Quadratic Moment-of-fluid Interface Reconstruction

Jacob Spainhour

Florida State University Department of Mathematics

Undergraduate Honors in the Major Thesis

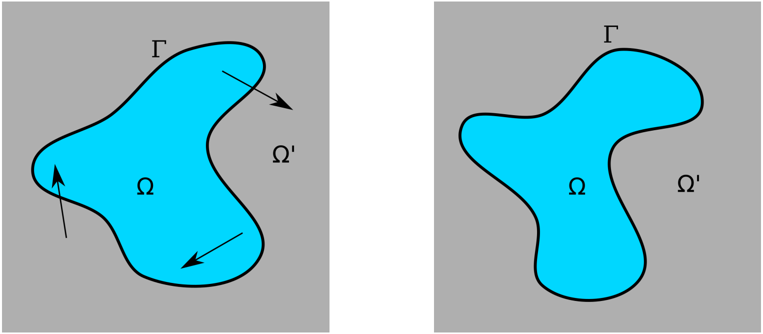
Outline

1. Motivation
2. Quadratic Moment-of-fluid Method
3. Piecewise Parabolic Interface Calculation
4. Numerical Results
5. Conclusions & Future Work

Outline

1. Motivation
   - Interface Reconstruction Problem
   - Timeline of Volume-of-fluid History
   - Timeline of Moment-of-fluid History
2. Quadratic Moment-of-fluid Method
3. Piecewise Parabolic Interface Calculation
4. Numerical Results
5. Conclusions & Future Work

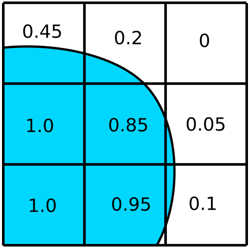
- Fluids are modeled by complicated equations
  - Need discretized simulations to approximate their motion



- Can we reconstruct the interface from discrete data?

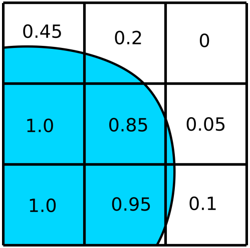
Volume-of-fluid (VOF) stores Volume Fractions

- Piecewise Linear Interface Calculation (PLIC) [Youngs, 1982]
- Piecewise Parabolic Interface Calculation [Price, 2000]
- Quadratic Spline Interpolation [Diwakar et al., 2009]
- Machine Learning Curvature Approximation [Qi et al., 2018]



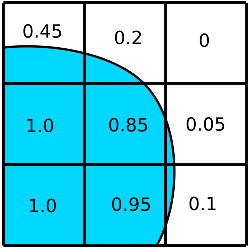
Volume-of-fluid (VOF) stores Volume Fractions

- Piecewise Linear Interface Calculation (PLIC) [Youngs, 1982]
- Piecewise Parabolic Interface Calculation [Price, 2000]
- Quadratic Spline Interpolation [Diwakar et al., 2009]
- Machine Learning Curvature Approximation [Qi et al., 2018]



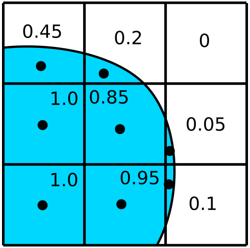
Volume-of-fluid (VOF) stores Volume Fractions

- Piecewise Linear Interface Calculation (PLIC) [Youngs, 1982]
- Piecewise Parabolic Interface Calculation [Price, 2000]
- Quadratic Spline Interpolation [Diwakar et al., 2009]
- Machine Learning Curvature Approximation [Qi et al., 2018]



Moment-of-fluid (MOF) stores Volume Fractions and Centroids

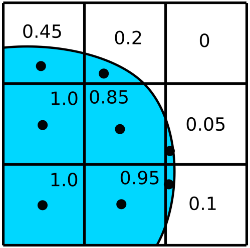
- Moment-of-fluid Method
  [Dyadechko and Sashkov, 2005]
- Adaptive Mesh Refinement Moment-of-fluid
  [Ahn and Shashkov, 2009]
- **Quadratic Moment-of-fluid**



QMOF: Use higher order moments for higher order interfaces

Moment-of-fluid (MOF) stores Volume Fractions and Centroids

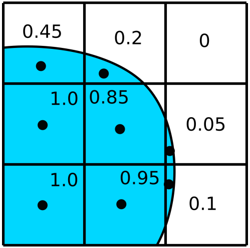
- Moment-of-fluid Method
  [Dyadechko and Sashkov, 2005]
- Adaptive Mesh Refinement
  Moment-of-fluid
  [Ahn and Shashkov, 2009]
- Quadratic Moment-of-fluid



QMOF: Use higher order moments for higher order interfaces

Moment-of-fluid (MOF) stores Volume Fractions and Centroids

- Moment-of-fluid Method
  [Dyadechko and Sashkov, 2005]
- Adaptive Mesh Refinement Moment-of-fluid
  [Ahn and Shashkov, 2009]
- **Quadratic Moment-of-fluid**



QMOF: Use higher order moments for higher order interfaces

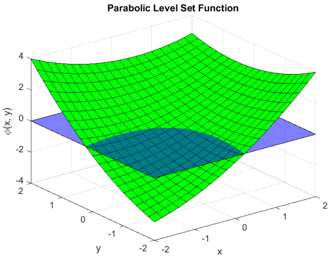
Outline

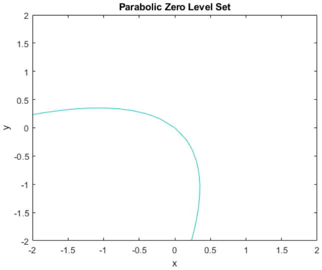
1. Motivation
2. Quadratic Moment-of-fluid Method
   - Theoretical Framework
   - Conditions for Moment-of-fluid Methods
   - Quadrature for Implicitly Defined Volumes
   - Time Advancing of Moment Data
3. Piecewise Parabolic Interface Calculation
4. Numerical Results
5. Conclusions & Future Work

- U: A computational cell
- Ω: The fluid interior to U
- Γ: The boundary of Ω interior to U

Define the approximation with a level set function ϕ where

$$\overline{\Omega} = \{(x, y) : \phi(x, y) < 0\} \cup U \qquad \overline{\Gamma} = \{(x, y) : \phi(x, y) = 0\}$$





Find the function ϕ that satisfies the following conditions:

$$\begin{aligned} \phi(x,y) = \phi(a_0,a_1,a_2) &= a_0x + a_1y + a_2, \quad ||\vec{a}|| = 1 \\ \overline{M} &= M \\ \phi &= \arg\min_{\phi}\{||M_x - \overline{M_x}|| + ||M_y - \overline{M_y}||\} \end{aligned}$$

where M, M_x, and M_y represent the reference moment data, and $\overline{M}$, $\overline{M_x}$, and $\overline{M_y}$ represent the moment data of the approximation.

Find the function ϕ that satisfies the following conditions:

$$\phi(x, y) = \phi(\vec{a}) = a_0x^2 + a_1y^2 + a_2xy + a_3x + a_4y + a_5$$

$$\text{where} \quad \vec{a} = [a_0, a_1, a_2, a_3, a_4, a_5]^T, ||\vec{a}|| = 1$$

$$\overline{M} = M$$

$$\phi = \arg\min_{\phi}\{||M_{x^2} - \overline{M_{x^2}}|| + ||M_{y^2} - \overline{M_{y^2}}|| + ||M_{xy} - \overline{M_{xy}}|| + \dots$$
$$||M_x - \overline{M_x}|| + ||M_y - \overline{M_y}||\}$$

where M, M_x, M_y, M_{xy}, M_{x^2}, and M_{y^2} represent the reference moment data, and $\overline{M}$, $\overline{M_x}$, $\overline{M_y}$, $\overline{M_{xy}}$, $\overline{M_{x^2}}$, and $\overline{M_{y^2}}$ represent the moment data of the approximation.

- 0^{th} order moment (volume):

$$M = \int_{\Omega} dxdy$$

- 0^{th} order moment (volume):

$$M = \int_\Omega dxdy$$

- 1^{st} order moments (centroid):

$$M_x = \frac{\int_\Omega x\, dxdy}{M} \quad M_y = \frac{\int_\Omega y\, dxdy}{M}$$

- 0^{th} order moment (volume):

$$M = \int_{\Omega} dxdy$$

- 1^{st} order moments (centroid):

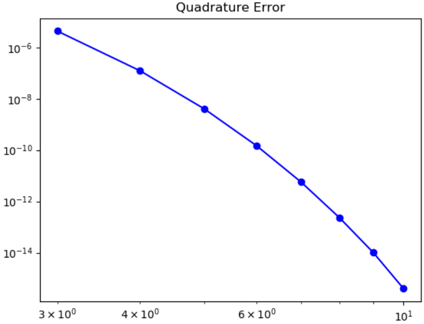
$$M_x = \frac{\int_{\Omega} x\,dxdy}{M} \quad M_y = \frac{\int_{\Omega} y\,dxdy}{M}$$

- 2^{nd} order moments:

$$M_{x^2} = \frac{\int_{\Omega} x^2\,dxdy}{M} \quad M_{xy} = \frac{\int_{\Omega} xy\,dxdy}{M} \quad M_{y^2} = \frac{\int_{\Omega} y^2\,dxdy}{M}$$

Moments of approximate fluid $\overline{\Omega}$ can be calculated analogously

- Recursively decompose the domain into height functions
- Perform Gaussian quadrature on each height function
- Implementation specialized for two-dimensional, second order level set functions [Saye, 2015]



- Fluid solver needs to be able to advance moment data

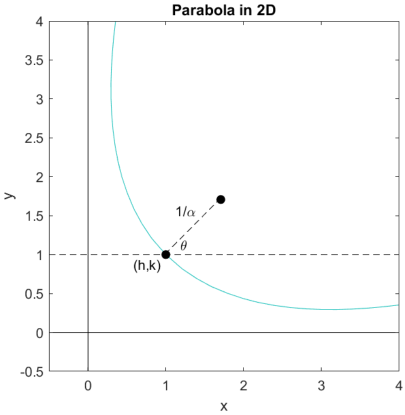
- Algorithm taken from Shashkov and Dyadechko [Dyadechko and Sashkov, 2005]
  1. Create Lagrangian Prototype: trace cells backwards in time
  2. Compute moment data for each prototype
  3. Transform moment data to current domain

- Each step needs to preserve second order accuracy

Outline

1. Motivation
2. Quadratic Moment-of-fluid Method
3. Piecewise Parabolic Interface Calculation
   - Level Set Construction
   - Problem Formulation
   - Initial Condition: Linear Moment-of-fluid
   - Initial Condition: Radial Basis Function Network
   - Optimization Algorithm
4. Numerical Results
5. Conclusions & Future Work

- Arbitrary quadratic curves are unnecessarily complicated
- Use parabolas: simpler parameterization than coefficients
  - Vertex (h, k), focus parameter α, angle of rotation θ



- Given a vector of moment data $\vec{M} = [M_{x^2}, M_{y^2}, M_{xy}, M_x, M_y, M]^T$, find the vector of level set function parameters $\vec{a}$ that minimizes the following cost function:

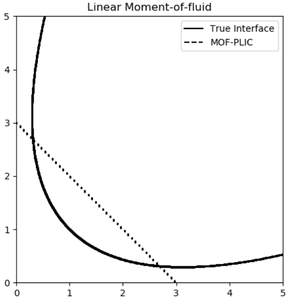
$$\begin{aligned} C(\vec{a}) =& (M_{x^2} - \overline{M_{x^2}})^2 + (M_{y^2} - \overline{M_{y^2}})^2 + (M_{xy} - \overline{M_{xy}})^2 \\ &+ (M_x - \overline{M_x})^2 + (M_y - \overline{M_y})^2 \end{aligned}$$

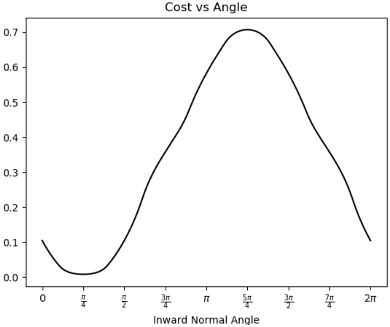
such that

$$\overline{M} = M$$

Transform $\vec{M}$ to $\vec{\widetilde{M}}$: equivalent moment data on the unit square $\widetilde{U} = [0,1]^2$.
Intelligently produce initial level set data $\vec{a}_0$.
while Error tolerance is not reached **do**
    Choose a search direction ∇C, step length γ
    $\vec{a}_{n+1} \leftarrow \vec{a}_n - \gamma \nabla C(\vec{a}_n)$
    Perform *flooding algorithm* on $\vec{a}_{n+1}$ to maintain volume
end while
Transform level set parameters $\vec{a}_n$ on $\widetilde{U}$ back to equivalent parameters on U.

- Properties of C are obscure: prone to local minima
  - Initialize QMOF with solution to MOF
- MOF reduces to a bounded, 1D optimization problem in θ
  - Computed with Golden Section search





- Overall, we want to solve $F(\vec{M}) = \vec{a}$ when we know F^{-1}
- Use three-layer (6-N-4) RBFN to approximate F [Wu et al., 2012]
  - Train network on randomly generated parabolas
  - Cluster resulting moment data into N hidden nodes
  - Interpolate reference data $\vec{M}$ among clusters to find an approximate parabola $\vec{a}$
- Training RBFN requires solving a dense linear system
  - Only needs to be done once across all runs of algorithm
- Converges significantly faster, but unreliably

- Given initial parameters $\vec{a}_0$, cost function $C(\vec{a})$, initial step size γ_0, and backtracking parameter $\beta \in (0, 1)$, compute optimal vector $\vec{a}$ that minimizes C

$\vec{a} \leftarrow \vec{a}_0$
$\gamma \leftarrow \gamma_0$
while $||\nabla C|| >$ error tolerance **do**
  while $C(\vec{a} - \gamma \nabla C(\vec{a})) \geq C(\vec{a})$ **do**
    $\gamma \leftarrow \beta \cdot \gamma$
  end while
  $\vec{a} \leftarrow \vec{a} - \gamma \nabla C(\vec{a})$
  flooding$(\vec{a}; M)$
  $\gamma \leftarrow \gamma_0$
end while

- Need to keep volume M constant across the algorithm
  - Done by adjusting the intercept of level set function:

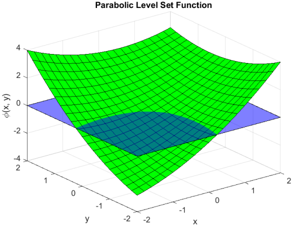
$$\phi(x,y) = \phi(\vec{a},M) = a_0x^2 + a_1y^2 + a_2xy + a_3x + a_4y + V(\vec{a},M)$$

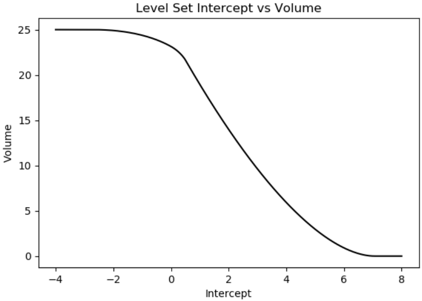
- Can calculate $V(\vec{a},M)$ using solution to bisection problem

$$g(a_5) = \overline{M} - M = \int_{\overline{\Omega}} dxdy - M = 0$$

for level set function

$$\phi(x,y) = a_0x^2 + a_1y^2 + a_2xy + a_3x + a_4y + a_5$$





- Volume is monotonic and bounded
  - Can find initial bracketing interval algorithmically

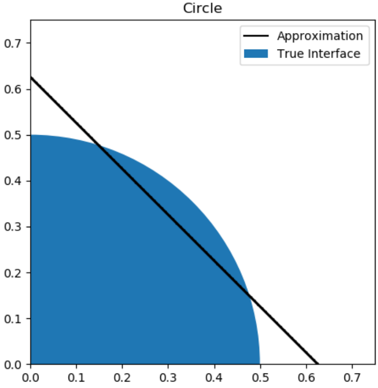
- We can reasonably assume C is continuous for parabolas
- Compute search direction with centered difference

$$\frac{\partial C(\vec{a})}{\partial a_j} \approx \frac{-C(a_j+2h)+8C(a_j+h)-8C(a_j-h)+C(a_j-2h)}{12h}$$

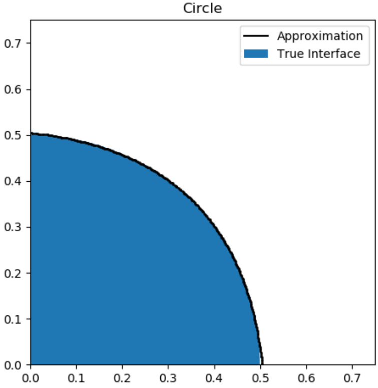
where $C(a_j)$ is shorthand for $C(a_0,\ldots,a_j,\ldots,a_n)$

Outline

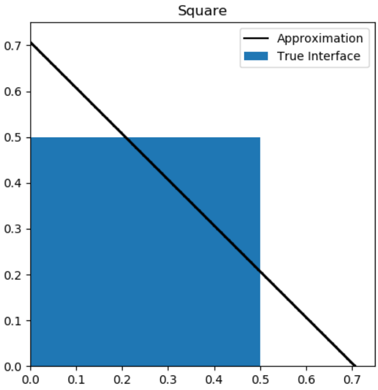
1. Motivation
2. Quadratic Moment-of-fluid Method
3. Piecewise Parabolic Interface Calculation
4. Numerical Results
   - Interface Accuracy: MOF vs QMOF
   - Initial Condition vs. Steepest Descent
   - Grid Refinement Tests
   - Curvature Approximation
5. Conclusions & Future Work



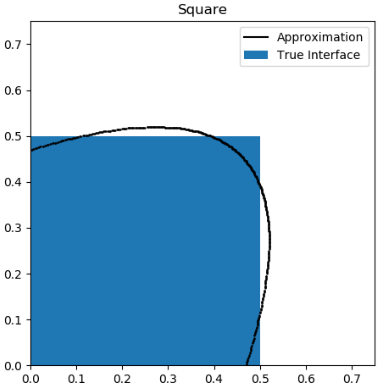
Linear MOF



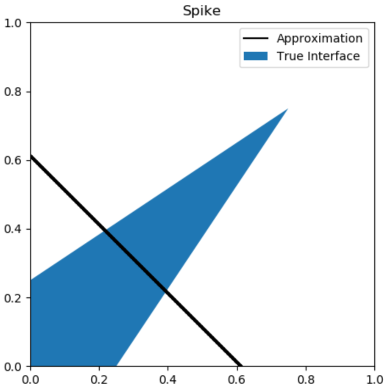
Parabolic QMOF



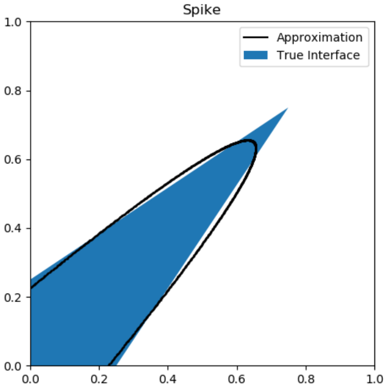
Linear MOF



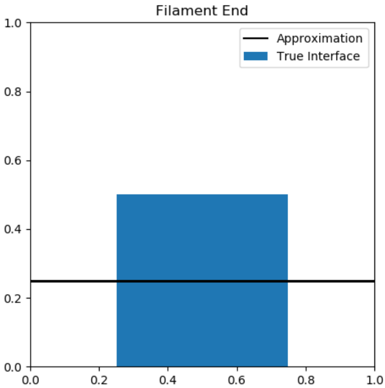
Parabolic QMOF



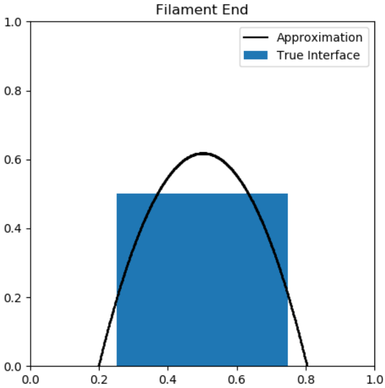
Linear MOF



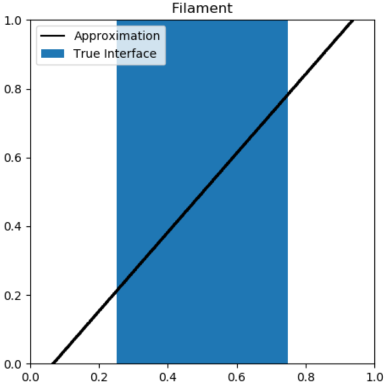
Parabolic QMOF



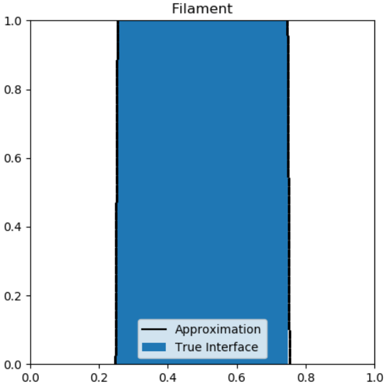
Linear MOF



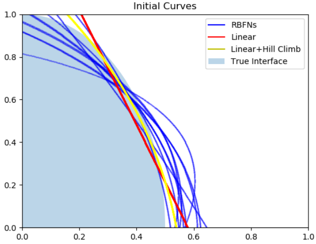
Parabolic QMOF

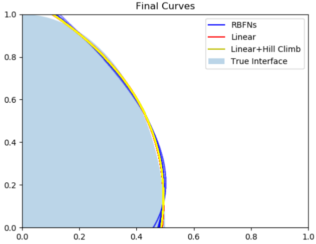


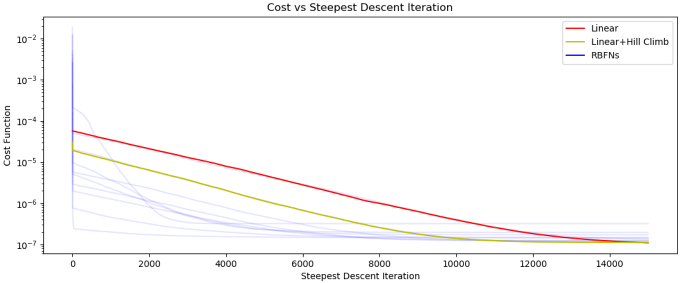
Linear MOF

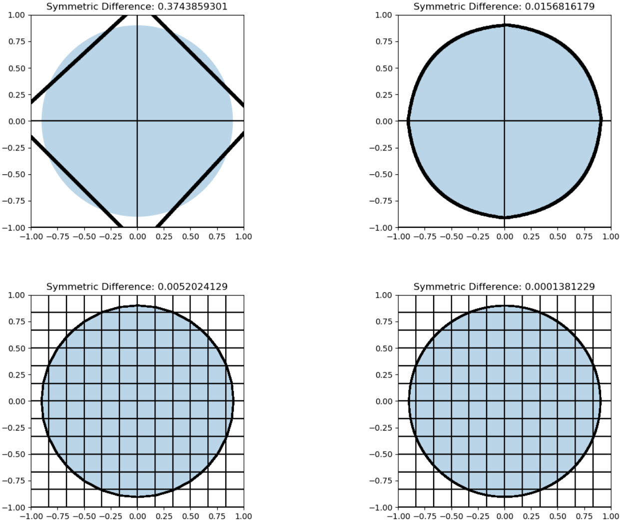


Parabolic QMOF

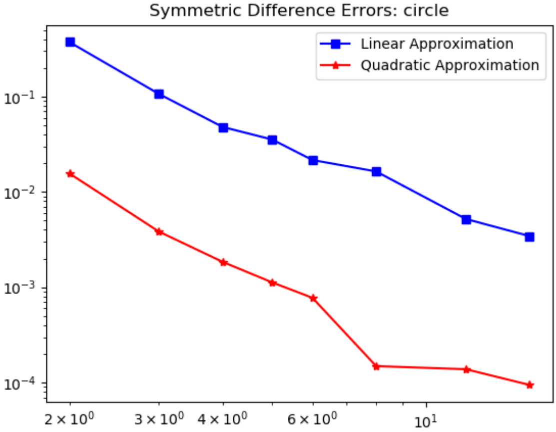

Initial Curves
RBFNs
Linear
Linear+Hill Climb
True Interface
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

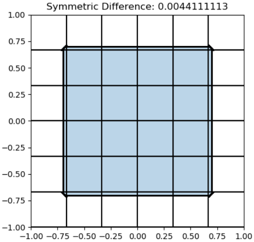

Final Curves
RBFNs
Linear
Linear+Hill Climb
True Interface
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

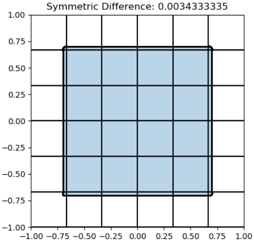

Cost vs Steepest Descent Iteration
Linear
Linear+Hill Climb
RBFNs
Cost Function
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
0
2000
4000
6000
8000
10000
12000
14000
Steepest Descent Iteration

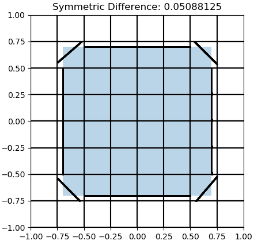

Symmetric Difference: 0.3743859301
Symmetric Difference: 0.0156816179
Symmetric Difference: 0.0052024129
Symmetric Difference: 0.0001381229
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

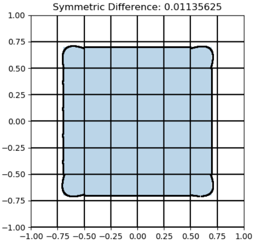
- Second Order Convergence, 4 times as effective as MOF

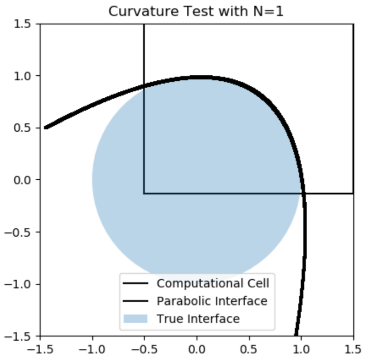



Symmetric Difference: 0.0044111113
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

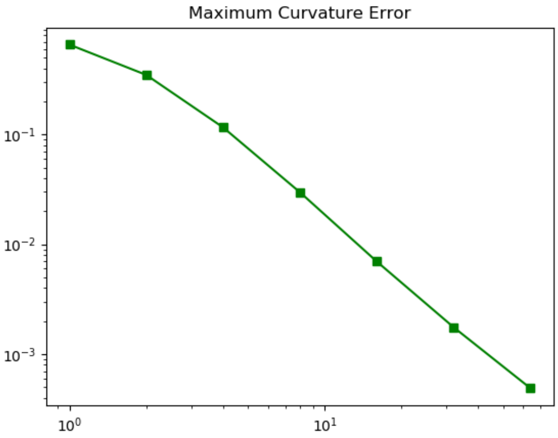

Symmetric Difference: 0.0034333335
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00


Symmetric Difference: 0.05088125
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00


Symmetric Difference: 0.01135625
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00


Curvature Test with N=1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5
Computational Cell
Parabolic Interface
True Interface
Curvature Test with N=4
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5
Computational Cell
Parabolic Interface
True Interface

- First order approximation of curvature



Outline

1 Motivation

2 Quadratic Moment-of-fluid Method

3 Piecewise Parabolic Interface Calculation

4 Numerical Results

5 Conclusions & Future Work
- Conclusions
- Future Work
- References

- A novel Piecewise Parabolic Interface Calculation method built on top of Moment-of-fluid, bringing locality advantages
- The second order interface is more accurate and can approximate subgrid features
- Second order reconstruction allows for accurate approximation of curvature

- Implement dynamic tests for time-advanced fluid
- Improve efficacy of Radial Basis Function Network
- Test alternative, gradient-free optimization methods

[Ahn and Shashkov, 2009] Ahn, H. T. and Shashkov, M. (2009).
Adaptive moment-of-fluid method for multi-material flow.
In Choi, H., Choi, H. G., and Yoo, J. Y., editors, *Computational Fluid Dynamics 2008*, pages 567–572, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Diwakar et al., 2009] Diwakar, S., Das, S. K., and Sundararajan, T. (2009).
A quadratic spline based interface (quasi) reconstruction algorithm for accurate tracking of two-phase flows.
Journal of Computational Physics, 228(24):9107–9130.

[Dyadechko and Sashkov, 2005] Dyadechko, V. and Sashkov, M. (2005).
Moment-of-fluid interface reconstruction.
Los Alamos Report LA-UR-05-7571.

[Noh and Woodward, 1976] Noh, W. F. and Woodward, P. (1976).
Slic (simple line interface calculation).
In van de Vooren, A. I. and Zandbergen, P. J., editors, *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2, 1976 Twente University, Enschede*, pages 330–340, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Price, 2000] Price, G. R. (2000).
A piecewise parabolic volume tracking method for the numerical simulation of interfacial flows.
Calgary.

[Qi et al., 2018] Qi, Y., Lu, J., Scardovelli, R., Zaleski, S., and Tryggvason, G. (2018).
Computing curvature for volume of fluid methods using machine learning.

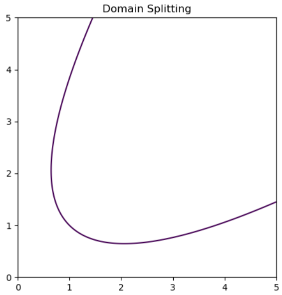
[Saye, 2015] Saye, R. (2015).
High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles.
SIAM Journal on Scientific Computing, 37(2):A993–A1019.

[Wu et al., 2012] Wu, Y., Wang, H., Zhang, B., and Du, K.-L. (2012).
Using radial basis function networks for function approximation and classification.
ISRN Applied Mathematics, 2012.

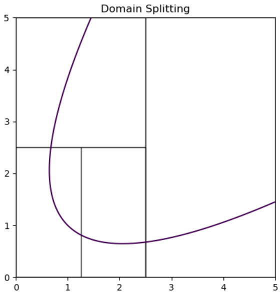
[Youngs, 1982] Youngs, D. (1982).
Time-Dependent Multi-material Flow with Large Fluid Distortion, volume 24, pages 273–285.

[Zinjala and Banerjee, 2017] Zinjala, H. K. and Banerjee, J. (2017).
Refined moment-of-fluid method.
Numerical Heat Transfer, Part B: Fundamentals, 71(6):574–591.

Recursive Quadrature Algorithm Example

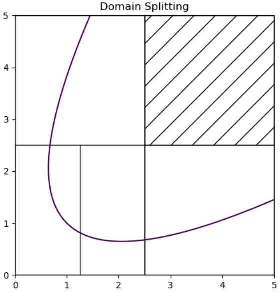


Initial Curve

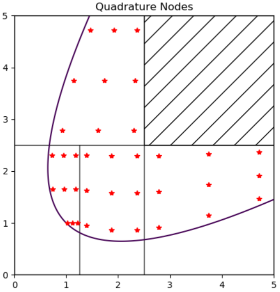


Subdivide Domain

Recursive Quadrature Algorithm Example



Identify Empty/Full Domains



Place Quadrature Nodes

Derivation of rotated parabola

- Begin with standard form of parabola, rotate around (h, k)

$$
\begin{aligned}
(y-k)^2 &= 4a(x-h) \\
\phi(x,y) &= \frac{\alpha}{4}(y-k)^2 - (x-h)
\end{aligned}
$$

$$
\begin{aligned}
x &\rightarrow (y-k)\sin(\theta) + (x-h)\cos(\theta) + h \\
y &\rightarrow (y-k)\cos(\theta) - (x-h)\sin(\theta) + k
\end{aligned}
$$

$$
\begin{aligned}
\phi(x,y) = \frac{\alpha}{4}&((y-k)\cos(\theta) - (x-h)\sin(\theta))^2 \\
&-((y-k)\sin(\theta) + (x-h)\cos(\theta))
\end{aligned}
$$

Can reorganize ϕ as coefficients for a second order polynomial

Maps data from $[x_l, x_u] \times [y_l, y_u]$ to $[0, 1]^2$

$$\widetilde{M} = \frac{M}{(x_u - x_l)(y_u - y_l)} \qquad \widetilde{M_{x^2}} = \frac{M_{x^2} - 2x_l M_x + x_l^2}{(x_u - x_l)^2}$$

$$\widetilde{M_x} = \frac{M_x - x_l}{x_u - x_l} \qquad \widetilde{M_{y^2}} = \frac{M_{y^2} - 2y_l M_y + y_l^2}{(y_u - y_l)^2}$$

$$\widetilde{M_y} = \frac{M_y - y_l}{y_u - y_l} \qquad \widetilde{M_{xy}} = \frac{M_{xy} - y_l M_x - x_l M_y + x_l y_l}{(x_u - x_l)(y_u - y_l)}$$

Map level set coefficients from $[0,1]^2$ to $[x_l, x_u] \times [y_l, y_u]$

$$a_0 = \frac{\widetilde{a}_0}{(x_u - x_l)^2}$$

$$a_1 = \frac{\widetilde{a}_1}{(y_u - y_l)^2}$$

$$a_2 = \frac{\widetilde{a}_2}{(x_u - x_l)(y_u - y_l)}$$

$$a_3 = \frac{-2\widetilde{a}_0 x_l}{(x_u - x_l)^2} - \frac{\widetilde{a}_2 y_l}{(x_u - x_l)(y_u - y_l)} + \widetilde{a}_3$$

$$a_4 = \frac{-2\widetilde{a}_1 y_l}{(y_u - y_l)^2} - \frac{\widetilde{a}_2 x_l}{(x_u - x_l)(y_u - y_l)} + \widetilde{a}_4$$

$$a_5 = \frac{\widetilde{a}_0 x_l^2}{(x_u - x_l)^2} + \frac{\widetilde{a}_1 y_l^2}{(y_u - y_l)^2} + \frac{\widetilde{a}_2 x_l y_l}{(x_u - x_l)(y_u - y_l)} - \frac{\widetilde{a}_3 x_l}{x_u - x_l} - \frac{\widetilde{a}_4 y_l}{y_u - y_l} + \widetilde{a}_5$$