Simulation of Triple Line Dynamics by Interface-Fluid Coupling

Simulation of Triple Line Dynamics by Interface-Fluid Coupling

NUR SHOFIANAH

September 30, 2014

This dissertation is dedicated to my beloved parents,
Ibu Masyhudah and Bapak M. Muchibbin,
for their amazing love, pure sincerity,
endless support and encouragement,
and for everything that I can not describe it in words.

Acknowledgements

In the name of Allah, the Most Merciful, the Most Compassionate. All praise be to Allah, the Lord of the worlds. I would not be able to finish my study without His amazing help. Prayers and peace be upon Mohammad, His servant and messenger.

In this opprtunity, I would like to thank my supervisor, professor Karel Svadlenka for endless support and valuable advice in both academic matter and practical life during my study. I can learn many things in my new research field, and also I can learn from him, how to be a good supervisor for my students in the future.

I would like to thank the Committe, professor Karel Svadlenka, professor Seiro Omata, professor Masato Kimura, professor Hiroshi Ohtsuka, and professor Katsuyoshi Ohara, for valuable suggestions in my dissertation.

I would like to thank the Directorate General of Higher Education (DIKTI), Ministry of National Education, Indonesia for the generous scholarship that made my study in Kanazawa University possible. And also for University of Brawijaya, especially Department of Mathematics, Faculty of Mathematics and Natural Sciences, for the approval and support for my study.

I would like to thank all of my Teachers for their great effort and sincerity since I was child and did not know everything until I can be myself now. May Allah give more reward for all of you.

My thanks also to my labmate, Rhudaina Z. Mohammad who help me checking the grammar and many things. She is the creator of the BMO program and numerical example of interface in Section 4.1. Also thanks to all my friends that I can not mention one by one, for all the supports.

Finally, I would like to thank my beloved parents and my beloved family. Your endless support and doa gave big influence for me. May Allah give us rahmah until together in Jannah.

Abstract

The dissertation develops a coupled interface-network and fluid model which can serve as a first step to simulate triple line dynamics. The main ingredients of this coupled model are the interface model with nonsymmetric junctions and the fluid model in moving domain. We build an interface model based on the gradient flow of surface energy and develop a method for its numerical solution by generalizing the reference vectors and diffusion system in vector-valued BMO algorithm. Moreover, we improve the scheme by using vector-valued projection triangle and use a vector type DMF to handle volume constraint via penalization. For the fluid part, we implement a numerical method adopting DSD/SST-SUPS, a stabilized space-time finite element method in moving domain. We also apply the appropriate boundary condition which is related to a moving triple line. We couple these two models weakly via pressure acting from the fluids on the interfaces and by the fact that the interfaces determine the domain for fluid motion. In the end, we present results of numerical experiments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this dissertation, we are interested in studying the dynamics of triple line. The term triple line refers to the intersection point between the interface of two immiscible fluids and a solid. At equilibrium state, based on the Young's law, the wetting properties of a fluid in contact with a solid are defined by the static contact angle with nonmoving triple line. Triple line dynamics arises when the triple line is moving due to the influence of some factors such as interface (surface) energy, fluid motion and inertial effects. In order to get the correct description of triple line dynamics, one needs to consider these factors in the system. In two dimension problem, triple line is also well-known with triple point, contact line, contact point, triple junction or junction. In the sequel, we will use the term triple line and triple junction term interchangeably.

Triple line has a number of impacts on material behaviors which is of great interest in material science [1]. Besides that, understanding the triple line dynamics is very useful to realize some kinds of important motions such as the motion of small droplets or bubbles which has important application in nanotechnology, the two-phase flow in microfluidic devices, inkjet printing, etc. An example of such a phenomenon is blowing of air from a small hole in the bottom of a container filled with water. Depending on the surface tension of the container's material, the air bubble evolution and its departure size vary significantly due to the different contact angle dynamics as can be seen on the experimental results taken from [48] in Figure 1.1. Moreover, there is a need to accurately model and simulate triple line dynamics in multiphase flow, an important phenomena in a wide range of industrial applications [2, 48].

Numerical simulation of flows with moving triple line have been devel-

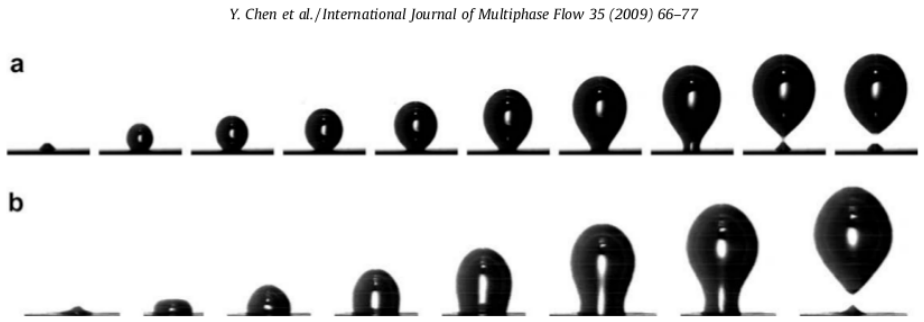


Figure 1.1: Experimental results taken from [48]: Air bubble formation in case blowing of air from a small hole in the bottom of a container filled with water with (a) stainless steel surface (wetting) (b) waxed surface (less wetting)

oped such as using the front tracking method [3], level set method [41, 48, 2, 4], volume-of-fluid method [5], phase field method [6, 23], immersed boundary method [7]. To model triple line motion, Bronsard et al. [23] uses the reaction-diffusion equation. This so-called phase-field method represents the phase boundary as an interfacial layer. In [23], the authors showed that the interfaces in the solution of reaction-diffusion equation move with a velocity proportional to the curvature of the interface and in its sharp interface limit ($\varepsilon \to 0$), where $\varepsilon > 0$ is a small parameter representing the thickness of interface, one obtains the Young's law at the junction. However, this method has difficulties in the numerical solution when $\varepsilon < \Delta x$, where Δx is the spacing grid. The only remedy to overcome these difficulties is to take $\frac{\Delta x}{\varepsilon} \ll 1$, which is numerically impractical [17]. Merriman, Bence and Osher [17] introduced BMO method, an implicit scheme for realizing interfacial motion by mean curvature for symmetric junctions, which alternately diffuses and sharpens characteristic function for each phase region. Zhao et al. [41] developed a numerical algorithm capturing the interfaces based on the level set method of Osher et al. [42]. This method is able to deal with topological singularities and nonsmooth data. Moreover, it can be extended to the multiphase setting by introducing as many level set functions as there are phases and imposing additional constraint on the constrained flow problem so that the level sets do not overlap or create vacuums. However, such a constraint has an unwanted influence on the motion and the volume of phases is not sufficiently controlled by the method. Ruuth [12] and Esedoglu et al. [49] generalized the BMO method to nonsymmetric triple junctions by re-

placing the thresholding step using projection triangle and using an update procedure which is analogue to the standard tresholding step of the BMO algorithm, respectively. However, these methods are in the scalar setting while we need a vector setting for implementing the volume constraint in the multiphase case. For this reason, Svadlenka et al. [11] reformulated the BMO algorithm [17] in vector setting. However, it is restricted to the symmetric triple line. Here, we propose numerical scheme which considers nonsymmetric triple line using BMO algorithm in vector-valued formulation and develop a coupled interface-network and fluid model weakly which requires incorporating the volume preservation for stable coupling.

The main purpose and contribution of the dissertation is to develop a coupled interface-network and fluid model which can serve as a first step to more complicated and precise models, such as those including inertia (hyperbolic mean curvature flow), hysteresis or transport [8]. In this dissertation, we focus mainly on triple junction motion with arbitrary surface tensions and discuss about DSD/SST-SUPS, formulation of fluid model in moving domain, in view of its application in the coupled problem.

We build an interface model based on gradient flow of surface energy and develop numerical model for interface motion with arbitrary surface tensions leading to interface-network with nonsymmetric triple junctions. We achieve the simulation of such a triple junction by generalizing the two main ingredients of the method in [11]: the reference vectors and the diffusion step. Moreover, we improve the scheme by including a modification of the projection step in [12]. The developed method is applicable to constrained motions. Another model that we build is the fluid model based on incompressible Navier-Stokes equations in moving domain. We use the Deforming-Spatial-Domain/Stabilized Space-Time with SUPG/PSPG Stabilization (DSD / SST-SUPS) which is developed for moving boundaries and interfaces [13, 14].

We couple weakly the interface model and fluid model via pressure acting from the fluids on the interfaces and by the fact that the interfaces determine the domain for fluid motion. Such kind of coupling strategy has lower computational cost but encounters instability related to global conditions as volume constraint and to violation of boundary condition on the interfaces [15]. We handle the first issue by incorporating the volume constraint in the interface model via penalization. In the fluid model, the second issue is handled by applying appropriate boundary conditions which is related to a moving triple line (in case two immiscible fluids in contact with solid surface).

To summarize our work, we propose a new numerical scheme to simulate

triple line dynamics:

1. We develop an interface model for arbitrary surface tensions / nonsymmetric junctions, a new contribution in numerical analysis of interface motion. In particular, we construct a numerical scheme for interface model that
   - stably imposes contact angles at junctions;
   - gives correct velocity of interfaces;
   - preserves volume.
2. We construct a fluid model in a moving domain, that is, a domain that changes as time progresses.
3. We propose a coupled interface-network and fluid model which can be extended to include other aspects such as inertia, hysteresis and transport.

In this dissertation, we simplify the situation by considering only the 2D problem.

The dissertation is organized as follows: In Chapter 2, we discuss the basics of interface and fluid models and their coupling strategy. Chapter 3 presents the numerical method for each of the interface and fluid models. Sequential coupling is also be addressed in this chapter. Finally, in Chapter 4, we present several numerical examples.

Chapter 2

Basic Model

In this chapter, we start the discussion by showing the physical setting of the problem. Then we derive equation of triple junction and present the incompressible Navier-Stokes equations as a model for fluid flow. In the 2D case, triple line is actually a triple point, called also a contact point or triple junction. Thus, our model problem is to simulate the motion of three immiscible fluids whose interfaces meet at one triple junction.

2.1 Problem Setting

To clearly illustrate the problem, we take a bubble rising from the bottom of the container filled with fluid as an example. The physical setting of the two phase problem is depicted in the Figure 2.1, where Ω_b and Ω_f denotes gas and fluid, respectively.

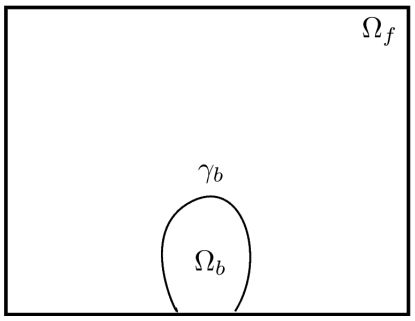


Figure 2.1: Physical setting

2.2 Fluid Model: Navier-Stokes Equation

Let $\Omega_t \in \mathbb{R}^2$ be the fluid domain with boundary Γ_t at time $t \in [0, T]$. The Navier-Stokes equation for incompressible flow in moving domain can be written as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{f}\right) - \nabla\cdot\widehat{\boldsymbol{\sigma}} = \mathbf{0}, \text{ in } \bigcup_{t\in(0,T)} \Omega_t \times \{t\}, \tag{2.1}$$

$$\nabla\cdot\mathbf{u} = \mathbf{0}, \text{ in } \bigcup_{t\in(0,T)} \Omega_t \times \{t\}, \tag{2.2}$$

where $\rho, \mathbf{u}$ and $\mathbf{f}$ are the density, velocity and the external force, respectively. We consider only gravity g as an external force. The stress tensor $\widehat{\boldsymbol{\sigma}}$ can be represented as

$$\widehat{\boldsymbol{\sigma}}(p, \mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}),$$

where p is pressure, $\mathbf{I}$ is the identity tensor, μ is viscosity, $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor,

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}((\nabla\mathbf{u}) + (\nabla\mathbf{u})^T).$$

The initial condition consists of a specified divergence-free velocity field

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0,$$

and hydrostatic pressure

$$p(\mathbf{x}, 0) = P_0 + \rho g h,$$

where P_0 is atmospheric pressure, g is gravitational acceleration, and h is the depth.

The following boundary conditions are prescribed in general,

$$\begin{aligned} \mathbf{u} &= \mathbf{g} \text{ on } (\Gamma_t)_{\mathbf{g}}, \\ \mathbf{n}\cdot\boldsymbol{\sigma} &= \mathbf{h} \text{ on } (\Gamma_t)_{\mathbf{h}}, \end{aligned} \tag{2.3}$$

where $(\Gamma_t)_{\mathbf{g}}$ and $(\Gamma_t)_{\mathbf{h}}$ are complementary subsets of the boundary Γ_t, $\mathbf{n}$ is the unit normal vector, $\mathbf{g}$ and $\mathbf{h}$ are given functions. Specific boundary conditions arise based on the fact that the boundary of the fluid domain Ω_f is moving, i.e., interface γ_b. We consider interface γ_b as a thin membrane that has mass and its motion is described by the Newton's second law,

$$\begin{aligned} m\mathbf{a} &= \mathbf{F} \\ m\frac{d\mathbf{v}}{dt} &= \widehat{\boldsymbol{\sigma}}\cdot\mathbf{n} + \sigma\kappa\mathbf{n} - \mathbf{f}_{fr}, \end{aligned}$$

where the third term of the right-hand side, $\mathbf{f}_{fr}$ is the friction force. The first and second terms are the influence from the fluid and interface, respectively. Writing out the components of stress tensor $\widehat{\boldsymbol{\sigma}}$ and the proportionality of the friction force to the velocity,

$$m\frac{d\mathbf{v}}{dt} = -p\mathbf{I}\cdot\mathbf{n} + 2\mu\boldsymbol{\epsilon}(\mathbf{u})\cdot\mathbf{n} + \sigma\kappa\mathbf{n} - C\mathbf{v}$$

where C is a constant. In this case, we assume that $C \gg m$ and that both the membrane and the fluid are composed of the same material, we obtain

$$C\mathbf{v} = -p\mathbf{I}\cdot\mathbf{n} + \sigma\kappa\mathbf{n}, \tag{2.4}$$

which implies that by the absence of the pressure term, the motion of the interface γ_b according to mean curvature flow.

At this stage, we have the boundary condition on the interface. We need another boundary condition on the moving triple line as free boundary in this problem. Based on what we got in (2.4), the interface evolves according to mean curvature flow. Such kind of motion is derived as the gradient flow of surface energy. Thus, we consider a system that describes the evolution of triple junction under gradient flow of surface energy as explained in the next section.

2.3 Interface Model: Equations of Triple Junction

We write the total surface energy of a curve network and compute its first variation to get the normal velocity for each interface and the condition that has to be satisfied at triple junction. In particular, we consider three evolving curves $\gamma_i(s) = (\gamma_{i1}(s), \gamma_{i2}(s))$, $s \in [p_i, q_i], i = 1, 2, 3$, which lie inside a fixed smooth region Ω of $\mathbb{R}^2$, meet the outer boundary $\partial\Omega$ at a right angle and get together at a triple junction $x_{TL} = \gamma_i(q_i)$. Each curve has different surface tension σ_i.
Then the surface energy of all curves is given by

$$L(\boldsymbol{\gamma}) = \sum_{i=1}^{3}\int_{\gamma_i}\sigma_i\,\mathrm{d}l = \sum_{i=1}^{3}\int_{p_i}^{q_i}\sigma_i\,|\gamma_i'(s)|\mathrm{d}s = \sum_{i=1}^{3}\int_{p_i}^{q_i}\sigma_i\,\sqrt{(\gamma_{i1}')^2 + (\gamma_{i2}')^2}\mathrm{d}s$$

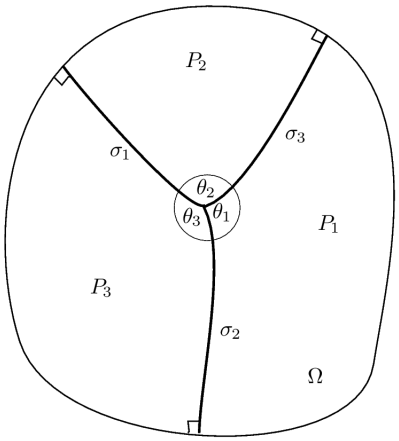


Figure 2.2: Triple Junction

The gradient flow of surface energy can be found from its variation. For a smooth vector field $\boldsymbol{\varphi} = (\varphi_1, \varphi_2)$ vanishing near the boundary $\partial\Omega$,

$$
\begin{aligned}
&\frac{d}{d\epsilon} L(\boldsymbol{\gamma} + \epsilon\boldsymbol{\varphi}(\boldsymbol{\gamma})) \\
&= \frac{d}{d\epsilon} \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i \, |\gamma_i' + \epsilon \frac{d}{ds}(\boldsymbol{\varphi}(\boldsymbol{\gamma}))| \, \mathrm{d}s \\
&= \frac{d}{d\epsilon} \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i \, \sqrt{\left(\gamma_{i1}' + \epsilon \frac{d}{ds}(\varphi_1(\boldsymbol{\gamma}))\right)^2 + \left(\gamma_{i2}' + \epsilon \frac{d}{ds}(\varphi_2(\boldsymbol{\gamma}))\right)^2} \, \mathrm{d}s \\
&= \frac{d}{d\epsilon} \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i \, \sqrt{(\gamma_{i1}' + \epsilon(\varphi_1(\boldsymbol{\gamma}))')^2 + (\gamma_{i2}' + \epsilon(\varphi_2(\boldsymbol{\gamma}))')^2} \, \mathrm{d}s \\
&= \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i A^{-\frac{1}{2}} \left(\gamma_{i1}'(\varphi_1(\boldsymbol{\gamma}))' + \epsilon \left((\varphi_1(\boldsymbol{\gamma}))'\right)^2 + \gamma_{i2}'(\varphi_2(\boldsymbol{\gamma}))' + \epsilon \left((\varphi_2(\boldsymbol{\gamma}))'\right)^2\right)
\end{aligned}
$$

where $A = (\gamma_{i1}' + \epsilon(\varphi_1(\boldsymbol{\gamma}))')^2 + (\gamma_{i2}' + \epsilon(\varphi_2(\boldsymbol{\gamma}))')^2$

$$\begin{aligned}
\frac{d}{d\epsilon} L(\boldsymbol{\gamma} + \epsilon\boldsymbol{\varphi}(\boldsymbol{\gamma}))|_{\epsilon=0} &= \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i \frac{(\gamma_{i1}'(\varphi_1(\boldsymbol{\gamma}))' + \gamma_{i2}'(\varphi_2(\boldsymbol{\gamma}))')}{\sqrt{(\gamma_{i1}')^2 + (\gamma_{i2}')^2}} \, \mathrm{d}s \\
&= \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i \frac{\gamma_i' \cdot (\boldsymbol{\varphi}(\boldsymbol{\gamma}))'}{|\gamma_i'|} \, \mathrm{d}s \\
&= \sum_{i=1}^{3} \int_{p_i}^{q_i} \sigma_i \boldsymbol{t}_i \cdot (\boldsymbol{\varphi}(\boldsymbol{\gamma}))' \, \mathrm{d}s \\
&= \sum_{i=1}^{3} \left(\sigma_i \boldsymbol{t}_i \cdot \boldsymbol{\varphi}(\boldsymbol{\gamma}(s))|_{s=p_i}^{q_i} - \int_{p_i}^{q_i} \sigma_i \boldsymbol{\varphi}(\boldsymbol{\gamma}) \cdot \boldsymbol{t}_i' \, \mathrm{d}s \right) \\
&= \sum_{i=1}^{3} \left(\sigma_i \boldsymbol{t}_i \cdot \boldsymbol{\varphi}(x_{TL}) - \int_{\gamma_i} \sigma_i \boldsymbol{\varphi}(\boldsymbol{\gamma}) \cdot \boldsymbol{t}_i' \frac{1}{|\gamma_i|} \, \mathrm{d}l \right) \\
&= \sum_{i=1}^{3} \left(\sigma_i \boldsymbol{t}_i \cdot \boldsymbol{\varphi}(x_{TL}) - \int_{\gamma_i} \sigma_i \kappa_i \mathbf{n}_i \cdot \boldsymbol{\varphi}(\boldsymbol{\gamma}) \, \mathrm{d}l \right)
\end{aligned}$$

where the tangential vector $\boldsymbol{t}_i$, curvature κ_i and outer normal $\boldsymbol{n}_i$ of curve γ_i are given by

$$\boldsymbol{t}_i = \frac{\gamma_i'}{|\gamma_i'|}, \quad \kappa_i = -\frac{\gamma_{i1}'\gamma_{i2}'' - \gamma_{i2}'\gamma_{i1}''}{|\gamma_i'|^3}, \quad \boldsymbol{n}_i = \frac{1}{|\gamma_i'|}(\gamma_{i2}', -\gamma_{i1}').$$

From this result, the motion by gradient flow satisfies

1. the normal velocity of interface

$$v_i = \sigma_i \kappa_i. \tag{2.5}$$

2. condition at triple junction

$$\sum_{i=1}^{3} \sigma_i \boldsymbol{t}_i = 0. \tag{2.6}$$

The junction condition (2.6) is the balance of forces which is well-known to be equivalent to the Young's law

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

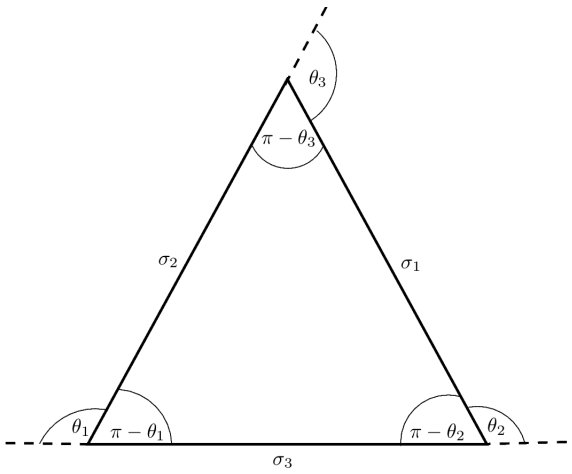


Figure 2.3: Relating surface tensions to junction angles.

where $\theta_1, \theta_2, \theta_3$ are the angles at the junction as be described in Figure 2.2. Note the connection of the above formula to the triangle in Figure 2.3. Using the law of cosines, we obtain the junction angles as in [26]:

$$\begin{aligned} \cos(\pi - \theta_1) &= \frac{\sigma_3^2 + \sigma_2^2 - \sigma_1^2}{2\sigma_2\sigma_3}, \\ \cos(\pi - \theta_2) &= \frac{\sigma_1^2 + \sigma_3^2 - \sigma_2^2}{2\sigma_1\sigma_3}, \end{aligned} \tag{2.7}$$

and $\theta_1 + \theta_2 + \theta_3 = 2\pi$.
Note that we can compute the stable angles for any given triple of surface tensions, as long as the triple satisfies the triangle inequality.

2.4 Coupling Strategy

In this section, we summarize the coupling strategies in our case. We treat the problem as depicted in the Figure 2.1 by considering three-phase case as in Figure 2.4. We add solid as an additional phase which does not change the original setting since the solid interface is straight and thus it will not move. We do so from the beginning in order to be able to suitably treat triple line motion based on the interface model in Section 2.3. Here, $P_i,\ i = 1, 2, 3$ denotes the i-th phase. In particular, P_1, P_2, P_3 represents gas, fluid and solid, respectively.

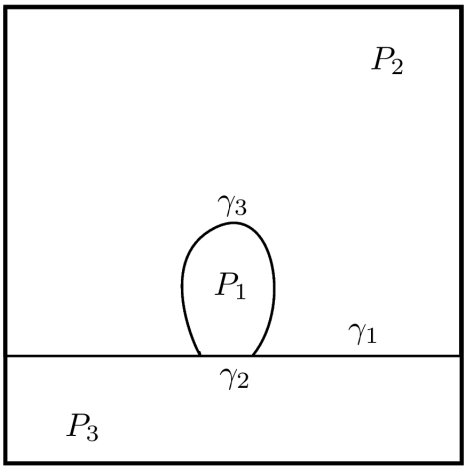


Figure 2.4: Physical setting

The interface and fluid models are coupled at interface γ_3 through pressure p acting from the fluid on the interfaces and by the fact that the interfaces determine the domain for fluid motion. Based on the conditions that we got from the previous sections, the coupling problem can be formulated as follows,

$$
\begin{aligned}
v_3 = \sigma_3\, \kappa_3 + p, \;\; \text{at} \;\; \gamma_3(t), \\
\sum_{i=1}^{3} \sigma_i t_i = 0, \;\; \text{at} \;\; (x_{TL})(t), \\
\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{f}\right) - \nabla\cdot(-p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})) = \mathbf{0}, \;\; \text{in} \bigcup_{t\in(0,T)} P_2(t)\times\{t\}, \\
\nabla\cdot\mathbf{u} = \mathbf{0}, \;\; \text{in} \bigcup_{t\in(0,T)} P_2(t)\times\{t\}.
\end{aligned}
$$

Here, x_{TL} are the triple points and $P_2(t) \subset \mathbb{R}^2$ is the fluid domain with boundary given by $\Gamma_t = \gamma_1(t) \cup \gamma_3(t) \cup \partial\Omega_t$, where $\partial\Omega_t$ is the part of the external boundary which has contact with the fluid.

In this problem, we have to apply appropriate boundary conditions which can handle moving triple line. At gas-fluid interface, we apply free-slip boundary condition, while at solid-fluid interface we apply no-slip boundary condition. The discrepancy of these conditions at triple line causes a nonphysical divergence of the hydrodynamic pressure and of the viscous dissipation at triple line [43, 44]. To avoid this discrepancy, we use some kind

of slip-boundary condition. We smoothly connect these two conditions at triple line by using smoothing length as in the formula below.

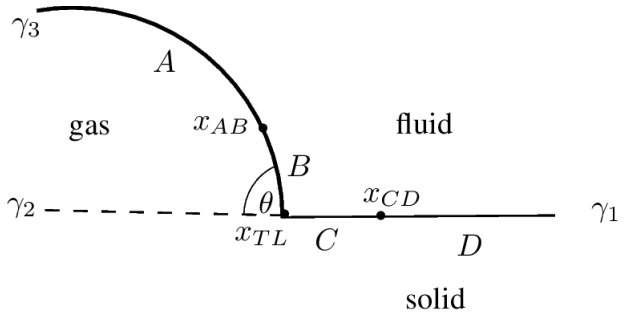


Figure 2.5: Vicinity of triple line

- If $x = x_{TL}$
  $\boldsymbol{u} \cdot \boldsymbol{t}_b = 0,\ \boldsymbol{u} \cdot \boldsymbol{n}_b = v,$
  $\boldsymbol{u} \cdot \boldsymbol{t} = v \sin\theta,\ \boldsymbol{u} \cdot \boldsymbol{n} = v \cos\theta,$
- If $x \in A$
  $\boldsymbol{t}_b \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n}_b = 0,\ \boldsymbol{u} \cdot \boldsymbol{n}_b = v,$
- If $x \in B$
  $\gamma\ \boldsymbol{t}_b \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n}_b = \beta(\boldsymbol{u} \cdot \boldsymbol{t}_b),\ \boldsymbol{u} \cdot \boldsymbol{n}_b = v,$
- If $x \in C$
  $\boldsymbol{u} \cdot \boldsymbol{t} = \alpha\ v \sin\theta,\ \boldsymbol{u} \cdot \boldsymbol{n} = \alpha\ v \cos\theta,$
- If $x \in D$
  $\boldsymbol{u} \cdot \boldsymbol{t} = 0,\ \boldsymbol{u} \cdot \boldsymbol{n} = 0,$

where $\beta = |x_{AB} - x|$, $\gamma = |x_{TL} - x|$, $\alpha = \frac{|x - x_{CD}|}{|x_{TL} - x_{CD}|}$ are the smoothing lengths, $\boldsymbol{u}$ is the velocity of the fluid, v is the normal velocity of the interface, $\boldsymbol{t}_b$, $\boldsymbol{t}$, $\boldsymbol{n}_b$, $\boldsymbol{n}$ are tangential and normal vectors of γ_3 and γ_1, respectively.

Note that in the coupling problem, volume preservation is essential for stable computation. In the numerical method, we incorporate the volume constraint in the interface model.

Chapter 3

Numerical Method

3.1 Numerical Solution of Interface Model: BMO Algorithm with Arbitrary Surface Tensions

The problem of simulating the motion of interfaces with triple junction according to some curvature-dependent speed arises in many applications, e.g., grain growth [9], image processing [10], multiphase flow [11]. Equal surface tensions lead to symmetric triple junction, which is well-known as the symmetric Herring condition (interfaces meeting at 120°). On the other hand, arbitrary surface tensions lead to nonsymmetric triple junctions.

To treat such motions, several methods have been developed. As has been mentioned in the introduction, Bronsard et al. [23] uses the reaction-diffusion equation

$$\mathbf{u}_t = \varepsilon \Delta \mathbf{u} - \frac{1}{\varepsilon} \nabla_{\mathbf{u}} W(\mathbf{u}), \tag{3.1}$$

to model triple junctions. Here, $\varepsilon > 0$ is a small parameter and W is a well potential, a non-negative function which has three minima for the three-phase case. This so-called phase-field method represents the phase boundary as an interfacial layer. In [23], the authors showed that the interfaces in the solution of (3.1) move with a velocity proportional to the curvature of the interface and in its sharp interface limit ($\varepsilon \to 0$), one obtains the Young's law at the junction. However, this method has difficulties in the numerical solution when $\varepsilon < \Delta x$, where Δx is the spacing grid. The only remedy to these difficulties is to take $\frac{\Delta x}{\varepsilon} \ll 1$, which is impractical numerically [17]. Moreover, it is also necessary to appropriately determine parameters of the well-potential function which amounts to a complicated

minimization problem. Garcke et al. [24] performed numerical simulations for a multiphase field model and showed that for suitably chosen parameters (determined based on numerical experiments), numerical results agree with the formal asymptotic expansion of [25] and yield the correct motion.

On the other hand, Zhao et al. [41] developed a numerical algorithm capturing the interfaces based on the level set method of Osher et al. [42]. This method is able to deal with topological singularities and nonsmooth data. Moreover, it can be extended to the multiphase setting by introducing as many level set functions as there are phases and imposing additional constraint on the constrained flow problem so that the level sets do not overlap or create vacuums. However, such a constraint has an unwanted influence on the motion and the volume of phases is not sufficiently controlled by the method.

Merriman, Bence and Osher [17] introduced the BMO method, an implicit scheme for realizing interfacial motion by mean curvature for symmetric junctions, which alternately diffuses and sharpens characteristic function for each phase region. For two-phase case, it has been shown that the method converges to motion by mean curvature [18, 19, 20]. Ruuth [12] generalized the BMO method to nonsymmetric triple junctions by replacing the thresholding step with a new decision, i.e., by using a projection triangle. This generalized method also allows for normal velocity equal to a positive multiple of the curvature of the interface plus the difference in bulk energies for prescribed junction angles. Svadlenka et al. [11] reformulated the BMO algorithm in vector-valued formulation for multiphase motion. This vector-valued formulation is essential for implementing constraints and for dealing with more general motions. However, it is restricted to the symmetric case. Mohammad et al. [21, 22] improved the symmetric multiphase BMO algorithm of [11] by introducing a signed distance vector-valued function.

In this work, we consider three evolving curves meeting at a junction and having arbitrary surface tensions. We achieve the simulation of such a triple junction by generalizing the two main ingredients of the method in [11]: the reference vectors (corresponding to the positions of wells in the phase-field method) and the diffusion step. Moreover, we improve the scheme by including a modification of the projection step in [12]. The developed method is applicable to constrained motions.

3.1.1 Vector-valued BMO

The basis of our method is the vector-valued BMO algorithm [11]:

1. Define reference vectors $\mathbf{p}_i$ of dimension two, each corresponding to a

phase P_i for $i = 1, 2, 3$.

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

3. Repeat

   - Solve the vector-valued heat equation with initial condition $\mathbf{u}_0$:

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

   - Update $\mathbf{u}_0$ by identifying the reference vector which is closest to the solution $\mathbf{u}(\Delta t, x)$:

$$\mathbf{u}_0(x) = \mathbf{p}_j,$$

     where $\mathbf{p}_j \cdot \mathbf{u}(\Delta t, x) = \max_{i=1,2,3} \mathbf{p}_i \cdot \mathbf{u}(\Delta t, x)$.

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

Note that this algorithm deals only with symmetric junctions, the use of symmetric reference vectors and simple heat equation in the diffusion step. However, for arbitrary junction angles, this setting is not sufficient and has to be generalized. We will do so in the following two sections.

3.1.2 Junction Stability

Consider three straight lines meeting at the origin with the given stable angles as in Figure 3.1. The fact that this configuration is stable will yield a condition on the selection of reference vectors for the BMO algorithm.

The triple junction does not move if $\mathbf{u}(t, 0, 0) = \mathbf{0}$ for all $t > 0$, where $\mathbf{u}$ is the solution of the heat equation. This means

$$\frac{1}{4\pi t} \sum_{i=1}^{3} \mathbf{p}_i \int_{\mathbb{R}^2 \cap P_i} \exp(-\frac{|x|^2}{4t})\, \mathrm{d}x = \mathbf{0}.$$

Since by radial symmetry, we can compute the integral of the wedge P_i as

$$\int_{P_i} e^{-|x|^2} dx = \frac{\theta_i}{2},$$

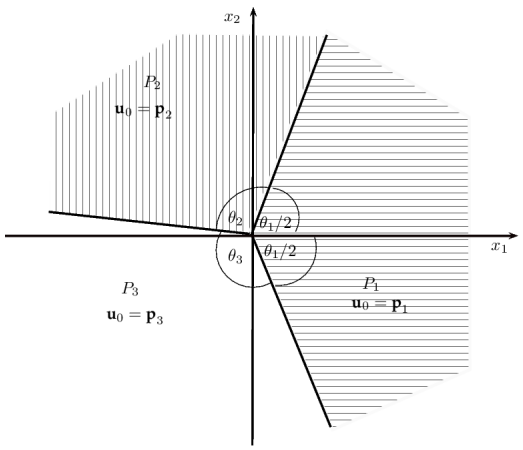


Figure 3.1: Configuration of a stable junction.

we get

$$\theta_1 \mathbf{p}_1 + \theta_2 \mathbf{p}_2 + \theta_3 \mathbf{p}_3 = \mathbf{0}. \tag{3.2}$$

The above relation is a one-dimension higher BMO analogue of Young's law (2.6) in the sense that the reference vectors $\mathbf{p}_i$ are distributed in the whole phase regions P_i and, thus, the equilibrium condition is related to area integrals, which results in weights equal to junction angles.

The vector equation (3.2) and the condition obtained in the next section that the lengths of $\mathbf{p}_i, i = 1, 2, 3$, must be equal, form a systems of equations for the components of $\mathbf{p}_i$:

$$\begin{aligned}
\theta_1 p_1^1 + \theta_2 p_2^1 + \theta_3 p_3^1 &= 0 \\
\theta_1 p_1^2 + \theta_2 p_2^2 + \theta_3 p_3^2 &= 0 \\
(p_1^1)^2 + (p_1^2)^2 &= 1 \\
(p_2^1)^2 + (p_2^2)^2 &= 1 \\
(p_3^1)^2 + (p_3^2)^2 &= 1
\end{aligned}$$

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 the system and its solution can be written as

$$
\begin{aligned}
p_1^1 &= 1 - \frac{2\pi}{\theta_1\theta_3}(\pi - \theta_2),\\
p_1^2 &= \pm\frac{2}{\theta_1\theta_3}\sqrt{\pi(\pi-\theta_1)(\pi-\theta_2)(\pi-\theta_3)},\\
p_2^1 &= 1 - \frac{2\pi}{\theta_2\theta_3}(\pi - \theta_1),\\
p_2^2 &= \mp\frac{2}{\theta_2\theta_3}\sqrt{\pi(\pi-\theta_1)(\pi-\theta_2)(\pi-\theta_3)}.
\end{aligned}
\tag{3.3}
$$

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

3.1.3 Interface Velocity

Here, we study the modification of the original BMO algorithm yielding the correct interface velocities $v_i = \sigma_i\kappa_i$. The idea is to consider, instead of the heat equation, the general diffusion system

$$
\begin{cases}
u_t^1 + du_t^2 = a\Delta u^1 + b\Delta u^2,\\
du_t^1 + eu_t^2 = b\Delta u^1 + c\Delta u^2,\\
\begin{pmatrix} u^1 \\ u^2 \end{pmatrix}(t=0) = \begin{pmatrix} u_0^1 \\ u_0^2 \end{pmatrix},
\end{cases}
\tag{3.4}
$$

and determine its coefficients a, b, c, d, e, so that we obtain the desired interface velocities. We will show that this leads to a system of nonlinear equations for a, b, c, d, e. Moreover, we check that the change of the underlying diffusion system does not influence the arguments of the previous section on junction stability.

Note that we restrict our considerations to symmetric coefficient matrices. This is due to the fact that in order to incorporate phase-volume preservation, we use the variational structure of the system in an essential way (see subsection 3.1.6).

The system (3.4) can be rewritten in the form

$$\tilde{\mathbf{u}}_t = A\Delta\tilde{\mathbf{u}},$$

where

$$\tilde{\mathbf{u}} = \begin{pmatrix} \tilde{u}^1 \\ \tilde{u}^2 \end{pmatrix} = \begin{pmatrix} 1 & d \\ d & e \end{pmatrix}\begin{pmatrix} u^1 \\ u^2 \end{pmatrix},$$

and

$$A = \frac{1}{e-d^2}\begin{pmatrix} ae-bd & b-ad \\ be-cd & c-bd \end{pmatrix}.$$

We assume that A is positive definite and diagonalize it in the following form

$$A = K\Lambda K^{-1} \quad \text{with} \quad \Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

The eigenvalues of A are given by

$$\lambda_{1,2} = \frac{ae + c - 2bd \pm r}{2(e-d^2)},$$

where $r = \sqrt{(ae-c)^2 + 4(ad-b)(cd-be)}$.
Hence, we get the matrix K consisting of eigenvectors as

$$K = \frac{1}{2r}\begin{pmatrix} ae-c+r & 2(ad-b) \\ 2(be-cd) & ae-c+r \end{pmatrix}.$$

The original problem is transformed into

$$\begin{cases} w_t^1 = \lambda_1 \Delta w^1, \\ w_t^2 = \lambda_2 \Delta w^2, \\ \begin{pmatrix} w^1 \\ w^2 \end{pmatrix}(t=0) = \begin{pmatrix} w_0^1 \\ w_0^2 \end{pmatrix} = M \begin{pmatrix} u_0^1 \\ u_0^2 \end{pmatrix}, \end{cases} \tag{3.5}$$

where $\begin{pmatrix} w^1 \\ w^2 \end{pmatrix} = M \begin{pmatrix} u^1 \\ u^2 \end{pmatrix}$, with $M = K^{-1}\begin{pmatrix} 1 & d \\ d & e \end{pmatrix}$.

The solution to the transformed problem (3.5) in the whole $\mathbb{R}^2$ is

$$w^i(t,x) = \frac{1}{4\pi\lambda_i t}\int_{\mathbb{R}^2} w_0^i(\xi) e^{-\frac{|x-\xi|^2}{4\lambda_i t}}\, d\xi, \quad i = 1,2, \tag{3.6}$$

where

$$w_0^i|_{P_j} = ((M\mathbf{u}_0)|_{P_j})^i = (M\mathbf{p}_j)^i, \quad i = 1,2, \quad j = 1,2,3.$$

Now, let us calculate the velocity of the interface for the above diffusion system. We consider a point on the interface $\gamma = \partial P_{ij}$ between phase P_i and P_j. We translate and rotate the coordinate system so that the chosen point lies in the origin and the outer normal at the point agrees with the positive x_2-direction. We define $Q = [-1,1] \times [-1,1]$. Then the normal velocity v

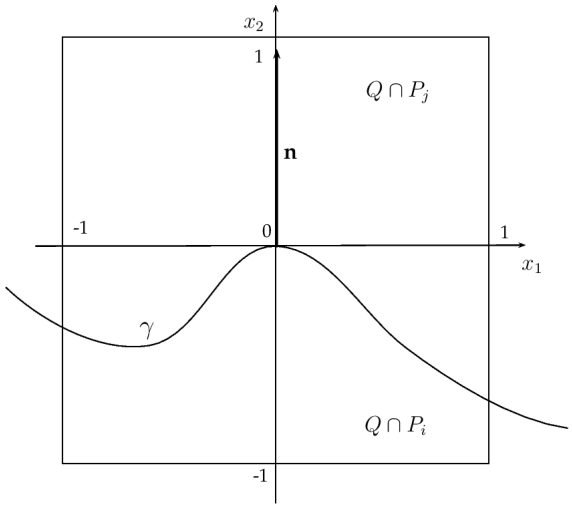


Figure 3.2: Configuration of the interface.

of the interface is found from the relation, expressing the condition on the interface position along the x_2-axis after time t, i.e.,

$$\mathbf{u}(t, 0, vt) \cdot (\mathbf{p}_i - \mathbf{p}_j) = 0. \tag{3.7}$$

We can write the solution of the transformed problem (3.5) as

$$\mathbf{w}(t, 0, vt) = \begin{pmatrix} \frac{1}{4\pi\lambda_1 t} \int_{\mathbb{R}^2} w_0^1(\xi) e^{-\frac{|\xi-(0,vt)|^2}{4\lambda_1 t}} d\xi \\ \frac{1}{4\pi\lambda_2 t} \int_{\mathbb{R}^2} w_0^2(\xi) e^{-\frac{|\xi-(0,vt)|^2}{4\lambda_2 t}} d\xi \end{pmatrix}.$$

By the techniques in [18] and [27], we get

$$\frac{1}{4\pi\lambda_1 t} \int_{Q\cap P_i} e^{-\frac{|\xi-(0,vt)|^2}{4\lambda_1 t}} d\xi = \frac{1}{2} + \frac{\sqrt{t}}{2\sqrt{\pi\lambda_1}}(\lambda_1\kappa - v) + O(t^{\frac{3}{2}}), \tag{3.8}$$

where κ is the curvature of ∂P_{ij} at the origin. Hence, we have for $l = 1, 2$,

and some $C > 0$,

$$\begin{aligned}
& w^l(t,0,vt) \\
= \; & \frac{w_0^l|_{P_i}}{4\pi\lambda_l t}\int_{Q\cap P_i} e^{-\frac{|\xi-(0,vt)|^2}{4\lambda_l t}}d\xi + \frac{w_0^l|_{P_j}}{4\pi\lambda_l t}\int_{Q\cap P_j} e^{-\frac{|\xi-(0,vt)|^2}{4\lambda_l t}}d\xi + O(e^{-\frac{C}{t}}) \\
= \; & w_0^l|_{P_i}\left(\frac{1}{2}+\frac{\sqrt{t}}{2\sqrt{\pi\lambda_l}}(\lambda_l\kappa - v)\right) + w_0^l|_{P_j}\left(\frac{1}{4\pi\lambda_l t}\int_{\mathbb{R}^2} e^{-\frac{|\xi-(0,vt)|^2}{4\lambda_l t}}d\xi\right) \\
& - w_0^l|_{P_j}\left(\frac{1}{2}+\frac{\sqrt{t}}{2\sqrt{\pi\lambda_l}}(\lambda_l\kappa - v)\right) + O(t^{\frac{3}{2}}) \\
= \; & w_0^l|_{P_i}\left(\frac{1}{2}+\frac{\sqrt{t}}{2\sqrt{\pi\lambda_l}}(\lambda_l\kappa - v)\right) + w_0^l|_{P_j}\left(\frac{1}{2}-\frac{\sqrt{t}}{2\sqrt{\pi\lambda_l}}(\lambda_l\kappa - v)\right) + O(t^{\frac{3}{2}}) \\
= \; & \frac{w_0^l|_{P_i}+w_0^l|_{P_j}}{2} + \frac{w_0^l|_{P_i}-w_0^l|_{P_j}}{2}\frac{\sqrt{t}}{\sqrt{\pi\lambda_l}}(\lambda_l\kappa - v) + O(t^{\frac{3}{2}}).
\end{aligned}$$

We obtain from (3.7) the identity

$$M^{-1}\left[M\frac{\mathbf{p}_i+\mathbf{p}_j}{2} + \frac{\sqrt{t}}{\sqrt{\pi}}DM\frac{\mathbf{p}_i-\mathbf{p}_j}{2}\right]\cdot(\mathbf{p}_i-\mathbf{p}_j) = O(t^{\frac{3}{2}}),$$

where

$$D = \begin{pmatrix} \frac{1}{\sqrt{\lambda_1}}(\lambda_1\kappa - v) & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}}(\lambda_2\kappa - v) \end{pmatrix}.$$

Notice that if the first dot product on the left-hand side does not vanish, then the order in time of the equation does not match. This leads to the condition $(\mathbf{p}_i+\mathbf{p}_j)\cdot(\mathbf{p}_i-\mathbf{p}_j) = 0$, meaning that the lengths of reference vectors have to be equal (see subsection 3.1.2). Then we obtain

$$(\mathbf{p}_i-\mathbf{p}_j)^T M^{-1}DM(\mathbf{p}_i-\mathbf{p}_j) = O(t),$$

which is, up to order $O(t)$, equivalent to

$$\mu_1\left(\sqrt{\lambda_1}\kappa - \frac{1}{\sqrt{\lambda_1}}v\right) + \mu_2\left(\sqrt{\lambda_2}\kappa - \frac{1}{\sqrt{\lambda_2}}v\right) = 0,$$

where

$$\begin{aligned}
\mu_1 = \; & m_{11}m_{22}(p_{ij}^1)^2 - m_{12}m_{21}(p_{ij}^2)^2 + (m_{12}m_{22}-m_{11}m_{21})p_{ij}^1p_{ij}^2, \\
\mu_2 = \; & -m_{12}m_{21}(p_{ij}^1)^2 + m_{11}m_{22}(p_{ij}^2)^2 - (m_{12}m_{22}-m_{11}m_{21})p_{ij}^1p_{ij}^2,
\end{aligned}$$

with $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ and $m_{11}, m_{12}, m_{21}, m_{22}$ denote the entries of the matrix M. Finally, we get the velocity of interface γ_k ($k \neq i, j$) in the form

$$v_k = \frac{\mu_1\sqrt{\lambda_1} + \mu_2\sqrt{\lambda_2}}{\mu_2\sqrt{\lambda_1} + \mu_1\sqrt{\lambda_2}}\sqrt{\lambda_1\lambda_2}\,\kappa_k.$$

Since there are only three types of interfaces, it is expected that we need only three parameters in order to design the velocities. Hence, it is reasonable to try to set $d = 0, e = 1$, so that our diffusion system becomes

$$\mathbf{u}_t = A\Delta\mathbf{u}, \tag{3.9}$$

with $A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$. We could have done so from the beginning of this section but, as we will see below, the resulting nonlinear system for a, b, c is hard to analyze theoretically, which led us to keep the parameters d and e as a last resort in case the three-parameter system cannot be solved. In this case, we get the interface velocities

$$v_k = \frac{\mu_1(a + c + r) + 2\mu_2\sqrt{ac - b^2}}{\mu_2(a + c + r) + 2\mu_1\sqrt{ac - b^2}}\sqrt{ac - b^2}\,\kappa_k, \tag{3.10}$$

where

$$\begin{aligned} r &= \sqrt{(a - c)^2 + 4b^2} \\ \mu_1 &= [(a - c + r)(p_i^1 - p_j^1) + 2b(p_i^2 - p_j^2)]^2, \\ \mu_2 &= [2b(p_i^1 - p_j^1) - (a - c + r)(p_i^2 - p_j^2)]^2. \end{aligned}$$

From (3.10) and (2.5), we have a nonlinear system consisting of three equations for the coefficients a, b and c, which is solved numerically. First, we simplify it by setting

$$\begin{aligned} x &= \frac{\alpha + \sqrt{\beta^2 + 1}}{\sqrt{\alpha^2 - \beta^2 - 1}}, \\ y &= \beta + \sqrt{\beta^2 + 1}, \\ z &= b\sqrt{\alpha^2 - \beta^2 - 1}, \end{aligned}$$

where $\alpha = (a + c)/(2b)$ and $\beta = (a - c)/(2b)$. Then the above system in terms of x, y, z becomes

$$\frac{(p_{ij}^1 y + p_{ij}^2)^2 x + (p_{ij}^1 - p_{ij}^2 y)^2}{(p_{ij}^1 - p_{ij}^2 y)^2 x + (p_{ij}^1 y + p_{ij}^2)^2} z = \sigma_k$$

for $(i, j, k) = (2, 3, 1), (1, 3, 2), (1, 2, 3)$. We take any two equations from this system and solve for x, z analytically, which is possible, since the corresponding equation is quadratic. Due to this fact, we get two different solutions for x, z. For each such pair of x, z, we use a numerical method (such as Newton's method) to find the root y of the remaining equation. There may exist several roots for y. For each such combination of x, y, z we recover α, β and b by

$$\alpha = \frac{(x^2+1)(y^2+1)}{2(x^2-1)y},\ \beta = \frac{y^2-1}{2y},\ b = \pm\frac{x(y^2+1)}{(x^2-1)y}.$$

The coefficients a and c are then obtained easily. The number of cases double because of two possible signs for b. All triples (a, b, c) obtained in the described way are then checked if they solve the system and the appropriate triple is selected.

Since the nonlinear system is very complicated, we were not able to prove the unique existence of solution using analytical methods, such as fixed point theory. However, in using the above method, we can always find a solution, except when $b = 0$ which occurs when two of the surface tensions are equal. In that case, the system simplifies in such a way that it can be solved fully analytically.

Table 3.1: Coefficients of Diffusion System in some Cases

$(\sigma_1, \sigma_2, \sigma_3)$	(a, b, c)
(1,1,1)	(1,0,1)
(1,1.5,1)	(1.43773,0.19887,0.86481)
(1.,1.8,1.)	(1.76785,0.18910,0.68793)
(1.5,0.75,1.)	(1.43308 -0.25468 0.67283)
(1.5,1.,1.)	(1.43773,-0.19887,0.86481)
(1.5,1.25,1.)	(1.50944,-0.10082,0.96859)
(1.5,2.,1.)	(2.02618,0.12516,0.89890)
(1.8,1.,1.)	(1.76785,-0.18909,0.68793)
(2.,1.5,1.)	(2.02618,-0.12516,0.89890)
(2.,2.,1.)	(2.2408,0,1)

Remark. For the initial condition in Figure 3.1, at the triple junction we

have

$$w^i(t,0) = \frac{1}{4\pi\lambda_i t}\left(\int_{P_1} + \int_{P_2} + \int_{P_3}\right) w_0^i(\xi) e^{-\frac{|\xi|^2}{4\lambda_i t}} d\xi$$
$$= \frac{1}{\pi}\left(\frac{\theta_1}{2} w_0^i|_{P_1} + \frac{\theta_2}{2} w_0^i|_{P_2} + \frac{\theta_3}{2} w_0^i|_{P_3}\right).$$

Therefore,

$$\mathbf{w}(t,0) = \frac{1}{2\pi} M\left(\theta_1 \mathbf{p}_1 + \theta_2 \mathbf{p}_2 + \theta_3 \mathbf{p}_3\right).$$

For junction stability, we require $\mathbf{u}(t,0) = M^{-1}\mathbf{w}(t,0) = \mathbf{0}$. This condition is equivalent to

$$\theta_1 \mathbf{p}_1 + \theta_2 \mathbf{p}_2 + \theta_3 \mathbf{p}_3 = \mathbf{0},$$

which is in agreement with (3.2).

This result shows that no matter how we change the diffusion equation, the stability condition (3.2) will not be affected. Hence, the selection of reference vectors can be done independently of the diffusion equation.

3.1.4 Correction by Projection Triangle

In the previous sections, we have derived an extension of the vector-valued BMO algorithm to include the case of nonsymmetric junctions and general interface velocities. We have shown that, for a suitably selected diffusion system, if an interface point is sufficiently far from the junction, the interface velocity at that point will satisfy the desired formula $v_i = \sigma_i \kappa_i$, and that, for a suitably selected reference vectors, the junction will be stationary for the initial configuration of three straight lines meeting at the junction with stable contact angles.

However, the above analysis does not address the close vicinity of the triple junction. In fact, by formal calculations it can be made clear (see Appendix 3.1.8) that the correct interface velocity is obtained only with an exponentially decreasing error with respect to the distance of the considered interface point to the junction. Indeed, numerical tests proved that the stable configuration consisting of three straight lines slightly curves in the neighborhood of the junction. This fact produces errors in the angles of the moving junction.

Therefore, we include a correction step based on the notion of a projection triangle given by Ruuth [12]. The idea is to first investigate how the stable configuration of three straight lines deforms, and use this information

to project the phase regions back into the correct position in each step of the BMO algorithm. For details of the construction of the projection triangle we refer to [12].

The original method in [12] uses the scalar setting, i.e., it diffuses separately characteristic functions for each of the three phase regions of the stable configuration (Figure 3.1) which yields three values of diffused function at every point in the domain. These three values are positive and sum up to one, thus, taken as points in $\mathbb{R}^3$, which lie inside a triangle on a hyperplane of $\mathbb{R}^3$. The straight lines of the stable configuration are mapped on the triangle as dividing curves. In order to relate our vector-valued formulation to the construction in [12], we introduce the functions for $(i, j, k) \in \{(1, 2, 3), (2, 3, 1), (3, 2, 1)\}$,

$$f_i(t, x) = \frac{\mathbf{u}(t, x) \cdot (\mathbf{p}_j + \mathbf{p}_k) - 1 - \mathbf{p}_j \cdot \mathbf{p}_k}{(\mathbf{p}_i - \mathbf{p}_j) \cdot (\mathbf{p}_j + \mathbf{p}_k)}, \tag{3.11}$$

Note that (3.11) is a generalization of the function $w_i(t, x)$ in [11], Section 2.2.2, for general reference vectors (3.3). Since $\mathbf{u}(0, x) = \mathbf{p}_i$ for $x \in P_i$, one can easily check that

$$f_i(0, x) = \chi_i(x), \quad i = 1, 2, 3,$$

where χ_i is the characteristic function of phase region P_i.

However, the advantage of our vector-valued approach is that we do not have to perform separate diffusions and subsequently identity the hyperplane $x_1 + x_2 + x_3 = 1$. It is sufficient to diffuse the vector-valued initial condition as in Figure 3.1 and record the images of the straight lines. In this way, we obtain a projection triangle that can be used directly for the function $\mathbf{u}$ in the BMO algorithm.

We summarize the construction of the projection triangle in the generalized vector-valued formulation. Given an angle configuration $\theta_1, \theta_2, \theta_3$, we perform the following steps:

1. Define the lines (in polar coordinates)

$$\ell_{12} = \{(r, \frac{1}{2}\theta_1) : r > 0\},$$

$$\ell_{13} = \{(r, -\frac{1}{2}\theta_1) : r > 0\},$$

$$\ell_{23} = \{(r, \frac{1}{2}\theta_1 + \theta_2) : r > 0\},$$

and regions P_1, P_2, P_3, as in Figure 3.1.

2. Set $\mathbf{u}_0(x) = \mathbf{p}_i$ for $x \in P_i$.

3. Apply the diffusion (3.4) to the initial condition $\mathbf{u}_0$ for a time $\tau \leq \Delta t$, where Δt is the BMO time step.

4. Map the values of the solution of step 3 along each line ℓ_{ij} onto the projection triangle to form the dividing lines $\tilde{\ell}_{ij}$:

$$\tilde{\ell}_{ij} = \{\mathbf{u}(\tau, x) : x \in \ell_{ij}\}.$$

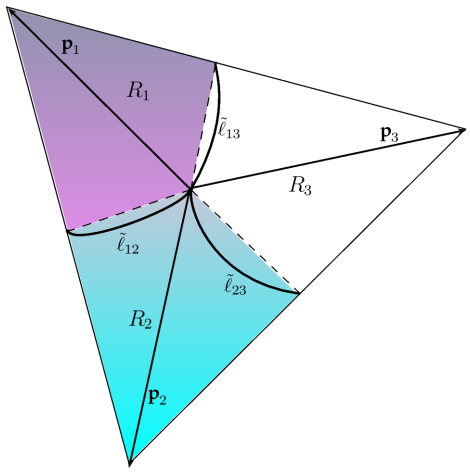


Figure 3.3: Projection triangle $\mathcal{T}$ in vector-valued setting.

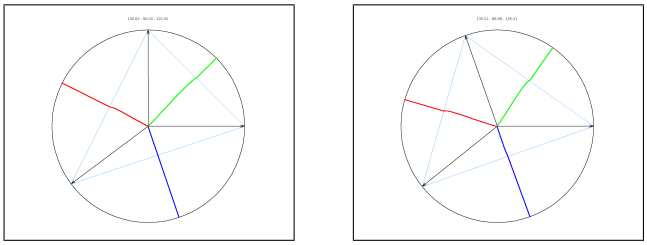
Notice that the values of the diffused function $\mathbf{u}$ fall inside the triangle $\mathcal{T}$ formed by the vectors $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ (Figure). Moreover, because of (3.2), the dividing lines $\tilde{\ell}_{12}, \tilde{\ell}_{13}, \tilde{\ell}_{23}$ will always meet at the origin which corresponds to the circumcenter of $\mathcal{T}$, and they approach the midpoints of edges of $\mathcal{T}$ on their other ends. This is in contrast to the projection triangles in [12], where the position of the junction shifts and the shape of dividing lines is distorted, especially for junction angles strongly deviating from the Herring symmetric case. This fact helps significantly to make the numerical computations stable, especially for the volume-preserving problem.

Figure 3.4 shows the resulting projection triangles for the $150° - 90° - 120°$ junction and the $135° - 90° - 135°$ junction. Since these two cases will be used in numerical tests, we list the coresponding parameters in Table 3.2. As it sometimes happens in the numerical computations that the values of

Table 3.2: Numerical parameters for case 1 and case 2

parameters		**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°
coefficients	a	0.881	0.954
	b	0.262	0.127
	c	0.656	0.639
reference vectors	$\mathbf{p}_1$	(-0.8,-0.6)	(-0.777,-0.628)
	$\mathbf{p}_2$	(0,1)	(-0.333,-0.943)
	$\mathbf{p}_3$	(1,0)	(1,0)

the function $\mathbf{u}$ fall out of the projection triangle, we extend the dividing curves by straight lines connecting the junction to the midpoints of the edges.



(a) (b)

Figure 3.4: Projection triangle for (a) $150° - 90° - 120°$ and (b) $135° - 90° - 135°$ triple junctions.

Remark. There is a close relation between the BMO algorithm and a splitting method for the phase-field equation (3.1), which repeats the following steps:

1. Diffuse for a time Δt

$$\mathbf{u}_t = \varepsilon \Delta \mathbf{u},$$

with initial condition $\mathbf{u}(0, x) = \mathbf{u}_0$.

2. Solve
$$\mathbf{u}_t = -\frac{1}{\varepsilon} W'(\mathbf{u}), \tag{3.12}$$
with initial condition $\mathbf{u}(\Delta t, x)$.

One can see that (3.12) corresponds to the thresholding step in BMO in the sense that the solution asymptotically separates into three regions with values equal to the minima of W. The dividing lines between these regions are the separatrices for the well potential W. The well potential can be chosen as in [24] but, as mentioned earlier, it is not easy to calibrate its parameters since they are not given by an explicit formula.

3.1.5 Generalized Vector-valued BMO Algorithm

In this subsection, we summarize the above deliberations into a generalized vector-valued BMO scheme. First we present the basic version of the BMO algorithm and then comment on the method of incorporating the volume constraint. The generalized vector-valued BMO for three-phases motion is as follows:

1. For given surface tensions, calculate junction angles θ_i by (2.7).
2. Define reference vectors $\mathbf{p}_i$ according to formula (3.3).
3. Find the solution a, b, c of (3.10) and set $A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$.
4. Construct projection triangle according to the algorithm in subsubsection 3.1.4.
5. For a given three phase initial configuration P_1, P_2, P_3, set
$$\mathbf{u}_0(x) = \mathbf{p}_i, \quad x \in P_i.$$
6. Repeat until desired time
   - Solve the diffusion system
$$\begin{aligned} \mathbf{u}_t &= A\Delta\mathbf{u} \quad \text{for } (t,x) \in (0,\Delta t] \times \Omega, \\ \frac{\partial \mathbf{u}}{\partial \mathbf{n}} &= 0 \qquad \text{on } (0,\Delta t] \times \Omega, \\ \mathbf{u}(0,x) &= \mathbf{u}_0(x) \qquad \text{in } \Omega. \end{aligned} \tag{3.13}$$

- Threshold according to the projection triangle defined in step 4, i.e.,

$$\mathbf{u}_0(x) = \mathbf{p}_i \quad \text{if } \mathbf{u}(x) \in R_i, \;\; i = 1, 2, 3,$$

where R_i are the regions in Figure .

The modified diffusion system is solved by using vector-type discrete Morse flow (DMF) [28], i.e., at each step we solve (3.13) by discretizing time $\Delta t = h \times N$ and successively minimizing the following functionals for $n = 1, .., N$ over $H^1(\Omega; R^2)$:

$$J_n(\mathbf{u}) = \int_\Omega \left(\frac{a}{2} |\nabla u^1|^2 + b \nabla u^1 \cdot \nabla u^2 + \frac{c}{2} |\nabla u^2|^2 \right) \mathrm{d}x + \int_\Omega \left(\frac{|\mathbf{u} - \mathbf{u}_{n-1}|^2}{2h} \right) \mathrm{d}x. \tag{3.14}$$

We approximate the functional (3.14) by using piecewise linear finite elements. The minimizers are found by steepest descent method.

3.1.6 Generalized Vector-valued BMO Algorithm with Volume Constraint

The minimization formulation of the vector-valued algorithm allows the inclusion of volume constraints via penalization. In particular, instead of the functional J_n in step 6, we minimize

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

Here $\epsilon > 0$ is a small penalty parameter, V_i is the prescribed volume of region P_i and the volumes corresponding to $\mathbf{u}$ are obtained from the sets

$$P_i^{\mathbf{u}} = \{x \in \Omega; \mathbf{u}(x) \in R_i\}.$$

This means that we have to employ the projection triangle each time we evaluate the functional F_n.

3.1.7 Appendix: Computation of General Reference Vectors

In this appendix, we show the computation of general reference vectors in more detail. We have a system of equations for the components of $\mathbf{p}_i$:

$$\theta_1 p_1^1 + \theta_2 p_2^1 + \theta_3 p_3^1 = 0 \tag{3.15}$$
$$\theta_1 p_1^2 + \theta_2 p_2^2 + \theta_3 p_3^2 = 0 \tag{3.16}$$
$$(p_1^1)^2 + (p_1^2)^2 = 1 \tag{3.17}$$
$$(p_2^1)^2 + (p_2^2)^2 = 1 \tag{3.18}$$
$$(p_3^1)^2 + (p_3^2)^2 = 1 \tag{3.19}$$
$$p_3^1 = 1 \text{ and } p_3^2 = 0 \tag{3.20}$$

Substitute (3.20) into (3.15)-(3.19), we obtain four equations,

$$p_1^1 = \frac{-\theta_3 - \theta_2 p_2^1}{\theta_1} \tag{3.21}$$
$$\theta_1 p_1^2 = -\theta_2 p_2^2 \tag{3.22}$$
$$(p_1^2)^2 = 1 - (p_1^1)^2 \tag{3.23}$$
$$(p_2^2)^2 = 1 - (p_2^1)^2, \tag{3.24}$$

for unknowns p_1^1, p_1^2, p_2^1 and p_2^2. Substitute (3.23), (3.24) and (3.21) into (3.22),

$$\begin{aligned}
\theta_1 p_1^2 &= -\theta_2 p_2^2 \\
\theta_1 \left(1 - (p_1^1)^2\right)^{\frac{1}{2}} &= -\theta_2 \left(1 - (p_2^1)^2\right)^{\frac{1}{2}} \\
\theta_1^2 \left(1 - \left(\frac{-\theta_3 - \theta_2 p_2^1}{\theta_1}\right)^2\right) &= -\theta_2^2 (1 - (p_2^1)^2) \\
p_2^1 &= \frac{\theta_1^2 - \theta_2^2 - \theta_3^2}{2\theta_2\theta_3}
\end{aligned} \tag{3.25}$$

Substitute (3.25) into (3.21),

$$
\begin{aligned}
p_1^1 &= \frac{-\theta_3 - \theta_2 p_2^1}{\theta_1} \\
&= \frac{-\theta_3 - \theta_2 \left(\dfrac{\theta_1^2 - \theta_2^2 - \theta_3^2}{2\theta_2\theta_3} \right)}{\theta_1} \\
&= \frac{-\theta_1^2 + \theta_2^2 - \theta_3^2}{2\theta_1\theta_3} \\
&= \frac{\theta_2^2 - \theta_1^2 - \theta_3^2 - 2\theta_1\theta_3 + 2\theta_1\theta_3}{2\theta_1\theta_3} \\
&= \frac{\theta_2^2 - (\theta_1 + \theta_3)^2 + 2\theta_1\theta_3}{2\theta_1\theta_3} \\
&= \frac{\theta_2^2 - (\theta_1 + \theta_3)^2}{2\theta_1\theta_3} + 1 \\
&= \frac{(2\pi - (\theta_1 + \theta_3))^2 - -(\theta_1 + \theta_3)^2}{2\theta_1\theta_3} + 1 \\
&= \frac{2\pi\left(\pi - (\theta_1 + \theta_3)\right)}{\theta_1\theta_3} + 1 \\
&= \frac{2\pi\left(2\pi - (\theta_1 + \theta_3) - \pi\right)}{\theta_1\theta_3} + 1 \\
&= 1 - \frac{2\pi}{\theta_1\theta_3}(\pi - \theta_2). \qquad (3.26)
\end{aligned}
$$

From (3.26) and (3.17), we can conclude that $\theta_2 \leq \pi$ should be taken. Substitute (3.26) into (3.23) to get p_1^2,

$$\begin{aligned}
(p_1^2)^2 &= 1 - (p_1^1)^2 \\
&= 1 - \left(1 - \frac{2\pi}{\theta_1\theta_3}(\pi - \theta_2)\right)^2 \\
&= \frac{4\pi}{\theta_1\theta_3}(\pi - \theta_2)\left(1 - \frac{\pi}{\theta_1\theta_3}(\pi - \theta_2)\right) \\
&= \frac{4\pi}{\theta_1\theta_3}(\pi - \theta_2)\left(\frac{\theta_1\theta_3 - \pi(\pi - \theta_2)}{\theta_1\theta_3}\right) \\
&= \frac{4\pi}{\theta_1^2\theta_3^2}(\pi - \theta_2)\left(\theta_1\theta_3 - \pi(\pi - (2\pi - (\theta_1 + \theta_3)))\right) \\
&= \frac{4\pi}{\theta_1^2\theta_3^2}(\pi - \theta_2)\left(\theta_1\theta_3 - \pi(-\pi + (\theta_1 + \theta_3))\right) \\
&= \frac{4\pi}{\theta_1^2\theta_3^2}(\pi - \theta_2)\left(\theta_1\theta_3 + \pi^2 - \pi(\theta_1 + \theta_3)\right) \\
&= \frac{4\pi}{\theta_1^2\theta_3^2}(\pi - \theta_2)(\pi - \theta_1)(\pi - \theta_3) \\
p_1^2 &= \pm\frac{2}{\theta_1\theta_3}\sqrt{\pi(\pi - \theta_1)(\pi - \theta_2)(\pi - \theta_3)}. \qquad (3.27)
\end{aligned}$$

where we take $\theta_1 \leq \pi$ and $\theta_3 \leq \pi$. In similar way, we obtain,

$$p_2^1 = 1 - \frac{2\pi}{\theta_2\theta_3}(\pi - \theta_1) \qquad (3.28)$$

$$p_2^2 = \mp\frac{2}{\theta_2\theta_3}\sqrt{\pi(\pi - \theta_1)(\pi - \theta_2)(\pi - \theta_3)}. \qquad (3.29)$$

3.1.8 Appendix: Formal Analysis of Nonsymmetric Triple Junction

We look at the formula for the solution of the system (3.4) with initial condition as in Figure 3.1 in more detail. From this, we will see that the interfaces close to the triple junction will not remain straight lines but will curve slightly.

Recalling the solution of the transformed problem (3.6), the solution of the original system is given by

$$\mathbf{u}(t, x) = M^{-1}w = M^{-1}\begin{pmatrix} w^1 \\ w^2 \end{pmatrix}, \qquad (3.30)$$

where w^1, w^2 are obtained from the formula

$$w^i(t,x) = \left(\sum_{j=1}^{3} (M\mathbf{p}_j)^i \int_{P_j} \right) \frac{1}{4\pi\lambda_i t} e^{-\frac{|x-\xi|^2}{4\lambda_i t}} d\xi, \quad i = 1, 2.$$

The main idea is that the above interface configuration should not change after one step of BMO algorithm if the junction has stable contact angles. To see the specific form of the respective terms, set

$$\operatorname{erfc}(s) = 1 - \operatorname{erf}(s) = \frac{2}{\sqrt{\pi}} \int_s^\infty e^{-\sigma^2} d\sigma,$$

and

$$\alpha(\vartheta) = x_1 \cos\vartheta + x_2 \sin\vartheta,$$

$$\beta(\vartheta) = x_1 \sin\vartheta - x_2 \cos\vartheta.$$

then compute the integral

$$\begin{aligned}
\int_{P_1} e^{-\frac{|x-\xi|^2}{4\lambda t}} d\xi &= \int_0^\infty \int_{-\frac{\theta_1}{2}}^{\frac{\theta_1}{2}} e^{-\frac{|(r\cos\vartheta, r\sin\vartheta)-(x_1,x_2)|^2}{4\lambda t}} r d\vartheta dr \\
&= \int_0^\infty \int_{-\frac{\theta_1}{2}}^{\frac{\theta_1}{2}} e^{-\frac{r^2-2r\alpha(\vartheta)+|x|^2}{4\lambda t}} r d\vartheta dr \\
&= \int_0^\infty \int_{-\frac{\theta_1}{2}}^{\frac{\theta_1}{2}} e^{-\frac{(r-\alpha(\vartheta))^2+\beta(\vartheta)^2}{4\lambda t}} r d\vartheta dr \\
&= \int_{-\frac{\theta_1}{2}}^{\frac{\theta_1}{2}} e^{-\frac{\beta(\vartheta)^2}{4\lambda t}} \int_{-\frac{\alpha(\vartheta)}{\sqrt{4\lambda t}}}^\infty \left(\sqrt{4\lambda t} z + \alpha(\vartheta)\right) \sqrt{4\lambda t} e^{-z^2} dz d\vartheta \\
&= \sqrt{4\lambda t} \int_{-\frac{\theta_1}{2}}^{\frac{\theta_1}{2}} \alpha(\vartheta) \frac{\pi}{2} e^{-\frac{|x-\xi|^2}{4\lambda t}} \operatorname{erfc}\left(-\frac{\alpha(\vartheta)}{\sqrt{4\lambda t}}\right) + \sqrt{\lambda t} e^{-\frac{\alpha(\vartheta)^2}{\sqrt{4\lambda t}}} d\vartheta \\
&= \sqrt{\pi\lambda t} \int_{-\frac{\theta_1}{2}}^{\frac{\theta_1}{2}} \alpha(\vartheta) e^{-\frac{\beta(\vartheta)^2}{4\lambda t}} \operatorname{erfc}\left(-\frac{\alpha(\vartheta)}{\sqrt{4\lambda t}}\right) d\vartheta \\
&\quad + 2\lambda t \int_{-\frac{\theta_1}{2}}^{\frac{\theta_1}{2}} e^{-\frac{\beta(\vartheta)^2+\alpha(\vartheta)^2}{4\lambda t}} d\vartheta.
\end{aligned}$$

Rearranging the terms, we get

$$\int_{P_1} e^{-\frac{|x-\xi|^2}{4\lambda t}} d\xi = \sqrt{\pi\lambda t} \int_{\nu}^{\bar{\beta}} e^{-\frac{\beta^2}{4\lambda t}} \quad \operatorname{erfc}\left(\pm\frac{\sqrt{|x|^2-\beta^2}}{\sqrt{4\lambda t}}\right) du + 2\lambda t\theta_1 e^{-\frac{|x|^2}{4\lambda t}}$$

$$= 2\lambda t\sqrt{\pi} \int_{\frac{\beta(\frac{-\theta_1}{2})}{\sqrt{4\lambda t}}}^{\frac{\beta(\frac{\theta_1}{2})}{\sqrt{4\lambda t}}} e^{-z^2} \operatorname{erfc}\left(\pm\sqrt{\frac{|x|^2}{4\lambda t}-z^2}\right) dz + 2\lambda t\theta_1 e^{-\frac{|x|^2}{4\lambda t}}.$$

Writing $x = (s\cos\varphi, s\sin\varphi)$, we get

$$\frac{1}{2\lambda t}\int_{P_1} e^{-\frac{|x-\xi|^2}{4\lambda t}} d\xi = \sqrt{\pi} \int_{\frac{s\sin(\frac{-\theta_1}{2}-\varphi)}{\sqrt{4\lambda t}}}^{\frac{s\sin(\frac{\theta_1}{2}-\varphi)}{\sqrt{4\lambda t}}} e^{-z^2} \operatorname{erfc}\left(\pm\sqrt{\frac{s^2}{4\lambda t}-z^2}\right) dz + \theta_1 e^{-\frac{s^2}{4\lambda t}}, \tag{3.31}$$

$$\frac{1}{2\lambda t}\int_{P_2} e^{-\frac{|x-\xi|^2}{4\lambda t}} d\xi \quad = \sqrt{\pi} \int_{\frac{s\sin(\frac{\theta_1}{2}-\varphi)}{\sqrt{4\lambda t}}}^{\frac{s\sin(\frac{\theta_1}{2}+\theta_2-\varphi)}{\sqrt{4\lambda t}}} e^{-z^2} \operatorname{erfc}\left(\pm\sqrt{\frac{s^2}{4\lambda t}-z^2}\right) dz + \theta_3 e^{-\frac{s^2}{4\lambda t}} \tag{3.32}$$

$$\frac{1}{2\lambda t}\int_{P_3} e^{-\frac{|x-\xi|^2}{4\lambda t}} d\xi \quad = \sqrt{\pi} \int_{\frac{s\sin(\theta_3-\varphi)}{\sqrt{4\lambda t}}}^{\frac{s\sin(-\varphi)}{\sqrt{4\lambda t}}} e^{-z^2} \operatorname{erfc}\left(\pm\sqrt{\frac{s^2}{4\lambda t}-z^2}\right) dz + \theta_1 e^{-\frac{s^2}{4\lambda t}} \tag{3.33}$$

From (3.31)-(3.33) we can conclude that:

1. If $\frac{s}{\sqrt{t}}$ is close to zero, (3.31)-(3.33) approach $\theta_1, \theta_2, \theta_3$, respectively since the first term is negligible. Hence, from (3.30) we get the value of the solution
$$\mathbf{u}(t,x) \approx \frac{\theta_1}{2\pi}\mathbf{p}_1 + \frac{\theta_2}{2\pi}\mathbf{p}_2 + \frac{\theta_3}{2\pi}\mathbf{p}_3,$$
which is in agreement with (3.2).

2. If the point x is away from the triple junction and close to some of the interfaces, for instance the interface between phases P_1 and P_2, then the second term is exponentially small. Moreover, $\varphi \approx \frac{\theta_1}{2}$, and so the first integral will be close to π. Similarly, the integral over the region P_2 will be close to π and the integral over P_3 will be close to zero. Therefore, the solution (3.30) at this point x will be
$$\mathbf{u}(t,x) \approx \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2),$$

which means that the interface will remain stationary as a straight line.

3. If the point x is in other position than the two cases mentioned above, the values of the above integrals will depend not only on x and t but also on the values of λ_1, λ_2. As a consequence, the solution $\mathbf{u}$ will depend also on the coefficients a, b, c, and the corresponding interfaces may curve.

3.2 Numerical Solution of Fluid Model: Stabilized Space-Time Finite Element in Moving Domain

In this section, we focus on the fluid model in moving domain. Solving fluid model involving moving domain is also the source of the computational challenges that we have. The spatial domain occupied by the fluid changes in time and the model to be used should be able to handle it. Here we use DSD/SST formulation that was introduced by the Team for Advanced Flow Simulation and Modeling (T⋆AFSM) in 1991 as a general-purpose interface-tracking technique for computation of flow problems with moving boundarieas and interfaces [13, 14, 39, 40].

3.2.1 Space-Time Finite Element Method

Deforming-Spatial Domain/Stabilized Space-Time is a method based on space-time finite element. In space-time finite element method, the discretization is applied not only in space but also in time. Consequently, the spatial deformation is taken into account automatically and the dimension of the problem increases by one. In our case, a 2D problem becomes a 3D problem including the time dimension.

As will be shown in the next subsection, the formulation of DSD/SST is written over a sequence of n_t spacetime slabs Q_n, where Q_n is the slice of the space-time domain between the time levels t_n and t_{n+1}. In order to construct the finite element function spaces, we partition the time interval $[0, T]$ into subintervals $T_n = (t_n, t_{n+1})$, where $n = 0, ..., n_{t-1}$, $t_0 = 0$ and $t_{n+1} = T$. Let Ω_n be the spatial domain at time level t_n with its boundary Γ_n. We denote by P_n, the surface described by the boundary Γ_n as t traverses T_n. Surface P_n can be decomposed into $(P_n)_{\mathbf{g}}$ and $(P_n)_{\mathbf{h}}$ where the Dirichlet and Neumann-type boundary conditions (2.3) are enforced, respectively. We define the

space-time slab Q_n as the domain enclosed by the surfaces Ω_n, Ω_{n+1} and P_n. Each Q_n is decomposed into elements Q_n^e, where $e = 1, 2, .., (n_e)_n$, and the number of space-time elements may different for each space-time slab. In each space-time slab Q_n, we define the finite element trial function spaces $(S_u^h)_n$ for velocity and $(S_p^h)_n$ for pressure and the test function spaces $(V_u^h)_n$ and $(V_p^h)_n = (S_p^h)_n$ as follows

$$
\begin{aligned}
(S_u^h)_n &= \{\mathbf{u}^h \in [H^{1h}(Q_n)]^2, \mathbf{u}^h = \mathbf{g}^h \text{ on } (P_n)_{\mathbf{g}}\},\\
(V_u^h)_n &= \{\mathbf{u}^h \in [H^{1h}(Q_n)]^2, \mathbf{u}^h = \mathbf{0} \text{ on } (P_n)_{\mathbf{g}}\},\\
(S_p^h)_n &= (V_p^h)_n = \{p^h \in [H^{1h}(Q_n)]\}.
\end{aligned}
$$

Here, $H^{1h}(Q_n)$ represent the finite dimensional function space over the domain Q_n,

$$
\begin{aligned}
H^{1h}(Q_n) = \{u^h : Q_n \to \mathbb{R} \mid u^h, \frac{\partial u^h}{\partial x}, \frac{\partial u^h}{\partial y}, \frac{\partial u^h}{\partial t} \in L^2(Q_n),\\
u^h|_{Q_n^e} = \text{polynomial in } x, y, t\}
\end{aligned}
\tag{3.34}
$$

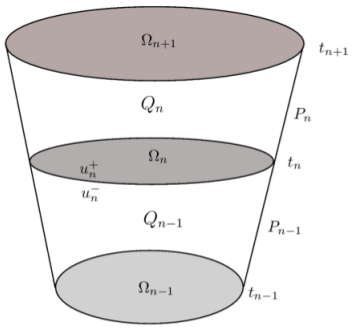


Figure 3.5: Space-time slabs Q_{n-1} and Q_n.

3.2.2 DSD/SST-SUPS Formulation

Deforming-Spatial Domain/Stabilized Space-Time SUPG-PSPG (DSD/SST-SUPS) is DSD/SST formulation based on Streamline-Upwind/Petrov-Galerkin (SUPG) stabilization and Pressure-Stabilizing/Petrov-Galerkin (PSPG) stablization. These stabilization terms assure the numerical stability of the

computations in advection-dominated flows and when using equal order interpolation functions for velocity and pressure, which simplifies the implementation. Moreover, we also use Least-Square on Incompressible Constraint (LSIC) stabilization. We will review these stabilization terms in the Appendix 3.2.8.

DSD/SST-SUPS Formulation for incompressible flow can be written as follows,
Given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in (S_u^h)_n$ and $p^h \in (S_p^h)_n$ such that $\forall \mathbf{w}^h \in (V_u^h)_n$ and $\forall q^h \in (V_p^h)_n$:

$$
\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h)\, \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\, \mathrm{d}Q \\
&- \int_{(P_n)_h} \mathbf{w}^h \mathbf{h}^h \, \mathrm{d}p + \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \frac{1}{\rho} \left[\tau_1 \, \rho \, (\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h) + \tau_2 \nabla q^h \right] \\
&\cdot \left[\rho(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \right] \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_3 \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q = 0,
\end{aligned}
\tag{3.35}
$$

where

$$(\mathbf{u}^h)_n^{\pm} = \lim_{\epsilon \to 0} \mathbf{u}(t_n \pm \epsilon).$$

The first four integrals are the Galerkin formulation of the problem. Here, $\boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h)$ represents component-wise scalar product between strain-rate tensor $\boldsymbol{\varepsilon}(\mathbf{w}^h)$ and stress tensor $\boldsymbol{\sigma}(p^h, \mathbf{u}^h)$. The fifth integral enforces, weakly, the temporal continuity of the velocity field since the basis functions are discontinous from one space-time slab to another. The remaining terms are the stabilization terms. Note that this stabilization leads to a consistent formulation, in the sense that an exact solution still satisfies the stabilized formulation. In this formula, τ_1, τ_2, τ_3 are the stabilization parameters, $\tau_{SUPG}, \tau_{PSPG}, \tau_{LSIC}$, respectively.

After space-time discretization, we obtain a nonlinear system of equa-

tions

$$\mathbf{K}(\mathbf{U})\mathbf{U} = \mathbf{F} \tag{3.36}$$
$$\begin{bmatrix} K_1 & K_2 & K_3 \\ K_4 & K_5 & K_6 \\ K_7 & K_8 & K_9 \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{v} \\ \bar{p} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}.$$

Here, $\bar{u}, \bar{v}, \bar{p}$ are the approximate solutions and matrix $\mathbf{K}$ consists of 9 block matrices. Each block matrix $K_i,\ i = 1, ..., 9$ represents the corresponding terms in (3.35). The size of each block matrix is $(2np) \times (2np)$ where np is the number of nodes at each time level in the space-time slab Q_n. We will write each term of DSD/SST formulation and use the index i to emphasize that it is related to the position in block K_i. We calculate the integrals for element Q_n^e of space-time slab and then assemble for all elements to get the global system matrix.

Matrix $\mathbf{K}$ can be written in more detail as

$$\mathbf{K} = \begin{bmatrix} K_1 & K_2 & K_3 \\ K_4 & K_5 & K_6 \\ K_7 & K_8 & K_9 \end{bmatrix},$$

with

$$\begin{aligned}
K_1 &= TD_1 + B_1 + C_1(\mathbf{U}) + M_1 + S1_1(\mathbf{U}) + S5_1 \\
K_2 &= B_2 + S1_2(\mathbf{U}) + S5_2 \\
K_3 &= G_3^T + S2_3 \\
K_4 &= B_4 + S1_4(\mathbf{U}) + S5_4 \\
K_5 &= TD_5 + B_5 + C_5(\mathbf{U}) + M_5 + S1_5(\mathbf{U}) + S5_5 \\
K_6 &= G_6^T + S2_6 \\
K_7 &= G_7 + S4_7(\mathbf{U}) \\
K_8 &= G_8 + S4_8(\mathbf{U}) \\
K_9 &= S3_9
\end{aligned}$$

The individual terms above have the following meaning:

- Time-dependent term (TD): $\int_{Q_n^e} \rho \mathbf{w}^h \frac{\partial \mathbf{u}^h}{\partial t}\, \mathrm{d}Q$
- Component-wise scalar product: $\int_{Q_n^e} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}\, \mathrm{d}Q_n^e$
  This term consist of Gradient matrix for pressure (G^T) and Diffusion term (B).

- Convection term $(C(\mathbf{U}))$:$\int_{Q_n^e} \rho \mathbf{w}^h (\mathbf{u}^h . \nabla \mathbf{u}^h)\, \mathrm{d}Q$
- Jump term (M):$\int_{\Omega_n^l} \rho (\mathbf{w}^h)_n^+ (\mathbf{u}^h)_n^+ \, \mathrm{d}\Omega$ where Ω_n^l is spatial domain of l-th triangle element at time level t_n.
- Gradient term (G);$\int_{Q_n^e} q^h . \nabla \mathbf{u}^h \, \mathrm{d}Q$
- Stabilization terms $(S(\mathbf{U}))$:

$$\begin{aligned}
&\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \frac{1}{\rho} \left[\tau_1 \, \rho \, (\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h) + \tau_2 \nabla q^h \right] \\
&\cdot \left[\rho (\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \right] \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_3 \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q
\end{aligned}$$

  Stabilization terms consist of five parts: $S1(\mathbf{U}), S2, S3, S4(\mathbf{U}), S5$. as in (3.48) - (3.52). We discuss more about the stabilization parameters τ_1, τ_2, τ_3 in the stabilization section.

The right-hand side is written as

$$F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} F_{f_1} + F_{s1_1} + F_{h_1} + F_{M_1} \\ F_{f_2} + F_{s1_2} + F_{h_2} + F_{M_2} \\ F_{s4_3} \end{bmatrix}.$$

The vectors F_1, F_2, F_3 have size $(2np)$. The detailed of each term will be explained in the next subsections.

3.2.3 Solution of Discrete Problem

In order to solve the nonlinear system (3.36), we use Newton-Raphson method. We compute a correction $\Delta \mathbf{U}$ of a current solution $\mathbf{U}^l$ at each iteration l, which yields a linear system

$$\mathbf{J}(\mathbf{U}^l)\, \Delta \mathbf{U}^l = \mathbf{F} - \mathbf{K}(\mathbf{U}^l)\mathbf{U}^l, \tag{3.37}$$

where $\mathbf{J}$ is the Jacobian matrix. We solve (3.37) by using GMRES with diagonal preconditioner.

As in matrix $\mathbf{K}$, matrix $\mathbf{J}$ also consists of nine block matrices.

$$\mathbf{J} = \begin{bmatrix} J_1 & J_2 & J_3 \\ J_4 & J_5 & J_6 \\ J_7 & J_8 & J_9 \end{bmatrix}$$

with
$J_1 = TD_1 + B_1 + C_1(\mathbf{U}^*) + CC_1(\mathbf{U}^*) + M_1 + S1_1(\mathbf{U}^*) + S5_1 + SS1_1(\mathbf{U}^*)$,
$J_2 = B_2 + S1_2(\mathbf{U}^*) + S5_2 + SS1_2(\mathbf{U}^*) + CC_2(\mathbf{U}^*)$,
$J_3 = G_3^T + S2_3$,
$J_4 = B_4 + S1_4(\mathbf{U}^*) + S5_4 + SS1_4(\mathbf{U}^*) + CC_4(\mathbf{U}^*)$,
$J_5 = TD_5 + B_5 + C_5(\mathbf{U}^*) + CC_5(\mathbf{U}) + M_5 + S1_5(\mathbf{U}^*) + S5_5 + SS1_5(\mathbf{U}^*)$,
$J_6 = G_6^T + S2_6$,
$J_7 = G_7 + S4_7(\mathbf{U}^*)$,
$J_8 = G_8 + S4_8(\mathbf{U}^*)$,
$J_9 = S3_9$,

where $\mathbf{U}^* = \mathbf{U}^l$, the current $\mathbf{U}$ at each iteration, $CC(\mathbf{U}^*)$ and $SS1(\mathbf{U}^*)$ be given by

- $CC(\mathbf{U}^*) :\ \rho \int_{Q_n^k} \rho \mathbf{w}^h (\Delta \mathbf{u}^h \cdot \nabla \mathbf{u}^*)\, \mathrm{d}Q$,

- $SS1(\mathbf{U}^*) : \tau_1 \rho \int_{Q_n^k} (\mathbf{u}^* \cdot \nabla \mathbf{w}^h)(\Delta \mathbf{u} \cdot \nabla \mathbf{u}^*)$,

and similarly for other terms as in matrix $\mathbf{K}$.

The right-hand side $\mathbf{F}$ is slightly different with the previous $\mathbf{F}$,

$$\mathbf{F} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} F_{f_1} + F_{ss1_1} + F_{h_1} + F_{M_1} \\ F_{f_2} + F_{ss1_2} + F_{h_2} + F_{M_2} \\ F_{s4_3} \end{bmatrix}, \tag{3.38}$$

where F_{ss1} is the right-hand side that comes from the derivative of stabilization $S1$, thus depend on $\mathbf{U}^*$ and since $\mathbf{U}^*$ is known, F_{ss1} be a component of the right-hand side. We compute F_{ss1} for each Newton's iteration; while other component of $\mathbf{F}$ are computed once in the beginning of Newton's iteration since they are independent of $\mathbf{U}^*$. The detailed computation of all terms will be explained in the next subsections.

3.2.4 Linear Finite Element in Global and Local Description

The construction of finite element basis functions begins at the level of individual element. Firstly, we construct the element-level interpolation

functions and then put together all elements into a finite element mesh. In this way, we define the global basis functions.

In physical domain, elements in the mesh may have different size and shape. However, every element can be considered as an image of parent element, a simple geometrical shape which is defined in the parametric domain.

Generally, there are two points of view when dealing with linear finite elements: global point of view and element or local point of view [30]. Let us look at it in our case. We will work in two space dimension and use triangular elements. In space-time finite element formulation, it becomes a 3D problem. In this case, every space-time element has six nodes, i.e., three triangular nodes for each time level. In global point of view, the basis functions are considered to be defined everywhere on the domain of the boundary-value problem. Here, we have the quantities,

- Physical domain : $\Omega \times [t_n, t_{n+1}]$.
- Coordinate of nodes of space-time slab Q_n^e :

$$\{\mathbf{x}_{A-1}^n, \mathbf{x}_A^n, \mathbf{x}_{A+1}^n, \mathbf{x}_{A+1}^{n-1}, \mathbf{x}_A^{n+1}, \mathbf{x}_{A+1}^{n+1}\}.$$

- Degrees of freedom: $\{\mathbf{u}_{A-1}^n, \mathbf{u}_A^n, \mathbf{u}_{A+1}^n, \mathbf{u}_{A-1}^{n+1}, \mathbf{u}_A^{n+1}, \mathbf{u}_{A+1}^{n+1}\}$.
- Shape functions: $\{\bar{N}_{A-1}^n, \bar{N}_A^n, \bar{N}_{A+1}^n, \bar{N}_{A-1}^{n+1}, \bar{N}_A^{n+1}, \bar{N}_{A+1}^{n+1}\}$.
- Approximate solution:

$$\begin{aligned}\mathbf{u}^h(\mathbf{x}) = \mathbf{u}_{A-1}^n \bar{N}_{A-1}^n + \mathbf{u}_A^n \bar{N}_A^n + \mathbf{u}_{A+1}^n \bar{N}_{A+1}^n + \mathbf{u}_{A-1}^{n+1} \bar{N}_{A-1}^{n+1} + \mathbf{u}_A^{n+1} \bar{N}_A^{n+1} \\ + \mathbf{u}_{A+1}^{n+1} \bar{N}_{A+1}^{n+1}\end{aligned}$$

These quantities are in terms of global parameters, i.e., global coordinates, global shape functions and global node ordering. The global point of view is useful in establishing the mathematical properties of the finite element method.

In local point of view, the above quantities are in terms of local parameters, i.e., local coordinates, local shape functions and local node ordering, as follows,

- Parametrical domain : $\widehat{\Omega} \times [\theta_1, \theta_2]$.
- Coordinate of nodes of space-time slab $\widehat{Q}_n^e$: $\{\mathbf{r}_1^1, \mathbf{r}_2^1, \mathbf{r}_3^1, \mathbf{r}_1^2, \mathbf{r}_2^2, \mathbf{r}_3^2\}$.
- Degrees of freedom: $\{\mathbf{u}_1^1, \mathbf{u}_2^1, \mathbf{u}_3^1, \mathbf{u}_1^2, \mathbf{u}_2^2, \mathbf{u}_3^2\}$.

- Shape functions: $\{N_1^1, N_2^1, N_3^1, N_1^2, N_2^2, N_3^2\}$.
- Approximate solution:

$$\mathbf{u}^h(\mathbf{r}) = \mathbf{u}_1^1 N_1^1 + \mathbf{u}_2^1 N_2^1 + \mathbf{u}_3^1 N_3^1 + \mathbf{u}_1^2 N_1^2 + \mathbf{u}_2^2 N_2^2 + \mathbf{u}_3^2 N_3^2.$$

In our implementation, we use $\theta_1 = -1,\ \theta_2 = 1, \mathbf{r} = (r, s)^T$, where $r, s \in [0, 1], r + s \leq 1$. For the specific forms of shape functions, see the following subsection. Note that in the local point of view, the nodal numbering is represented by numbering beginning with 1 to show that this is in local description. The local point of view is useful in the computer implementation of finite element mehod. We will use this local point of view in computation of the terms of DSD/SST that has been mentioned before.

The domains of the global and local description are related by the transformation

$$\xi : \Omega \times [t_n, t_{n+1}] \to \widehat{\Omega} \times [\theta_1, \theta_2], \tag{3.39}$$

such that $\xi(\mathbf{x}_{A-1}^n) = \mathbf{r}_1^1,\ \xi(\mathbf{x}_A^n) = \mathbf{r}_2^1,\ \xi(\mathbf{x}_{A+1}^n) = \mathbf{r}_3^1,\ \xi(\mathbf{x}_{A-1}^{n+1}) = \mathbf{r}_1^1,\ \xi(\mathbf{x}_A^{n+1}) = \mathbf{r}_2^1$ and $\xi(\mathbf{x}_{A+1}^{n+1}) = \mathbf{r}_2^2$, where $\mathbf{x} = (x, y)^T, \mathbf{r} = (r, s)^T$. Similarly, we also can define the inverse of ξ,

$$\xi^{-1} : \widehat{\Omega} \times [\theta_1, \theta_2] \to \Omega \times [t_n, t_{n+1}],$$

3.2.5 Shape Functions of Space-Time Element

In order to get the component of system $\mathbf{K}$ and $\mathbf{J}$ in (3.36),(3.37) and right-hand side $\mathbf{F}$ in (3.38), we need to understand the shape functions of space-time element and their derivatives. Here, we assume some special features of space-time slab Q_n, i.e., Q_n has uniform thickness in time direction and all the nodes of the space-time slab are either on its upper or lower surface (Special DSD/SST) [29]. Uniform thickness means that we take the same time-step Δt in the program. We also assume that the mesh on the upper surface of the slab is obtained by a deformation of the mesh on the lower surface which is governed by elasticity equation. The Special DSD/SST formulation offers efficiency in computational cost, i.e., it simplifies both the computation of shape function derivatives at Gaussian quadrature points and the formation of the element-level vectors and matrices. Since for each iteration at every time step, these computations need to be done for each element of the space-time mesh, these improvements will give a significant impact on the overall computational performance in a simulation.

The Special DSD/SST formulation uses shape function having a form of the tensor product of its spatial and temporal shape functions in parametric domain,

$$N_a^\alpha(\mathbf{r},\theta) = N_a(\mathbf{r})T^\alpha(\theta), \tag{3.40}$$

where $a = 1, 2, .., n_{el},\ \alpha = 1, 2$, with n_{el} is the number of nodes in spatial domain. We use triangular elements, i.e., $n_{el} = 3$. In this case, the shape functions $N_a(\mathbf{r})$ read

$$N_1(\mathbf{r}) = r, \qquad N_2(\mathbf{r}) = s, \qquad N_3(\mathbf{r}) = 1 - r - s,$$

and their derivatives,

$$N_{1,r} = 1, \qquad N_{2,r} = 0, \qquad N_{3,r} = -1,$$

$$N_{1,s} = 0, \qquad N_{2,s} = 1, \qquad N_{3,s} = -1,$$

with $r, s \in [0, 1]$ and $r + s <= 1$. The temporal shape functions can be defined by

$$T^1(\theta) = \frac{1}{2}(1 - \theta), \qquad T^2(\theta) = \frac{1}{2}(1 + \theta),$$

and their derivatives,

$$T^1_\theta = -\frac{1}{2}, \qquad T^2_\theta = \frac{1}{2},$$

with $\theta \in [-1, 1]$.

As the mapping (3.39), the spatial coordinate of each element in parametrical domain can be written as

$$\begin{aligned}
\mathbf{x}(\mathbf{r},\theta) &= \sum_{\alpha=1}^{2}\sum_{a=1}^{3} N_a^\alpha(\mathbf{r},\theta)\mathbf{x}_a^\alpha \\
&= \sum_{\alpha=1}^{2}\sum_{a=1}^{3} N_a(\mathbf{r})T^\alpha(\theta)\mathbf{x}_a^\alpha \\
&= \sum_{a=1}^{3} N_a(\mathbf{r})(T^1(\theta)\mathbf{x}_a^1 + T^2(\theta)\mathbf{x}_a^2) \\
&= \sum_{a=1}^{3} N_a(\mathbf{r})\mathbf{x}_a(\theta),
\end{aligned}$$

where $\mathbf{x}_a^\alpha = (x_a^\alpha, y_a^\alpha)$ with $a = 1, 2, 3;\ \alpha = 1, 2$, are the coordinate of space-time element nodes in physical domain. We list formula for each component of $\mathbf{x}(\mathbf{r}, \theta)$ and their derivatives:

$$\begin{aligned}
x(\mathbf{r},\theta) &= \sum_{a=1}^{3} N_a(\mathbf{r}) x_a \\
&= N_1(\mathbf{r})x_1(\theta) + N_2(\mathbf{r})x_2(\theta) + N_3(\mathbf{r})\mathbf{x}_3(\theta) \\
&= N_1(\mathbf{r})(T^1(\theta)x_1^1 + T^2(\theta)x_1^2) + N_2(\mathbf{r})(T^1(\theta)x_2^1 + T^2(\theta)x_2^2) + N_3(\mathbf{r})(T^1(\theta)x_3^1 \\
&\quad + T^2(\theta)x_3^2) \\
&= \frac{r}{2}\left[(1-\theta)x_1^1 + (1+\theta)x_1^2\right] + \frac{s}{2}\left[(1-\theta)x_2^1 + (1+\theta)x_2^2\right] \\
&\quad + \frac{1-r-s}{2}\left[(1-\theta)x_3^1 + (1+\theta)x_3^2\right].
\end{aligned}$$

$$x_r(\theta) = \tfrac{1}{2}\left[(1-\theta)(x_1^1 - x_3^1) + (1+\theta)(x_1^2 - x_3^2)\right].$$

$$x_s(\theta) = \tfrac{1}{2}\left[(1-\theta)(x_2^1 - x_3^1) + (1+\theta)(x_2^2 - x_3^2)\right].$$

$$x_\theta(\mathbf{r}) = \frac{r(x_1^2 - x_1^1) + s(x_2^2 - x_2^1) + (1-r-s)(x_3^2 - x_3^1)}{2}.$$

$$\begin{aligned}
y(\mathbf{r},\theta) &= \sum_{a=1}^{3} N_a(\mathbf{r}) y_a \\
&= N_1(\mathbf{r})y_1(\theta) + N_2(\mathbf{r})y_2(\theta) + N_3(\mathbf{r})y_3(\theta) \\
&= N_1(\mathbf{r})(T^1(\theta)y_1^1 + T^2(\theta)y_1^2) + N_2(\mathbf{r})(T^1(\theta)y_2^1 + T^2(\theta)y_2^2) + N_3(\mathbf{r})(T^1(\theta)y_3^1 \\
&\quad + T^2(\theta)y_3^2) \\
&= \frac{r}{2}\left[(1-\theta)y_1^1 + (1+\theta)y_1^2\right] + \frac{s}{2}\left[(1-\theta)y_2^1 + (1+\theta)y_2^2\right] \\
&\quad + \frac{1-r-s}{2}\left[(1-\theta)y_3^1 + (1+\theta)y_3^2\right].
\end{aligned}$$

$$y_r(\theta) = \tfrac{1}{2}\left[(1-\theta)(y_1^1 - y_3^1) + (1+\theta)(y_1^2 - y_3^2)\right].$$

$$y_s(\theta) = \tfrac{1}{2}\left[(1-\theta)(y_2^1 - y_3^1) + (1+\theta)(y_2^2 - y_3^2)\right].$$

$$y_\theta(\mathbf{r}) = \frac{r(y_1^2 - y_1^1) + s(y_2^2 - y_2^1) + (1-r-s)(y_3^2 - y_3^1)}{2}.$$

The temporal coordinate can be written as,

$$t(\theta) = \sum_{\alpha=1}^{2}\sum_{a=1}^{3} N_a^\alpha(\mathbf{r},\theta)t_a^\alpha.$$

Note that in special DSD/SST formulation, $t_a^\alpha = t^\alpha,\ a = 1,2,3$ since the nodes of a slab are either on its upper ($\alpha = 2$) or lower surface ($\alpha = 1$), we have

$$\begin{aligned} t(\theta) &= \sum_{\alpha=1}^{2}\sum_{a=1}^{3} N_a^\alpha(\mathbf{r},\theta)t^\alpha = \sum_{\alpha=1}^{2}\sum_{a=1}^{3} N_a(\mathbf{r})T^\alpha(\theta)t^\alpha = \sum_{a=1}^{3} N_a(\mathbf{r})\sum_{\alpha=1}^{2} T^\alpha(\theta)t^\alpha \\ &= 1(T^1(\theta)t^1 + T^2(\theta)t^2) = \frac{(1-\theta)t^1 + (1+\theta)t^2}{2} = \frac{t^1 + t^2 + \theta(t^2 - t^1)}{2}, \end{aligned}$$

and for derivatives,

$$\begin{aligned} t_r &= 0, \\ t_s &= 0, \\ t_\theta &= \frac{t^2 - t^1}{2} = \frac{\Delta t}{2}. \end{aligned}$$

The next step is to get the derivative of shape-function of space-time element $\bar{N}_a^\alpha(\mathbf{x},t)$ (in physical domain) with respect to x, y and t since the components of system matrix contain these derivative as well. By using chain rule,

$$\begin{aligned} \frac{\bar{N}_a^\alpha(\mathbf{x},t)}{\partial x} &= \frac{N_a^\alpha(\mathbf{x}(\mathbf{r},\theta),t(\theta))}{\partial r}\frac{\partial r}{\partial x} + \frac{N_a^\alpha(\mathbf{x}(\mathbf{r},\theta),t(\theta))}{\partial s}\frac{\partial s}{\partial x} \\ \frac{\bar{N}_a^\alpha(\mathbf{x},t)}{\partial y} &= \frac{N_a^\alpha(\mathbf{x}(\mathbf{r},\theta),t(\theta))}{\partial r}\frac{\partial r}{\partial y} + \frac{N_a^\alpha(\mathbf{x}(\mathbf{r},\theta),t(\theta))}{\partial s}\frac{\partial s}{\partial y} \\ \frac{\bar{N}_a^\alpha(\mathbf{x},t)}{\partial t} &= \frac{N_a^\alpha(\mathbf{x}(\mathbf{r},\theta),t(\theta))}{\partial \theta}\frac{\partial \theta}{\partial t}, \end{aligned}$$

which can be rewritten in the following matrix form,

$$\begin{bmatrix} \bar{N}_{a,x}^\alpha(\mathbf{x},t) \\ \bar{N}_{a,y}^\alpha(\mathbf{x},t) \\ \bar{N}_{a,t}^\alpha(\mathbf{x},t) \end{bmatrix} = P \begin{bmatrix} N_{a,r}^\alpha(\mathbf{r},\theta) \\ N_{a,s}^\alpha(\mathbf{r},\theta) \\ N_{a,\theta}^\alpha(\mathbf{r},\theta) \end{bmatrix} = P \begin{bmatrix} N_{a,r}T^\alpha(\theta) \\ N_{a,s}T^\alpha(\theta) \\ N_a(\mathbf{r})T_\theta^\alpha \end{bmatrix} \tag{3.41}$$

where

$$P = \begin{bmatrix} r_x & s_x & \theta_x \\ r_y & s_y & \theta_y \\ r_t & s_t & \theta_t \end{bmatrix}$$

Note that we have already expressions for $N_{a,r}, N_{a,s}$ and T_θ. However, we do not have explicit expressions for $r = r(x, y, t), s = s(x, y, t)$, and $\theta = \theta(x, y, t)$, so matrix P cannot be computed directly. It can be obtained from $P = Q^{-1}$, where

$$Q = \begin{bmatrix} x_r & y_r & t_r \\ x_s & y_s & t_s \\ x_\theta & y_\theta & t_\theta \end{bmatrix} = \begin{bmatrix} x_r(\theta) & y_r(\theta) & 0 \\ x_s(\theta) & y_s(\theta) & 0 \\ x_\theta(\mathbf{r}) & y_\theta(\mathbf{r}) & \frac{\Delta t}{2} \end{bmatrix}.$$

We define

$$J^{st}(\theta) = \det Q = \frac{\Delta t}{2} \det \begin{bmatrix} x_r(\theta) & y_r(\theta) \\ x_s(\theta) & y_s(\theta) \end{bmatrix} = \frac{\Delta t}{2} \Upsilon(\theta),$$

the Jacobian of the transformation from physical domain to parametrical domain in space-time element. Here, $\Upsilon(\theta)$ can be written as,

$$\Upsilon(\theta) = \det \begin{bmatrix} x_r(\theta) & y_r(\theta) \\ x_s(\theta) & y_s(\theta) \end{bmatrix}. \tag{3.42}$$

Computing the inverse of matrix Q, we get

$$\begin{aligned} Q^{-1} = P &= \frac{1}{J^{st}(\theta)} \begin{bmatrix} \frac{\Delta t}{2} y_s & -\frac{\Delta t}{2} y_r & 0 \\ -\frac{\Delta t}{2} x_s & \frac{\Delta t}{2} x_r & 0 \\ -x_\theta y_s + y_\theta x_s & x_\theta y_r - y_\theta x_r & \Upsilon(\theta) \end{bmatrix} \\ &= \begin{bmatrix} \frac{y_s(\theta)}{\Upsilon(\theta)} & -\frac{y_r(\theta)}{\Upsilon(\theta)} & 0 \\ -\frac{x_s(\theta)}{\Upsilon(\theta)} & \frac{x_r(\theta)}{\Upsilon(\theta)} & 0 \\ V_r(\mathbf{r}, \theta) & V_s(\mathbf{r}, \theta) & \frac{2}{\Delta t} \end{bmatrix}, \end{aligned}$$

where

$$V_r(\mathbf{r}, \theta) = \frac{-x_\theta(\mathbf{r}) y_s(\theta) + y_\theta(\mathbf{r}) x_s(\theta)}{J^{st}(\theta)}, \qquad V_s(\mathbf{r}, \theta) = \frac{x_\theta(\mathbf{r}) y_r(\theta) - y_\theta(\mathbf{r}) x_r(\theta)}{J^{st}(\theta)}.$$

Then (3.41) can be rewritten as

$$\begin{bmatrix} \bar{N}^{\alpha}_{a,x}(\mathbf{x},t) \\ \bar{N}^{\alpha}_{a,y}(\mathbf{x},t) \\ \bar{N}^{\alpha}_{a,t}(\mathbf{x},t) \end{bmatrix} = \begin{bmatrix} \frac{y_s(\theta)}{\Upsilon(\theta)} & -\frac{y_r(\theta)}{\Upsilon(\theta)} & 0 \\ -\frac{x_s(\theta)}{\Upsilon(\theta)} & \frac{x_r(\theta)}{\Upsilon(\theta)} & 0 \\ V_r(\mathbf{r},\theta) & V_s(\mathbf{r},\theta) & \frac{2}{\Delta t} \end{bmatrix} \begin{bmatrix} N_{a,r}T^{\alpha}(\theta) \\ N_{a,s}T^{\alpha}(\theta) \\ N_a(\mathbf{r})T^{\alpha}_{\theta} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{y_s(\theta)N_{a,r} - y_r(\theta)N_{a,s}}{\Upsilon(\theta)}T^{\alpha}(\theta) \\ \frac{-x_s(\theta)N_{a,r} + x_r(\theta)N_{a,s}}{\Upsilon(\theta)}T^{\alpha}(\theta) \\ (V_r(\mathbf{r},\theta)N_{a,r} + V_s(\mathbf{r},\theta)N_{a,s})T^{\alpha}_{\theta} + \frac{(-1)^{\alpha}}{\Delta t}N_a(\mathbf{r}) \end{bmatrix}. \tag{3.43}$$

3.2.6 Computation of Component System Matrix and Right-hand Side of The Linearized System

Using (3.43), we can compute all components of the finite element systems matrices **K** and **J** in (3.36),(3.37) and right-hand side **F** in (3.38). The components of system matrix and the right-hand side are decomposed from spatial and temporal shape functions due to (3.40). We compute the integral over temporal domain using Gaussian quadrature with Gaussian quadrature points $\tilde{\theta}_i$ and weights W_i (Appendix 3.2.10); while the integral over spatial domain will be calculated analytically for efficiency of computation reason [29]. It is possible to perform integration over spatial domain analytically, since we use linear triangular as spatial element. Here, we use the formula in [30]

$$\int_{\triangle} N_1^{\alpha} N_2^{\beta} N_3^{\gamma} \, d\triangle = \frac{\alpha!\beta!\gamma!}{(\alpha+\beta+\gamma+2)!} 2A$$

where A is the area of triangular $\triangle$ that can be obtained from

$$2A = \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix},$$

with (x_i, y_i) is the coordinate of node i in triangular $\triangle$. In the computation of component system matrix, we change the domain of integration from physical domain into parametrical domain. In this case, A represent the area of triangle in parametrical domain with $A = \frac{1}{2}$.

We will calculate the linear terms first and then nonlinear terms (convection term, stabilization stab1 term, stabilization stab4 term) in increment form. We denote each term with its abbreviated name as we mentioned in the earlier part of this section, e.g., TD for time-dependent term. In $TD_i^e(a,b)$, index i represents the position of block matrix of this term, superindex e is the number of space-time slab element Q_n^e. This symbol also emphasizes that this computation is done for each element separately. a and b are the local numbers of spatial nodes in Q_n^e, $a = 1, 2, 3$; $b = 1, 2, 3$, α and β represent time level at Q_n^e, 1 for time level t_n and 2 for t_{n+1}, $\widehat{Q}_n^e$ is Q_n^e in parametrical domain, $\widehat{Q}_n^{\triangle}$ represents the triangular (spatial element) in $\widehat{Q}_n^e$.

1) Time-dependent term (TD),

$$\int_{Q_n^e} \rho \mathbf{w}^h \frac{\partial \mathbf{u}^h}{\partial t} \, \mathrm{d}Q \tag{3.44}$$

$$\begin{aligned}
&TD_1^e(a,b) \\
&= \rho \int_{Q_n^e} \bar{N}_a^{\alpha}(\mathbf{x},t) \bar{N}_{b,t}^{\beta}(\mathbf{x},t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\
&= \rho \int_{\widehat{Q}_n^e} \bar{N}_a^{\alpha}(\mathbf{x}(\mathbf{r},\theta),t(\theta)) \bar{N}_{b,t}^{\beta}(\mathbf{x}(\mathbf{r},\theta),t(\theta)) J^{st}(\theta) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\theta \\
&= \rho \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} N_a^{\alpha}(\mathbf{r},\tilde{\theta}_i) N_{b,t}^{\beta}(\mathbf{r},\tilde{\theta}_i) J^{st}(\tilde{\theta}_i) W_i \, \mathrm{d}\mathbf{r} \\
&= \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \int_{\widehat{Q}_n^{\triangle}} N_a(\mathbf{r}) T^{\alpha}(\tilde{\theta}_i) N_{b,t}^{\beta}(\mathbf{r},\tilde{\theta}_i) \, \mathrm{d}\mathbf{r} \\
&= \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \left[T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} N_a(\mathbf{r}) (V_r(\mathbf{r},\tilde{\theta}_i) N_{b,r} + V_s(\mathbf{r},\tilde{\theta}_i) N_{b,s}) \mathrm{d}\mathbf{r} \right] \\
&\quad + \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \left[T^{\alpha}(\tilde{\theta}_i) \frac{(-1)^{\beta}}{\Delta t} \int_{\widehat{Q}_n^{\triangle}} N_a(\mathbf{r}) N_b(\mathbf{r}) \mathrm{d}\mathbf{r} \right]
\end{aligned}$$

We obtain time-dependent term,

$$TD_1^e(a,b)\begin{cases} = \rho\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)\frac{2A}{24}\left[A_1+A_2+2J^{st}(\tilde{\theta}_i)\frac{(-1)^{\beta}}{\Delta t}\right], \text{ if } a=b \\ = \rho\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)\frac{2A}{24}\left[A_1+A_2+J^{st}(\tilde{\theta}_i)\frac{(-1)^{\beta}}{\Delta t}\right], \text{ otherwise} \end{cases}$$

$$TD_1^e(a,b)\begin{cases} = \rho\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)\frac{2A}{24}\left[A_1+A_2+\Upsilon(\tilde{\theta}_i)(-1)^{\beta}\right], \text{ if } a=b \\ = \rho\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)\frac{2A}{24}\left[A_1+A_2+\frac{\Upsilon(\tilde{\theta}_i)}{2}(-1)^{\beta}\right], \text{ otherwise} \end{cases}$$

where

$$A_1 = T^{\beta}(\tilde{\theta}_i)\left(\sum N_c(\mathbf{r})(-\tfrac{\Delta x_c}{2}-\tfrac{\Delta x_a}{2})y_s(\tilde{\theta}_i)+\sum N_c(\mathbf{r})(\tfrac{\Delta y_c}{2}+\tfrac{\Delta y_a}{2})x_s(\tilde{\theta}_i)\right)N_{b,r}$$
$$A_2 = T^{\beta}(\tilde{\theta}_i)\left(\sum N_c(\mathbf{r})(\tfrac{\Delta x_c}{2}+\tfrac{\Delta x_a}{2})y_r(\tilde{\theta}_i)-\sum N_c(\mathbf{r})(-\tfrac{\Delta y_c}{2}-\tfrac{\Delta y_a}{2})x_r(\tilde{\theta}_i)\right)N_{b,s}$$

$TD_5^e(a,b) = TD_1^e(a,b)$
2) Component wise scalar product

$$\int_{Q_n^e} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}\, \mathrm{d}Q_n^e \tag{3.45}$$

This term consist of

- Gradient matrix for pressure (G^T)

$$
\begin{aligned}
(G_3^T)^e(a,b) &= -\int_{Q_n^e} \bar{N}_{a,x}^{\alpha}(t)\bar{N}_b^{\beta}(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= -\int_{\widehat{Q}_n^e} \bar{N}_{a,x}^{\alpha}(t(\theta))\bar{N}_b^{\beta}(\mathbf{x}(\mathbf{r},\theta),t(\theta))J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta \\
&= -\int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} N_{a,x}^{\alpha}(\tilde{\theta}_i)N_b^{\beta}(\mathbf{r},\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_i\ d\mathbf{r} \\
&= -\sum_{i=1}^{2} J^{st}(\tilde{\theta}_i)W_i \int_{\widehat{Q}_n^{\triangle}} N_{a,x}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)N_b(\mathbf{r})T^{\beta}(\tilde{\theta}_i)\ \mathrm{d}\mathbf{r} \\
&= -\sum_{i=1}^{2} J^{st}(\tilde{\theta}_i)W_i T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} N_b(\mathbf{r}) \\
&= -\sum_{i=1}^{2} \frac{\Delta t}{2}\Upsilon(\tilde{\theta}_i)W_i T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)\frac{2A}{6} \\
&= -\frac{\Delta t}{12}\sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)
\end{aligned}
$$

and similarly with $(G_5^T)^e(a,b)$ by changing $N_{a,x}$ with $N_{a,y}$

$$
\begin{aligned}
(G_6^T)^e(a,b) &= -\int_{Q_n^e} \bar{N}_{a,y}^{\alpha}(t)\bar{N}_b^{\beta}(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= -\frac{\Delta t}{12}\sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)
\end{aligned}
$$

3) Diffusion term (B):

$$
\begin{aligned}
B_1^e(a,b) &= \mu \int_{Q_n^e} 2\bar{N}_{a,x}^{\alpha}(t)\bar{N}_{b,x}^{\beta}(t) + \bar{N}_{a,y}^{\alpha}(t)\bar{N}_{b,y}^{\beta}(t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= \mu \int_{\widehat{Q}_n^e} 2\bar{N}_{a,x}^{\alpha}(t(\theta))\bar{N}_{b,x}^{\beta}(t(\theta))J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta \\
&\quad + \mu \int_{\widehat{Q}_n^e} \bar{N}_{a,y}^{\alpha}(t(\theta))\bar{N}_{b,y}^{\beta}(t(\theta))J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta
\end{aligned}
$$

$$
\begin{aligned}
B_1^e(a,b) &= \mu \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} 2N_{a,x}^{\alpha}(\tilde{\theta}_i) N_{b,x}^{\beta}(\tilde{\theta}_i) J^{st}(\tilde{\theta}_i) W_i \, \mathrm{d}\mathbf{r} \\
&\quad + \mu \int_{\widehat{Q}_n^{e}} \sum_{i=1}^{2} N_{a,y}^{\alpha}(\tilde{\theta}_i) N_{b,y}^{\beta}(\tilde{\theta}_i) J^{st}(\tilde{\theta}_i) W_i \, \mathrm{d}\mathbf{r} \\
&= \mu \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \int_{\widehat{Q}_n^{\triangle}} 2N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \, \mathrm{d}\mathbf{r} \\
&\quad + \mu \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \int_{\widehat{Q}_n^{\triangle}} N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \, \mathrm{d}\mathbf{r} \\
&= \mu \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i 2N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} \mathrm{d}\mathbf{r} \\
&\quad + \mu \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} \mathrm{d}\mathbf{r} \\
&= \mu \sum_{i=1}^{2} \frac{\Delta t}{2} \Upsilon(\tilde{\theta}_i) W_i 2N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{2A}{2} \\
&\quad + \mu \sum_{i=1}^{2} \frac{\Delta t}{2} \Upsilon(\tilde{\theta}_i) W_i N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{2A}{2} \\
&= \frac{\mu \Delta t}{2} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \\
&\quad + \frac{\mu \Delta t}{4} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i).
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
B_2^e(a,b) &= \mu \int_{Q_n^e} \bar{N}_{a,y}^{\alpha}(t) \bar{N}_{b,x}^{\beta}(t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\
&= \mu \int_{\widehat{Q}_n^{e}} \bar{N}_{a,y}^{\alpha}(t(\theta)) \bar{N}_{b,x}^{\beta}(t(\theta)) J^{st}(\theta) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\theta \\
&= \mu \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} N_{a,y}^{\alpha}(\tilde{\theta}_i) N_{b,x}^{\beta}(\tilde{\theta}_i) J^{st}(\tilde{\theta}_i) W_i \, \mathrm{d}\mathbf{r} \\
&= \mu \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \int_{\widehat{Q}_n^{\triangle}} N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \, \mathrm{d}\mathbf{r}
\end{aligned}
$$

$$
\begin{aligned}
B_2^e(a,b) &= \mu \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} \mathrm{d}\mathbf{r} \\
&= \mu \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{2A}{2} \\
&= \mu \sum_{i=1}^{2} \frac{\Delta t}{2} \Upsilon(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{2A}{2} \\
&= \frac{\mu \Delta t}{4} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i).
\end{aligned}
$$

$$
\begin{aligned}
B_4^e(a,b) &= \mu \int_{Q_n^e} \bar{N}_{a,x}^{\alpha}(t) \bar{N}_{b,y}^{\beta}(t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\
&= \frac{\mu \Delta t}{4} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i).
\end{aligned}
$$

$$
\begin{aligned}
B_5^e(a,b) &= \mu \int_{Q_n^e} \bar{N}_{a,x}^{\alpha}(t) \bar{N}_{b,x}^{\beta}(t) + 2\bar{N}_{a,y}^{\alpha}(t) \bar{N}_{b,y}^{\beta}(t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\
&= \frac{\mu \Delta t}{4} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \\
&\quad + \frac{\mu \Delta t}{2} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i).
\end{aligned}
$$

4) Jump term (M),

$$
\int_{\Omega_n^l} \rho (\mathbf{w}^h)_n^+ (\mathbf{u}^h)_n^+ \, \mathrm{d}\Omega \tag{3.46}
$$

$$
\begin{aligned}
M_1^e(a,b) &= \rho \int_{Q_n^e} N_a(\mathbf{x}) N_b(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \rho \int_{Q_n^e} N_a(\mathbf{r}) N_b(\mathbf{r}) \tilde{\Upsilon} \, \mathrm{d}\mathbf{r} \\
&= \begin{cases} \rho \tilde{\Upsilon} \frac{2A}{12} = \frac{\rho}{12} \tilde{\Upsilon}, & \text{if } a = b \\ \rho \tilde{\Upsilon} \frac{2A}{24} = \frac{\rho}{24} \tilde{\Upsilon}, & \text{otherwise} \end{cases} \\
M_5^e(a,b) &= M_1^e(a,b)
\end{aligned}
$$

5) Gradient term (G),

$$
\int_{Q_n^e} q^h . \nabla \mathbf{u}^h \, \mathrm{d}Q \tag{3.47}
$$

The integration is analogous to that of (G^T) term

$$G_7^e(a,b) = \int_{Q_n^e} \bar{N}_a^\alpha(\mathbf{x},t)\bar{N}_{b,x}^\beta(t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t = \frac{\Delta t}{12}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)T^\beta(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)$$

$$G_8^e(a,b) = \int_{Q_n^e} \bar{N}_a^\alpha(\mathbf{x},t)\bar{N}_{b,y}^\beta(t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t = \frac{\Delta t}{12}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)T^\beta(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)$$

6) Stabilization terms $(S(\mathbf{U}))$:

$$\sum_{e=1}^{(n_{el})_n}\int_{Q_n^e}\frac{1}{\rho}\left[\tau_1\,\rho\,(\frac{\partial \mathbf{w}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{w}^h)+\tau_2\nabla q^h\right]$$

$$\cdot\left[\rho(\frac{\partial \mathbf{u}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{u}^h-\mathbf{f}^h)-\nabla\cdot\boldsymbol{\sigma}(p^h,\mathbf{u}^h)\right]\mathrm{d}Q+\sum_{e=1}^{(n_{el})_n}\int_{Q_n^e}\tau_3\nabla\cdot\mathbf{w}^h\rho\nabla\cdot\mathbf{u}^h\,\mathrm{d}Q$$

In order to explain the calculation of stabilization terms, we divide the above expression into five parts.

a. Stab1 :

$$\int_{Q_n^e}\tau_1\frac{1}{\rho}\left[\rho(\frac{\partial \mathbf{w}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{w}^h)\rho(\frac{\partial \mathbf{u}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{u}^h)\right]\mathrm{d}Q \tag{3.48}$$

Stab1 is nonlinear term, it will be explained later.

b. Stab2

$$\int_{Q_n^k}\rho(\frac{\partial \mathbf{w}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{w}^h)\frac{\tau_1}{\rho}\nabla p^h\,\mathrm{d}Q \tag{3.49}$$

In x-component : $\tau_1\int_{Q_n^e}\rho(\frac{\partial w^h}{\partial t}+u\frac{\partial w^h}{\partial x}+v\frac{\partial w^h}{\partial y})\frac{\partial p^h}{\partial x}\,\mathrm{d}Q$

$$\begin{aligned}
&S2_3^e(a,b)\\
&=\tau_1\int_{Q_n^e}\left[\bar{N}_{a,t}^\alpha(\mathbf{x},t)+\bar{N}_{a,x}^\alpha(\mathbf{x},t)u+\bar{N}_{a,y}^\alpha(\mathbf{x},t)v\right]\bar{N}_{b,x}^\beta(t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\
&=\tau_1\int_{\widehat{Q}_n^e}\left[N_{a,t}^\alpha(\mathbf{r},\theta)+N_{a,x}^\alpha(\theta)(\textstyle\sum_{\gamma=1}^{2}\sum_{c=1}^{3}u_c^{*\gamma}N_c(\mathbf{r})T^\gamma(\theta)\right]N_{b,x}^\beta(\theta)J^{st}(\theta)\,\mathrm{d}\mathbf{r}\,\mathrm{d}\theta\\
&\quad+\tau_1\int_{\widehat{Q}_n^e}\left[N_{a,y}^\alpha(\theta)(\textstyle\sum_{\gamma=1}^{2}\sum_{c=1}^{3}v_c^{*\gamma}N_c(\mathbf{r})T^\gamma(\theta)\right]N_{b,x}^\beta(\theta)J^{st}(\theta)\,\mathrm{d}\mathbf{r}\,\mathrm{d}\theta
\end{aligned}$$

$$
\begin{aligned}
&S2_3^e(a,b)\\
&= \tau_1 \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left[N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) (\textstyle\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_c(\mathbf{r}) T^{\gamma}(\tilde{\theta}_i) \right]\\
&+ \tau_1 \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left[N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) (\textstyle\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_c(\mathbf{r}) T^{\gamma}(\tilde{\theta}_i) \right]\\
&+ \tau_1 \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left[(V_r(\mathbf{r},\tilde{\theta}_i) N_{a,r} + V_s(\mathbf{r},\tilde{\theta}_i) N_{a,s}) T^{\alpha}(\tilde{\theta}_i) \right]\\
&+ \tau_1 \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{(-1)^{\alpha}}{\Delta t} N_a(\mathbf{r})\\
&= \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i)) \int_{\widehat{Q}_n^{\triangle}} N_c(\mathbf{r}) \mathrm{d}\mathbf{r} \right]\\
&+ \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (v_c^{*1} T^1(\tilde{\theta}_i) + v_c^{*2} T^2(\tilde{\theta}_i)) \int_{\widehat{Q}_n^{\triangle}} N_c(\mathbf{r}) \mathrm{d}\mathbf{r} \right]\\
&+ \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} \left(\frac{-x_{\theta}(\mathbf{r}) y_s(\tilde{\theta}_i) + y_{\theta}(\mathbf{r}) x_s(\tilde{\theta}_i)}{J^{st}(\tilde{\theta}_i)} \right) N_{a,r} \mathrm{d}\mathbf{r}\\
&+ \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} \left(\frac{x_{\theta}(\mathbf{r}) y_r(\tilde{\theta}_i) - y_{\theta}(\mathbf{r}) x_r(\tilde{\theta}_i)}{J^{st}(\tilde{\theta}_i)} \right) N_{a,s} \mathrm{d}\mathbf{r}\\
&+ \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{(-1)^{\alpha}}{\Delta t} \int_{\widehat{Q}_n^{\triangle}} N_a(\mathbf{r}) \mathrm{d}\mathbf{r}
\end{aligned}
$$

$$
\begin{aligned}
&S2_3^e(a,b)\\
&= \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i)) \frac{det}{6}\right]\\
&\quad + \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (v_c^{*1} T^1(\tilde{\theta}_i) + v_c^{*2} T^2(\tilde{\theta}_i)) \frac{det}{6}\right]\\
&\quad + \tau_1 \sum_{i=1}^{2} N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left(A_3 N_{a,r} + A_4 N_{a,s}\right)\\
&\quad + \tau_1 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{(-1)^{\alpha}}{\Delta t} \frac{2A}{6}\\
&= \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i))\right]\\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (v_c^{*1} T^1(\tilde{\theta}_i) + v_c^{*2} T^2(\tilde{\theta}_i))\right]\\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left(-\textstyle\sum \frac{\Delta x_c}{2} y_s(\tilde{\theta}_i) + \sum \frac{\Delta y_c}{2} x_s(\tilde{\theta}_i)\right) N_{a,r}\\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \left(\textstyle\sum \frac{\Delta x_c}{2} y_r(\tilde{\theta}_i) - \sum \frac{\Delta y_c}{2} x_r(\tilde{\theta}_i)\right) N_{a,s}\\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \frac{(-1)^{\alpha}}{\Delta t},
\end{aligned}
$$

where

$$
\begin{aligned}
A_3 &= \left(-\textstyle\sum \frac{\Delta x_c}{2} y_s(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} N_c(\mathbf{r}) \mathrm{d}\mathbf{r} + \sum \frac{\Delta y_c}{2} x_s(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} N_c(\mathbf{r}) \mathrm{d}\mathbf{r}\right)\\
A_4 &= \left(\textstyle\sum \frac{\Delta x_c}{2} y_r(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} N_c(\mathbf{r}) \mathrm{d}\mathbf{r} - \sum \frac{\Delta y_c}{2} x_r(\tilde{\theta}_i) \int_{\widehat{Q}_n^{\triangle}} N_c(\mathbf{r}) \mathrm{d}\mathbf{r}\right)
\end{aligned}
$$

Similarly, in y-component : $\tau_1 \int_{Q_n^e} \rho(\frac{\partial w^h}{\partial t} + u\frac{\partial w^h}{\partial x} + v\frac{\partial w^h}{\partial y}) \frac{\partial p^h}{\partial y} \, \mathrm{d}Q$
(using $N_{b,y}$ instead of $N_{b,x}$)

$$
S2_6^e(a,b) = \tau_1 \int_{Q_n^e} \left[\bar{N}_{a,t}^{\alpha}(\mathbf{x},t) + \bar{N}_{a,x}^{\alpha}(\mathbf{x},t) u + \bar{N}_{a,y}^{\alpha}(\mathbf{x},t) v\right] \bar{N}_{b,y}^{\beta}(t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t
$$

$$
\begin{aligned}
&S2_6^e(a,b)\\
&= \frac{\tau_1}{6}\sum_{i=1}^{2} J^{st}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)\left[\textstyle\sum_{c=1}^{3}(u_c^{*1}T^1(\tilde{\theta}_i)+u_c^{*2}T^2(\tilde{\theta}_i))\right]\\
&\quad+\frac{\tau_1}{6}\sum_{i=1}^{2} J^{st}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)\left[\textstyle\sum_{c=1}^{3}(v_c^{*1}T^1(\tilde{\theta}_i)+v_c^{*2}T^2(\tilde{\theta}_i))\right]\\
&\quad+\frac{\tau_1}{6}\sum_{i=1}^{2} N_{b,y}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)\left(-\textstyle\sum\frac{\Delta x_c}{2}y_s(\tilde{\theta}_i)+\sum\frac{\Delta y_c}{2}x_s(\tilde{\theta}_i)\right)N_{a,r}\\
&\quad+\frac{\tau_1}{6}\sum_{i=1}^{2} N_{b,y}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)\left(\textstyle\sum\frac{\Delta x_c}{2}y_r(\tilde{\theta}_i)-\sum\frac{\Delta y_c}{2}x_r(\tilde{\theta}_i)\right)N_{a,s}\\
&\quad+\frac{\tau_1}{6}\sum_{i=1}^{2} J^{st}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\frac{(-1)^{\alpha}}{\Delta t}
\end{aligned}
$$

c. Stab3,

$$\int_{Q_n^e}\frac{\tau_2}{\rho}\nabla q^h\cdot\nabla p^h\,\mathrm{d}Q \tag{3.50}$$

$$
\begin{aligned}
&S3_9^e(a,b)\\
&=\frac{\tau_2}{\rho}\int_{Q_n^e}\nabla\bar{N}_a^{\alpha}(\mathbf{x},t)\cdot\nabla\bar{N}_b^{\beta}(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\
&=\frac{\tau_2}{\rho}\int_{Q_n^e}\nabla N_a^{\alpha}(\mathbf{r},\theta)\cdot N_b^{\beta}(\mathbf{r},\theta)J^{st}(\theta)\,\mathrm{d}\mathbf{r}\,\mathrm{d}\theta\\
&=\frac{\tau_2}{\rho}\int_{Q_n^e}\left(N_{a,x}^{\alpha}(\theta)N_{b,x}^{\beta}(\theta)+N_{a,y}^{\alpha}(\theta)N_{b,y}^{\beta}(\theta)\right)J^{st}(\theta)\,\mathrm{d}\mathbf{r}\,\mathrm{d}\theta\\
&=\frac{\tau_2}{\rho}\int_{Q_n^e}\left(N_{a,x}(\theta)T^{\alpha}(\theta)N_{b,x}(\theta)T^{\beta}(\theta)+N_{a,y}(\theta)T^{\alpha}(\theta)N_{b,y}(\theta)T^{\beta}(\theta)\right)J^{st}(\theta)\,\mathrm{d}\mathbf{r}\,\mathrm{d}\theta\\
&=\frac{\tau_2}{\rho}\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_i\left(N_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)+N_{a,y}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)\right)\int_{Q_n^{\triangle}}\mathrm{d}\mathbf{r}\\
&=\frac{\tau_2}{\rho}\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_i\left(N_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)+N_{a,y}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)\right)\frac{2A}{2}
\end{aligned}
$$

$$
\begin{aligned}
&S3_9^e(a,b)\\
&= \frac{\tau_2}{2\rho}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)\left(N_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)+N_{a,y}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)\right)\\
&= \frac{\tau_2\Delta t}{4\rho}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\Upsilon(\tilde{\theta}_i)\left(N_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)+N_{a,y}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)\right)
\end{aligned}
$$

d. Stab4,

$$\tau_2\int_{Q_n^e}\nabla q^h\left(\frac{\partial \mathbf{u}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{u}^h\right) \tag{3.51}$$

Stab4 is nonlinear term, it will be explained later.

e. Stab5,

$$\tau_3\int_{Q_n^e}\nabla\mathbf{w}^h\rho\nabla\mathbf{u}^h\,\mathrm{d}Q \tag{3.52}$$

$$
\begin{aligned}
S5_1^e(a,b) &= \tau_3\rho\int_{Q_n^e}\bar{N}_{a,x}^{\alpha}(t)\bar{N}_{b,x}^{\beta}(t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\
&= \tau_3\rho\int_{Q_n^e}N_{a,x}^{\alpha}(\theta)N_{b,x}^{\beta}(\theta)J^{st}(\theta)\,\mathrm{d}\mathbf{r}\,\mathrm{d}\theta\\
&= \tau_3\rho\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_iN_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)\int_{Q_n^{\triangle}}\mathrm{d}\mathbf{r}\\
&= \tau_3\rho\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_iN_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)\frac{2A}{2}\\
&= \frac{\tau_3\rho}{2}\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)\\
&= \frac{\tau_3\rho\Delta t}{4}\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\Upsilon(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
S5_2^e(a,b) &= \tau_3\rho\int_{Q_n^e}\bar{N}_{a,x}^{\alpha}(t)\bar{N}_{b,y}^{\beta}(t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\
&= \frac{\tau_3\rho}{2}\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)\\
&= \frac{\tau_3\rho\Delta t}{4}\sum_{i=1}^{2}T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\Upsilon(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)
\end{aligned}
$$

$$
\begin{aligned}
S5_4^e(a,b) &= \tau_3\rho \int_{Q_n^e} \bar{N}_{a,y}^{\alpha}(t)\bar{N}_{b,x}^{\beta}(t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= \frac{\tau_3\rho}{2}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i) \\
&= \frac{\tau_3\rho\Delta t}{4}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\Upsilon(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)
\end{aligned}
$$

$$
\begin{aligned}
S5_5^e(a,b) &= \tau_3\rho \int_{Q_n^e} \bar{N}_{a,y}^{\alpha}(t)\bar{N}_{b,y}^{\beta}(t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= \frac{\tau_3\rho}{2}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i) \\
&= \frac{\tau_3\rho\Delta t}{4}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\Upsilon(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)
\end{aligned}
$$

7) F_1 and F_2

$$\int_{Q_n^e} \rho\mathbf{w}^h\mathbf{f}\,\mathrm{d}Q \tag{3.53}$$

$$
\begin{aligned}
F_1^e(a) &= \rho \int_{Q_n^e} \bar{N}_a^{\alpha}(\mathbf{x},t)f_x\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= \rho \int_{\widehat{Q}_n^e} \bar{N}_a^{\alpha}(\mathbf{x}(\mathbf{r},\theta),t(\theta))f_x J^{st}(\theta)\mathrm{d}\mathbf{r}\ \mathrm{d}\theta \\
&= \rho \int_{\widehat{Q}_n^{\triangle}} \sum_{i=1}^{2} N_a(\mathbf{r})T^{\alpha}(\tilde{\theta}_i)f_x J^{st}(\tilde{\theta}_i)W_i\ \mathrm{d}\mathbf{r} \\
&= \rho f_x \sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_i \int_{\widehat{Q}_n^{\triangle}} N_a(\mathbf{r})\mathrm{d}\mathbf{r} \\
&= \rho f_x \sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_i \frac{2A}{6} \\
&= \frac{\rho f_x}{6}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)J^{st}(\tilde{\theta}_i) \\
&= \frac{\rho f_x\Delta t}{12}\sum_{i=1}^{2} T^{\alpha}(\tilde{\theta}_i)\Upsilon(\tilde{\theta}_i)
\end{aligned}
$$

Similarly for $F_2^e(a)$,

$$\begin{aligned} F_2^e(a) &= \rho \int_{Q_n^e} \bar{N}_a^\alpha(\mathbf{x}, t) f_y \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\ &= \frac{\rho f_y \Delta t}{12} \sum_{i=1}^{2} T^\alpha(\tilde{\theta}_i) \Upsilon(\tilde{\theta}_i) \end{aligned}$$

8) F_{s4}

$$\int_{Q_n^e} \tau_2 \nabla q \mathbf{f} \, \mathrm{d}Q \tag{3.54}$$

$$\begin{aligned} & F_{S4_3}^e(a) \\ &= \tau_2 \int_{Q_n^e} \bar{N}_{a,x}^\alpha(t) f_x + \bar{N}_{a,y}^\alpha(t) f_y \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\ &= \tau_2 \left[\int_{\widehat{Q}_n^e} \bar{N}_{a,x}^\alpha(\theta), t(\theta)) f_x + \bar{N}_{a,y}^\alpha(\theta), t(\theta)) f_y \right] J^{st}(\theta) \mathrm{d}\mathbf{r} \, \mathrm{d}\theta \\ &= \tau_2 \sum_{i=1}^{2} \left[\int_{\widehat{Q}_n^\triangle} N_{a,x}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_x + N_{a,y}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_y \right] J^{st}(\tilde{\theta}_i) W_i \mathrm{d}\mathbf{r} \\ &= \tau_2 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \left[N_{a,x}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_x + N_{a,y}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_y \right] \int_{\widehat{Q}_n^\triangle} \mathrm{d}\mathbf{r} \\ &= \tau_2 \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i \left[N_{a,x}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_x + N_{a,y}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_y \right] \frac{2A}{2} \\ &= \tau_2 \frac{2A}{2} \frac{\Delta t}{2} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) W_i \left[N_{a,x}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_x + N_{a,y}(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) f_y \right] \\ &= \frac{\tau_2 \Delta t}{4} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^\alpha(\tilde{\theta}_i) \left[N_{a,x}(\tilde{\theta}_i) f_x + N_{a,y}(\tilde{\theta}_i) f_y \right] \end{aligned}$$

9) Right-hand side of jump term F_M,

$$\int_{\Omega^e} \rho \mathbf{w}(\mathbf{u}^-) \, \mathrm{d}\Omega \tag{3.55}$$

$$\begin{aligned} F^e_{M_1}(a) &= \rho \int_{\Omega^e} N_a(\mathbf{x}) \sum_{b=1}^{3} u_b^- N_b(\mathbf{x})\, \mathrm{d}x \\ &= \rho \int_{\Omega^e} N_a(\mathbf{r}) \sum_{b=1}^{3} u_b^- N_b(\mathbf{r}) \widehat{\Upsilon}\, \mathrm{d}r \\ &= \rho \frac{2A}{24} \sum_{b=1}^{3} (u_b^- + u_a) \widehat{\Upsilon} \\ &= \frac{\rho \widehat{\Upsilon}}{24} \sum_{b=1}^{3} (u_b^- + u_a) \end{aligned}$$

Similarly,

$$\begin{aligned} F^e_{M_2}(a) &= \rho \int_{\Omega^e} N_a(\mathbf{x}) \sum_{b=1}^{3} v_b^- N_b(\mathbf{x})\, \mathrm{d}x \\ &= \frac{\rho \widehat{\Upsilon}}{24} \sum_{b=1}^{3} (v_b^- + v_a) \end{aligned}$$

The nonlinear terms will be written in increment form, i.e., we take the solution $\mathbf{u}$ in the form

$$\mathbf{u} = \mathbf{u}^* + \Delta\mathbf{u},$$

and substitute it into the corresponding terms and calculate the resulting contribution to the left-hand side of the system.
1) Convection

$$\begin{aligned} &\int_{Q_n^e} \rho \mathbf{w}(\mathbf{u} \cdot \nabla \mathbf{u})\, \mathrm{d}Q \\ &= \rho \int_{Q_n^e} \mathbf{w}((\mathbf{u}^* + \Delta\mathbf{u}) \cdot \nabla(\mathbf{u}^* + \Delta\mathbf{u}))\, \mathrm{d}Q \\ &= \rho \left[\int_{Q_n^e} \mathbf{w}(\mathbf{u}^* \cdot \nabla \mathbf{u}^*)\, \mathrm{d}Q + \int_{Q_n^e} \mathbf{w}(\mathbf{u}^* \cdot \nabla \Delta\mathbf{u})\, \mathrm{d}Q + \int_{Q_n^e} \mathbf{w}(\Delta\mathbf{u} \cdot \nabla \mathbf{u}^*)\, \mathrm{d}Q \right] \\ &\quad + \rho \left[\int_{Q_n^e} \mathbf{w}(\Delta\mathbf{u} \cdot \nabla \Delta\mathbf{u})\, \mathrm{d}Q \right] \end{aligned}$$

The first integral contributes to right-hand side since $\mathbf{u}^*$ is known. The second integral contributes to the left-hand side and we will call it, convection term ($C(\mathbf{U}^*)$). The third integral contributes to the left-hand side and

we will call it convection derivative term ($CC(\mathbf{U}^*)$). The last integral is dropped since it is quadratic in $\mathbf{\Delta u}$ [31].
a. Convection term ($C(\mathbf{U}^*)$)

$$\rho \int_{Q_n^e} \mathbf{w}(\mathbf{u}^* \cdot \nabla \Delta \mathbf{u}) \, \mathrm{d}Q \tag{3.56}$$

$$
\begin{aligned}
&C_1^e(\mathbf{U}^*)(a,b) \\
&= \rho \int_{Q_n^e} \bar{N}_a^\alpha(\mathbf{x},t)(\mathbf{u}^* \cdot \nabla \bar{N}_b^\beta(t)) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\
&= \rho \int_{Q_n^e} \bar{N}_a^\alpha(\mathbf{x},t) \left[u^* \bar{N}_{b,x}^\beta(t) + v^* \bar{N}_{b,y}^\beta(t) \right] \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \\
&= \rho \int_{\widehat{Q}_n^e} N_a^\alpha(\mathbf{x}(\mathbf{r},\theta),t(\theta)) \left[u^* N_{b,x}^\beta(t(\theta)) + v^* N_{b,y}^\beta(t(\theta)) \right] J^{st}(\theta) \, \mathrm{d}\mathbf{r} \, \mathrm{d}t \\
&= \rho \int_{\widehat{Q}_n^e} N_a(\mathbf{r}) T^\alpha(\theta) \left[u^* N_{b,x}(\theta) T^\beta(\theta) + v^* N_{b,y}(\theta) T(\theta) \right] J^{st}(\theta) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\theta \\
&= \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i T^\alpha(\tilde{\theta}_i) T^\beta(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) \int_{\widehat{Q}_n^\triangle} N_a(\mathbf{r}) \textstyle\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_c(\mathbf{r}) T^\gamma(\tilde{\theta}_i) \, \mathrm{d}\mathbf{r} \\
&\quad + \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i T^\alpha(\tilde{\theta}_i) T^\beta(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) \int_{\widehat{Q}_n^\triangle} N_a(\mathbf{r}) \textstyle\sum_{\gamma=1}^{2} \sum_{c=1}^{3} v_c^{*\gamma} N_c(\mathbf{r}) T^\gamma(\tilde{\theta}_i) \, \mathrm{d}\mathbf{r} \\
&= \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i T^\alpha(\tilde{\theta}_i) T^\beta(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) \\
&\quad \left[\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i)) \int_{\widehat{Q}_n^\triangle} N_c(\mathbf{r}) N_a(\mathbf{r}) \, \mathrm{d}\mathbf{r} \right] \\
&\quad + \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i T^\alpha(\tilde{\theta}_i) T^\beta(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) \\
&\quad \left[\sum_{c=1}^{3} (v_c^{*1} T^1(\tilde{\theta}_i) + v_c^{*2} T^2(\tilde{\theta}_i)) \int_{\widehat{Q}_n^\triangle} N_c(\mathbf{r}) N_a(\mathbf{r}) \, \mathrm{d}\mathbf{r} \right] \\
&= \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i T^\alpha(\tilde{\theta}_i) T^\beta(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) \frac{2A}{24} (A_5 + A_6)
\end{aligned}
$$

$$C_1^e(\mathbf{U}^*)(a,b) = \rho \sum_{i=1}^{2} \frac{\Delta t}{2} \Upsilon(\tilde{\theta}_i) W_i T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) \frac{2A}{24} (A_5 + A_6)$$
$$= \rho \frac{\Delta t}{48} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) (A_5 + A_6)$$

where

$$A_5 = \left[\sum_{c=1}^{3} (u_c^{*1} + u_a^{*1}) T^1(\tilde{\theta}_i) + (u_c^{*2} + u_a^{*2}) T^2(\tilde{\theta}_i) \right]$$
$$A_6 = \left[\sum_{c=1}^{3} (v_c^{*1} + v_a^{*1}) T^1(\tilde{\theta}_i) + (v_c^{*2} + v_a^{*2}) T^2(\tilde{\theta}_i) \right] \tag{3.57}$$

$$C_5^e(\mathbf{U}^*)(a,b) = C_1^e(\mathbf{U}^*)(a,b)$$

b. Convection derivative ($CC(\mathbf{U}^*)$),

$$\rho \int_{Q_n^e} \mathbf{w}(\Delta \mathbf{u} \cdot \nabla \mathbf{u}^*) \, \mathrm{d}Q \tag{3.58}$$

$$\rho \int_{Q_n^e} \mathbf{w}(\Delta \mathbf{u} \cdot \nabla \mathbf{u}^*) \, \mathrm{d}Q = \begin{cases} CC_1(\mathbf{U}^*) = \rho \int_{Q_n^e} w_1 (\Delta u \frac{\partial u^*}{\partial x}) \, \mathrm{d}Q \\ CC_2(\mathbf{U}^*) = \rho \int_{Q_n^e} w_1 (\Delta v \frac{\partial u^*}{\partial y}) \, \mathrm{d}Q \\ CC_4(\mathbf{U}^*) = \rho \int_{Q_n^e} w_2 (\Delta u \frac{\partial v^*}{\partial x}) \, \mathrm{d}Q \\ CC_1(\mathbf{U}^*) = \rho \int_{Q_n^e} w_2 (\Delta v \frac{\partial v^*}{\partial y}) \, \mathrm{d}Q \end{cases}$$

$CC_1^e(\mathbf{U}^*)(a,b)$

$$= \rho \int_{Q_n^e} \bar{N}_a^{\alpha}(\mathbf{x},t) \bar{N}_b^{\beta}(\mathbf{x},t) \left(\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} \bar{N}_{c,x}^{\gamma}(t) \right) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t$$
$$= \rho \int_{\widehat{Q}_n^e} N_a^{\alpha}(\mathbf{r},\theta) N_b^{\beta}(\mathbf{r},\theta) \left(\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_{c,x}^{\gamma}(\theta) \right) J^{st}(\theta) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\theta$$
$$= \rho \int_{Q_n^{\triangle}} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i N_a(\mathbf{r}) T^{\alpha}(\tilde{\theta}_i) N_b(\mathbf{r}) T^{\beta}(\tilde{\theta}_i) \left(\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_{c,x}(\tilde{\theta}_i) T^{\gamma}(\theta) \right) \, \mathrm{d}\mathbf{r}$$
$$= \rho \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) W_i T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_{c,x}(\tilde{\theta}_i) T^{\gamma}(\tilde{\theta}_i) \right) \int_{Q_n^e} N_a(\mathbf{r}) N_b(\mathbf{r}) \, \mathrm{d}\mathbf{r}$$

$$
\begin{aligned}
&CC_1^e(\mathbf{U}^*)(a,b)\\
&= \begin{cases}
\rho \sum_{i=1}^{2} \frac{\Delta t}{2} \Upsilon(\tilde{\theta}_i) W_i T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_{c,x}(\tilde{\theta}_i) T^{\gamma}(\tilde{\theta}_i) \right) \frac{2A}{12}, \\
\quad \text{if } a = b. \\
\rho \sum_{i=1}^{2} \frac{\Delta t}{2} \Upsilon(\tilde{\theta}_i) W_i T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} N_{c,x}(\tilde{\theta}_i) T^{\gamma}(\tilde{\theta}_i) \right) \frac{2A}{24}, \\
\quad \text{otherwise.}
\end{cases}\\
&= \begin{cases}
\frac{\rho \Delta t}{24} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i)) N_{c,x}(\tilde{\theta}_i) \right), \\
\quad \text{if } a = b. \\
\frac{\rho \Delta t}{48} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i) N_{c,x}(\tilde{\theta}_i) \right), \\
\quad \text{otherwise.}
\end{cases}
\end{aligned}
$$

And similarly for other $(CC(\mathbf{U}^*))$

$$
\begin{aligned}
&CC_2^e(\mathbf{U}^*)(a,b)\\
&= \rho \int_{Q_n^e} \bar{N}_a^{\alpha}(\mathbf{x},t) \bar{N}_b^{\beta}(\mathbf{x},t) \left(\sum_{\gamma=1}^{2} \sum_{c=1}^{3} u_c^{*\gamma} \bar{N}_{c,y}^{\gamma}(t) \right) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t\\
&= \begin{cases}
\frac{\rho \Delta t}{24} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i)) N_{c,y}(\tilde{\theta}_i) \right), \\
\quad \text{if } a = b. \\
\frac{\rho \Delta t}{48} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i) N_{c,y}(\tilde{\theta}_i) \right), \\
\quad \text{otherwise.}
\end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&CC_4^e(\mathbf{U}^*)(a,b)\\
&= \rho \int_{Q_n^e} \bar{N}_a^\alpha(\mathbf{x},t)\bar{N}_b^\beta(\mathbf{x},t)\left(\sum_{\gamma=1}^{2}\sum_{c=1}^{3} v_c^{*\gamma}\bar{N}_{c,y}^{\gamma}(t)\right)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\
&= \begin{cases} \frac{\rho\Delta t}{24}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)T^\beta(\tilde{\theta}_i)\left(\sum_{c=1}^{3}(v_c^{*1}T^1(\tilde{\theta}_i)+v_c^{*2}T^2(\tilde{\theta}_i))N_{c,x}(\tilde{\theta}_i)\right), \\ \quad \text{if } a=b. \\ \frac{\rho\Delta t}{48}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)T^\beta(\tilde{\theta}_i)\left(\sum_{c=1}^{3}(v_c^{*1}T^1(\tilde{\theta}_i)+v_c^{*2}T^2(\tilde{\theta}_i)N_{c,x}(\tilde{\theta}_i)\right), \\ \quad \text{otherwise.} \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&CC_5^e(\mathbf{U}^*)(a,b)\\
&= \rho \int_{Q_n^e} \bar{N}_a^\alpha(\mathbf{x},t)\bar{N}_b^\beta(\mathbf{x},t)\left(\sum_{\gamma=1}^{2}\sum_{c=1}^{3} v_c^{*\gamma}\bar{N}_{c,y}^{\gamma}(t)\right)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\
&= \begin{cases} \frac{\rho\Delta t}{24}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)T^\beta(\tilde{\theta}_i)\left(\sum_{c=1}^{3}(v_c^{*1}T^1(\tilde{\theta}_i)+v_c^{*2}T^2(\tilde{\theta}_i))N_{c,y}(\tilde{\theta}_i)\right), \\ \quad \text{if } a=b \\ \frac{\rho\Delta t}{48}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)T^\beta(\tilde{\theta}_i)\left(\sum_{c=1}^{3}(v_c^{*1}T^1(\tilde{\theta}_i)+v_c^{*2}T^2(\tilde{\theta}_i)N_{c,y}(\tilde{\theta}_i)\right), \\ \quad \text{otherwise} \end{cases}
\end{aligned}
$$

2) Stab4

$$
\begin{aligned}
&\tau_2\int_{Q_n^e}\nabla q\left(\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)\mathrm{d}Q\\
&= \tau_2\left[\int_{Q_n^e}\nabla q\left(\frac{\partial\Delta\mathbf{u}}{\partial t}+\mathbf{u}^*\cdot\nabla\Delta\mathbf{u}\right)\mathrm{d}Q+\int_{Q_n^e}\nabla q\left(\frac{\partial\Delta\mathbf{u}}{\partial t}+\Delta\mathbf{u}\cdot\nabla\mathbf{u}^*\right)\mathrm{d}Q\right]\\
&\quad+\tau_2\left[\int_{Q_n^e}\nabla q\left(\frac{\partial\Delta\mathbf{u}}{\partial t}+\Delta\mathbf{u}\cdot\nabla\Delta\mathbf{u}\right)\mathrm{d}Q+\int_{Q_n^e}\nabla q\left(\frac{\partial\mathbf{u}^*}{\partial t}+\mathbf{u}^*\cdot\nabla\mathbf{u}^*\right)\mathrm{d}Q\right]
\end{aligned}
$$

The first integral contributes to the left-hand side and we will call it Stab4 term ($S4(\mathbf{U}^*)$). The second integral is dropped because of convergence reasons. The third integral is also dropped since it is quadratic in $\Delta\mathbf{u}$. The last integral contributes to the right-hand side since $\mathbf{u}^*$ is known.

Stab4 term ($S4(\mathbf{U}^*)$),

$$\tau_2 \int_{Q_n^e} \nabla q \left(\frac{\partial \Delta \mathbf{u}}{\partial t} + \mathbf{u}^* \cdot \nabla \Delta \mathbf{u} \right) \mathrm{d}Q \tag{3.59}$$

Stab4 is similar with stab2 term, only we reverse a with b and α with β.

$$\begin{aligned}
&S4_7^e(\mathbf{U}^*)(a,b) \\
&= \tau_1 \int_{Q_n^e} \left[\bar{N}_{b,t}^{\beta}(\mathbf{x},t) + \bar{N}_{b,x}^{\beta}(\mathbf{x},t)u + \bar{N}_{b,y}^{\beta}(\mathbf{x},t)v \right] \bar{N}_{a,x}^{\alpha}(t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i)) \right] \\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (v_c^{*1} T^1(\tilde{\theta}_i) + v_c^{*2} T^2(\tilde{\theta}_i)) \right] \\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(- \textstyle\sum \frac{\Delta x_c}{2} y_s(\tilde{\theta}_i) + \sum \frac{\Delta y_c}{2} x_s(\tilde{\theta}_i) \right) N_{b,r} \\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(\textstyle\sum \frac{\Delta x_c}{2} y_r(\tilde{\theta}_i) - \sum \frac{\Delta y_c}{2} x_r(\tilde{\theta}_i) \right) N_{b,s} \\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) \frac{(-1)^{\beta}}{\Delta t}
\end{aligned}$$

$$\begin{aligned}
&S4_8^e(\mathbf{U}^*)(a,b) \\
&= \tau_1 \int_{Q_n^e} \left[\bar{N}_{b,t}^{\beta}(\mathbf{x},t) + \bar{N}_{b,x}^{\beta}(\mathbf{x},t)u + \bar{N}_{b,y}^{\beta}(\mathbf{x},t)v \right] \bar{N}_{a,y}^{\alpha}(t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \\
&= \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,x}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i)) \right] \\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} J^{st}(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{b,y}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left[\textstyle\sum_{c=1}^{3} (v_c^{*1} T^1(\tilde{\theta}_i) + v_c^{*2} T^2(\tilde{\theta}_i)) \right] \\
&\quad + \frac{\tau_1}{6} \sum_{i=1}^{2} N_{a,y}(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) T^{\beta}(\tilde{\theta}_i) \left(- \textstyle\sum \frac{\Delta x_c}{2} y_s(\tilde{\theta}_i) + \sum \frac{\Delta y_c}{2} x_s(\tilde{\theta}_i) \right) N_{b,r}
\end{aligned}$$

$$+\frac{\tau_1}{6}\sum_{i=1}^{2}N_{a,y}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\left(\sum\frac{\Delta x_c}{2}y_r(\tilde{\theta}_i)-\sum\frac{\Delta y_c}{2}x_r(\tilde{\theta}_i)\right)N_{b,s}$$
$$+\frac{\tau_1}{6}\sum_{i=1}^{2}J^{st}(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)\frac{(-1)^{\beta}}{\Delta t}$$

3) Stab1

$$\int_{Q_n^e}\tau_1\frac{1}{\rho}\left[\rho\left(\frac{\partial\mathbf{w}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{w}\right)\rho\left(\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)\right]\mathrm{d}Q \tag{3.60}$$

$$\int_{Q_n^e}\tau_1\frac{1}{\rho}\left[\rho\left(\frac{\partial\mathbf{w}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{w}\right)\rho\left(\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)\right]\mathrm{d}Q$$
$$=\int_{Q_n^e}\tau_1\rho\left[\left(\frac{\partial\mathbf{w}}{\partial t}+\mathbf{u}^*\cdot\nabla\mathbf{w}\right)\left(\frac{\partial(\mathbf{u}^*+\Delta\mathbf{u})}{\partial t}+(\mathbf{u}^*+\Delta\mathbf{u})\cdot\nabla(\mathbf{u}^*+\Delta\mathbf{u})\right)\right]\mathrm{d}Q$$

Dropping $\Delta\mathbf{u}$ in the first bracket improves the convergence rate [31].

$$=\tau_1\rho\left[\int_{Q_n^e}\left(\frac{\partial\mathbf{w}}{\partial t}+\mathbf{u}^*\cdot\nabla\mathbf{w}\right)\left(\frac{\partial\mathbf{u}^*}{\partial t}+\mathbf{u}^*\cdot\nabla\mathbf{u}^*\right)\mathrm{d}Q\right]$$
$$+\tau_1\rho\left[\int_{Q_n^e}\left(\frac{\partial\mathbf{w}}{\partial t}+\mathbf{u}^*\cdot\nabla\mathbf{w}\right)\left(\frac{\partial\Delta\mathbf{u}}{\partial t}+\mathbf{u}^*\cdot\nabla\Delta\mathbf{u}\right)\mathrm{d}Q\right]$$
$$+\tau_1\rho\left[\int_{Q_n^e}\left(\frac{\partial\mathbf{w}}{\partial t}+\mathbf{u}^*\cdot\nabla\mathbf{w}\right)(\Delta\mathbf{u}\cdot\nabla\mathbf{u}^*)\,\mathrm{d}Q\right]$$
$$+\tau_1\rho\left[\int_{Q_n^e}\left(\frac{\partial\mathbf{w}}{\partial t}+\mathbf{u}^*\cdot\nabla\mathbf{w}\right)(\Delta\mathbf{u}\cdot\nabla\Delta\mathbf{u})\,\mathrm{d}Q\right]$$

The first integral contributes to the the right-hand side since $\mathbf{u}^*$ is known. The second integral contributes to the left-hand side and we will call it stab1 term ($S1(\mathbf{U}^*)$). The third integral contributes to the left-hand side and we will call it stab1 derivative term ($SS1(\mathbf{U}^*)$). The last integral is dropped since it is quadratic in $\Delta\mathbf{u}$. Since stab1 term (second integral) is complicated, we divide it into stab1a term, stab1b term, stab1c term and stab1d term,

a. stab1a

$$\tau_1\rho\int_{Q_n^e}\frac{\partial\mathbf{w}}{\partial t}\frac{\partial\Delta\mathbf{u}}{\partial t}\,dQ \tag{3.61}$$

$$
\begin{aligned}
&S1a_1^e(a,b)\\
&=\tau_1\rho\int_{Q_n^e}\bar{N}_{a,t}^{\alpha}(\mathbf{x},t)\bar{N}_{b,t}^{\beta}(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t\\
&=\tau_1\rho\int_{\widehat{Q}_n^e}\bar{N}_{a,t}^{\alpha}(\mathbf{x}(\mathbf{r},\theta),t(\theta))\bar{N}_{b,t}^{\beta}(\mathbf{x}(\mathbf{r},\theta),t(\theta))J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta\\
&=\tau_1\rho\int_{\widehat{Q}_n^{\triangle}}\sum_{i=1}^{2}N_{a,t}^{\alpha}(\mathbf{r},\tilde{\theta}_i)N_{b,t}^{\beta}(\mathbf{r},\tilde{\theta}_i)J^{st}(\tilde{\theta}_i)W_i\ \mathrm{d}\mathbf{r}\\
&=\tau_1\rho\sum_{i=1}^{2}J^{st}(\tilde{\theta}_i)W_i\ \int_{\widehat{Q}_n^{\triangle}}N_{a,t}^{\alpha}(\mathbf{r},\tilde{\theta}_i)N_{b,t}^{\beta}(\mathbf{r},\tilde{\theta}_i)\mathrm{d}\mathbf{r}\\
&=\tau_1\rho\sum_{i=1}^{2}J^{st}(\tilde{\theta}_i)W_i\int_{\widehat{Q}_n^{\triangle}}\left[\left(V_r(\mathbf{r},\tilde{\theta}_i)N_{a,r}+V_s(\mathbf{r},\tilde{\theta}_i)N_{a,s}\right)T^{\alpha}(\tilde{\theta}_i)+\frac{(-1)^{\alpha}}{\Delta t}N_a(\mathbf{r})\right]\\
&\quad\left[\left(V_r(\mathbf{r},\tilde{\theta}_i)N_{b,r}+V_s(\mathbf{r},\tilde{\theta}_i)N_{b,s}\right)T^{\beta}(\tilde{\theta}_i)+\frac{(-1)^{\beta}}{\Delta t}N_b(\mathbf{r})\right]\mathrm{d}\mathbf{r}
\end{aligned}
$$

$$S1a_5^e(a,b)=S1a_1^e(a,b)$$

b. stab1b

$$\tau_1\rho\int_{Q_n^e}\frac{\partial\mathbf{w}}{\partial t}\mathbf{u}^*\cdot\nabla\Delta\mathbf{u}\mathrm{d}Q \tag{3.62}$$

$$
\begin{aligned}
&S1b_1^e(a,b)\\
&=\tau_1\rho\int_{Q_n^e}\bar{N}_{a,t}^{\alpha}(\mathbf{x},t)\mathbf{u}^*\cdot\nabla barN_b^{\beta}(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t\\
&=\tau_1\rho\int_{\widehat{Q}_n^e}\bar{N}_{a,t}^{\alpha}(\mathbf{x}(\mathbf{r},\theta),t(\theta))\bar{N}_{b,t}^{\beta}(\mathbf{x}(\mathbf{r},\theta),t(\theta))J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta\\
&=\tau_1\rho\int_{\widehat{Q}_n^e}N_{a,t}^{\alpha}(\mathbf{r},\theta)\left(\mathbf{u}^*\cdot N_{b,x}^{\beta}(\mathbf{r},\theta)+\mathbf{v}^*\cdot N_{b,y}^{\beta}(\mathbf{r},\theta)\right)J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta\\
&=\tau_1\rho\sum_{i=1}^{2}J^{st}(\tilde{\theta}_i)W_i\int_{\widehat{Q}_n^{\triangle}}\left[\left(V_r(\mathbf{r},\tilde{\theta}_i)N_{a,r}+V_s(\mathbf{r},\tilde{\theta}_i)N_{a,s}\right)T^{\alpha}(\tilde{\theta}_i)+\frac{(-1)^{\alpha}}{\Delta t}N_a(\mathbf{r})\right]\\
&\quad\left[\sum_{\gamma=1}^{2}\sum_{c=1}^{3}u_c^{*\gamma}N_c(\mathbf{r})T^{\gamma}(\tilde{\theta}_i)N_{b,x}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)+\sum_{\gamma=1}^{2}\sum_{c=1}^{3}v_c^{*\gamma}N_c(\mathbf{r})T^{\gamma}(\tilde{\theta}_i)N_{b,y}(\tilde{\theta}_i)T^{\beta}(\tilde{\theta}_i)\right]\mathrm{d}\mathbf{r}
\end{aligned}
$$

$S1b_5^e(a,b) = S1b_1^e(a,b)$

c. stab1c

$$\tau_1\rho\int_{Q_n^e}\mathbf{u}^*\cdot\nabla\mathbf{w}\frac{\partial\Delta\mathbf{u}}{\partial t}\mathrm{d}Q \tag{3.63}$$

$$\begin{aligned}
&S1c_1^e(a,b)\\
&=\tau_1\rho\int_{Q_n^e}\bar{N}_{b,t}^{\beta}(\mathbf{x},t)\mathbf{u}^*\cdot\nabla\bar{N}_a^{\alpha}(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t\\
&=\tau_1\rho\int_{\widehat{Q}_n^e}\bar{N}_{b,t}^{\beta}(\mathbf{x}(\mathbf{r},\theta),t(\theta))\bar{N}_{a,t}^{\alpha}(\mathbf{x}(\mathbf{r},\theta),t(\theta))J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta\\
&=\tau_1\rho\int_{\widehat{Q}_n^e}N_{b,t}^{\beta}(\mathbf{r},\theta)\left(u^*\cdot N_{a,x}^{\alpha}(\mathbf{r},\theta)+v^*\cdot N_{a,y}^{\alpha}(\mathbf{r},\theta)\right)J^{st}(\theta)\ \mathrm{d}\mathbf{r}\ \mathrm{d}\theta\\
&=\tau_1\rho\sum_{i=1}^{2}J^{st}(\tilde{\theta}_i)W_i\int_{\widehat{Q}_n^{\triangle}}\left[\left(V_r(\mathbf{r},\tilde{\theta}_i)N_{b,r}+V_s(\mathbf{r},\tilde{\theta}_i)N_{b,s}\right)T^{\beta}(\tilde{\theta}_i)+\frac{(-1)^{\beta}}{\Delta t}N_b(\mathbf{r})\right]\\
&\left[\sum_{\gamma=1}^{2}\sum_{c=1}^{3}u_c^{*\gamma}N_c(\mathbf{r})T^{\gamma}(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)+\sum_{\gamma=1}^{2}\sum_{c=1}^{3}v_c^{*\gamma}N_c(\mathbf{r})T^{\gamma}(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)T^{\alpha}(\tilde{\theta}_i)\right]\mathrm{d}\mathbf{r}
\end{aligned}$$

$S1c_5^e(a,b) = S1c_1^e(a,b)$

d. stab1d

$$\tau_1\rho\int_{Q_n^e}\left(\mathbf{u}^*\cdot\nabla\mathbf{w}\right)\left(\mathbf{u}^*\cdot\nabla\Delta\mathbf{u}\right)\mathrm{d}Q \tag{3.64}$$

$$
\begin{aligned}
&S1d_1^e(a,b)\\
&= \tau_1\rho \int_{Q_n^e} \left(\mathbf{u}^* \cdot \nabla \bar{N}_a^\alpha(\mathbf{x},t)\right)\left(\mathbf{u}^* \cdot \nabla \bar{N}_b^\beta(\mathbf{x},t)\right)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\\
&= \tau_1\rho \int_{Q_n^e} \left(u^* \bar{N}_{a,x}^\alpha(\mathbf{x},t) + v^* \bar{N}_{a,y}^\alpha(\mathbf{x},t)\right)\left(u^* \bar{N}_{b,x}^\beta(\mathbf{x},t) + v^* \bar{N}_{b,y}^\beta(\mathbf{x},t)\right)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\\
&= \tau_1\rho \int_{\widehat{Q}_n^e} \left(u^* N_{a,x}(\theta) T^\alpha(\theta) + v^* N_{a,y}(\theta) T^\alpha(\theta)\right)\\
&\quad \left(u^* N_{b,x}(\theta) T^\beta(\theta) + v^* N_{b,y}(\theta) T^\beta(\theta)\right) J^{st}(\theta)\, \mathrm{d}\mathbf{r}\, \mathrm{d}\theta\\
&= \tau_1\rho \sum_{i=1}^{2} T^\alpha(\tilde{\theta}_i) T^\beta(\tilde{\theta}_i) J^{st}(\tilde{\theta}_i) W_i\\
&\quad \int_{\widehat{Q}_n^\triangle} \left(u^* N_{a,x}(\theta) + v^* N_{a,y}(\theta)\right)\left(u^* N_{b,x}(\theta) + v^* N_{b,y}(\theta)\right)\, \mathrm{d}\mathbf{r}
\end{aligned}
$$

$$S1d_5^e(a,b) = S1d_1^e(a,b)$$

Stab1 derivative term ($SS1(\mathbf{U}^*)$) can be written as,

$$\tau_1\rho \int_{Q_n^e} \left(\frac{\partial \mathbf{w}}{\partial t} + \mathbf{u}^* \cdot \nabla \mathbf{w}\right)\left(\Delta \mathbf{u}^* \cdot \nabla \mathbf{u}^*\right)\, \mathrm{d}Q \tag{3.65}$$

or,

$$SS1_1^e(a,b) = \tau_1\rho \int_{Q_n^e} \left(\frac{\partial w}{\partial t} + u^* w_{1,x} + v^* w_{1,y}\right) \Delta u\; u_{,x}^*\, \mathrm{d}Q$$

$$SS1_2^e(a,b) = \tau_1\rho \int_{Q_n^e} \left(\frac{\partial w}{\partial t} + u^* w_{1,x} + v^* w_{1,y}\right) \Delta v\; u_{,y}^*\, \mathrm{d}Q$$

$$SS1_4^e(a,b) = \tau_1\rho \int_{Q_n^e} \left(\frac{\partial w}{\partial t} + u^* w_{1,x} + v^* w_{1,y}\right) \Delta u\; v_{,x}^*\, \mathrm{d}Q$$

$$SS1_5^e(a,b) = \tau_1\rho \int_{Q_n^e} \left(\frac{\partial w}{\partial t} + u^* w_{1,x} + v^* w_{1,y}\right) \Delta v\; v_{,y}\, \mathrm{d}Q$$

with

$$
\begin{aligned}
&SS1_1^e(a,b)\\
&= \tau_1\rho \int_{Q_n^e} \left(\bar{N}_{a,t}^{\alpha}(\mathbf{x},t) + u^* \bar{N}_{a,x}^{\alpha}(\mathbf{x},t) + v^* \bar{N}_{a,y}^{\alpha}(\mathbf{x},t)\right) \bar{N}_b^{\beta}(\mathbf{x},t)\\
&\quad \sum_{c=1}^{3} (u_c^{*1} T^1 + u_c^{*2} T^2) \bar{N}_{c,x}(\theta)\, \mathrm{d}Q\\
&= \tau_1\rho \int_{\widehat{Q}_n^e} \left(N_{a,t}(\mathbf{r},\theta) T^{\alpha}(\theta) + u^* N_{a,x}(\theta) T^{\alpha}(\theta) + v^* N_{a,y}(\theta) T^{\alpha}(\theta)\right) N_b(\mathbf{r}) T^{\beta}(\theta)\\
&\quad \sum_{c=1}^{3} (u_c^{*1} T^1 + u_c^{*2} T^2) N_{c,x}(\theta) J^{st}(\theta)\, d\mathbf{r} d\theta
\end{aligned}
$$

$$
\begin{aligned}
&SS1_2^e(a,b)\\
&= \tau_1\rho \int_{Q_n^e} \left(\bar{N}_{a,t}^{\alpha}(\mathbf{x},t) + u^* \bar{N}_{a,x}^{\alpha}(\mathbf{x},t) + v^* \bar{N}_{a,y}^{\alpha}(\mathbf{x},t)\right) \bar{N}_b^{\beta}(\mathbf{x},t)\\
&\quad \sum_{c=1}^{3} (u_c^{*1} T^1 + u_c^{*2} T^2) \bar{N}_{c,y}(\theta)\, \mathrm{d}Q\\
&= \tau_1\rho \int_{\widehat{Q}_n^e} \left(N_{a,t}(\mathbf{r},\theta) T^{\alpha}(\theta) + u^* N_{a,x}(\theta) T^{\alpha}(\theta) + v^* N_{a,y}(\theta) T^{\alpha}(\theta)\right) N_b(\mathbf{r}) T^{\beta}(\theta)\\
&\quad \sum_{c=1}^{3} (u_c^{*1} T^1 + u_c^{*2} T^2) N_{c,y}(\theta) J^{st}(\theta)\, d\mathbf{r} d\theta
\end{aligned}
$$

$$
\begin{aligned}
&SS1_4^e(a,b)\\
&= \tau_1\rho \int_{Q_n^e} \left(\bar{N}_{a,t}^{\alpha}(\mathbf{x},t) + u^* \bar{N}_{a,x}^{\alpha}(\mathbf{x},t) + v^* \bar{N}_{a,y}^{\alpha}(\mathbf{x},t)\right) \bar{N}_b^{\beta}(\mathbf{x},t)\\
&\quad \sum_{c=1}^{3} (v_c^{*1} T^1 + v_c^{*2} T^2) \bar{N}_{c,x}(\theta)\, \mathrm{d}Q\\
&= \tau_1\rho \int_{\widehat{Q}_n^e} \left(N_{a,t}(\mathbf{r},\theta) T^{\alpha}(\theta) + u^* N_{a,x}(\theta) T^{\alpha}(\theta) + v^* N_{a,y}(\theta) T^{\alpha}(\theta)\right) N_b(\mathbf{r}) T^{\beta}(\theta)\\
&\quad \sum_{c=1}^{3} (v_c^{*1} T^1 + v_c^{*2} T^2) N_{c,x}(\theta) J^{st}(\theta)\, d\mathbf{r} d\theta
\end{aligned}
$$

$$
\begin{aligned}
&SS1_5^e(a,b)\\
&= \tau_1\rho \int_{Q_n^e} \left(\bar{N}_{a,t}^{\alpha}(\mathbf{x},t) + u^* \bar{N}_{a,x}^{\alpha}(\mathbf{x},t) + v^* \bar{N}_{a,y}^{\alpha}(\mathbf{x},t)\right) \bar{N}_b^{\beta}(\mathbf{x},t)\\
&\quad \sum_{c=1}^{3} (v_c^{*1} T^1 + v_c^{*2} T^2) \bar{N}_{c,y}(\theta)\, \mathrm{d}Q\\
&= \tau_1\rho \int_{\widehat{Q}_n^e} \left(N_{a,t}(\mathbf{r},\theta) T^{\alpha}(\theta) + u^* N_{a,x}(\theta) T^{\alpha}(\theta) + v^* N_{a,y}(\theta) T^{\alpha}(\theta)\right) N_b(\mathbf{r}) T^{\beta}(\theta)\\
&\quad \sum_{c=1}^{3} (v_c^{*1} T^1 + v_c^{*2} T^2) N_{c,y}(\theta) J^{st}(\theta)\, d\mathbf{r} d\theta
\end{aligned}
$$

There is another term in the right-hand side $\mathbf{F}$ (3.38) which is changing for each Newton iteration, since it depend on $\mathbf{u}^*$,

$$
\int_{Q_n^e} \tau_1\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^* \cdot \nabla \mathbf{w}\right) \mathbf{f}\, \mathrm{d}Q \tag{3.66}
$$

$$
\begin{aligned}
&F_{SS1_1}^e(a)\\
&= \tau_1\rho \int_{Q_n^e} \left[\bar{N}_{a,t}^{\alpha}(\mathbf{x},t) + \mathbf{u}^* \nabla \bar{N}_a^{\alpha}(\mathbf{x},t)\right] f_x\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\\
&= \frac{\tau_1\rho f_x}{6} \sum_{i=1}^{2} \left[T^{\alpha}(\tilde{\theta}_i) \left(-\sum \tfrac{\Delta x_c}{2} y_s(\tilde{\theta}_i) + \sum \tfrac{\Delta y_c}{2} x_s(\tilde{\theta}_i)\right) N_{a,r}\right]\\
&\quad + \frac{\tau_1\rho f_x}{6} \sum_{i=1}^{2} \left[T^{\alpha}(\tilde{\theta}_i) \left(\sum \tfrac{\Delta x_c}{2} y_r(\tilde{\theta}_i) - \sum \tfrac{\Delta y_c}{2} x_r(\tilde{\theta}_i)\right) N_{a,s} + \Upsilon(\tilde{\theta}_i) \frac{(-1)^{\alpha}}{2}\right]\\
&\quad + \frac{\tau_1\rho f_x \Delta t}{12} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{a,x}(\tilde{\theta}_i) \textstyle\sum_{c=1}^{3} (u_c^{*1} T^1(\tilde{\theta}_i) + u_c^{*2} T^2(\tilde{\theta}_i))\\
&\quad + \frac{\tau_1\rho f_x \Delta t}{12} \sum_{i=1}^{2} \Upsilon(\tilde{\theta}_i) T^{\alpha}(\tilde{\theta}_i) N_{a,y}(\tilde{\theta}_i) \textstyle\sum_{c=1}^{3} (v_c^{*1} T^1(\tilde{\theta}_i) + v_c^{*2} T^2(\tilde{\theta}_i))
\end{aligned}
$$

Similarly for $F^e_{SS1_2}(a)$,

$$
\begin{aligned}
&F^e_{SS1_2}(a)\\
&= \tau_1\rho \int_{Q^e_n} \left[\bar{N}^\alpha_{a,t}(\mathbf{x},t) + \mathbf{u}^*\nabla\bar{N}^\alpha_a(\mathbf{x},t)\right] f_y \,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\
&= \frac{\tau_1\rho f_y}{6}\sum_{i=1}^{2}\left[T^\alpha(\tilde{\theta}_i)\left(-\sum\tfrac{\Delta x_c}{2}y_s(\tilde{\theta}_i)+\sum\tfrac{\Delta y_c}{2}x_s(\tilde{\theta}_i)\right)N_{a,r}\right]\\
&\quad+\frac{\tau_1\rho f_y}{6}\sum_{i=1}^{2}\left[T^\alpha(\tilde{\theta}_i)\left(\sum\tfrac{\Delta x_c}{2}y_r(\tilde{\theta}_i)-\sum\tfrac{\Delta y_c}{2}x_r(\tilde{\theta}_i)\right)N_{a,s}+\Upsilon(\tilde{\theta}_i)\frac{(-1)^\alpha}{2}\right]\\
&\quad+\frac{\tau_1\rho f_y\Delta t}{12}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)N_{a,x}(\tilde{\theta}_i)\textstyle\sum_{c=1}^{3}(u_c^{*1}T^1(\tilde{\theta}_i)+u_c^{*2}T^2(\tilde{\theta}_i))\\
&\quad+\frac{\tau_1\rho f_y\Delta t}{12}\sum_{i=1}^{2}\Upsilon(\tilde{\theta}_i)T^\alpha(\tilde{\theta}_i)N_{a,y}(\tilde{\theta}_i)\textstyle\sum_{c=1}^{3}(v_c^{*1}T^1(\tilde{\theta}_i)+v_c^{*2}T^2(\tilde{\theta}_i))
\end{aligned}
$$

3.2.7 Appendix: The Core of Stabilization

Solving differential equations using numerical methods often encounters some severe problems such as oscillation, singular matrix, etc. In such cases, stabilization is needed to get satisfactory result. Some of the circumstances in which such problems occur are convection-dominated problems or violation of the Babuska-Brezzi condition, which, in mixed formulation such as Navier-Stokes equation, may be caused by using equal order of basis functions for velocity and pressure. The underlying idea about the stabilization can be found in [32]. In this appendix, we discuss briefly about the core idea of stabilization for time-dependent advection-diffusion 1D problem in usual finite element method (not space-time finite element method). For more details, we refer to [47].

In order to understand the core idea of stabilization, we use simple equation, i.e., time-dependent advection-diffusion equation 1D,

$$\frac{\partial\phi}{\partial t} + u\frac{\partial\phi}{\partial x} - \nu\frac{\partial^2\phi}{\partial x^2} = f.$$

Assume that we have some suitably-defined finite-dimensional function spaces for trial function S^h and test function V^h. Weak formulation for time-dependent advection-diffusion equation 1D can be written as: find $\phi^h \in S^h$

such that $\forall w^h \in V^h$:

$$\int_\Omega w^h \frac{\partial \phi^h}{\partial t} d\Omega + \underbrace{\int_\Omega w^h u^h \frac{\partial \phi^h}{\partial x} d\Omega}_{\text{Adv}} + \underbrace{\frac{\partial w^h}{\partial x} \nu \frac{\partial \phi^h}{\partial x} d\Omega}_{\text{Diff}} = \int_\Gamma w^h h^h + \int_\Omega w^h f^h.$$

We compare the convection/advection term with diffusion term (in parametrical domain) to know whether the problem convection-dominated,

$$\frac{\text{Adv}}{\text{Diff}} = \frac{\int_{-1}^{1} N_a u^h \frac{2}{h} \frac{\partial N_b}{\partial \xi} \Upsilon d\xi}{\nu \int_{-1}^{1} \frac{2}{h} \frac{\partial N_a}{\partial \xi} \nu \frac{2}{h} \frac{\partial N_b}{\partial \xi} \Upsilon d\xi} = \frac{u^h \int_{-1}^{1} N_a \frac{\partial N_b}{\partial \xi} d\xi}{\nu \frac{2}{h} \int_{-1}^{1} \frac{\partial N_a}{\partial \xi} \frac{\partial N_b}{\partial \xi} d\xi}.$$

In this case, $\Upsilon = \frac{h}{2}$ is the Jacobian of the transformation from physical domain to parametrical domain. The integrand of these two integrals have no dimension. So, to know which term is dominant, we compare only the dimension part of the formulation,

$$\frac{u^h}{\frac{2}{h}\nu} = \frac{u^h}{2\nu}, \tag{3.67}$$

which represents the element Peclet number Pe_h or element Reynold numbers Re_h in Navier-Stokes equation. The problem is convection-dominated if $Pe_h \gg 1$ and non convection-dominated if $Pe_h \approx 1$. Note that the criterion based on the element Peclet number (element Reynold numbers) and not based on the global Peclet number (global Reynold numbers). It is closely related to the element length h in the formulation. We can make the mesh finer and finer (very small h) such that the stabilization is not necessary. However, very small h is numerically impractical.

Consider the stabilized formulation of time-dependent advection-diffusion equation 1D,

$$\int_\Omega w^h \frac{\partial \phi^h}{\partial t} d\Omega + \underbrace{\int_\Omega w^h u^h \frac{\partial \phi^h}{\partial x} d\Omega}_{\text{Adv}} + \underbrace{\frac{\partial w^h}{\partial x} \nu \frac{\partial \phi^h}{\partial x} d\Omega}_{\text{Diff}} +$$

$$\underbrace{\sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau u^h \frac{\partial w^h}{\partial x} \left(\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} - \nu \frac{\partial^2 \phi}{\partial x^2} - f \right) d\Omega}_{\text{Stab}} = \int_{\Gamma_h} w^h h^h d\Gamma + f \int_\Omega w^h f d\Omega.$$

Note that the formulation is still consistent since the stabilization term is residual based formulation. By comparing "Adv", "Diff", and "Stab" terms, we can get some useful information,

- If we compare "Diff" and "Stab", we get the dimension of τ. We obtain that $\tau(u^h)^2$ has the same dimension with ν. It means we can consider $\tau(u^h)^2$ as numerical viscosity $\tilde{\nu}$ and stabilization can be viewed as adding the numerical viscosity. We have to add the numerical viscosity as needed, otherwise, we can lost the accuracy of the solution.

- If we compare "Adv" and "Diff", we get element Peclet number Pe_h as in (3.67).

$$Pe_h = \frac{u^h h}{2\nu},$$

- If we compare "Adv" and "Stab", we get numerical element Peclet number,

$$\tilde{Pe}_h = \frac{u^h h}{2\tilde{\nu}} = \frac{u^h h}{2\tau(u^h)^2} = \frac{h}{2u^h}\frac{1}{\tau}.$$

From this information, we can avoid the numerical difficulties caused by convection-dominated problem (assume $Pe_h \gg 1$) by setting the stabilization parameter τ with

$$\begin{aligned} \tilde{Pe}_h &\approx 1, \\ \frac{h}{2u^h}\frac{1}{\tau} &\approx 1, \\ \tau &= \frac{h}{2u^h}. \end{aligned} \tag{3.68}$$

Note that (3.68) is one of the several selections that we take such that we add stabilization by adding the numerical viscosity as needed. In the higher dimensional case, we can represent (3.68) as

$$\tau = \frac{h}{2||\mathbf{u}^h||},$$

where $||\mathbf{u}^h||$ is the magnitude of $\mathbf{u}^h$.

3.2.8 Appendix: Stabilization in DSD/SST-SUPS formulation

In this appendix, we discuss briefly about the stabilization that we use in DSD/SST-SUPS formulation in (3.35) (refer to [47] for more details).

The formula and the role for each stabilization term are as follows,

- Streamline-Upwind/Petrov-Galerkin (SUPG)

$$\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_1 \left[\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h\right] \cdot \underbrace{\left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h\right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\right]}_{\text{residual}} \mathrm{d}Q$$

  To avoid numerical instability that is caused by dominating convection.

- Pressure Stabilizing/Petrov-Galerkin (PSPG)

$$\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \frac{1}{\rho} \tau_2 \nabla q^h \underbrace{\left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h\right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\right]}_{\text{residual}} \mathrm{d}Q$$

  To avoid pressure oscillations that are generated due to using equal order of basis functions for velocity and pressure with the purpose of simplifying the implementation. Moreover, without PSPG, we have zero block matrix K_9 in (3.36), which leads to an ill-conditioned matrix.

- Least Square on Incompressible Constraint (LSIC)

$$\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_3 (\nabla \cdot \mathbf{w}^h) \rho \underbrace{(\nabla \cdot \mathbf{u}^h)}_{\text{residual}} \mathrm{d}Q$$

  To reduce the divergence error that can occur in low or high Reynold numbers. To handle problems with very high Reynold numbers, beside LSIC, we also need additional stabilization terms as in DSD/SST-VMST (DSD version with the variational multiscale turbulence model).

Note that when adding the stabilizations to the formulation, we have to make sure that the resulting formulation is still consistent, i.e., the exact solution still satisfies the resulting formulation. This is one reason we use a formula with the residual as the factor, i.e., momentum equation is used as a factor in SUPG and PSPG while continuity equation is used as a factor in LSIC.

We briefly explain the calculation of the stabilization parameters τ. In the stabilized formulation, the appropriate stabilization parameter τ plays an important role in the accuracy of the formulation [14]. In the DSD/SST formulation (3.35), we use three stabilization parameters τ_1, τ_2, τ_3 which are $\tau_{SUPG}, \tau_{PSPG}, \tau_{LSIC}$, respectively. The definition of each stabilization parameter can be found in [14]. Here, we show the way to compute it in the S-DSD/SST context.

- Stabilization parameter τ_1 or τ_{SUPG}

$$\tau_1 = \left(\frac{1}{\tau_{SUGN12}^2} + \frac{1}{\tau_{SUGN3}^2} \right)^{-\frac{1}{2}}, \tag{3.69}$$

where

$$\begin{aligned}
&\tau_{SUGN12} \\
&= \left(\sum_{\alpha=1}^{2} \sum_{a=1}^{3} \left| \frac{\partial \bar{N}_a^\alpha(\mathbf{x},t)}{\partial t} + \mathbf{u}^h \cdot \nabla \bar{N}_a^\alpha(\mathbf{x},t) \right| \right)^{-1} \\
&= \left(\sum_{\alpha=1}^{2} \sum_{a=1}^{3} \left| \frac{\partial \bar{N}_a^\alpha(\mathbf{x},t)}{\partial t} + u \bar{N}_{a,x}^\alpha(\mathbf{x},t) + v \bar{N}_{a,y}^\alpha(\mathbf{x},t) \right| \right)^{-1} \\
&= \left(\sum_{\alpha=1}^{2} \sum_{a=1}^{3} \left| \frac{\partial \bar{N}_a^\alpha(\mathbf{x}(\mathbf{r},\theta),t(\theta))}{\partial t} + u N_{a,x}(\theta) T^\alpha(\theta) + v N_{a,y}(\theta) T^\alpha(\theta) \right| \right)^{-1}
\end{aligned}$$

with

$$u = \sum_{c=1}^{3} \left(u_c^{*1} T^1(\theta) + u_c^{*2} T^2(\theta) \right) N_c(\mathbf{r}),$$

$$v = \sum_{c=1}^{3} \left(v_c^{*1} T^1(\theta) + v_c^{*2} T^2(\theta) \right) N_c(\mathbf{r}),$$

and $\bar{N}_{a,t}^\alpha(\mathbf{x},t), \bar{N}_{a,x}^\alpha(\mathbf{x},t), \bar{N}_{a,y}^\alpha(\mathbf{x},t)$ as in (3.43).

$$\tau_{SUGN3} = \frac{h_{RGN}^2}{4\nu}$$

with

$$h_{RGN} = 2\left(\sum_{\alpha=1}^{2}\sum_{a=1}^{3}\left|\mathbf{z}\cdot\nabla\bar{N}_a^{\alpha}(\mathbf{x},t)\right|\right)^{-1}$$
$$= 2\left(\sum_{\alpha=1}^{2}\sum_{a=1}^{3}\left|z_1\bar{N}_{a,x}^{\alpha}(\mathbf{x},t)+z_2\bar{N}_{a,y}^{\alpha}(\mathbf{x},t)\right|\right)^{-1}$$
$$= 2\left(\sum_{\alpha=1}^{2}\sum_{a=1}^{3}\left|z_1N_{a,x}(\theta)T^{\alpha}(\theta)+z_2N_{a,y}(\theta)T^{\alpha}(\theta)\right|\right)^{-1}$$

and the solution gradient unit vector is defined as

$$\mathbf{z} = \frac{\nabla \parallel \mathbf{u} \parallel}{\parallel \nabla \parallel \mathbf{u} \parallel\parallel}$$

Note that this quantity is not constant. Practically, we can evaluate it either using the center of each element or at the quadrature points. In our simulation, we use the first approach.

- Stabilization parameter τ_2 or τ_{PSPG}

$$\tau_2 = \tau_1 \tag{3.70}$$

- Stabilization parameter τ_3 or τ_{LSIC}

$$\tau_3 = \tau_1 \parallel \mathbf{u} \parallel^2 \tag{3.71}$$

This quantity is also not constant, we evaluate it using the center of each element.

3.2.9 Appendix: GMRES

GMRES is a projection method computing approximate solution x_n to the system $Ax = b$ as minimizers of the residual norm

$$\|r_n\| = \|b - Ax_n\|,$$

in the Krylov subspace

$$K_n = \text{span}\,[r_0, Ar_0, A^2r_0, ..., A^{n-1}r_0],$$

upon finding an orthonormal basis $\{q_1, q_2, ..., q_n\}$ of K_n and denoting by $\tilde{Q_n}$ the matrix with columns $q_1, q_2, ..., q_n$. The above minimization is equivalent to the minimization of $\|\tilde{H}_n y - \|r_0\|e_0\|$ where $e_0 = (1, 0, ..., 0)$ and $\tilde{H}_n$ is the Hessenberg matrix satisfying the similarity transformation

$$A\tilde{Q}_n = \tilde{Q}_{n+1}\tilde{H}_n,$$

and x_n is obtained from $x_n = x_0 + \tilde{Q}_n y$.

Note that the corresponding problem in our case is to approximate solution $\Delta\mathbf{U}^l$ to the system (3.37) as minimizers of the residual norm

$$\|r^l\| = \|E^l - \mathbf{J}^l \Delta\mathbf{U}^l\|,$$

in the Krylov subspace

$$K_l = \text{span } [r_0, \mathbf{J}r_0, \mathbf{J}^2 r_0, ..., \mathbf{J}^{l-1} r_0],$$

where E^l is the right-hand side of (3.37).

The outline of the GMRES algorithm can be written as follows [35]

Algorithm : GMRES
Given initial value x_0, we have initial residual is $r_0 = b - Ax_0$.
1 $q_1 = \frac{r_0}{\|r_0\|}$
2 for $j = 1, 2, ..., m$
3 compute $v_j = Aq_j$
4 for $i = 1, ..., j$
5 $h_{i,j} = (v_j, q_i)$
6 $v_j = v_j - h_{i,j} q_i$
7 end
8 compute $h_{j+1,j} = \|v_j\|_2$ and $v_{j+1} = \frac{v_j}{h_{j+1,j}}$
9 end
10 define $\tilde{Q}_m := [q_1, ..., q_m], \tilde{H}_m = \{h_{ij}\}_{1 \le i \le m+1; 1 \le j \le m}$
11 compute y_m that minimizes $\|\tilde{H}_m y - \|r_0\|e_0\|$ and $x_m = x_0 + \tilde{Q}_m y_m$

Step 2-9 of the algorithm is well-known with Arnoldi Iteration (modified Gram-Schmidt). Arnoldi iteration as orthogonal projection onto Krylov subspaces is an algorithm for building an orthogonal basis of the Krylov subspace K_n. This algorithm is based on the similarity transformation

$$A\tilde{Q} = \tilde{Q}\tilde{H},$$

where $\tilde{Q}$ is an orthogonal matrix and $\tilde{H}$ is Hessenberg matrix that results-from Arnoldi iteration. Let $A, \tilde{Q}, \tilde{H}$ be $m \times m$ matrices.
We can consider only a part of the system by taking $n < m$, ,

$$A_{(m\times m)}\tilde{Q}_{(m\times n)} = \tilde{Q}_{(m\times(n+1))}\tilde{H}_{((n+1)\times n)},$$

In order to solve the problem more efficiently, instead of using A we can use $\tilde{H}_n$ which is smaller but similar matrix due to satisfy similarity transformation, i.e., A and $\tilde{H}_n$ have the same eigenvalues. Moreover, the efficiency of QR factorization in step 4 increases considerably if we first put A into Hessenberg form $\tilde{H}$ (almost triangular matrix).

In step 11, we find y that minimize $\|\tilde{H}_n y - \|r_0\|e_0\|$ using QR factorization. Note that $\tilde{H}_n$ is $(n+1)\times n$ matrix, hence it gives an over-constrained linier system of $(n+1)$ equations for n unknowns. In this case, we can compute the minimum by solving the least square problem. One of the methods that can be used is QR factorization : find an orthogonal $(n+1)\times(n+1)$ matrix Q and $(n+1)\times n$ matrix R such that

$$\tilde{H}_n = QR. \tag{3.72}$$

Using (3.72), we get

$$\begin{aligned}\|\tilde{H}_n y - \|r_0\|e_0\| &= \|QRy - \|r_0\|e_0\| \\ &= \|Ry - Q^{-1}\|r_0\|e_0\| \\ &= \|Ry - \beta\|,\end{aligned}$$

and the minimization in step 12 of GMRES algorithm becomes: Find y that minimizes $\|Ry - \beta\|$, where $\beta = Q^{-1}\|r_0\|e_0$

We use the Householder reflectors H to obtain the QR factorization of $\tilde{H}_n$. The idea is to successively introduce zeros starting from the first column of $\tilde{H}_n$ and continue in this way until $\tilde{H}_n$ becomes upper triangular R. For each vector x (column of $\tilde{H}_n$), we construct householder matrix

$$H = I - \frac{2zz^T}{z^T z},$$

which is a symmetric orthogonal matrix where $Hx = w$ and $z = w - x$ [37]. Note that the triangular matrix R has one more row than its columns, so its bottom row consists of zero.

Using Householder reflector, $(n+1)\times n$ matrix $\tilde{H}_n$ can be factorized as

$$\tilde{H}_n = QR$$

where

$R = H_n H_{n-1} ... H_1 \tilde{H}_n$
$Q = (H_n H_{n-1} ... H_1)^{-1} = H_1^{-1} ... H_{n-1}^{-1} H_n^{-1} = H_1 ... H_{n-1} H_n$
note that $H_i^{-1} = H_i$ because H is a symmetric orthogonal matrix

Additional notes concerning the practical implementation of GMRES:

- The computational cost of GMRES algorithm depends on the cost of the method used for the least square problem and Arnoldi iteration.

- Practically, norm of the residual r_n is used as a stopping criterion. However, in order to save the computational cost, we can use another stopping criterion equivalent to the residual norm which is the $(n+1)st$ component of the right-hand side obtained by premultiplying $|r_0|e_0|$ by the n successive multiplication of Householder reflector [36],

  $$\|r_n\| \approx |\beta_{n+1}| = |Q_{n+1}^{-1}\|r_0\|e_0|.$$

  This stopping criterion enables us to obtain the residual norm of the approximation solution without computing x_n explicitly, thus we avoid unnecessary operations.

- In practice, GMRES is usually applied in collaboration with a preconditioner. The choice of preconditioner has a great importance for the efficiency of GMRES. Since we already use stabilization in the formulation, the condition of the matrix system is not too bad so diagonal preconditioner is expected to be sufficient.

  We use right preconditioned GMRES algorithm which is based on solving

  $$AP^{-1}\bar{x} = b, \bar{x} = Px.$$

  By writing Arnoldi iteration in detail, right-preconditioned GMRES can be written as follows [35]:
  Let x_0 is a given initial value. Then the initial residual is $r_0 = b - Ax_0$.

  Algorithm : Right preconditioned GMRES
  Given initial value x_0, we have initial residual is $r_0 = b - Ax_0$.
  1 $q_1 = \frac{r_0}{\|r_0\|}$
  2 for $j = 1, 2, ..., m$

3 compute $v_j := AP^{-1}q_j$
4 for $i = 1, ..., j$
5 $h_{i,j} = (v_j, q_i)$
6 $v_j := v_j - h_{i,j}q_i$
7 end
8 compute $h_{j+1,j} = \|v_j\|_2$ and $q_{j+1} = \frac{w_j}{h_{j+1,j}}$
9 end
10 define $\tilde{Q}_m := [q_1, ..., q_m], \tilde{H}_m = \{h_{ij}\}_{1 \le i \le m+1; 1 \le j \le m}$
11. compute y_m that minimizes $\|\tilde{H}_m y - \|r_0\|e_0\|$
and $x_m = x_0 + P^{-1}\tilde{Q}_m y_m$

In this case, the Arnoldi loop builds an orthogonal basis of the right preconditioned Krylov subspace

$$\text{span}[r_0, AP^{-1}r_0, (AP^{-1})^2 r_0, ..., (AP^{-1})^{n-1}r_0]$$

Notice that the residual norm is relative to the initial system, i.e., $b - Ax_m$, since the algorithm obtains the residual $b - Ax_m = b - AP^{-1}\bar{x}_m$ implicitly. Then, we still can use stopping criterion as explained before.

3.2.10 Appendix: Gaussian Quadrature

We briefly review the basics of multidimensional Gaussian quadrature [39] that we used in subsection 3.2.6.

Let us consider integration of $f(\mathbf{x})$ over element domain Ω^e. In this case, we evaluate the integration by transforming variables from physical domain Ω^e to parametric domain $\widehat{\Omega}^e$ as follows,

$$\int_{\Omega^e} f(\mathbf{x})\, d\Omega = \int_{\widehat{\Omega}^e} f(\mathbf{x}(\mathbf{r}))\Upsilon(\mathbf{r})\, d\widehat{\Omega}^e = \int_{\widehat{\Omega}^e} g(\mathbf{r})\, d\widehat{\Omega}^e$$

where Υ is the Jacobian of the transformation. Then, approximate the integral using

$$\int_{\widehat{\Omega}^e} g(\mathbf{r})\, d\widehat{\Omega}^e \approx \sum_{i=1}^{n_{gauss}} g(\tilde{\mathbf{r}}_i) W_i$$

where $\tilde{\mathbf{r}}_i$'s are the quadrature points in parametric domain $\widehat{\Omega}^e$, W_i's are the quadrature weights, n_{gauss} is the number of point integration. The choice in using these parameters is based on the quadrature rule. In our case, we use Gaussian quadrature to compute the integral over temporal domain which is

1D. We take two-point Gaussian quadrature rule in 1D as in Table 3.3. We observed that in computing all of the integral over temporal domain, it is enough to use two-point quadrature which is exact up to cubic polynomial.

Table 3.3: Two-points quadrature rule

n_{gauss}	i	$\tilde{\mathbf{r}}_i$	W_i
2	1	$-\frac{1}{\sqrt{3}}$	1
	2	$\frac{1}{\sqrt{3}}$	1

3.2.11 Remark on Optimization and Acceleration of The Algorithm

As mentioned before, solving fluid model involving moving domain is also a source of computational challenges that we have. Using space-time finite element, the dimension of the problem increases by one and after space-time discretization, we get large nonlinear system of equations which has four times bigger size than usual space discretization, leading to higher computational cost and used memory. We apply some strategies to improve overall computational performance such as exploiting the special features of S-DSD/SST (Special-DSD/SST); using the appropriate iterative method to solve this large nonlinear system; avoiding unnecessary operations as we did in setting stopping criterion for GMRES; and designing a good structure data for the large coupled program so that it is more efficient. However, due to the big complexity of the problem, improvements in implementation of the developed model are needed. Some of these include parallelization of the implementation, compression of the memory, e.g., using compressed row storage (CSR) or using more powerful preconditioner in iterative solution of fluid model. Also, in solving problem with very high Reynolds numbers, a more suitable model can be used, e.g., DSD/SST-VMST, DSD/SST version with the variational multiscale (VMS) turbulence model [39].

3.3 Sequential Coupling

In this section, we address the sequential coupling between interface and fluid model: the coupling algorithm, boundary condition, and elasticity equation

to move the nodal points of fluid mesh as time progresses.

3.3.1 The Algorithm

As mentioned in Section 2.4, the coupling between interface and fluid model is done via pressure acting from the fluid on the interfaces and by the fact that the interfaces determine the domain for fluid motion. Numerically, this is a weak coupling where the interface model and fluid model are solved independently for each time step, as in the algorithm below.
Repeat for $n = 0, 1, ...$

- Run one step of the interface model with current pressure as outer force to get the interface position at t_{n+1}. We minimize the functional

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

  where

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

  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 $\boldsymbol{p}_i, \boldsymbol{p}_j$ are the BMO reference vectors.

- Using the information of interface position at t_n and t_{n+1}, determine the domain occupied by fluid and run one step of the fluid model to obtain velocity and pressure of fluids at t_{n+1}.

The coupling algorithm in flow chart form is shown in Figure 3.6.

Notice that in this problem, we consider the interface as a thin membrane which is composed of the fluid particles, so it has the same physical quantities, e.g., viscosity. Only chemical properties such as adhesion, are different with the fluid particles. Therefore, viscous stress is not used for the coupling, we only use pressure as an outer force that act on the interface. This assumption has to be verified through comparison of numerical results with experimental data.

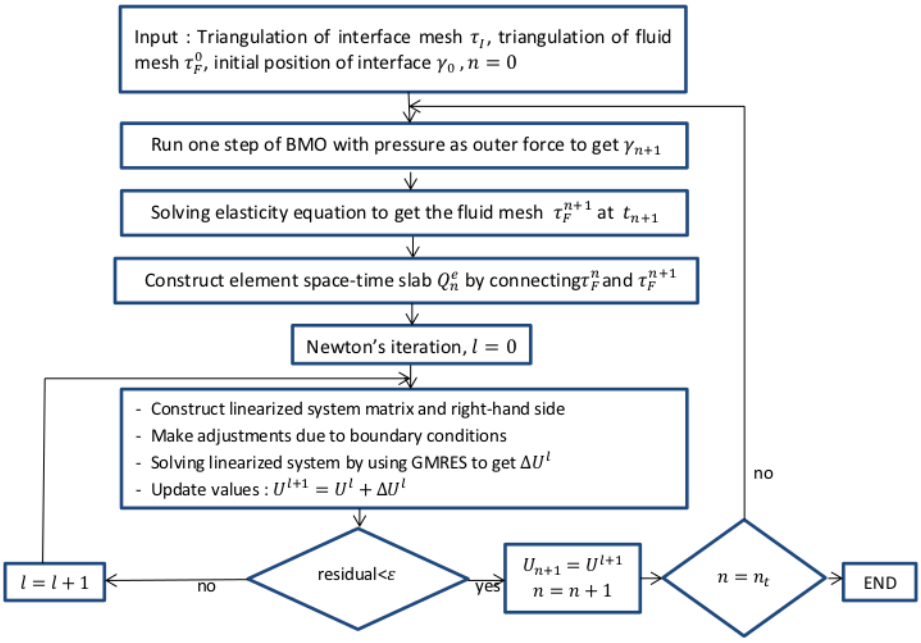


Figure 3.6: Numerical algorithm for coupled model

3.3.2 Boundary Condition

The two models are coupled at the interface through compatibility conditions of kinematics and tractions. We discretize the boundary condition that has been shown in Section 2.4, as follows

- Boundary condition in region D, E and triple line C are Dirichlet boundary conditions. We incorporate this boundary condition by adjusting the left-hand side and right-hand side of the resulting system matrix in (3.36). We adjust left-hand side **K** by setting the diagonal positions with 1 and 0 otherwise, for rows that correspond to velocity and adjust the right-hand side **F** by setting it with the corresponding boundary velocity.

- Boundary condition in region A and B consist of mixed Neumann and Dirichlet boundary condition which is related to the third integral in (3.35). As what we have discussed in Section 2.4, we apply a kind

of slip-boundary condition to avoid the discrepancy between no-slip and free-slip boundary condition at triple line. In discretization level, we need to be careful such that we get a good approximation of the continuum of such slip boundary condition in the discrete problem. Since the bubble interface is a curve, the ambigous nature of the normal vector in the discretized problem also happens here, which can interfere with the application of such boundary conditions [45]. Even if we use consistent normal direction instead of usual normal direction, it does not guarantee a good approximation. Therefore, we adopt a discretization of such boundary conditions in the presence of curved boundaries, as in [45], as follows

$$\begin{aligned}&\int_{(P_n)_h} \tilde{N}_{a_2}\mathbf{h}^h \, dP_n \\ &= \left[\int_{(l_n^i)} \tilde{N}_{a_2}\boldsymbol{\sigma}(\mathbf{u}^h, p^h)\cdot\mathbf{n}_i \, dl_n^i + \int_{(l_n^j)} \tilde{N}_{a_2}\boldsymbol{\sigma}(\mathbf{u}^h, p^h)\cdot\mathbf{n}_i \, dl_n^j\right]\cdot[\mathbf{I} - \mathbf{n}_{a_2}\mathbf{n}_{a_2}],\end{aligned} \tag{3.73}$$

where, as in Figure 3.7, $l_n^i,\ l_n^j$ are the boundary space-time (P_n) edges on space domain; node $a_1,\ a_2,\ a_3$ are boundary nodes; node a_2 adjacent with l_n^i and l_n^j; $\mathbf{n}_i$ and $\mathbf{n}_j$ are normal vectors which are assumed to be uniform along edges l_n^i and l_n^j, respectively. Here, $\tilde{N}_{a_2}$ is the boundary trace of the basis function associated with node a_2, $\mathbf{n}_{a_2}$ is a consistent normal vector at node a_2 [45] defined by

$$\mathbf{n}_{a_2} = \frac{\int_{(l_n^i)} \tilde{N}_{a_2}\ \mathbf{n}_i \, dl_n^i + \int_{(l_n^j)} \tilde{N}_{a_2}\ \mathbf{n}_j \, dl_n^j}{|\int_{(l_n^i)} \tilde{N}_{a_2}\ \mathbf{n}_i \, dl_n^i + \int_{(l_n^j)} \tilde{N}_{a_2}\ \mathbf{n}_j \, dl_n^j|}$$

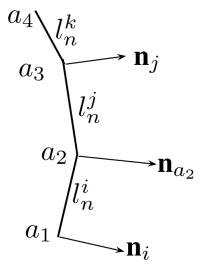


Figure 3.7: normal vectors $\mathbf{n}_i$, $\mathbf{n}_j$ and consisten normal $\mathbf{n}_{a_2}$ at boundary node a_2

In more detail, we can write the third integral in (3.35) as follows,

$$\begin{aligned}
&\int_{(P_n)_h} \tilde{N}^{\alpha}_{a_2}(x,t)\mathbf{h}^h \, dP_n \\
&= \int_{(\widehat{P}_n)_h} \tilde{N}^{\alpha}_{a_2}(x(\xi),t(\theta))\mathbf{h}^h \tilde{J}^{ST} \, d\widehat{P}_n \\
&= \int_{(\widehat{P}_n)_h} \tilde{N}^{\alpha}_{a_2}(\xi,\theta)\mathbf{h}^h \tilde{J}^{ST} \, d\widehat{P}_n \\
&= \int_{(\widehat{P}_n)_h} \tilde{N}_{a_2}(\xi) T^{\alpha}(\theta)\mathbf{h}^h \tilde{J}^{ST} \, d\widehat{P}_n \\
&= \sum_{i=1}^{i=2} T^{\alpha}(\tilde{\theta}_i)\tilde{J}^{ST} W_i \underbrace{\int_{(\bar{P}_n)_h} \tilde{N}^{\alpha}_{a_2}(x,t)\mathbf{h}^h \, d\bar{P}_n}_{(3.73)}
\end{aligned}$$

where $\widehat{P}_n$ is P_n in parametrical domain, $\bar{P}_n$ is one-dimension part of P_n in spatial domain, $\bar{x}$ is coordinate (one-dimension) of $\bar{P}_n$, ξ is $\bar{x}$ in parametrical domain.

Note that the consistent normal directions are the most suitable one for proper mass and momentum conservation [46]. The application of slip boundary condition in this sense still can be distinguished into two: a slip boundary, which is planar and which is curved. In the planar case, it has uniform distribution of the normal vector; while in the curved case, the normal vector varies depending on the location. Hence, in our problem, we consider the curved boundary case such that we get a good approximation of the continuum of such boundary condition in the discrete problem.

3.3.3 Elasticity Equation

In this coupling problem, we have separate mesh for interface and fluid model. The mesh for the interface does not change during simulation but the fluid mesh changes as the interfaces evolves. To move the nodal points of fluid mesh while preserving the good properties of the mesh, we use an automatic mesh moving scheme where the displacement of internal nodes is determined by solving the elasticity equation [39].

Elasticity equation in the continuum setting is formulated as follows: find the displacement of fluid mesh $\mathbf{y} \in S_f$, such that $\forall \mathbf{w} \in V_f$,

$$\int_{\Omega_t} \boldsymbol{\varepsilon}(\mathbf{w}) \cdot \mathbf{D}\boldsymbol{\varepsilon}(\mathbf{y}) \, d\Omega = 0, \tag{3.74}$$

where Ω_t is fluid subdomain at time t; S_f and V_f are the sets of trial and test functions for the fluid domain, respectively. Moreover, $\boldsymbol{\varepsilon}$ is the strain vector, $\mathbf{D}$ is the elasticity tensor given by

$$\mathbf{D} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

with λ and μ are the Lamé parameters. We prescribe boundary condition at the interface γ_3 by

$$\mathbf{y} \cdot \mathbf{n} = \mathbf{y}_I \cdot \mathbf{n}$$

where $\mathbf{y}_I$ is the displacement of interface points.

The elasticity equation is solved in fluid mesh at each time-step (or once in several time-steps) which gives rise to a time-dependent fluid mesh deformation. The size of elements becomes a criterion for the extension of deformation of the fluid mesh, where we stiffen smaller elements more than the larger ones by choosing appropriate elasticity coefficients. We do so by altering the way we account the Jacobian of the transformation from element domain to physical domain in the space element. Practically, we take smaller elements near the interfaces as in Figure 3.8. In this case, the elasticity tensor is mesh-dependent, with the mesh Lamé parameters μ^h and λ^h defined by

$$\mu^h = \frac{E_f^h}{2(1 + z_f)} \tag{3.75}$$

$$\lambda^h = \frac{z_f E_f^h}{(1 + z_f)(1 - 2z_f)} \tag{3.76}$$

where

$$E_f^h = E_f \left(\frac{\Upsilon}{\Upsilon^0} \right)^{-\chi}$$

is the fluid mesh Young's modulus, Υ is the Jacobian (3.42), $\chi > 0$ is a real parameter, Υ^0 is global scaling value which we can take arbitrarily, E_f and z_f are the constant fluid mesh Young's modulus and Poisson's ratio, which are prescribed beforehand. In [39], the value for Poisson's ratio is chosen as $z_f \in [0, 0.5)$. In the numerical example, we also use $z_f = 0.3$ as they used in the simulations.

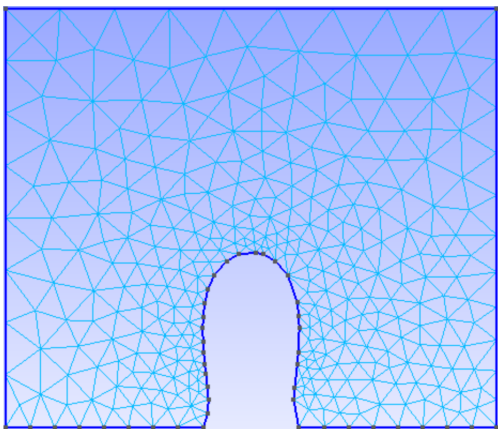


Figure 3.8: Example of fluid mesh around the bubble

The output of elasticity equation is the displacement for each node of fluid mesh. We use this displacement to calculate the new position of each node. Moreover, we also use the displacement at interfaces to get the interface velocity which will be the input for boundary conditions in fluid model.

Chapter 4

Numerical Examples

4.1 Interface Model

We now present some numerical tests and examples of our method. All numerical experiments in this section are conducted on a $[0,1]\times[0,1]$ domain with time step $\Delta t = 0.005$ and DMF partition $N = 30$. Moreover, the phase interior angles at the triple junction are measured using the tangents to the quadratic interpolation of the piecewise linear interface in the neighborhood of the junction.

We begin by investigating the stability of the triple junction in terms of the angle measure and location of the junction. Next, we look at the behavior of the triple junction motion for two cases (with and without axial symmetry). Finally, we present an example which incorporates the volume constraint and check the stability of the junction and preservation of the phase volumes of its stationary solution.

In the experiments below, we use two types of setting as in Table 4.1.

4.1.1 Junction Stability Test

Consider an initial condition where phase P_2 consists of two disjoint square regions on opposite corners of the domain; while the remaining regions above and below the line $y = x$ are taken as phase P_1 and P_3, respectively (refer to Figure 4.1a). In this setting, the interface network remains stationary for a 135°–90°–135° junction angle configuration. We then, run our method under the case 2 setting and check the stability of the triple junction when the maximum dot product (without projection triangle) or projection triangle is used for phase detection. Here, the domain is triangulated into $12,800$ elements ($\Delta x = 0.0125$).

Table 4.1: Numerical parameters for case 1 and case 2

parameters		**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°
coefficients	a	0.881	0.954
	b	0.262	0.127
	c	0.656	0.639
reference vectors	$\mathbf{p}_1$	(-0.8,-0.6)	(-0.777,-0.628)
	$\mathbf{p}_2$	(0,1)	(-0.333,-0.943)
	$\mathbf{p}_3$	(1,0)	(1,0)

The relative error of the stationary junction angle measures are summarized in Table 4.2. We see that when the maximum dot product is used to locate the interface, the stationary interior angle in phase P_2 is of measure $90^\circ \pm 5.11$, while the other two junction angles are approximately $135^\circ \pm 4.10$; thereby, yielding a relative error of at most 5.68%. On the other hand, utilizing the projection triangle in the phase detection scheme reduces the relative error to at most 0.88% with stationary junction angles of measure $90^\circ \pm 0.78$ and $135^\circ \pm 1.19$.

Table 4.2: Relative Error in Junction Angle Measures

phase	maximum dot product		projection triangle	
	junction J_1	junction J_2	junction J_1	junction J_2
P_1	-0.0083	-0.0081	0.0088	0.0077
P_2	-0.0290	-0.0296	-0.0062	-0.0019
P_3	0.0561	0.0568	-0.0039	-0.0087

In addition, both phase detection scheme shifted the junction to a distance of at most 0.0065 for the dot product scheme and at most 0.0057 for the projection triangle. This can be accounted for by the approximation error in the construction of the projection triangle. Hence, our method stably preserves the angle conditions, in this case $135^\circ - 90^\circ - 135^\circ$. This also confirms that using the projection triangle in the phase detection scheme

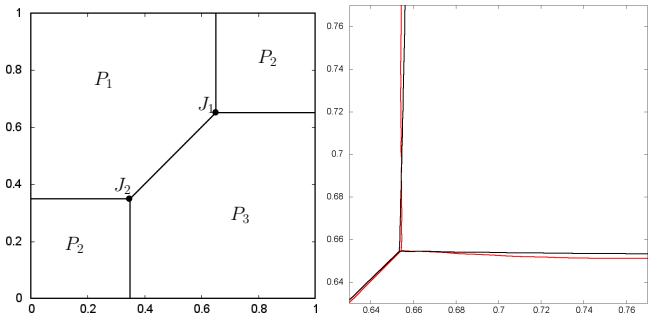


Figure 4.1: (a) Initial condition. (b) The stationary interface network around junction J_1 after time $100\Delta t$ using dot product (red) and projection triangle (black) for phase detection.

correctly detects the interface near the junction, as opposed to using the maximum dot product.

4.1.2 Triple Junction Motion

In this subsection, we start with an initial condition where a T-shaped interface is rotated 90° counterclockwise where the T-junction is at point $(0.25, 0.5)$. We take phase P_2 as the region to the left of line $x = 0.25$, and the remaining top and bottom regions as phases P_1 and P_3, respectively. We triangulate the domain into $20,000$ elements with $\Delta x = 0.01$, and evolve the interface via our method using the projection triangle to determine the different phase regions at each time step. We then investigate the evolution of the triple junction in both cases.

Junction without Axial Symmetry (Case 1)

Under the first case setup, we plot the evolution of the initial T-junction and its underlying interface network at different times in Figure 4.2.

Notice that for the first 10 time steps, the junction angles rapidly adjusts to approximate the $150°-90°-120°$ angle conditions. Thereafter, the triple junction maintains phase interior angles of measure within 2.5% relative error (refer to Figure 4.3). Note that this approximation error in the stationary junction angles can be made smaller by increasing the precision in the construction of the projection triangle.

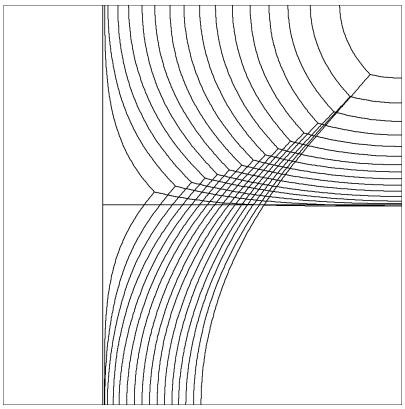


Figure 4.2: Evolution of the triple junction for case 1.

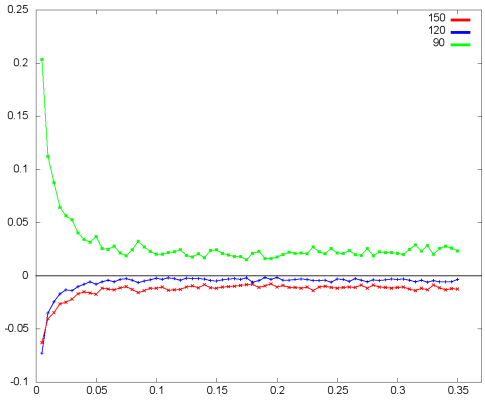


Figure 4.3: Relative error of the junction angles at each time step.

Axially Symmetric Junction (Case 2)

We now look at the behavior of the junction motion when subjected to the second case. Note that since the surface tensions on the $1-2$ and $2-3$ interfaces are equal, that is, $\sigma_1 = \sigma_3 = \frac{1}{\sqrt{2}}$, we expect these interfaces to symmetrically evolve with respect to the horizontal line $y = 0.5$. This is in agreement with our numerical simulation shown in Figure 4.4.

In the first 10 time steps, the triple junction rapidly approaches the $135°-90°-135°$ angle conditions. After which, the interface gradually starts to move horizontally to the right. The transport velocity s of our numerical interface solution approaches $\frac{\pi}{2\sqrt{2}}$, which is the exact velocity of the constantly transported solution of the sharp interface problem, as shown in

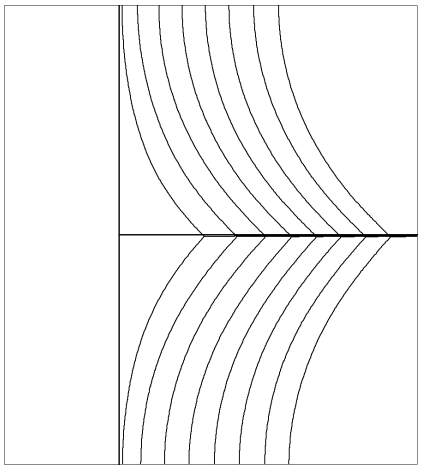


Figure 4.4: Evolution of the triple junction for case 2.

Figure 4.5.

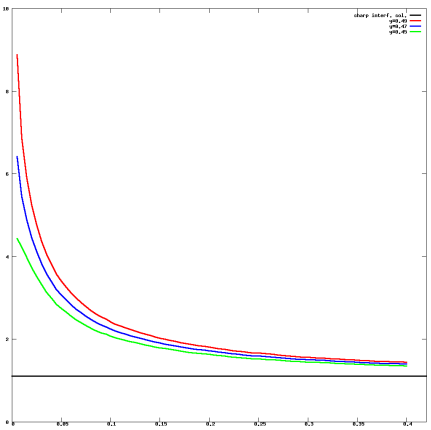


Figure 4.5: Transport velocities of the numerical interface solution at $y = 0.45, 0.47, 0.49$ (colored) vs the constantly transported solution (black).

Moreover, we note that the shape of such a constantly transported profile in the axially symmetric case is determined by:

$$\begin{aligned} v(y) &= -\tfrac{\sigma_1}{s}\log(\cos(\tfrac{s}{\sigma_1}y)) + c \\ &= -\tfrac{2}{\pi}\log(\cos(\tfrac{\pi}{2}y)) + c, \end{aligned}$$

where the constant c of horizontal shift may be chosen appropriately [24].

Comparing this with the numerical interface solution obtained via our method, we see that it is in a good agreement with the exact shape of the profile (Figure 4.6).

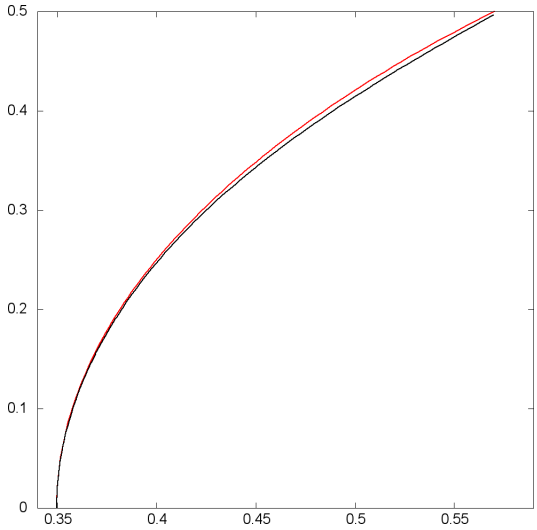


Figure 4.6: The shape of the numerical interface at time $t = 30\Delta t$ (black) vs the constantly transported solution (red).

4.1.3 Volume-preserving $150°-90°-120°$ Double Bubble

Consider a three-phase volume-preserving case where two phases are identical squares with one common side of length 0.28. We take the outside region as phase P_1, and the left and right square regions as phase P_2 and P_3, respectively (refer Figure 4.7.a).

We evolve this configuration via our method under the first case setting. Moreover, we wish to preserve the volume of each phase region employing a penalization technique with parameter $\epsilon = 10^{-6}$. Here, the domain is triangulated into $12,800$ elements with $\Delta x = 0.0125$. To locate the interface, we use a projection scheme determined by the maximum dot product. The numerical stationary interface solution is shown in Figure 4.7.b.

To check the stability of the triple junction, we measure the junction angles at each time step and calculate the corresponding errors. We observe that the interior phase angles at the top and bottom triple junctions behave in the same manner, hence, we only plot the relative error in the measure of the top junction angle in Figure 4.8.

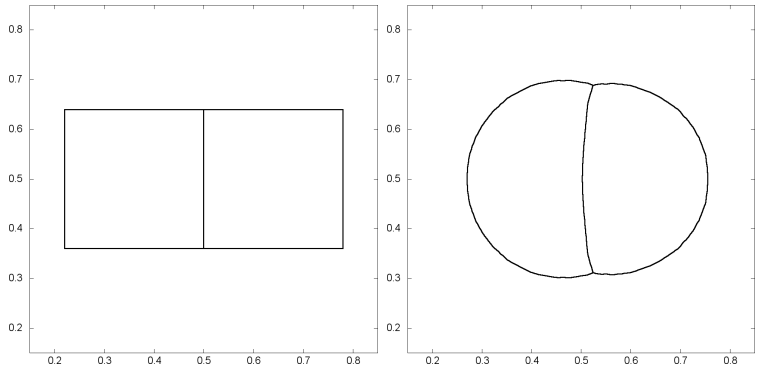


Figure 4.7: (a) Initial Condition. (b) The stationary numerical solution in case 1.

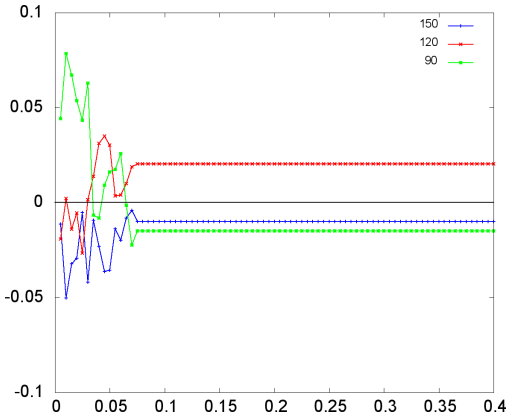


Figure 4.8: Relative error of the top junction angles at each time step.

It is evident from the error plot that the triple junction first rapidly approximates the angle conditions, which is consistent with the previous results. Thereafter, the numerical solution gradually reaches a stationary state whose junction angles are of measure $150^\circ \pm 1.50$, $120^\circ \pm 2.43$, and $90^\circ \pm 1.36$ yielding a relative error of at most 2%. This is consistent with the result in our stability test. Hence, one can achieve a more precise result using the projection triangle to locate the interface.

Moreover, it is clear from Table 4.3 that the phase volumes are preserved. Hence, the stationary numerical solution obtain using our method fairly approximates the volume-preserving solution of case 1.

Table 4.3: Phase Volumes under penalty parameter $\epsilon = 10^{-6}$.

phase	prescribed vol	stationary state vol	absolute error
P_1	0.8432	0.84321	1.0×10^{-5}
P_2	0.0784	0.07840	3.0×10^{-6}
P_3	0.0784	0.07839	8.0×10^{-6}

4.1.4 Moving Bubble

We simulate the bubble motion with buoyancy (β) as outer force in two settings, $\theta = 60^\circ$ and $\theta = 120.1^\circ$. The numerical examples are conducted on a $[0,1] \times [0,1]$ domain which is triangulated into 12,800 elements, $\epsilon = 10^{-5}$, $\beta = -150$, $\delta_1 = \delta_2 = \Delta x$ (mesh size). Under the same buoyant force, we see that for a large contact angle ($\theta = 120.1^\circ$), the bubble detaches from the bottom phase, while for $\theta = 60^\circ$ the bubble remains attached (Figure 4.9).

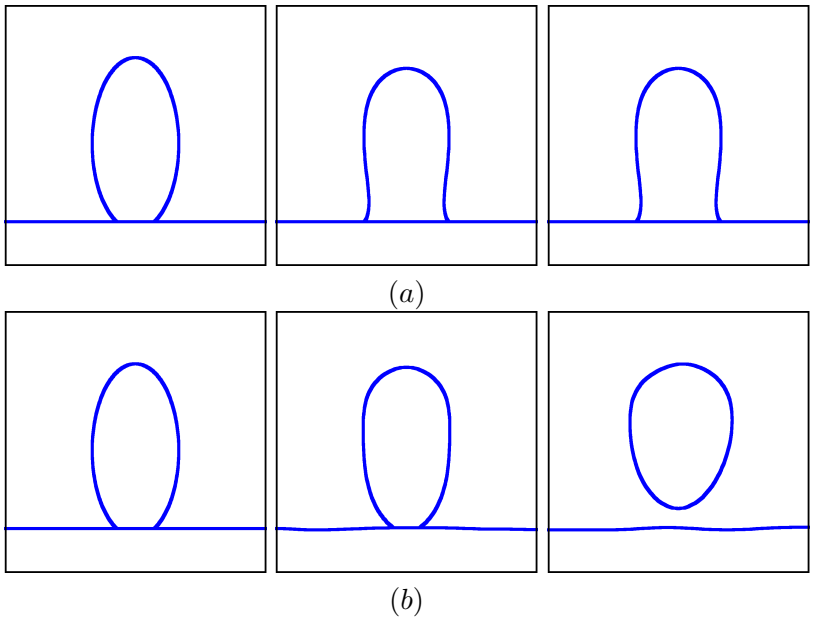


Figure 4.9: Bubble motion with (a) $\theta = 60^\circ$, $\Delta t = 0.005$ (b) $\theta = 120.1^\circ$, $\Delta t = 0.001$

4.2 Fluid Model

4.2.1 Cavity Flow

In this subsection, we simulate the cavity flow as test problem of fluid model using DSD/SST-SUPS formulation. The numerical test was conducted on a fixed $[0,1] \times [0,1]$ domain with time step $\Delta t = 0.002\ s$, upper boundary velocity $v = 0.02\ m/s$ with x positive direction, and no-slip boundary conditions on the bottom, right and left walls. We simulate some fluid with different Reynold numbers as shown in Table 4.4.

Table 4.4: Parameter of fluid test problems

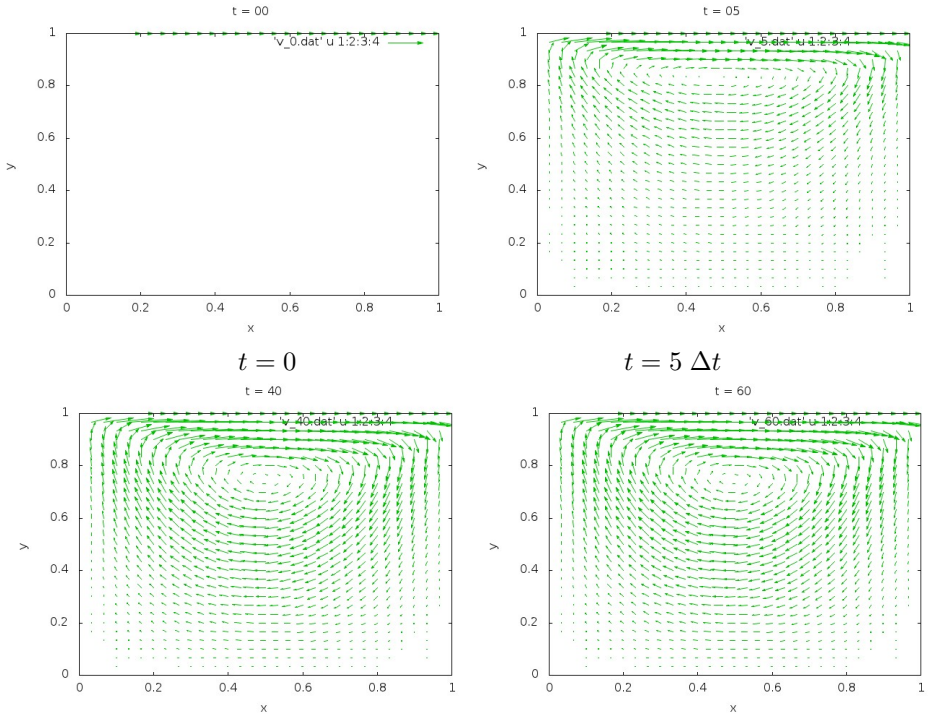
fluid	$\nu\ (m^2/s)$	$\rho\ (kg/m^3)$	$\mu\ (kg/ms)$	Re
1	1	1000	1000	0.02
2	0.01	1000	10	2
3	0.0001	1000	0.1	200

The vector field for several time-steps are shown in Figures 4.10, 4.11, and 4.12 for fluid 1, 2 and 3, respectively. The numerical simulations can catch the flow of cavity flow case. Moreover, it still works even in high Reynold numbers. Notice that we obtained all these result without having to refine the time step. This shows that the stability restrictions of the space-time approach are not so severe as in the usual spatial finite element method.

Note that in cavity flow problem, nonphysical singularities happened at the upper corner where stationary and moving walls meet [50, 51]. One of strategies to handle these singularities is to apply Navier-slip boundary condition [51]. The idea is almost the same with our setting in boundary condition of coupled problem (Section 2.4). Another strategy is by using fine enough elements close to the walls especially at the upper corner.

4.2.2 One-way Coupling

In this subsection, we simulate the examples of one-way coupling in hydrophilic and hydrophobic cases using Navier-Stokes equation in moving domain (DSD/SST-SUPS formulation). Hydrophilic materials have a good affinity to water, that cause droplets to spread. Hydrophobic materials repel water naturally, that cause droplets to form. These two kinds of materials



;

Figure 4.10: Velocity field of cavity flow for fluid 1

are characterized by the contact angle between the edge of droplet and the material's surface underneath it. If the contact angle less than 90°, the materials considered as hydrophilic and more than 90° for hydrophobic. In this example, we mimic the hydrophilic case by using a half of circle with radius $r(0) = 0.2$ as initial condition and evolve it based on the increasing of radius $r(t)$ while maintain the area A_0 (Figure 4.13). Here,

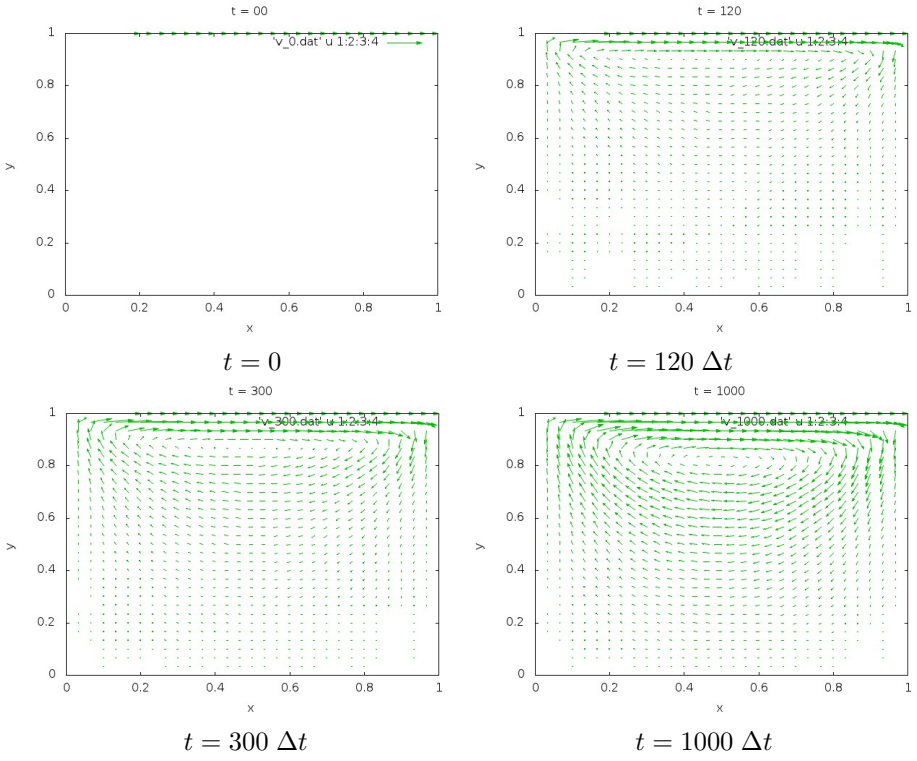
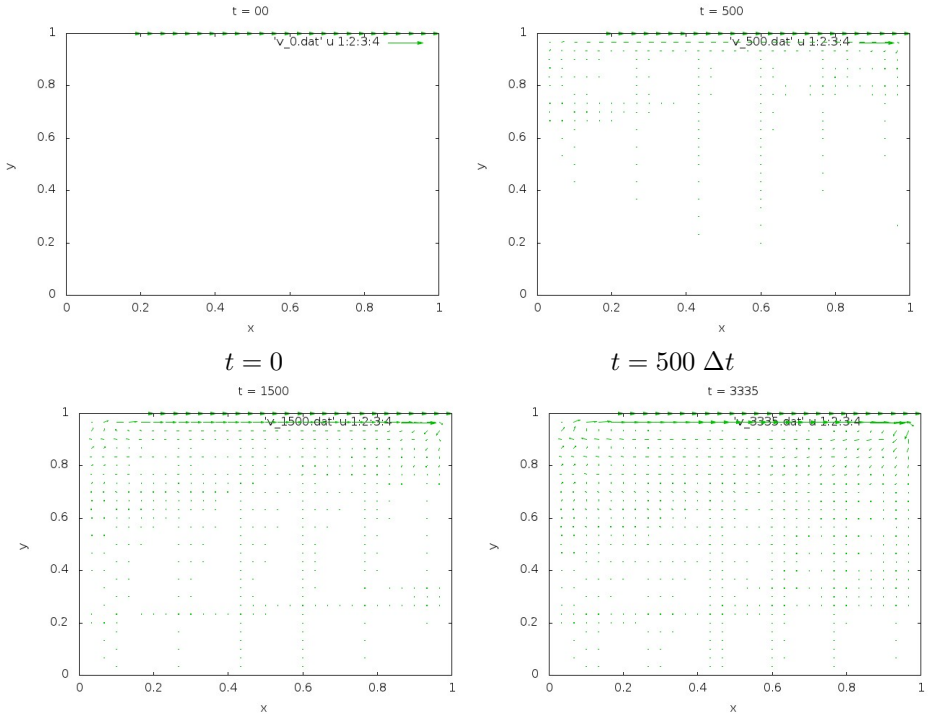


Figure 4.11: Velocity field of cavity flow for fluid 2

$$r(t) = \sqrt{\frac{A_0}{\alpha(t) - \frac{1}{\tan(\alpha(t))}}}$$
$$A_0 = \frac{1}{2}\pi(r(0))^2$$
$$\alpha(t) = \frac{\pi}{2} - c\ t\ dt$$

with $c = 0.25$ and $dt = 0.002$. Note that area preservation is important in this one-way coupling case, otherwise, the Navier-Stokes equation will collapse.

We simulate the hydrophobic case starting from the same initial condition as hydrophilic case and evolve it by decreasing the radius $r(t)$ while maintain the area A_0 (Figure 4.15). Here,



$t = 0$ $t = 500\ \Delta t$

$t = 1500\ \Delta t$ $t = 3335\ \Delta t$

;

Figure 4.12: Velocity field of cavity flow for fluid 3

$$r(t) = \sqrt{\frac{A_0}{\beta(t) - \frac{1}{\tan(\beta(t))}}}$$
$$A_0 = \frac{1}{2}\pi(r(0))^2$$
$$\beta(t) = \frac{\pi}{2} + c\ t\ dt$$

We solve Navier-Stokes equation in moving domain Ω_f. We apply boundary condition as in Section 2.4 with some simplifications. We extend region A until the triple point and extend region C until the endpoint that intersect external boundary. This means, we set velocity of the fluid equal to normal velocity of the whole gas-fluid interface and set the velocity of solid-fluid

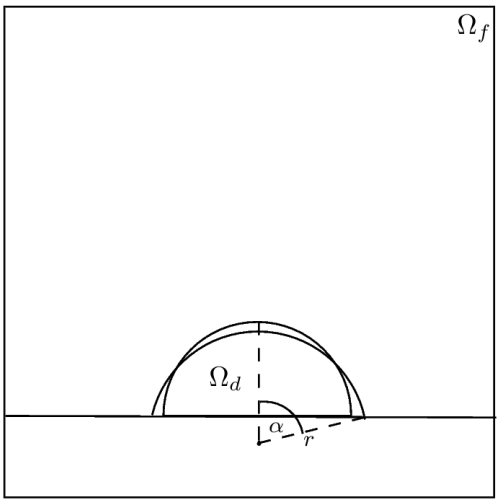


Figure 4.13: Hydrophilic setting

interface is only tangential, determined by the motion of equidistantly distributed boundary nodes. Here we write the algorithm to compute normal velocity of the whole gas-fluid interface:

1. Given data files: gas-fluid interface nodes at time t (nodes.dat), line segment of gas-fluid at time $t+1$ (segment.dat), circle center node at time t (center.dat).
2. Iteration for all nodes in nodes.dat.
3. Construct the line connecting each node with the center of the circle.
4. Find the intersection of this line with a certain segment in segment.dat
5. Compute the displacement of intersection point and the node (step 2).
6. Compute the velocity by dividing the displacement with time-step.

The flow of hydrophilic and hydrophobic case are depicted in Figures 4.14 and 4.16, respectively.

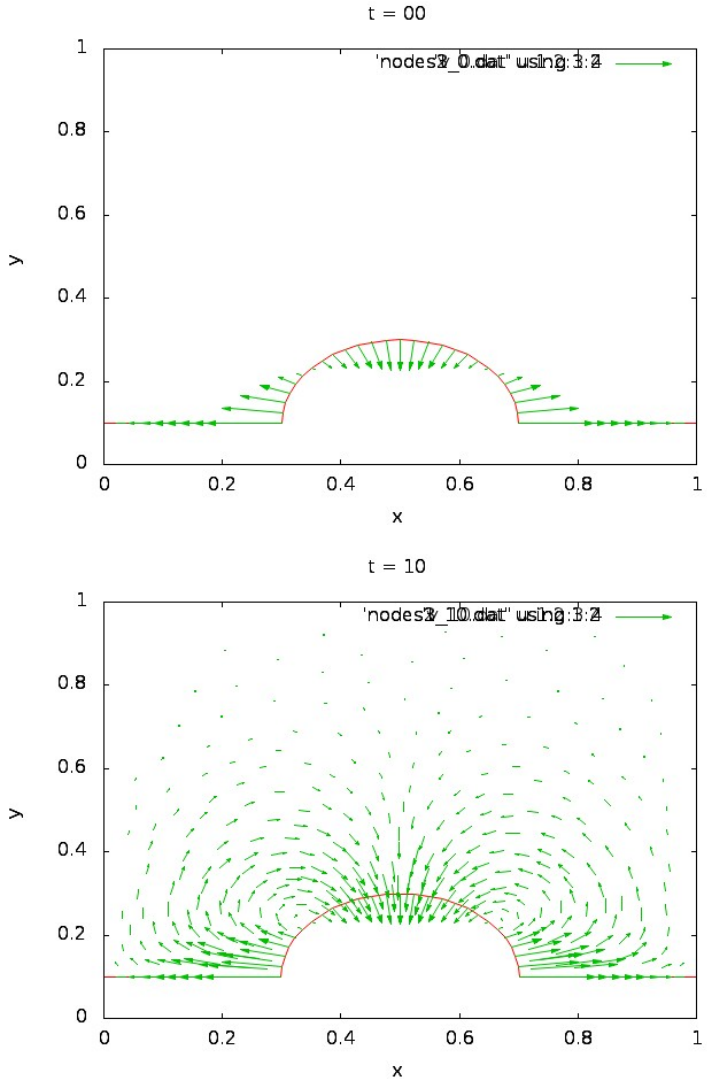


Figure 4.14: One-way coupling in hydrophilic case at $t = 0$ and $t = 10\ \Delta t$

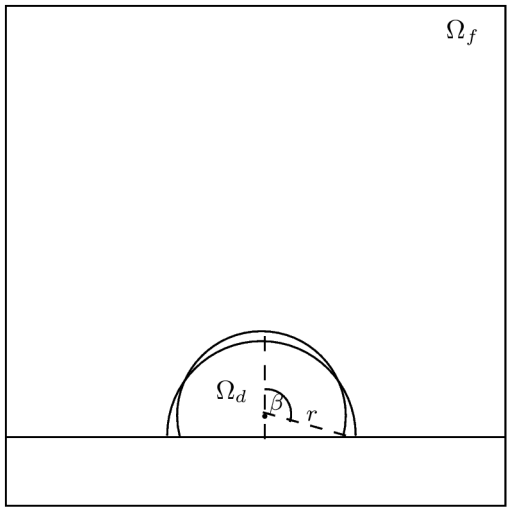


Figure 4.15: Hydrophobic setting

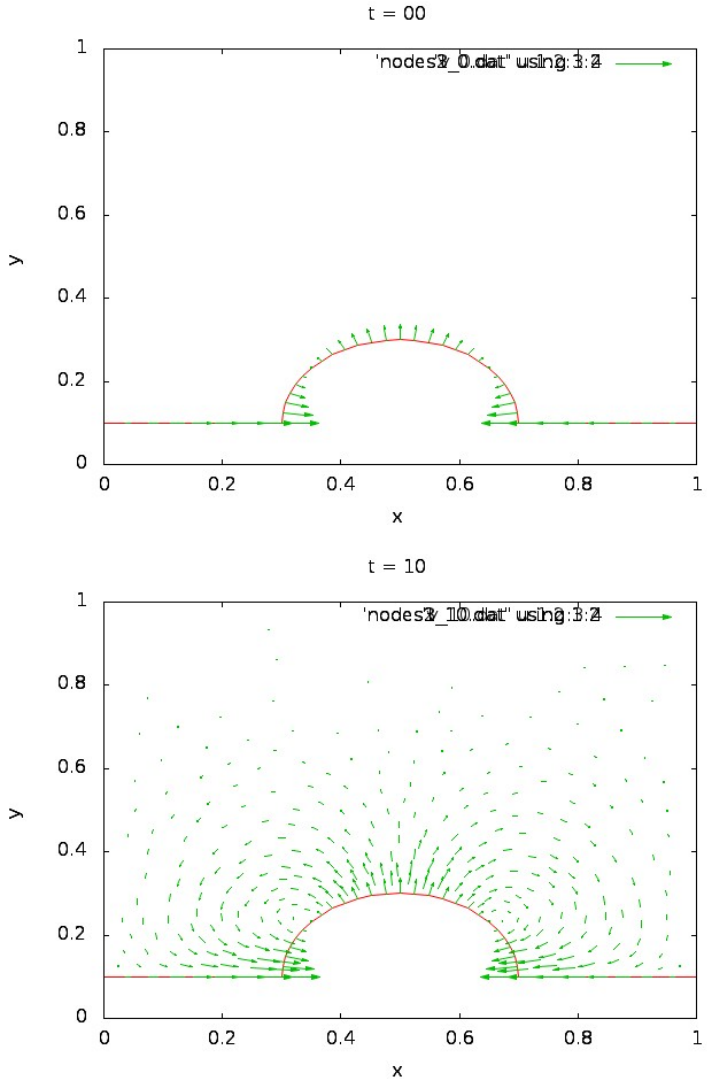


Figure 4.16: One-way coupling in hydrophobic case at $t = 0$ and $t = 10\ \Delta t$

Chapter 5

Conclusion

The dissertation developed a coupled interface-network and fluid model. We derived equation of the triple junction and adopted Navier-Stokes equation for incompressible flow as the basic interface model and fluid model, respectively. For each model, we developed a method for its numerical solution. This coupled model is expected to serve as a first step to more complicated and precise models, such as those including inertia, hysteresis or transport, which can be used to simulate triple line dynamics accurately.

We proposed a generalized vector-valued BMO algorithm to treat triple junction motion with nonsymmetric junction. Based on several numerical experiments and formal analysis, this new algorithm is able to realize desired triple junction motion for arbitrary surface tensions, i.e., it stably imposes contact angles at the junctions, gives correct velocity of interfaces and preserves volume. These features, especially the third one, support the interface-network model to be coupled with the fluid model as a first step toward accurate simulation of triple line dynamics.

There are many important and interesting problems for the future research, such as to

- understand the contact angle hysteresis, develop the interface model including the effect of hysteresis and analyze the influence of fluid motion;
- understand the inertia of the interfaces, develop the interface model including the effect of inertia (hyperbolic-type MCF) and investigate the impact on contact angle dynamics;
- study the physically appropriate boundary conditions related to the coupling of fluid and interface in case of interface between two fluids;

and

- use these models to investigate the laws behind triple line and contact angle dynamics.

Bibliography

[1] A. H. King, "Triple lines in materials science and engineering," in *Science Direct*, vol. 62, pp. 889-893, 2010.

[2] S. Zahedi, G. Kreiss, K. Gustavsson, "An interface capturing method for tw0-phase flow with moving contact lines," in *Proceeding of the 1st European Conference on Microfluidics - Microfluidics 2008*, 2008.

[3] H. Huang, D. Liang, B. Wetton, "Computation of a moving drop/bubble on a solid surface using a front-tracking method," in *Comm. Math. Sci.*, pp. 535-552, 2004.

[4] P. D. M. Spelt, "A level set approach for simulations of flows with multiple moving contact lines with hysteresis," in *J. Comp. Physics*, vol. 207, pp. 389-404, 2005.

[5] M. Renardy, Y. Renardy, J. Li, "Numerical simulation of moving contact line problem using a volume-fluid method," in *J. Comp. Physics*, vol. 171, pp. 243-263, 2001.

[6] D. Jacqmin, "Contact-line dynamics of a diffuse fluid interface," in *J. Fluid Mech.*, vol. 402, pp. 57-88, 2000.

[7] M. C. Lai, Y. H. Tseng, H. Huang, "Numerical simulation of moving contact lines with surfactant by immersed boundary method," in *Comm. Comp. Phys.*, vol. 8, no. 4, pp. 735-757, 2010.

[8] E. Ginder, K. Svadlenka, "A thresholding algorithm generating motion by hyperbolic mean curvature flow," preprint, 2014.

[9] M. Elsey, S. Esodoglu, P. Smereka, "Diffusion generated motion for grain growth in two and three dimensions," in *J. Comp. Physics*, vol. 228, pp. 8015-8033, 2010.

[10] F. Catte, P. L. Lions, J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion," in *SIAM J. Num. Anal.*, vol. 29, pp. 182-183, 1992.

[11] K. Svadlenka, E. Ginder, S. Omata, "A variational method for multiphase volume-preserving interface motions," in *Journal of Computational and Applied Mathematics*, vol. 257, pp. 157-179, 2014.

[12] S. J. Ruuth, "A diffusion-generated approach to multiphase motion," in *J. Comp. Physics*, vol. 145, pp. 166-192, 1998.

[13] T. E. Tezduyar, "Stabilized finite element methods for computation of flows with moving boundaries and interfaces" ,Lecture notes on Finite element simulation of flow problems (basic advanced course), Japan Society of Computational Engineering and Science, 2003.

[14] T. E. Tezduyar and S. Sathe, "Stabilization parameters in SUPG and PSPG formulations," in *Journal of Computational and Applied Mechanics, Vol 4., N0.1, (2003)*, pp. 71-88.

[15] S. Omata, M. Kazama, K. Svadlenka, "Numerical computation of coupled problems comprising elastic membrane," in *Proceedings of Computational Engineering Conference, JSCES*, vol. 15, 2010.

[16] N. Shofianah, R. Z. Mohammad, K. Svadlenka, "Simulation of triple junction motion with arbitrary surface tensions," preprint,2013.

[17] B. Merriman, J. Bence, S. Osher, "Motion of multiple junctions: a level set approach," in *J. Comp. Physics*, vol. 112, pp. 334-363, 1994.

[18] L. Evans, "Convergence of an algorithm for mean curvature motion," in *Indiana Univ. Math. J.*, vol. 42, pp. 533-557, 1993.

[19] G. Barles, C. Georgelin, "A simple proof of convergence of an approximation scheme for computing motions by mean curvature," in *SIAM J. Numer. Anal.*, vol. 32, pp. 484-500, 1995.

[20] Y. Goto, K. Ishii, T. Ogawa, "Method of the distance function to the Bence-Merriman-Osher algorithm for motion by mean curvature," in *Comm. Pure Appl. Anal.*, vol. 4, pp. 311-339, 2005.

[21] R. Z. Mohammad, K. Svadlenka, "Multiphase Volume-preserving Interface Motions via Localized Signed Distance Vector Scheme", preprint, 2013.

[22] R. Z. Mohammad, "Multiphase mean curvature flow: signed distance vector approach," in *Recent development in computational science: selected papers from the ISCS*, vol. 4, pp. 115-123, 2013.

[23] L. Bronsard, F. Reitich, "On three-phase boundary motion and the singular limit of a vector-valued Ginzburg - Landau equation," in *Arch Ration. Mech. Anal.*, vol. 124, pp. 355-379, 1993.

[24] H. Garcke, B. Nestler, B. Stoth, "A multiphase field concept: numerical simulations of moving phase boundaries and multiple junctions," in *SIAM J. Appl. Math* , vol. 60, pp. 295-315, 1999.

[25] H. Garcke, B. Nestler, B. Stoth, "On anisotropic order parameter models for multi-phase systems and their sharp interface limits," in *Physica D*, vol. 115, pp. 87-108, 1998.

[26] X. Chen, J-S. Guo, "Self-similar solutions of a 2-D multiple-phase curvature flow," in *J. Comp. Physics*, D 229, pp. 22-34, 2007.

[27] K. Ishii, "Mathematical analysis to an approximation scheme for mean curvature flow, in: S. Omata, K. Svadlenka (Eds.), International Symposium on Computational Science 2011," in *Mathematical Sciences and Applications*, vol. 34, GAKUTO International Series, pp. 67-85, 2011.

[28] E. Rothe, "Zweidimensionale parabolische Randwertaufgaben als Grenzfall eindimensionaler Randwertaufgaben," in *Math. Ann.*, vol. 102, pp. 650-670, 1930.

[29] T. E. Tezduyar, S. Sathe, R. Keedy, K. Stein, "Space-time finite element techniques for computation of fluid-structure interactions," in *Comp. Methods Appl. Mech. Engrg.*, vol. 195, pp. 2002-20027, 2006.

[30] T. J. R. Hughes, "The finite element method: Linear static and dynamic finite element analysis," Prentice-Hall, Inc, 1987.

[31] S. Elgeti, M. Behr, "Block (2D Space-time incompressible Navier-Stokes) Derivation," 2013.

[32] T.P. Fries and H.G.Matthies "A review of petrov-galerkin stabilization approaches and an extension to meshfree methods", Technische Universitat Braunschweig , 2003.

[33] S. Gross and A. Reusken, "Numerical methods for two-phase incompressible flows", Springer, 2011.

[34] M. Benzi, G.H. Golub, J. Liesen "Numerical solution of saddle point problems", Cambridge University Press, United Kingdom, 2005.

[35] Y. Saad, "Iterative methods for sparse linier systems", second edition, Society for Industrial and Applied Mathematics, 2003.

[36] Y. Saad and M. Schultz "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems", Society for Industrial and Applied, Vol.7, N0.3, 1986.

[37] T. Sauer, "Numerical analysis", Pearson Education, Inc, 2006.

[38] R.L. Burden and J.D. Faires "Numerical Analysis", ninth edition, Brooks/Cole Cengage Learning, 2011.

[39] Y. Bazilevs, K. Takizawa, T. E. Tezduyar, "Computational Fluid-Structure Interaction: methods and applications", Wilwy series in computational mechanics, 2013.

[40] K. Takizawa, T. E. Tezduyar, "Multiscale space-time fluid-structure interaction techniques", in *Comp. Mech.*, vol. 48, pp. 247-267, 2011.

[41] H. Zhao, T. Chan, B. Merriman, S. Osher, "A variational level set approach to multiphase motion", in *J. Comp. Physics*, vol. 127, pp. 179-195, 1996.

[42] S. Osher, J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations", in *J. Comp. Physics*, vol. 79, pp. 12-49, 1988.

[43] V. S. Nikolayev, S. L. Gavrilyuk, H. Gouin, "MModeling of the moving deformed triple contact line: influence of the fluid inertia," in *J. of Colloid and Interface Science*, vol. 302, pp. 605-612, 2006.

[44] C. Huh, L. Scriven, "Hydrodynamic model of steady movement of a solid/liquid/fluid contact line," in *J. Coll. Int. Sci.*, vol. 35, pp. 85-101, 1971.

[45] M. Behr, "On the application of slip boundary condition on curved boundaries,", in *International Journal of Numerical Methods in Fluids*, vol. 45, pp. 43-51, 2004.

[46] M. S. Engelman, R. L. Sani, "The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible

fluid flow,", in *International Journal for Numerical Methods in Fluids*, vol. 2, pp. 225-238, 1982.

[47] T. E. Tezduyar, K. Takizawa, Y. Bazilevs,"Short-course on Fluid-Structure Interaction", March 22-23, 2014, Waseda University.

[48] Y. Chen, R. Mertz, R. Kulenovic, "Numerical simulation of bubble formation on orifice plates with a moving contact line," in *International Journal of Multiphase Flow*, vol. 35, pp. 66-77, 2009.

[49] S. Esedoglu, F. Otto "Threshold dynamics for networks with arbitrary surface tensions,"preprint, 2013.

[50] T. E. Tezduyar, S. Mittal, S. E. Ray, R. Shih, "Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements," in *Computer Methods in Applied Mechanics and Engineering*, vol. 95, pp. 221-242, 1992.

[51] Q. He, X. P. Wang, "Numerical study of the effect of navier slip on the driven cavity flow," in *ZAMM Z. Angew. Math. Mech.*, vol. 89 no. 10, pp. 857-868, 2009.