A Particle Method for Fokker-Planck Equations in High Dimensions

G. Venkiteswaran

Vom Fachbereich Mathematik
der Universität Kaiserslautern
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation

1. Gutachter: Prof. Michael Junk
2. Gutachter: Prof. Christian Lécot

Datum der Disputation: 31. Januar 2003

D386

To all my teachers.

I take this opportunity to thank Prof. Helmut Neunzert for suggesting this interesting problem and allowing me to work on the same. I am indebted to Prof. Michael Junk for his support and encouragement during the three years of my work without which this work would have just remained a dream.

I am grateful to Prof. Helmut Neunzert for his valuable technical suggestions on the subject and other members of AG Technomathematik with whom I had discussions time and again. I would also like to acknowledge the help of Dr. J. Ravi Prakash, Department of Chemical Engineering, Monash University, Australia for introducing me to the interesting subject of Polymers and for some fruitful discussions.

Finally, I wish to express my gratitude to the German Research Foundation (DFG) for providing me with financial support within the Graduiertenkolleg Mathematik und Praxis.

Contents

CHAPTER 1

Introduction

Fluctuations are a very common feature in a large number of fields ranging from stock markets to circuit theory. Nearly every system is subjected to a complicated external or internal influence that are not completely known and are termed as noise. The *Fokker-Planck* equation deals with those disturbances which change the variables of a system in a random but small way. This equation was applied to the Brownian motion problem, first studied by Robert Brown, who observed the zig-zag motion of a pollen grain in water due to thermal fluctuations. Due to these fluctuations the position of the particle cannot be determined precisely and only a probabilistic estimate can be given. With the help of Fokker-Planck equations such a probability can be determined.

If $u(\boldsymbol{x},t)d\boldsymbol{x}$ denotes the probability of finding the particle between positions $\boldsymbol{x}$ and $\boldsymbol{x}+d\boldsymbol{x}$ at time t, then the Fokker-Planck equation for $u(\boldsymbol{x},t)$ has the general form

$$\frac{\partial u}{\partial t}(\boldsymbol{x},t)+\nabla\cdot(\mathbf{v}u)(\boldsymbol{x},t)=\nabla\cdot(\mathbf{D}\nabla u)(\boldsymbol{x},t),\quad \boldsymbol{x}\in\mathbb{R}^s,\ \ t>0 \tag{1.0.1}$$

where $\mathbf{v}(\boldsymbol{x},t)$ is the velocity field of the solvent and $\mathbf{D}$ is the diffusion matrix. Initial and boundary conditions are to be prescribed, which are dependent on the physical problem under consideration. Refer [**23**] for a detailed treatment on such equations.

In the present work we are interested in studying the dynamics of polymeric liquids. Polymeric liquids are non-Newtonian liquids possessing special properties. This is attributed to their chemical composition, which is quite complex. A microscopic study is therefore necessary to understand the qualitative difference between Newtonian and polymeric liquids. In fact, evaluation of material functions like viscosity, normal stresses etc., are important for the classification of polymeric liquids. This subject has come to be known as polymer kinetic theory and the material functions as viscometric functions.

We now turn on our attention to describe the qualitative differences between the behavior of Newtonian and polymeric liquids mentioned above. These are not to be considered as abnormalities but rather as properties common of liquids having large molecules in them. We shall now describe two experiments illustrating properties specific to polymeric liquids. Refer [**1**] for other experiments characterizing polymeric liquids.

1.0.1. Non-Newtonian viscosity. In this experiment, we consider two identical tubes, one filled with Newtonian fluid and the other with a polymeric fluid to the same volume as shown in figure 1.1. The fluids are so chosen such that they have the same viscosity, meaning that a ball put into both take the same time to reach a specific position inside the fluids. The tubes are initially covered at the bottom by a base. In such a situation one would expect that on removing the base, the two liquids drain out at the same rate. But what is observed is the contrary, the polymeric liquid drains out faster than the Newtonian liquid. This effect is related to the fact that the viscosity of a polymeric

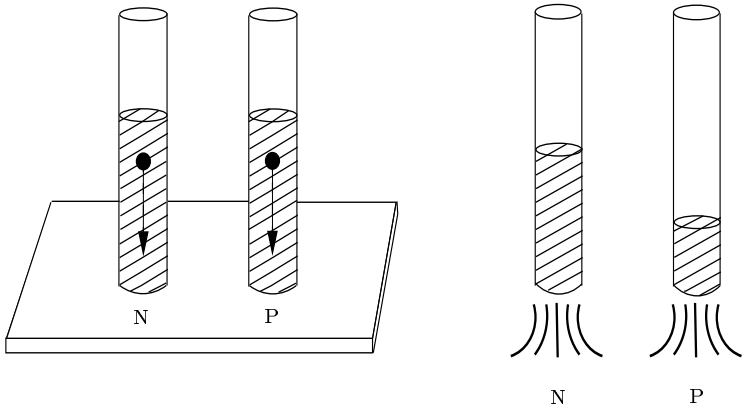


FIGURE 1.1. Tubes containing Newtonian liquid N and polymeric liquid P. The polymeric liquid flows faster than Newtonian liquid.

liquid decreases with increasing shear rate, which is called shear-thinning. This property of polymeric fluids is made use of in speeding up oil transport over large distances.

1.0.2. Normal stress effects. Here, we consider two beakers, one filled up with a Newtonian liquid and the other with a polymeric liquid. We then insert two identical rotating rods at the center to both these beakers. In the case of

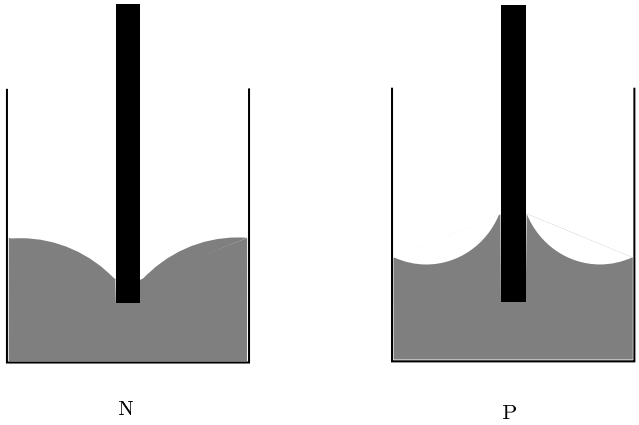


FIGURE 1.2. Beakers containing Newtonian liquid N and polymeric liquid P. On inserting a rotating rod, the Newtonian liquid develops a dip whereas the polymeric liquid climbs up.

Newtonian liquid, we see a dip in the middle near to the rotating center. What happens in the case of polymeric liquids is that the liquid climbs up the rod (figure 1.2), quite opposite to what is observed for Newtonian liquids. This phenomenon can be interpreted with the notion of an extra tension along the streamlines, which causes the effect.

It is therefore of interest to compute these material functions, namely normal stress and viscosity, which serve as a theoretical way of verifying the properties. It is to be noted that there are several other phenomena exhibited by polymeric liquids (see [**1**]), and they correspond to different physical properties, but we shall not consider them for the present study.

Polymeric liquids, have in them long macromolecules or polymers and these possess large number of degrees of freedom. Due to the irregular thermal fluctuations between the polymer molecules and the solvent particles, the former undergoes configurational changes. So we end up with a Fokker-Planck equation for the configurational distribution function, which is the probability of finding the polymer in a certain configuration at a prescribed time. Since the polymer molecule possesses large number of degrees of freedom, the resulting equation is posed on a high-dimensional space. What we are finally interested in is to evaluate the steady state values of the material functions, viscosity and first normal stress difference coefficient, which are integral functionals of the configurational distribution function. The aim of the thesis is to find out a fast and accurate method of evaluating the same in high dimension.

Traditional methods of computing these functionals, like finite differences etc, are not applicable in high dimension since the number of node points needed to achieve a prescribed accuracy grows exponentially with the dimension. This is referred to as the **curse of dimension**. The next approach would then be to use a Monte Carlo method. Though this method is applicable, the error estimate in terms of the number of particles N, goes only like $1/\sqrt{N}$.

The idea now is to replace Monte Carlo points by well determined sequences (quasi random points) which are better uniformly distributed than the former. Though this trick works for plain integration and order close to $1/N$ can be achieved, it cannot be simply applied to particle simulations. This is because of the correlation among the quasi random points. This problem was first studied by Lécot [**13**], for the spatially homogeneous Boltzmann equation and he gave a convergence proof when the quasi random points were used. The idea of Lécot was to reorder the particle positions at each time step to break the correlations. Morokoff and Caflisch [**17**], applied this technique to solve the heat equation in one and two dimensions and they obtained significant improvement over the Monte Carlo approach for certain problems. However, for the high dimensional case, the idea of reordering was not clear.

Lécot [**15**], introduced a sorting algorithm which was adaptable to higher dimensions and also shuffled the particle positions at each time step. The sorting was done with respect to each coordinate of the particle position and convergence was proved for arbitrary dimension s. For the simple diffusion problem, there was some improvement achieved over the standard Monte Carlo method. The method however had a drawback, namely, the particle numbers were drastically increasing. For a problem in s dimension, a Faure generator of base b, a prime $\geq 2s$ was taken. The minimal particle number was then b^s if sorting is to be done in each coordinate. To be concrete, for the case $s = 10$, the base b is 23 and the minimal particle number is of the order $23^{10} (\approx 10^{13})$.

Lécot and Schmid [**16**], modified the previous scheme of Lécot and replaced the $2s$ dimensional sequence by a $s+1$ dimensional sequence. Again, the minimal particle number is b^s if sorting is done with respect to all the coordinates, but now b is the smallest prime $\geq s$. This means that for the example case $s = 10$, a minimum of 11^{10} particles have to be considered and this is still too big. The method was based on partial discretization and numerical results were presented only for one and two dimensions.

In our method, we apply the scheme presented in [**15**], in a way which is not justified by the error estimate given in [**15**]. The idea is to reorder the particle position only with respect to the first coordinate which allows us to work with a $s+1$ dimensional sequence instead of a $2s$ dimensional one and get a low minimum particle number. With our method, the possible particle numbers that can be considered for the case $s = 10$ are $11, 11^2, 11^3, \ldots$. Even for $s = 100$, the possible particle numbers are $101, 101^2, 101^3, \ldots$, which seems quite reasonable. This makes the scheme more faster compared to the full reordering done in [**15**] and the memory requirement is drastically reduced. We prove an error estimate for this modification of Lécot's algorithm in [**15**] which shows essentially an $1/\sqrt{N}$ behavior but in practice the algorithm outperforms standard Monte Carlo method for the Fokker-Planck equation describing polymeric liquids.

The thesis is organized as follows. In Chapter 2, some physical models for polymers are discussed and we derive the Fokker-Planck equation from force balance equations. In Chapter 3 we look into the existence and uniqueness of solutions for the derived Fokker-Planck equation and later the splitting method is introduced, the method in which we split the convection and diffusion processes occurring in our equation, and first order convergence in time is proved. Chapter 4 is about particle methods in general and we illustrate the superiority of quasi-Monte Carlo over Monte Carlo and conclude with a convergence proof of the particle method for the diffusion equation. The last chapter shows the numerical results of our simulations.

CHAPTER 2

Polymeric Liquids: Model and Dynamics

In order to study polymeric liquids at a microscopic level, we first need a model for polymers. The mechanical model should have in it the high degree of freedom and should reflect the stretching and elongational flow properties observed in polymeric liquids. The subject polymer kinetic theory started around 1930 and various mechanical models have been suggested, (see [**2**]). They are, the chain model with fixed bond lengths and fixed angles, the bead-rod model and the bead-spring model.

It should be evident that the system we are studying is extraordinarily complicated. In modeling such a system it should be kept in mind that the model is able to account for the physical behavior of polymeric liquids and give reasonable results. The nature of the model actually depends on the final result in which one is interested. For a fluid dynamist, the flow behavior would be important whereas for a rheologist, it is interesting to have an accurate description of material functions. We shall now discuss the above three models in detail.

2.1. Model for polymer molecules

2.1.1. Chain model with fixed bond lengths and bond angles. It was observed by Flory [**6**], that the bond lengths and bond angles between adjacent bonds are restricted to very narrow ranges. At normal temperatures, the deflections are about 3% of their equilibrium values. So in this model it is reasonable to fix the bond length d and the bond angle β. In case of the polyethylene molecule, the angle between the successive C-C bonds is restricted to be $\beta = \arccos \frac{1}{3} = 70.5$ (see figure 2.1). In order to specify the configuration of the N bead chain (N large) at any time, we need approximately N internal coordinates.

2.1.2. Bead-rod model. We start with a model which is called the bead-rod chain model where we have N beads freely jointed together with $N-1$

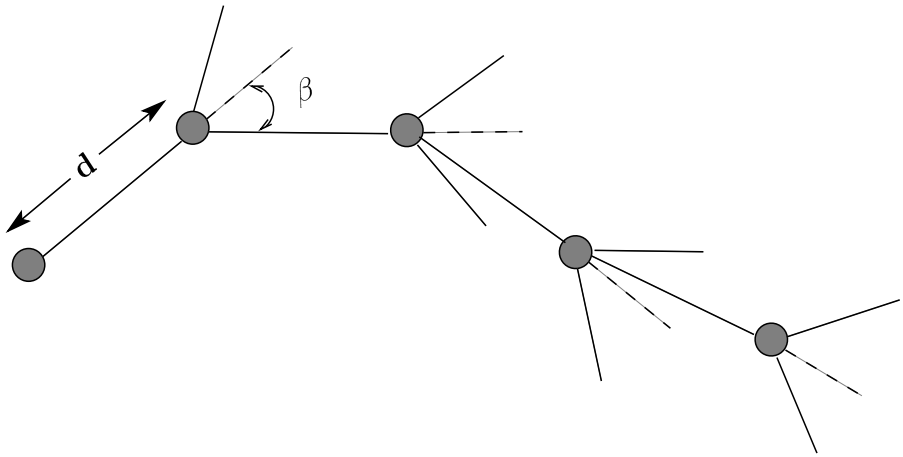


FIGURE 2.1. Fixed bond length and bond angle chain model. The spherical balls here represent the carbon atoms in polyethylene chain. Taken from [**2**].

massless rods, called connector vectors, of fixed length a as shown in figure 2.2. The beads do not represent the polymer atoms but are rather concentration of 10 or 20 monomer units. Since the model was proposed by Kramers [**9**], we call it a Kramers chain.

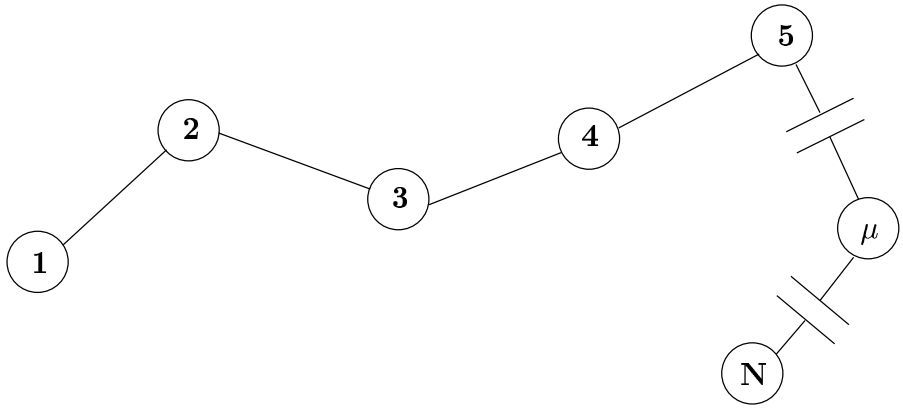


FIGURE 2.2. Kramers bead-rod model of polymer chain.

It is to be observed that the model has a few drawbacks namely, the connector vectors all have the length a so that the contour length of the chain remains a constant, being $(N-1)a$. But there are also some features which are characteristic of polymer molecules: it has a large number of internal degrees of freedom (for a large N); it can be oriented, stretched and deformed.

It can be shown that (see [**2**]), the average tension in the rod is

$$\mathbf{F}(\boldsymbol{r}) = \frac{3k_BT}{(N-1)a^2}\,\boldsymbol{r}$$

where k is the Boltzmann's constant, T is the temperature of the solvent, and $\boldsymbol{r}$ is the end-to-end vector. This suggests that the rods in the bead-rod chain behaves like a Hookean spring with the spring constant given by $H = \frac{3k_BT}{(N-1)a^2}$.

2.1.3. Bead-spring model. The bead-spring model was introduced by Rouse in 1950. The bead-spring chain or the Rouse chain is similar to the bead-rod model except that the rods are now replaced by Hookean springs of same spring constant H. Figure 2.3 depicts a typical chain with N beads.

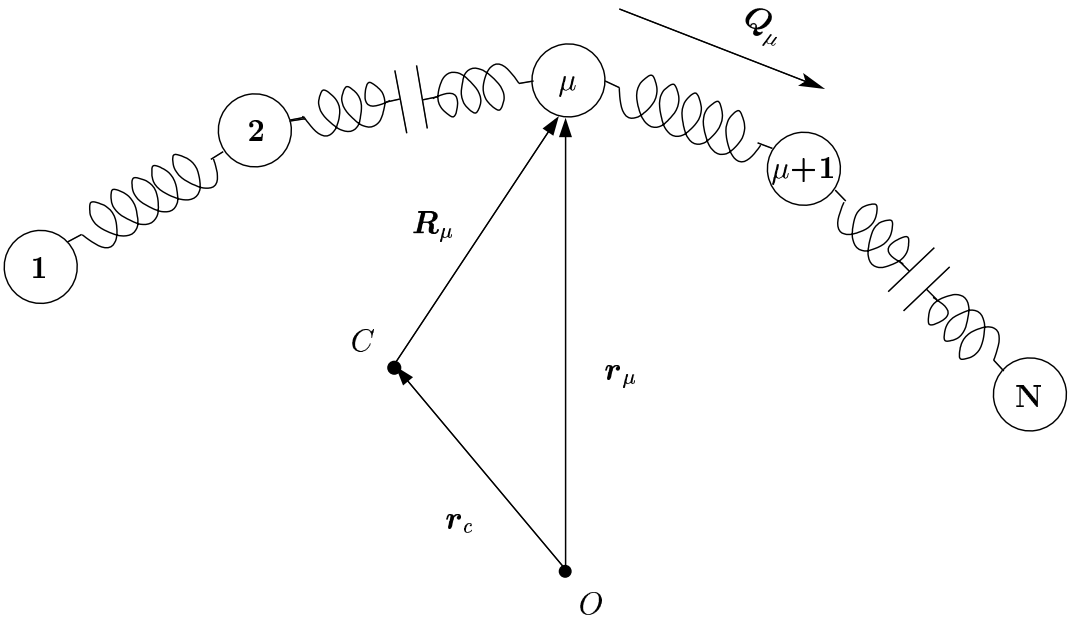


FIGURE 2.3. Rouse model of bead-spring chain.

The N beads of the chain are connected by $N-1$ connector vectors $\boldsymbol{Q}_i$, $i = 1, \ldots, N-1$. The configuration of the chain can either be described by specifying all the position vectors $\boldsymbol{r}_\nu$, $\nu = 1, \ldots, N$ with respect to a fixed point in space O or by prescribing the $N-1$ connector vectors $\boldsymbol{Q}_i = \boldsymbol{r}_{i+1} - \boldsymbol{r}_i$ and the position of the center of mass C, $\boldsymbol{r}_c = \overrightarrow{OC}$. So the configuration depends on N vectors and since each vector has 3 components, the configurational space is $\mathbb{R}^{3N}$.

Out of the three models described so far, the bead-spring chain model is advantageous because it is able to capture the basic properties of polymers without

the need of considering complicated side constraints like fixed rod lengths or bond angles. So from now on in our discussion, we shall restrict ourselves to the bead-spring model.

In doing so we shall use the following index conventions: μ, ν, η ... would be used to number the beads and they run from $1, \ldots, N$; i, j, k would be used to number the connector vectors and they take values $1, \ldots, N-1$. With this terminology

$$\begin{aligned} \boldsymbol{r}_c &= \frac{1}{N}\sum_{\nu} \boldsymbol{r}_\nu \\ \boldsymbol{Q}_k &= \boldsymbol{r}_{k+1} - \boldsymbol{r}_k \end{aligned}$$

The relation between the two systems can be expressed by

$$\begin{aligned} \boldsymbol{Q}_k &= \sum_{\nu} \bar{\mathbf{B}}_{k\nu} \boldsymbol{r}_\nu \\ \boldsymbol{r}_\nu - \boldsymbol{r}_c &= \sum_{k} \mathbf{B}_{\nu k} \boldsymbol{Q}_k \end{aligned}$$

where the $(N-1) \times (N)$ matrix $\bar{\mathbf{B}}$ is given by

$$\bar{\mathbf{B}}_{k\nu} = \delta_{k+1,\nu} - \delta_{k,\nu} \tag{2.1.2}$$

and the $(N) \times (N-1)$ matrix $\mathbf{B}$ are given by

$$\mathbf{B}_{\nu k} = \begin{cases} \frac{k}{N} & \text{if} \quad k < \nu \\ -\left[1 - \frac{k}{N}\right] & \text{if} \quad k \geq \nu \end{cases} \tag{2.1.3}$$

We define two $(N-1) \times (N-1)$ symmetric nonsingular matrices $(\boldsymbol{A}_{ij})$ and $(\boldsymbol{C}_{ij})$ as follows

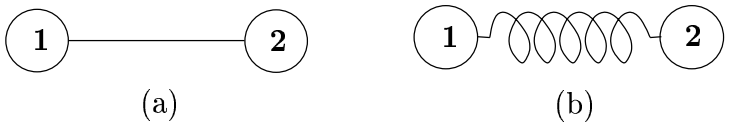
$$\sum_{\nu} \bar{\mathbf{B}}_{i\nu} \bar{\mathbf{B}}_{j\nu} = \boldsymbol{A}_{ij} = \begin{cases} 2 & \text{if} \quad |i-j| = 0 \\ -1 & \text{if} \quad |i-j| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.1.4}$$

$$\sum_{\nu} \mathbf{B}_{\nu i} \mathbf{B}_{\nu j} = \boldsymbol{C}_{ij} = \begin{cases} i(N-j)/N & \text{if} \quad i \leq j \\ j(N-i)/N & \text{if} \quad j \leq i \end{cases} \tag{2.1.5}$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ called the Rouse and Kramers matrix respectively are inverses of each other and their eigenvalues a_j and c_j are given by

$$a_j = \frac{1}{c_j} = 4\sin^2\left(\frac{j\pi}{2N}\right), \qquad j = 1, \ldots, N-1 \tag{2.1.6}$$

A simplification of the last two models is the dumbbell model. As the name suggests, the dumbbell model (figure 2.4) consists of two beads connected either by a rigid rod or a Hookean spring. Accordingly it is called as rigid dumbbell or elastic dumbbell.



(a) (b)

FIGURE 2.4. Dumbbell model. (a) is a rigid dumbbell and (b) is a Hookean or elastic dumbbell

It is clear that the dumbbell model is too crude since it in no way accounts for the details in molecular architecture. It also does not have enough number of degrees of freedom to account for fine structure in the motion of polymers. On the other hand the elastic dumbbell model is orientable and stretchable. Also the dynamics can be studied quite easily with little mathematical effort. The dynamics have been studied in detail [**22**] and they serve as a benchmark for our simulations.

2.2. Dynamics of polymeric liquids

To start with, a polymeric liquid is modeled as a Newtonian solvent having as solute the polymer molecules. The type of flow often used to characterize polymeric liquids is the *shear flow*. A shear flow is a one-parameter family of material surfaces which slide relative to one another without stretching. This means that two points on a surface which are initially separated by a horizontal distance l continue to be so for all future times.

The velocity field which describes a simple shear flow (figure 2.5) is given by

$$v_x = \beta y \quad v_y = 0 \quad v_z = 0$$

In general the velocity can be written as $v(\mathbf{x}) = \boldsymbol{\kappa}\,\mathbf{x}$ where $\boldsymbol{\kappa}$ is the gradient of the velocity field given by

$$\boldsymbol{\kappa} = \begin{pmatrix} 0 & \beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.2.7}$$

where β is the shear rate. It is to be noted that since $tr(\boldsymbol{\kappa}) = 0$, the flow is incompressible.

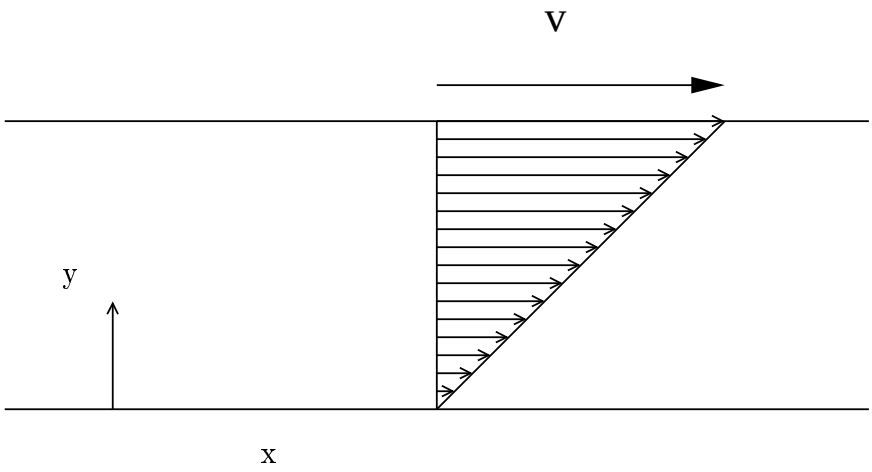


FIGURE 2.5. A simple shear flow

Now that we have the model for the polymer and know the nature of the flow, we can write the equation of motion for the beads. We analyze the different forces acting on the bead as they move in the flow. In doing so, we assume that the inertial term is zero owing to the negligible mass and the sluggish motion they undergo. Each bead experiences the following forces

(a) Hydrodynamic drag force $\mathbf{F}_\nu^{(h)}$. This is the force of resistance the bead experiences as it moves through the solution. Under the assumption that the beads are spherical in shape, an expression for this force can be written using Stokes' law.

$$\mathbf{F}_\nu^{(h)} = -\xi \cdot [\dot{\boldsymbol{r}}_\nu - \mathbf{v}_\nu] \tag{2.2.8}$$

According to this law, the force on bead ν is directly proportional to the difference between the bead velocity $\dot{\boldsymbol{r}}_\nu$ and the velocity of the solution $\mathbf{v}_\nu$ at bead ν. The parameter ξ is the Stokes' friction coefficient and the minus sign appears since the force is repulsive in nature. The solvent velocity $\mathbf{v}_\nu$ is taken as $\boldsymbol{\kappa}\boldsymbol{r}_\nu$, where $\boldsymbol{\kappa}$ is the gradient of the velocity field.

Due to the interaction between the solvent molecules and a bead the flow can be perturbed and this in turn can affect the motion of the other beads. This phenomenon is called the *hydrodynamic interaction* but we shall neglect this in our study.

(b) Brownian force $\mathbf{F}_\nu^{(b)}$. Due to the thermal fluctuations of the solvent molecules, the bead experiences a random force and this force is modeled by a Wiener process.

Definition 2.2.1. *A n-dimensional Wiener process on* $[0,T]$ *is a random process* $\boldsymbol{W}(t)$ *with values in* $\boldsymbol{R}^n$*, which depends continuously on* $t \in [0,T]$ *and satisfies the following conditions*

$$\begin{aligned} \boldsymbol{W}(0) &= 0 \\ \boldsymbol{W}(t) - \boldsymbol{W}(s) &\sim \sqrt{t-s}\,\mathcal{N}(0,I) \end{aligned} \tag{2.2.9}$$

where $\mathcal{N}(0,I)$ *is the standard normal distribution with mean 0 and covariance matrix* I*.*

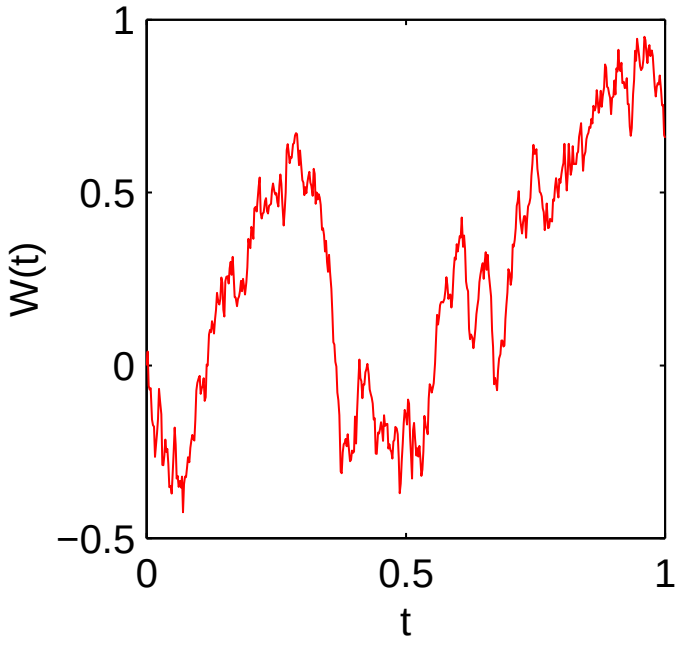


FIGURE 2.6. Trajectory of a Wiener process

The random force experienced by a bead is then given by

$$\mathbf{F}_\nu^{(b)}\,dt = \sqrt{2k_BT\xi}\,d\boldsymbol{W}_\nu \tag{2.2.10}$$

where $\boldsymbol{W}_\nu$ is a three dimensional Wiener process. The factor $\sqrt{2k_BT\xi}$ signifies the fact that the energy of the solvent molecules is due to the temperature of the solvent, T, and this energy influences the collision with the beads.

(c) Potential force $\mathbf{F}_\nu^{(\phi)}$. The potential force results from two contributions. One is the traditional spring potential, ϕ_{sp} which is due to the presence of the springs and prevents the beads from going too far apart. The potential is

attractive and is quadratic in nature

$$\phi_{sp} = \tfrac{1}{2}\sum_i H\mathbf{Q}_i \cdot \mathbf{Q}_i \tag{2.2.11}$$

where H is the spring constant.
The other one is the *excluded volume potential* ϕ_{ev}, which prevents the beads from coming too close to each other. This is a repulsive potential and is modeled as a narrow Gaussian (see section 4.3.1 of [**21**]).

$$\phi_{ev} = k_B T \frac{z}{d^3} \sum_{\substack{\mu,\nu=1 \\ \mu\neq\nu}}^{N} \exp\left(-\frac{H}{k_B T}\frac{r_{\mu\nu}^2}{2d^2}\right). \tag{2.2.12}$$

Here, $r_{\mu\nu}$ is the magnitude of the vector $\boldsymbol{r}_{\mu\nu} = \boldsymbol{r}_\mu - \boldsymbol{r}_\nu$, connecting the pair of beads μ and ν. The parameter d controls the *extent* of the repulsive potential, and z describes its *strength*. Note that as d approaches zero, ϕ_{ev} behaves like a delta function (see figure 2.7).

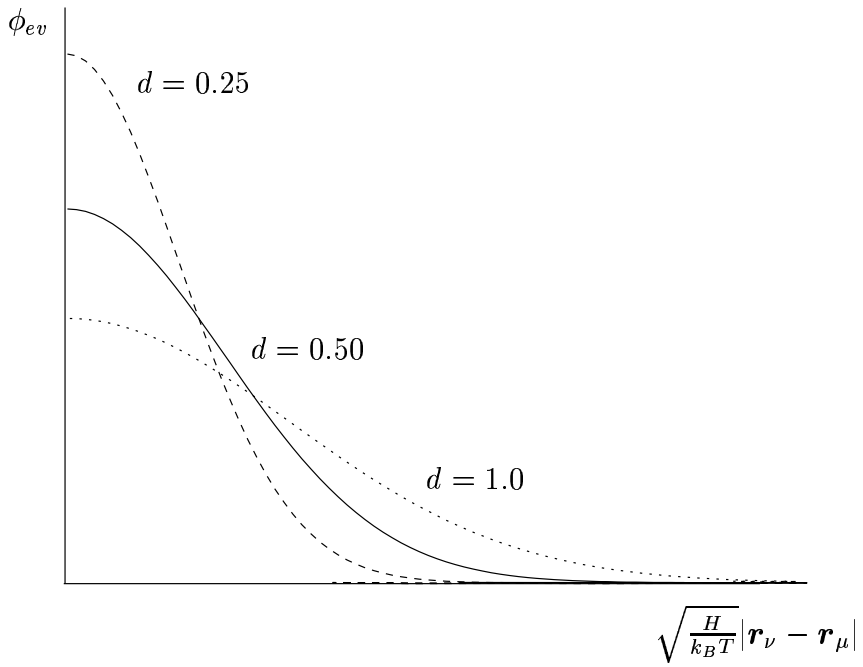


FIGURE 2.7. Excluded volume potential for various values of the parameter d.

The total potential is the sum of the spring potential (2.2.11) and the excluded volume (2.2.12), $\phi = \phi_{sp} + \phi_{ev}$ and the potential force is given by

$$\mathbf{F}_\nu^{(\phi)} = -\nabla\phi \tag{2.2.13}$$

(d)External force $\mathbf{F}_\nu^{(e)}$. There could also be external forces like electrical forces, gravity etc, but we shall not consider them for the present study.

2.3. Diffusion equation for the configurational distribution function

As the bead-spring system moves in the flow and there are random forces, the exact configuration of the system cannot be determined, but rather only a probabilistic estimate can be given. This probability, which describes the likelihood of finding the chain in a particular configuration is called the configurational distribution function ψ. We shall now derive the Fokker-Planck equation for the configurational distribution function.

As mentioned earlier in the previous section, we shall neglect the bead inertia. Writing the force balance with the above assumption leads us to

$$(\mathbf{F}_\nu^{(h)} + \mathbf{F}_\nu^{(b)} + \mathbf{F}_\nu^{(\phi)})\, dt \;=\; 0 \tag{2.3.14}$$

Substituting (2.2.8), (2.2.10), (2.2.13) and using $\dot{\boldsymbol{r}}_\nu dt = d\boldsymbol{r}_\nu$, we get

$$\begin{aligned} d\boldsymbol{r}_\nu &= \left[\mathbf{v}_\nu - \frac{1}{\xi}\frac{\partial \phi}{\partial \boldsymbol{r}_\nu}\right] dt + \sqrt{\frac{2k_BT}{\xi}}\, d\mathbf{W}_\nu \\ &= \left[\boldsymbol{\kappa}\boldsymbol{r}_\nu - \frac{1}{\xi}\frac{\partial \phi}{\partial \boldsymbol{r}_\nu}\right] dt + \sqrt{\frac{2k_BT}{\xi}}\, d\mathbf{W}_\nu \end{aligned} \tag{2.3.15}$$

This is nothing but a stochastic differential equation for the bead position. We now transform the equation to the connector vectors $\boldsymbol{Q}_j$. Thus we move from a system with $3N$ variables to a system with $3(N-1)$ variables.

Subtracting (2.3.15) for $\nu = j$ and $\nu = j+1$, we get

$$\begin{aligned} d\boldsymbol{r}_{j+1} - d\boldsymbol{r}_j = &\left[\boldsymbol{\kappa}(\boldsymbol{r}_{j+1} - \boldsymbol{r}_j) - \frac{1}{\xi}\left(\frac{\partial \phi}{\partial \boldsymbol{r}_{j+1}} - \frac{\partial \phi}{\partial \boldsymbol{r}_j}\right)\right] dt \\ &+ \sqrt{\frac{2k_BT}{\xi}}\left[d\mathbf{W}_{j+1} - d\mathbf{W}_j\right] \end{aligned} \tag{2.3.16}$$

This can be simplified into

$$d\boldsymbol{Q}_j = \left[\boldsymbol{\kappa}\boldsymbol{Q}_j - \frac{1}{\xi}\sum_\nu \bar{\mathbf{B}}_{j\nu}\frac{\partial \phi}{\partial \boldsymbol{r}_\nu}\right] dt + \sqrt{\frac{2k_BT}{\xi}}\left[\sum_\nu \bar{\mathbf{B}}_{j\nu}\, d\mathbf{W}_\nu\right] \tag{2.3.17}$$

where $\bar{\mathbf{B}}_{j\nu}$ is the transformation matrix introduced earlier. We now rewrite the derivative of ϕ in terms of the $\boldsymbol{Q}_i$s. Observe that

$$\frac{\partial \phi}{\partial \boldsymbol{r}_\nu} = \sum_k \frac{\partial \phi}{\partial \boldsymbol{Q}_k} \frac{\partial \boldsymbol{Q}_k}{\partial \boldsymbol{r}_\nu} = \sum_k \bar{\mathbf{B}}_{k\nu} \frac{\partial \phi}{\partial \boldsymbol{Q}_k} \tag{2.3.18}$$

Hence (2.3.17) can be rewritten as

$$d\boldsymbol{Q}_j = \left[\boldsymbol{\kappa}\boldsymbol{Q}_j - \frac{1}{\xi}\sum_k \boldsymbol{A}_{jk}\frac{\partial \phi}{\partial \boldsymbol{Q}_k}\right]dt + \sqrt{\frac{2k_BT}{\xi}}\left[\sum_\nu \bar{\mathbf{B}}_{j\nu}\, d\mathbf{W}_\nu\right] \tag{2.3.19}$$

From the theory of stochastic differential equation, (see [**21**]) we know that

$$d\boldsymbol{X}_t = \boldsymbol{A}(t, \boldsymbol{X}_t)\, dt + \mathbf{B}(t, \boldsymbol{X}_t) d\boldsymbol{W}_t \tag{2.3.20}$$

is related to the partial differential equation

$$\frac{\partial}{\partial t} p(t, \boldsymbol{x}) = -\frac{\partial}{\partial \boldsymbol{x}} \cdot \left[\boldsymbol{A}(t, \boldsymbol{x}) p(t, \boldsymbol{x})\right] + \frac{1}{2}\frac{\partial}{\partial \boldsymbol{x}}\frac{\partial}{\partial \boldsymbol{x}} : \left[\boldsymbol{D}(t, \boldsymbol{x}) p(t, \boldsymbol{x})\right] \tag{2.3.21}$$

where $p(t, \boldsymbol{x})$ is the probability density that characterizes the continuous distribution $\boldsymbol{X}_t$, $\boldsymbol{D}(t, \boldsymbol{x}) = \mathbf{B}(t, \boldsymbol{x})\mathbf{B}^T(t, \boldsymbol{x})$,

$$\frac{\partial}{\partial \boldsymbol{x}} \cdot \left[\boldsymbol{A}(t, \boldsymbol{x}) p(t, \boldsymbol{x})\right] = \sum_k \frac{\partial}{\partial x_k}(A_i(t, \boldsymbol{x}) p(t, \boldsymbol{x})) \tag{2.3.22}$$

and

$$\frac{\partial}{\partial \boldsymbol{x}}\frac{\partial}{\partial \boldsymbol{x}} : \left[\boldsymbol{D}(t, \boldsymbol{x}) p(t, \boldsymbol{x})\right] = \sum_{j,k} \frac{\partial}{\partial x_j}\frac{\partial}{\partial x_k} D_{jk}(t, \boldsymbol{x}) p(t, \boldsymbol{x}) \tag{2.3.23}$$

Hence the Fokker-Planck equation for the configurational distribution function ψ can be written down from (2.3.17) using (2.1.4) as,

$$\frac{\partial \psi}{\partial t} = -\sum_j \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \left(\boldsymbol{\kappa}\boldsymbol{Q}_j - \frac{1}{\xi}\sum_k A_{jk}\, \frac{\partial \phi}{\partial \boldsymbol{Q}_k}\right)\psi + \frac{k_BT}{\xi}\sum_{j,k} A_{jk}\, \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \psi}{\partial \boldsymbol{Q}_k} \tag{2.3.24}$$

For the investigation of the flow behavior of polymeric liquids, the calculation of the stress tensor $\boldsymbol{\tau}$ is of special interest. The stress tensor consists of two contributions, one from the solvent $\boldsymbol{\tau}^s$, and the other from the polymer $\boldsymbol{\tau}^p$,

$$\boldsymbol{\tau} = \boldsymbol{\tau}^s + \boldsymbol{\tau}^p \tag{2.3.25}$$

The rheological properties of the polymer solution can be obtained by calculating the polymer contribution to the stress tensor, $\boldsymbol{\tau}^p$, which is given by

Kramers expression (see [**2**]),

$$\boldsymbol{\tau}^p = -\sum_j \int_{\mathbb{R}^{3(N-1)}} \boldsymbol{Q}_j \otimes \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \psi d\boldsymbol{Q} \tag{2.3.26}$$

where ψ is the steady state solution of (2.3.24).

We know for simple shear flow that there are only four distinct components of the stress tensor, $\tau^p_{xx}, \tau^p_{xy}, \tau^p_{yy}$ and τ^p_{zz}. Furthermore, we can show that, $\tau^p_{yy} = \tau^p_{zz}$ (see [**1**]). The two important rheological properties of a dilute polymer solution, undergoing simple shear flow, that can then be calculated are, the *viscosity,*

$$\eta = -\frac{\tau^p_{xy}}{\beta} \tag{2.3.27}$$

and the *first normal-stress-difference* coefficient,

$$\Psi = -\frac{\tau^p_{xx} - \tau^p_{yy}}{\beta^2} \tag{2.3.28}$$

The problem can be summarized as follows. We need to find the stationary solution of (2.3.24) and evaluate the stress tensor $\boldsymbol{\tau}^p$ to calculate the quantities η and Ψ given by (2.3.27) and (2.3.28) respectively.

2.4. Analytical solutions for special flows

We start by defining certain flow states in terms of the gradient $\boldsymbol{\kappa}$ of the velocity field.

Definition 2.4.2. *The flow field $v(\boldsymbol{x}) = \boldsymbol{\kappa}\boldsymbol{x}$ is called*

1. *fluid at rest when $\boldsymbol{\kappa} = 0$*
2. *incompressible when $tr(\boldsymbol{\kappa}) = 0$*
3. *potential flow when $\boldsymbol{\kappa}$ is symmetric*
4. *shear flow when*

$$\boldsymbol{\kappa} = \begin{pmatrix} 0 & \beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

We first look for a stationary solution of (2.3.24) for the case $\boldsymbol{\kappa} = 0$. The solution so obtained is called the *equilibrium distribution* ψ_{eq}.

Lemma 2.1. *ψ_{eq} given by*

$$\psi_{eq} = N_{eq} \exp\left[-\phi/k_B T\right] \tag{2.4.29}$$

where N_{eq} is the normalization constant, is a solution of (2.3.24) in the stationary case of a fluid at rest.

Proof.

$$\frac{\partial \psi_{eq}}{\partial \boldsymbol{Q}_k} = N_{eq} \exp\left[-\phi/k_B T\right] \left(\frac{-1}{k_B T}\right) \cdot \frac{\partial \phi}{\partial \boldsymbol{Q}_k} = -\frac{\psi_{eq}}{k_B T} \frac{\partial \phi}{\partial \boldsymbol{Q}_k} \tag{2.4.30}$$

Also in the stationary case $\partial \psi_{eq}/\partial t = 0$ and $\boldsymbol{\kappa} = 0$, hence the right hand side of (2.3.24) reduces to

$$\frac{1}{\xi} \sum_{j,k} \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \left(A_{jk} \frac{\partial \phi}{\partial \boldsymbol{Q}_k} \right) \psi_{eq} + \frac{k_B T}{\xi} \sum_{j,k} A_{jk} \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \psi_{eq}}{\partial \boldsymbol{Q}_k} \tag{2.4.31}$$

Substituting the expression for $\partial \psi_{eq}/\partial \boldsymbol{Q}_k$ from (2.4.30), we get

$$\frac{1}{\xi} \sum_{j,k} \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \left(A_{jk} \frac{\partial \phi}{\partial \boldsymbol{Q}_k} \right) \psi_{eq} - \frac{1}{\xi} \sum_{j,k} A_{jk} \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \left(\frac{\partial \phi}{\partial \boldsymbol{Q}_k} \right) \psi_{eq} = 0$$

Hence (2.3.24) is satisfied and that proves our lemma. ■

The constant N_{eq} can be obtained using the condition, $\int \psi d\boldsymbol{Q} = 1$, ψ being a probability density. Having obtained the equilibrium distribution, we are now interested in solving (2.3.24) for incompressible potential flows at steady state. We have the following lemma.

Lemma 2.2. *A solution of* (2.3.24) *for an incompressible potential flow at steady state is given by*

$$\psi = N_{fl} \exp\left[\frac{\xi}{2 k_B T} \sum_{m,n} C_{mn} \boldsymbol{Q}_m^T \boldsymbol{\kappa} \boldsymbol{Q}_n \right] \psi_{eq}$$

where $\boldsymbol{C}$ is the matrix given by (2.1.5).

Proof. We write,

$$\psi(\boldsymbol{Q}, t) = \psi_{eq}(\boldsymbol{Q})\, \phi_{fl}(\boldsymbol{Q}, t) \tag{2.4.32}$$

where the first term in the product is the equilibrium distribution and is given by (2.4.29) and ϕ_{fl} is the fluid part which needs to be evaluated (as in [**2**]).

We now substitute the above expression for ψ in (2.3.24) to get the governing equation for ϕ_{fl}. With ψ given by (2.4.32), we have

$$\frac{\partial \psi}{\partial t} = \psi_{eq} \frac{\partial \phi_{fl}}{\partial t} \tag{2.4.33}$$

and

$$\frac{\partial \psi}{\partial \boldsymbol{Q}_k} = \frac{\partial \psi_{eq}}{\partial \boldsymbol{Q}_k}\phi_{fl} + \psi_{eq}\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} = -\frac{\psi_{eq}}{k_B T}\phi_{fl}\frac{\partial \phi}{\partial \boldsymbol{Q}_k} + \psi_{eq}\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} \tag{2.4.34}$$

Substituting (2.4.34) in (2.3.24) we get

$$\begin{aligned}
\psi_{eq}\frac{\partial \phi_{fl}}{\partial t} &= -\sum_j \frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left[\boldsymbol{\kappa}\boldsymbol{Q}_j\,\psi_{eq}\phi_{fl}\right] + \frac{1}{\xi}\sum_j\sum_k A_{jk}\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left[\frac{\partial \phi}{\partial \boldsymbol{Q}_k}\,\psi_{eq}\,\phi_{fl}\right] \\
&+ \frac{k_B T}{\xi}\sum_{j,k} A_{jk}\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left[\psi_{eq}\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} - \frac{\psi_{eq}}{k_B T}\phi_{fl}\frac{\partial \phi}{\partial \boldsymbol{Q}_k}\right] \\
&= -\sum_j \frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left[\boldsymbol{\kappa}\boldsymbol{Q}_j\,\psi_{eq}\phi_{fl}\right] + \frac{k_B T}{\xi}\sum_{j,k} A_{jk}\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left[\psi_{eq}\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k}\right] \qquad (2.4.35)
\end{aligned}$$

Now

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left[(\boldsymbol{\kappa}\boldsymbol{Q}_j)\phi_{fl}\psi_{eq}\right] &= (\boldsymbol{\kappa}\boldsymbol{Q}_j)\cdot\frac{\partial}{\partial \boldsymbol{Q}_j}(\phi_{fl}\psi_{eq}) + \phi_{fl}\psi_{eq}\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot(\boldsymbol{\kappa}\boldsymbol{Q}_j) \\
&= (\boldsymbol{\kappa}\boldsymbol{Q}_j)\cdot\frac{\partial}{\partial \boldsymbol{Q}_j}(\phi_{fl}\psi_{eq}) + tr(\boldsymbol{\kappa})\phi_{fl}\psi_{eq} \\
&= (\boldsymbol{\kappa}\boldsymbol{Q}_j)\cdot\left(\psi_{eq}\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_j} + \phi_{fl}\left(\frac{-\psi_{eq}}{k_B T}\right)\frac{\partial \phi}{\partial \boldsymbol{Q}_j}\right) \\
&= \psi_{eq}(\boldsymbol{\kappa}\boldsymbol{Q}_j)\cdot\left(\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_j} - \left(\frac{\phi_{fl}}{k_B T}\right)\frac{\partial \phi}{\partial \boldsymbol{Q}_j}\right) \qquad (2.4.36)
\end{aligned}$$

The term $\partial/\partial \boldsymbol{Q}_j \cdot (\boldsymbol{\kappa}\boldsymbol{Q}_j)$ drops out in the second step since $tr(\boldsymbol{\kappa}) = 0$ for incompressible flows.
Again

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left[\psi_{eq}\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k}\right] &= -\frac{\psi_{eq}}{k_B T}\frac{\partial \phi}{\partial \boldsymbol{Q}_j}\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} + \psi_{eq}\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\left(\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k}\right) \\
&= \psi_{eq}\left[\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} - \frac{1}{k_B T}\frac{\partial \phi}{\partial \boldsymbol{Q}_j}\cdot\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k}\right] \qquad (2.4.37)
\end{aligned}$$

Substituting (2.4.36) and (2.4.37) in (2.4.35), we get

$$\begin{aligned}
\psi_{eq}\frac{\partial \phi_{fl}}{\partial t} &= -\psi_{eq}\sum_j (\boldsymbol{\kappa}\boldsymbol{Q}_j)\cdot\left[\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_j} - \frac{\phi_{fl}}{k_B T}\frac{\partial \phi}{\partial \boldsymbol{Q}_j}\right] \\
&+ \frac{k_B T}{\xi}\psi_{eq}\sum_{j,k} A_{jk}\left[\frac{\partial}{\partial \boldsymbol{Q}_j}\cdot\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} - \frac{1}{k_B T}\frac{\partial \phi}{\partial \boldsymbol{Q}_j}\cdot\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k}\right] \qquad (2.4.38)
\end{aligned}$$

Since $\psi_{eq} \neq 0$, we get

$$\begin{aligned} \frac{\partial \phi_{fl}}{\partial t} &= -\sum_j (\boldsymbol{\kappa} \boldsymbol{Q}_j) \cdot \left[\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_j} - \frac{\phi_{fl}}{k_B T} \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \right] \\ &+ \frac{k_B T}{\xi} \sum_{j,k} A_{jk} \left[\frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} - \frac{1}{k_B T} \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} \right] \end{aligned} \tag{2.4.39}$$

We rearrange this equation as

$$\begin{aligned} \frac{\partial \phi_{fl}}{\partial t} &= \frac{k_B T}{\xi} \sum_{j,k} A_{jk} \left[\frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} \right] - \sum_j (\boldsymbol{\kappa} \boldsymbol{Q}_j) \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_j} \\ &+ \sum_j \left[\left(\frac{\phi_{fl}}{k_B T} (\boldsymbol{\kappa} \boldsymbol{Q}_j) - \frac{1}{\xi} \sum_k A_{jk} \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} \right) \cdot \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \right] \end{aligned} \tag{2.4.40}$$

We now try as ansatz the function

$$\phi_{fl} = N_{fl} \exp \left[\alpha \sum_{m,n} C_{mn} \, \boldsymbol{Q}_m^T \, \boldsymbol{\kappa} \, \boldsymbol{Q}_n \right] \tag{2.4.41}$$

where $\boldsymbol{C}$ is the inverse of the Rouse matrix, N_{fl} is the normalization constant and α is a constant to be determined. $\boldsymbol{Q}_j$ is always taken as a column vector and the superscript T on these vectors refer to the transpose. Using the fact that $\boldsymbol{\kappa}$ is symmetric, we get

$$\frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} = 2\alpha \, \phi_{fl} \left[\sum_n C_{kn} (\boldsymbol{\kappa} \boldsymbol{Q}_n) \right].$$

Now, since $\boldsymbol{A}$ is the inverse of $\boldsymbol{C}$,

$$\frac{1}{\xi} \sum_k A_{jk} \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} = \frac{2\alpha \phi_{fl}}{\xi} \sum_{k,n} A_{jk} C_{kn} (\boldsymbol{\kappa} \boldsymbol{Q}_n) = \frac{2\alpha \phi_{fl}}{\xi} (\boldsymbol{\kappa} \boldsymbol{Q}_j)$$

and using again $tr(\boldsymbol{\kappa}) = 0$, we eventually find

$$\frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} = \sum_{r,s} 2\alpha \, \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_{jr}} \left[\sum_n C_{kn} \, \kappa_{rs} \, Q_{ns} \right]. \tag{2.4.42}$$

The variables r and s run from 1 to 3 in the rest of the calculation. So,

$$\begin{aligned} \frac{k_B T}{\xi} \sum_{j,k} A_{jk} \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_k} &= 2\alpha \, \frac{k_B T}{\xi} \left[\sum_{j,k} \sum_{r,s} \sum_n A_{jk} \, C_{kn} \, \kappa_{rs} \, Q_{ns} \, \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_{jr}} \right] \\ &= 2\alpha \, \frac{k_B T}{\xi} \left[\sum_{j,n} \sum_{r,s} \delta_{jn} \, \kappa_{rs} \, \boldsymbol{Q}_{ns} \, \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_{jr}} \right] \\ &= 2\alpha \, \frac{k_B T}{\xi} \left[\sum_j \left((\boldsymbol{\kappa} \boldsymbol{Q}_j) \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_j} \right) \right] \end{aligned}$$

So (2.4.40) can now be written as,

$$\begin{aligned}\frac{\partial \phi_{fl}}{\partial t} &= (2\alpha \frac{k_B T}{\xi} - 1) \sum_j \left((\boldsymbol{\kappa} \boldsymbol{Q}_j) \cdot \frac{\partial \phi_{fl}}{\partial \boldsymbol{Q}_j} \right) \\ &+ \phi_{fl} \left(\frac{1}{k_B T} - \frac{2\alpha}{\xi} \right) \sum_j \left[(\boldsymbol{\kappa} \boldsymbol{Q}_j) \cdot \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \right]\end{aligned}$$

From the above it is clear that when $\alpha = \frac{\xi}{2k_B T}$, (2.3.24) is satisfied at steady state. Thus the steady state solution can be written as

$$\psi = N_{fl} \exp \left[\frac{\xi}{2k_B T} \sum_{m,n} C_{mn} \boldsymbol{Q}_m^T \boldsymbol{\kappa} \boldsymbol{Q}_n \right] \psi_{eq}$$

and the lemma is proved. ■

To fix N_{fl} we use the normalization condition $\int \psi = 1$, ψ being a probability density. Thus we get,

$$N_{fl} \left\langle \exp \left[\frac{\xi}{2k_B T} \sum_{m,n} C_{mn} \boldsymbol{Q}_m^T \boldsymbol{\kappa} \boldsymbol{Q}_n \right] \right\rangle = 1$$

where, $\langle X \rangle$ denotes the average of any quantity X,

$$\langle X \rangle = \int X \, \psi \, d\boldsymbol{Q}_1 d\boldsymbol{Q}_2 \ldots d\boldsymbol{Q}_{N-1}$$

It may be noted that the above solution holds good for any choice of the potential ϕ.

We have thus been able to find analytical stationary solutions for the cases (a) the fluid at rest and (b) incompressible potential flows. It is now desired to consider the case of shear flow. But since there is no analytical solution possible in this case, we must resort to numerical simulations. Before we do that, we first simplify our equation.

2.5. Algebraic simplifications

We first non-dimensionalize the various quantities appearing in our equation (2.3.24). After that the anisotropic diffusion term is transformed into the Laplacian by a suitable change of variables.

2.5.1. Non-dimensionalisation. We introduce the following dimensionless variables

$$t' = \frac{t}{\lambda_H}, \quad \boldsymbol{Q}'_k = \frac{\boldsymbol{Q}_k}{\sqrt{\frac{k_B T}{H}}}, \quad \boldsymbol{\kappa'} = \lambda_H \boldsymbol{\kappa}, \quad \phi'(\boldsymbol{Q}') = \frac{\phi(\boldsymbol{Q})}{k_B T} \tag{2.5.43}$$

where $\lambda_H = \xi/4H$ is the time constant and $\sqrt{\frac{k_B T}{H}}$ is the length scale. Our equation (2.3.24) written in terms of the non-dimensional variables reads as

$$\begin{aligned}\frac{4H}{\xi}\frac{\partial \psi'}{\partial t'} &= -\sum_j \sqrt{\frac{H}{k_B T}}\frac{\partial}{\partial \boldsymbol{Q}'_j}\cdot\left(\frac{4H}{\xi}\boldsymbol{\kappa'}\sqrt{\frac{k_B T}{H}}\boldsymbol{Q}'_j - \frac{k_B T}{\xi}\sum_k A_{jk}\sqrt{\frac{H}{k_B T}}\frac{\partial \phi'}{\partial \boldsymbol{Q}'_k}\right)\psi' \\ &+ \frac{k_B T}{\xi}\sum_{j,k} A_{jk}\sqrt{\frac{H}{k_B T}}\frac{\partial}{\partial \boldsymbol{Q}'_j}\cdot\sqrt{\frac{H}{k_B T}}\frac{\partial \psi'}{\partial \boldsymbol{Q}'_k} \\ &= -\sum_j \frac{\partial}{\partial \boldsymbol{Q}'_j}\cdot\left(\frac{4H}{\xi}\boldsymbol{\kappa'}\boldsymbol{Q}'_j - \frac{H}{\xi}\sum_k A_{jk}\frac{\partial \phi'}{\partial \boldsymbol{Q}'_j}\right)\psi' + \frac{H}{\xi}\sum_{j,k} A_{jk}\frac{\partial}{\partial \boldsymbol{Q}'_j}\cdot\frac{\partial \psi'}{\partial \boldsymbol{Q}'_k}\end{aligned}$$

which simplifies to,

$$\frac{\partial \psi'}{\partial t'} = -\sum_j \frac{\partial}{\partial \boldsymbol{Q}'_j}\cdot\left(\boldsymbol{\kappa'}\boldsymbol{Q}'_j - \frac{1}{4}\sum_k A_{jk}\frac{\partial \phi'}{\partial \boldsymbol{Q}'_k}\right)\psi' + \frac{1}{4}\sum_{j,k} A_{jk}\frac{\partial}{\partial \boldsymbol{Q}'_j}\cdot\frac{\partial \psi'}{\partial \boldsymbol{Q}'_k} \tag{2.5.44}$$

2.5.2. Transformation to normal coordinates. In the diffusion equation (2.3.24), we note that there is coupling among the various connector vectors because of the Rouse matrix A_{ij}. This coupling can be removed by diagonalising the Rouse matrix and we specifically carry out by introducing a new set of variables $\boldsymbol{Q}^*_j$. The Cartesian components of $\boldsymbol{Q}^*_j$ are called the normal coordinates. We now introduce the $(N-1)\times(N-1)$ orthogonal matrix Ω_{ij} which diagonalises A_{ij}, given by

$$\Omega_{ij} = \sqrt{\frac{2}{N}}\sin\frac{ij\pi}{N} \qquad i,j = 1,\ldots,N \tag{2.5.45}$$

and satisfies the relations

$$\begin{aligned}\sum_k \Omega_{kj}\Omega_{ki} &= \delta_{ji} \\ \sum_j\sum_k \Omega_{ji}A_{jk}\Omega_{kl} &= a_l\delta_{il}\end{aligned} \tag{2.5.46}$$

where a_i are the eigenvalues of A_{ij}. If we denote the matrix $a_l\delta_{il}$ by D, then the last relation can be written as

$$\Omega^T A \Omega = D$$

Hence,

$$\left(\Omega\sqrt{D}^{-1}\right)^T A\left(\Omega\sqrt{D}^{-1}\right) = I \tag{2.5.47}$$

since D is a diagonal matrix. Utilizing the above we make the transformation $Z = 2\Omega\sqrt{D}^{-1}$. This actually transforms A into

$$\sum_{j,k} Z_{ji} A_{jk} Z_{kl} = 4\ \delta_{il} \tag{2.5.48}$$

and satisfies

$$\sum_k Z_{kj} Z_{ki} = 4\frac{\delta_{ji}}{a_j}$$

The advantage of this transformation is that the second derivatives of ψ occurring in (2.5.44) reduces to a Laplacian. The connector vectors transform as

$$\boldsymbol{Q}_j^* = \sum_k Z_{jk} \boldsymbol{Q}_k' \tag{2.5.49}$$

and the derivatives transform as follows

$$\frac{\partial}{\partial \boldsymbol{Q}_j'} = \sum_k Z_{kj} \frac{\partial}{\partial \boldsymbol{Q}_k^*} \tag{2.5.50}$$

With these, (2.5.44) reduces to

$$\frac{\partial \psi'}{\partial t'} = -\sum_j \frac{\partial}{\partial \boldsymbol{Q}_j^*} \cdot \left(\boldsymbol{\kappa}' \boldsymbol{Q}_j^* - \frac{\partial \phi'}{\partial \boldsymbol{Q}_j^*}\right)\psi' + \sum_j \frac{\partial}{\partial \boldsymbol{Q}_j^*} \cdot \frac{\partial \psi'}{\partial \boldsymbol{Q}_j^*} \tag{2.5.51}$$

The polymer contribution to the stress tensor, $\boldsymbol{\tau}^p$ can be written down in terms of the non-dimensional variables as

$$\boldsymbol{\tau}^p = -\sum_j \int_{\mathbb{R}^{3(N-1)}} \boldsymbol{Q}_j^* \otimes \frac{\partial \phi'}{\partial \boldsymbol{Q}_j^*} \psi' d\boldsymbol{Q}^* \tag{2.5.52}$$

The scaled problem can now be formulated as follows: Solve (2.5.51) for steady state and then evaluate (2.5.52) to calculate the non-dimensional viscosity η' given by

$$\eta' = -\frac{\tau_{xy}^p}{\beta'} \tag{2.5.53}$$

and the non-dimensional *first normal-stress-difference* coefficient Ψ' given by,

$$\Psi' = -\frac{\tau_{xx}^p - \tau_{yy}^p}{(\beta')^2} \tag{2.5.54}$$

where $\beta' = \beta\lambda_H$.

Since we are interested in calculating the steady state solution, the choice of the initial condition only matters to that extent that how close we start to the final solution. We later employ several initial conditions,

1. the position of the connector vectors are independent Gaussian distributed with mean zero and variance one.
2. all beads are on top of each other at the origin, i.e., $\psi(\boldsymbol{Q}, 0) = \delta_0(\boldsymbol{Q})$.
3. take $\psi(\boldsymbol{Q}, 0) = \psi_{eq}$, that is we start with the equilibrium steady state solution of the flow problem corresponding to fluid at rest.

Numerical simulations using Monte Carlo methods have been carried out for the special case of dumbbells and the above two viscometric functions have been calculated by Ravi Prakash and Öttinger [**22**] in the presence of excluded volume. It is now desired to compute these material functions for a larger number of beads.

For the sake of simplicity, we shall drop the dashes and stars appearing in our equation and shall consider the non-dimensional form in the rest of the thesis.

CHAPTER 3

Existence of Solutions and the Splitting Method

In this chapter we shall address the issue of existence and uniqueness of solution of equation (2.5.51), and later outline the splitting approach. The existence theorem assures the existence of a unique classical solution for our parabolic equation. In our numerical solution we will split the convection part and the diffusion part to simulate the Fokker-Plank equation. Therefore in the splitting method, instead of considering the problem as a whole, we consider two sub-problems and the approximate solution of the original problem is written as a composition of the solution operators of the sub-problems. We conclude the chapter by showing the convergence of the splitting approach.

3.1. Existence and uniqueness of solution

In the previous chapter we derived the Fokker-Planck equation for the configuration distribution function ψ. Written in terms of the non-dimensional variables $\boldsymbol{Q}$ and t,

$$\frac{\partial \psi}{\partial t} = -\sum_j \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \left(\boldsymbol{\kappa} \cdot \boldsymbol{Q}_j - \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \right) \psi + \sum_j \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \psi}{\partial \boldsymbol{Q}_j} \tag{3.1.1}$$

$$\psi(\boldsymbol{Q}, 0) = \psi_0(\boldsymbol{Q})$$

If we denote,

$$L = \sum_j \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial}{\partial \boldsymbol{Q}_j} - \sum_j \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \left(\boldsymbol{\kappa} \cdot \boldsymbol{Q}_j - \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \right) - \frac{\partial}{\partial t} \tag{3.1.2}$$

then (3.1.1) can be rewritten as

$$L\psi = 0, \quad \psi(\boldsymbol{Q}, 0) = \psi_0(\boldsymbol{Q}) \tag{3.1.3}$$

We have the following definition.

Definition 3.1.1. *An operator M associated with the partial differential equation*

$$Mu = \sum_{i,j=1}^{s} a_{ij}(\boldsymbol{x},t)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i}^{s} b_i(\boldsymbol{x},t)\frac{\partial u}{\partial x_i} + c(\boldsymbol{x},t)u - \frac{\partial u}{\partial t} = 0 \tag{3.1.4}$$

where a_{ij}, b_i and c are defined in $\Omega = \bar{D} \times [T_0, T_1]$, $D \subseteq \mathbb{R}^s$ is said to be parabolic at the point $(\boldsymbol{x}, t)$ if the matrix $a_{ij}(\boldsymbol{x}, t)$ is positive definite. That is, if for every real vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_s) \neq 0$, we have

$$\sum_{i,j=1}^{s} a_{ij}(\boldsymbol{x},t)\xi_i\xi_j > 0 \tag{3.1.5}$$

If the above condition holds good for all points $(\boldsymbol{x}, t) \in \Omega$, then M is said to be parabolic in Ω. If there exists positive constants, μ_1 and μ_2 such that for any real vector $\boldsymbol{\xi}$,

$$\mu_1\|\boldsymbol{\xi}\|^2 \leq \sum_{i,j=1}^{s} a_{ij}\xi_i\xi_j \leq \mu_2\|\boldsymbol{\xi}\|^2 \tag{3.1.6}$$

for all $(\boldsymbol{x}, t) \in \Omega$, then M is said to be uniformly parabolic in Ω.

Consistent with the definition, we see that the matrix a_{ij} in our case is the identity matrix. Thus (3.1.5) is trivially satisfied and this proves that L is parabolic and in fact uniformly parabolic ($\mu_1 = 1$ and $\mu_2 = 1$).

Definition 3.1.2 (The Cauchy Problem)**.** *Given a function $f(\boldsymbol{x}, t)$ in $\Omega \equiv \mathbb{R}^s \times [0, T]$ and a function $\varphi(\boldsymbol{x})$ in $\mathbb{R}^s$, the problem of finding a function $u(\boldsymbol{x}, t)$ satisfying the parabolic equation*

$$Mu(\boldsymbol{x},t) = f(\boldsymbol{x},t) \qquad \text{in } \Omega \equiv \mathbb{R}^{\mathrm{s}} \times (0, \mathrm{T}], \tag{3.1.7}$$

where M is as defined in (3.1.4), and the initial condition

$$u(\boldsymbol{x}, 0) = \varphi(\boldsymbol{x}) \text{ on } \mathbb{R}^{\mathrm{s}} \tag{3.1.8}$$

is called a Cauchy problem (in the strip $0 \leq t \leq T$).

It can be immediately seen from the above definition that (3.1.3) is a Cauchy problem for ψ. Before we go on to prove the existence theorem, we need the following definition.

Definition 3.1.3. *A fundamental solution of $Mu = 0$ in $\Omega = \mathbb{R}^s \times (0, T]$ is a function $\Gamma(\boldsymbol{x}, t; \boldsymbol{\xi}, \tau)$ defined for all $(\boldsymbol{x}, t), (\boldsymbol{\xi}, \tau) \in \Omega$ with $t > \tau$ which satisfies the following two conditions*

1. *for a fixed* $(\boldsymbol{\xi}, \tau) \in \Omega$, $u(\boldsymbol{x}, t) = \Gamma(\boldsymbol{x}, t; \boldsymbol{\xi}, \tau)$ *satisfies* $Mu = 0$;
2. *for every continuous function* $f(x)$ *in* $\overline{D}$,

$$\lim_{t \searrow \tau} \int_D \Gamma(\boldsymbol{x}, t; \boldsymbol{\xi}, \tau) f(\boldsymbol{\xi}) d\boldsymbol{\xi} = f(\boldsymbol{x}) \tag{3.1.9}$$

wherein $d\boldsymbol{\xi} = d\xi_1 \cdots d\xi_s$.

The procedure of constructing the fundamental solution for the Cauchy problem is outlined in [**7**] and is based on the parametrix method of E. E. Levy. We have the following theorem on the existence of solutions taken from [**7**].

Theorem 3.1.1. *Suppose that*

A. M *is uniformly parabolic in* $\Omega = \mathbb{R}^s \times [0, T]$;
B. *the coefficients of* M *are bounded continuous functions in* $\mathbb{R}^s$ *and there exists* $A \in \mathbb{R}$ *and* $\alpha \in (0, 1]$ *such that for all* $(\boldsymbol{x}, t), (\boldsymbol{x}^0, t^0) \in \Omega$

$$|a_{ij}(\boldsymbol{x}, t) - a_{ij}(\boldsymbol{x}^0, t^0)| \leq A(\|\boldsymbol{x} - \boldsymbol{x}^0\|^\alpha + |t - t^0|^\alpha) \tag{3.1.10}$$

$$|b_i(\boldsymbol{x}, t) - b_i(\boldsymbol{x}^0, t^0)| \leq A\|\boldsymbol{x} - \boldsymbol{x}^0\|^\alpha \tag{3.1.11}$$

$$|c(\boldsymbol{x}, t) - c(\boldsymbol{x}^0, t^0)| \leq A\|\boldsymbol{x} - \boldsymbol{x}^0\|^\alpha \tag{3.1.12}$$

C. $f(\boldsymbol{x}, t)$ *and* $\varphi(\boldsymbol{x})$ *are continuous functions satisfying*

$$|f(\boldsymbol{x}, t)| \leq \text{const.} \exp[\,\mathrm{h}\,\|\,\boldsymbol{x}\,\|^2], \tag{3.1.13}$$

$$|\varphi(\boldsymbol{x})| \leq \text{const.} \exp[\,\mathrm{h}\,\|\,\boldsymbol{x}\,\|^2] \tag{3.1.14}$$

where h *is any positive constant satisfying*

$$h < \frac{\lambda_0}{4T} \tag{3.1.15}$$

and λ_0 *is a constant depending on* μ_1, μ_2, A.

D. $f(\boldsymbol{x}, t)$ *is locally Hölder continuous of exponent* α *in* $\boldsymbol{x} \in \mathbb{R}^s$, *uniformly with respect to* t;

Then there exists a fundamental solution $\Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, \tau)$ *of* $Mu = 0$ *and the function given by*

$$u(\boldsymbol{x}, t) = \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, 0)\varphi(\boldsymbol{\xi}) d\boldsymbol{\xi} - \int_0^t \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, \tau) f(\boldsymbol{\xi}, \tau) d\boldsymbol{\xi} d\tau \tag{3.1.16}$$

is a solution of the Cauchy problem (3.1.7) *with* (3.1.8) *and satisfies the bound*

$$|u(\boldsymbol{x}, t)| \leq \text{const.} \exp[\,\mathrm{k}\,\|\,\boldsymbol{x}\,\|^2] \tag{3.1.17}$$

where k *is a constant depending on* h, λ_0, T.

Proof. The proof is exactly as given in [**7**] taking $\alpha = 1$. ■

To apply the result to our problem, we map the $3N-3$ components of the $N-1$ connector vectors $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_{N-1}$ to the $3N-3$ components of a single vector $\boldsymbol{x}$. That is,

$$\boldsymbol{x} = (Q_{11}, Q_{12}, Q_{13}, \ldots, Q_{3N-1}, Q_{3N-2}, Q_{3N-3})$$

We define

$$a_{ij} = \delta_{ij} \qquad i, j = 1, \ldots, 3N-3. \tag{3.1.18}$$

The $3N-3$-dimensional vector $\boldsymbol{b}$ is defined as,

$$b_j = \boldsymbol{\kappa} \boldsymbol{Q}_j - \frac{\partial \phi}{\partial \boldsymbol{Q}_j}, \quad j = 1, \ldots, 3N-3$$

where,

$$\phi = \frac{1}{2} \sum_{i=1}^{N-1} \left(\frac{a_i}{4} \sum_{r=1}^{N-1} Z_{ri} \boldsymbol{Q}_r \right) \cdot \left(\frac{a_i}{4} \sum_{m=1}^{N-1} Z_{mi} \boldsymbol{Q}_m \right) + \frac{z}{d^3} \sum_{\substack{\mu,\nu=1 \\ \mu \neq \nu}}^{N} \exp \left[- \frac{\left| \sum_{k=1}^{N-1} \left((B_{\mu k} - B_{\nu k}) \frac{a_k}{4} \sum_{r=1}^{N-1} Z_{rk} \boldsymbol{Q}_r \right) \right|^2}{2d^2} \right],$$

with B, a_i, Z defined in (2.1.3), (2.1.6) and (2.5.48) respectively and z, d are the excluded volume parameters.

The coefficient c is defined by

$$c = \operatorname{div} \boldsymbol{b}.$$

The source term f does not exist in our case. So $f = 0$. Then the equation in (3.1.3) has exactly the form (3.1.4). Now, we just verify the conditions A, B, C and D in theorem 3.1.1.

Since the matrix (a_{ij}), in our case is an identity matrix, A is satisfied. Condition D is also fulfilled since $f = 0$ for our problem. So it remains to show that B and C are satisfied.
Regarding C, (3.1.13) is satisfied again since $f = 0$. Since the initial conditions we shall consider for our problem are of the form const.$\exp(-\gamma |\boldsymbol{x}|^2)$, $\gamma > 0$, we see that (3.1.14) is fulfilled.
Condition B is verified with the help of the following lemma.

Lemma 3.1. *The function* $\boldsymbol{b}$ *defined above satisfies*

$$\boldsymbol{b} \in C^\infty; \qquad \frac{\|\boldsymbol{b}(\boldsymbol{x})\|}{\|\boldsymbol{x}\|} < C \quad \text{for } \|\boldsymbol{x}\| \to \infty$$

$$\|\nabla b_k\| \leq C; \qquad \left\| \frac{\partial^2 b_i}{\partial x_j \partial x_k} \right\| \leq C$$

and there exists $A \in \mathbb{R}$ *such that* (3.1.11) *is satisfied.*

Proof. We observe that the components of $\boldsymbol{b}$ have a linear term and terms which are first order polynomials times a decaying exponential. Thus $\boldsymbol{b} \in C^\infty$ for it is the sum of two C^∞ functions. Since the latter component remains bounded (for the exponential decays faster) we see that (3.1.11) is also satisfied. The higher order derivatives of $\boldsymbol{b}$ would all have terms involving a polynomial multiplied with a decaying exponential and as before since the exponential decays faster than a polynomial, we conclude that they are all bounded. ■

Since (a_{ij}) is an identity matrix, A in (3.1.10) can be chosen arbitrarily. By arguments provided in lemma 3.1, we can conclude that the derivatives of c remain bounded and that we can choose a A such that (3.1.12) is satisfied.

Having established the existence of solutions of the Cauchy problem we now consider uniqueness. The following theorem taken from [**7**] states

Theorem 3.1.2. *Let* M *satisfy the assumptions*

1. M *is uniformly parabolic in* $\Omega = \mathbb{R}^s \times [0, T]$;
2. *The functions*

$$a_{ij}, \quad \frac{\partial}{\partial x_h} a_{ij}, \quad \frac{\partial^2}{\partial x_h \partial x_k} a_{ij}; \quad b_i, \quad \frac{\partial}{\partial x_h} b_i; \quad c \tag{3.1.19}$$

are bounded functions on Ω*; they satisfy a uniform Hölder condition of exponent* α *in* $\boldsymbol{x} \in \mathbb{R}^s$*, uniformly with respect to* t *and* (3.1.10) *holds throughout* Ω*. Then there exists at most one solution to the Cauchy problem* (3.1.7) *with* (3.1.8) *satisfying the boundedness condition*

$$\int_0^T \int_{\mathbb{R}^s} |u(\boldsymbol{x}, t)| \exp[-k \,\|\, \boldsymbol{x} \,\|^2] d\boldsymbol{x} dt < \infty \tag{3.1.20}$$

for some positive number k*.*

Proof. The proof is as given in [**7**] with $\alpha = 1$. ■

For our case the assumptions are satisfied by arguments similar to those given after theorem 3.1.1. Theorem 3.1.1 assures the existence of a fundamental solution $\Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, \tau)$ and in view of theorems 3.1.1 and 3.1.2 we have the following.

Theorem 3.1.3. *Problem* (3.1.3) *admits a unique classical solution given by*

$$\psi(\boldsymbol{x}, t) = \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, 0)\psi_0(\boldsymbol{\xi})d\boldsymbol{\xi}. \tag{3.1.21}$$

Proof. Follows from (3.1.16), taking $f(\boldsymbol{x}, t) = 0$ and $\varphi(\boldsymbol{x}) = \psi_0(\boldsymbol{x})$. ■

3.2. The splitting method

It can be observed from the Fokker-Planck equation we derived in the last chapter that the equation has a convective part and a diffusive part. The convective part of the equation is very easy to handle, by moving the initial data along the characteristics, that is the integral curves of $\boldsymbol{b}$. The diffusive part is again quite easily solved, for the solution is the convolution of the Gauss kernel with the initial condition. Thus the two subproblems are easier to solve than the original problem.

Having this background, the main idea of the splitting method is to write (3.1.1) in the form

$$\begin{aligned} \partial_t\psi + \operatorname{div}(\boldsymbol{b}\psi) &= \Delta\psi, \qquad \text{in } \mathbb{R}^s \times (0, T) \\ \psi(\boldsymbol{y}, 0) &= \psi_0(\boldsymbol{y}), \qquad \text{in } \mathbb{R}^s \end{aligned} \tag{3.2.22}$$

and then to consider the two subproblems

$$\begin{aligned} \partial_t\psi + \operatorname{div}(\boldsymbol{b}\psi) &= 0, \\ \psi(\boldsymbol{y}, 0) &= \psi_0(\boldsymbol{y}) \end{aligned} \tag{3.2.23}$$

and

$$\begin{aligned} \partial_t\psi &= \Delta\psi \\ \psi(\boldsymbol{y}, 0) &= \psi_0(\boldsymbol{y}) \end{aligned} \tag{3.2.24}$$

separately.

If we introduce the operators $C = \operatorname{div}(\boldsymbol{b}\cdot)$ and $D = -\Delta$, then we can write (3.2.22) as

$$\partial_t\psi + C\psi + D\psi = 0, \qquad \psi|_{t=0} = \psi_0 \tag{3.2.25}$$

and the two subproblems as

$$\partial_t \psi + C\psi = 0, \qquad \psi|_{t=0} = \psi_0 \tag{3.2.26}$$

and

$$\partial_t \psi + D\psi = 0, \qquad \psi|_{t=0} = \psi_0 \tag{3.2.27}$$

We observe that $\boldsymbol{b}$ in our case is independent of t. Hence (3.2.25), (3.2.26) and (3.2.27) are autonomous systems. Denoting the solution operators by $\mathcal{S}_t$, $\mathcal{T}_t$ and $\mathcal{D}_t$ respectively, the approximate solution of (3.2.25) at time Δt is obtained by first solving (3.2.26) for a time Δt, feeding it in (3.2.27) and solving it in $[0, \Delta t]$. We then obtain

$$\mathcal{S}_{\Delta t}\psi_0 \approx \mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t}\psi_0$$

and recursively for the n^{th} time step

$$\mathcal{S}_{n\Delta t}\psi_0 \approx (\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^n \psi_0. \tag{3.2.28}$$

The approximate solution $\tilde{\psi}$ of (3.2.22) at time $\tau \in (n\Delta t, (n+1)\Delta t]$ is written as

$$\tilde{\psi}(\boldsymbol{y}, \tau) = \mathcal{D}_{\tau - n\Delta t}\mathcal{T}_{\tau - n\Delta t}(\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^n \psi_0(\boldsymbol{y}).$$

In the analysis to follow, we shall assume that $\tau, \Delta t < T$.

3.2.1. Consistency analysis. To formally assess the consistency of the splitting method, we write $\mathcal{T}_t = e^{-tC}$, $\mathcal{D}_t = e^{-tD}$ and $\mathcal{S}_t = e^{-t(C+D)}$. Expand the exponentials, we get

$$e^{-\Delta t(C+D)} = I - \Delta t(C+D) + \frac{1}{2}\Delta t^2(C^2 + DC + CD + D^2) + O(\Delta t^3),$$
$$e^{-\Delta tC} = I - \Delta tC + \frac{1}{2}\Delta t^2 C^2 + O(\Delta t^3),$$
$$e^{-\Delta tD} = I - \Delta tD + \frac{1}{2}\Delta t^2 D^2 + O(\Delta t^3).$$

Also,

$$e^{-\Delta tD}e^{-\Delta tC} = I - \Delta t(C+D) + \frac{1}{2}\Delta t^2(C^2 + 2CD + D^2) + O(\Delta t^3).$$

By introducing the commutator $[C, D] = DC - CD$, we have

$$e^{-\Delta t(C+D)} - e^{-\Delta tD}e^{-\Delta tC} = \frac{1}{2}\Delta t^2[C, D] + O(\Delta t^3).$$

Consequently the splitting method introduced by (3.2.28) is first order consistent unless we have $[C, D] = 0$. In our case,

$$[C, D]\psi = -\mathrm{div}(\boldsymbol{b}\Delta\psi) + \Delta(\mathrm{div}\boldsymbol{b}\psi) \neq 0$$

Having studied the consistency of the splitting method, we now analyze the convergence of the method, starting with the properties of the operators $\mathcal{D}_t$ and $\mathcal{T}_t$.

3.2.2. Properties of $\mathcal{D}_t$. It is well known that the solution of (3.2.24) can be written as the convolution of the initial condition ψ_0 with the Gaussian kernel

$$G_t(\boldsymbol{x}) = \frac{1}{(4\pi t)^{\frac{s}{2}}} \exp\left(-\frac{\|\boldsymbol{x}\|^2}{4t}\right). \tag{3.2.29}$$

Proposition 3.2.1. *The operator $\mathcal{D}_t$ is given by*

$$\mathcal{D}_t\psi = G_t * \psi. \tag{3.2.30}$$

We shall now estimate $\mathcal{D}_t\psi$ in C^2 norm which is defined as follows.

Definition 3.2.4. *Let Ω be a domain in $\mathbb{R}^s$. For $h \in C^2(\Omega)$,*

$$\|h\|_{C^2} = \max_{0\le|\alpha|\le 2} \sup_{\boldsymbol{x}\in\Omega} |D^\alpha h(\boldsymbol{x})|$$

Lemma 3.2. *For any function $\psi \in C^2(\mathbb{R}^s)$, we have*

$$\|\mathcal{D}_t\psi\|_{C^2} \le \|\psi\|_{C^2} \tag{3.2.31}$$

Proof. The result we shall refer to quite often is that the fundamental solution G_t given by (3.2.29) integrates to one, proved later in lemma 4.1. Now,

$$\begin{aligned} |\mathcal{D}_t\psi(\boldsymbol{x})| &= \left|\int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y})\psi(\boldsymbol{y})d\boldsymbol{y}\right| \\ &\le \int_{\mathbb{R}^s} |G_t(\boldsymbol{x}-\boldsymbol{y})|\,|\psi(\boldsymbol{y})|d\boldsymbol{y} \\ &\le \sup_{\boldsymbol{x}\in\mathbb{R}^s} |\psi(\boldsymbol{x})|\cdot 1 = \sup_{\boldsymbol{x}\in\mathbb{R}^s} |\psi(\boldsymbol{x})| \end{aligned}$$

using lemma 4.1.
Similarly, for $0 \le |\alpha| \le 2$, on exchanging differentiation and convolution, we get,

$$\begin{aligned} |D^\alpha\mathcal{D}_t\psi(\boldsymbol{x})| &= \left|D^\alpha\int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y})\psi(\boldsymbol{y})d\boldsymbol{y}\right| \\ &= \left|D^\alpha\int_{\mathbb{R}^s} G_t(\boldsymbol{y})\psi(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y}\right| \\ &\le \int_{\mathbb{R}^s} |G_t(\boldsymbol{y})|\,|D^\alpha\psi(\boldsymbol{x}-\boldsymbol{y})|d\boldsymbol{y} \\ &\le \sup_{\boldsymbol{x}\in\mathbb{R}^s} |D^\alpha\psi(\boldsymbol{x})|\cdot 1 = \sup_{\boldsymbol{x}\in\mathbb{R}^s} |D^\alpha\psi(\boldsymbol{x})| \end{aligned}$$

again using lemma 4.1.
Thus, combining the results we get, $\|\mathcal{D}_t\psi\|_{C^2} \leq \|\psi\|_{C^2}$. ■

We conclude this subsection stating a property of G_t we shall need later in our analysis.

Result 3.1. *If $P : \mathbb{R}^s \to \mathbb{R}$ is a homogeneous polynomial of degree r, that is $P(t\boldsymbol{x}) = t^r P(\boldsymbol{x})$ then,*

$$\int_{\mathbb{R}^s} |P(\boldsymbol{y})G_t(\boldsymbol{y})|\, d\boldsymbol{y} \leq C_P t^{\frac{r}{2}}, \quad t > 0 \tag{3.2.32}$$

where C_P is a constant depending on P.

3.2.3. Properties of $\mathcal{T}_t$. The characteristics of (3.2.23) are got by solving

$$\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x}) \qquad \boldsymbol{x}|_{t=\tau} = \boldsymbol{\xi} \tag{3.2.33}$$

We shall denote the solution of (3.2.33) by $\boldsymbol{X}(t;\boldsymbol{\xi},\tau)$. Since $\boldsymbol{b}$ is smooth and grows at most linearly at infinity (see lemma 3.1), $\boldsymbol{X}(t;\cdot,\tau)$ is a C^∞ diffeomorphism, [**3**]. With this notation we have the following proposition.

Proposition 3.2.2. *The operator $\mathcal{T}_t$ is given by*

$$\mathcal{T}_t\psi = \psi(\boldsymbol{X}(0;\boldsymbol{x},t))J(0;\boldsymbol{x},t), \tag{3.2.34}$$

where $J(t;\boldsymbol{x},\tau)$ is the Jacobian $\partial_{\boldsymbol{x}}\boldsymbol{X}(t;\boldsymbol{x},\tau)$ given by

$$J(t;\boldsymbol{x},\tau) = \exp\left(\int_\tau^t \operatorname{div} \boldsymbol{b}(\boldsymbol{X}(t';\boldsymbol{x},\tau))dt'\right).$$

The solution of (3.2.23) can now be written as $\mathcal{T}_t\psi_0$. For the sake of simplicity we denote $\mathcal{T}_t\psi_0$ by v. We now study the regularity of $\boldsymbol{X}$, J and v.

Remark 1. *In order to simplify the notation, we shall henceforth use a generic constant C which depends on the given field $\boldsymbol{b}$ and the length of the time interval $[0,T]$.*

Lemma 3.3. *The first and second order partial derivatives of $\boldsymbol{X}(t;\boldsymbol{\xi},\tau)$ can be estimated by*

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^s} \left|\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\boldsymbol{\xi},\tau)\right| \leq 1 + C(t-\tau), \tag{3.2.35}$$

and

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^s} \left|\frac{\partial}{\partial\xi_i}\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\boldsymbol{\xi},\tau)\right| \leq C(t-\tau). \tag{3.2.36}$$

Proof. Observe that $\boldsymbol{X}(t;\boldsymbol{\xi},\tau)$ satisfies (3.2.33). So

$$\frac{\partial}{\partial\xi_j}\frac{d}{dt}X_k(t;\boldsymbol{\xi},\tau)=\frac{\partial}{\partial\xi_j}b_k(\boldsymbol{X}(t;\boldsymbol{\xi},\tau))$$

Changing the order of differentiation, we get

$$\frac{d}{dt}\frac{\partial}{\partial\xi_j}X_k(t;\boldsymbol{\xi},\tau)=\sum_{r=1}^{s}\frac{\partial}{\partial x_r}b_k(\boldsymbol{X}(t;\boldsymbol{\xi},\tau))\frac{\partial}{\partial\xi_j}X_r(t;\boldsymbol{\xi},\tau) \tag{3.2.37}$$

with the initial condition,

$$\frac{\partial}{\partial\xi_j}X_k(t;\boldsymbol{\xi},\tau)\Big|_{t=\tau}=\delta_{jk}$$

Introducing the matrix,

$$U(t)=\nabla\boldsymbol{b}(\boldsymbol{X}(t;\boldsymbol{\xi},\tau))$$

and writing (3.2.37) for $k=1,\ldots,s$, we get,

$$\frac{d}{dt}\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\boldsymbol{\xi},\tau)=U(t)\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\boldsymbol{\xi},\tau)$$

along with

$$\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\boldsymbol{\xi},\tau)\Big|_{t=\tau}=\boldsymbol{e}_j$$

Rewriting we get,

$$\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\boldsymbol{\xi},\tau)=\boldsymbol{e}_j+\int_\tau^t U(t')\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t';\boldsymbol{\xi},\tau)dt'.$$

Thus,

$$\left\|\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\cdot,\tau)\right\|_\infty\leq 1+\int_\tau^t\|U(t')\|\left\|\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t';\cdot,\tau)\right\|_\infty dt'. \tag{3.2.38}$$

Hence using Gronwall's lemma and lemma 3.1 we conclude that,

$$\left\|\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\cdot,\tau)\right\|_\infty\leq\exp\left(\int_\tau^t\|U(t')\|dt'\right)\leq\exp(C(t-\tau))\leq\exp(CT). \tag{3.2.39}$$

Now substituting (3.2.39) in (3.2.38), we get (3.2.35) using the boundedness of U.

The second estimate follows by a similar argument. For a fixed i and j, differentiating (3.2.37) once more, yields the following ode for $\mathbf{H}:=\frac{\partial}{\partial\xi_i}\frac{\partial}{\partial\xi_j}\boldsymbol{X}(t;\boldsymbol{\xi},\tau)$

$$\frac{d\mathbf{H}}{dt}=\mathbf{F}+U\mathbf{H}, \tag{3.2.40}$$

where,

$$\mathbf{F}(t) = \sum_{r,m=1}^{s} \frac{\partial}{\partial x_r}\frac{\partial}{\partial x_m}\boldsymbol{b}\,\frac{\partial X_m}{\partial \xi_i}\frac{\partial X_r}{\partial \xi_j},$$

with zero initial condition. Observe that from lemma 3.1 and (3.2.39) that $\|\mathbf{F}\|_\infty \leq C$. The integral representation of the ode (3.2.40) is,

$$\mathbf{H} = \int_\tau^t (\mathbf{F}(t') + U(t')\mathbf{H}(t'))dt'.$$

So it follows that,

$$\|\mathbf{H}(t)\|_\infty \leq \int_\tau^t \|\mathbf{F}(t')\|_\infty dt' + \int_\tau^t \|U(t')\|_\infty \|\mathbf{H}(t')\|_\infty dt',$$

where the $\mathbb{L}_\infty$ norms are taken with respect to the space variable $\boldsymbol{\xi}$. Using the boundedness of $\mathbf{F}$ and U and applying Gronwall's lemma yields,

$$\|\mathbf{H}(t)\|_\infty \;\leq\; C(t-\tau)\exp(C(t-\tau)) \leq C(t-\tau)\exp(CT)$$

and that completes the proof of the lemma. ■

Lemma 3.4. *The $\mathbb{L}^\infty$ norm of J satisfies*

$$\|J(t;\cdot,\tau)\|_\infty \leq 1 + C(t-\tau) \tag{3.2.41}$$

where C is a generic constant.

Proof. We have $J(t;\boldsymbol{\xi},\tau) = \exp\left(\int_\tau^t \operatorname{div}\boldsymbol{b}(\boldsymbol{X}(t';\boldsymbol{\xi},\tau))dt'\right)$ and so

$$\|J(t;\cdot,\tau)\|_\infty \;\leq\; \exp\left(\|\operatorname{div}\boldsymbol{b}\|_\infty (t-\tau)\right)$$

on using lemma 3.1. Since $\|\operatorname{div}\boldsymbol{b}\|_\infty$ can be calculated apriorily as $\boldsymbol{b}$ is known, we can find a C such that (3.2.41) is satisfied for $t-\tau < T$. ■

Lemma 3.5. *The first and second order partial derivatives of J are uniformly bounded in $[0,\Delta t]\times\mathbb{R}^s\times[0,\Delta t]$*

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^s}\left|\frac{\partial}{\partial\xi_j}J(t;\boldsymbol{\xi},\tau)\right| \leq C(t-\tau),$$

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^s}\left|\frac{\partial}{\partial\xi_i}\frac{\partial}{\partial\xi_j}J(t;\boldsymbol{\xi},\tau)\right| \leq C(t-\tau).$$

Proof. We have $J(t;\boldsymbol{\xi},\tau) = \exp\left(\int_\tau^t \operatorname{div}\boldsymbol{b}(\boldsymbol{X}(t';\boldsymbol{\xi},\tau))dt'\right)$ and so

$$\frac{\partial}{\partial\xi_j}J(t;\boldsymbol{\xi},\tau) = \exp\left(\int_\tau^t (\operatorname{div}\boldsymbol{b}(\boldsymbol{X}(t';\boldsymbol{\xi},\tau))dt'\right)\left(\int_\tau^t (\nabla\operatorname{div}\boldsymbol{b})\cdot(\nabla\boldsymbol{X})_j dt'\right)$$

where the subscript j in $(\nabla \boldsymbol{X})_j$ denotes the j^{th} column of the matrix $\nabla \boldsymbol{X}$. Using lemma 3.1 and (3.2.39) yields,

$$\sup_{\boldsymbol{\xi}\in\mathbb{R}^s} \left| \frac{\partial}{\partial \xi_j} J(t;\boldsymbol{\xi},\tau) \right| \leq \exp\left(C(t-\tau)\right) C(t-\tau) \leq \exp(CT) C(t-\tau)$$

and the result follows. Differentiating once more yields,

$$\begin{aligned} \frac{\partial}{\partial \xi_i}\frac{\partial}{\partial \xi_j} J(t;\boldsymbol{\xi},\tau) = & J\left(\int_\tau^t (\nabla \mathrm{div}\boldsymbol{b})\cdot(\nabla \boldsymbol{X})_j dt'\right)\left(\int_\tau^t (\nabla \mathrm{div}\boldsymbol{b})\cdot(\nabla \boldsymbol{X})_i dt'\right) \\ & + J\left(\int_\tau^t \left(\frac{\partial}{\partial \xi_i}(\nabla \mathrm{div}\boldsymbol{b})\cdot(\nabla \boldsymbol{X})_j + (\nabla \mathrm{div}\boldsymbol{b})\cdot\frac{\partial}{\partial \xi_i}(\nabla \boldsymbol{X})_j\right) dt'\right) \end{aligned}$$

Using lemmas 3.1, 3.3, 3.4, 3.5 and (3.2.39), we get

$$\begin{aligned} \left\| \frac{\partial}{\partial \xi_i}\frac{\partial}{\partial \xi_j} J(t;\cdot,\tau) \right\|_\infty &\leq (1+C(t-\tau))\left((C(t-\tau))^2 + C(t-\tau) + C(t-\tau)^2\right) \\ &\leq (1+C(t-\tau))(C(t-\tau)T + C(t-\tau) + C(t-\tau)T) \end{aligned}$$

Recombining generic constants, we find

$$\left\| \frac{\partial}{\partial \xi_i}\frac{\partial}{\partial \xi_j} J(t;\cdot,\tau) \right\|_\infty \leq (1+CT)(2CT+C)(t-\tau)$$

and the lemma is proved. ■

In view of lemmas 3.4 and 3.5 we have proved the following.

Result 3.2. *The C^2 norm of J can be estimated by*

$$\|J(t;\cdot,\tau)\|_{C^2} \leq \exp(C(t-\tau))$$

We now turn on to study the regularity of $v = \mathcal{T}_t\psi_0$.

Lemma 3.6. *If $\psi_0 : \mathbb{R}^s \to \mathbb{R}$ is bounded and has bounded first and second order derivatives, then for $t \leq \Delta t$*

$$\|v(\cdot,t)\|_\infty \leq \|\psi_0\|_{C^2}\|J(0;\cdot,t)\|_{C^2} \tag{3.2.42}$$

$$\sup_{\boldsymbol{x}\in\mathbb{R}^s} \left| \frac{\partial}{\partial x_j} v(\boldsymbol{x},t) \right| \leq \|\psi_0\|_{C^2} \exp\left(Ct\right) \tag{3.2.43}$$

$$\sup_{\boldsymbol{x}\in\mathbb{R}^s} \left| \frac{\partial}{\partial x_k}\frac{\partial}{\partial x_j} v(\boldsymbol{x},t) \right| \leq \|\psi_0\|_{C^2} \exp\left(Ct\right). \tag{3.2.44}$$

Proof. The quantity v is explicitly given by

$$v(\boldsymbol{\xi},t) = \psi_0(\boldsymbol{X}(0;\boldsymbol{\xi},t)) J(0;\boldsymbol{\xi},t)$$

Thus,

$$\|v(\cdot,t)\|_\infty \leq \|\psi_0\|_\infty \|J(0;\cdot,t)\|_\infty \leq \|\psi_0\|_{C^2}\|J(0;\cdot,t)\|_{C^2}$$

and (3.2.42) is proved.

Again from the solution formula for v,

$$\frac{\partial}{\partial \xi_j} v(\boldsymbol{\xi},t) = \sum_{k=1}^{s} \frac{\partial \psi_0}{\partial x_k}\frac{\partial X_k}{\partial \xi_j} J(0;\boldsymbol{\xi},t) + \psi_0(\boldsymbol{X}(0;\boldsymbol{\xi},t))\frac{\partial}{\partial \xi_j} J(0;\boldsymbol{\xi},t)$$

Thus, using lemmas 3.3, 3.4 and 3.5 it follows that,

$$\begin{aligned}\left\|\frac{\partial}{\partial \xi_j} v(\cdot,t)\right\|_\infty &\leq \langle \nabla\psi_0, \nabla \boldsymbol{X}\rangle \|J(0;\cdot,t)\|_\infty + \|\psi_0\|_\infty \left\|\frac{\partial}{\partial \xi_j} J(0;\cdot,t)\right\|_\infty \\ &\leq \|\psi_0\|_{C^2}((1+Ct)(1+Ct)+Ct) \\ &\leq \|\psi_0\|_{C^2}(1+(3C+C^2T)t)\end{aligned}$$

Recombining generic constants, we get

$$\left\|\frac{\partial}{\partial \xi_j} v(\cdot,t)\right\|_\infty \leq \|\psi_0\|_{C^2}(1+Ct) \leq \|\psi_0\|_{C^2}\exp(Ct)$$

The estimate for the second order partial derivatives can be verified in a similar manner. ■

Finally we have the following result for v.

Result 3.3. *If $\psi \in C^2$, then for $t \leq \Delta t$*

$$\|\mathcal{T}_t\psi\|_{C^2} \leq \exp(Ct)\|\psi\|_{C^2} \tag{3.2.45}$$

Proof. Follows from lemmas 3.3, 3.4, 3.5 and 3.6. ■

3.3. Convergence of the splitting method

Recall that if $\mathcal{T}_t$ and $\mathcal{D}_t$ denote the solution operators of (3.2.23) and (3.2.24) at time t respectively, then the approximate solution $\tilde{\psi}$ of (3.2.22) at time $\tau \in (n\Delta t, (n+1)\Delta t]$ can be written as

$$\tilde{\psi}(\boldsymbol{y},\tau) = \mathcal{D}_{\tau-n\Delta t}\mathcal{T}_{\tau-n\Delta t}(\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^n\psi_0(\boldsymbol{y}).$$

The bound for the values of $\tilde{\psi}$ at intermediate time steps is given by the following lemma.

Lemma 3.7. *If $\psi_0 \in C^2$ then,*

$$\|\tilde{\psi}(\cdot,t)\|_{C^2} \leq \exp(CT)\|\psi_0\|_{C^2}, \quad 0 \leq t \leq T \tag{3.3.46}$$

Proof. We observe that for a given $t \in [0,T]$, there exists an n such that $t \in (n\Delta t, (n+1)\Delta t]$ and

$$\tilde{\psi}(\boldsymbol{y},t) = \mathcal{D}_{t-n\Delta t}\mathcal{T}_{t-n\Delta t}(\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^n\psi_0(\boldsymbol{y}).$$

Repeated use of (3.2.31) and (3.2.45) yields,

$$\begin{aligned}
\|\tilde{\psi}(\cdot,t)\|_{C^2} &= \|\mathcal{D}_{t-n\Delta t}\left(\mathcal{T}_{t-n\Delta t}(\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^n\psi_0\right)\|_{C^2} \leq \|\mathcal{T}_{t-n\Delta t}\left((\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^n\psi_0\right)\|_{C^2} \\
&\leq \exp(C(t-n\Delta t))\ \|\mathcal{D}_{\Delta t}\left(T_{\Delta t}(\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^{n-1}\psi_0\right)\|_{C^2} \\
&\leq \exp(C(t-n\Delta t))\ \|T_{\Delta t}\left((\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^{n-1}\psi_0\right)\|_{C^2} \\
&\leq \exp(C(t-(n-1)\Delta t))\ \|(\mathcal{D}_{\Delta t}\mathcal{T}_{\Delta t})^{n-1}\psi_0\|_{C^2} \\
&\vdots \\
&\leq \exp(Ct)\ \|\psi_0\|_{C^2} \leq \exp(CT)\ \|\psi_0\|_{C^2}
\end{aligned}$$

and the lemma is proved. ■

Let $\tau = n\Delta t + t$, $t \in (0,\Delta t]$. If $\tilde{\psi}_n$ denotes the approximate solution after n time steps, then an equation for $\tilde{\psi}_n$ similar to (3.2.22) can be derived as,

$$\frac{\partial\tilde{\psi}_n}{\partial t} = \left(\frac{\partial}{\partial t}\mathcal{D}_t\right)(\mathcal{T}_t\tilde{\psi}_n) + \mathcal{D}_t\left(\frac{\partial}{\partial t}(\mathcal{T}_t\tilde{\psi}_n)\right) = \Delta\tilde{\psi}_n - \mathcal{D}_t(\mathrm{div}\,(\boldsymbol{b}\mathcal{T}_t\tilde{\psi}_n)) \tag{3.3.47}$$

Hence,

$$\frac{\partial\tilde{\psi}_n}{\partial t} + \mathrm{div}\,(\boldsymbol{b}\tilde{\psi}_n) - \Delta\tilde{\psi}_n = \mathrm{div}\,(\boldsymbol{b}\tilde{\psi}_n) - \mathcal{D}_t(\mathrm{div}\,(\boldsymbol{b}\mathcal{T}_t\tilde{\psi}_n)) \tag{3.3.48}$$

If $e_n = \psi(\boldsymbol{x},n\Delta t) - \tilde{\psi}_n$, is the error after n time steps, then the equation for e can be written down by subtracting (3.3.47) from (3.2.22) as,

$$\frac{\partial e_n}{\partial t} + \mathrm{div}\,(\boldsymbol{b}e_n) - \Delta e_n = \mathcal{D}_t(\mathrm{div}\,(\boldsymbol{b}\mathcal{T}_t\tilde{\psi}_n)) - \mathrm{div}\,(\boldsymbol{b}\mathcal{D}_t\mathcal{T}_t\tilde{\psi}_n) = g \tag{3.3.49}$$

We shall now find an upper bound for the $\mathbb{L}^\infty$ norm of g. For ease of notation, as before, we set $v = \mathcal{T}_t\tilde{\psi}_n$.

Lemma 3.8. *There exists a constant $C > 0$ such that*

$$\|g(\cdot,t)\|_\infty \leq C\Delta t, \quad 0 \leq t \leq T \tag{3.3.50}$$

The proof of the lemma is quite lengthy and we split it into several steps.

Observe that

$$\begin{aligned}
\|g\|_\infty &= \|\mathcal{D}_t(\operatorname{div}(\boldsymbol{b}\, v)) - \operatorname{div}(\boldsymbol{b}\mathcal{D}_t v)\|_\infty \\
&= \|\mathcal{D}_t((\operatorname{div}\boldsymbol{b})v) + \mathcal{D}_t(\boldsymbol{b}\cdot\nabla v) - (\operatorname{div}\boldsymbol{b})\mathcal{D}_t v - \boldsymbol{b}\cdot\nabla(\mathcal{D}_t v)\|_\infty \\
&\leq \|\mathcal{D}_t((\operatorname{div}\boldsymbol{b})v) - (\operatorname{div}\boldsymbol{b})\mathcal{D}_t v\|_\infty + \|\mathcal{D}_t(\boldsymbol{b}\cdot\nabla v) - \boldsymbol{b}\cdot\nabla(\mathcal{D}_t v)\|_\infty \\
&= E_1 + E_2
\end{aligned}$$

We estimate E_1 and E_2 separately.

Lemma 3.9. *E_1 and E_2 satisfy the bounds,*

$$\begin{aligned}
E_1 &= \|\mathcal{D}_t((\operatorname{div}\boldsymbol{b})v) - (\operatorname{div}\boldsymbol{b})\mathcal{D}_t v\|_\infty \leq Ct \\
E_2 &= \|\mathcal{D}_t(\boldsymbol{b}\cdot\nabla v) - \boldsymbol{b}\cdot\nabla(\mathcal{D}_t v)\|_\infty \leq Ct
\end{aligned}$$

Proof. We start with E_1.

$$\begin{aligned}
E_1 &= \|\mathcal{D}_t((\operatorname{div}\boldsymbol{b})v) - (\operatorname{div}\boldsymbol{b})\mathcal{D}_t v\|_\infty \\
&= \sup_{\boldsymbol{x}\in\mathbb{R}^s}\left|\int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y})v(\boldsymbol{y})\{\operatorname{div}\boldsymbol{b}(\boldsymbol{y}) - \operatorname{div}\boldsymbol{b}(\boldsymbol{x})\}d\boldsymbol{y}\right|
\end{aligned}$$

By the regularity of v and $\operatorname{div}\boldsymbol{b}(\boldsymbol{y})$ established earlier, both of them can be expanded in Taylor series. Thus we get,

$$v(\boldsymbol{y}) = v(\boldsymbol{x}) + (\boldsymbol{y}-\boldsymbol{x})\cdot\nabla v(\boldsymbol{\xi}_1)$$

$$\operatorname{div}\boldsymbol{b}(\boldsymbol{y}) = \operatorname{div}\boldsymbol{b}(\boldsymbol{x}) + (\boldsymbol{y}-\boldsymbol{x})\cdot\nabla\operatorname{div}\boldsymbol{b}(\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})\mathbf{A}_2(\boldsymbol{\xi}_2)(\boldsymbol{y}-\boldsymbol{x})^T$$

where $\boldsymbol{\xi}_1 \in (\boldsymbol{x},\boldsymbol{y})$ and $\mathbf{A}_2$ is the matrix

$$\left(\frac{\partial^2(\operatorname{div}\boldsymbol{b})}{\partial x_i \partial x_j}\right)_{ij}$$

evaluated at some intermediate point $\boldsymbol{\xi}_2 \in (\boldsymbol{x},\boldsymbol{y})$ respectively.

Inserting the expansions, performing term by term integration and using triangle inequality we get four terms E_{11}, E_{12}, E_{13}, E_{14}. We now estimate each term separately, using result 3.1.

$$E_{11} = \left\|\int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y})v(\boldsymbol{x})((\boldsymbol{y}-\boldsymbol{x})\cdot\nabla\operatorname{div}\boldsymbol{b}(\boldsymbol{x}))d\boldsymbol{y}\right\|_\infty = 0$$

using the even odd properties of the integrand.

$$\begin{aligned} E_{12} &= \left\| \frac{1}{2} \int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y}) v(\boldsymbol{x}) (\boldsymbol{y}-\boldsymbol{x}) A_2(\boldsymbol{\xi}_2) (\boldsymbol{y}-\boldsymbol{x})^T d\boldsymbol{y} \right\|_\infty \\ &\leq \frac{1}{2} \|A_2\|_\infty \|v\|_\infty \int_{\mathbb{R}^s} \left\| G_t(\boldsymbol{x}-\boldsymbol{y}) \, (\boldsymbol{y}-\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})^T \right\|_\infty d\boldsymbol{y} \\ &\leq Ct \end{aligned}$$

using lemmas 3.1 and 3.4 and the boundedness of ψ_0.

$$\begin{aligned} E_{13} &= \left\| \int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y}) ((\boldsymbol{y}-\boldsymbol{x}) \cdot \nabla v(\boldsymbol{\xi}_1)) ((\boldsymbol{y}-\boldsymbol{x}) \cdot \operatorname{div} \boldsymbol{b}(\boldsymbol{x})) d\boldsymbol{y} \right\|_\infty \\ &= 2t \|\nabla v \cdot \nabla \boldsymbol{b}\|_\infty \leq Ct \end{aligned}$$

again on using lemmas 3.1 and 3.6.

$$\begin{aligned} E_{14} &= \left\| \frac{1}{2} \int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y}) ((\boldsymbol{y}-\boldsymbol{x}) \cdot \nabla v(\boldsymbol{\xi}_1)) ((\boldsymbol{y}-\boldsymbol{x}) A_2(\boldsymbol{\xi}_2) (\boldsymbol{y}-\boldsymbol{x})^T) d\boldsymbol{y} \right\|_\infty \\ &\leq Ct\sqrt{t} \leq Ct\sqrt{T} \end{aligned}$$

on using lemmas 3.1 and 3.6.

Thus, $E_1 \leq E_{11} + E_{12} + E_{13} + E_{14} \leq Ct$. The estimate for E_2 follows by a similar argument. ■

We shall now show the continuity of g with respect to t in each of the subintervals. Recall that,

$$\begin{aligned} g(\boldsymbol{x},t) &= \mathcal{D}_t(\operatorname{div}(\boldsymbol{b}\mathcal{T}_t\psi_0)) - \operatorname{div}(\boldsymbol{b}\mathcal{D}_t\mathcal{T}_t\psi_0) \\ &= \mathcal{D}_t(\operatorname{div}(\boldsymbol{b}\mathcal{T}_t\psi_0)) - \mathcal{D}_t\mathcal{T}_t\psi_0 \operatorname{div}\boldsymbol{b} - \boldsymbol{b} \cdot \nabla \mathcal{D}_t\mathcal{T}_t\psi_0 \\ &:= g_1 - g_2 - g_3 \end{aligned}$$

The required continuity of g is essentially contained in the following lemma.

Lemma 3.10. *If f is a continuous function and satisfies $\|f(\cdot,t)\|_\infty \leq C$ for $0 \leq t \leq T$, then the function F defined by*

$$F(\boldsymbol{x},t) = \int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y}) f(\boldsymbol{y},t) d\boldsymbol{y} \tag{3.3.51}$$

is continuous with respect to t and moreover

$$\lim_{t \to 0} F(\boldsymbol{x},t) = f(\boldsymbol{x},0) \tag{3.3.52}$$

Proof. By the substitution $\boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{y})/\sqrt{4t}$, (3.3.51) reduces to

$$F(\boldsymbol{x}, t) = \int_{\mathbb{R}^s} \frac{1}{\pi^{s/2}} \exp(-\boldsymbol{z}^2) f(\boldsymbol{x} - \boldsymbol{z}\sqrt{4t}, t) d\boldsymbol{z} \tag{3.3.53}$$

Since f is assumed to be bounded, we have

$$\left\| \frac{1}{\pi^{s/2}} \exp(-\boldsymbol{z}^2) f(\cdot, t) \right\|_\infty \leq \frac{1}{\pi^{s/2}} \exp(-\boldsymbol{z}^2) C := h(\boldsymbol{z}) \tag{3.3.54}$$

Since h is integrable, we conclude from result 25(a) of appendix in [**31**] that F is continuous with respect to t.

On taking the limits as $t \to 0$ in (3.3.53) we get,

$$\lim_{t \to 0} F(\boldsymbol{x}, t) = \int_{\mathbb{R}^s} \frac{1}{\pi^{s/2}} \exp(-\boldsymbol{z}^2) f(\boldsymbol{x}, 0) d\boldsymbol{z} = f(\boldsymbol{x}, 0) \tag{3.3.55}$$

and the lemma is proved. ■

By the smoothness of the involved function, their boundedness (established earlier) and lemma 3.10 the continuity of g_1, g_2 and g_3 is established.

Having seen that g is continuous in each of the time intervals , we now estimate the error at the end of each time step. Applying theorem 3.1.1 to (3.3.49) yields,

(3.3.56)

$$e_n(\boldsymbol{x}) = \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, 0) e_{n-1}(\boldsymbol{\xi}) d\boldsymbol{\xi} - \int_0^{\Delta t} \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, \Delta t, \boldsymbol{\xi}, \tau) g(\boldsymbol{\xi}, \tau) d\boldsymbol{\xi} d\tau.$$

The fundamental solution $\Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, \tau)$ can be bounded by (refer (6.12) of [**7**]),

$$|\Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, \tau)| \leq \text{const.}(t - \tau)^{-s/2} \exp\left[-\frac{\lambda_0^* \|\boldsymbol{x} - \boldsymbol{\xi}\|^2}{4(t - \tau)} \right], \quad \lambda_0^* < \lambda_0$$

and so the second term in (3.3.56) can be estimated by using (3.3.50) as

$$\left| \int_0^{\Delta t} \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, \Delta t, \boldsymbol{\xi}, \tau) g(\boldsymbol{\xi}, \tau) d\boldsymbol{\xi} d\tau \right| \leq C \Delta t \, \Delta t = C(\Delta t)^2. \tag{3.3.57}$$

In the first time step, we have $e_0 = \psi_0 - \tilde{\psi}_0 = 0$ and so

$$\|e_1\|_\infty \leq \left| \int_0^{\Delta t} \int_{\mathbb{R}^s} \Gamma(\cdot, \Delta t, \boldsymbol{\xi}, \tau) g(\boldsymbol{\xi}, \tau) d\boldsymbol{\xi} d\tau \right| \leq C(\Delta t)^2 \tag{3.3.58}$$

In the second time step, again from (3.3.56),

$$\begin{aligned} |e_2(\boldsymbol{x})| &\leq \left| \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, t, \boldsymbol{\xi}, 0) e_1(\boldsymbol{\xi}) d\boldsymbol{\xi} \right| + \left| \int_0^{\Delta t} \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, \Delta t, \boldsymbol{\xi}, \tau) g(\boldsymbol{\xi}, \tau) d\boldsymbol{\xi} d\tau \right| \\ &\leq \|e_1\|_\infty + \left| \int_0^{\Delta t} \int_{\mathbb{R}^s} \Gamma(\boldsymbol{x}, \Delta t, \boldsymbol{\xi}, \tau) g(\boldsymbol{\xi}, \tau) d\boldsymbol{\xi} d\tau \right| \end{aligned}$$

On using (3.3.58) we have

$$\|e_2\|_\infty \leq \|e_1\|_\infty + C(\Delta t)^2 = 2C(\Delta t)^2$$

Continuing in the same way, we get

$$\|e_n\|_\infty \leq nC(\Delta t)^2 \leq \frac{T}{\Delta t} C(\Delta t)^2 = CT\Delta t$$

which shows that the splitting scheme is first order accurate.

In the next chapter, we will present a particle method for the Fokker-Planck equation we had derived earlier, which uses the splitting idea presented here. We shall later see that in order to present a complete convergence proof of that method, we would require the estimates in the $\mathbb{L}^1$ norm rather than the C^2 norm considered here. It may be noted that the C^2 estimates do not generalize to estimates in $\mathbb{L}^1$ because of the unbounded domain. The necessary $\mathbb{L}^1$ estimates can be obtained in a similar manner but we shall not pursue the same.

CHAPTER 4

Particle Methods

Most of the partial differential equations modeling various physical processes cannot be solved analytically. It is hence important to have good numerical schemes which are computationally efficient. In connection with our problem, the numerical scheme should not only be applicable in high dimensions but also cost effective. In general numerical methods which are based on discretization do not prove to be efficient in high dimensions and we illustrate this point in the following section.

4.1. Why not traditional methods?

We are interested in the numerical solution of a partial differential equation in high dimensions. For a certain class of partial differential equations this is the same as integrating a function in a high dimensional space. Consider, for example, the diffusion equation

$$\begin{aligned} u_t &= \Delta u \qquad \text{in } \Omega = \mathbb{R}^s \times [0,T] \\ u(\boldsymbol{x},0) &= u_0(\boldsymbol{x}) \qquad \text{in } \mathbb{R}^s. \end{aligned} \tag{4.1.1}$$

It is well known that the solution of (4.1.1) can be written as the convolution of the Gaussian kernel $G_t(\boldsymbol{x})$, given by (3.2.29) with the initial condition $u_0(\boldsymbol{x})$. Thus,

$$u(\boldsymbol{x},t) = \int_{\mathbb{R}^s} G_t(\boldsymbol{x}-\boldsymbol{y}) u_0(\boldsymbol{y}) d\boldsymbol{y}.$$

That is to say that we have an underlying integral which must be evaluated in high dimensions. So we start our discussion with the problem of numerical integration in dimension s.

For the one dimensional $s = 1$, case there are a number of conventional integration rules such as the trapezoidal rule and Simpson's rule [**11**]. These formulas are of the interpolatory type; that is to say

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} w_i f(x_i) \tag{4.1.2}$$

where

$$w_i = \int_a^b p_i(x)dx$$

and p_i are the Lagrange interpolation polynomials

$$p_i = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

The w_is and x_is, $i = 0, 1, \ldots, n$ are called the weights and nodes of the quadrature formula (4.1.2).

In the case of trapezoidal rule for the interval [a,b] with uniform spacing $h = (b-a)/n$, the approximation reads,

$$\int_a^b f(u)du \approx h \sum_{i=0}^{n}{}'' f(a+ih) \tag{4.1.3}$$

where the double prime denotes that the first term and the last term in the sum are halved. Consistent with (4.1.2), the weights of the above formula are given by $w_0 = w_n = h/2$ and $w_i = h$ for $1 \leq i \leq n-1$. This rule is exact for all functions $f \in \mathcal{P}_1$, that is for all polynomials of degree at most one. Moreover the error term is

$$-\frac{1}{12}(b-a)h^2 f''(\xi)$$

where $\xi \in (a, b)$, provided f'' exists.

In the multidimensional case $s \geq 2$ with an interval $[a, b]^s = I^s$ as integration domain, the classical numerical integration methods use tensor product of one dimensional integration rules. In such multi-dimensional quadrature rules, the node set is the Cartesian product of one-dimensional node sets and the weights are appropriate products of weights from the one-dimensional rules.

The total number of nodes used in the case of multidimensional integrals is then $N = (n+1)^s$, being $(n+1)$ in each dimension. From the error bound for (4.1.2), it follows that the error is $O(n^{-2})$, provided that $\partial^2 f/\partial^2 u_i^2$ is continuous on I^s for $1 \leq i \leq$ s. In terms of the number N of nodes, the error is $O(N^{-2/s})$. With increasing dimension s, the usefulness of the error bound declines drastically. To be more precise, to guarantee an error which is in absolute value $\leq 10^{-2}$ one needs to use roughly 10^s nodes; hence, the required number of nodes increases exponentially with d. This phenomenon is often called the "curse of dimensionality" and has been studied by Novak [**20**].

We next turn to the memory requirements to store the grid points. If we consider l points per direction then the total number of points is l^s and the total memory need to store these values is proportional to l^s. Since s appears in the exponent, enormous memory is required to carry on the computation for large s. So the method of Cartesian product does not prove to be computationally economical in high dimensions due to the huge node set and enormous memory requirements.

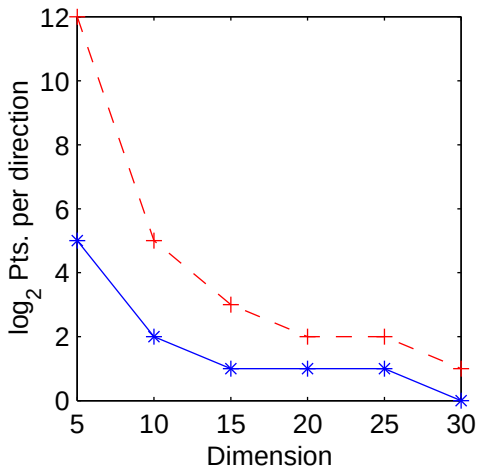


FIGURE 4.1. Comparison between finite difference method (thick line) and sparse grids (dashed line), assuming available memory is 1GB.

For a special class of functions Smolyak's construction [**27**] has been effective in tackling the problem. Though this method is again based on the tensor product structure, we do not consider the entire grid but only a few selected points. Compared to the traditional tensor product rule, this method uses only $n(\log_2 n)^{s-1}$ points, being $n = 2^l$ in each direction, instead of n^s points. Though this method is promising in low dimensions, it does not apply to high dimensions as can be seen from figure 4.1.

4.2. The Monte Carlo method

It is important to have good quadrature formulas for evaluating definite integrals occurring in many physical problems numerically. It is evident from the above discussion that any method which is based on grids will not generalize to high dimensions. This motivates us to go in for particle methods. Some

interesting facts in the case of multivariate integration is that the Monte Carlo method

1. does not require the integrand to be regular.
2. assures that the order of magnitude of the integration errors in terms of the number of nodes is independent of the dimension.

These make the method readily applicable to high dimensional integration. We shall now explain the method in detail.

The deterministic quadrature rules discussed in the previous section require the integrand to be regular. But if the integrand fails to be regular, the rules become less attractive. In such cases, it is convenient and simple to apply a Monte Carlo method. Though this method may be less accurate, the implementation is quite straightforward. The basic idea behind a Monte Carlo method is the representation of an integral as a *sample mean* unlike the traditional methods which interpret it as an area. This suggests that the method is strongly based on sampling to calculate the mean. Hence, the crucial task in the application of Monte Carlo method is the generation of random samples.

Suppose that random variables $x_1, x_2, \ldots, x_n, \ldots$ are all drawn from a probability distribution $f(x)$. A function G may be defined by

$$G = \sum_{n=1}^{N} \frac{1}{N} g(x_n) \tag{4.2.4}$$

where g is a given function. Now g being a function of a random variable x_n is itself a random variable and also since the sum of random variables is again a random variable, one can conclude that G is a random variable. The expected value of G can be defined to be

$$E(G) = E\left(\sum_{n=1}^{N} \frac{1}{N} g(x_n)\right) = \sum_{n=1}^{N} \frac{1}{N} E(g(x_n)) = E(g(x)) \tag{4.2.5}$$

since expectation value is a linear operation. In other words G and g have the same mean. More generally, G is an estimator of a quantity like $\int g(x)f(x)dx$.

If the x_is are chosen independent of each other then one can write

$$\mathrm{var}(G) = E(G^2) - (E(G))^2 \tag{4.2.6}$$

Now the variance of G in (4.2.4) becomes

$$\text{(4.2.7)} \qquad \text{var}(G) = \text{var}\left(\frac{1}{N}\sum_{n=1}^{N} g(x_n)\right) = \sum \frac{1}{N^2}\text{var}(g(x)) = \frac{1}{N}\text{var}(g(x)).$$

From the above one can conclude that as the number of samples increases, the variance of the mean value of G decreases like $1/N$. This forms the basis of the Monte Carlo method; that is that an integral may be estimated by a sum of the form (4.2.5). To sum up the method, to evaluate the integral $\int f(x)g(x)dx$, draw a series of random variables x_n from $f(x)$ and calculate $g(x)$ for each x_n. The arithmetic mean of all the values of g is an estimate of the integral, wherein the variance of the estimate reduces as the number of samples increases.

We shall now discuss the accuracy and convergence properties of the Monte Carlo method. The most general result we need to accomplish is the *strong law of large numbers.*

Theorem 4.2.1 (Strong law of large numbers)**.** *Let X be a random variable on $(A, \mathcal{A}, \lambda)$ and let f be the density of X. If $x_1, x_2, \ldots, x_N$ are all drawn independent of each other from the same distribution f, so that the expectation of each is μ, then as $N \to \infty$, the average value of the x's*

$$\bar{x}_N = \frac{1}{N}\sum_{i=1}^{N} x_i$$

converges to μ almost surely, in the sense

$$P\{\lim_{N\to\infty} \bar{x}_N = \mu\} = 1.$$

The above result shows that the mean of n sampled variables converges in probability to its expected value. In order to estimate the speed of convergence, we need a stronger assumption on the existence of the variance. In this case we have from the Chebychev's inequality

$$\text{(4.2.8)} \qquad P\left\{|G - E(G)| \geq \left[\frac{\text{var}(G)}{\epsilon}\right]^{1/2}\right\} \leq \epsilon$$

where ϵ is a positive number. This inequality could be called the first fundamental theorem of the Monte Carlo method for it gives an estimation of generating a large deviation in the calculation. For definiteness if we have $\epsilon = 1/1000$ then from the above inequality we can infer that

$$P\{(|G - E(G)|)^2 \geq 1000\ \text{var}(G)\} \leq \frac{1}{1000}.$$

Since $\mathrm{var}(G) = (1/N)\mathrm{var}(g)$ we have

$$P\{(|G - E(G)|)^2 \geq \frac{1000}{N}\mathrm{var}(g)\} \leq \frac{1}{1000}.$$

By making N large enough, one can make the variance of G as small as possible. In other words the probability of getting a large deviation between the value of the integral and the estimate becomes very small. A much stronger result than Chebychev's inequality is the central limit theorem which describes the range of values of G in the course of the Monte Carlo evaluation as $N \to \infty$.

Theorem 4.2.2 (Central limit theorem)**.** *Let X be a random variable defined on $(A, \mathcal{A}, \lambda)$ with mean μ and finite variance σ^2 and let f be the density of X. Let $\bar{x}_N$ be the sample mean of a random sample of size N from $f(\cdot)$, that is*

$$\bar{x}_N = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Let the random variable Z_N be defined by

$$Z_N = \frac{\bar{x}_N - E(\bar{x}_N)}{\sqrt{\mathrm{var}(\bar{x}_N)}}.$$

Then the distribution of Z_N approaches the standard normal distribution as n approaches infinity. That is,

$$\lim_{N\to\infty} P\{a \leq Z_N \leq b\} = \int_a^b \frac{\exp[-t^2/2]}{\sqrt{2\pi}}dt$$

That is, if we set

$$G_N = \frac{1}{N}\sum_n g(x_n)$$

and

$$S_N = (G_N - E(G_N))/[\mathrm{var}(G_N)]^{1/2}$$

then

$$\lim_{N\to\infty} P\{c_1 \leq S_N \leq c_2\} = \int_{c_1}^{c_2} \frac{\exp[-t^2/2]}{\sqrt{2\pi}}dt$$

or

$$\lim_{N\to\infty} P\left(\frac{c_1\sigma(g)}{\sqrt{N}} \leq \frac{1}{N}\sum_{n=1}^{N} g(x_n) - E(g) \leq \frac{c_2\sigma(g)}{\sqrt{N}}\right) = \frac{1}{\sqrt{2\pi}}\int_{c_1}^{c_2} e^{-t^2/2}dt$$

for any constants c_1 and c_2.

To sum up, the results can be interpreted to mean that the absolute value of the error in a Monte Carlo evaluation is, on the average, $\sigma(g)N^{-1/2}$, where

$\sigma(g)$ is the *standard deviation* of g. On the basis of this fact, we can state that the Monte Carlo method for numerical integration yields a probabilistic error bound of the form $O(N^{-1/2})$ in terms of the number N of nodes, independent of the dimension. This feature of the method makes it readily applicable for the high dimensions.

We next turn our attention onto the computational time needed for a Monte Carlo evaluation of an integral in comparison to other deterministic quadrature rules (DQR). Again assume that we are interested in the numerical integration of

$$G = \int_{I^s} f(x)g(x)dx. \tag{4.2.9}$$

The numerical procedure can be written as

$$G \approx \sum_{i=1}^{N} w_i f(x_i) g(x_i) \tag{4.2.10}$$

where $w_i, i = 1, 2, 3, \ldots, N$ are the weights and $x_i, i = 1, 2, 3, \ldots, N$ are the lattice points that fill the hypercube. The error ϵ, associated with this quadrature is bounded by

$$\epsilon \leq \alpha h^k \tag{4.2.11}$$

where h is the size of the interval separating the x_i. The constants c and k are dependent on the numerical scheme and k normally increases with more sophisticated rules. If we assume that the computational time is proportional to the total number of points used, then

$$T(DQR) \propto N = N_0 \left(\frac{1}{h}\right)^s \tag{4.2.12}$$

where N_0 is a constant of the order of 1. Equation (4.2.11) can be rewritten as

$$h \geq \left(\frac{\epsilon}{\alpha}\right)^{1/k} \tag{4.2.13}$$

and so (4.2.12) reduces to

$$T(DQR) \propto N_0 \left(\frac{c}{\epsilon}\right)^{s/k} = O(\epsilon^{-s/k}). \tag{4.2.14}$$

It is evident that as the accuracy needed becomes greater, the greater is the computational time. Also to get same accuracy in higher dimension, the method has to have higher and higher order.

In the case of Monte Carlo (MC), the total time required for the computation is the product of the time for an individual sampling, t_1, times the total number of points,

$$T(MC) \propto t_1 N. \tag{4.2.15}$$

Since the error in a Monte Carlo evaluation goes like

$$\epsilon \approx \frac{\sigma}{N^{1/2}} \tag{4.2.16}$$

we get,

$$T(MC) = t_1 \sigma^2/\epsilon^2 = O(\epsilon^{-2}). \tag{4.2.17}$$

We observe that (4.2.17) is independent of the dimension. Also since it is very difficult to find a numerical scheme in high dimensions which renders a $k > d/2$, we conclude that Monte Carlo calculation is more advantageous than the numerical integration.

We now present computational results for following the test integral taken from [8].

$$I = \int_{[0,1]^s} (1 + 1/s)^s \prod_{i=1}^{s} (x_i)^{1/s} \, d\boldsymbol{x}. \tag{4.2.18}$$

For each of the dimensions, various particle numbers, $10^i, i = 1, 2, \ldots, 6$ were

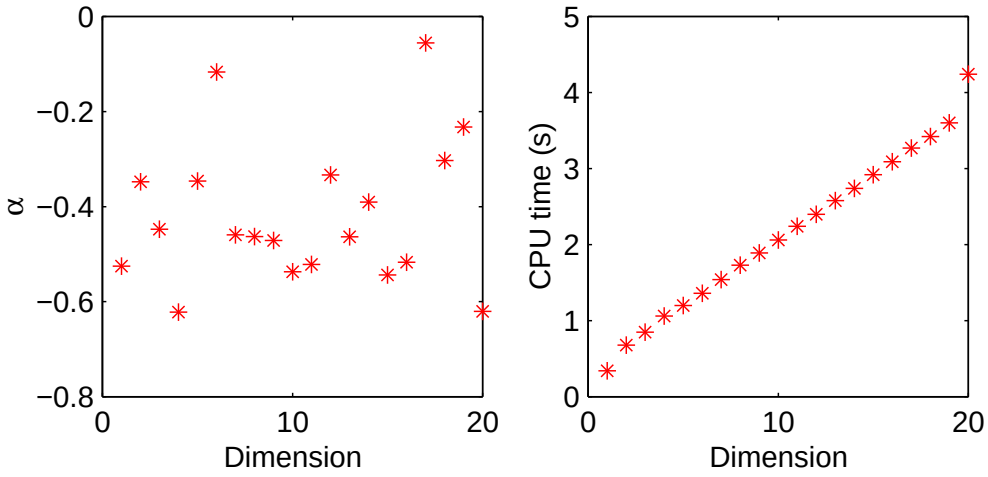


FIGURE 4.2. The figure on the left shows the order of convergence for various dimensions in a single run of Monte Carlo algorithm and the one on the right shows the CPU time taken to carry on the computation with 10^6 particles.

considered and the slope of the least square fit of the particle number versus error plot is taken as the order of convergence for that dimension. Typically, the order of convergence is the exponent α for which the error is $O(N^{\alpha})$. It can be inferred from figure 4.2 that order of convergence close to $1/\sqrt{N}$ is achieved. For some dimensions the result shows worse than $\alpha = -0.5$. This is because we considered only one run of the algorithm. Generally in Monte Carlo algorithms, results are averaged over several runs. The CPU time shows a linear growth with the dimension.

4.3. Sampling

We have sketched how a Monte Carlo calculation works. The next step consists in designing and carrying out such a process. In doing so, it is usually required that random variables be drawn from distribution functions that define the process. For example, in order to evaluate $\int f(x)g(x)dx$, values of x must be drawn from $f(x)$ and the average value of $g(x)$ over such a set of x calculated. With this background, let us define the term *sampling.*

Definition 4.3.1. *Consider a set $\Omega_0 \subset \mathbb{R}^d$ and $x \in \Omega_0$, together with a probability density function $f(x)$ on Ω_0. A sampling procedure is an algorithm that can produce a sequence of values of "x" (random variables) $x_1, x_2, \cdots$ such that for any $\Omega \subset \Omega_0$ we have*

$$P(x_k \in \Omega) = \int_{\Omega} f(x)dx.$$

It will be possible to do this only by having already a sequence of some basic random variables. It has become conventional to start with random variables that are independent and uniformly distributed in [0,1]. We now outline the method of generating random variates according to a given distribution. There are standard methods like the inverse transform method, acceptance-rejection technique etc., for carrying out the process ([**25**], [**10**], [**4**]). We shall now discuss the methods for the univariate case, restricting ourselves to continuous distributions. A similar procedure holds for the multidimensional case.

4.3.1. Inversion method. Let X be a random variable with cumulative distribution function $F_X(x)$. Let Y be uniformly distributed in $(0, 1)$. Then its

cumulative distribution function is given by

$$F_Y(y) = \begin{cases} 0, & y < 0 \\ y, & 0 \leq y \leq 1 \\ 1, & y \geq 1 \end{cases}$$

A random point x distributed according to F_X is generated by solving the following equation,

$$F_X(x) = y \tag{4.3.19}$$

for x. We have the following theorem.

Theorem 4.3.3. *x obtained by solving (4.3.19) is distributed according to F_X.*

Proof. Since F_X is an increasing function, (4.3.19) has a unique solution. Consider,

$$P(x \leq \xi) = P(F_X^{-1}(y) \leq \xi) = P(y \leq F_X(\xi)) = F_X(\xi).$$

Hence the result. ■

We shall exemplify the above procedure for the case of normal distribution.

Definition 4.3.2. *A continuous random variable X has a normal distribution if the p.d.f is*

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\big[-\frac{(x-\mu)^2}{2\sigma^2}\big], \quad -\infty < x < \infty.$$

and is denoted by $\mathcal{N}(\mu, \sigma^2)$. Here μ is the mean and σ^2 is the variance.

Since $X = \mu + \sigma Z$, where Z is the standard normal variable denoted by $\mathcal{N}(0,1)$, we consider only generation from $\mathcal{N}(0,1)$. The cumulative distribution function of Z is given by,

$$F_Z(z) = \begin{cases} \frac{1}{2}\left(1 - \text{erf}\left(\frac{z}{\sqrt{2}}\right)\right) & z \leq 0 \\ \frac{1}{2}\left(1 + \text{erf}\left(\frac{z}{\sqrt{2}}\right)\right) & z \geq 0. \end{cases}$$

If u is a uniform random variable in $(0,1)$, then we consider the following cases. If $u \leq 0.5$, then we solve

$$\frac{1}{2}\left(1 - \text{erf}\left(\frac{z}{\sqrt{2}}\right)\right) = u$$

which yields

$$z = \sqrt{2}\,\text{erf}^{-1}(1-2u)$$

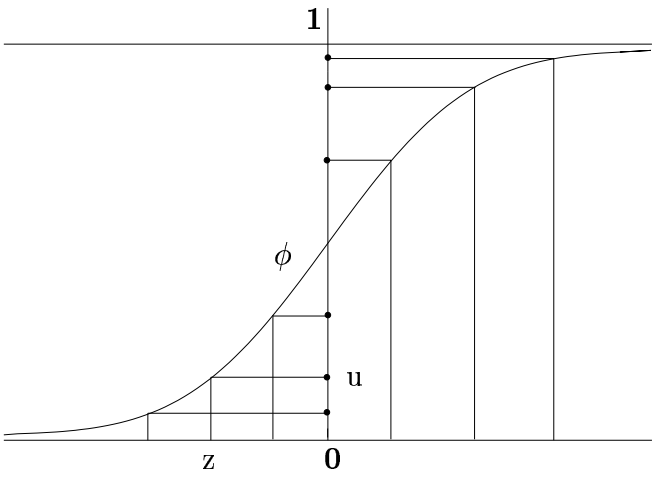


FIGURE 4.3. Sampling from standard normal distribution $\mathcal{N}(0,1)$. ϕ is the cumulative distribution function of $\mathcal{N}(0,1)$. u is a uniform random variable in (0,1) and z is the corresponding standard normal variable.

else we solve,

$$\frac{1}{2}\left(1+\operatorname{erf}\left(\frac{z}{\sqrt{2}}\right)\right)=u$$

to get

$$z=\sqrt{2}\,\operatorname{erf}^{-1}(2u-1).$$

4.3.2. Acceptance-rejection method. This method is due to von Neumann [**18**]. The main idea behind this technique is to sample a random variable according to some appropriate distribution and then subject it to a test to decide whether or not to accept the point. Suppose that X is to be generated from $f_X(x)$. We try to represent

$$f(x)\leq Ch(x)$$

where $C\geq 1$ and $h(x)$ is a distribution which can be sampled easily, for example, using the inversion technique etc. If Y is sampled according to $h(x)$, we check if

$$X\leq CY.$$

If so, then Y is accepted, else it is rejected. The main drawbacks of this method are the determination of a suitable function h and the existence of a C which is close to one so that $Ch(x)$ actually approximates the function f. If one

cannot find good candidates C and h, this algorithm works quite slow for a lot of points are rejected before one is selected.

To summarize, what we have observed in this section is that the Monte Carlo method can be applied to approximate integrals in high dimensions. The error estimate is of the order of $1/\sqrt{N}$, where N is the number of particles considered, independent of the dimension. The question which arises now is that, is it possible to remain in the setup of particle methods and achieve a better order of convergence. The answer to this question is yes and we name the method as quasi-Monte Carlo and describe it in the following section.

4.4. Quasi-Monte Carlo method

The basic idea of a quasi-Monte Carlo method is to replace random samples in a Monte Carlo method by well-chosen deterministic points. The choice of deterministic points depends on the numerical problem we are dealing with. For the problem of numerical integration, the selection criterion is easy to find and leads to the concepts of uniformly distributed sequence and discrepancy. In a weak sense, we say that nodes $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \bar{I}^s$ are uniformly distributed over $\bar{I}^s$, if every subinterval of $\bar{I}^s$ has its share of points. The discrepancy can be viewed as a quantitative measure for the deviation from uniform distribution. The significance of the discrepancy for quasi-Monte Carlo integration will become clear from the Koksma-Hlawka inequality, which bounds the integration error in terms of the discrepancy.

4.4.1. Discrepancy. For the sake of convenience, we normalize the integration domain to be $\bar{I}^s := [0,1]^s$, the closed s-dimensional unit cube. For a given integrand f, the quasi-Monte Carlo approximation yields,

$$\int_{\bar{I}^s} f(\mathbf{u})\, d\mathbf{u} \approx \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{x}_n) \tag{4.4.20}$$

with $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \bar{I}^s$.

Let P be a point set consisting of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \bar{I}^s$. For an arbitrary subset B of $\bar{I}^s$, we define the counting function $A(B; P)$, which indicates the number of n for which $\boldsymbol{x}_n \in P$. Formally,

$$A(B; P) = \sum_{n=1}^{N} c_B(\mathbf{x}_n)$$

where c_B is the characteristic function of B. If $\mathcal{B}$ is a nonempty family of Lebesgue-measurable subsets of $\bar{I}^s$, then a general notion of the discrepancy of the point set P is given by

$$D_N(\mathcal{B}; P) = \sup_{B \in \mathcal{B}} \left| \frac{A(B; P)}{N} - \lambda_s(B) \right| \tag{4.4.21}$$

where λ_s denotes the s-dimensional Lebesgue measure. It can be seen from the definition that $0 \leq D_N(\mathcal{B}; P) \leq 1$. By considering suitable specializations of the family $\mathcal{B}$, we obtain the following two important concepts of discrepancy. We put $I^s = [0, 1)^s$.

Definition 4.4.3. *The star discrepancy $D_N^*(P) = D_N^*(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ of the point set P is defined by $D_N^*(P) = D_N^*(\mathcal{J}^*; P)$, where $\mathcal{J}^*$ is the family of all subintervals of I^s of the form $\prod_{i=1}^s [0, u_i)$, $\mathbf{u} \in \bar{I}^s$.*

Definition 4.4.4. *The extreme discrepancy $D_N(P) = D_N(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of the point set P is defined by $D_N(P) = D_N(\mathcal{J}; P)$, where $\mathcal{J}$ is the family of all subintervals of I^s of the form $\prod_{i=1}^s [u_i, v_i)$, $\mathbf{u}, \mathbf{v} \in \bar{I}^s$.*

The following propositions follow directly from the definition. Refer [**19**] for proofs.

Proposition 4.4.1. For any P consisting of points in $\bar{I}^s$, we have

$$D_N^*(P) \leq D_N(P) \leq 2^s D_N^*(P).$$

Proposition 4.4.2. If $0 \leq x_1 \leq x_2 \ldots \leq x_N \leq 1$ then,

$$D_N^*(x_1, \ldots, x_N) = \frac{1}{2N} + \max_{1 \leq n \leq N} \left| x_n - \frac{2n-1}{2N} \right|.$$

We now turn our attention on to some important error estimates for the quasi-Monte Carlo approximation. For the sake of simplicity we start with the one dimensional case due to Koksma. Refer [**19**] for proofs.

Theorem 4.4.4. *If f is a function of bounded variation and has variation $V(f)$ on $[0, 1]$, then for any point set $x_1, x_2, \ldots, x_N \in [0, 1]$, we have*

$$\left| \int_0^1 f(u)\, du - \frac{1}{N} \sum_{i=1}^N f(x_i) \right| \leq V(f) D_N^*(x_1, \ldots, x_N). \tag{4.4.22}$$

For the multidimensional case, we first need to extend the notion of variation for functions of several variables. For a function f on $\bar{I}^s$ and a subinterval J

of $\bar{I}^s$, let $\Delta(f;J)$ be an alternating sum of the values of f at the vertices of J. Then the variation of f in the sense of Vitali is defined by,

$$V^{(s)}(f) = \sup_{\mathcal{P}} \sum_{J \in \mathcal{P}} |\Delta(f;J)| \tag{4.4.23}$$

A more handy formula is given by

$$V^{(s)}(f) = \int_0^1 \cdots \int_0^1 \left| \frac{\partial^s f}{\partial u_1 \cdots \partial u_s} \right| du_1 \cdots du_s. \tag{4.4.24}$$

whenever the partial derivatives occurring are continuous on $\bar{I}^s$. For $1 \leq k \leq s$ and $1 \leq i_1 < i_2 < \cdots < i_k \leq s$, let $V^{(k)}(f;i_1,\ldots,i_k)$ be the variation of f in the sense of Vitali, restricted to the k-dimensional face $\{(u_1,\ldots,u_s) \in \bar{I}^s : u_j = 1 \text{ for } j \neq i_1,\ldots,i_k\}$. Then

$$V(f) = \sum_{k=1}^{s} \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq s} V^{(k)}(f;i_1,\ldots,i_k) \tag{4.4.25}$$

is called the variation of f in the sense of Hardy and Krause. Now we can state the Koksma-Hlawka inequality.

Theorem 4.4.5. *If f has bounded variation V(f) in the sense of Hardy and Krause, then, for any point set $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N \in \bar{I}^s$, we have*

$$\left| \int_{\bar{I}^s} f(\mathbf{u})\, d\mathbf{u} - \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i) \right| \leq V(f) D_N^*(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N). \tag{4.4.26}$$

The above inequality is not a crude estimate as it can be seen from the following theorem.

Theorem 4.4.6. *For any $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N \in \bar{I}^s$ and any $\epsilon > 0$, there exists a function $f \in C^\infty(\bar{I}^s)$ with $V(f) = 1$, and*

$$\left| \int_{\bar{I}^s} f(\mathbf{u})\, d\mathbf{u} - \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i) \right| > D_N^*(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N) - \epsilon. \tag{4.4.27}$$

A comprehensive list of discrepancy estimates of some QMC sequences can be found in [**19**]. In general the star discrepancy in s dimensions satisfies,

$$D_N^* = O\left(\frac{(\log N)^s}{N} \right) \tag{4.4.28}$$

and one cannot expect anything better than this. Since QMC sequences have small discrepancy, they are sometimes called low discrepancy sequences. Some

of the commonly used low discrepancy sequences are the Sobol sequence, Hammersley sequence, Halton sequence and the Faure sequence. The following figure 4.4 shows the uniformity of a low discrepancy sequence in comparison to random points.

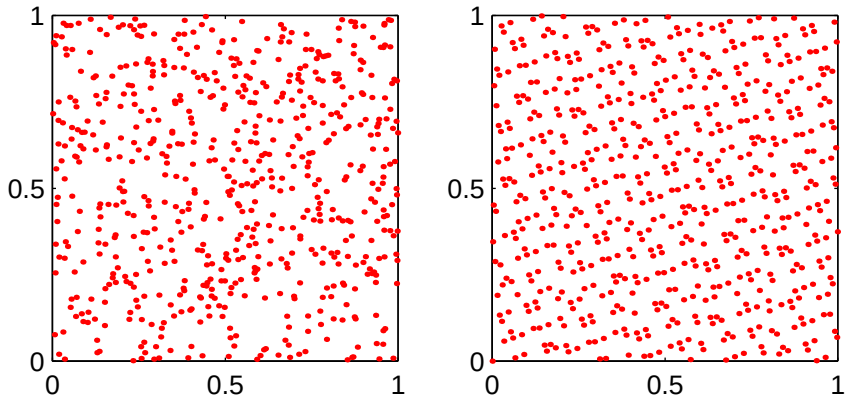


FIGURE 4.4. The figure on the left shows 625 random points and the figure on the right shows 625 Faure points.

We now evaluate the same integral, (4.2.18), we took for the Monte Carlo case and compare the results.

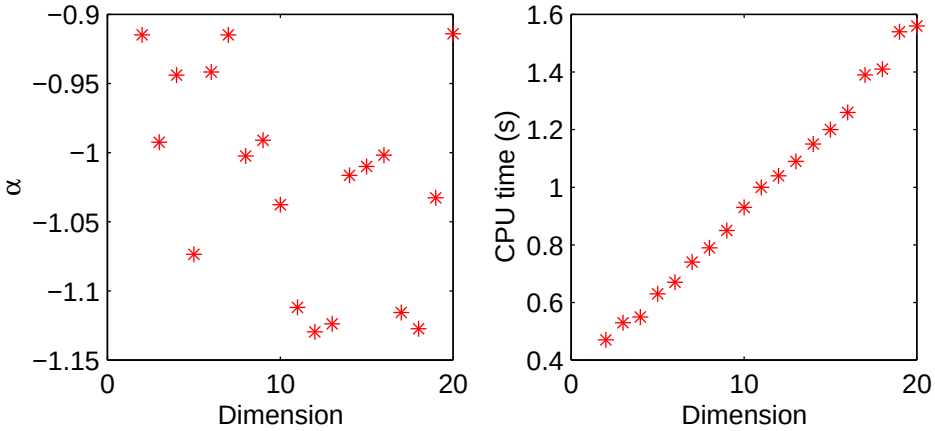


FIGURE 4.5. The figure on the left shows the order of convergence for dimensions 2 to 20. The figure on the right shows the computational time with 10^6 Faure particles.

It can be inferred from figure 4.5 in comparison with figure 4.2, that QMC beats Monte Carlo both in accuracy and computational time.

Remark 2. *To get QMC points with respect to other measures, one can use the same methods as described in section 4.3.*

Though QMC is promising for plain integration, it cannot be applied directly for particle simulations as will see in section 4.6 of partial differential equations.

The original problem we are concerned with is the Fokker-Planck equation. As we saw in the last chapter, the idea of the splitting method is to solve the advection part and diffusion part separately. We now consider the particle approximation of the two subproblems.

4.5. The advection equation

The advection equation can be written as

$$\begin{aligned} &\frac{\partial u}{\partial t} + \operatorname{div}(\boldsymbol{b}u) = 0, \qquad \boldsymbol{x} \in \mathbb{R}^s, \qquad t > 0 \\ &\qquad\qquad u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) \qquad \boldsymbol{x} \in \mathbb{R}^s. \end{aligned} \tag{4.5.29}$$

The operator $\Gamma : \frac{\partial}{\partial t} + \operatorname{div}(\boldsymbol{b}\cdot\,)$ is called the advection operator and physically it signifies the movement of a conserved quantity. Geometrically it can be interpreted as the transfer of information along the curves given by $\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x}, t)$, which are called the characteristics. The first step in applying the particle method would be to sample the initial value $u_0(\boldsymbol{x})$ using either MC or QMC.

$$u_0(\boldsymbol{x}) \to \prod_{u_0}^N = \sum_{i=1}^N \boldsymbol{\omega}_i \, \delta_{\boldsymbol{x}_i^0} \tag{4.5.30}$$

where $\boldsymbol{\omega}_i$, $i = 1, \ldots, N$ are the weights and $\boldsymbol{x}_i^0$, $i = 1, \ldots, N$ are the positions of the sampled particles. Particles now move along the characteristics given by $\dot{\boldsymbol{x}} = \boldsymbol{b}(\boldsymbol{x}, t)$, so that, with the simple Euler discretization

$$\boldsymbol{x}_i^{n+1} = \boldsymbol{x}_i^n + \Delta t \, \boldsymbol{b}(\boldsymbol{x}_i^n). \tag{4.5.31}$$

In terms of the particle distribution we now define the iteration as follows,

$$u^{n+1} = \sum_{i=1}^N \boldsymbol{\omega}_i \, \boldsymbol{x}_i^{n+1} \tag{4.5.32}$$

Now for any test function $\varphi \in C^1(\mathbb{R}^d)$ we have

$$\begin{aligned}
\left\langle \frac{u^{n+1}-u^n}{\Delta t}, \varphi \right\rangle &= \sum_{i=1}^{N} \frac{\boldsymbol{\omega}_i\, \delta_{\boldsymbol{x}_i^n + \boldsymbol{b}(\boldsymbol{x}_i^n)\,\Delta t} - \boldsymbol{\omega}_i\, \delta_{\boldsymbol{x}_i^n}}{\Delta t} \varphi(\boldsymbol{x}) \\
&= \frac{1}{\Delta t} \sum_{i=1}^{N} \boldsymbol{\omega}_i\, [\varphi(\boldsymbol{x}_i^n + \boldsymbol{b}(\boldsymbol{x}_i^n)\,\Delta t) - \varphi(\boldsymbol{x}_i^n)] \\
&\approx \frac{1}{\Delta t} \sum_{i=1}^{N} \boldsymbol{\omega}_i\, [\nabla\, \varphi(\boldsymbol{x}_i^n) \cdot \boldsymbol{b}(\boldsymbol{x}_i^n)]\, \Delta t \\
&= \sum_{i=1}^{N} \boldsymbol{\omega}_i\, \nabla\, \varphi(\boldsymbol{x}_i^n) \cdot \boldsymbol{b}(\boldsymbol{x}_i^n) \\
&= \langle u^n, \boldsymbol{b} \cdot \nabla \varphi \rangle \\
&= -\langle \mathrm{div}(\boldsymbol{b} u^n), \varphi \rangle
\end{aligned}$$

So it is clear that

$$\frac{u^{n+1}-u^n}{\Delta t} \approx -\mathrm{div}(\boldsymbol{b} u^n) \tag{4.5.33}$$

So the particle approximation u^n approximates (4.5.29) in a weak sense.

We again consider the system (4.5.29). We have already seen in section 2 of chapter 3, that the solution of the transport problem can be written as $u_0(\boldsymbol{X}(0;\boldsymbol{x},t))J(0;\boldsymbol{x},t)$. Sampling the initial value gives,

$$\tilde{u}^0 = \sum_{i=1}^{N} \frac{1}{N} \delta_{\boldsymbol{x}_i^0}. \tag{4.5.34}$$

The position of the particles at time t is given by,

$$\boldsymbol{x}_i(t) = \boldsymbol{X}(t;\boldsymbol{x}_i^0, 0) \tag{4.5.35}$$

and the approximate solution is

$$\tilde{u}(t) = \sum_{i=1}^{N} \frac{1}{N} \delta_{\boldsymbol{x}_i(t)}. \tag{4.5.36}$$

To compute the star discrepancy, we consider the family

$$\mathcal{F} = \{\boldsymbol{Q} \,|\, \boldsymbol{Q} = \textstyle\prod_{i=1}^{s}(-\boldsymbol{\infty}, \omega_i),\ \boldsymbol{\omega} \in \mathbb{R}^s\}. \tag{4.5.37}$$

Now,

$$D_N^* = \sup_{\boldsymbol{Q} \in \mathcal{F}} \left| \int_{\boldsymbol{Q}} u_0(\boldsymbol{X}(0;\boldsymbol{x},t)) J(0;\boldsymbol{x},t) d\boldsymbol{x} - \int_{\boldsymbol{Q}} d\tilde{u}(t) \right|. \tag{4.5.38}$$

We consider the two terms in (4.5.38) separately. On using the transformation $\boldsymbol{y} = \boldsymbol{X}(0; \boldsymbol{x}, t)$, the first integral reduces to,

$$\int_{\boldsymbol{Q}} u_0(\boldsymbol{X}(0; \boldsymbol{x}, t)) J(0; \boldsymbol{x}, t) d\boldsymbol{x} = \int_{\boldsymbol{X}(0;\boldsymbol{Q},t)} u_0(\boldsymbol{y}) d\boldsymbol{y}.$$

The second term can be written as,

$$\int_{\boldsymbol{Q}} d\tilde{u}(t) = \int_{\boldsymbol{Q}} d\left(\sum_{i=1}^{N} \frac{1}{N} \delta_{\boldsymbol{x}_i(t)}(\boldsymbol{x})\right) = \int_{\boldsymbol{Q}} d\left(\sum_{i=1}^{N} \frac{1}{N} \delta_{\boldsymbol{X}(t;\boldsymbol{x}_i^0,0)}(\boldsymbol{x})\right).$$

Again using the transformation $\boldsymbol{X}(0; \boldsymbol{x}, t)$, we get,

$$\int_{\boldsymbol{Q}} d\tilde{u}(t) = \int_{\boldsymbol{X}(0;\boldsymbol{Q},t)} d\tilde{u}_0.$$

Combining the results, we get

$$D_N^* = \sup_{\boldsymbol{Q} \in \mathcal{F}} \left| \int_{\boldsymbol{X}(0;\boldsymbol{Q},t)} u_0(\boldsymbol{y}) d\boldsymbol{y} - \int_{\boldsymbol{X}(0;\boldsymbol{Q},t)} d\tilde{u}_0 \right|.$$

The above expression is actually a discrepancy, though in the transformed sets $\boldsymbol{X}(0; \boldsymbol{Q}, t)$. So what we actually require is the relation between $\boldsymbol{X}(0; \boldsymbol{Q}, t)$ discrepancy and $\boldsymbol{Q}$ discrepancy. Niederreiter [**19**], shows a similar result for the case of Jordan measurable subsets of $\bar{I}^s$, but further investigation needs to be done to generalize the same to the family $\mathcal{F}$ considered above.

4.6. The diffusion equation

The diffusion equation also known as the heat equation describes the evolution in time of the density of some quantity such as heat, chemical concentration etc. The heat equation appears as well in the study of Brownian motion.

We are interested in the solution of the following

$$\begin{aligned} \frac{\partial u}{\partial t} &= \Delta u \qquad \boldsymbol{x} \in \mathbb{R}^s, \qquad t > 0 \\ u(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}) \qquad \boldsymbol{x} \in \mathbb{R}^s \end{aligned} \tag{4.6.39}$$

subject to the extra conditions that $u_0(\boldsymbol{x})$ remains bounded and

$$\int_{\mathbb{R}^s} u_0(\boldsymbol{x}) d\boldsymbol{x} = 1 \tag{4.6.40}$$

In this case we know that the solution for any time t can be written down as

$$u(\boldsymbol{x}, t) = G_t * u_0 = \int_{\mathbb{R}^s} G_t(\boldsymbol{x} - \boldsymbol{y}) u_0(\boldsymbol{y}) d\boldsymbol{x} \tag{4.6.41}$$

where $G_t(\boldsymbol{x}, t)$ is the fundamental solution of the heat equation and is given by

$$G_t(\boldsymbol{x}) = \begin{cases} (4\pi t)^{-d/2} \exp\left[-\|\boldsymbol{x}\|^2/4t\right] & t > 0 \\ 0 & t < 0. \end{cases} \tag{4.6.42}$$

Note that G is singular at $(0, 0)$.

Consequent from the definition, we have the following.

Lemma 4.1. *For each $t > 0$,*

$$\int_{\mathbb{R}^s} G_t(\boldsymbol{x}) d\boldsymbol{x} = 1.$$

Proof. Observe that the integrand is separable and can be written as

$$\int_{\mathbb{R}^s} G_t(\boldsymbol{x}) d\boldsymbol{x} = \prod_{i=1}^{s} \int_{\mathbb{R}} \frac{1}{\sqrt{4\pi t}} \exp\left[-\frac{{x_i}^2}{4t}\right] dx_i. \tag{4.6.43}$$

Each of the integral appearing in the product is equal to one as it can be seen as a Gaussian density with mean 0 and variance $2t$. Hence the result. ■

If one assumes that the initial condition is a Gaussian distribution, then it follows from the solution formula that $u(\boldsymbol{x}, t) > 0$ for all $t > 0$. This can be easily seen from the solution formula as the convolution of two Gaussians is again a Gaussian and Gaussian is always positive. Typically for the heat equation, the temperature at any point $\boldsymbol{x}$ is positive meaning that the velocity of propagation of heat is infinite.

With (4.6.40) it can be further shown that

Result 4.1. *For all $t > 0$,*

$$\int_{\mathbb{R}^s} u(\boldsymbol{x}, t) d\boldsymbol{x} = 1.$$

Proof. By 4.6.41,

$$\int_{\mathbb{R}^s} u(\boldsymbol{x}, t) d\boldsymbol{x} = \int_{\mathbb{R}^s} \int_{\mathbb{R}^s} G_t(\boldsymbol{x} - \boldsymbol{y}) u_0(\boldsymbol{y}) d\boldsymbol{y} d\boldsymbol{x}.$$

Letting $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y}$, the integral reduces to

$$\int_{\mathbb{R}^s} \int_{\mathbb{R}^s} G_t(\boldsymbol{z}) u_0(\boldsymbol{y}) d\boldsymbol{y} d\boldsymbol{z} = \int_{\mathbb{R}^s} G_t(\boldsymbol{z}) d\boldsymbol{z} \int_{\mathbb{R}^s} u_0(\boldsymbol{y}) d\boldsymbol{y} = 1 \tag{4.6.44}$$

on using lemma 4.6.43 and (4.6.40). ■

As far as the particle approximation of advection is concerned, it is clear from the previous section that the operator $\partial_t + \mathrm{div}(\boldsymbol{b}\cdot)$ is taken care of by moving the particles along the integral curves of $\boldsymbol{b}$. However, in the case of diffusion the dynamics of the particle is not so straightforward. This was overcome by a method proposed by A. Chorin. He used a random process to simulate diffusion. The random process, however, should be chosen in a suitable manner to approximate diffusion.

Although, the algorithm works for Monte Carlo points, it is not straightforward to replace MC by QMC and expect better accuracy as illustrated in the following example.

Example 1. *Suppose, we would like to solve Cauchy problem for diffusion,*

$$\frac{\partial u}{\partial t} = \frac{1}{2}\sum_{i=1}^{s} \frac{\partial^2 u}{\partial^2 x_i} \tag{4.6.45}$$
$$u(0, \boldsymbol{x}) = G_1(\boldsymbol{x})$$

where G_t is the fundamental solution of (4.6.45) *given by*

$$G_t(\boldsymbol{x}) = (2\pi t)^{-s/2} \exp\left(-\frac{\|x\|^2}{2t}\right). \tag{4.6.46}$$

A particle method algorithm would typically be to sample a set of N particles according to the initial condition and increment the particle position at each time step by $\mathcal{N}(0, \Delta t I)$ distributed random numbers. Figure 4.6 (left) shows the result of the simulation using Monte Carlo points. A similar computation

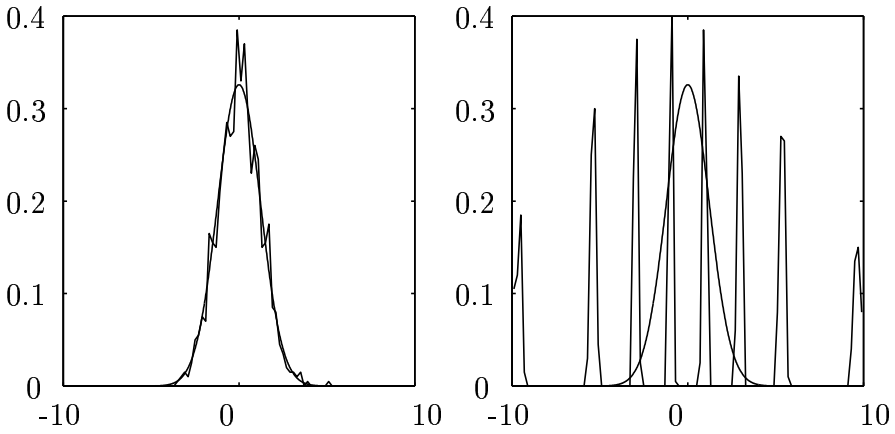


FIGURE 4.6. Monte Carlo simulation of diffusion (left). QMC simulation of diffusion (right).

replacing MC points by QMC points, does not yield the expected result. This is because of the correlation among the QMC points which spoils the convergence and can be explained with the following 1-d argument, [**17**]. Assume we had taken N as a power of two and used the van der Corput sequence,

$$(\boldsymbol{x}_n) = \left\{ \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \frac{5}{16}, \frac{13}{16}, \cdots \right\}$$

then all the odd particles would always get a positive increment and all the even particles would get a negative increment, thereby the particles get drifted away. The problem of correlation was first studied by Lécot and he proposed to sort

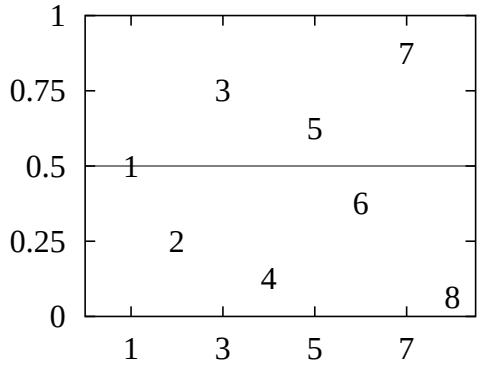


FIGURE 4.7. The van der Corput sequence. All even numbered points are less than 0.5 and all odd numbered points are greater than 0.5

the particles and give increments in a quasi random way. A convergence proof was also given for the spatially homogeneous Boltzmann problem. Morokoff and Caflisch [**17**], applied the idea of Lécot to simulate diffusion in one and two dimensions and obtained significant improvement over Monte Carlo. The results for the 1d case is as depicted in figure 4.8. However the idea of sorting was not clear in high dimensions.

Lécot [**15**], introduced a sorting algorithm which was adaptable to higher dimensions and also shuffled the particle positions at each time step. The sorting is done with respect to each coordinate of the particle position and convergence is proved for any dimension s. For the simple diffusion problem, there is some improvement achieved over the standard Monte Carlo method. However, in order to beat MC in terms of order of convergence the required particle numbers are very large in high dimensions. For a problem in s dimension, a Faure

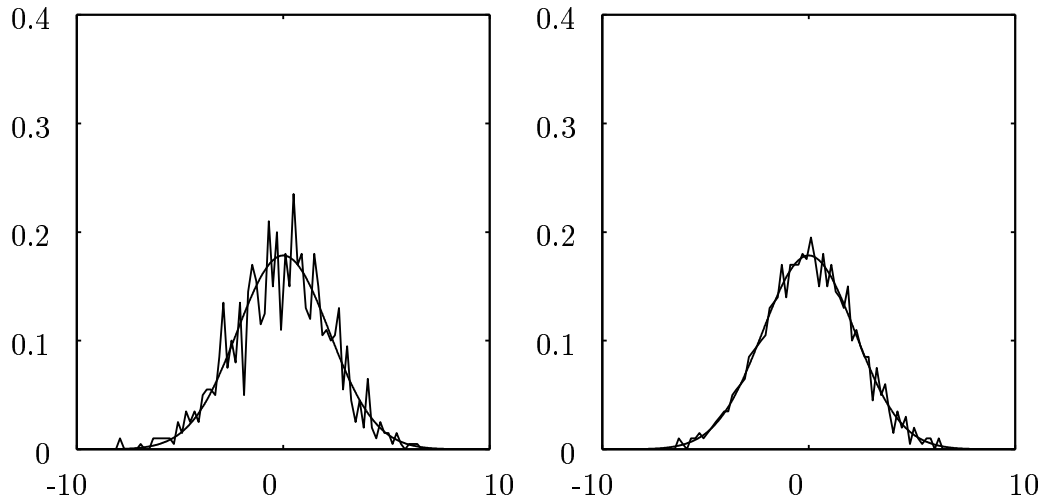


FIGURE 4.8. (left)Monte Carlo simulation of diffusion. (right) QMC simulation of diffusion with sorting.

generator of base b, a prime $\geq 2s$ has to be taken. The minimal particle number is then b^s. To be concrete, for the case $s = 10$, the base b is 23 and the minimal particle number is of the order $23^{10} (\approx 10^{13})$.

Lécot and Schmid [**16**], improved the previous scheme of Lécot and replaced the $2s$ dimensional sequence by a $s + 1$ dimensional sequence. The method was based on partial discretization and numerical results were presented only for the two dimensional case.

We consider now the convergence of the method developed in [**15**]. To do so, we need the following definition.

Definition 4.6.5. *Let X be a point set consisting of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. If ρ is a non-negative Riemann integrable function on $\mathbb{R}^s$ with the property $\int_{\mathbb{R}^s} \rho(\boldsymbol{x}) d\boldsymbol{x} = 1$, then the star ρ-discrepancy of X is defined as*

$$D_N^*(X; \rho) := \sup_{\boldsymbol{\omega} \in \mathbb{R}^s} \left| \frac{1}{N} \sum_{j=1}^{N} \sigma_{\boldsymbol{\omega}}(\boldsymbol{x}_j) - \int_{\mathbb{R}^s} \sigma_{\boldsymbol{\omega}}(\boldsymbol{y}) \rho(\boldsymbol{y}) d\boldsymbol{y} \right|,$$

where $\sigma_{\boldsymbol{\omega}}$ denotes the characteristic function of the interval $\prod_{i=1}^{s} (-\infty, w_i)$.

With this definition, the result in [**15**] can be stated as follows. Let $\boldsymbol{X}^n$ be the point set consisting of $\boldsymbol{x}_1^{(n)}, \ldots, \boldsymbol{x}_N^{(n)}$, then the star c-discrepancy of $\boldsymbol{X}^n$

satisfies

$$D_N^*(X^{(n)};c_n) \leq D_N^*(X^{(0)};c_0) + b^{d_1+\cdots+d_{s-1}+\lfloor d_s/2\rfloor}\sum_{m=0}^{n-1} D_N(Y^m) + n\left(\frac{1}{b^{d_1}}+\cdots+\frac{1}{b^{d_s-1}}+\frac{1}{b^{\lfloor d_s/2\rfloor}}\right) \tag{4.6.47}$$

where b is the smallest prime $\geq 2s+1$, $N = b^{d_1+\cdots+d_{s-1}+d_s}$ with $d_i \geq 0$, $i = 1,\ldots,s$, is the total particle number considered, Y is the (t,m,s)-net used and c_n is the exact solution of the diffusion equation at the n^{th} time step. The estimate shows that the method is better than MC for $d_i \geq 1$. This actually leads to large particle numbers and also the order of convergence is close to 0.5. However, the estimate fails on setting one or more of the d_is to zero. To be concrete, if we set $d_1 = 0$, then from (4.6.47), it is evident that the last term on the right has a leading term n, and this does not go to zero as $N \to \infty$.

We pick up the idea of Lécot in [**15**]. In our method, we consider a $s+1$ dimensional QMC sequence and choose $d_i = 0, i \geq 2$, so that we can work with less number of particles. For example, in the case $s = 10$ we can start with 11 particles and continue further with multiples of 11. Though estimate (4.6.47) fails to show convergence in this case, the method works as we shall see later in the chapter on numerical results.

We now focus on the particle approximation. As before,

$$\tilde{u}^0 = \prod_{u_0}^N = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{x}_i^0}. \tag{4.6.48}$$

Using the solution formula, the solution after one time step (Δt) can be written as

$$\tilde{u}^1(\boldsymbol{x}) = \int_{\mathbb{R}^s} G_{\Delta t}(\boldsymbol{x}-\boldsymbol{y})d\prod_{u_0}^N(\boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^N G_{\Delta t}(\boldsymbol{x}-\boldsymbol{x}_i^0). \tag{4.6.49}$$

Observe that (4.6.49) is no longer a particle approximation but a smooth function of $\boldsymbol{x}$. In order to get back to particles, we need to do another step namely,

$$\tilde{u}^1 = \prod_{\tilde{u}^1}^M = \sum_{i=1}^M \boldsymbol{\omega}_i\, \delta_{\boldsymbol{x}_i^1}. \tag{4.6.50}$$

Observing that two measures are close to each other if they integrate general test functions to approximately the same value, we consider

$$\begin{aligned}\int_{\mathbb{R}^s} \varphi(\boldsymbol{x})\tilde{u}^1(\boldsymbol{x})d\boldsymbol{x} &= \frac{1}{N}\sum_{i=1}^{N}\int_{\mathbb{R}^s} \varphi(\boldsymbol{x})G_{\Delta t}(\boldsymbol{x}-\boldsymbol{x}_i^0)d\boldsymbol{x} \\ &= \frac{1}{N}\sum_{i=1}^{N}\int_{\mathbb{R}^s} \varphi(\boldsymbol{y}+\boldsymbol{x}_i^0)G_{\Delta t}(\boldsymbol{y})d\boldsymbol{y}.\end{aligned}$$

We now use the transformation $z_i = H_i(y_i)$ where

$$H_i(x) = \frac{1}{2}\left(1+\operatorname{erf}\left(\frac{x}{\sqrt{4\Delta t}}\right)\right).$$

Thus, $\mathbf{H}:\mathbb{R}^s \to [0,1]^s$ and the above integral transforms as

$$\int_{\mathbb{R}^s} \varphi(\boldsymbol{x})\tilde{u}^1(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N}\int_{I^s} \varphi(\mathbf{H}^{-1}(\boldsymbol{z})+\boldsymbol{x}_i^0)d\boldsymbol{z}.$$

Evaluating the $\boldsymbol{z}$ integral with N' QMC points yields

$$\int_{I^s} \varphi(\mathbf{H}^{-1}(\boldsymbol{z})+\boldsymbol{x}_i^0)d\boldsymbol{z} = \frac{1}{N'}\sum_{j=1}^{N'} \varphi(\mathbf{H}^{-1}(\boldsymbol{z}_j)+\boldsymbol{x}_i^0) \tag{4.6.51}$$

so that

$$\int_{\mathbb{R}^s} \varphi(\boldsymbol{x})\tilde{u}^1(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\frac{1}{N'}\sum_{i=1}^{N}\sum_{j=1}^{N'} \varphi(\mathbf{H}^{-1}(\boldsymbol{z}_j)+\boldsymbol{x}_i^0).$$

We observe that in the above process we use $M = NN'$ particles. Since we need a large N' for good accuracy, the particle number increases drastically at each step. In fact

$$u_0 \longrightarrow \prod_{u_0}^{N} \longrightarrow \tilde{u}^1 \longrightarrow \prod_{\tilde{u}^1}^{NN'} \longrightarrow \tilde{u}^2 \longrightarrow \prod_{\tilde{u}^2}^{NN'N'} \cdots \tag{4.6.52}$$

So after k steps the particle number is $N(N')^k$, which is enormously large even for reasonable values of k. Also the memory required to store these values is huge and this makes the scheme expensive.

In order to stay with a fixed number of particles, we do the following. We start with partitioning the unit interval $[0,1]$ into disjoint subintervals $I_j, j = 1,2,\ldots,N$ with the property that $|I_j| = 1/N, j = 1,2,\ldots,N$ and set $\chi_j = 1_{I_j}$, that is

$$\chi_j(x) = \begin{cases} 1 & \text{if } x \in I_j \\ 0 & \text{if } x \notin I_j. \end{cases}$$

Then, we have

$$\int_0^1 \chi_i(s)ds = \frac{1}{N}$$

This is a result we need presently for simplification.

$$\begin{aligned}\int_{\mathbb{R}^s} \varphi(\boldsymbol{x})\tilde{u}^1(\boldsymbol{x})d\boldsymbol{x} &= \frac{1}{N}\sum_{i=1}^N \int_{I^s} \varphi(\mathbf{H}^{-1}(\boldsymbol{z}) + \boldsymbol{x}_i^0)d\boldsymbol{z} \\ &= \int_0^1 \int_{I^s} \sum_{i=1}^N \chi_i(s)\varphi(\mathbf{H}^{-1}(\boldsymbol{z}) + \boldsymbol{x}_i^0)d\boldsymbol{z}ds \\ &= \int_{I^{s+1}} F_N(\boldsymbol{z}, s)d\boldsymbol{z}ds\end{aligned}$$

with

$$F_N(\boldsymbol{z}, s)d\boldsymbol{z}ds = \sum_{i=1}^N \int_{I^s} \varphi(H^{-1}(\boldsymbol{z}) + \boldsymbol{x}_i^0).$$

We now approximate the above integral with N QMC points,

$$\begin{aligned}\int_{I^{s+1}} F_N(\boldsymbol{z}, s)d\boldsymbol{z}ds &\approx \frac{1}{N}\sum_{k=1}^N F_N(\boldsymbol{z}_k, s_k) \\ &= \frac{1}{N}\sum_{k=1}^N \sum_{i=1}^N \chi_i(s_k)\varphi(\mathbf{H}^{-1}(\boldsymbol{z}_k) + \boldsymbol{x}_i^0).\end{aligned}$$

Observe that for each $k \in \{1, 2, \ldots, N\}$, there exists some $i = \sigma(k)$ such that $\chi_i(s_k) = 1$.

Since $\bigcup_{i=1}^N I_i = [0, 1]$,

$$\begin{aligned}\frac{1}{N}\sum_{k=1}^N F_N(\boldsymbol{z}_k, s_k) &= \frac{1}{N}\sum_{k=1}^N \varphi(\underbrace{\mathbf{H}^{-1}(\boldsymbol{z}_k) + \boldsymbol{x}_{\sigma(k)}^0}_{\boldsymbol{x}_k^1}) \\ &= \int_{\mathbb{R}^s} \varphi(\boldsymbol{x})\, d\prod_{\tilde{u}^1}^N(\boldsymbol{x})\end{aligned}$$

where,

$$\prod_{\tilde{u}^1}^N = \frac{1}{N}\sum_{k=1}^N \delta_{\boldsymbol{x}_k^1}. \tag{4.6.53}$$

Result 4.2. *If each of the intervals I_j contains exactly one point s_k, then the map σ is invertible, i.e, given $k \in \{1, \ldots, N\}$ there exists a unique $i \in \{1, \ldots, N\}$ such that $k = \sigma^{-1}(i)$, and*

$$\{\boldsymbol{x}_k^1 \mid k = 1, 2, \ldots, N\} = \{\boldsymbol{x}_k^0 + \mathbf{H}^{-1}(\boldsymbol{z}_{\sigma^{-1}(k)}) \mid k = 1, 2, \ldots, N\}. \tag{4.6.54}$$

According to the above result, the particle position after one time step is obtained by incrementing each component of the present position by an amount sampled from normal distribution with mean zero and variance $2\Delta t$.

In order to ensure that each of the intervals I_j contains exactly one point, we use the concept of (t, m, s)-nets.

Definition 4.6.6. *An elementary interval in base $b \geq 2$ in dimension $s \geq 1$ is a subinterval E of $\bar{I}^s$ of the form*

$$E = \prod_{i=1}^{s} \left[a_i b^{-d_i}, (a_i + 1) b^{-d_i}\right)$$

with $a_i, d_i \in \mathbb{Z}, d_i \geq 0, 0 \leq a_i < b^{d_i}$ for $1 \leq i \leq s$.

With this notion, we define:

Definition 4.6.7. *Let $0 \leq t \leq m$ be integers. A (t, m, s)-net in base b is a point set P consisting of b^m points in $\bar{I}^s$ such that every elementary interval E of volume b^{t-m} contains exactly b^t points.*

In our analysis, since we require that each interval I_j should just contain a point, we choose $t = 0$.

Now it remains to see how good we have approximated

$$\int_{I^{s+1}} F_N(\boldsymbol{z}, s) d\boldsymbol{z} ds \approx \frac{1}{N} \sum_{k=1}^{N} F_N(\boldsymbol{z}_k, s_k)$$

using N QMC points. The answer is provided by the Koksma-Hlawka inequality. Accordingly, the error is $V(F_N) \cdot D_N$, where $V(F_N)$ is the variation of the function F_N and D_N is the discrepancy of the $s + 1$ dimensional QMC sequence. In order to keep this error small, we wish to have the variation and the discrepancy as small as possible. It may be noted that $V(F_N)$ depends on N and can be as large as $2N$ as seen from the following example.

Example 2. *Consider the 1d case with*

$$u_0(x) = \frac{1}{\sqrt{2\pi}} \exp[-x^2/2]$$

and the error after one time step $\Delta t = 1/2$.

We take as uniform random numbers $u_i = (N - i + 1/2)/N$ which is evenly distributed in $[0, 1]$. Since the initial value is a standard Gaussian, we do the

initial sampling by the inversion method outlined earlier. In our notation this is

$$x_i^0 = H^{-1}(u_i) \qquad i = 1, 2, \ldots, N.$$

As test function we take $\mathbf{1}_{(-\infty,0)}(x)$. Then

$$F_N(z,s) = \sum_{i=1}^{N} \chi_i(s)\varphi(x_i^0 + H^{-1}(z)).$$

Note that

$$\infty < x_i^0 + H^{-1}(z) < 0$$

is equivalent to

$$0 < z < H(-x_i^0) = 1 - H(x_i^0) = 1 - u_i = \frac{i - 1/2}{N}.$$

Hence $F_N(z,s)$ can be written as,

$$F_N(z,s) = \sum_{i=1}^{N} \mathbf{1}_{(0,1-u_i)}(z)\mathbf{1}_{(I_i)}(s).$$

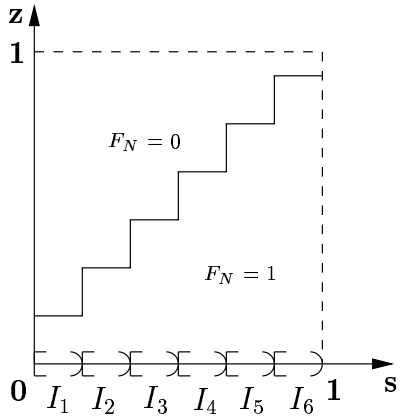


FIGURE 4.9. Graph of $F_6(z,s)$ in the z,s plane.

If we choose a special partition as shown in fig. 4.10, then since the support of function F_N has $2N-2$ corners, $V(F_N(z,s)) \geq 2N-2$ and Koksma-Hlawka inequality does not ensure convergence in this case. It is therefore desirable to make the variation less thereby making the error small, again by Koksma-Hlawka inequality. Consistent with later use, we shall denote the support of F_N in the (z,s) plane by $\mathcal{E}_N$. The trick is to reduce the number of corners of $\mathcal{E}_N$ in order to make the variation small. We return now to approximating the area

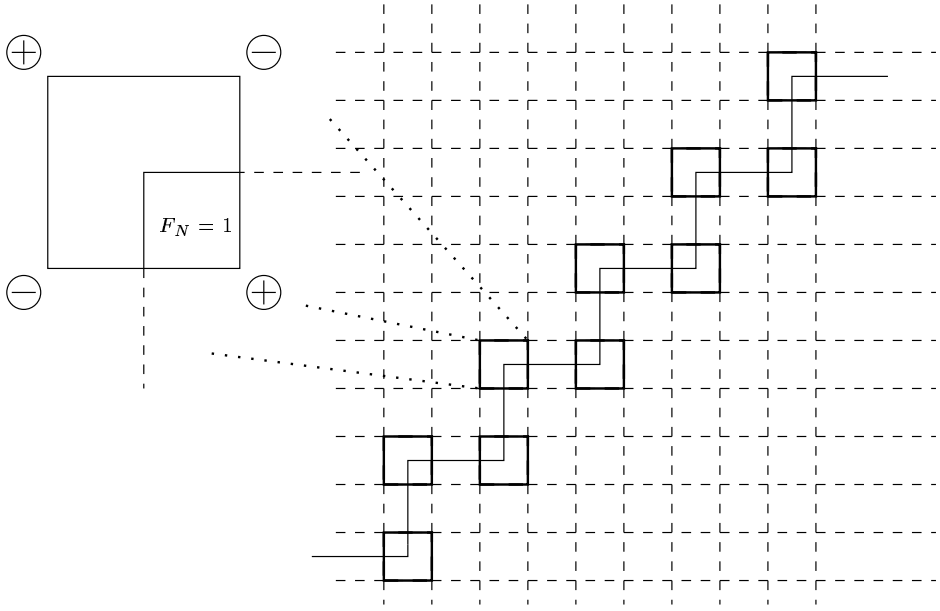


FIGURE 4.10. A partition to calculate the variation. Observe that each square containing a corner contributes a 1 to the variation.

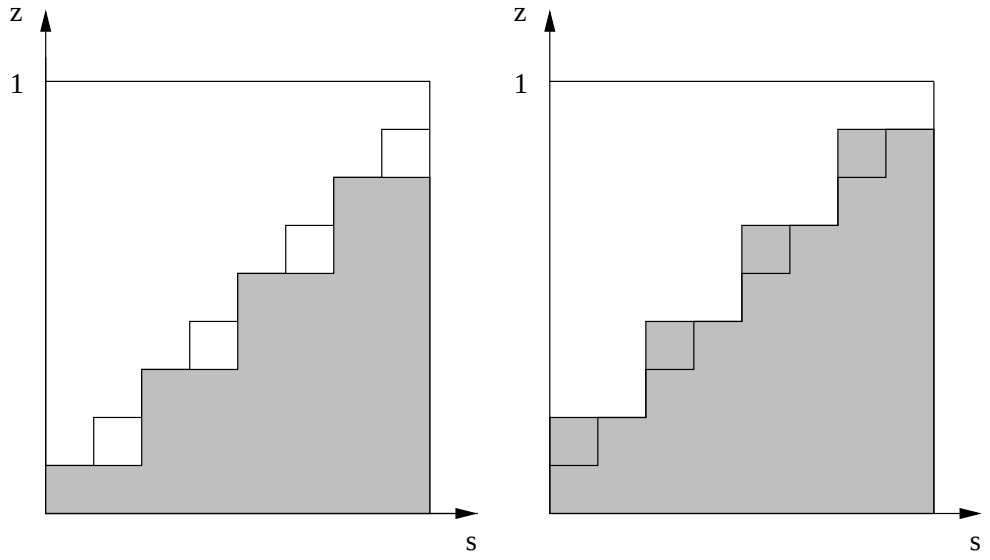


FIGURE 4.11. Graph of the approximations $\underline{\mathcal{E}}_N$ (left) and $\bar{\mathcal{E}}_N$ (right) in the (z, s) plane. Note that $\underline{\mathcal{E}}_N \subset \mathcal{E}_N \subset \bar{\mathcal{E}}_N$.

of $\mathcal{E}_N$. We consider instead of N corners, the upper and lower approximations of $\mathcal{E}_N$, $\bar{\mathcal{E}}_N, \underline{\mathcal{E}}_N$ respectively, with $\sqrt{N}$ corners as shown in figure 4.11. Thus we get by Koksma-Hlawka inequality, that the error, using QMC points is $\sqrt{N}/N$, where the factor $1/N$ is taken as the discrepancy of the QMC sequence up to a logarithmic factor. However, we have considered a different function and hence we have to take into account the area of difference between the upper and lower estimate $\lambda(\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N)$.

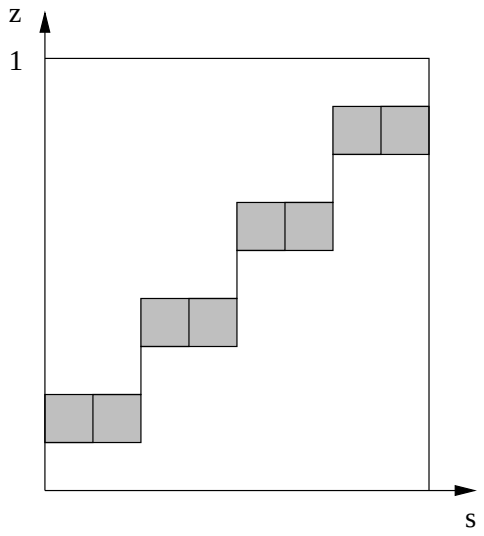


FIGURE 4.12. The shaded area shows the graph of $\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N$.

The area of difference can be seen from figure 4.12 to be $1/\sqrt{N}$. Thus we see that we can achieve $1/\sqrt{N}$ convergence with our algorithm. This seems to be the optimal choice since we need to strike a balance between approximating the function and keep the area of difference roughly equal.

We return to the problem of estimating the approximation, by comparing the measures on the family of intervals

$$(-\boldsymbol{\infty}, \boldsymbol{\omega}) = \prod_{i=1}^{d} (-\infty, \omega_i) \qquad \boldsymbol{\omega} \in \mathbb{R}^s$$

in the general multidimensional case. Our test functions are now $\varphi(\boldsymbol{x}) = \mathbf{1}_{(-\infty,\omega)}(\boldsymbol{x})$. Observe that

$$-\boldsymbol{\infty} < \mathbf{H}^{-1}(\boldsymbol{z}) + \boldsymbol{x}_i^0 < \boldsymbol{\omega}$$

is defined componentwise

$$-\infty < H_j^{-1}(z) + (\boldsymbol{x}_i^0)_j < \omega_j \quad \forall\, j = 1, \ldots, s.$$

Hence we have

$$0 < \boldsymbol{z} < \mathbf{H}(\boldsymbol{\omega} - \boldsymbol{x}_i^0)$$

We introduce

$$\mathcal{E}_N = \bigcup_{i=1}^{N} (0, \mathbf{H}(\boldsymbol{\omega} - \boldsymbol{x}_i^0)) \times I_i$$

Then,

$$\begin{aligned} \mathbf{1}_{\mathcal{E}_N}(\boldsymbol{z}, s) &= \sum_{i=1}^{N} \mathbf{1}_{(0, \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{x}_i^0)))}(\boldsymbol{z})\, \mathbf{1}_{I_i}(s) \\ &= \sum_{i=1}^{N} \chi_i(s)\, \mathbf{1}_{(-\infty, \boldsymbol{w})}(\mathbf{H}^{-1}(\boldsymbol{z}) + \boldsymbol{x}_i^0) \\ &= F_N(\boldsymbol{z}, s) \end{aligned}$$

Hence with $\prod^N = \frac{1}{N} \sum_{i=1}^{N} \delta_{(\boldsymbol{z}_i, s_i)}$, we have

$$\begin{aligned} &\left| \prod_{\tilde{u}^1}^{N}(-\infty, \boldsymbol{\omega}) - (\tilde{u}^1 \lambda_s)(-\infty, \boldsymbol{\omega}) \right| \\ &= \left| \int \mathbf{1}_{\mathcal{E}_N}(\boldsymbol{z}, s) \prod^N (d\boldsymbol{z} ds) - \int_{I^{s+1}} \mathbf{1}_{\mathcal{E}_N}(\boldsymbol{z}, s) d\boldsymbol{z} ds \right| \\ &= \left| \prod^N (\mathcal{E}_N) - \lambda_{s+1}(\mathcal{E}_N) \right| \end{aligned}$$

From the definition of $\mathcal{E}_N$, it is clear that it depends on $\boldsymbol{\omega}$. But what we are interested in is an estimate independent of $\boldsymbol{\omega}$. Since $\mathcal{E}_N$ has too many corners we go over to the approximations $\underline{\mathcal{E}}_N$ and $\bar{\mathcal{E}}_N$ with less corners. Note that if $\underline{\mathcal{E}}_N \subset \mathcal{E}_N \subset \bar{\mathcal{E}}_N$, then

$$\prod^N (\underline{\mathcal{E}}_N) \leq \prod^N (\mathcal{E}_N) \leq \prod^N (\bar{\mathcal{E}}_N)$$

$$\Longrightarrow \qquad \prod^N (\underline{\mathcal{E}}_N) - \lambda_{s+1}(\mathcal{E}_N) \leq \prod^N (\mathcal{E}_N) - \lambda_{s+1}(\mathcal{E}_N) \leq \prod^N (\bar{\mathcal{E}}_N) - \lambda_{s+1}(\mathcal{E}_N)$$

We now need an estimate of $\lambda_{s+1}(\mathcal{E}_N)$. Note that

$$\begin{aligned} -\lambda_{s+1}(\mathcal{E}_N) &= -\lambda_{s+1}(\underline{\mathcal{E}}_N) - \lambda_{s+1}(\mathcal{E}_N \setminus \underline{\mathcal{E}}_N) \;\geq\; -\lambda_{s+1}(\underline{\mathcal{E}}_N) - \lambda_{s+1}(\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N) \\ -\lambda_{s+1}(\mathcal{E}_N) &= -\lambda_{s+1}(\bar{\mathcal{E}}_N) + \lambda_{s+1}(\bar{\mathcal{E}}_N \setminus \mathcal{E}_N) \;\leq\; -\lambda_{s+1}(\bar{\mathcal{E}}_N) + \lambda_{s+1}(\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N) \end{aligned}$$

Hence,

$$\begin{aligned} -\left| \prod^N (\underline{\mathcal{E}}_N) - \lambda_{s+1}(\underline{\mathcal{E}}_N) \right| - \lambda_{s+1}(\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N) &\leq \left| \prod^N (\mathcal{E}_N) - \lambda_{s+1}(\mathcal{E}_N) \right| \\ &\leq \left| \prod^N (\bar{\mathcal{E}}_N) - \lambda_{s+1}(\bar{\mathcal{E}}_N) \right| + \lambda_{s+1}(\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N) \end{aligned}$$

If we set,

$$\begin{aligned} A_N &:= \max\left\{\left|\prod^N(\underline{\mathcal{E}}_N) - \lambda_{s+1}(\underline{\mathcal{E}}_N)\right|, \left|\prod^N(\bar{\mathcal{E}}_N) - \lambda_{s+1}(\bar{\mathcal{E}}_N)\right|\right\} \\ B_N &:= \lambda_{s+1}(\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N) \end{aligned}$$

Then,

$$\left|\prod^N(\mathcal{E}_N) - \lambda_{s+1}(\mathcal{E}_N)\right| \leq A_N + B_N \tag{4.6.55}$$

Having reached this stage, what is now important is to construct the suitable sets $\underline{\mathcal{E}}_N$ and $\bar{\mathcal{E}}_N$. We need to have a nice structure for $\mathcal{E}_N$ so that we can have the bounding boxes $\underline{\mathcal{E}}_N$ and $\bar{\mathcal{E}}_N$. Assume that the points $\boldsymbol{x}_i^0$ are sorted with respect to the first coordinate in decreasing order. Then,

$$H_1(\boldsymbol{\omega} - \boldsymbol{x}_i^0)_1$$

are sorted in ascending order. The subscript 1 denotes the first component of the vector under consideration.

A typical plot of $\mathcal{E}_N$ intersected with (z_1, s) plane Γ looks like as shown in figure 4.14.

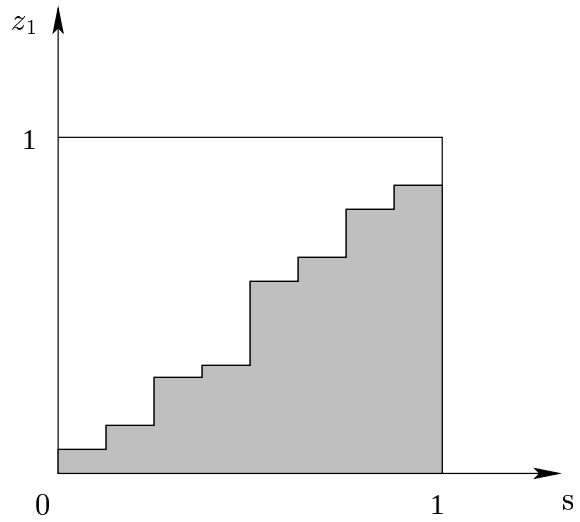


FIGURE 4.13. The shaded region shows the graph of $\mathcal{E}_N \cap \Gamma$ in the (z_1,s) plane.

It may be observed that $\mathcal{E}_N$ has lot of corners and so as shown in example 2, the variation of F_N can be large.

Assume we start with N particles and let $N = b^m$ for some b and m. Consider groups of particles of length b^μ for $0 \leq \mu \leq m$. So there are in total $b^{m-\mu}$

groups. Define,

$$\underline{\mathcal{E}}_N := \bigcup_{k=1}^{b^{m-\mu}} \Big[0, H_1((\boldsymbol{\omega} - \boldsymbol{x}^0_{(k-1)b^\mu+1})_1)\Big) \times I^{s-1} \times \big[(k-1)b^{\mu-m}, kb^{\mu-m}\big) \tag{4.6.56}$$

$$\bar{\mathcal{E}}_N := \bigcup_{k=1}^{b^{m-\mu}} \big[0, H_1((\boldsymbol{\omega} - \boldsymbol{x}^0_{kb^\mu})_1)\big) \times I^{s-1} \times \big[(k-1)b^{\mu-m}, kb^{\mu-m}\big) \tag{4.6.57}$$

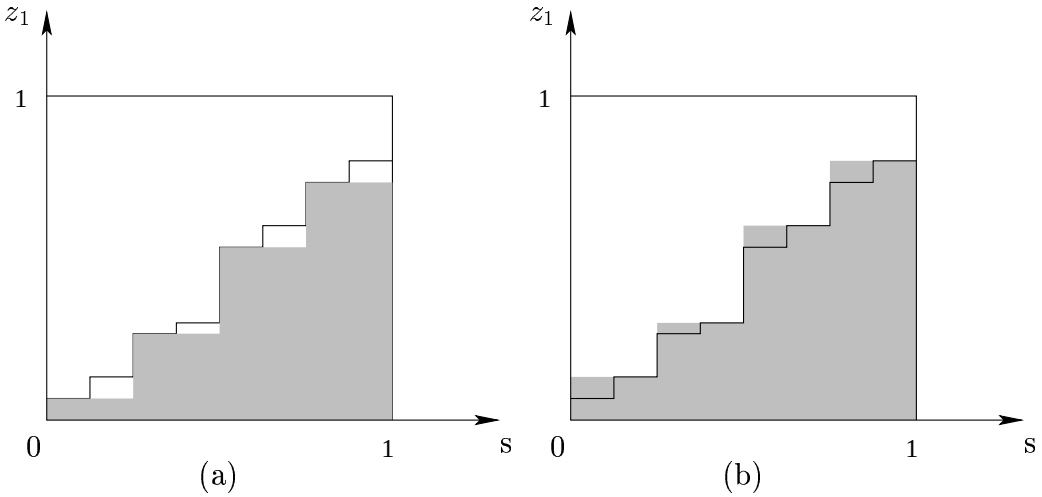


FIGURE 4.14. The shaded region shows the graph of (a) $\underline{\mathcal{E}}_N \cap \Gamma$ and (b) $\bar{\mathcal{E}}_N \cap \Gamma$ in the (z_1,s) plane. In this case $b = 2, m = 3, \mu = 1$

It is clear from the construction that $\underline{\mathcal{E}}_N \subset \mathcal{E}_N \subset \bar{\mathcal{E}}_N$ and we have

$$\begin{aligned} \lambda_{s+1}(\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N) &\leq \sum_{k=1}^{b^{m-\mu}} \lambda_1 \left(H_1((\boldsymbol{\omega} - \boldsymbol{x}^0_{(k-1)b^\mu+1})_1), H_1((\boldsymbol{\omega} - \boldsymbol{x}^0_{kb^\mu})_1)\right) b^{\mu-m} \\ &\leq \lambda_1 \left(H_1((\boldsymbol{\omega} - \boldsymbol{x}^0_1)_1), H_1((\boldsymbol{\omega} - \boldsymbol{x}^0_N)_1)\right) b^{\mu-m} \\ &\leq b^{\mu-m} \end{aligned} \tag{4.6.58}$$

Note that in the second step, we have sorted the position of the particles with respect to the first coordinate. Thus,

$$B_N \leq b^{\mu-m} \tag{4.6.59}$$

What we finally need is an estimate on the discrepancy of a point set on an interval in I^s. We introduce,

$$D_N = \sup_{a', b' \in I^s} \left| \prod^N [a', b') - \lambda_{s+1}[a', b') \right| \tag{4.6.60}$$

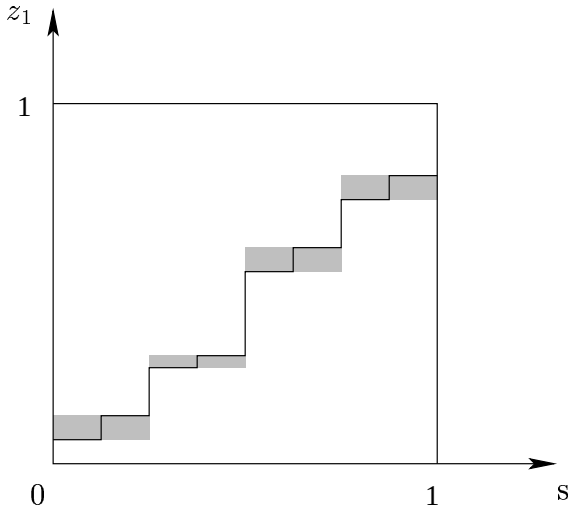


FIGURE 4.15. The shaded region shows the graph of $\bar{\mathcal{E}}_N \setminus \underline{\mathcal{E}}_N$ in the (z_1,s) plane. Note that it is the union of disjoint sets.

Then,

$$\left|\prod^N(\underline{\mathcal{E}}_N) - \lambda_{s+1}(\underline{\mathcal{E}}_N)\right| = b^{m-\mu}D_N \tag{4.6.61}$$

since $\underline{\mathcal{E}}_N$ consists of $b^{m-\mu}$ rectangles and each of them are bounded by D_N. A similar estimate holds for $\bar{\mathcal{E}}_N$ since it has a similar structure and hence we have,

$$\left|\prod^N(\bar{\mathcal{E}}_N) - \lambda_{s+1}(\bar{\mathcal{E}}_N)\right| = b^{m-\mu}D_N \tag{4.6.62}$$

In our terminology,

$$A_N \leq b^{m-\mu}D_N \tag{4.6.63}$$

Note that a rectangle $[\boldsymbol{a}', \boldsymbol{b}') \in I^s$ has 2^s corners and

$$V(\mathbf{1}_{[\boldsymbol{a}',\boldsymbol{b}')}) \leq 2^s$$

By Koksma-Hlawka inequality we have,

$$D_N \leq V(\mathbf{1}_{[\boldsymbol{a}',\boldsymbol{b}')})D_N^*$$

Combining the results we get,

$$\begin{aligned}\left|\prod^N(\mathcal{E}_N) - \lambda_{s+1}(\mathcal{E}_N)\right| &= A_N + B_N \\ \left|\prod^N(\mathcal{E}_N) - \lambda_{s+1}(\mathcal{E}_N)\right| &= A_N + B_N \\ &\leq b^{m-\mu}D_N^* + b^{\mu-m}\end{aligned}$$

We thus have the following lemma.

Lemma 4.2. *The error in the quasi-Monte Carlo approximation is given by*

$$D_N \leq b^{m-\mu} D_N^* + b^{\mu-m}$$

where $N = b^m$ is the number of particles, μ is an integer with $0 \leq \mu \leq m$ and D_N^* is the star discrepancy of the $(0, m, s+1)$-net.

We replace our estimate obtained in lemma 4.2 with the estimate obtained in lemma 2 of [**15**] and the rest of the proof is as done in [**15**]. In our discussion on the particle method for the diffusion equation, we have proved the following theorem.

Theorem 4.6.7. *Let $\boldsymbol{X}^n$ be the point set consisting of $\boldsymbol{x}_1^{(n)}, \ldots, \boldsymbol{x}_N^{(n)}$, then the star c-discrepancy of $\boldsymbol{X}^n$ satisfies,*

$$D_N^*(\boldsymbol{X}^{(n)}; c_n) \leq D_N^*(\boldsymbol{X}^0; c_0) + b^{m-\mu} \sum_{m=0}^{n-1} D_N^*(Y^{(m)}) + n b^{\mu-m} \tag{4.6.64}$$

where c_n is the exact solution of the diffusion equation at the n^{th} time step and Y is a $(0, m, s+1)$-net in base b, the smallest prime $\geq s+1$.

The star discrepancy of a low discrepancy is as given in (4.4.28), which is essentially $1/N$ behavior, except for the logarithmic factor. From (4.6.64), we see that the error has two contributions, namely, $b^{m-\mu} D_N^*(Y)$ and $b^{\mu-m}$. In order to minimize the error we choose μ such that

$$-\mu = \mu - m \tag{4.6.65}$$

which yields $\mu = \lfloor m/2 \rfloor$ or $\mu = \lceil m/2 \rceil$, up to a logarithmic factor. This means that we can achieve an order of convergence $1/\sqrt{N}$, up to a logarithmic factor, which is comparable to the Monte Carlo estimate. It is now the comparison between deterministic $1/\sqrt{N}$ versus the stochastic $1/\sqrt{N}$ convergence. We shall later see in the chapter on numerical results, that there is still some advantage in using QMC.
Combining the observations in the last two sections, the algorithm of our particle method in s dimensions can be summarized as follows.

Algorithm 4.1.

1. **Input** $m \geq 1$, Δt *and* n; $T = n\Delta t$
2. **Sample** $N = b^m, m \geq 1$ *particles from the given initial distribution*
3. **Initialize** $n \leftarrow 0$
4. **Transport** *all the* N *particles along the integral curves of* $\boldsymbol{b}$
5. **Sort** *the* N *particles according to the magnitude of the first coordinate of the particle positions*
6. **Diffusion**
   for $i = 1, N$
   (i) *Produce a* $(0, m, s+1)$*-net* $Y = (y_1, \ldots, y_{s+1})$
   (ii) $\lfloor b^m y_1 \rfloor$ *yields a particle number* P
   (iii) *Use the remaining* s *coordinates of the generated sequence to increment the coordinates of the* P*th particle by* $\mathcal{N}(0, 2\Delta t)$ *variables*
   end;
7. $n \leftarrow n + 1$
8. if *(*$n\Delta t < T$*)*
   goto *4*
   else
   stop

CHAPTER 5

Numerical Results

In this chapter we summarize the results of various numerical simulations that have been carried out using different approaches. We mainly compare the computations carried out on the algorithms presented in [**15**] and [**16**] with the one developed in this thesis. All the computations are done on a AMD Athlon 1400 MHz machine with 1.5GB memory running Debian Linux 3.0. The CPU time we shall refer to is as measured on this machine. The complete implementation is done in ANSI C language.

In our computations we take as $(t, m, s) - net$ in base b the Faure sequence [**5**]. Construction of such nets have been proposed for example by Sobol [**28**], Faure [**5**], Niederreiter [**19**]. Generally it takes $O(m^2 s)$ time to generate a point using the straightforward algorithm. The method proposed by Antonov and Saleev to generate a $(t, m, s) - net$ in base 2 requires only $O(ms)$ time. This method was generalized by Eric Thiémard [**30**] to an arbitrary base b based on the idea presented in [**29**]. The calculation uses integer arithmetic thereby guaranteeing high precision. For the Monte Carlo simulation, we use the Unix inbuilt random number generator function $drand48$.

For the sorting, we use the *quicksort* algorithm proposed by Hoare, [**26**]. Quicksort is based on the method of recursion and is the fastest known algorithm requiring about $O(n \log n)$ steps (for a large number of elements) in contrast to the other sorting rules like insertion sort, bubble sort which require $O(n^2)$ steps. The worst case for quicksort corresponds to already sorted data in which it takes $O(n^2)$ steps.

The chapter is organized as follows. In the first section we review the superiority of QMC over MC both in accuracy and computational time for sampling initial values. The second section deals with the problem of plain diffusion in high dimensions. In the third section we present results of the simulation carried out on the Fokker-Planck equation derived earlier.

5.1. Integration

As a first task, we compare the time taken by the Faure generator and the

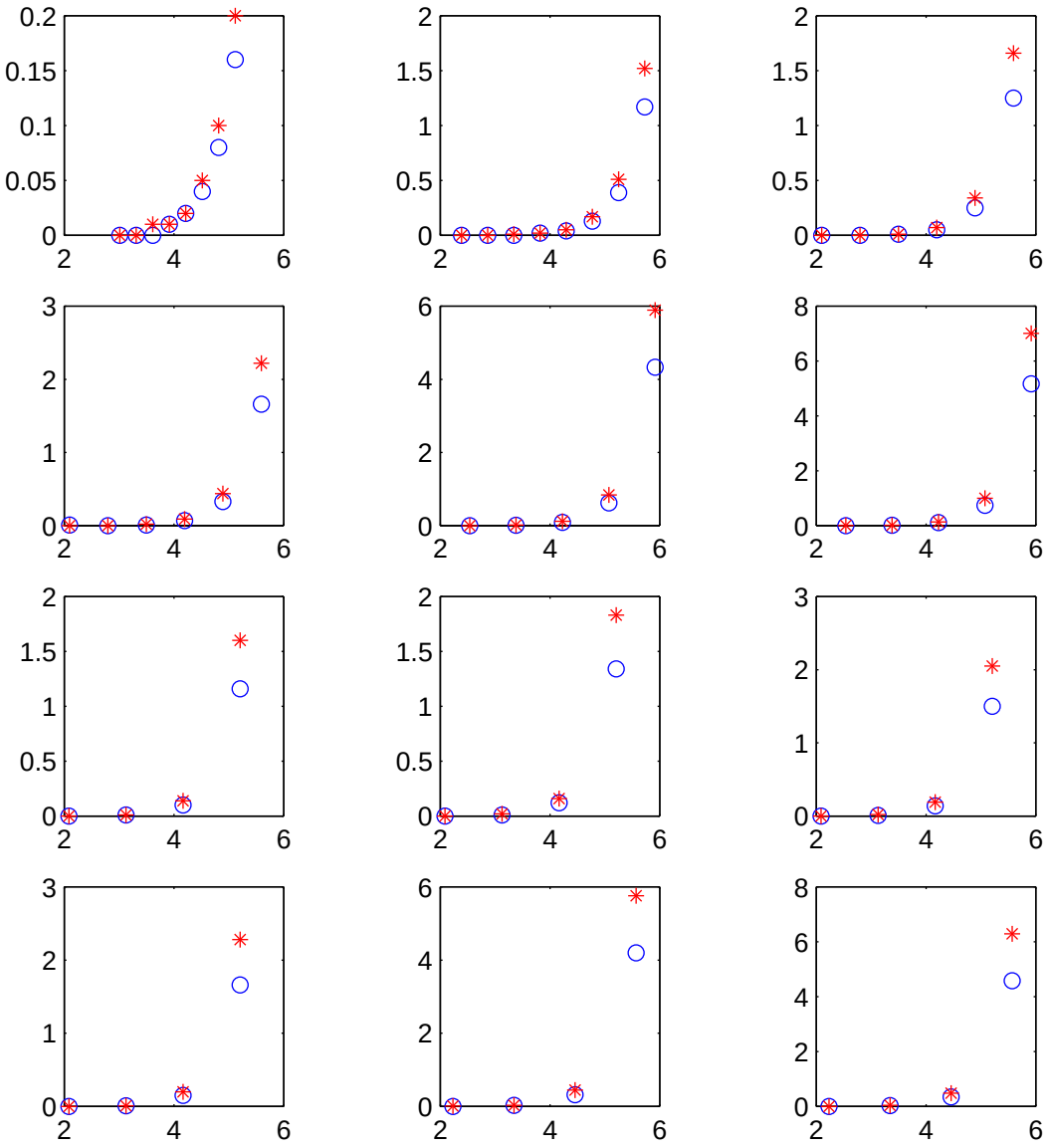


FIGURE 5.1. Time taken to sample particles distributed according to standard normal for dimensions 1 to 12 (the first row refers to dimensions 1,2 and 3, the second row 4, 5 and 6 and so on). The stars correspond to MC whereas the circles correspond to QMC. The x-axis is set to log scale and denotes the total number of particles considered; the y axis shows the CPU time taken in seconds.

random generator to sample particles distributed according to standard normal distribution in various dimensions. We use the inversion technique outlined earlier for sampling. Since the standard normal distribution, $\mathcal{N}(0, I)$, is radial, the inversion technique can be applied to each coordinate to produce a vector of size equal to the given dimension distributed according to $\mathcal{N}(0, I)$. It is clear from figure 5.1 that it is indeed economical to generate the Faure points even in high dimensions.

Having seen that it is faster to generate and use QMC points, we now show that they approximate the sampled function better than MC points. Consistent with later use, we consider sampling from the standard normal distribution $f(x)$. If we take $\mathcal{B}$ as the set of 1000 boxes having a random center and random length uniformly distributed in (0,1) and define

$$D_N^{1000} = \sup_{B \in \mathcal{B}} \left| \frac{A(B; P)}{N} - \int_B f \right| \tag{5.1.1}$$

where P is the set of points sampled from f, then for a fixed dimension, D_N^{1000} can be calculated for each N. A parameter α is fit in such a way that

$$D_N^{1000} = CN^{\alpha} \tag{5.1.2}$$

in the sense of least squares. We see from figure 5.2 that the order of convergence of QMC is significantly better than MC.

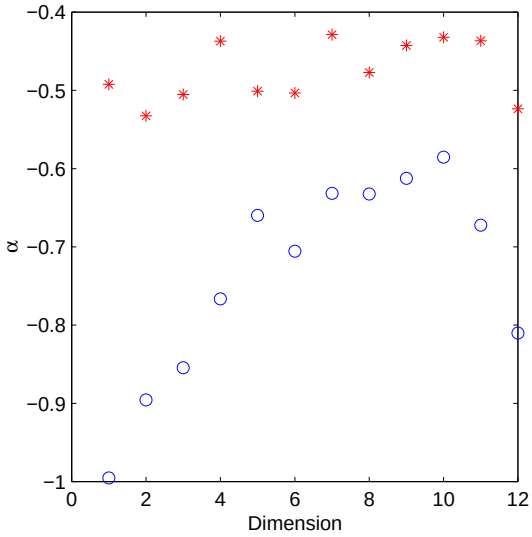


FIGURE 5.2. Initial sampling error. Value of α in MC (stars) and QMC (circles) simulations restricting to a maximum of 10^5 particles.

Remark 3. *It is to be noted that the rate of convergence we get by the least squares fit is dependent heavily on the data under consideration. Especially for the strongly fluctuating results obtained with MC and QMC, an extra data point can improve or worsen the order of convergence. For the data set presented in figure 5.3 for example, the order of convergence is reduced by including the error value corresponding to the largest particle number.*

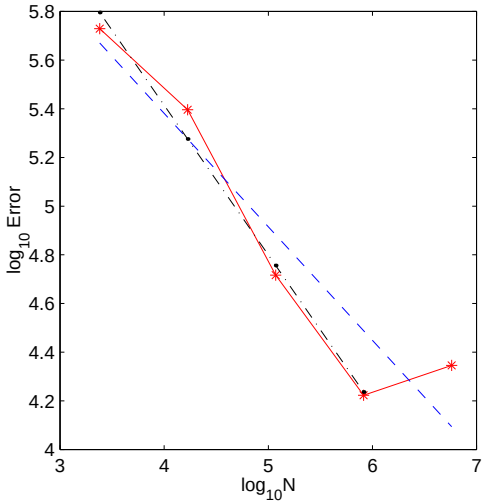


FIGURE 5.3. The data set is represented by the solid line. The least square fit of the data is shown by the dashed line and the same for the data without the last point included is shown by the dash dot line.

Since the particle numbers cannot be freely chosen in our QMC algorithm for diffusion problems, the size of the data set used for fitting is eventually restricted by memory limitations. Thus the estimated convergence order is not of high precision and it should just give an indication of the general behavior of the algorithm.

5.2. Plain diffusion

We now consider the plain diffusion problem in dimension s.

$$u_t = \Delta u$$
$$u(\boldsymbol{x}, 0) = \prod_{i=1}^{s} \frac{1}{\sqrt{\pi}} \exp(-x_i^2) \tag{5.2.3}$$

The reason for choosing this problem is that the analytical solution can be calculated quite easily and this is helpful in comparing our numerical results. The exact solution for this problem can be written down as the convolution of the Gauss kernel $G(\boldsymbol{x}, t)$ with the initial value $u(\boldsymbol{x}, 0)$. Thus

$$u(\boldsymbol{x}, t) = \frac{1}{\sqrt{\pi(1+4t)}} \exp\left(-\frac{\|\boldsymbol{x}\|^2}{(1+4t)}\right) \tag{5.2.4}$$

The nature of the exact solution facilitates calculating moments of the form $\int_{\mathbb{R}^d} \|\boldsymbol{x}\|_2^\gamma u(\boldsymbol{x}, t) d\boldsymbol{x}$, where $\|\boldsymbol{x}\|_2$ denotes the 2-norm of $\boldsymbol{x}$. In fact, we have for $\gamma \in \mathbb{N}$, with $\beta = \sqrt{\frac{1+4t}{2}}$

$$\int_{\mathbb{R}^s} \|\boldsymbol{x}\|_2^\gamma u(\boldsymbol{x}, t) d\boldsymbol{x} = \begin{cases} 0 & \text{if} \quad \gamma = 2\mu + 1 \\ \beta^{2\mu} \prod_{m=0}^{\mu-1} (s + 2m) & \text{if} \quad \gamma = 2\mu \end{cases} \tag{5.2.5}$$

We now compare our implementation with the one described in [**15**] and Monte Carlo. The first aspect in this regard would be to check the accuracy of the methods. With the same notion of discrepancy explained earlier, we calculate α for dimensions 1 to 8 taking 10 time steps of 0.0001 each.

Dim	MC	QMC	QMC [**15**]
2	-0.56	-0.47	-0.58
3	-0.45	-0.44	-0.59
4	-0.57	-0.49	-0.64
5	-0.48	-0.63	-0.56
6	-0.47	-0.54	-0.57
7	-0.64	-0.59	-0.62
8	-0.44	-0.53	-0.57

TABLE 5.1. Simulation of diffusion.

From table 5.1 we conclude that our method nearly obeys the estimate we have established earlier. One can also observe that the algorithm QMC [**15**] outperforms both MC and QMC. In view of the error estimate in [**15**], this is actually not expected. Choosing, for example, $N = b^{ks}$ and $d_i = k$ for all i, the estimate predicts a convergence like $1/N^{1/2s}$. The d_i's in our calculation have been chosen as shown below for the case $s = 4$.

d_1	d_2	d_3	d_4	$N = b^{\sum d_i}$
1	1	0	0	121
1	1	1	0	1331
1	1	1	1	14641
2	1	1	1	161051

TABLE 5.2. Various combinations of d_i's used in our computation.

Regarding the computational time we start with a comparison of MC and QMC again doing 10 steps of 0.0001 each. The results are summarized in figure 5.4. It is clear from figure 5.4 that QMC is faster compared to MC especially in high dimensions which is in accordance with the results in section 5.1.

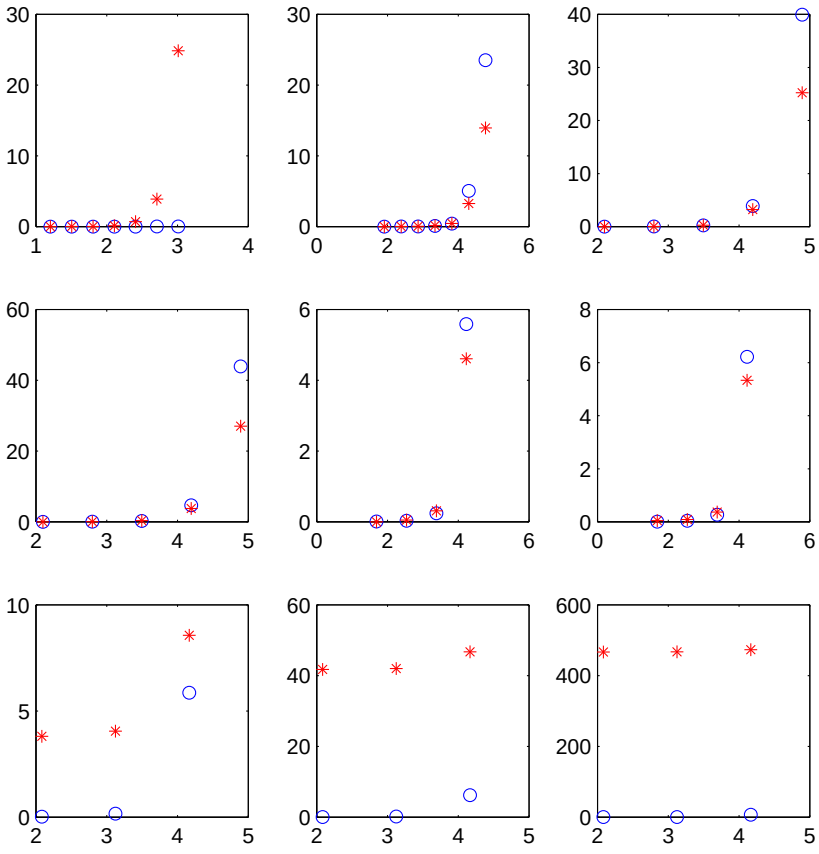


FIGURE 5.4. CPU time required by MC and our method for ten steps of plain diffusion for dimensions 1 to 9. The stars correspond to MC whereas the circles correspond to QMC. The x-axis is set to log scale.

A comparison between QMC and QMC [**15**] is not so straight forward, because the two methods work on completely different particle numbers. For a problem in s dimension, we just consider a $s+1$ dimensional Faure sequence in base b, the least prime number $\geq (s+1)$, whereas in [**15**], Faure sequence of dimension $2s$ in base b, the least prime $\geq (2s+1)$ is considered. So we cannot estimate the time required to carry on the computation with a fixed number of particles.

The main difference between QMC and QMC [**15**] is in the sorting rule. So the difference in computational times can be viewed as the time to sort the particles according to the two implementations. From figure 5.5, it is clear that multi-index sorting takes considerably much more time compared to sorting only along one dimension. Observe that for $1d$ case both the methods coincide. The percentage of sorting time in QMC can be estimated by comparing figures 5.4 and 5.5.

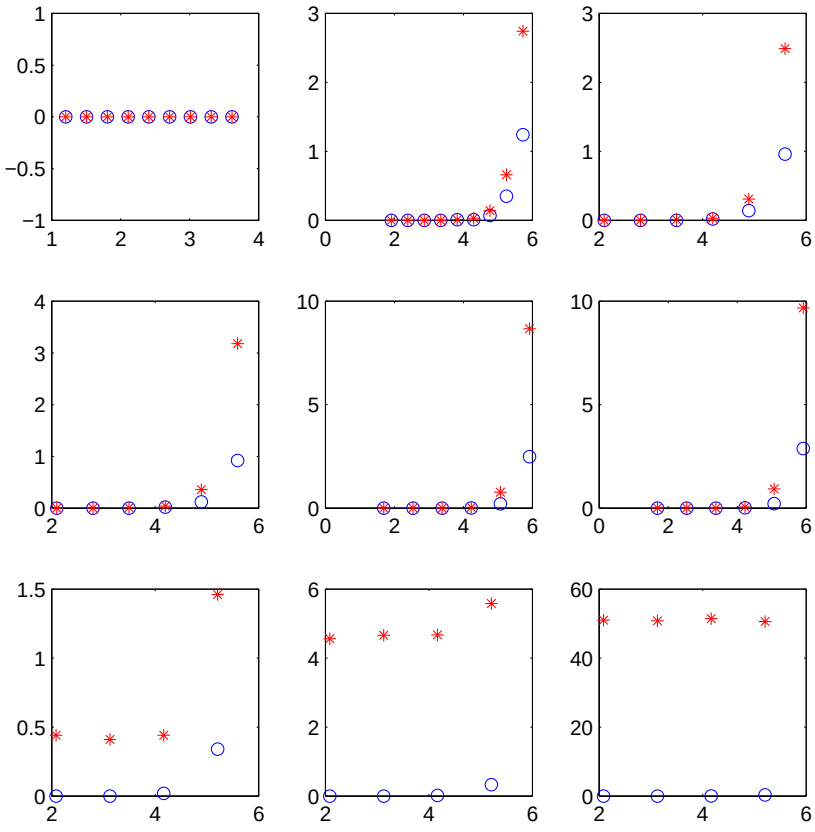


FIGURE 5.5. Comparison between sorting with respect to the first coordinate and full sorting as in [**15**] for dimensions 1 to 9. The stars correspond to QMC [**15**] whereas the circles correspond to QMC. The x-axis is set to log scale.

Since we are finally interested in calculating certain second order functionals of our original Fokker-Planck equation, we now proceed calculating the same for the simple diffusion problem.

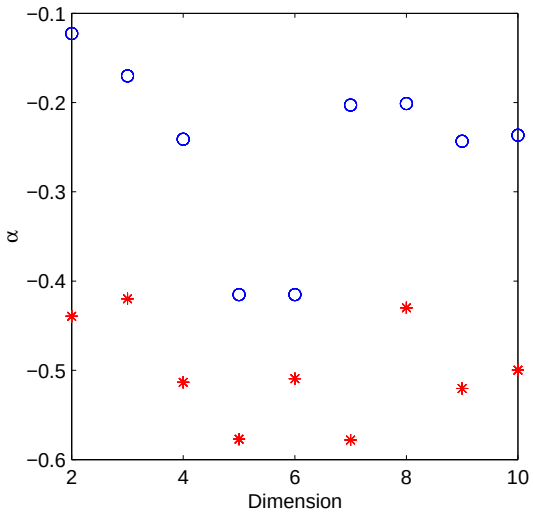


FIGURE 5.6. Convergence of MC and our method for dimensions 2 to 10 in calculating the second order functionals. 10 time steps with $\Delta t = 0.1$ were taken. The circles correspond to QMC whereas the stars correspond to MC. The convergence order of the worst case error is plotted.

By worst case error we mean the maximum error taking all the second order moments

$$\int_{\mathbb{R}^s} x_i x_j u(\boldsymbol{x}, t) d\boldsymbol{x} \qquad i, j = 1, 2, \ldots, s \tag{5.2.6}$$

into consideration.

It is clear from figure 5.6 that MC outperforms QMC quite considerably. This is not surprising in so far as the Koksma-Hlawka inequality cannot be applied to the present situation for the variation of the quadratic functions is infinite. In order to understand the dismal performance of QMC, let us consider a 1d case. We consider just 64 particles (figure 5.7) and allow them to undergo diffusion until time $t = 1$ with a time step of 0.01. It is seen that while MC particles remain confined, a few QMC particles drift away with considerable velocity. The reason for the particles drifting away is obviously due to correlation among the QMC points. Since the quadratic functionals give high weight at large distances, it now seems natural that the non-diffusive behavior of the far out

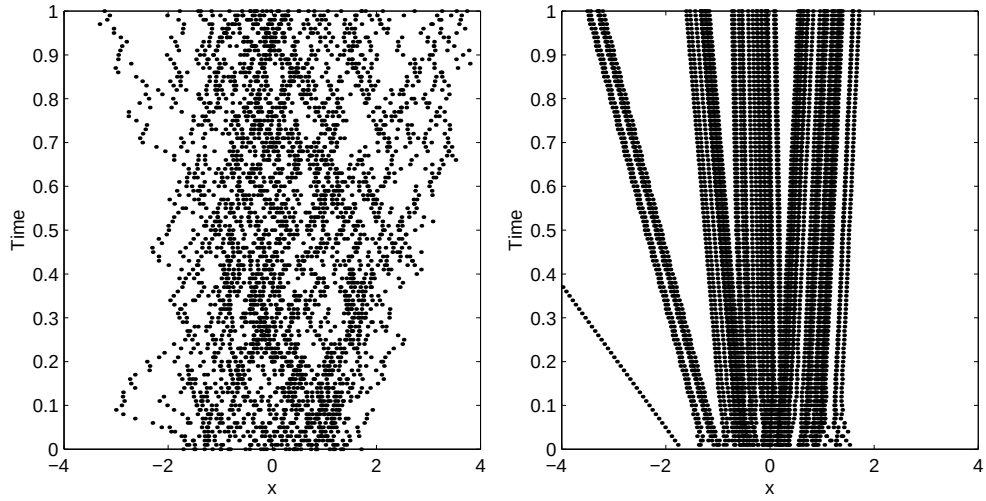


FIGURE 5.7. Particles performing diffusion in 1-d. MC to the left and QMC to the right.

particles spoils the convergence order. Consequently, one expects even worse convergence orders by taking higher order moments of the solution like

$$\int_{\mathbb{R}^s} \|\boldsymbol{x}\|_2^{10} u(\boldsymbol{x}, t) d\boldsymbol{x}. \tag{5.2.7}$$

However, these expectations are not satisfied as can be seen from figure 5.8,

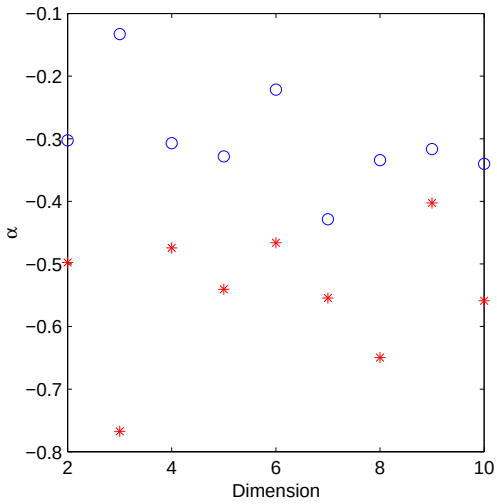


FIGURE 5.8. Convergence of MC and our method for dimensions 2 to 6 in calculating $\int \|\boldsymbol{x}\|_2^{10} u(\boldsymbol{x}, t) d\boldsymbol{x}$. The circles correspond to QMC whereas the stars correspond to MC. The worst case error is plotted.

and we have to leave the question concerning the reduction of convergence order unanswered.

At this point it might be disappointing to apply the method for our Fokker-Planck problem, in which we have to evaluate the moment functional with an unbounded weight function. It turns out however, that the algorithm works remarkably well. In view of the comments above, this could be explained by the presence of the additional drift term in the equation (spring forces) which counteract diffusion and suppress particles drifting away.

5.3. Numerical simulation of Fokker-Planck equation

We are interested in the steady state solution of

$$\frac{\partial \psi}{\partial t} = -\sum_j \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \left(\boldsymbol{\kappa} \boldsymbol{Q}_j - \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \right) \psi + \sum_j \frac{\partial}{\partial \boldsymbol{Q}_j} \cdot \frac{\partial \psi}{\partial \boldsymbol{Q}_j} \tag{5.3.8}$$

subject to suitable initial condition in order to evaluate the stress tensor,

$$\boldsymbol{\tau}^p = -\sum_j \int_{\mathbb{R}^{3(N-1)}} \boldsymbol{Q}_j \otimes \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \psi d\boldsymbol{Q} \tag{5.3.9}$$

and calculate the viscosity η and first normal stress difference coefficients given by (2.5.53) and (2.5.54) respectively. In other words this means to say that we would like to evaluate,

$$\boldsymbol{\tau}^p = -\lim_{t \to \infty} \sum_j \int_{\mathbb{R}^{3(N-1)}} \boldsymbol{Q}_j \otimes \frac{\partial \phi}{\partial \boldsymbol{Q}_j} \psi d\boldsymbol{Q} \tag{5.3.10}$$

Numerically we do it as follows. Consistent with algorithm 4.1, P particles are sampled from the initial distribution and they undergo transport and diffusion according to the dynamics of the equation. The material functions η and Ψ are calculated at every time step till steady state is reached.

A typical plot of viscosity and first normal stress difference coefficient is as shown in figure 5.9 (taken from the dumbbell case with 5^6 particles). However, in the higher dimensional case there is an initial spike in the values of viscosity and first normal stress difference coefficient as can be seen from 5.10. This just means that one needs to run the simulation for a longer time till the stationary values are obtained. The following figure 5.11 shows the results of simulation run up to time $T = 500$ in the case of 8 beads and the stationary situation can be observed.

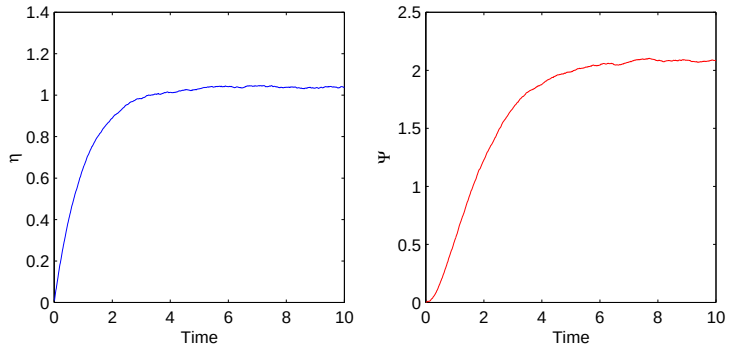


FIGURE 5.9. Typical plot of viscosity (left) and first normal stress difference coefficient (right) versus time.

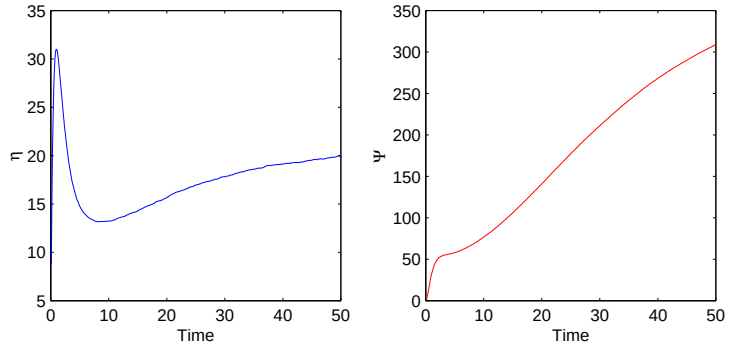


FIGURE 5.10. Plot of viscosity (left) and first normal stress difference coefficient (right) versus time showing initial spike in the case of 8 beads.

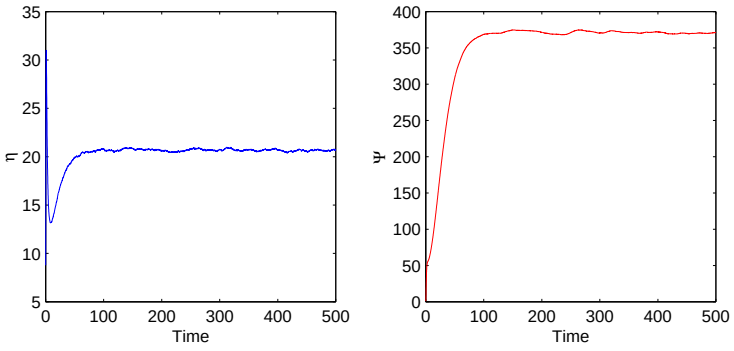


FIGURE 5.11. Plot of viscosity (left) and first normal stress difference coefficient (right) versus time showing stationary situation in the case of 8 beads.

Remark 4. *It can be observed from the expression for* $\boldsymbol{\tau}^p$*, that the tensor is symmetric. The only property which needs to be verified is the equality of* τ^p_{yy} *and* τ^p_{zz}*. It is observed that the relative difference in their values remain within 4%.*

Now we study the dependence of the steady state values of η and Ψ on the initial conditions. We consider the following three cases with reference to the dimensional variables, but we do suitable transformations to adapt to non-dimensional form.

1. The position of the connector vectors are independent Gaussian distributed with mean zero and variance one.
2. We start from a delta distribution, that is to say that all beads are on top of each other at the origin
3. We take $\psi(\boldsymbol{Q}, 0) = \psi_{eq}$, that is we start with the equilibrium steady state solution of the flow problem corresponding to fluid at rest.

In the first case, we do the usual inversion technique to generate the standard normal variable and use the transformation Z introduced in subsection 2.5.2. For the last case, we use the acceptance rejection technique to sample according to ψ_{eq}. The first result is marked with an SN to represent sampling from normal distribution, the second with DD to signify the delta distribution and the last with EQ to emphasize equilibrium distribution. From figure 5.12 and 5.13, it

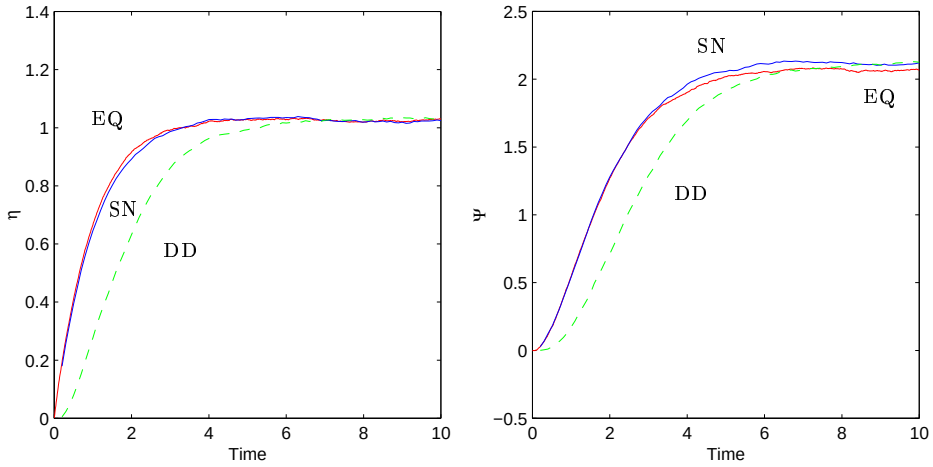


FIGURE 5.12. Non-dimensional viscosity (left) and first normal stress difference coefficient (right) for different choices of initial condition for the case of dumbbell.

is clear that as expected the convergence is not affected by the choice of initial condition. But a careful examination shows that steady state is attained ahead of time in the case of choices 1 and 3 compared to 2. The only disadvantage of choice 3 is that, many particles should be produced before a required number is selected. In the case of dumbbells only 10% of the particles are accepted on an average. So we conclude that choice 1 is the optimal choice both accuracy wise and computational cost wise.

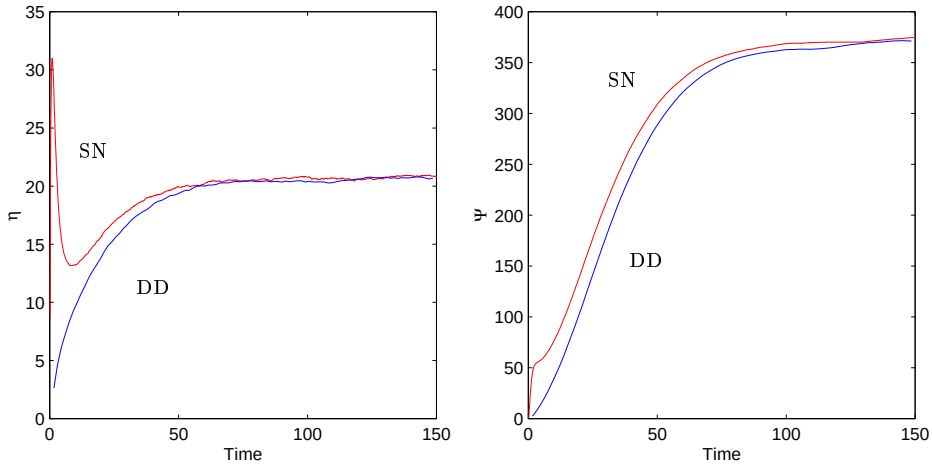


FIGURE 5.13. Non-dimensional viscosity (left) and first normal stress difference coefficient (right) for choices 1 and 2 of initial condition for the 8 beads case.

Now that we have fixed the initial condition, we have all the ingredients to compare MC and QMC. Figure 5.14 (left) shows two independent runs of the

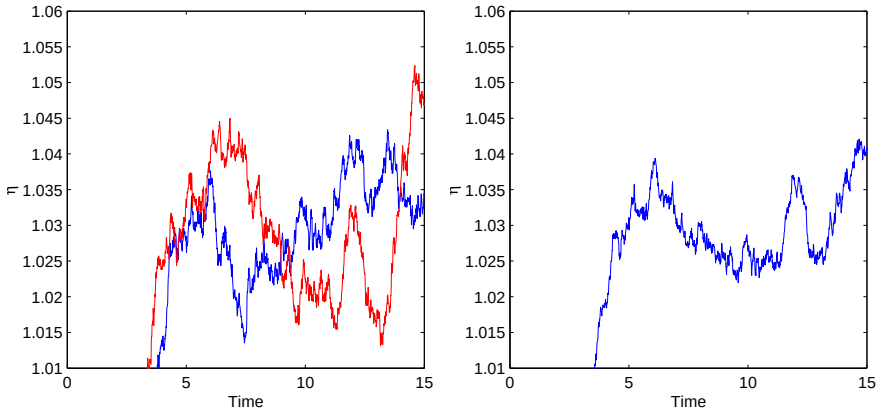


FIGURE 5.14. Monte Carlo simulation to calculate the viscosity.

Monte Carlo algorithm for calculating the viscosity. The average values obtained from the two trajectories is shown in figure 5.14 (right). Note that we have chosen the η scale to show the behavior more clearly in the stationary part of the curve.

The trajectory obtained using our scheme, (figure 5.15), is quite smooth compared to the Monte Carlo simulation meaning that the stationary value is well attained at a earlier time.

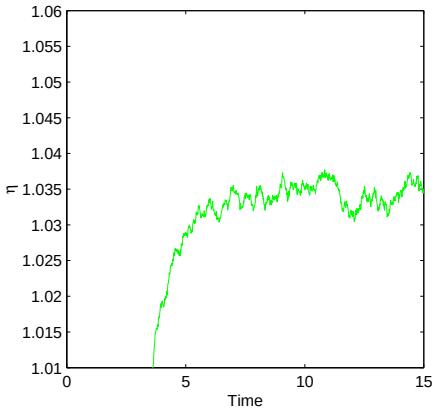


FIGURE 5.15. Quasi-Monte Carlo simulation to calculate the viscosity

A similar behavior is observed also in the case of first normal stress difference coefficient as shown in figure 5.16.

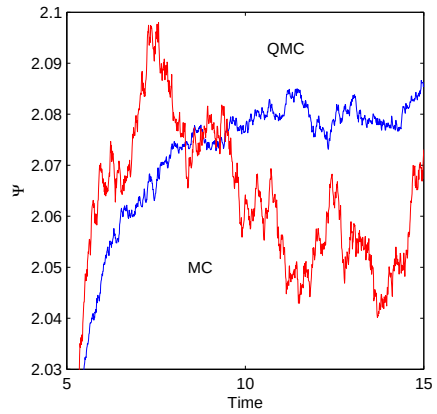


FIGURE 5.16. Comparing MC and QMC simulation in calculating the first normal stress difference coefficient.

For a single run with 5^7 particles, the Monte Carlo method took 2275 seconds whereas our scheme took about 4448 seconds. Since we need to average values

over several trajectories, to get a trajectory as smooth as in the case of QMC, we conclude that our scheme works significantly better both in accuracy and computational time.

Since we are interested in the steady state values of the viscosity and the first normal stress difference coefficient, we would now study the influence of the time step on the stationary value of our method. Again as a test case we take the dumbbell case with 5^6 particles. Three different time steps, namely 0.01, 0.1 and 1.0 are considered. From figure 5.17, it is evident that the difference in the values obtained using $\Delta t = 0.1$ and $\Delta t = 0.01$ is less than 2%. Also the case $\Delta t = 0.01$ takes ten times more time compared to $\Delta t = 0.1$. In view of these facts, we conclude that it is judicious to consider $\Delta t = 0.1$ for the simulations.

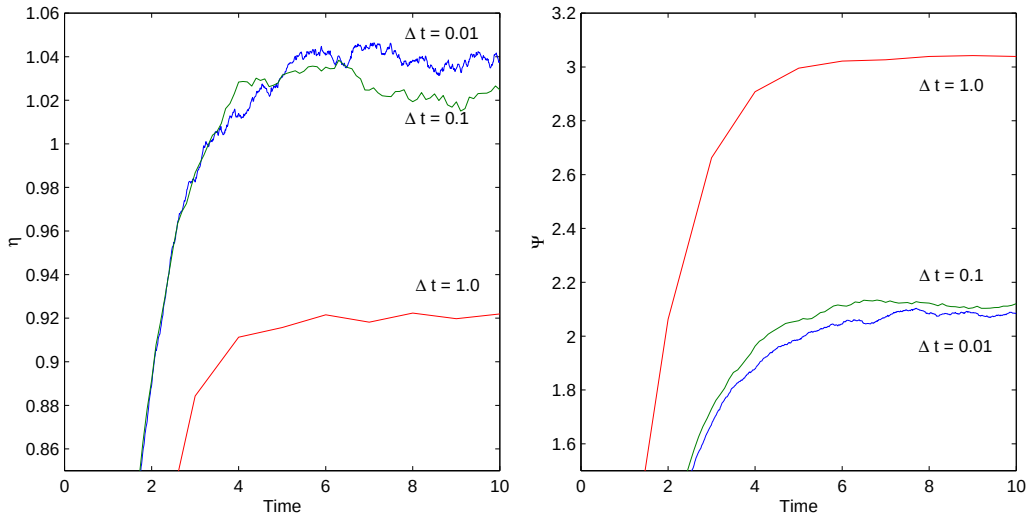


FIGURE 5.17. Time accuracy of the scheme for calculating viscosity (left) and first normal stress difference coefficient (right).

Having studied the time dependence with a fixed number of particles, we now study the dependence of the viscosity and first normal stress difference coefficient on the sample size. We consider the dumbbell case and march with a time step of 0.1 till time $T = 10.0$ with 5^5, 5^6 and 5^7 particles. It can be seen from figure 5.18 that as the particle number increases, the oscillations decline in magnitude and steady state is attained ahead in time.

At this stage, we have the following situation: MC and QMC both work for high dimensions, the former can be implemented in a straightforward manner whereas the latter requires sorting and shuffling the particle positions at each

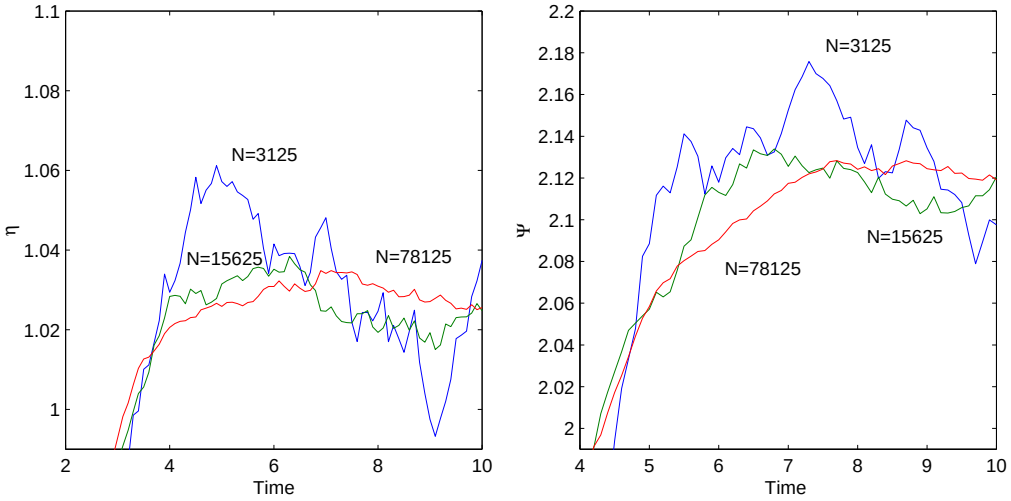


FIGURE 5.18. Dependence of viscosity (left) and first normal stress difference coefficient on the sample size.

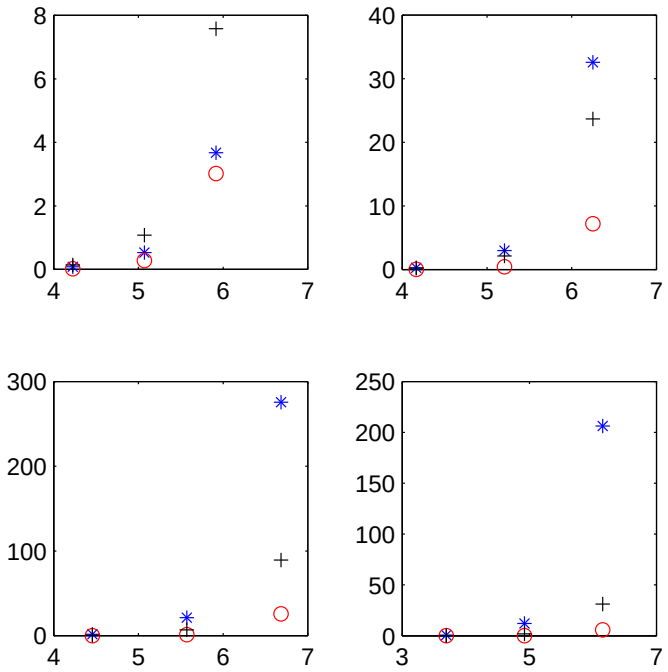


FIGURE 5.19. Time taken for transport (stars), sorting (circles) and diffusion (+) per time step for 3, 4, 5 and 6 beads. The x axis shows the particle numbers in log scale and y axis, the CPU time in seconds.

time step. The advantage is that QMC results have less noise compared to

MC, but the extra processes may take up additional time. But what we observe (figure 5.19), is that this does not contribute significantly as transport dominates the total computational time. Though the time taken for diffusion is quite significant in the case of 3 beads, it is overtaken by transport in higher dimensions (4, 5 and 6 beads).

As the last task, we compare our implementation with that of [**16**]. As in the case of diffusion, the error estimate does not converge if, for example, we consider some of the d_is to be equal to zero. This leads us to consider minimal particle numbers of the form b^s. So we could address only the dumbbell case with this algorithm for otherwise the particle numbers grow very large. Figure 5.20 shows the time evolution of η and Ψ by both the methods in the dumbbell case.

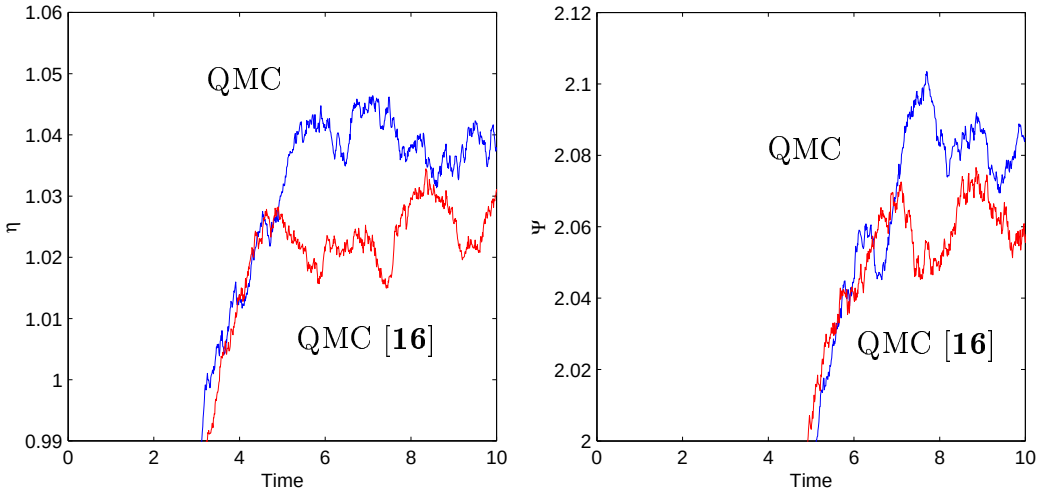


FIGURE 5.20. Comparison between QMC and QMC [**16**]. 5^6 particles were considered and a time step of 0.01 was chosen. For QMC [**16**], Δx was taken to be such that $2s\Delta t/\Delta x^2 \leq 1/25$ to satisfy the stability condition.

We now take up the benchmark case, i.e, dumbbell case, to compare with the results presented in [**22**]. Various combinations of z, β and d are taken and the non-dimensional viscosity and first normal stress difference coefficient are calculated. A single computation took about 476 seconds. The results shown in figures 5.21 and 5.22 are in accordance with the ones reported in [**22**].

We conclude our discussion with simulation results for a real high dimensional case. We consider the case of 10 beads, that is a 27 dimensional Fokker-Planck

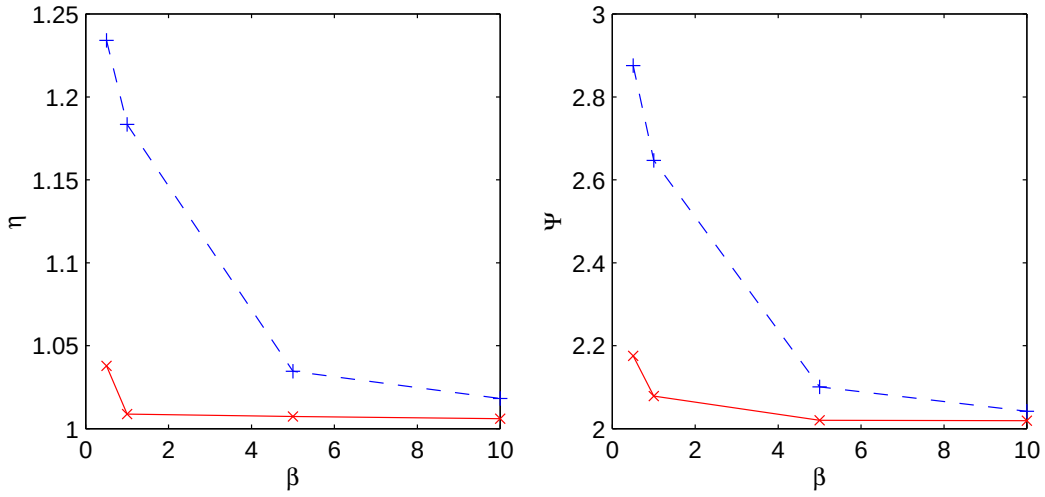


FIGURE 5.21. Non-dimensional viscosity (left) and first normal stress difference coefficient (right) for a fixed $d = 2.5$. The broken line corresponds to $z = 30$ and the solid line corresponds to $z = 3$

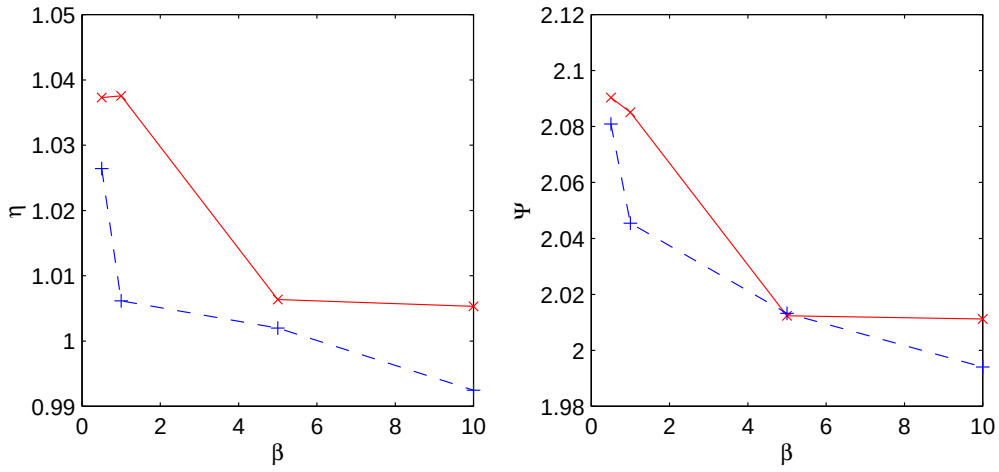


FIGURE 5.22. Non-dimensional viscosity (left) and first normal stress difference coefficient (right) for a fixed $z = 0.1$. The broken line corresponds to $d = 1.0$ and the solid line corresponds to $d = 0.5$

equation. A total of 29^3 particles were considered and we march with a time step of 0.5 seconds till time $T = 250$ seconds. The average CPU time taken

for a time step was about 107 seconds and η and Ψ were calculated after every 5 time steps.

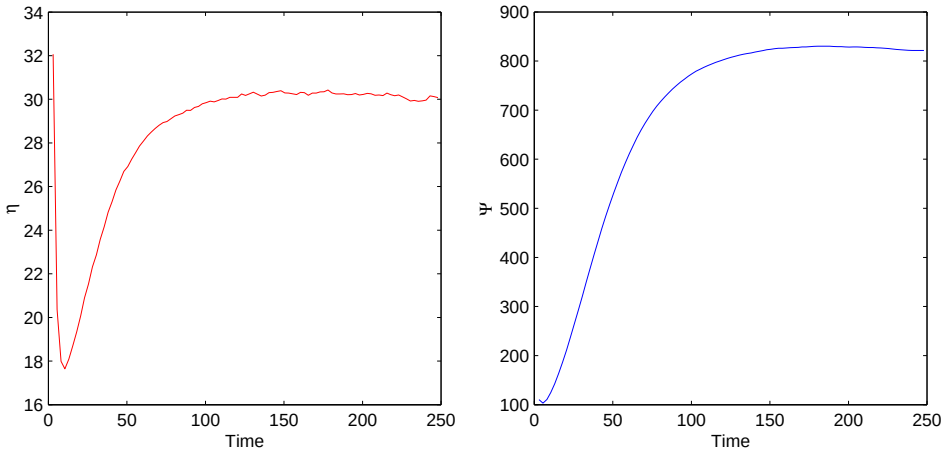


FIGURE 5.23. Non-dimensional viscosity (left) and first normal stress difference coefficient (right) for the 10 beads case.

Conclusions

With the method developed in this thesis, the two important material functions of a dilute polymer solution undergoing shear flow, namely the viscosity η and the first normal stress difference coefficient Ψ can be calculated for larger number of beads subjected to the memory restrictions. In the two beads case, the decline of viscosity with increasing shear rate β is verified and this confirms the experimental result presented in subsection 1.0.1. The quantity Ψ is non zero and this explains the rod climbing phenomenon described in subsection 1.0.2.

In the course of our analysis, we have compared the effectiveness of MC and QMC in integration, diffusion and Fokker-Plank equation for high dimensional problems. The scenario is mixed and is summarized in the following table.

	Integration	Diffusion	Fokker-Planck equation
MC	-	+	-
QMC	+	-	+

TABLE 5.3. Comparing MC and QMC for different applications.

The superiority of QMC over MC in plain integration is well known. We discuss it in section 5.1, where it is also seen that QMC method is faster compared to MC. The better order of convergence (eventually $1/N$ versus $1/\sqrt{N}$) however, does not carry over to plain diffusion problems due to the correlation among QMC points. In fact, if integral functionals with weight functions of unbounded variation like quadratic, biquadratic of the solution are considered, QMC performs worse than MC. This may be due to the fact that because the particles are drifting away, the errors are amplified by the weight function. Fortunately for our application, though we also consider functionals with unbounded variation, due to an extra drift term (spring forces), the QMC particles do not move away so much and we achieve improvement over MC. Regarding the computational time for our application, the sorting time does not contribute significantly to the total time since the former is dominated by advection. Thus we achieve improvement both in accuracy and computational time using our method.

The scheme presented in this thesis is based on the one presented in [**15**]. However compared to the methods in [**15**] and [**16**] our method works in real

high dimensions with feasible particle numbers, thus allowing us to consider large number of beads to carry out simulations of polymer models.

Bibliography

[1] R. Byron Bird, Charles F. Curtiss, Robert C. Armstrong and Ole Hassager, Dynamics of Polymeric liquids, Vol I, Wiley Interscience, 1987.

[2] R. Byron Bird, Charles F. Curtiss, Robert C. Armstrong and Ole Hassager, Dynamics of Polymeric liquids, Vol II, Wiley Interscience, 1987.

[3] E. A. Coddington and N. Levinson, Theory of ordinary differential equations, McGraw-Hill, New York, 1955.

[4] L. Devroye, Non-uniform random variate generation, Springer, New York, 1986.

[5] H. Faure, Discrépance de suites associées à un systéme de numération (en dimension s), *Acta Arith.*, **41**, 1982, pp. 337-351.

[6] P. J. Flory, Statistical mechanics of chain molecules, Wiley-Interscience, New York, 1969, p. 13.

[7] A. Friedman, Partial differential equations of parabolic type, Englewood Cliffs Prentice-Hall, New York, 1964.

[8] T. Gerstner and M. Griebel, Numerical integration using sparse grids, Numerical algorithms **18**, pp. 209-232, also as Sonderforschungsbereich 256, Universität Bonn, 1998.

[9] H.A. Kramers, *Physica*, **11**, 1944, pp. 1-19.

[10] Kalos and Whitlock, Monte Carlo methods, Wiley, 1986.

[11] D. Kincaid, W. Cheney, Numerical analysis, Brooks/Cole Publications, 1996, pp. 517.

[12] L. Kuipers and H. Niederreiter, Uniform distribution of sequences, Wiley-Interscience, New York, 1974.

[13] C. Lécot, Low discrepancy sequences for solving the Boltzmann equation, *Journal of Comp. and Applied Math.*, **25**, 1989, pp. 237-249.

[14] C. Lécot and I. Coulibaly, Simulation of diffusion using quasi-random walk methods, *Mathematics and Computers in Simulation*, **47**, 1998, pp. 153-163.

[15] C. Lécot and F. E. Khettabi, Quasi-Monte Carlo simulation of diffusion, *Journal of Complexity*, **15**, 1999, pp. 342-359.

[16] C. Lécot and W. Ch. Schmid, Particle approximation of convection-diffusion equations, *Mathematics and Computers in Simulation*, **55**, 2001, pp. 123-130.

[17] W. J. Morokoff and R. E. Caflisch, A quasi-Monte Carlo approach to particle simulation of the heat equation, *Siam J. Numer. Anal.*, Vol **30**, No. 6, 1993, pp. 1558-1573.

[18] J. von Neumann, Various techniques used in connection with random digits, *U.S. Nat. Bur. Stand. Appl. Math. Ser.*,**12**, 1951, pp. 36-38.

[19] H. Niederreiter, Random number generation and quasi-Monte Carlo methods, *Society for Industrial and Applied Mathematics* - VI 1992.

[20] E. Novak and K. Ritter, The curse of dimension and a universal method for numerical integration, *Multivariate approximation and splines*, G. Nüneberger, J.W. Schmidt, G. Walz (eds.), 1998.

[21] H.C. Öttinger, Stochastic Processes in Polymeric Fluids, Springer, 1996.

[22] J. Ravi Prakash and H. C. Öttinger, Viscometric functions for a dilute solution of polymers in a good solvent. *Macromolecules*, Volume **32**, Number *6*, 1998, pp. 2028-2043.

[23] Risken, The Fokker-Planck Equation, Methods of Solution and Applications, Springer-Verlag, 1984.

[24] P.E. Rouse, J. Chem. Phys.,**21**, 1953, pp. 1272-1280.

[25] R. Y. Rubinstein, Simulation and the Monte Carlo method, Wiley, 1981.

[26] R. Sedgewick, Algorithms in C, Reading, Mass. : Addison-Wesley, 1990.

[27] S.A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, *Dokl. Akad. Nauk SSSR* **4**, 1963, pp. 240-243.

[28] I. M. Sobol, On the distribution of points in a cube and the approximate evaluation of integrals, *USSR Comput. Math. Math. Phys*, 1967.

[29] S. Tezuka, Uniform random numbers: Theory and practice, Kluwer Academic Publishers, Boston, 1995.

[30] E. Thiémard, Economic generation of low-discrepancy sequences with a b-ary Gray code, *EPFL-DMA-ROSO, RO981201.*

[31] E. Zeidler, Nonlinear functional analysis and its applications II/B, Springer-Verlag, New York, (1990).

Curriculum Vitae

19th May 1974	Born in Madras, Tamilnadu, India
1977-1990	Secondary Education Vivekananda Vidyalaya, Madras, India
1990-1992	Senior Secondary Education DTEA Senior Secondary School, New Delhi, India
1992-1995	Bachelor of Science (Honours) in Mathematics University of Delhi, New Delhi, India
1995-1997	Master of Science in Mathematics Indian Institute of Technology, Madras, India
1997-1999	Master of Science in Industrial Mathematics University of Kaiserslautern, Germany
1999-present	Doctoral Student in AG Technomathematik University of Kaiserslautern, Germany