

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING IN THERMAL
AND FLUID DYNAMICS

**LES stochastic modelling of cavitation
with its applications in OpenFOAM**

BOXIONG CHEN

Division of Combustion
Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

LES stochastic modelling of cavitation with its applications in OpenFOAM

Boxiong Chen

Copyright © Boxiong Chen, 2016.

Thesis for Licentiate of Engineering no. 2016-18

ISSN 1652-8565

Department of Applied Mechanics

Division of Combustion

Chalmers University of Technology

SE-412 96 GÖTEBORG, Sweden

Phone: +46 (0)31-772 10 00

Author e-mail: `chenb@chalmers.se`

Chalmers Reproservice

Göteborg, Sweden 2016

LES stochastic modelling of cavitation with its applications in OpenFOAM

Boxiong Chen

Division of Combustion, Chalmers University of Technology

ABSTRACT

Cavitation is a vaporisation process that commonly happens in high pressure injector nozzles nowadays. It has been shown by previous studies that cavitation has a significant influence on the subsequent atomization process, the quality of which would in turn heavily affect the quality of combustion. With the coming of the ever increasingly restrictive emission standard, studies into cavitation phenomena has attracted rapidly increasing amount of interest from both the academic and the industrial circle. However, due to the inheritant difficulties, cavitation still render itself a process that is hard to be quantified with the experimental facilities nowadays. While many of the commercial softwares, e.g. Ansys, STAR-CCM+, have integrated cavitation modules into their packages, the variety of models is still in a rapidly expanding phase.

The open source CFD package OpenFOAM (Open Field Operation and Manipulation) has gone through a long developing stage and has been proved to be a highly matured and convenient CFD tool for many engineering applications. Although a great number of fundamental applications have been developed, the development of solvers and libraries which caters the need of specific physical problems remain to be an ongoing task. In this work, focus has been put specifically on the development of cavitation modeling. Several readily available cavitation models have been used, based on which, a more complicated stochastic ordinary differential equation (ODE) solver has been developed to facilitate a Monte-Carlo type treatment to the non-linear terms involved in the cavitation model.

Keywords: Cavitation, Volume of Fluid Method, Rayleigh-Plesset Model, Modeling, Simulation, OpenFOAM

Preface

Parts, but far from all, of the contributions presented in this thesis have previously been accepted to conferences or submitted to journals.

- ▷ **Boxiong Chen**, Zachary Falgout, Mark Linne, Michael Oevermann, “LES modelling of cavitation flow in a diesel injector nozzle,” in *International Conference on Multiphase Flow*, Firenze, Italy, May 22nd to 27th, 2016.

Acknowledgments

First off, I am grateful to my advisor, Professor Michael Oevermann, for his genuine interest into science, and all the insightful discussions we had, which has finally built up to this thesis. Without our common persistence to physical sense, and the affluent time he allowed for the fundamentals, the work could have withered at many times when major problems present themselves.

A thanks goes to Professor Ingemar Denbratt, and assistant Ellenor Norberg, and all other fellow colleagues for making the combustion division a cozy and pleasant place to work in. Life could have been so much tougher, without proper assistance and guidance from the peers.

Special thanks due to Professor Mark Linne for his excellent skills in presenting ideas and communication. Without those, the financial support of this work would have to be from other sources. In the same line, the Knut & Alice Wallenberg Foundation is acknowledged for the funding.

I would like to also thank many colleagues in Fluid Dynamics division, especially Håkan Nilsson, Xin Zhao, Ardalan Javadi, among others, for their invaluable experiences on OpenFOAM, and ICEM-CFD. A significantly more amount of work and time would be required without their assistance on details and open-mindedness on sharing their expertise.

Boxiong Chen
Göteborg, February 2016

Contents

Abstract	i
Preface	iii
Acknowledgments	v
1 Introduction	1
1.1 Introduction	1
Bibliography	4
2 Cavitation and multiphase modelling	7
2.1 Introduction	7
2.2 Cavitation modelling: an overview	7
2.3 Governing equations and models	10
Bibliography	12
3 Stochastic numerical integration	15
3.1 Introduction	15
3.2 Monte Carlo method	15
3.3 PDF and Ito stochastic differential equations	17
3.4 Stochastic integration schemes	21
3.4.1 Euler-Maruyama method	22
3.4.2 Some Theoretical Tools and Milstein method	23
3.4.3 2nd order stochastic Runge-Kutta method	26

3.5	Numerical tests	26
3.5.1	Test One	27
3.5.2	Test Two	27
3.6	An Eulerian Field Monte Carlo Formulation of Volume Fraction	29
	Bibliography	30
4	Test case	35
4.1	Introduction	35
4.2	Geometry and case setup	36
4.3	Results	37
	Bibliography	37
	Appendices	47
A	OpenFOAM implementation	49
A.1	Introduction	49
A.2	OpenFOAM and its standard solvers	50
A.3	Implementation of SRK2	52
A.4	Implementation of stochastic model	53
A.4.1	stochasticPhaseChangeTwoPhaseMixture	53
A.4.2	stochasticModel and stochasticSolver	58
	Bibliography	66

List of Figures

1.1	Global crude oil consumption in 2012	3
3.1	Test 1 for SRK and EM method, dt=0.01, dx=0.1, realizations=40	28
3.2	Test 2 for SRK and EM method, realizations=1000	32
3.3	Test 2 for SRK and EM method, realizations=10000	32
3.4	Test 2 for SRK and EM method, time step comparison	33
4.1	2D nozzle geometry	36
4.2	Velocity field, interPhaseChangeFoam	39
4.3	Velocity field, interPhaseChangeFDFFoam	40
4.4	Volume fraction field, interPhaseChangeFoam	41
4.5	Mean volume fraction field, interPhaseChangeFDFFoam	42
4.6	Volume fraction field number 1, interPhaseChangeFDFFoam	43
4.7	Volume fraction field number 2, interPhaseChangeFDFFoam	44
4.8	Volume fraction field number 3, interPhaseChangeFDFFoam	45
4.9	Volume fraction field number 4, interPhaseChangeFDFFoam	46

1

Introduction

1.1 Introduction

Energy has been one of the major concern of human civilization since the dawn of industrial revolution. As we enjoy the fruit of modernization, problems caused by the consumption of fossil fuels have become more pronounced than ever. In general, the development of countries with a less developed industrial basis depends heavily on oil. Take crude oil consumption of China as an example, in 1980 the consumption was around 2 million barrels per day, it increases five times to around 10 million barrels per day until 2013 [1]; A almost 6 times increase from around 640 to 3600 has also occurred during the same period of time in India [2]. The figures in EU countries suggest a lesser dependence [3] [4], however, this is largely due to the more developed industrial

basis, and the ongoing effort to discover alternative resources and reduce the oil consumption by political intervention.

Fossil fuels are in general not renewable, and considering the restriction on the available amount of oil, there could be a time of depletion. New efforts to exploit non-fossil energy resources, such as solar, wind, and nuclear power, as well as the novel technologies to obtain fuels, such as shale oil extraction, and the application of bio-fuel, have given us new hopes to the solution of potential shortage of fossil fuel. However, at present the combustion of fossil fuels still accounts for 80 percent of the energy consumption of today's world. Before the coming of any matured, and economically feasible solutions out of any of the above mentioned technologies, the only sane response to avoid such shortage is to use fossil fuel in a controlled and more efficient way. From the environmental prospective, combustion gives off CO_2 as one of the major chemical product, which is vastly believed to be a major cause of the global warming problem. In the less ideal situation when the combustion is insufficient, the process yields more harmful products such as CO and NO_x , unburned hydrocarbons and soot, which could give rise to severe health problems. Thus, it is necessary to reduce the consumption of fossil fuel and to improve combustion efficiency.

Among all the sources of fuel consumption, consumption by transportation has been a major portion of the figure. According to [5], 63.7 percent of the world oil consumption goes to the transportation sector in 2012, see Figure 1.1. Therefore, it is critical to reduce the oil consumption of motor vehicles. From the political side, emission standards have become ever-increasingly strict. In the automotive engineering realm, researchers from both the industrial side and the academic side have been struggling hard to study the combustion inside the internal combustion engine from various aspects so as to meet the emission requirements. Among multiple factors, the quality of the fuel spray greatly affects the quality of combustion. Previous studies [6] have demonstrated that the cavitation that happens inside the needle following the passage contraction of flow enhances spray atomization, which in turn affects strongly the quality of subsequent combustion. And it is speculated that the enhancement is achieved by either inducing a direct disintegration locally, or through enhanc-

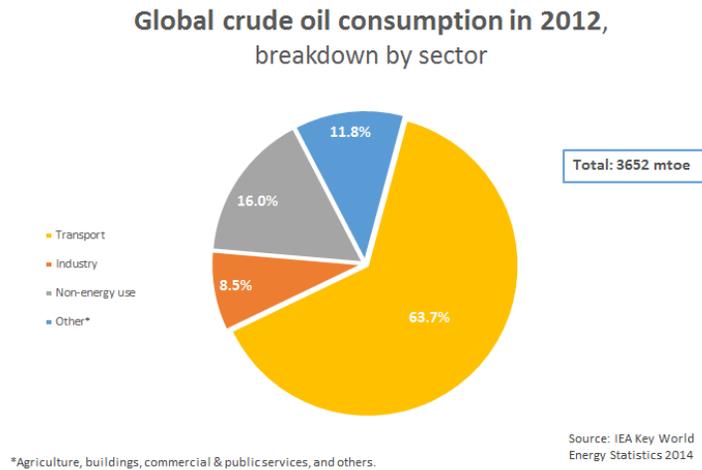


Figure 1.1: *Global crude oil consumption in 2012*

ing turbulence, which in turn, induces the breakup. Up to now, neither of the two mechanisms have got convincing validations from experiments or direct numerical simulations. More insight into the phenomena is required for both fundamental understanding of the process and to provide predictive simulations on relevant applications. However, it is commonly admitted that experimental study of cavitation on realistic nozzle geometries and injection conditions is difficult. As pointed out in [7], direct observation of cavities in realistic injectors requires high resolutions in both space and time, due to the small sizes and rapid evolvments of bubbles. Therefore previous experimental studies have mostly been performed on up-scaled nozzles, lower injection pressure and velocities with simple geometries. Results and conclusions obtained in such simplified designs could not be directly transferred to realistic injectors, they are rather used for validation and references for cavitation models.

Previous computational studies on cavitation with various levels of complexities have been carried out. Direct numerical resolving of the interface between phases could be computationally very expensive, and is therefore only feasi-

ble for fundamental studies. Various interface modelling methodologies have been testified in previous studies. Although some of them have achieved good agreement with experiments, they would be less applicable to simulations of realistic nozzle geometries. Since we are more concerned about the applicability to real size, high pressure nozzle geometries, a homogeneous mixture model is adopted in the current work, which does not attempt to resolve or model the interface, but observe on a more macroscopic level and regard the bubbly flow as a mixture of fluid and gas. The phases are represented by their volume fraction. Previous studies, e.g. [8] have developed mass exchange models based on the Rayleigh-Plesset equation. In this work, a stochastic Monte-Carlo type of treatment has been applied to solve the volume fraction equation with higher complexity.

Bibliography

- [1] "China crude oil consumption by year," <http://www.indexmundi.com/energy/?country=cn&product=oil&graph=consumption>, Accessed: 2016-08-22.
- [2] "India crude oil consumption by year," <http://www.indexmundi.com/energy/?country=in&product=oil&graph=consumption>, Accessed: 2016-08-22.
- [3] "Sweden crude oil consumption by year," <http://www.indexmundi.com/energy/?country=se&product=oil&graph=consumption>, Accessed: 2016-08-22.
- [4] "Germnay crude oil consumption by year," <http://www.indexmundi.com/energy/?country=de&product=oil&graph=consumption>, Accessed: 2016-08-22.
- [5] "Breakdown of oil consumption by sector," <http://www.globalpetrolprices.com/articles/39/>, Published: 2015-10-07, Accessed: 2016-08-22.

- [6] Akira Sou, Shigeo Hosokawa, and Akio Tomiyama, “Effects of cavitation in a nozzle on liquid jet atomization,” *International journal of heat and mass transfer*, vol. 50, no. 17, pp. 3575–3582, 2007.
- [7] Cyril Mauger, Loïc Méès, Marc Michard, Alexandre Azouzi, and Stéphane Valette, “Shadowgraph, schlieren and interferometry in a 2D cavitating channel flow,” *Experiments in fluids*, vol. 53, no. 6, pp. 1895–1913, 2012.
- [8] Lord Rayleigh, “On the pressure developed in a liquid during the collapse of a spherical cavity, VIII.,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 34, no. 200, pp. 94–98, 1917.

2

Cavitation and multiphase modelling

2.1 Introduction

In this chapter, the physics and modelling aspects of cavitation are discussed. An overview of some previous cavitation modelling studies will be given. What then follows is a description of the cavitation model in the current work.

2.2 Cavitation modelling: an overview

Cavitation is a vaporisation of a liquid following a drastic pressure drop below the vapour pressure. The transition from liquid to vapour can be achieved by either heating the liquid at a constant pressure, which is known as boiling, or

through decreasing the pressure below the vapour pressure while maintaining constant temperature, which is known as cavitation.

The cavitation processes could present itself in many engineering applications. For example, in hydraulic applications like pumps and marine propellers, the rapid growth and collapse of cavity bubbles could chip off the turbomachinery over time; in water treatment, the same phenomena incurred by bubbles could be harnessed to break the pollutant particles. Specifically in spray nozzles of internal combustion engines, previous studies have found that cavitation phenomena helps in breaking up liquid fuel spray which is important to proper fuel-air mixing and reduce the formation of pollutant. Therefore, cavitation attracts substantial interests from both the automobile industry and the research side.

The influence of cavitation in injector nozzles on spray formation and breakup has been investigated both experimentally and computationally by previous works. Sou in [1] have demonstrated that the cavitation inside the fluid passage strongly affects the quality of the subsequent combustion. Whereas it remains and open question whether it enhances spray disintegration by contributing to the turbulent kinetic energy or by inducing a direct breakup mechanism locally. Experimental studies of cavitation on realistic nozzle geometries and injection conditions are difficult to perform. The geometric size and the residence time of the bubbles requires restrictively high resolutions of both time and space of optical access, for this reason, many experimental studies thus far have been based on up-scaled nozzles and low pressure conditions.

Computational studies with various modelling methodologies have been carried out. On the basis of Lagrangian approach, a Lagrangian description of individual bubbles or bubble clusters can be used to represent cavities, see, for instance, [2] and [3]. The interaction between bubbles, and bubble-wall interactions are modelled. On the Eulerian modelling side, both the liquid phase and the vapour phase are described by Eulerian fields. Since directly resolving interface normally takes formidably high computational cost, the treatment of interface is critical for Eulerian methods. [4] and [5] applied a level set method to represent the phase boundary between liquid and vapour phase. The onset

of cavitation is identified when the maximum tensile stress exceeds a critical value. Based on this assumption a model of the sub-grid shape of the interface is applied. The curvature of the interface is then used to calculate surface tension forces in the momentum equation. Marcer in [6] applied a volume of fluids (VOF) method, reconstructed the interface with planes of arbitrary orientations in each cell. In the above mentioned studies, the sub-grid shape of interface is considered either by geometrical reconstruction or direct modelling. Methods of this kind fit well into fundamental studies of the behaviour of few bubbles. A more macroscopic class of models, called “interface diffuse models” [7], regard bubbly flow as a continuous dispersed phase. The approach spends no effort tracking the fronts of each individual bubbles, but regard the bubbly flow as a mixture instead. The phases are represented by volume fractions, and the interface is treated as a zone where two phases coexists. Since physically bubbles exist on the sub-grid level, the volume fraction field does not define a sharp interface between the phases but provides the volumetric fraction of liquid and vapour within a computational cell. In the category of interface diffuse models, the number of equations that are used to describe the two phase further subcategorise different models. [8] proposed a seven equation model that consists of a conservative set of equations for mass, momentum, and energy for each of the two phases and an additional transport of equation for the volume fraction. Six-equation model by [9], [10], and [11] consist of the conservation equations for both phases, but only one pressure is kept assuming either incompressibility of one of the phases, or pressure equilibrium between the two phases. This approach reduced the computational cost involved in solving a volume fraction transport equation at the expense of a reduced validity on problems where transient wave propagation is important. Based on the seven-equation models, a five-equation model was derived in [12], assuming a mechanical equilibrium between the two phases. It involves transport of volume fraction, one set of momentum and energy equation, but two phase balance equations. [7] applied a discrete equation method (DEM) and a splitting method on the five-equation model to preserve the positivity of the solution and reduce the computational cost at the same time. The even simpler models make equilibrium assumption

for both pressure and temperature in the different phases, thereby regarding the two phases practically as one. [13] and [14] applied the volume of fluids method, combined with a $k - \omega$ turbulence model to describe the flow field. The Rayleigh-Plesset relation was used to provide a source term for the volume fraction equation.

In this work, we focus on the Eulerian Volume of Fluids method (VOF) of multiphase modelling. The expansion and shrinkage of vapour bubbles is indicated by the volume fraction of the vapour phase. The vapour-liquid mixture is assumed to be a homogenous mixture in each cell. Therefore, no effort is needed to reconstruct the interface and to model surface tension. This simplification is justified by our focus on applications that are close to realistic industrial designs, in which a large number of bubbles is involved and tracking every bubbles is computationally excessive or impossible. The Rayleigh-Plesset model is applied as a prediction for the mass transfer between the two phases. A Large Eddy Simulation approach is adopted to model sub-grid turbulent fluid motion. In the next section, a mathematical description of the governing equations and the cavitation model will be used.

2.3 Governing equations and models

We start with the definition of volume fraction. As proposed in [15], the vapor is assumed to consist of spherical bubbles. Therefore the liquid volume fraction can be written as

$$\alpha_l = \frac{1}{1 + n * 4\pi R^3 / 3}, \quad (2.1)$$

where R denotes bubble radius and n is the nuclei number density, which must be prescribed. The closure for α_l can be achieved by taking the material derivative:

$$\frac{d\alpha_l}{dt} = -3\alpha_l(1 - \alpha_l) \frac{\dot{R}}{R} \quad (2.2)$$

or

$$\dot{\alpha}_l + \vec{u} \cdot \nabla \alpha_l = -3\alpha_l(1 - \alpha_l) \frac{\dot{R}}{R} \quad (2.3)$$

A model for the bubble growth rate R is needed to close the equation above, which will be discussed shortly.

The global continuity and the momentum equation for the density ρ and the momentum $\rho\vec{u}$ are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\vec{u}) = 0 \quad (2.4)$$

$$\frac{\partial}{\partial t}(\rho\vec{u}) + \nabla \cdot (\rho\vec{u}\vec{u}) = \nabla \cdot (\mu\nabla\vec{u}) - \nabla p \quad (2.5)$$

The mixture averaged density is given by

$$\rho = (1 - \alpha_l)\rho_v + \alpha_l\rho_l, \quad (2.6)$$

where ρ_l and ρ_v denote the density of the liquid phase and the vapor phase respectively. Mass balance equations for the two phases are

$$\frac{\partial}{\partial t}(\alpha_v\rho_v) + \nabla \cdot (\alpha_v\rho_v\vec{u}) = \dot{m}_v, \quad (2.7)$$

$$\frac{\partial}{\partial t}(\alpha_l\rho_l) + \nabla \cdot (\alpha_l\rho_l\vec{u}) = -\dot{m}_v. \quad (2.8)$$

In the present study we use a pressure based scheme of the PISO type [20] for the numerical solution of the governing equations. A one-equation LES model, as is used in [16], is applied to model the turbulent behavior of the flow.

There have been numerous investigations since the beginning of the last century to achieve the closure of \dot{R} . A simple yet effective description by Rayleigh, see [15], is widely applied in cavitation modelling. Neglecting thermal effects and surface tension forces (justified by the assumption of homogeneity of the mixture) we have

$$\dot{R} = \sqrt{\frac{2}{3} \frac{p(R) - p}{\rho_l}}, \quad (2.9)$$

where $p(R)$ is the pressure in the liquid at the bubble boundary, which in the current model is assumed to be equivalent to the saturation pressure, and p is the pressure at a large distance from the bubble which is in practice the local pressure of the flow solver.

With the above closure for \dot{R} , the focus is then placed on the solving of Equation (2.3). The solving framework that has been presented up to this point constitutes the methodology of several previous works, e.g., [14] [16], and cutting-edge solvers, e.g., `interPhaseChangeFoam` in `OpenFOAM`, which will be discussed again in the later chapters. However, it is evident that, following the definitions, the single volume fraction field enjoys the correspondence with only one bubble radius (for each computational cell). It is, therefore, natural to think of a representation, through which, a distribution of radii can be rendered. In previous works, various methodologies have been developed on account of this extension. Pope in [17] solved the PDF transport equation by tracking Lagrangian particles in combination with Monte-Carlo method in a combustion problem. The spatial position of each particle is included as one of the stochastic variables and evolves according to PDF equation. This approach, therefore, entails no Eulerian grids. In [18], a more Eulerian-based approach was developed. The particles reside on a Eulerian field, and moves from one node to another following rules that are based on transportation. In the current work, we adopt the full Eulerian framework as in [19] and [20]. With certain stochastic procedures, which will be discussed in the next chapter, an ensemble of Eulerian fields is generated to represent a distribution of radii.

In the next chapter, our discussion will be focused on the stochastic treatment of a typical scalar transport problem that bears a close mathematical resemblance to Equation (2.3). The stochastic fields cavitation model of the current work can be derived based on the same procedures. However, since the final mathematical formulation would make little sense without a detailed discussion on stochastic method, the final formulation will not be shown until the end of the next chapter. Interested readers who have backgrounds pertinent to PDF method are referred to Section 3.6.

Bibliography

- [1] Akira Sou, Shigeo Hosokawa, and Akio Tomiyama, “Effects of cavitation in a nozzle on liquid jet atomization,” *International journal of heat and mass transfer*,

- vol. 50, no. 17, pp. 3575–3582, 2007.
- [2] E Giannadakis, M Gavaises, and C Arcoumanis, “Modelling of cavitation in diesel injector nozzles,” *Journal of Fluid Mechanics*, vol. 616, pp. 153–193, 2008.
- [3] Emmanouil Giannadakis, *Modelling of cavitation in automotive fuel injector nozzles*, Ph.D. thesis, Imperial College London (University of London), 2005.
- [4] Sadegh Dabiri, WA Sirignano, and DD Joseph, “Cavitation in an orifice flow,” *Physics of Fluids (1994-present)*, vol. 19, no. 7, pp. 072112, 2007.
- [5] S Dabiri, WA Sirignano, and DD Joseph, “A numerical study on the effects of cavitation on orifice flow,” *Physics of Fluids (1994-present)*, vol. 22, no. 4, pp. 042102, 2010.
- [6] R Marcer and JM LeGouez, “Simulation of unsteady cavitating flows in diesel injector with an improved VOF-method,” in *17th ILASS-Europe Conf*, 2001.
- [7] Maria Giovanna Rodio and Rémi Abgrall, “An innovative phase transition modeling for reproducing cavitation through a five-equation model and theoretical generalization to six and seven-equation models,” *International Journal of Heat and Mass Transfer*, vol. 89, pp. 1386–1401, 2015.
- [8] MR Baer and JW Nunziato, “A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials,” *International journal of multiphase flow*, vol. 12, no. 6, pp. 861–889, 1986.
- [9] Mamoru Ishii, “Thermo-fluid dynamic theory of two-phase flow,” *NASA STI/Recon Technical Report A*, vol. 75, pp. 29657, 1975.
- [10] JA Boure and JM Delhay, “General equations and two-phase flow modeling,” 1982.
- [11] Patrick Barry Butler, MF Lembeck, and H Krier, “Modeling of shock development and transition to detonation initiated by burning in porous propellant beds,” *Combustion and Flame*, vol. 46, pp. 75–93, 1982.
- [12] Richard Saurel, Fabien Petitpas, and Rémi Abgrall, “Modelling phase transition in metastable liquids: application to cavitating and flashing flows,” *Journal of Fluid Mechanics*, vol. 607, pp. 313–350, 2008.
- [13] Weixing Yuan and Günter H Schnerr, “Numerical simulation of two-phase flow in injection nozzles: Interaction of cavitation and external jet formation,” *Journal of Fluids Engineering*, vol. 125, no. 6, pp. 963–969, 2003.

- [14] Weixing Yuan, Jürgen Sauer, and Günter H Schnerr, “Modeling and computation of unsteady cavitation flows in injection nozzles,” *Mécanique & industries*, vol. 2, no. 5, pp. 383–394, 2001.
- [15] Lord Rayleigh, “On the pressure developed in a liquid during the collapse of a spherical cavity, VIII.,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 34, no. 200, pp. 94–98, 1917.
- [16] Boxiong Chen, Zachary Falgout, Mark Linne, and Michael Oevermann, “LES modelling of cavitation flow in a diesel injector nozzle,” *International Conference on Multiphase Flow*, 2016.
- [17] Stephen B Pope, “PDF methods for turbulent reactive flows,” *Progress in Energy and Combustion Science*, vol. 11, no. 2, pp. 119–192, 1985.
- [18] Stephen Bailey Pope, “A monte carlo method for the PDF equations of turbulent reactive flow,” 1981.
- [19] Luis Valiño, “A field monte carlo formulation for calculating the probability density function of a single scalar in a turbulent flow,” *Flow, turbulence and combustion*, vol. 60, no. 2, pp. 157–172, 1998.
- [20] J Dumond, F Magagnato, and A Class, “Stochastic-field cavitation model,” *Physics of Fluids (1994-present)*, vol. 25, no. 7, pp. 073302, 2013.

3

Stochastic numerical integration

3.1 Introduction

In this chapter, we start with a discussion on Monte Carlo method, which gradually transit to an introduction of Ito stochastic differential equation, then an extensive discussion on stochastic numerical integration schemes will be given.

3.2 Monte Carlo method

Monte Carlo methods are a broad class of computational algorithms that are used on a wide range of mathematical and physical problems. Several previous studies [1], [2] and [3] have applied Monte Carlo methods on the numerical so-

lution of probability density functions (PDF). In essence, Monte Carlo method is not concerned with capturing any particular realization of the solving variable. Through the random sampling process, a series of stochastic fields are taken as representations of the variable, who stochastically resembles the solving variable. Therefore, stochastic variables such as mean value of the original variable can be estimated by the stochastic variables of the fields. These approaches rely on random sampling processes to obtain numerical results, whereas vastly different procedures are seen in different approaches. In [3], a Monte Carlo method is presented to simulate the finite difference solution of the PDF transport equation of turbulent reactive flows. By considering the joint probability density function of the flow variables, the closure problem associated with non-linearities in the Navier-Stokes equation is avoided. In reactive flows, the typically non-linear chemical source terms are closed through the application of PDF method. A complete set of statistical description of chemical species and thermodynamic properties can be accessed through the introduction of joint pdf. The computational work involved in such method scales only linearly, in contrast to the power law dependence of a standard finite difference approach, with the number of independent variables of the joint pdf, therefore making it feasible for turbulent reactive flow with multiple species.

In [4], an Eulerian field approach is designed to solve a dynamically scalar transport equation.

As has been pointed out in [5], cavitation flows are inherently stochastic because of the uncertainty involved in water quality (nuclei size and nuclei number PDF variance) and turbulence-cavitation interaction. In the modelling realm, an inspection on the cutting edge cavitation models [6] would reveal the non-linear nature of such models. Given the restriction on computational and experimental capacities, solving for instantaneous number and size of bubbles at specific locations would be neither economically feasible nor necessary for most flow applications, in which mean variables suffice for a complete description of thermodynamic properties of the fluid. Therefore, a numerical tool that is capable of reproducing the statistical properties of the mean flow would

be of higher practical interest. In the current work, a stochastic field method that resembles the methodology being used in [5] has been applied. Some fundamental numerical experiments are performed to testify the various stochastic integration schemes available. In [5], a standard forth-order Range-Kutta scheme was applied to solve the stochastic field due to the robustness concern. However, such scheme is inherently inconsistent with the stochastic formulation. Some ground-breaking studies on stochastic integration schemes have already been conducted in the mathematical realm, e.g. [7], [8]. A comparison of these stochastic numerical integration schemes is needed to obtain the necessary insight into them. The particularly interesting questions are, what are the theoretically consistent options, and how their performances compare in practice. In the current work, a Matlab code was developed to perform 1-D and 2-D stochastic simulations of sample problems, based on which we concluded that under the ideal condition of relatively lower number of realizations, 2nd order stochastic Runge-Kutta (SRK2) method performs noticeably better than 1st order Eulerian-Maruyama (EM1) method, and therefore was chosen as the tool for the numerical integration of the stochastic cavitation model implemented in OpenFOAM.

3.3 PDF and Ito stochastic differential equations

In [4], a novel PDF representation is developed, in which the description of PDF does not rely on the concept of particles, as in previous approaches. In line with this pure Eulerian representation, the state-of-the-art numerical algorithms for partial differential equations can be applied. While the method has been elaborated on in [4], a brief summary of the method will be given in this section in order to give a self-contained explanation of the method.

The typical transport equation of a dynamically scalar field $\phi(x, t)$ that undergoes turbulent convection, molecular diffusion, and chemical source bears the following form:

$$\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} = D \frac{\partial^2 \phi}{\partial x_j \partial x_j} + S \quad (3.1)$$

where u_j, x_j follows the convention of tensor notation. And u_j denotes a solenoidal velocity field, and D represents the diffusion coefficient.

The time evolution equation for the scalar PDF of ϕ , with turbulent fluctuation and molecular mixing terms modeled [4], is

$$\frac{\partial P}{\partial t} + U_i \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial \psi} [SP] = \frac{\partial}{\partial x_i} (D' \frac{\partial P}{\partial x_i}) + \frac{\partial}{\partial \psi} [\frac{\omega_c}{2} (\psi - C) P]. \quad (3.2)$$

where C represents scalar mean value. The main idea of the stochastic fields method is that, instead of solving the transport equation of the PDF itself, with some stochastic procedures, we come up with a series of Eulerian fields, the purpose of which is not to recover any particular realization of the scalar field ϕ , but more importantly they shares the same PDF with scalar field ϕ . In the above equation, we have gradient terms in both the spatial coordinates and the sampling space. Before applying the stochastic procedure on the PDF equation above, we would need to reformulate it into a formula that is conducive to stochastic integration (i.e. a Fokker-Planck equation).

In stochastic theory, a Fokker-Planck equation (FPE) takes the following form [9]:

$$\frac{\partial P(\phi, t)}{\partial t} = - \frac{\partial}{\partial \phi} [a(\phi, t) P(\phi, t)] + \frac{1}{2} \frac{\partial^2}{\partial \phi^2} [b(\phi, t) P(\phi, t)], \quad (3.3)$$

where P denotes the PDF of ϕ , a, b are some arbitrary coefficients.

We will now reformulate the PDF into a FPE. In fact, what we need to do is just transform all spatial derivative terms into derivatives in the sampling variable space. It is therefore important to notice that:

$$\frac{\partial P}{\partial x_i} = - \frac{\partial}{\partial \psi} \left[\left\langle \frac{\partial \tau}{\partial x_i} \middle| \tau = \psi \right\rangle P \right] \quad (3.4)$$

where the angle brackets denotes the ensemble average of the contained quantity. For a more detailed discussion on PDF, please refer to [10]. Here we will give a proof of the above equation without covering much of the fundamentals.

As in [4], we start with the definition of PDF:

$$P(\psi; x, t) = \frac{1}{N} \sum_{n=1}^N \delta[\psi - \tau^n(x, t)] := \langle \delta[\psi - \tau(x, t)] \rangle \quad (3.5)$$

where δ denotes the Dirac delta function. It is well known, see e.g. [10], that the delta function has the following properties:

$$\delta(x - a) = \delta(a - x) \quad (3.6)$$

$$\delta^{(1)}(x - a) = -\delta^{(1)}(a - x) \quad (3.7)$$

where the superscript in brackets indicates a derivative and its order. Following the above relations, we have:

$$\begin{aligned} \frac{\partial P}{\partial x_i} &= \frac{\partial \langle \delta(\tau - \psi) \rangle}{\partial x_i} \\ &= \frac{\partial}{\partial x_i} \left(\frac{1}{N} \sum_{n=1}^N \delta[\tau^n(x, t) - \psi] \right) \\ &= \frac{1}{N} \sum_{n=1}^N \frac{\partial}{\partial x_i} \delta(\tau^n - \psi) \\ &= \frac{1}{N} \sum_{n=1}^N \frac{\partial \tau^n}{\partial x_i} \frac{\partial \delta(\tau^n - \psi)}{\partial \tau^n} \\ &= -\frac{1}{N} \sum_{n=1}^N \frac{\partial \tau^n}{\partial x_i} \frac{\partial \delta(\tau^n - \psi)}{\partial \psi} \\ &= \frac{\partial}{\partial \psi} \left[\frac{1}{N} \sum_{n=1}^N \frac{\partial \tau^n}{\partial x_i} \delta(\tau^n - \psi) \right] \\ &= \frac{\partial}{\partial \psi} \left[\left\langle \frac{\partial \tau}{\partial x_i} \middle| \tau = \psi \right\rangle P \right] \end{aligned}$$

Looking back at the PDF equation, the combined turbulent and molecular dif-

fusion term requires applying the above relation twice:

$$\begin{aligned} \frac{\partial}{\partial x_i} (\Gamma' \frac{\partial P}{\partial x_i}) &= \frac{\partial \Gamma'}{\partial x_i} \frac{\partial P}{\partial x_i} + \Gamma' \frac{\partial^2 P}{\partial x_i \partial x_i} \\ &= \frac{\partial \Gamma'}{\partial x_i} \frac{\partial P}{\partial x_i} - \Gamma' \frac{\partial}{\partial \psi} \left(\left\langle \frac{\partial^2 \tau}{\partial x_i \partial x_i} \middle| \tau = \psi \right\rangle P \right) \\ &\quad + \Gamma' \frac{\partial^2}{\partial \psi^2} \left(\left\langle \frac{\partial \tau}{\partial x_i} \frac{\partial \tau}{\partial x_i} \middle| \tau = \psi \right\rangle P \right) \end{aligned} \quad (3.8)$$

The first term on the RHS can be dealt together with the convection term by applying the derivative variable exchange relation for another time, then we get:

$$\begin{aligned} \frac{\partial P}{\partial t} &= \frac{\partial}{\partial \psi} \left[(U_i \left\langle \frac{\partial \tau}{\partial x_i} \middle| \tau = \psi \right\rangle - \frac{\partial \Gamma'}{\partial x_i} \left\langle \frac{\partial \tau}{\partial x_i} \middle| \tau = \psi \right\rangle \right. \\ &\quad \left. - \Gamma' \left\langle \frac{\partial^2 \tau}{\partial x_i \partial x_i} \middle| \tau = \psi \right\rangle \right) P \right] \\ &\quad + \frac{\partial^2}{\partial \psi^2} \left(\Gamma' \left\langle \frac{\partial \tau}{\partial x_i} \frac{\partial \tau}{\partial x_i} \middle| \tau = \psi \right\rangle P \right) \end{aligned} \quad (3.9)$$

Now the convection-diffusion components of the PDF transport equation has been reformulated into a FPE. All the remaining terms in the PDF are naturally derivatives of sampling variable.

In [9], it is proved that a stochastic process described by a FPE is equivalent to the Ito stochastic differential equation (SDE).

$$d\phi(t) = a[\phi(t), t]dt + \sqrt{b[\phi(t), t]}dW(t) \quad (3.10)$$

where $W(t)$ is a Wiener process, which is a continuous Gaussian stochastic process. The rigorous mathematical definition is not necessary here for the particular interest of engineering applications. Interested readers will find a scrutinised discussion in [9]. It is, however, important to understand that,

- $W(t) \sim N(0, t)$ for any $t \geq 0$
- for any $0 \leq s < t$, $[W(t) - W(s)] \sim N(0, t - s)$

where N denotes normal distribution. The above conclusions are drawn from the fact that $W(t)$ is a Gaussian process. We can further argue that,

$$\Delta W = \epsilon \sqrt{\Delta t}, \quad (3.11)$$

where $\epsilon \sim N(0, 1)$. We let Δt get infinitesimally small, then we get,

$$dW = \epsilon \sqrt{dt}. \quad (3.12)$$

Equipped with the knowledge on Wiener process we are ready to deal with the integral of the Ito SDE, namely

$$\phi(t) = \phi(t_0) + \int_{t_0}^t dt' a[\phi(t'), t'] + \int_{t_0}^t dW(t') b[\phi(t'), t']. \quad (3.13)$$

The next section will be devoted to the numerical integration of the Ito SDE.

3.4 Stochastic integration schemes

Having shown the theoretical correspondence of the PDF equation and the Ito stochastic differential equation, the priority of our work now is to think about what kind of numerical schemes we could possibly use to tackle a stochastic numerical integral. Out of intuition, it is natural to understand that the deterministic portions are not different with the normal integral, and therefore can be treated with any typical numerical integration schemes, e.g., Runge-Kutta type methods. In fact, in [5] a conventional Runge-Kutta scheme was applied due to the convenience and robustness of conventional numerical integration schemes. However in the strict sense, the existence of the Wiener process calls for a special type of numerical scheme.

The subject of SDE integration is a relatively young one with many intensive ongoing research. To be able to label the order of accuracy of SDE schemes as in conventional integration analysis, different definitions of accuracy would be needed, an elaboration which is beyond the scope of this work. Interested readers are referred to [11]. Based on the affinity with deterministic integration, some of the methods are selected in this work for further discussions:

- Euler-Maruyama method, a direct extension from traditional Euler first order scheme.
- Milstein method, a slightly more complex scheme compared to the Euler-Maruyama method.
- Stochastic 2nd order Runge-Kutta method(SRK2).

Below we would give a brief introduction to the Euler-Maruyama method, then some theoretical tools will be discussed to facilitate the derivation of the Milstein method that follows. Finally, the higher order SRK2, which will be applied in our cavitation model, will be presented.

3.4.1 Euler-Maruyama method

Euler-Maruyama method [12] is the simplest generalization of the Euler method of ordinary differential equation (ODE) to SDE. Consider the SDE:

$$d\phi(t) = a(\phi, t)dt + b(\phi, t)dW(t) \quad (3.14)$$

Suppose we solve the SDE on the interval $[0, T]$, with time step $\Delta t = T/N$, where N is the number of time steps, and initial condition $\phi(0) = \phi_0$. The **Euler-Maruyama approximation** to the true solution ϕ is the following series Y :

- set $Y_0 = \phi_0$
- recursive march the time

$$Y_{n+1} = Y_n + a(Y_n)\Delta T + b(Y_n)\Delta W_n \quad (3.15)$$

where $0 \leq n \leq N - 1$, $\Delta W_n = W_{n+1} - W_n$

The advantage of such scheme is obviously its simplicity, although, not surprisingly, the accuracy of such method is low. And unfortunately the idea of such simple extension cannot be directly transplanted into higher order SRK

schemes, as will be shown in the later section. Despite of the disadvantages, it is a convenient method especially suitable for preliminary solution and comparison purposes.

3.4.2 Some Theoretical Tools and Milstein method

Euler-Maruyama method is the simplest integration scheme that does not take much theoretical consideration. But in applications where higher accuracy is important, schemes with higher order of accuracy are usually adopted. However, without delicate theoretical tools, it is impossible to develop or to understand the gist of higher order schemes. In the discussion that follows, we will demonstrate the derivation of **Ito's Lemma** and **Lemperti transformation**. Then a derivation of the Milstein Method will be given by applying Taylor expansion.

Ito's Lemma

The basis of Ito SDE is **Ito's Lemma**, which is commonly compared with the chain rule in calculus. Suppose for any function $f(\phi, t)$ that is first order continuous in t and second order continuous in ϕ , we have,

$$\begin{aligned} df &= \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial \phi} d\phi + \frac{1}{2} \frac{\partial^2 f}{\partial \phi^2} d\phi^2 + o(d\phi^2) \\ &= \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial \phi} (a(\phi, t)dt + b(\phi, t)dW^2) \\ &\quad + \frac{1}{2} \frac{\partial^2 f}{\partial \phi^2} [a^2(\phi, t)dt^2 + 2a(\phi, t)b(\phi, t)dt dW + b^2(\phi, t)dW^2] + o(d\phi^2) \end{aligned} \quad (3.16)$$

From the previous discussion on the Wiener's process, we know that the dW term is of order $dt^{\frac{1}{2}}$, neglecting terms with order higher than dt in the above equation, we get,

$$df = \left(\frac{\partial f}{\partial t} + a(\phi, t) \frac{\partial f}{\partial \phi} + \frac{1}{2} b^2(\phi, t) \frac{\partial^2 f}{\partial \phi^2} \right) dt + b(\phi, t) \frac{\partial f}{\partial \phi} dW \quad (3.17)$$

Lamperti Transformation

Another tool that will be used later is called **Lamperti transformation**. For any SDE with the diffusion coefficient depending only on the state variable but not on time t ,

$$d\phi(t) = a(\phi, t)dt + b(\phi)dW(t) \quad (3.18)$$

Such a SDE can always be transformed into one with unitary diffusion coefficient by applying the Lamperti transformation

$$Y_t = F(\phi_t) = \int_z^{\phi_t} \frac{1}{\sigma(u)} du. \quad (3.19)$$

Here z is an arbitrary value in the state space of ϕ . The process Y_t solves the SDE below

$$dY_t = \left(\frac{a(t, F^{-1}(y))}{b(F^{-1}(y))} - \frac{1}{2}\sigma_\phi(F^{-1}(y)) \right) dt + dW_t, \quad (3.20)$$

which is just

$$dY_t = \left(\frac{a(t, \phi_t)}{b(\phi_t)} - \frac{1}{2}\sigma_\phi(\phi_t) \right) dt + dW_t. \quad (3.21)$$

Also the following properties will be used extensively later on in the scheme derivations

$$\begin{aligned} f(t, \phi) &= \int_z^\phi \frac{1}{\sigma(u)} du, & f_t(t, \phi) &= 0, \\ f_\phi(t, \phi) &= \frac{1}{\sigma(u)}, & f_{\phi\phi}(t, \phi) &= -\frac{\sigma_\phi(\phi)}{\sigma^2(\phi)}. \end{aligned}$$

Milstein Method

Milstein method is published the first time in [13]. We assume the time discretization and initial condition as in the previous section. The Milstein approximation to the true solution ϕ is the series Y defined as follows:

- set $Y_0 = \phi_0$

- recursive march the time

$$Y_{n+1} = Y_n + a(Y_n)\Delta T + b(Y_n)\Delta W_n + \frac{1}{2}b(Y_n)b'(Y_n)((\Delta W_n)^2 - \Delta t) \quad (3.22)$$

where b' denotes the derivative, $0 \leq n \leq N-1$, and $\Delta W_n = W_{n+1} - W_n$.

We will now start to look at the derivation of the scheme. Think about an SDE with diffusion coefficient independent of t . For any transformation $y = f(x)$ on the state variable, by Ito's Lemma we have

$$dY_t = \left(\frac{\partial f}{\partial t} + a \frac{\partial f}{\partial \phi} + \frac{1}{2}b^2 \frac{\partial^2 f}{\partial \phi^2} \right) dt + b \frac{\partial f}{\partial \phi} dW. \quad (3.23)$$

Let f be the Lamperti transformation, we get

$$f'(\phi) = \frac{1}{b}, \quad f''(\phi) = -\frac{b_\phi}{b^2}.$$

Plug the above into Ito's formulation,

$$dY_t = \left(\frac{a}{b} - \frac{1}{2}b_\phi \right) dt + dW, \quad (3.24)$$

and note that $\Delta Y \sim O(\Delta t^{\frac{1}{2}})$.

Denote the inverse of f as g , so $\phi = f^{-1}(y) = g(y)$. Then we apply the Taylor expansion on g

$$g(Y_i + \Delta Y_i) = g(Y_i) + g'(Y_i)\Delta Y_i + \frac{1}{2}g''(Y_i)(\Delta Y_i)^2 + O(\Delta Y_i^3). \quad (3.25)$$

Applying the chain rule of derivatives on g , we get

$$g'(y) = b, \quad g''(y) = bb_\phi.$$

Therefore, we have

$$\begin{aligned} \phi_{i+1} - \phi_i &= g(Y_i + \Delta Y_i) - g(Y_i) \\ &= b\Delta Y_i + \frac{1}{2}bb_\phi(\Delta Y_i)^2 + O(\Delta Y_i^3) \\ &= b\left(\frac{a}{b} - \frac{1}{2}b_\phi\right)\Delta t + b\Delta W + \frac{1}{2}bb_\phi(\Delta W)^2 \\ &= a\Delta t + b\Delta W + \frac{1}{2}bb_\phi((\Delta W)^2 - \Delta t) + O(\Delta t^{\frac{3}{2}}). \end{aligned} \quad (3.26)$$

Note that:

- When the diffusion term is not a function of ϕ , then $b_\phi = 0$, this method degrades to the Euler-Maruyama method.
- Due to the addition of the derivative term, this method normally has a higher accuracy as compared to the Euler-Maruyama method. However, it requires the knowledge of the derivative term.

In the next section, we will introduce the 2nd order stochastic Runge-Kutta scheme, which requires no knowledge of the derivative term and enjoys a high order of accuracy.

3.4.3 2nd order stochastic Runge-Kutta method

Other than the first-order methods discussed above, A.Rößler in [14] has developed a range of Runge-Kutta schemes with 2nd order accuracy in the weak sense (for the definition of strong and weak accuracy, please refer to [11]), by different choices of coefficients and number of stages. A discussion on the general scheme and the order of accuracy would be beyond the scope of the current work. Here we adopt a specific scheme that is proposed in [7] and [8]:

$$Y_{n+1} = Y_n + \frac{1}{2}a(t_n, Y_n)\Delta t + \frac{1}{2}a(t_n + \Delta t, Y_n + a(t_n, Y_n)\Delta t + b\Delta W) + b\Delta W \quad (3.27)$$

3.5 Numerical tests

Having all the theoretical background covered, some sample problems will be testified with the Euler-Maruyama and the SRK2 schemes. A finite difference code is written in matlab to perform these preliminary numerical experiments. In this section some results will be shown and discussed.

3.5.1 Test One

As in [3] and [4], we will solve a PDF transport equation that corresponds to a plug flow reactor configuration. With a simple molecular mixing model, the PDF equation becomes Equation (3.2) with the boundary conditions:

$$P(\phi; 0, t) = \delta(\phi), \quad \frac{\partial}{\partial x} P(\phi; 1, t) = 0$$

and the initial condition:

$$P(\phi; x, 0) = \delta(1 - \phi). \quad (3.28)$$

In order to apply Monte Carlo method, the transport equation for the PDF is transformed into the Eulerian field formulation:

$$\begin{aligned} d\phi^n = & \left[-\frac{\partial \phi^n}{\partial x_i} + D' \frac{\partial^2 \phi^n}{\partial x_i \partial x_i} - \frac{\omega}{2} (\phi^n - \langle \phi^n \rangle) + a_1 (1 - \phi^n) \right] dt \\ & + \sqrt{2D'} \frac{\partial \phi^n}{\partial x_i} dW. \end{aligned} \quad (3.29)$$

The above equation is non-dimensionalized. The first term on the RHS is the convection by mean velocity, which aligns x_1 direction in this plug flow problem and is non-dimensionalized into 1. The constant values adopted in the current problem are:

$$a_1 = 3, \quad \omega = 20, \quad D' = 0.1.$$

For the above values, the scalar mean value has the following analytical solution:

$$\bar{\phi}(x) = 1 - \exp(-2.416x). \quad (3.30)$$

We obtained the result as in Figure 3.1. Both methods seem to have produced satisfactory results.

3.5.2 Test Two

The second test resembles the one-dimensional problem in [7], a typical advection-diffusion equation with a stochastic forcing term:

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = \sigma \frac{\partial W}{\partial x}, \quad (3.31)$$

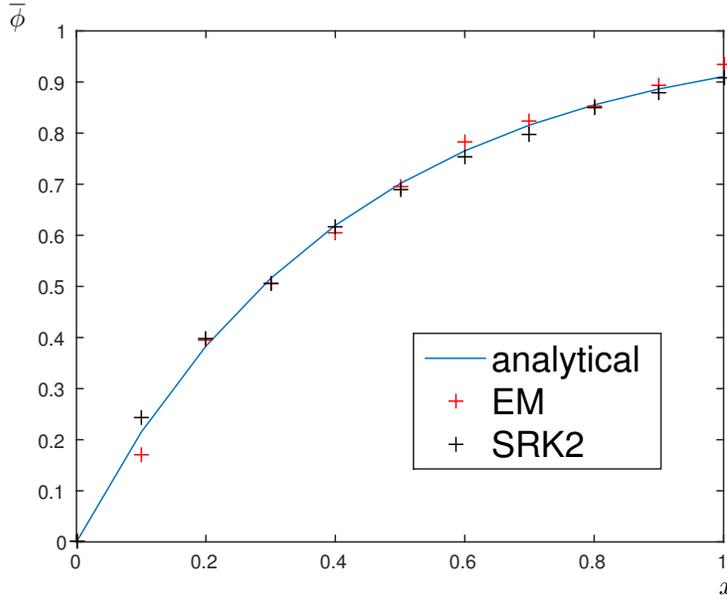


Figure 3.1: Test 1 for SRK and EM method, $dt=0.01$, $dx=0.1$, realizations=40

where $x \in [0, 1]$, velocity $v = 0.6$, diffusion coefficient $\nu = 0.005$, $\sigma = 2.5$ subject to the initial condition

$$u(0, x) = \exp\left(-\frac{(x - 0.2)^2}{\nu}\right), x \in [0, 1] \quad (3.32)$$

and the boundary conditions

$$\begin{aligned} u(t, 0) &= \frac{1}{\sqrt{4t + 1}} \exp\left(-\frac{(-0.2 - vt)^2}{\nu(4t + 1)}\right), \\ u(t, 1) &= \frac{1}{\sqrt{4t + 1}} \exp\left(-\frac{(-0.8 - vt)^2}{\nu(4t + 1)}\right). \end{aligned} \quad (3.33)$$

It has the following analytical solution for the expected value of u

$$\bar{u}(t, x) = \frac{1}{\sqrt{4t + 1}} \exp\left(-\frac{(x - 0.2 - vt)^2}{\nu(4t + 1)}\right), x \in [0, 1] \quad (3.34)$$

The results from the simulation, which are shown in Figure 3.2-Figure 3.4, agree well with the analytical solution. Here we are more interested in knowing how the two schemes behave with different time step and different number of realizations. As later on when calculating realistic problems, it is critical to control the computational cost. Figure 3.2 and Figure 3.3 offer a comparison on the number of realizations. As is revealed in the 10000 realizations cases, at high number of realizations, both methods produce very satisfactory results. Even the variances are highly converged. With a close inspection, SRK2 slightly outperforms Euler-Maruyama method at the range close to the peak. However, at a lower number of realizations, the SRK2 clearly achieves a higher level of agreement to the analytical expected value, especially at the peak region.

Figure 3.4 demonstrates the behavior of the two methods with different time step values. A trend similar to that with the number of realizations can be observed. Both of the methods agree very well with the analytical expected value, nonetheless the Euler-Maruyama gets a higher and higher deviation in contrast to SRK2, as the time step gets larger. Also SRK2 again captures the peak better than Euler-Maruyama method.

In summary, from the above figures, it is safe to conclude that with a refined time step and large number of realizations, both methods perform well. For realistic problems, in which the affordable computational cost limits the number of realizations (in the current studies, the number of Eulerian fields), and when a larger time step is usually desirable, SRK2 has a considerably high advantage over the Euler-Maruyama method. Therefore, we selected the SRK2 method in the final implementation of the cavitation model.

3.6 An Eulerian Field Monte Carlo Formulation of Volume Fraction

The details on stochastics covered in this chapter enable us to finalize the discussion on stochastic cavitation model that we left opened in the end of the last

chapter.

Recall that in the last chapter, a volume fraction equation, Equation (2.3), which is mathematically identical to Equation (3.1), has been derived. Likewise, the volume fraction shares the same PDF transport equation. Therefore, a similar transformation from PDF equation to Eulerian field Monte carlo formulation can be applied, which leads to:

$$d\alpha^n = \left[-u_i \frac{\partial \alpha^n}{\partial x_i} + D' \frac{\partial^2 \alpha^n}{\partial x_i \partial x_i} - \frac{\omega}{2} (\alpha^n - \langle \alpha \rangle) + S(\alpha^n) \right] dt + \sqrt{2D'} \frac{\partial \alpha^n}{\partial x_i} dW. \quad (3.35)$$

where the first term denotes the convection by filtered velocity field. The unresolved velocity from our LES model contributes via D' . At this point, the formulation could readily be handled by the stochastic numerical integration schemes.

Bibliography

- [1] Stephen B Pope, "PDF methods for turbulent reactive flows," *Progress in Energy and Combustion Science*, vol. 11, no. 2, pp. 119–192, 1985.
- [2] BJ Delarue and SB Pope, "Calculations of subsonic and supersonic turbulent reacting mixing layers using probability density function methods," *Physics of Fluids (1994-present)*, vol. 10, no. 2, pp. 487–498, 1998.
- [3] Stephen Bailey Pope, "A monte carlo method for the PDF equations of turbulent reactive flow," 1981.
- [4] Luis Valiño, "A field monte carlo formulation for calculating the probability density function of a single scalar in a turbulent flow," *Flow, turbulence and combustion*, vol. 60, no. 2, pp. 157–172, 1998.
- [5] J Dumond, F Magagnato, and A Class, "Stochastic-field cavitation model," *Physics of Fluids (1994-present)*, vol. 25, no. 7, pp. 073302, 2013.
- [6] Christopher E Brennen, *Cavitation and bubble dynamics*, Cambridge University Press, 2013.

- [7] Andreas Röbler, Mohammed Seaïd, and Mostafa Zahri, “Method of lines for stochastic boundary-value problems with additive noise,” *Applied Mathematics and Computation*, vol. 199, no. 1, pp. 301–314, 2008.
- [8] Mostafa Zahri and KSA Al Madinah, “On numerical schemes for solving a stochastic advection-diffusion,” .
- [9] Crispin W Gardiner et al., *Handbook of stochastic methods*, vol. 3, Springer Berlin, 1985.
- [10] Stephen B Pope, “Turbulent flows,” 2001.
- [11] Stefano M Iacus, *Simulation and inference for stochastic differential equations: with R examples*, Springer Science & Business Media, 2009.
- [12] Klöden Platen, “Numerical solution of stochastic differential equations berlin springer,” 1995.
- [13] GN Mil’shtejn, “Approximate integration of stochastic differential equations,” *Theory of Probability & Its Applications*, vol. 19, no. 3, pp. 557–562, 1975.
- [14] Andreas Röbler, “Runge–Kutta methods for Itô stochastic differential equations with scalar noise,” *BIT Numerical Mathematics*, vol. 46, no. 1, pp. 97–110, 2006.

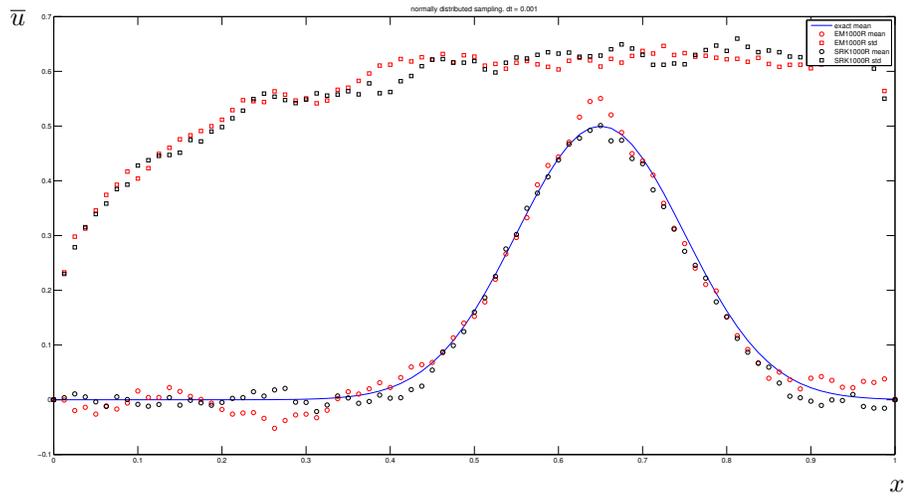


Figure 3.2: Test 2 for SRK and EM method, realizations=1000

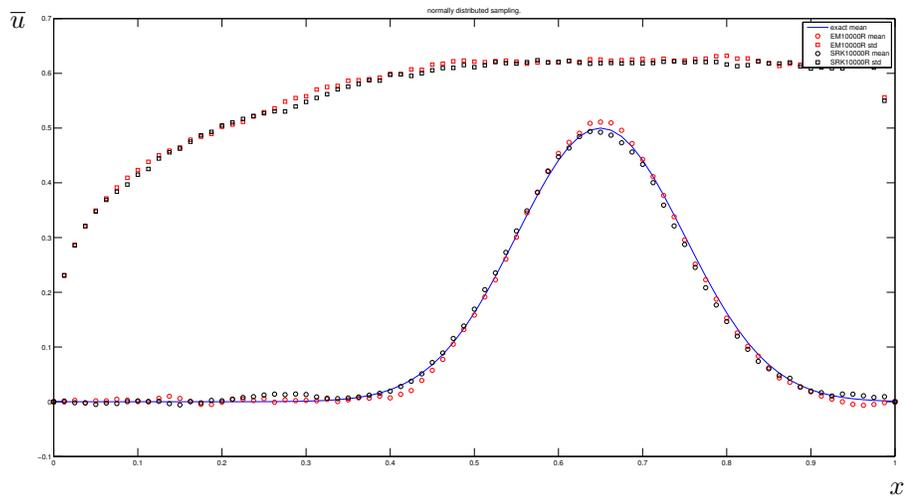


Figure 3.3: Test 2 for SRK and EM method, realizations=10000

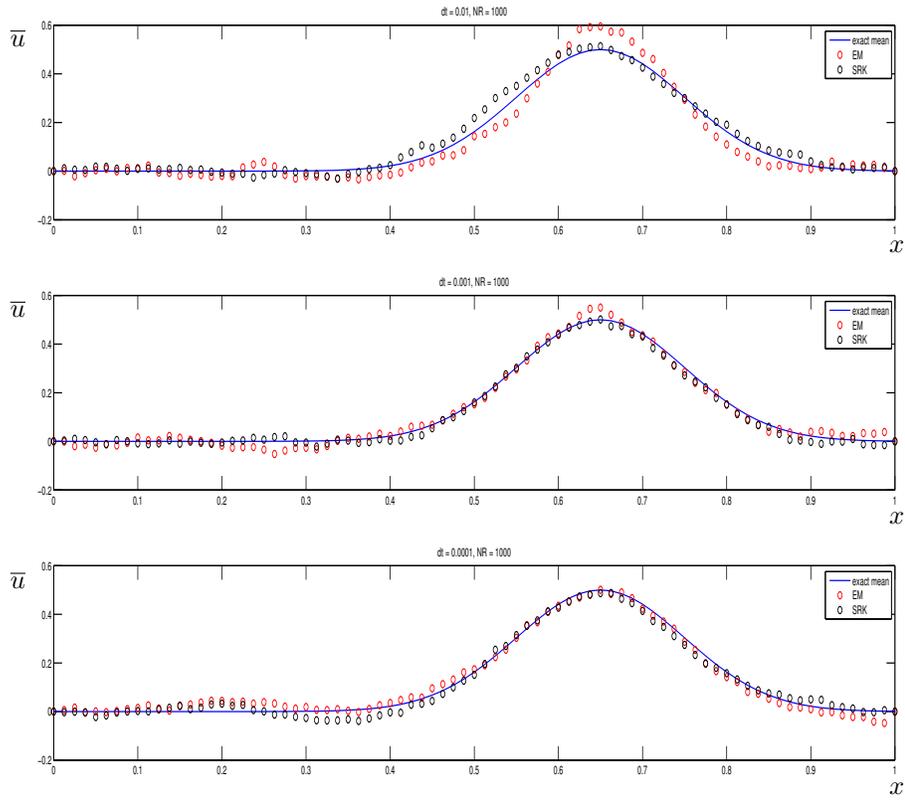


Figure 3.4: Test 2 for SRK and EM method, time step comparison

4

Test case

4.1 Introduction

In this chapter, a preliminary test case for stochastic field method will be demonstrated. We opted for a 2 dimensional configuration as in standard cavitating-Foam case (available in the majority of OpenFOAM releases) for its simplicity. Comparison is made between the results from stochastic field solver, `interPhaseChangeFDFFoam` and regular `interPhaseChangeFoam`.

4.2 Geometry and case setup

A simplified geometry of an academic nozzle geometry (see Figure 4.1), with a rectangular contraction in the middle of the domain, is used to present both the cavitation inside the nozzle and the flow downstream. A higher mesh concentration is adopted around the nozzle and immediate upstream and downstream portion of the domain.

With respect to the boundary conditions, we followed the setting in cavitating-

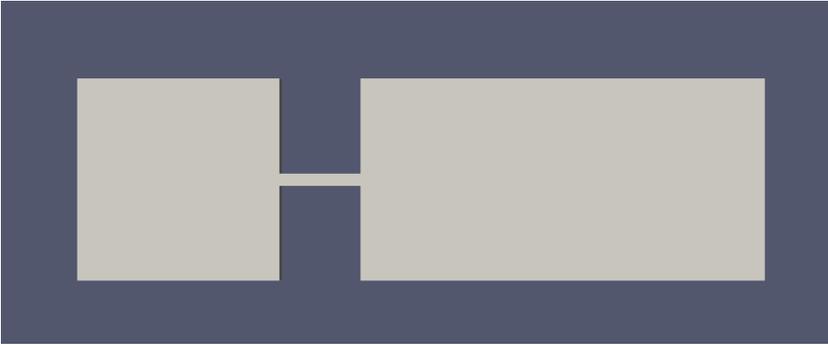


Figure 4.1: *2D nozzle geometry*

Foam case, in which 300bar, and 100bar pressure are imposed on the inlet and outlet respectively. Velocity boundaries (mass fluxes) are left free to adjust to the pressure difference, much like in the case of a Poiseuille flow.

4.3 Results

Two cases with the same settings have been simulated with `interPhaseChangeFoam` and `interPhaseChangeFDFFoam` respectively. In the latter case, we take 10 stochastic fields as the presentation of the PDF, for 8 were used in previous works [1] [2]. Snapshots of velocity fields and volume fraction fields are taken at 0.01s, which is several times of flow through time.

In Figure 4.2 and Figure 4.3, very good agreement on the velocity prediction

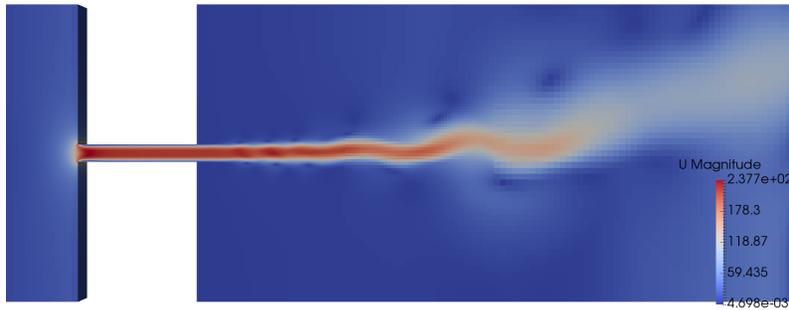


Figure 4.2: *Velocity field, interPhaseChangeFoam*

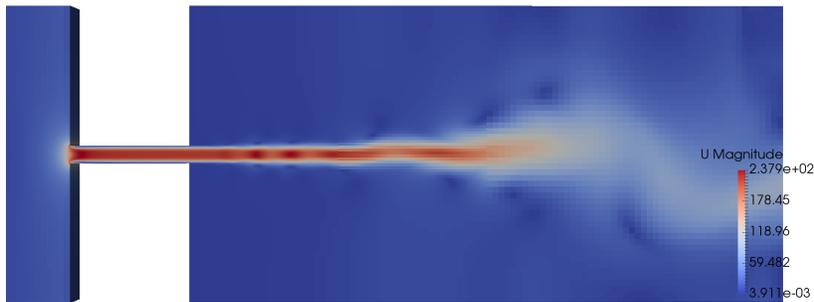


Figure 4.3: *Velocity field, interPhaseChangeFDFFoam*

has been achieved both in terms of the scale of velocities and the flow behavior downstream, as the scales of the curvatures are similar.

With respect to alpha fields in Figure 4.4 and Figure 4.5, in both cases we observe cavity regions extended from the nozzle inlet all the way out of the nozzle, after which, shear induced vorticities close to the nozzle outlet create low pressure spots at their centers, hence the inception of cavitation, which correspond well with the wiggles we have seen in the velocity fields. The value of vol-

ume fraction is also predicted with satisfactory accuracy. For the purpose of demonstration for the stochastic method, snapshots of 4 volume fraction fields are provided, see Figure 4.6- Figure 4.9. The difference involved in them can easily be observed.

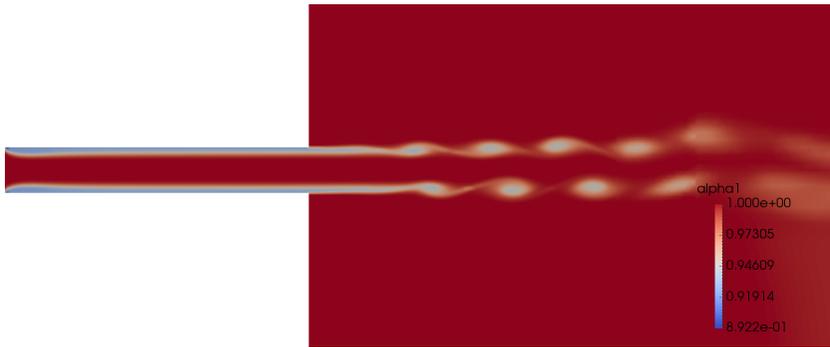


Figure 4.4: *Volume fraction field, interPhaseChangeFoam*

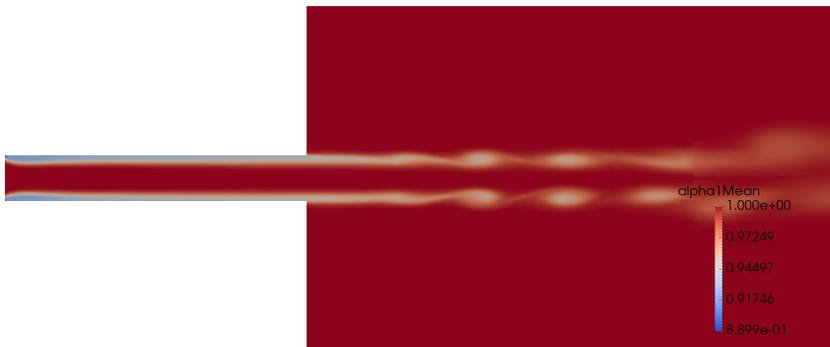


Figure 4.5: *Mean volume fraction field, interPhaseChangeFDFoam*

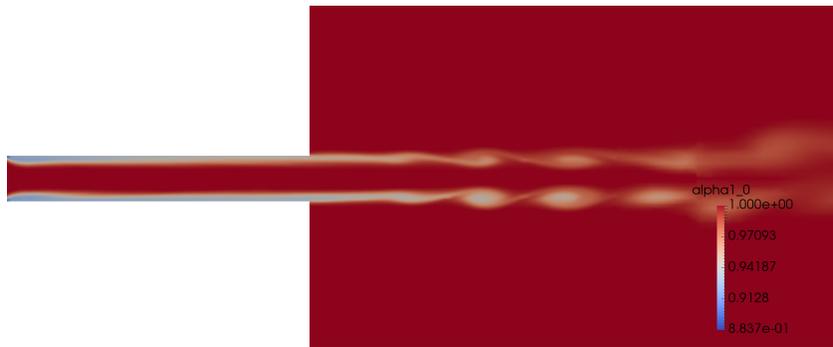


Figure 4.6: Volume fraction field number 1, *interPhaseChangeFDFFoam*

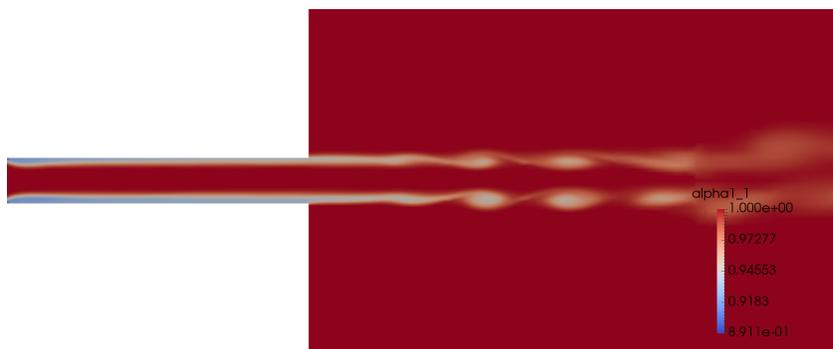


Figure 4.7: Volume fraction field number 2, *interPhaseChangeFDFFoam*

Bibliography

- [1] J Dumond, F Magagnato, and A Class, “Stochastic-field cavitation model,” *Physics of Fluids (1994-present)*, vol. 25, no. 7, pp. 073302, 2013.
- [2] WP Jones and S Navarro-Martinez, “Large eddy simulation and the filtered probability density function method,” in *LES AND DNS OF IGNITION PROCESSES AND COMPLEX-STRUCTURE FLAMES WITH LOCAL EXTINCTION Proceedings of the International COST Workshop*. AIP Publishing, 2009, vol. 1190, pp. 39–62.

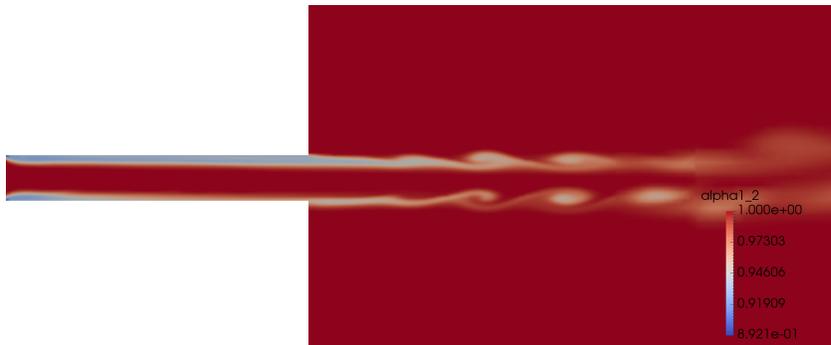


Figure 4.8: Volume fraction field number 3, *interPhaseChangeFDFFoam*

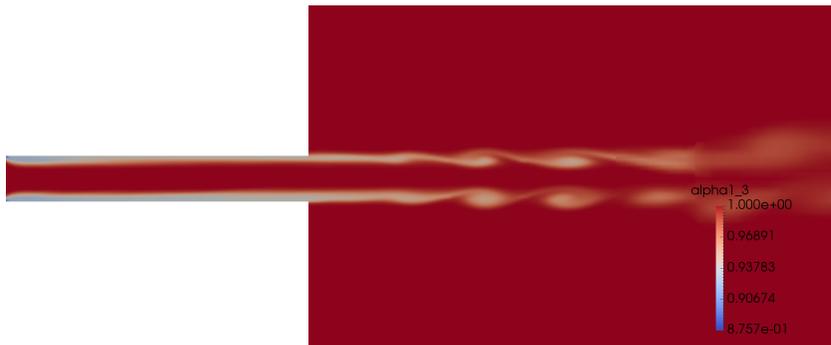


Figure 4.9: Volume fraction field number 4, *interPhaseChangeFDFFoam*

Appendices



OpenFOAM implementation

A.1 Introduction

In the previous chapters, the theoretical and algorithmic details have been discussed in preparation for the code implementation of the current study. This chapter focus specifically on the details involved in the coding aspect. Firstly, an introduction and general perspectives will be given on the numerical tool that will be used, namely OpenFOAM (Open Field Operation and Manipulation) and the major standard solvers pertinent to the current study, with the particular aim of connecting the standard implementation with the theoretical background. Secondly, we will briefly walk through the code structure of the cavitation models that has already been available in the standard release, in order to lay a perceptual foundation for the implementations of stochastic mod-

ule. In the last section, a demonstration of the implementation of the stochastic model, as well as the underlying stochastic integration library will be given.

Much unlike in the previous chapters, where theory presents heavily, the contents of this chapter are organised with the aim that is twofold: First is to show the code implementation of the model in a reader-friendly way; second is to provide an example on how to harness the available code snippets in OpenFOAM package to fulfil our purpose in practice. It has always been a sincere hope of the author that more pragmatic tutorials of OpenFOAM would become available to the users, so that they could handle the code with ease, and such convenience would hopefully in turn attracts more users into the open source community. Because the collaboration under the spirit of open source is, in the author's opinion, the most effective way of fostering the development of CFD.

A.2 OpenFOAM and its standard solvers

Computational Fluid Dynamics (CFD) has been well established since its earliest applications in meteorology in the early 20's. With the increasing computational power becoming available at a lower price, the application of computational models has become an indispensable part of the studies on the respective physical problems in both academical and industrial practices. Several commercial CFD packages, e.g., Star-CD, Fluent, CFX, FIRE, have accomplished great success and have long been available in the market. However, with the ever increasing complexities and amount of details involved in the problems of interest, reducing the overhead involved in the licence costs has also become increasingly more important for CFD users. In line with this concern, OpenFOAM (Open Field Operation and Manipulation) has attracted a substantial amount of attention from both academical and commercial users since its first release in 2004. Researchers in the multiphase flow area have been using OpenFOAM extensively since some of the earliest release, and a range of well developed solvers are readily available in the OpenFOAM standard releases. The ongoing OpenFOAM implementation and debugging effort in the multi-

phase area by the community has made OpenFOAM an excellent platform for high-level solver development. Some major solvers that are worth mentioning are (Lagrangian particle based solvers are not included here due to the lower relevance to the current work):

- `interFoam` [1], solver for 2 incompressible, isothermal immiscible fluids using a VOF (volume of fluid) phase-fraction based interface capturing approach. Its multiple fluids counterpart `multiphaseInterFoam` has adopted the same approach.
- `twoPhaseEulerFoam`, solver for a system of 2 compressible fluid phases with one phase dispersed, e.g. gas bubbles in a liquid including heat-transfer.
- `cavitatingFoam`, transient cavitation code based on the homogeneous equilibrium model from which the compressibility of the liquid/vapour “mixture” is obtained.
- `interPhaseChangeFoam` [2], solver for 2 incompressible, isothermal immiscible fluids with phase-change (e.g. cavitation). Uses a VOF (volume of fluid) phase-fraction based interface capturing approach.

Since the purpose of the project is to develop an Eulerian stochastic model in the VOF framework, and the state-of-the-art simplified Rayleigh-Plesset model [3] [4] has already been implemented in `interPhaseChangeFoam`, `interPhaseChangeFoam` is selected as the basis of the further development of the current work. In the next chapter, below we would focus mainly on the implementation details of the new model, namely the SRK2 stochastic integration module, and the stochastic model library. The details of the code provided in this section is based on Foam extended 3.1, minor variations may apply to different releases.

A.3 Implementation of SRK2

A good practice of OpenFOAM development involves selecting the available code pieces to start with. And as was mentioned before, the deterministic part of the stochastic integration closely resembles the normal numerical integration. As an open source project with some past development, OpenFOAM has an ODE library that serves this purpose. In the ODE library, the RK directory contains the components for a 4th order Runge-Kutta integration. A few steps can be taken to modify the ODE libraries into the stochastic integration library we need:

- Take a copy of the original ODE library, then remove the numerical methods that are irrelevant to our purpose. In foam extended 3.1, Euler, KRR4, SIBS directories should be removed. Meanwhile corresponding changes in Make files have to be made.
- Tailor down the 4th order RK scheme into a 2nd order RK. This mainly involves deleting many integration coefficients defined in the first few lines of RK.C file, and those in the constructor. Then remove the excessive integration stages in function "solve". A compile and test on the 2nd order RK method at this point is highly recommended before proceeding.
- Implement the stochastic part. The class "Random" is a random number generator in OpenFOAM that can be used to perform a Gaussian sampling (cachedRandom class is not kept in foam extended 3.1, but it can also be used if available). The stochastic contribution should then be added into the corresponding lines of the "solve" function.

Note that keeping the code structure of ODE library enables an easy potential addition of other numerical integration schemes into the library. Another scheme can be easily implemented independent on the existing SRK2 components.

A.4 Implementation of stochastic model

The stochastic ODE library (which will be referred to as `stchODE` from now on, following the naming in the implementation) only solves the stochastic ODE, but a module that assembles the equation is still needed. In standard `interPhaseChangeFoam` solver, the equation of volume fraction exist in `alphaEqn.H`, and it is numerically solved in the `MULES` solver which is located in `/finiteVolume/fvMatrices/solvers`. What we need to achieve here is to build a substitution for `alphaEqn.H` which will be solved by our `stchODE` library.

The question that follows during the implementation was: "What are the essential features of the stochastic model that are fundamentally different from those of the `interPhaseChangeFoam`?" In author's opinion, implementation-wise the differences lie in two aspects:

- Stochastic method solves realisations of volume fraction on multiple Eulerian fields instead of just one. So we need to find where the Eulerian field is defined in `interPhaseChangeFoam`, and add the multi-fields feature into that structure for our application.
- The solving of stochastic method relies on the `stchODE` module we have developed, which means an ideal target template should be some class that resorts to ODE library for solution. That template class can then be modified to assemble our stochastic equation, and call `stchODE` for solution.

The following two sections contain brief discussions that support these two aspects with details on the implementation level. For better clarity, readers are recommended to refer to the corresponding code snippets.

A.4.1 `stochasticPhaseChangeTwoPhaseMixture`

First off, we have to locate the Eulerian field in `interPhaseChangeFoam`. It is obvious that in `creatFields.H` of `interPhaseChangeFoam`,

```
volScalarField alpha1
```

```

(
    IObject
    (
        "alpha1",
        runTime.timeName(),
        mesh,
        IObject::MUST_READ,
        IObject::AUTO_WRITE
    ),
    mesh
);

```

declares the Eulerian field. Then we notice that in the declaration of `phaseChangeTwoPhaseMixture`,

```

...
    autoPtr<phaseChangeTwoPhaseMixture> twoPhaseProperties =
        phaseChangeTwoPhaseMixture::New(U, phi, "alpha1");
...

```

A character string is passed to the `runTime` selection (of the cavitation models). Digging deeper into the class `phaseChangeTwoPhaseMixture`, we find no trail of `alpha1` inside the `phaseChangeTwoPhaseMixture` class itself. However, a closer inspection on `phaseChangeTwoPhaseMixture.H` reveals that,

```

class phaseChangeTwoPhaseMixture
:
    public twoPhaseMixture
{
    ...

```

could have made the manipulation of `alpha1` possible for `phaseChangeTwoPhaseMixture` (such manipulations can be found in any of the cavitation models, e.g., in `SchnerrSauer.C`), through the inheritance from `twoPhaseMixture` structure. Indeed, after locating the `twoPhaseMixture` files (in `/src/transportModels`

/incompressible/incompressibleTwoPhaseMixture), in twoPhaseMixture.H, we find the following line in the class declaration:

```
...
    const volScalarField& alpha1_;
...
```

which hints at the chance that twoPhaseMixture might communicate with the alpha1 in createFields.H through the above reference. This can easily be confirmed by the constructor in twoPhaseMixture.C:

```
...
    alpha1_
    (
        U_.db().lookupObject<const volScalarField>
        (
            alpha1Name
        )
    ),
...
```

so the lookupObject function searches for the alpha1 field, and returns the reference to twoPhaseMixture.

Having understood the principle of how phaseChangeTwoPhaseMixture works, we can start to build the block for our purpose. In order to operate on the multiple Eulerian fields, the PtrList structure is used in our current implementation. In author's implementation, stochasticTwoPhaseMixture.H

```
...
    PtrList<volScalarField> alpha1_;
...
```

And in the constructor of stochasticTwoPhaseMixture in the corresponding C file, the number of the fields is first passed to the constructor, and then a mechanism that identify, name, and initialise the Eulerian fields is implemented.

```
...
forall(alpha1_, i)
{

    char buffer[8];

    sprintf(buffer, "%d", i);

    Foam::word stringI(buffer);

    IOobject header
    (
        "alpha1_" + stringI,
        U_.mesh().time().timeName(),
        U_.mesh(),
        IOobject::NO_READ
    );

    // check if field exists and can be read
    if (header.headerOk())
    {
        alpha1_.set
        (
            i,
            new volScalarField
            (
                IOobject
                (
                    "alpha1_" + stringI,
                    U_.mesh().time().timeName(),
                    U_.mesh(),
                    IOobject::MUST_READ,

```

```

        IOobject::AUTO_WRITE
    ),
    U_.mesh()
)
);
}
else
{
    alpha1_.set
    (
        i,
        new volScalarField
        (cd
        IOobject
        (
            "alpha1_" + stringI,
            U_.mesh().time().timeName(),
            U_.mesh(),
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        alpha1Mean_
    )
);
}
}
...

```

Other auxiliary functionalities, such as obtaining the mean value of Eulerian fields, are also actuated in `stochasticPhaseChangeTwoPhaseMixture`. Meanwhile, the cavitation models, for instance, `SchnerrSauer.C` require some modification to fit into the multi-field structures. However, since the work involved in these aspects are fairly straight forward but scattered in the code, they will

not be covered here due to the limited volume of this instruction. Interested readers are highly encouraged to refer to the code for more details. While the application of such mechanism will be touched upon again in the next section, where the equation is assembled.

A.4.2 stochasticModel and stochasticSolver

As has been mentioned before, since we designed our stchODE library in a way largely similar to the ODE library. So for the equation assembling purpose, it is natural to seek for a structure in foam that resorts to ODE for solution. Some code pieces in foam provides such examples, e.g. chemistryModel and chemistrySolver in thermophysicalModels. So the design of stochastic model vaguely follows the pattern in chemistryModel in the sense that stochasticModel provides necessary functions that interface with ODE library, such as nEqns, derivatives, and jacobian, and stochasticSolver inherits stochasticModel, unwraps it, and calls the stchODE solver for solution in the end. In the following content of this section, we would briefly walk through the final implementation of stochasticModel and stochasticSolver and commentate as it is due.

stochasticModel.H

stochasticModel inherits stchODE class:

```
...
class stochasticModel
:
    public stchODE
...

```

Such inheritance guarantees that stochasticModel, or any derived class of it would be able to call the solver in stchODE. In fact we would see later that the this is done by stochasticSolver, who includes stochasticModel as a member.

```
...
```

```

    autoPtr<stochasticPhaseChangeTwoPhaseMixture>
    twoPhaseProperties_;

    transportModel& twoPhaseTransport_;
...

```

These references and pointer are basically a replication of those in the createFields.H of interPhaseChangeFoam, as the transport properties are necessary for the update of properties, when the new solution becomes available.

As mentioned previously,

```

...
    virtual label nEqns() const;

    virtual void derivatives
    (
        const scalar t,
        const scalarField& y,
        scalarField& dydt
    ) const;

    virtual void stochasticTerm
    (
        const scalar t,
        const scalarField& y,
        vectorField& stch
    ) const;

    virtual void jacobian
    (
        const scalar t,
        const scalarField& y,
        scalarField& dfdt,

```

```

        scalarSquareMatrix& dfdy
    ) const;
...

```

Are the functions that interface with stchODE library. Here, they are virtual functions but not pure virtual ones because in stochasticMode.C they will be loaded with our stochastic equations.

stochasticModel.C

As shown in the last section,

```

...
    twoPhaseProperties_
    (
        stochasticPhaseChangeTwoPhaseMixture::New
        (
            NoR,
            U,
            phi,
            alpha1Name
        )
    ),
    twoPhaseTransport_
    (
        dynamic_cast<transportModel&>
        (
            twoPhaseProperties_()
        )
    ),
...

```

These are the handles for the properties' update.

```

...

```

```

Foam::label Foam::stochasticModel::nEqns() const
{
    return twoPhaseProperties_
        ->alpha1Mean_.internalField().size();
}
...

```

We will pass all the internal nodes of the Eulerian fields to stchODE for solution.

```

...
void Foam::stochasticModel::derivatives
(
    const scalar t,
    const scalarField &y,
    scalarField& dydt
) const
{
    volScalarField yc(twoPhaseProperties_->alpha1Mean_);

    volScalarField detRHS
    (
        IOobject
        (
            "dydt",
            mesh_.time().timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::NO_WRITE,
            false
        ),
        mesh_,
        dimensionedScalar("zero", dimless/dimTime, 0.0)
    );
}

```

```

);

yc.internalField() = y;
yc.correctBoundaryConditions();
...

```

The derivatives function provides the deterministic portion of the RHS source. The reason why we build `volScalarField` `yc` here is because, we need the solution of the old step (both `internalField` and the boundary condition) later on to calculate the deterministic source. However, the interface of virtual function derivatives has to follow the same pattern as that in `stchODE`, and the solution of the old step can only be seen through the reference of `scalarField` `y` (which means boundary conditions are missing). So here we recover the old solution by taking a new field `yc`, which is declared and initialised with the mean solution field. It follows that `yc` would get the same boundary conditions as the mean field, which actually are the same boundary conditions with solution fields of each realisations. And once `yc` has taken `y`, the boundary conditions is updated, then the `yc` becomes the full old solution field.

```

...
const surfaceScalarField& phiRef
    = mesh_.lookupObject<surfaceScalarField>("phi");

Pair<tmp<volScalarField> > vDotAlphal
    = twoPhaseProperties_->vDotAlphal(yc);

const volScalarField& vDotvAlphal = vDotAlphal[1]();

detRHS = (
    //- Diffusion
    fvc::laplacian
    (
        turbulence_->nuEff(),

```

```

        yc
    )
    //- Convection
- fvc::div
    (
        phiRef,
        yc
    )
+ fvc::Sp(fvc::div(phiRef),yc)
    //- sgs molecular mixing
    // In the explicit source function,
    // only the first input get returned.
- fvc::Su
    (
        scalar(0.5)
        * (yc - twoPhaseProperties_->alpha1Mean_)
        / tsgs(),
        yc
    )
);

dydt = detRHS.internalField();
dydt += vDotvAlpha1.internalField()*yc.internalField();
...

```

These portion assembles the deterministic source. We can see that all the fvc operations work on volScalarField, not scalarField, and this justifies our round-about representation of the old solution field. Since dydt has to be scalarField, it takes back only the internalField information from detRHS. The last line represents the contribution from the cavitation model. A similar mechanism has been used in function stochasticTerm to recover the full solution field, which needs no further elaboration.

stochasticSolver

Since the content of stochasticSolver is rather straightforward, we would not go over the files line by line. However it is worth mentioning that,

```

...
    PtrList<volScalarField>& alpha1 = stchModel_.alpha1();

    forAll(alpha1, i)
    {

        scalarField& ci = alpha1[i].internalField();

        stchODESolver_->solve
        (
            stchModel_,
            t0,
            t0 + dt,
            ci,
            deltaT
        );
    }
...

```

extracts the internal nodes values and pass them to the stchODE solver. If we have a look at the stchODE solver (The reader is recommended to check out the stchODESolver.C file in the directory: /applications/solvers/interPhaseChangeFDFFoam /stchODE/stchODESolvers/stchODESolver as continue reading),

```

...
void Foam::stchODESolver::solve
(
    const stchODE& ode,
    const scalar xStart,
    const scalar xEnd,

```

```

    scalarField& y,
    scalar& h
)
{
    const label MAXSTP = 10000;

    scalar x = xStart;

    for (label nStep=0; nStep<MAXSTP; nStep++)
    {
        ode.derivatives(x, y, dydx_);

        ode.stochasticTerm(x, y, stch_);
    ...
        solve(ode, x, y, dydx_, stch_, h);
    ...
}

```

Comparing the interface of the above function and that of `stochasticSolver::solve` function (Note that there is another `solve` function that has a different interface. It is a different function, following C++ syntax), it becomes obvious that `stchODESolver::solver` takes the `internalField` values of each of the alpha field from `stochasticSolver::solve`, pass them to `ode.derivatives` and `ode.stochasticTerms`, which are in fact the corresponding functions in `stochasticModel` that we have covered in the previous section. These functions return the deterministic and stochastic source in the form of `dydx_` and `stch_`, which are then passed to the core `stchODE` solver. When the solution is completed, back in `stochasticSolver.C` file,

```

...
    stchODESolver_ -> solve
    (
        stchModel_,
        t0,

```

```
        t0 + dt,  
        ci,  
        deltaT  
    );  
  
    alpha[i].correctBoundaryConditions();  
    ...
```

The `ci`, which refers to the `internalField` of `alpha[i]` gets updated, and the last line updates the boundary conditions of `alpha[i]`. When the loop of `i` is finished, the solution of all Eulerian fields is completed.

Bibliography

- [1] Suraj S Deshpande, Lakshman Anumolu, and Mario F Trujillo, “Evaluating the performance of the two-phase flow solver `interFoam`,” *Computational science & discovery*, vol. 5, no. 1, pp. 014016, 2012.
- [2] ST Miller, Hrvoje Jasak, DA Boger, EG Paterson, and A Nedungadi, “A pressure-based, compressible, two-phase flow finite volume method for underwater explosions,” *Computers & Fluids*, vol. 87, pp. 132–143, 2013.
- [3] Weixing Yuan, Jürgen Sauer, and Günter H Schnerr, “Modeling and computation of unsteady cavitation flows in injection nozzles,” *Mécanique & industries*, vol. 2, no. 5, pp. 383–394, 2001.
- [4] Weixing Yuan and Günter H Schnerr, “Numerical simulation of two-phase flow in injection nozzles: Interaction of cavitation and external jet formation,” *Journal of Fluids Engineering*, vol. 125, no. 6, pp. 963–969, 2003.