Simulation of a Rising Oil Droplet using an Interface-Fluid Coupling

Ullul Azmy[a,b]

[a]Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jl. Ganesha 10, Bandung 40132 Indonesia
[b]Graduate School of Natural Science and Technology, Kanazawa University, Kakuma, Kanazawa 920-1192 Japan
Email: ullul.azmy@hotmail.com

Abstract. *We develop a coupled interface-network and fluid model to simulate nonsymmetric triple junction motion with arbitrary surface tension in two dimension. The motion of the interface is governed by the gradient flow of a surface energy. For the numerical method, we adapt a vector valued BMO algorithm. To advance the BMO algorithm, we use a vector-type discrete Morse flow that handles the volume constraint via a penalization. Then, we add buoyancy force as an outer force to the interface model. By using this method, we simulate phenomena of a rising oil droplet in water. Lastly, we present the results of numerical experiments.*

Keywords: triple junction motion, mean curvature flow, vector-valued thresholding

1 Introduction

Triple line dynamics appears when triple line, which refers to interface of three immiscible fluids that intersect at one point, is moving due to some factors such as surface energy, fluid motion and inertial effects. Understanding the triple line dynamics is very useful to realize some kind of important motions. An example of such phenomena is the motion of a rising oil droplet. When the oil droplet rise to the water surface, the interface of three different fluids meet at a single point and adjust into the shape of triple line.

Numerical simulation of flows with moving triple line have been developed by using some methods. One of them is so-called Bence-Merriman-Osher (BMO) algorithm. Bence, Merriman, and Osher [1] introduced the original BMO method, an implicit scheme for realizing interfacial motion by mean curvature flow. Svadlenka et al. [2] reformulated the BMO algorithm in vector-valued formulation for multiphase motion. However, it is restricted to the symmetric case. Shofianah et al. [4] modified the original vector-valued BMO algorithm of [2] by generalizing the reference vectors and the way of diffusing so that it can accomodate motions for any triple of surface tensions.

In this work, we consider three evolving curves meeting at a junction and having arbitrary surface tensions. We adapt method in [4] to achieve the simulation of such a triple junction by generalizing the reference vectors.

2 Basic Model

Triple line for two dimension case is actually a triple point which is called also a triple junction. In our case, when the droplet touches the water surface, the triple junction occurs so that we have three immiscible fluids whose interfaces meet at triple junction. Thus, to get the normal velocity and the condition that has to be satisfied at triple junction, we have to consider the total surface energy of the interfaces and compute its variation.

For a fixed smooth region Ω of $\mathbb{R}^2$, we consider three evolving curves $\gamma_i(s), s \in [p_i, q_i], i = 1, 2, 3$. These curves meet the outer boundary $\partial\Omega$ at a right angle and there will be a point, called triple junction $x_T = \gamma_i(q_i), i = 1, 2, 3$, at which the curves meet. Each curve has different surface tension σ_i. Then the surface energy of all curves is given by

$$L(\gamma) = \sum_{i=1}^{3} \int_{\gamma_i} \sigma_i \mathrm{d}l = \sum_{i=1}^{3} \int_{p_i}^{q_i} |\gamma_i'(s)| \mathrm{d}s$$

From its variation, we can find the gradient flow of surface energy. For a smooth vector field $\varphi(s) = (\varphi_1, \varphi_1)$, we compute

$$\begin{aligned} \frac{d}{d\epsilon} L(\gamma + \epsilon\varphi)|_{\epsilon=0} &= \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i \mathbf{t}_i \cdot \frac{d}{ds}(\varphi(\gamma_i)) \mathrm{d}s \\ &= \sum_{i=1}^{3} \left(- \int_{\gamma_i} (\sigma_i \kappa_i \mathbf{n}_i) \cdot \varphi \mathrm{d}l + \sigma_i \mathbf{t}_i \cdot \varphi(x_T) \right) \end{aligned}$$

where $\mathbf{t}_i$ is the tangential vector, κ_i is curvature, and $\mathbf{n}_i$ is outer normal of γ_i with

$$\mathbf{t}_i = \frac{\gamma_i'}{|\gamma_i'|}, \kappa_i = -\frac{\gamma_{ix}'\gamma_{iy}'' - \gamma_{iy}'\gamma_{ix}''}{|\gamma_i'|^3}, \mathbf{n}_i = \frac{1}{|\gamma_i'|}(\gamma_{iy}' - \gamma_{ix}')$$

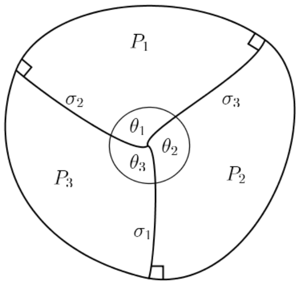


Figure 1: Triple junction

From this result, the motion by gradient flow satisfies

1. The normal velocity of interface
$$v_i = \sigma_i \kappa_i.$$

2. Condition at triple junction
$$\sum_{i=1}^{3} \sigma_i \mathbf{t}_i = 0, \tag{1}$$

The junction condition (1) is the balance of forces which is well-known to be equivalent to the Young's law

$$\frac{\sin\theta_1}{\sigma_1} = \frac{\sin\theta_2}{\sigma_2} = \frac{\sin\theta_3}{\sigma_3},$$

where θ_1, θ_2, and θ_3 are the angles at the junction (see Figure 1). By connecting this formula to the triangle as in [5], we obtain the junction angles by law of cosines:

$$\begin{aligned}
\cos(\pi - \theta_1) &= \frac{\sigma_3^2 + \sigma_2^2 - \sigma_1^2}{2\sigma_2\sigma_3}, \\
\cos(\pi - \theta_2) &= \frac{\sigma_1^2 + \sigma_3^2 - \sigma_2^2}{2\sigma_1\sigma_3}, \\
\theta_1 + \theta_2 + \theta_3 &= 2\pi
\end{aligned}$$

Note that as long as any given triple of surface tensions satisfies the triangle inequality, we can compute the stable angles.

3 Numerical Method

3.1 Vector-valued BMO algorithm

The BMO algorithm is a process for realizing mean curvature motion of interfaces. Originally, it takes advantage from the fact that the characteristic function of a region enclosed by an interface is evolved for a small time by the heat equation according to its mean curvature. Then, a step called *truncation* is implemeneted to obtain the new interface (given by the 1/2-level set of the diffused function). The convergence of this algorithm goes to motion by mean curvature as the short-time $\mathit{\Delta t}$ goes to zero (see [3]).

Svadlenka et al. in [2] modified the BMO algorithm with different approach so that it can treat any number of phases in any dimension and can be extended to more general motions such motion with transport. By reformulating this algorithm into vector-type setting, then:

1. Define reference vectors $\mathbf{p}_i$, each corresponding to a phase P_i for $i = 1, 2, 3$.

2. Given regions P_i, $i = 1, 2, 3$, set $\mathbf{u}_0(x) = \mathbf{p}_i$ for $x \in P_i$.

3. Solve the vector-valued heat equation with initial condition $\mathbf{u}_0(x)$

$$\begin{aligned}
\mathbf{u}_t(x,t) &= \Delta\mathbf{u}(x,t) && \text{for } (t,x) \in (0, \Delta t] \times \Omega \\
\frac{\partial \mathbf{u}}{\partial \mathbf{n}} &= 0 && \text{on } (0, \Delta t] \times \partial\Omega \\
\mathbf{u}(x,0) &= \mathbf{u}_0(x) && \text{in } \Omega
\end{aligned} \tag{2}$$

4. Update $\mathbf{u}_0$ by identifying the reference vector which is closest to the solution $\mathbf{u}(x, \Delta t)$ (see Figure 2 for illustration)

$$\mathbf{u}_0 = \mathbf{p}_j, \text{ where } \mathbf{p}_j \cdot \mathbf{u}(x, \Delta t) = \max_{i=0,1,\dots,k} \mathbf{p}_i \cdot \mathbf{u}(x, \Delta t)$$

   The redistribution of reference vectors determines the configuration of each phase after time Δt.

5. Repeat from step three for the next time step until desired time.

However, this method is only related to symmetric junctions. For arbitrary junction angles, the reference vectors have to be generalized. The main ideas of the generalization were already explained in [4] and will be outlined in the following part.

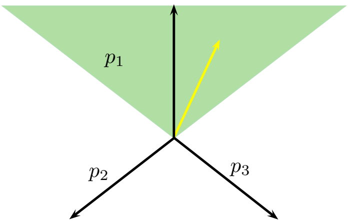


Figure 2: Reidentification of reference vector which is closest to the solution (yellow line)

3.2 Junction stability

Based on the stable configuration for triple junction that was explained in [4], we get

$$\theta_1 \mathbf{p}_1 + \theta_2 \mathbf{p}_2 + \theta_3 \mathbf{p}_3 = \mathbf{0}. \tag{3}$$

Since the reference vectors are determined up to rotation and scaling, we can choose one reference vector arbitrarily, e.g., we set $\mathbf{p}_3 = (1, 0)$. This closes system containing equation (3) and condition of $\mathbf{p}_i, i = 1, 2, 3$, whose lengths must be equal, then we get the reference vectors:

$$\mathbf{p}_1 = \left(1 - \frac{2\pi}{\theta_1\theta_3}(\pi - \theta_2), \pm\frac{2}{\theta_1\theta_3}\sqrt{\pi(\pi-\theta_1)(\pi-\theta_2)(\pi-\theta_3)}\right)$$
$$\mathbf{p}_2 = \left(1 - \frac{2\pi}{\theta_2\theta_3}(\pi - \theta_1), \mp\frac{2}{\theta_2\theta_3}\sqrt{\pi(\pi-\theta_1)(\pi-\theta_2)(\pi-\theta_3)}\right)$$

The possible choices for the sign of the second component follow from the invariance of the reference vectors with respect to flipping.

3.3 Minimizing movements

The heat equation is solved by using vector-type discrete Morse flow (DMF). For a given $N > 0$, we solve (2) by discretizing time $\Delta t = h \times N$ at each step and successively minimizing the following functionals for $n = 1, \cdots, N$ over $H^1(\Omega; R^2)$

$$J_n(\mathbf{u}) = \int_\Omega \left(\frac{|\mathbf{u} - \mathbf{u}_{n-1}|^2}{h} + |\nabla \mathbf{u}|^2\right) dx$$

The minimizers are found by conjugate gradient method.

In the volume constrained case, the minimization formulation of the vector-valued algorithm allows the inclusion of volume constraints via a penalization. In particular, instead of the functional J_n, we minimize

$$F_n(\mathbf{u}) = J_n(\mathbf{u}) + \frac{1}{\epsilon}\sum_{i=1}^{3} |V_i - \mathrm{meas}(P_i^u)|^2,$$

where $\epsilon > 0$ is a small penalty parameter, V_i is the prescribed volume of region P_i.

For the buoyancy effect, we include the transport term to the minimization formulation as in [4], so we minimize

$$\widetilde{F}_n(\mathbf{u}) = F_n(\mathbf{u}) + \int_\Omega \frac{\boldsymbol{f} \cdot \mathbf{u}}{\sqrt{4\pi n h}},$$

where

$$\boldsymbol{f} = \begin{cases} p\dfrac{(\mathbf{p}_i \cdot \mathbf{p}_j - 1)}{|\mathbf{p}_i - \mathbf{p}_j|^2}(\mathbf{p}_i - \mathbf{p}_j), & \text{if} \quad \operatorname{dist}(x, \gamma_{ij}) < \delta_1, \\ & \qquad \operatorname{dist}(x, P_k) > \delta_2 \\ 0, \quad \text{otherwise.} & \end{cases}$$

Here, $\gamma_k, (k \neq i, j)$ is the interface between phase P_i and P_j. δ_1, δ_2 are small positive constants (usually taken as several times the mesh size), p is fluid pressure, and $\mathbf{p}_i, \mathbf{p}_j$ are the BMO reference vectors.

4 Numerical Tests

Now, we present some numerical examples of the method. In this section, all numerical examples are conducted on a $[0,1] \times [0,1]$ domain with time step $\Delta t = 0.002$ and DMF partition $N = 25$. We present the behaviour of the triple junction motion for two cases (with and without axial symmetry) with types of setting as in Table 1.

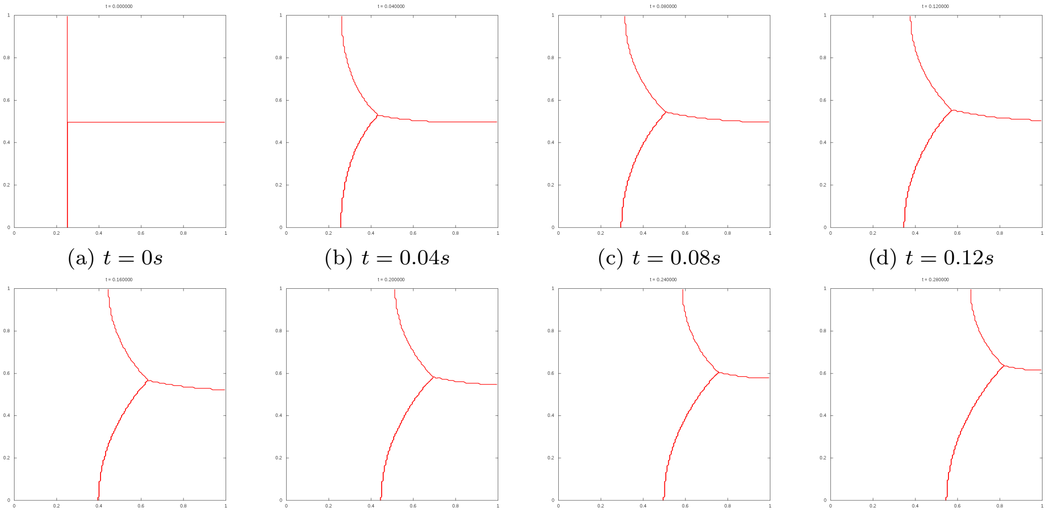
Table 1: Numerical parameters for case 1 and case 2

parameter		case 1	case 2
surface tensions	σ_1	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$
	σ_2	1	1
	σ_3	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$
angles	θ_1	$150°$	$135°$
	θ_2	$90°$	$90°$
	θ_3	$120°$	$135°$
reference vectors	$\boldsymbol{p}_1$	$(-0.8, -0.6)$	$(-0.777, -0.628)$
	$\boldsymbol{p}_2$	$(0, 1)$	$(-0.333, -0.943)$
	$\boldsymbol{p}_3$	$(1, 0)$	$(1, 0)$

For examining the behaviour of the triple junction motion, we start with an initial condition where a T-shaped interface is rotated 90° counterclockwise and the T-junction is at point (0.25,0.5). We take the region that is on the left of the line $x = 0.25$ as P_2, and another top and bottom region as P_1 and P_3 respectively. Here, the domain is triangulated into uniform grid with $\Delta x = 0.00625$. Then, we investigate the evolution of the triple junction for both cases.

1. $150° - 90° - 120°$ angle condition
   For the first case, we plot the evolution of the initial T-junction for each time as in Figure 3.

   For the first 20 time steps, the junction angles rapidly adjusts to approximate the $150° - 90° - 120°$ angle conditions. Note that the interface move to the region with the smallest surface tension over time.



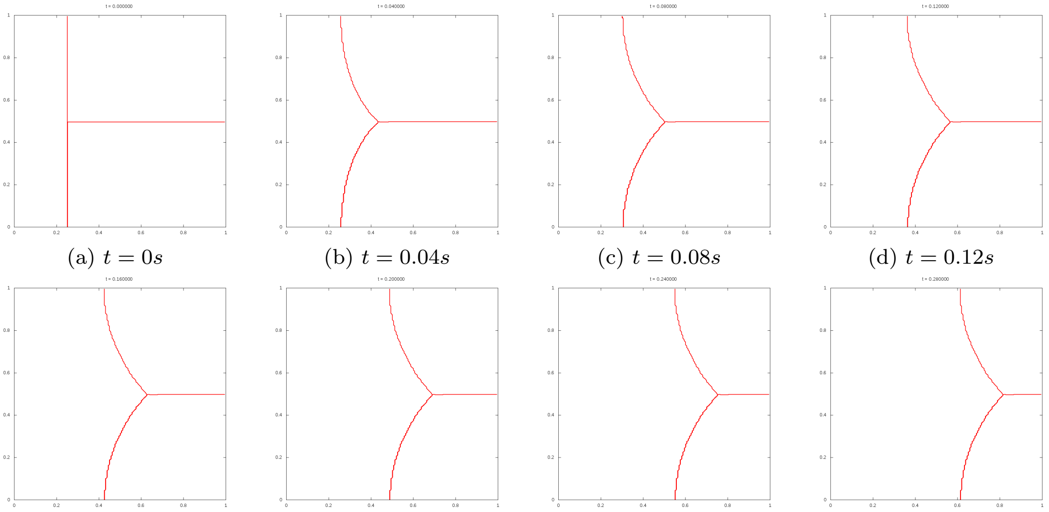
(a) $t = 0s$ (b) $t = 0.04s$ (c) $t = 0.08s$ (d) $t = 0.12s$

(e) $t = 0.16s$ (f) $t = 0.2s$ (g) $t = 0.24s$ (h) $t = 0.28s$

Figure 3: Evolution of triple junction for case 1

2. $135° - 90° - 135°$ angle condition

   Now we look at the behaviour of the junction motion with the parameter as on the second condition. We expect these interfaces will evolve symmetrically with respect to the horizontal line $y = 0.5$ since the surface tensions on the $1 - 2$ and $2 - 3$ interfaces are equal. This is in accordance with the numerical result shown in Figure 4.



(a) $t = 0s$ (b) $t = 0.04s$ (c) $t = 0.08s$ (d) $t = 0.12s$

(e) $t = 0.16s$ (f) $t = 0.2s$ (g) $t = 0.24s$ (h) $t = 0.28s$

Figure 4: Evolution of triple junction for case 2

Note that the triple junction rapidly adjusts to approximate the $135° - 90° - 135°$ angle conditions after the first 20 time steps. Afterwards, the interface gradually is begun to move horizontally to the right.

5 Conclusion

The triple junction motion with T-shaped interface as initial condition was presented. We took two cases of parameters to know the behaviour of this motion. As remarks the test, the method seems working well, so we would like to apply it for three dimensional rising droplet simulation including the buoyancy force also.

Acknowledgment

The author would like to thank to Prof. Seiro Omata and Prof. Norbert Pozar for the guidance during this research. The work of the author is financially supported by the Japanese Government (Monbukagakusho: MEXT) Scholarship Grant.

References

[1] B. Merriman, J. K. Bence, S. J. Osher (1993). Motion of multiple junctions: a level set approach. *Journal of Computational Physics*, **112**, 334-363.

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

[3] L. Evans (1993). Convergence of an algorithm for mean curvature motion. *Indiana Univ. Math. Journals*, **42**, 533-557.

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

[5] X. Chen, J. Guo (2007). Self-similar solution of a 2-D multi-phase curvature flow. *J. Comp. Physics*, D 229, 22-34.