A Fully Adaptive Block-Structured Multiresolution Scheme for Reactive Flows

Brandon Gusto

Dept. of Scientific Computing
Florida State University

August 26, 2019

Introduction

Many engineering applications depend on numerically solving systems of conservation laws of the form

$$\begin{cases} u_t + f(u)_x = s(u) \\ u(x,0) = u_0(x), \end{cases} \tag{1}$$

where $x \in \Omega$ and $t \in [t_0, t_f]$. Volume averages are defined for each cell $I_i = \left[x_{i-1/2}, x_{i+1/2}\right]$,

$$\overline{u}_i(t) = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(\xi, t) d\xi, \tag{2}$$

Discretization

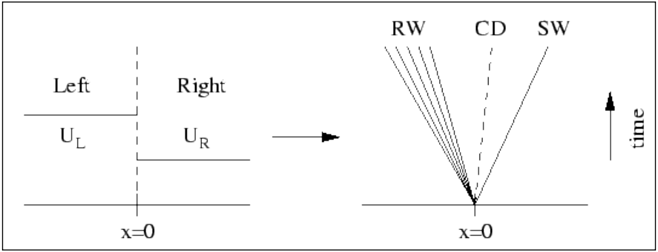
The governing equations are cast into the semi-discrete conservative form,

$$\frac{d\overline{u}_i(t)}{dt} = L(\overline{\boldsymbol{u}}) = -\frac{1}{\Delta x}\left(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}\right) + \overline{s}_i, \tag{3}$$

where $\overline{s} = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} s(u)d\xi$, and the numerical flux is evaluated as

$$\hat{f}_{i\pm1/2} = \hat{f}(\overline{u}^{-}_{i\pm1/2}, \overline{u}^{+}_{i\pm1/2}). \tag{4}$$

Riemann problem



Time integration

This system of ODEs is evolved forward in time

$$\overline{\boldsymbol{u}}^{n+1} = \overline{\boldsymbol{u}}^{n} + \Delta t \sum_{j=1}^{s} b_j \boldsymbol{k}_j, \tag{5}$$

where the stages k_j are solved either implicitly or explicitly.

Motivation for non-uniform refinement

Adaptive mesh refinement

These equations are typically solved on non-uniform mesh spacing:

Adaptive mesh refinement

These equations are typically solved on non-uniform mesh spacing:

- refinement is associated with localized features

Adaptive mesh refinement

These equations are typically solved on non-uniform mesh spacing:

- refinement is associated with localized features
- some type of estimator of the local error is needed

Adaptive mesh refinement

These equations are typically solved on non-uniform mesh spacing:

- refinement is associated with localized features
- some type of estimator of the local error is needed
- typically a collection of cells (a block) is refined for efficiency

Adaptive mesh refinement

These equations are typically solved on non-uniform mesh spacing:

- refinement is associated with localized features
- some type of estimator of the local error is needed
- typically a collection of cells (a block) is refined for efficiency
- blocks introduce inherent "overresolution" in some regions of the mesh

Constraints of AMR

Mesh hierarchy

Define multiple levels of representation of the discrete data

$$\mathcal{G}_l = \left\{ x_i^l \right\}_{i=0}^{N_l}, \quad x_i^l = i \cdot \Delta x_l, \;\; \Delta x_l = 2^{L-l} \cdot \Delta x_L, \;\; N_l = N_L / 2^{L-l},$$

where Δx_l and N_l denote the cell width and number of cells, respectively, on level l. The index space of cells on each level of the hierarchy is denoted by $\mathcal{I}^l = \{1, \ldots, N_l\}$.

Decomposition

Project: The cells at level $l+1$ are projected by means of averaging, onto the coarser grid level l. The projection is defined by a linear operator which performs the mapping $\boldsymbol{P}^{l}_{l+1} : \overline{\boldsymbol{u}}^{l+1} \mapsto \overline{\boldsymbol{u}}^{l}$.

Decomposition

Project: The cells at level $l+1$ are projected by means of averaging, onto the coarser grid level l. The projection is defined by a linear operator which performs the mapping $\boldsymbol{P}_{l+1}^{l} : \overline{\boldsymbol{u}}^{l+1} \mapsto \overline{\boldsymbol{u}}^{l}$.

Predict: Cell averages at level $l+1$ are predicted by an average-interpolating polynomial constructed of cells on level l. The prediction operator performs the mapping $\boldsymbol{P}_{l}^{l+1} : \overline{\boldsymbol{u}}^{l} \mapsto \tilde{\boldsymbol{u}}^{l+1}$.

Project

Coarsening/projection is done via

$$\overline{u}_i^l = \left(\boldsymbol{P}_{l+1}^l \overline{\boldsymbol{u}}^{l+1} \right)_i = \frac{1}{2}(\overline{u}_{2i-1}^{l+1} + \overline{u}_{2i}^{l+1}), \quad \forall i \in \mathcal{I}^l. \tag{6}$$

Predict

Prediction is done using average-interpolating polynomials, with $m \in \{0, 1\}$,

$$\tilde{u}^{l+1}_{2i-m} = \overline{u}^{l}_{i} - (-1)^m \sum_{p=1}^{s} \gamma_p \left(\overline{u}^{l}_{i+p} - \overline{u}^{l}_{i-p} \right), \quad \forall i \in \mathcal{I}^l.$$

Difference information and thresholding

We compute a difference between the true value at the higher resolution $l+1$, and its prediction,

$$d_i^l = \overline{u}_{2i}^{l+1} - \tilde{u}_{2i}^{l+1}, \quad \forall i \in \mathcal{I}^l. \tag{7}$$

Then we can eliminate small detail coefficients (where solution is smooth enough),

$$\tilde{d}_i^l = \begin{cases} d_i^l, \text{ if } |d_i^l| > \varepsilon \\ 0, \text{ if } |d_i^l| \leq \varepsilon. \end{cases} \tag{8}$$

Then reconstructing field is done by

$$\overline{u}_{2i}^{l+1} = \tilde{u}_{2i}^{l+1} + \tilde{d}_i^l.$$

Multiresolution compression of function

Grid adaptation

(MOVIE)

Buffer region

Adaptive flux and source computations on blocks

Where the mask is not active (solution is smooth enough), we can also replace redundant flux calculations with interpolation,

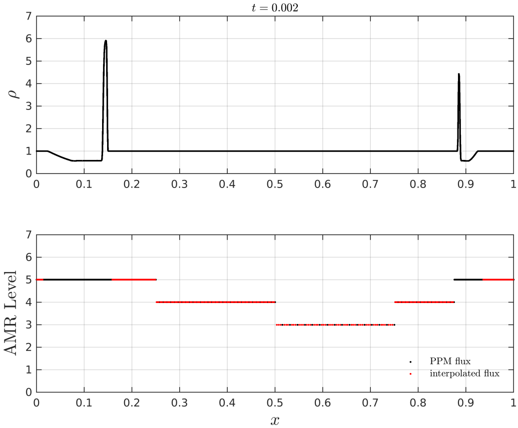
$$\hat{f}^{l+1}_{2i+1} \approx \sum_{p=1}^{s+1} \beta_p \left(\hat{f}^{l}_{i-p+1} + \hat{f}^{l}_{i+p} \right). \tag{9}$$

We can do the same for the source terms,

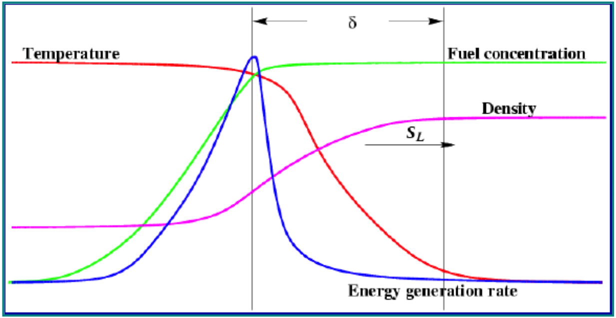
$$\overline{s}^{l+1}_{2i-m} \approx \overline{s}^{l}_{i} - (-1)^m \sum_{p=1}^{k} \gamma_p \left(\overline{s}^{l}_{i+p} - \overline{s}^{l}_{i-p} \right). \tag{10}$$

Load balancing

Two interacting blast waves



Laminar flame



Nuclear burning