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
Virginia Beach, Virginia, October 5-9, 2009

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.

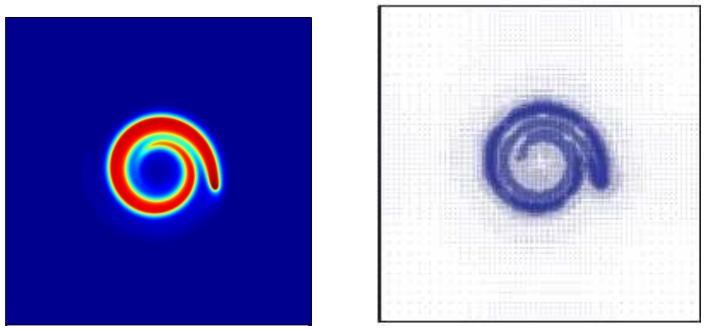


Figure: Flameball-vortex interaction– computed temperature field and adaptive grid.

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

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

Dynamic Spatially 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.)

- 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.
- Recent solutions obtained using machines in the GX cluster at NASA Glenn Research Center.

Intrinsic Low-Dimensional Manifolds (ILDM)

Strang Splitting

- A Strang splitting is included as an option.
- Time integration occurs as a 2-step process:
  - Reaction step – ρ, u, and e are held constant and each spatial point is considered as a homogeneous pre-mixed reactor.
  - Advection-diffusion step – reactive source terms are suppressed, and the inert system is integrated.
- Splitting allows for separate handling of reaction terms, such as with the ILDM method.
- Time-step is controlled by the advection-diffusion step.
- Splitting is second-order accurate in time.

Data Structure Improvements

- New data structure needed to reduce memory footprint and allow straightforward use on parallel architectures.
- Partitioned hash table:
  - Each process has a simple array storing the data elements.
  - Location of elements found by evaluation of hash function (grid indices $\rightarrow$ array index).
  - Hash function generally not injective – collisions (multiple data points mapped to same space in storage array) may occur.
  - Collisions resolved by chaining – creating linked lists from storage array, requires resizing array if chains become too long.
- Constant time data access for random data and worse case proportional to collision chain length.
- Hilbert space-filling curve used as a hash function, may also be used for domain partitioning/load balancing.

Hilbert Space-Filling Curves

- Space-filling curves built from recursive application of a basic pattern.
- Maps n-dimensional points onto a 1-dimensional curve.
- Retains spatial locality – points close in space are close on the curve.
- Constructed by Morton ordering (bit-interleaving) of spatial indices and translation to Hilbert ordering by table lookups.

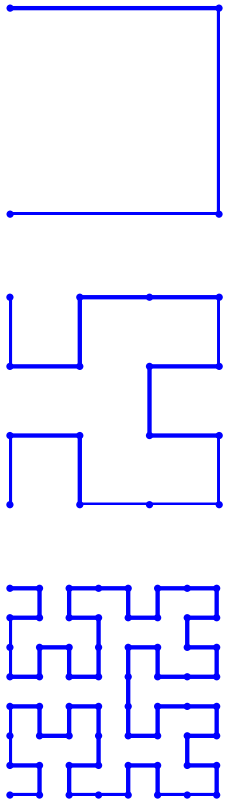
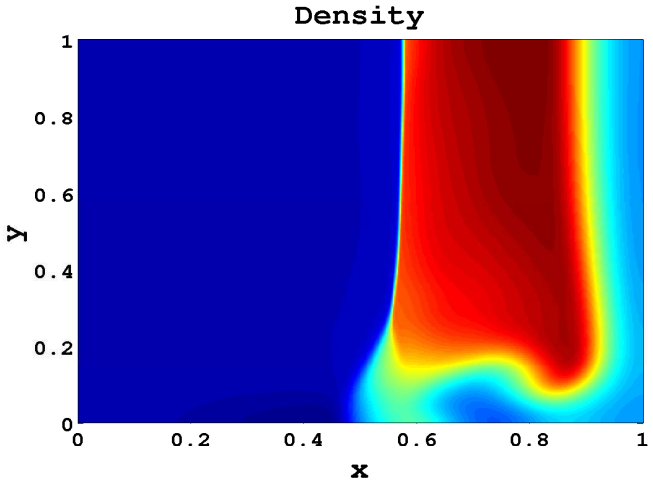


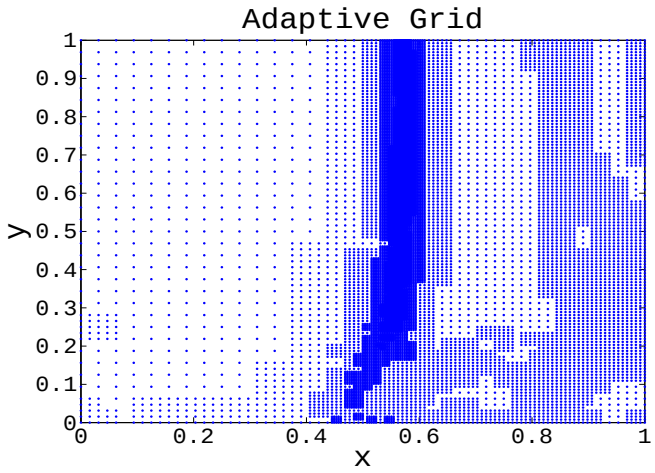
Figure: Three levels of Hilbert curve construction.

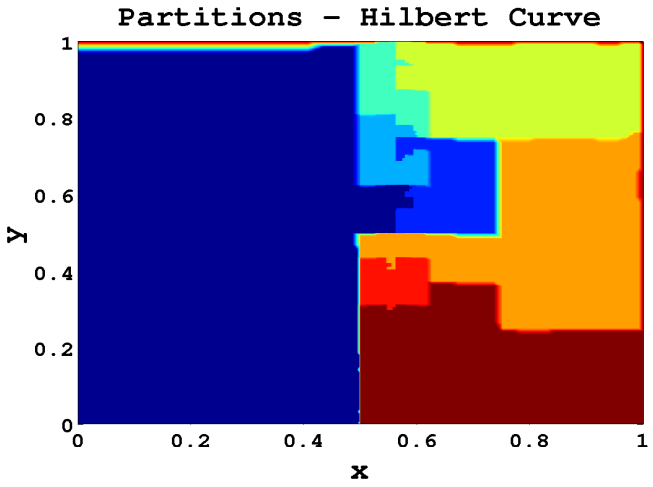
Domain Partitioning and Dynamic Load-Balancing

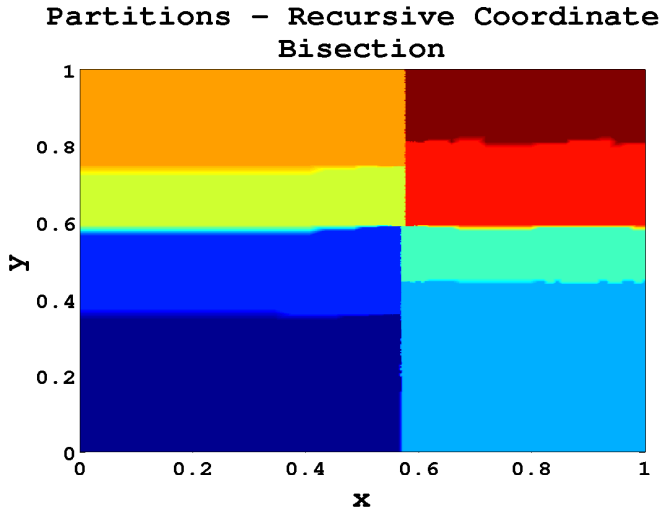
- Needed to equally distribute work among processes and furthermore, to retain equal distribution as points are added or removed during execution.
- Partitioning methods:
  - Bisection – recursively subdivide data along medians of coordinate directions, longest dimension, *etc.*
  - Space-filling curve – trivial partitioning by dividing 1-d curve into equal parts.
- Two major costs:
  - Data movement – changing ownership of a point.
  - Communications – amount of data that must be passed each time-step.

Comparison of Hilbert and Coordinate Bisection

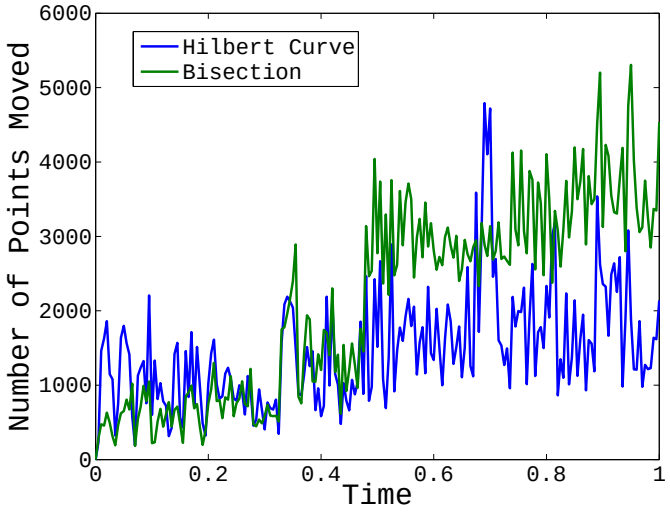


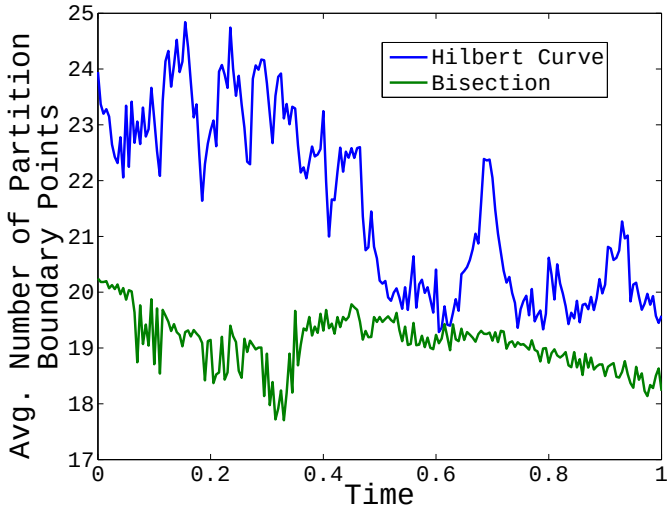


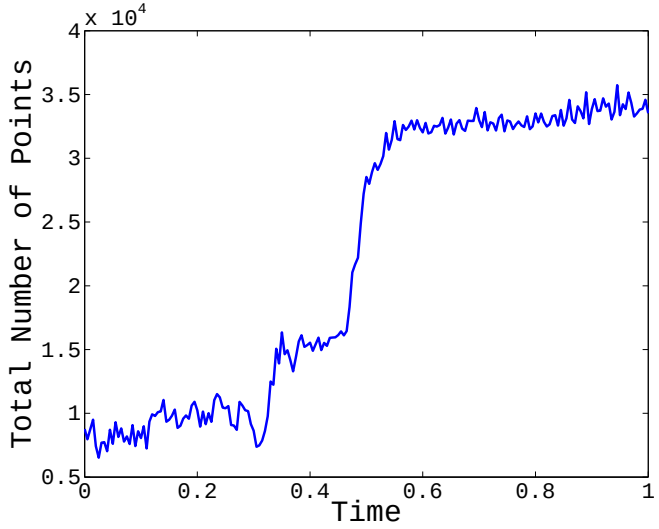




Comparison of Hilbert and Coordinate Bisection

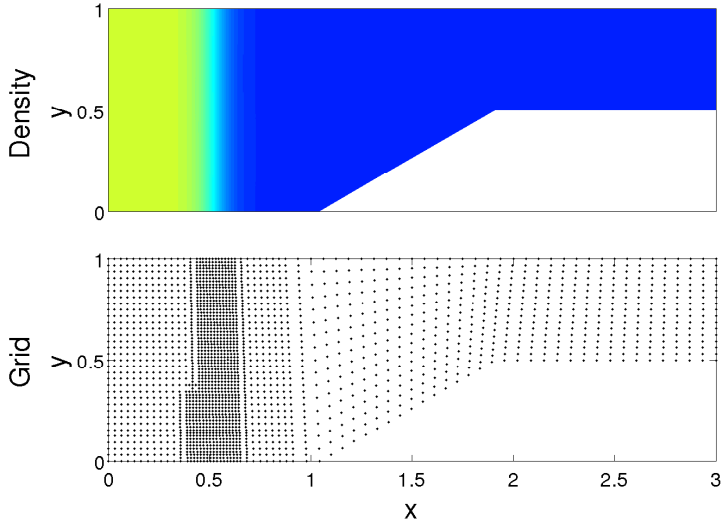






Domain Transformation

Initial conditions:



Geometry:

$0 \leq x < 1.045$
$H = 1.0$

$1.045 \leq x < 1.91$
30°incline

$1.91 \leq x \leq 3.0$
$H = 0.5$

$2H_2 : 1O_2 : 7Ar$ mixture
9 species, inert

State 1: $0 \leq x < 0.5$

$\rho_1 = 2.77 \times 10^{-4}\ gm\ cm^{-3}$
$P_1 = 716400\ dyne\ cm^{-2}$
$u_1 = 59849\ cm\ s^{-1}$

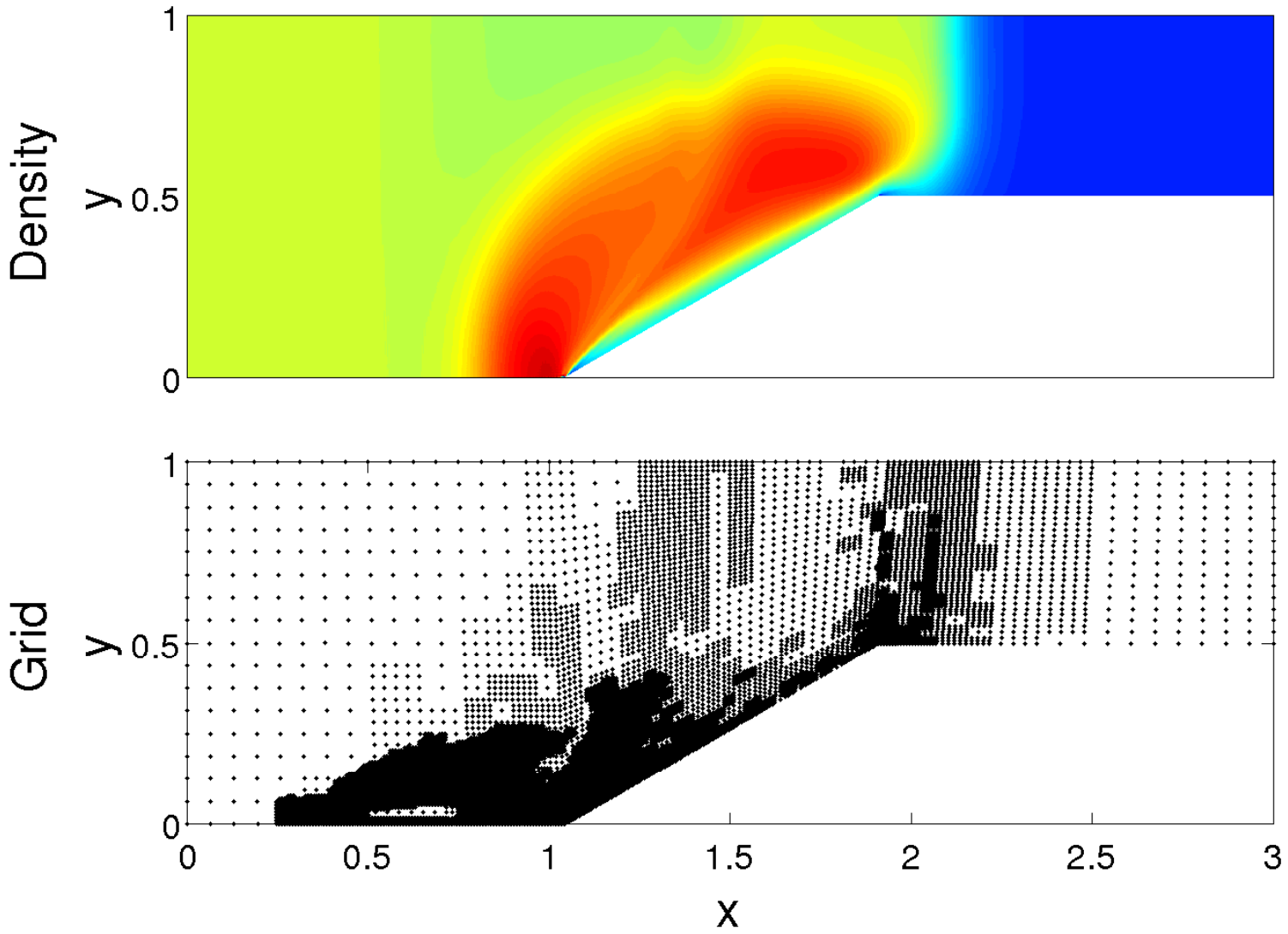
State 2: $0.5 \leq x \leq 3.0$

$\rho_2 = 8.44 \times 10^{-5}\ gm\ cm^{-3}$
$P_2 = 66700\ dyne\ cm^{-2}$
$u_2 = 0\ cm\ s^{-1}$

Wavelet parameters:

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

Domain Transformation (cont.)

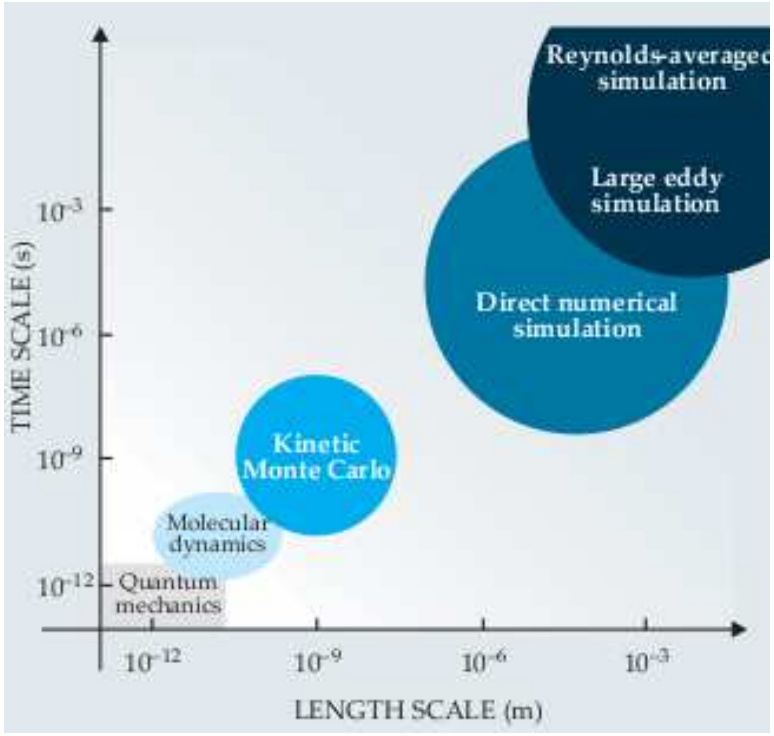


Future Work

- Continue with coarse-grained message passing-based parallelization.
- Include dynamic load-balancing.
- Implement non-reflecting boundary conditions for problems in open domains.
- 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.