UNIVERSITY OF CALIFORNIA

Los Angeles

A Particle Method for Multiphase Mechanics Simulation

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Yi-Jui Chang

2020

ABSTRACT OF THE DISSERTATION

A Particle Method for Multiphase Mechanics Simulation

by

Yi-Jui Chang
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2020
Professor Jeff D. Eldredge, Chair

Modeling the multiphase mechanics with coupled fluid and elastic material is important for many applications such as blood perfused soft tissues, wicking in porous medium. The liquid-solid interaction results in complicated effect on structure deformation and liquid transportation. The current study aims to develop high visual and physical fidelity simulations of multiphase mechanics, particularly within the context of soft tissue swelling, human injuries, medical treatments, the transport of blood through damaged tissue under bleeding or hemorrhaging conditions and droplet spreading on a fabric. The solid material is considered as a dynamic poro-hyperelastic material with liquid-filled voids. A biphasic formulation—effectively, a generalization of Darcy's law—is utilized, treating the phases as occupying fractions of the same volume. A Stokes-like friction force, a pressure that penalizes deviations from volume fractions summing to unity and the surface tension between multiphase interface, serve as the interaction force between solid and liquid phases. The resulting equations for both phases are discretized with the method of Smoothed Particle Hydrodynamics (SPH). The solver is validated separately on each phase and demonstrates good agreement with exact solutions in test problems. Simulations of oozing, hysteresis, swelling, drying and shrinkage, tissue fracturing and hemorrhage, liquid droplet spreading on a fabric are shown in this work. Besides the physical-based SPH solver, the technique called dynamic mode decomposition (DMD) from data science also applies on the results from SPH solver to extract the system features without any knowledge of governing equations, providing benefits such as data compression and efficient data manipulation, raising

the potential of developing data-driven computational solver in the future.

The dissertation of Yi-Jui Chang is approved.

Pirouz Kavehpour

Lihua Jin

Joseph M. Teran

Jeff D. Eldredge, Committee Chair

University of California, Los Angeles

2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis and research advisor, Prof. Jeff Eldredge, for being a supportive mentor, great teacher and advisor during my PhD life in UCLA. This work would never have been possible without Jeff's guidance and encouragement.

I want to thank my committee members, Prof. Pirouz Kavehpour, Prof. Joseph Teran, Prof. Lihua Jin for their time and valuable insights.

Thanks also goes to all members in SOFIA lab, for all our discussion, experience sharing, brainstorming on all aspects in these years.

Last but not least, I would like to thank my family, my significant other, An-Yun, and all my friends. I could never complete this journey without their mental support, understanding and companionship.

VITA

2013	B.S. in Mechanical Engineering, National Taiwan University, Taipei, Taiwan.
2014	M.S. in Mechanical Engineering, National Taiwan University, Taipei, Taiwan.

PUBLICATIONS AND PRESENTATIONS

- Li, L., Maccabi, A., Abiri, A., Juo, Y.Y., Zhang, W., Chang, Y.J.*, Saddik, G.N., Jin, L., Grundfest, W.S., Dutson, E.P. and Eldredge, J.D., Characterization of perfused and sectioned liver tissue in a full indentation cycle using a visco-hyperelastic model. *Journal of the mechanical behavior of biomedical materials*, 2019.
- Chang, Y.J.*, Eldredge, J.D., Benharash, P.,Dutson, E.P., Smoothed particle hydrodynamics simulation of biphasic soft tissue and its medical applications, 2020, in prep.

- Chang, Y.J.*, Chong, K., Eldredge, J.D., Teran, J., Benharash, P.,Dutson, E.P., A biphasic model for bleeding in soft tissue,*70st Annual Meeting of the APS Division of Fluid Dynamics*, Denver, CO, November 19-21, 2017. Presentation.
- Chang, Y.J.*, Canuto, D., Chong, K., Eldredge, J.D., Teran, J., Benharash, P.,Dutson, E.P., A biphasic computational model of blood-perfused soft tissue coupled with a cardiovascular model,*12th Southern California Flow Physics Symposium (SoCal Fluids XII)*, Los Angeles, CA, April 14, 2018. Presentation.

- Chang, Y.J.*, Canuto, D., Chong, K., Eldredge, J.D., Teran, J., Benharash, P.,Dutson, E.P., A biphasic computational model of the mechanics of the blood-perfused liver, *71st Annual Meeting of the APS Division of Fluid Dynamics*, Atlanta, GA, November 25-27, 2018. Presentation.
- Chang, Y.J.*, Eldredge, J.D., Benharash, P.,Dutson, E.P., Application of deep learning to smoothed particle hydrodynamics,*13th Southern California Flow Physics Symposium (SoCal Fluids XIII)*, Santa Barbara, CA, April 20, 2019. Presentation.

CHAPTER 1

Introduction

1.1 Background

There has recently been great deal of interest in numerical simulation on multiphase mechanism. In general, multi-phases mechanism usually refers to the physical phenomena that dealing with interaction of fluid and solid phase. Fluid here, may be liquid or gas, usually acts on the solid phase by its pressure force and friction force, while solid phase usually play a role as structure to sustain the movement of fluid. In small scale problems, the imbalance of adhesion and cohesion between each phase interface that resulting in surface tension should also be taken into consideration. Besides developing a reliable numerical method for simulating multiphase mechanics, to accelerate the computation to real time is another important effort. This acceleration requires the replacement of the computationally-intensive steps with faster steps that suitably mimic their results. For this replacement, we are addressing to the using tools from data science that may just learning the feature of the system directly from the data instead of solving the governing equations.

This chapter outlines the features and major applications of multiphase problem and the relevant work from the literature, closing with the objectives and the structure of this work.

1.1.1 Soft Tissue

Investigations of blood-perfused tissue with solid-liquid interaction have drawn a great deal of attention in the research community over the past few decades [CD07, RDD10, EW15, CJR^{+}17, BFH18]. The soft tissue can be regarded as an elastic porous medium whose sponge-like structure contains pores that may contain fluids in both liquid and gaseous phase. Because of the complexity of the micro-scale interactions between the fluids and solid inside the tissue, the macroscopic behavior remains quantitatively vague from both the experimental and numerical points of view. However, many medical applications, such as surgical training, pre-surgery evaluation, or trauma care, would benefit greatly from integrated numerical simulation of this behavior, at a level of resolution that preserves physical and visual fidelity—for example, to enable virtual interaction with an organ that bleeds realistically when injured or under surgical intervention. The micro-scale behavior is irrelevant to this level of application, aside from the degree to which it determines the macroscopic behavior.

Such simulations clearly have need for mesoscopic resolution—a level of resolution at which the microstructure is averaged out in favor of a locally homogeneous treatment of each phase. Mathematical models that attempt to capture the interactions between phases without resolving their interface, sometimes referred to as "biphasic" models (or "triphasic" if three phases are included), have been available for several decades. The theory of poroelasticity, proposed first by Biot [Bio41, DC93], has been widely applied in the fields of geomechanics and biomechanics under the conditions of quasi-static deformation of the solid material. This framework raised two new parameters, porosity and permeability, to describe the volume fraction of void in an infinitesimal parcel of material and the ability of a porous medium to allow fluids to pass through it. These parameters serve as coefficients of linear relations between stress, strain and pore pressure in the constitutive equation of solid material in the transport equations. An alternative foundation for the porous media model, based on the thermodynamic point of view, was based on the theory of mixtures [Bow80, MKLA80]. These concepts were later extended to constitute the theory of porous media (TPM), developed by de Boer and his coworkers [DBK83, DBE86, DB06], which relies on a definition of phase volume fraction as the ratio of the volume of each phase in an infinitesimal parcel of the biphasic material to the total volume of that parcel. The associated equations of motion for each phase, outfitted with interaction forces and the saturation constraint that the summation of volume fraction should remain unity, provides a thorough mathematical framework for studying multi-phases problems.

1.1.2 Numerical Tool

Numerical simulation of these equations, particularly in cases of complex geometry and with non-linear material behaviors, generally requires high computational cost. The finite element method is a common choice for treating multi-phases problems [DB06, RSB07, CGSMVC10, RDD10, RE10, EW15, BFH18]. However, many problems of biomedical nature involve changes in topology—fracturing of solid material, liquid–gas interfaces and associated phenomena—that are challenging for mesh-based methods. Particle methods, in contrast, rely on advecting computational elements that share no inherent connectivity, and thus naturally adapt to changing structure and topology. Smoothed particle hydrodynamics (SPH), in particular, is a well-known particle method first used in the field of astrophysics [GM77] but later applied to general continuum theory [LP91, OVSM98, Mon00, GMS01, LL03, MCG03, Kel06]. This method divides the continuum material into a set of discrete particles carrying their own physical quantities, from which the continuum fields can be locally interpolated via a particle kernel function. The method has been extended in recent years to materials of various constitutive behaviors of biomedical relevance, such as hyperelastic materials [KS07], non-Newtonian fluids [SL03, VM08], and viscoelastic materials [ET05, RMH07]. The aspect of these methods that incur the greatest computational expense, the radius searching required to populate nearest-neighbor lists for each particle, can rely on well-established libraries that allow it to be carried out efficiently[BSC15]. Other techniques for computational acceleration have also been applied, such as parallelization [SWB$^+$06, SWP$^+$06, HKK07, CDB$^+$11, IABT11, DCVB$^+$13, CDR$^+$15] or numerical pressure treatments that enable larger time step size [SP09, TOSF11].

For these reasons, SPH has been applied for biomedical applications in recent years (see, e.g., [CJR$^+$17, TGK$^+$17]), with some limited interaction between phases. However, in spite of these recent advances, an SPH algorithm for biphasic modeling of perfused soft tissue is still missing from the literature. Therefore, in this work, we seek to provide an alternative for simulating problems that commonly arise in biomedical applications, such as the mechanics of blood-perfused soft tissues, and particularly, their dynamic response to various forcing modalities that arise in injury or surgery. To address this, we extend the single-phase SPH framework to a biphasic formulation. Our proposed methodology draws inspiration from the existing weakly-compressible treatment in SPH to penalize deviations of the phase volume fraction from the saturation condition. The exchange of momentum between phases is manifested in the pressure that arises from these deviations from saturation and from a Stokes-like friction force proportional to the relative local velocities between phases. This approach enables the method to capture various biphasic phenomena, including fracturing, swelling, shrinkage, hysteresis, oozing, and hemorrhaging.

1.1.3 Wicking Problem

When it comes to small scale multiphase problem, precisely, small Capillary number, defined as $Ca = \frac{\mu V}{\gamma}$, the effect of surface tension become more and more important. Surface tension actually comes from the difference of attraction force between liquid/liquid, liquid/solid and liquid/air molecules, resulting in a tendency of liquid surfaces to shrink into minimum surface area. The flow with such small length scale in the porous medium, is referred as imbibition or wicking, which defined as replacement of one fluid by another fluid in a porous medium. If liquid in the pores of porous medium was not saturated, creating some liquid-solid interface inside the pore, the surface tension would act as driven force to transport the liquid. Wicking problem usually connects with application with moisture management of clothing [PRKG06, SK14, PVR$^+$17, Ozd17], where absorption and transportation of sweat influence on our daily life comfort.

The mathematical model of this type of problems is similar with that of soft tissue problem. The model still focuses on mesoscopic resolution, but requires additional mathematical form to represent surface tension. The pore size of the porous medium will not explicitly appear in the governing equation but associated with the strength of surface tension. Another important parameter, saturation, which indicates the percentage of volume occupied by the liquid inside the pore, is required here to switch the importance of surface tension at the mesoscopic point such that surface tension would have no effect as whole volume was occupied by the liquid.

First attempt of SPH formulation for surface tension can be found in [Mor00, MCG03]. They defined a new field quantity called color field to construct surface of the flow and thus obtaining surface normal as well as curvature. However, in Kelager's work [Kel06], he addressed an issue that this asymmetrical surface tension formulation would cause unphysical fluid motion in some types of flow problem. Later, three different ways to resolve the issue were published [TM05, Zha10, BPHK13, TP16]. Breinlinger *et al.*[BPHK13] improved the previous method by correcting surface normal to ensure smooth transition. Zhang [Zha10] utilized Lagrangian interpolation polynomial and moving least squares method to reconstruct the actual boundary surface of the flow, providing more accurate description on surface normal and curvature. Tartakovsky and his coworkers [TM05, TP16] raise another approach that considering particle-particle interaction instead of obtaining a boundary surface, to mimic the intrinsic nature of adhesion and cohesion between molecules. In this manner, some promising simulations such as droplet between two parallel plates and spreading of a droplet on a horizontal surface were demonstrated. However, although surface tension force seems to be modeling well in SPH, wicking problem, resulting from surface tension force as well, involves with diffusion and transportation of fluid in the porous medium, is still missing thorough SPH formulation. Therefore, we extend the current particle-particle interaction force to consider the similar interaction force from solid material and incorporate with permeability and saturation parameter to model wicking problem. This approach enables the SPH method to capture the mechanism of flow spreading in the porous medium due to the surface tension in the pore size scale.

1.1.4 Data Science on Particle Method

Applying knowledge from data science on fluid dynamics to analyze large amount of data no matter from numerical simulation or experimental data has been re-drawn attention from fluid community during the last decade since the recent development of computational resource. In such a highly non linear system governed by Navier-Stokes' equations, data science technique paves a new path to analyze the fluid dynamics instead of solving the mathematical equations directly. Dynamic mode decomposition (DMD) is one of a useful regression technique of data science that have been widely applied to various fluid systems to extract the features of system and analyze the flow.

The work here aims to extend the application of DMD on the simulation results from specific Lagrangian particle method, smoothed particle hydrodynamics (SPH). These state vector from the particle dynamics, served as the input data to the regression, is totally different with usual input data with inherent position information such as fixed grid or sensor location. We provide a new perspective with DMD technique on particle method and successfully reconstruct original flow from the extracting DMD modes and corresponding eigenvalues. The effort in the work provides more insightful understanding of DMD on SPH, extending the usage of DMD on different types of input data, forming the basis of future application that building a data-driven fluid solver from the knowledge of DMD modes.

1.2 Objectives

Although the TPM has been widely used to describe multi-phases problem and some kind of numerical tools are available to simulate multiphase mechanism. Due to the huge potential of particle method from recent progress of computational improvement, the present research aims to develop a numerical tool with SPH to simulate multi-phases interaction and focuses on the soft tissue and wicking problem, and therefore has the following objectives:

1. Develop a numerical tool with particle method to capture liquid solid interaction such as fracturing, swelling, shrinkage, hysteresis, oozing, and hemorrhaging

2. The numerical method should be adaptive with general constitutive equations such as nonlinear elasticity, non-Newtonian fluid.

3. Surface tension that causes wicking and droplet formation should be incorporated in the SPH method.

4. High performance computing that including parallel computing, GPU computing should be used in the SPH method.

5. Apply whole numerical tool to simulate a real organ, the liver, and couple the cardiovascular system as the inflow condition of the blood vessel in the liver.

6. Perform data science technique from the above numerical tool to achieve faster, physically acceptable, simulation.

Chapter 2 and 3 focus on the first four objectives by detailing the SPH formulations for solid liquid interaction and surface tension as well as their implementation. Chapter 4 address the fifth objective and provide validation test on the new designed SPH method and the real liver simulation. Then chapter 5 shows our thought and work on the final objective. Finally, chapter 6 concludes with a summary of accomplished goals and some thoughts for future work.

CHAPTER 2

Biphasic Modeling with SPH

In this chapter, we briefly review the basic concept and governing equations of multiphase materials, which are developed from the theory of porous media, as well as other concepts from continuum mechanics. Then we summarize the method of smoothed particle hydrodynamics (SPH) and present our extension of SPH method to the biphasic system and its implementation details.

2.1 Theory of Porous Media

Here we present the governing equations for biphasic material mechanics. This is meant only as a brief review, with details provided only insofar as necessary to support the methodology of this work. Deeper details can be found in the works of [MKLA80, Bow80, DBK83, DBE86, DBE90, RDB03, dBD04, DB06, RSB07, RDD10, EM01]

2.1.1 Representative Volume Element in Mesoscale

In continuum mechanics, materials are regarded as completely filled by the substance of that material. In such a viewpoint, a point in the continuum is associated with an infinitesimal material parcel consisting of a collection of molecules and atoms, and the point inherits the bulk properties of this parcel, such as velocity and density. This idea was illustrated in the figure 2.1 from Cowin's work [CD07]. The material parcel in biphasic problem is called as representative volume element (RVE) here, containing both solid material and fluid material in the continuum point. The length scale of this material parcel, L_{RVE}, is much larger than the microstuctural scale, L_M, and smaller than the problem scale, L_p. (i.e.

$L_p >> L_{RVE} >> L_m$)

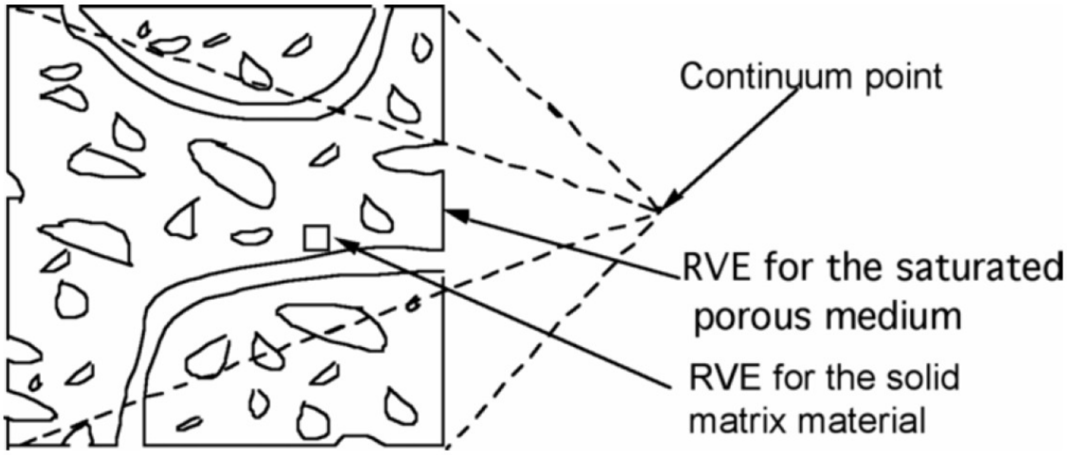

Continuum point
RVE for the saturated porous medium
RVE for the solid matrix material

Figure 2.1: RVE representation of material parcel in porous medium [CD07]

2.1.2 Volume Fraction

In the theory of porous media, a new parameter, volume fraction of any phase, is introduced to illustrate the multiphase status in material parcels, defined as the ratio between the volume occupied by a single phase in the parcel and the volume of the material parcel, written as $n^\alpha = \delta V^\alpha / \delta V_{\text{total}}$, where superscript α indicates the phase and δV represents the volume. Another interpretation of the phase volume fraction is the ratio of the effective density of the phase in the multiphase material parcel and its “real" density as a single phase material, which can be easily derived from $n^\alpha = (\delta m^\alpha / \delta V_{\text{total}})(\delta V^\alpha / \delta m^\alpha) = \rho^\alpha / \rho_0^\alpha$, where δm^α is the mass of phase α in the material parcel, so that the definition of density in the parcel becomes $\rho^\alpha = \delta m^\alpha / \delta V_{\text{total}}$, which is different from the reference density, associated with the single phase density $\rho_0^\alpha = \delta m^\alpha / \delta V^\alpha$. An important condition on volume fraction can be obtained from the fact that all space in the parcel should be occupied by a phase of matter—solid, liquid, or gas—so that the summation of volumes of each phase equals the total parcel volume, or equivalently,

$$\sum_\alpha n^\alpha = 1. \tag{2.1}$$

2.1.3 Kinematics

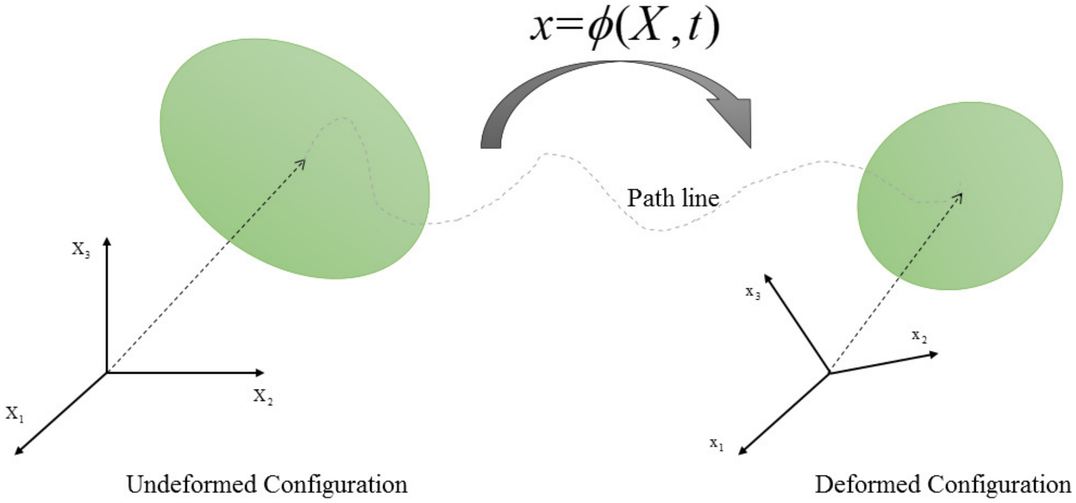


Figure 2.2: Motion of a continuum body

Continuum mechanics, for single or multiple phases, relies heavily on the relative kinematics of material points. To describe the motion of material particles, two configurations should be defined in advance. The first, the spatial coordinates (also called Eulerian coordinates), denoted by $\boldsymbol{x}$, specify the instantaneous location of a point in space; the second, the reference configuration (or Lagrangian or material coordinates), $\boldsymbol{X}$, represent the location of a point in the material frame of reference; these are usually taken to be the spatial coordinates in the initial configuration of the system (i.e., $\boldsymbol{X} = \boldsymbol{x}$ at $t = 0$). Then, a mapping function $\boldsymbol{\phi}(\boldsymbol{X}, t)$ describes the current position (at time t) of a particle with material coordinates $\boldsymbol{X}$:

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t). \tag{2.2}$$

This concept is shown in figure 2.2. The displacement $\boldsymbol{u}$ is thus defined as the difference between current position of the material point and its initial position,

$$\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{\phi}(\boldsymbol{X}, t) - \boldsymbol{X}. \tag{2.3}$$

The deformation gradient, $\boldsymbol{F}$, is a rank-2 tensor and defined as the derivative of the mapped coordinates $\boldsymbol{x}$ with respect to the reference coordinates $\boldsymbol{X}$ (the gradient of the mapping function):

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{\phi}(\boldsymbol{X}, t)}{\partial \boldsymbol{X}}; \tag{2.4}$$

this tensor describes the deformation of a material line element from the reference configuration to its deformed configuration. The determinant of the deformation gradient, $J = \det \boldsymbol{F} = \rho_0/\rho$, represents the ratio of the reference density of the material configuration and the density at the current configuration. In a dynamical system, the deformation gradient can be obtained at state n from a previous state $n-1$ via the updated Lagrangian formulation,

$$\boldsymbol{F}^{<n>} = \hat{\boldsymbol{F}} \cdot \boldsymbol{F}^{<n-1>}, \tag{2.5}$$

where $\hat{\boldsymbol{F}} = \boldsymbol{I} + \boldsymbol{\nabla u}$ (with $\boldsymbol{\nabla u} = \partial \boldsymbol{u}/\partial \boldsymbol{x}$) represents the current deformation in the deformed configuration (rather than the reference). The velocity gradient $\dot{\boldsymbol{F}}$ in Lagrangian configuration is defined as

$$\dot{\boldsymbol{F}} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \boldsymbol{\nabla v} \cdot \boldsymbol{F}, \tag{2.6}$$

where $\boldsymbol{v}$ is the velocity and the spatial velocity gradient is denoted as $\partial \boldsymbol{v}/\partial \boldsymbol{x} = \boldsymbol{\nabla v}$. The velocity gradient, most commonly used to describe fluid motion, can be further decomposed into a symmetrical part, also called the rate of strain, $\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}(\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T)$, and the anti-symmetrical part, the rotation rate tensor $\dot{\boldsymbol{R}} = \frac{1}{2}(\boldsymbol{\nabla v} - (\boldsymbol{\nabla v})^T)$. The right and left Cauchy–Green deformation tensors—$\boldsymbol{C}$ and $\boldsymbol{B}$, respectively—are then defined as $\boldsymbol{C} = \boldsymbol{F}^T \cdot \boldsymbol{F}$, and $\boldsymbol{B} = \boldsymbol{F} \cdot \boldsymbol{F}^T$, while the Cauchy strain tensor, $\boldsymbol{E}$, is defined as $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$.

2.1.4 Governing equations

In the theory of porous media, the phases are effectively superimposed on each other, and material points associated with the different phases may occupy the same location. Each phase has an associated mapping function, $\boldsymbol{\phi}^\alpha$, and thus, its own set of kinematics. The material derivative of a phase is defined as

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}t} = \frac{\partial}{\partial t} + \boldsymbol{v}^\alpha \cdot \boldsymbol{\nabla}, \tag{2.7}$$

where $\boldsymbol{v}^\alpha$ is the local velocity of phase α. For brevity, we will generally suppress the α superscript on the derivative, and take it for granted that any quantity associated with a given phase is differentiated with respect to that phase.

The equation of conservation of mass is applied to each phase under the assumption that the phases are microscopically immiscible and do not exchange mass; the equation for phase α is

$$\frac{\mathrm{d}\rho^\alpha}{\mathrm{d}t} = -\rho^\alpha \boldsymbol{\nabla} \cdot \boldsymbol{v}^\alpha. \tag{2.8}$$

Conservation of momentum in the theory of porous media is described by

$$\rho^\alpha \frac{\mathrm{d}\boldsymbol{v}^\alpha}{\mathrm{d}t} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}^\alpha + \rho^\alpha \boldsymbol{F}_b^\alpha + \boldsymbol{\pi}^\alpha, \tag{2.9}$$

where $\boldsymbol{\sigma}^\alpha$ is the Cauchy stress tensor, $\boldsymbol{F}_b^\alpha$ is the body force of phase α, and $\boldsymbol{\pi}^\alpha$ is the interaction force acting on the phase α from all other phases in the same infinitesimal material parcel. This interaction force will be discussed in greater detail below.

2.1.5 Constitutive relations

The mechanics of the individual phases that comprise the multiphase material are specified by these phases' constitutive relations. Here, we present the relevant relations for this work.

2.1.5.1 Hyperelasticity

General elastic materials whose work is independent of the loading path are called hyperelastic or Green elastic materials. In such cases, there exists a strain energy function, $\psi(\boldsymbol{F})$, which is a single-valued function of the deformation gradient only. This function can be used to derive the first Piola–Kirchhoff stress tensor, $\boldsymbol{P}$,

$$\boldsymbol{P} = \frac{\partial \psi(\boldsymbol{F})}{\partial \boldsymbol{F}}, \tag{2.10}$$

and the Cauchy stress tensor, $\boldsymbol{\sigma}$,

$$\boldsymbol{\sigma} = \frac{1}{J} \boldsymbol{P} \cdot \boldsymbol{F}^T. \tag{2.11}$$

The first Piola–Kirchhoff stress represents the force expressed in the current configuration divided by the area of the material configuration, while the Cauchy stress is calculated at the current configuration for both force and area. Also, the Cauchy stress can be rewritten as a function of the left Cauchy–Green deformation tensor as

$$\boldsymbol{\sigma} = \frac{2}{J}\left[\frac{1}{J^{2/3}}\left(\frac{\partial\psi}{\partial\bar{I}_1} + \bar{I}_1\frac{\partial\psi}{\partial\bar{I}_2}\right)\boldsymbol{B} - \frac{1}{J^{4/3}}\frac{\partial\psi}{\partial\bar{I}_2}\boldsymbol{B}^2\right] + \left[\frac{\partial\psi}{\partial J} - \frac{2}{3J}\left(\bar{I}_1\frac{\partial\psi}{\partial\bar{I}_1} + 2\bar{I}_2\frac{\partial\psi}{\partial\bar{I}_2}\right)\right]\boldsymbol{I}, \tag{2.12}$$

where $\boldsymbol{B}^2 = \boldsymbol{B}\cdot\boldsymbol{B}$, $\bar{I}_1 = J^{-2/3}I_1$, and $\bar{I}_2 = J^{-4/3}I_2$, while I_1 and I_2 are, respectively, the first and second invariant of the left Cauchy–Green deformation tensor. Note that for an incompressible material, $J = 1$.

Soft tissues are usually modeled as hyperelastic material [CKC+04, CKC+07, JSZH09, GLD10, UCB+11, LGD11, UDB+13], due to this model's ability to capture the typical J curve of the stress–strain relation and its easily obtainable fitting parameters. In particular, a Mooney–Rivlin material model [Moo40, Riv48] has been frequently used to model nearly-incompressible and rubber-like materials with the generalized hyperelastic energy function,

$$\psi = \sum_{p,q=0}^{N} C_{pq}(\bar{I}_1 - 3)^p(\bar{I}_2 - 3)^q + \sum_{m=1}^{M} D_m(J-1)^{2m}. \tag{2.13}$$

The most common-used and simplified three coefficients energy function of Mooney-Rivlin material is

$$\psi = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) + D_1(J-1)^2, \tag{2.14}$$

such that the Cauchy stress tensor can be written as

$$\boldsymbol{\sigma} = \frac{2}{J}\left[\left(C_{10} + C_{01}\bar{I}_1\right)\boldsymbol{B} - C_{01}\boldsymbol{B}^2\right] + \left[2D_1(J-1) - \frac{2}{3J}\left(\bar{I}_1C_{10} + 2\bar{I}_2C_{01}\right)\right]\boldsymbol{I}, \tag{2.15}$$

2.1.5.2 Newtonian fluid

In the mechanics of a Newtonian fluid, the Cauchy stress is proportional to the rate of strain,

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\eta\dot{\boldsymbol{\varepsilon}}, \tag{2.16}$$

where η is the fluid viscosity and p is the hydraulic pressure to ensure the incompressible condition.

Interaction Force The interaction force, discussed in detail in de Boer's book [DB06], is written as

$$\boldsymbol{\pi}^{\alpha} = p^{\alpha} \boldsymbol{\nabla} n^{\alpha} - \sum_{\beta} K^{\beta} (\boldsymbol{v}^{\alpha} - \boldsymbol{v}^{\beta}), \tag{2.17}$$

where p^{α} is the isotropic force of each phase, coming from the relation $p^{\alpha} = n^{\alpha} P$ that P plays a role as Lagrangian multiplier to ensure the constraint of equation (2.1), exerting on its own phase in the multiphase continuum and the second term represents the frictional force from other phase in the same material parcel. The coefficient K^{β}, depends on the local volume fraction and represents the degree to which a given velocity difference between the phases produces a friction between them; it is similar to the resistivity in the Stokes drag exerted on objects that move through fluids at low Reynolds number. Here, we assume that the coefficient has the form $K = (1 - n^s)\bar{K}$, where $\bar{K}$ is a constant and is sometimes referred to as the reciprocal of permeability. The $(1 - n^s)$ is local porosity that ensures that the friction term vanishes in a single phase region (where $n^s = 1$).

2.2 Background and Basic Implementation for SPH

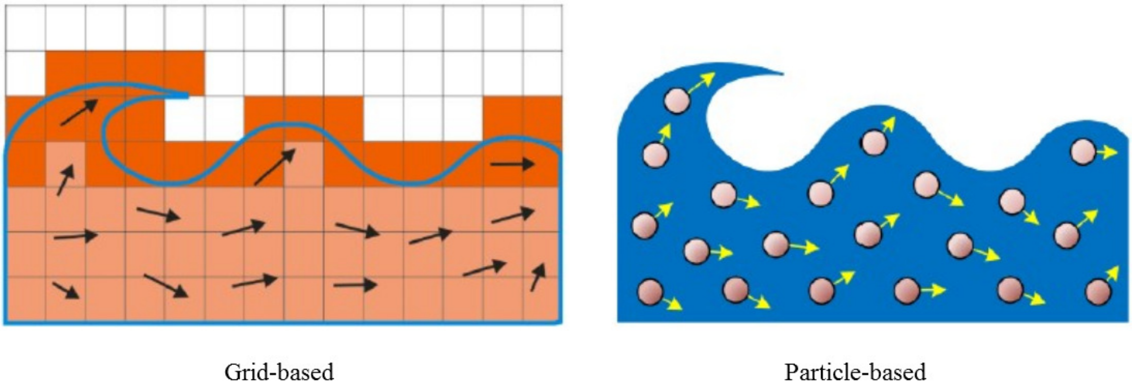


Figure 2.3: Comparison between particle-based method and grid-based method [LBWW18]

Illustrated in figure 2.3 was published in Liu's work [LBWW18], depicting the differences between particle-based method and grid-based method. For grid-based numerical method, we discretize the domain into structure grid as the left figure shown. The physical quantities are calculated on the grid point such that the time derivative at the left hand side in equation 2.8 and equation 2.9 will be modified in material derivative $\frac{D}{Dt}$. On the other hand, in particle-based method, the time dericative at that two equations will be just total time derivative since we track with the particles.

In particular, the method of smoothed particle hydrodynamics (SPH) discretizes the governing equations with a set of advecting computational particles. Each particle in the set carries constant mass and moves with that phase's local velocity. In SPH, any physical field quantity, $f(\boldsymbol{x})$—density, velocity—at specified position, $\boldsymbol{x}$, is approximated by interpolation over the particles as follows,

$$f(\boldsymbol{x}) \approx \sum_j \frac{m_j}{\rho_j} f_j W(\boldsymbol{x} - \boldsymbol{x}_j, h), \tag{2.18}$$

where $\boldsymbol{x}_j$ is the position of particle j, f_j represents the field value associated that particle, and $W(r, h)$ is a smooth, radially-symmetric interpolation (or kernel) function with compact support and support radius h. The ratio m_j/ρ_j represents the phase volume associated with the particle, essentially a finite version of the single-phase parcel volume δV^α discussed earlier. The kernel function has the basic properties

$$\int W(\boldsymbol{x} - \boldsymbol{x}', h)\, \mathrm{d}\boldsymbol{x}' = 1, \tag{2.19}$$

$$\lim_{h \to 0} W(\boldsymbol{x} - \boldsymbol{x}', h) = \delta(\boldsymbol{x} - \boldsymbol{x}'). \tag{2.20}$$

The first property ensures that the integral of the field is preserved by the summation over the particles; the second identifies the kernel as an approximate Dirac measure, shrinking its support as the particle density increases. The support radius, h, represents the radius of a spherical region surrounding the particle centroid; particles in this region have computational influence on the evaluation of a field quantity at this centroid. Thus, each particle has an evolving set of neighboring particles that must be identified by radius searching. The searching radius should not be unreasonable large due to stability requirement but have to choose an appropriate value to ensure enough particles in the searching region. The searching radius also determines the mass of each particle via equally allocating the total mass in the region to each particle inside it. The gradient of field $f(\boldsymbol{x})$ is easily computed by applying the gradient to the kernel:

$$\boldsymbol{\nabla} f(\boldsymbol{x}) \approx \sum_j \frac{m_j}{\rho_j} f_j \boldsymbol{\nabla} W(\boldsymbol{x} - \boldsymbol{x}_j, h). \tag{2.21}$$

A higher accuracy and symmetric version of the gradient between particles can be derived from the product rule,

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \frac{1}{\rho} \left(\boldsymbol{\nabla}(\rho f(\boldsymbol{x})) - f(\boldsymbol{x}) \boldsymbol{\nabla} \rho \right), \tag{2.22}$$

or

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \rho \left(\boldsymbol{\nabla} \left(\frac{f(\boldsymbol{x})}{\rho} \right) + \frac{f(\boldsymbol{x})}{\rho^2} \boldsymbol{\nabla} \rho \right), \tag{2.23}$$

such that the gradient is written as

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \frac{1}{\rho_i} \sum_j (f(\boldsymbol{x_j}) - f(\boldsymbol{x_i})) m_j \boldsymbol{\nabla} W(\boldsymbol{x} - \boldsymbol{x}_j, h), \tag{2.24}$$

or

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \rho_i \sum_j \left(\frac{f(\boldsymbol{x}_j)}{\rho_j^2} + \frac{f(\boldsymbol{x}_i)}{\rho_i^2} \right) m_j \boldsymbol{\nabla} W(\boldsymbol{x} - \boldsymbol{x}_j, h), \tag{2.25}$$

where the former equation is usually used as we want to compute strain tensor, while the latter equation is applied to the formulation of pressure or stress gradient.

Higher derivatives, such as the Laplacian, are calculated analogously, written as

$$\nabla^2 f(\boldsymbol{x}_i) = \sum_j \left(f(\boldsymbol{x}_j) - f(\boldsymbol{x}_i) \right) \frac{m_j}{\rho_j} \nabla^2 W(\boldsymbol{x} - \boldsymbol{x}_j, h). \tag{2.26}$$

It is useful to note that the radial symmetry of the kernel ensures that

$$\boldsymbol{\nabla} W(\boldsymbol{x} - \boldsymbol{x}', h) = -\boldsymbol{\nabla}' W(\boldsymbol{x} - \boldsymbol{x}', h), \tag{2.27}$$

where $\boldsymbol{\nabla}'$ denotes the gradient with respect to $\boldsymbol{x}'$. The symmetry also implies that

$$\int \boldsymbol{\nabla} W(\boldsymbol{x} - \boldsymbol{x}', h) \, \mathrm{d}\boldsymbol{x}' = 0, \tag{2.28}$$

2.2.1 Kernel Functions

Kernel functions will highly affect stability, accuracy and speed of the SPH method. In this subsection, all kernel functions that used in our implementation will be introduced here and the kernel function curve, and its curve undergoing gradient and Laplacian operator will also be displayed by different colors under the assumption with unity of searching/support radius in one dimensional case. Basically, the kernel function is a kind of interpolation function, so that it should be always positive everywhere and own the most weighting magnitude at the center just like Dirac function. The gradient of the kernel function is a vector that always point out to the center.

The cubic spline kernel: The cubic spline kernel function, also known as B-spline function, is a common used function when SPH method was applied to solid mechanics [LP91, OVSM98, Mon00]. It provides some basic properties like Gaussian function but the second derivative of the cubic spline is a piecewise linear function, which resulting in stability issue in some cases [LL10]. The curves of the cubic spline kernel function is shown in figure 2.4. The blue, red and yellow curves represent the original kernel function, gradient of kernel function and the Laplacian of it, respectively.

$$W(\boldsymbol{r},h) = \alpha_0 \begin{cases} 1 - \frac{3}{2}(\frac{|\boldsymbol{r}|}{h})^2 + \frac{3}{4}(\frac{|\boldsymbol{r}|}{h})^3, & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ \frac{1}{4}(2 - \frac{|\boldsymbol{r}|}{h})^3, & 1 \leq \frac{|\boldsymbol{r}|}{h} < 2 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 2 \end{cases} \tag{2.29}$$

$$\boldsymbol{\nabla} W(\boldsymbol{r},h) = -\frac{\alpha_0}{h}\frac{\boldsymbol{r}}{|\boldsymbol{r}|} \begin{cases} 3(\frac{|\boldsymbol{r}|}{h}) - \frac{9}{4}(\frac{|\boldsymbol{r}|}{h})^2, & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ \frac{3}{4}(2 - \frac{|\boldsymbol{r}|}{h})^2, & 1 \leq \frac{|\boldsymbol{r}|}{h} < 2 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 2 \end{cases} \tag{2.30}$$

$$\nabla^2 W(\boldsymbol{r},h) = \frac{\alpha_0}{h^2} \begin{cases} 9(\frac{|\boldsymbol{r}|}{h} - 1), & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ \frac{3}{2}(2 - \frac{|\boldsymbol{r}|}{h})(1 - \frac{h}{|\boldsymbol{r}|}), & 1 \leq \frac{|\boldsymbol{r}|}{h} < 2 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 2 \end{cases} \tag{2.31}$$

where $\alpha_0 = 2/(3h)$, $10/(7\pi h^2)$, $1/(\pi h^3)$ in 1, 2 and 3 dimensions, respectively.

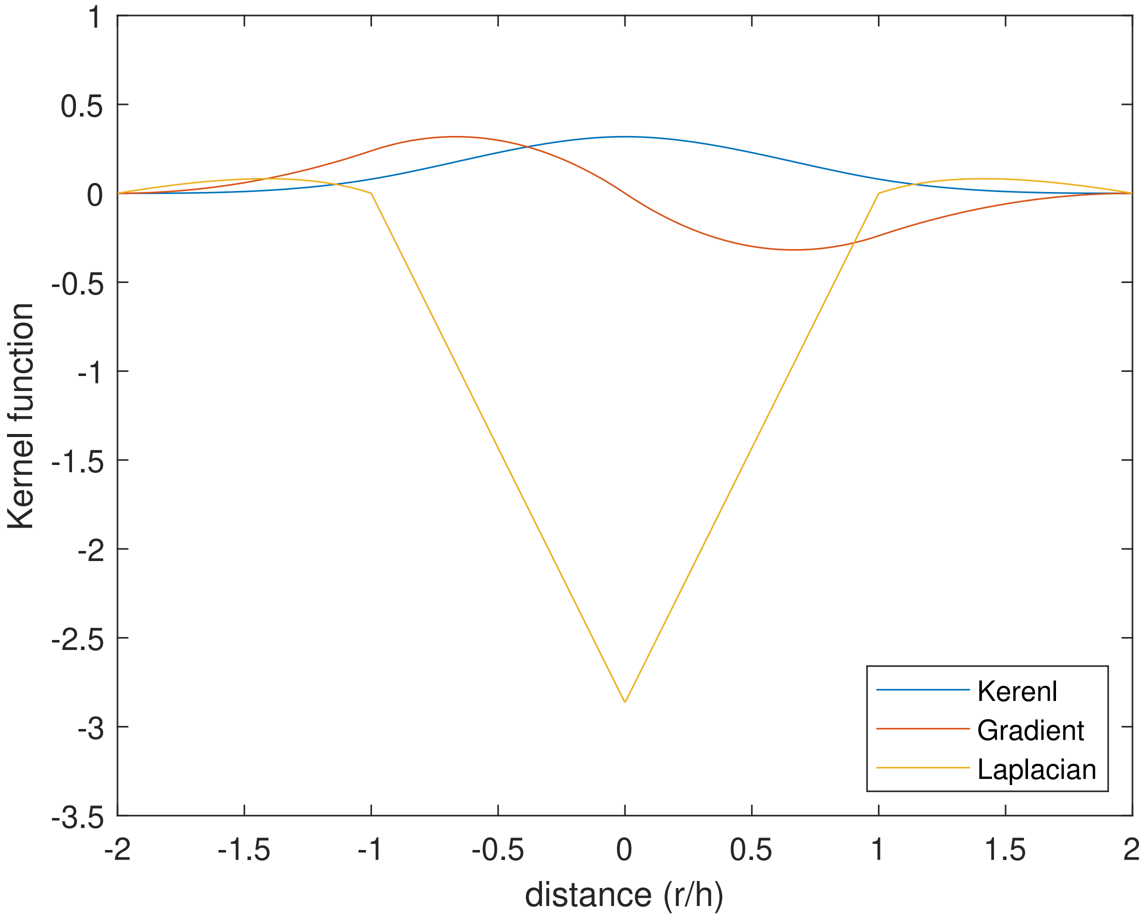


Figure 2.4: The cubic spline kernel function (blue) and its gradient (red), Laplacian (yellow).

The third degree polynomial kernel: The third degree polynomial kernel was proposed in [MCG03] and applied in following works [Kel06, CJR$^+$17] as a kernel function for viscous force formulation. The kernel ensures the property that relative velocity between two particles is the only effect to determine the strength of viscous term rather than getting unrealistic damping with negative resulting in forces that increase their relative velocity. Also, in coarsely sampled velocity fields, the kernel can improve the stability issue significantly [MCG03, Kel06]. The curves of the third degree polynomial kernel function is shown in figure 2.5. The blue, red and yellow curves represent the original kernel function, gradient of kernel function and the Laplacian of it, respectively. Note that the zero distance is a singular point approaching infinity but in the viscous force formulation, particle will not act with itself.

$$W(\boldsymbol{r},h) = \alpha_0 \begin{cases} -\frac{1}{2}(\frac{|\boldsymbol{r}|}{h})^3 + (\frac{|\boldsymbol{r}|}{h})^2 + \frac{1}{2}(\frac{h}{|\boldsymbol{r}|}) - 1, & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 1 \end{cases} \tag{2.32}$$

$$\boldsymbol{\nabla} W(\boldsymbol{r},h) = \frac{\alpha_0}{h}\frac{\boldsymbol{r}}{h} \begin{cases} (-\frac{3|\boldsymbol{r}|}{2h} + 2 - \frac{h^3}{2|\boldsymbol{r}|}), & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 1 \end{cases} \tag{2.33}$$

$$\nabla^2 W(\boldsymbol{r},h) = \frac{6\alpha_0}{h^2} \begin{cases} 1 - \frac{|\boldsymbol{r}|}{h}, & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 1 \end{cases} \tag{2.34}$$

where $\alpha_0 = 10/(3\pi h^2)$, $15/(2\pi h^3)$ in 2 and 3 dimensions, respectively.

The sixth degree polynomial kernel: The sixth degree polynomial kernel function is a function to mimic the Gaussian bell curve in high order polynomial approximation. Note that the Gaussian kernel has a good mathematical properties but owns exponential function that causes expensive computation compared with simple arithmetic. Therefore, the sixth degree polynomial kernel function is a good replacement that served as the most general kernel function in SPH method except for internal fluid force field. The curves of the sixth degree polynomial kernel function is shown in figure 2.6. The blue, red and yellow curves represent the original kernel function, gradient of kernel function and the Laplacian of it, respectively.

$$W(\boldsymbol{r},h) = \alpha_0 \begin{cases} (1 - (\frac{|\boldsymbol{r}|}{h})^2)^3, & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 1 \end{cases} \tag{2.35}$$

$$\boldsymbol{\nabla} W(\boldsymbol{r},h) = \frac{6\alpha_0}{h}\frac{\boldsymbol{r}}{h} \begin{cases} (1 - (\frac{|\boldsymbol{r}|}{h})^2)^2, & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 1 \end{cases} \tag{2.36}$$

$$\nabla^2 W(\boldsymbol{r},h) = \frac{6\alpha_0}{h^2} \begin{cases} (1 - (\frac{|\boldsymbol{r}|}{h})^2)(3 - 7(\frac{|\boldsymbol{r}|}{h})^2), & 0 \leq \frac{|\boldsymbol{r}|}{h} < 1 \\ 0, & \frac{|\boldsymbol{r}|}{h} \geq 1 \end{cases} \tag{2.37}$$

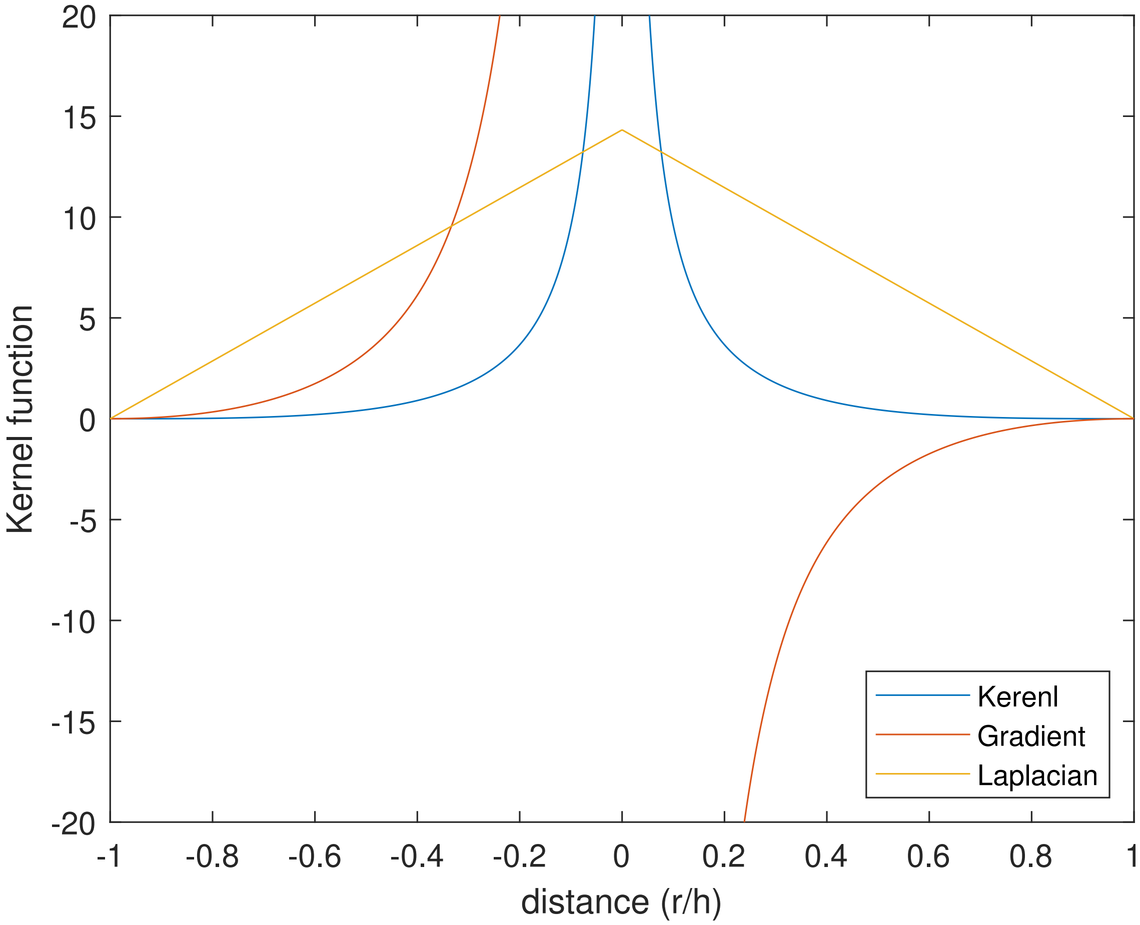


Figure 2.5: The third degree polynomial kernel function (blue) and its gradient (red), Laplacian (yellow).

where $\alpha_0 = 35/(16h)$, $4/(\pi h^2)$, $315/(64\pi h^3)$ in 1, 2 and 3 dimensions, respectively.

The spiky kernel: The spiky kernel is chosen as the kernel function for pressure force because the ability to prevent clustering in high pressure region. We can observe that for other types of kernels, the gradient of kernel function is approaching zero as the distance between two particles becomes smaller, which means the repulsive force vanishes. However, in real word, or in numerical sense, any two particles that being too close should suffer large amount of repelling force from collapsing. Therefore, in the spiky kernel function, we exaggerate the magnitude of force as the distance approaches zero. The curves of the spiky kernel function is shown in figure 2.4. The blue, red and yellow curves represent the original

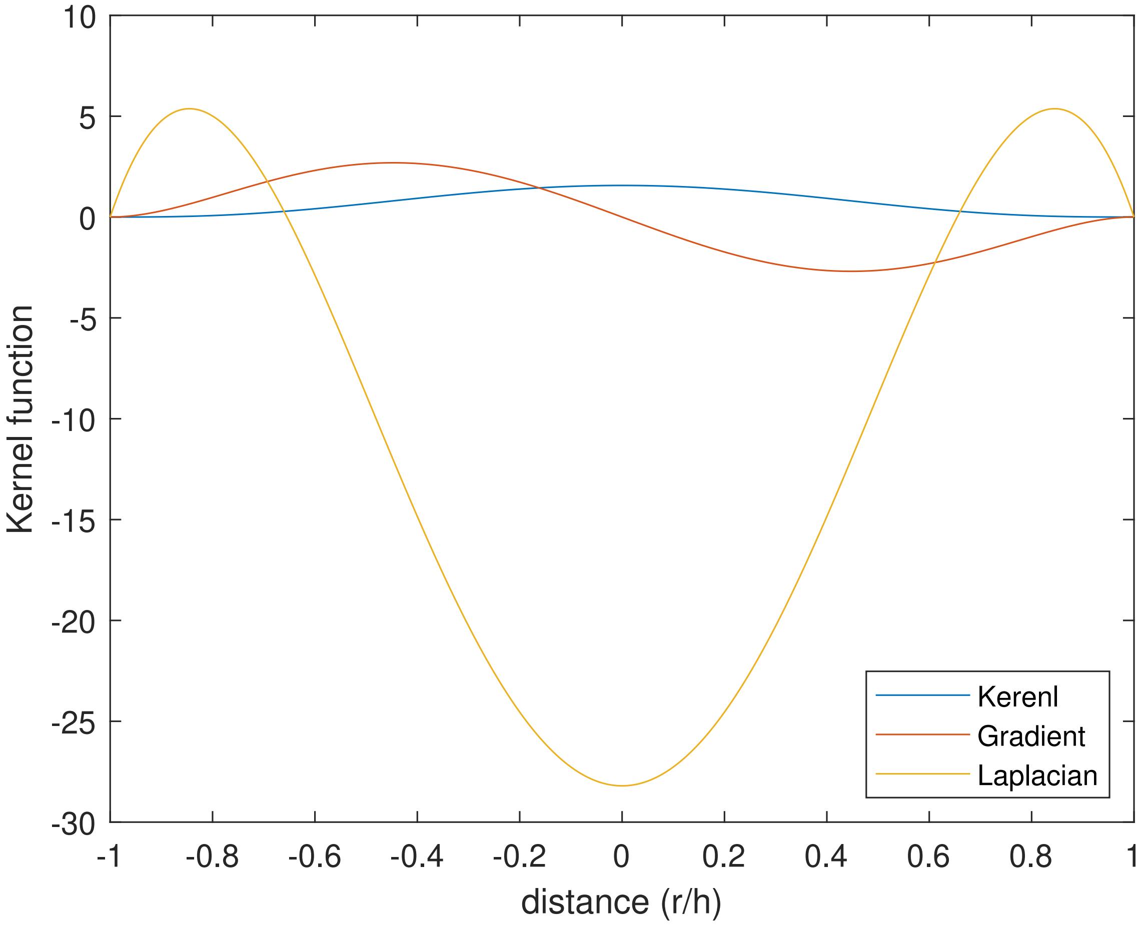


Figure 2.6: The sixth degree polynomial kernel function (blue) and its gradient (red), Laplacian (yellow).

kernel function, gradient of kernel function and the Laplacian of it, respectively.

$$W(\boldsymbol{r},h)=\alpha_0\begin{cases}(1-\frac{|\boldsymbol{r}|}{h})^3, & 0\leq\frac{|\boldsymbol{r}|}{h}<1\\ 0, & \frac{|\boldsymbol{r}|}{h}\geq 1\end{cases}\tag{2.38}$$

$$\boldsymbol{\nabla}W(\boldsymbol{r},h)=-\frac{3\alpha_0}{h}\frac{\boldsymbol{r}}{|\boldsymbol{r}|}\begin{cases}(1-\frac{|\boldsymbol{r}|}{h})^2, & 0\leq\frac{|\boldsymbol{r}|}{h}<1\\ 0, & \frac{|\boldsymbol{r}|}{h}\geq 1\end{cases}\tag{2.39}$$

$$\nabla^2W(\boldsymbol{r},h)=-\frac{4\alpha_0}{h^2}\begin{cases}\frac{h}{|\boldsymbol{r}|}(1-\frac{|\boldsymbol{r}|}{h})(1-\frac{2|\boldsymbol{r}|}{h}), & 0\leq\frac{|\boldsymbol{r}|}{h}<1\\ 0, & \frac{|\boldsymbol{r}|}{h}\geq 1\end{cases}\tag{2.40}$$

where $\alpha_0=4/h$, $10/(\pi h^2)$, $15/(\pi h^3)$ in 1, 2 and 3 dimensions, respectively.

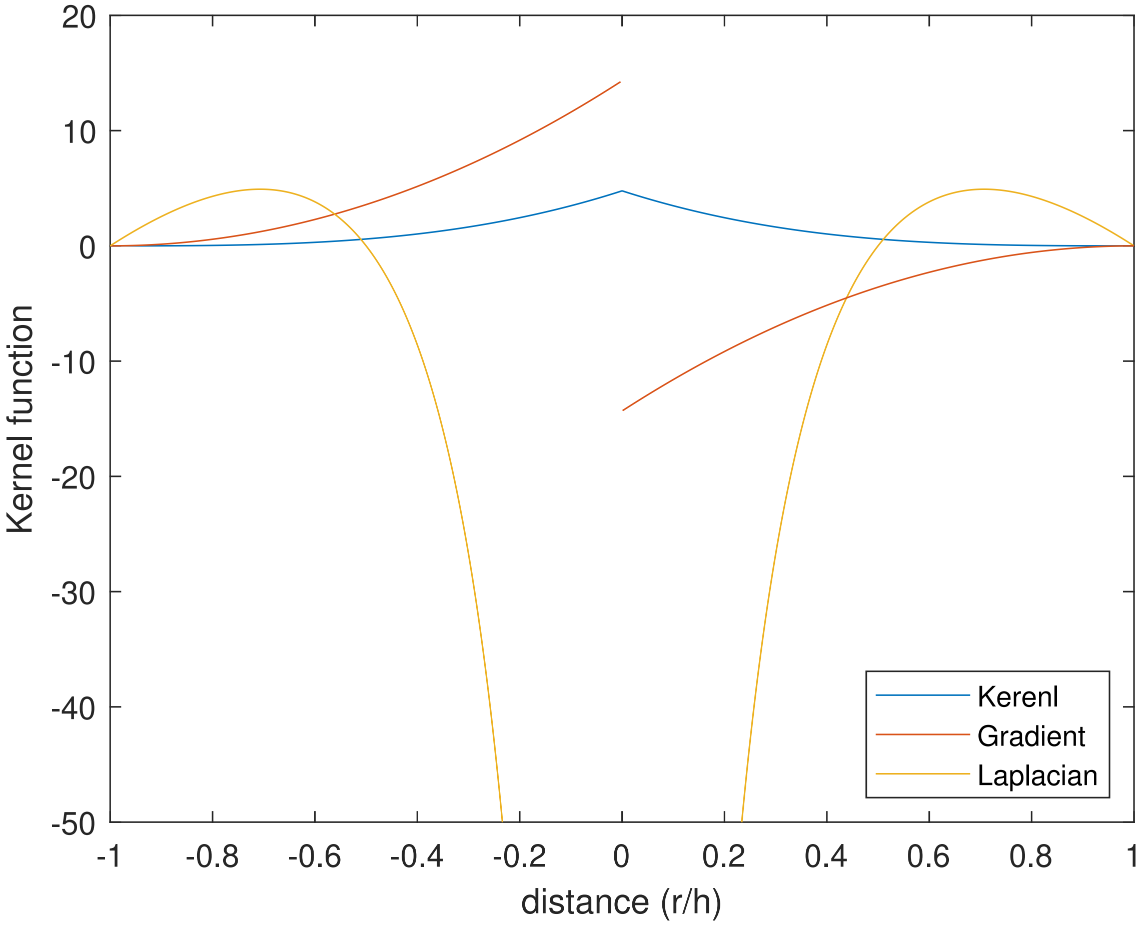


Figure 2.7: The spiky kernel function (blue) and its gradient (red), Laplacian (yellow).

2.2.2 Kinematic calculations

The SPH formalism (2.18) and its gradient (2.24) provide the necessary tools to evaluate the complete set of kinematic field quantities in each phase. At each time step, we will have the position and velocity of every particle available. From these data, we can calculate the material deformation and its rates of change. For example, the updated Lagrangian formulation (2.5) can be applied at particle i to advance the deformation gradient from step $n-1$ to n:

$$\boldsymbol{F}_i^{<n>} = \hat{\boldsymbol{F}}_i \boldsymbol{F}_i^{<n-1>}. \tag{2.41}$$

This update requires that we compute the deformation gradient in the current configuration,

$$\hat{\boldsymbol{F}}_i = \boldsymbol{I} + \boldsymbol{\nabla} \boldsymbol{u}_i, \tag{2.42}$$

for which we only need to calculate the local displacement gradient. This task can be carried out over the set of particles,

$$\boldsymbol{\nabla u}_i = \sum_j \frac{m_j}{\rho_j}(\mathrm{d}\boldsymbol{x}_j - \mathrm{d}\boldsymbol{x}_i)^T \boldsymbol{\nabla} W(\boldsymbol{x}_i - \boldsymbol{x}_j, h), \tag{2.43}$$

where $\mathrm{d}\boldsymbol{x}_i = \boldsymbol{x}_i^{<n>} - \boldsymbol{x}_i^{<n-1>}$ is the particle displacement over the previous time step.

The rate of strain and rate of rotation are calculated from the gradient of the SPH-approximated velocity field,

$$\dot{\boldsymbol{\varepsilon}}_i = \frac{1}{2}\sum_j \frac{m_j}{\rho_j}[(\boldsymbol{v}_j - \boldsymbol{v}_i)^T \boldsymbol{\nabla} W + \boldsymbol{\nabla} W^T(\boldsymbol{v}_j - \boldsymbol{v}_i)], \tag{2.44}$$

and

$$\dot{\boldsymbol{R}}_i = \frac{1}{2}\sum_j \frac{m_j}{\rho_j}[(\boldsymbol{v}_j - \boldsymbol{v}_i)^T \boldsymbol{\nabla} W - \boldsymbol{\nabla} W^T(\boldsymbol{v}_j - \boldsymbol{v}_i)], \tag{2.45}$$

respectively.

2.2.3 SPH Discretization of Governing Equations

In SPH, global conservation of mass is automatically satisfied because each particle carries its own mass with constant value. The local density can be found from equation (2.18), dependent on the summation of mass over neighboring particles with the weighting kernel function,

$$\rho_i(\boldsymbol{x}) \approx \sum_j m_j W(\boldsymbol{x} - \boldsymbol{x}_j, h). \tag{2.46}$$

Another way to obtain local density is to solve equation (2.8) as

$$\frac{\mathrm{d}\rho_i}{\mathrm{d}t} = \sum_j m_j(\boldsymbol{v}_i - \boldsymbol{v}_j) \cdot \boldsymbol{\nabla} W(\boldsymbol{x_i} - \boldsymbol{x}_j, h). \tag{2.47}$$

Usually, the equation (2.46) is applied to liquid phase while the equation (2.47) is applied to solid phase.

For the discretization of momentum equation (2.9), we require Newton's third law for action and reaction to hold in the discrete sense. Thus, to ensure that two particles exert equal and opposite stress on each other, it is typical to rely on the equation (2.25) and the anti-symmetry of the kernel, equation (2.27), to write the momentum equation as

$$\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \sum_j m_j \left(\frac{\boldsymbol{\sigma}_i}{\rho_i^2} + \frac{\boldsymbol{\sigma}_j}{\rho_j^2} + \boldsymbol{R}_{ij} f_{ij}^n + \boldsymbol{\Pi}_{ij} \right) \cdot \boldsymbol{\nabla} W(\boldsymbol{x}_i - \boldsymbol{x}_j, h) + \boldsymbol{F_{ib}} + \frac{\boldsymbol{\pi}_i}{\rho_i}, \tag{2.48}$$

where $\boldsymbol{R}_{ij} f_{ij}^n$, $\boldsymbol{\Pi}_{ij}$ are different forms of artificial dissipation to remove, respectively, the tensile instability/particle clumping and unphysical high frequency vibration when applying SPH method into solid phase. Details will be explained later.

The stress and the interaction force require constitutive equations. The isotropic and deviatoric parts of the stress are treated separately,

$$\boldsymbol{\sigma}^\alpha = \boldsymbol{\tau}_{iso}^\alpha + \boldsymbol{\tau}_{dev}^\alpha, \tag{2.49}$$

where $\boldsymbol{\tau}_{iso}^\alpha = -n_i^\alpha p_i^\alpha \boldsymbol{I}$ is from [DB06] and would reduce back to Lagrangian multiplier that ensure incompressibility if we deal with single phase problem($n^\alpha = 1$).

SPH method on hyperelastic material has already been investigated by Kawashima [KS07]. In his implementation, an equivalent two steps predictor-corrector method was utilized. The incompressible constraint was enforced at the second step by solving the pressure Poisson equation. An comparison between his implementation and the finite element results was shown in the paper. Here, we followed his idea on the updated Lagrangian formulation to describe hyperelastic material motion as shown from equation (2.41) to equation (2.43). Then compute only the deviatoric part (i.e. $\frac{\partial \psi}{\partial J} = 0$) in the Cauchy stress tensor of each particle via equation (2.12). The volumetric stress tensor is modeled in equation 2.52. From this manner, we avoid solving Poisson equation on the pressure term and only account for isotopic force once instead of calculating it twice on the pressure force of the correction step and the volumetric stress in the Cauchy stress tensor.

Regarding to the SPH on Newtonian fluid, equation (2.16), we can further decompose the stress tensor in equation (2.50) and rewrite it as

$$\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \sum_{i \neq j} m_j \left(\frac{p_i^l}{\rho_i^2} + \frac{p_j^l}{\rho_j^2} \right) \boldsymbol{\nabla} W_{ij} + \frac{\mu}{\rho_i} \sum_{i \neq j} m_j (\boldsymbol{v}_j - \boldsymbol{v}_i) \nabla^2 W_{ij} + \boldsymbol{F_{ib}} + \frac{\boldsymbol{\pi}_i}{\rho_i}, \tag{2.50}$$

where the kernel functions of pressure force and viscous force are different as mentioned before.

2.2.3.1 Isotropic Stress

In single-phase SPH, it is common to treat incompressible materials as weakly compressible, so that instead of enforcing the incompressibility constraint strongly, the pressure is obtained from an isothermal equation of state linking two thermodynamic variables: the density and the pressure. The most simple relation for the single phase problem is written as

$$p_i = k\left(\frac{\rho_i}{\rho_0} - 1\right) = k(n_i - 1), \tag{2.51}$$

where the k is a numerical parameter representing the gas stiffness, related to the material's speed of sound. In such formulation, deviations of the local volume fraction n_i from unity are penalized, so that if the particles are locally dense or sparse, the pressure will provide a repelling or attractive force, respectively, to maintain homogeneity of the local density. The basic concept of this comes from the intrinsic physical role of pressure for propagating any density disturbance at the speed of sound. This speed is infinite in an incompressible medium. In the numerical context, a large speed of sound requires a small time step size, so the numerical speed of sound is artificially reduced to ensure a computationally tractable algorithm. This treatment results in weakly compressible representations of each material.

The treatment of isotropic stress we propose in the biphasic problem (liquid and solid) draws inspiration from the penalization idea implied in the equation of state (2.51). For the solid material, we propose an isotropic stress of the form

$$p_i^s = k_s(n_i^s - n_{\text{sat}}^s) + k_{\text{sat}}(n_i^s + n_i^l - 1), \tag{2.52}$$

where n_{sat}^s is a material constant representing the solid volume fraction at the equilibrium state of liquid perfusion and k_s, k_{sat} are two new gas stiffnesses in the biphasic problem. The first term on the right-hand side represents the structural behavior of the solid and the second term enables the isotropic stress to try to preserve the saturated condition of both phases (2.1). The equilibrium volume fractions of the solid material are implied in the equation (2.52) by letting P_i^s go to zero in both full perfusion and dry conditions. In the fully perfused case, $n_i^s + n_i^l$ is unity and thus, $n_i^s = n_{\text{sat}}^s$. In the dry case, there are no liquid particles nearby, implying that $n_i^l = 0$ so that the lowest limit of n_i^s is $(k_s n_{\text{sat}}^s + k_{\text{sat}})/(k_s + k_{\text{sat}})$.

In this case, the sum of volume fractions $n_i^s + n_i^l = n_i^s$ is less than or equal to unity, implying that there exists a void inside the material parcel, as one would expect for a dry material without an explicit representation of the gas phase. The range of solid volume fractions in general circumastances is $(k_s n_{\text{sat}}^s + k_{\text{sat}})/(k_s + k_{\text{sat}}) \leq n_i^s \leq n_{\text{sat}}^s$.

The formulation of isotropic force in the liquid is much simpler than in the solid because of the lack of structural pores in the liquid. (We do not treat bubbly flows in this work.) The isotropic stress in the liquid is proposed to be

$$p_i^l = k_{\text{sat}}(n_i^s + n_i^l - 1). \tag{2.53}$$

Note that it is wise to combine all the isotropic force from material constitutive relation and that from interaction force in momentum equation via $-\boldsymbol{\nabla}(n^\alpha p^\alpha) + p^\alpha \boldsymbol{\nabla} n^\alpha = -n^\alpha \boldsymbol{\nabla} p^\alpha$.

2.2.3.2 Artificial Dissipation

In SPH method, artificial dissipation is essential since the numerical pressure does not act as perfect as the real pressure that is able to remove all small density disturbance instantly, causing unphysical vibration upon the particles. Furthermore, if local numerical pressure is too small, it will result in large attraction force toward neighboring particle and then induce particle clumping. Artificial dissipation here is designed to prevent both of these two effects. First, the term Π_{ij} is the first term being investigated to remove high frequency vibration. The common used form was proposed by Monaghan and his colleagues [MG83, MP85], allowing the feasibility with the shock front simulation to convert kinetic energy into heat, formulating as

$$\Pi_{ij} = \begin{cases} \frac{-\alpha_d c \mu_d + \beta_d \mu_d^2}{\bar{\rho_{ij}}}, & \boldsymbol{v_{ij}} \cdot \boldsymbol{x_{ij}} < 0 \\ 0, & \boldsymbol{v_{ij}} \cdot \boldsymbol{x_{ij}} \geq 0 \end{cases}, \tag{2.54}$$

where

$$\mu_d = \frac{h(\boldsymbol{v_{ij}} \cdot \boldsymbol{x_{ij}})}{|x_{ij}|^2 + \gamma_d h^2}, \tag{2.55}$$

$$\bar{\rho_{ij}} = \frac{1}{2}(\rho_i + \rho_j), \tag{2.56}$$

$$\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j, \tag{2.57}$$

$$\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j, \tag{2.58}$$

c is the numerical speed of sound, and $\alpha_d, \beta_d, \gamma_d$ are some parameters suggested as $\alpha_d = 1, \beta_d = 1, \gamma_d = 0.01$ respectively. Note that here this dissipation only affect as $\boldsymbol{v_{ij}} \cdot \boldsymbol{x_{ij}} < 0$, which means the dissipation force only activate as two particles effectively tend to move toward each other.

The another term $\boldsymbol{R}_{ij} f_{ij}$ in the equation (2.9) follows from these literature [Mon00, GMS01, LL03]. The tensor $\boldsymbol{R}_i j$ is determined by $\boldsymbol{R}_{ij} = \boldsymbol{R}_i + \boldsymbol{R}_j$, with

$$\boldsymbol{R}_i = -\epsilon \boldsymbol{\sigma}_i, \tag{2.59}$$

for all positive component in the stress tensor $\boldsymbol{\sigma}$; otherwise, the component with negative value in stress tensor is set as zero in corresponding component in tensor $\boldsymbol{R}_i$. The ϵ is a small number, representing the strength of dissipation. The suggestion value of n, ϵ were mentioned in [Mon00, GMS01], depending on the ratio of searching radius and the particle spacing Δx.

The term f_{ij} is defined as

$$f_{ij} = \frac{W(\boldsymbol{x}_i - \boldsymbol{x}_j)}{W(\Delta x)}, \tag{2.60}$$

where Δx is particle spacing and n is the exponent of f_{ij}. The parameter n in the first term was set to 4 throughout; the strength of dissipation was tuned for each case, according to the suggestions in [Mon00, GMS01]. Basically, the magnitude of $\boldsymbol{R}$ is related to the local attraction isotropic force and utilized here to ease the effect of negative pressure force, preventing tendency of particle clumping. It also weighted by the f_{ij}^n, which provided strong amplification as two particles are too close each other.

2.2.4 Numerical Parameters

There are some numerical parameters to be determined before running the simulation. Here, some guidance of how to choose appropriate parameters is listed below.

The size of support radius has to be determined before setting any other parameters. It should be proportional to the particle spacing Δx and the ratio is usually between 1 $\sim$

2 (i.e. $2\Delta x \geq h \geq \Delta x$). The precise ratio is chosen by its corresponding number of neighboring particles. In liquid simulation, the number of neighboring particles is usually suggested between of $25 \sim 40$, while in solid simulation, the number of it is usually between $50 \sim 70$. Note that the cubic kernel function is the most common weighting function applied in solid problem, which utilizes twice support radius as the length of searching radius. In consequence, the most common ratio between support radius and particle spacing is $h = 1.3\Delta x$, which is able to fulfil the resolution requirement of number of neighboring particles in both phases.

Once the support radius is determined, the next step is to decide the mass of each particle. Due to the consideration of stability, we must choose a particle mass such that the density calculation by equation 2.46 is close to the material reference density at the initial stage. The mass is proportional to the cubed of the particle spacing in three dimensional problem, written as $m = C\Delta x^3$, where C is a constant usually between $1 \sim 10$, determined by the equation 2.46.

The gas stiffness in the numerical pressure formulation is a special parameter in SPH method. In real world, it should approach infinity if the material is incompressible. In SPH, we ease the constraint and then gas stiffness reduces to finite number depending on how incompressible we want to enforce on the simulation. It is also an important parameter relating to the time step size. In the thermodynamic point of view, the gas stiffness is just a square of the speed of sound in the material, written as $k = c_0^2$. The speed of sound appears in the artificial dissipation as well to damp out high frequency vibration. The common number of c_0 for liquid problem is between $1 \sim 5$ while that in solid problem is between $3 \sim 10$. Note that the speed of sound in solid material is faster it in liquid phase.

2.2.5 Multiphase SPH

In the multiphase SPH method, the basic SPH formulation is applied separately to each phase of material. Each material is discretized with its own set of particles. Particles from the same set are precluded from occupying the same position, but particles from different sets are allowed to coincide without explicit interference. A physical quantity of any phase can be recovered at a location by summing over the nearby particles associated with that phase. The evolution of each phase by equation (2.9) requires some information about the other phase(s), such as the local volume fraction and the phase-specific velocity. Thus, in order to calculate the behavior of a given particle in a particular phase's set, radius searching and summation are done separately over each particle set. For example, in the biphasic problem in the focus of this work, liquid and solid phases are discretized into two separates set of particles. Four neighboring particle index tables should be obtained in the radius searching step: indexes of solid/liquid particles at the searching centroid of both solid and liquid particle position. After having the neighboring particle indexes, all information required to obtain the force in momentum equation are ready for computing. The basic steps of the algorithm are summarized in Algorithm 1.

Algorithm 1 Biphasic SPH algorithm

```
Initialize particle sets of liquid/solid position $\boldsymbol{x}^l, \boldsymbol{x}^s$
Assign initial value of solid density $\rho^s$
while animating do
  for all $\boldsymbol{x}^l, \boldsymbol{x}^s$ do
    Radius searching of neighboring particles
  for all $\boldsymbol{x}^l$ do
    Compute density $\rho^l, d\rho^s/dt$, volume fraction $n^l, n^s$ and solid velocity
    Compute force $\boldsymbol{F}^{p,\pi,\nu}$ (r.h.s. in equation (2.9))
  for all $\boldsymbol{x}^s$ do
    Compute density $\rho^l, d\rho^s/dt$, volume fraction $n^l, n^s$ and liquid velocity
    Update deformation gradient $\boldsymbol{F}^{<n>} = \hat{\boldsymbol{F}} \cdot \boldsymbol{F}^{<n-1>}$
    Compute force $\boldsymbol{F}^{p,\pi,\text{hyper}}$ (r.h.s. in equation (2.9))
  for all $\boldsymbol{x}^l, \boldsymbol{x}^s$ do
    Advance $\boldsymbol{x}^l, \boldsymbol{x}^s$
    Evolve $\rho^s$
```

2.2.6 Volume/area representation by particles

The volume/area representation by particles is very different from grid. It is very straightforward that the distance between grid point is the side length in grid-based discretization and then the volume/area can be obtained corresponding to the side length. On the other hand, in particle method, all particles represent a concentration of neighboring mass with their searching radius. This concept is depicting in the figure 2.8, where the blue box represents the area/volume in the grid-based discretization, while the green box is actually the real volume/area represented by the particles. This effective area with extra length is definitely relating to the length of searching radius in SPH method, which is really important correlation as we validate our solver in the chapter 4.

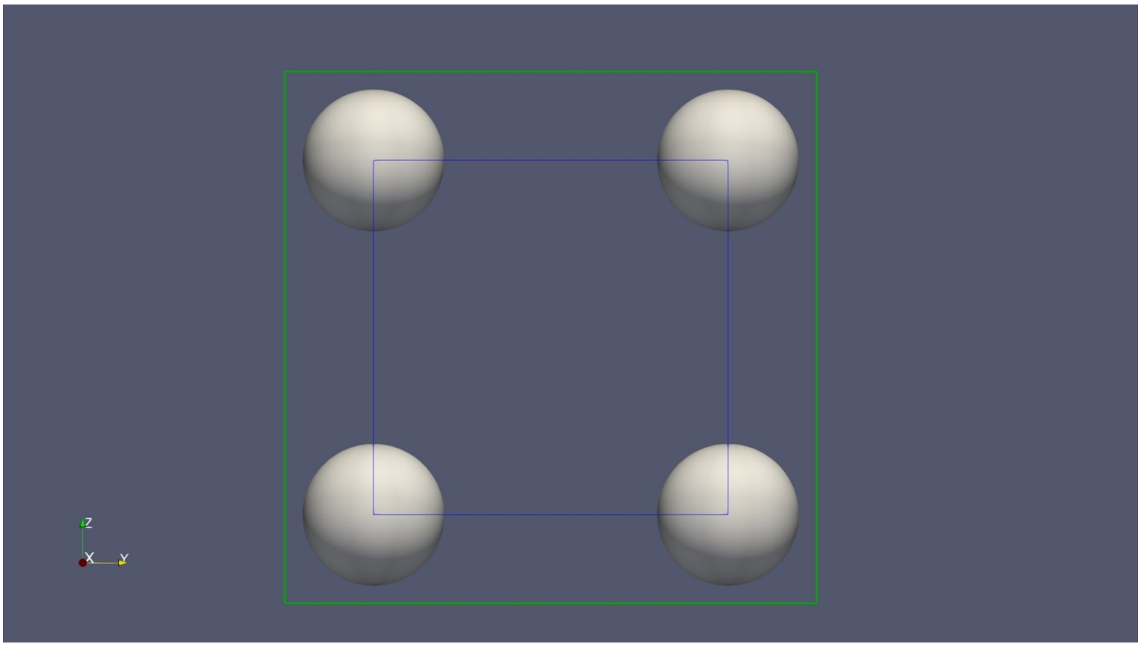


Figure 2.8: Effective area representation by the SPH method (green) and the grid-based method (blue)

2.2.7 Boundary Treatment

No-penetration, no-slip boundary conditions, such as at a wall, a tank, or a piston, are achieved in our implementation by placing virtual particles at the boundary. The positions of these virtual particles are assigned at each time step, with prescribed values of density, pressure, and velocity. As material particles pass nearby, they are affected by the fields contributed from the virtual particles. These contributions also improve the resolution deficiency of the material particles near the boundary. Sometimes, to prevent the leakage of liquid at the boundary, the particle spacing for the virtual particles may be much higher than it for the liquid particles. It should be determined case by case.

2.2.8 Time Scheme

We use a predictor-corrector leapfrog method for time integration. For the equations with time evolving quantities: velocity $\boldsymbol{v}$, position $\boldsymbol{r}$, density ρ,

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = \boldsymbol{F}, \tag{2.61}$$

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{v}, \tag{2.62}$$

and

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = D, \tag{2.63}$$

we first calculate the right hand side of above equations and denoted as superscript 0 and then find the predicting state as

$$\boldsymbol{v}^p = \boldsymbol{v}^0 + \Delta t \boldsymbol{F}^0, \tag{2.64}$$

$$\boldsymbol{r}^p = \boldsymbol{r}^0 + \Delta t \boldsymbol{v}^0 + \frac{1}{2}(\Delta t)^2 \boldsymbol{F}^0, \tag{2.65}$$

and

$$\rho^p = \rho^0 + \Delta t D^0. \tag{2.66}$$

Then we correct the value of velocity $\boldsymbol{v}$ and density ρ by recalculating the right hand side of the equations (2.61), (2.62), and (2.63)

$$\boldsymbol{v} = \boldsymbol{v}^p + \frac{1}{2}\Delta t(\boldsymbol{F} - \boldsymbol{F}^0), \tag{2.67}$$

$$\rho = \rho^p + \frac{1}{2}\Delta t(D - D^0), \tag{2.68}$$

Sometimes, the velocity in equation 2.62 will be replaced by $\widetilde{\boldsymbol{v}}$, defined as

$$\widetilde{\boldsymbol{v}}_i = \boldsymbol{v}_i + \widetilde{\epsilon} \sum_j \frac{m_j}{\rho_{ij}} (\boldsymbol{v}_j - \boldsymbol{v}_i) W(\boldsymbol{x} - \boldsymbol{x}_j, h), \tag{2.69}$$

called XSPH [Mon89, GMS01] to stabilize the numerical method by smoothing the velocity with neighboring particles again.

The maximum time step size is determined by

$$\Delta t \leq Min \left(0.125 \frac{\rho h^2}{\eta}, 0.25 \frac{h}{3c}, 0.25 \sqrt{\frac{h}{3|\boldsymbol{F}|}} \right), \tag{2.70}$$

where the $\boldsymbol{F}$ is the magnitude of the total force on a particle.

2.2.9 High Performance Computing

In SPH, since we solve the governing equation in each position of the particle based on the current particle position and velocity, it is straightforward to obtain parallel computing into each particle point. Theoretically, if we have a machine with infinite core in one processors, then we can assign computing task of each particle into different processors, meaning that no matter how many points we have in one problem, the simulation time remains the same. The application programming interface (API) OpenMP (Open Multi-Processing) [CDK$^+$01, CJVDP08] is a powerful application that usually used in SPH, supporting shared-memory multiprocessing programming, speeding up the simulation by the number of cores in central processing unit (CPU), with comparison to single core simulation. However, in current era, the best personal computer (PC) sports only eight to twelve cores in a processor. Even though hyper-threading technique, which developed by Intel and allows the two logical CPU cores to share physical execution resources, can pretend the machine have twice number of cores, they are still not fast enough for high performance computing with millions of particles.

Graphics processing unit (GPU) computing, then gained more and more interest because it has totally different hardware architecture that comes from demands of large, parallel computation requirements [OHL$^+$08, ND10]. GPU divides the resources of the processor to execute different computing task at the same time to achieve parallelism as well as the relatively larger memory bandwidth that can be manipulated to reduce the data loading latency. Some drawbacks should never be ignored such as data transfer between CPU and GPU will be additional cost for computing time, dynamic memory allocation is not supported, and difficult to debug. Compute Unified Device Architecture (CUDA), developed by Nvidia corporation, is a well-known and common use parallel computing platform for general-purpose computing on GPU (GPGPU). CUDA is compatible with different programming languages such as C, C++, Fortran, Python and Julia, providing many CUDA-accelerated libraries and some useful functions. For example, atomic function is a read-modify-write atomic operation that guaranteed to be performed without interference from other thread during operation process in a specific memory address, which is useful when we want to run two operations in parallel with data in the same memory address.

Based on the characteristics of the GPU computing, we implemented SPH on the GPU to accelerate simulation time. During radius searching, we first partition space equally into bins, whose length scale is close to the particle searching radius. Then use atomic operation to insert each particle into bins and create a table that containing the particle indexes in each bins. Last, we can find the neighboring particle by only checking the particles found in current and neighboring bins. Other procedures list in algorithm 1 are easy to allocate needed computational operation of each particle to different GPU cores.

Shown in figure 2.9 is the comparison of computational time cost of each time step with CPU-based and GPU-based solver. The CPU-based solver, denoted as blue marker, was run in Intel Core i7-6700HQ processor, while the GPU-based solver, denoted as red marker, was run in the same laptop with NVIDIA GeForce GTX 960M graphics card. The x axis indicates the total number of particles of the testing problem and the y axis is the average time cost of each time step. We can notice that the GPU-based solver generally has ten times faster performance than the CPU-based simulation. The purple line and the green line in the figure represent the slope with $O(N \log N)$ and $O(N)$. Note that the slope with $O(N \log N)$ is the theoretical complexity of N particles radius searching problem.

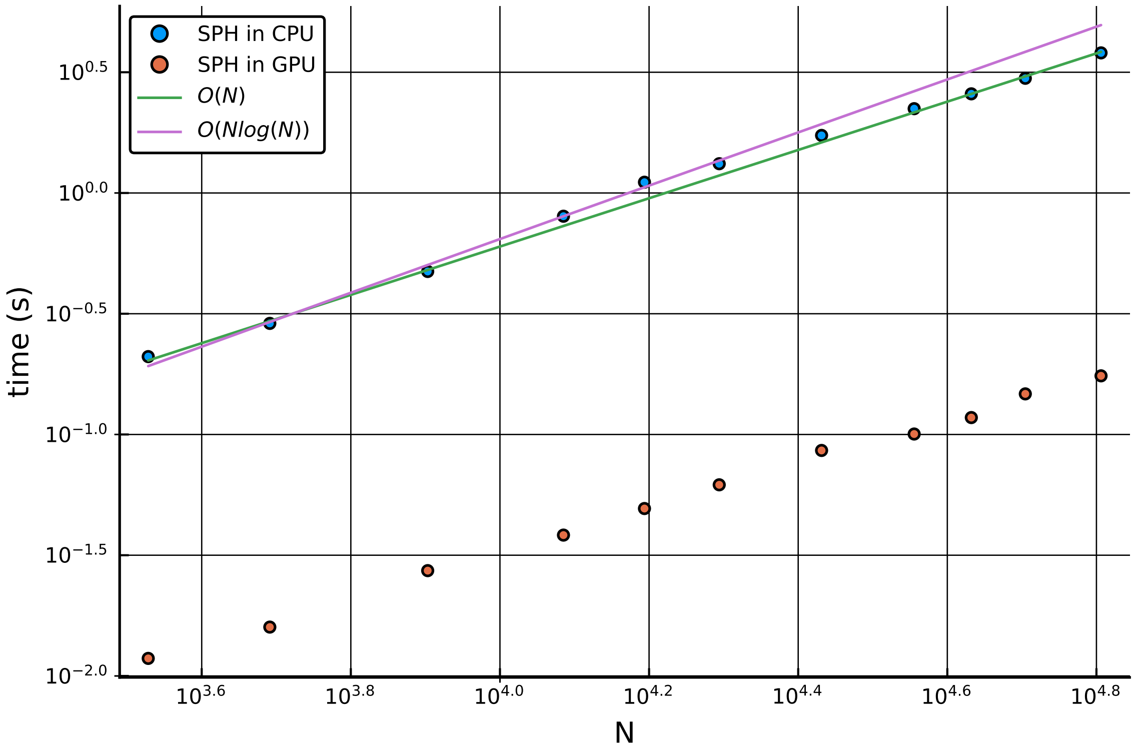


Figure 2.9: Analysis of acceleration by GPU

CHAPTER 3

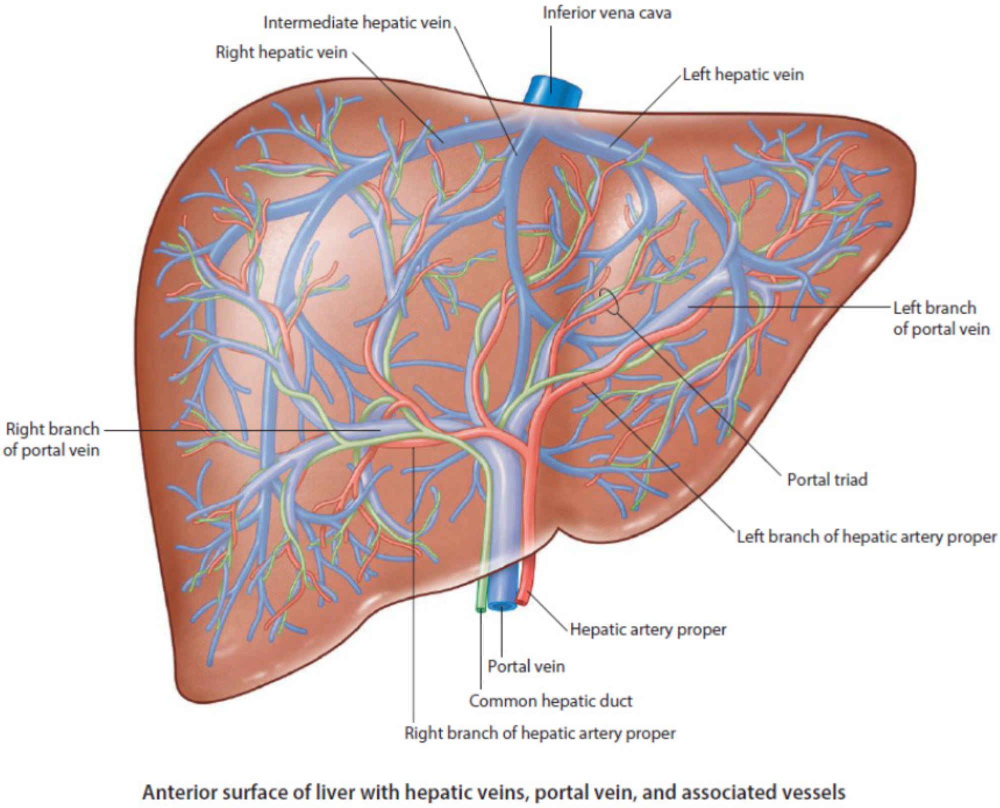
Application for Biphasic Modeling

In this chapter, we introduce the practical applications of multiphase problem such as highly perfused human organ and the flow in small scale porous medium that surface tension should be considered. More details about how to apply our multiphase SPH method to these applications is provided such as boundary resolution enhancement, membrane-like material formulation, fracture modeling, coupling with cardiovascular system, and formulation of surface tension.

3.1 Soft Tissue

Tissues, recognized as composite materials with constituents and small structures, consisting of cells and extracellular matrices, will continually deform according to the variation of mechanical and physiological environment. Tissues usually perform specialized functions and for animal tissues, they can be simply classified into four main groups[CD07]: connective, epithelial, muscle and nerve. Connective tissues have much intercellular substance and fewer cells, including cartilage, tendons, ligaments, bone matrix, adipose tissues, skin, blood, and lymph. Most of these are categorized into soft tissue including tendons, ligaments, fascia, skin, fibrous tissues, fat, synovial membranes, muscles, nerves, and blood vessel.[CD07, Sta15] The soft tissue performs nonlinear stiffness under different deformation and with many void inside, allowing blood and tissue fluid to flow into it. The nonlinear stiffness of the soft tissue comes from different contributors corresponding to the different degree of deformation. At small deformation state, the elastin provides the major contribution of the stiffness, while at large deformation state, the collagen fibers, which are loose at the small deformation, play the most important role to prevent severe rupture from the large strain. A typical J curve of stress and strain is used to describe the nonlinear stiffness. Besides, the voids in the soft tissue, usually occupying by blood or tissue fluid, provide the changeable response of viscous force as the soft tissue deformed.

3.1.1 The Liver



Intermediate hepatic vein
Inferior vena cava
Right hepatic vein
Left hepatic vein
Left branch of portal vein
Right branch of portal vein
Portal triad
Left branch of hepatic artery proper
Hepatic artery proper
Portal vein
Common hepatic duct
Right branch of hepatic artery proper
Anterior surface of liver with hepatic veins, portal vein, and associated vessels

Figure 3.1: Illustration of the liver in anterior side [Sta15].

The liver is the most vital organ of human, providing multiple functions such as detoxification, production of red blood cells and bile, storage of glycogen, protein synthesis, locating at human abdomen, can simply consider as consisting of parenchymal tissue and nonparenchymal tissue. About 70 85% of the liver volume is occupied by parenchymal cell, also known as hepatocytes. Figure 3.1 and 3.2 illustrate the view from anterior side and posterior side of the liver as well as the vascular system through the liver. The liver is attached with

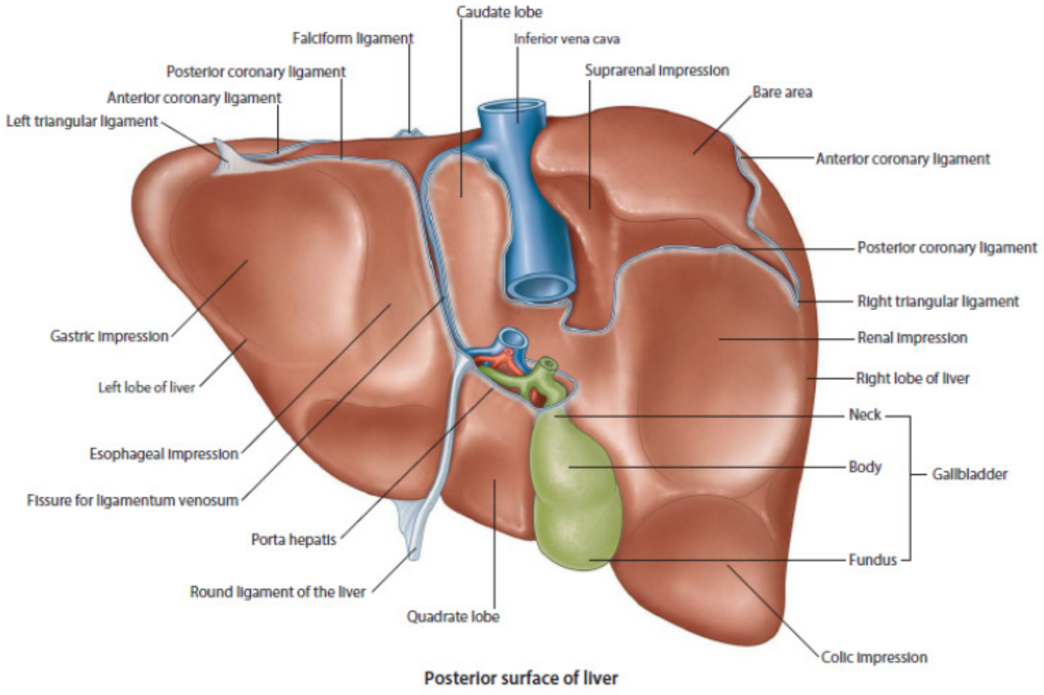


Figure 3.2: Illustration of the liver in posterior side [Sta15].

human body by numerous ligaments listed in figure 3.2 and covered with a membrane called Glisson's capsule. The blood and bile flow enter the liver at the visceral surface through hepatic artery, portal vein and hepatic duct. These three main vessels then subdivide into small branches and reach capillaries called liver sinusoids. All blood flow and tissue fluid interact with the hepatocyte at sinusoids before aggregating to hepatic vein, and then go back to cardiovascular system through the inferior vena cava. The oxygen-rich blood from hepatic artery and the nutrient-rich blood from portal vein provide essential elements of life for hepatocytes. For the whole liver, expect for large blood vessels, it consists of microscopic unit as shown in figure 3.3. All flow including the blood from both hepatic artery and portal vein as well as the bile pass through the sinusoid unit from the vertices to the center of hexagon. The sinusoid is a open core capillary, causing the liver to be similar with sponge-like material, having high interaction between solid and liquid, where we should apply our biphasic model and numerical solver on it.

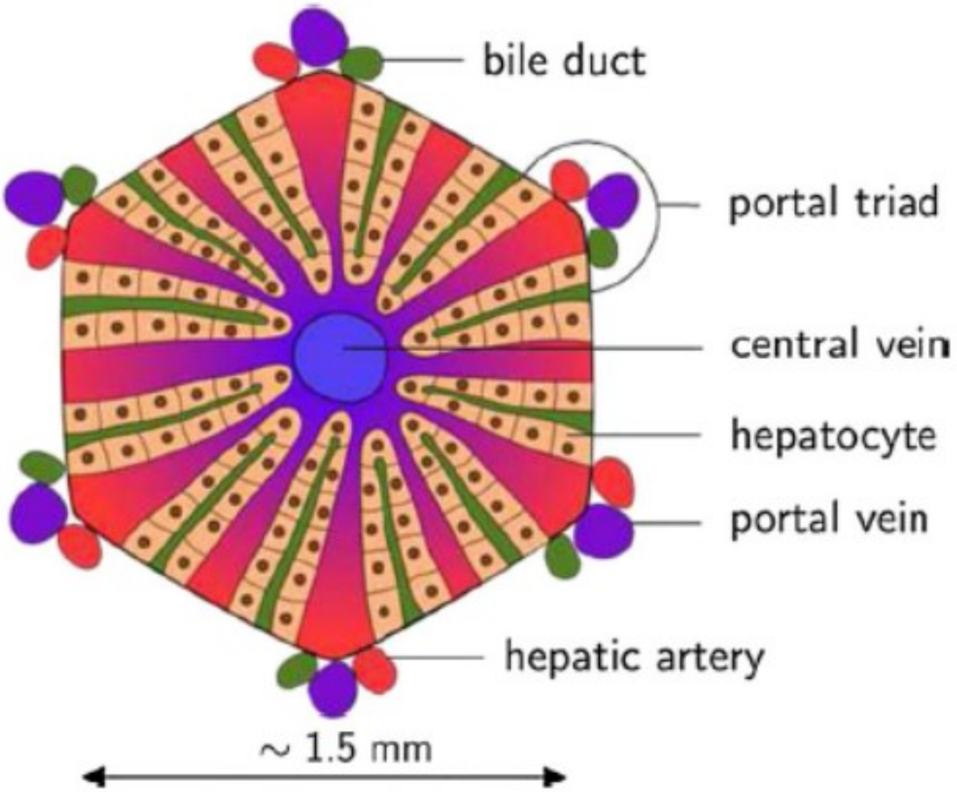


Figure 3.3: The microscopic unit of the liver [RDD10].

3.1.2 Boundary Resolution Enhancement

As the discussion in the previous section, the liver is covered with the Glisson's capsule that prevent all the liquid inside the liver to leak out. One issue pops out here is how to create a non-penetrable membrane to enclose all liquid. Usually, the initial position of particles are discretized from images such as computed tomography (CT) scan, which provide detailed three dimensional cross sectional picture. The resolution in whole domain stays the same level. However, the SPH method requires higher resolution at the boundary to block liquid particle passing through it and thus avoid leakage. At the mean time, too many particles would cause high computational resource demand. Under these concerns, we developed a method to locally enhance boundary resolution to prevent leakage but keeps the resolution inside the liver stay the same level. The concept of the method is depicted in the figure 3.4. For each particle that represents the membrane as marked in red in the figure, radius

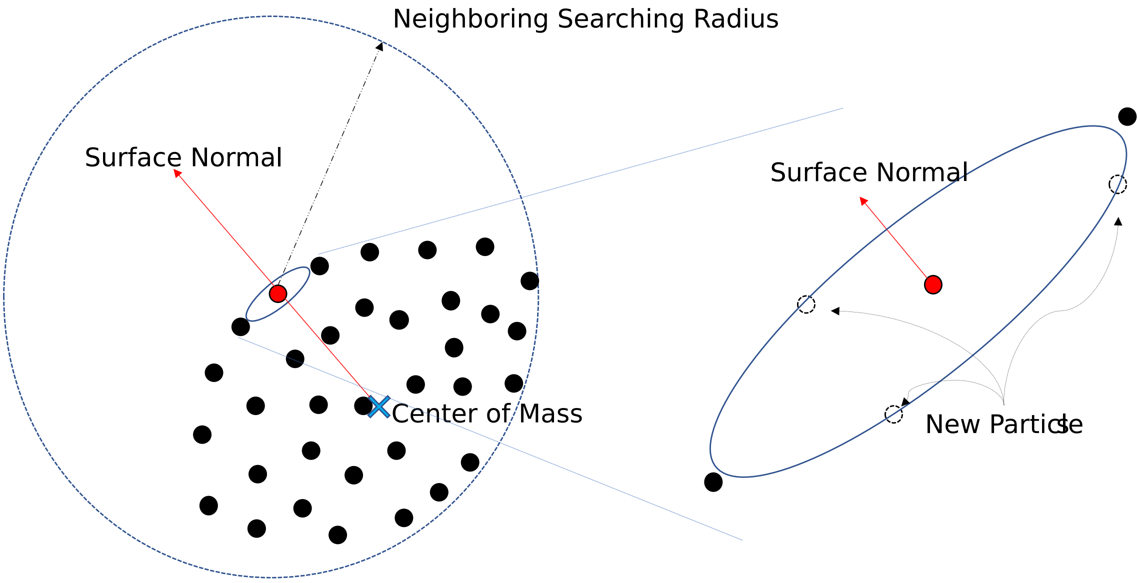


Figure 3.4: Illustration of boundary resolution enhancement.

searching was performed to obtain all neighboring particles. Then the center of mass, shown as blue cross in the figure, can be found by averaging the position of these particles. A local surface normal then defined as the direction from center of mass to the membrane particle. From the surface normal, a perpendicular circular plane can be constructed with the center of the membrane particle and a prescribed radius. Last, new particles can be create at the perimeter of the circle. The radius of the circle, and the numbers of new particle are two parameters depend on the demand of resolution.

The result of applying the resolution enhancement method on the liver is shown in figure 3.6. It increases fourth times of the surface particle numbers with the original surface shown in figure 3.5. The resolution of the boundary increase significantly here.

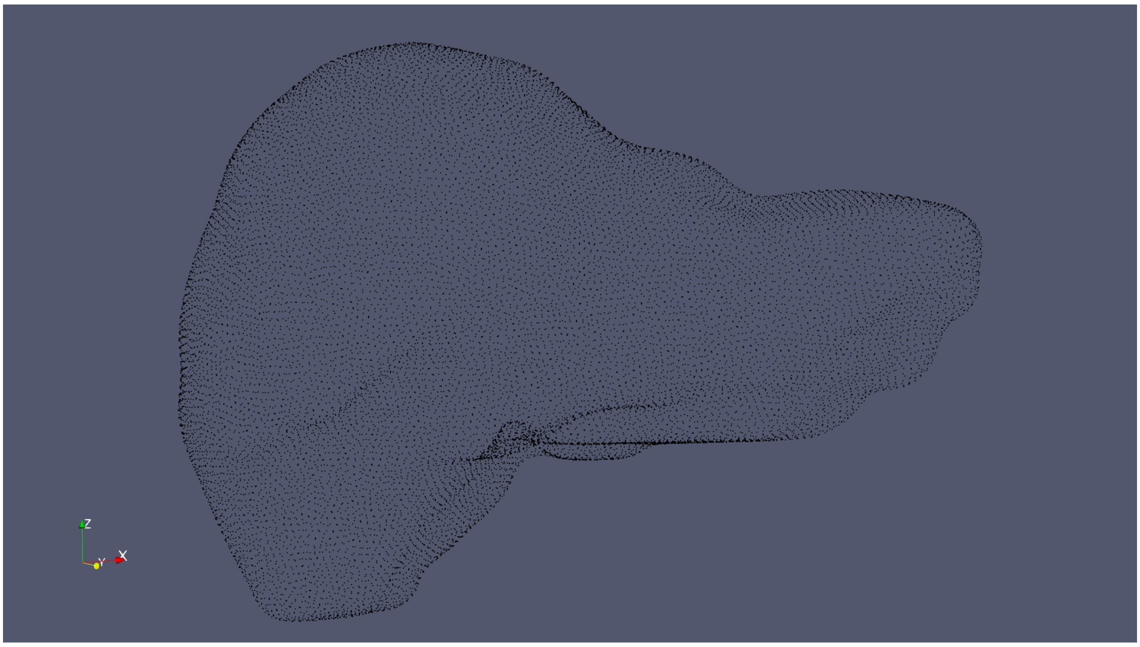


Figure 3.5: Initial membrane particles of the liver.

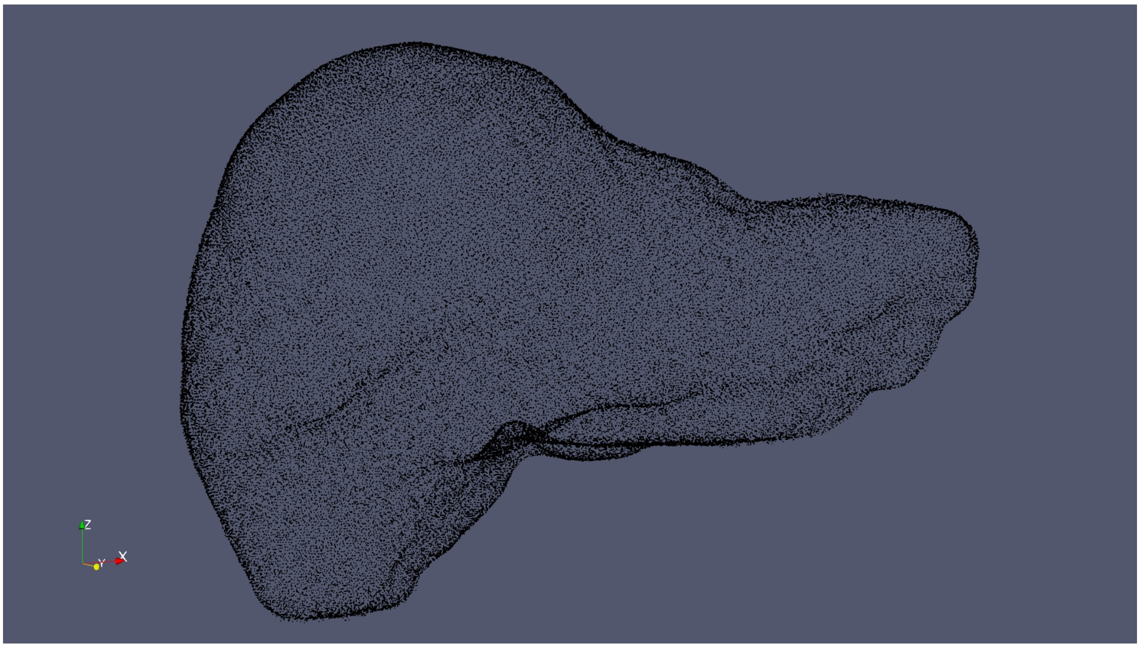


Figure 3.6: High resolution of the membrane by using resolution enhancement technique.

3.1.3 Membrane in SPH

The membrane is important in the liver simulation that it encloses all liquid inside without occurrence of leakage but still deforms with the inner particles of soft tissue. To fulfil the requirements, we have to first determine which particle is counted as membrane. We use the same technique that shown in the figure 3.4. Once we found the center of mass of and then calculate the magnitude of the vector from it to the corresponding particle, all vectors with a magnitude larger than a threshold indicate they are belong to the membrane. Note that if the particle is not located at the boundary, the center of mass and the particle position would be very close. After obtaining the membrane particles, we treat them as normal solid particle and still compute the deformation gradient of them at each time step. It is still affected by the non-membrane particles just like attaching onto them. One manipulation for membrane particles is to manually set the local liquid volume fraction of them being the equilibrium liquid volume fraction of the soft tissue, such that the isotropic stress would stay in reasonable magnitude. Note that there almost no liquid particles near the membrane since we keep updating membrane particles as the virtual boundary of the liquid particles, so that the membrane particles repel liquid particles to achieve our goal that encloses all the liquid.

3.1.4 Fracture Mechanics in SPH

In order to provide more feasibility on the liver simulation, fracture mechanics should be incorporated in the SPH method. In particle method, there is an implicit fracture mechanics that once two particles are separated further than the searching radius, they thus disconnect with each other. The disconnection here is equivalent to fracture of the material. Note that the fracture should be irreversible, which means the disconnected particles should not have influence on each other except for the isotropic force that preventing them penetrate to each other. The work flow of solid SPH solver incorporated fracture mechanics is listed in algorithm 2. A connectivity table have to construct at the beginning of the simulation, recording the initial elastic force contribution from connected particles. The table only contain the indexes of the connecting particles. Note that the construction of the table only required performing radius searching once, and then remove the index if the distance of two particles is further than the searching radius. In addition, all disconnected particles cannot become connected again since its a irreversible process, which corresponds to our sense that two broken materials will not have any elastic force between them any more. It is worthy to point out that we still have to perform radius searching on all solid particles each time step due to two broken particles still have contact force that we represent as the pressure force in the momentum equations. Here, we just provide a simple fracture model from the nature of particle method. Further complicated model can be integrated into the solver by using the information of local deformation gradient tensor and the knowledge of plasticity until fracture point being reached in the stress strain curve.

Algorithm 2 Algorithm on Fracture Mechanics

```
Initialize particle sets of solid position $\boldsymbol{x}^s$
Assign initial value of solid density $\rho^s$
Perform initial connectivity table for each solid particles
while animating do
    for all $\boldsymbol{x}^s$ do
        Radius searching of neighboring particles
        Compute density $\rho^l, d\rho^s/dt$, volume fraction $n^s$
        Update deformation gradient based on the connectivity table $\boldsymbol{F}^{<n>} = \hat{\boldsymbol{F}} \cdot \boldsymbol{F}^{<n-1>}$
        Compute deviatoric force based on the deformation gradient
        Compute isotropic force based on the current radius searching results
        Advance $\boldsymbol{x}^s$
        Evolve $\rho^s$
        Fracture check and update connectivity table
```

3.1.5 Coupled with Cardiovascular System

Under the injury condition, the simulation on the liver relies on additional information about how much blood will come from the portal vein and hepatic artery. These amount determines the maximum bleeding volume. Here, we hooked our SPH method with the thorough cardiovascular model proposed and implemented by Canuto [CCB+18, Can19]. The full-scale cardiovascular model is illustrated in figure 3.8. It basically couples zero dimensional models of the heart, pulmonary, peripheral vasculature and one dimensional models of the major arteries of human body. Each artery has numerous segments corresponding to its length. The model provides spatial information such as blood flow rate, blood pressure at specific segment of a specific artery. For the hepatic artery, is denoted as node fourteen in the figure 3.9. The model is also incorporated with a feedback model of the baroreflex, allowing control of heart rate, cardiac contractility, and peripheral impedance. The inflow condition of a healthy hepatic artery is depicted in blue color in figure 3.7. The pulsatile flow is corresponded to the heart beat, being squeezed out from left ventricle and passing through aorta to the liver. ZeroMQ [Hin13], which is a high performance asynchronous messaging library, was utilized to exchange and pass the information between full-scale cardiovascular system solver and the biphasic SPH solver despite their implementation is based on different programming languages.

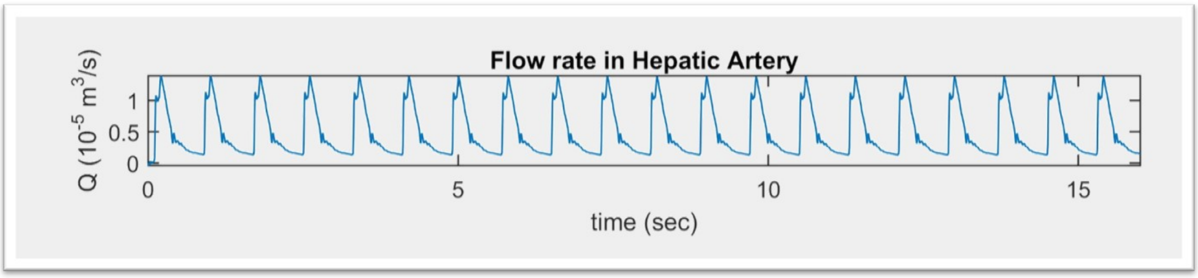


Figure 3.7: Inflow condition of hepatic artery from cardiovascular model.

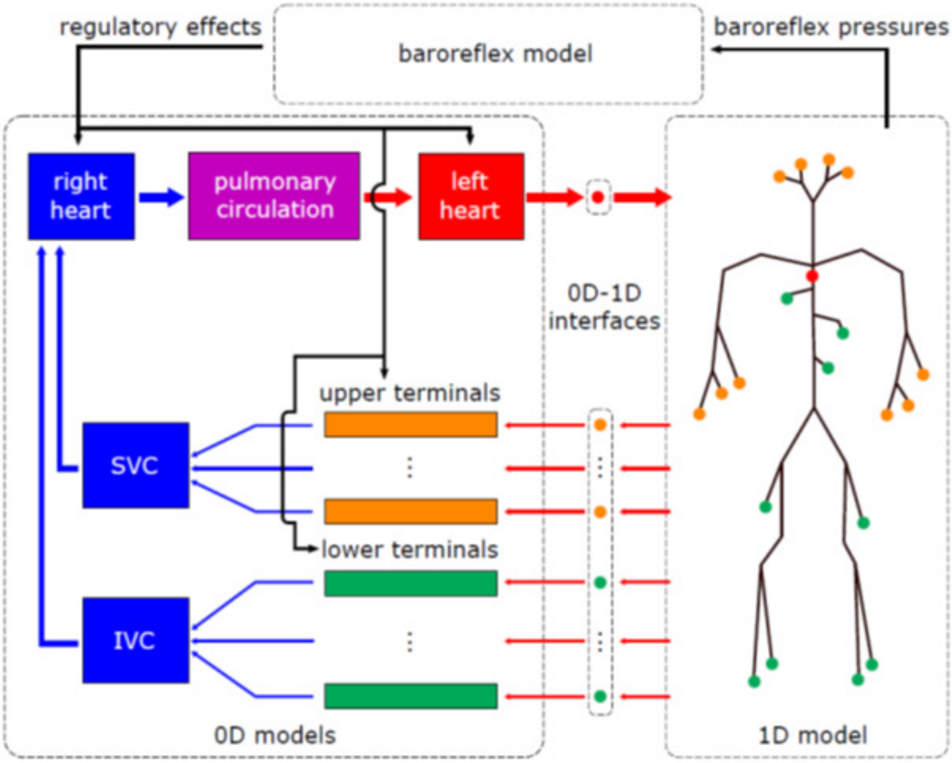


Figure 3.8: A high-level view of the closed-loop model architecture [CCB+18, Can19]

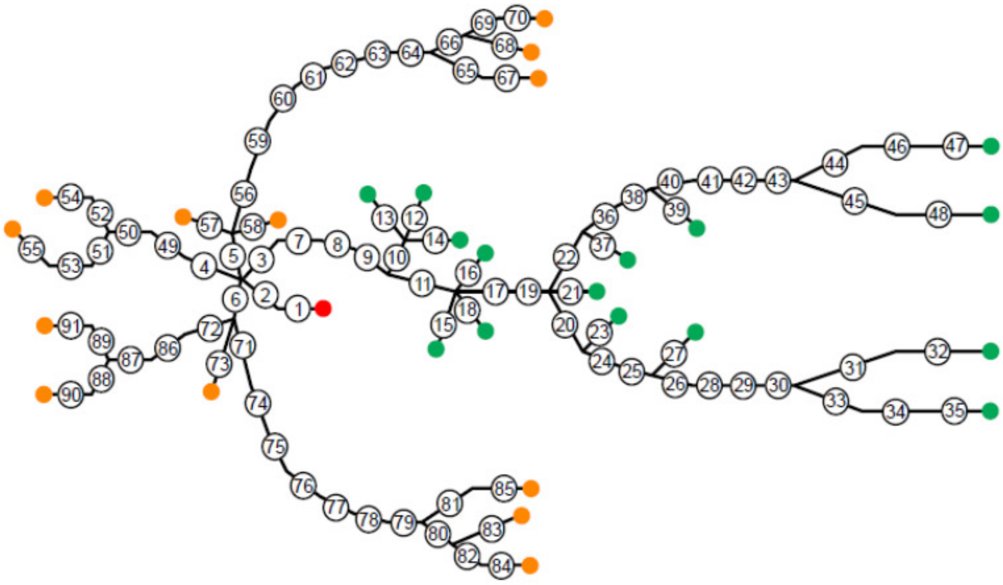


Figure 3.9: Connectivity diagram of complete one-dimensional arterial network. [CCB+18, Can19]

3.2 Surface Tension and Wicking

Surface tension is a term to represent the imbalance of cohesion and the adhesion between two different phases of molecules, with the dimension of energy per unit area. A typical example is a liquid droplet on a impermeable solid surface. The Young-Duprè equation that describes equilibrium status of liquid contacting with solid is

$$\gamma_{SV} - \gamma_{SL} = \gamma_{LV} \cos\theta, \tag{3.1}$$

where γ represents the interfacial surface tension between two phases of liquid, solid, and vapor denoted with subscript L, S, and V respectively, and θ is the equilibrium contact angle shown in figure 3.10.

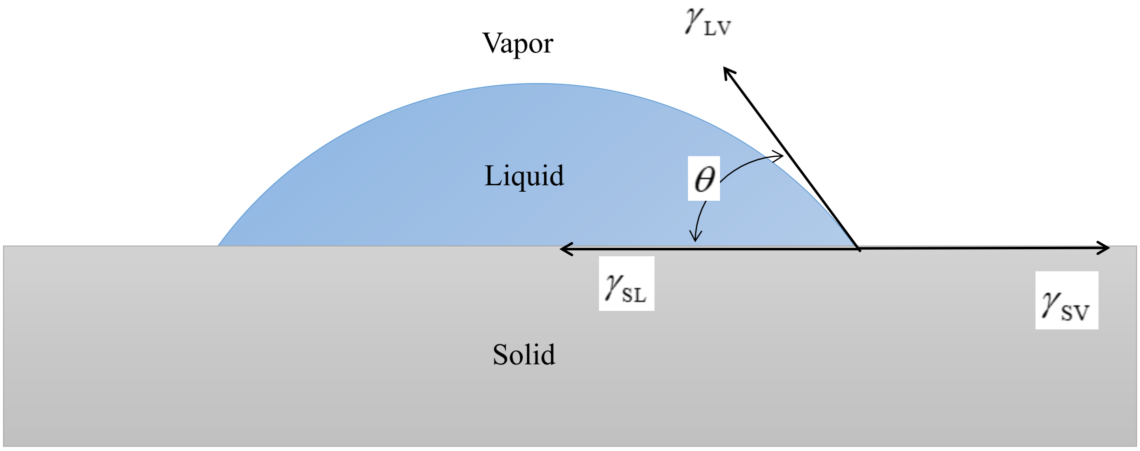
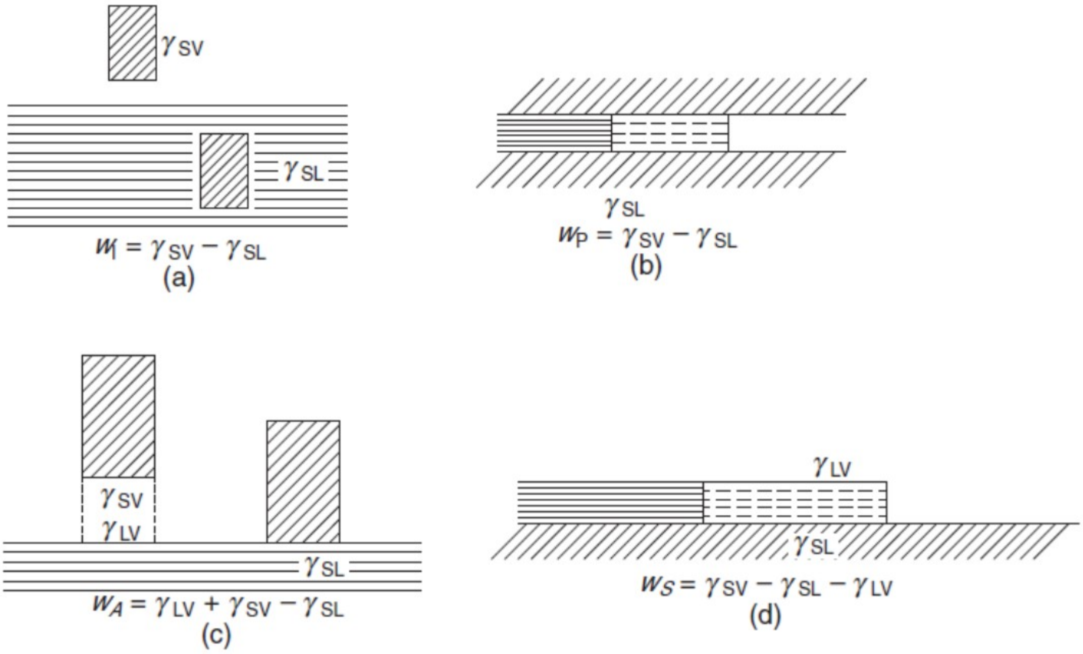


Figure 3.10: Illustration of surface tension and contact angle.

The general mechanisms of wicking are shown in figure 3.11, including following processes: immersion, capillary sorption, adhesion and spreading [Gil58, Lew84, PRKG06]. These processes are corresponding with energy releasing or consumption per unit area can be written as the combinations of γ_{SV},γ_{LV} and γ_{SL}. These mechanisms will happen if surface energy is released. Immersion and capillary sorption will occur as the interfacial energy of the solid/vapor interface exceeds that of solid/liquid interface. Adhesion releases surface energy from the separation of two contact surface. Spreading is a flow motion that causes increasing of solid/liquid and liquid/vapor interface.

If we zoom into the molecule scale at the triple point in the figure 3.10 and revisit the microscopic physics, the surface tension terms such as γ_{SV},γ_{LV} and γ_{SL} are actually the result from the imbalanced interaction force between molecule pairs due to different energy potential between heterogeneous molecules. Demonstrated in the figure 3.12 is the illustration to explain this concept, for the center molecule, the net force from liquid/liquid pairs, liquid/air pairs, and liquid/solid pairs can be written as equation 3.1. This concept is useful for our future formulation of surface tension in SPH method.



γ_{SV}
γ_{SL}
$w_I = \gamma_{SV} - \gamma_{SL}$
(a)
γ_{SL}
$w_P = \gamma_{SV} - \gamma_{SL}$
(b)
γ_{SV}
γ_{LV}
γ_{SL}
$w_A = \gamma_{LV} + \gamma_{SV} - \gamma_{SL}$
(c)
γ_{LV}
γ_{SL}
$w_S = \gamma_{SV} - \gamma_{SL} - \gamma_{LV}$
(d)

Figure 3.11: Wetting mechanism: (a) Immersion of a solid in a liquid; (b) capillary sorption; (c) adhesion between liquid and solid; (d) spreading of liquid on solid [Lew84, PRKG06]

3.2.1 Dimensionless Parameters

In the wicking problem, there are some dimensionless parameters obtained from Buckingham Pi theorem to evaluated the importance of surface tension as shown in table 3.1. Darcy number is defined as the ratio of permeability and the cross section area. Larger Darcy number indicates that the liquid flows through porous medium more easily. Bond number is the ratio of length scale and the effective capillary length scale. Larger Bond number

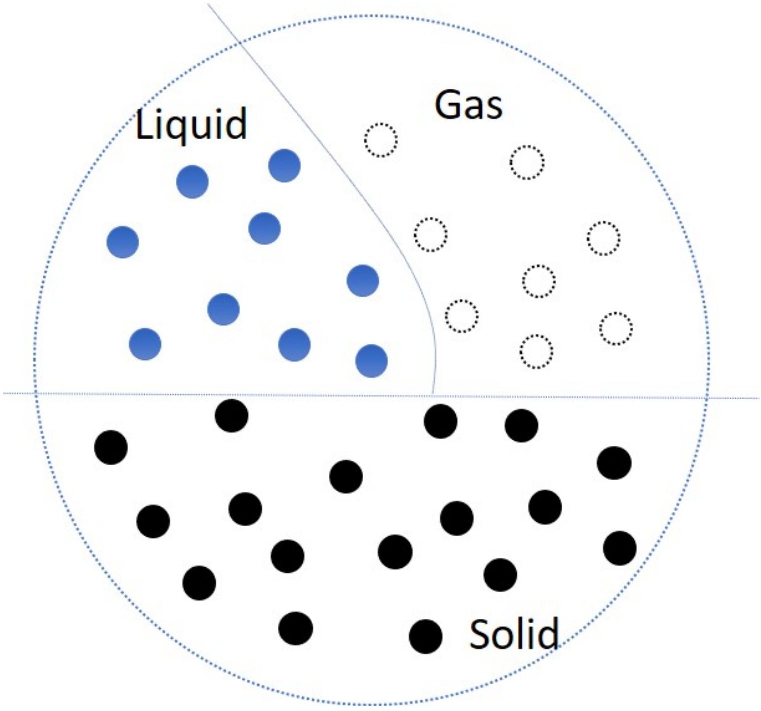


Figure 3.12: Surface tension under the viewpoint of molecule level

means that surface tension term is less important. Weber number is used to evaluate the importance off luid inertia compared to its surface tension. Last, Ohnesorge number is the ratio of square root of Weber number and the Reynolds number, representing the ration of viscous forces and the combination of inertia and surface tension. Larger Ohnesorge number indicates greater influence of the viscosity. Usually, in typical wicking problem, we can dimensionless spreading velocity with characteristic velocity $v_{ch} = \sqrt{\frac{\gamma}{\rho\sqrt{K}}}$ as a function of other dimensionless parameters,

$$\frac{v}{v_{ch}} = fn(\frac{\eta^2}{\rho\gamma\sqrt{K}}, \frac{\rho g K}{\gamma}) = fn(\frac{Oh^2}{\sqrt{Da}}, BoDa,) \tag{3.2}$$

Table 3.1: Lists of Dimensionless Parameters

Name	Parameter
Darcy number	$Da = \frac{K}{L^2}$
Bond number	$Bo = \frac{\rho g L^2}{\gamma}$
Weber number	$We = \frac{\rho v^2 L}{\gamma}$
Ohnesorge number	$Oh = \frac{\eta}{\sqrt{\rho \gamma L}}$

3.2.2 SPH Formulation for Surface Tension

As mentioned before, Tartakovsky and his coworkers [TM05, TP16] raise another approach that considering particle-particle interaction instead of obtaining a boundary surface, to mimic the intrinsic nature of adhesion and cohesion between molecules. They modeled the interaction force that causing surface tension as

$$\boldsymbol{f}_i = \sum_j s_{ij} \cos\left(\frac{\pi}{2}\frac{|\boldsymbol{r}_{ij}|}{h}\right)\frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}, \tag{3.3}$$

where the value of s_{ij} is the strength of surface tension and the magnitude of it is determined by the wetting condition and the contact angle. Note that here the SI unit of s_{ij} is not as usual as the common unit we saw in the traditional unit of surface tension. This additional force will be added back to the momentum equations. The method is impressive that convert the mathematical form of surface tension from boundary only to a general form for all particles. The formulation will automatically cancel itself if the particle is not located at the boundary. On simple example that an droplet on a plate is shown in the figure 3.13. The droplet stands on the plate and form a shape as we expect. The example shows the ability of these particle-particle interaction force formulation can capture the surface tension effect well.

Based on the concept, we extend the formulation being more general to the wicking process. We discretize the textile as a set of solid particles, which carry the average physical quantities like description in chapter 2 including permeability and volume fraction of each phase. The voids in the porous medium, whose size have been already captured by solid volume fraction, providing attraction force to the neighboring liquid particle due to surface tension from the interface between void and liquid in material parcel microscopically. Under mesoscopic viewpoints, the attraction force can be regarded as the force from the void in solid particle, acting on liquid particle. We thus model this attraction force due to surface tension in the inner voids as

$$\boldsymbol{f}_{ij} = \sum_j (1 - n_j^s - n_j^l) \frac{\gamma_{ij}}{h^2} \cos\left(\frac{\pi}{2}\frac{|\boldsymbol{r}_{ij}|}{2h}\right) \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}, \tag{3.4}$$

where $(1-n_j^s-n_j^l)$ is a weighting coefficient acted like a switch that make the effect of surface tension disappear as the local status reach saturation, and we scale the surface tension γ by the square of searching radius to match the SI unit of surface tension, which is force per unit length $\frac{N}{m}$. Note that $(1 - n_j^s - n_j^l)$ can be rewritten as $(1 - n_j^s)(1 - S_j)$, where S_j is the local saturation defined as $\frac{n_j^l}{1-n_j^s}$, representing the ratio of the current liquid volume with the maximum liquid volume can be occupied in the biphasic material parcel where the attraction force comes from.

In addition to the extra force formulation in mesoscale, the constraint that in the numerical pressure has to be reconsidered since it basically implicitly determines the final equilibrium state of the volume fraction. In chapter 2, we model the numerical pressure as $p^l = \rho c^2(n^l + n^s - 1)$, which implies the liquid will occupied all voids as reaching equilibrium, such that the equilibrium state is $n_{eq}^l = 1 - n^s$. However, in the wicking mechanism, the driving force is originally from the liquid-solid interface in the void, meaning that $n^l \neq 1 - n^s$ at the equilibrium. To address the issue, we should modify the pressure formulation as

$$P = \rho c^2 \left((1 - n_i^s - n_i^l) + C_1 n^l n^s\right), \tag{3.5}$$

where C_1 is the coefficient to be determined by the equilibrium state. Note that if the liquid particle is located at the pure liquid region, then the n^s in the last term will cancel out the modification term and then reduce back to the regular problem without consideration of surface tension from the void in the porous medium. The volume fraction of liquid at the equilibrium state, which is then considered as a material property here, can be obtained by letting numerical pressure be zero, which is $n_{eq}^l = \frac{1-n^s}{1+C_1 n^s}$. Larger value of C_1, means lower prescribed liquid volume fraction at the equilibrium state, macroscopically representing that liquid may spread further in the porous medium.

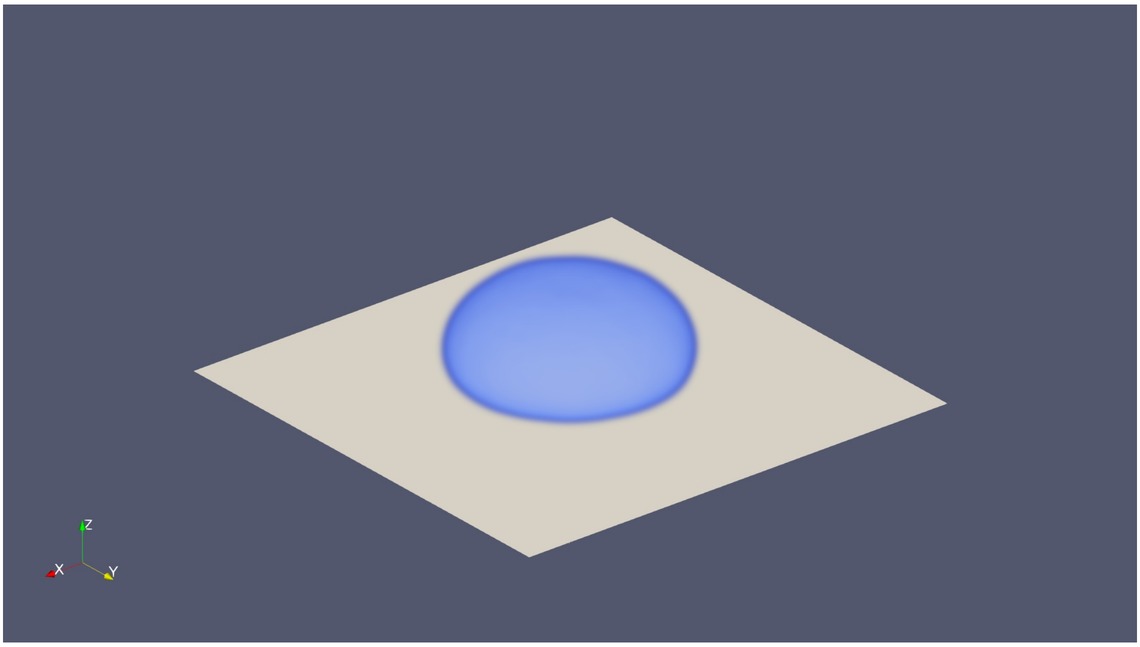


Figure 3.13: Schematic of a droplet on a plate

.

CHAPTER 4

Results and Discussion

4.1 Simulation of Validation Problems

The biphasic SPH methodology presented in the chapter 2 is validated in three manners in this section. First, we apply the method to two single phase problems to verify the correctness of implementation; these problems, applied to solid and fluid, respectively, are oscillations of a cantilevered beam and the transient pressure-driven flow through a channel. Both have analytical solutions to which the numerical solutions can be compared. The second problem is the pressure-driven flow of liquid through a static porous medium; the results of the method applied to this problem are compared with the solution supplied by the classical Brinkman equations for porous media. Finally, we perform a convergence test on this problem by decreasing the initial spacing between particles to confirm that the solution converges in this limit of high particle resolution.

4.1.1 Fluid Solver

The start-up of two-dimensional pressure-driven flow of a Newtonian fluid through a channel, also known as Poiseuille flow, is a benchmark problem in fluid dynamics. In lieu of a pressure gradient, the flow can also be driven by a constant and uniform body force F_b in the streamwise direction. The analytical solution of this is

$$v^*(y^*,t^*) = \frac{3}{2}(1-y^*)(1+y^*) - \sum_{k=0}^{\infty} \frac{48}{N_k^3} \sin\left[\frac{(1+y^*)N_k}{2}\right] \exp\left(-\frac{N_k^2}{4}t^*\right), \tag{4.1}$$

where the channel extends from $y^* = -1$ to $y^* = 1$ and $N_k = (2k+1)\pi$. The superscript $*$ implies dimensionless quantities, non-dimensionalized by the characteristic velocity, time and length defined, respectively, as $V = H^2F_b/(3\eta)$, $T = \rho H^2/\eta$, and H, where η is the viscosity, ρ is the density, and H is the channel width. Note that the Reynolds number is defined as $Re = \rho VH/\eta$, though the solution is independent of this parameter.

In the SPH solution, the number of fluid particles in each cross section was 106 and the total number of particles along the length of the channel was $N = 5355$. All particles that exited downstream of the channel re-entered at the upstream end. Fixed particles were placed along the walls at $y^* = \pm 1$ to achieve no-slip boundary conditions. The flow started from rest and then gradually reached steady-state. The results in Figure 4.1 depict the time-varying velocity along the centerline and show very good agreement between the SPH implementation and the analytical solution. The velocity profile at several instants is shown in Figure 4.2, also exhibiting good agreement across the entire channel, including at the no-slip walls.

4.1.2 Solid Solver

For small vertical displacements w from equilibrium, the governing equation of a cantilevered beam whose length is L and whose square cross section has area a^2 is

$$EI\frac{\partial^4 w}{\partial x^4} = -\mu_m \frac{\partial^2 w}{\partial t^2} + q, \tag{4.2}$$

where $I = a^4/12$ is the second moment of area, μ_m is the mass per unit length, q is the external load. The first mode's natural frequency of vibration is $\omega_1 = 3.516(EI/\mu_m)^{1/2}/L^2$. The results of an example of the SPH simulation of the free vibration of the beam are depicted in Figure 4.3. Here, the parameters are chosen to be $C_{10} = 250000, C_{01} = 9167.5$ Pa, $K = 900000$ Pa, $L = 0.5$ m, $a = 0.025$ m, and total mass $M = 0.4267$ kg; the corresponded natural frequency is $\omega_1 = 3.13$ rad/sec. The shear modulus was approximated by the relation $G \approx 2(C_{10} + C_{01})$, which valid in the range of small deformation.Then the Young' modulus was calculated as $E = \frac{9KG}{3K+G}$. The material parameters chooing here were as small as possible to mimic soft tissue behaviour but have to be large enough to have oscilating motion. The total number of particles used in the SPH simulation was $N = 24321$. The beam was fixed at one end of its length and the other end was free. The free end was imposed a initial velocity

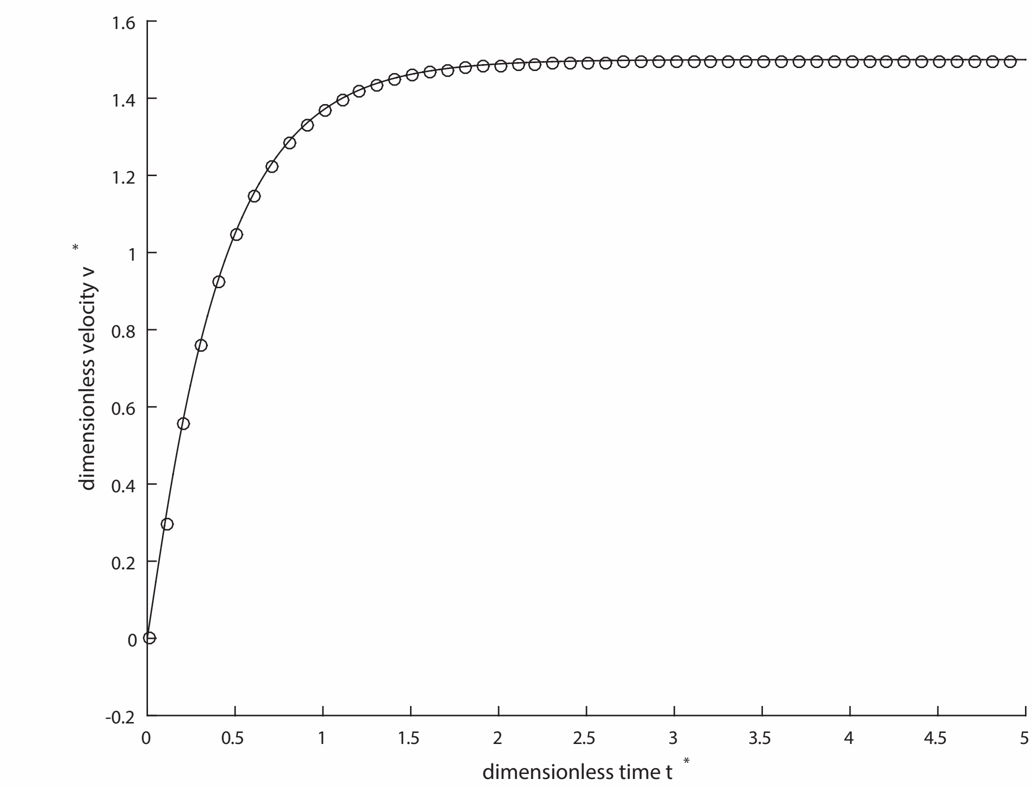


Figure 4.1: Velocity along the centerline of a channel for start-up flow of Newtonian fluid on SPH result(○) and analytical solution(−)

at the time $t = 0$ in order to supply the initial perturbation of the beam [Mon00, GMS01]. Body force q was turned off for whole simulation. The resulting time-varying positions of all particles were recorded and the vertical position of a point at the end of the beam is shown in Figure 4.3. Good agreement with the linear analytical solution is achieved, though at a somewhat higher frequency. Furthermore, it is also important to note that there is very little dissipation over several oscillation cycles.

4.1.3 Fluid-Solid Interaction

The flow of a Newtonian fluid through a static porous medium is governed by the Brinkman equations, which can be solved analytically when the flow is driven by a uniform pressure gradient in one direction. Thus, this problem serves as a useful validation of our biphasic

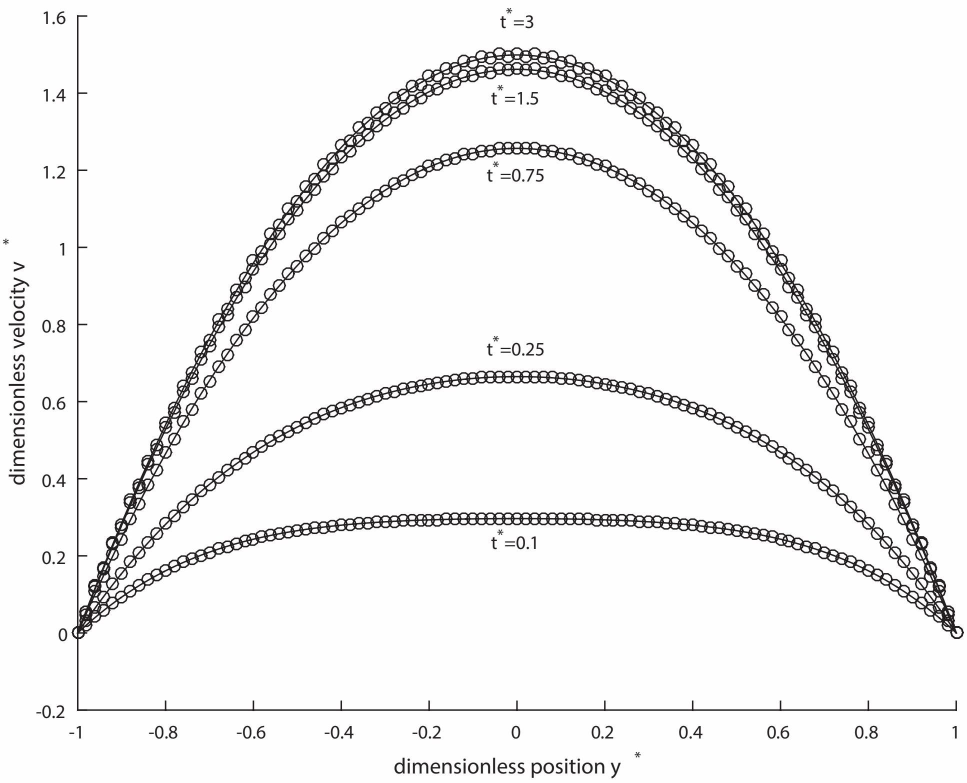


Figure 4.2: Comparison of SPH result(○) and analytical solution(−) of velocity profile at several instants for start-up flow of Newtonian fluid through a channel.

SPH model, since the equations should reduce to the Brinkman equations when the solid volume fraction is set constant and the velocity of the solid maintained to be zero.

The basic equations and numerical choices of this simulation are the same as those used in the previous Poiseuille flow problem, but with an additional interaction force term representing the friction force, with the dimensionless inverse permeability set to $\bar{K}^* = \bar{K}L^2/\eta = 4$. The Reynolds number is set to 5×10^{-6} and the solid volume fraction is set to $n^s = 0.5$. The results for the biphasic SPH simulation are compared with the steady-state solution of the Brinkman equation in Figure 4.4. The results agree well once steady state is achieved in the transient solution. The results of the basic channel flow (infinite permeability, or zero interphase friction) are shown for reference, and show the effect of the interphase friction for establishing a different steady state (Darcy flow).

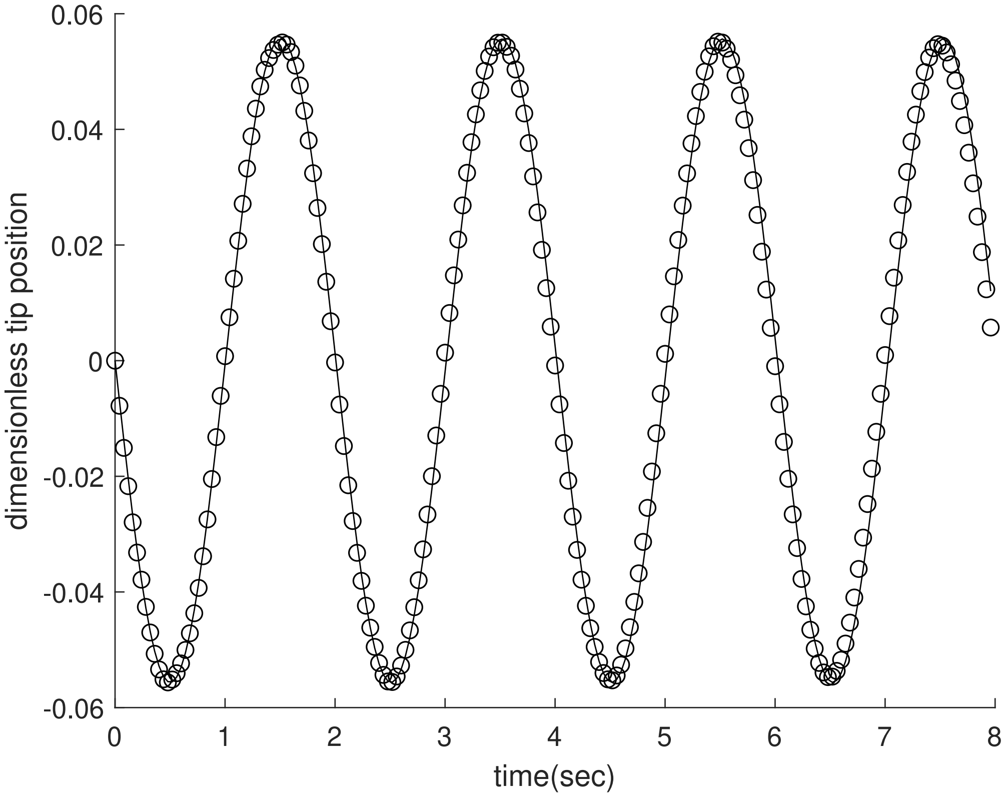


Figure 4.3: The vertical tip position (non-dimensionalized by its initial value) of the cantilevered beam. The result from SPH solver ($\circ$) has natural frequency $\omega = 3.15$ while analytical solution ($-$) suggests frequency as $\omega = 3.13$.

4.1.4 Convergence test

In this section, we carry out the SPH solution of the previous porous channel problem for various choices of initial spacing between particles and check that the solution converges. The results for the centerline velocity, v_0 in each case are compared with that of the analytical solution, v_{Br}, after steady state is achieved; the relative error was defined as $|v_0 - v_{Br}|/|v_{Br}|$. Initial inter-particle spacings were set to $\delta x = 0.05$, 0.04, 0.025, 0.02, 0.0125, and 0.01, and the population of the set of neighbors of each particle (i.e., the ratio $h/\delta x$) was fixed among all cases. The results of the error versus particle spacing are shown in Figure 4.5. As the initial particle spacing δx decreases, the relative error also decreases but eventually the error saturates. This error saturation has been discussed previously in the SPH literature

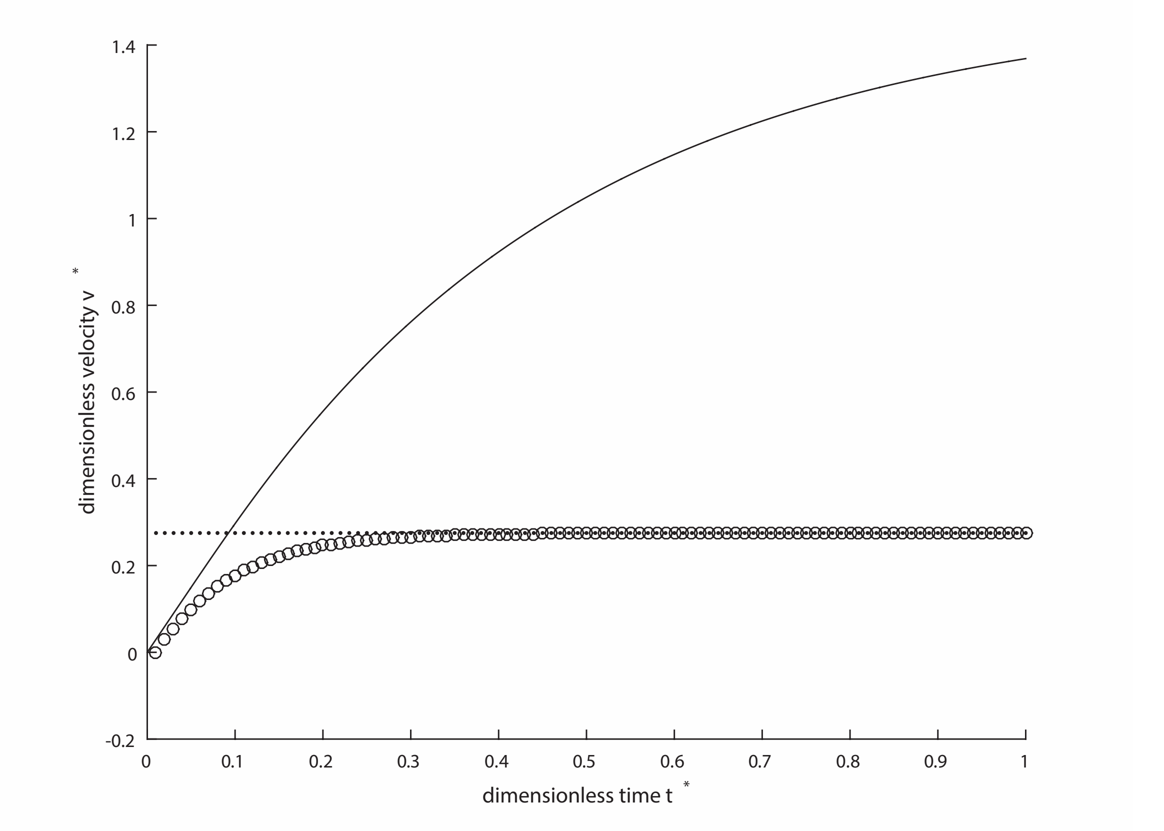


Figure 4.4: Comparison of velocity on the centerline between the biphasic model(○) and the steady-state Brinkman equations(·). The results for Poiseuille flow(−) are shown for reference.

[Spr10, ZHL15], and results from the fact that the convergence of SPH depends on the particle spacing, searching radius and the number of particles in the neighboring region of each particle. Zhu [ZHL15] states that numerical convergence of SPH only occurs in the dual limits were satisfied that decreasing inter-particle spacing and increasing number of particles in the neighboring region; otherwise, there is still a numerical error coming from limiting number of particle in the neighboring region.

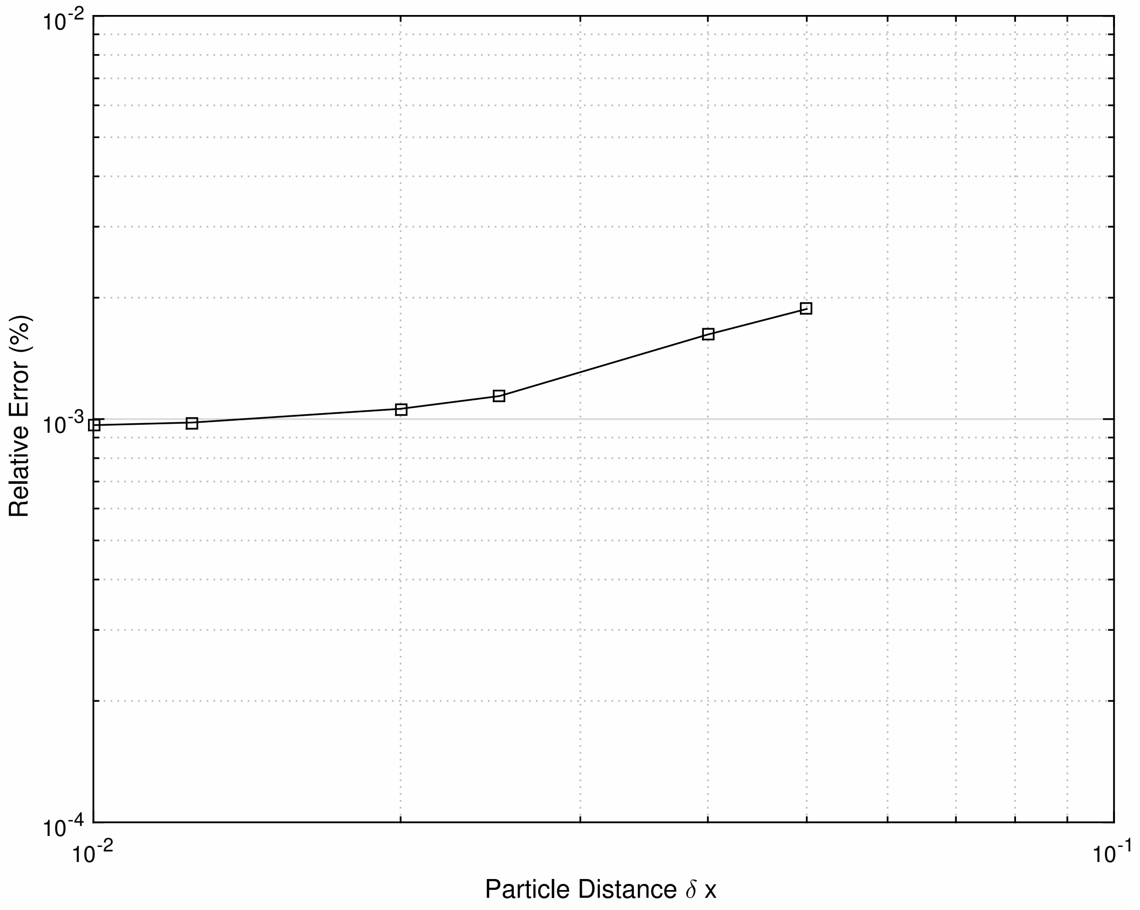


Figure 4.5: Relative error of the biphasic SPH simulation of flow passing through static porous medium.

4.2 Simulation on Medical Applications

In this section, we use the SPH methodology presented in chapter 2 and chapter 3 to carry out biphasic simulations of several scenarios involving a cubic specimen perfused with liquid first; this specimen is assigned the properties typical of soft biological tissue[KS07, CKC+07, GLD10], $C_{10} = 100, C_{01} = 0.4$, and the viscosity of liquid was selected as $\eta = 0.35$ Pa·sec, which is more viscous than the blood, as suggested by Kelager [Kel06] to provide more dissipation to stabilize the numerical method. The results are visualized with ParaView [AGL05, Aya15], which has the useful capability of interpolating point-based objects to generate surfaces. We assumed throughout that the initial volume fraction of the solid and liquid were 0.6 and 0.4 **??**, respectively. In each case, the specimen was subjected to gravity and mounted on a square flat plate below it. All impenetrable boundaries, such as a wall, a plate, and a projectile, were treated with virtual particles with prescribed properties, as described in chapter 2. The specimen was allowed to relax toward equilibrium before simulations were carried out, so that the initially cubic geometry was slightly deformed under the influence of gravity. The number of solid and liquid particles in the cubic specimen were $N^s = 32768$ and $N^l = 27000$, respectively. The reference densities of the solid and liquid were both set to $\rho^s = \rho^l = 1000$ kg/m^3.

4.2.1 Oozing

In the first example, the initially-perfused specimen is simply allowed to evolve naturally under the influence of gravity. As observed in Figure 4.6, the perfusing liquid drains from the specimen, accumulating on the supporting plate, spreading radially outward on this plate and eventually falling over the edge. After some time, the upper part of the specimen dries out (i.e., loses its liquid particles) and contracts toward the dry material volume fraction.

4.2.2 Indentation Test on a Membrane-enclosed Specimen

Figure 4.7 depicts the SPH simulation results for an indentation test, a standard test for evaluating the constitutive behavior of a material. In this simulation—as well as all of the remaining examples—the biphasic cubic specimen is enclosed entirely by a thin membrane that can deform but is impenetrable to liquid, preventing the liquid from escaping the specimen. An indenter with a spherical head, shown transparently in the snapshots, was initially placed above the specimen, prescribed to move steadily downward to compress the specimen for a few seconds, and then pulled back to release the compression. The specimen deforms laterally during the applied compression, as one would expect as the specimen preserves its volume. A residual indentation is apparent in the material after the indenter has pulled back without any influence on specimen, demonstrating the hysteresis in the biphasic specimen.

4.2.3 Blood Drawing

Figure 4.8 exhibits the results of an SPH simulation in which blood is drawn from the soft tissue specimen through a syringe. In the images, the syringe is shown translucently and consists of a needle tip inserted into the specimen. The upper part of the syringe, of larger internal diameter, applies a pressure lower than that inside the specimen. As apparent in the figure, liquid particles are drawn upward into the syringe. Furthermore, though it is difficult to see in the images, the solid specimen is elastically deformed toward the syringe, as well.

4.2.4 Coupled with Cardiovascular System

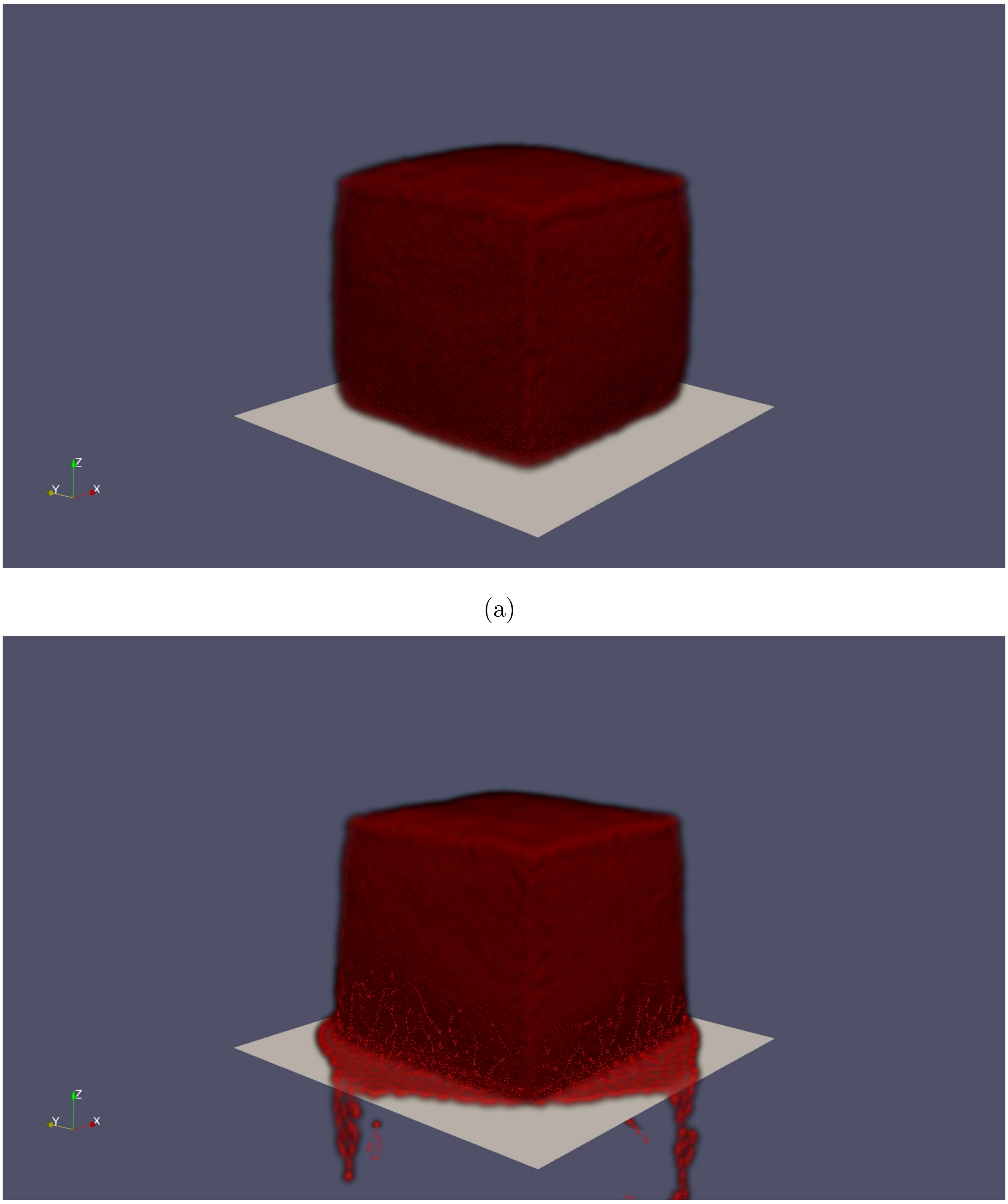
Shown in the figure 4.9 is to show the inflow condition from Canuto's work [CCB$^+$18, Can19] and demonstrated in the figure as new liquid particles coming from a pipe. The rate of adding new liquid particles follows the volume flow rate in hepatic artery. The flow is pulsatile correspond to the heat beat. In the figure, we show two snapshots of minimum and maximum flow rate respectively. The mass of each liquid particle remain constant so that more new liquid particles coming from the pipe means larger volume flow rate as shown in the bottom figure.

4.2.5 External Forcing on Soft Tissue

In the example, whose results are shown in Figure 4.10, a ballistic projectile passes through the perfused soft tissue specimen with constant prescribed velocity. The projectile provides pressure force when it approaches the specimen that may separates tissue particles. As the projectile encounters the membrane-enclosed tissue, a damage model is applied to the tissue particles, resulting in a breach of the membrane, fracturing of the tissue, and the creation of a large void along the path of the projectile. Damage model comes from the intrinsic feature of SPH that once adjacent particles being separated more than the searching radius, they disconnect without any interaction between them. We then record these disconnectivity that avoids any elastin force interaction between these disconnected particles after failure, just like an appearance of wound. Liquid particles freely escape through the damage region. In this example, rather than simulate drainage of a fixed reservoir of perfusing liquid, we created a persistent source of new liquid particles at the center of the specimen, mimicking an arterial supply of blood. Here, the new liquid particles were introduced at a constant rate, but the choice is arbitrary and can be time-varying and determined from the physiological state of the cardiovascular system.

4.2.6 Scalpel and the Liver

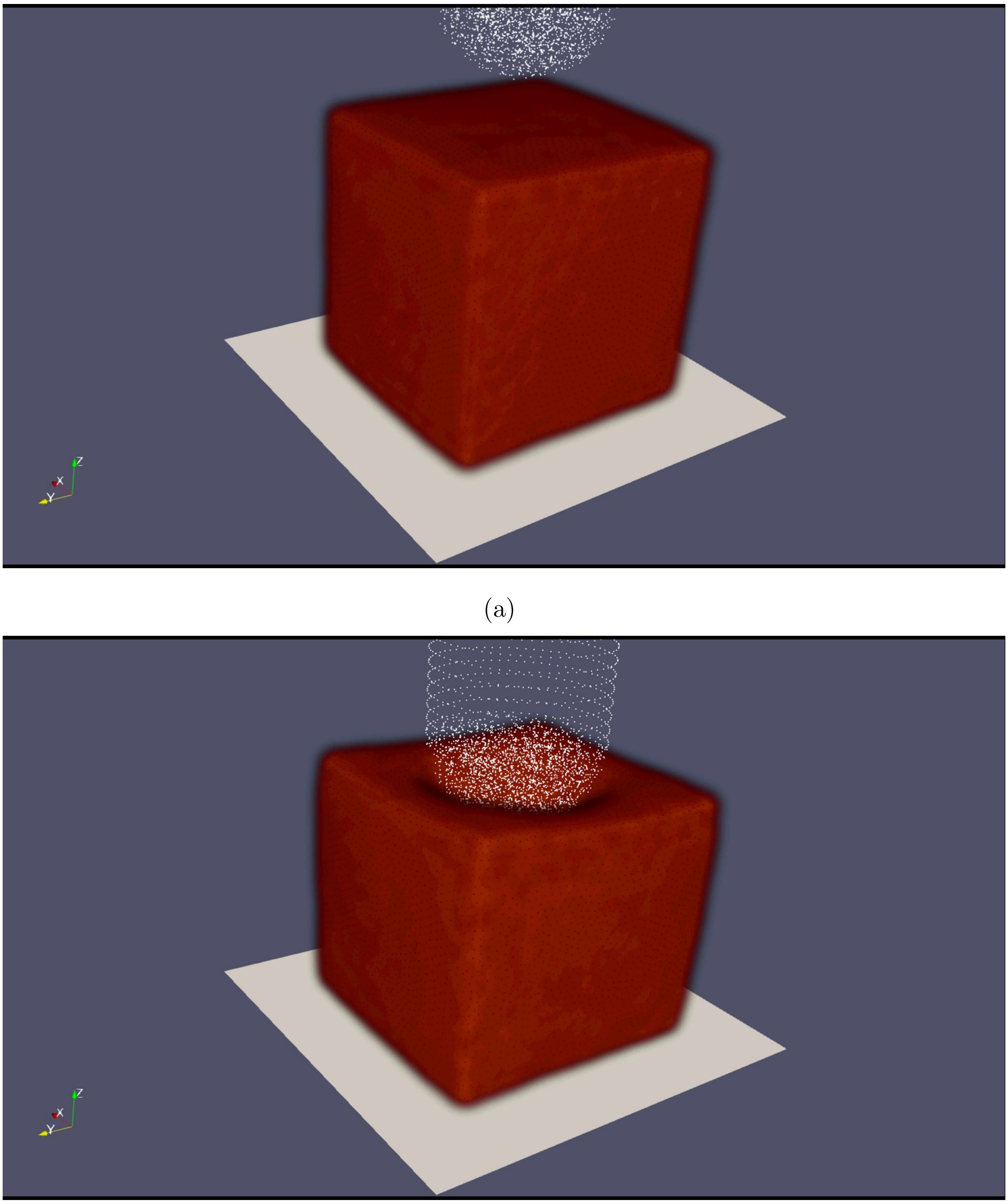
In this simulation, cubic specimen is replaced by a real organ: the liver as shown in figure 4.11. The liver is discretized by 307113 solid particles, including 17027 membrane particles to enclose all other solid particles. A rectangular thin plate mimicking a scalpel is put above the liver initially and then move downward after simulation begins. Another bigger rectangular plate is located below the liver to let the blood and the segment of the cutting liver drop on after failure of the liver and bleeding occur. The liver is hung in the air by letting the left upper surface of the liver particles fixed, while the other part of the liver would be affected by the gravity. The elastic force between fixed-position particles and others makes the liver stay in the space before any external force exerts on the liver. After the simulation starts, the scalpel passes through and cuts the liver, disconnecting the connection of the particles separating by it, causing the liver to divide into two segments. The left segment is still connecting with fixed-position particles but the right segment falls down due to the gravity. After failure of the liver, so does the non-penetrable membrane, the liquid particles inside the liver then start to ooze through the cutting plane of the liver, dripping down to the plate below the liver.



(a)

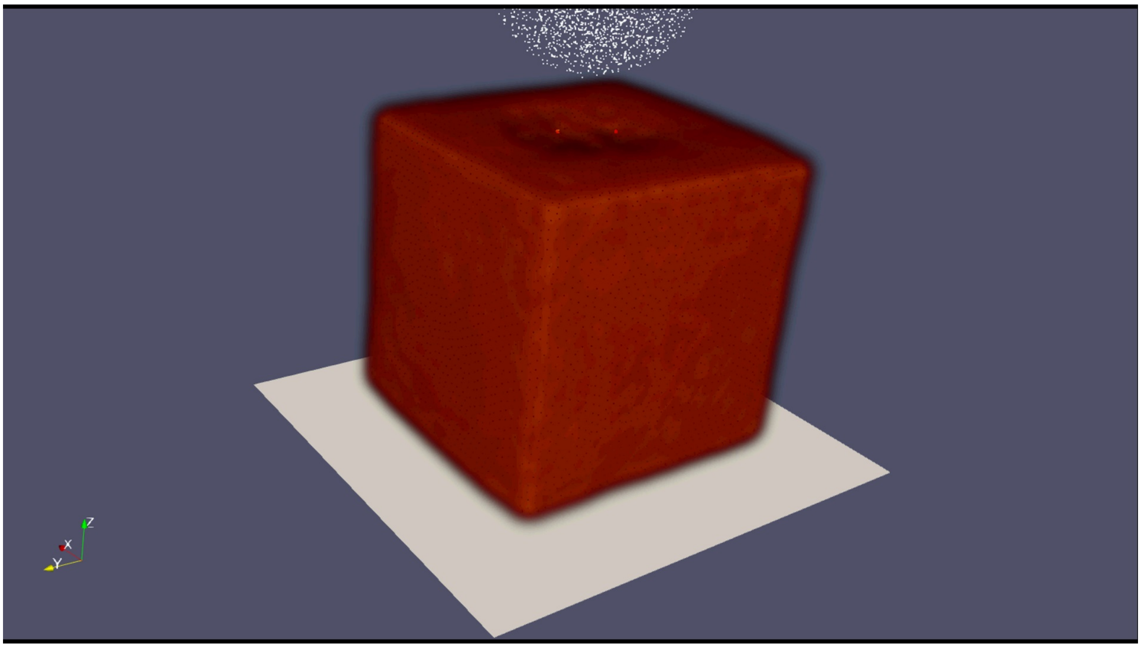
(b)

Figure 4.6: Drainage from soft tissue with a permeable surface. Solid specimen is shown in brown, while liquid particles are shown in red. Left: initial configuration: Right: snapshot at $t = 5$.



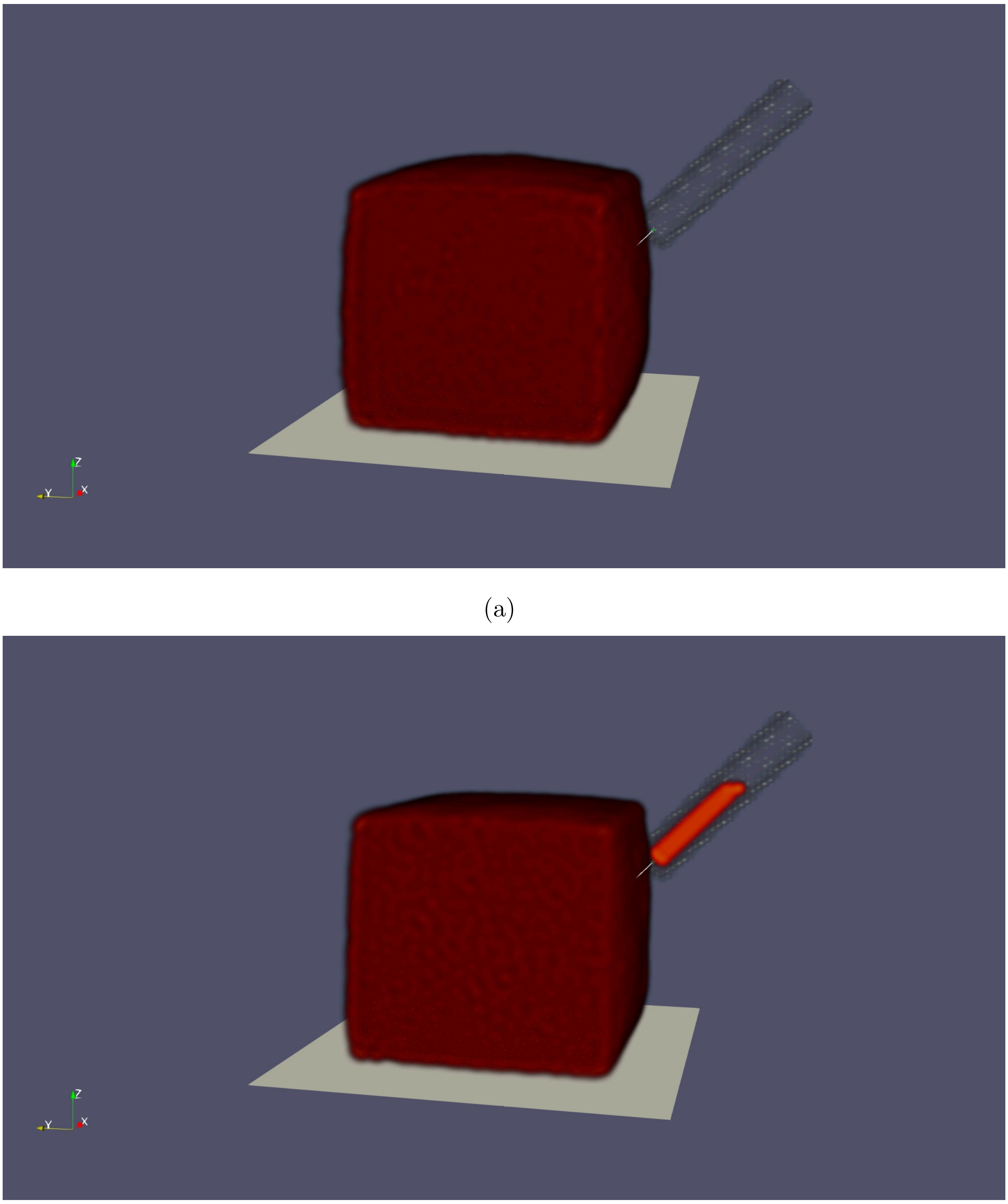
(a)

(b)



(c)

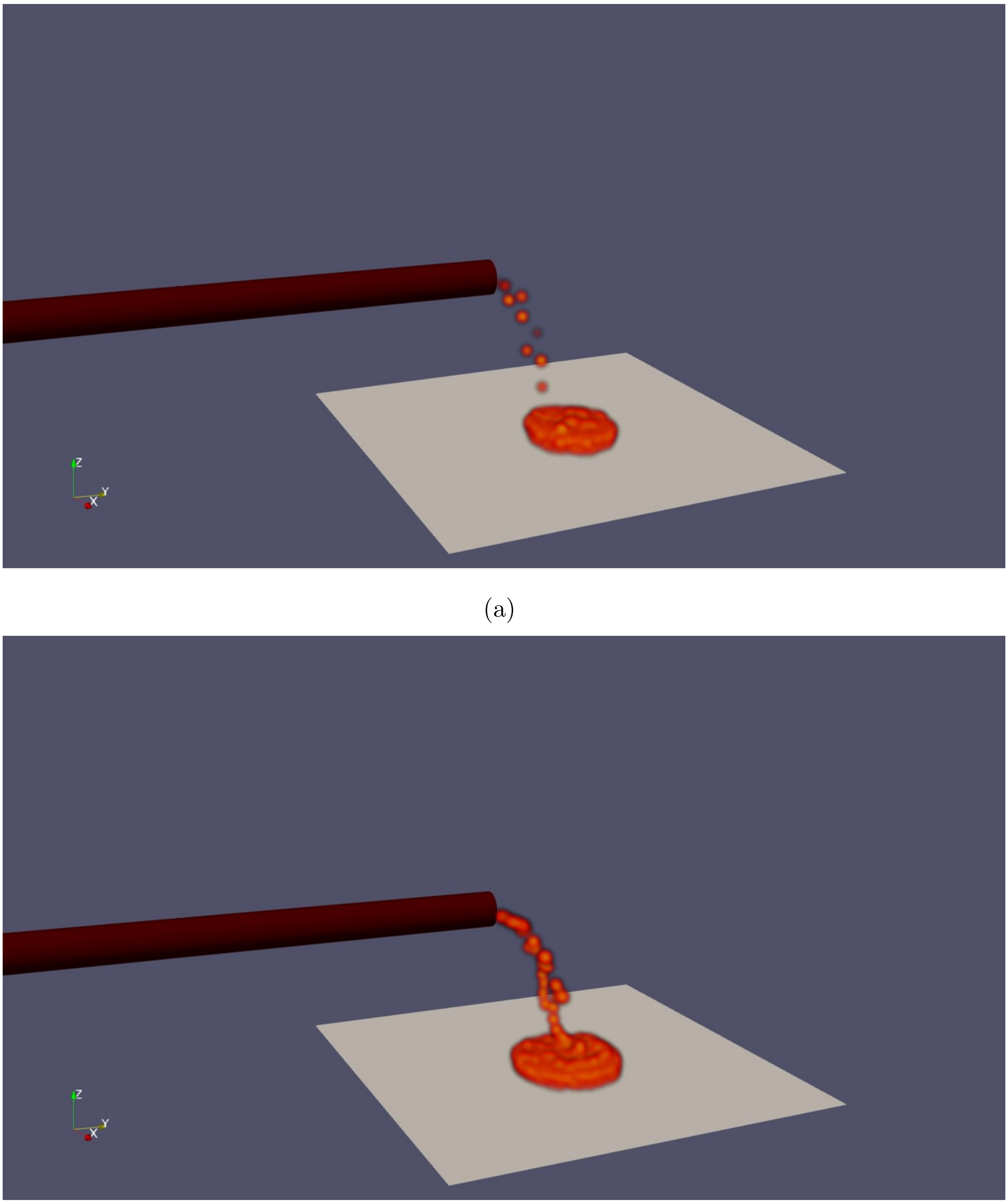
Figure 4.7: Indentation of perfused soft tissue specimen enclosed by a membrane.



(a)

(b)

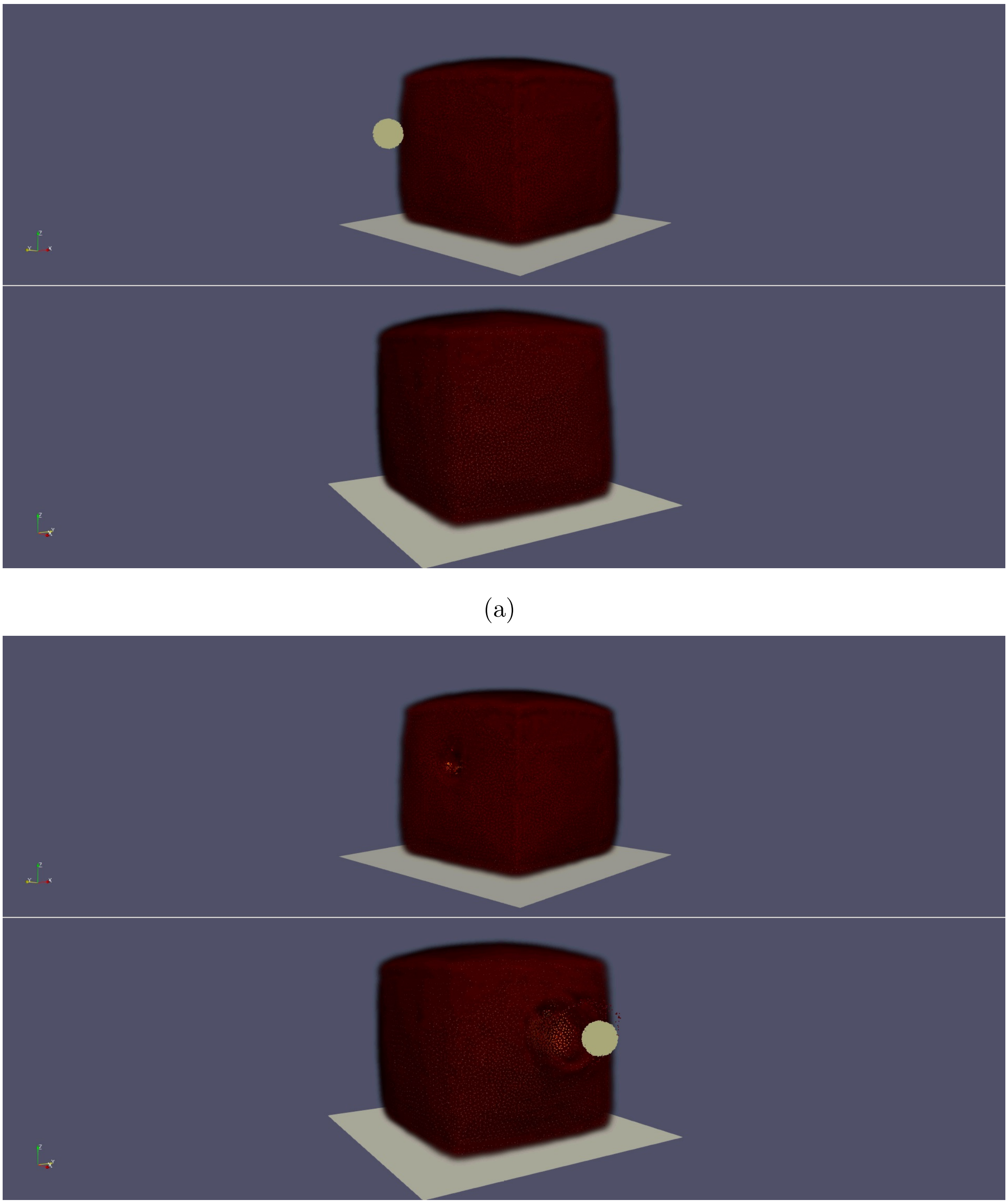
Figure 4.8: Drawing blood through a syringe from perfused soft tissue specimen enclosed by a membrane.



(a)

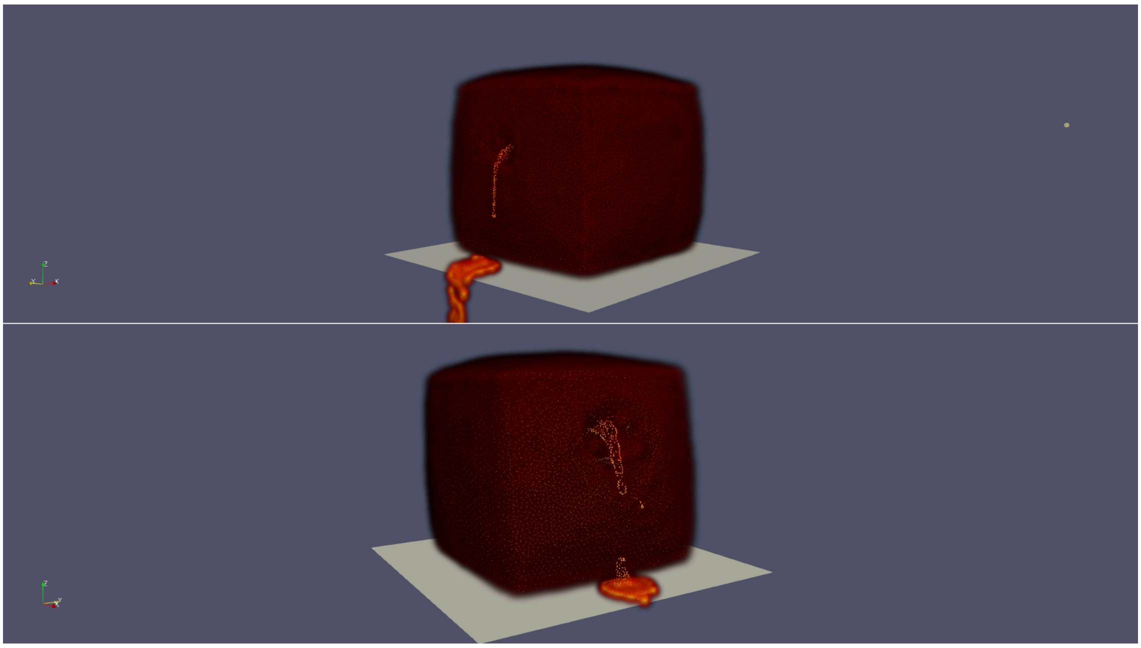
(b)

Figure 4.9: Inflow condition from cardiovascular system



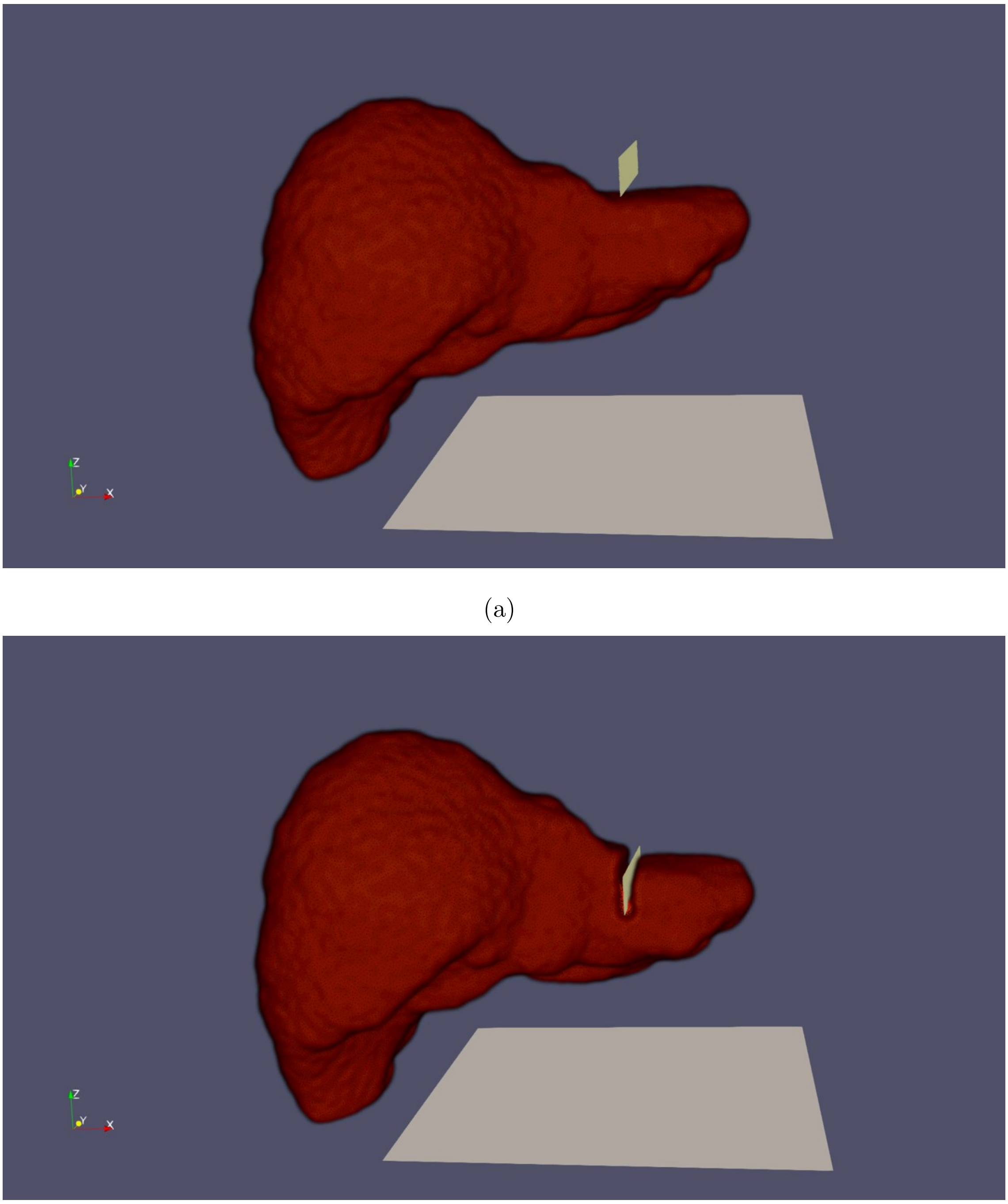
(a)

(b)



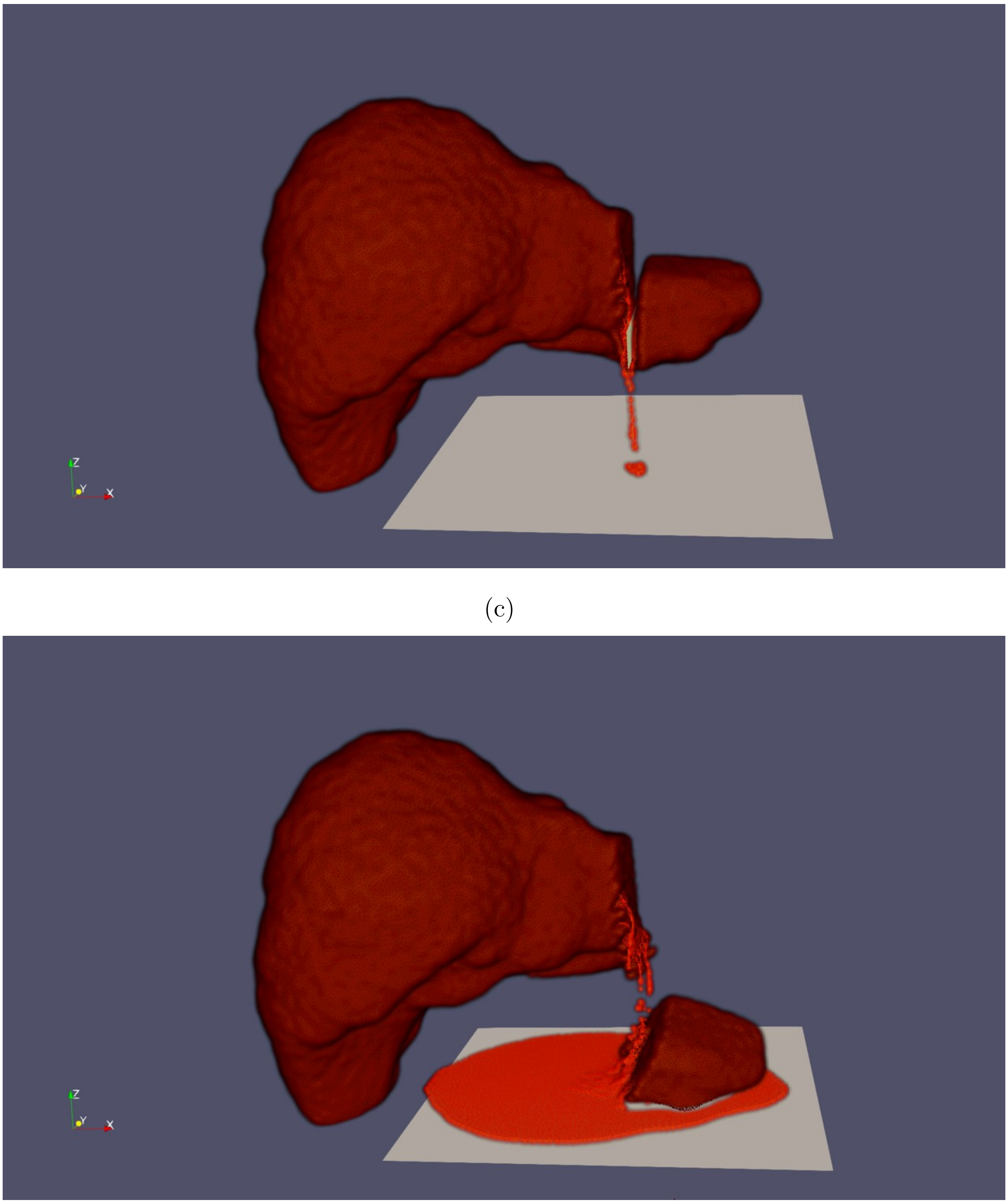
(c)

Figure 4.10: Perfused soft tissue specimen, enclose by a membrane, injured by a spherical projectile. Each panel contains a front and rear view, arranged vertically. Top panel: Just before entry. Middle panel: Just after exit. Bottom panel: After bleeding commences.



(a)

(b)



(c)

(d)

Figure 4.11: Simulation on a scalpel cutting the liver.

4.3 Simulation on Surface Tension Problems

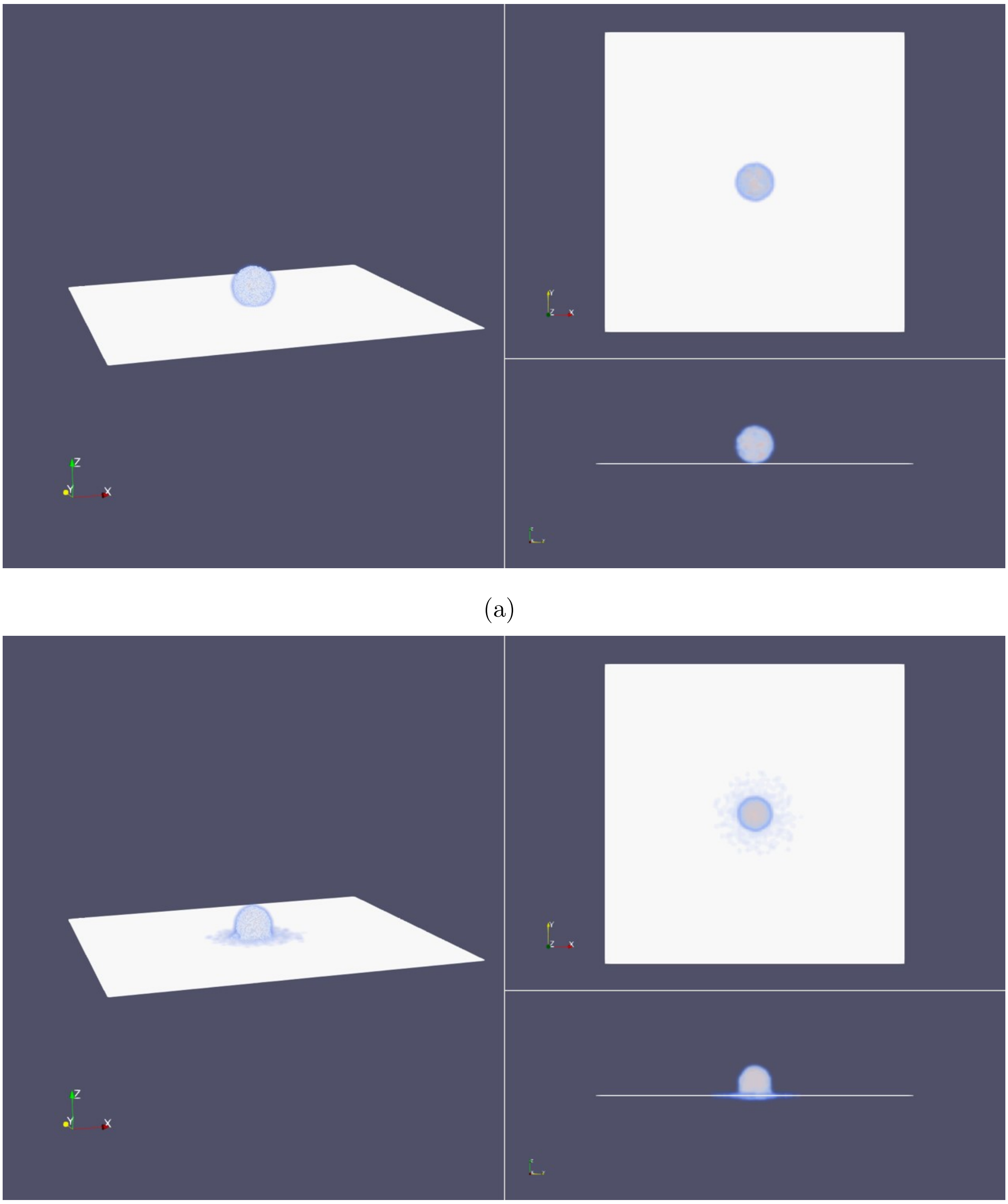
In this section, we show the simulations involving surface tension term in the implementation and demonstrate three scenarios with different material parameters. The simulations are about to deposit a liquid droplet at the center of a fabric-type porous medium. The fabric should be really thin to match the practical thickness— for example, the thickness of clothes is around 0.15 (mm). As mentioned in the chapter 2, each particle actually represents a region in the space whose dimension is related to the neighboring searching radius as well as the the particle spacing. For such a tiny scale of the fabric thickness, we have to dramatically increase the resolution of solid particles. In the simulation, the fabric is discretized with 250000 single layer particles with solid particle spacing $\Delta x_s = 0.052$ (mm) to represent the fabric with both width and length being 25.9 (mm). The searching radius is then set as $h = 1.3\Delta x_s$, such that the thickness represented by the single particle is around 0.15 (mm). The liquid droplet is discetized by 8000 liquid particles to represent 15 (μL) droplet. These parameter settings such as the size of the fabric, the volume of the droplet are chosen to match the experimental setup from our collaboration team for future validation purpose.

We utilize three sets of parameters to demonstrate three different scenarios. In each case showing later, we demonstrate the wicking process via four snapshots, which contain three sub-figures including global view (left), top view (right upper) and side view (right lower) respectively.

In the first simulation as shown in figure 4.12, we demonstrate the basic process of wicking. The final liquid volume fraction is set as $n^l_{eq} = 0.5$. The permeability is $K = 10^{-6}$ (m^2). The attraction force between liquid and solid particle is $s = 0.0728 (N/m)$. Liquid properties are same as the previous simulation that the density $\rho = 1000$ (kg/m^3), the viscosity $\eta = 0.35$ ($Pa \cdot s$). The figure 4.12(a) is the initial setup of the system: a droplet placing at the center and just above the white fabric material. The figure 4.12(b) shows the droplet is undergoing the influence of the gravity that falling toward the fabric. Later in the figure 4.12(c), the surface tension between liquid and solid dominate the flow resulting the liquid spread along the fabric. Finally, in the last one, figure 4.12(d), an equilibrium state is reached that all liquid particles do not move anymore.

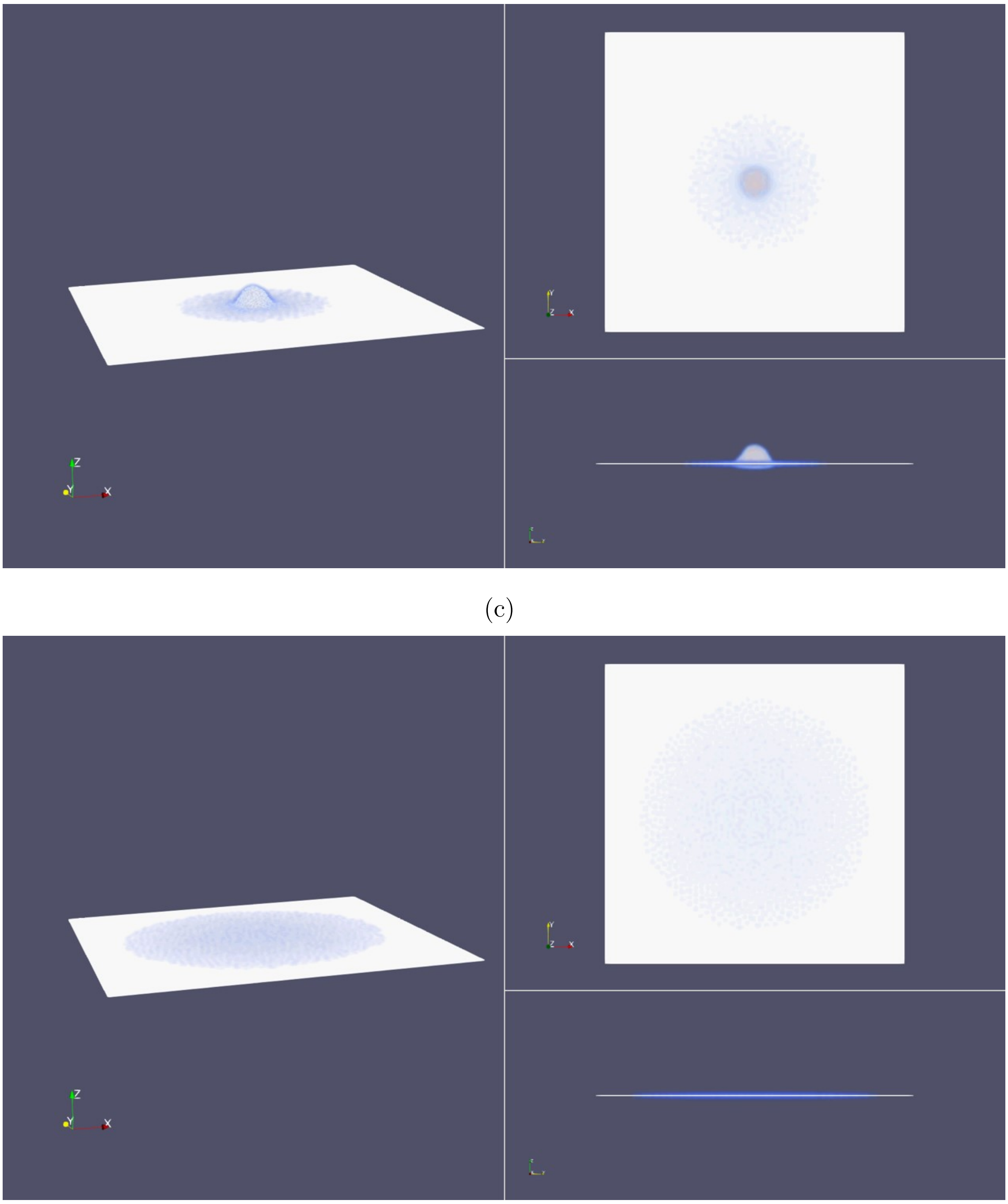
The second case is shown in figure 4.13 with four snapshots as well. The material parameters is similar to previous case but decreasing the equilibrium liquid volume fraction $n^l_{eq} = 0.15$. The setting is chosen to let the liquid droplet to spread smoothly at the fabric sheet and showing the importance the of the modification of numerical pressure in equation 3.5, where the modification can explicitly determine the stain size by solving the equilibrium volume fraction of the liquid. By just comparing the last sub-figure with it in figure 4.12(d), two different equilibrium state (stain size) is reached because of specified with the initial setting of n^l_{eq}. Apparently, the liquid finally wetting all fabric in this case, while the droplet only wet the central part of the fabric in the previous one. Similar mechanism, due to the gravity and the attraction force from the pore inside the fabric, the droplet falls down initially, then affected by the friction generated by the fabric resulting to change the flow direction to spread in the fabric plane. The wicking mechanism keeps it spreading outward until the equilibrium state, in this case, wetting all fabric, was reached.

The third case is shown in figure 4.14, while only one parameter, strength of the surface tension, reduces ten times smaller. The parameters chosen here is to make the droplet partially penetrate the fabric. Compared with the second case shown in 4.13, the ratio between permeability and the surface tension is larger here, indication the material is more permeable such that the liquid may accumulates below the liquid depositing point first, and then detach with the fabric when the total droplet mass beating the attraction force from the surface tension.



(a)

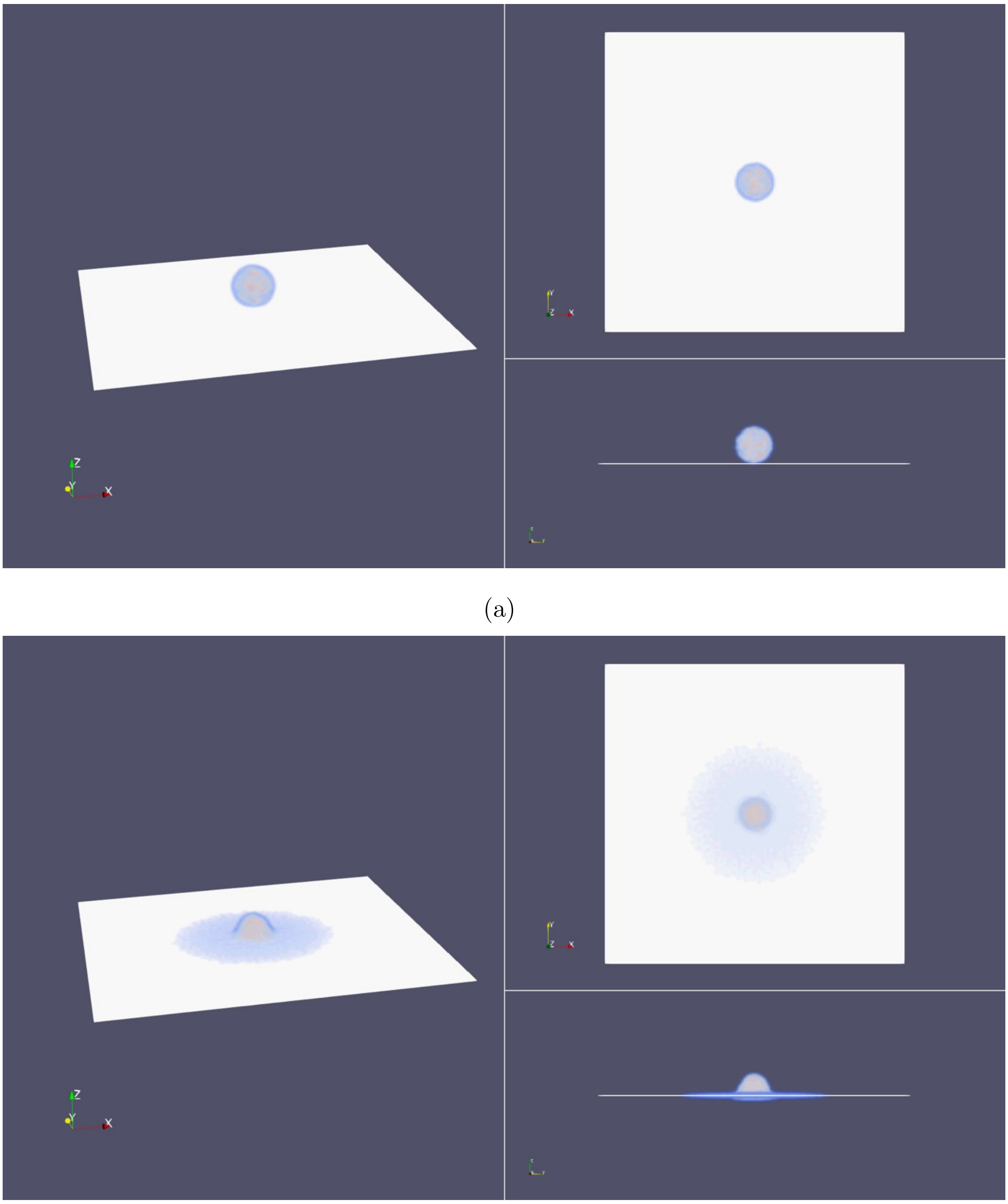
(b)



(c)

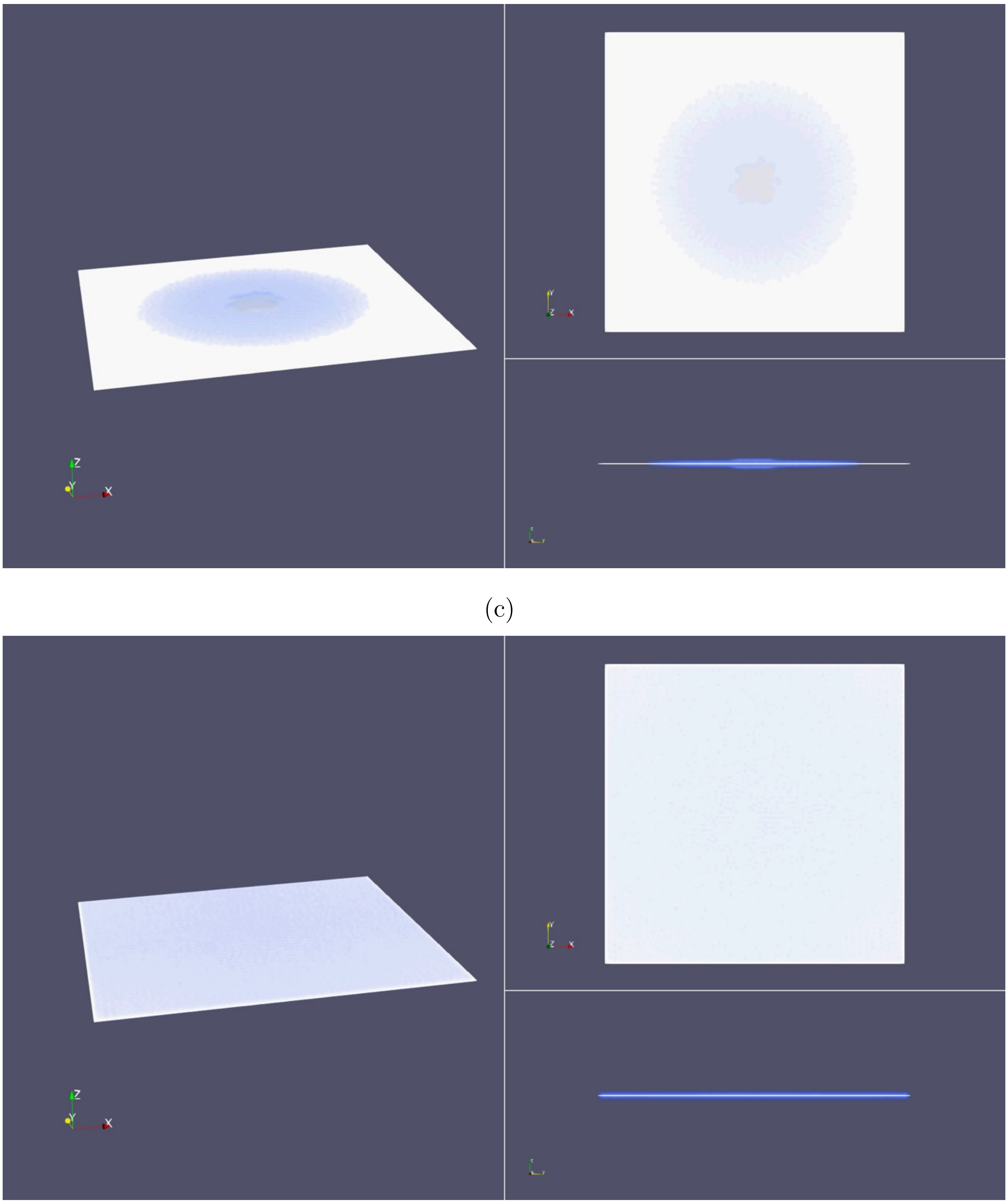
(d)

Figure 4.12: Liquid droplet spreading on a fabric.



(a)

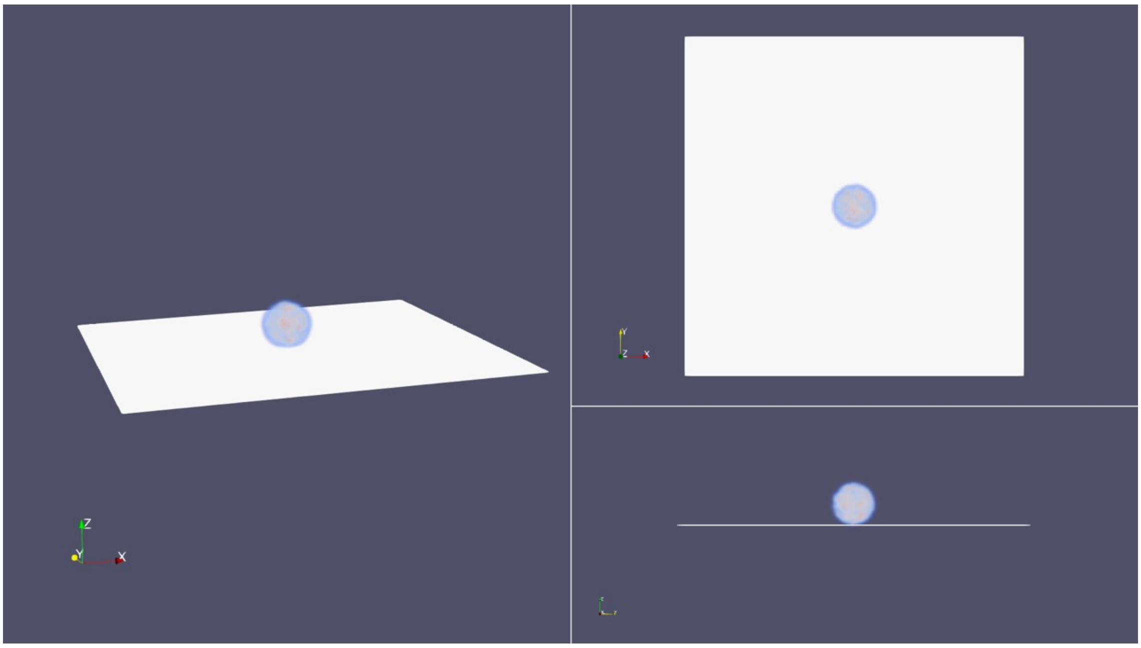
(b)



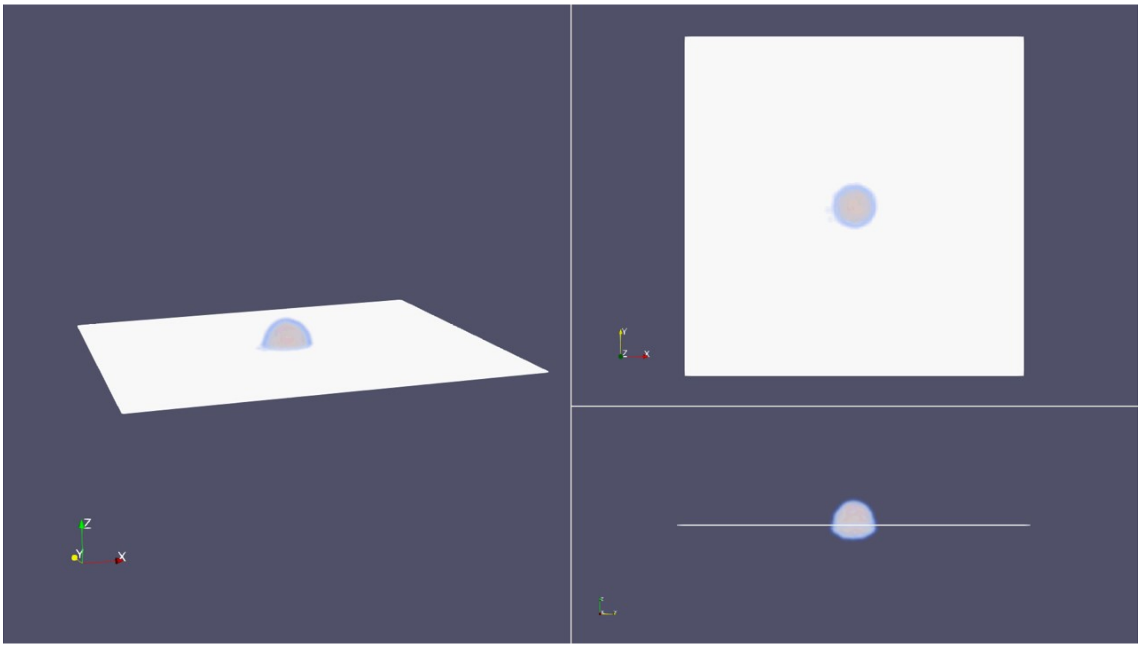
(c)

(d)

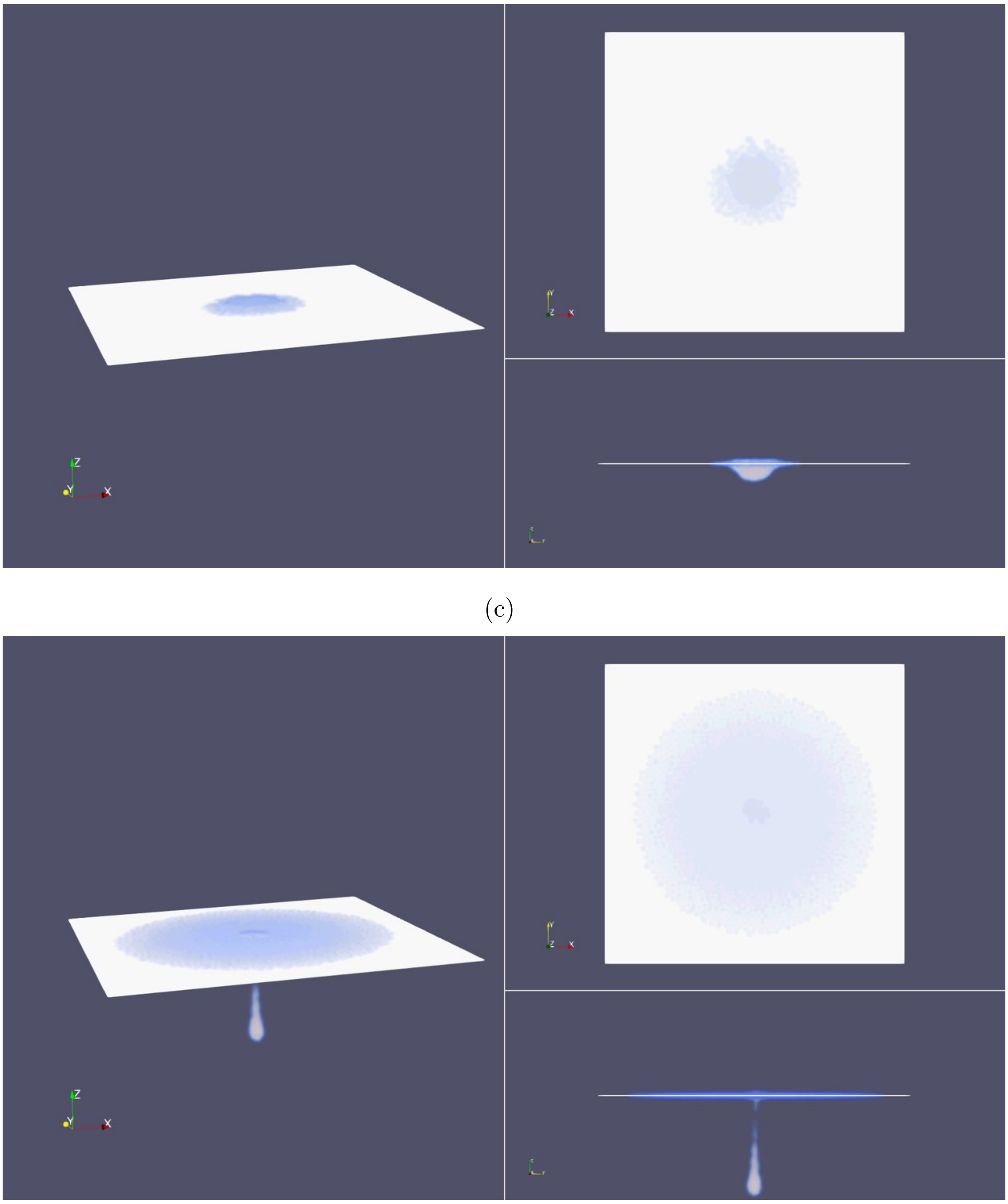
Figure 4.13: Liquid droplet wetting through a fabric.



(a)



(b)



(c)

(d)

Figure 4.14: Liquid droplet on a fabric (partially penetrate).

CHAPTER 5

Data Science on Smoothed Particle Hydrodynamics

5.1 Introduction

Dynamic mode decomposition (DMD), being introduced by Schmid [Sch10], was originally utilized to decompose large numerical and experimental data sets into different spatiotemporal modes with their own complex number frequencies that represent oscillation and growth/decay rate on each coherent structure. DMD performs an elegant way to extract the features of dynamic information from data without any knowledge of governing equation but just obtain a best-fit operator through linear algebra manipulation. Extension of application of DMD on nonlinear system was usually linked with the Koopman operator [Koo31] that mapping a finite dimensional nonlinear systems to an infinite-dimensional linear systems, such that improving the ability to capture the features of nonlinearity via linear operator in DMD [CTR12, TRL$^+$13, KBBP16].

The goal of applying DMD on a dynamic system is to obtain a best-fit linear operator $\boldsymbol{A}$ to relate two data matrices $\boldsymbol{X}, \boldsymbol{X'}$, containing only one step forward of snapshots of the system state in time and placed them column by column, while each row represented one specific state of the system. Usually, the data set is really large so that obtaining operator $\boldsymbol{A}$ directly is impossible. Instead, DMD establishes a standard procedure to achieve full dimensional eigenvectors of $\boldsymbol{A}$ as following steps: performing singular value decomposition on the data matrix $\boldsymbol{X}$, projecting data matrix $\boldsymbol{A}$ on the basis of truncated left-singular vectors, applying eigendecomposition on the projecting data matrix, and reconstructing full eigenvectors of data matrix $\boldsymbol{A}$ from the eigenvectors of projecting data matrix, right-singular vectors and singular values.

Although DMD has been widely applied into diverse fields [GK14, BSV$^+$15, PE15, KBBP16, ED16, MK16, BJOK16] to extract the features of the systems as an entirely data-driven analysis, all components existing in state vector of each column of the data matrix $\boldsymbol{X}$ have their own intrinsic well-structured relation. For example, the velocity in the flow field no matter from numerical results or experimental results, representing a specific position—may be the grid position or a measurement position—on the flow field. These position information will not list into the state vector since position is always fixed and will be redundant information in the data matrix. Same situations can be found in other DMD application such as pixel point data in video/picture processing [GK14, KGB16, ED16], data from fixed-position sensor in robotics [BSV$^+$15] and neuroscience [BJOK16], flu activity in a certain area [PE15], and portfolio data of different companies [MK16]. Each state variables in these applications all have naturally association and implicitly imply the space information with each other.

In fluid dynamics, these nature space information comes from the Eulerian description of flow field that representing the focus on the control volume rather than individual fluid particles. However, there is another branch of computational fluid dynamics (CFD) that solving the governing equations via Lagragian description, where the observer keeps following each individual fluid parcel as time evolved. In such Lagragian method, the position of each point where we solve the governing equation at each time step is different and considered as an explicit state variables. Smoothed particle hydrodynamics (SPH), in particular, is a well-known particle method first used in the field of astrophysics [GM77] but later applied to general continuum theory [LP91, Mon00, GMS01, LL03, MCG03, Kel06]. This method divides the continuum material into a set of discrete particles carrying their own physical quantities, from which the continuum fields can be locally interpolated via a particle kernel function. The method has been extended in recent years to materials of various constitutive behaviors such as non-Newtonian fluids [SL03, VM08], and viscoelastic materials [ET05, RMH07] as well as turbulent flow [VI07, Mon11, DMACM17].

SPH method has its unique strengths such as dealing with complex boundary problems without requirement of grid refinement, easier algorithm implementation and computational parallelization, but small time step size restriction as well as analysis on the SPH simulation results still require other powerful tool. Performing DMD on SPH simulation results is a straightforward idea and expect to extract the nature behind the full flow field. However, to the best of the author's knowledge, DMD on particle method is still missing over the recently booming development of data science. The state variables here, from the SPH results, should contain not only velocity field but also position vector and make the size of state vector twice larger. The effect of larger state vector on the rank-r truncation reduced order model at SVD step are also vague. Understanding how DMD working with explicit position vector will not only limits in the application in the fluid dynamics but also extends DMD to other problems such as human behaviour involved with geographical data, biological cell functionality at the different location of bodies.

In this work, two simulations from SPH solver will be utilized to generate data matrices. The position vectors and velocity vectors of each particle in SPH are set as the components of state vector. In first case, some quiescent liquid are placing in a rectangular container under gravity force with left side wall of container having horizontal oscillating motion. After the wall starts to move, the surface wave generated on the top of the flow and whole liquid are oscillating. This case has only one mode in order to test how well does the ability of DMD to extract features on particle method. The second case is a broken-dam problem that all liquid are placed at the left half side of a tank with a fictitious plate in the middle of it. As the simulation starts, the plate is setting to disappear immediately, causing liquid collapse and move to the other side of the tank, resulting in surface wave propagation and reflection of the liquid. There is no natural mode in this problem but we expect DMD can decompose the complex system into different dynamic mode to be analyzed. There are two main objectives in this work. First, learning how to apply DMD on particle method such as obtaining a best input data information from SPH results, minimum requirement of time series snapshots. Second, using the knowledge of linear operator $\boldsymbol{A}$ to improve the SPH method with higher computational efficiency, and data compressing that reducing the computational memory storage requirement.

5.2 Dynamic Mode Decomposition

The basic idea of DMD is to obtain a linear operator for following two data matrices

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & ... & x_{m-1} \\ | & | & & | \end{bmatrix}, \boldsymbol{X'} = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & ... & x_m \\ | & | & & | \end{bmatrix}, \tag{5.1}$$

where x_m represented the $m-th$ snapshot of system with state vector x. The relation between them is

$$\boldsymbol{X'} = \boldsymbol{A}\boldsymbol{X}. \tag{5.2}$$

Usually, the data matrix $\boldsymbol{X}$ and $\boldsymbol{X'}$ are with large dimension, such that we try to find eigenvectors of $\boldsymbol{A}$ instead of obtain $\boldsymbol{A}$ via $\boldsymbol{A} = \boldsymbol{X'}\boldsymbol{X}^{-1}$. The first step is performing SVD on $\boldsymbol{X}$,

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{*}, \tag{5.3}$$

where $\boldsymbol{U}, \boldsymbol{V}$ are left-singular vector matrix and right-singular vector matrix respectively;$\boldsymbol{\Sigma}$ is singular values matrix. Sometimes, We can truncate the SVD results by only taking first r component of singular value as $\boldsymbol{X} \approx \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{V}}^{*}$, called rank-$r$ truncation. Next, we project matrix $\boldsymbol{A}$ to $r \times r$ matrix $\widetilde{\boldsymbol{A}}$ on eigenvector $\widetilde{\boldsymbol{U}}$,

$$\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{U}^{*}}\boldsymbol{A}\widetilde{\boldsymbol{U}} = \widetilde{\boldsymbol{U}^{*}}\boldsymbol{X'}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Sigma}}^{-1}. \tag{5.4}$$

After having above $\widetilde{\boldsymbol{A}}$, we perform eigendecomposition on it via

$$\widetilde{\boldsymbol{A}}\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Lambda}, \tag{5.5}$$

where $\boldsymbol{W}, \boldsymbol{\Lambda}$ are eigenvectors matrix and eigenvalues respectively. Last, we can reconstruct full dimensional eigenvectors of $\boldsymbol{A}$ by

$$\boldsymbol{\Phi} = \boldsymbol{X'}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{W}. \tag{5.6}$$

The columns of $\boldsymbol{\Phi}$ are DMD modes of the system. Then the representation of data in terms of DMD is given by

$$\boldsymbol{X} = \sum_j a_j \boldsymbol{\phi}_j \lambda_j, \tag{5.7}$$

where $|a_j|$ can be obtained by solving initial value problem and considered as the amplitude of each DMD mode. If all components in input state vector are velocity field, then the square of each component of the singular value in equation 5.3 represent the flow energy of each mode like proper orthogonal decomposition (POD).

5.2.1 DMD on SPH

In SPH, the time step size is usually very small ($\Delta t < 10^{-4}$). It is because switching the incompressible constraint into weakly compressible flow (equation 2.46) caused the particle require small time step size to relax the small disturbance of density like the action of the actual pressure force. However, the flow information in every small time step size require huge computer memory space to store. Therefore, we usually define another larger time interval as time step size of each frame, at which we store current data such as particle position for future visualization purpose. To perform DMD on the SPH data based on the frame size sounds attractable. Once the linear operator $\boldsymbol{A}$ being found, then the operator is so powerful that can evolve the system vectors without the constraint of Courant number and Fourier number that limiting the largest time step size of the computational method. In addition, the extracting mode information from DMD can also provide some benefit on data storing as the sense of data compression.

5.3 Results of DMD on SPH

5.3.1 DMD on Oscillating Wall Simulation

In this section, we first performed a DMD on oscillating wall problem. Five snapshots for a half period of this problem was shown in figure 5.1. 1980 quiescent particles were placed in the container. The left wall then moved oscillatory for 50 second with frequency $f = 0.5$ and the amplitude was set as a quarter of container width. The liquid inside the container was squeezed toward right as the left wall moves to the right during the first half period. After few periods, since energy was continuously input into the system, the wave amplitude of the liquid would become larger and larger near both side walls. The trajectories of three chosen particles for fifty seconds simulation are shown with different colors in figure 5.2, while the trajectory of each particle is the key feature we want to extract by the DMD.

The first question to be answered is the form of the state vector of each snapshot. In DMD, there is no certain type regulation of the input vector. It could be the results from experimental measurement or numerical simulation. The input state vector sometimes may not have the same size as the complete state vector that we usually used to describe the whole system. For example, for a well-known choatic dynamics, Lorenz system, there are three components in the system state vector. However, we may use only one variable as the input in the HAVOK analysis but still extract the features of the dynamics as shown in Brunton's work [BBP$^+$17]. On the other hand, a fluid solver like SPH method, it relies on the whole set of the system state vector: in SPH case, the position vector and the velocity vector of each particle. In addition, another key point of the input data matrix is the time step size for each snapshot, which could be chosen arbitrary but only requiring the least sampling rate indicating by Nyquist frequency. With these background knowledge in mind, we first examine how to formulate input data matrix and evaluate the quality of the data matrix by reconstructing the flow simulation with extracted DMD modes and corresponding eigenvalues. We thus figured out the input data matrix should be only contained position vector. We did try the following cases as shown in table 5.1. Case one is inspired by the intrinsic nature of SPH solver, where the input vector is same as the input information for each time step of SPH method. The idea of case two is from the common usage of DMD onto compressible flow, where the density and the energy are two additional physical quantities of the flow. Note that the density would not be a constant and computed by equation 2.46 in SPH method as we loosen the incompressible constraint into weakly compressible flow. The last two cases are just to mimic the measurement from experiment with only input data matrix with either position vector or velocity vector. Surprisingly, we found the DMD mode extracting from third case has the better performance. We conjecture that it is because we evaluate the performance by the reconstruction of the flow simulation, such that the position information at each time step is much more important than the velocity information. Besides, the time step size of the snapshots is not as small as that for SPH solver, meaning that the velocity information that only valid at such small time interval is redundant. Therefore, focusing on the mode that extracting only from position information sounds a reasonable choice of the input state vector. The time step size of each snapshots have to compromise on the memory of the computer resource. We found it is not necessary to limit the time step size as small as it of SPH solver. Choosing the time step size of each snapshots as the same order of magnitude of the frame size works well in our preliminary trial test. For all results being shown later, the particle position vectors at each frame are served as the primary input state vector of the DMD methods.

Table 5.1: Components of Input State Vectors

case	density	position	velocity	momentum	energy
1		V	V		
2	V	V		V	V
3		V			
4			V		

The eigenvalues extracted by the DMD is shown in the figure 5.3. There are 1980 particles in the simulation and the input state vector only contains two dimensional position information, such that the full DMD modes number is 3960. These all eigenvalues are plotted in a complex plane and denoted by red if the absolute value of it is larger than one. These eigenvalues of unstable mode are located at the right side of the complex plane, meaning that as we reconstruct the flow with equation 5.7, the magnitude of position in state vector will have unreasonable amplification after few iterations. The eigenvalue distribution disperses at the left complex plane but approaching to the unit circle as the growth rate increases at the right complex plane.

We can easily decompose the complex eigenvalue into growth rate and the frequency, which corresponded to the real part and imaginary part respectively. Shown in figure 5.4 is the mode frequency and its amplitude $|a|$ that obtained from equation 5.7]. The mode frequency span large range but there are only few primary modes with high amplitude, which indicates reduce-order model may be applied here and with non-important modes truncation. Another feature from eigenvalue of DMD is the relation between growth rate and the mode amplitude, which is shown in figure 5.5. This plot explicitly display that the growth rate of most DMD modes are less than one and almost collapse in the left y axis in the plot.

For the same type of flow, oscillating wall problem, we dimmed the oscillating amplitude of the side wall, comparing the eigenvalues of it with the eigenvalues in first case, showing the results in figure 5.6. These two cases have a similar pattern on the eigenvalue distribution, indicating the consistency of DMD performing on the same type of flow problems.

After performing DMD on SPH method, the physical interpretation of each mode is illustrated in figure 5.7. Recall that the input state vector is just the position of each particle. The superposition of each mode reconstruct the position vector. Therefore, each mode actually play a role similar to the concept of displacement of the particle. In the figure 5.7, we plot the different mode of only one selective particle at initial time step, where the red diamond marker represents the average position of the particle during entire simulation, and red filled circle indicates the current position of the particle. Blue dots connecting with red line between current particle position and the average position divide different DMD modes and the red line is the magnitude of each mode, showing the displacement contribution from corresponding mode. As the time evolves, the length of each mode will change, implying the dominant mode at different time step is different.

From the eigenvalues plot in figure 5.3 and figure 5.10, we notice that there exist few unstable modes such that the flow field reconstructed from DMD modes will go to infinity after few iterations. In order to reconstruct the flow field correctly without occurrence of instability. An required modification that have to subject into all unstable modes is to turn these modes into neutrally stable. In the mathematical description, the modification is just $\mu_m = \frac{\mu}{|\mu|}$, where the subscript m represents modified eigenvalue. The reconstruction flow field is shown in figure 5.8, where the blue particles is the results from SPH solver, and the red particles is the reconstruction flow filed. Note that in the figure, the snapshots of the system is the first half of the tenth period, illustrating that the reconstruction flow field is still stable and captures all particle well even after few time evolution.

To understand the possibility of reduced-order model when applying DMD on SPH method, we borrowed the mode energy analysis from POD method, that the energy for each mode is corresponding to its square of the singular value. The DMD also has the singular value decomposition at the first step such that we can utilize the singular value obtained here to evaluate the mode energy. The cumulative plot of the square of the singular values is shown in figure 5.9. From the figure, we can observe that the energy of first five hundred modes are around ninety nine percent of total energy. We later truncated the last half modes in equation 5.4 and then follow the standard procedure to obtain remaining 1980 modes, plotting the eigenvalues of then in the figure 5.10. The red dots still represent the absolute value of eigenvalue larger than unity. The eigenvalue pattern here seems to shrink into right complex plane. The reason for that is the modes with negative growth rate (real part of eigenvalue) are less important and would vanish after few time step iteration, so that they are the targeting mode to be truncated. Reconstruction of the flow field with these truncated was compared with figure 5.8 but have some deficiency on the spatial resolution that causing particles aggregation at some locations.

5.3.2 DMD on Broken Dam Problem

With the experience of applying DMD analysis on SPH method, we then tried the same technique on another flow simulation: broken dam problem. The snapshots of broken dam problem are shown in figure5.11 and figure 5.12. 3399 numbers of particles were initially located of the left side of the container with a virtual wall located at the center to keep particles at the left as shown in the first snapshot in figure 5.11. At time $t = 0^+$, the virtual plate at the center was removed immediately, causing the static fluid becoming unstable and tend to move toward right due to gravity. After few second, the liquid reached the right end of the container, and suddenly changed the flow direction to bounce back to the left. The surface wave then moved back and forth inside the container until all energy was dissipated due to viscosity. Again, the blue particles in the figures are the results from SPH method, while the red particles are the reconstruction from the DMD analysis with modification of the large eigenvalue as described in previous section. The reconstruction is still promising and shows the great ability of DMD to extract the physics from a certain type flow and then reconstruct it. The eigenvalues of broken dam problem is plotted in figure 5.13. Red dots are the cases of absolute value of eigenvalue larger than one. It is difficult to have a deep insight on the eigenvalue pattern, but the pattern is slightly different with that of the oscillating wall problem in figure 5.3, meaning that the features extracted from DMD indeed adaptive to different problems. Also, the most eigenvalues are away with the unit circle, indicating the certain flow is more stable than than oscillating wall problem, which matches our knowledge that the flow in broken dam problem would go back to quiescent status due to viscosity dissipation without any additional energy input.

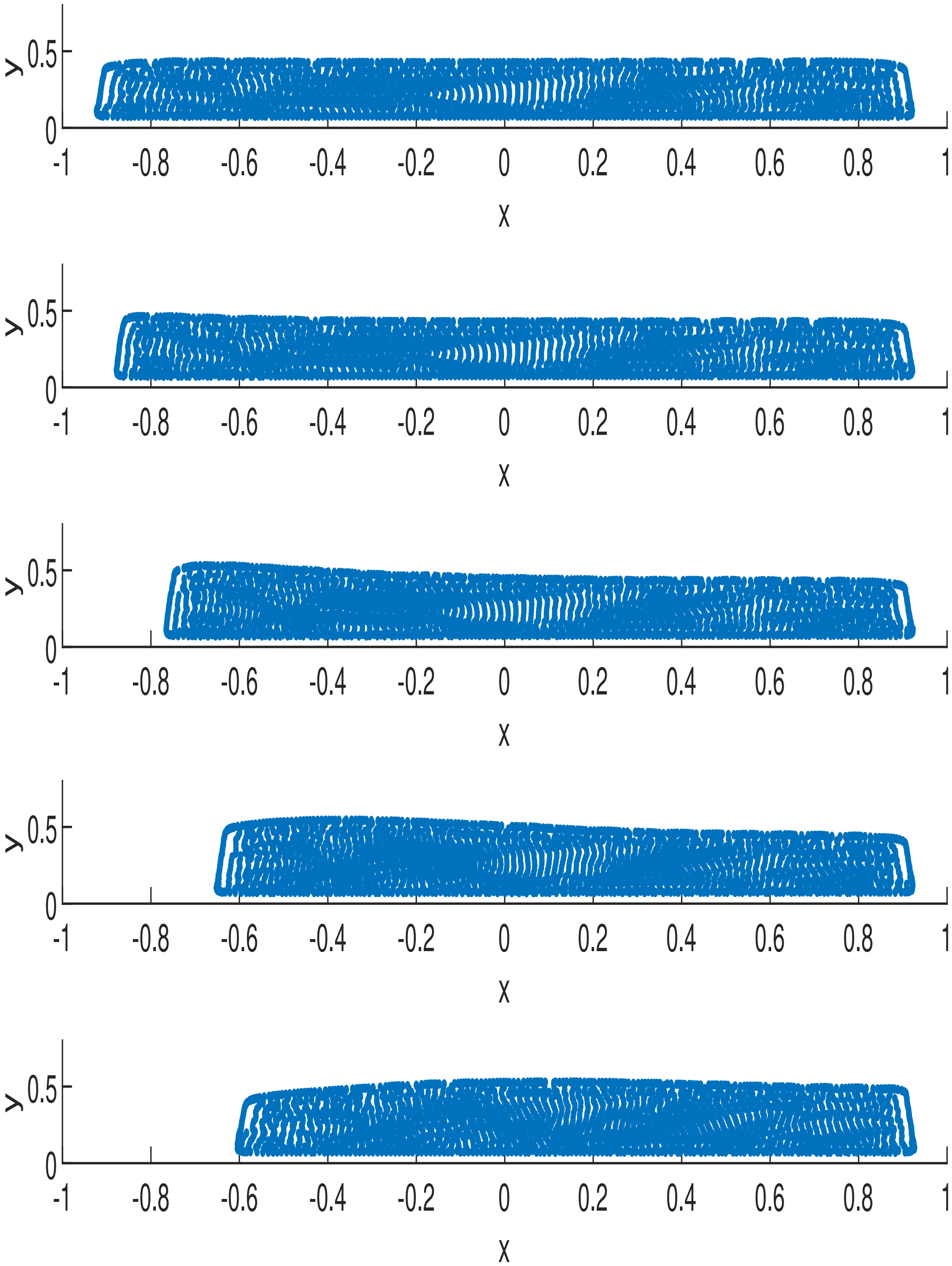


Figure 5.1: Snapshots of oscillating wall problem in half period

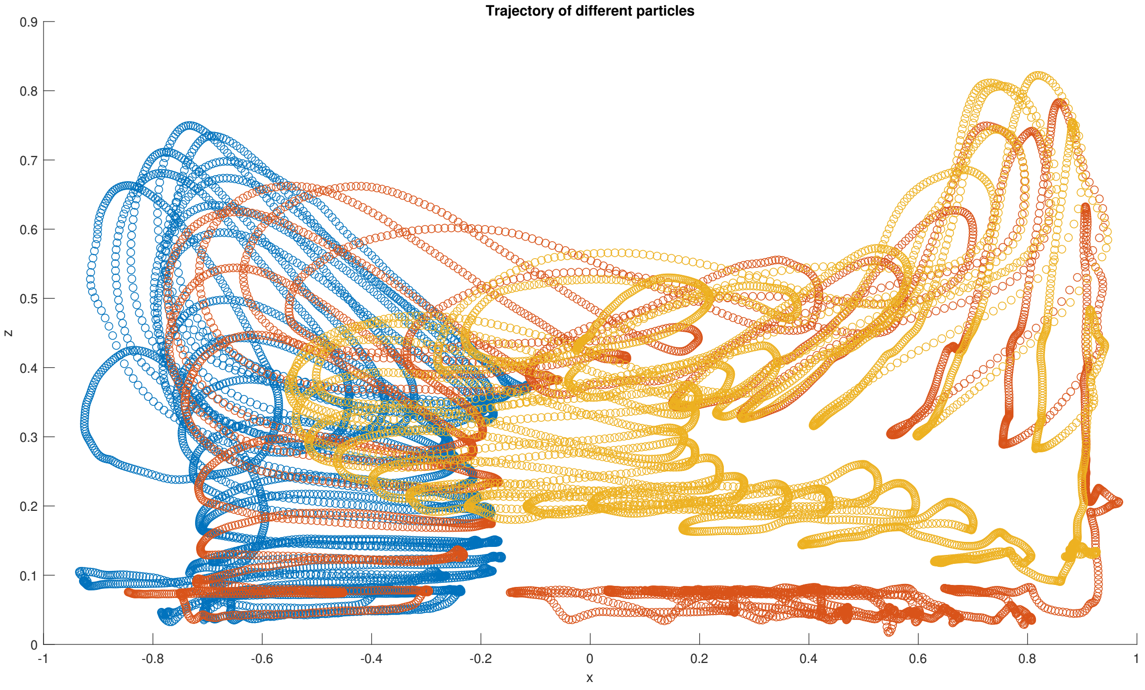


Figure 5.2: oscillating wall problem

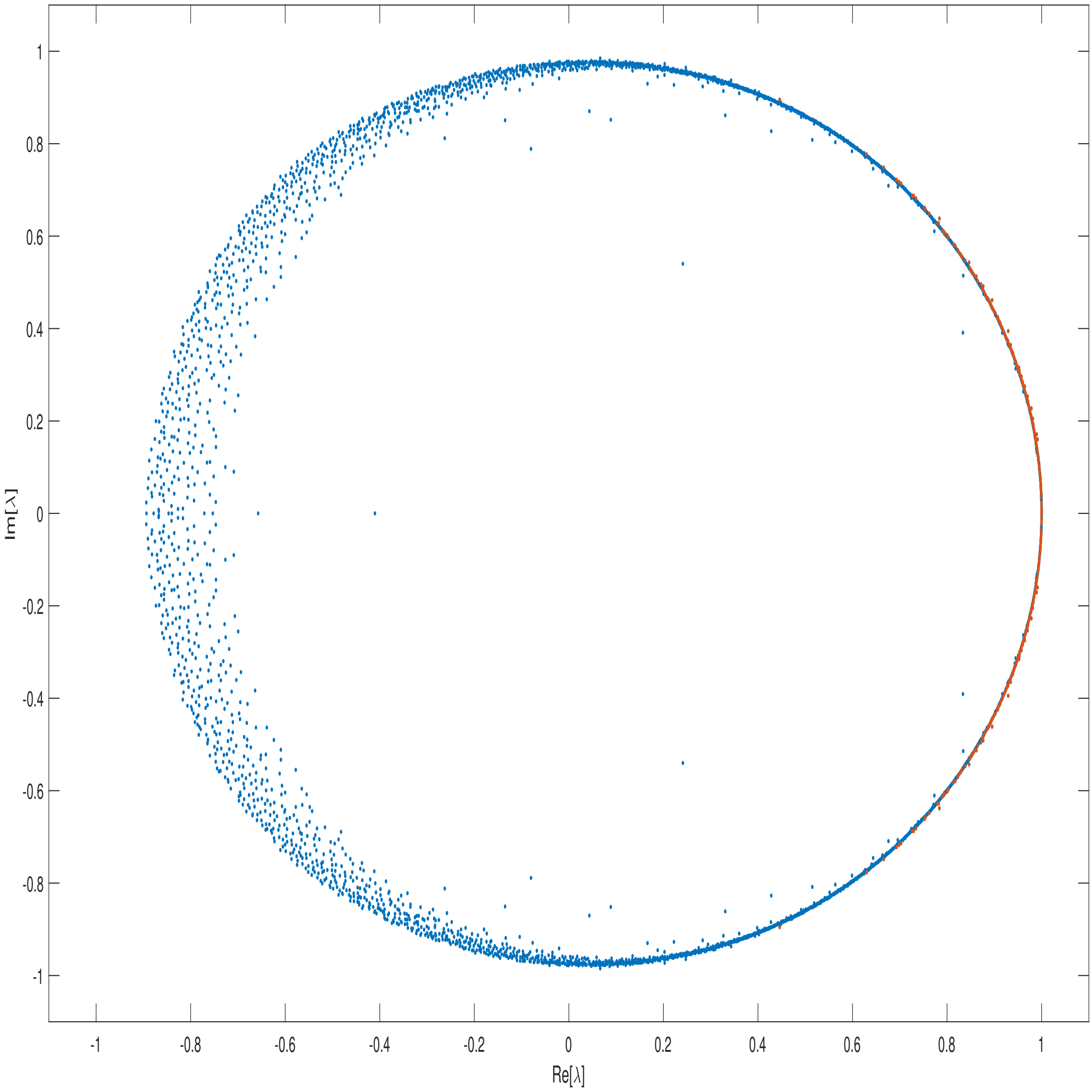


Figure 5.3: Eigenvalues of oscillating wall problem.

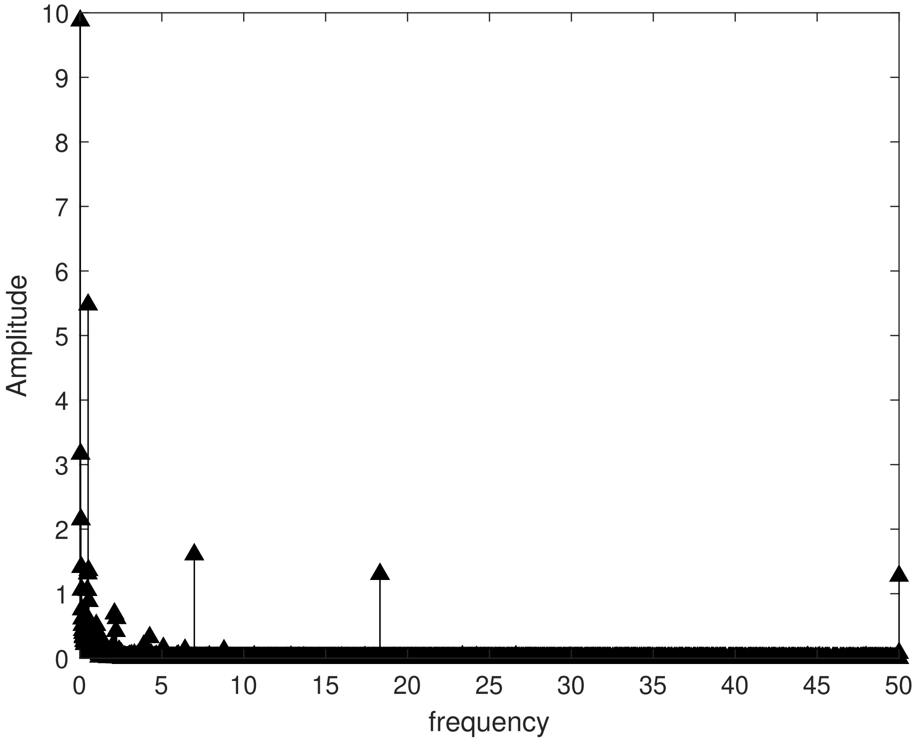


Figure 5.4: Mode frequency of oscillating wall problem.

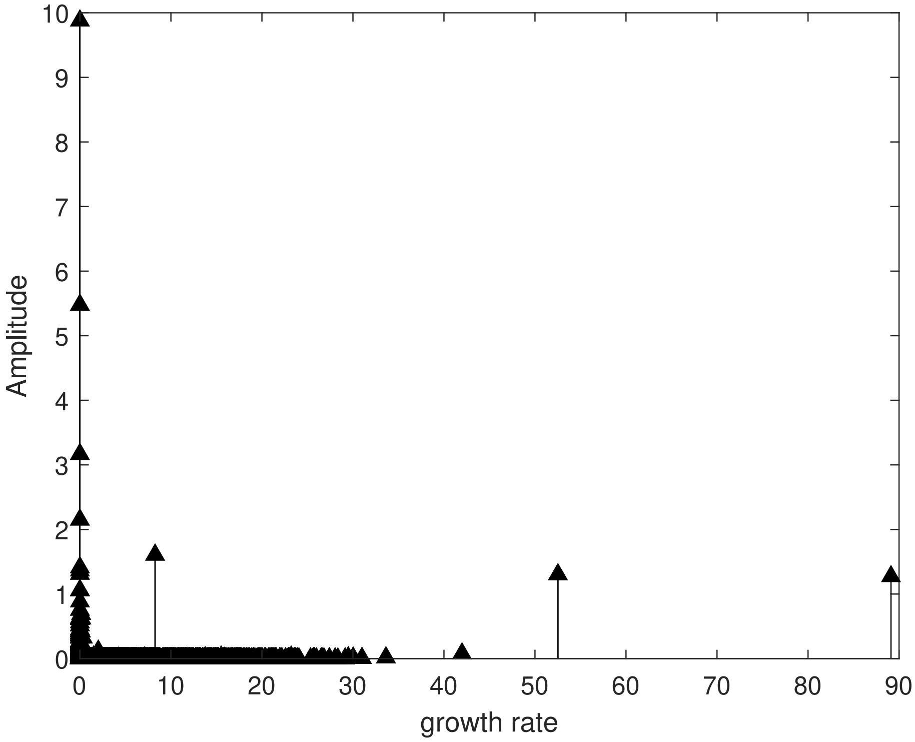


Figure 5.5: Mode growth rate of oscillating wall problem.

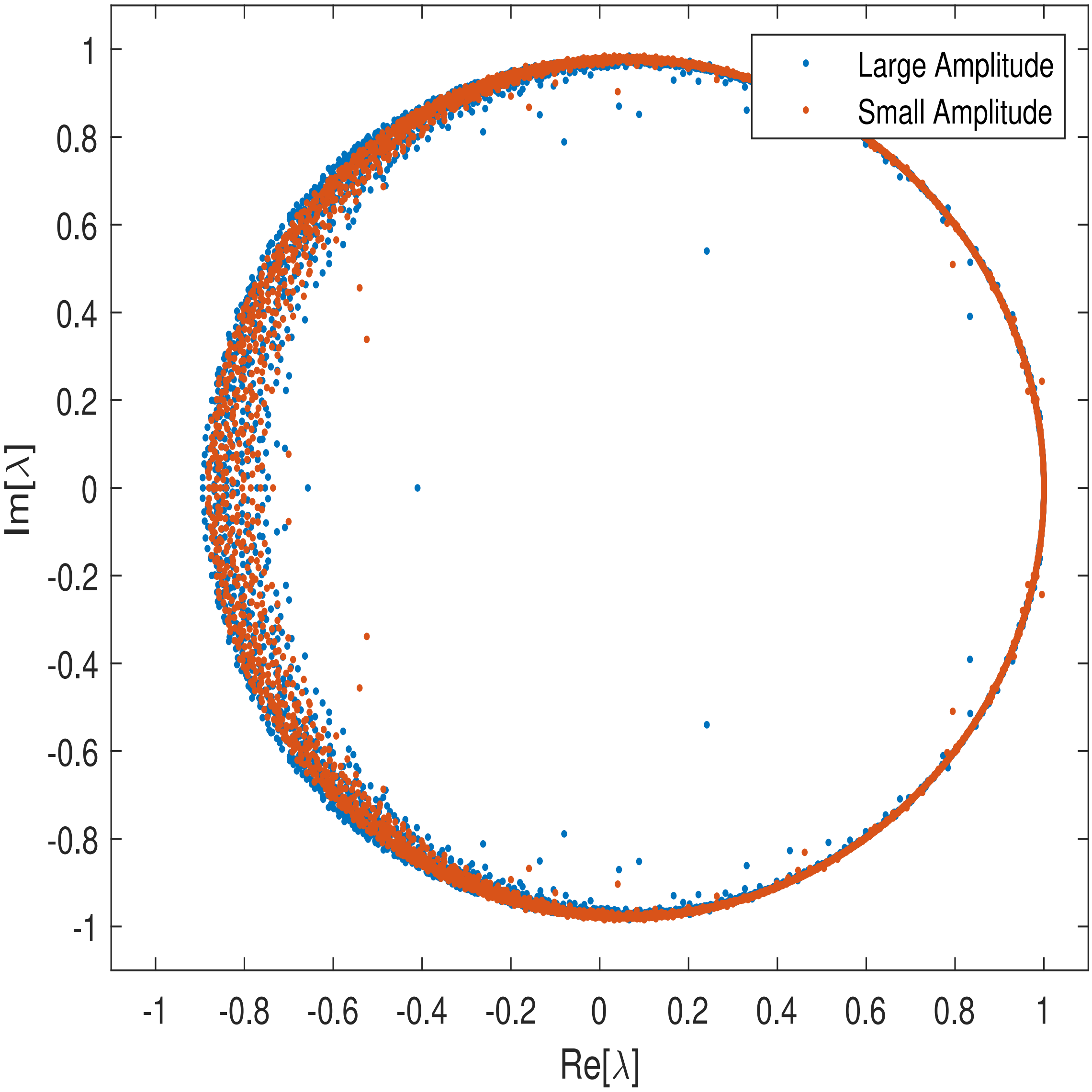


Figure 5.6: Eigenvalues of two different amplitudes oscillating wall problem.

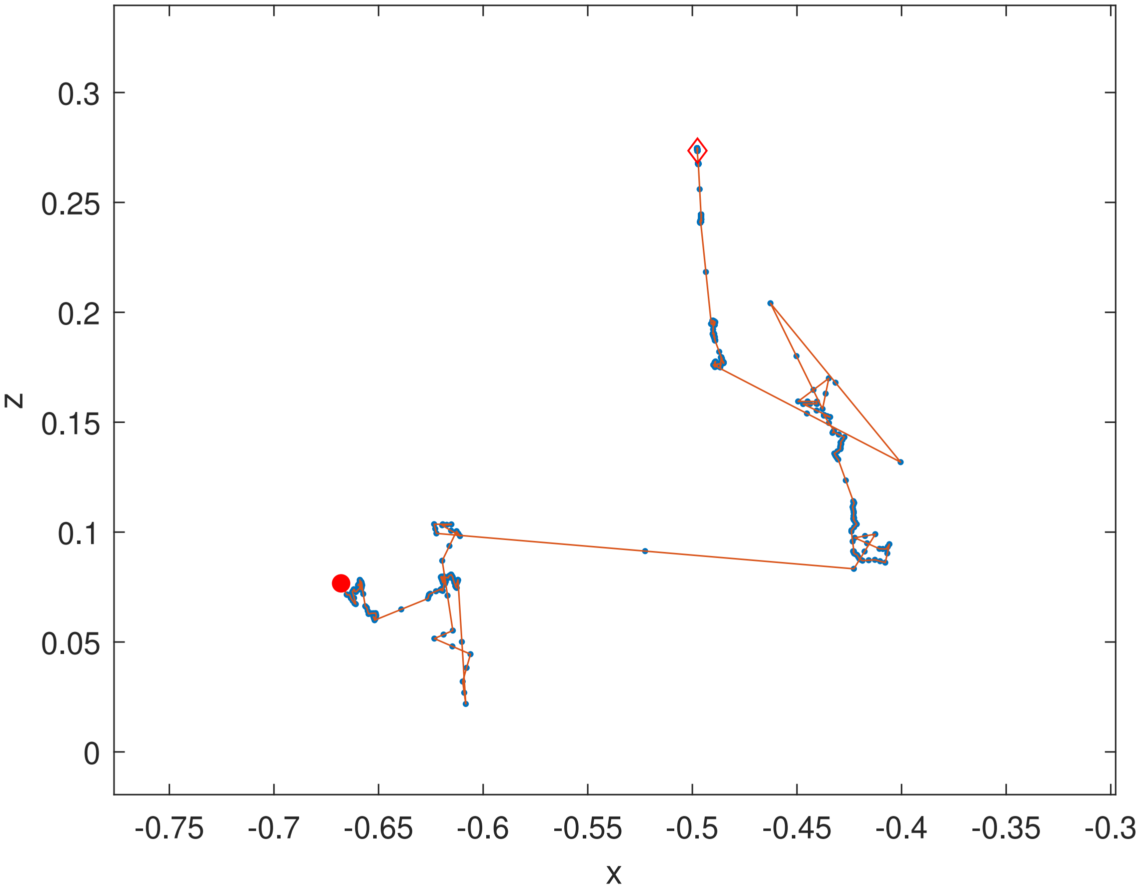


Figure 5.7: Physical interpretation of the DMD mode.

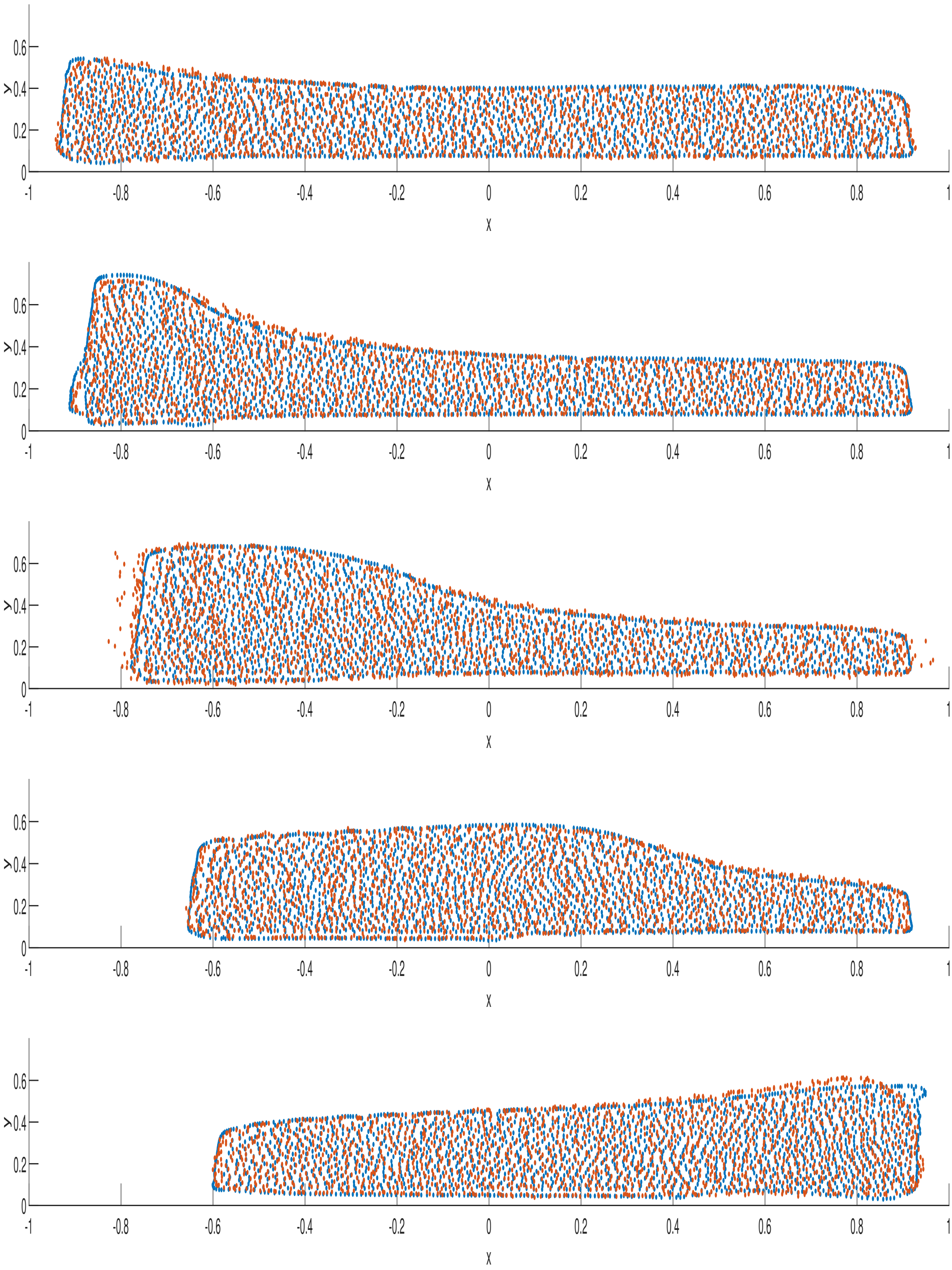


Figure 5.8: Comparison of original SPH results (blue) and the reconstruction flow field from DMD (red).

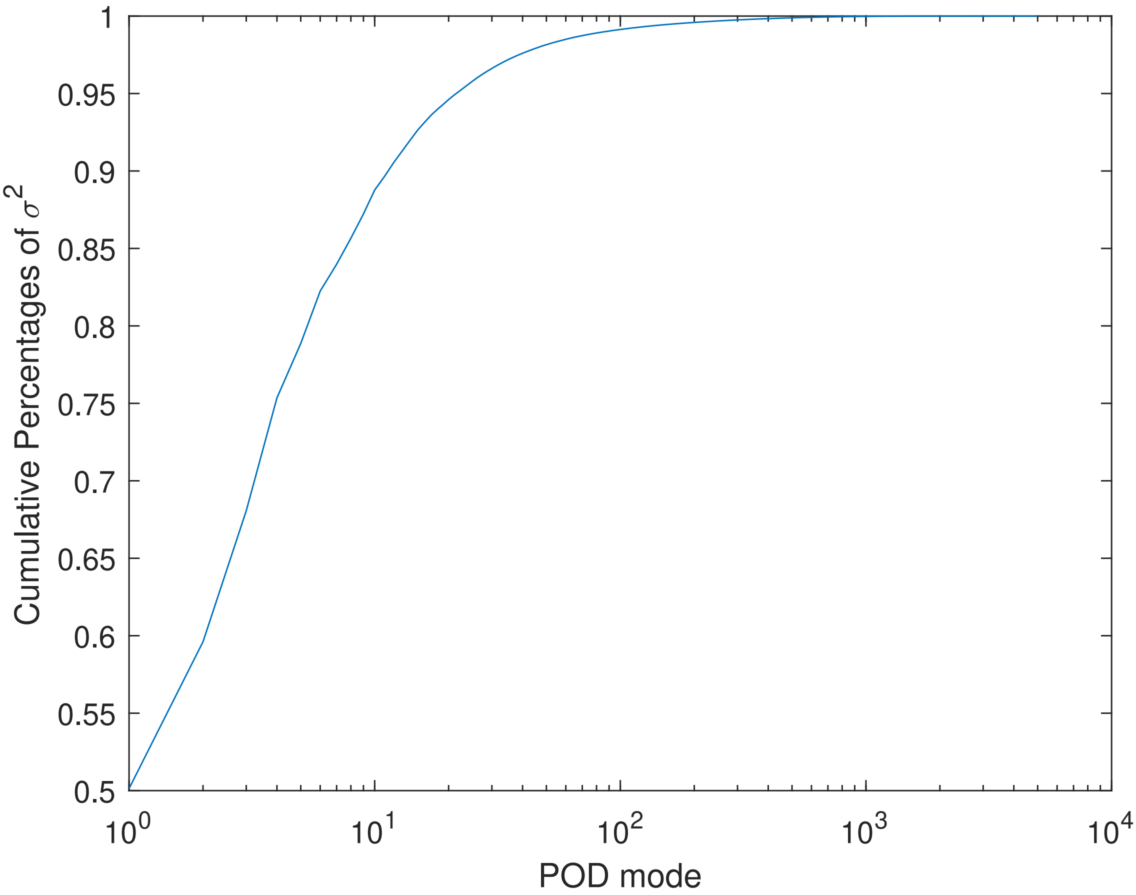


Figure 5.9: Percentage of cumulative energy of all POD modes.

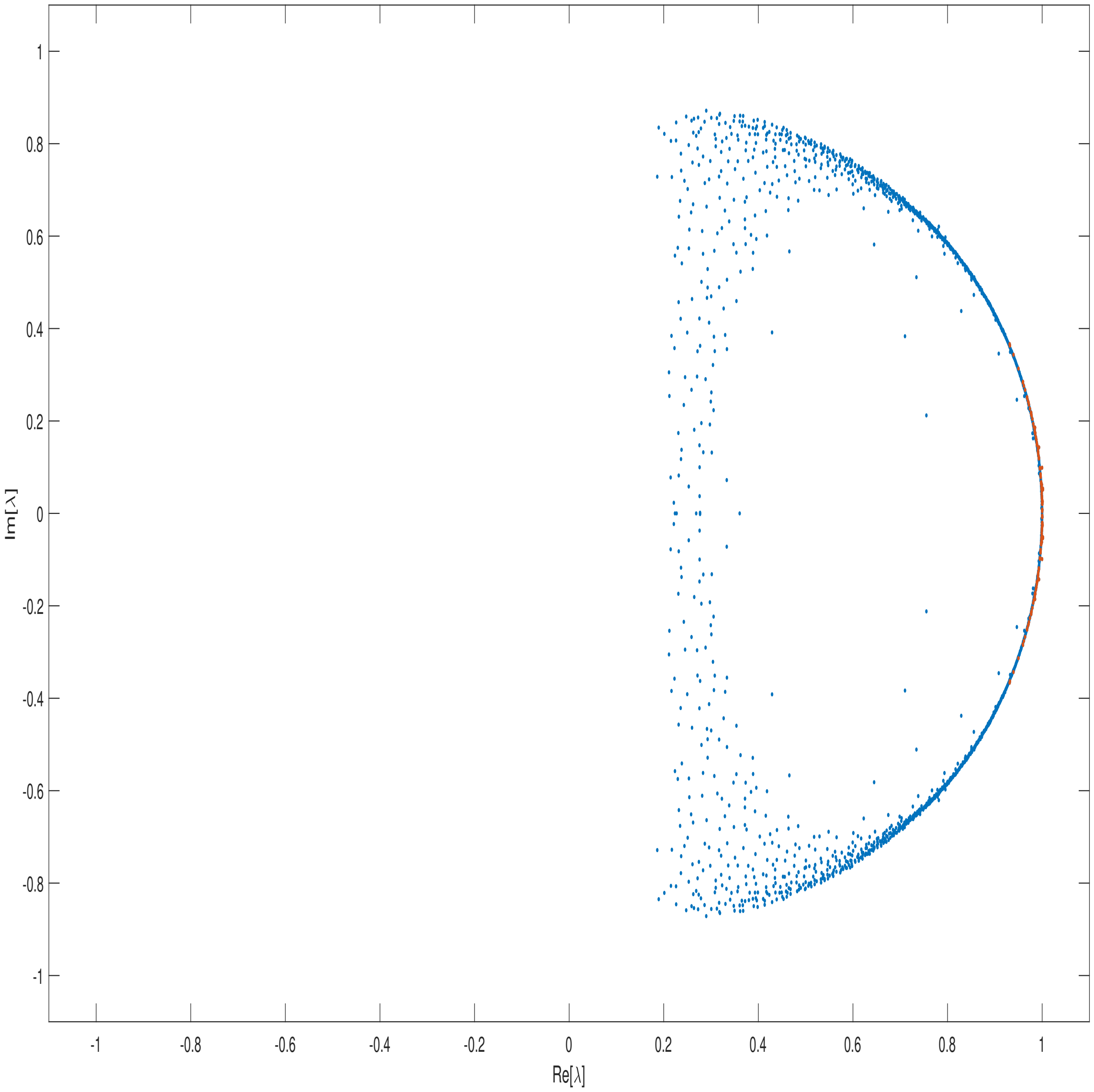


Figure 5.10: Eigenvalues of Reduced-Order model

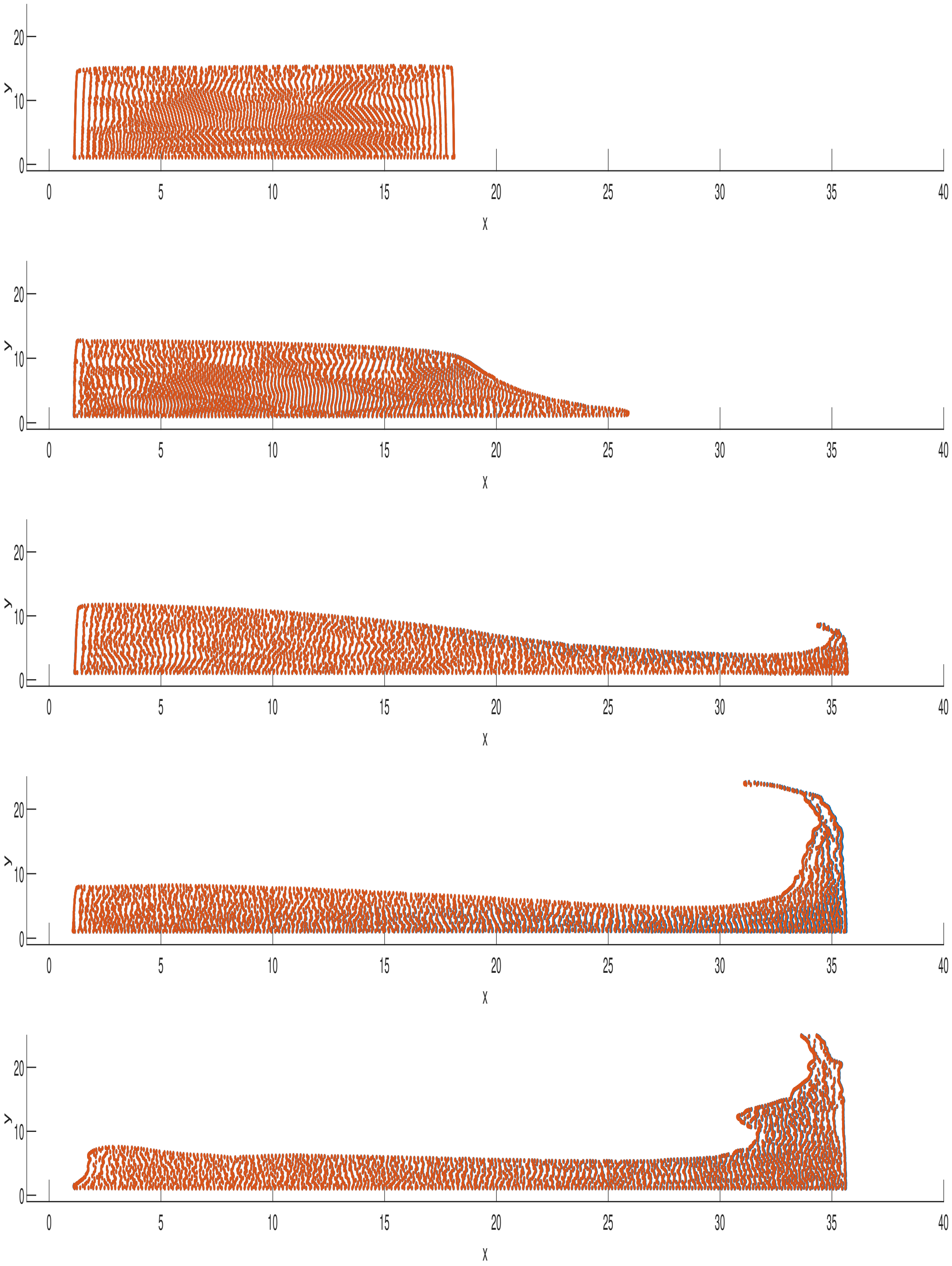


Figure 5.11: Comparison of original SPH results (blue) and the reconstruction flow field from DMD (red) with broken dam problem ($t = 0, 1, 2, 3, 4$).

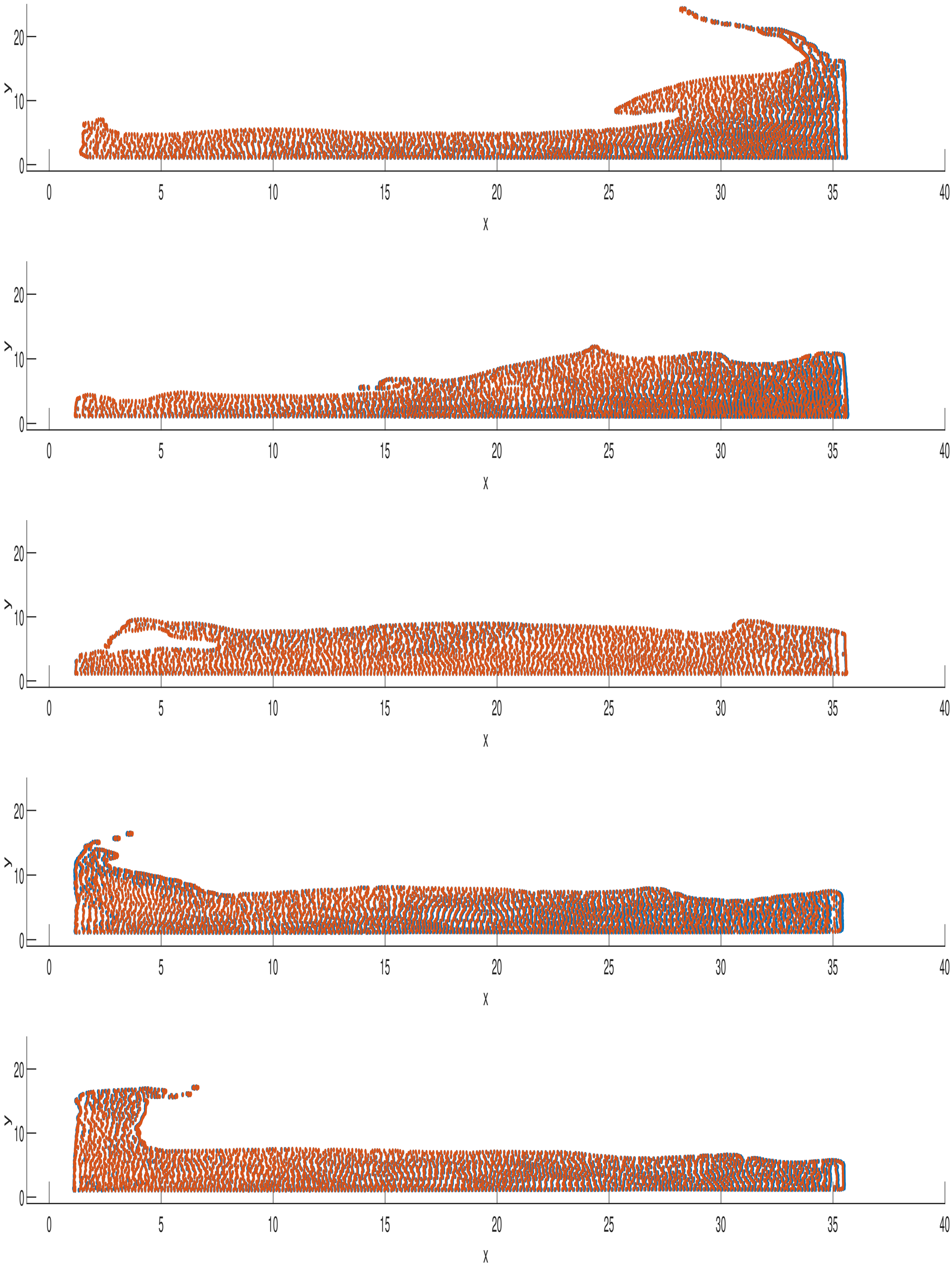


Figure 5.12: Comparison of original SPH results (blue) and the reconstruction flow field from DMD (red) with broken dam problem ($t = 5, 6, 7, 8, 9$).

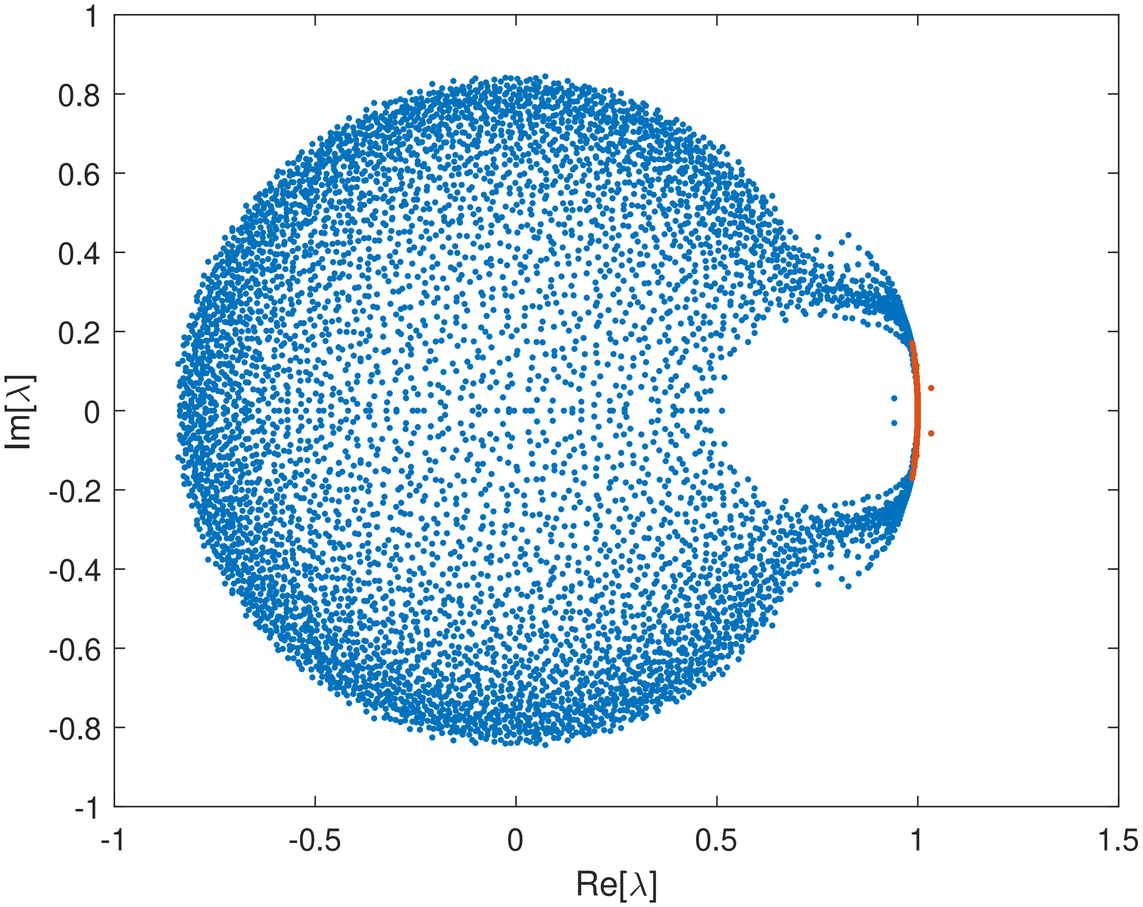


Figure 5.13: Eigenvalues of broken dam problem.

CHAPTER 6

Conclusions and Future Work

6.1 Summary

In this work, we have developed a methodology for the numerical simulation of the full dynamic response of blood-perfused soft tissues and the surface tension involved wicking problem. The methodology relies on the method of smoothed particle hydrodynamics (SPH), utilizing separate sets of computational particles to discretize the governing equations developed from the theory of porous media. Solid behavior in this work was treated as hyperelastic and the liquid as a Newtonian fluid. Good agreement was found between the method and available analytical solution in several test problems. The method was applied to several problems that demonstrated its ability to capture phenomena that are entirely dependent on the interaction between the solid and liquid phases—draining (oozing), hysteresis, swelling, drying and shrinkage, and tissue fracturing, hemorrhage and wicking.

It is important to note that the methodology we have presented here is agnostic to the specific constitutive models of the materials, and other material behaviors can readily be substituted for the solid and liquid phases. A third (gas) phase can also be included in problems in which macroscopic gas transport must be explicitly accounted for.

In addition, we first explored how to apply DMD analysis on the SPH method, raising a strategy to formulate the input state vectors only consist of position vectors, and then reconstruct the original flow from superposition of the DMD modes with modification of eigenvalues of corresponding modes. Eigenvalues and their growth rate, amplitude, frequency of two exapmle flow simulations are shown in the paper, while the comparison between eigenvalues pattern and the unit circle in complex plane provides implication of the stability status of the flow. The possibility of reduce-order model was also discussed although the flow field reconstruction of reduce-order DMD modes is not as good as that of non-truncated one. The physical interpretation of each mode and its corresponding amplitude are connected with the concept of displacement of current position with respect to average position. The extracting DMD modes show the possible application in the sense of data compressing, feature extraction. The time step size of each snapshots of the input data matrix is much larger than the time step size of SPH method, implying the possibility of future application that predicting system state without the concerns of limitation time step size indicating by Courant number and Fourier number.

With reference to the research objectives outlined in Sec. 1.2, the work presented here has demonstrated significant progress. Specifically, we have achieved the following goals:

1. Implemented and validated numerical solvers on both solid mechanics and fluid mechanics with Smoothed Particle Hydrodynamics.

2. Proposed first model on SPH to incorporate into liquid-solid particle interaction such as wicking, surface tension, swelling, hysteresis.

3. Simulated multiphase phenomena on high visual and physical fidelity of bleeding and hemorrhage, within the context of human injuries, medical treatments, and surgical intervention.

4. Coupled the cardiovascular system as the inflow condition of the blood vessel in the liver simulation.

5. Devised unsupervised learning approaches (dynamic mode decomposition) to extract spatiotemporal behavior of particle system.

6. Reduced dimensionality of flow field and reconstructed the fluid system for data compression purpose.

6.2 Future Work

Besides the current achievements in the work, it could be improved or extended by:

1. Acceleration of the implementation: High performance computing is still highly developing. Beside the techniques we utilized in this work such as OpenMP, GPU computing, there are several tools emerged recently and may be applied in the future such as TPU computing, quantum computing. Both of them are claimed to have much more powerful computational tool than GPU.

2. Validation with the experiments: In the work, we only validate our solver with analytical solution of some benchmark problems. Comparison with the real experiment may help us to verify and improve our modeling.

3. Data science on fluid mechanics: To model a highly non-linear system by just extracting the mechanism from pure data is attractable to fluid community. It may provides the answer for computationally-efficient solver instead of solving Navier Stokes' equation anymore.

REFERENCES

[AGL05] James Ahrens, Berk Geveci, and Charles Law. Paraview: An end-user tool for large data visualization. *The visualization handbook*, 717, 2005.

[Aya15] Utkarsh Ayachit. *The paraview guide: a parallel visualization application.* Kitware, Inc., 2015.

[BBP$^+$17] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):1–9, 2017.

[BFH18] Reza Behrou, Hamid Foroughi, and Fardad Haghpanah. Numerical study of temperature effects on the poro-viscoelastic behavior of articular cartilage. *J. Mech. Behav. Biomed. Mater.*, 78:214–223, 2018.

[Bio41] Maurice A. Biot. General theory of three-dimensional consolidation. *J. Appl. Phys.*, 12(2):155–164, 1941.

[BJOK16] Bingni W Brunton, Lise A Johnson, Jeffrey G Ojemann, and J Nathan Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of neuroscience methods*, 258:1–15, 2016.

[Bow80] Ray M. Bowen. Incompressible porous media models by use of the theory of mixtures. *Int. J. Eng. Sci.*, 18(9):1129–1148, 1980.

[BPHK13] Thomas Breinlinger, Pit Polfer, Adham Hashibon, and Torsten Kraft. Surface tension and wetting effects with smoothed particle hydrodynamics. *Journal of Computational Physics*, 243:14–27, 2013.

[BSC15] Jens Behley, Volker Steinhage, and Armin B Cremers. Efficient radius neighbor search in three-dimensional point clouds. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3625–3630. IEEE, 2015.

[BSV$^+$15] Erik Berger, Mark Sastuba, David Vogt, Bernhard Jung, and Heni Ben Amor. Estimation of perturbations in robotic behavior using dynamic mode decomposition. *Advanced Robotics*, 29(5):331–343, 2015.

[Can19] Daniel Joseph Canuto. *Multiscale and Patient-Specific Cardiovascular Modeling.* PhD thesis, UCLA, 2019.

[CCB$^+$18] Daniel Canuto, Kwitae Chong, Cayley Bowles, Erik P. Dutson, Jeff D. Eldredge, and Peyman Benharash. A regulated multiscale closed-loop cardiovascular model, with applications to hemorrhage and hypertension. *Int. J. Numer. Meth. Biomed. Engng.*, page e2975, 2018.

[CD07] Stephen C. Cowin and Stephen B. Doty. *Tissue Mechanics.* Springer Science & Business Media, 2007.

[CDB$^+$11] Alejandro C. Crespo, Jose M. Dominguez, Anxo Barreiro, Moncho Gómez-Gesteira, and Benedict D. Rogers. GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods. *PLoS One*, 6(6):e20685, 2011.

[CDK$^+$01] Rohit Chandra, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, and Jeff McDonald. *Parallel programming in OpenMP.* Morgan kaufmann, 2001.

[CDR$^+$15] Alejandro J. C. Crespo, José M. Domínguez, Benedict D. Rogers, Moncho Gómez-Gesteira, S. Longshaw, R. Canelas, Renato Vacondio, A. Barreiro, and O. García-Feal. DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH). *Comput. Phys. Commun.*, 187:204–216, 2015.

[CGSMVC10] Dominique Chapelle, J.-F. Gerbeau, J. Sainte-Marie, and I. E. Vignon-Clementel. A poroelastic model valid in large strains with applications to perfusion in cardiac modeling. *Comput. Mech.*, 46(1):91–101, 2010.

[CJR$^+$17] Kwitae Chong, Chenfanfu Jiang, Daniel Ram, Anand Santhanam, Demetri Terzopoulos, Peyman Benharash, Erik Dutson, Joseph Teran, and Jeff D. Eldredge. Visualization of vascular injuries in extremity trauma. *Med. Biol. Eng. Comput.*, pages 1–10, 2017.

[CJVDP08] Barbara Chapman, Gabriele Jost, and Ruud Van Der Pas. *Using OpenMP: portable shared memory parallel programming*, volume 10. MIT press, 2008.

[CKC$^+$04] C. Chui, E. Kobayashi, X. Chen, T. Hisada, and I. Sakuma. Combined compression and elongation experiments and non-linear modelling of liver tissue for surgical simulation. *Med. Biol. Eng. Comput.*, 42(6):787–798, 2004.

[CKC$^+$07] C. Chui, Etsuko Kobayashi, Xian Chen, Toshiaki Hisada, and Ichiro Sakuma. Transversely isotropic properties of porcine liver tissue: experiments and constitutive modelling. *Med. Biol. Eng. Comput.*, 45(1):99–106, 2007.

[CTR12] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.

[DB06] Reint De Boer. *Trends in Continuum Mechanics of Porous Media*, volume 18. Springer Science & Business Media, 2006.

[dBD04] Reint de Boer and Anjani Kumar Didwania. Two-phase flow and the capillarity phenomenon in porous solids–a continuum thermomechanical approach. *Transp. Porous Media*, 56(2):137–170, 2004.

[DBE86] R. De Boer and W. Ehlers. On the problem of fluid-and gas-filled elastoplastic solids. *Int. J. Solids Struct.*, 22(11):1231–1242, 1986.

[DBE90] R. De Boer and W. Ehlers. Uplift, friction and capillarity: three fundamental effects for liquid-saturated porous solids. *Int. J. Solids Struct.*, 26(1):43–57, 1990.

[DBK83] Reint De Boer and Stefan Jan Kowalski. A plasticity theory for fluid-saturated porous solids. *Int. J. Eng. Sci.*, 21(11):1343–1357, 1983.

[DC93] E. Detournay and Alexander H.-D. Cheng. Fundamentals of poroelasticity. In *Analysis and Design Methods: Comprehensive Rock Engineering: Principles, Practice and Projects*, volume 2, page 113. Elsevier, 1993.

[DCVB$^+$13] José M. Domínguez, Alejandro J. C. Crespo, Daniel Valdez-Balderas, Benedict D. Rogers, and Moncho Gómez-Gesteira. New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters. *Comput. Phys. Commun.*, 184(8):1848–1860, 2013.

[DMACM17] A Di Mascio, M Antuono, A Colagrossi, and S Marrone. Smoothed particle hydrodynamics method from a large eddy simulation perspective. *Physics of Fluids*, 29(3):035102, 2017.

[ED16] N Benjamin Erichson and Carl Donovan. Randomized low-rank dynamic mode decomposition for motion detection. *Computer Vision and Image Understanding*, 146:40–50, 2016.

[EM01] Wolfgang Ehlers and Bernd Markert. A linear viscoelastic biphasic model for soft tissues based on the theory of porous media. *J. Biomech. Eng.*, 123(5):418–424, 2001.

[ET05] M. Ellero and R. I. Tanner. SPH simulations of transient viscoelastic flows at low Reynolds number. *J. Non-Newtonian Fluid Mech.*, 132(1):61–72, 2005.

[EW15] Wolfgang Ehlers and Arndt Wagner. Multi-component modelling of human brain tissue: a contribution to the constitutive and computational description of deformation, flow and diffusion processes with application to the invasive drug-delivery problem. *Comput. Methods Biomech. Biomed. Engin.*, 18(8):861–879, 2015.

[Gil58] T Gillespie. The spreading of low vapor pressure liquids in paper. *Journal of colloid science*, 13(1):32–50, 1958.

[GK14] Jacob Grosek and J Nathan Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.

[GLD10] Zhan Gao, Kevin Lister, and Jaydev P. Desai. Constitutive modeling of liver tissue: experiment and theory. *Ann. Biomed. Eng.*, 38(2):505–516, 2010.

[GM77] Robert A. Gingold and Joseph J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.*, 181(3):375–389, 1977.

[GMS01] J. P. Gray, J. J. Monaghan, and R. P. Swift. SPH elastic dynamics. *Comput. Methods Appl. Mech. Eng.*, 190(49):6641–6662, 2001.

[Hin13] Pieter Hintjens. *ZeroMQ: messaging for many applications.* " O'Reilly Media, Inc.", 2013.

[HKK07] Takahiro Harada, Seiichi Koshizuka, and Yoichiro Kawaguchi. Smoothed particle hydrodynamics on GPUs. In *Computer Graphics International*, volume 40, pages 63–70. SBC Petropolis, 2007.

[IABT11] Markus Ihmsen, Nadir Akinci, Markus Becker, and Matthias Teschner. A parallel sph implementation on multi-core cpus. In *Computer Graphics Forum*, volume 30, pages 99–112. Wiley Online Library, 2011.

[JSZH09] P. Jordan, S. Socrate, T. E. Zickler, and R. D. Howe. Constitutive modeling of porcine liver in indentation using 3D ultrasound imaging. *J. Mech. Behav. Biomed. Mater.*, 2(2):192–201, 2009.

[KBBP16] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems.* SIAM, 2016.

[Kel06] Micky Kelager. Lagrangian fluid dynamics using smoothed particle hydrodynamics. Master's thesis, University of Copenhagen, 2006.

[KGB16] Jake Nathan Kutz, J Grosek, and Steven L Brunton. Dynamic mode decomposition for robust pca with applications to foreground/background subtraction in video streams and multi-resolution analysis. *CRC Handbook on Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, 2016.

[Koo31] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5):315, 1931.

[KS07] Yoichi Kawashima and Yuzuru Sakai. Large Deformation Analysis of Hyperelastic Materials Using SPH Method. *e-Journal of Soft Materials*, 3:21–28, 2007.

[LBWW18] Sinuo Liu, Xiaojuan Ban, Ben Wang, and Xiaokun Wang. A symmetric particle-based simulation scheme towards large scale diffuse fluids. *Symmetry*, 10(4):86, 2018.

[Lew84] Menachem Lewin. *Handbook of Fiber Science and Technology Volume 2: Chemical Processing of Fibers and Fabrics–Functional Finishes*, volume 2. CRC Press, 1984.

[LGD11] Kevin Lister, Zhan Gao, and Jaydev P. Desai. Development of in vivo constitutive models for liver: Application to surgical simulation. *Ann. Biomed. Eng.*, 39(3):1060–1073, 2011.

[LL03] Gui-Rong Liu and Moubin B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method.* World Scientific, 2003.

[LL10] MB Liu and GR Liu. Smoothed particle hydrodynamics (sph): an overview and recent developments. *Archives of computational methods in engineering*, 17(1):25–76, 2010.

[LP91] Larry D. Libersky and A. G. Petschek. Smooth particle hydrodynamics with strength of materials. In *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*, pages 248–257. Springer, 1991.

[MCG03] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 154–159. Eurographics Association, 2003.

[MG83] Joseph J Monaghan and Robert A Gingold. Shock simulation by the particle method sph. *Journal of computational physics*, 52(2):374–389, 1983.

[MK16] Jordan Mann and J Nathan Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, 16(11):1643–1655, 2016.

[MKLA80] Van C. Mow, S. C. Kuei, W. Michael Lai, and Cecil G. Armstrong. Biphasic creep and stress relaxation of articular cartilage in compression: Theory and experiments. *J. Biomech. Eng.*, 102(1):73–84, 1980.

[Mon89] JJ Monaghan. On the problem of penetration in particle methods. *Journal of Computational physics*, 82(1):1–15, 1989.

[Mon00] Joseph J. Monaghan. SPH without a tensile instability. *J. Comput. Phys.*, 159(2):290–311, 2000.

[Mon11] Joe J Monaghan. A turbulence model for smoothed particle hydrodynamics. *European Journal of Mechanics-B/Fluids*, 30(4):360–370, 2011.

[Moo40] M. Mooney. A theory of large elastic deformation. *J. Appl. Phys.*, 11(9):582–592, 1940.

[Mor00] Joseph P Morris. Simulating surface tension with smoothed particle hydrodynamics. *International journal for numerical methods in fluids*, 33(3):333–353, 2000.

[MP85] JJ Monaghan and H Pongracic. Artificial viscosity for particle methods. *Applied Numerical Mathematics*, 1(3):187–194, 1985.

[ND10] John Nickolls and William J Dally. The gpu computing era. *IEEE micro*, 30(2):56–69, 2010.

[OHL+08] John D Owens, Mike Houston, David Luebke, Simon Green, John E Stone, and James C Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899, 2008.

[OVSM98] J. Michael Owen, Jens V. Villumsen, Paul R. Shapiro, and Hugo Martel. Adaptive smoothed particle hydrodynamics: Methodology. II. *The Astrophysical Journal Supplement Series*, 116(2):155, 1998.

[Ozd17] Hakan Ozdemir. Permeability and wicking properties of modal and lyocell woven fabrics used for clothing. *Journal of Engineered Fibers and Fabrics*, 12(1):155892501701200102, 2017.

[PE15] Joshua L Proctor and Philip A Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.

[PRKG06] Amalendu Patnaik, RS Rengasamy, VK Kothari, and A Ghosh. Wetting and wicking in fibrous materials. *Textile Progress*, 38(1):1–105, 2006.

[PVR+17] Marcelo Parada, Peter Vontobel, René M Rossi, Dominique Derome, and Jan Carmeliet. Dynamic wicking process in textiles. *Transport in Porous Media*, 119(3):611–632, 2017.

[RDB03] T. Ricken and R. De Boer. Multiphase flow in a capillary porous medium. *Computational Materials Science*, 28(3):704–713, 2003.

[RDD10] Tim Ricken, Uta Dahmen, and Olaf Dirsch. A biphasic model for sinusoidal liver perfusion remodeling after outflow obstruction. *Biomech. Model. Mechanobiol.*, 9(4):435–450, 2010.

[RE10] R. A. Regueiro and D. Ebrahimi. Implicit dynamic three-dimensional finite element analysis of an inelastic biphasic mixture at finite strain: Part 1: application to a simple geomaterial. *Comput. Methods Appl. Mech. Eng.*, 199(29-32):2024–2049, 2010.

[Riv48] R. S. Rivlin. Large elastic deformations of isotropic materials. IV. Further developments of the general theory. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 241(835):379–397, 1948.

[RMH07] A. Rafiee, M. T. Manzari, and M. Hosseini. An incompressible SPH method for simulation of unsteady viscoelastic free-surface flows. *Int. J. Non Linear Mech.*, 42(10):1210–1223, 2007.

[RSB07] Tim Ricken, Alexander Schwarz, and Joachim Bluhm. A triphasic model of transversely isotropic biological tissue with applications to stress and biologically induced growth. *Computational Materials Science*, 39(1):124–136, 2007.

[Sch10] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[SK14] Canan Saricam and F Kalaoğlu. Investigation of the wicking and drying behaviour of polyester woven fabrics. *Fibres & Textiles in Eastern Europe*, 2014.

[SL03] Songdong Shao and Edmond Y. M. Lo. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Adv. Water Resour.*, 26(7):787–800, 2003.

[SP09] Barbara Solenthaler and Renato Pajarola. Predictive-corrective incompressible SPH. *ACM Trans. Graphics*, 28(3):40, 2009.

[Spr10] Volker Springel. Smoothed particle hydrodynamics in astrophysics. *Annual Review of Astronomy and Astrophysics*, 48:391–430, 2010.

[Sta15] Susan Standring. *Gray's Anatomy E-Book: The Anatomical Basis of Clinical Practice.* Elsevier Health Sciences, 2015.

[SWB+06] Ivo F Sbalzarini, Jens H Walther, Michael Bergdorf, Simone Elke Hieber, Evangelos M Kotsalis, and Petros Koumoutsakos. Ppm–a highly efficient parallel particle–mesh library for the simulation of continuum systems. *Journal of Computational Physics*, 215(2):566–588, 2006.

[SWP+06] Ivo F Sbalzarini, Jens H Walther, B Polasek, Philippe Chatelain, Michael Bergdorf, Simone Elke Hieber, Evangelos M Kotsalis, and Petros Koumoutsakos. A software framework for the portable parallelization of particle-mesh simulations. In *European Conference on Parallel Processing*, pages 730–739. Springer, 2006.

[TGK+17] Andre Pradhana Tampubolon, Theodore Gast, Gergely Klár, Chuyuan Fu, Joseph Teran, Chenfanfu Jiang, and Ken Museth. Multi-species simulation of porous sand and water mixtures. *ACM Trans. Graphics*, 36(4):105, 2017.

[TM05] Alexandre Tartakovsky and Paul Meakin. Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E*, 72(2):026301, 2005.

[TOSF11] Hiroki Takamiya, Hiroshi Okada, Yuzuru Sakai, and Yasuyoshi Fukui. Smoothed particle hydrodynamics analysis on semi-solid metal forming process. *Jpn. J. Ind. Appl. Math.*, 28(1):183–203, 2011.

[TP16] Alexandre M Tartakovsky and Alexander Panchenko. Pairwise force smoothed particle hydrodynamics model for multiphase flow: surface tension and contact line dynamics. *Journal of Computational Physics*, 305:1119–1146, 2016.

[TRL+13] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: Theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.

[UCB+11] Sagar Umale, Simon Chatelin, Nicolas Bourdet, Caroline Deck, Michele Diana, Parag Dhumane, Luc Soler, Jacques Marescaux, and Remy Willinger. Experimental in vitro mechanical characterization of porcine Glisson's capsule and hepatic veins. *J. Biomech.*, 44(9):1678–1683, 2011.

[UDB+13] Sagar Umale, Caroline Deck, Nicolas Bourdet, Parag Dhumane, Luc Soler, Jacques Marescaux, and Remy Willinger. Experimental mechanical characterization of abdominal organs: liver, kidney & spleen. *J. Mech. Behav. Biomed. Mater.*, 17:22–33, 2013.

[VI07] Damien Violeau and Reza Issa. Numerical modelling of complex turbulent free-surface flows with the sph method: an overview. *International Journal for Numerical Methods in Fluids*, 53(2):277–304, 2007.

[VM08] Mehran Vakilha and Mehrdad T. Manzari. Modelling of power-law fluid flow through porous media using smoothed particle hydrodynamics. *Transp. Porous Media*, 74(3):331–346, 2008.

[Zha10] Mingyu Zhang. Simulation of surface tension in 2d and 3d with smoothed particle hydrodynamics method. *Journal of Computational Physics*, 229(19):7238–7259, 2010.

[ZHL15] Qirong Zhu, Lars Hernquist, and Yuexing Li. Numerical convergence in smoothed particle hydrodynamics. *The Astrophysical Journal*, 800(1):6, 2015.