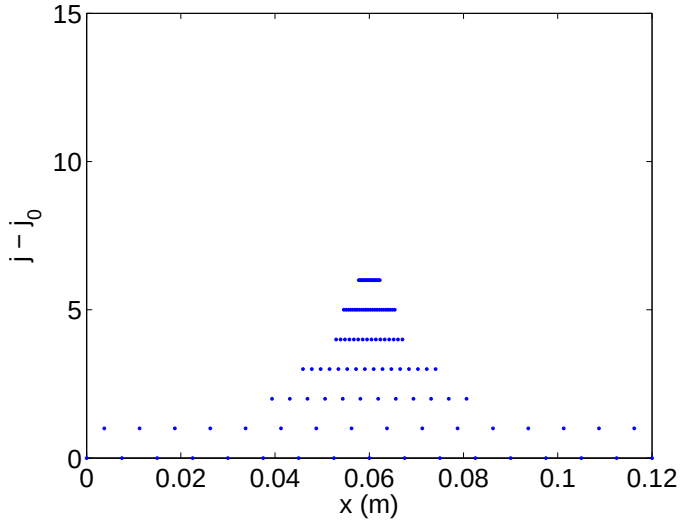
- Serial 1-, 2-, 3-dimensional code is implemented.
- Model includes detailed chemical kinetics, multi-component and thermal diffusion.
- Includes state-dependent specific heats and transport properties.
- CHEMKIN and TRANLIB libraries used for evaluation of transport properties, thermodynamics, and chemical source terms.
- Solutions obtained using hardware at the Center for Research Computing at the University of Notre Dame:

  Dual-Core 2.2 GHz AMD Opteron Model 175 64-bit Linux
  2 GB RAM, 80 GB SATA Disk

1-D Viscous Detonation

Initial conditions:





$2H_2 : 1O_2 : 7Ar$ mixture

9 species, 37 reactions

State 1: $0\ m \leq x < 0.06\ m$

$\rho_1 = 0.18075\ kg\ m^{-3}$

$P_1 = 35594\ Pa$

$u_1 = 487.34\ m\ s^{-1}$

State 2: $0.06\ m \leq x \leq 0.12\ m$

$\rho_2 = 0.072\ kg\ m^{-3}$

$P_2 = 7173\ Pa$

$u_2 = 0\ m\ s^{-1}$

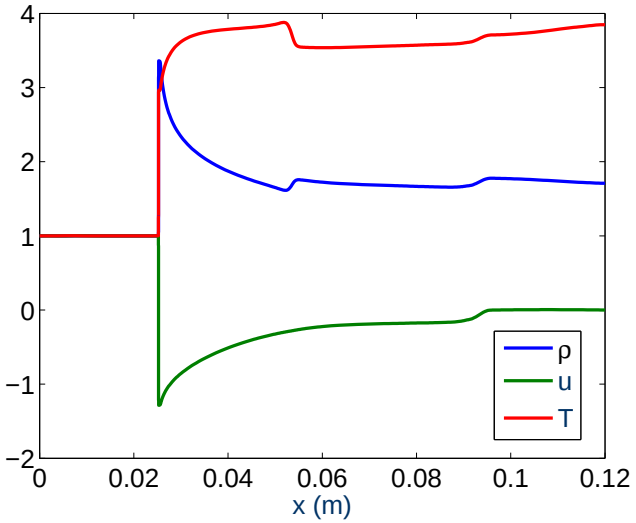
Wavelet parameters:

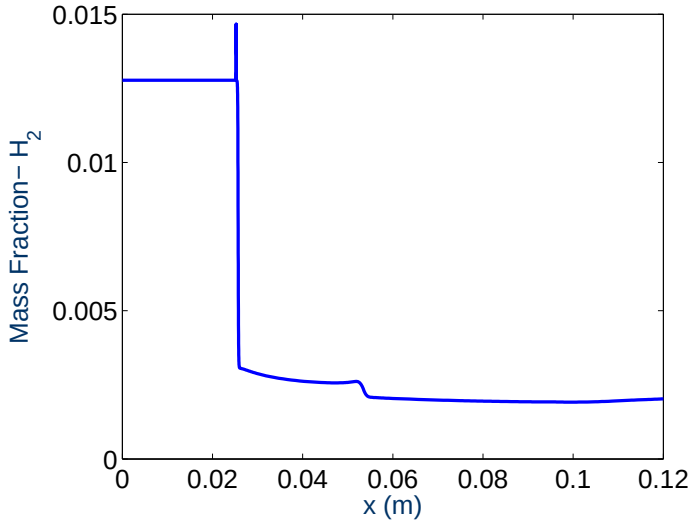
$\epsilon = 1 \times 10^{-4}$

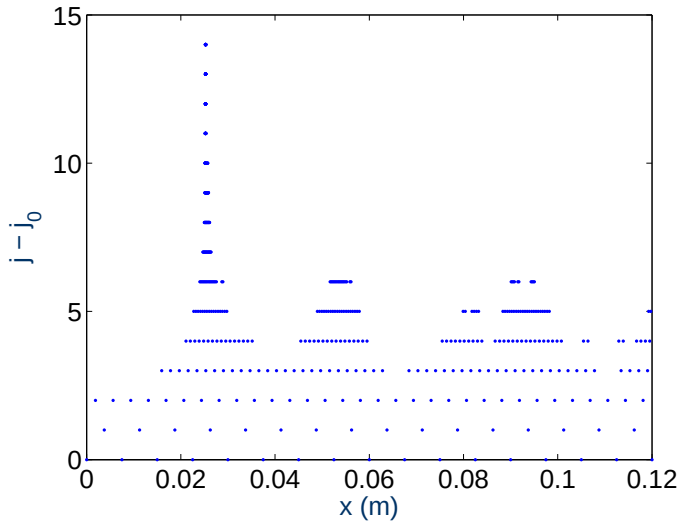
$p = 6, \quad n = 4$

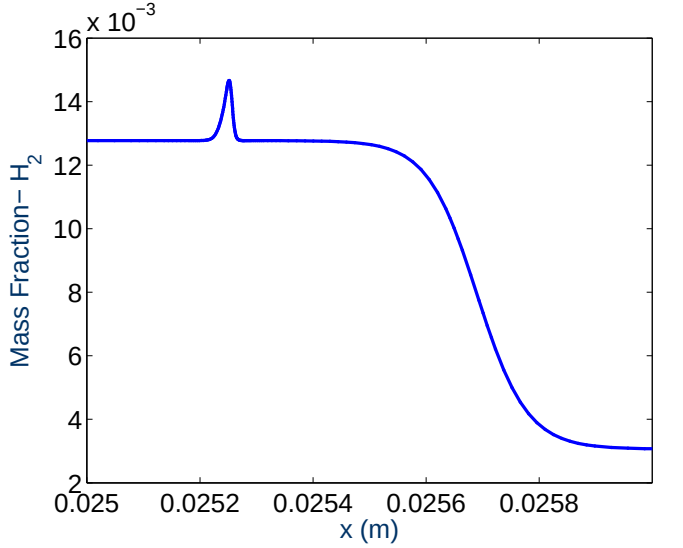
$j_0 = 4, \quad J - j_0 = 15$

1-D Viscous Detonation (cont.)





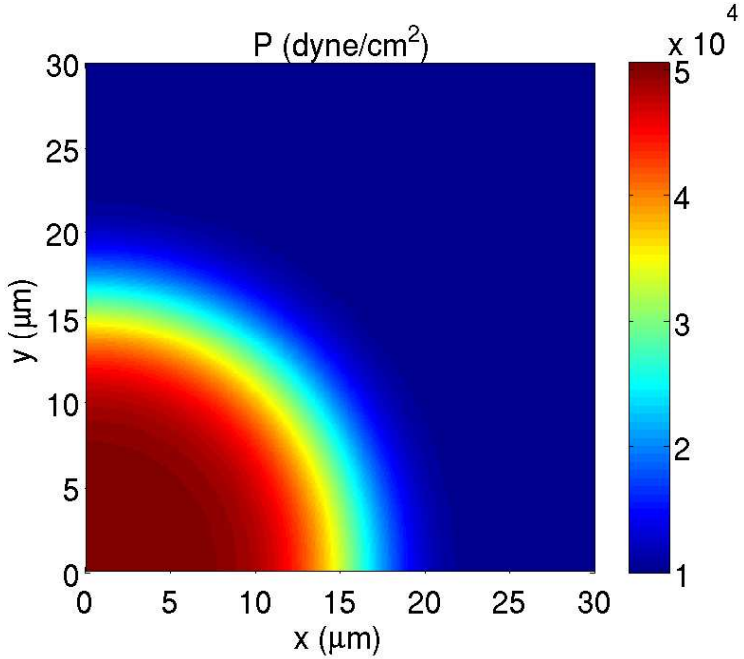




Taylor/Sedov Blast Wave

$78N_2 : 21O_2 : 1Ar$ (air) mixture
3 species, inert

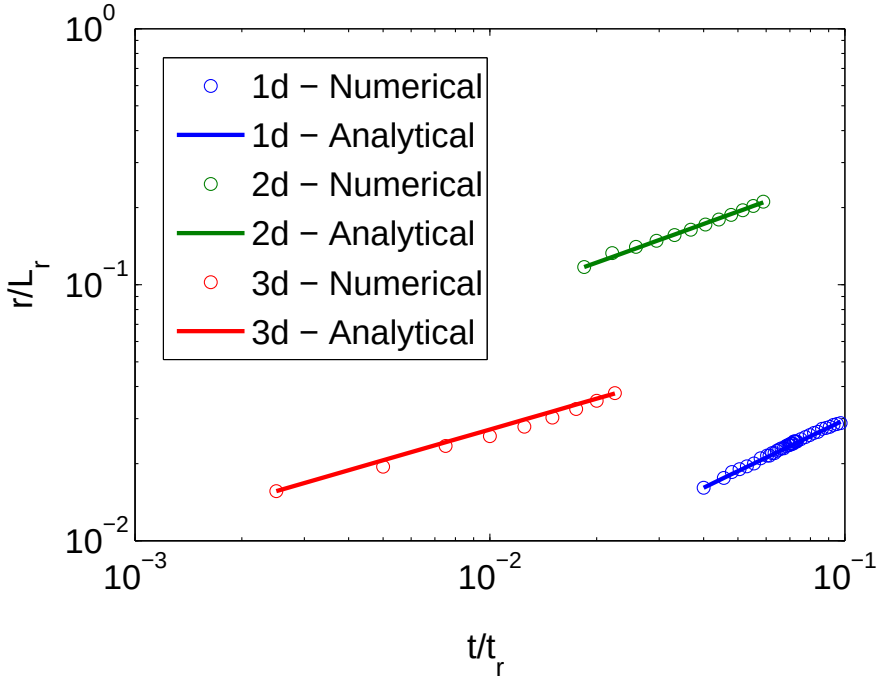
$\rho(\mathbf{x}, 0) = 3 \times 10^{-5}\ gm\ cm^{-3}$
$\mathbf{u}(\mathbf{x}, 0) = 0\ cm\ s^{-1}$
$P_0 = 1 \times 10^4\ dyne\ cm^{-2}$
$P_{max}/P_0 = 50$
$P(\mathbf{x}, 0) = P_0 + P_{max} \exp(-500\|\mathbf{x}/L\|^2)$
$L = 100\ \mu m$



Wavelet parameters:

$\epsilon = 1 \times 10^{-3}$
$p = 6, \quad n = 4$
$j_0 = 3, \quad J - j_0 = 9$ (1-d), 6 (2-,3-d)

Taylor/Sedov Blast Wave (cont.)

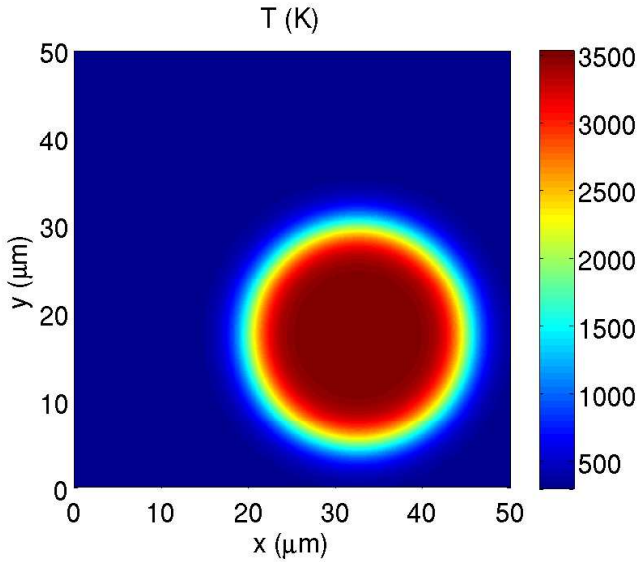


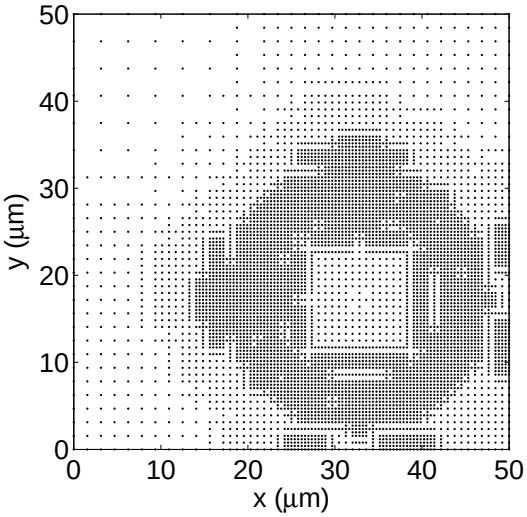
$$r(t) = \left(\frac{E}{\rho_0}\right)^a t^{2a}$$

$$a = (2+d)^{-1}$$

d	a-Analytical	a-Numerical
1	0.6667	0.6645
2	0.5000	0.4842
3	0.4000	0.3979

2-D Flameball





$2H_2 : 1O_2 : 7Ar$ mixture

9 species, 37 reactions

$\mathbf{x}_0 = (32.5\mu m,\ 17.5\mu m)$

$r = \|\mathbf{x} - \mathbf{x}_0\|_2$

$\mathbf{u} = 0\ cm\ s^{-1}$

State 1: $r > 12.5\ \mu m$

$\rho_1 = 1.265\ kg\ m^{-3}$

$T_1 = 300\ K$

State 2: $r \leq 12.5\ \mu m$

$\rho_2 = 1.265\ kg\ m^{-3}$

$T_2 = 3530\ K$

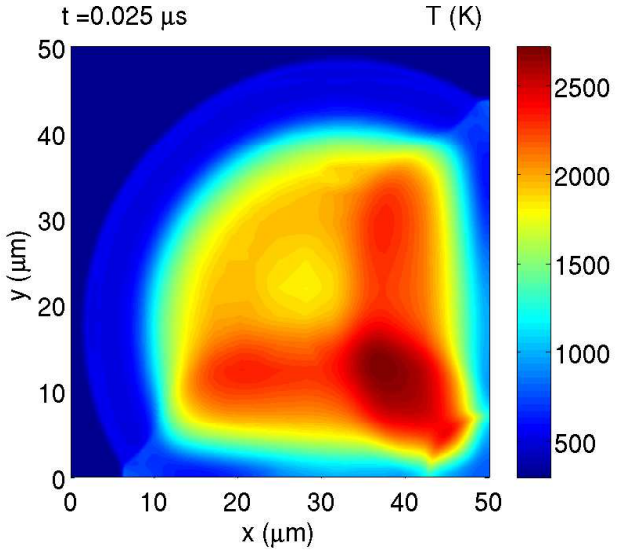
Wavelet parameters:

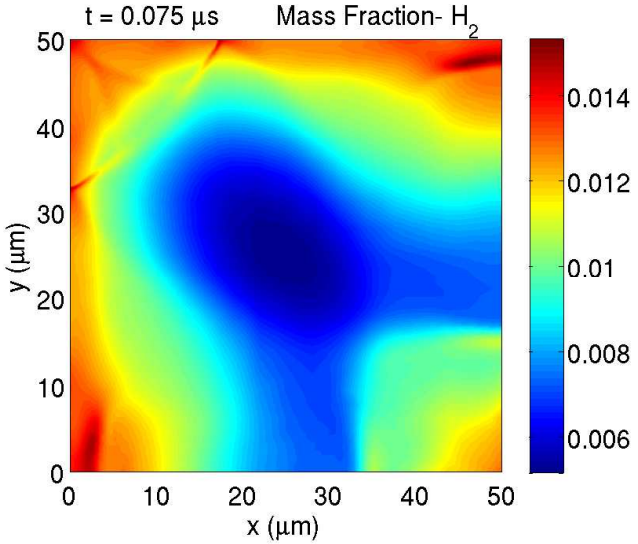
$\epsilon = 1 \times 10^{-3}$

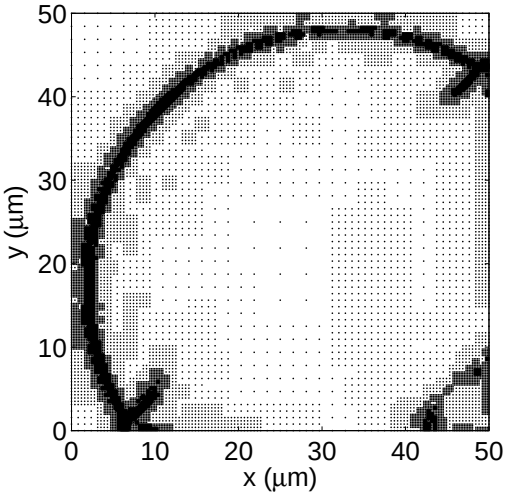
$p = 6, \quad n = 4$

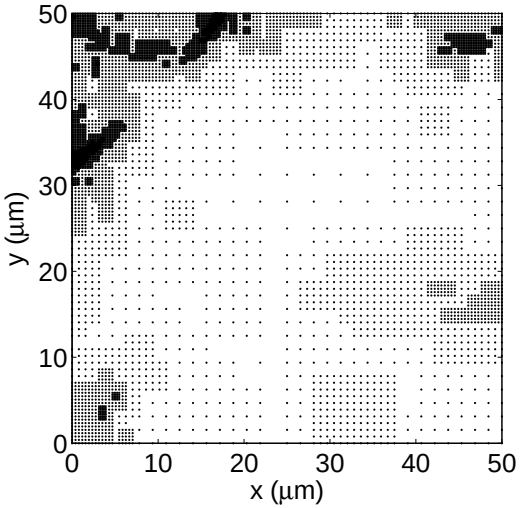
$j_0 = 3, \quad J - j_0 = 7$

2-D Flameball (cont.)









Runtime Comparisons

Case	N_a	N_f	t_{adap} (hr)	t_{full} (hr)	Speedup
1-D Detonation	275	2.6×10^5	343	3.3×10^5	950
1-D Blast Wave	305	4.1×10^3	0.06	0.8×10^0	13
2-D Blast Wave	2566	2.6×10^5	0.83	8.5×10^1	102
3-D Blast Wave	23084	1.3×10^8	29.5	1.7×10^5	5800
2-D Flameball	12784	1.0×10^6	29	2.4×10^3	82

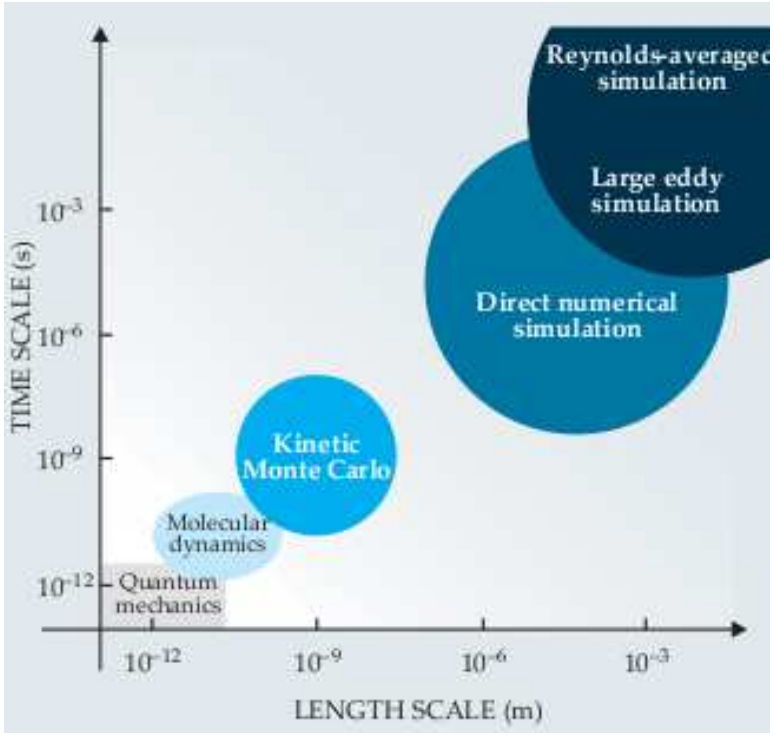
N_a - average number of points in adaptive grid
N_f - total number of points in equivalent uniform grid
t_{adap} - runtime of adaptive routine $[CPU\ hr]$
t_{full} - est. runtime of routine with equivalent full grid $[CPU\ hr]$
Speedup - t_{full}/t_{adap}

Future Work

- Perform coarse-grained message passing-based parallelization.
- Improve data structure and memory management to reduce storage requirements to $O(N)$ and maintain constant-time data access.
- Implement non-reflecting boundary conditions for problems in open domains.
- Include generalized coordinates/domain transformation for non-Cartesian geometries.
- Solve more complex problems with good experimental databases for validation.

Project Challenges

- To maintain time accuracy, time step is restricted by finest spatial grid size.
- We need better time integration strategies, *i.e.* multiple time stepping or a time-adaptive method.
- Parallel domain decomposition and load balancing is challenging on an adaptive grid.
- Verified solutions with large geometries require large computational resources, even with an adaptive method.
  Powers and Paolucci *AIAA J* 2005; Powers *JPP* 2006



"Research needs for future internal combustion engines,"
Physics Today, Nov. 2008, pp 47-52.