# Triple junction simulation using acceleration dependent BMO

Reza F. Arifin [a,b]

[a]Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jl. Ganesha 10, Bandung 40132 Indonesia
[b]Institute of Science and Engineering, Kanazawa University, Kakuma, Kanazawa 920-1192 Japan
E-mail: reza_1622@students.itb.ac.id

**Abstract.** *We develop a method to simulate multiple bubbles and the triple junction motion in two dimensions driven by the mean curvature acceleration. The core of this method is the acceleration dependent BMO algorithm in a vector-valued formulation. To avoid a discontinuity in the vector-valued function, we adopt a signed distance vector formulation. Moreover we also develop the method for the area preserving motion.*

**Keywords:** BMO algorithm, acceleration dependent BMO, triple juction, mean curvature acceleration, hyperbolic mean curvature flow.

## 1 Introduction

Bubbles motion phenomenon has become an interesting research objective and there are many models and methods to approach such motions. One of the methods is called Bence-Merriman-Osher (BMO) algorithm. The original BMO algorithm [1] was introduced for realizing interfacial motion by mean curvature flow. Ginder and Svadlenka [2] introduced the modified version of the BMO algorithm for curvature-dependent interfacial acceleration. The method uses mean curvature acceleration instead of mean curvature flow on the surfaces to evolve.

In particular we consider the motion of interface governed by equation below

$$A = -\kappa \mathbf{n} \tag{1}$$

where $A$ is acceleration of interface, $\kappa$ is the mean curvature and $\mathbf{n}$ is unit normal vector.

We compare this method with Runge-Kutta fourth order for shrinking circle problem to see behaviour of the result. In this research, we also implement the method for multiple bubbles. In such phenomenon, we have to deal with triple junction. In order to handle multiple bubbles, we implement vector-valued BMO.

## 2 General model

In general we assume that the interface of bubble on certain point is moving accelerated by its mean curvature. The direction of its movement is on the opposite direction of normal vector. We can write the problem as bellow.

Let $\alpha$ be a position function and $\mathbf{n}$ be a unit normal vector, then

$$\begin{cases} \alpha_{tt} = -\kappa \mathbf{n}, \\ \alpha_t(t=0,s) = v_0(s), \quad s \in [0,1) \\ \alpha(t=0,s) = \gamma(s), \quad s \in [0,1). \end{cases} \tag{2}$$

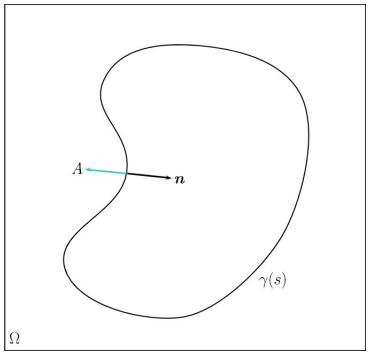



Figure 1: General model

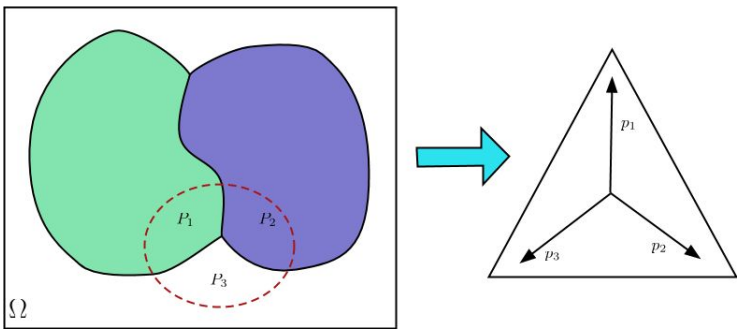



Figure 2: Three regions

where $\gamma(s)$ is a parameterized closed curve with initial velocity $v_0$ and $\kappa$ is the mean curvature.

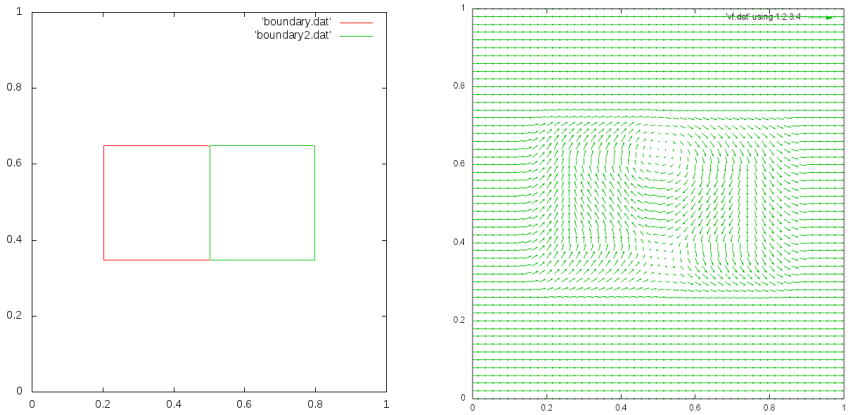
For multiple bubbles, each region $P_i$ is assigned by reference vector $p_i$(see Figure.2). The way to construct reference vector we can see further at [3]. Each reference vector represent corresponding region. Analog with characteristic function in the original BMO [1], 0 and 1 represent two different regions.

## 3 Method

We approximate the motion with thresholding the solution to the wave equation which evolves from its initial condition. For one bubble case, it uses signed distance function constructed by its initial region. In the similar ways, for two or more bubbles, we implement signed distance vector [4] defined by,

**Definition 3.1** *(Signed distance vector)* For $m$ number of region and $\varepsilon > 0$, we define the *signed distance vector* $\mathbf{d}_\varepsilon : \mathbb{R}^N \to \mathbb{R}^{m-1}$ by:

$$\mathbf{d}_\varepsilon(x) := \sum_{i=1}^{m} \left[1 - \min\left(1, \frac{d_i(x)}{\varepsilon}\right)\right]\mathbf{p}_i,$$




Three phases region Three phases SDV

Figure 3: Signed distance vector

where $d_i(\cdot) := \text{dist}(\cdot, P_i)$ is distance to region $P_i$ and $\mathbf{p}_i$ is reference vector corresponding to each region $P_i$. For example we can see three region signed distance vector at Figure.3.

## 3.1 Acceleration dependent BMO algorithm

Here is the algorithm for Acceleration dependent BMO.

- For given time $T$, take $h = T/M$, where $M$ is positive integer.
- Define $n-1$ dimensional reference vectors $\mathbf{p}_i$ for each corresponding to region $P_i$, $i = 1, 2, ..., n$.
- Set the signed distance vector using initial condition $\mathbf{d}_0$ and prepare $\mathbf{d}_{-1}$ obtained from the initial velocity along the interface.
- For $k = 0, 1, ..., M$
  1. Set $\mathbf{u}_0(x) = 2\mathbf{d}_k - \mathbf{d}_{k-1}$.
  2. Solve the vector-valued wave equation with initial condition $\mathbf{u}_0$ for time interval $h$ and zero initial velocity
$$\begin{cases} \mathbf{u}_{tt} = \Delta \mathbf{u} & \text{in } (0,h) \times \Omega \\ \frac{\partial \mathbf{u}}{\partial \nu} = 0 & \text{on } (0,h) \times \partial\Omega \\ \mathbf{u}(t=0,x) = \mathbf{u}_0 & \text{in } \Omega \\ \mathbf{u}_t(t=0,x) = 0 & \text{in } \Omega \end{cases}$$
  3. Update each regions and set signed distance vector $\mathbf{d}_{k+1}$.

To update region, here we use "closest vector" to the reference vector. For illustration we can see Figure.4. Here we have $\mathbf{u}(h,x)$ as a solution of wave equation, then we update $\mathbf{u}_0$ as

$$\mathbf{u}_0 = p_j, \text{where } \mathbf{p}_j \cdot \mathbf{u}(h,x) = \max_{i=1,2,...,n} \mathbf{p}_i \cdot \mathbf{u}(h,x)$$

After we determine phase region each nodes on the domain, we set signed distance vector by Definition.3.1.

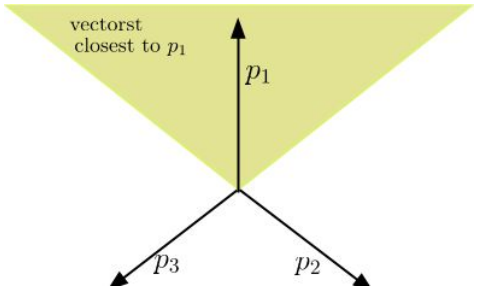



Figure 4: Updating region to closest vector

### 3.2 Minimizing movements

To solve wave equation, we implement the discrete Morse flow method. Let $\tau = h/N$ be a time discretization for given positive integer $N$. For each $n = 1, \ldots, N$, functional $\mathcal{J}_n(\mathbf{u})$ defined over $H^1(\Omega; \mathbb{R}^{n-1})$:

$$\mathcal{J}_n(\mathbf{u}) = \int_\Omega \left( \frac{|\mathbf{u} - 2\mathbf{u}_{n-1} + \mathbf{u}_{n-2}|^2}{2\tau^2} + \frac{|\nabla \mathbf{u}|^2}{2} \right) dx, \tag{3}$$

where $\mathbf{u}_{n-1}$ and $\mathbf{u}_{n-2}$ are given functions. To preserve the area, we include a constraint via penalization,

$$\mathcal{F}_n(\mathbf{u}) = \mathcal{J}_n(\mathbf{u}) + \frac{1}{\varepsilon} \sum_{i=1}^{3} |V_i - \text{meas}(P_i^{\mathbf{u}})|^2 \tag{4}$$

where $\varepsilon > 0$ is a small penalty parameter and $V_i$ is prescribed area of region $P_i$.

We approximate the functional (3) and (4) by using piecewise linear finite element. To find minimizer of the functional here we use conjugate gradient method.

## 4 Numerical Result

Before we see numerical result, we do a numerical test for the method by comparing with Runge-Kutta fourth order for shrinking circle problem.

### 4.1 Numerical Test

The shrinking circle problem satisfies (5),

$$\begin{cases} r_{tt}(t) = -\frac{1}{r(t)}, & t \in [0, T) \\ r_t(0) = 0, \\ r(0) = 0.3, \end{cases} \tag{5}$$

For testing numerical result, we use various of parameters resolution and space/time. We measure the error by computing time-average of the absolute difference between radius of numerical result and Runge-Kutta fourth order. For Runge-Kutta fourth order, we use very small time step so it can represent the exact solution.

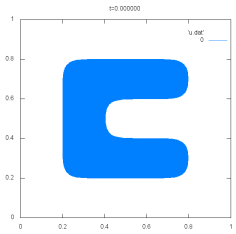
Table 1: Comparison with Runge-Kutta fourth order

| resolution | space/time | average radius error |
|---|---|---|
| $40 \times 40$ | 4 | 0.00666 |
| $40 \times 40$ | 8 | 0.00521 |
| $40 \times 40$ | 16 | 0.00537 |
| $80 \times 80$ | 4 | 0.00259 |
| $80 \times 80$ | 8 | 0.00278 |
| $80 \times 80$ | 16 | 0.00328 |
| $160 \times 160$ | 4 | 0.00164 |
| $160 \times 160$ | 8 | 0.00156 |
| $160 \times 160$ | 16 | 0.00145 |

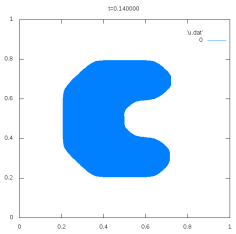
From the comparison result at Table.1, we can see that the accuracy roughly increase almost twice as we increase the resolution twice. And the error is smaller than corresponding space discretization $\Delta x$.

## 4.2 Numerical Simulation

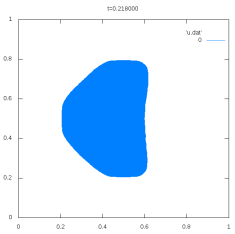
1. This is the simulation of two phases case without area preservation, resolution $160 \times 160$ and time discretization $\tau = 0.005/10$. For initial condition, here we use such kind of area because we want to see inertia effect while it evolve.



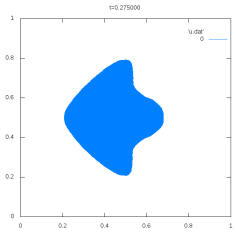
$t = 0.000$



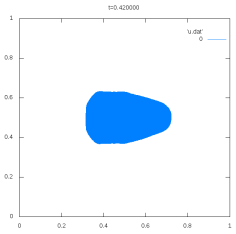
$t = 0.140$



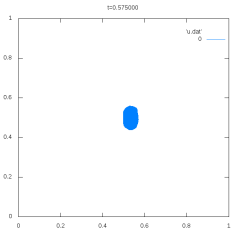
$t = 0.215$



$t = 0.345$



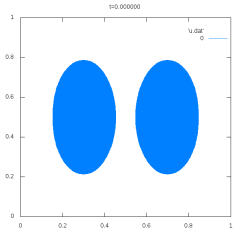
$t = 0.495$



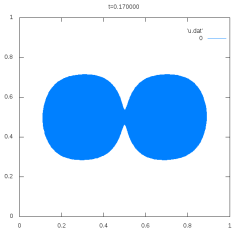
$t = 0.555$

2. This is the simulation of two phases case with area preservation, resolution $160 \times 160$ and time discretization $\tau = 0.005/10$. We use two ellipses area as initial condition. Because these two bubbles have the same phase, as the result it merges at certain time and remain as one

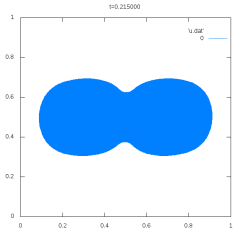
region. And the area is also preserved.



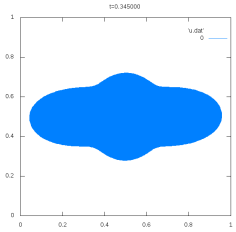
$t = 0.000$



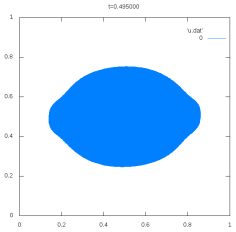
$t = 0.170$



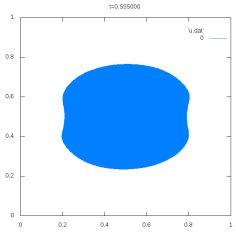
$t = 0.215$



$t = 0.275$

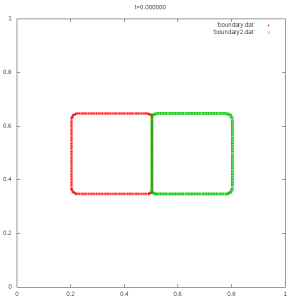


$t = 0.42$

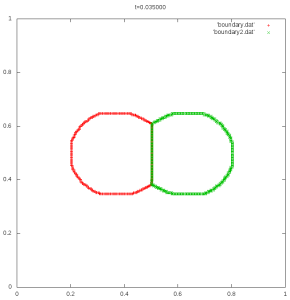


$t = 0.575$

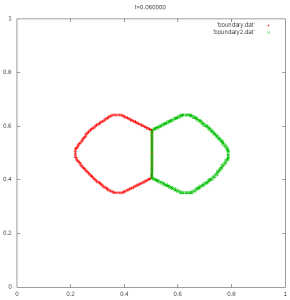
3. This is the simulation of three phases case without area preservation, resolution $160 \times 160$ and time discretization $\tau = 0.005/10$. For initial condition here we use symmetric area of two squares. We can see inertia effect at the edge of the square.



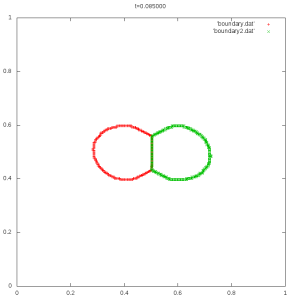
$t = 0$



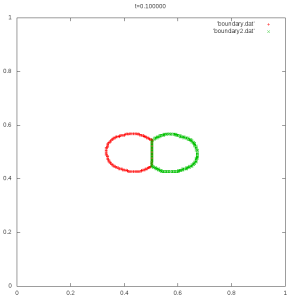
$t = 0.035$



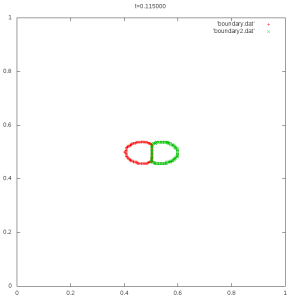
$t = 0.060$



$t = 0.085$



$t = 0.100$



$t = 0.115$

## 5 Conclusion

As conclusion, we implemented the acceleration dependent BMO method for simulating bubble motion. We compared the numerical result for shrinking circle problem with Runge-Kutta fourth order. We also implemented the method for area preservation and for multiple bubbles (three area regions). For future work, this method can be applied for three dimensional bubble motion simulation that involve external force also.

## 6 Acknowledgment

I would like to thank Prof. Seiro Omata and Prof. Norbert Pozar for guiding me during this research in Japan. Financial support from DIKTI is acknowledged. Also big thanks to all DDP students for their support.

## References

[1] B. Merriman, J. Bence, S. Osher (1994). Motion of multiple Junction: a level set approach. *J.Comp.Physics.*, textbf112, 334–363.

[2] E.Ginder, K.Svadlenka (2014). On algorithm for curvature-dependent interfacial acceleration. *Proceeding of Computational Engineering Conference.*, **19**.

[3] K.Svadlenka, E. Ginder, S. Omata (2014). A variational method for multiphase volume-preserving interface motion. *Journal of Computational and Applied Mathematics.*, **257**, 157–179

[4] N. Sofianah, R.Z. Mohammad, K. Svadlenka (2014). On a numerical method for the simulation of contact angle dynamics. *Proceeding of Computational Engineering Conference.*, **19**.

[5] P. LeFloch, K. Smoczyk (2008). The hyperbolic mean curvature flow. *Journal de Mathematiques Pures et Appliques.*, textbf90, 591–614.

[6] R.Z. Mohammad (2014). Numerical analysis of multiphase curvature-driven interface evolution with volume constraint. *Ph.D. Dissertation of Kanazawa University.*

[7] S. Omata (1997). A Numerical method based on the discrete Morse semiflow related to parabolic and hyperbolic equation. *Nonlinear Analysis.*, **30**, 2181-2187.