

Adaptive Reconnection-based Arbitrary Lagrangian Eulerian Method **A-ReALE**

Mikhail Shashkov, XCP-4, XCP Division, LANL
Wurigen Bo, CCS-2, CCS Division, LANL

Thanks to

S. Sambasivan, S. Diot and S. Mao

R. Loubere, J. Breil, S. Galera, P.-H. Maire, A. Barlow, G. Ball

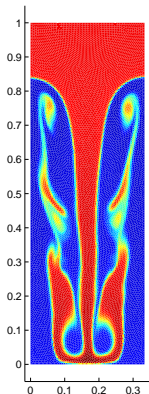
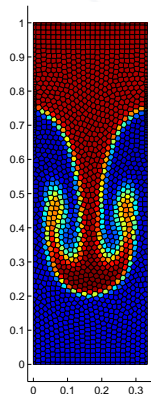
LA-UR-15-25092

Outline

- ReALE
- Adaptive Reconnection-based ALE - A-ReALE
 - A-ReALE Concept
 - Monitor Function, Equidistribution Principle
 - Main tool: Weighted-centroidal Voronoi Tessellation
 - Adaptations strategies: Maintaining spatial resolution
 - How to construct weighted-centroidal Voronoi Tessellation?
 - Numerical Examples, triple point, shock-cavity interaction
- Conclusion and Future Work

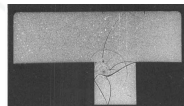
Standard Reconnection-based Arbitrary Lagrangian-Eulerian ReALE Methods

- Lagrangian Phase — General polygonal meshes
- Rezone Phase
 - Allows mesh reconnection —
Voronoi Meshes
 - Mesh Smoothing - Centroidal Voronoi
 - Number of Cell does not change
- Remap Phase
 - Remapping between polygonal meshes with different connectivity
 - Intersection-based Remap

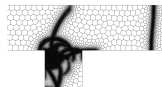
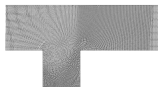


Adaptive Reconnection-based Arbitrary Lagrangian-Eulerian A-ReALE Methods

- Lagrangian Phase — General polygonal meshes
- Rezone Phase
 - Add/Remove Cells - Monitor, Equidistribution, Weighted-centroidal Voronoi
 - Mesh Reconnection - Voronoi
 - Number of Cells **May Change**
- Remap Phase
 - Remapping between polygonal meshes with different connectivity and different number of cells
 - Intersection-based Remap



Experiment



Standard ReALE

A-ReALE

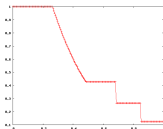
Shock over Cavity

Adaptation - Design Principles

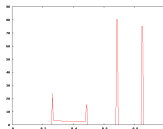
- **Monitor Function** $\varphi(\mathbf{x}, t) > 0$
Problem Dependent and Goal Oriented
 - Interpolation Error, Error Estimation
 - Feature Resolution - shock, contact, interface, boundary-layer
- **Equidistribution Principle** - $\int_{\Omega_i} \varphi(\mathbf{x}, t) dV = E(t)$ - **Equidistribution Level**
- **Adaptation strategies**
 - Fixed number of generators - adapting to monitor using equidistribution principle
 - Maintaining constant equidistribution level during the calculation $E(t) = E(t_0) = \text{const}$ - number of generators changes
 - Maintaining given spatial resolution where monitor reaches its max values - number of generators changes
- **Main Tool - Weighted Centroidal Voronoi Meshes**

Monitor Function

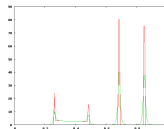
- **Raw Monitor** - Interpolation error: $\varphi = \sqrt{\max(|\lambda_1|, |\lambda_2|)}$
 Hessian $H_u = \begin{pmatrix} \partial^2 u / \partial x^2 & \partial^2 u / \partial x \partial y \\ \partial^2 u / \partial x \partial y & \partial^2 u / \partial y^2 \end{pmatrix} = Q^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} Q$
- **Smoothing** - smoothing and removing noise
- **Shape Preserving Scaling** - increase min and decreases max values - to avoid very small cells - dt ; and very big cells - accuracy



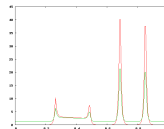
Sod problem



Raw Monitor



Raw & Smoothed



Smoothed & Scaled

Equidistribution Principle

- **Equidistribution Principle**
 - Computational domain Ω ; Monitor function $\varphi(\mathbf{x})$
 - Mesh with cells Ω_i satisfies equidistribution principle if
$$\int_{\Omega_i} \varphi(\mathbf{x}) dV = \text{const} = E \rightarrow \varphi_i |\Omega_i| \approx E$$
 - Large φ_i - small $|\Omega_i|$
- **How to** construct mesh which satisfies equidistribution principle?
 - **Weighted centroidal Voronoi tessellation**

Voronoi, Weighted-Centroidal Voronoi

Set of generators: $\mathbf{g}_i = (x_i, y_i)$

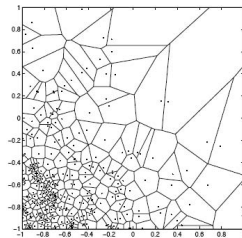
Voronoi cell: $\Omega_i = \{\mathbf{r} = (x, y) : |\mathbf{r} - \mathbf{g}_i| < |\mathbf{r} - \mathbf{g}_j|, \text{ for all } j \neq i\}$

Mass centroid of the cell ($\psi(\mathbf{r}) > 0$ - given weight function)

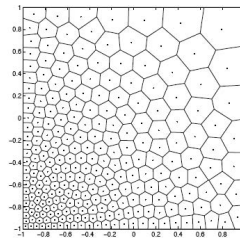
$$\mathbf{c}_i^\psi = \int_{\Omega_i} \mathbf{r} \psi(\mathbf{r}) d\mathbf{r} / \int_{\Omega_i} \psi(\mathbf{r}) d\mathbf{r},$$

If $\mathbf{g}_i = \mathbf{c}_i^\psi$ - **weighted-centroidal Voronoi** tessellation

Left - Random with weight; Right - weighted-centroidal



Random sampling



Centroidal Voronoi

Weighted-Centroidal Voronoi Tessellation

■ Properties:

- Local uniformity - spherical hexagons - accuracy of discretization
- $\psi(\mathbf{g}_i) h_i^{d+2} \approx \psi(\mathbf{g}_j) h_j^{d+2}$, $h_i = 2 \max_{\mathbf{r} \in \Omega_i} \|\mathbf{g}_i - \mathbf{r}\|$
 d dimension of the space

■ How to construct mesh which satisfies equidistribution principle?

- For $d = 2$ - $\psi(\mathbf{g}_i) h_i^4 \approx \psi(\mathbf{g}_j) h_j^4$
- For weighted centroidal Voronoi tessellation in 2D $|\Omega_i| \sim h_i^2$
- $\sqrt{\psi(\mathbf{g}_i)} |\Omega_i| \approx \sqrt{\psi(\mathbf{g}_j)} |\Omega_j|$
- If we set $\psi = \varphi^2$, then $\varphi(\mathbf{g}_i) |\Omega_i| \approx \varphi(\mathbf{g}_j) |\Omega_j|$ - Approximate Equidistribution

Resolution and Required Number of Generators

- If mesh satisfies equidistribution principle then

$$\varphi(\mathbf{g}_i) |\Omega_i| \approx \varphi(\mathbf{g}_j) |\Omega_j| \approx \left(\int_{\Omega} \varphi dV \right) / N_g = E$$

N_g - number of generators, E - Equid. Level

- If we **want to achieve desired Equid. Level E** in domain Ω then we **need**

$$N_g^E \approx \left(\int_{\Omega} \varphi dV \right) / E$$

generators

Adaptation Strategy - Maintaining Spatial Resolution

- Monitor function depends on space and time - φ_i^n
- For meshes which satisfy equidistribution $\varphi_i^n A_i^n = E^n$ and therefore $\varphi_{max}^n A_{min}^n = E^n$
- If we would like to **maintain** constant spatial **resolution** for in regions where **monitor** has its **maximum** values - $A_{min}^n = A_{min} = \text{const}$, then Equid. Level has to be $E^n = \varphi_{max}^n A_{min}$
- Number of generators need to maintain prescribed spatial resolution A_{min}

$$N^n \approx \int_{\Omega(t)} \varphi(\mathbf{r}, t) d\mathbf{r} / (\varphi_{max}^n A_{min})$$

How to construct weighted-centroidal Voronoi tessellation?

■ Variational formulation

— Functional

$$\mathcal{K}(\{\mathbf{x}_i\}_{i=1}^N) = \sum_{i=1}^N \int_{\Omega_i} \varphi(\mathbf{x}) \|\mathbf{x} - \mathbf{x}_i\|^2 d\mathbf{x},$$

$$\min_{\{\mathbf{x}_i\}_{i=1}^N} \mathcal{K}(\{\mathbf{x}_i\}_{i=1}^N), \text{ subject to } \mathbf{x}_j = \mathbf{G}_j, j = M + 1, \dots, N.$$

where Ω_i are the Voronoi cells corresponding to the \mathbf{x}_i .

- A quasi-Newton method, the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS). Need to compute Voronoi tessellation on each iteration
- Derivative

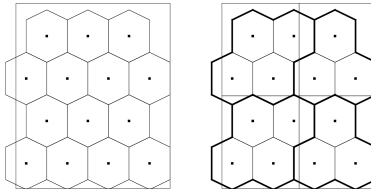
$$\frac{\partial \mathcal{K}}{\partial \mathbf{x}_i} = 2(\mathbf{x}_i - \mathbf{G}_i^*) \int_{\Omega_i} \varphi(\mathbf{x}) d\mathbf{x}$$

where \mathbf{G}_i^* is the mass centroid of Ω_i .

How to construct weighted-centroidal Voronoi tessellation?

■ Initial guess - Quadtree approach

- Total # of required generators $\Phi_i = \int_{\Omega_i^L} \phi(\mathbf{x}) d\mathbf{x}$, $N_{total}^{req} = \left\lfloor \left(\sum_{i=1}^{N_L} \Phi_i \right) / E \right\rfloor$
- Number of generators required in subregion $N_{\sigma_k}^{req} = \left\lfloor \left(\sum_{i: G_i \in \sigma_k} \Phi_i \right) / E \right\rfloor$



(a)

(b)

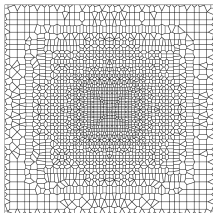
Quad-tree cells and its associated cells. The dots are the centroids of the cells.

(a) a parent quad-tree cell.

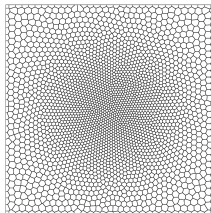
(b) The four children of the parent. The bold lines represent the boundaries of the domains associated with the children.

How to construct weighted-centroidal Voronoi tessellation?

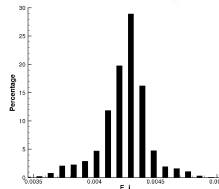
Example



Initial guess



"Equidistributed mesh"

Distribution of E_i

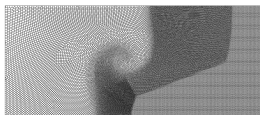
The weight function is given by $\varphi(\mathbf{x}) = (1 + 8e^{-\|\mathbf{x}\|^2/0.32})^2$

The desired equi-distribution level $E = 0.00423$, 2856 generators required

Triple Point Problem - ReALE vs. **A-ReALE**

Approximately the same number number of generators in space-time

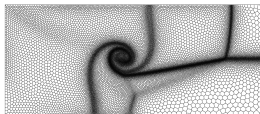
$$N_{\text{ReALE}} \approx 34000, A_{\text{min}}^{\text{A-ReALE}} = 5 \times 10^{-5}$$



ReALE - Mesh, $N_g = 33556$



ReALE - IE



A-ReALE - Mesh, $N_g = 24374$

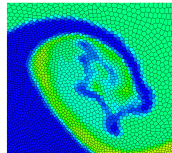
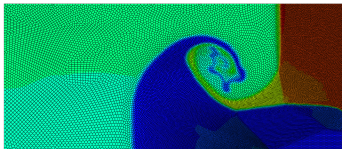


A-ReALE - IE

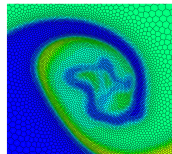
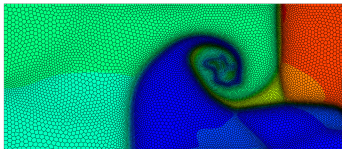
$t = 3$

Triple Point Problem - ReALE vs. **A-ReALE**

Vortex Resolution



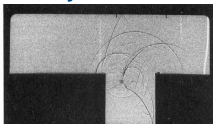
ReALE



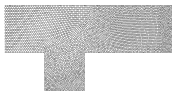
A-ReALE

$t = 5$

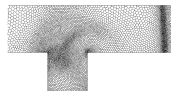
Interaction between a shock and cavity



The shadowgraph from the experimental results - Igra et al.



ReALE, Mesh - $N = 6051$



A-ReALE, Mesh - $A_{min} = 2 \times 10^{-3}$, $N(420) = 6450$



ReALE, $N = 39593$



A-ReALE, $A_{min} = 1.25 \times 10^{-4}$, $N(420) = 33179$

	$N(420)$	$N_{st}(420)$	time steps
ReALE $N = 39593$	39593	159559790	4030
A-ReALE $A_{min} = 1.25 \times 10^{-4}$	33179	177856789	7086

UNCLASSIFIED

Conclusion and Future Work

- Developed new adaptive reconnection based framework
- Demonstrated A-ReALE performance on series of examples
- Future work: Development of parallel version
- Future work: Different monitors

Thanks



Picture by M. Berndt (LANL) from Kokoman Pojoaque Valley, NM Liquor Store

Papers, Reports: https://www.researchgate.net/profile/Mikhail_Shashkov