Numerical Methods for Simulating Dilute Polymeric Fluids

Shenghan Ye

Keble College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2018

Acknowledgements

I would like to express my special gratitude to my supervisor, Professor Endre Süli, who helped me a lot in conducting my research these past four years. Furthermore, I would also like to thank Professor Mike Giles, who has helped me many times by answering questions related to the project and invited me to the multilevel Monte Carlo group. Finally, I would like to thank all members of the Numerical Analysis Group at the University of Oxford for all their support. It has been a joy to work in such a nice environment.

Abstract

Numerical Methods for Simulating Dilute Polymeric Fluids

Shenghan Ye, Keble College, Doctor of Philosophy, Trinity 2017

Navier–Stokes–Fokker–Planck systems are coupled systems of partial differential equations arising from statistical physics as mathematical models of dilute polymeric fluids. The solvent is assumed to be a viscous incompressible Newtonian fluid, whose evolution in time is modelled by the Navier–Stokes equations; the elastic effects exhibited by the dilute polymeric fluid are modelled by the elastic extra stress tensor, whose spatial divergence appears on the right-hand side of the Navier–Stokes momentum equation. In this thesis we investigate and implement several numerical methods for the computational simulation of dilute polymeric fluids. The Fokker–Planck equation featuring in the model is a high-dimensional transport-diffusion equation, whose numerical solution by conventional means is extremely challenging since standard numerical methods applied to this equation suffer from the curse of dimensionality. The key objective of the thesis is therefore to develop an efficient numerical approximation scheme for the model, where deterministic approximation techniques for the Fokker–Planck equation are replaced by a multilevel Monte Carlo method. Under suitable assumptions, we can prove exponential convergence in time of the solution to an equilibrium solution: the Maxwellian of the model. At the end of the thesis we perform a series of numerical experiments, including multi-bead simulations for polymer molecules modelled as bead-spring chains, with FENE-type (finitely extensible nonlinear elastic) spring potentials, to explore the practical performance of the proposed numerical method.

Contents

List of Figures

List of Symbols

$\mathbb{R}$	set of all real numbers
d	space dimension
$\vec{x}, \vec{q}$	physical and configuration space variables
t	time variable
$\vec{u}$	velocity
$\boldsymbol{\kappa}$	velocity gradient
p	pressure
ρ	solvent density
$\boldsymbol{\tau}$	polymeric extra-stress tensor
Ω, D	physical and configuration domain
$\vec{F}$	FENE force
b	maximum extension to which the dumbbell can be stretched
Wi, Re	Weissenberg number and Reynolds number
ψ	solution to the Fokker–Planck equation
ψ_0	initial condition to the Fokker–Planck equation
M	Maxwellian for the force $\vec{F}$
P	Poincaré's constant
$\vec{Q}$	solution to the stochastic differential equation
$\vec{W}$	Brownian motion
L	level for the multilevel Monte Carlo method
ϵ	root mean square error
N	number of simulations
Δt	time step
$\mathbb{E}$	expectation
R	refinement factor
σ	clamping factor
π	invariant measure
λ	contractive Lipschitz constant
K	number of chains
σ	extension rate of the flow

Chapter 1

Introduction

1.1 Motivation

Fluid motion can be seen every day and everywhere in our life, and it therefore attracted researchers for centuries. The well-known Navier–Stokes equations are used to describe the motion of a class of liquids and gases, which are called Newtonian fluids. A fluid is said to be Newtonian if the shear stress at every point is proportional to the local strain rate; such fluids include water and most gases. However there still remain a class of fluids which can not be correctly described by the Navier–Stokes equations; for instance, blood and oils with polymeric additives. These fluids have more complex constitutive relations and are often referred to as non-Newtonian fluids. The study of non-Newtonian fluids has been an active research topic since the last century. With the advances in scientific computing, researchers have become interested in more and more complicated flow problems and their numerical solution. The computational modelling of dilute polymeric fluids has also attracted the attention of researchers recently. Dilute polymeric fluids contain long chain molecules immersed in a Newtonian fluid. One of the most popular models for dilute polymeric fluids is based on a multiscale approach. On the macroscopic scale, an incompressible non-Newtonian fluid is governed by the Navier–Stokes equations with an additional stress tensor. On the microscopic scale, the simplest model used is the dumbbell model for approximating a polymer molecule, which results in the Fokker–Planck equation modelling the evolution of the probability density function associated with the ran-

dom configuration vector of the polymer molecule. Then, the additional polymeric stress tensor appearing in the Navier–Stokes equations is defined by the Kramers expression [47]. The full model for dilute polymeric fluids is therefore the coupled Navier–Stokes–Fokker–Planck system, which is often seen as a macro-micro system [28].

The difficulties in computing solutions to this model arise from the microscopic scale. The polymer molecules in the fluid are represented by two beads of a dumbbell connected with a massless spring according to some spring force law. Then, the Fokker–Planck equation is used to describe the evolution of the probability density function of the configuration of the dumbbells in the fluid. The equation involves the time variable $t \in [0, T]$, the position variable $\vec{x}$ in the physical flow domain and the configuration variable $\vec{q}$, which leads to a seven dimensional partial differential equation. More generally, when a polymer chain is modelled by $K+1$ beads which are linearly connected with K massless springs, the number of independent variables is potentially very large. Due to the curse of dimensionality, one needs special numerical techniques to solve this problem.

An alternative approach to overcoming the curse of dimensionality is to rewrite the Fokker–Planck equation as a system of stochastic differential equations, and use the Monte Carlo method to calculate the extra stress tensor. One drawback of this approach is that the Monte Carlo method introduces a stochastic error with slow decay ($O(N^{-1/2})$ as $N \to \infty$), N being the number of simulations. We will focus on this stochastic approach in this thesis and replace the Monte Carlo simulation by a recently developed multilevel Monte Carlo path simulation algorithm, which has been proved to reduce the cost significantly compared to the standard Monte Carlo method [23]. The multilevel Monte Carlo method reduces the total cost from $O(\epsilon^{-3})$ to $O(\epsilon^{-2}(\log \epsilon)^2)$ under suitable conditions, where ϵ^2 is the mean square error.

In this thesis, we investigate a particular non-Newtonian fluid problem: the simulation of dilute polymeric fluids using a multiscale model. We shall review some existing numerical methods for this multiscale model of dilute polymeric fluid and investigate the improved scheme for this problem.

The thesis is organized as follows. In Chapter 1, we will discuss the background and the mathematical modelling of dilute polymeric fluids; then both the deterministic and stochastic methods discussed above will be reviewed. We will then investigate the equilibrium solutions of the Navier–Stokes–Fokker–Planck equations. After that, we will present the multilevel Monte Carlo technique which will be used in the improved numerical scheme based on the stochastic approach. Chapter 4 will describe some numerical experiments. We will investigate a homogeneous planar extensional flow, which has an analytical solution. Then we will consider a 2D non-homogeneous flow for which there are some published results with which we can compare our own computations. Moreover we will perform multi-bead simulations. Finally we will draw some conclusions in Chapter 5.

1.2 Macroscopic Equation

The classical model for describing the dynamics of an incompressible fluid is well established. From the macroscopic point of view, the fluid is considered as a continuum, hence the fundamental laws of classical mechanics can be applied to represent its behaviour. The conservation of mass and linear momentum are used to derive the Navier–Stokes equations (for the full details, see [1], [7]). In this section, we consider the basic laws of macroscopic Newtonian fluids and derive the Navier–Stokes equations. Our overview is based on the books of Elman, Silvester and Wathen [15] and Owens and Phillips [47].

1.2.1 Conservation of Mass

Consider a fluid moving in a three-dimensional domain Ω with density $\rho : \mathbb{R}^3 \times [0, t_{\max}] \to \mathbb{R}$. Suppose that the volume of a fluid element at time $t \in [0, t_{\max}]$ is $V(t)$ with a boundary point $x(t)$ moving at fluid velocity $u(x, t)$; then, the total mass at time t is $m(t) = \int_{V(t)} \rho \, \mathrm{d}\Omega$, where $\mathrm{d}\Omega = \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z$ in three-dimensional space.

Theorem 1.1 Reynolds Transport Theorem. [Theorem in [47]] *Let $f(\vec{x},t)$ be a smooth scalar- or vector-valued function defined over a volume $V(t)$. Then,*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} f\,\mathrm{d}V = \int_{V(t)} \frac{\partial f}{\partial t}\,\mathrm{d}V + \int_{\partial V(t)} f\vec{u}\cdot\vec{n}\,\mathrm{d}A,$$

where $\vec{n}$ is the outward-pointing unit normal vector on the surface of $V(t)$ and $\vec{u}$ is the velocity of the fluid.

The law of conservation of mass states that the total mass of the system does not change with time. We express this mathematically as

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho\,\mathrm{d}\Omega = 0. \tag{1.1}$$

We apply the Reynolds Transport Theorem and the Divergence Theorem in (1.1) to obtain

$$\int_{V(t)} \left(\frac{\partial \rho}{\partial t} + \nabla_x\cdot(\rho\vec{u})\right)\mathrm{d}\Omega = 0. \tag{1.2}$$

Here we have put a subscript x on the differential operators to indicate that these are spatial derivatives. Since $V(t)$ is arbitrary, we can ignore the integration to deduce that

$$\frac{\partial \rho}{\partial t} + \nabla_x\cdot(\rho\vec{u}) = 0 \quad \text{in } \Omega. \tag{1.3}$$

If the density ρ is assumed to be constant as will be the case throughout this thesis, the equation (1.3) can be simplified as:

$$\nabla_x\cdot\vec{u} = 0. \quad \text{in } \Omega \tag{1.4}$$

1.2.2 Conservation of Momentum

The momentum of the fluid is the product of mass and velocity of the fluid. We can calculate the acceleration of a particle at position $\vec{x}(t)\in\Omega$ in the following way:

$$\vec{a}(\vec{x}(t),t) = \frac{\mathrm{d}}{\mathrm{d}t}\vec{u}(\vec{x}(t),t) = \frac{\partial\vec{u}}{\partial t} + \frac{\partial\vec{u}}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial\vec{u}}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial\vec{u}}{\partial z}\frac{\partial z}{\partial t} = \frac{\partial\vec{u}}{\partial t} + (\vec{u}\cdot\nabla_x)\vec{u}. \tag{1.5}$$

In fluid mechanics, this derivative is called material derivative.

Definition 1.1 Material Derivative. *The material derivative operator describes the time-dependent change of a scalar- or vector-valued function that evolves along a*

fluid field with velocity $\vec{u}$. We define the operator as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + (\vec{u} \cdot \nabla_x).$$

Then we can express the rate of change of momentum of the fluid in the case of constant density, ρ, as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_V \rho\, \vec{u}(\vec{x}(t), t) \mathrm{d}\Omega = \int_V \rho \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla_x)\vec{u} \right) \mathrm{d}\Omega. \tag{1.6}$$

Newton's second law of motion requires that the rate of change of momentum of the fluid is equal to the sum of external forces on the fluid. There are two kinds of force acting on the fluid. One is body force, such as gravity. The other are surface forces, which act by contact with the surface. For an ideal incompressible flow without viscosity, the only surface force is due to the pressure. Thus the total force is

$$\int_V \rho \vec{f}\, \mathrm{d}\Omega + \int_{\partial V} \boldsymbol{\sigma} \cdot \vec{n}\, \mathrm{d}A, \tag{1.7}$$

where $\vec{f}$ is the body force per unit mass, $\boldsymbol{\sigma}$ is the stress tensor and $\mathrm{d}A$ is the surface element. Now conservation of momentum yields

$$\int_V \rho \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla_x)\vec{u} \right) \mathrm{d}\Omega = \int_V \rho \vec{f}\, \mathrm{d}\Omega + \int_V \nabla_x \cdot \boldsymbol{\sigma}\, \mathrm{d}\Omega. \tag{1.8}$$

For the second term on the right-hand side, we used the Divergence Theorem. Since the equation holds for any volume V, equation (1.8) can be simplified as

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla_x)\vec{u} = \vec{f} + \frac{1}{\rho} \nabla_x \cdot \boldsymbol{\sigma} \quad \text{in } \Omega. \tag{1.9}$$

In (1.9) the stress tensor is still unknown. Different types of fluid flow have different $\boldsymbol{\sigma}$ and this results in specific forms of the Navier–Stokes equations.

1.2.3 Stress Tensor

We can expand the stress tensor into three parts

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau_s} + \boldsymbol{\tau}, \tag{1.10}$$

where p is the pressure, $\boldsymbol{\tau_s}$ represents the solvent contribution to the stress and $\boldsymbol{\tau}$ is the extra stress for non-Newtonian fluids. In the case of Newtonian fluids, we have $\boldsymbol{\tau} = 0$ and Newton's constitutive law states that the shear stress tensor is a linear function of the rate of strain tensor

$$\boldsymbol{\tau_s} = \frac{\mu_s}{2}(\nabla_x \vec{u} + (\nabla_x \vec{u})^T), \tag{1.11}$$

where μ_s is the molecular viscosity. Combining (1.10), (1.11) with (1.9), in the case of an incompressible Newtonian fluid we obtain the Navier–Stokes equations:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla_x)\vec{u} = \vec{f} - \frac{1}{\rho}\nabla_x p + \nu \nabla_x^2 \vec{u} \quad \text{in } \Omega, \tag{1.12}$$

$$\nabla_x \cdot \vec{u} = 0, \quad \text{in } \Omega, \tag{1.13}$$

where $\nu := \mu_s/\rho$ is called the kinematic solvent viscosity.

Polymeric fluids are non-Newtonian fluids. The polymer molecules will contribute an extra stress $\boldsymbol{\tau}(\vec{x}, t)$ to the total stress tensor while the equations for the conservation of momentum and mass still remain the same, hence the model problem takes the following form:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla_x)\vec{u} = -\frac{1}{\rho}\nabla_x p + \nu \nabla_x^2 \vec{u} + \vec{f} + \nabla_x \cdot \boldsymbol{\tau} \quad \text{in } \Omega, \tag{1.14}$$

$$\nabla_x \cdot \vec{u} = 0 \quad \text{in } \Omega. \tag{1.15}$$

The system models polymeric flow in a macroscopic sense, and the contribution of microscopic polymeric molecules is reflected by the polymeric extra stress $\boldsymbol{\tau}$.

In the next section, we will consider the microscopic scale to build a model for polymer molecules, which will then allow us to compute the polymeric extra stress tensor $\boldsymbol{\tau}$.

1.3 Microscopic Equations

The dynamics of polymeric fluids is frequently modelled with the aid of bead-spring chain and bead-rod chain models to capture the properties of the polymer molecules. These models only include descriptions of the most basic characteristics of the polymer

molecules in which the beads model monomer units and the springs or rods represent the internal forces in the molecules. They are the most frequently used models in practice but true polymer molecule motions are of course much more complicated [51]. Among these models, the simplest one is the dumbbell model, which only considers two beads connected with a massless elastic spring; then, only one configuration variable (the end-to-end vector) $\vec{q}$ is needed, which will reduce the number of internal degrees of freedom.

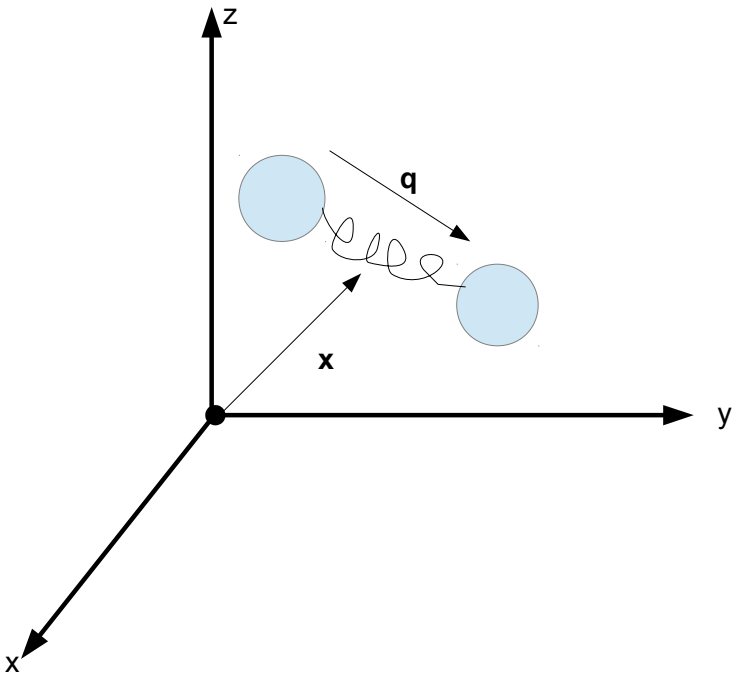


Figure 1.1: Illustration of the dumbbell model

Figure 1.1 illustrates the details of the dumbbell model. In a 3D flow domain, the state of a dumbbell is described by a position vector of the centre of mass $\vec{x} \in \mathbb{R}^3$ and a configuration vector (or end-to-end vector) $\vec{q} \in \mathbb{R}^3$. This end-to-end vector

is sufficient to describe the dynamical behaviour, including the stretching and the orientation of the dumbbell.

A spring force law must also be defined to complete the model. The simplest is Hooke's law, which states that the force needed to stretch a spring by some distance is proportional to that distance. Mathematically, the force of a Hookean spring can be represented as:

$$\vec{F}(\vec{q}) = H\vec{q}, \tag{1.16}$$

where H is the spring constant. However, this model is not realistic because the spring can not be infinitely stretched in reality. Therefore it is necessary to impose a nonlinear force law, which controls the maximum extension of the dumbbell. A remedy is to consider the FENE (Finitely Extensible Nonlinear Elastic) model, which is given by:

$$\vec{F}(\vec{q}) = \frac{H\vec{q}}{1 - \frac{|\vec{q}|^2}{b}}, \tag{1.17}$$

where $b = |\vec{q}_{max}|^2$ indicates the maximum extension to which the dumbbell can be stretched. The spring force $\vec{F}$ becomes unbounded as $|\vec{q}|^2 \to b$ and consequently the dumbbell can be only extended to a finite length. Therefore the configuration space D becomes a ball centred at the origin with radius $\sqrt{b}$. Note also that $\vec{F}(-\vec{q}) = -\vec{F}(\vec{q})$ in agreement with Newton's third law.

After choosing the dumbbell model for the polymeric fluid, we can formulate the Fokker–Planck equation to describe the evolution of the probability density function (denoted by $\psi = \psi(\vec{x}, \vec{q}, t)$) of the dumbbell configuration $\vec{q}$. The associated dimensionless Fokker–Planck equation has the following form:

$$\frac{\partial \psi}{\partial t} + \nabla_x \cdot (\vec{u}\psi) + \nabla_q \cdot ((\nabla_x \vec{u})\vec{q}\psi) = \epsilon \Delta_x \psi + \frac{1}{2\mathrm{Wi}}(\nabla_q \cdot (\nabla_q \psi + \vec{F}(\vec{q})\psi)), \tag{1.18}$$

where ϵ is a small positive constant of the order 10^{-8} according to [8] and Wi is the non-dimensional Weissenberg number with the order of 1 [28]. Here $\nabla_x \vec{u}$ is a matrix representing the velocity gradient, i.e., $(\nabla_x \vec{u})_{ij} = \partial_j u_i$ and $(\nabla_x \vec{u})\vec{q}$ represents the multiplication of the matrix and the vector $\vec{q}$. In most of the literature, the term $\epsilon \Delta_x \psi$ has been ignored because of the small coefficient ϵ.

In the case of a multi-bead spring model, the model involves $K+1$ beads connected with K elastic springs to represent a polymer chain, hence the configuration vector becomes $\vec{q} := (\vec{q_1}^T, \ldots, \vec{q_K}^T)^T \in D :=$ Cartesian product of D_i with $\vec{q_i}$ representing the conformation and orientation of the ith spring and the Fokker–Planck equation takes the form

$$\frac{\partial \psi}{\partial t} + \nabla_x \cdot (\vec{u}\psi) + \sum_{i=1}^{K} \nabla_{q_i} \cdot ((\nabla_x \vec{u})\vec{q_i}\psi) = \epsilon \Delta_x \psi + \frac{1}{2\mathrm{Wi}} \sum_{i=1}^{K} \sum_{j=1}^{K} A_{ij} (\nabla_{q_i} \cdot (\nabla_{q_j} \psi + \vec{F_j}(\vec{q_j})\psi)), \tag{1.19}$$

where $\mathbf{A} = (A_{ij})$ is a symmetric positive definite matrix with smallest eigenvalue $a_0 \in \mathbb{R}_{>0}$, and D_i, $i = 1, \ldots, K$, are bounded balls centred at the origin (cf. Figure 1.2 on page 30).

1.3.1 Derivation of the Fokker–Planck Equation

We shall derive the Fokker–Planck equation (1.18) from first principles. The details of the derivation can be found in the paper by Barrett & Süli [6].

Consider a bead-spring representation of a polymer molecule with $K + 1$ beads that are joined by K massless elastic springs in a Newtonian solvent with velocity given by $\vec{u}(\vec{x}, t)$. Each bead has mass m and its position at time t is determined by the vector $\vec{x}_i(t)$ for $i = 1, \ldots, K + 1$. The centre of mass of the system is $\vec{x} = \frac{1}{K+1} \sum_{i=1}^{K+1} \vec{x}_i$, and the end-to-end vector of the ith spring is $\vec{q_i} = \vec{x}_{i+1} - \vec{x}_i, i = 1, \ldots, K$. The state of the molecule can be fully described by these vectors.

There are three kinds of force acting on the ith bead of the bead-spring chain in the fluid. One is the spring force $\vec{F_i}(\vec{q_i}, t)$. The second is the Brownian force $\vec{B_i}(t)$ due to random collisions of solvent molecules with bead i. The third is the drag force $\vec{F}_i^{\mathrm{drag}}(t)$ due to bead i moving through the viscous solvent.

Notice that the ith bead can be pulled by either the ith spring in the $\vec{q_i}$ direction or the $(i-1)$st spring in the $\vec{q}_{i-1}$ direction; therefore we define a $(K+1) \times K$ graph incidence matrix $\mathbf{G}$ given by

$$\mathbf{G}= \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & -1 \end{pmatrix}.$$

Then the total spring force acting on the ith bead is:

$$\sum_{j=1}^{K} G_{i,j}\vec{F}_j(\vec{q}_j,t), \quad \text{for } i=1,\ldots,K+1. \tag{1.20}$$

The Brownian force $\vec{B}_i(t)$ is defined as:

$$\vec{B}_i(t)\mathrm{d}t = \sqrt{2k_BT\zeta}\,\mathrm{d}\vec{W}_i(t), \tag{1.21}$$

where $\vec{W}_i(t)$ is a vector of $K+1$ independent Wiener processes, the constant factor $k_B = 1.38\times10^{-23}m^2kgs^{-2}K^{-1}$ is Boltzmann's constant and T is the absolute temperature measured in Kelvin, K, ζ is a characteristic drag coefficient, which is fixed to be a positive number.

The drag force $\vec{F}_i^{\text{drag}}(t)$ is modelled by the Stokes law for viscous drag on a sphere at low Reynolds number, defined by

$$\vec{F}_i^{\text{drag}}(t) = \zeta\left(\vec{u}(\vec{x}_i,t) - \frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t}(t)\right), \tag{1.22}$$

which is proportional to the relative velocity of the ith bead to the fluid.

Applying Newton's second law to the ith bead, we obtain the following:

$$m\vec{a}_i(t) = \sum_{j=1}^{K} G_{i,j}\vec{F}_j(\vec{q}_j,t) + \vec{B}_i(t) + \vec{F}_i^{\text{drag}}(t), \quad i=1,\ldots,K+1, \tag{1.23}$$

where $\vec{a}_i(t)$ is the acceleration of the ith bead and m is its mass. In [53], Schieber and Öttinger give reasons why the mass on the left-hand side can be formally neglected. For a rigorous proof, we refer to Süli & Yahiaoui (2018) [55]. From the physical point of view, the justification for this simplification is that the beads are assumed to be massless in the model. By considering the zero-mass limit for the beads and formally multiplying by $\mathrm{d}t$ on both sides of the equation, we obtain:

$$\zeta\left(\mathrm{d}\vec{x}_i(t) - \vec{u}(\vec{x}_i,t)\,\mathrm{d}t\right) = \sum_{j=1}^{K} G_{i,j}\vec{F}_j(\vec{q}_j,t)\,\mathrm{d}t + \sqrt{2k_BT\zeta}\,\mathrm{d}\vec{W}_i(t),\ i=1,\ldots,K+1. \tag{1.24}$$

These equations are referred to as Langevin's equations. We can rearrange them to get the following system of stochastic differential equations:

$$\mathrm{d}\vec{x}_i(t) = \left(\vec{u}(\vec{x}_i,t) + \zeta^{-1}\sum_{j=1}^{K} G_{i,j}\vec{F}_j(\vec{x}_{j+1}-\vec{x}_j,t)\right)\mathrm{d}t + \sqrt{\frac{2k_BT}{\zeta}}\mathrm{d}\vec{W}_i(t),\ i=1,\ldots,K+1. \tag{1.25}$$

Defining

$$\vec{X}(t) = (\vec{x}_1(t)^T, \vec{x}_2(t)^T, \ldots, \vec{x}_{K+1}(t)^T)^T, \tag{1.26}$$

$$\vec{b}(\vec{X}(t)) = (\vec{u}(\vec{x}_1,t) + \zeta^{-1}\sum_{j=1}^{K} G_{1,j}\vec{F}_j(\vec{x}_{j+1}-\vec{x}_j,t),\ldots)^T, \tag{1.27}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{2k_BT}{\zeta}}\mathbf{I}, \tag{1.28}$$

the equations can be rewritten as the following stochastic differential equation for $\vec{X}(t)$:

$$\mathrm{d}\vec{X}(t) = \vec{b}(\vec{X}(t))\mathrm{d}t + \boldsymbol{\sigma}(\vec{X}(t))\mathrm{d}\vec{W}(t). \tag{1.29}$$

We can now use the associated forward Kolmogorov equation to derive the equation (1.18) for the dumbbell model when $K = 1$.

Theorem 1.2 Forward Kolmogorov (Fokker–Planck) equation. [Theorem in [3]] *Let the random variable $\vec{X}(t)$, which evolves according to (1.29), have probability density function $(\vec{z},t) \mapsto \psi(\vec{z},t)$ of class $C^{2,1}(\mathbb{R}^{2d}\times[0,\infty))$ (i.e., twice continuously differentiable with respect to $\vec{z}\in\mathbb{R}^{2d}$ and once with respect to t), and let $\vec{X}(0) = \vec{X}$ be a square-integrable random variable with density function $\psi_0 \in C^2(\mathbb{R}^{2d})$. Also, suppose that $\vec{b}$ and $\boldsymbol{\sigma}$ in the stochastic differential equation are globally Lipschitz continuous, and $\boldsymbol{a}(\vec{z}) = \boldsymbol{\sigma}(\vec{z})\boldsymbol{\sigma}(\vec{z})^T$. Then,*

$$\frac{\partial\psi}{\partial t} + \sum_{j=1}^{2d}\frac{\partial}{\partial z_j}(b_j\psi) = \frac{1}{2}\sum_{i,j=1}^{2d}\frac{\partial^2}{\partial z_i\partial z_j}(a_{ij}\psi), \tag{1.30}$$

in $\mathbb{R}^{2d}\times[0,\infty)$ where $\psi(\vec{z},0) = \psi_0(\vec{z})$ for $\vec{z}\in\mathbb{R}^d$.

For the proof of this theorem we refer to [3].

Remark 1.1 *In the theorem, it is assumed that the drift term $\vec{b}$ is globally Lipschitz continuous. In our model the Hookean spring force is globally Lipschitz continuous, whereas the FENE spring force does not satisfy this assumption. Indeed, the FENE force is only locally Lipschitz continuous on D and is not defined on all of $\mathbb{R}^d$. Nevertheless, we shall proceed as if Theorem 1.2 held in the FENE case also.*

The bead-spring chain model is simplified to the dumbbell model when $K = 1$ (single bond). Therefore

$$\vec{X}(t) = (\vec{x}_1(t)^T, \vec{x}_2(t)^T)^T,$$

$$\vec{b}(\vec{X}(t)) = \begin{pmatrix} \vec{u}(\vec{x}_1, t) + \zeta^{-1}\vec{F}(\vec{x}_2(t) - \vec{x}_1(t)) \\ \vec{u}(\vec{x}_2, t) + \zeta^{-1}\vec{F}(\vec{x}_1(t) - \vec{x}_2(t)) \end{pmatrix},$$

$$\mathbf{a}(\vec{X}(t)) = \frac{2k_BT}{\zeta}\mathbf{I}.$$

Applying Theorem 1.2 to the stochastic differential equation we obtain:

$$\begin{aligned} \frac{\partial \psi^{12}}{\partial t} &+ \nabla_{x_1} \cdot \left(\vec{u}(\vec{x}_1, t)\psi^{12} + \zeta^{-1}\vec{F}(\vec{x}_2(t) - \vec{x}_1(t))\psi^{12} \right) \\ &+ \nabla_{x_2} \cdot \left(\vec{u}(\vec{x}_2, t)\psi^{12} + \zeta^{-1}\vec{F}(\vec{x}_1(t) - \vec{x}_2(t))\psi^{12} \right) \\ &= \frac{k_BT}{\zeta}\Delta_{x_1}\psi^{12} + \frac{k_BT}{\zeta}\Delta_{x_2}\psi^{12}, \end{aligned} \tag{1.31}$$

where $\psi^{12}(\vec{x}_1, \vec{x}_2, t)$ is the probability density function of the random variables $\vec{x}_1, \vec{x}_2$. The probability density function $\psi(\vec{x}, \vec{q}, t)$ in the Fokker–Planck equation (1.18) depends on the position of the centre of mass $\vec{x}$ and the configuration vector $\vec{q}$. Therefore we perform the linear change of variables: $\vec{x} = \frac{\vec{x}_1 + \vec{x}_2}{2}$, $\vec{q} = \vec{x}_2 - \vec{x}_1$. By the chain rule, we have:

$$\begin{aligned} \frac{\partial}{\partial \vec{x}_1} &= \frac{1}{2}\frac{\partial}{\partial \vec{x}} - \frac{\partial}{\partial \vec{q}}, \\ \frac{\partial}{\partial \vec{x}_2} &= \frac{1}{2}\frac{\partial}{\partial \vec{x}} + \frac{\partial}{\partial \vec{q}}, \\ \frac{\partial^2}{\partial \vec{x}_1^2} &= \frac{1}{4}\frac{\partial^2}{\partial \vec{x}^2} - \frac{\partial^2}{\partial \vec{x}\partial \vec{q}} + \frac{\partial^2}{\partial \vec{q}^2}, \end{aligned}$$

$$\frac{\partial^2}{\partial \vec{x}_2^2} = \frac{1}{4}\frac{\partial^2}{\partial \vec{x}^2} + \frac{\partial^2}{\partial \vec{x}\partial \vec{q}} + \frac{\partial^2}{\partial \vec{q}^2}.$$

Letting $\psi(x, \vec{q}, t) = \psi^{12}(\vec{x}_1, \vec{x}_2, t)$, we obtain

$$\begin{aligned}
&\frac{\partial \psi}{\partial t} + \nabla_q \cdot \left((\vec{u}(\vec{x} + \frac{\vec{q}}{2}, t) - \vec{u}(\vec{x} - \frac{\vec{q}}{2}, t))\psi - \frac{2}{\zeta}\vec{F}(\vec{q})\psi \right) \\
&\quad + \nabla_x \cdot \left(\frac{\vec{u}(\vec{x} - \frac{\vec{q}}{2}, t) + \vec{u}(\vec{x} + \frac{\vec{q}}{2}, t)}{2}\psi \right) \\
&\quad = \frac{k_B T}{2\zeta}\Delta_x \psi + \frac{2k_B T}{\zeta}\Delta_q \psi, \qquad (1.32)
\end{aligned}$$

where we have used the fact that $\vec{F}(\vec{q}) = -\vec{F}(-\vec{q})$.

Using the approximations $\frac{\vec{u}(\vec{x}_1,t)+\vec{u}(\vec{x}_2,t)}{2} \approx \vec{u}(\vec{x}, t)$ and the Taylor expansion:

$$\vec{u}(\vec{x} + \frac{\vec{q}}{2}, t) = \vec{u}(\vec{x}, t) + \nabla_x \vec{u}(\vec{x}, t)\frac{\vec{q}}{2} + \cdots, \qquad (1.33)$$

$$\vec{u}(\vec{x} - \frac{\vec{q}}{2}, t) = \vec{u}(\vec{x}, t) - \nabla_x \vec{u}(\vec{x}, t)\frac{\vec{q}}{2} + \cdots, \qquad (1.34)$$

yields, with the approximate equality replaced by equality:

$$\frac{\partial \psi}{\partial t} + \nabla_x \cdot (\vec{u}\psi) + \nabla_q \cdot \left((\nabla_x \vec{u})\vec{q}\psi - \frac{2}{\zeta}\vec{F}(\vec{q})\psi \right) = \frac{k_B T}{2\zeta}\Delta_x \psi + \frac{2k_B T}{\zeta}\Delta_q \psi. \qquad (1.35)$$

In these equations the expression $(\nabla_x \vec{u})\vec{q}$ is the product of the 3×3 matrix function $\nabla_x \vec{u}$ and the 3-component column vector $\vec{q}$. Notice that the first approximation will be exact if $\vec{u}$ is linear in its spatial variable.

The last step is to put the equation into an appropriate nondimensionalised form. Applying

$$x = L_0 \hat{x}, \quad q = l_0 \hat{q}, \quad u = U_0 \hat{u}, \quad t = \frac{L_0}{U_0}\hat{t} \qquad (1.36)$$

to the equation and multiplying it through by L_0/U_0 gives us the Fokker–Planck equation (with the hats over the symbols omitted for the sake of simplicity of the notation)

$$\frac{\partial \psi}{\partial t} + \nabla_x \cdot (\vec{u}\psi) + \nabla_q \cdot ((\nabla_x \vec{u})\vec{q}\psi - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{q})\psi) = \frac{1}{8\mathrm{Wi}}\left(\frac{l_0}{L_0}\right)^2 \Delta_x \psi + \frac{1}{2\mathrm{Wi}}\Delta_q \psi, \quad (1.37)$$

where $l_0 = \sqrt{k_B T/H}$, is a typical microscopic length scale (of between tens to a few hundred nanometers), and H is the spring constant, $\mathrm{Wi} = \frac{\zeta U_0}{4HL_0}$ is the non-dimensional Weissenberg number and L_0, U_0 are the characteristic length and velocity

of the macroscopic flow. The Weissenberg number is the ratio between the microscopic relaxation time $\frac{\zeta}{4H}$ and the characteristic fluid process time $\frac{L_0}{U_0}$ and it can be regarded as the ratio of the microscopic to macroscopic time-scale. This parameter plays an important role in the numerical experiment and reader can refer to Section 4.4.3. Wi is typically of order 1, while $\frac{l_0}{L_0}$ is typically of size 10^{-11} to 10^{-9}.

In this thesis, we will focus on the FENE spring force, hence $\vec{F}(\vec{q})$ in equation (1.18) takes the form of (1.17). The function $\psi(\vec{x}, \vec{q}, t)$ is the probability of finding a dumbbell with centre of mass at $\vec{x}$ and its configuration given by $\vec{q}$ at time t. The Fokker–Planck equation is typically high-dimensional (since ψ is a function of $3(K+1)+1$ independent variables, where K is the number of springs in the bead-spring chain) and most standard numerical algorithms for its solutions become inefficient due to the exponential growth of the computational cost with the number of dimensions. Moreover, this equation contains an unbounded nonlinear term $\vec{F}(\vec{q})$, which makes its solution even more challenging. In next section, we will combine these equations to complete the description of the multiscale model.

1.4 The Coupled Micro-Macro System

In Section 1.2, we introduced the extra stress tensor $\boldsymbol{\tau}$ in the Navier–Stokes equations (1.14), which will serve as a source term. It is also the link between the Navier–Stokes equation and the Fokker–Planck equation. Having computed the probability density function $\psi(\vec{x}, \vec{q}, t)$, the polymeric extra stress tensor is given by what is known as the Kramers expression [46]:

$$\boldsymbol{\tau}(\vec{x}, t) := n_p(\langle \vec{q} \otimes \vec{F}(\vec{q}) \rangle - \mathbf{I}) = n_p \left(\int_D \vec{q} \otimes \vec{F}(\vec{q}) \psi(\vec{x}, \vec{q}, t) \mathrm{d}\vec{q} - \mathbf{I} \right), \tag{1.38}$$

where n_p is a polymer number density and $\mathbf{I}$ is the identity matrix, $\otimes$ represents the tensor product of two vectors. The derivation of the Kramers expression (1.38) can be found in Bird [9] and Li [40]. Since $\boldsymbol{\tau}$ appears in the Navier–Stokes equations (1.14) by its divergence, the constant $n_p\mathbf{I}$ in (1.38) makes no contribution to the coupled system and therefore we could omit this factor in the simulation. In an analogous manner

as for the Fokker–Planck equation, we non-dimensionalise the Kramers expression according to (1.36),

$$\boldsymbol{\tau}(\vec{x},t) = n_p k_B T \left(\int_D \vec{q} \otimes \vec{F}(\vec{q}) \psi(\vec{x},\vec{q},t)\, \mathrm{d}\vec{q} - \mathbf{I} \right). \tag{1.39}$$

It can be shown that [52] for dilute FENE dumbbell solutions in a shear flow, the (1,2) component of $\boldsymbol{\tau}$ can be approximated by

$$\tau_{12} \approx \dot{\gamma} \frac{\mathrm{Wi} L_0}{U_0} n_p k_B T \left(\frac{b}{b+d+2} \right), \tag{1.40}$$

where $\dot{\gamma}$ is the shear rate and d is the dimension of the flow, $d \in \{2,3\}$. Based on this, we can define the polymeric viscosity μ_p for the FENE dumbbell as,

$$\mu_p = \frac{\mathrm{Wi} L_0}{U_0} n_p k_B T \left(\frac{b}{b+d+2} \right). \tag{1.41}$$

Inserting (1.41) into (1.39) we obtain

$$\boldsymbol{\tau}(\vec{x},t) = \frac{U_0}{L_0} \frac{\mu_p}{\mathrm{Wi}} \left(\frac{b+d+2}{b} \right) \left(\int_D \vec{q} \otimes \vec{F}(\vec{q}) \psi(\vec{x},\vec{q},t) \mathrm{d}\vec{q} - \mathbf{I} \right). \tag{1.42}$$

We also need to non-dimensionalise the Navier–Stokes equations (1.14) in the same manner as the polymeric stress tensor according to (1.36) and multiply it through by L_0/U_0, which yields the following equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla_x \vec{u} = -\nabla_x p + \frac{\gamma}{\mathrm{Re}} \Delta_x \vec{u} + \frac{b+d+2}{b} \frac{1-\gamma}{\mathrm{ReWi}} \nabla_x \cdot \boldsymbol{\tau} + \vec{f}, \tag{1.43}$$

$$\nabla_x \cdot \vec{u} = 0, \tag{1.44}$$

where $\mathrm{Re} = L_0 U_0/(\nu + \mu_p)$ is the Reynolds number and $\gamma = \nu/(\nu + \mu_p)$. Note that we absorbed the coefficients into the source term $\vec{f}$ and the pressure p. Assuming that the physical domain is denoted by $\Omega \subset \mathbb{R}^d$ and the configuration space is $D = B(0, \sqrt{b}) \subset \mathbb{R}^d$ and combining all the equations subject to appropriate initial and boundary conditions, we obtain the following system for $\vec{u} = \vec{u}(\vec{x},t), p = p(\vec{x},t)$ and $\psi = \psi(\vec{x},\vec{q},t)$:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla_x \vec{u} = -\nabla_x p + \frac{\gamma}{\mathrm{Re}} \Delta_x \vec{u} + \nabla_x \cdot \boldsymbol{\tau} + \vec{f}, \tag{1.45}$$

$$\nabla_x \cdot \vec{u} = 0, \tag{1.46}$$

$$\frac{\partial \psi}{\partial t} + \nabla_x \cdot (\vec{u}\psi) + \nabla_q \cdot ((\nabla_x \vec{u})\vec{q}\psi) = \frac{1}{2\mathrm{Wi}} (\nabla_q \cdot (\nabla_q \psi + \vec{F}(\vec{q})\psi)), \tag{1.47}$$

$$\boldsymbol{\tau}(\vec{x}, t) = \frac{b+d+2}{b} \frac{1-\gamma}{\mathrm{ReWi}} \left(\int_D \vec{q} \otimes \vec{F}(\vec{q}) \psi(\vec{x}, \vec{q}, t) \mathrm{d}\vec{q} - \mathbf{I} \right), \tag{1.48}$$

$$\vec{u}(\vec{x}, 0) = \vec{u}_0(\vec{x}), \quad \psi(\vec{x}, \vec{q}, 0) = \psi_0(\vec{q}), \tag{1.49}$$

where $\vec{x} \in \Omega$, $\vec{q} \in D$ and $t \in [0, T]$. Equations (1.45)–(1.49) are the coupled micro-macro system for modelling the dilute polymeric fluid. In the next section, we shall review some existing numerical methods for this problem.

Remark 1.2 *As indicated before, in most of the literature, the term $\epsilon \Delta_x \psi$ is neglected because the coefficient ϵ is typically on the order of 10^{-8} according to [8]; therefore, from now on we consider the Fokker–Planck equation (1.47) without the $\vec{x}$ diffusion term as our model problem.*

1.5 Review of Methods

There are many methods which have been developed over several decades for the numerical simulation of polymeric fluids. These techniques can be divided into three categories: fully deterministic multiscale methods, combined stochastic-deterministic multiscale methods and purely macroscopic methods.

1.5.1 Purely Macroscopic Methods

Originally researchers used continuum mechanics to study the flow and developed macroscopic methods. Purely macroscopic methods model non-Newtonian fluids by ignoring the effect of the microscopic length scale, which can save huge amounts of computational effort; therefore numerical methods for macroscopic models have been widely used in practice since the 1970's. It can be shown that certain macroscopic models are equivalent to multiscale models in some cases; for example, a microscopic

dumbbell model with a Hookean spring [4] is formally equivalent to the Oldroyd-B model. However this model is not sufficient to capture minute features of complex polymeric flows. We consider one of the most famous models (Oldroyd-B model) here; the details can be found in [45], [47].

Definition 1.2 Upper convected derivative. *For an arbitrary tensor $\boldsymbol{C}$, we define the upper convected derivative as*

$$\frac{\delta \boldsymbol{C}}{\delta t} = \frac{\partial \boldsymbol{C}}{\partial t} + (\vec{u} \cdot \nabla_x)\boldsymbol{C} - \left((\nabla_x \vec{u})\boldsymbol{C} + \boldsymbol{C}(\nabla_x \vec{u})^T\right). \tag{1.50}$$

Here $\nabla_x \vec{u}$ denotes the 3×3 gradient matrix of the 3-component vector function $\vec{u}$.

The Oldroyd-B model combines solvent and polymeric contributions to the stress. Using equation (1.11) for the solvent stress and applying the upper convected derivative (1.50) we obtain the constitutive equation for the stress tensor

$$\boldsymbol{\sigma} = -p\mathbf{I} + \frac{\eta_s}{2}(\nabla_x \vec{u} + (\nabla_x \vec{u})^T) + \boldsymbol{\tau}, \tag{1.51}$$

$$\boldsymbol{\tau} + \mathrm{Wi}\frac{\delta \boldsymbol{\tau}}{\delta t} = \frac{\eta_p}{2}(\nabla_x \vec{u} + (\nabla_x \vec{u})^T), \tag{1.52}$$

where η_s and η_p stand for the corresponding viscosity for the solvent and polymer.

The Oldroyd-B model is a fully macroscopic model for functions in $\vec{x}$ and t and replaces the high dimensional (involving the probability density function of 3+3+1 variables) Fokker–Planck equation (1.47) by the equation (1.52) and therefore reduces the amount of computation. We shall show that this model is formally equivalent to a multiscale model involving a Hookean spring, assuming that ψ and $|\nabla_q \psi|$ decay to zero sufficiently fast with $|\vec{q}| \to \infty$. Multiplying (1.47) by $\vec{q} \otimes \vec{q} = \vec{q}\vec{q}^T$ and integrating over the configuration domain D, we have

$$\begin{aligned}\int_D \frac{\partial \psi}{\partial t}\vec{q}\vec{q}^T \mathrm{d}\vec{q} + (\vec{u} \cdot \nabla_x)\int_D \psi \vec{q}\vec{q}^T \mathrm{d}\vec{q} + \int_D \nabla_q \cdot ((\nabla_x \vec{u})\vec{q}\psi)\vec{q}\vec{q}^T \mathrm{d}\vec{q} \\ = \frac{1}{2\mathrm{Wi}}\int_D \nabla_q \cdot (\nabla_q \psi + \vec{F}(\vec{q})\psi)\vec{q}\vec{q}^T \mathrm{d}\vec{q}.\end{aligned} \tag{1.53}$$

Performing integration by parts yields

$$\int_D \frac{\partial \psi}{\partial t}\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} + (\vec{u}\cdot\nabla_x)\int_D \psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} - \int_D [(\nabla_x\vec{u})\vec{q}\psi\cdot\nabla_q]\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} = \frac{1}{2\mathrm{Wi}}\int_D [\Delta_q\psi - \vec{F}(\vec{q})\psi\cdot\nabla_q]\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q}. \tag{1.54}$$

The boundary terms have been dropped due to the assumption that ψ and $|\nabla_q\psi|$ decay to zero sufficiently fast with $|\vec{q}|\to\infty$. Noting for any vector $\vec{r}$ that

$$(\vec{r}\cdot\nabla_q)\vec{q}\,\vec{q}^{\,T} = \vec{r}\,\vec{q}^{\,T} + \vec{q}\,\vec{r}^{\,T}, \tag{1.55}$$

the equation (1.54) becomes upon noting that $\nabla_x\vec{u}$ is independent of $\vec{q}$ and taking $\vec{r} = \vec{q}\psi$,

$$\int_D \frac{\partial \psi}{\partial t}\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} + (\vec{u}\cdot\nabla_x)\int_D \psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} - \left[(\nabla_x\vec{u})\int_D \psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} + (\nabla_x\vec{u})^T\int_D \psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q}\right] = \frac{1}{2\mathrm{Wi}}\left[\int_D \Delta_q\psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} - \int_D \psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} - \int_D \psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q}\right], \tag{1.56}$$

where we have used the fact that the spring force $F(\vec{q}) = \vec{q}$ for the Hookean spring with the spring constant H set equal to 1 for ease of writing.

Defining $\mathbf{C} = \int_D \psi\vec{q}\vec{q}^{\,T}\mathrm{d}\vec{q}$ and substituting in (1.56) we obtain

$$\mathrm{Wi}\left(\frac{\partial \mathbf{C}}{\partial t} + (\vec{u}\cdot\nabla_x)\mathbf{C} - \left[(\nabla_x\vec{u})\mathbf{C} + \mathbf{C}(\nabla_x\vec{u})^T\right]\right) = \frac{1}{2}\int_D \Delta_q\psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q} - \mathbf{C}. \tag{1.57}$$

Using Definition 1.2 gives

$$\mathrm{Wi}\frac{\delta \mathbf{C}}{\delta t} + \mathbf{C} = \frac{1}{2}\int_D \Delta_q\psi\vec{q}\,\vec{q}^{\,T}\mathrm{d}\vec{q}. \tag{1.58}$$

Performing integration by parts twice on the right-hand side and using the fact $\Delta_q(\vec{q}\vec{q}^{\,T}) = 2\mathbf{I}$, we obtain

$$\mathrm{Wi}\frac{\delta \mathbf{C}}{\delta t} + \mathbf{C} = \left(\int_D \psi(\vec{x},\vec{q},t)\mathrm{d}\vec{q}\right)\mathbf{I}. \tag{1.59}$$

Observe also that ψ satisfies the following equation

$$\frac{\partial}{\partial t}\int_D \psi \,\mathrm{d}\vec{q} + (\vec{u}\cdot\nabla_x)\int_D \psi \,\mathrm{d}\vec{q} = 0. \tag{1.60}$$

This can be proved by integrating (1.47) over the configuration space D and applying the divergence theorem:

$$\begin{aligned}
&\frac{\partial}{\partial t}\int_D \psi \,\mathrm{d}\vec{q} + (\vec{u}\cdot\nabla_x)\int_D \psi \,\mathrm{d}\vec{q} \\
&= \int_D \nabla_q \cdot \Big(-(\nabla_x \vec{u})\vec{q}\psi + \nabla_q \psi + \vec{F}(\vec{q})\psi\Big)\,\mathrm{d}\vec{q} \\
&= \int_{\partial D} \Big(-(\nabla_x \vec{u})\vec{q}\psi + \nabla_q \psi + \vec{F}(\vec{q})\psi\Big)\cdot \vec{n_q}\,\mathrm{ds} = 0,
\end{aligned} \tag{1.61}$$

where $\vec{n_q}$ is the outward unit normal on the boundary of D and the boundary terms vanish due to the assumptions on ψ. Since ψ is a probability density function for each $\vec{x} \in \Omega$, the following normalisation property should also be satisfied:

$$\int_D \psi(\vec{x},\vec{q},t)\,\mathrm{d}\vec{q} = \int_D \psi(\vec{x},\vec{q},0)\,\mathrm{d}\vec{q} = 1. \tag{1.62}$$

Another observation is that in the case of the Hookean spring model, the Kramers expression for $\boldsymbol{\tau}$ can be written in terms of $\mathbf{C}$ defined above; more precisely,

$$\boldsymbol{\tau} = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{ReWi}}(\mathbf{C}-\mathbf{I}). \tag{1.63}$$

Combining (1.59), (1.60) (1.62) and (1.63), we find that $\boldsymbol{\tau}$ satisfies

$$\boldsymbol{\tau} + \mathrm{Wi}\frac{\delta \boldsymbol{\tau}}{\delta t} = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{Re}}(\nabla\vec{u} + (\nabla\vec{u})^T), \tag{1.64}$$

which is the Oldroyd-B constitutive equation.

1.5.2 Fully Deterministic Multiscale Methods

From the previous section, we can see that the multiscale model for polymeric fluids involves a high-dimensional Fokker–Planck equation for the probability density function $\psi = \psi(\vec{x},\vec{q},t)$ coupled with the Navier–Stokes equations for the velocity $\vec{u} = \vec{u}(\vec{x},t)$ and the pressure $p(\vec{x},t)$. With a flow domain $\Omega \subset \mathbb{R}^d$, the configuration space domain D is a subset of $\mathbb{R}^{Kd}$, where K is the number of springs in the

bead-spring chain. As the name implies, this approach solves the system directly without introducing any stochastic elements. Deterministic methods are not popular and only a limited literature exists. The main shortcoming of this approach is the high-dimensional Fokker–Planck equation, which requires significant computational effort due to the curse of dimensionality. In 1972, Warner [58] simulated a simple shear flow of FENE dumbbells. The flow was simplified so that the probability density function ψ in the Fokker–Planck equation did not depend on the space variable $\vec{x}$ and therefore the dimension was halved. Such flows are also called homogeneous flows. In 1989, Fan [16] made an attempt to simulate a non-homogeneous flow field using a bead-rod model. No further work on the deterministic approach had been performed until 2003 when Chauvière and Lozinski published a series of papers, which developed the deterministic multiscale approach for simulating the FENE dumbbell model [12], [13], [41].

The key numerical technique used in the papers of Chauvière and Lozinski, motivated by the classical Peaceman–Rachford (1955) and Douglas–Rachford (1956) alternating direction schemes, is the alternating-direction approach based on splitting the differential operator in the Fokker–Planck equation. They separated the operator on the physical space from the one on the configuration space, hence decomposing the Fokker–Planck equation into two parts, and then used a spectral method to calculate the numerical solutions. The Fokker–Planck equation (1.47) is written in the following form:

$$\frac{\partial \psi}{\partial t} + (L_x + L_q)\psi = 0, \tag{1.65}$$

where L_q and L_x are differential operators in the variables $\vec{q}$ and $\vec{x}$ respectively, defined by

$$L_q\psi = \nabla_q \cdot ((\nabla_x \vec{u})\vec{q}\psi) - \frac{1}{2\mathrm{Wi}}(\nabla_q \cdot (\nabla_q \psi + \vec{F}(\vec{q})\psi)), \tag{1.66}$$

$$L_x\psi = \nabla_x \cdot (\vec{u}\psi) - \epsilon \Delta_x \psi. \tag{1.67}$$

Given ψ at time t^n, the two-stage alternating-direction method solves the q-direction equation (1.66) over half a time step $\Delta t/2$ to obtain $\psi(t^n + \frac{\Delta t}{2})$, and then

we advance another half time step starting from $t^n + \frac{\Delta t}{2}$ by solving the x-direction equation (1.67) to get the numerical approximation for $\psi(t^{n+1})$.

There are two advantages of this operator splitting technique. As highlighted in the previous paragraph, the Fokker–Planck equation for a non-homogeneous d-dimensional flow system with the dumbbell model is $2d$-dimensional; hence standard numerical schemes suffer from the curse of dimensionality. However, using the two-stage alternating-direction method, we reduce the dimension of the equation at each time step. The second advantage is the diversity of numerical methods that can be used. Before the splitting, we can only use one numerical scheme to solve the equation (1.47), while we are allowed to treat equation (1.66) and (1.67) with different numerical schemes. In 2009, based on the work of Chauvière and Lozinski, Knezevic and Süli [37] developed a similar technique using the alternating-direction method. On the physical domain Ω, they used a finite element approximation because FEM is a flexible method to deal with a complicated geometry of the flow domain. On the configuration space D, a Galerkin spectral method was used because the configuration domain is always a ball. By performing a polar (or spherical polar) coordinate transformation, the Galerkin spectral method is well suited for solving equation (1.66). Knezevic and Süli were the first researchers to simulate the six-dimensional Fokker–Planck equation coupled with the three-dimensional Navier–Stokes equations. The advantage of the splitting method considered in these papers is that is reduces the solution of the Fokker–Planck equations in six dimensions at each time level to a pair of three dimensional solves.

1.5.3 Stochastic Multiscale Methods

We have already mentioned this method in the previous section. The key idea is to use the system of stochastic differential equations instead of the high-dimensional Fokker–Planck equation. Then, the Kramers expression in the equation (1.48) can be replaced by $\mathbb{E}[\vec{Q}(t) \otimes \vec{F}(\vec{Q}(t))]$ where $\vec{Q}(t)$ is the stochastic process representing the configuration vector $\vec{q}(t)$; hence the extra stress tensor $\boldsymbol{\tau}$ can be calculated by the Monte Carlo method. After that the Navier–Stokes equations can be solved using a standard finite element method.

The first work considering such type of method dates back to 1993. Laso and Öttinger presented the idea of CONNFFESSIT, which stands for "Calculation of Non-Newtonian Flow: Finite Elements and Stochastic Simulation Technique" [39]. In this method, the finite element method is applied to the Navier–Stokes equations and in each element the polymeric stress is obtained from a large number of model polymer molecules. Each individual dumbbell is carried by the velocity field of the flow and the configuration is calculated through a stochastic process. In that paper, the authors considered the Hookean dumbbell model and applied the scheme to a 2D homogeneous flow problem. For the Hookean dumbbell model, the spring force is $\vec{F}(\vec{Q}) = H\vec{Q}$ and the stochastic differential equation is given by:

$$\mathrm{d}\vec{Q}(t) = \left((\nabla_x \vec{u})\vec{Q}(t) - \frac{\mathrm{H}}{2\mathrm{Wi}}\vec{Q}(t) \right) \mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}}\mathrm{d}\vec{W}(t), \tag{1.68}$$

where $\vec{W}(t)$ is a vectorial Brownian motion. The stochastic process $\vec{Q}(t)$ represents the configuration variable $\vec{q}$ and the equation (1.68) can be solved numerically, typically using the Euler–Maruyama method. Once the configurations are known, the extra stress is computed by the sample average:

$$\mathbb{E}[\vec{Q} \otimes \vec{F}(\vec{Q})] = \frac{1}{N}\sum_{i=1}^{N} \vec{Q}(t)^i \otimes \vec{F}(\vec{Q}(t)^i), \tag{1.69}$$

where N is the number of independent realizations and $Q(t)^i$ represents the ith dumbbell in the ensemble. We know that the extra stress tensor $\boldsymbol{\tau}$ is a variable of both time t and physical space $\vec{x}$. Here the random variable $\vec{Q}$ is governed by the equation (1.68) and the equation involves the term $(\nabla_x \vec{u})$, which depends on $\vec{x}$; therefore we can obtain the value of $\boldsymbol{\tau}(\vec{x}, t)$ by using equation (1.69). Three years later Hule, van Heel and van den Brule presented a new approach to this problem in the paper "Simulation of viscoelastic flows using Brownian configuration fields" [32]. For non-homogeneous flows, they note that it is better to use an expression for $\boldsymbol{\tau}$ that depends on both time and space. Instead of considering many discrete dumbbells, they proposed a collection of N continuous configuration fields (Brownian configuration fields) $\vec{Q}(\vec{x}, t)$. The values of this field are sampled independently from the initial distribution prescribed as the initial datum of the Fokker–Planck equation. From

equation (1.49), we know that the initial datum for ψ does not depend on the spatial variable $\vec{x}$; hence the configuration field is also spatially uniform at time zero. Then the velocity gradient, the spring force and the Brownian motion will act on the field as in equation (1.68) and the field will be also convected by the flow. Therefore the new stochastic differential equation reads

$$\mathrm{d}\vec{Q}(\vec{x},t) = \left(-\vec{u}(\vec{x},t)\cdot\nabla_x\vec{Q}(\vec{x},t) + (\nabla_x\vec{u}(\vec{x},t))\cdot\vec{Q}(\vec{x},t) - \frac{\mathrm{H}}{2\mathrm{Wi}}\vec{Q}(\vec{x},t)\right)\mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}}\mathrm{d}\vec{W}(t). \quad (1.70)$$

After solving this stochastic PDE, the stress tensor $\boldsymbol{\tau}$ is computed as the ensemble average, as before. Based on the idea of Brownian configuration fields, several papers ([38], [54], [57]) have developed these ideas further using the stochastic approach for micro-macro simulations. Recently, Griebel and Rüttgers [28] published the first simulation of 3D square-square contraction flow using the FENE model and the Brownian configuration field method; they performed the computations on a parallel computer to reduce the computing time.

An important advantage of the stochastic method is that it avoids the curse of dimensionality in the deterministic method. On the other hand it is still a computationally intensive procedure due to the low accuracy of the Monte Carlo method. For example, if we simulate a flow with $M = 10^4$ grid points and in each grid point we use $N = 10^4$ Monte Carlo realisations for computing the stress tensor, then in total we will need $M \cdot N = 10^8$ stochastic simulations per time step. Therefore a variance reduction method is essential.

In the stochastic approach, we need to solve the stochastic differential equation (1.70) and compute the expectation (1.69). Stochastic differential equations are only solvable analytically in some special cases. Therefore, we have to concentrate on numerical approximation schemes for SDEs involving a nonlinear FENE spring force. Here we need to introduce some concepts and definitions of numerical stochastic analysis [10], [20], [35], [52].

The Euler–Maruyama method. Let us consider the stochastic differential equation

$$\mathrm{d}S(t) = a(S(t), t)\mathrm{d}t + b(S(t), t)\mathrm{d}W(t), \quad 0 < t \leq T, \tag{1.71}$$

which is a shorthand for the integral form:

$$S(t) = S(0) + \int_0^t a(S(s), s)\,\mathrm{d}s + \int_0^t b(S(s), s)\,\mathrm{d}W_s, \tag{1.72}$$

where $S(0) = S_0$ is the initial condition, $W(t)$ is Brownian motion. The drift and volatility terms a and b are assumed to satisfy the usual conditions:

1. Lipschitz continuity in space:

$$|a(x,t) - a(y,t)| + |b(x,t) - b(y,t)| \leq L|x - y|. \tag{1.73}$$

2. linear growth bound:

$$|a(x,t)| + |b(x,t)| \leq L(1 + |x|). \tag{1.74}$$

3. square–root continuity in time:

$$|a(x,s) - a(x,t)| + |b(x,s) - b(x,t)| \leq L(1 + |x|)\sqrt{s - t}, \tag{1.75}$$

where L is some positive constant.

To apply a numerical scheme to (1.71), we first need to discretize the time interval. Let $\Delta t = \frac{T}{J}$ for some positive integer J and $t_j = j\Delta t$. We then deduce from the integral form that, for $j = 1, \ldots, J$:

$$\begin{aligned} S(t_j) &= S(t_{j-1}) + \int_{t_{j-1}}^{t_j} a(S(s), s)\mathrm{d}s + \int_{t_{j-1}}^{t_j} b(S(s), s)\mathrm{d}W_s \\ &\approx S(t_{j-1}) + a(S(t_{j-1}), t_{j-1})\Delta t + b(S(t_{j-1}), t_{j-1})(W(t_j) - W(t_{j-1})). \end{aligned} \tag{1.76}$$

Remark 1.3 *With the use of the Ito integral, the stochastic integrals have to be evaluated using the left-hand endpoint of the time interval $[t_{j-1}, t_j]$ and therefore all numerical schemes have to be explicit in the stochastic term, as otherwise the scheme would produce incorrect results. On the other hand the drift term can be evaluated implicitly, which leads to the backward Euler–Maruyama method.*

Let $\hat{S}_j$ denote the numerical approximation to $S(t_j)$. Then the Euler–Maruyama method takes the form

$$\hat{S}_j = \hat{S}_{j-1} + a(\hat{S}_{j-1}, t_{j-1})\Delta t + b(\hat{S}_{j-1}, t_{j-1})(W(t_j) - W(t_{j-1})), \tag{1.77}$$

where the Brownian increments $W(t_j) - W(t_{j-1}), \quad j = 1, \ldots, J$, are independent random variables of the form $\sqrt{\Delta t}\mathcal{N}(0,1)$, where $\mathcal{N}(0,1)$ denotes the normal distribution with mean 0 and standard deviation 1.

One can assess the quality of schemes by introducing the concepts of strong and weak convergence.

Definition 1.3 Weak convergence. *A method is said to have weak order of convergence equal to $\gamma > 0$ if there exists a constant C such that for all functions of class $g \in C^{2\gamma+1}(\mathbb{R})$ one has*

$$|\mathbb{E}g(\hat{S}_n) - \mathbb{E}g(S(\tau))| \leq C\Delta t^{\gamma}$$

at any fixed $\tau = n\Delta t \in [0, T]$ and Δt sufficiently small.

If conditions (1.73)–(1.75) are all satisfied, it can be shown that [35] the Euler–Maruyama method has weak order of convergence $\gamma = 1$.

Definition 1.4 Strong convergence. *A method is said to have strong order of convergence equal to γ if there exists a constant $C > 0$ such that one has*

$$\mathbb{E}|\hat{S}_n - S(\tau)| \leq C\Delta t^{\gamma}$$

for any fixed $\tau = n\Delta t \in [0, T]$ and Δt sufficiently small.

If the conditions (1.73)–(1.75) are all satisfied, then it can be shown that the Euler–Maruyama method has strong order of convergence $\gamma = \frac{1}{2}$. The strong order of convergence measures the rate at which the "mean of the error" decays as $\Delta t \to 0$. It plays a very important role in the analysis of the multilevel Monte Carlo method. Therefore we recall the following, more general, result concerning the strong convergence of the Euler–Maruyama method: $\mathbb{E}\left[|S(t) - \hat{S}_t|^p\right] = O(\Delta t^{p/2})$ for any $p \geq 2$ and $t \in [0, T]$. The detailed proof can be found in the book by Kloeden and Platen [35].

The Milstein method. The strong convergence order of the Euler–Maruyama method is only $\frac{1}{2}$. It is possible to raise the strong order to 1 [35] by adding a correction to the stochastic increment, giving Milstein's method.

Consider the autonomous stochastic differential equation

$$\mathrm{d}S(t) = a(S(t))\mathrm{d}t + b(S(t))\mathrm{d}W(t), \quad 0 < t \leq T.$$

Recall that Ito formula [35] for a general field Φ asserts that

$$\mathrm{d}\Phi = \frac{\partial \Phi}{\partial t}\mathrm{d}t + \frac{\partial \Phi}{\partial S}\mathrm{d}S + \frac{1}{2}\frac{\partial^2 \Phi}{\partial S^2}b^2\mathrm{d}t = \left(\frac{\partial \Phi}{\partial t} + \frac{\partial \Phi}{\partial S}a + \frac{1}{2}\frac{\partial^2 \Phi}{\partial S^2}b^2\right)\mathrm{d}t + \frac{\partial \Phi}{\partial S}b\mathrm{d}W(t).$$

We apply the Ito formula with $\Phi = a$ and $\Phi = b$ to expand the coefficients a and b in the SDE to obtain:

$$\mathrm{d}a(S(t)) = (a'(S(t))a(S(t)) + \frac{1}{2}a''(S(t))b^2(S(t)))\mathrm{d}t + a'(S(t))b(S(t))\mathrm{d}W(t),$$

$$\mathrm{d}b(S(t)) = (b'(S(t))a(S(t)) + \frac{1}{2}b''(S(t))b^2(S(t)))\mathrm{d}t + b'(S(t))b(S(t))\mathrm{d}W(t),$$

where the prime refers to differentiation in S. The integral form of the coefficients at time s $(t_{j-1} < s < t_j)$ is

$$\begin{aligned} a(S(s)) = a(S(t_{j-1})) &+ \int_{t_{j-1}}^{s} \left(a'(S(u))a(S(u)) + \frac{1}{2}a''(S(u))b^2(S(u))\right)\mathrm{d}u \\ &+ \int_{t_{j-1}}^{s} a'(S(u))b(S(u))\,\mathrm{d}W(u), \end{aligned}$$

$$\begin{aligned} b(S(s)) = b(S(t_{j-1})) &+ \int_{t_{j-1}}^{s} \left(b'(S(u))a(S(u)) + \frac{1}{2}b''(S(u))b^2(S(u))\right)\mathrm{d}u \\ &+ \int_{t_{j-1}}^{s} b'(S(u))b(S(u))\,\mathrm{d}W(u). \end{aligned}$$

Substituting $a(S(s)), b(S(s))$ into the integral form of the stochastic differential equation yields

$$\begin{aligned} S(t_j) = S(t_{j-1}) &+ \int_{t_{j-1}}^{t_j} \left(a + \int_{t_{j-1}}^{s} \left(a'a + \frac{1}{2}a''b^2\right)\mathrm{d}u + \int_{t_{j-1}}^{s} (a'b)\mathrm{d}W(u)\right)\mathrm{d}s \\ &+ \int_{t_{j-1}}^{t_j} \left(b + \int_{t_{j-1}}^{s} \left(b'a + \frac{1}{2}b''b^2\right)\mathrm{d}u + \int_{t_{j-1}}^{s} (b'b)\mathrm{d}W(u)\right)\mathrm{d}W(s), \end{aligned}$$

where $\mathrm{d}s \cdot \mathrm{d}u = O(\Delta t^2)$, $\mathrm{d}W(u) \cdot \mathrm{d}s = O(\Delta t^{3/2})$ and $\mathrm{d}W(u) \cdot \mathrm{d}W(s) = O(\Delta t)$. We ignore the higher order terms and only keep the $O(\Delta t)$ term. This leaves us with

$$\begin{aligned} S(t_j) &\approx S(t_{j-1}) + \int_{t_{j-1}}^{t_j} a\mathrm{d}s + \int_{t_{j-1}}^{t_j} \left(b + \int_{t_{j-1}}^{s} (b'b)\mathrm{d}W(u) \right) \mathrm{d}W(s) \\ &\approx S(t_{j-1}) + a(S(t_{j-1}))\Delta t + b(S(t_{j-1}))(W(t_j) - W(t_{j-1})) \\ &+ \int_{t_{j-1}}^{t_j} \int_{t_{j-1}}^{s} b'(S(u))b(S(u))\mathrm{d}W(u)\mathrm{d}W(s). \end{aligned}$$

The first two terms in the equation above give the Euler–Maruyama scheme. We approximate the third term above by:

$$\int_{t_{j-1}}^{t_j} \int_{t_{j-1}}^{s} b'(S(u))b(S(u))\mathrm{d}W(u)\mathrm{d}W(s) \approx b'(S(t_{j-1}))b(S(t_{j-1})) \int_{t_{j-1}}^{t_j} \int_{t_{j-1}}^{s} \mathrm{d}W(u)\mathrm{d}W(s),$$

and the stochastic integral is $\int_{t_{j-1}}^{t_j} \int_{t_{j-1}}^{s} \mathrm{d}W(u)\mathrm{d}W(s) = \frac{1}{2}((W(t_j) - W(t_{j-1}))^2 - \Delta t)$. Substituting this in our previous approximation, we obtain the Milstein scheme:

$$\begin{aligned} \hat{S}_j = \hat{S}_{j-1} &+ a(\hat{S}_{j-1})\Delta t + b(\hat{S}_{j-1})(W(t_j) - W(t_{j-1}) \\ &+ \frac{1}{2} b'(\hat{S}_{j-1})b(\hat{S}_{j-1})(W(t_j) - W(t_{j-1}))^2 - \Delta t). \end{aligned}$$

One can prove that the Milstein scheme converges strongly with order 1 to the solution of the SDE [35].

The Monte Carlo method. We mentioned that the stress tensor $\boldsymbol{\tau}$ is the expected value of $\vec{Q} \otimes \vec{F}(\vec{Q})$, where $\vec{Q}$ is the solution to the SDE and it is computed by Monte Carlo simulation. Despite the widespread use of the Monte Carlo method and numerous descriptions of the method in articles, it is virtually impossible to find a succint definition of the Monte Carlo method in the literature. Some authors prefer to use the term stochastic simulation for almost everything, reserving Monte Carlo only for Monte Carlo Integration and Monte Carlo Tests [50]. In this thesis, we use the terminology Monte Carlo to express the fact that we are approximating an expectation by the sample mean of a function of simulated random variables and we invoke the law of large numbers to approximate expectations.

Consider a random variable S having probability density function $f_S(x)$, which is greater than zero on a set of values χ. Then the expected value of a function g of S

is

$$\mathbb{E}[g(S)] = \sum_{x \in \chi} g(x) f_S(x),$$

if S is discrete, and

$$\mathbb{E}[g(S)] = \int_{x \in \chi} g(x) f_S(x) \mathrm{d}x,$$

if S is continuous.

Now, if we take N independent samples of S, $(S_1, S_2, \ldots, S_N)$ and we compute the average of f over the sample, then we will have the Monte Carlo estimator E_N of $\mathbb{E}[g(S)]$:

$$E_N = \frac{1}{N} \sum_{i=1}^{N} g(S_i).$$

One thing to note at this point is that E_N is an unbiased estimator for $\mathbb{E}[g(S)]$:

$$\mathbb{E}[E_N] = \mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} g(S_i)] = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[g(S_i)] = \mathbb{E}[g(S)].$$

Also, due to independence,

$$\mathbb{V}[E_N] = N^{-1} \mathbb{V}[S].$$

If $\mathbb{E}[g(S)]$ exists, then the weak law of large numbers tells us that for an arbitrarily small positive number ϵ, we have

$$\lim_{N \to \infty} P(|E_N - \mathbb{E}[g(S)]| \geq \epsilon) = 0.$$

The weak law essentially states that with a sufficiently large sample, there will be a very high probability that the average of the observations will be close to the expected value. Also, the strong law of large number tells us that the sample average converges almost surely to the expected value; i.e.,

$$P(\lim_{N \to \infty} E_N) = \mathbb{E}[g(S)]) = 1.$$

The Central Limit Theorem states that as N gets larger, the distribution of the difference between the sample average E_N and $\mathbb{E}[E_N]$, when multiplied by the factor $\sqrt{N}$ (that is $\sqrt{N}(E_N - \mathbb{E}[E_N])$), approximates the normal distribution with mean 0

and variance $\sigma^2 = \mathbb{V}[S]$. Loosely speaking, the Central Limit Theorem proves that, for large N,

$$|E_N - \mathbb{E}[g(S)]| \sim \sigma N^{-1/2} Z,$$

where Z is a $N(0,1)$ random variable. From the last equation, we can see that the error of the Monte Carlo method will decay like $O(N^{-1/2})$ as $N \to \infty$ provided σ is finite.

1.6 Multi-bead Simulations

So far, we have introduced the single-bead model (dumbbell model), however the polymer molecule is more accurately modelled by a chain with several beads and therefore a more complex multi-bead system is required. In Section 1.3 we have already mentioned the idea of multi-bead simulation. As we can see in Figure 1.2 (taken from Figueroa [19]), we will use $K+1$ ($K \geq 2$) beads connected with K elastic springs to represent a polymer chain. For each spring, we will need one configuration vector $\vec{q}_i$ to represent the conformation and orientation of the ith spring, hence the total configuration vector is $\vec{q} := (\vec{q_1}^T, \ldots, \vec{q_K}^T)^T \in D$, and the total configuration space D is the Cartesian product of all the sets D_i with $\vec{q}_i \in D_i, i = 1, \ldots, K$. The dimension of the vector $\vec{q}$ is K times the dimension of the physical space because the number of the components of $\vec{q}_i$ is the same as the dimension of the physical space. The resulting configuration space is therefore a high-dimensional space.

Due to the curse of dimensionality, the multi-bead simulation is still a challenging problem [28], [43]. The central goal and main contribution of this thesis is to develop a stochastic multiscale numerical method for the multi-bead FENE model.

Recall from Section 1.3 that the total spring force acting on the ith bead is:

$$\vec{F}_i^E = \sum_{j=1}^{K} G_{i,j} \vec{F}_j(\vec{q}_j, t).$$

More precisely, the spring force on bead i is given by:

$$\vec{F}_i^E = \begin{cases} \vec{F}_1(\vec{q}_1) & i = 1, \\ \vec{F}_i(\vec{q}_i) - \vec{F}_{i-1}(\vec{q}_{i-1}) & 1 < i \leq K, \\ -\vec{F}_K(\vec{q}_K) & i = K+1. \end{cases} \tag{1.78}$$

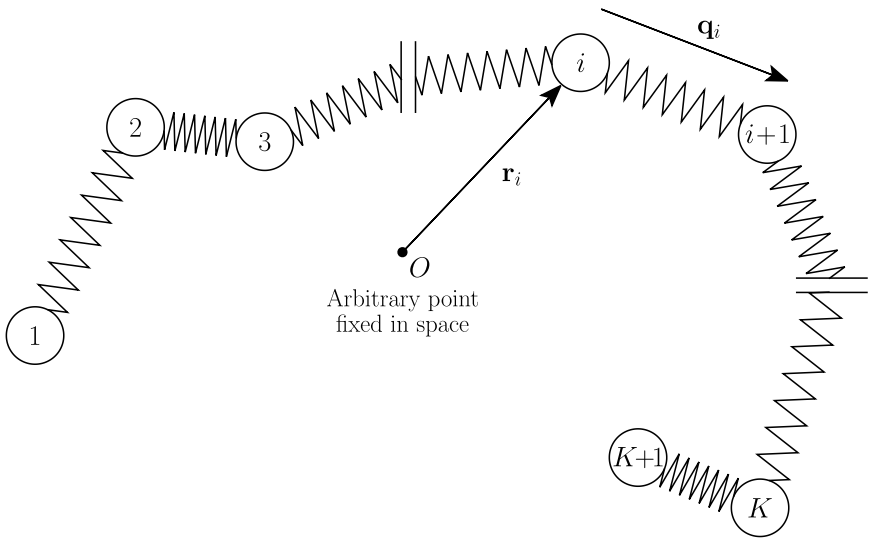


Figure 1.2: Illustration of the multi-bead spring model

Therefore the new system of coupled stochastic differential equations for the configuration vector is the following:

$$\begin{aligned}
\mathrm{d}\vec{Q}_1(t) &= \left((\nabla_x \vec{u})\vec{Q}_1(t) - \frac{1}{2\mathrm{Wi}}\left(-\vec{F}_2(\vec{Q}_2(t)) + 2\vec{F}_1(\vec{Q}_1(t))\right)\right)\mathrm{d}t \\
&\quad + \sqrt{\frac{1}{\mathrm{Wi}}}(\mathrm{d}\vec{W}_2(t) - \mathrm{d}\vec{W}_1(t)), \\
\mathrm{d}\vec{Q}_i(t) &= \left((\nabla_x \vec{u})\vec{Q}_i(t) - \frac{1}{2\mathrm{Wi}}\left(-\vec{F}_{i+1}(\vec{Q}_{i+1}(t)) + 2\vec{F}_i(\vec{Q}_i(t)) - \vec{F}_{i-1}(\vec{Q}_{i-1}(t))\right)\right)\mathrm{d}t \\
&\quad + \sqrt{\frac{1}{\mathrm{Wi}}}(\mathrm{d}\vec{W}_{i+1}(t) - \mathrm{d}\vec{W}_i(t)), \quad 1 < i < K, \\
\mathrm{d}\vec{Q}_K(t) &= \left((\nabla_x \vec{u})\vec{Q}_K(t) - \frac{1}{2\mathrm{Wi}}\left(-\vec{F}_{K-1}(\vec{Q}_{K-1}(t)) + 2\vec{F}_K(\vec{Q}_K(t))\right)\right)\mathrm{d}t \\
&\quad + \sqrt{\frac{1}{\mathrm{Wi}}}(\mathrm{d}\vec{W}_{K+1}(t) - \mathrm{d}\vec{W}_K(t)).
\end{aligned} \tag{1.79}$$

The corresponding Kramers expression for the polymeric extra stress tensor is the following:

$$\boldsymbol{\tau}(\vec{x},t) = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{ReWi}}\left(\sum_{i=1}^{K}\mathbb{E}[\vec{Q}_i(t)\otimes\vec{F}_i(\vec{Q}_i(t))] - \mathbf{I}\right). \tag{1.80}$$

1.7 Summary

We will focus on the stochastic multiscale numerical method described above to simulate the flow of dilute polymers using the FENE spring model. As discussed above, the computational cost is still large due to the low efficiency of the Monte Carlo method since the total cost of the standard Monte Carlo method is $O(\epsilon^{-3})$ where ϵ^2 is the mean square error. Therefore the central goal of this work is to use the recently developed multilevel Monte Carlo method in order to reduce the cost. Furthermore we are going to use the parallel programming and computing platform CUDA to implement the numerical scheme for the stochastic differential equation on a GPU. Meanwhile, we will compute the Navier–Stokes equations on the CPU. We can reduce the computational time significantly with the help of CUDA, which allows us to simulate more complicated problems, including for example the multi-bead spring model. So far much of the literature has been restricted to the dumbbell model, while only a few papers discuss simulations for the multi-bead model [52].

In Chapter 2, we start by considering the equilibrium behaviour of the Navier–Stokes–Fokker–Planck equation. We will show that when the velocity is zero on the boundary of the physical domain and the body force is ignored, a weak solution to the model decays exponentially in time. In Chapter 3, we will introduce the multilevel Monte Carlo method, which generalises the idea of variance reduction method and reduces the total cost. After that, we will illustrate our numerical algorithm for our stochastic differential equation, the Navier–Stokes equations and the coupled system. In Chapter 4, we will perform several numerical experiments based on the dumbbell model and the multi-bead model. The benchmark problems will be an enclosed flow on a square domain and the 4:1 contraction flow. Finally, we will discuss some difficulties arising in the analysis of the numerical algorithm and propose possible directions for future work.

Chapter 2

Exponential Decay to the Equilibrium Solution

The purpose of this chapter is to establish the rate of decay to the equilibrium for the multiscale model. We restrict our attention to the problem with homogeneous velocity boundary condition. We will prove that the velocity decays to zero exponentially at a rate which is independent of the choice of the initial datum. Meanwhile the solution to the Fokker–Planck equation converges to the Maxwellian M of the model, which means that all of the configuration vectors $\vec{q}_i$ have the same equilibrium probability density function $M_i(\vec{q}_i)$.

The reason for proving Theorem 2.1 which comes from [5] is to establish a test problem for our numerical experiments. We can use this test problem to assess our numerical algorithm because we know how the true solution behaves under some special conditions. We will perform a numerical experiment in Chapter 4 to check whether our numerical solutions match the results in Theorem 2.1. Without considering this model problem, it would be difficult to ascertain that the numerical algorithms are reliable because we do not know the true solution for a complicated model problem such as multi-bead simulations in an L-shaped domain.

This chapter considers the equilibrium solutions of the Navier–Stokes and Fokker–Planck equations for the FENE spring model. We shall show that, in the absence of a body force (i.e., with $\vec{f} = \vec{0}$), weak solutions to the equations (1.44)-(1.49) decay exponentially in time at a rate which is independent of the choice of the initial data for the Navier–Stokes and Fokker–Planck equations. It will be assumed throughout this

chapter that the unique global-in-time weak solution exists and it is regular enough. The corresponding proof can be found in the paper by Barrett and Süli [5]. Also the velocity is assumed to be zero on the boundary of the physical domain (i.e., $\vec{u} = \vec{0}$ on $\partial\Omega$). The motivation for studying this question is that we can understand the long-time behaviour of dilute polymers and we can also use the result to test the numerical scheme. In a later chapter, we will perform numerical experiments to show that our numerical results converge to the equilibrium solution at an exponential rate. The focus of this chapter is on the analysis of weak solutions and the proof of the exponential decay.

Recall that the Fokker–Planck equation in Chapter 1 takes the form:

$$\frac{\partial\psi}{\partial t}+(\vec{u}\cdot\nabla_x)\psi+\sum_{i=1}^{K}\nabla_{q_i}\cdot((\nabla_x\vec{u})\vec{q}_i\psi)=\epsilon\Delta_x\psi+\frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K}A_{ij}(\nabla_{q_i}\cdot(\nabla_{q_j}\psi+\vec{F}_j(\vec{q}_j)\psi)). \tag{2.1}$$

In general, the elastic spring force $\vec{F}_i$ of the ith spring in the chain is assumed to be defined by a (sufficiently smooth) potential U_i via

$$\vec{F}_i(\vec{q}_i)=U_i'(\frac{1}{2}|\vec{q}_i|^2)\vec{q}_i,\quad i=1,\dots,K. \tag{2.2}$$

The normalized Maxwellian M_i for each spring is defined by:

$$M_i(\vec{q}_i):=\frac{1}{Z_i}\exp(-U_i(\frac{1}{2}|\vec{q}_i|^2)),\quad Z_i:=\int_{D_i}\exp(-U_i(\frac{1}{2}|\vec{q}_i|^2))\,\mathrm{d}\vec{q}_i. \tag{2.3}$$

The total Maxwellian in the model is then the product of the individual Maxwellians:

$$M(\vec{q})=\prod_{i=1}^{K}M_i(\vec{q}_i). \tag{2.4}$$

We can now transform the equation (2.1) into a form, in which the spring force $\vec{F}$ has been absorbed into a weighted diffusion term,

$$\begin{aligned}&\frac{\partial\psi}{\partial t}+(\vec{u}\cdot\nabla_x)\psi+\sum_{i=1}^{K}\nabla_{q_i}\cdot((\nabla_x\vec{u})\vec{q}_i\psi)\\&\quad=\epsilon\Delta_x\psi+\frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K}A_{ij}\nabla_{q_i}\cdot\left(M\nabla_{q_j}\left(\frac{\psi}{M}\right)\right).\end{aligned} \tag{2.5}$$

We shall show that the equation (2.5) is equivalent to (2.1). First differentiating the term $\nabla_{q_j}\left(\frac{\psi}{M}\right)$, we obtain

$$\begin{aligned}
&\frac{\partial\psi}{\partial t} + (\vec{u}\cdot\nabla_x)\psi + \sum_{i=1}^{K}\nabla_{q_i}\cdot((\nabla_x\vec{u})\vec{q_i}\psi) \\
&\quad = \epsilon\Delta_x\psi + \frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K}A_{ij}\nabla_{q_i}\cdot\left(M\frac{(\nabla_{q_j}\psi)M - \psi(\nabla_{q_j}M)}{M^2}\right) \\
&\quad = \epsilon\Delta_x\psi + \frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K}A_{ij}\nabla_{q_i}\cdot(\nabla_{q_j}\psi) \\
&\quad - \frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K}A_{ij}\nabla_{q_i}\cdot\left(\frac{\psi(\nabla_{q_j}M)}{M}\right). \qquad (2.6)
\end{aligned}$$

Using the definition of the Maxwellian (2.3), we can compute

$$\begin{aligned}
\nabla_{q_j}M_j(\vec{q_j}) &= \nabla_{q_j}\frac{\exp(-U_j(\frac{1}{2}|\vec{q_j}|^2))}{Z_j} \\
&= -\frac{1}{Z_j}\exp(-U_j(\frac{1}{2}|\vec{q_j}|^2))U_j'(\frac{1}{2}|\vec{q_j}|^2)q_j \\
&= -M_j(\vec{q_j})U_j'(\frac{1}{2}|\vec{q_j}|^2)q_j. \qquad (2.7)
\end{aligned}$$

We observe that, for $i = 1, \ldots, K$,

$$M(\vec{q}) = M_1(\vec{q_1})M_2(\vec{q_2})\cdots M_K(\vec{q_K})$$

and

$$\nabla_{q_j}M = M_1(\vec{q_1})\cdots M_{j-1}(\vec{q}_{j-1})M_{j+1}(\vec{q}_{j+1})\cdots M_K(\vec{q_K})\nabla_{q_j}M_j(\vec{q_j}). \qquad (2.8)$$

Inserting (2.7) into (2.8) yields

$$\nabla_{q_j}M = -MU_j'(\frac{1}{2}|\vec{q_j}|^2)q_j. \qquad (2.9)$$

Combining the results from (2.6) and (2.9), we deduce that

$$\frac{\partial\psi}{\partial t} + (\vec{u}\cdot\nabla_x)\psi + \sum_{i=1}^{K}\nabla_{q_i}\cdot((\nabla_x\vec{u})\vec{q_i}\psi)$$

$$
\begin{aligned}
&= \epsilon\Delta_x\psi + \frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K} A_{ij}\nabla_{q_i}\cdot(\nabla_{q_j}\psi) \\
&+ \frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K} A_{ij}\nabla_{q_i}\cdot\left(\psi U_j'(\frac{1}{2}|\vec{q_j}|^2)q_j\right). \\
&= \epsilon\Delta_x\psi + \frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K} A_{ij}\nabla_{q_i}\cdot\left(\nabla_{q_j}\psi + U_j'(\frac{1}{2}|\vec{q_j}|^2)q_j\psi\right) \\
&= \epsilon\Delta_x\psi + \frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K} A_{ij}\nabla_{q_i}\cdot\left(\nabla_{q_j}\psi + \vec{F}_j(\vec{q_j})\psi\right), \qquad (2.10)
\end{aligned}
$$

and that completes the proof. The Maxwellian transformation used in (2.5) allows us to circumvent analytical difficulties introduced by the unbounded convection term, $\vec{F}$, and we will focus on the transformed equation to perform the analysis.

Definition 2.1 *The equations (1.45)-(1.49) and (2.10) will be referred as the model problem (P_ϵ).*

Definition 2.2 *The pair $(\vec{u},\hat{\psi})$ is a global-in-time weak solution to problem (P_ϵ), such that:*

$$
\begin{aligned}
&-\int_0^T\int_\Omega \vec{u}\cdot\frac{\partial\vec{w}}{\partial t}\mathrm{d}\vec{x}\,\mathrm{d}t + \int_0^T\int_\Omega\left((\vec{u}\cdot\nabla_x\vec{u})\cdot\vec{w} + \frac{\gamma}{Re}\nabla_x\vec{u}:\nabla_x\vec{w}\right)\mathrm{d}\vec{x}\,\mathrm{d}t \\
&\quad = \int_\Omega\left(\vec{u}_0(\vec{x})\cdot\vec{w}(\vec{x},0)\mathrm{d}\vec{x} + \int_0^T \vec{f}\cdot\vec{w} - \boldsymbol{\tau}:\nabla_x\vec{w}\right)\mathrm{d}t \\
&\quad \forall\vec{w}\in W^{1,1}(0,T;V_\sigma), s.t.\ \vec{w}(\cdot,T)=0;
\end{aligned}
$$

where V_σ denote the closure of the set of all divergence-free functions in the norm of the Sobolev space $H_0^1(\Omega)\cap H^\sigma(\Omega), \sigma\geq 1$, and

$$
\begin{aligned}
&-\int_0^T\int_{\Omega\times D} M\hat{\psi}\frac{\partial\hat{\varphi}}{\partial t}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\,\mathrm{d}t + \int_0^T\int_{\Omega\times D} M[\epsilon\nabla_x\hat{\psi} - \vec{u}\hat{\psi}]\cdot\nabla_x\hat{\varphi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\,\mathrm{d}t \\
&+ \frac{1}{2\mathrm{Wi}}\int_0^T\int_{\Omega\times D}\sum_{i,j=1}^{K} A_{ij}M\nabla_{q_j}\hat{\psi}\cdot\nabla_{q_i}\hat{\varphi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\,\mathrm{d}t \\
&-\int_0^T\int_{\Omega\times D} M\sum_{i=1}^{K}(\nabla_x\vec{u})\vec{q}_i\hat{\psi}\cdot\nabla_x\hat{\varphi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\,\mathrm{d}t \\
&= \int_{\Omega\times D}\hat{\psi}_0(\vec{x},\vec{q})\hat{\varphi}(\vec{x},\vec{q},0)\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}
\end{aligned}
$$

$$\forall \hat{\varphi} \in W^{1,1}(0,T;H^s(\Omega\times D)), s.t.\, \hat{\varphi}(\cdot,\cdot,T)=0.$$

where $s > 1+\frac{1}{2}(K+1)d$ *and* $:$ *represents the componentwise scalar product of matrices, that is, for square matrices* $\boldsymbol{a}$ *and* $\boldsymbol{b}$, $\boldsymbol{a}:\boldsymbol{b} := \sum_{i,j} a_{ij}b_{ij}$.

Remark 2.1 *The next theorem will involve the so called Bakry–Emery condition. The Bakry–Emery condition is the following: there exists a positive constant* μ *such that* $\mathbf{Hess}(-\log M(\vec{q})) \geq \mu\mathbf{I}$ *on* D *and can be understood as logarithmic concavity of the Maxwellian on* D. *The validity of the Bakry–Emery condition for the FENE Maxwellian follows from Remark 5.3 in Barrett and Süli [5].*

Theorem 2.1. [Theorem 7.1 in [6]] *Suppose the unique global-in-time weak solution* $(\vec{u},\hat{\psi})$ *to problem* (P_ϵ) *exists; the velocity is zero on the boundary of the physical domain* Ω *and* M *satisfies the Bakry–Emery condition (see Remark 2.1 above); then, for all* $T>0$,

$$\begin{aligned}&\|\vec{u}(T)\|^2 + \frac{k}{|\Omega|}\|\hat{\psi}(T)-1\|^2_{L^1_M(\Omega\times D)}\\ &\leq \exp(-\gamma_0 T)\left(\|\vec{u}_0\|^2 + 2k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0)\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right) + \frac{P^2\mathrm{Re}}{\gamma}\int_0^T \|\vec{f}(s)\|^2\,\mathrm{d}s, \quad (2.11)\end{aligned}$$

where $\hat{\psi} = \frac{\psi}{M}$, $\gamma_0 := \min(\frac{\gamma}{P^2\mathrm{Re}}, \frac{a_0\delta}{\mathrm{Wi}})$, $k = \frac{b+d+2}{b}\frac{1-\gamma}{ReWi} \in \mathbb{R}_{>0}$. *Here* P *is Poincaré's constant for* $\vec{u}$ *and* δ *comes from the logarithmic Sobolev inequality we will introduce later.* $\mathcal{F}(s) = s(\log s - 1) + 1$ *for* $s \geq 0$. *In particular if* $\vec{f} = 0$ *(no body force), the following inequality holds:*

$$\begin{aligned}&\|\vec{u}(T)\|^2 + \frac{k}{|\Omega|}\|\hat{\psi}(T)-1\|^2_{L^1_M(\Omega\times D)}\\ &\leq \exp(-\gamma_0 T)\left(\|u_0\|^2 + 2k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0)\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right). \quad (2.12)\end{aligned}$$

Here, $L^p_M(\Omega\times D)$, for $p\in[1,\infty)$, denotes the Maxwellian-weighted L^p space over $\Omega\times D$ with norm

$$\|\hat{\psi}\|_{L^p_M(\Omega\times D)} := \left\{\int_{\Omega\times D} M|\hat{\psi}|^p\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right\}^{\frac{1}{p}},$$

and $\|\cdot\|$ denotes the L^2 norm over Ω.

From (2.12) we can see that if the time T tends to infinity, the right-hand side of the inequality will decay to zero exponentially. On the other hand, $\|\vec{u}(T)\|^2$ and $\|\hat{\psi}(T)-1\|^2_{L^1_M(\Omega\times D)}$ are all nonnegative, therefore both must decay to zero. This also indicates that the equilibrium solution of the Fokker–Planck equation is the Maxwellian M.

2.1 Analysis of the Fokker–Planck Equation ($\vec{u}=\vec{0}$)

We shall prove Theorem 2.1 by considering a homogeneous flow first (i.e., with $\vec{u}=\vec{0}$). We start with establishing the weak formulation of the Fokker–Planck equation. By introducing the notation

$$\hat{\psi}=\frac{\psi}{M}, \tag{2.13}$$

we can rewrite the Fokker–Planck equation as

$$M\frac{\partial\hat{\psi}}{\partial t}-\epsilon M\Delta_x\hat{\psi}-\frac{1}{2\mathrm{Wi}}\sum_{i=1}^{K}\sum_{j=1}^{K}A_{ij}\nabla_{q_i}\cdot\left(M\nabla_{q_j}\hat{\psi}\right)=0. \tag{2.14}$$

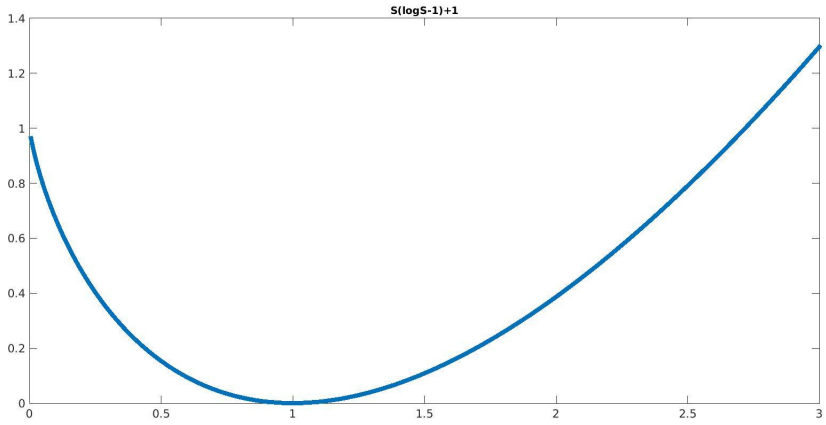
First of all, we define the function spaces relevant to the weak formulation of (2.14). Letting

$$H^1_M(\Omega\times D):=\left\{\hat{\varphi}\in L^1_{\mathrm{loc}}(\Omega\times D):\|\hat{\varphi}\|^2_{H^1_M(\Omega\times D)}:=\int_{\Omega\times D}M(|\hat{\varphi}|^2+|\nabla_q\hat{\varphi}|^2)\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}<\infty\right\}. \tag{2.15}$$

The weak formulation of (2.14) is then defined as follows. Given $\hat{\psi}_0\in L^1_M(\Omega\times D)$ such that $\mathcal{F}(\hat{\psi}_0)\in L^1_M(\Omega\times D)$ and $\hat{\psi}_0\geq 0$, find $\hat{\psi}\geq 0$ such that $\mathcal{F}(\hat{\psi})\in L^\infty(0,T;L^1_M(\Omega\times D))$, $\sqrt{\hat{\psi}}\in L^2(0,T;H^1_M(\Omega\times D))$ and

$$\begin{aligned}&\int_{\Omega\times D}M\frac{\partial\hat{\psi}}{\partial t}\varphi\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}+\epsilon\int_{\Omega\times D}M\nabla_x\hat{\psi}\cdot\nabla_x\varphi\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\&+\frac{1}{2\mathrm{Wi}}\sum_{i,j=1}^{K}A_{ij}\int_{\Omega\times D}M\nabla_{q_j}\hat{\psi}\cdot\nabla_{q_i}\varphi\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}=0\quad\forall\varphi\in H^1_M(\Omega\times D).\end{aligned} \tag{2.16}$$

Here the function $\mathcal{F}(s)$ is defined by $\mathcal{F}(s)=s(\log s-1)+1, s\geq 0$. Choosing



The function $\mathcal{F}(s) = s(\log s - 1) + 1$

$\varphi = \mathcal{F}'(\hat{\psi}) = \log \hat{\psi}$ in (2.16) yields that

$$\begin{aligned}
&\int_{\Omega\times D} M\frac{\partial \hat{\psi}}{\partial t}\mathcal{F}'(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} + \epsilon\int_{\Omega\times D} M\nabla_x\hat{\psi}\cdot\nabla_x(\mathcal{F}'(\hat{\psi}))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&+\frac{1}{2\mathrm{Wi}}\sum_{i,j=1}^{K} A_{ij}\int_{\Omega\times D} M\nabla_{q_j}\hat{\psi}\cdot\nabla_{q_i}(\mathcal{F}'(\hat{\psi}))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} = 0,\\
&= \mathrm{S}_1 + \mathrm{S}_2 + \mathrm{S}_3. \qquad (2.17)
\end{aligned}$$

We begin by considering S_1:

$$\begin{aligned}
\mathrm{S}_1 = \int_{\Omega\times D} M\frac{\partial \hat{\psi}}{\partial t}\mathcal{F}'(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} &= \int_{\Omega\times D} M\frac{\partial}{\partial t}(\mathcal{F}(\hat{\psi}))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \qquad (2.18)
\end{aligned}$$

Next, we consider the term S_2; noting that $\mathcal{F}''(s) = \frac{1}{s}$, we have that

$$\begin{aligned}
\mathrm{S}_2 &= \epsilon\int_{\Omega\times D} M\nabla_x\hat{\psi}\cdot\nabla_x(\mathcal{F}'(\hat{\psi}))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} = \epsilon\int_{\Omega\times D} M\nabla_x\hat{\psi}\cdot\frac{1}{\hat{\psi}}\nabla_x\hat{\psi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&= \epsilon\int_{\Omega\times D} M\frac{|\nabla_x\hat{\psi}|^2}{\hat{\psi}}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&= 4\epsilon\int_{\Omega\times D} M|\nabla_x\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \qquad (2.19)
\end{aligned}$$

Similarly as above,

$$\int_{\Omega\times D} M\nabla_{q_j}\hat{\psi}\nabla_{q_i}(\mathcal{F}'(\hat{\psi}))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}$$

$$= \int_{\Omega\times D} M\nabla_{q_j}\hat{\psi}\cdot\nabla_{q_i}\log\hat{\psi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}$$
$$= \int_{\Omega\times D} M\nabla_{q_j}\hat{\psi}\cdot\frac{\nabla_{q_i}\hat{\psi}}{\hat{\psi}}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \tag{2.20}$$

Finally, for the term S_3, we have that

$$\begin{aligned}\mathrm{S}_3 &= \frac{1}{2\mathrm{Wi}}\sum_{i,j=1}^{K} A_{ij}\int_{\Omega\times D} M\nabla_{q_j}\hat{\psi}\cdot\nabla_{q_i}(\mathcal{F}'(\hat{\psi}))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\ &= \frac{1}{2\mathrm{Wi}}\sum_{i,j=1}^{K} A_{ij}\int_{\Omega\times D}\frac{M}{\hat{\psi}}\nabla_{q_i}\hat{\psi}\cdot\nabla_{q_j}\hat{\psi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\ &= \frac{1}{2\mathrm{Wi}}\int_{\Omega\times D}\frac{M}{\hat{\psi}}\sum_{i,j=1}^{K} A_{ij}\nabla_{q_i}\hat{\psi}\cdot\nabla_{q_j}\hat{\psi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}.\end{aligned} \tag{2.21}$$

We recall that the matrix $\mathbf{A}\in\mathbb{R}^{K\times K}$ is a symmetric positive definite matrix with smallest eigenvalue $a_0\in\mathbb{R}_{>0}$, and therefore the following inequality holds:

$$\sum_{i,j=1}^{K} A_{ij}\nabla_{q_i}\hat{\psi}\cdot\nabla_{q_j}\hat{\psi}\mathrm{d}\vec{q}\mathrm{d}\vec{x} \geq a_0\sum_{i=1}^{k}|\nabla_{q_i}\hat{\psi}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \tag{2.22}$$

Substituting (2.22) into (2.21), we obtain

$$\begin{aligned}\mathrm{S}_3 &= \frac{1}{2\mathrm{Wi}}\int_{\Omega\times D}\frac{M}{\hat{\psi}}\sum_{i,j=1}^{K} A_{ij}\nabla_{q_i}\hat{\psi}\cdot\nabla_{q_j}\hat{\psi}\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\ &\geq \frac{a_0}{2\mathrm{Wi}}\int_{\Omega\times D}\frac{M}{\hat{\psi}}\sum_{i=1}^{K}|\nabla_{q_i}\hat{\psi}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\ &= \frac{a_0}{2\mathrm{Wi}}\int_{\Omega\times D}\frac{M}{\hat{\psi}}|\nabla_q\hat{\psi}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\ &= \frac{2a_0}{\mathrm{Wi}}\int_{\Omega\times D} M|\nabla_q\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}.\end{aligned} \tag{2.23}$$

Combining (2.18), (2.19) and (2.23), we have that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\mathrm{d}\vec{x} + 4\epsilon\int_{\Omega\times D} M|\nabla_x\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\ + \frac{2a_0}{\mathrm{Wi}}\int_{\Omega\times D} M|\nabla_q\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\leq 0.\end{aligned} \tag{2.24}$$

In (2.24) the second term $4\epsilon \int_{\Omega\times D} M|\nabla_x \sqrt{\hat{\psi}}|^2 \mathrm{d}\vec{q}\,\mathrm{d}\vec{x}$ is nonnegative; therefore we can ignore this term and the inequality still holds.

In fact, there is a connection between the terms $\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}$ and $\int_{\Omega\times D} M|\nabla_q \sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}$; this can be seen by noting the logarithmic Sobolev inequality.

Remark 2.2 *The logarithmic Sobolev inequality has the form:*

$$\int_D M(\vec{q})|\hat{\rho}(\vec{q})|^2 \log \frac{|\hat{\rho}(\vec{q})|^2}{\|\hat{\rho}\|^2_{L^2_M(D)}}\,\mathrm{d}\vec{q} \le \frac{2}{\delta}\int_D M(\vec{q})|\nabla_q \hat{\rho}(\vec{q})|^2\,\mathrm{d}\vec{q} \quad \forall \hat{\rho} \in H^1_M(D), \tag{2.25}$$

where $\delta > 0$ is a positive constant. This inequality holds when the Maxwellian M satisfies the Bakry–Emery condition (cf Remark 2.1).

Choosing $|\hat{\rho}(\vec{q})|^2 = \hat{\psi}(\vec{q})$ in (2.25), we obtain that

$$\int_D M(\vec{q})\hat{\psi}(\vec{q}) \log \frac{\hat{\psi}(\vec{q})}{\|\hat{\psi}\|_{L^1_M(D)}}\,\mathrm{d}\vec{q} \le \frac{2}{\delta}\int_D M(\vec{q})|\nabla_q \sqrt{\hat{\psi}(\vec{q})}|^2\,\mathrm{d}\vec{q}, \tag{2.26}$$

for all $\hat{\psi} \ge 0$ on D and $\sqrt{\hat{\psi}} \in H^1_M(D)$. Using the definition of $\|\cdot\|_{L^1_M(D)}$ and (2.13), we have that

$$\|\hat{\psi}\|_{L^1_M(D)} = \int_D M\hat{\psi}\,\mathrm{d}\vec{q} = \int_D \psi\,\mathrm{d}\vec{q} = 1. \tag{2.27}$$

The last equality holds due to the fact that ψ is a probability density function on D.

Now we are ready to find the connection between S_1 and S_3. Substituting $\mathcal{F}(\hat{\psi}) = \hat{\psi}(\log\hat{\psi} - 1) + 1$ into the integral, we deduce that

$$\begin{aligned}
&\int_D M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q} = \int_D M\left[\hat{\psi}(\log\hat{\psi} - 1) + 1\right]\mathrm{d}\vec{q} \\
&= \int_D M\hat{\psi}\log\hat{\psi}\,\mathrm{d}\vec{q} - \int_D M\hat{\psi}\,\mathrm{d}\vec{q} + \int_D M\,\mathrm{d}\vec{q} \\
&= \int_D M\hat{\psi}\log\hat{\psi}\,\mathrm{d}\vec{q},
\end{aligned} \tag{2.28}$$

where $\int_D M\hat{\psi}\,\mathrm{d}\vec{q} = \int_D M\,\mathrm{d}\vec{q} = 1$. Applying the logarithmic Sobolev inequality (2.26) to (2.28), we have that

$$\begin{aligned}
&\int_D M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q} \le \frac{2}{\delta}\int_D M|\nabla_q\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q} \\
&\Rightarrow \int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \le \frac{2}{\delta}\int_{\Omega\times D} M|\nabla_q\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}
\end{aligned}$$

$$\Rightarrow \int_{\Omega\times D} M|\nabla_q \sqrt{\hat{\psi}}|^2 \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \geq \frac{\delta}{2} \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \tag{2.29}$$

Using the result from (2.29) and plugging it into (2.24), we deduce that

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} + \frac{a_0\delta}{\mathrm{Wi}} \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \leq 0. \tag{2.30}$$

Multiplying (2.30) by $\exp(\frac{a_0\delta}{\mathrm{Wi}}t)$, we have that

$$\frac{\mathrm{d}}{\mathrm{d}t} \left[\exp\left(\frac{a_0\delta}{\mathrm{Wi}}t\right) \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \right] \leq 0. \tag{2.31}$$

Integrating (2.31) from time 0 to s yields

$$\begin{aligned}
&\int_0^s \frac{\mathrm{d}}{\mathrm{d}t} \left[\exp\left(\frac{a_0\delta}{\mathrm{Wi}}t\right) \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \right] \mathrm{d}t \\
&= \exp\left(\frac{a_0\delta}{\mathrm{Wi}}s\right) \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}(\vec{x},\vec{q},s)) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} - \exp\left(\frac{a_0\delta}{\mathrm{Wi}}0\right) \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \\
&\Rightarrow \exp\left(\frac{a_0\delta}{\mathrm{Wi}}s\right) \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}(\vec{x},\vec{q},s)) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \leq \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \\
&\Rightarrow \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}(\vec{x},\vec{q},s)) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \leq \exp\left(-\frac{a_0\delta}{\mathrm{Wi}}s\right) \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}.
\end{aligned} \tag{2.32}$$

Next we need to apply the Csiszár–Kullback inequality [2] to obtain an upper bound for $\|\hat{\psi}(s) - 1\|_{L^1_M(D)}$; it states that

$$\|\hat{\psi}(s) - 1\|_{L^1_M(D)} \leq \left[2 \int_D M\mathcal{F}(\hat{\psi}(\vec{x},\vec{q},s)) \,\mathrm{d}\vec{q} \right]^{\frac{1}{2}}. \tag{2.33}$$

Using the definition of the norm of Maxwellian-weighted space and integrating (2.33) over Ω implies:

$$\int_\Omega \left(\int_D M|\hat{\psi}(\vec{x},\vec{q},s) - 1| \mathrm{d}\vec{q} \right)^2 \mathrm{d}\vec{x} \leq 2 \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}(\vec{x},\vec{q},s)) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \tag{2.34}$$

Now we are able to bound $\|\hat{\psi} - 1\|^2_{L^1_M(\Omega\times D)}$ in Theorem 2.1. By the definition of $\|\cdot\|_{L^1_M(\Omega\times D)}$ and the Cauchy–Schwarz inequality, we have that

$$\begin{aligned}
\|\hat{\psi}(\cdot,\cdot,s) - 1\|^2_{L^1_M(\Omega\times D)} &= \left[\int_{\Omega\times D} M(\hat{\psi}(\vec{x},\vec{q},s) - 1) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \right]^2 \\
&= \left[\int_\Omega 1 \cdot \int_D M(\hat{\psi}(\vec{x},\vec{q},s) - 1) \,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \right]^2
\end{aligned}$$

$$\leq \left[\left(\int_\Omega 1^2 \mathrm{d}\vec{x} \right)^{\frac{1}{2}} \cdot \left(\int_\Omega (\int_D M(\hat{\psi}(\vec{x},\vec{q},s) - 1)\, \mathrm{d}\vec{q})^2 \mathrm{d}\vec{x} \right)^{\frac{1}{2}} \right]^2$$
$$= |\Omega| \int_\Omega \left(\int_D M(\hat{\psi}(\vec{x},\vec{q},s) - 1)\, \mathrm{d}\vec{q} \right)^2 \mathrm{d}\vec{x}. \tag{2.35}$$

Combining (2.35) with (2.34) yields

$$\|\hat{\psi}(\cdot,\cdot,s) - 1\|^2_{L^1_M(\Omega\times D)} \leq 2|\Omega| \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}(\vec{x},\vec{q},s))\, \mathrm{d}\vec{q}\mathrm{d}\vec{x}. \tag{2.36}$$

Finally we use (2.32) to bound the right-hand side in (2.36), and we deduce that

$$\|\hat{\psi}(\cdot,\cdot,s) - 1\|^2_{L^1_M(\Omega\times D)} \leq 2|\Omega| \exp\left(-\frac{a_0\delta}{\mathrm{Wi}} s \right) \int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0)\, \mathrm{d}\vec{q}\mathrm{d}\vec{x}. \tag{2.37}$$

Taking $\vec{u} = \vec{0}, \vec{f} = \vec{0}$, multiplying (2.37) by a positive constant $\frac{k}{|\Omega|}$ and choosing $s = T$ yields (2.11) in Theorem 2.1. In the rest of this chapter we will take the velocity into consideration and prove the complete Theorem 2.1.

2.2 Analysis of the Full Navier–Stokes–Fokker–Planck Equation

In the last section, we proved Theorem 2.1 when the velocity was taken to be zero, therefore we did not need to consider the Navier–Stokes equations. Recall the Navier–Stokes equations from Chapter 1:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla_x\vec{u} = -\nabla_x p + \frac{\gamma}{\mathrm{Re}}\Delta_x\vec{u} + \nabla_x\cdot\boldsymbol{\tau} + \vec{f} \quad \text{in } \Omega, \tag{2.38}$$
$$\nabla_x\cdot\vec{u} = 0 \quad \text{in } \Omega. \tag{2.39}$$

We consider the following Hilbert spaces:

$$H := \left\{ \vec{w} \in L^2(\Omega) : \nabla\cdot\vec{w} = 0, \text{ and } \vec{w}\cdot\vec{n}|_{\partial\Omega} = 0 \right\},$$
$$V := \left\{ \vec{w} \in H^1_0(\Omega) : \nabla\cdot\vec{w} = 0 \right\}. \tag{2.40}$$

Then the weak formulation of the Navier–Stokes equations is the following: find $\vec{u} \in L^\infty(0,T;H) \cap L^2(0,T;V)$ such that

$$\int_\Omega \frac{\partial\vec{u}}{\partial t}\cdot\vec{v}\,\mathrm{d}\vec{x} + \int_\Omega (\vec{u}\cdot\nabla\vec{u})\cdot\vec{v}\,\mathrm{d}\vec{x} + \frac{\gamma}{\mathrm{Re}}\int_\Omega \nabla\vec{u} : \nabla\vec{v}\,\mathrm{d}\vec{x}$$

$$= \int_\Omega p\nabla\cdot\vec{v}\,\mathrm{d}\vec{x} - \int_\Omega \boldsymbol{\tau} : \nabla\vec{v}\,\mathrm{d}\vec{x} + \int_\Omega \vec{f}\cdot\vec{v}\,\mathrm{d}\vec{x} \quad \forall\vec{v}\in V. \tag{2.41}$$

All the boundary terms are dropped due to the assumption $\vec{u} = \vec{0}$ on $\partial\Omega$. The term $\int_\Omega p\nabla\cdot\vec{v}\,\mathrm{d}\vec{x}$ also vanishes because the test function $\vec{v}$ is in the space V and the divergence of $\vec{v}$ is therefore zero.

Choosing $\vec{v} = \vec{u}$ in (2.41), we have that

$$\begin{aligned}
&\int_\Omega \frac{\partial\vec{u}}{\partial t}\cdot\vec{u}\,\mathrm{d}\vec{x} + \int_\Omega (\vec{u}\cdot\nabla\vec{u})\cdot\vec{u}\,\mathrm{d}\vec{x} + \frac{\gamma}{\mathrm{Re}}\int_\Omega \nabla\vec{u} : \nabla\vec{u}\,\mathrm{d}\vec{x} \\
&= -\int_\Omega \boldsymbol{\tau} : \nabla\vec{u}\,\mathrm{d}\vec{x} + \int_\Omega \vec{f}\cdot\vec{u}\,\mathrm{d}\vec{x}. \qquad (2.42)\\
&\Rightarrow \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\vec{u}\|^2 + \int_\Omega (\nabla\cdot(\vec{u}\otimes\vec{u}))\cdot\vec{u}\,\mathrm{d}\vec{x} + \frac{\gamma}{\mathrm{Re}}\|\nabla\vec{u}\|^2 \\
&= -\int_\Omega \boldsymbol{\tau} : \nabla\vec{u}\,\mathrm{d}\vec{x} + \int_\Omega \vec{f}\vec{u}\,\mathrm{d}\vec{x}. \qquad (2.43)
\end{aligned}$$

Here we observe that $\vec{u}\cdot\nabla\vec{u} = \nabla\cdot(\vec{u}\otimes\vec{u})$. Thus we can eliminate this term by performing integration by parts and using the fact $\nabla\cdot\vec{u} = 0$ and we can deduce that

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\vec{u}\|^2 + \frac{\gamma}{\mathrm{Re}}\|\nabla\vec{u}\|^2 = \int_\Omega \vec{f}\cdot\vec{u}\,\mathrm{d}\vec{x} - \int_\Omega \boldsymbol{\tau} : \nabla\vec{u}\,\mathrm{d}\vec{x}. \tag{2.44}$$

Let us now go back to the Fokker–Planck equation. In Section 2.1, we began with the weak formulation of the Fokker–Planck equation. Then we created a convex function $\mathcal{F}(s)$ and picked the test function $\varphi = \mathcal{F}'(\hat{\psi})$. After that we considered these terms separately and finally obtained the inequality (2.24). For the full Fokker–Planck equation, we will repeat the same analysis except that we have two more terms in the weak formulation of the Fokker–Planck equation, which are related to the velocity $\vec{u}$. By noting (2.5), we write these two terms as:

$$\mathrm{T}_1 = \int_{\Omega\times D} \left(M\vec{u}\cdot\nabla_x\hat{\psi}\right)\mathcal{F}'(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}, \tag{2.45}$$

$$\mathrm{T}_2 = \int_{\Omega\times D} \sum_{i=1}^{K} \left(\nabla_{q_i}\cdot((\nabla_x\vec{u})\vec{q}_i M\hat{\psi})\right)\mathcal{F}'(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \tag{2.46}$$

Therefore the weak formulation of the Fokker–Planck equation with the above choice of test function becomes $\mathrm{S}_1+\mathrm{S}_2+\mathrm{S}_3+\mathrm{T}_1+\mathrm{T}_2 = 0$. We perform the same analysis for the first three terms as before; then we can rewrite (2.24) as the following inequality:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} + \int_{\Omega\times D}\left(M\vec{u}\cdot\nabla_x\hat{\psi}\right)\mathcal{F}'(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}$$

$$+\int_{\Omega\times D}\sum_{i=1}^{K}\left(\nabla_{q_i}\cdot((\nabla_x\vec{u})\vec{q}_i M\hat{\psi})\right)\mathcal{F}'(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}+\frac{2a_0}{\mathrm{Wi}}\int_{\Omega\times D}M|\nabla_q\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\leq 0. \tag{2.47}$$

We start by considering the term T_1. On noting that $(\nabla_x\hat{\psi})\mathcal{F}'(\hat{\psi})=\nabla_x\mathcal{F}(\hat{\psi})$ and M is independent of x,

$$\begin{aligned}
\mathrm{T}_1&=\int_{\Omega\times D}\left(M\vec{u}\cdot\nabla_x\hat{\psi}\right)\mathcal{F}'(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&=\int_{\Omega\times D}M\vec{u}\cdot\nabla_x\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&=-\int_{\Omega\times D}(\nabla_x\cdot(M\vec{u}))\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}+\int_{\partial\Omega\times D}(M\vec{u}\cdot\vec{n}_x)\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&=-\int_{\Omega\times D}M(\nabla_x\cdot\vec{u})\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\\
&=0.
\end{aligned}\tag{2.48}$$

Again the boundary term is zero due to the assumption that $\vec{u}=\vec{0}$ on $\partial\Omega$. Next we shall focus on the second term T_2. We will find later that this term has an important relationship with the polymeric extra tensor $\boldsymbol{\tau}$. We perform an integration by parts first, and note that the boundary term is

$$\sum_{i=1}^{K}\int_{\Omega\times D_1\times\cdots\times\partial D_i\times\cdots\times D_K}\left((\nabla_x\vec{u})\vec{q}_i M\hat{\psi}\cdot\vec{n}_{q_i}\right)\mathcal{F}'(\hat{\psi})\mathrm{d}\vec{q}_1\cdots\mathrm{d}\vec{q}_{i-1}\mathrm{d}s(\vec{q}_i)\mathrm{d}\vec{q}_{i+1}\cdots\mathrm{d}\vec{q}_K\mathrm{d}\vec{x}, \tag{2.49}$$

where $\vec{n}_{q_i}$ is the outward unit normal vector to ∂D_i and

$$M(\vec{q})=M_1(\vec{q_1})M_2(\vec{q_2})\cdots M_K(\vec{q_K}). \tag{2.50}$$

We decompose M in (2.49) into the product of the individual Maxwellians. Furthermore, for the FENE spring, the spring force $\vec{F}_i$ and the corresponding potential U_i are given by

$$\vec{F}_i(\vec{q}_i)=\frac{\vec{q}}{1-\frac{|\vec{q}_i|^2}{b}},\qquad U_i(s)=-\frac{b}{2}\log\left(1-\frac{2s}{b}\right). \tag{2.51}$$

Recall that by the definition of the Maxwellian (2.3) we have that

$$M_i(\vec{q}_i):=\frac{1}{Z_i}\exp\left(-U_i(\frac{1}{2}|\vec{q}_i|^2)\right)=\frac{1}{Z_i}\left(1-\frac{|q_i|^2}{b}\right)^{\frac{b}{2}}. \tag{2.52}$$

From (2.52), we can clearly see that $M_i(\vec{q}_i) = 0$ on ∂D_i. Combining this with (2.49) and (2.50), we deduce that

$$\sum_{i=1}^{K} \int_{\Omega \times D_1 \times \cdots \times \partial D_i \times \cdots \times D_K} \left((\nabla_x \vec{u}) \vec{q}_i M \hat{\psi} \cdot \vec{n}_{q_i} \right) \mathcal{F}'(\hat{\psi}) \\ \mathrm{d}\vec{q}_1 \cdots \mathrm{d}\vec{q}_{i-1} \mathrm{d}s(\vec{q}_i) \mathrm{d}\vec{q}_{i+1} \cdots \mathrm{d}\vec{q}_K \mathrm{d}\vec{x} = 0. \tag{2.53}$$

Performing integration by parts and using the result from (2.53), we have that

$$\begin{aligned} \mathrm{T}_2 &= \int_{\Omega \times D} \sum_{i=1}^{K} \left(\nabla_{q_i} \cdot ((\nabla_x \vec{u}) \vec{q}_i M \hat{\psi}) \right) \mathcal{F}'(\hat{\psi}) \, \mathrm{d}\vec{q} \mathrm{d}\vec{x} \\ &= -\sum_{i=1}^{K} \int_{\Omega \times D} (\nabla_x \vec{u}) \vec{q}_i M \hat{\psi} \cdot \nabla_{q_i} \mathcal{F}'(\hat{\psi}) \, \mathrm{d}\vec{q} \mathrm{d}\vec{x} \\ &= -\sum_{i=1}^{K} \int_{\Omega \times D} (\nabla_x \vec{u}) \vec{q}_i M \hat{\psi} \cdot \mathcal{F}''(\hat{\psi}) \nabla_{q_i} \hat{\psi} \, \mathrm{d}\vec{q} \mathrm{d}\vec{x} \\ &= -\sum_{i=1}^{K} \int_{\Omega \times D} (\nabla_x \vec{u}) \vec{q}_i M \hat{\psi} \cdot \frac{1}{\hat{\psi}} \nabla_{q_i} \hat{\psi} \, \mathrm{d}\vec{q} \mathrm{d}\vec{x} \\ &= -\sum_{i=1}^{K} \int_{\Omega \times D} (\nabla_x \vec{u}) \vec{q}_i M \cdot \nabla_{q_i} \hat{\psi} \, \mathrm{d}\vec{q} \mathrm{d}\vec{x}. \end{aligned} \tag{2.54}$$

Here we have used the fact that $\mathcal{F}''(s) = \frac{1}{s}$. Observing that $\nabla_x \vec{u}$ does not depend on $\vec{q}$ and the index i, we can separate the integral $\int_{\Omega \times D}$ into two parts. Therefore T_2 becomes

$$\mathrm{T}_2 = -\int_{\Omega} (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(\int_D \vec{q}_i M (\nabla_{q_i} \hat{\psi}) \mathrm{d}\vec{q} \right) \mathrm{d}\vec{x}. \tag{2.55}$$

Using integration by parts again on the term $\int_D \vec{q}_i M (\nabla_{q_i} \hat{\psi}) \mathrm{d}\vec{q}$, we have that

$$\begin{aligned} \mathrm{T}_2 &= -\int_{\Omega} (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(-\int_D (\nabla_{q_i} (\vec{q}_i M)) \hat{\psi} \, \mathrm{d}\vec{q} \right) \mathrm{d}\vec{x} \\ &= -\int_{\Omega} (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(-\int_D (M \mathbf{I} + \vec{q}_i \otimes \nabla_{q_i} M) \hat{\psi} \, \mathrm{d}\vec{q} \right) \mathrm{d}\vec{x}. \\ &= -\int_{\Omega} (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(-\int_D (\vec{q}_i \otimes \nabla_{q_i} M) \hat{\psi} \, \mathrm{d}\vec{q} - \mathbf{I} \int_D M \hat{\psi} \, \mathrm{d}\vec{q} \right) \mathrm{d}\vec{x} \\ &= -\int_{\Omega} (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(-\int_D (\vec{q}_i \otimes \nabla_{q_i} M) \hat{\psi} \, \mathrm{d}\vec{q} - \mathbf{I} \int_D \psi \, \mathrm{d}\vec{q} \right) \mathrm{d}\vec{x} \end{aligned}$$

$$= -\int_\Omega (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(-\int_D (\vec{q}_i \otimes \nabla_{q_i} M)\hat{\psi}\, \mathrm{d}\vec{q} - \mathbf{I} \right) \mathrm{d}\vec{x}. \tag{2.56}$$

In Section 2.1, we have already calculated the term $\nabla_{q_i} M$ (see equation (2.9)). Inserting (2.9) into (2.56) and noting that $\vec{q}_i \otimes \vec{q}_i = \vec{q}_i \vec{q}_i^T$, we obtain that

$$\begin{aligned} \mathrm{T}_2 &= -\int_\Omega (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(\int_D \vec{q}_i \vec{q}_i^T M \hat{\psi} U_i'(\frac{1}{2}|\vec{q}_i|^2) \mathrm{d}\vec{q} - \mathbf{I} \right) \mathrm{d}\vec{x} \\ &= -\int_\Omega (\nabla_x \vec{u}) : \sum_{i=1}^{K} \left(\int_D \vec{q}_i \vec{q}_i^T \psi U_i'(\frac{1}{2}|\vec{q}_i|^2) \mathrm{d}\vec{q} - \mathbf{I} \right) \mathrm{d}\vec{x}. \end{aligned} \tag{2.57}$$

In Section 1.4 we stated that the polymeric extra stress tensor $\boldsymbol{\tau}$ for the dumbbell model is defined by the Kramers expression (equation (1.39)):

$$\boldsymbol{\tau}(\vec{x}, t) = n_p k_B T (\int_D \vec{q} \otimes \vec{F}(\vec{q}) \psi(\vec{x}, \vec{q}, t) \mathrm{d}\vec{q} - \mathbf{I}). \tag{2.58}$$

Similarly, for the multi-bead model introduced in Section 1.6, the Kramers expression is given by

$$\boldsymbol{\tau}(\vec{x}, t) = k \left(\sum_{i=1}^{K} \int_D \vec{q}_i \otimes \vec{F}_i(\vec{q}_i) \psi(\vec{x}, \vec{q}, t)\, \mathrm{d}\vec{q} - \mathbf{I} \right), \tag{2.59}$$

where $k = \frac{b+d+2}{b} \frac{1-\gamma}{\mathrm{ReWi}} > 0$ is a dimensionless constant. Noting that $\vec{F}_i(\vec{q}) = U_i'(\frac{1}{2}|\vec{q}_i|^2)\vec{q}_i$, we can rewrite (2.59) as

$$\boldsymbol{\tau}(\vec{x}, t) = k \left(\sum_{i=1}^{K} \int_D \vec{q}_i \vec{q}_i^T \psi U_i'(\frac{1}{2}|\vec{q}_i|^2)\, \mathrm{d}\vec{q} - \mathbf{I} \right). \tag{2.60}$$

Combining (2.60) with (2.57), we have that

$$\mathrm{T}_2 = -\frac{1}{k} \int_\Omega (\nabla_x \vec{u}) : \boldsymbol{\tau}\, \mathrm{d}\vec{x}. \tag{2.61}$$

Finally, using the results from (2.47), (2.48) and (2.61), we conclude from the Fokker–Planck equation that

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega \times D} M \mathcal{F}(\hat{\psi})\, \mathrm{d}\vec{q}\mathrm{d}\vec{x} - \frac{1}{k} \int_\Omega (\nabla_x \vec{u}) : \boldsymbol{\tau}\, \mathrm{d}\vec{x} \\ &+ \frac{2a_0}{\mathrm{Wi}} \int_{\Omega \times D} M |\nabla_q \sqrt{\hat{\psi}}|^2\, \mathrm{d}\vec{q}\mathrm{d}\vec{x} \leq 0. \end{aligned} \tag{2.62}$$

Comparing (2.62) with (2.44) we find that by multiplying (2.62) with k and summing (2.62) and (2.44), the term involving $\boldsymbol{\tau}$ is cancelled. Therefore, we deduce

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\|\vec{u}\|^2 + k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right]\\&+\frac{\gamma}{\mathrm{Re}}\|\nabla\vec{u}\|^2 + \frac{2ka_0}{\mathrm{Wi}}\int_{\Omega\times D} M|\nabla_q\sqrt{\hat{\psi}}|^2\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \le \int_\Omega \vec{f}\cdot\vec{u}\,\mathrm{d}\vec{x}.\end{aligned} \tag{2.63}$$

Thanks to Poincaré's inequality there exists a positive constant P, such that

$$\|\vec{u}\| \le P\|\nabla\vec{u}\|. \tag{2.64}$$

Also the right-hand side of (2.63) can be bounded further by the following:

$$\int_\Omega \vec{f}\cdot\vec{u}\,\mathrm{d}\vec{x} \le \frac{\gamma}{2P^2\mathrm{Re}}\|\vec{u}\|^2 + \frac{P^2\mathrm{Re}}{2\gamma}\|\vec{f}\|^2. \tag{2.65}$$

Here we used the fact that $ab \le \frac{\epsilon}{2}a^2 + \frac{1}{2\epsilon}b^2$, for any $\epsilon > 0$. Using the results from (2.29), (2.63), (2.64) and (2.65) we have that

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\|\vec{u}\|^2 + k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right]\\&+\frac{\gamma}{P^2\mathrm{Re}}\|\vec{u}\|^2 + \frac{ka_0\delta}{\mathrm{Wi}}\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \le \int_\Omega \vec{f}\cdot\vec{u}\,\mathrm{d}\vec{x}\\&\le \frac{\gamma}{2P^2\mathrm{Re}}\|\vec{u}\|^2 + \frac{P^2\mathrm{Re}}{2\gamma}\|\vec{f}\|^2.\end{aligned} \tag{2.66}$$

Rearranging (2.66) we deduce that

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{1}{2}\|\vec{u}\|^2 + k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right]\\&+\min\left(\frac{\gamma}{P^2\mathrm{Re}}, \frac{a_0\delta}{\mathrm{Wi}}\right)\left[\frac{1}{2}\|\vec{u}\|^2 + k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi})\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right]\\&\le \frac{P^2\mathrm{Re}}{2\gamma}\|\vec{f}\|^2.\end{aligned} \tag{2.67}$$

Let us introduce the following notation:

$$\gamma_0 := \min\left(\frac{\gamma}{P^2\mathrm{Re}}, \frac{a_0\delta}{\mathrm{Wi}}\right), \tag{2.68a}$$

$$A(t) := \frac{1}{2}\|\vec{u}(t)\|^2 + k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi}(t))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}. \tag{2.68b}$$

Using the notations introduced above and multiplying (2.67) by $\exp(\gamma_0 t)$, we have that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\exp(\gamma_0 t)A(t)\right] \leq \frac{P^2\mathrm{Re}}{2\gamma}\exp(\gamma_0 t)\|\vec{f}\|^2. \tag{2.69}$$

Integrating (2.69) from time 0 to s yields:

$$\begin{aligned}&\int_0^s \frac{\mathrm{d}}{\mathrm{d}t}[\exp(\gamma_0 t)A(t)]\,\mathrm{d}t \leq \frac{P^2\mathrm{Re}}{2\gamma}\int_0^s \exp(\gamma_0 t)\|\vec{f}\|^2\mathrm{d}t \\ &\Rightarrow \exp(\gamma_0 s)A(s) - A(0) \leq \frac{P^2\mathrm{Re}}{2\gamma}\int_0^s \exp(\gamma_0 t)\|\vec{f}\|^2\mathrm{d}t. \end{aligned}\tag{2.70}$$

Multiplying (2.70) by $\exp(-\gamma_0 s)$, we have that

$$A(s) \leq \exp(-\gamma_0 s)A(0) + \frac{P^2\mathrm{Re}}{2\gamma}\int_0^s \exp(\gamma_0(t-s))\|\vec{f}\|^2\,\mathrm{d}t. \tag{2.71}$$

Observe that t is between 0 and s, therefore $0 < \exp(\gamma_0(t-s)) < 1$, which implies

$$A(s) \leq \exp(-\gamma_0 s)A(0) + \frac{P^2\mathrm{Re}}{2\gamma}\int_0^s \|\vec{f}\|^2\,\mathrm{d}t. \tag{2.72}$$

Substituting (2.68b) into (2.72), we have the following:

$$\begin{aligned}&\|\vec{u}(s)\|^2 + 2k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi}(s))\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x} \\ &\leq \exp(-\gamma_0 s)\left(\|\vec{u_0}\|^2 + 2k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0)\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right) + \frac{P^2\mathrm{Re}}{\gamma}\int_0^s \|\vec{f}\|^2\,\mathrm{d}t. \end{aligned}\tag{2.73}$$

Using the results from (2.36) and (2.73) and choosing $s = T$ we can conclude that

$$\begin{aligned}&\|\vec{u}(T)\|^2 + \frac{k}{|\Omega|}\|\hat{\psi}(T) - 1\|^2_{L^1_M(\Omega\times D)} \\ &\leq \exp(-\gamma_0 T)\left(\|\vec{u_0}\|^2 + 2k\int_{\Omega\times D} M\mathcal{F}(\hat{\psi}_0)\,\mathrm{d}\vec{q}\,\mathrm{d}\vec{x}\right) + \frac{P^2\mathrm{Re}}{\gamma}\int_0^s \|\vec{f}\|^2\,\mathrm{d}t, \end{aligned}\tag{2.74}$$

and that completes the proof. We note that the rate of decay $\gamma_0 = \min\left(\frac{\gamma}{P^2\mathrm{Re}}, \frac{a_0\delta}{\mathrm{Wi}}\right)$ depends on the viscosity of the solvent γ, the Reynolds number, the minimum eigenvalue of the matrix $\mathbf{A}$, the Weissenberg number, the Poincaré constant P and the Bakry–Emery constant δ.

2.3 Conclusion

In Section 2.1, we first proved the exponential decay of the solutions to the Fokker–Planck equation when $\vec{u}$ is taken to be identically zero; therefore we did not need to consider the Navier–Stokes equations. We focused on the weak formulation of the Fokker-Planck equation (2.16) and picked a special test function $\mathcal{F}'(\hat{\psi})$. Then we applied the logarithmic Sobolev inequality (2.26) and the Csiszár–Kullback inequality (2.33) to obtain the resulting bound (2.37).

In Section 2.2, we extended our proof to the case when the Navier–Stokes equations were taken into account. We first investigated the weak formulation of the Navier–Stokes equations (2.41) and then took the test function to be $\vec{u}$. Then we performed a similar analysis as before except that we had two additional terms: T_1 defined in (2.45) and T_2 defined in (2.46). We proved that $\mathrm{T}_1 = 0$ and T_2 is related to the polymeric extra stress tensor $\boldsymbol{\tau}$. We summed the results from the Navier–Stokes equations and the Fokker–Planck equation and applied the logarithmic Sobolev inequality and the Csiszár–Kullback inequality to complete the proof.

The numerical methods for this coupled system including the multilevel Monte Carlo method, the adaptive Euler–Maruyama method and the mixed finite element method will be introduced in Chapter 3.

Chapter 3

Numerical Methods

In this section, we will first illustrate the idea of the Multilevel Monte Carlo method and consider a simple numerical example in order to compare it with the standard Monte Carlo method. Next, we will consider computing the stress tensor for homogeneous flows using the Multilevel Monte Carlo. Then, we will describe the proposed numerical method for nonhomogeneous flows.

3.1 Multilevel Monte Carlo Method

In practice, one is usually interested in the expected value of some function g of the solution to the stochastic equation (1.71) at some fixed time T, i.e., $\mathbb{E}[g(S(T))]$. The Monte Carlo method estimates the expectation by taking the mean of N independent samples $g(\hat{S}(T))$, where each sample is obtained by simulating (1.71) with some numerical scheme:

$$\hat{Y} = \frac{1}{N}\sum_{i=1}^{N} g(\hat{S}^{(i)}(T)). \tag{3.1}$$

Here $\hat{S}(T)$ denotes the numerical solution of (1.71) at time T, and $S(T)$ is the exact solution of (1.71) at time T. Furthermore $\hat{S}^i(T), i = 1, \ldots, N$, is the i-th sample path.

The mean square error of the approximation $\hat{Y}$ is given by:

$$\begin{aligned}\mathbb{E}\left[(\hat{Y} - \mathbb{E}[g(S(T))])^2\right] &= \mathbb{E}\left[(\hat{Y} - \mathbb{E}[\hat{Y}] + \mathbb{E}[\hat{Y}] - \mathbb{E}[g(S(T))])^2\right] \\ &= \mathbb{E}\left[(\hat{Y} - \mathbb{E}[\hat{Y}])^2\right] + \Big(\mathbb{E}[\hat{Y}] - \mathbb{E}[g(S(T))]\Big)^2. \end{aligned} \tag{3.2}$$

The mean square error is hence split into two terms. The first term is the variance of the Monte Carlo estimator $\mathbb{V}[\hat{Y}]$ and the second is the square of the bias or weak error of the numerical approximation. Let us take a look at the variance of the estimator:

$$\begin{aligned}\mathbb{V}[\hat{Y}] &= \mathbb{V}\left[\frac{1}{N}\sum_{i=1}^{N} g(\hat{S}^{(i)}(T))\right] \\ &= \frac{1}{N^2}\mathbb{V}\left[\sum_{i=1}^{N} g(\hat{S}^{(i)}(T))\right] \\ &= \frac{1}{N}\mathbb{V}\left[g(\hat{S}(T))\right]. \end{aligned} \tag{3.3}$$

The final equality indicates that the Monte Carlo variance is order $O(\frac{1}{N})$. If we use the Euler–Maruyama or the Milstein scheme to approximate $g(\hat{S}(T))$, the weak order will be proportional to the time step Δt as long as conditions (1.73)–(1.75) are satisfied [30], [35]. Then the mean square error of the Monte Carlo method is $O(\frac{1}{N}) + O(\Delta t^2)$.

In order to achieve a mean square error of size $O(\epsilon^2)$ (ϵ is a small positive number), the standard Monte Carlo method in conjunction with the Euler–Maruyama or the Milstein scheme requires $N = O(\epsilon^{-2})$ and $\Delta t = O(\epsilon)$. The total computational cost C is the number of simulations N times the number of time steps in each simulation $\frac{T}{\Delta t}$, so that $C = O(\epsilon^{-3})$.

Notice that the mean square error depends on the number of samples N, the variance of the sample $\mathbb{V}[g(\hat{S}(T)]$ and the weak error. The weak error can be reduced by decreasing the time step Δt, however it takes more time steps to perform the simulation on a fine mesh. The mean square error can also be reduced by increasing the number of samples N, however the Monte Carlo convergence rate is slow and the resulting error reduction may therefore be small. If one can make the sample variance very small, then one will not need to perform a large number of simulations, which will reduce the computational cost. Therefore variance reduction techniques are usually applied to the Monte Carlo method. The Multilevel Monte Carlo method generalises the idea of the variance reduction method and under some conditions reduces the total computational cost from $O(\epsilon^{-3})$ to $O(\epsilon^{-2}(\log \epsilon)^2)$ or $O(\epsilon^{-2})$ [23], [26].

Consider a sequence of numerical simulations $g(\hat{S}_0(T)), g(\hat{S}_1(T)), \ldots, g(\hat{S}_L(T))$ with different time steps $\Delta t_l = \frac{T}{R^l}$, $l = 0, 1, \ldots, L$, where R is a positive integer called refinement factor. We are interested in computing on the finest level $\mathbb{E}[g(\hat{S}_L(T))]$, and it is obvious that this quantity can be represented as a telescoping sum

$$\mathbb{E}[g(\hat{S}_L(T))] = \mathbb{E}\left[g(\hat{S}_0(T))\right] + \sum_{l=1}^{L} \mathbb{E}\left[g(\hat{S}_l(T)) - g(\hat{S}_{l-1}(T))\right]. \tag{3.4}$$

The multilevel Monte Carlo method uses independent estimations at each level and combines them according to equation (3.4). The idea behind this method is to control the variance of $g(\hat{S}_l(T)) - g(\hat{S}_{l-1}(T))$ in order to move most of the work from fine level to coarse level, which is much cheaper to simulate. The smaller the variance $\mathbb{V}\left[g(\hat{S}_l(T)) - g(\hat{S}_{l-1}(T))\right]$ is, the fewer samples N are required. The way to control the variance is to use the same Brownian path to approximate $g(\hat{S}_l(T))$ and $g(\hat{S}_{l-1}(T))$. Then, the difference between them will be small thanks to strong (pathwise) convergence.

Let $\hat{Y}_0$ be a Monte Carlo estimator for $\mathbb{E}[g(\hat{S}_0(T))]$, i.e., $\hat{Y}_0 = \frac{1}{N_0}\sum_{i=1}^{N_0} g(\hat{S}_0^{(i)}(T))$. Similarly, let $\hat{Y}_l$ be a Monte Carlo estimator for $\mathbb{E}\left[g(\hat{S}_l(T)) - g(\hat{S}_{l-1}(T))\right]$ using N_l simulations. Hence the multilevel Monte Carlo estimator for $\mathbb{E}[g(\hat{S}_L(T))]$ is the sum of the individual estimators $\hat{Y}_l$, i.e.,

$$\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l. \tag{3.5}$$

The total variance of the multilevel Monte Carlo estimator is

$$\begin{aligned} \mathbb{V}[\hat{Y}] &= \mathbb{V}\left[\sum_{l=0}^{L} \hat{Y}_l\right] \\ &= \sum_{l=0}^{L} \mathbb{V}[\hat{Y}_l] \\ &= \sum_{l=0}^{L} N_l^{-1} V_l, \end{aligned} \tag{3.6}$$

where $V_0 = \mathbb{V}[g(\hat{S}_0(T))]$, $\quad V_l = \mathbb{V}\left[g(\hat{S}_l(T)) - g(\hat{S}_{l-1}(T))\right]$, $\quad 0 < l \leq L$.

Recall that the mean square error is split into the weak error and the variance of the estimator. To make the mean square error less than ϵ^2, we can choose L such that the weak error $\mathbb{E}[g(\hat{S}_L(T))] - \mathbb{E}[g(S(T))]$ is less than $\frac{\epsilon}{\sqrt{2}}$, and choose N_l to make the total variance less than $\frac{\epsilon^2}{2}$. If the numerical method for approximating the SDE has first weak order, the time step Δt_L should be $O(\epsilon)$ to make the weak error less than $\frac{\epsilon}{\sqrt{2}}$. Therefore the associated number of levels is $L = \lceil \frac{\log \epsilon^{-1}}{\log R} \rceil + O(1)$.

For the Euler–Maruyama scheme and a Lipschitz continuous function g under conditions (1.73)–(1.75), one can prove that V_l decays like $O(\Delta t_l)$, [23]. If we choose $N_l = O(\epsilon^{-2} L \Delta t_l)$, then the variance of the estimator $\hat{Y}$ will be

$$\begin{aligned} \mathbb{V}[\hat{Y}] &= \sum_{l=0}^{L} N_l^{-1} V_l \\ &= \sum_{l=0}^{L} O(\epsilon^2 L^{-1} \Delta t_l^{-1}) \cdot O(\Delta t_l) \\ &= O(\epsilon^2). \end{aligned} \tag{3.7}$$

The total computational cost is the sum of the individual costs at each level:

$$C = \sum_{l=0}^{L} N_l \Delta t_l^{-1}. \tag{3.8}$$

Hence the cost of the multilevel Monte Carlo method for achieving mean square error ϵ^2 is

$$\begin{aligned} C &= \sum_{l=0}^{L} N_l \Delta t_l^{-1} \\ &= \sum_{l=0}^{L} O(\epsilon^{-2} L \Delta t_l) \cdot \Delta t_l^{-1} \\ &= O(\epsilon^{-2} L^2) \\ &= O(\epsilon^{-2} (\log \epsilon)^2). \end{aligned} \tag{3.9}$$

The 2008 [23] and 2015 paper [24] by Giles include a complexity estimate theorem for the multilevel Monte Carlo method for more general circumstances. The theorem

states that if there exist independent estimators $\hat{Y}_l$ based on N_l Monte Carlo samples and positive constants $\alpha \geq \frac{1}{2}$, β, c_1, c_2, c_3, c_4, L such that

1.
$$\mathbb{E}\left[g(\hat{S}_l(T)) - g(S(T))\right] \leq c_1 \Delta t_l^{\alpha}, \tag{3.10a}$$

2.
$$\mathbb{E}[\hat{Y}_l] = \begin{cases} \mathbb{E}\left[g(\hat{S}_0(T))\right] & l = 0, \\ \mathbb{E}\left[g(\hat{S}_l(T)) - g(\hat{S}_{l-1}(T))\right] & l > 0, \end{cases} \tag{3.10b}$$

3.
$$\mathbb{V}[\hat{Y}_l] \leq c_2 N_l^{-1} \Delta t_l^{\beta}, \tag{3.10c}$$

4. C_l, the computational cost of $\hat{Y}_l$, is bounded by

$$C_l \leq c_3 N_l \Delta t_l^{-1}, \tag{3.10d}$$

then the overall cost C of the multilevel Monte Carlo estimator

$$\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l,$$

with mean square error ϵ^2, is bounded by

$$C \leq \begin{cases} c_4 \epsilon^{-2}, & \beta > 1, \\ c_4 \epsilon^{-2} (\log \epsilon)^2, & \beta = 1, \\ c_4 \epsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1. \end{cases} \tag{3.11}$$

This theorem is rigorously proved in Section 3 of [23]. The interested reader is referred to that paper for details.

The mean square error relies on the weak error. Condition 1 of the theorem ensures the weak convergence of the approximation. The number α is the weak convergence order of the numerical scheme. As long as Condition 1 is satisfied, we are guaranteed to find the finest level L for which the weak error is order ϵ. Condition 2 is used to make sure that the telescoping sum is valid (equation (3.4)). Condition 3 is the most important one. The variance of $\hat{Y}_l$ should decay as the level number increases. We

control the variance of the estimator so that we can reduce the computational cost. The number β plays a very important role here: it represents the convergence rate of V_l. A bigger β will produce better results, because a large decay rate indicates that most of the work is moved to the coarse level, and for $\beta = 1$ the work is evenly distributed across all levels. Recall that the multilevel Monte Carlo method uses the same Brownian path to simulate $g(\hat{S}_l(T))$ and $g(\hat{S}_{l-1}(T))$; hence strong convergence will ensure that the difference between them is small. In the case of a Lipschitz function g, β will be twice the strong convergence order of the numerical method, however in general fast strong convergence might not imply fast decay of V_l. In [22], the Milstein scheme is used to further reduce the cost to $O(\epsilon^{-2})$ for a Lipschitz function g to achieve a mean square error $O(\epsilon^2)$ because the Milstein method has better strong convergence order than the Euler–Maruyama method. However for the standard Monte Carlo method, both Euler–Maruyama and Milstein require $O(\epsilon^{-3})$ operations.

The algorithm contains the following steps:

1. We start with $L = 0$ and obtain an estimation $\hat{Y}_0$ based on an initial set of $N_0 = 10^4$ samples.

2. Calculate the variance V_L.

3. Calculate the optimal number of simulations N_L.

4. If $N_L > N_0$, we evaluate the extra samples and update the results; if $N_l < N_0$, we keep the results.

5. Terminate the algorithm when the weak error is less than $\frac{\epsilon}{\sqrt{2}}$; otherwise proceed to the next level ($L := L + 1$).

3.1.1 Numerical Examples

We will perform a simple numerical experiment using both the standard and the multilevel Monte Carlo method. Consider the linear stochastic differential equation

$$\mathrm{d}S(t) = \lambda S(t)\,\mathrm{d}t + \mu S(t)\,\mathrm{d}W(t), \tag{3.12}$$

Mean square error ϵ^2	$\hat{Y}_{MLMC}$	Computing time (MLMC)	$\hat{Y}_{MC}$	Computing time
0.1^2	0.0955	0.002s	0.0953	0.002s
0.01^2	0.1090	0.003s	0.1077	0.03s
0.001^2	0.1039	0.01s	0.1046	2.74s
0.0001^2	0.1044	0.819s	0.1045	9.2s

Table 3.1: Numerical results for $\mathbb{E}[e^{-\lambda}\max(0, S(1)-1)]$

with $S(0) = 1$, $\lambda = 0.05$ and $\mu = 0.2$. We are interested in computing

$$\mathbb{E}[e^{-\lambda}\max(0, S(1)-1)].$$

The results are calculated using the Euler–Maruyama method and the refinement number $R = 4$. Table 3.1 shows the expected value based on the various values of ϵ for both the multilevel Monte Carlo and the standard Monte Carlo method. The time step Δt for the standard Monte Carlo method was chosen to be the same as the time step in the last level of the multilevel Monte Carlo simulation. The number of samples was chosen to be of the order of $O(\epsilon^{-2})$. From the table we can see that for a large error the computing time is similar, however as the error decreases the difference is significant. It takes almost ten times longer to use the standard Monte Carlo method when $\epsilon = 0.0001$.

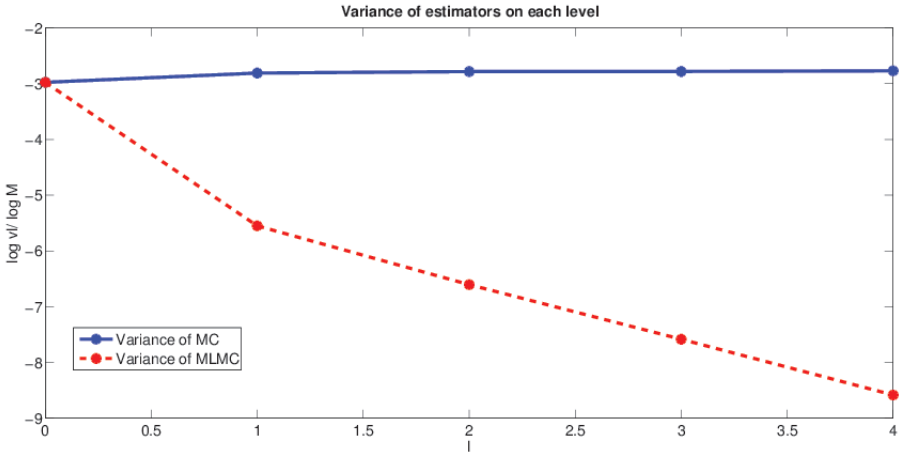


Figure 3.1: Euler–Maruyama scheme with 10^5 sample paths used on each level

Figure 3.1 shows $\mathbb{V}[g(\hat{S}_l(T))]$ in blue and $\mathbb{V}[g(\hat{S}_l(T)) - g(\hat{S}_{l-1}(T))]$ in red based on

10^5 samples on each level. The logarithm of the variance is plotted versus the level so that the slope of the red line represents the decay rate of $\mathbb{V}[\hat{Y}_l] = O(\Delta t)$. The strong convergence order for the Euler–Maruyama method is $O(\Delta t^{\frac{1}{2}})$, which in turn gives $\beta = 1$ as explained before. If we use the standard Monte Carlo method, the variance of the estimator shown by the blue line is almost 1000 times larger than with the multilevel Monte Carlo estimator when $l = 4$ with the same time step.

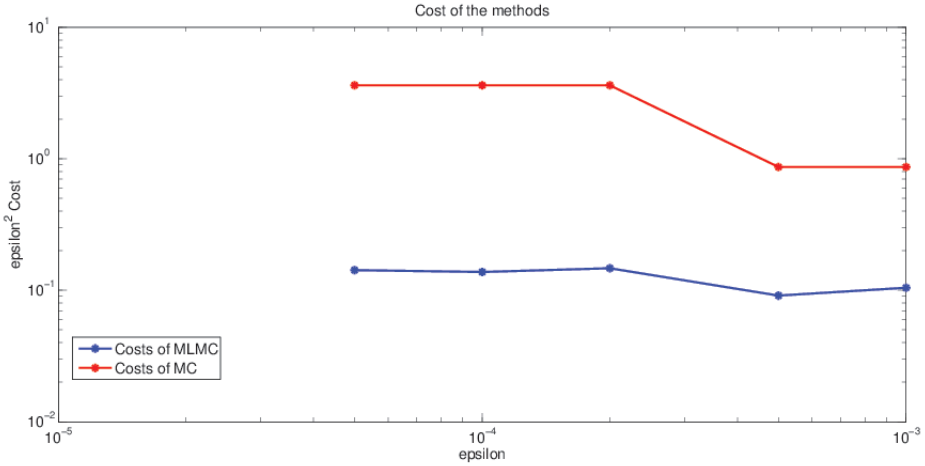


Figure 3.2: Comparison of the computational cost of MLMC and MC

The total computational cost of the multilevel Monte Carlo method is the number of time steps times the number of simulations on all levels:

$$C = N_0 + \sum_{l=1}^{L} N_l (R^l + R^{l-1}). \tag{3.13}$$

The cost of the standard Monte Carlo method is on the other hand given by

$$C^* = \sum_{l=0}^{L} N_l^* R^l, \tag{3.14}$$

where $N_l^* = 2\epsilon^{-2}\mathbb{V}[g(\hat{S}_l(T))]$ in order to make sure that the total variance is less than $\frac{1}{2}\epsilon^2$. The level L is the same as with the multilevel Monte Carlo method so that the weak error is also less than $\frac{1}{2}\epsilon^2$. The summation in (3.14) is used to apply the same heuristic termination criterion as the multilevel method [23].

Figure 3.2 is the log-log plot of $\epsilon^2 C$ against ϵ for the multilevel Monte Carlo method and $\epsilon^2 C^*$ for the standard Monte Carlo method against the tolerance ϵ. It is clear that the multilevel Monte Carlo method is much cheaper. There is a change of the value $\epsilon^2 C$ in the figure corresponding to the fact that L is changed. We see a very slow increase in the cost of the multilevel Monte Carlo method, while for the standard Monte Carlo method, this change is much larger.

3.2 Stochastic Simulation for Homogeneous Flows

In this section, we will consider homogeneous flows for which the velocity gradient $\nabla_x \vec{u}(\vec{x}, t) := \boldsymbol{\kappa}$ ($\kappa_{ij} = \partial_j u_i(\vec{x}, t)$) is a constant matrix throughout the physical domain Ω. In that case, the probability density function ψ and the Fokker–Planck equation do not depend on the space variable $\vec{x}$. Furthermore, we will focus our attention on the dumbbell model in this section. Therefore this section is concerned with the numerical approximation of the following stochastic differential equation:

$$\mathrm{d}\vec{Q}(t) = \left(\boldsymbol{\kappa} \cdot \vec{Q}(t) - \frac{1}{2\mathrm{Wi}} \vec{F}(\vec{Q}(t))\right) \mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}} \mathrm{d}\vec{W}(t), \quad t \in (0, T], \tag{3.15}$$

$$\vec{Q}(0) \sim \psi_0(\vec{q}), \tag{3.16}$$

where $\vec{F}(\vec{Q}(t)) = \frac{\vec{Q}(t)}{1 - \frac{|\vec{Q}(t)|^2}{b}}$, $\vec{Q}(t) = (Q_1(t), Q_2(t))^T$ represents the dumbbell orientation at time $t \in [0, T]$ in a 2D flow. Let us denote $\vec{u} := (u, v)^T$; then the velocity gradient $\boldsymbol{\kappa}$ is the following:

$$\boldsymbol{\kappa} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}.$$

Our target is to calculate the extra stress tensor $\boldsymbol{\tau}$ according to the Kramers expression by taking the expected value of the numerical solution to the stochastic differential equation (3.15) at a fixed time T, i.e., $\boldsymbol{\tau}(T) = \mathbb{E}[\vec{Q}(T) \otimes \vec{F}(\vec{Q}(T))]$ given $\boldsymbol{\kappa}$. The first method we suggest to simulate the stochastic differential equation (3.15) is the Euler–Maruyama method and the multilevel Monte Carlo method will be used to estimate the expectation.

3.2.1 The Euler–Maruyama Method

First, we discretize the time interval into m pieces with a uniform time step $\Delta t = \frac{T}{m}$. Let $\vec{Q}_n$ be the numerical approximation at time $t_n = n\Delta t, \quad n = 0, 1, \ldots, m$. Then, the Euler–Maruyama method for equation (3.15) takes the following form:

$$\vec{Q}_{n+1} = \vec{Q}_n + (\boldsymbol{\kappa} \cdot \vec{Q}_n - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}_n))\Delta t + \sqrt{\frac{1}{\mathrm{Wi}}}\Delta\vec{W}_n, \quad \text{for } n = 0, 1, \ldots, m, \quad (3.17)$$

$$\vec{Q}_0 \sim \psi_0(\vec{q}). \quad (3.18)$$

Here $\Delta\vec{W}_n = \vec{W}(t_{n+1}) - \vec{W}(t_n) = \sqrt{\Delta t}\vec{N}(0,1)$, where $\vec{N}(0,1)$ is a two-dimensional vector of independent normally distributed random variables with mean zero and variance one. The initial value $\vec{Q}_0$ is randomly generated from the initial condition of ψ. The componentwise description of (3.17) is the following:

$$\begin{aligned}\begin{pmatrix} Q_{1,n+1} \\ Q_{2,n+1} \end{pmatrix} = \begin{pmatrix} Q_{1,n} \\ Q_{2,n} \end{pmatrix} + \begin{pmatrix} \frac{\partial u}{\partial x}Q_{1,n} + \frac{\partial u}{\partial y}Q_{2,n} \\ \frac{\partial v}{\partial x}Q_{1,n} + \frac{\partial v}{\partial y}Q_{2,n} \end{pmatrix}\Delta t \\ - \frac{1}{2\mathrm{Wi}}\frac{1}{1 - \frac{Q_{1,n}^2 + Q_{2,n}^2}{b}}\begin{pmatrix} Q_{1,n} \\ Q_{2,n} \end{pmatrix}\Delta t + \sqrt{\frac{\Delta t}{\mathrm{Wi}}}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}\end{aligned} \quad (3.19)$$

for $n = 0, 1, \ldots, m$ and $z_1, z_2 \sim N(0,1)$.

This method suffers from a serious stability problem since the drift is not globally Lipschitz continuous in the stochastic differential equation. In our model, the drift term contains the FENE spring force $\vec{F}(\vec{Q})$, which tends to infinity as $|\vec{Q}| \to \sqrt{b}$. In particular, $\vec{F}$ is locally Lipschitz continuous, but it exhibits superlinear growth and is not globally Lipschitz continuous on its domain of definition. The Euler–Maruyama method is known to converge to the exact solution of a stochastic differential equation with globally Lipschitz continuous drift and diffusion coefficients. Recent results in the literature have extended this convergence to coefficients that are locally Lipschitz continuous and both the exact and numerical solution have a bounded pth moment for $p > 2$ [31], [59], [60]. However the bounded moment assumption might not hold for the forward Euler method. In [31], the authors proved that the exact solution has bounded moments based on the following conditions: the volatility is globally Lipschitz and the drift satisfies a one-sided Lipschitz condition. They were unable to establish the

boundedness of the moments for the numerical solution using the Euler–Maruyama method. In fact for superlinearly growing coefficients, it has been proved in [33] that, for a large class of SDEs with non-globally Lipschitz continuous coefficients, the Euler–Maruyama approximation converges neither in the strong mean-square sense nor in the numerically weak sense to the exact solution at a finite time point. Even worse, the difference of the exact solution and the numerical approximation at a finite time point was shown there to diverge to infinity in the strong mean-square sense and in the numerically weak sense. While the paper [33] does not cover the FENE SDE considered here, it gives strong indication that the Euler–Maruyama approximation of the FENE SDE may be similarly badly behaved. Also, for a large time step, the numerical approximation might exit the configuration domain D, which is not allowed.

In practice, this last problem might not occur when we use an extremely small time step because of the finite precision representation of floating point numbers in the computer. If we generate a random number from $N(0,1)$, the probability of generating a large outcome is very small and the computer might not generate it. This method is easy to implement, however it is time-consuming due to the small time step used. An alternative approach is an adaptive time stepping method.

Below is the numerical algorithm of the multilevel Monte Carlo method combined with the standard Euler–Maruyama method. L is the finest level starting from $L = 0$ and we increase L by one every time when we move forward to the next level until we have convergence and $L > 2$. For each level $l = 0, \ldots, L$, we will first use the Euler–Maruyama method to simulate the SDE using an initial set of N_0 samples and estimate the variances V_l and the Monte Carlo estimator $\hat{Y}_l$, then we will calculate the sample size N_l needed for each level and if it is larger than N_0, we need to update the results by simulating more SDEs using a set of $N_l - N_0$ samples.

Algorithm 1: This algorithm is the combination of the Euler–Maruyama method for solving (3.15) and the multilevel Monte Carlo method to find the expected value.

Data: constant $\boldsymbol{\kappa}$, initial probability density distribution $\psi_0(\vec{q})$, initial number of samples N_0, mean square error ϵ^2, refinement factor R
Result: $\mathbb{E}[\vec{Q}(T) \otimes \vec{F}(\vec{Q}(T))]$
We start with level $L = -1$ and initialize all the variables;
while not converged or $L < 2$ **do**
    $L = L + 1$;
    for k=1:N_0 **do**
        if $L = 0$ **then**
            Sample the initial value $\vec{Q}(0)$ from $\psi_0(\vec{q})$ and simulate the stochastic differential equation (3.15) according to (3.18) using $\Delta t = T$;
        else
            Sample initial value $\vec{Q}(0)$ from $\psi_0(\vec{q})$ for both level L and level $L-1$;
            $nf := R^L$, $nc := R^{L-1}$, $\Delta t_f := \frac{T}{nf}$, $\Delta t_c := \frac{T}{nc}$;
            for i=1:nc **do**
                $\Delta \vec{W}_c := \vec{0}$;
                for j=1:R **do**
                    $\Delta \vec{W}_f := \sqrt{\Delta t_f} \times \vec{N}(0,1)$;
                    $\Delta \vec{W}_c := \Delta \vec{W}_c + \Delta \vec{W}_f$;
                    Simulate the stochastic differential equation (3.15) according to (3.18) at the fine level using Δt_f and $\Delta \vec{W}_f$;
                end
                Simulate the stochastic differential equation (3.15) according to (3.18) at the coarse level using Δt_c and $\Delta \vec{W}_c$;
            end
        end
    end
    Estimate the variance V_L using initial set of N_0 samples, the Monte Carlo estimator $\hat{Y}_L$ and the optimal number of samples needed N_L;
    for l=0:L **do**
        if The optimal number of samples is bigger than the current sample size **then**
            Evaluate extra samples at each level as needed and update the results;
        end
    end
    Test for convergence;
end
Obtain the multilevel Monte Carlo estimator $\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l$;

3.2.2 Adaptive Time Stepping Method

Unlike in the standard Euler–Maruyama method, here the time step may vary from time step to time step in an adaptive fashion. The algorithm is still an Euler type method, but the time step is varied, based on our numerical approximation. The discretisation of our SDE leads to the following equation:

$$\vec{Q}_{n+1} = \vec{Q}_n + (\boldsymbol{\kappa} \cdot \vec{Q}_n - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}_n))\Delta t_n + \sqrt{\frac{1}{\mathrm{Wi}}}\Delta\vec{W}_n, \tag{3.20}$$

$$t_{n+1} = t_n + \Delta t_n. \tag{3.21}$$

From (3.20), we can see that the numerical approximation $\vec{Q}_{n+1}$ at the next time step depends on the current approximation $\vec{Q}_n$, the time step Δt_n and the random variable $\Delta\vec{W}_n$. Therefore we can force our numerical solution to remain within the domain of definition of $\vec{F}$ by selecting the time step properly if we can control the size of the random part. With the FENE model, the length of the configuration vector can not exceed $\sqrt{b}$, therefore we need to put the following restriction on the time step:

$$\begin{aligned} &\sqrt{\frac{1}{\mathrm{Wi}}}\Delta\vec{W}_n = \sqrt{\frac{\Delta t_n}{\mathrm{Wi}}}z_i \leq \sqrt{b} - \|\vec{Q}_n\| \\ &\Rightarrow \Delta t_n \leq b\mathrm{Wi}\left(\frac{1}{z_i}\left(1 - \frac{\|\vec{Q}_n\|}{\sqrt{b}}\right)\right)^2, \quad i = 1, 2, \end{aligned} \tag{3.22}$$

where z_i here is the same as in the equation (3.19). From (3.22) we notice that the time step depends on b. If b is small we need a smaller time step, which leads to an increase in the computational cost, and therefore the complexity of this problem using the adaptive time stepping method above depends on the radius of the configuration domain.

From the previous section, we know that z_1, z_2 follow the standard normal distribution. Assuming that the random variables z_1, z_2 have maximum magnitude 10 in a finite precision computer, we pick the time step Δt_n to be

$$\Delta t_n = b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{\|\vec{Q}_n\|}{\sqrt{b}}\right)\right)^2. \tag{3.23}$$

Then, the numerical approximation at the next step will be controlled and in fact it will remain inside the configuration domain D and never reach the boundary as long as $\boldsymbol{\kappa}$ is not large. If we have a very large value for $\boldsymbol{\kappa}$, then we also need to put a restriction on the time step to control the term $(\boldsymbol{\kappa} \cdot \vec{Q}_n)\Delta t_n$; therefore we can pick the time step to be the norm square of the configuration vector over the norm square of the drift:

$$\Delta t_n = C_{\Delta t} \frac{\|\vec{Q}_n\|^2}{\|\boldsymbol{\kappa} \cdot \vec{Q}_n - \frac{1}{2\mathrm{Wi}} \vec{F}(\vec{Q}_n)\|^2}, \tag{3.24}$$

where $C_{\Delta t}$ is some positive constant we need to choose. Then we can set the time step to be the minimum of (3.23) and (3.24) to make sure both restrictions are achieved:

$$\Delta t_n = \min \left(b\mathrm{Wi} \left(\frac{1}{10} \left(1 - \frac{\|\vec{Q}_n\|}{\sqrt{b}} \right) \right)^2, C_{\Delta t} \frac{\|\vec{Q}_n\|^2}{\|\boldsymbol{\kappa} \cdot \vec{Q}_n - \frac{1}{2\mathrm{Wi}} \vec{F}(\vec{Q}_n)\|^2} \right). \tag{3.25}$$

The implementation of this method is as simple as that of the classical Euler–Maruyama method, however the total cost is reduced because the time step does not need to be small when the trajectory is far away from the boundary. Moreover, most of the stochastic trajectories are controlled to remain within the configuration domain D. If we use the Euler–Maruyama method with a fixed time step, we can not make sure that the trajectories stay within the domain.

In practice, the adaptive time stepping requires an additional constraint to work. When the numerical approximation is very close to the boundary, the corresponding time step will be too small to be recognized by the computer due to the finite precision representation. Therefore we can not move forward because the time step is set to be zero by the computer. A remedy is to clamp the trajectory before it gets near the boundary. We first set a safety margin which is a small perturbation of the original domain with radius $\sqrt{b} - \delta$ for some small $\delta \ll 1$. When the trajectory goes beyond this margin, we force it back by setting $\vec{Q} = \frac{\sqrt{b}-\delta}{|\vec{Q}|}\vec{Q}$. We believe that the additional error due to the clamping is negligible. One computational confirmation of this is that we performed ten thousands simulations for the 1-D FENE stochastic differential equation with $\kappa = 1, \mathrm{Wi} = 1, b = 10$ and $\delta = 10^{-5}$. Only five simulations went beyond $\sqrt{b} - \delta$.

From the analytical point of view, the adaptive time stepping method has the following shortcoming: we use the clamping strategy to avoid a small time step and the analysis for the clamping might be difficult. However, based on the numerical experiments reported above, the probability of this seems to be small, and we can simply ignore them in practice. Another concern is that we assumed that every random variable with normal distribution $N(0,1)$ has maximum magnitude 10 in the computer, which is not necessarily true.

3.2.3 Polymeric Stress Tensor

The polymeric stress tensor involves the expectation of the solution to the stochastic differential equation (3.15), according to (1.48):

$$\boldsymbol{\tau}(t) = C \begin{pmatrix} \mathbb{E}\left(\frac{Q_1^2(t)}{1-\frac{Q_1^2(t)+Q_2^2(t)}{b}}\right) - 1 & \mathbb{E}\left(\frac{Q_1(t)Q_2(t)}{1-\frac{Q_1^2(t)+Q_2^2(t)}{b}}\right) \\ \mathbb{E}\left(\frac{Q_1(t)Q_2(t)}{1-\frac{Q_1^2(t)+Q_2^2(t)}{b}}\right) & \mathbb{E}\left(\frac{Q_2^2(t)}{1-\frac{Q_1^2(t)+Q_2^2(t)}{b}}\right) - 1 \end{pmatrix}, \tag{3.26}$$

where $C = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{ReWi}}$ is a constant and Q_1, Q_2 are the two components of the vector $\vec{Q}$ in the 2D case. The (1,2) component is the same as the (2,1) component; therefore in practice we only need to compute three components of $\boldsymbol{\tau}$.

For the Monte Carlo method, we will generate N independent values $\vec{Q}^{(s)}(t), \quad s = 1, \ldots, N$, and use the numerical scheme (3.17) or (3.20) to simulate the stochastic differential equation (3.15). We then compute the three components of the polymeric stress tensor $\boldsymbol{\tau}$ at time t by taking the average of N outcomes. For the multilevel Monte Carlo method, we need to pay more attention to the adaptive time stepping method because the time grids are not nested. The time step in the multilevel Monte Carlo method at level l is the following:

$$\Delta t_{n,l} = \Delta t_n R^{-l}, \quad l = 0, 1, \ldots, L. \tag{3.27}$$

Figure 3.3 shows two different simulation times using the standard Euler–Maruyama method and the adaptive time stepping method. We can see that the time grids are nested for the Euler–Maruyama method while the time grids for the adaptive method are not. Using the figure as an example, for the standard Euler–Maruyama method

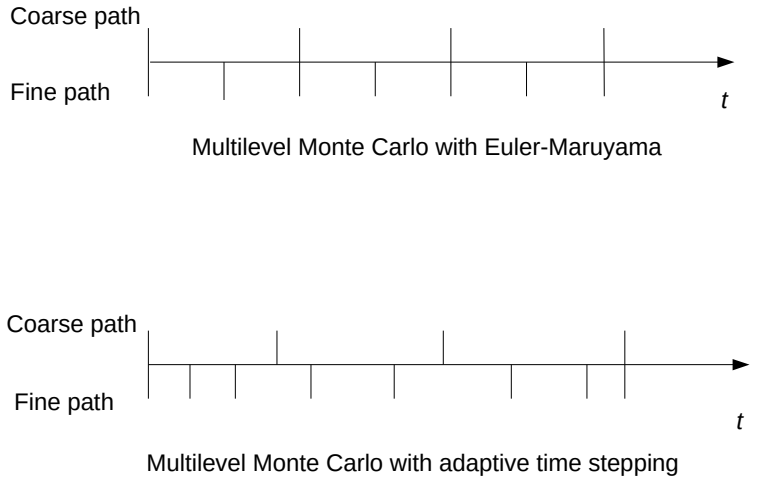


Figure 3.3: Simulation times for the multilevel Monte Carlo method with the Euler–Maruyama method and the adaptive time stepping method

we can simulate the stochastic differential equation twice at the fine level and add two Brownian increments together to obtain the Brownian increments over the coarse level and repeat this process until time T. For the adaptive time stepping method, although the time grids are not nested, the time grid on the fine level still has approximately R times as many time steps as on the coarse level. Therefore we simulate the stochastic differential equation simultaneously for both the coarse and the fine level in order to make sure that the same Brownian path is used for the coarse and fine levels. The algorithm 2 is a single multilevel Monte Carlo path calculation for level $L \geq 0$ equipped with the adaptive time stepping method.

Algorithm 2: This algorithm is the single multilevel Monte Carlo simulation for level L for solving (3.15) using the Euler–Maruyama method with adaptive time stepping and clamping in the time interval $[0, T]$.

Data: constant $\boldsymbol{\kappa}$, initial probability density distribution $\psi_0(\vec{q})$, initial number of samples N_0, clamping factor δ, level L, refinement factor R
Result: $\hat{Y}_L$
Sample the initial value $\vec{Q}(0)$ from $\psi_0(\vec{q})$ and initialize the following variables;
$t := 0$, $t_f := 0$, $t_c := 0$, $\Delta t := 0$, $\Delta t_f := 0$, $\Delta t_c := 0$, $\Delta \vec{W}_f = \vec{0}$, $\Delta \vec{W}_c = \vec{0}$;
if $L = 0$ **then**
  while $t < T$ **do**
    Clamp the trajectories if they go beyond the safety margin;
    Calculate the time step Δt using (3.25);
    $\Delta t := \min(\Delta t, T - t)$;
    Simulate the stochastic differential equations (3.15) according to (3.20) using Δt;
    $t = t + \Delta t$;
  end
else
  $\Delta t_f := b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{|\vec{Q}(t)|}{\sqrt{b}}\right)\right)^2 R^{-L}$;
  $\Delta t_c := b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{|\vec{Q}(t)|}{\sqrt{b}}\right)\right)^2 R^{-L+1}$;
  $\Delta t := \min(\Delta t_f, \Delta t_c)$, $t_f := t_f + \Delta t_f$, $t_c := t_c + \Delta t_c$;
  while $t < T$ **do**
    $\Delta \vec{W} := \sqrt{\Delta t} \times \vec{N}(0, 1)$;
    $\Delta \vec{W}_f := \Delta \vec{W}_f + \Delta \vec{W}$, $\Delta \vec{W}_c := \Delta \vec{W}_c + \Delta \vec{W}$;
    if $t_f < t_c$ **then**
      $t := \min(t_f, T)$, Clamp the trajectories if needed;
      Simulate the fine path using time step Δt and $\Delta \vec{W}_f$;
      Compute new adapted fine path time step
      $\Delta t_f := b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{|\vec{Q}(t)|}{\sqrt{b}}\right)\right)^2 R^{-L}$, $t_f := t_f + \Delta t_f$,
      $\Delta t := \min(\Delta t_f, t_c - t, T - t)$, $\Delta \vec{W}_f = \vec{0}$;
    else
      $t := \min(t_c, T)$, Clamp the trajectories if needed;
      Simulate the coarse path using time step Δt and $\Delta \vec{W}_c$;
      Compute new adapted coarse path time step
      $\Delta t_c := b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{|\vec{Q}(t)|}{\sqrt{b}}\right)\right)^2 R^{-L+1}$, $t_c := t_c + \Delta t_c$,
      $\Delta t := \min(\Delta t_c, t_f - t, T - t)$, $\Delta \vec{W}_c = \vec{0}$;
    end
  end
end
Compute $\hat{Y}_L$ using the numerical solution at level L and $L - 1$;

1. Start with time $t = 0$, calculate the required time steps for both coarse and fine levels: $\Delta t_c, \Delta t_f$, set $t_c = \Delta t_c, t_f = \Delta t_f$, choose the minimum between Δt_c and Δt_f and set it to be Δt, set the Brownian increment $\Delta \vec{W}_f$ and $\vec{W}_c$ to be zero.

2. calculate the Brownian increment $\Delta \vec{W}$ using Δt and set $\Delta \vec{W}_f := \Delta \vec{W}_f + \Delta \vec{W}$, $\Delta \vec{W}_c := \Delta \vec{W}_c + \Delta \vec{W}$.

3. If $t_f < t_c$, set the current time $t = \min(t_f, T)$, update the fine path using Δt and $\Delta \vec{W}_f$, then reset $\vec{W}_f$ to be zero, calculate the next required time step Δt_f and set $t_f := t_f + \Delta t_f$, set Δt to be the minimum among Δt_f, $t_c - t$ and $T - t$, then go to step 5.

4. If $t_f \geq t_c$, set the current time $t = \min(t_c, T)$, update the coarse path using Δt and $\Delta \vec{W}_c$, then reset $\vec{W}_c$ to be zero, calculate the next required time step Δt_c and set $t_c := t_c + \Delta t_c$, set Δt to be the minimum among Δt_c, $t_f - t$ and $T - t$, then go to step 5.

5. Repeat steps 2 to 4 until time T.

The most important step is to make sure that the coarse path and the fine path share the same Brownian path, which is the key idea in the multilevel Monte Carlo method to reduce the variance. For more details about the non-nested adaptive grids in multilevel Monte Carlo computations, readers are referred to [25].

Remark 3.1 *Algorithm 2 shows only one path simulation, however we need to compute N different simulations. In practice the variables in the algorithm will be modified in order to simulate N simulations and some* `For`*-loops might be added. The way to achieve this in MATLAB is to change the time variables into vectors with N components and* $\vec{Q}$ *into a matrix.*

3.2.4 Stability and Strong Convergence

In this section, we will investigate the stability of both the numerical and the exact solution to the FENE stochastic differential equation and the strong convergence of the adaptive Euler–Maruyama method by making suitable assumptions. Recall that

the equation we are interested in is the following:

$$\mathrm{d}\vec{Q}(t) = \left(\boldsymbol{\kappa}\cdot\vec{Q}(t) - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}(t))\right)\mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}}\,\mathrm{d}\vec{W}(t), \quad t\in(0,T]. \tag{3.28}$$

The drift term of the stochastic differential equation involves the FENE spring force $\vec{F}(\vec{Q}(t)) = \frac{\vec{Q}(t)}{1-\frac{|\vec{Q}(t)|^2}{b}}$, which is only defined in the region $D = B(0,\sqrt{b})$. In order to carry out the analysis, we need to extend $\vec{F}$ to the whole space.

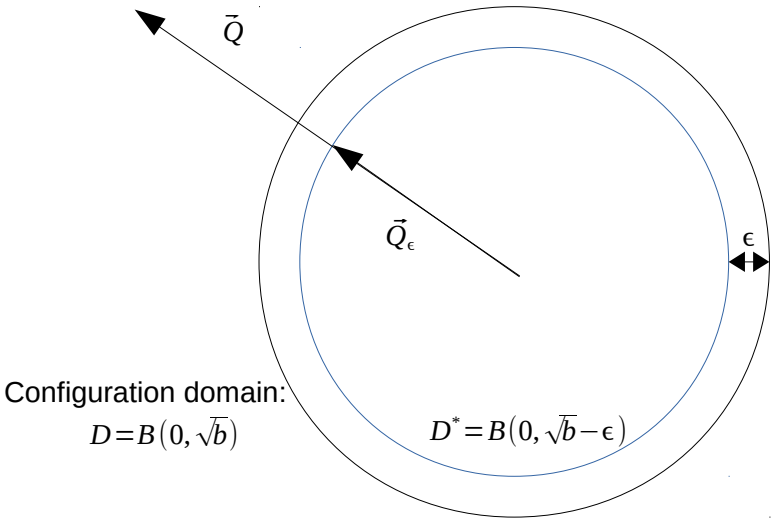


Figure 3.4: Extension of the FENE spring force

We consider a new region $D^\star$, which is smaller than the original one and the distance between them is $\epsilon \ll 1$ as shown in the figure. For all the variables which are inside the $D^\star$, we use the same expression for $\vec{F}$ as before, while for the variables which are outside this region, we first truncate $\vec{Q}$ to $\vec{Q}_\epsilon$, which is on the boundary of $D^\star$ pointing in the direction of $\vec{Q}$; then we use a linear function with positive derivative passing through the point $(\vec{Q}_\epsilon, \vec{F}(\vec{Q}_\epsilon))$ as the extension of the spring force. The mathematical expression for the extended FENE force $\vec{F}_\epsilon$ is the following:

$$\vec{F}_\epsilon(\vec{Q}) = \begin{cases} \vec{F}(\vec{Q}), & \|\vec{Q}\| < \sqrt{b}-\epsilon, \\ k\vec{Q} + \vec{F}(\vec{Q}_\epsilon) - k\vec{Q}_\epsilon, & \|\vec{Q}\| \geq \sqrt{b}-\epsilon, \end{cases} \tag{3.29}$$

where k is some positive number and $\vec{Q}_\epsilon = \frac{\vec{Q}}{\|\vec{Q}\|}(\sqrt{b}-\epsilon)$. The function $\vec{F}_\epsilon$ has bounded first derivative on the whole space, therefore it is globally Lipschitz continuous.

Remark 3.2 *We can choose the slope k so that the extended FENE force $\vec{F}_\epsilon$ is in C^1. Furthermore, we can also use a higher degree polynomial function as the extension. The extended FENE force is for the purpose of the analysis. In our computations the length of the spring $\|\vec{Q}\|$ is not allowed to stretch to infinity and we use the adaptive time stepping to make sure that $\|\vec{Q}\| < \sqrt{b} - \epsilon$.*

In order to show that the exact solution to the SDE with extended FENE force has a uniform pth moment bound for any $p > 1$, we introduce the following assumption.

Assumption 3.1 *Consider an n-dimensional stochastic differential equation (3.28) with extended FENE force $\vec{F}_\epsilon(\vec{Q})$, driven by an n-dimensional Brownian motion. We define $\vec{a}(\vec{Q}(t)) = \left(\boldsymbol{\kappa}\cdot\vec{Q}(t) - \frac{1}{2\,Wi}\vec{F}_\epsilon(\vec{Q}(t))\right)$ and $\boldsymbol{b} = \sqrt{\frac{1}{Wi}}\boldsymbol{I}$. Suppose that the drift term $\vec{a}:\mathbb{R}^n \to \mathbb{R}^n$ and volatility term $\boldsymbol{b}:\mathbb{R}^n \to \mathbb{R}^{n\times n}$ are both locally Lipschitz, and there exist some constants $\alpha, \beta > 0$ such that:*

$$\langle \vec{Q}, \vec{a}(\vec{Q})\rangle \le -\alpha\|\vec{Q}\|^2 + \beta \quad \forall \vec{Q} \in \mathbb{R}^n, \tag{3.30}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product in $\mathbb{R}^n$. Also the volatility term $\boldsymbol{b}$ is assumed to be bounded and non-degenerate:

$$\|\boldsymbol{b}(\vec{Q})\|^2 \le \beta \quad \forall \vec{Q} \in \mathbb{R}^n. \tag{3.31}$$

We need to verify that the extended FENE stochastic differential equation satisfies Assumption 3.1. Clearly the volatility term is constant, therefore it is globally bounded and (3.31) is satisfied. Before we verify the condition (3.30) for the drift term, there is an assumption about the size of the velocity gradient $\boldsymbol{\kappa}$, which is a important factor in the contractivity we will introduce later.

Assumption 3.2 *The velocity gradient $\boldsymbol{\kappa}$ is bounded by the Weissenberg number such that:*

$$\|\boldsymbol{\kappa}\|_F < \frac{1}{2\mathrm{Wi}}, \tag{3.32}$$

where $\|\boldsymbol{\kappa}\|_F$ stands for the Frobenius norm of the matrix.

Assumption 3.2 implies that there exists a constant $B > 0$ such that:

$$\|\boldsymbol{\kappa}\|_F \leq B < \frac{1}{2\mathrm{Wi}}. \tag{3.33}$$

Assumption 3.2 is very important because it determinates the probability density function ψ as illustrated in Figure 3.5 below. If Assumption 3.2 is satisfied, there will be only one peak in the probability density function near the central of the configuration domain D. On the other hand, if Assumption 3.2 is not satisfied, there will be double peaks near the boundary of the configuration domain D. The drift term is dominated by the spring force when Assumption 3.2 is satisfied and the spring force always tries to shrink the spring, therefore the trajectories of the SDE are all near the original. However, if we increase the size of the velocity gradient $\boldsymbol{\kappa}$ and the term $\boldsymbol{\kappa} \cdot \vec{Q}$ might be larger than the spring force, this will cause the spring to expand. We will perform one numerical experiment in Section 4.2.3 to illustrate this phenomenon and the results are shown in Figure 4.19.

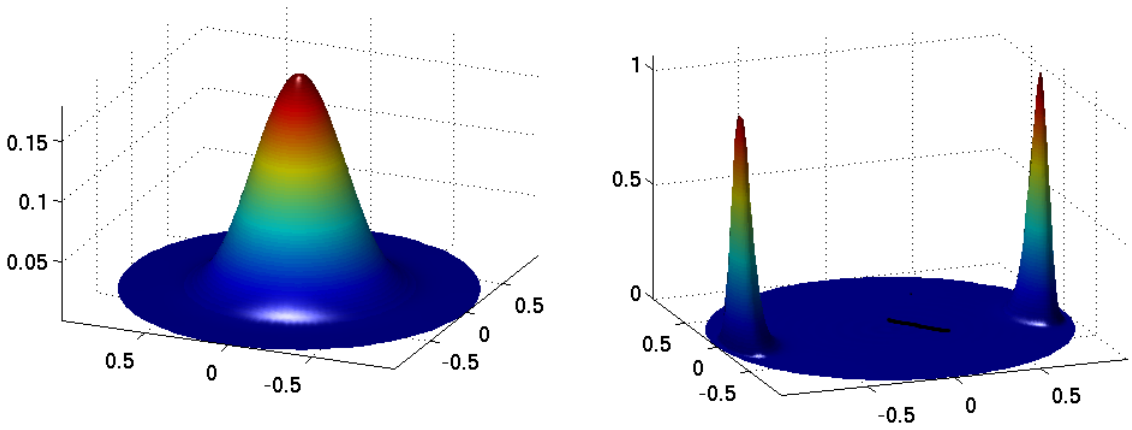


Figure 3.5: Examples of the probability density function ψ

According to [13], an analytical expression for the steady-state solution of the Fokker-Planck equation exists when $\boldsymbol{\kappa} = \mathrm{diag}(\sigma, -\sigma)$, where σ is the flow extension rate (see Section 4.1). Figure 3.5 plots two steady-state solutions of the Fokker-Planck equation when $\mathrm{Wi} = 1$, $\sigma = 0$ (left) and $\sigma = 5$ (right).

We can now prove that the drift term satisfies the condition (3.30). We have

$$\begin{aligned}\langle \vec{Q}, \vec{a}(\vec{Q})\rangle &= -\frac{1}{2\mathrm{Wi}}\vec{F}_\epsilon(\vec{Q})\cdot\vec{Q} + \vec{Q}^T\boldsymbol{\kappa}\vec{Q} \\ &\le -\frac{1}{2\mathrm{Wi}}\vec{F}_\epsilon(\vec{Q})\cdot\vec{Q} + B\|\vec{Q}\|^2. \end{aligned} \tag{3.34}$$

Case 1: $\|\vec{Q}\| < \sqrt{b}-\epsilon$:

$$\begin{aligned}\langle \vec{Q}, \vec{a}(\vec{Q})\rangle &\le -\frac{1}{2\mathrm{Wi}}\frac{\|\vec{Q}\|^2}{1-\frac{\|\vec{Q}\|^2}{b}} + B\|\vec{Q}\|^2 \\ &= -\frac{1}{2\mathrm{Wi}}\frac{\|\vec{Q}\|^2}{1-\frac{\|\vec{Q}\|^2}{b}} + B\frac{\|\vec{Q}\|^2 - \frac{\|\vec{Q}\|^4}{b}}{1-\frac{\|\vec{Q}\|^2}{b}} \\ &= -\left(\frac{1}{2\mathrm{Wi}} - B\right)\frac{\|\vec{Q}\|^2}{1-\frac{\|\vec{Q}\|^2}{b}} - B\frac{\frac{\|\vec{Q}\|^4}{b}}{1-\frac{\|\vec{Q}\|^2}{b}} \\ &\le -\left(\frac{1}{2\mathrm{Wi}} - B\right)\|\vec{Q}\|^2. \end{aligned} \tag{3.35}$$

Therefore we have

$$\langle \vec{Q}, \vec{a}(\vec{Q})\rangle \le -\alpha\|\vec{Q}\|^2 + \beta,$$

where $\alpha = \frac{1}{2\mathrm{Wi}} - B > 0, \beta > 0$ under the Assumption 3.2. On the other hand, if Assumption 3.2 does not hold (i.e., there exists a constant $B > 0$ such that: $\frac{1}{2\mathrm{Wi}} < \|\boldsymbol{\kappa}\|_F \le B$), the equation (3.35) is still bounded by its maximum value within a bounded domain when $\|\vec{Q}\| < \sqrt{b}-\epsilon$, therefore we can still find a pair of constants $\alpha, \beta > 0$ such that (3.30) holds.

Case 2: $\|\vec{Q}\| \ge \sqrt{b}-\epsilon$:

$$\langle \vec{Q}, \vec{a}(\vec{Q})\rangle \le -\frac{1}{2\mathrm{Wi}}\left(k\vec{Q} + \vec{F}(\vec{Q}_\epsilon) - k\vec{Q}_\epsilon\right)\cdot\vec{Q} + B\|\vec{Q}\|^2. \tag{3.36}$$

Using the expressions for $\vec{Q}_\epsilon$ and $\vec{F}$, we obtain:

$$\begin{aligned}\langle \vec{Q}, \vec{a}(\vec{Q})\rangle &\le -\frac{1}{2\mathrm{Wi}}\left(k\|\vec{Q}\|^2 + \frac{\|\vec{Q}\|(\sqrt{b}-\epsilon)}{1-\frac{(\sqrt{b}-\epsilon)^2}{b}} - k\|\vec{Q}\|(\sqrt{b}-\epsilon)\right) + B\|\vec{Q}\|^2 \\ &\le -\frac{1}{2\mathrm{Wi}}\left(k\|\vec{Q}\|^2 + \|\vec{Q}\|(\sqrt{b}-\epsilon) - k\|\vec{Q}\|(\sqrt{b}-\epsilon)\right) + B\|\vec{Q}\|^2 \\ &= -\frac{1}{2\mathrm{Wi}}\left(k\|\vec{Q}\|^2 + (1-k)\|\vec{Q}\|(\sqrt{b}-\epsilon)\right) + B\|\vec{Q}\|^2 \end{aligned}$$

$$\leq -\frac{1}{2\mathrm{Wi}}\left(k\|\vec{Q}\|^2 + (1-k)(\sqrt{b}-\epsilon)^2\right) + B\|\vec{Q}\|^2$$
$$= -(\frac{k}{2\mathrm{Wi}} - B)\|\vec{Q}\|^2 - \frac{1}{2\mathrm{Wi}}(1-k)(\sqrt{b}-\epsilon). \tag{3.37}$$

Therefore we have

$$\langle \vec{Q}, \vec{a}(\vec{Q})\rangle \leq -\alpha\|\vec{Q}\|^2 + \beta,$$

where $\alpha = \frac{k}{2\mathrm{Wi}} - B > 0$ under Assumption 3.2 and $\beta = -\frac{1}{2\mathrm{Wi}}(1-k)(\sqrt{b}-\epsilon)$ when $k > 1$ or β can be any positive number when $0 < k \leq 1$. On the other hand, if Assumption 3.2 does not hold, we can still choose the slope k so that $\alpha = \frac{k}{2\mathrm{Wi}} - B > 0$ and the condition (3.30) still holds. Combining the results from case 1 and case 2, we have shown that the extended FENE stochastic differential equation satisfies Assumption 3.1.

Remark 3.3 *We can see that with the help of Assumption 3.2, it is straightforward to show that the extended FENE stochastic differential equation satisfies Assumption 3.1, and from case 2 we can see that for the extended part we are free to choose the slope k in equation (3.37) to make sure that the equation (3.30) is satisfied. The size of the velocity gradient does not affect the result here, and we will come back to Assumption 3.2 when we talk about contractivity in Section 3.2.5.*

Now we can show based on Assumption 3.1 that the exact solution $\vec{Q}(t)$ to the stochastic differential equation (3.28) with $\vec{F}_\epsilon(\vec{Q})$ has a uniformly bounded pth moment, which only depends on p and the initial condition.

Lemma 3.1 *If the stochastic differential equation (3.28) with $\vec{F}_\epsilon(\vec{Q})$ satisfies the Assumption 3.1, then for each $p \in (1, \infty)$ there is a constant C, which only depend on p and $\vec{Q}(0)$, such that, for all $t \geq 0$,*

$$\mathbb{E}\left[\|\vec{Q}(t)\|^p\right] \leq C. \tag{3.38}$$

Proof. Consider the stochastic differential equation

$$\mathrm{d}\vec{Q}(t) = \vec{a}(\vec{Q}(t))\,\mathrm{d}t + \mathbf{b}(\vec{Q}(t))\,\mathrm{d}\vec{W}(t).$$

By Itô's formula applied to the quantity of interest $\Phi(\vec{Q}) = \mathrm{e}^{\frac{\alpha p t}{2}}\|\vec{Q}(t)\|^p$, for any $t > 0$ we have

$$\begin{aligned}\mathrm{e}^{\frac{\alpha p t}{2}}\|\vec{Q}(t)\|^p - \|\vec{Q}(0)\|^p \leq \int_0^t \Big(\frac{\alpha p}{2}\|\vec{Q}(t)\|^2 + p\langle\vec{Q}(s), \vec{a}(\vec{Q}(s))\rangle\Big)\|\vec{Q}(s)\|^{p-2}\mathrm{e}^{\frac{\alpha p s}{2}}\mathrm{d}s \\ + \frac{p(p-1)}{2}\int_0^t \|\vec{Q}(s)\|^{p-2}\|\mathbf{b}(\vec{Q}(s))\|^2\mathrm{e}^{\frac{\alpha p s}{2}}\mathrm{d}s \\ + p\int_0^t \mathrm{e}^{\frac{\alpha p s}{2}}\|\vec{Q}(s)\|^{p-2}\langle\vec{Q}(s), \mathbf{b}(\vec{Q}(s))\mathrm{d}\vec{W}(s)\rangle, \qquad (3.39)\end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product of vectors.

Taking the expectation on the both sides of (3.39) and applying the Assumption 3.1, we obtain

$$\begin{aligned}\mathbb{E}[\mathrm{e}^{\frac{\alpha p t}{2}}\|\vec{Q}(t)\|^p] - \mathbb{E}[\|\vec{Q}(0)\|^p] \leq -\frac{\alpha p}{2}\int_0^t \mathbb{E}[\mathrm{e}^{\frac{\alpha p s}{2}}\|\vec{Q}(s)\|^p]\mathrm{d}s \\ + \frac{\beta p(p+1)}{2}\int_0^t \mathbb{E}[\mathrm{e}^{\frac{\alpha p s}{2}}\|\vec{Q}(s)\|^{p-2}]\mathrm{d}s. \qquad (3.40)\end{aligned}$$

By Young's inequality (see the Appendix, with p and ψ there taken to be $\frac{p}{p-2}$ and $\frac{\alpha p^2}{2(p-2)}$, and with $a = \|\vec{Q}(s)\|^{p-2}$ and $b = \frac{\beta p(p+1)}{2}$) we have

$$\frac{\beta p(p+1)}{2}\mathrm{e}^{\frac{\alpha p s}{2}}\|\vec{Q}(s)\|^{p-2} \leq \frac{\alpha p}{2}\mathrm{e}^{\frac{\alpha p s}{2}}\|\vec{Q}(s)\|^p + C_{\alpha\beta p}\mathrm{e}^{\frac{\alpha p s}{2}},$$

where $C_{\alpha\beta p} = (\beta(p+1))^{\frac{p}{2}}(\frac{p-2}{\alpha p})^{\frac{p-2}{2}}$, which is a constant that only depends on α, β, p. Substituting into inequality (3.40), we have

$$\mathbb{E}[\mathrm{e}^{\frac{\alpha p t}{2}}\|\vec{Q}(t)\|^p] \leq \mathbb{E}[\|\vec{Q}(0)\|^p] + \int_0^t C_{\alpha\beta p}\mathrm{e}^{\frac{\alpha p s}{2}}\mathrm{d}s.$$

Multiplying this inequality by $\mathrm{e}^{\frac{-\alpha p t}{2}}$ we obtain

$$\mathbb{E}[\vec{Q}(t)^p] \leq \mathrm{e}^{\frac{-\alpha p t}{2}}\mathbb{E}[\|\vec{Q}(0)\|^p] + \frac{2C_{\alpha\beta p}}{\alpha p}(1 - \mathrm{e}^{\frac{-\alpha p t}{2}}).$$

For any $t > 0$, $\mathrm{e}^{\frac{-\alpha p t}{2}} \in (0, 1)$, we conclude that

$$\mathbb{E}\left[\|\vec{Q}(t)\|^p\right] \leq \mathbb{E}[\|\vec{Q}(0)\|^p] + \frac{2C_{\alpha\beta p}}{\alpha p} \triangleq C. \qquad \square$$

In [31], Higham *et al.* proved the strong convergence of the Euler–Maruyama approximation to the exact solution assuming that both the exact and the numerical solution have pth moment bounded and the SDE has a non-globally Lipschitz

drift. Lemma 3.1 guarantees the boundedness of the pth moment of the exact solution, however the boundedness of the pth moment of the Euler approximation is not guaranteed for a large class of stochastic differential equations. Hutzenthaler *et al.* [33] showed that the explicit Euler–Maruyama method will diverge in the case of a drift term with superlinear growth, therefore the standard Euler–Maruyama method with a fixed time step is unstable for non-globally Lipschitz stochastic differential equations. For the FENE SDE, it is necessary to apply the adaptive time stepping algorithm. Next, we will make an assumption on the time step in order to ensure the stability of the numerical scheme.

Assumption 3.3 *Suppose the adaptive time step is continuous and bounded and there exist some constants $\alpha, \beta > 0$ such that the adaptive time step $\Delta t(x)$ satisfies*

$$\langle \vec{x}, \vec{a}(\vec{x}) \rangle + \frac{1}{2}\Delta t(\vec{x})\|\vec{a}(\vec{x})\|^2 \leq -\alpha\|\vec{x}\|^2 + \beta \quad \forall x \in \mathbb{R}^n. \tag{3.41}$$

Note that if the condition (3.41) holds then the condition (3.30) also holds with the same constants. Under Assumptions 3.1 and 3.3, one can show the stability of the numerical scheme [18].

Lemma 3.2 [Theorem 1 in [18]] *Suppose Assumptions 3.1 and 3.3 are satisfied; then for each $p > 0$, there is a constant C_p, which only depends on p, α, β and the initial condition, such that, for all $t \geq 0$,*

$$\mathbb{E}\left[\|\vec{Q}_n\|^p\right] < C_p, \tag{3.42}$$

where $\vec{Q}_n$ denotes the numerical solution.

The proof of Lemma 3.2 can be found in the Appendix. Now we need to verify that our adaptive time step satisfies the Assumption 3.3. Recall that the time step we have used is

$$\Delta t(\vec{Q}) = \min\left(b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{\|\vec{Q}\|}{\sqrt{b}}\right)\right)^2, C_{\Delta t}\frac{\|\vec{Q}\|^2}{\|\boldsymbol{\kappa}\cdot\vec{Q} - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q})\|^2}\right). \tag{3.43}$$

Then we obtain:

$$\frac{1}{2}\Delta t(\vec{Q})\|\vec{a}(\vec{Q})\|^2 \leq \frac{1}{2}C_{\Delta t}\frac{\|\vec{Q}\|^2}{\|\boldsymbol{\kappa}\cdot\vec{Q} - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q})\|^2}\|\vec{a}(\vec{Q})\|^2 \leq \frac{C_{\Delta t}}{2}\|\vec{Q}\|^2. \tag{3.44}$$

From (3.35) and (3.37), we deduce that $\langle \vec{Q}, \vec{a}(\vec{Q}) \rangle \leq -\alpha \|\vec{Q}\|^2 + \beta$, and therefore we only need to choose the constant $C_{\Delta t}$ carefully to obtain $\langle \vec{Q}, \vec{a}(\vec{Q}) \rangle + \frac{1}{2}\Delta t(\vec{Q})\|\vec{a}(\vec{Q})\|^2 \leq -\alpha^\star \|\vec{x}\|^2 + \beta$.

Combining Lemmas 3.1 and 3.2, we can obtain the desired strong convergence result. Normally, the notion of strong convergence involves the difference between the exact and the numerical solution for a fixed time step Δt and by letting $\Delta t \to 0$, one can measure the strong convergence order. For the adaptive time step method, we need to characterise the time step by some parameter θ and measure the strong convergence order using that parameter.

Assumption 3.4 *Suppose the time step $\Delta t(x)$ satisfies Assumption 3.3 and $\Delta t^\theta(x)$ satisfies the following condition:*

$$\theta \min\left[\Delta t(x), \max(\Delta t(x))\right] \leq \Delta t^\theta(x) \leq \min\left[\Delta t(x), \theta \max(\Delta t(x))\right]. \tag{3.45}$$

Theorem 3.1 (Strong Convergence) [Theorem 2 in [17]] *Suppose that both the exact and the numerical solution have bounded pth moment and both the drift and volatility are locally Lipschitz continuous. Suppose also that Assumption 3.4 holds for the adaptive time step. Then we have strong convergence for all $p > 0$, and for all $t \geq 0$:*

$$\lim_{\theta \to 0} \mathbb{E}\left[\|\vec{Q}(t) - \vec{Q}_n\|^p\right] = 0. \tag{3.46}$$

The proof can be found in Theorem 2.2 in [31]. For the extended FENE force, we have shown that Assumption 3.1 is met, and therefore the exact solution has bounded pth moment. We have also shown that the adaptive time step we used satisfies Assumption 3.3, and hence the numerical solution also has bounded pth moment. In addition, the drift and volatility are locally Lipschitz continuous, and therefore the adaptive time stepping method for the equation (3.28) exhibits strong convergence. In order to obtain the strong convergence order, we need to make additional assumptions on the drift and volatility which will be discussed in the next section.

Remark 3.4 *For the purpose of the analysis, we extend the drift to the whole space because it is not defined outside the configuration domain, however the analysis is based on the whole space $\mathbb{R}^n$. Although we can use some linear function as the*

extension which makes the extended system become globally Lipschitz continuous, we are interested in the non-linear part and in practice we never use the extension part; therefore the extension part is not important to us. In Remark 3.2 we mentioned that we can also use an appropriate polynomial function as the extension (then the extended function is only locally Lipschitz continuous) and the analysis is similar.

3.2.5 Improved Adaptive Multilevel Monte Carlo Method

This section is originally motivated by the research done by Fang and Giles [17]. After establishing the stability of the numerical solution and the strong convergence order using an adaptive time stepping method for SDEs with a non-globally Lipschitz drift term, we can propose a better adaptive multilevel Monte Carlo method. The equation we consider is the stochastic differential equation

$$\mathrm{d}\vec{Q}(t) = \vec{a}(\vec{Q}(t))\mathrm{d}t + \mathbf{b}(\vec{Q}(t))\mathrm{d}\vec{W}_t. \tag{3.47}$$

Here $\vec{Q}(t) \in \mathbb{R}^n$ and $\vec{a} : \mathbb{R}^n \to \mathbb{R}^n$ is the non-global Lipschitz drift satisfying the one-sided Lipschitz condition:

$$\langle \vec{Q}_1 - \vec{Q}_2, \vec{a}(\vec{Q}_1) - \vec{a}(\vec{Q}_2)\rangle \leq \mu \|\vec{Q}_1 - \vec{Q}_2\|^2 \quad \forall \vec{Q}_1, \vec{Q}_2 \in \mathbb{R}^n, \tag{3.48}$$

where $\mu > 0$. We are interested in computing the expectation of some smooth function $g(x)$ with respect to the invariant measure π of the SDE:

$$\pi(g) = \lim_{t\to\infty} \mathbb{E}[g(\vec{Q}(t))]. \tag{3.49}$$

Lemma 3.3 (Existence and uniqueness of the invariant measure). [Lemma 6 in [18]] *Suppose that the drift a and volatility b are both locally Lipschitz. In addition, assume that the drift term satisfies equation (3.30) and the volatility term is globally bounded; then the stochastic differential equation (3.47) has a unique invariant measure π. Assuming that g is globally Lipschitz continuous, we have*

$$|\mathbb{E}[g(\vec{Q}(t)) - \pi(g)]| \leq C_1(1 + \|\vec{Q}(0)\|^2)\,\mathrm{e}^{-C_2 t}$$

where C_1 and C_2 are positive constants and $\vec{Q}(0)$ is the initial datum of the stochastic differential equation.

The proof can be found in [42], and by Lemma 3.3, we are guaranteed to have a unique invariant measure for the equation (3.47).

Note that the time interval here is $[0, \infty)$ instead of a finite time interval. In order to compute the expected value when t tends to infinity, we need to truncate the infinite time interval at some large T and then apply the standard multilevel Monte Carlo method to the problem over the interval $[0, T]$; however the difficulty is how to choose a reasonable time T.

A new multilevel Monte Carlo method is introduced to approach this problem. Instead of using a fixed time interval $[0, T]$ across all the levels, the authors of [18] use time intervals of different lengths T_l for different levels l, and therefore from the coarse level to the fine level not only the number of time steps is increased but also the length of the time interval is increased. In the standard multilevel Monte Carlo method, we simulate both the coarse and the fine level using the same driven Brownian motion on $[0, T]$ and thanks to the strong convergence, the difference between them at time T is small. However we can not control the difference between the numerical results at time T_l and T_{l-1} if T_l is not equal to T_{l-1}. The idea is that we start the simulation at time $-T_l$ at fine level l and $-T_{l-1}$ at coarse level $l-1$ and then stop both of them at time zero. Due to the fact that both the drift term and the volatility term are independent of time t, the probability density function of the numerical solution simulated on the time interval $[0, T_l]$ is the same as on the time interval $[-T_l, 0]$ at the fine level and the same result applies to the coarse level as well.

Lemma 3.4 (Contractivity). [Lemma 3 in [18]] *Suppose that the stochastic differential equation (3.47) satisfies the contractive Lipschitz properties:*

$$\langle \vec{Q}_1 - \vec{Q}_2, \vec{a}(\vec{Q}_1) - \vec{a}(\vec{Q}_2) \rangle + \frac{p^\star - 1}{2} \| \boldsymbol{b}(\vec{Q}_1) - \boldsymbol{b}(\vec{Q}_2) \|^2 \leq -\lambda \| \vec{Q}_1 - \vec{Q}_2 \|^2 \quad \forall \vec{Q}_1, \vec{Q}_2 \in \mathbb{R}^n$$

$$\| \boldsymbol{b}(\vec{Q}_1) - \boldsymbol{b}(\vec{Q}_2) \|^2 \leq \eta \| x - y \|^2 \quad \forall \vec{Q}_1, \vec{Q}_2 \in \mathbb{R}^n$$

for some $p^\star \in (1, \infty)$, $\lambda, \eta > 0$. Then, for any $p \in (0, p^\star]$, any two solutions $\vec{X}_t, \vec{Y}_t$ to the SDE (3.47) driven by the same Brownian motion but starting from different initial values $\vec{X}(-T), \vec{Y}(-T)$ will converge to each other exponentially, i.e.,

$$\mathbb{E}[\| \vec{X}(0) - \vec{Y}(0) \|^p] \leq \mathrm{e}^{-\lambda p T} \mathbb{E}[\| \vec{X}(-T) - \vec{Y}(-T) \|^p] \quad \forall T > 0.$$

The proof of Lemma 3.4 can be found in the Appendix. The contractivity property makes sure that the fine path and coarse path driven by the same Brownian motion will converge to each other exponentially, and therefore the variance of the difference between fine and coarse path solutions at time zero will be small.

Now we need to verify that the extended FENE SDE satisfies the contractivity property. In the extended FENE SDE, the volatility in the equation (3.28) is constant, therefore it is globally Lipschitz continuous and globally bounded and it also implies that the contractive Lipschitz property in Lemma 3.4 becomes:

$$\langle \vec{Q}_1 - \vec{Q}_2, \vec{a}(\vec{Q}_1) - \vec{a}(\vec{Q}_2) \rangle \leq -\lambda \|\vec{Q}_1 - \vec{Q}_2\|^2 \quad \forall \vec{Q}_1, \vec{Q}_2 \in \mathbb{R}^n$$

This is equivalent showing that the drift in the equation (3.28) with $\vec{F}_\epsilon(\vec{Q})$ is globally one-sided Lipschitz continuous with negative Lipschitz constant because in Lemma 3.4 we are required to have $-\lambda$ on the right hand side of the inequality with $\lambda > 0$. Suppose that we consider the extend FENE drift in the 1D case: $a(x) = \kappa x - \frac{1}{2\mathrm{Wi}} \frac{x}{1-x^2}$ for $x \in (-1, 1)$ and a linear function for the extended part. Then this condition requires that the product of $x_1 - x_2$ and $a(x_1) - a(x_2)$ is always less or equal than zero for all $x \in \mathbb{R}$, which means that the function $a(x)$ must be monotonically decreasing. Furthermore, if the function $a \in C^1$, the first derivative must be bounded above by zero for all $x \in \mathbb{R}$. Hence we can deduce that for the 1D case we must have $\kappa \leq \frac{1}{2\mathrm{Wi}}$ in order to satisfy the contractive Lipschitz property.

Note that if the drift is differentiable, the contractive Lipschitz property is equivalent to showing the following condition:

$$\langle \vec{e}, \nabla \vec{a}(\vec{Q}) \cdot \vec{e} \rangle \leq -\lambda,$$

where $\vec{e} \in \mathbb{R}^n$ with $\|\vec{e}\| = 1$.

By computing the gradient of the drift we obtain $\nabla \vec{a}(\vec{Q}) = -\frac{1}{2\mathrm{Wi}} \nabla \vec{F}_\epsilon + \boldsymbol{\kappa}$ and we have:

$$\begin{aligned} \langle \vec{e}, \nabla \vec{a}(\vec{Q}) \cdot \vec{e} \rangle &= \langle \vec{e}, -\frac{1}{2\mathrm{Wi}} \nabla \vec{F}_\epsilon \vec{e} + \boldsymbol{\kappa} \vec{e} \rangle \\ &= -\frac{1}{2\mathrm{Wi}} \vec{e}^T \nabla \vec{F}_\epsilon \vec{e} + \vec{e}^T \boldsymbol{\kappa} \vec{e} \end{aligned}$$

$$
\begin{aligned}
&\leq -\frac{1}{2\mathrm{Wi}}\|\nabla \vec{F}_\epsilon\|_F\|\vec{e}\| + \|\boldsymbol{\kappa}\|_F\|\vec{e}\| \\
&= -\frac{1}{2\mathrm{Wi}}\|\nabla \vec{F}_\epsilon\|_F + \|\boldsymbol{\kappa}\|_F \\
&\leq -\frac{1}{2\mathrm{Wi}} + \|\boldsymbol{\kappa}\|_F.
\end{aligned}
$$

We require that $-\frac{1}{2\mathrm{Wi}} + \|\boldsymbol{\kappa}\|_F$ has a negative upper bound; therefore we should have that $\|\boldsymbol{\kappa}\|_F$ is less than $\frac{1}{2\mathrm{Wi}}$, i.e., Assumption 3.2 must be satisfied. Notice that if the SDE satisfies contractivity, we can deduce that the SDE also satisfies Assumption 3.1 while the reverse is not true.

Similarly to (3.5), the multilevel Monte Carlo estimator is again the telescopic sum:

$$
\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l,
$$

where

$$
\hat{Y}_0 = \frac{1}{N_0}\sum_{i=1}^{N_0} g(\hat{S}_0^{(i)}(0)),
$$

$$
\hat{Y}_l = \frac{1}{N_l}\sum_{i=1}^{N_l} \left(g(\hat{S}_l^{(i)}(0)) - g(\hat{S}_{l-1}^{(i)}(0)) \right). \tag{3.50}
$$

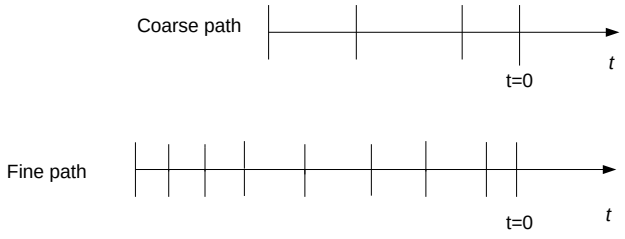


Figure 3.6: Illustration of the multilevel Monte Carlo method at coarse and fine level

The variance of the multilevel Monte Carlo estimator with time step refinement R has been proved to satisfy [18] :

$$V_l \leq c_1 R^{-l} + c_2 e^{-2\lambda T_{l-1}}. \tag{3.51}$$

By choosing $T_{l-1} = \frac{\log R}{2\lambda} l$, we ensure that the variance is of the order of R^{-l}, which is the same as (3.10c) in the standard multilevel Monte Carlo method.

We can now apply this method to the FENE SDE. In our case, the volatility term is $\frac{1}{\sqrt{\mathrm{Wi}}}$, which is globally Lipschitz continuous and globally bounded and we checked that the system satisfies Assumption 3.1. Then by Lemma 3.3 we know that there exists a unique invariant measure to the SDE. In addition Lemma 3.4 ensures that the SDE starting from different values but driven by the same Brownian motion will converge to each other eventually as long as contractivity holds. For our problem, in order to have contractivity we need to make sure that the velocity gradient $\boldsymbol{\kappa}$ is not too large because the drift term should be one-sided Lipschitz continuous with negative Lipschitz constant as we discussed in the previous paragraph. Once we have contractivity, we can use the improved adaptive multilevel Monte Carlo method to reduce the computational cost. We show a numerical example to illustrate this in Chapter 4.

Theorem 3.2 (Strong convergence order) [Theorem 2 in [18]] *Suppose that the stochastic differential equation satisfies the contractive Lipschitz properties and the time step satisfies the Assumption 3.4; then for all $p > 0$ and for all $t \geq 0$, we have*

$$\mathbb{E}[\|\vec{Q}(t) - \vec{Q}_n\|^p] \leq C_p \theta^p,$$

where C_p is a constant.

Algorithm 3: This algorithm is the single path adaptive multilevel Monte Carlo simulation for level L for solving (3.15) using adaptive time stepping and clamping.

Data: constant $\boldsymbol{\kappa}$, initial probability density distribution $\psi_0(\vec{q})$, initial number of samples N_0, clamping factor δ, level L, refinement factor R, the length of time interval $T_l = \frac{\log R}{2\lambda}(l+1)$

Result: $\hat{Y}_L$

Sample the initial value from $\psi_0(\vec{q})$ and initialize the following variables;

$t := -T_l$, $t_f := -T_l$, $t_c := -T_{l-1}$, $\Delta t := 0$, $\Delta t_f := 0$, $\Delta t_c := 0$, $\Delta \vec{W}_f = \vec{0}$, $\Delta \vec{W}_c = \vec{0}$;

if $L = 0$ **then**

  while $t < 0$ **do**

    Clamp the trajectories if they go beyond the safety margin;

    Calculate the time step Δt using (3.25);

    $\Delta t := \min(\Delta t, -t)$;

    Simulate the stochastic differential equations (3.15) according to (3.20) using Δt;

    $t = t + \Delta t$;

  end

else

  while $t < t_c$ **do**

    Clamp the trajectories if needed;

    Compute the time step Δt_f according to (3.25) at fine level;

    $\Delta t_f = \min(\Delta t_f, t_c - t_f)$, $t_f = t_f + \Delta t_f$;

    $\Delta \vec{W} = \sqrt{\Delta t_f} \times \vec{N}(0,1)$;

    Update the fine path using Δt_f and $\Delta \vec{W}$;

    $t = t_f$

  end

  Compute both Δt_f and Δt_c according to (3.25) for fine and coarse levels;

  $\Delta t := \min(\Delta t_f, \Delta t_c)$, $t_f = t_f + \Delta t_f$, $t_c = t_c + \Delta t_c$;

  while $t < 0$ **do**

    $\Delta \vec{W} := \sqrt{\Delta t} \times \vec{N}(0,1)$;

    $\Delta \vec{W}_f := \Delta \vec{W}_f + \Delta \vec{W}$, $\Delta \vec{W}_c := \Delta \vec{W}_c + \Delta \vec{W}$;

    if $t_f < t_c$ **then**

      $t := \min(t_f, 0)$, Clamp the trajectories if needed;

      Simulate the fine path using time step Δt and $\Delta \vec{W}_f$;

      Compute new Δt_f using (3.25), $t_f := t_f + \Delta t_f$,

       $\Delta t := \min(\Delta t_f, t_c - t, -t)$, $\vec{W}_f = \vec{0}$;

    else

      $t := \min(t_c, 0)$, Clamp the trajectories if needed;

      Simulate the coarse path using time step Δt and $\Delta \vec{W}_c$;

      Compute new Δt_c using (3.25) $t_c := t_c + \Delta t_c$,

       $\Delta t := \min(\Delta t_c, t_f - t, -t)$, $\vec{W}_c = \vec{0}$;

    end

  end

end

Compute $\hat{Y}_L$ using the numerical solution at level L and $L-1$;

In the last section, Theorem 3.1 states the strong convergence of the numerical solution using the adaptive time stepping scheme. With a further condition on the drift, Theorem 3.2 states that the strong convergence order of the adaptive time stepping scheme is 1. In our problem, if the size of $\boldsymbol{\kappa}$ is large (Assumption 3.2 is violated), the drift will not satisfy the contractive Lipschitz property, and then we are unable to show that the strong convergence order is 1. The proof of Theorem 3.2 is given in [18].

3.3 Stochastic Simulation for Nonhomogeneous Flows

In this section, we will focus on the numerical algorithm for nonhomogeneous flows. The configuration variable is not only dependent on time, but it also depends on the physical space variable $\vec{x}$. The velocity gradient $\boldsymbol{\kappa}(\vec{x},t) = \nabla_x \vec{u}(\vec{x},t)$ is no longer a constant matrix either. For a given velocity field $\vec{u}(\vec{x},t)$ we can first approximate the gradient $\boldsymbol{\kappa}(\vec{x},t)$ at any time and any specified position by using a central finite difference approximation. Then we can use these computational results and apply either Algorithm 1 or Algorithm 2 to obtain the value of the polymeric extra stress tensor at these spatial points $\vec{x}$. Therefore we need to repeatedly apply the Euler–Maruyama method or the adaptive time stepping method for all different grid points in the flow domain. We must calculate all their individual trajectories and they are independent of each other, which leads to a high computational cost.

3.3.1 Polymeric Stress Tensor

For nonhomogeneous flows, the polymeric stress tensor is a function of time and space. We will calculate the Kramers expression $\boldsymbol{\tau}(\vec{x},t)$ at each node in the discretisation. In order to save work, we will apply the Multilevel Monte Carlo method at the same time to obtain all of the expected values simultaneously. In particular we want to avoid approximating the SDE too many times, and we do not want to calculate the individual expectations one by one. We will obtain the estimations at each of the

nodes after we have finished calculating the solution to the SDE. Then we obtain the variance of each estimator and calculate the corresponding required number of simulations. Finally, we pick the largest number of simulations to update all the estimators.

3.4 Numerical Algorithms for the Coupled System

In this section we develop an algorithm for the coupled system involving the macroscopic Navier–Stokes equations and the microscopic stochastic differential equations and the Kramers expression for the extra stress tensor coupling them. Recall the system stated in Chapter 1:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla_x \vec{u} = -\nabla_x p + \frac{\gamma}{\mathrm{Re}} \Delta_x \vec{u} + \nabla_x \cdot \boldsymbol{\tau} + \vec{f}, \tag{3.52}$$

$$\nabla_x \cdot \vec{u} = 0, \tag{3.53}$$

$$\mathrm{d}\vec{Q} = \left(\boldsymbol{\kappa} \cdot \vec{Q} - \frac{1}{2\mathrm{Wi}} \vec{F}(\vec{Q}) \right) \mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}} \mathrm{d}\vec{W}(t), \tag{3.54}$$

$$\boldsymbol{\tau}(\vec{x}, t) = \frac{b+d+2}{b} \frac{1-\gamma}{\mathrm{ReWi}} \left(\mathbb{E}[\vec{Q} \otimes \vec{F}(\vec{Q})] - \mathbf{I} \right), \tag{3.55}$$

$$\vec{u}(\vec{x}, 0) = \vec{u}_0(\vec{x}), \quad \vec{Q}(\vec{x}, 0) \sim \psi_0(\vec{q}). \tag{3.56}$$

3.4.1 Mixed Finite Element Method

Next, we introduce the mixed finite element approximation of the incompressible Navier–Stokes equations. The detailed analysis of the method can be found in [15] and [27]. First let us consider the weak formulation of the Navier–Stokes equations (3.52) and (3.53). As in Chapter 2, let $\vec{v}$ and q denote the test functions in suitable spaces which will be specified below. Then we have:

$$\begin{aligned} &\int_\Omega \frac{\partial \vec{u}}{\partial t} \cdot \vec{v} \, \mathrm{d}\vec{x} + \int_\Omega (\vec{u} \cdot \nabla_x \vec{u}) \cdot \vec{v} \, \mathrm{d}\vec{x} \\ &= -\int_\Omega \nabla_x p \cdot \vec{v} \, \mathrm{d}\vec{x} + \frac{\gamma}{\mathrm{Re}} \int_\Omega \Delta_x \vec{u} \cdot \vec{v} \, \mathrm{d}\vec{x} + \int_\Omega (\nabla_x \cdot \boldsymbol{\tau}) \cdot \vec{v} \, \mathrm{d}\vec{x} + \int_\Omega \vec{f} \cdot \vec{v} \, \mathrm{d}\vec{x}, \end{aligned} \tag{3.57}$$

$$\int_\Omega q \nabla_x \cdot \vec{u} \, \mathrm{d}\vec{x} = 0. \tag{3.58}$$

Performing integration by parts yields for all choices of the test functions $\vec{v}$ and q in suitable function spaces:

$$\int_\Omega \frac{\partial \vec{u}}{\partial t} \cdot \vec{v}\,\mathrm{d}\vec{x} + \int_\Omega (\vec{u} \cdot \nabla_x \vec{u}) \cdot \vec{v}\,\mathrm{d}\vec{x} - \int_\Omega p \nabla_x \cdot \vec{v}\,\mathrm{d}\vec{x}$$
$$+ \frac{\gamma}{\mathrm{Re}} \int_\Omega \nabla_x \vec{u} : \nabla_x \vec{v}\,\mathrm{d}\vec{x} + \int_\Omega \boldsymbol{\tau} : \nabla_x \vec{v}\,\mathrm{d}\vec{x}$$
$$+ \int_{\partial\Omega} \left(p\mathbf{I} - \frac{\gamma}{\mathrm{Re}} \nabla_x \vec{u} - \boldsymbol{\tau} \right) \vec{v} \cdot \vec{n}\,\mathrm{d}s = \int_\Omega \vec{f} \cdot \vec{v}\,\mathrm{d}\vec{x}, \tag{3.59}$$
$$\int_\Omega q \nabla_x \cdot \vec{u}\,\mathrm{d}\vec{x} = 0. \tag{3.60}$$

In this thesis, we consider enclosed flow and channel flow problems. We split the boundary into the Dirichlet part $\partial\Omega_D$ and Neumann part $\partial\Omega_N$. For the enclosed flow problem, we have zero boundary condition $\vec{u} = \vec{0}$ for all the boundaries $\partial\Omega$. As a result of Dirichlet boundary condition, we choose $\vec{v} = \vec{0}$ on these boundaries, and therefore the boundary term in (3.59) vanishes.

On the other hand, for the channel flow problem, we will have inflow boundary $\partial\Omega_{\mathrm{in}}$, outflow boundary $\partial\Omega_{\mathrm{out}}$ and the wall boundary $\partial\Omega_0$. The velocity is considered to be zero on the wall boundary, and therefore we impose a no-slip boundary condition on $\partial\Omega_0$. For the inflow boundary, the velocity is specified by a given velocity function $\vec{u}_{\mathrm{in}}$. Therefore the test function $\vec{v}$ is chosen to be zero on these Dirichlet boundaries $(\partial\Omega_0 \cup \partial\Omega_{\mathrm{in}})$ and we impose

$$(p\mathbf{I} - \frac{\gamma}{\mathrm{Re}} \nabla_x \vec{u} - \boldsymbol{\tau}) \cdot \vec{n} = \vec{0}, \quad \text{on} \quad \partial\Omega_{\mathrm{out}}. \tag{3.61}$$

Hence the boundary term in (3.59) is zero. Let us define the following spaces

$$H^1_E = \{\vec{u} \in H^1(\Omega) : \vec{u} = \vec{w} \quad \text{on} \quad \partial\Omega_D\}, \tag{3.62}$$

$$H^1_{E_0} = \{\vec{v} \in H^1(\Omega) : \vec{v} = \vec{0} \quad \text{on} \quad \partial\Omega_D\}, \tag{3.63}$$

$$X = H^1_E. \tag{3.64}$$

Then the weak formulation is the following: find $\vec{u}(t) \in X$ and $p(t) \in L^2(\Omega)$ such that

$$\int_\Omega \frac{\partial \vec{u}}{\partial t} \cdot \vec{v}\,\mathrm{d}\vec{x} + \int_\Omega (\vec{u} \cdot \nabla_x \vec{u}) \cdot \vec{v}\,\mathrm{d}\vec{x} - \int_\Omega p \nabla_x \cdot \vec{v}\,\mathrm{d}\vec{x}$$

$$+\frac{\gamma}{\mathrm{Re}}\int_\Omega \nabla_x\vec{u}:\nabla_x\vec{v}\,\mathrm{d}\vec{x}+\int_\Omega \boldsymbol{\tau}:\nabla_x\vec{v}\,\mathrm{d}\vec{x}=\int_\Omega \vec{f}\cdot\vec{v}\,\mathrm{d}\vec{x}\quad\forall\vec{v}\in H^1_{E_0}, \tag{3.65}$$

$$\int_\Omega q\nabla_x\cdot\vec{u}\,\mathrm{d}\vec{x}=0\quad\forall q\in L^2(\Omega). \tag{3.66}$$

Suppose $X_h\subset X$, $X_{h,0}=X_h\cap H^1_{E_0}$, and $M_h\subset L^2(\Omega)$ are finite-dimensional linear subspaces of the Hilbert spaces X and $L^2(\Omega)$, respectively, which we choose to be finite element spaces on a given triangulation, of granularity h, of the computational domain Ω. Then the corresponding discrete weak formulation of (3.65)–(3.66) is defined using these finite dimensional spaces. Assuming $\vec{u}_h^0\in X_h$, $p_h^0\in M_h$ and $\boldsymbol{\tau}_h^n$ is given for all the previous time steps, then for $n=0,1,\ldots,\frac{T}{\Delta t_{\mathrm{NS}}}-1$, ($\Delta t_{\mathrm{NS}}$ is the fixed time step for the Navier–Stokes equations), find $\vec{u}_h^{n+1}\in X_h$ and $p_h^{n+1}\in M_h$ such that

$$\int_\Omega \frac{\vec{u}_h^{n+1}-\vec{u}_h^n}{\Delta t}\cdot\vec{v}_h\,\mathrm{d}\vec{x}+\int_\Omega(\vec{u}_h^n\cdot\nabla_x\vec{u}_h^n)\cdot\vec{v}_h\,\mathrm{d}\vec{x}-\int_\Omega p_h^n\nabla_x\cdot\vec{v}_h\,\mathrm{d}\vec{x}$$
$$+\frac{\gamma}{\mathrm{Re}}\int_\Omega\nabla_x\vec{u}_h^n:\nabla_x\vec{v}_h\,\mathrm{d}\vec{x}+\int_\Omega\boldsymbol{\tau}_h^n:\nabla_x\vec{v}_h\,\mathrm{d}\vec{x}=\int_\Omega\vec{f}\cdot\vec{v}_h\,\mathrm{d}\vec{x}\quad\forall\vec{v}_h\in X_{h,0} \tag{3.67}$$

$$\int_\Omega q_h\nabla_x\cdot\vec{u}_h^n\,\mathrm{d}\vec{x}=0\quad\forall q_h\in M_h. \tag{3.68}$$

We consider the initial condition for the pressure $p_h^0=0$ on the whole of Ω in our numerical experiment.

3.4.2 Coupled System

So far, we have introduced the individual numerical algorithms for the various parts of the system of equations. Now we can combine these separate pieces together for the coupled system. The Navier–Stokes equations are simulated by the mixed finite element method with fixed time step Δt_{NS} in the time interval $t\in[0,T]$. During each time step $t\in[t_{n-1},t_n]$, we need to simulate the stochastic differential equation by the adaptive time stepping method and perform multilevel Monte Carlo simulations to obtain the value for the tensor $\boldsymbol{\tau}$ assuming the velocity is constant during that time interval. The whole algorithm contains the following steps:

1. Start with $t=0$ and we are given the initial condition for the velocity $\vec{u}_0$ and the initial distribution of the configuration ψ_0. From ψ_0 we can compute the

initial value for the tensor $\boldsymbol{\tau}_0$ and generate an initial condition $\vec{Q}_0$ for the stochastic differential equation. Also we can calculate the initial velocity gradient $\boldsymbol{\kappa}_0$ from $\vec{u}_0$.

2. Using $\vec{u}_0$ and $\boldsymbol{\tau}_0$, we can obtain the numerical approximation to the velocity $\vec{u}_1$ at the next time step, and then we can find the velocity gradient $\boldsymbol{\kappa}_1$ at that time as well.

3. Using $\vec{Q}_0$ and the corresponding $\boldsymbol{\kappa}_0$, we can perform the numerical simulation of the SDE for $\vec{Q}_1$ and the estimation for $\boldsymbol{\tau}_1$.

4. Using $\vec{u}_{n-1}$ and $\boldsymbol{\tau}_{n-1}$, we can obtain $\vec{u}_n$ and $\boldsymbol{\kappa}_n$.

5. Using $\vec{Q}_{n-1}$ and $\boldsymbol{\kappa}_{n-1}$, we can obtain $\vec{Q}_n$ and $\boldsymbol{\tau}_n$.

6. Repeat steps 4 and 5 until we reach the final time T.

We have two different time scales in our problem. One is for the Navier–Stokes equations and another is for the SDE during each time interval $[t_{n-1}, t_n]$. Recall that in Section 1.3.1, we mentioned that the Weissenberg number can be regarded as the ratio of the microscopic to macroscopic time scales. A small Weissenberg number implies that the microscopic relaxation time of the beads is smaller than the characteristic fluid process time; then the time scale for the SDE will be smaller than the time scale for the Navier–Stokes equations. In that case the SDE achieves the invariant measure quickly, which means that we usually do not need to simulate over the entire interval $[t_{n-1}, t_n]$. In the previous section, we introduced the adaptive multilevel Monte Carlo method for the SDE, which has an invariant measure. The idea of the method is that the time interval is not fixed but it keeps changing as the level increases and the algorithm will stop when the invariant measure is obtained to within a given tolerance. We can use this method in our computation for a small Weissenberg number problem to reduce the computational cost because we do not need to simulate the SDE over the entire time interval $[t_{n-1}, t_n]$. On the other hand, if the Weissenberg number is large, the time for the SDE converging to the invariant measure might be larger than $t_n - t_{n-1}$; then we do not gain any benefits using the adaptive multilevel Monte Carlo method.

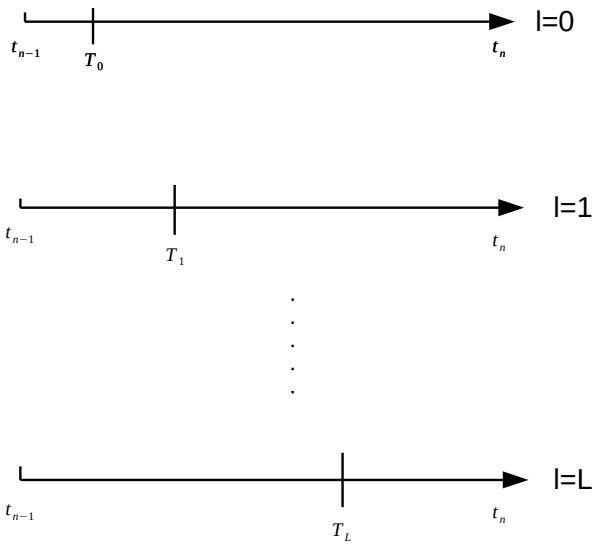


Figure 3.7: Time interval of the adaptive multilevel Monte Carlo method at different levels

Figure 3.7 indicates that we use different time intervals for different levels within $[t_{n-1}, t_n]$. Suppose the SDE achieves the invariant measure to within the given tolerance at level L and the time we used is smaller than $t_n - t_{n-1}$. Then we only need to simulate the SDE from t_{n-1} to T_L and there is no need to simulate over the entire time interval. On the other hand, if the invariant measure is still not achieved when we have already simulated over the whole time interval, we may not gain extra benefit from using the adaptive multilevel Monte Carlo method for our problem.

Remark 3.5 *For a small Weissenberg number, we may apply the adaptive multilevel Monte Carlo method to reduce the overall costs, however we need to be careful when we apply the algorithm, because we must check the contractivity of the SDE.*

The contractivity depends not only on the Weissenberg number but also the velocity gradient $\boldsymbol{\kappa}$.

3.5 Multi-Bead Simulations

In Section 1.6, we introduced the multi-bead simulations, which include $K+1$ ($K>1$) beads connected with K elastic springs to represent a polymer chain. For the dumbbell model ($K=1$) discussed above we only need one stochastic differential equation to represent one configuration vector; however for the multi-bead model, we need K coupled stochastic differential equations to represent K different configuration vectors. Recall the equations:

$$\begin{aligned}
\mathrm{d}\vec{Q}_1(t) &= \left((\nabla_x \vec{u})\vec{Q}_1(t) - \frac{1}{2\mathrm{Wi}}\left(-\vec{F}_2(\vec{Q}_2(t)) + 2\vec{F}_1(\vec{Q}_1(t))\right)\right)\mathrm{d}t \\
&\quad + \sqrt{\frac{1}{\mathrm{Wi}}}(\mathrm{d}\vec{W}_2(t) - \mathrm{d}\vec{W}_1(t)), \\
\mathrm{d}\vec{Q}_i(t) &= \left((\nabla_x \vec{u})\vec{Q}_i(t) - \frac{1}{2\mathrm{Wi}}\left(-\vec{F}_{i+1}(\vec{Q}_{i+1}(t)) + 2\vec{F}_i(\vec{Q}_i(t)) - \vec{F}_{i-1}(\vec{Q}_{i-1}(t))\right)\right)\mathrm{d}t \\
&\quad + \sqrt{\frac{1}{\mathrm{Wi}}}(\mathrm{d}\vec{W}_{i+1}(t) - \mathrm{d}\vec{W}_i(t)), \quad 1<i<K, \\
\mathrm{d}\vec{Q}_K(t) &= \left((\nabla_x \vec{u})\vec{Q}_K(t) - \frac{1}{2\mathrm{Wi}}\left(-\vec{F}_{K-1}(\vec{Q}_{K-1}(t)) + 2\vec{F}_K(\vec{Q}_K(t))\right)\right)\mathrm{d}t \\
&\quad + \sqrt{\frac{1}{\mathrm{Wi}}}(\mathrm{d}\vec{W}_{K+1}(t) - \mathrm{d}\vec{W}_K(t)).
\end{aligned} \tag{3.69}$$

The stress tensor $\boldsymbol{\tau}$ is now the sum of all the individual expectations; more precisely

$$\boldsymbol{\tau}(\vec{x},t) = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{ReWi}}\left(\sum_{i=1}^{K}\mathbb{E}[\vec{Q}_i(t)\otimes\vec{F}_i(\vec{Q}_i(t))] - \mathbf{I}\right). \tag{3.70}$$

The adaptive time stepping method for the multi-bead system is similar as for the dumbbell model. For the dumbbell model we have $-\vec{F}(\vec{Q}(t))$ in the drift term, which always tries to shrink the length of the spring, and therefore we only need to pay attention to the velocity gradient and the Brownian term, while for the multi-bead system, the FENE force becomes $\vec{F}_{i+1}(\vec{Q}_{i+1}(t)) - 2\vec{F}_i(\vec{Q}_i(t)) + \vec{F}_{i-1}(\vec{Q}_{i-1}(t))$ in the ith equation. If we take too large a time step, it is possible that the length of the

ith spring will be greater than the maximum extension to which the dumbbell can be stretched. In Section 3.2.2, we first control the size of the stochastic part by equation (3.22). We have $(\mathrm{d}\vec{W}_{i+1}(t) - \mathrm{d}\vec{W}_i(t))$ in the multi-bead system instead of a single Brownian increment in the dumbbell model and the difference of two independent Brownian increments follows a normal distribution with zero mean and twice the time step. Therefore the time step for the ith equation is:

$$\Delta t^i = \frac{\mathrm{Wi}}{2}\left(\frac{1}{10}\left(1 - \frac{\|\vec{Q}_i\|}{\sqrt{b}}\right)\right)^2. \tag{3.71}$$

Similarly to equation (3.25), we also need the restriction on the drift term and we pick the time step to be:

$$\Delta t^i = \min\left(\frac{\mathrm{Wi}}{2}\left(\frac{1}{10}\left(1 - \frac{\|\vec{Q}_i\|}{\sqrt{b}}\right)\right)^2, C_{\Delta t}\frac{\|\vec{Q}_i\|^2}{\|\boldsymbol{\kappa}\cdot\vec{Q}_i - \frac{1}{2\mathrm{Wi}}\vec{F}_i^M\|^2}\right), \tag{3.72}$$

where

$$\vec{F}_i^M = \begin{cases} -\vec{F}_2(\vec{Q}_2(t)) + 2\vec{F}_1(\vec{Q}_1(t))) & i = 1, \\ -\vec{F}_{i+1}(\vec{Q}_{i+1}(t)) + 2\vec{F}_i(\vec{Q}_i(t)) - \vec{F}_{i-1}(\vec{Q}_{i-1}(t)) & 1 < i < K, \\ -\vec{F}_{K-1}(\vec{Q}_{K-1}(t)) + 2\vec{F}_K(\vec{Q}_K(t)) & i = K. \end{cases} \tag{3.73}$$

Remark 3.6 *Instead of one equation, we now have K equations that need to be simulated at the same time. In order to make sure that the length of each spring does not exceed the bound imposed by the diameter of the domain D, we still need to use the adaptive time stepping method for each equation, however the time step Δt^i (the time step for the ith equation) now depends on the norm of the configuration vector $\|\vec{Q}_i\|$ and the state of each equation is not the same, which leads to different time steps. A remedy is to compute all the required time steps $\Delta t^1, \Delta t^2, \ldots, \Delta t^K$ first, then take the minimum of them. Then we can make sure that all the configuration vectors will stay inside the configuration domain.*

Remark 3.7 *We do not need to apply the multilevel Monte Carlo method to obtain all the individual expectations. We can focus on the sum of them and only apply the multilevel Monte Carlo method once and simulate all the stochastic differential equations simultaneously to obtain $\boldsymbol{\tau}$.*

3.6 Parallel Computing

As we have indicated in Chapter 1, the Fokker-Planck equation suffers from the curse of dimensionality, therefore we choose to simulate a system of SDEs. The number of SDEs we need to simulate depends on the number of grid points in the physical domain because we have different velocity gradient in the physical domain. Furthermore, for the multi-bead system, we have more SDEs in the system. It is time-consuming to perform the calculation on a single CPU core without any parallel computing. Due to the fact that all the Monte Carlo simulations are independent from each other and the only difference in the calculations is a random number, we can perform parallel computing to do several calculations at the same time in order to reduce the overall computational time.

The parallel computing platform we used for our problem in this thesis is CUDA. CUDA was introduced by NVIDIA in November 2006, and it is designed to perform parallel computing on a GPU core instead of a CPU core to solve complex problems. The advantage of GPU is that is has more cores than a CPU and it is especially well-suited to address problems that can be expressed as data-parallel computations [44]. For example, Monte Carlo simulation reuses the same programming code with different data elements (random numbers generated) in parallel.

In our problem, it is clear that the main computational cost arises from solving the SDEs. As indicated above, Monte Carlo simulations are very well suited to perform parallel computing. If we perform either Algorithm 2 or Algorithm 3 on a single core, we need to run this piece of code N times using the for loop. The only difference in each loop is the value of the random number. The programme can be very slow depending on the size of N. In CUDA, we create a kernel function which runs on the GPU. The kernel function contains the routine for simulating the SDE and it is used to do these parallel executions. We use 1D grid of blocks and 1D set of threads within each block. The number of threads we used per block is 512 and the block size is determined by the number of simulations N. We use the following code to create the CUDA kernel in MATLAB:

```
Kernel = parallel.gpu.CUDAKernel('MLMC.ptx','MLMC.cu');
```

```
threadsPerBlock = 512;
blockSize = ceil(N/threadsPerBlock);
Kernel.ThreadBlockSize = [threadsPerBlock, 1, 1];
Kernel.GridSize = [blockSize, 1, 1];
```

Here MLMC.cu is the file that contains the CUDA code of the kernel and MLMC.ptx is the name of the file that contains the PTX code. We can run the CUDA kernel in MATLAB using the feval command. For different threads, the only difference is the random number generated. The way we generate the random number in the kernel function is the following code:

```
int id=threadIdx.x+blockIdx.x*blockDim.x;
curandState localRNGstate;
curand_init(seed,id,0,&localRNGstate);
```

Here id represents the position of the thread; then we can obtain different random number values for different threads. CUDA manages and executes threads in groups of 32 called warps. 32 threads all execute the same instruction at the same time which is the key to high performance [14].

Chapter 4

Numerical Experiments

In this chapter, we shall present some numerical results for a polymeric fluid based on the algorithm discussed in the previous sections. We will start with some simple cases and then move to our main goal in this thesis: multi-bead simulations. The chapter contains the following numerical experiments:

- We will simulate the FENE stochastic differential equation for which the velocity gradient $\boldsymbol{\kappa}$ is a given known constant matrix throughout the physical domain Ω; therefore we do not need to solve the Navier–Stokes equations in this case. We will then use both the Monte Carlo and the multilevel Monte Carlo method for the FENE stochastic differential equation to compute the extra stress tensor $\boldsymbol{\tau}$ and compare the results with the steady-state solution which has an analytical expression in this case [13]. We will use both Algorithm 2 and Algorithm 3 for the problem and compare their performance.

- We will consider a 2D nonhomogeneous flow in a square domain. For the sake of simplicity, we will first consider using a given velocity $\vec{u}$ and we take $\vec{u}$ to be constant in time; then we will compute the extra stress tensor $\boldsymbol{\tau}$ and compare our results with those of Knezevic [37] who used a deterministic method.

- We will design an experiment in order to verify the exponential decay to the equilibrium solution we discussed in Chapter 2. The reason to do this is that we can use this special case to test our numerical schemes because we know the true behaviour of the flow from the analysis in Chapter 2 and we can

compare our numerical results with the exact solution. First, we will consider the homogeneous flow with $\vec{u} = \vec{0}$ and by Theorem 2.1, we should find that the probability density function ψ of the configuration random variable $\vec{Q}$ converges to the Maxwellian function M with an exponential rate. Then we will investigate the case when the velocity is zero on the boundary of the physical domain Ω. The velocity should then decay to zero exponentially. In this experiment, we will start coupling the FENE stochastic differential equation with the Navier–Stokes equation.

- We will perform a numerical experiment based on the model problem which is a 2D planar expansion flow. The physical domain is an L-shaped domain and the solution is singular in the corner, which makes the problem more complicated than in the case of the flow in a square domain. We will investigate the streamlines of the flow and the extra stress tensor $\boldsymbol{\tau}$.

- All the experiments mentioned above are based on the dumbbell model. We know that when the number of beads increases the deterministic approach is no longer efficient due to the curse of dimensionality; therefore we will perform some experiments using the stochastic approach based on the multi-bead model.

For simplicity, we will use the same function as the initial probability density function (if not mentioned), which is $\psi_0(\vec{q}) = \frac{1}{Z}(1 - \frac{|\vec{q}|^2}{b})^{\frac{b}{2}}$ for all the 2D problems, where Z is a normalisation constant to make sure that the probability density integrates to one. In fact, this is one of the solutions to the Fokker–Planck equation and ψ_0 is also the equilibrium probability density for the configurations. The initial condition for the stochastic differential equation will be randomly generated according to this probability density function. This distribution is not Gaussian, and the random number generator in MATLAB can not generate numbers according to ψ_0, therefore the von Neumann rejection sampling method is used here to generate the initial condition. For more information, the reader is referred to Section 7.3.6 in [49]. The left side of Figure 4.1 is the probability density function ψ_0 and the right side of Figure 4.1 illustrates 1000 pairs of two-dimensional random numbers generated from

$\psi_0(\vec{q}) = \frac{1}{Z}(1 - \frac{|\vec{q}|^2}{10})^5$ and the red dots in the figure can be regarded as the initial condition of the stochastic differential equation.

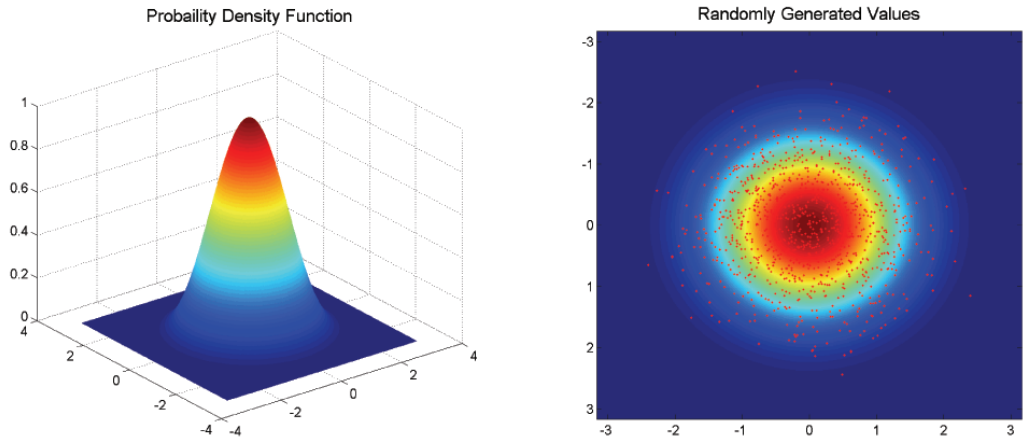


Figure 4.1: The figure illustrates 1000 two-dimensional random numbers whose corresponding density function is ψ_0

4.1 2D Homogeneous Flows

First, we consider a two-dimensional planar extensional flow to investigate the accuracy of our algorithm. The velocity field of the planar extensional flow is given by $\vec{u} = (\sigma x, -\sigma y)$, where σ is the extension rate. The velocity gradient therefore becomes a diagonal matrix $\boldsymbol{\kappa} = \mathrm{diag}(\sigma, -\sigma)$. For such a velocity field, there exists an analytical expression for the steady-state solution of the Fokker–Planck equation (1.18), which has the following form:

$$\frac{1}{Z}\left(1 - \frac{|\vec{q}|^2}{b}\right)^{\frac{b}{2}} \exp(\sigma \mathrm{Wi} \vec{q}^T \boldsymbol{\kappa} \vec{q}), \tag{4.1}$$

with Z being a normalisation constant. We can integrate this function to obtain the exact value of $\boldsymbol{\tau}$ at steady-state and perform the comparison with our numerical result.

In this experiment, we need to solve the following stochastic differential equation:

$$\mathrm{d}\vec{Q}(t) = \left(\boldsymbol{\kappa} \cdot \vec{Q}(t) - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}(t))\right)\mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}}\mathrm{d}\vec{W}(t), \quad t \in (0, T], \tag{4.2}$$

$$\vec{Q}(0) \sim \psi_0 \tag{4.3}$$

subject to a constant velocity gradient: $\boldsymbol{\kappa} = \mathrm{diag}(\sigma, -\sigma)$, and then perform both a standard Monte Carlo and a multilevel Monte Carlo simulation to calculate the extra stress tensor

$$\boldsymbol{\tau}(t) = \frac{b+d+2}{b}\frac{1}{\mathrm{Wi}}\left(\mathbb{E}[\vec{Q}(t) \otimes \vec{F}(\vec{Q}(t))]\right). \tag{4.4}$$

We will present the results with the extension rate $\sigma = 1$ and the other parameters are chosen to be $b = 10, \mathrm{Wi} = 1$ and $d = 2$ since the dimension of the problem is 2. We will use a zero initial condition for our stochastic differential equation as we are interested in the steady-state solution and we can see how the solution converges to the steady-state shown in Figures 4.2 and 4.3. In the paper by Chauvière and Lozinski [13], the authors reported the value of the first component of the extra tensor based for these parameters to be 9.372422, i.e., $\boldsymbol{\tau}_{1,1} = 9.372422$. Another example they considered is to increase the extension rate to $\sigma = 5$, which requires much more refined meshes for solving the Fokker–Planck equation. The value they reported is $\boldsymbol{\tau}_{1,1} = 122.295817$.

4.1.1 Monte Carlo Approach

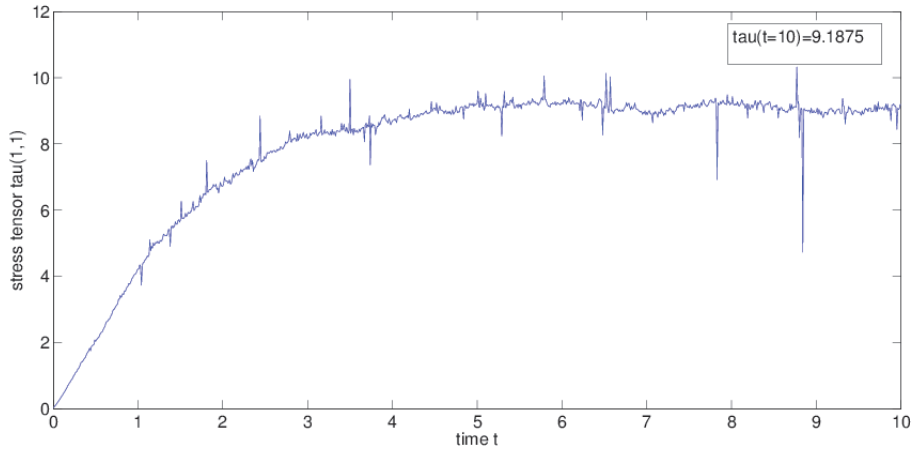


Figure 4.2: Numerical results for $\boldsymbol{\tau}_{1,1}$ from $t = 0$ to $t = 10$, $\Delta t = 0.01$, $N = 10000$

We first solve the stochastic equation (4.3) with the Euler–Maruyama method (with a uniform time step $\Delta t = 0.01$); then we compute $\boldsymbol{\tau}_{1,1}$ by the Monte Carlo

method (the number of simulations is set to $N = 10000$). The extension rate is $\sigma = 1$. In the Monte Carlo approach, we simulate the stochastic process from $t = 0$ to $t = 10$ with a fixed time step Δt and a fixed number of realizations N. Then at each time step, we calculate the expected value.

We can see that the approximation of $\boldsymbol{\tau}_{1,1}$ is approaching 9 in Figure 4.2, but it is not accurate and there are some outliers as well. These bad numerical approximations might come from some very large sample values, which contaminate the results. This can be improved upon by increasing the number of realizations N and decreasing the time step Δt and hence the total computational complexity (cost) will increase to make the mean square error smaller.

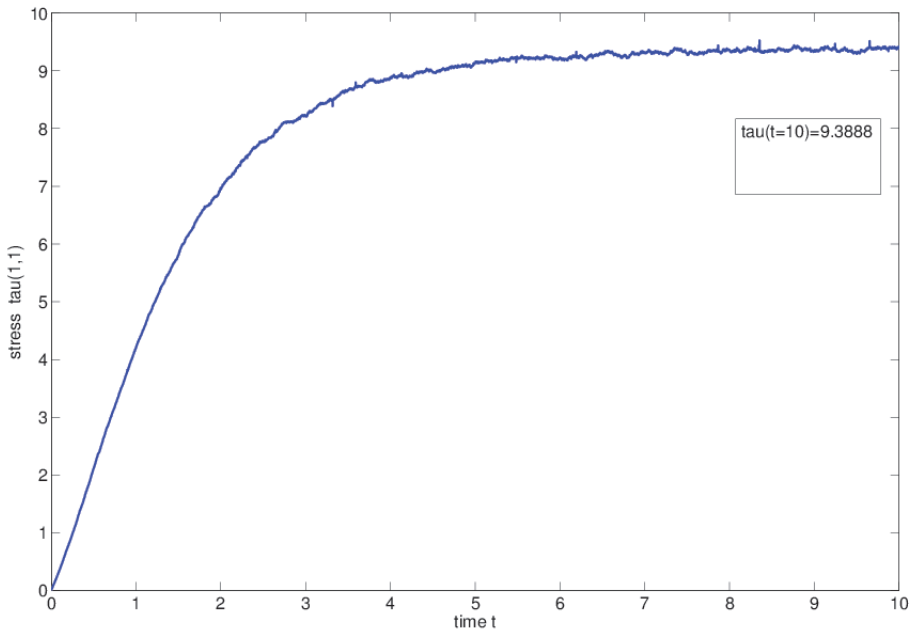


Figure 4.3: Numerical results for $\boldsymbol{\tau}_{1,1}$ from $t = 0$ to $t = 10$, $\Delta t = 0.001$, $N = 100000$

In Figure 4.3, we used a smaller time step $\Delta t = 0.001$ and more samples $N = 100000$. This time the numerical result looks better than the previous one and the first component of the extra-stress tensor $\boldsymbol{\tau}_{1,1} = 9.3888$ at time $t = 10$. There is still some noise because of the large sample produced by the Euler–Maruyama method. Should we wish to obtain more correct digits, we would need to further decrease the

time step and increase the number of samples. Therefore, the Monte Carlo method combined with the Euler–Maruyama solver is of low accuracy. Next we will use the Multilevel Monte Carlo technique to reduce the cost and obtain more accurate results.

4.1.2 Multilevel Monte Carlo Approach

In the previous section, we have seen the low efficiency of the standard Monte Carlo method combined with the fixed time step Euler–Maruyama method. We now apply the multilevel Monte Carlo method to the same problem to assess the improvement. First, we will use Algorithm 2 for the problem and then we will use the improved multilevel Monte Carlo method in Algorithm 3. For Algorithm 2, we need to choose the terminal time T in order to calculate the extra stress tensor $\boldsymbol{\tau}(T)$ and we will pick the times $T = 0, 1, 2, \ldots, 10$ to compute the value of $\boldsymbol{\tau}(T)$. However, for the improved Monte Carlo method, we do not need to specify the terminal time.

Algorithm 2 is the multilevel Monte Carlo technique combined with the adaptive time stepping method (3.20)–(3.25). The adaptive time stepping method at level l and step n has time step

$$\Delta t_l^n = \min\left(b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{\|\vec{Q}_n\|}{\sqrt{b}}\right)\right)^2, C_{\Delta t}\frac{\|\vec{Q}_n\|^2}{\|\boldsymbol{\kappa}\cdot\vec{Q}_n - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}_n)\|^2}\right) R^{-l}, \quad (4.5)$$

where $\vec{Q}_n$ represents the numerical approximation of $\vec{Q}$ at time t_n. Also, we need to use the clamping strategy to make sure that the algorithm will not be stuck inside an infinite loop due to tiny time steps. The clamping radius is $\sqrt{b} - \delta$, where δ is a small number. The effect of different clamping radii is negligible in practice. In the rest of this chapter, the numerical results will be based on taking $\delta = 10^{-6}$.

In Figure 4.4, we computed 11 values using Algorithm 2. The mesh refinement factor was $R = 2$ and the root mean square tolerance was $\epsilon = 0.1$. Table 4.1 shows the numerical results for $\boldsymbol{\tau}$ at time $T = 10$ based on different tolerances ϵ. The relative error is defined as the absolute error divided by the true value.

In order to estimate the cost of the method, we need to estimate the parameters α and β in the complexity theorem for the multilevel Monte Carlo method, which are the weak convergence order and the convergence rate of the variance. In our case

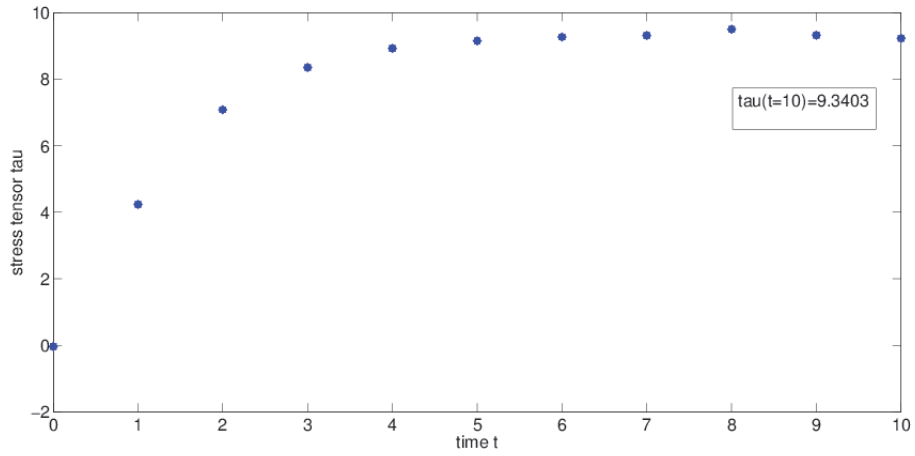


Figure 4.4: Numerical results for $\boldsymbol{\tau}_{1,1}$ from $t = 0$ to $t = 10$ using Algorithm 2, $\epsilon = 0.1$

Tolerance	Computed	Relative error
0.1	9.3403	0.0034
0.05	9.3556	0.0017
0.01	9.3766	0.00045

Table 4.1: Numerical results for the first component of $\boldsymbol{\tau}$ at time $T = 10$ (the value at the steady-state reported in [13] is 9.3724)

the volatility of the stochastic differential equation is constant. The drift function is $\boldsymbol{\kappa} \cdot \vec{Q} - \frac{1}{2\mathrm{Wi}} \frac{\vec{Q}}{1 - \frac{\|\vec{Q}\|^2}{b}}$ with $\boldsymbol{\kappa} = \mathrm{diag}(1, -1)$ and $\mathrm{Wi} = 1$. From Section 3.2.5 we know that the strong convergence order will be 1 if the drift satisfies the contractive Lipschitz properties from Theorem 3.2, however in this case $\|\boldsymbol{\kappa}\|_F = \sqrt{2}$ is larger than $\frac{1}{2\mathrm{Wi}} = \frac{1}{2}$, therefore Assumption 3.2 is invalid and we do not have first order strong convergence and β should be less than 2.

Figure 4.5 shows the $\log_2$(variance) of the Monte Carlo estimators in blue and the multilevel Monte Carlo estimators in red based on 5000 samples for the experiment. The decay rate of the red line tells us that $\beta \approx 1.64 < 2$, however we can still see that the variance of the Monte Carlo estimator is much larger than that of the multilevel level Monte Carlo estimator for this problem, therefore more samples are required to reduce the statistical error using a standard Monte Carlo method. Next we are going to perform a numerical experiment with small $\boldsymbol{\kappa}$ such that Assumption

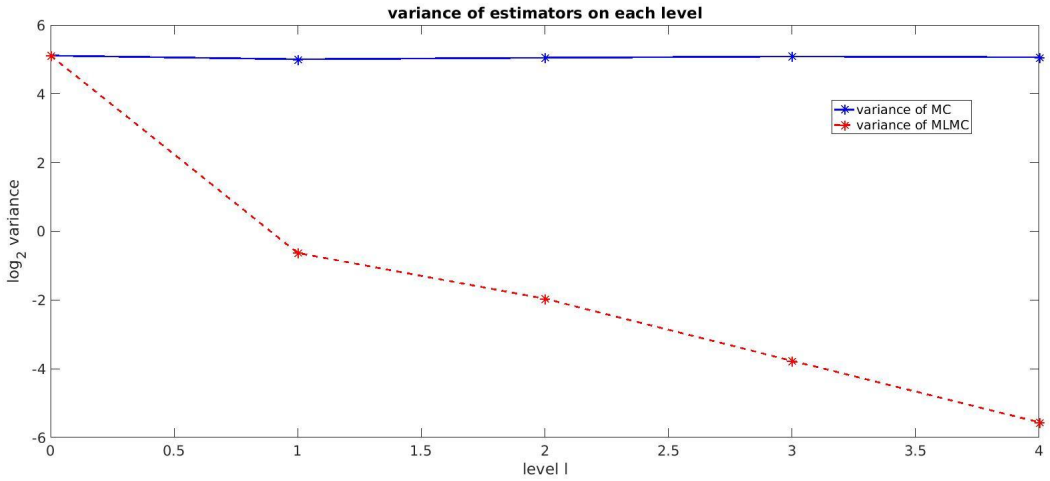


Figure 4.5: Variance of MC estimator and MLMC estimator (Algorithm 2, $\sigma = 1$, $\mathrm{Wi} = 1$, $T = 10$)

3.2 is satisfied. We choose $\mathrm{Wi} = 1$, $\boldsymbol{\kappa} = \mathrm{diag}(0.1, -0.1)$ and therefore the strong convergence order is expected to be 1. Figure 4.6 shows the multilevel Monte Carlo results using Algorithm 2. The top left figure shows the variance decay based on 5000 samples; the top right figure is the mean at different levels; the bottom left figure is the number of simulations in the multilevel Monte Carlo method based on different given tolerances; the bottom right figure is the product of the cost of the method and the mean square error. We can see that the cost of the standard Monte Carlo method is much higher than that of the multilevel Monte Carlo method. We can also see from the bottom right plot that the computational cost for root mean square accuracy is $O(\epsilon^{-2})$ for the multilevel Monte Carlo method. In this experiment the decay rate of the variance of the multilevel Monte Carlo method is approximately 2, which is consistent with Theorem 3.2. Finally we show some results using the multilevel Monte Carlo method for a strong extensional flow, for which the extension rate is $\sigma = 5$, in Table 4.2.

If we use the deterministic approach to solve this problem, i.e., solving the Fokker–Planck equation (1.47) by the finite element method to obtain the numerical solution of the probability density function ψ, the numerical results might be diverging according to the report by Chauvière and Lozinski [13] for strong extensional flow. The

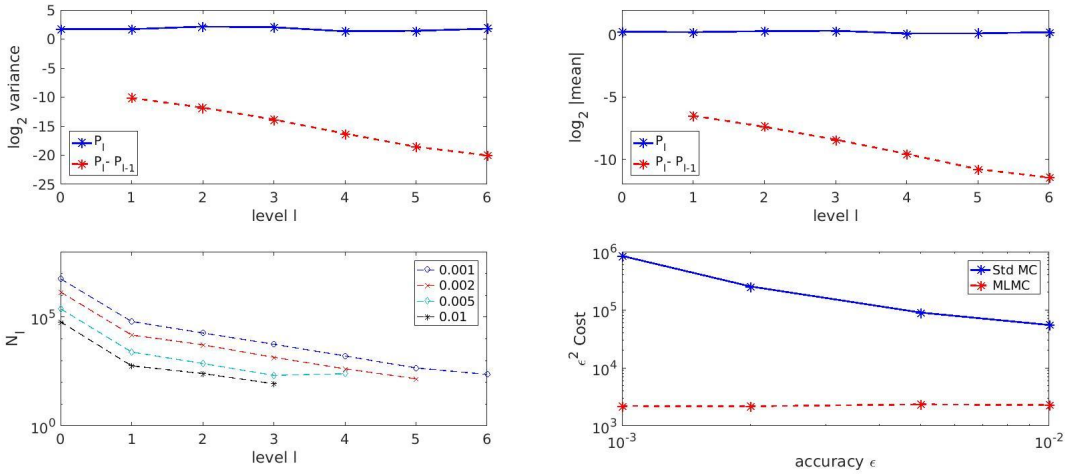


Figure 4.6: The plot of multilevel Monte Carlo results (Algorithm 2, $\sigma = 0.1$, Wi $= 1$, $T = 10$)

Tolerance	Computed	Relative error
0.1	121.7347	0.0046
0.05	122.3721	0.00062
0.01	122.2803	0.00012

Table 4.2: Numerical results for strong extensional flow ($\sigma = 5$) at time $T = 10$ (the value of $\boldsymbol{\tau}_{1,1}$ at the steady-state reported in [13] is 122.2958)

solution to the PDE has two peaks which are close to the boundary and a very fine mesh needs to be used in the numerical scheme to ensure the stability, and the cost of doing that is very high. Similarly, for the stochastic approach, a strong extension rate indicates that the size of $\boldsymbol{\kappa}$ is large, which means that we do not have contractivity and first order strong convergence, and the overall cost will increase. We will discuss the large $\boldsymbol{\kappa}$ problem in a later section.

In this experiment (planar extensional flow), in order to approximate the equilibrium solution for $\boldsymbol{\tau}$, we need to compute $\mathbb{E}[\vec{Q}(t) \otimes \vec{F}(\vec{Q}(t))]$ for a sufficiently large time t if we use Algorithm 2, however Algorithm 3 can deal with such a problem better as long as contractivity is satisfied. Recall that Algorithm 3 uses different lengths of the time interval for different levels and it will stop when the desired mean square error is achieved. First, we need to choose $T_l = \frac{\log R}{2\lambda}(l+1)$ for each level l according to

Section 3.25 where the parameter $\lambda > 0$ comes from Lemma 3.4 (contractivity). We can see that λ plays an important role here, as it controls the size of the time interval. Lemma 3.4 suggests that the rate of convergence is exponential depending on λ, and a larger value of λ implies faster convergence. This is consistent with Algorithm 3 because a larger value of λ implies a smaller T_l. For our problem, we can choose the value of λ by computing $\|\boldsymbol{\kappa}\|_F - \frac{1}{2\mathrm{Wi}}$. Suppose that we take $\mathrm{Wi} = 1$ and $\boldsymbol{\kappa} = \mathbf{0}$; then the range of λ is $0 < \lambda \leq 0.5$.

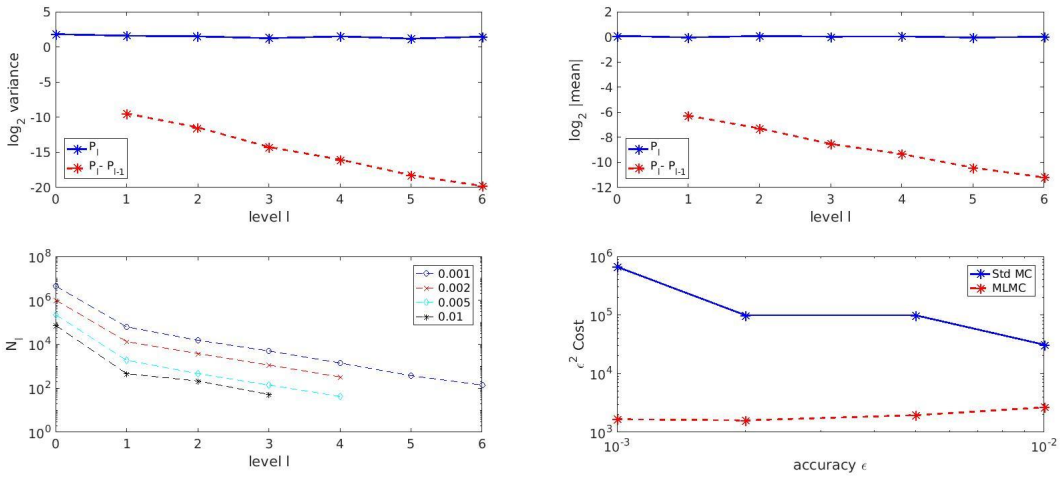


Figure 4.7: The plot of multilevel Monte Carlo results (Algorithm 3, $\sigma = 0.1$, $\mathrm{Wi} = 1$, $\lambda = 0.05$)

Figure 4.7 shows the results obtained by multilevel Monte Carlo using Algorithm 3 with $\sigma = 0.1$, $\mathrm{Wi} = 1$, $\lambda = 0.05$. The variance plot is based on 5000 samples. If we repeat this experiment with a smaller value of λ, which means that we simulate over a longer time interval at each level according to $T_l = \frac{\log R}{2\lambda}(l+1)$. Due to contractivity, the difference between the fine and coarse paths exponentially decays with time, and therefore we expect a smaller variance. However we may not benefit from using smaller λ because we need to simulate over at longer time interval for each simulation. The bottom left plot shows the number of simulations taken for each level, given different tolerances. It also shows how many levels we need in order to achieve the tolerance; then we are able to compute T_L, where L is the finest level.

In Section 2.1 we showed that the solution to the Fokker–Planck equation converges to the Maxwellian function exponentially when $\vec{u} = \vec{0}$ and the rate is independent of the choice of the initial datum. This corresponds to taking $\boldsymbol{\kappa} = \mathbf{0}$ in the SDE (4.2) and the probability density function of the random variable $\vec{Q}$ should converge to the Maxwellian. Lemma 3.3 in Section 3.2.5 tells us that there exists a unique invariant measure and the convergence rate is exponential. Combining Lemma 3.3 with Lemma 3.4, we can see that if the SDE has the contractivity property, this rate is also independent of the initial datum of the SDE. The problem is that we are not solving the Fokker–Planck equation to obtain ψ; therefore it is difficult to show that the probability density function of the random variable sequence generated by the SDE converges to the Maxwellian. The idea is therefore to simulate the moment-generating function of both the Maxwellian and our numerical simulations, and compare the results.

In order to show that the probability density function of a 2D random variable $\vec{Q} = (Q_1, Q_2)^T$ converges to the Maxwellian, we will apply Algorithm 3 to the SDE (4.2) with zero velocity gradient and simulate the moment-generating functions (a function of the real number y): $\mathbb{E}[\exp(Q_1 y)]$ and $\mathbb{E}[\exp(Q_2 y)], y \in \mathbb{R}$. Then we will compare the results with the moment-generating function based on the Maxwellian to see whether they match. If we expand the moment-generating function, we will find that it is the combination of all the moments of the distribution. The reason to use the moment-generating function here is because of an important property: if two distributions have the same moment-generating function, then they are identical at almost all points [21]. We can not simulate the moment-generating function at all points of the real line in a computer; we will therefore only simulate $\mathbb{E}[\exp(Q_1 y)]$ and $\mathbb{E}[\exp(Q_2 y)]$ on a finite interval. We use $\vec{Q}(0) = \vec{0}$, $b = 10$, $\lambda = 0.05$ and Wi=1 in the experiment.

Figure 4.8 is the moment-generating function plotted on the interval $[-1, 1]$. The red curve is the moment-generating function generated by the Maxwellian function and the blue curve is the numerical simulation using Algorithm 3 with tolerance $\epsilon = 0.01$. We can see that our numerical results match the Maxwellian very well, which suggests that the probability density function of $\vec{Q}$ is also close to the Maxwellian.

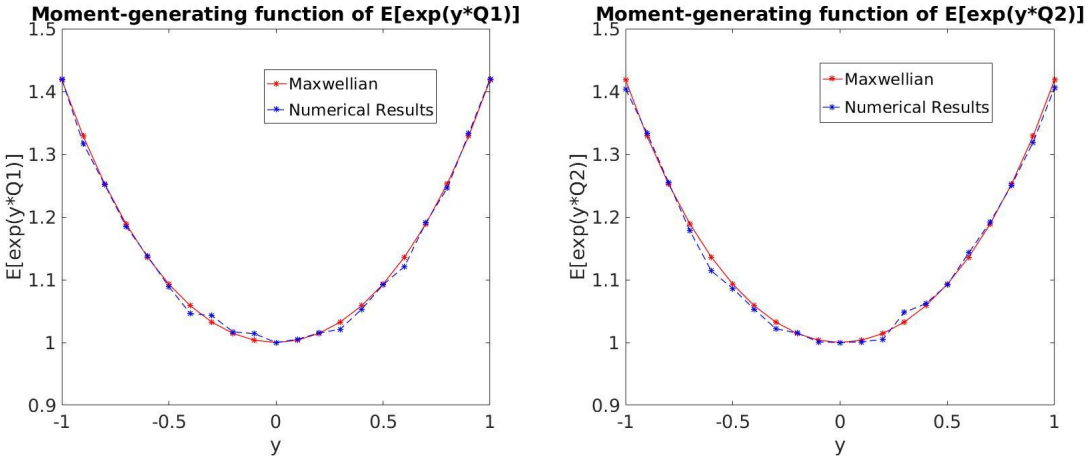


Figure 4.8: The moment-generating function of the Maxwellian and its numerical simulations

4.2 2D Non-Homogeneous Flows

In this section, we will consider the 2D non-homogeneous flow. In the case of a non-homogeneous flow, the velocity gradient $\boldsymbol{\kappa}(\vec{x},t)$ is no longer constant, but now also depends on space and time. For the homogeneous flow considered in the last section, we only had to simulate one SDE and $\boldsymbol{\tau}$ is only based on that simulation; however in this section, due to the different values of $\boldsymbol{\kappa}(\vec{x},t)$ at different space and time points, the number of SDEs we need to simulate is as many as the number of the grid points in the physical domain $\Omega \times (0,T)$; therefore the cost of simulating non-homogeneous flows is much higher. The stochastic differential equation depends on the space variable $\vec{x}$ implicitly and now becomes:

$$\mathrm{d}\vec{Q}(\vec{x},t) = \left(\boldsymbol{\kappa}(\vec{x},t)\cdot\vec{Q}(\vec{(}x),t) - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}(\vec{x},t))\right)\mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}}\mathrm{d}\vec{W}(t), \quad t\in(0,T], \tag{4.6}$$

$$\vec{Q}(\vec{x},0) \sim \psi_0 \quad \forall \vec{x}\in\Omega. \tag{4.7}$$

In addition the extra stress tensor also depends on the spatial variable now:

$$\boldsymbol{\tau}(\vec{x},t) = \frac{b+d+2}{b}\frac{1}{\mathrm{Wi}}\left(\mathbb{E}[\vec{Q}(\vec{x},t)\otimes\vec{F}(\vec{Q}(\vec{x},t))] - \mathbf{I}\right). \tag{4.8}$$

4.2.1 2D Steady Non-Homogeneous Flow

A flow is said to be steady if the velocity and pressure at every point in the flow domain are independent of time. In such a case $\boldsymbol{\kappa}$ will only depend on $\vec{x}$, which means that we only need to compute $\boldsymbol{\kappa}$ once. In this section, we will show a numerical example based on a given steady velocity field $\vec{u}(\vec{x})$ in the unit square domain.

We take the velocity field $\vec{u}(\vec{x})$ to be the solution of the steady incompressible Navier–Stokes equations in the unit square domain $\Omega = (0,1)^2$ of the following form:

$$(\vec{u} \cdot \nabla_x)\vec{u} = -\nabla_x p + \Delta_x \vec{u} + \vec{f}, \tag{4.3}$$

$$\nabla_x \cdot \vec{u} = 0, \tag{4.4}$$

where the body force is $\vec{f}(x,y) = (5\sin(2\pi y), -5\sin(2\pi x))^T$. A homogeneous Dirichlet boundary condition $\vec{u} = \vec{0}$ on $\partial\Omega$ is imposed. This is an enclosed flow because $\vec{u} \cdot \vec{n} = 0$ on the boundary.

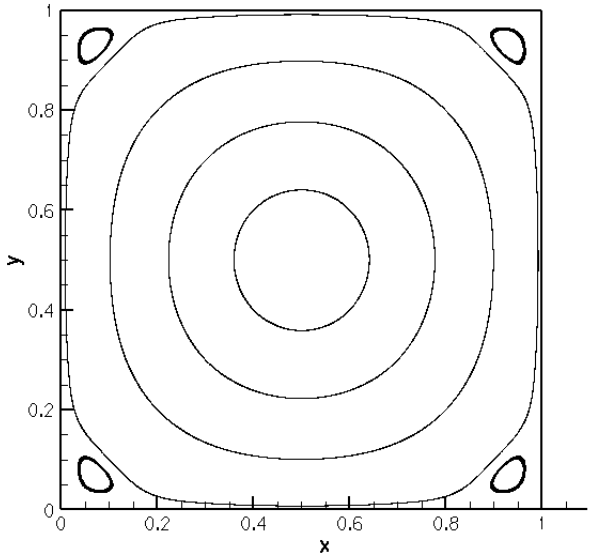


Figure 4.9: Streamlines of the velocity field $\vec{u}$. This is the solution of the steady Navier–Stokes equations with body force $\vec{f}(x,y) = (5\sin(2\pi y), -5\sin(2\pi x))^T$.

Figure 4.9 shows the streamlines of the velocity field $\vec{u}$ obtained by solving (4.3), (4.4) with a Taylor–Hood mixed finite element scheme, see [36]. A continuous piecewise biquadratic approximation is used for $\vec{u}$ and a continuous piecewise bilinear

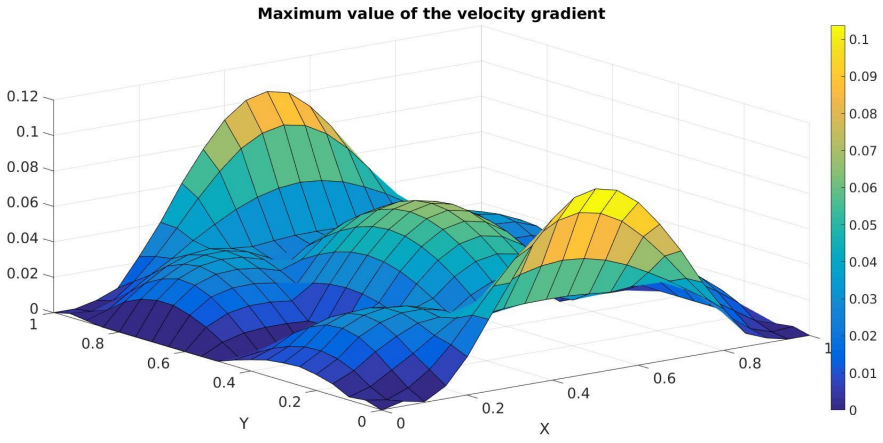


Figure 4.10: The second component of $\boldsymbol{\tau}(\vec{x}, t)$ at $t = 0.2$

approximation is used for the pressure p. The mesh sizes are $\Delta x = \Delta y = 0.05$. We will use the same mesh for the SDE and compute $\boldsymbol{\tau}(\vec{x})$ at every node based on this velocity field. The velocity gradient $\boldsymbol{\kappa}(\vec{x})$ is then approximated by a central difference scheme. Figure 4.10 shows the maximum value of $\boldsymbol{\kappa}(\vec{x})$ in the physical domain, i.e., $\max(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial u}{\partial y})$. We can see that $\boldsymbol{\kappa}$ is not large in this experiment, which is helpful for us because Assumption 3.2 will be satisfied and then Lemma 3.4 will apply. We will study a more difficult problem later when we move to an L-shaped domain, which has a large value of $\boldsymbol{\kappa}$ near the reentrant corner of the domain.

Figure 4.11 is the plot from Knezevic [37] who used a deterministic approach to compute $\boldsymbol{\tau}(\vec{x}, t)$ at $t = 0.2$. We want to compare our results with those of Knezevic and therefore Algorithm 2 will be used to simulate the SDE (4.2) based on the velocity $\vec{u}$ given above for a fixed time interval $t \in [0, 0.2]$. We then compute $\boldsymbol{\tau}(\vec{x}, 0.2)$ at every grid point in the unit square domain. The applications of Algorithm 2 at each grid point used to compute $\boldsymbol{\tau}(\vec{x}, 0.2)$ are independent of each other, and therefore parallel computing (CUDA) reduces the computing time.

The parameters used in this experiment are $b = 12$, $\mathrm{Wi} = 1$. Also, the initial condition is $\psi_0 = \frac{1}{Z}(1 - \frac{|\vec{q}|^2}{b})^{\frac{b}{2}}$ (Z is the normalisation constant). We randomly generate the initial value for $\vec{Q}$ according to ψ_0 as we discussed in the beginning of the chapter. The true solution for this problem is not available, hence we compare our results with those of Knezevic [37]. The author used a deterministic approach in

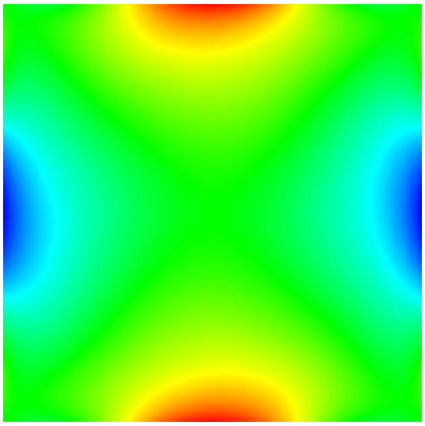


Figure 4.11: The second component of $\boldsymbol{\tau}(\vec{x}, t)$ at $t = 0.2$ (Knezevic result)

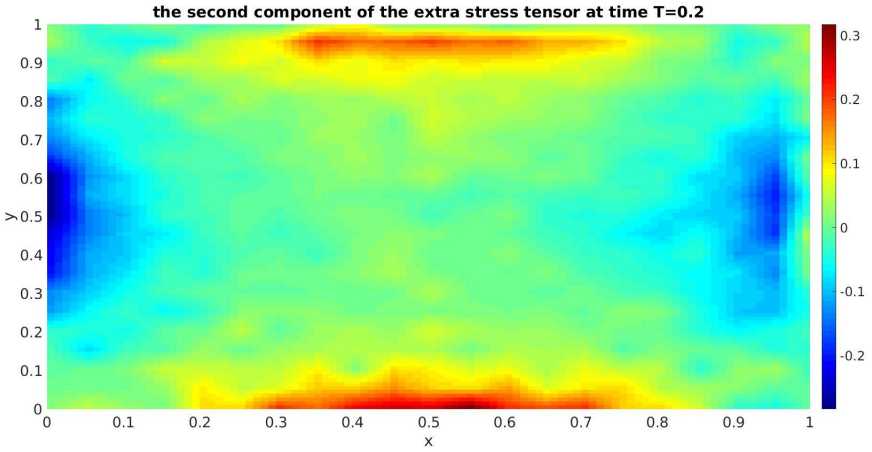


Figure 4.12: The second component of $\boldsymbol{\tau}(\vec{x}, t)$ at $t = 0.2$

his work to perform this numerical experiment. Figure 4.12 is the second component of $\boldsymbol{\tau}(\vec{x}, t)$ at time $t = 0.2$ using the stochastic approach. The root mean square tolerance used in the multilevel Monte Carlo is $\epsilon = 0.05$ and the refinement factor is $R = 2$ in order to generate these figures. The value of $\boldsymbol{\tau}$ in Figure 4.11 ranges from -0.229 (blue) to 0.229 (red) according to the work of Knezevic. We can see that our results (Figure 4.12) are close to what Knezevic obtained. The left plot in Figure 4.13 suggests that the decay rate of the variance is near 2. In this problem, the variance is a function of the space variable, i.e., $V(\vec{x})$, because at each point in the physical space there will be an MLMC simulation associated with it. The plot is based on taking the maximum value of $V(\vec{x})$. The right plot in Figure 4.13 is the number of samples

needed for the MLMC estimator at different levels subject to different tolerances. Also we only consider the maximum value among all the simulations.

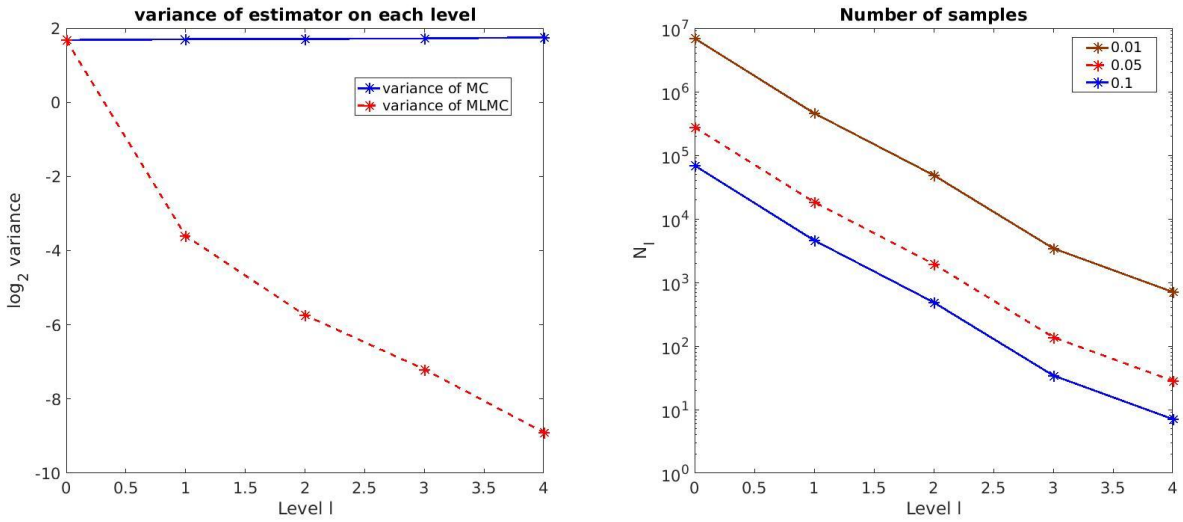


Figure 4.13: MLMC results for the 2D steady non-homogeneous flow

4.2.2 2D Unsteady Non-Homogeneous Flow

In this section, we investigate the unsteady flow whose velocity gradient $\boldsymbol{\kappa}(\vec{x},t)$ depends on time as well, and therefore instead of solving equation (4.6)–(4.8), now we need to focus on the entire coupled system:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla_x\vec{u} = -\nabla_x p + \frac{\gamma}{\mathrm{Re}}\Delta_x\vec{u} + \nabla_x\cdot\boldsymbol{\tau}, \tag{4.5}$$

$$\nabla_x\cdot\vec{u} = 0, \tag{4.6}$$

$$\mathrm{d}\vec{Q}(\vec{x},t) = \left(\boldsymbol{\kappa}(\vec{x},t)\cdot\vec{Q}(\vec{x},t) - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}(\vec{x},t))\right)\mathrm{d}t + \sqrt{\frac{1}{\mathrm{Wi}}}\mathrm{d}\vec{W}(t), \tag{4.7}$$

$$\boldsymbol{\tau}(\vec{x},t) = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{ReWi}}\left(\mathbb{E}[\vec{Q}\otimes\vec{F}(\vec{Q})] - \mathbf{I}\right), \tag{4.8}$$

$$\vec{u}(\vec{x},0) = \vec{u}_0(\vec{x}), \quad \vec{Q}(\vec{x},0)\sim\psi_0(\vec{q}). \tag{4.9}$$

We consider a 2D unsteady non-homogeneous flow on a square domain with zero boundary condition and no external body force acting on the flow. According to the results proved in Chapter 2, the velocity of the flow should decay to zero exponentially

and the solution to the Fokker–Planck equation should converge to the Maxwellian M. In addition, the asymptotic decay rate is independent of the choice of the initial data.

We need to pick a divergence-free velocity field with homogeneous Dirichlet conditions applied on all boundaries as the initial datum $\vec{u}_0(\vec{x})$. In the last section, we computed the numerical solution to the steady incompressible Navier–Stokes equations (4.3), (4.4) with boundary condition $\vec{u} = \vec{0}$ on $\partial\Omega$; therefore we can use this result as the initial datum for the Navier–Stokes equations.

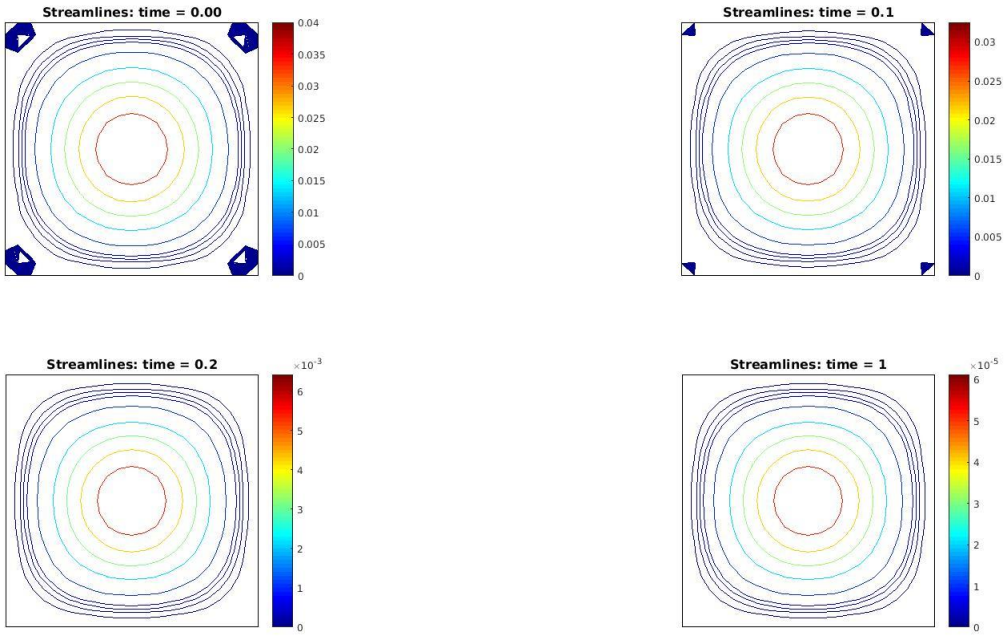


Figure 4.14: Contour plot

Figure 4.14 shows the contour plot of the stream function at times $t = 0, 0.1, 0.2, 1$ by simulating the coupled system (4.5) to (4.9). The initial condition for the velocity is suggested above and the initial condition for the pressure is taken to be zero throughout Ω. The initial condition for the SDE, $\vec{Q}(\vec{x}, 0)$, is zero in the numerical experiment. The parameters used in this experiment are $b = 12$, $\mathrm{Wi} = 1, \mathrm{Re} = 1, \gamma = 0.5$.

According to Section 3.4.2, we apply Algorithm 2 to simulate the SDE (4.7) given

the velocity field $\vec{u}(\vec{x}, t_{n-1})$ over the time interval $[t_{n-1}, t_n]$ (assuming $\vec{u}$ does not change over the time interval) and estimate the extra stress tensor $\boldsymbol{\tau}(\vec{x}, t_n)$, (4.8), (in the same way as we did in Section 4.2.1). After computing $\boldsymbol{\tau}(\vec{x}, t_n)$, we can substitute it into equations (4.5), (4.6), then use the Taylor–Hood mixed finite element scheme for the incompressible Navier–Stokes equations (4.5), (4.6) (the mesh size is $\Delta x = \Delta y = 0.05$). The time stepping for the Navier–Stokes equations from t_n to t_{n+1} is done by the forward Euler scheme. For the weak formulation of the problem, we refer to Section 3.4.1. In this experiment, we take the time step $\Delta t = 0.05$ for the Navier–Stokes equations.

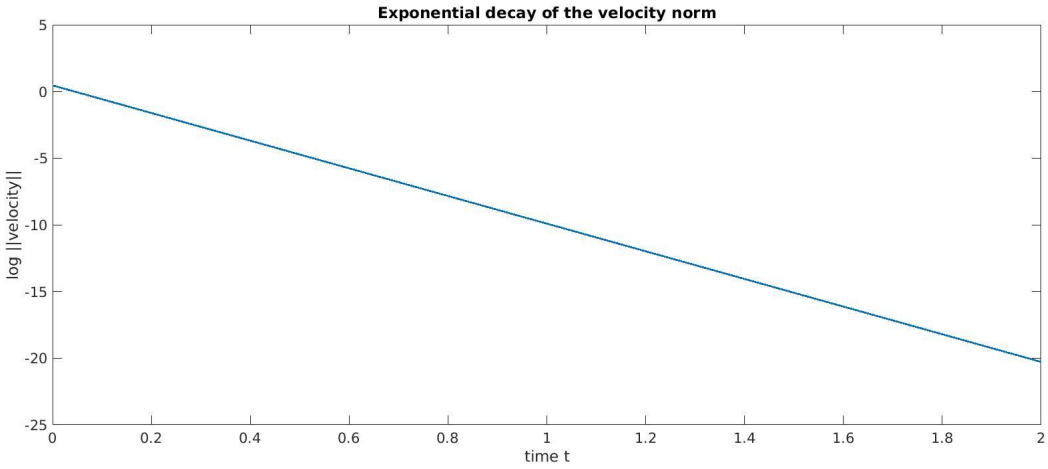


Figure 4.15: Decay of the velocity norm

Figure 4.15 depicts the L^2 norm of the velocity field $\vec{u}$ over the whole flow domain. We can see that the norm decays to zero exponentially as Theorem 2.1 predicts. Figure 4.16 is the velocity gradient in the flow domain at time $t = 0$ and $t = 2$ and we plot the maximum value among the four components of $\boldsymbol{\kappa}(\vec{x}, t)$. We can see that the velocity gradient $\boldsymbol{\kappa}(\vec{x}, t)$ also tends to zero as the velocity decays and hence we have $\boldsymbol{\kappa}(\vec{x}, t) \to \mathbf{0}$ as $t \to \infty$:

$$\lim_{t\to\infty} \boldsymbol{\kappa}(\vec{x}, t) = \lim_{t\to\infty} \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix} \to \mathbf{0}.$$

In Section 4.1, we introduced the steady-state solution of the Fokker–Planck equation given the velocity gradient is a diagonal matrix (equation (4.1)). Plugging the

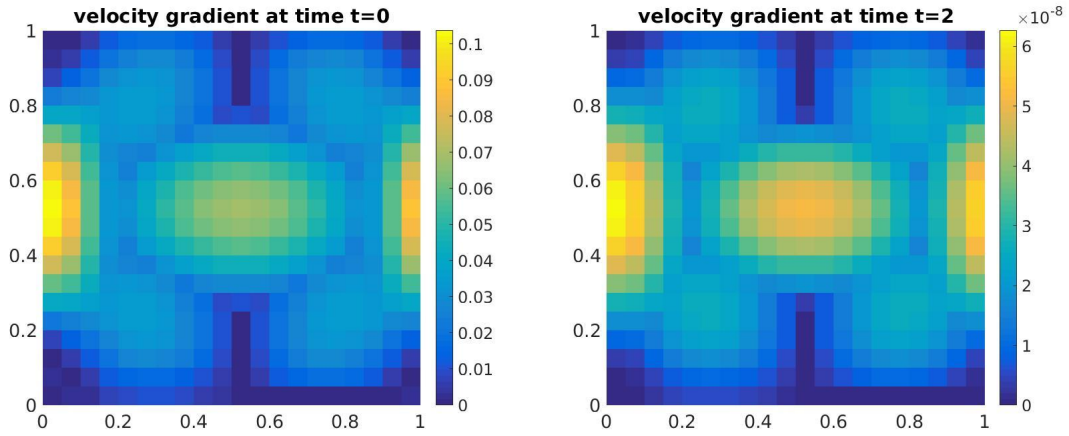


Figure 4.16: The maximum value of the velocity gradient $\boldsymbol{\kappa}(\vec{x})$ at time $t = 0$ (left) and $t = 2$ (right)

zero matrix for $\boldsymbol{\kappa}$ into equation (4.1), we can see that the steady-state solution is exactly the Maxwellian function, which is in agreement with Theorem 2.1.

4.2.3 Plane Expansion Flow in an L-shaped Domain

In this section, we consider the expansion flow in an L-shaped domain. Similarly to the procedure in the last section, we need to solve a time-dependent Navier–Stokes equations coupled with a stochastic differential equation. The difference is that we have changed the flow domain from a unit square to an L-shaped domain and we will impose a Poiseuille flow profile on the inflow boundary, which is a parabolic function vanishing on the top and bottom wall. A natural boundary condition (zero Neumann boundary condition) will be applied on the outflow boundary and we impose a no-slip condition on the rest of the wall.

Figure 4.17 is the mesh in the flow domain with mesh sizes $\Delta x = \Delta y = 0.125$. The channel is 6 units long, 1 unit wide on the inflow boundary and 2 units wide on the outflow boundary. The inflow boundary condition at $x = -1$ is $\vec{u}(x, y) = 4y(1-y)\vec{e}_x$. We use the zero initial conditions for the velocity and pressure: $\vec{u}_0(\vec{x}) = \vec{0}$. The parameters used in this experiment are $b = 12$, $\mathrm{Wi} = 0.8, \mathrm{Re} = 1, \gamma = 0.5$. We take the time step $\Delta t = 0.05$ for the Navier–Stokes equations.

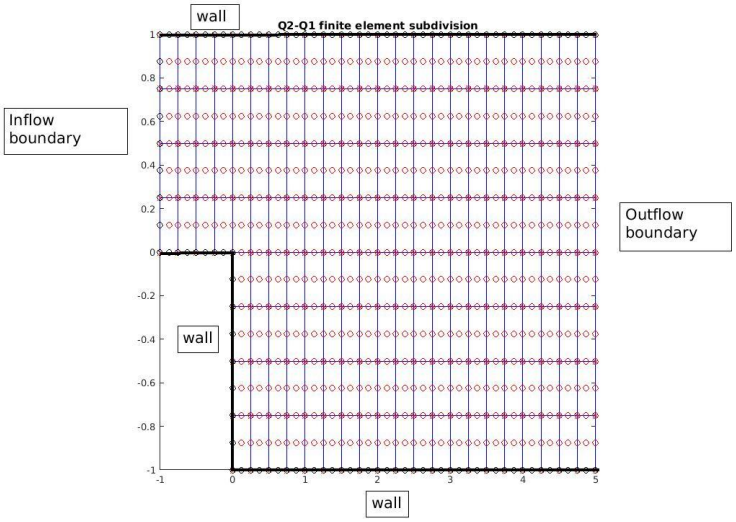


Figure 4.17: The mesh for the expansion flow

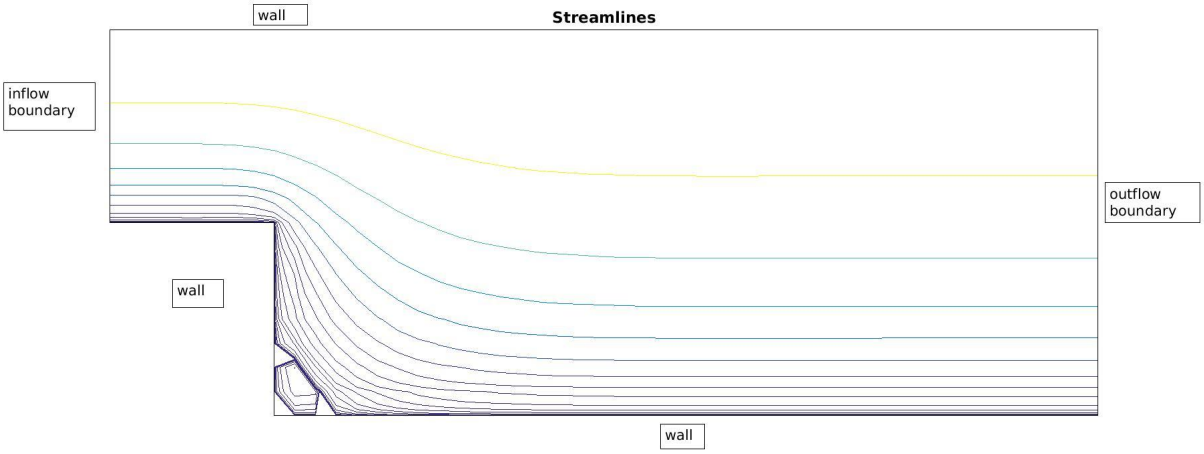


Figure 4.18: Streamlines of the velocity field $\vec{u}(\vec{x}, t)$ at time $t = 5$

Figure 4.18 shows the streamlines for the velocity field at time $t = 5$. The maximum value of the velocity gradient of the numerical solution at this time is around 10 and it occurs near the reentrant corner of the flow domain. The large value of $\boldsymbol{\kappa}$ makes this problem more difficult to solve. The first reason is that Assumption 3.2 is invalid and we do not have the contractivity. Without the contractivity we are unable to use Algorithm 3 and the strong convergence order using the adaptive time stepping method is worse. Also the variance of the Monte Carlo estimator becomes larger, so we need to perform more simulations to achieve the same mean

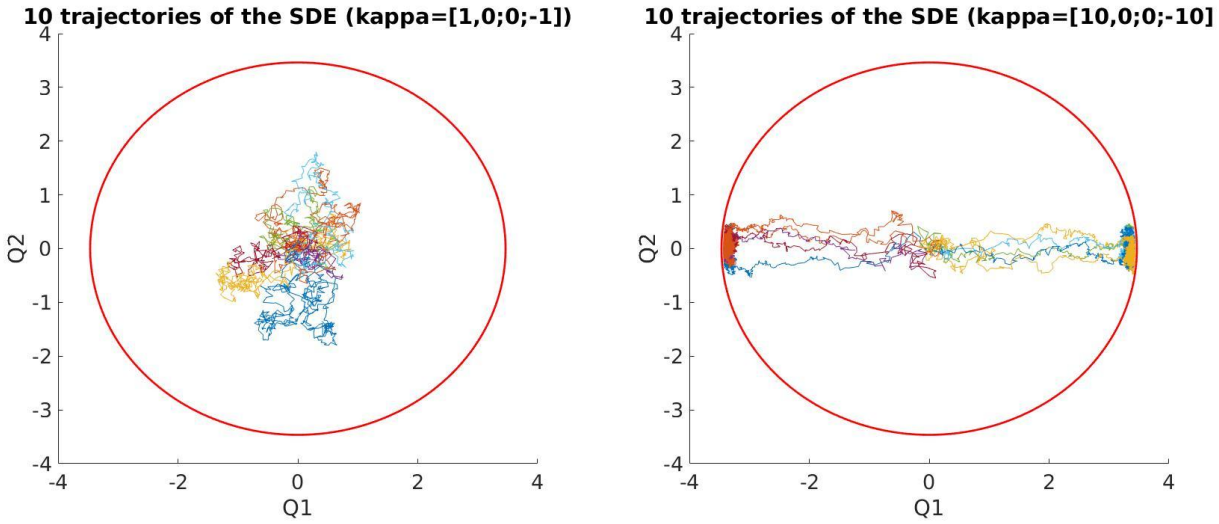


Figure 4.19: The trajectories of the SDE simulations using the adaptive time stepping scheme with different velocity gradients

square error. The reason for a large variance is that the drift term in the SDE (4.7) is dominated by the large value of $\boldsymbol{\kappa}$ when $\|\vec{Q}\|$ is small. Recall that the drift is $\boldsymbol{\kappa}(\vec{x},t)\cdot\vec{Q}(\vec{x},t)-\frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}(\vec{x},t))$. The FENE spring force $-\frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}(\vec{x},t))$ always tries to shrink the lengths of the springs while the velocity gradient $\boldsymbol{\kappa}(\vec{x},t)\cdot\vec{Q}(\vec{x},t)$ tries to stretch the springs if the velocity gradient is positive. When $\|\vec{Q}\|$ is small, the effect of the FENE spring force is smaller than the effect of $\boldsymbol{\kappa}$, so then the trajectory of the SDE will reach the boundary of the configuration domain sooner. Figure 4.19 shows 10 simulations of the SDE until time $t=1$ using the adaptive time stepping scheme with different values of the velocity gradient, starting at the origin. The left plot uses $\boldsymbol{\kappa}=\mathrm{diag}(1,-1)$ and the right plot uses $\boldsymbol{\kappa}=\mathrm{diag}(10,-10)$. We can clearly see that the trajectories of the SDE with large velocity gradient exhibit oscillations near the boundary at the left and right edge of the configuration domain. This is because the equation for the Q_1 component is associated with a large positive value (10) and the equation for the Q_2 component is associated with a large negative value (-10). From Figure 4.19, we can also say that the variance of the Monte Carlo estimator shown in the right plot is larger than the variance of $\vec{Q}$ shown in the left plot. For the multilevel Monte Carlo method, we still have the variance decay as the level increases, however the decay rate is worse if Assumption 3.2 is invalid. Also the variance will be larger

at level 0, which requires more simulations. We will discuss the issue of $\boldsymbol{\kappa}$ in Section 4.4.2.

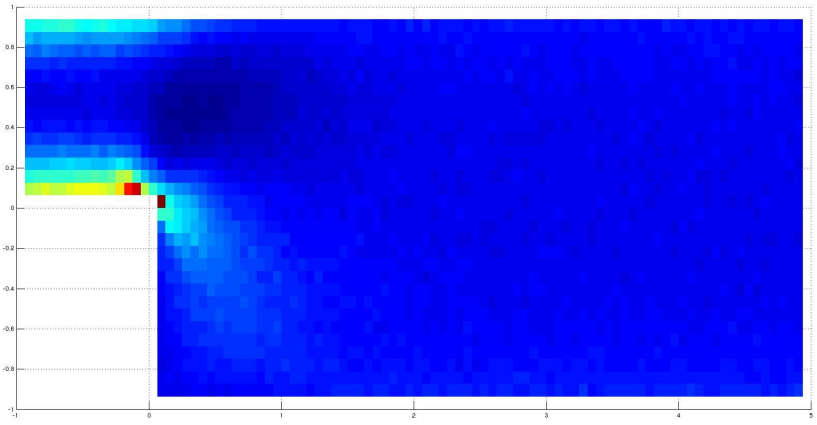


Figure 4.20: The first component of $\boldsymbol{\tau}(\vec{x}, t)$ at $t = 5$ for expansion flow in an L-shaped domain

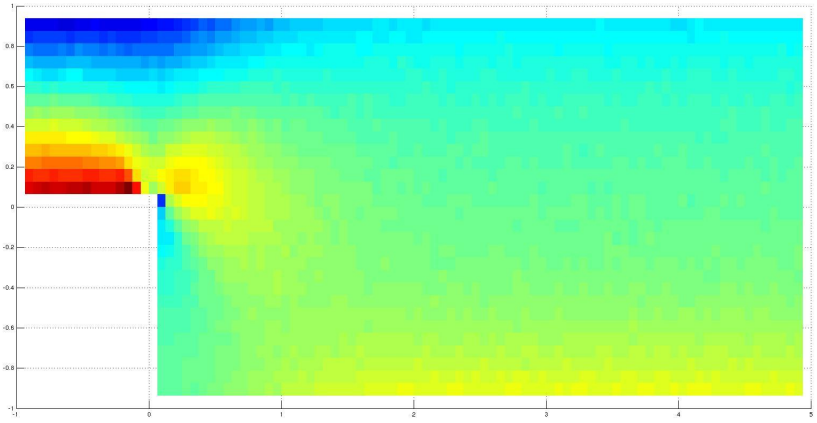


Figure 4.21: The second component of $\boldsymbol{\tau}(\vec{x}, t)$ at $t = 5$ for expansion flow in an L-shaped domain

Figures 4.20 and 4.21 are the plots of the first and second components of the extra stress tensor. In [37], the author used a deterministic approach to simulate a contraction flow in an L-shaped domain. The author reports a problem with the corner singularity so he solved the problem with a rounded corner.

4.3 Multi-Bead Simulations

So far, we have reported numerical experiments based on the dumbbell model. In this section, we consider the multi-bead model, which has $K+1$ beads connected with K springs. For the multi-bead model with K springs, the maximum extension to which the whole spring can be stretched is the sum of the individual lengths of the stretched springs. In order to compare the result with the dumbbell model, we set $\sqrt{b_i} = \frac{\sqrt{b}}{K}$, where $\sqrt{b_i}$ represents the maximum extension of the ith spring and $\sqrt{b}$ is the maximum extension in the dumbbell model. We will repeat the numerical experiments in Section 4.1.2, 4.2.2 and 4.2.3 with our multi-bead model.

First, we consider the planar extensional flow with $\boldsymbol{\kappa} = \mathrm{diag}(1,-1)$ in two space dimensions. From the previous experiment, we know that there is a steady-state solution to the Fokker–Planck equation and the reference value for $\boldsymbol{\tau}_{1,1} = 9.372$ for $b = 10$ and $\mathrm{Wi} = 1$. We apply Algorithm 2 to the same experiment based on the multi-bead model with tolerance 0.01. Table 4.3 shows the numerical results based on the dumbbell model and the multi-bead model with $K = 2, \ldots, 5$. We can see that when we increase the number of beads, the extra stress tensor also increases with a linear rate. The joint distribution of all the configuration variables converges to the Maxwellian $M(\vec{Q})$, which is the product of the partial Maxwellians $M_i(\vec{Q}_i) = Z_i(1-\frac{\|\vec{Q}_i\|^2}{b_i})^{\frac{b_i}{2}}$. Because we use the same b_i for each spring, all configuration variables $\vec{Q}(i)$ have the same probability density function, and therefore the extra stress tensor in this experiment can be simplified as:

$$\boldsymbol{\tau}(\vec{x},t) = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{ReWi}}\left(K\mathbb{E}[\vec{Q}_i(t)\otimes\vec{F}_i(\vec{Q}_i(t))] - \mathbf{I}\right).$$

Next, we consider a steady non-homogeneous flow using the multi-bead model. Figure 4.22 is the contour plot of the velocity based on the experiment in Section 4.2.2 using the dumbbell model and the multi-bead model with the number of springs $K = 3, 6, 10$ at time $t = 0.1$. The top left plot is the same as the top right plot in Figure 4.14, as expected. The rest of the figure is the velocity field based on different numbers of springs. Compared to Figure 4.14, we can see that with more beads considered, the velocity decays more quickly. Theorem 2.1 tells us that the

Number of spring K	$\boldsymbol{\tau}_{1,1}$
1	9.3724
2	9.7164
3	10.7636
4	11.9846
5	13.2752

Table 4.3: Numerical results for $\boldsymbol{\tau}_{1,1}$

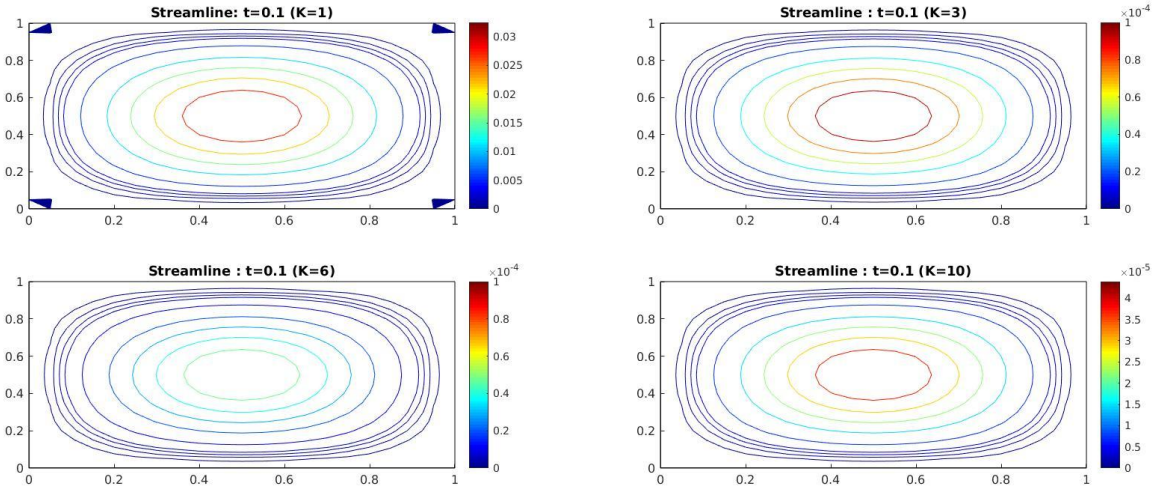


Figure 4.22: The contour plot of the stream function at time $t = 0.1$. Top left: dumbbell model; Top right $K = 3$ springs; Bottom left: $K = 6$ springs; Bottom right: $K = 10$ springs

rate at which the fluid relaxes to equilibrium is given by $\min(\frac{\gamma}{P^2\mathrm{Re}}, \frac{a_0\delta}{\mathrm{Wi}})$, where γ is the dimensionless viscosity of the solvent, a_0 is the minimum eigenvalue of the tridiagonal matrix $\mathbf{A} := (A_{ij})_{i,j=1}^{k} = \mathrm{tridiag}[-1, 2, -1]$ defined in equation (1.19), Wi is the Weissenberg number, Re is the Reynolds number, P is Poincaré's constant and δ is as in the inequality (2.66). By increasing the number of beads, all the parameters remain the same except for a_0. We can see that a_0 decreases as K increases in Figure 4.23, however the decay rate of a_0 suggests that we do not need to use a large value of K because the change becomes smaller for large K.

We repeat the numerical experiment we did in Section 4.2.3 with our multi-bead model. The parameters we used are the same as in the previous experiment and we

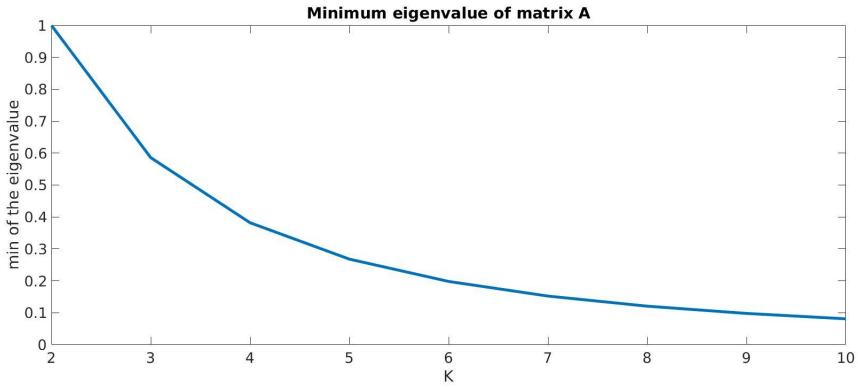


Figure 4.23: The minimum eigenvalue of the tridiagonal matrix: a_0

should use b_i instead of b for each spring. Therefore the total maximum extensibility remains unchanged. The contour plots of the velocity field at time $t = 5$ are shown in Figure 4.24. The top left plot shows the velocity streamlines generated by a Newtonian fluid, i.e., there is no extra stress tensor $\boldsymbol{\tau}$ in the Navier–Stokes equations. The top right plot shows the results based on the dumbbell model ($K = 1$) and the bottom plots are the numerical results using the multi-bead model. The number of springs we used is 6 and 10 respectively. We can see that the vortex near the corner becomes smaller when we increase the number of beads. From the pure Newtonian to the dumbbell model, the change in the vortex is significant, but there is no significant change when we increase the number of springs from 6 to 10. The top two plots suggest that adding the extra stress tensor $\boldsymbol{\tau}$ into the Navier–Stokes equations is similar to increasing the viscosity, therefore we can see that the vortex becomes smaller. The bottom two plots suggest that it is not necessary to use many beads in this experiment because the influence becomes smaller and smaller when we add more beads in the model.

Number of spring K	$\boldsymbol{\tau}_{1,1}(t)$	$\boldsymbol{\tau}_{1,2}(t)$	$\boldsymbol{\tau}_{2,2}(t)$
1	15.7	1.41	11.5
6	16.4	1.5	11.9
10	16.5	1.53	12.1

Table 4.4: The maximum value of $\boldsymbol{\tau}(\vec{x}, t)$ at time $t = 5$ over $\vec{x}$

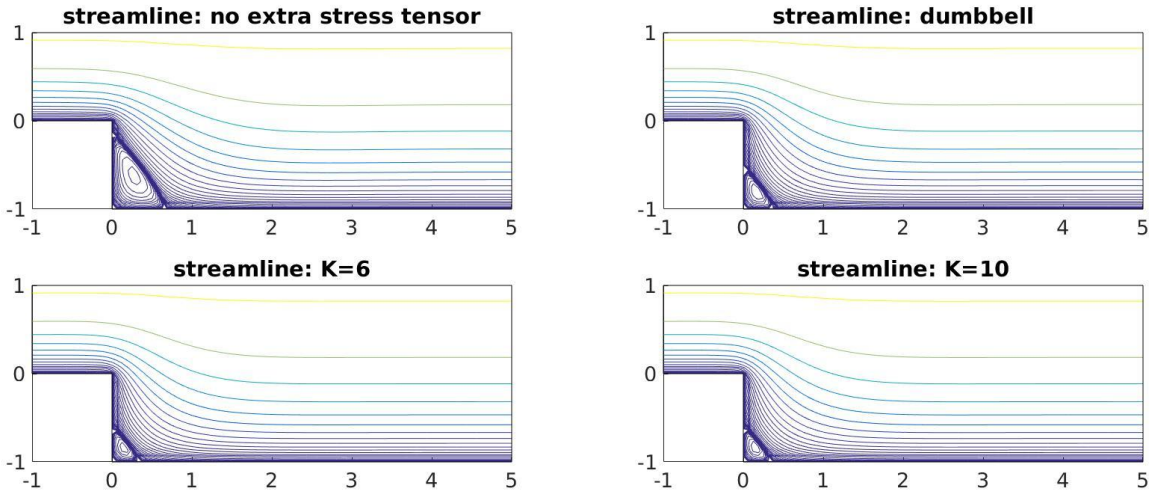


Figure 4.24: The contour plot of the velocity field at time $t = 5$. Top left: no stress tensor; Top right: dumbbell model; Bottom left: $K = 6$ springs; Bottom right: $K = 10$ springs

Table 4.4 shows the $\max_{\vec{x}\in\Omega}$ of the three components of $\boldsymbol{\tau}(\vec{x}, t)$ at time $t = 5$ for different numbers of beads and we can see from the table that the value of $\boldsymbol{\tau}$ increases only slightly as the number of beads increases.

Recall that the deterministic method for solving the multi-bead model suffers from the curse of dimensionality because the Fokker-Planck equation featuring in this model is a high-dimensional transport-diffusion equation, whose numerical solution by conventional means is extremely challenging. The cost of solving the partial differential equation numerically grows exponentially as the number of bonds increases. The advantage of using the stochastic approach is that it avoids this. Suppose we use the Monte Carlo method with an adaptive time stepping algorithm to estimate the expectation of the solution to the SDE; then the total cost is approximately the product of the number of simulations N and the average number of time steps used. In Section 1.6, we saw that adding one more bead into the chain is equivalent to adding one more SDE into the system. We only need to simulate one extra SDE in this case. The cost of simulating one SDE also equals to the number of simulations N times the average number of time steps used for that SDE. Similarly, the cost of

the multilevel Monte Carlo method is approximately

$$\text{Cost} = \sum_{l=0}^{L} N_l \overline{\Delta t_l},$$

where $\overline{\Delta t_l}$ represents the average number of time steps used in level l.

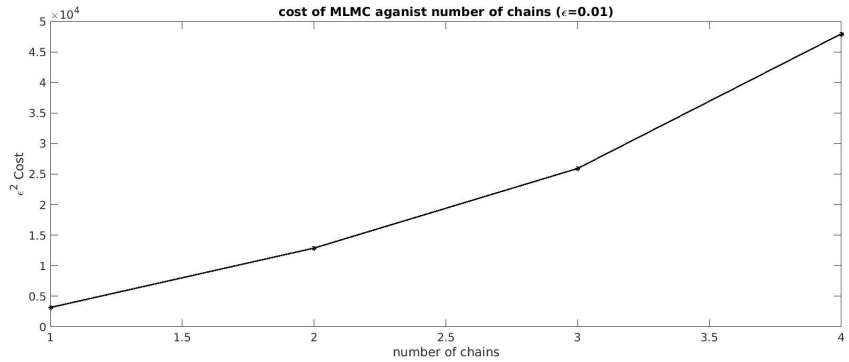


Figure 4.25: Cost of MLMC against the number of chains

Figure 4.25 shows ϵ^2Cost against the number of chains K. This numerical experiment is based on $\epsilon = 0.01, \text{Wi} = 1, \boldsymbol{\kappa} = \text{diag}(0.1, -0.1), b = 10, \sqrt{b_i} = \frac{\sqrt{b}}{K}$ using algorithm 3. We can see that the growth rate is worse than linear. A possible reason is that we decrease the size of b when we add more beads into the system and a small value of b sometimes requires a small time step in the adaptive scheme, see Section 4.4.1. It would be worthwhile to develop a rigorous analysis of the complexity of the multi-bead model using MLMC.

4.4 Some Discussion on the Parameters and the Velocity Gradient

In this section, we consider the influence of the velocity gradient $\boldsymbol{\kappa}$, the maximum extensibility b and the Weissenberg number Wi. We have already seen some of the effects of these parameters in the numerical experiments we had performed. For example, a large value of $\boldsymbol{\kappa}$ was considered in Section 4.2.3. The choice of these parameters is very important in the numerical simulation because they can seriously affect the complexity of the simulation.

4.4.1 Large b vs. Small b

First of all, we consider the maximum extensibility b. This determines the size of the configuration domain $D = B(0, \sqrt{b})$ and it is closely related to the time step we use in the adaptive scheme. Recall that the adaptive time step is:

$$\Delta t_n = \min\left(b\mathrm{Wi}\left(\frac{1}{10}\left(1 - \frac{\|\vec{Q}_n\|}{\sqrt{b}}\right)\right)^2, C_{\Delta t}\frac{\|\vec{Q}_n\|^2}{\|\boldsymbol{\kappa}\cdot\vec{Q}_n - \frac{1}{2\mathrm{Wi}}\vec{F}(\vec{Q}_n)\|^2}\right).$$

For a large value of b, we are allowed to take a larger time step Δt because the trajectories are less likely to exit the configuration domain, while for a small value of b, it is dangerous to take a large time step because the configuration domain is too small. Figure 4.26 plots the time steps generated by the adaptive time stepping method to simulate the SDE (4.2) using $\mathrm{Wi} = 1$, $\boldsymbol{\kappa} = \mathrm{diag}(1, -1)$ and $b = 100, 50, 10$. In general, the time step is large when b is large. However, there is a drawback for large values of b, which is that the variance of the random variable $\vec{Q}$ increases as well. This is because increasing the size of the configuration domain increases the range of $\vec{Q}$.

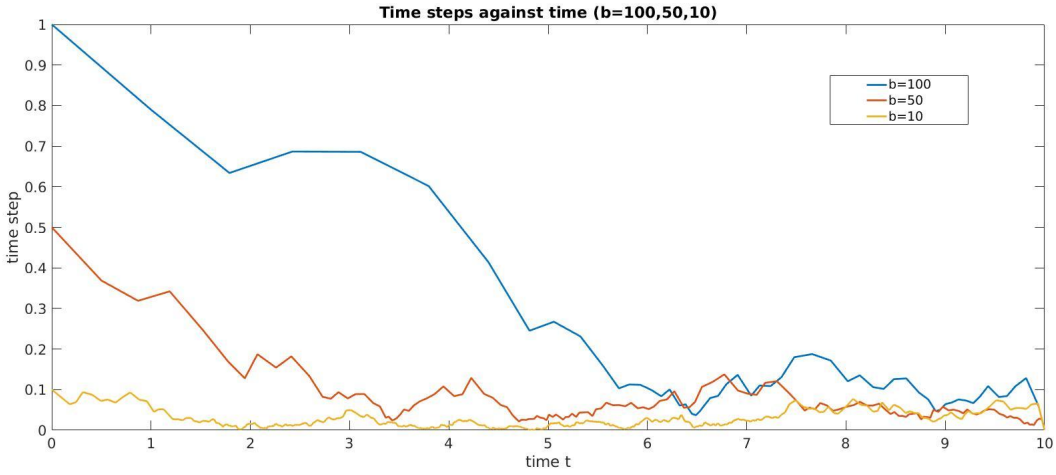


Figure 4.26: The time steps Δt for different values of $b = 10, 50, 100$

We used the same value of b for all springs when dealing with the multi-bead model. The more beads we add, the smaller b_i we have, and therefore the number of time steps grows, which increases the computational cost generally.

4.4.2 Large $\boldsymbol{\kappa}$ vs. Small $\boldsymbol{\kappa}$

We have already discussed the difficulties caused by a large velocity gradient when simulating a expansion flow in an L-shaped domain. We have the following disadvantages of having a large $\boldsymbol{\kappa}$.

First, we require $\|\boldsymbol{\kappa}\|_F - \frac{1}{2\mathrm{Wi}} < 0$ in order to show that the drift term in the stochastic differential equation satisfies the contractive Lipschitz property. If the size of the velocity gradient is large, we will not have the contractivity unless the Weissenberg number is very small. We can not use Algorithm 3 without the contractivity. Moreover the strong convergence order of the adaptive time stepping method is less than one in such a case, which suggests that we will have a worse variance decay rate. In Section 4.1.2, we have done two numerical experiments with $\boldsymbol{\kappa} = \mathrm{diag}(1, -1)$ and $\boldsymbol{\kappa} = \mathrm{diag}(0.1, -0.1)$. The Weissenberg number we used in these experiments is 1. Figure 4.6 suggests that the variance decay rate is near 2 while in Figure 4.5 the variance decay rate is worse.

In Section 4.2.3, we simulated the SDE with $\boldsymbol{\kappa} = \mathrm{diag}(1, -1)$ and $\boldsymbol{\kappa} = \mathrm{diag}(10, 10)$ and in Figure 4.19 we can see that the trajectories exhibit oscillations near the boundary at the left and right edge of the domain with a large velocity gradient. Hence the variance of the Monte Carlo estimator should be large. This is equivalent to saying that the multilevel Monte Carlo estimator at level 0 has large variance. Figure 4.27 represents the variance of the MLMC estimator for different velocity gradients based on 5000 samples using Algorithm 2. The Weissenberg number is fixed to be 1 in this numerical experiment. We can see that the variance still decays, however the decay rate is 2 for small $\boldsymbol{\kappa}$ due to the contractivity. On the other hand, the decay rate is worse when we have large $\boldsymbol{\kappa}$, which is the blue line. Also we can see from the picture that the variance is large with a large velocity gradient because we have the blue line ($\boldsymbol{\kappa} = \mathrm{diag}(5, -5)$) on the top and the purple line ($\boldsymbol{\kappa} = \mathrm{diag}(0.1, -0.1)$) at the bottom.

In Section 3.2.2, we derived an adaptive time stepping method. The equation (3.25) is the time step required to control the trajectories of the SDE. We can see from the equation that the time step will be smaller if the size of the velocity gradient is larger.

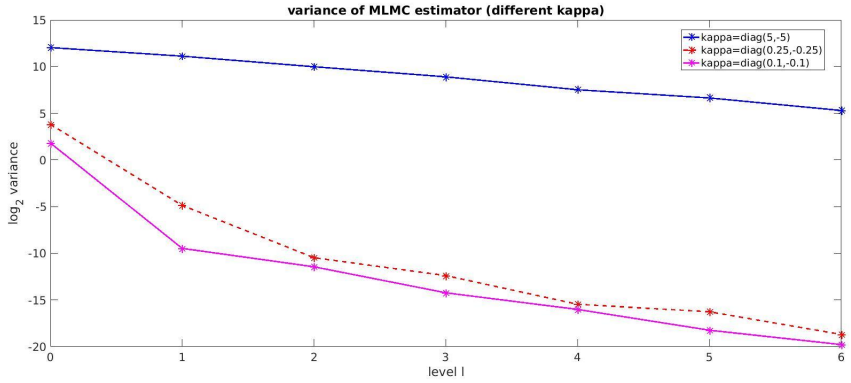


Figure 4.27: The variance of the MLMC estimator for different values of $\boldsymbol{\kappa}$

4.4.3 Large Wi vs. Small Wi

The Weissenberg number represents the ratio of the microscopic to macroscopic lengthscales, and therefore it determines the time-scale of the problem. For a small Weissenberg number, the time-scale for the SDE is smaller and relaxation is fast; therefore Algorithm 3 is a good choice because we do not need to simulate the SDE for a long time interval. From Section 3.2.5, we know that a small value of Wi implies a large value of λ as long as Assumption 3.2 holds and hence the time interval $T_l = \frac{\log R}{2\lambda} l$ is small, which means that we do not need to simulate the SDE over a large time interval. In addition, a small value of Wi implies that the FENE force dominates the drift term so we do not need to worry about the trajectory moving outside the configuration domain. On the other hand, large values of Wi imply that the relaxation of the polymer molecule is slower. In our case, a large value of Wi implies that the FENE spring force is weak compared with the velocity gradient, which means that the polymers are mainly stretched by the flow. We have already commented on the adverse influence of a large $\boldsymbol{\kappa}$. For example, the time steps need to be small and large variances are observed in the stochastic simulations. Also when Assumption 3.2 is not met, we cannot guarantee that the contractivity for the drift term holds.

Chapter 5

Conclusion

In this thesis, we considered and implemented a stochastic approach for simulating dilute polymeric fluids based on a multiscale model. The coupled system includes a stochastic differential equation for the configuration of the polymer molecules and the Navier–Stokes equations for the motion of the fluid. The link between them is the extra stress tensor $\boldsymbol{\tau}$, which is defined through the expectation of the solution to the SDE. The spring model we used is the FENE model. The problem is challenging due to the curse of dimensionality and the super-linear growth of the FENE force. Therefore we implemented a multilevel Monte Carlo strategy and an adaptive time stepping scheme to deal with the problem. With the help of the algorithms developed and a parallel computing environment, we were able to perform the multi-bead simulations with 10 springs.

In Chapter 2, we considered the steady-state solution of the coupled system. We showed that the solution to the Fokker–Planck equation converges to the Maxwellian and the velocity decays to zero given that a homogeneous Dirichlet boundary condition is imposed and no external force exists. We also showed that the decay rate is independent of the choice of the initial data. We verified these results in Chapter 4.

In Chapter 3, we introduced the concept of multilevel Monte Carlo method and developed Algorithm 2, which is an adaptive time stepping scheme combined with a multilevel Monte Carlo method. Then we extended the FENE spring force to the whole space in order to carry out a strong convergence analysis. We showed that, under suitable conditions, both the exact and the numerical solution to the extended

FENE SDE have uniform finite pth moment; hence the numerical simulation to the extended FENE SDE equipped with the adaptive time stepping scheme is strongly convergent. With a further condition on the drift term, we also showed that the strong convergence order is 1 thanks to the contractivity. For our problem, we showed that Assumption 3.2 ($\|\boldsymbol{\kappa}\|_F - \frac{1}{2\mathrm{Wi}} \leq -\lambda$) must be satisfied. With the help of contractivity, we then introduced Algorithm 3, which is an adaptive multilevel Monte Carlo method. The idea of Algorithm 3 is that we start the simulation at time $-T_l$ at level l and $-T_{l-1}$ at level $l-1$ and then stop both of them at time zero. Due to the contractivity, we know that any two paths starting from different points but driven by the same Brownian motion will converge to each other and therefore the difference between the fine and coarse path at time zero will be small. This method is more suitable for a small Weissenberg number, because the relaxation time for the SDE is small. We can also argue that small Weissenberg number implies that λ is large and we know by Lemma 3.4 that the decay rate of perturbations is $\mathrm{e}^{-\lambda p t}$. With the help of Algorithm 3, it is easy to find the steady state solution efficiently. After that, we introduced our algorithm for the coupled system, which includes the mixed finite element method for solving the Navier–Stokes equations. Finally we explained how we aim to perform multi-bead simulations. We also showed that our schemes are well suited to perform parallel computations. With the help of GPU cores, it is possible to perform the multi-bead simulations.

Chapter 4 presented the numerical results across a range of problems. We implemented the adaptive time stepping scheme for a homogeneous flow using both the Monte Carlo and the multilevel Monte Carlo method and compared their results with the true solution. We also applied the adaptive multilevel Monte Carlo method to the same problem and investigated the results obtained. We found that the cost of multilevel Monte Carlo method is $O(\epsilon^{-2})$ if assumption 3.2 is hold. We then moved to the non-homogeneous problem including both steady and unsteady flow. Then we considered the multi-bead model in these experiments and compared the results with the dumbbell model. We found that adding more beads into the model is similar to increasing the viscosity of the flow and the numerical experiments suggest that we do not need to use too many beads, because the results in the flow only changed slightly

when the number of beads is sufficiently large. Finally, we discussed the parameter values in the experiments, and noted that their choice influences the complexity of the problem. A large value of b requires more time steps and a large value of $\|\boldsymbol{\kappa}\|_F$ or large value of Wi might lead to a worse result if Assumption 3.2 does not hold.

5.1 Future Perspectives

We found in the numerical experiments that the velocity gradient and the Weissenberg number are very important in our model. A large velocity gradient and a large Weissenberg number make the problem harder from the computational point of view (the variance and the number of time steps are large). Also when Assumption 3.2 is invalid, i.e.,

$$\|\boldsymbol{\kappa}\|_F - \frac{1}{2\mathrm{Wi}} > 0, \tag{5.1}$$

we do not have contractivity, which means that we can not apply Algorithm 3 and we are not able to prove first order strong convergence. We have already seen in Chapter 4 that the variance decay rate β is worse in such a case. It would be interesting and useful to develop a numerical method for stochastic differential equations without contractivity.

Supposing that Assumption 3.2 is satisfied, a large Weissenberg number will introduce new problems. It is also frequently noted in the literature that with a high value of Wi, special care must be taken to deal with the associated stability issue of the numerical scheme [56]. For our problem, a large value of the Weissenberg number suggests that the microscopic relaxation time is larger than the characteristic fluid process time, which means that it takes a longer time for the SDE to converge to the invariant measure.

Also we did not consider the numerical analysis of the multi-bead model. In order to prove the strong convergence of the stochastic differential equation, we need to show that the exact solution has finite pth moment bound, which requires that the drift term should be one-sided Lipschitz continuous. The challenge for the multi-bead model is that the drift term involves $\vec{F}_{i+1}, -\vec{F}_i$ and $\vec{F}_{i-1}$. In order to show that the drift is one-sided Lipschitz, we need to bound the $\vec{F}_{i+1}$ and $\vec{F}_{i-1}$ in terms of $\vec{F}_i$.

It would be useful to develop a rigorous analysis of the multi-bead model including proofs of strong convergence. In addition, we did not consider the analysis of the computational cost of the multilevel Monte Carlo method for the multi-bead model.

Another possible direction is to develop a suitable numerical algorithm for small values of b. In Chapter 4, we performed numerical experiments with different values of b and we found that a smaller b requires more time steps, which then increases the computational cost.

Finally, it is possible to investigate different models such as the FENE-P model. The FENE-P model was introduced by Peterlin [48].

Recall that for the FENE model, The Kramers expression for the extra stress tensor $\boldsymbol{\tau}$ is defined by:

$$\boldsymbol{\tau}(\vec{x},t) = \frac{b+d+2}{b}\frac{1-\gamma}{\mathrm{ReWi}}\left(\int_D \vec{q}\otimes\frac{\vec{q}}{1-\frac{\|\vec{q}\|^2}{b}}\psi(\vec{x},\vec{q},t)\mathrm{d}\vec{q}-\mathbf{I}\right).$$

For the FENE-p model, the extra stress tensor $\boldsymbol{\tau}$ becomes the following:

$$\boldsymbol{\tau}(\vec{x},t) = \frac{b+d}{b}\frac{1-\gamma}{\mathrm{ReWi}}\left(\frac{\mathbf{A}}{1-\mathrm{tr}(\mathbf{A})/b}-\mathbf{I}\right),$$

where $\mathbf{A} = \int_D \vec{q}\otimes\vec{q}\psi(\vec{x},\vec{q},t)\mathrm{d}\vec{q}$ and $\mathrm{tr}(\cdot)$ denotes trace.

Consequently, the FENE-P model also leads to spring extensions that can not exceed a maximum length of $\sqrt{b}$. It would be interesting to investigate this kind of spring force and compare the results with those reported here.

Appendix

We will prove Lemma 3.2 and Lemma 3.4. Before proving the results, we need some definitions and preliminary inequalities. Our presentation of these is based on [11], [29], [34], [35].

Definition (Adaptedness). *The sequence* $\boldsymbol{X} = (X_n : n \geq 0)$ *is adapted to the sequence* $\boldsymbol{Z} = (Z_n : n \geq 0)$ *if, for each* $n \geq 0$*, there exists a (deterministic) function* $g_n(\cdot)$ *such that:*

$$X_n = g_n(Z_0, Z_1, \ldots, Z_n).$$

Similarly, the process $\boldsymbol{X} = (X_t : t \geq 0)$ *is adapted to the process* $\boldsymbol{Z} = (Z_t : t \geq 0)$ *if, for each* $t \geq 0$*, there exists a (deterministic) function* $g_t(\cdot)$ *such that:*

$$X_t = g_t(Z(s) : 0 \leq s \leq t).$$

Definition (Martingale). *A discrete sequence* $(M_n : n \geq 0)$ *is a discrete martingale (adapted to* $(Z_n : n \geq 0)$*) if:*

1. $\mathbb{E}|M_n| < \infty, \quad n \geq 0;$
2. $(M_n : n \geq 0)$ *is adapted to* $(Z_n : n \geq 0);$
3. $\mathbb{E}[M_{n+1}|Z_0, \ldots, Z_n] = M_n, \quad n \geq 0.$

Similarly, a continuous process $(M(t) : t \geq 0)$ *is a continuous martingale (adapted to* $(Z(t) : t \geq 0)$*) if:*

1. $\mathbb{E}|M(t)| < \infty, \quad t \geq 0;$
2. $(M(t) : t \geq 0)$ *is adapted to* $(Z(t) : t \geq 0);$
3. $\mathbb{E}[M(s)|Z(u) : 0 \leq u \leq t] = M(t), \quad s > t.$

Remark. *In the definition of martingale the sequence (or process)* $Z_n(Z(t))$ *can be the martingale* $M_n(M(t))$ *itself. Therefore the expected value of* M *at a future time, given complete knowledge of* M *at the current time, is equal to its current value.*

Definition (Quadratic variation). *Suppose that* $(X(t) : t \geq 0)$ *is a real-valued stochastic process defined on a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$*. Consider partitions* P *of*

the time interval $[0,t]$*; then the quadratic variation of the process* $X(t)$*, written as* $[X]_t$*, is defined as*

$$[X]_t = \lim_{||P||\to 0} \sum_{k=1}^{n} (X_{t_k} - X_{t_{k-1}})^2,$$

where P *ranges over partitions of the interval* $[0,t]$*. In the case of*

$$X(t) = X(0) + \int_0^t a(X(s),s)\,\mathrm{d}s + \int_0^t b(X(s),s)\,\mathrm{d}\,W_\mathrm{s},$$

the quadratic variation of $X(t)$ *is*

$$[X]_t = \int_0^t b^2(X(s),s)\,\mathrm{d}s.$$

Jensen's inequality. Consider a convex function φ defined on the real line, i.e., assume that the inequality

$$\varphi((1-\lambda)x_1 + \lambda x_2) \le (1-\lambda)\varphi(x_1) + \lambda\varphi(x_2)$$

holds for all $x_1, x_2 \in \mathbb{R}$ and for any $0 < \lambda < 1$.

If $\sum_{i=1}^{N} \lambda_i = 1$ with $0 < \lambda_i < 1$, then

$$\varphi\left(\sum_{i=1}^{N} \lambda_i x_i\right) \le \sum_{i=1}^{N} \lambda_i \varphi(x_i).$$

Assuming that f and $\varphi(\cdot)$ are Riemann integrable on $[a,b]$, the following integral version of Jensen' inequality holds:

$$\varphi\left(\frac{1}{b-a}\int_a^b f(x)\mathrm{d}x\right) \le \frac{1}{b-a}\int_a^b \varphi(f(x))\mathrm{d}x.$$

Gronwall's inequality. If $\beta(s)$ is a continuous and nonnegative function and u satisfies the integral inequality

$$u(t) \le \alpha(t) + \int_0^t \beta(s)u(s)\mathrm{d}s$$

where $\alpha(t)$ is integrable, then

$$u(t) \le \alpha(t) + \int_0^t \alpha(s)\beta(s)\exp\left(\int_s^t \beta(r)\mathrm{d}r\right)\mathrm{d}s.$$

Burkholder–Davis–Gundy inequality. For any $p \geq 1$, there are constants c_p, C_p such that, for a continuous time martingale M_t, the following inequality holds:

$$c_p \mathbb{E}[[M]_T^{p/2}] \leq \mathbb{E}[\sup_{[0,T]} |M_t|^p] \leq C_p \mathbb{E}[[M]_T^{p/2}],$$

where $[\cdot]_t$ denotes the quadratic variation.

Young's inequality. Given two positive real numbers p, q such that $\frac{1}{p} + \frac{1}{q} = 1$, then for any $a, b, \psi > 0$, we have

$$ab \leq \frac{a^p \psi^p}{p} + \frac{b^q}{q \psi^q}.$$

Proof of Lemma 3.2:

This proof comes from [18]. We define $\phi(\vec{x}) = \vec{x} + \Delta t(\vec{x})\vec{a}(\vec{x})$ and $\hat{Q}_t$ to be the numerical solution at time t; then the adaptive time stepping method gives:

$$\begin{aligned}\|\vec{Q}_{n+1}\|^2 \leq \|\vec{Q}_n\|^2 + 2\Delta t(\vec{Q}_n) \left(\langle \vec{Q}_n, \vec{a}(\vec{Q}_n) \rangle + \frac{1}{2} \Delta t(\vec{Q})_n \|\vec{a}(\vec{Q}_n)\|^2 \right) \\ + 2\langle \phi(\vec{Q}_n), \sqrt{\frac{1}{\mathrm{Wi}}} \Delta \vec{W}_n \rangle + \| \sqrt{\frac{1}{\mathrm{Wi}}} \Delta \vec{W}_n \|^2.\end{aligned}$$

Applying equation (3.41) in Assumption 3.3 for the time step we obtain:

$$\|\vec{Q}_{n+1}\|^2 \leq \|\vec{Q}_n\|^2 - \left(2\alpha \|\vec{Q}_n\|^2 - 2\beta \right) \Delta t(\vec{Q}_n) \tag{1}$$

$$+ 2\langle \phi(\vec{Q}_n), \sqrt{\frac{1}{\mathrm{Wi}}} \Delta \vec{W}_n \rangle + \| \sqrt{\frac{1}{\mathrm{Wi}}} \Delta \vec{W}_n \|^2. \tag{2}$$

We multiply $\exp(2\alpha t_{n+1})$ on both sides of equation (1) and (2), and we notice that $1 - 2\alpha \Delta t(\vec{Q}_n) \leq \exp(-2\alpha \Delta t(\vec{Q}_n))$; also we can bound the time step Δt by $\Delta t_{\max}$.

$$\begin{aligned}\exp(2\alpha t_{n+1}) \|\vec{Q}_{n+1}\|^2 \leq \exp(2\alpha t_n) \|\vec{Q}_n\|^2 + 2\exp(2\alpha(t_n + \Delta t_{\max}))\beta \Delta t(\vec{Q}_n) \\ + \exp(2\alpha(t_n + \Delta t_{\max}))\beta \|\Delta \vec{W}_n\|^2 \\ + 2\exp(2\alpha t_{n+1}) \langle \phi(\vec{Q}_n), \sqrt{\frac{1}{\mathrm{Wi}}} \Delta \vec{W}_n \rangle. \end{aligned} \tag{3}$$

Defining $\underline{t} = \max\{t_n : t_n \leq t\}$ to be the largest time point before time t, we have

$$\exp(2\alpha t) \|\hat{Q}_t\|^2 \leq \exp(2\alpha \underline{t}) \|\hat{Q}_{\underline{t}}\|^2 + 2\exp(2\alpha(\underline{t} + \Delta t_{\max}))\beta(t - \underline{t})$$

$$+\exp\left(2\alpha(\underline{t}+\Delta t_{\max})\right)\beta\|W_t-W_{\underline{t}}\|^2$$
$$+2\exp\left(2\alpha t\right)\langle\phi(\hat{Q}_{\underline{t}}),\sqrt{\frac{1}{\mathrm{Wi}}}(|W_t-W_{\underline{t}})\rangle. \tag{4}$$

We sum through all the time steps in equation (3) and then add equation (4); then we obtain

$$\begin{aligned}\exp\left(2\alpha t\right)\|\hat{Q}_t\|^2 \le \|\vec{Q}(0)\|^2 + 2\beta\exp\left(2\alpha\Delta t_{\max}\right)\left(\sum_{k=0}^{k=n-1}\exp(2\alpha t_k)\Delta t(\vec{Q}_{t_k})+\exp(2\alpha\underline{t})(t-\underline{t})\right)\\ +2\sum_{k=0}^{k=n-1}\exp(2\alpha t_{k+1})\langle\phi(\hat{Q}_{t_k}),\sqrt{\frac{1}{\mathrm{Wi}}}\Delta\vec{W}_k\rangle\\ +\beta\exp(2\alpha\Delta t_{\max})\sum_{k=0}^{k=n-1}\exp(2\alpha t_k)\|\Delta\vec{W}_k\|^2\\ +2\exp(2\alpha t)\langle\hat{Q}_{\underline{t}}+\vec{a}(\hat{Q}_{\underline{t}})(t-\underline{t}),\sqrt{\frac{1}{\mathrm{Wi}}}(W_t-W_{\underline{t}})\rangle\\ +\beta\exp(2\alpha(\underline{t}+\Delta t_{\max}))\|W_t-W_{\underline{t}}\|^2. \end{aligned}\tag{5}$$

We take the $p/2$-th power of both sides of inequality (5) and apply Jensen's inequality; we obtain

$$\begin{aligned}\mathrm{e}^{\alpha pt}\|\hat{Q}_t\|^p \le 6^{p/2-1}\mathrm{e}^{\alpha p\Delta t\max}\Bigg(\|\vec{Q}(0)\|^p+(2\beta\int_0^t\mathrm{e}^{2\alpha s}\mathrm{d}s)^{p/2}\\ +|2\int_0^{\underline{t}}\mathrm{e}^{2\alpha(\underline{s}+\Delta t(\vec{Q}_s))}\langle\phi(\vec{Q}_s),\sqrt{\frac{1}{\mathrm{Wi}}}\mathrm{d}W_s\rangle|^{p/2}+(\beta\sum_{k=0}^{k=n-1}\mathrm{e}^{2\alpha t_k}\|\Delta W_k\|^2)^{p/2}\\ +|2\mathrm{e}^{2\alpha\underline{t}}\langle\vec{Q}_{\underline{t}}+\vec{a}(\vec{Q}_{\underline{t}})(t-\underline{t}),\sqrt{\frac{1}{\mathrm{Wi}}}(W_t-W_{\underline{t}})\rangle|^{p/2}\\ +\beta^{p/2}\mathrm{e}^{\alpha p\underline{t}}\|W_t-W_{\underline{t}}\|^p\Bigg).\end{aligned}\tag{6}$$

So far, we have found an upper bound for $\mathrm{e}^{\alpha pt}\|\hat{Q}_t\|^p$. Next we take the supremum and the expectation on both sides of (6) to obtain:

$$\mathbb{E}\left[\sup_{0\le s\le t}\mathrm{e}^{\alpha ps}\|\hat{Q}_s\|^p\right]\le 6^{p/2-1}\mathrm{e}^{\alpha p\Delta t\max}(I_1+I_2+I_3+I_4+I_5).$$

For I_1 we have $I_1=\|\vec{Q}(0)\|^p+(2\beta)^{\frac{p}{2}}(\frac{\mathrm{e}^{2\alpha t-1)}}{2\alpha})^{\frac{p}{2}}\le\|\vec{Q}(0)\|^p+(\frac{\beta}{\alpha})^{\frac{p}{2}}\mathrm{e}^{\alpha pt}$.

For I_2, we use condition (3.41) from Assumption 3.3 and apply Jensen inequality, Young's inequality and the Burkholder–Davis–Gundy inequality to obtain

$$\begin{aligned}
I_2 &= \mathbb{E}\left[\sup_{0\le s\le t}|2\int_0^s \mathrm{e}^{2\alpha(\underline{u}+\Delta t(\vec{Q}_u))}\langle\phi(\vec{Q}_u),\sqrt{\frac{1}{\mathrm{Wi}}}\mathrm{d}W_u\rangle|^{p/2}\right]\\
&\le \mathbb{E}\left[C_p^1\left(\int_0^t \mathrm{e}^{4\alpha u}\|\phi(\vec{Q}_u)\sqrt{\frac{1}{\mathrm{Wi}}}\|^2\mathrm{d}u\right)^{p/4}\right]\\
&\le \mathbb{E}\left[C_p^1(\mathrm{e}^{2\alpha t}/2\alpha)^{p/4-1}\int_0^t \mathrm{e}^{\alpha(p/2+2)u}\|\phi(\vec{Q}_u)\sqrt{\frac{1}{\mathrm{Wi}}}\|^{p/2}\mathrm{d}u\right]\\
&\le 6^{-p/2+1}\mathrm{e}^{-\alpha p\Delta t}\max\mathbb{E}\left[\sup_{0\le s\le t}\mathrm{e}^{\alpha ps}\|\hat{(}Q)_s\|^p\right]+C_p^2\mathrm{e}^{\alpha pt},
\end{aligned}$$

where C_p^1 and C_p^2 are constants. We also apply Jensen's inequality to I_3 to obtain the following

$$\begin{aligned}
I_3 &\le \beta^{p/2}\left(\int_0^t \mathrm{e}^{2\alpha s}\mathrm{d}s\right)^{p/2-1}\mathbb{E}\left[\sum_{k=0}^{k=n-1}\mathrm{e}^{2\alpha t_k}\|\Delta W_k\|^p\right]\\
&\le C_p^3\left(\beta\int_0^t \mathrm{e}^{2\alpha s}\mathrm{d}s\right)\le C_p^3(\frac{\beta}{2\alpha})^{p/2}\mathrm{e}^{\alpha pt},
\end{aligned}$$

where C_p^3 is some constant. We derive the upper bound for I_4 based on a similar argument as for I_2.

$$\begin{aligned}
I_4 &\le 2^{p/2}\mathbb{E}\left[\sup_{0\le s\le t}\mathrm{e}^{\alpha p\underline{s}}|\langle\vec{Q}_{\underline{st}}+\vec{a}(\vec{Q}_{\underline{s}})(s-\underline{s}),\sqrt{\frac{1}{\mathrm{Wi}}}(W_s-W_{\underline{s}})\rangle|^{p/2}\right]\\
&\le 2^{p/2}\mathbb{E}\left[\sup_{0\le s\le t}\mathrm{e}^{\alpha p\underline{s}}\|\vec{Q}_{\underline{s}}+\vec{a}(\vec{Q}_{\underline{s}})(s-\underline{s})\|^{p/2}\|\sqrt{\frac{1}{\mathrm{Wi}}}\|^{p/2}\|(W_s-W_{\underline{s}})\|^{p/2}\right]\\
&\le 2^{\frac{3p}{4}-1}\beta^{\frac{p}{2}}C_p^3\Delta t_{\max}^{\frac{p}{2}-1}\mathbb{E}\left[\int_0^t \mathrm{e}^{\alpha ps}(\|\vec{Q}_s\|^{p/2}+(2\beta)^{p/4})\mathrm{d}s\right]\\
&\le \frac{1}{4}6^{-p/2+1}\mathrm{e}^{-\alpha p\Delta t}\max\mathbb{E}\left[\sup_{0\le s\le t}\mathrm{e}^{\alpha ps}\|\hat{Q}_s\|^p\right]+C_p^4\mathrm{e}^{\alpha pt},
\end{aligned}$$

where C_p^4 is some constant. For I_5, we have the same bound as for I_3

$$I_5 \le C_p^3(\frac{\beta}{2\alpha})^{p/2}\mathrm{e}^{\alpha pt}$$

.

Combining the results, we obtain

$$\mathbb{E}\left[\sup_{0\leq s\leq t}\mathrm{e}^{\alpha ps}\|\hat{Q}_s\|^p\right]\leq C_p^5\|\vec{Q}(0)\|^p+\frac{1}{2}\mathbb{E}\left[\sup_{0\leq s\leq t}\mathrm{e}^{\alpha ps}\|\hat{Q}_s\|^p\right]+C_p^6\mathrm{e}^{\alpha pt},$$

where C_p^5 and C_p^6 are constants, and therefore we have

$$\mathbb{E}\left[\sup_{0\leq s\leq t}\mathrm{e}^{\alpha ps}\|\hat{Q}_s\|^p\right]\leq \mathrm{e}^{\alpha p\Delta t}\mathrm{max}(2C_p^5\|\vec{Q}(0)\|^p+2C_p^6\mathrm{e}^{\alpha pt}).$$

For any $t\geq 0$, we have

$$\mathbb{E}\left[\mathrm{e}^{\alpha pt}\|\hat{Q}_t\|^p\right]\leq\mathbb{E}\left[\sup_{0\leq s\leq t}\mathrm{e}^{\alpha ps}\|\hat{Q}_s\|^p\right]\leq 2C_p^7\|\vec{Q}(0)\|^p+2C_p^8\mathrm{e}^{\alpha pt}.$$

Hence we deduce that for any $t\geq 0$, we have

$$\mathbb{E}[\|\hat{Q}_t\|^p]\leq 2C_p^7\|\vec{Q}(0)\|^p\mathrm{e}^{-\alpha pt}+2C_p^8\leq 2C_p^7\|\vec{Q}(0)\|^p+2C_p^8\leq C_p.\quad\square$$

Proof of Lemma 3.4:

This proof comes from [18]. We define $\vec{e}(t)=\vec{X}(t)-\vec{Y}(t)$, where $\vec{X}(t)$ and $\vec{Y}(t)$ are any two solutions to the SDE (3.47) driven by the same Brownian motion; and therefore we have

$$\mathrm{d}\vec{e}(t)=(\vec{a}(\vec{X}(t))-\vec{a}(\vec{Y}(t)))\mathrm{d}t+(\mathbf{b}(\vec{X}(t))-\mathbf{b}(\vec{Y}(t)))\mathrm{d}\vec{W}_t.$$

By Ito's formula, we have, for any $0<t\leq T$,

$$\begin{aligned}\mathrm{e}^{\lambda pt}\|\vec{e}(t)\|^p-\|\vec{e}(0)\|^p\leq&\int_0^t\lambda p\mathrm{e}^{\lambda ps}\|\vec{e}(s)\|^p\mathrm{d}s\\&+\int_0^t p\mathrm{e}^{\lambda ps}\langle\vec{e}(s),\vec{a}(\vec{X}(s))-\vec{a}(\vec{Y}(s))\rangle\|\vec{e}(s)\|^{p-2}\mathrm{d}s\\&+\int_0^t\frac{p(p-1)}{2}\mathrm{e}^{\lambda ps}\|\mathbf{b}(\vec{X}(s))-\mathbf{b}(\vec{Y}(s))\|^2\|\vec{e}(s)\|^{p-2}\mathrm{d}s\\&+\int_0^t p\mathrm{e}^{\lambda ps}\langle\vec{e}(s),(\mathbf{b}(\vec{X}(s))-\mathbf{b}(\vec{Y}(s)))\mathrm{d}\vec{W}(s)\rangle\|\vec{e}(s)\|^{p-2}.\end{aligned}$$

We take the expectation on both sides and apply the condition for contractivity to obtain

$$\mathbb{E}[\mathrm{e}^{\lambda pt}\|\vec{e}(t)\|^p]-\mathbb{E}[\|\vec{e}(0)\|^p]\leq 0,$$

and therefore we can deduce that

$$\mathbb{E}[\|\vec{X}(t) - \vec{Y}(t)\|^p] \leq \mathrm{e}^{-\lambda p t}\mathbb{E}[\|\vec{X}(0) - \vec{Y}(0)\|^p]. \qquad \square$$

References

[1] D. J. Acheson. *Elementary Fluid Dynamics*. Clarendon,Oxford, 1990.

[2] A. Arnold, P. Markowich, G. Toscani, and A. Unterreiter. On generalized Csiszár-Kullback inequalities. *Monatshefte für Mathematik*, (131):235–253, 2000.

[3] L. Arnold. *Stochastic dierential equations: theory and applications*. Wiley-Interscience, 1974.

[4] J. W. Barrett, C. Schwab, and E. Süli. Existence of global weak solutions for some polymeric flow models. *Math. Models and Methods in Applied Sciences*, 15(3):939–983, 2005.

[5] J. W. Barrett and E. Süli. Existence and equilibration of global weak solutions to kinetic models for dilute polymers i: Finitely extensible nonlinear bead-spring chains. *Mathematical Models and Methods in Applied Sciences*, 21(06):1211–1289, 2011.

[6] J. W. Barrett and E. Süli. Existence of global weak solutions to finitely extensible nonlinear bead-spring chain models for dilute polymers with variable density and viscosity. *Journal of Differential Equations*, 253(12):3610 – 3677, 2012.

[7] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

[8] A. V. Bhave, R. C. Armstrong, and R. A. Brown. Kinetic theory and rheology of dilute, nonhomogeneous polymer solutions. *The Journal of Chemical Physics*, 95(4), 1991.

[9] R. B. Bird, R. C. Armstrong, and O. Hassager. Dynamics of Polymeric Liquids. Volume 2: Kinetic theory. *AIChE Journal*, 35(8):1399–1402, 1989.

[10] C. Bris and T. Lelièvre. Multiscale modelling of complex fluids: A mathematical initiation. In *Multiscale Modeling and Simulation in Science*, volume 66 of *Lecture Notes in Computational Science and Engineering*, pages 49–137. Springer Berlin Heidelberg, 2009.

[11] D. L. Burkholder, B. J. Davis, and R. F. Gundy. Integral inequalities for convex functions of operators on Martingales. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, pages 223–240, Berkeley, Calif., 1972. University of California Press.

[12] C. Chauvière and A. Lozinski. Simulation of complex viscoelastic flows using the Fokker-Planck equation: 3D FENE model. *Journal of Non-Newtonian Fluid Mechanics*, 122(13):201 – 214, 2004.

[13] C. Chauvière and A. Lozinski. Simulation of dilute polymer solutions using a Fokker-Planck equation. *Computers and Fluids*, 33:689–696, 2004.

[14] J. Cheng, M. Grossman, and T. McKercher. *Professional CUDA C Programming*. Wrox, 2014.

[15] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Oxford University Press, Oxford, 2005.

[16] X. J. Fan. Molecular models and flow calculations: II. Simulation of steady planar flow. *Acta Mechanica Sinica*, 5:216–226, 1989.

[17] W. Fang and M. B. Giles. Adaptive Euler-Maruyama method for SDEs with non-globally Lipschitz drift: Part i, finite time interval. *arXiv preprint arXiv:1609.08101*, 2016.

[18] W. Fang and M. B. Giles. Adaptive Euler-Maruyama method for SDEs with non-globally Lipschitz drift: Part ii, infinite time interval. *arXiv preprint arXiv:1703.06743*, 2017.

[19] L. Figueroa. *Deterministic Simulation of Multi-Beaded Models of Dilute Polymer Solutions*. PhD thesis, University of Oxford, 2011.

[20] T. C. Gard. *Introduction to stochastic differential equations*. Marcel Dekker Inc, New York, 1988.

[21] G. Geoffrey. *Probability - An Introduction*. Oxford University Press, 1986.

[22] M. B. Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. In Alexander Keller, Stefan Heinrich, and Harald Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods, 2006*, pages 343–358. Springer Berlin Heidelberg, 2008.

[23] M. B. Giles. Multilevel Monte Carlo path simulation. *Oper. Res.*, 56(3):607–617, 2008.

[24] M. B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.

[25] M. B. Giles, C. Lester, and J. Whittle. Non-nested adaptive timesteps in multilevel Monte Carlo computations. *Monte Carlo and Quasi-Monte Carlo Methods 2014*, Springer, 2015.

[26] M. B. Giles and L. Szpruch. Multilevel Monte Carlo methods for applications in finance. *Recent Developments in Computational Finance, World Scientific*, 2013.

[27] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*. Springer, 1986.

[28] M. Griebel and A. Rüttgers. Multiscale simulations of three-dimensional viscoelastic flows in a squaresquare contraction. *Journal of Non-Newtonian Fluid Mechanics*, 205(0):41 – 63, 2014.

[29] T. H. Gronwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, 20(4):pp. 292–296, 1919.

[30] D. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001.

[31] D. J. Higham, X. Mao, and A. M. Stuart. Strong convergence of Euler-type methods for nonlinear stochastic differential equations. *SIAM Journal on Numerical Analysis*, 40(3):1041–1063, 2003.

[32] M. A. Hulsen, A. P. G. van Heel, and B. H. A. A. van den Brule. Simulation of viscoelastic flows using Brownian configuration fields. *Journal of Non-Newtonian Fluid Mechanics*, 70(12):79 – 101, 1997.

[33] M. Hutzenthale, A. Jentzen, and P. E. Kloeden. Strong and weak divergence in finite time of Euler's method for stochastic differential equations with non-globally Lipschitz continuous coefficients. *Proceedings of the Royal Society*, 2010.

[34] M. Hutzenthaler and A. Jentzen. Convergence of the stochastic Euler scheme for locally Lipschitz coefficients. *Foundations of Computational Mathematics*, 11(6):657–706, 2011.

[35] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, Berlin, 1992.

[36] D. J. Knezevic. *Analysis and Implementation of Numerical Methods for Simulating Dilute Polymeric Fluids*. PhD thesis, University of Oxford, 2008.

[37] D. J. Knezevic and E. Süli. A heterogeneous alternating-direction method for a micro-macro dilute polymeric fluid model. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(6):1117–1156, 8 2009.

[38] A. P. Koppol, R. Sureshkumar, and B. Khomami. An efficient algorithm for multiscale flow simulation of dilute polymeric solutions using bead-spring chains. *Journal of Non-Newtonian Fluid Mechanics*, 141(23):180 – 192, 2007.

[39] M. Laso and H.C. Öttinger. Calculation of viscoelastic flow using molecular models: the CONNFFESSIT approach. *Journal of Non-Newtonian Fluid Mechanics*, 47(0):1 – 20, 1993.

[40] T. Li and P. Zhang. Mathematical analysis of multi-scale models of complex fluids. *Commun, Math. Sci.*, 5(1):1–51, 2007.

[41] A. Lozinski and C. Chauvière. A fast solver for Fokker-Planck equation applied to viscoelastic flows calculations: 2D FENE model. *Journal of Computational Physics*, 189(2):607 – 625, 2003.

[42] J. C. Mattingly, A. M. Stuart, and D. J. Higham. Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise. *Stochastic Processes and their Applications*, 101(2):185–232, 2002.

[43] Z. Min, W. Cao, and C. Shen. Numerical study of dilute polymer solutions using FENE bead-spring chain model. *Polymer-Plastics Technology and Engineering*, 47(6):630–634, 2008.

[44] NVIDIA. CUDA C programming guide version 7.0. , 2017.

[45] J. G. Oldroyd. On the formulation of rheological equations of state. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 200(1063):523–541, 1950.

[46] H. C. Öttinger. *Stochastic Processes in Polymeric Fluids*. Springer, Berlin, 1996.

[47] R. G. Owens and T. N. Phillips. *Computational Rheology*. Imperial College Press, London, 2002.

[48] A. Peterlin. Hydrodynamics of macromolecules in a velocity field with longitudinal gradient. *Journal of Polymer Science Part B: Polymer Letters*, 4(4):287–291, 1966.

[49] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes*. Cambridge university press, 2007.

[50] B. D. Ripley. *Stochastic Simulation*. Wiley Sons, New York, 1987.

[51] J. W. Rudisill and P. T. Cummings. Brownian dynamics simulation of model polymer fluids in shear flow. I. Dumbbell models. *Journal of Non-Newtonian Fluid Mechanics*, 41(3):275 – 288, 1992.

[52] A. Rüttgers. Multiscale Modelling of Dilute Polymeric Fluids with Stochastic and Fokker-Planck-based Methods. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2010.

[53] J. D. Schieber and H. C. Öttinger. The effects of bead inertia on the Rouse model. *The Journal of Chemical Physics*, 89(11), 1988.

[54] K. D. Smith and A. Sequeira. Micromacro simulations of a shear-thinning viscoelastic kinetic model: applications to blood flow. *Applicable Analysis*, 90(1):227–252, 2011.

[55] E. Süli and G. Yahiaoui. McKean-Vlasov diffusion and the well-posedness of the Hookean bead-spring-chain model for dilute polymeric fluids: small-mass limit and equilibration in momentum space. *arXiv preprint arXiv:1802.06268*, 2018.

[56] R. I. Tanner. The stability of some numerical schemes for model viscoelastic fluids,. *J. Non-Newt. Fl. Mech*, 10:169–174, 1982.

[57] R. O. Vargas, O. Manero, and T.N. Phillips. Viscoelastic flow past confined objects using a micro-macro approach. *Rheologica Acta*, 48(4):373–395, 2009.

[58] H. R. Warner. Kinetic theory and rheology of dilute suspensions of finitely extensible dumbbells. *Ind. Eng. Chem. Fundamentals*, 11:379–387, 1972.

[59] L. Yan. The Euler scheme with irregular coefficients. *The Annals of Probability*, 30(3):1172–1194, 07 2002.

[60] C. Yuan and X. Mao. A note on the rate of convergence of the Euler-Maruyama method for stochastic differential equations. *Stochastic Analysis and Applications*, 26(2):325–333, 2008.