



CHALMERS
UNIVERSITY OF TECHNOLOGY



Long–time behaviour of Hamiltonian PDEs driven by an additive one dimensional standard Brownian motion and their numerical discretisations

Master's thesis in Engineering Mathematics and Computational Science

Sandy Abonaser
Homam Mowed

DEPARTMENT OF Mathematical Sciences

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

MASTER'S THESIS 2026

**Long-time behaviour of Hamiltonian PDEs
driven by an additive one dimensional standard
Brownian motion and their numerical
discretisations**

Sandy Abonaser
Homam Mowed



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Long-time behaviour of Hamiltonian PDEs driven by an additive one dimensional standard Brownian motion and their numerical discretisations

Sandy Abonaser

Homam Mowed

© Sandy Abonaser, Homam Mowed 2026.

Supervisor: David Cohen, Department of Mathematical Sciences

Examiner: David Cohen, Department of Mathematical Sciences

Master's Thesis 2026

Department of Mathematical Sciences

Division of Applied Mathematics and Statistics

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Typeset in L^AT_EX

Printed by Chalmers Reproservice

Gothenburg, Sweden 2026

Long-time behaviour of Hamiltonian PDEs driven by an additive Brownian motion and its numerical discretisations

Sandy Abonaser

Homam Mowed

Department of Mathematical Sciences

Chalmers University of Technology

Abstract

This thesis investigates the long-time behaviour of Hamiltonian partial differential equations driven by an additive one dimensional standard Brownian motion. The study aims to understand how random fluctuations affect fundamental properties, mostly in classical systems, analytically and numerically. Deterministic Hamiltonian ODEs and PDEs are first examined, after which stochastic perturbations are introduced and analyzed using stochastic differential equations, Itô calculus and numerical methods. A numerical integrator called Strang-AVF is derived, which preserves the long-time behaviour of the expected value of the Hamiltonian in stochastic systems.

Keywords: Hamiltonian ODEs, Hamiltonian PDEs, Stochastic Hamiltonian, Brownian motion, AVF, Strang-AVF

Acknowledgements

We would like to thank our supervisor and examiner David Cohen, for all his guidance, patience and valuable feedback throughout this thesis work.

Sandy Abonaser, Homam Mowed, Gothenburg, October 2026

Contents

List of Figures	xi
1 Introduction	1
2 Hamiltonian ODEs and their numerical discretization	2
2.1 Hamiltonian ODEs	2
2.2 Numerical discretization of the mathematical pendulum	3
2.3 The AVF method	5
3 Hamiltonian PDEs and their numerical discretizations	9
3.1 Hamiltonian PDEs	9
3.1.1 The sine-Gordon equation	10
3.1.1.1 First order system for the sine-Gordon equation . . .	10
3.1.1.2 Numerical discretizations of the sine-Gordon equation	12
3.1.1.3 The AVF scheme for the sine-Gordon equation . . .	15
3.1.2 The time-dependent Schrödinger equation	16
3.1.2.1 The Schrödinger equation as a Hamiltonian system .	17
3.1.2.2 The AVF scheme for the Schrödinger equation	19
4 Stochastic differential equations	21
4.1 Brownian Motion	21
4.2 Itô Integral	22
4.3 Stochastic Differential Equations	24
4.4 The Euler-Maruyama method	25
4.5 Application: Black-Scholes-Merton model	25
5 Hamiltonian stochastic differential equations	28
5.1 The stochastic pendulum	28
5.2 The expected total energy of the pendulum	28
5.3 The Strang-AVF scheme for the stochastic pendulum	30
5.3.1 Derivation of the Strang-AVF scheme	30
5.3.2 Numerical experiments	32
6 Hamiltonian stochastic partial differential equations	34
6.1 The stochastic sine-Gordon equation	34
6.1.1 Expected total energy for the sine-Gordon equation	35

6.1.2	The Strang–AVF scheme for the stochastic sine–Gordon equation	36
6.1.3	Numerical experiments	38
6.2	Long-time behaviour of the stochastic Schrödinger equation	40
6.2.1	Strang–AVF scheme for the stochastic Schrödinger	41
6.2.2	Numerical experiments	42
7	Conclusion	44
	Bibliography	44
A	Numerical implementation	I
A.1	Code used for the pendulum experiments	I
A.2	Code used for the sine–Gordon experiments	VI
A.3	Code used for the simulation of Brownian motion path and the Black–Scholes–Merton model	IX
A.4	Code used for the stochastic pendulum experiments	XI
A.5	Code used for the stochastic sine–Gordon experiments	XV
A.6	Code used for the time–dependent Schrödinger equation	XIX
A.7	The AVF quotient	XXVI

List of Figures

2.1	Evolution of the total energy for the mathematical pendulum (2.4) using the explicit Euler scheme (2.8) and the Störmer–Verlet scheme (2.10) for time step $k = 0.5$, number of discretization points $N = 100$, final time $T = 50$, initial condition $(p_0, q_0) = (1.5, 1)$ and initial energy $H_0 = 0.5847$	5
2.2	Evolution of the total energy $H(p_n, q_n)$ for the mathematical pendulum (2.4) using the AVF scheme (2.22) and the Störmer–Verlet scheme (2.10), for number of time step $k = 0.5$, discretization points $N = 100$, final time $T = 50$, initial energy $H_0 = 0.5847$, initial condition $(p_0, q_0) = (1.5, 1)$, fixed point iteration tolerance $\text{TOL} = 10^{-8}$. . .	8
3.1	Evolution of the discretized total energy (3.13) for the sine–Gordon dynamics (3.19) using the AVF scheme and the Störmer–Verlet scheme for initial energy $\mathcal{H}_d^0 = 11.6940$, initial conditions $(\mathbf{u}_0, \mathbf{v}_0) = (1, 1.5)$, spatial domain $[0, 20]$, number of spatial discretization points $N_{\text{spatial}} = 200$, mesh size $\Delta x = 0.1$, number of time discretization points $N_{\text{time}} = 100$, time step $k = 0.5$, fixed point iteration tolerance $\text{Tol} = 10^{-8}$. . .	16
3.2	Evolution of the L^2 norm squared for the Schrödinger equation (3.23) using the AVF scheme for spatial domain $[-20, 20]$, number of discretization points $N_{\text{spatial}} = 200$, spatial step size $\Delta x = 0.2$, number of discretization points in time $N_{\text{time}} = 300$, time step $\Delta t = 0.7$, initial conditions $(\mathbf{u}_{\text{real}}^0, \mathbf{u}_{\text{im}}^0) = (e^{-\frac{(x-1)^2}{2}}, e^{-\frac{x^2}{2}})$, initial L^2 norm = 3.5449, fixed point iteration tolerance $\text{TOL} = 10^{-6}$	20
4.1	Simulation of discretized Brownian motion path using (4.2) for final time $T = 1$ and number of discretization points $N = 500$	22
4.2	Numerical approximation of the SDE (4.17) using the Euler–Maruyama scheme (4.15) compared to the exact solution (4.18), which is evaluated using a discretized Brownian motion path.	27
5.1	Evolution of the expected total energy for the stochastic pendulum (5.7) using the constructed Strang–AVF scheme (5.11) and the Euler–Maruyama scheme (4.15). For final time $T = 20$, number of discretization points $N = 100$, diffusion coefficient $\sigma = 1$, initial condition $(q_0, p_0) = (1, 1.5)$, initial exact total energy $H_0 = 0.5847$, step size $h = \frac{T}{N}$, tolerance for fixed point iteration $\text{TOL} = 10^{-8}$, and number of realizations $M = 10^6$, run time ≈ 2.29 minutes.	33

- 6.1 Evolution of the expected total energy for the stochastic sine-Gordon equation (6.1) using the constructed Strang–AVF scheme (6.15) and Euler–Maruyama scheme (4.15). For final time $T = 20$, number of discretization points of the time domain $N_{\text{time}} = 100$, time step size $h = \frac{T}{N_{\text{time}}}$, spatial domain $[0, L] = [0, 10]$, number of discretization points of spatial domain $N_{\text{spatial}} = 150$, spatial mesh $\Delta x = \frac{L}{N_{\text{spatial}}}$, initial condition $(\mathbf{u}_0, \mathbf{v}_0) = (1, 1.5)$, initial energy $\mathcal{H}_0 = 5.8470$, tolerance for fixed point iteration $\text{TOL} = 10^{-8}$, number of realizations $M = 10^5$, run time ≈ 27.18 minutes. 39
- 6.2 Evolution of the expected value of the discretized L^2 norm squared (3.39) for the Stochastic Schrödinger equation (6.17) equation using Strang–AVF scheme (6.28), for final time $T = 2$, number of discretization points of the time domain $N_{\text{time}} = 300$, spatial domain $[-20, 20]$, number of discretization points of spatial domain $N_{\text{spatial}} = 200$, initial conditions $(\mathbf{u}_{\text{real}}^0, \mathbf{u}_{\text{im}}^0) = (e^{-\frac{(x-1)^2}{2}}, e^{-\frac{x^2}{2}})$, initial L^2 norm squared $M_0 = 3.5449$, time step size $h = \frac{T}{N_{\text{time}}}$, spatial step size $\Delta x = \frac{40}{N_{\text{spatial}}}$, $M = 10^5$ number of realizations, run time ≈ 52.70 minutes. 43

1

Introduction

The Hamiltonian formulation of physical systems is fundamental in classical and quantum mechanics as it represents the total energy of the dynamics. The study of Hamiltonian equations of motion reveals the energy conservative nature that certain dynamics possess. For this reason, different numerical methods were developed in an attempt to mimic the underlying behaviour of the dynamics in question.

In the context of the thesis, one differentiates between two categories of dynamics, namely deterministic and nondeterministic dynamics. In the deterministic case, the Hamiltonian represents the total energy of the system which remains constant over time. However, in many applications, the effects of external random force for instance must be taken into account. The introduction of a Brownian motion serves as an imitation tool for the resulting perturbation. This, in turn, gives rise to stochastic Hamiltonian systems. One main objective of this work is to get insight into the behaviour of the Hamiltonian in the stochastic context.

The thesis aims to study the long-time behaviour of Hamiltonian stochastic partial differential equations analytically and numerically. In order to achieve this goal, the thesis begins by examining deterministic ODEs (chapter 2) and PDEs (chapter 3) along with a structure preserving numerical scheme. Namely, the Average Vector Field (AVF). Then, we consider stochastic perturbations driven by one additive Brownian motion and analyze the evolution of the system affected by noise in the sense of Itô (chapter 4). Building on the developments for deterministic and stochastic Hamiltonian ODEs and PDEs, we are finally able to investigate the long-time behaviour of Hamiltonian SDEs (chapter 5) and Hamiltonian SPDEs (chapter 6). In these chapters, we combine two numerical ideas. In particular, we consider the Strang-splitting strategy combined to the AVF scheme and thus we introduce a new numerical integrator, which we call the Strang-AVF method. This novel method will later turn out to preserve the behaviour of the stochastic Hamiltonian over long-time intervals.

2

Hamiltonian ODEs and their numerical discretization

This chapter is devoted to Hamiltonian systems introduced within the framework of ordinary differential equations. A discussion of one of the main properties of Hamiltonian systems, namely the existence of a first integral, will be presented, along with a practical illustrative example, see section 2.1. We then study the behaviour of the Hamiltonian function utilizing different numerical integrators, see sections 2.2 and 2.3. The same framework will be generalized to partial differential equations settings in the next chapter. The main reference for this chapter is [9].

2.1 Hamiltonian ODEs

Hamiltonian systems are dynamical systems whose evolution can be written in terms of a Hamiltonian function $H(p_1, \dots, p_d, q_1, \dots, q_d)$ that represents the total energy of the system [9]. More precisely, using the position coordinates q_i and the momenta p_i for $i = 1, \dots, d$, where d denotes the number of degrees of freedom, the dynamics can be written as¹

$$\begin{cases} \dot{p} = -\nabla_q H(p, q), \\ \dot{q} = \nabla_p H(p, q), \\ p(0) = p_0, \\ q(0) = q_0, \end{cases} \quad (2.1)$$

where $p = p(t), q = q(t)$. Furthermore, the Hamiltonian function is a first integral; meaning that the total energy of the system is conserved [19]. This can be shown as follows

$$\begin{aligned} \frac{dH(p(t), q(t))}{dt} &= \nabla_p H \frac{dp(t)}{dt} + \nabla_q H \frac{dq(t)}{dt} = \\ &= \left[\text{Using (2.1)} \right], \\ &= -\nabla_p H \nabla_q H + \nabla_q H \nabla_p H = 0. \end{aligned} \quad (2.2)$$

A simple example of such a Hamiltonian system is the mathematical pendulum with mass $m = 1$, rod length $l = 1$ and gravitational acceleration $g = 1$. This system has

¹the dots in (2.1) on the momenta p and the position coordinate q stand for differentiation with respect to time

the kinetic energy $T(p) = \frac{1}{2}p^2$, the potential energy $V(q) = -\cos(q)$ and is classified as a Hamiltonian system defined by the Hamiltonian function [9]

$$H(p, q) = T(p) + V(q) = \frac{1}{2}p^2 - \cos(q), \quad (2.3)$$

which represent the total energy of the system. Using (2.1) the following equations of motion are obtained

$$\begin{cases} \dot{p} = -\sin(q), \\ \dot{q} = p. \end{cases} \quad (2.4)$$

In order to write the dynamics of the mathematical pendulum (2.4) in the Hamiltonian canonical form, denote the state of the system (2.4) by $\mathbf{z} = (p, q)^T$. The phase space is defined as the even dimensional Euclidean space $\mathbf{R}^2$ [21]. Taking $\nabla_{\mathbf{z}}$ of (2.3) gives

$$\nabla_{\mathbf{z}}H(\mathbf{z}) = \begin{bmatrix} \nabla_p H(\mathbf{z}) \\ \nabla_q H(\mathbf{z}) \end{bmatrix} = \begin{bmatrix} p \\ \sin(q) \end{bmatrix}. \quad (2.5)$$

Using (2.5), the evolution of the state $\mathbf{z}$ over time can be described as

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \nabla_p H(\mathbf{z}) \\ \nabla_q H(\mathbf{z}) \end{bmatrix},$$

which corresponds to (2.1). Thus, the Hamiltonian canonical form of the system (2.4) is given by [9]

$$\dot{\mathbf{z}} = J^{-1} \nabla_{\mathbf{z}}H(\mathbf{z}), \quad (2.6)$$

where $\mathbf{z} = \mathbf{z}(t)$ and J is a constant skew symmetric matrix given by

$$J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

2.2 Numerical discretization of the mathematical pendulum

As shown in (2.2), the total energy of the mathematical pendulum along the trajectories governed by (2.4) is conserved. Thus, it is of great interest to study numerical schemes that preserve this behaviour. In order to numerically integrate the dynamics (2.4) for the final time $T = 50$, step size $k = 0.5$ and $N = 100$ number of discretization points, the time domain $[0, T]$ is discretized according to

$$0 = t_0 < t_1 < \dots < t_N = T, \quad t_n = n \cdot k, \quad \text{for } n = 0, 1, 2, \dots, N. \quad (2.7)$$

Using the explicit Euler method with time step k yields the following update of the dynamics in (2.4)

$$p_{n+1} = p_n - k \sin(q_n), \quad (2.8a)$$

$$q_{n+1} = q_n + kp_n. \quad (2.8b)$$

Symplectic Euler scheme can be derived by treating the update for the position variable (2.8b) implicitly in the momenta variable p , which gives the following symplectic integrator of order one [9]

$$p_{n+1} = p_n - k \sin(q_n), \quad (2.9a)$$

$$q_{n+1} = q_n + kp_{n+1}. \quad (2.9b)$$

Another version of the symplectic Euler scheme can be obtained by treating the update for the momenta variable (2.8a) implicitly in the position variable q . Composing those two versions of symplectic Euler schemes yields what is known as the Störmer–Verlet scheme

$$p_{n+\frac{1}{2}} = p_n + \frac{k}{2} \sin(q_n), \quad (2.10a)$$

$$q_{n+1} = q_n + kp_{n+\frac{1}{2}}, \quad (2.10b)$$

$$p_{n+1} = p_{n+\frac{1}{2}} + \frac{k}{2} \sin(q_{n+1}), \quad (2.10c)$$

which is a symplectic integrator of order 2 [2],[9]. The code used to implement the explicit Euler scheme (2.8) and the Störmer–Verlet scheme (2.10) is presented in Appendix A.1. Figure 2.1 below, shows the evolution of the total energy using the explicit Euler scheme (2.8) and the Störmer–Verlet scheme (2.10). One observes that the total energy drifts away over long–time intervals when using the explicit Euler scheme. In contrast, the Strömer–Verlet scheme shows much better long–time behaviour. But, the Hamiltonian is not exactly conserved.

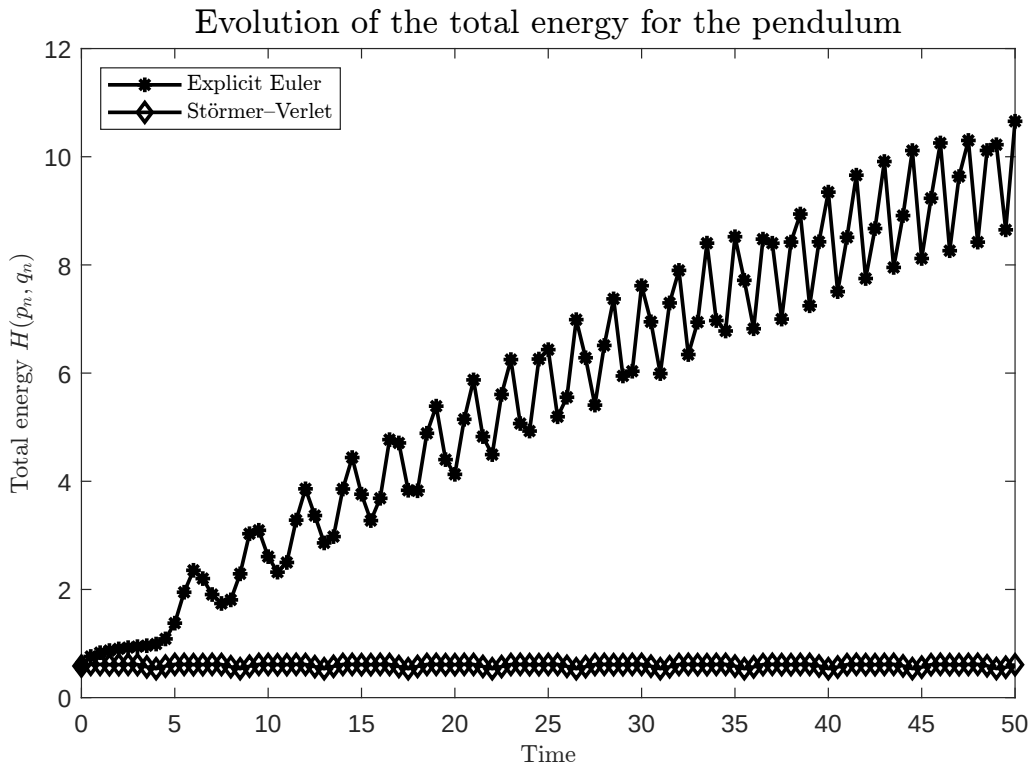


Figure 2.1: Evolution of the total energy for the mathematical pendulum (2.4) using the explicit Euler scheme (2.8) and the Störmer–Verlet scheme (2.10) for time step $k = 0.5$, number of discretization points $N = 100$, final time $T = 50$, initial condition $(p_0, q_0) = (1.5, 1)$ and initial energy $H_0 = 0.5847$.

2.3 The AVF method

Although the Strörmer–Verlet method, as shown in the previous section, exhibits improved long-time behaviour, we seek a method that exactly mirrors the behaviour of the original Hamiltonian function. In this section, a novel numerical integration method called the averaged vector field method (AVF) is introduced. One of the main properties of this method is that it preserves the total energy H exactly as it takes only the vector field as input. The method will be studied and then applied to the mathematical pendulum (2.4).

Given the general ODE

$$\frac{dx}{dt} = f(x), \quad x = x(t), \quad x(0) = x_0 \quad (2.11)$$

where $f(x)$ is a given vector field. The AVF scheme is given by [17]

$$\frac{x_{n+1} - x_n}{k} = \int_0^1 f((1 - \xi)x_n + \xi x_{n+1}) d\xi, \quad (2.12)$$

where k is the step size, $(1 - \xi)x_n + \xi x_{n+1}$ is the convex combination of x_n and x_{n+1} for $\xi \in [0, 1]$ i.e. the line segment between x_n and x_{n+1} . The idea behind the AVF

method is that, at each update, the vector field is evaluated along the line segment connecting x_n and x_{n+1} and then averaged.

Proposition 2.3.1 below, shows the assumptions needed to prove why the AVF scheme (2.12) conserves the energy exactly. It is then followed by the proof, which is taken from [17].

Proposition 2.3.1. *Suppose that the vector field $f(x)$ in (2.11) can be written as*

$$f(x) = S\nabla H(x), \quad (2.13)$$

where S is a skew-symmetric matrix and H is a Hamiltonian function. Then the AVF method (2.12) preserves the energy exactly. That is, the energy does not change with time and $H(x_{n+1}) = H(x_n) = \dots = H(x_0)$.

Proof. Substituting (2.13) into (2.12) yields

$$\frac{x_{n+1} - x_n}{k} = \int_0^1 S\nabla H((1 - \xi)x_n + \xi x_{n+1}) d\xi. \quad (2.14)$$

For convenience, define $v_n := \int_0^1 \nabla H((1 - \xi)x_n + \xi x_{n+1}) d\xi$. Then, the expression becomes

$$\frac{x_{n+1} - x_n}{k} = Sv_n.$$

Taking the inner product of both sides with v_n gives

$$v_n^T \frac{x_{n+1} - x_n}{k} = v_n^T Sv_n. \quad (2.15)$$

S is skew-symmetric, then $S^T = -S$ and therefore

$$\langle v_n, Sv_n \rangle = v_n^T Sv_n = (v_n^T Sv_n)^T = (v_n^T S^T v_n) = -\langle v_n, Sv_n \rangle. \quad (2.16)$$

Based on (2.16), one obtains

$$2\langle v_n, Sv_n \rangle = 0,$$

which means that the vectors v_n and Sv_n are orthogonal, and

$$\langle v_n, Sv_n \rangle = 0. \quad (2.17)$$

Taking into account the LHS of (2.15), we thus have

$$\frac{1}{k} \int_0^1 \underbrace{\nabla H((1 - \xi)x_n + \xi x_{n+1})^T (x_{n+1} - x_n)}_{\frac{d}{d\xi} H((1 - \xi)x_n + \xi x_{n+1})} d\xi = 0. \quad (2.18)$$

Applying the fundamental theorem of calculus to (2.18) gives

$$\frac{1}{k} (H(x_{n+1}) - H(x_n)) = 0,$$

which implies that the energy is conserved at each time step, i.e.

$$H(x_{n+1}) = H(x_n) = \dots = H(x_0).$$

□

Now that we have established the main property of the AVF scheme (2.12), the method will be used to numerically integrate the dynamics of the mathematical pendulum (2.4) and the total energy (2.3) will be studied. As shown previously in section 2.1, the vector field for the mathematical pendulum can be written in the form (2.6) and hence satisfies the assumptions required for the AVF method to exactly preserve the total energy at each time step.

Using the AVF scheme (2.12), one obtains the following

$$p_{n+1} = p_n - k \int_0^1 \sin((1 - \xi)q_n + \xi q_{n+1}) d\xi, \quad (2.19a)$$

$$q_{n+1} = q_n + k \int_0^1 (1 - \xi)p_n + \xi p_{n+1} d\xi. \quad (2.19b)$$

By computing the integrals in (2.19a)-(2.19b) analytically, one obtains

$$\int_0^1 \sin((1 - \xi)q_n + \xi q_{n+1}) d\xi = \frac{\cos(q_n) - \cos(q_{n+1})}{q_{n+1} - q_n}, \quad (2.20)$$

and

$$\int_0^1 (1 - \xi)p_n + \xi p_{n+1} d\xi = \frac{1}{2}(p_n + p_{n+1}). \quad (2.21)$$

Inserting (2.20) and (2.21) in (2.19a) and (2.19b) respectively, one obtains the following AVF scheme

$$p_{n+1} = p_n - k \frac{\cos(q_n) - \cos(q_{n+1})}{q_{n+1} - q_n}, \quad (2.22a)$$

$$q_{n+1} = q_n + \frac{k}{2}(p_n + p_{n+1}). \quad (2.22b)$$

The AVF quotient derived in the AVF scheme (2.22a) should be handled carefully when the dominator $q_{n+1} - q_n$ approaches zero. Thus, the AVF quotient is implemented as suggested in [5], for details see Appendix A.7. Furthermore, in the derivation of the AVF scheme (2.22) for the mathematical pendulum (2.4) the integrals are exactly computed. When numerical methods are used to approximate the integrals, e.g. quadrature formulas, the mechanism of the AVF method is affected in a way that may fail to exactly preserve the energy.

In order to implement the AVF scheme (2.22), which is an implicit update consisting of a non-linear system with two unknowns, a fixed point iteration method is employed, see, for instance, algorithm 2.2 in [4]. Furthermore, the code used to implement the AVF scheme (2.22) is presented in Appendix A.1. Figure 2.2 below, shows the evolution of the total energy for long-times using the AVF scheme (2.22) and the Störmer–Verlet scheme (2.10) derived earlier in section 2.2. The AVF method shows exact conservation of energy as expected due to the Proposition 2.3.1, while the Störmer–Verlet scheme shows an oscillation of energy which is bounded from above by ≈ 0.6157 and from below by ≈ 0.5341 .

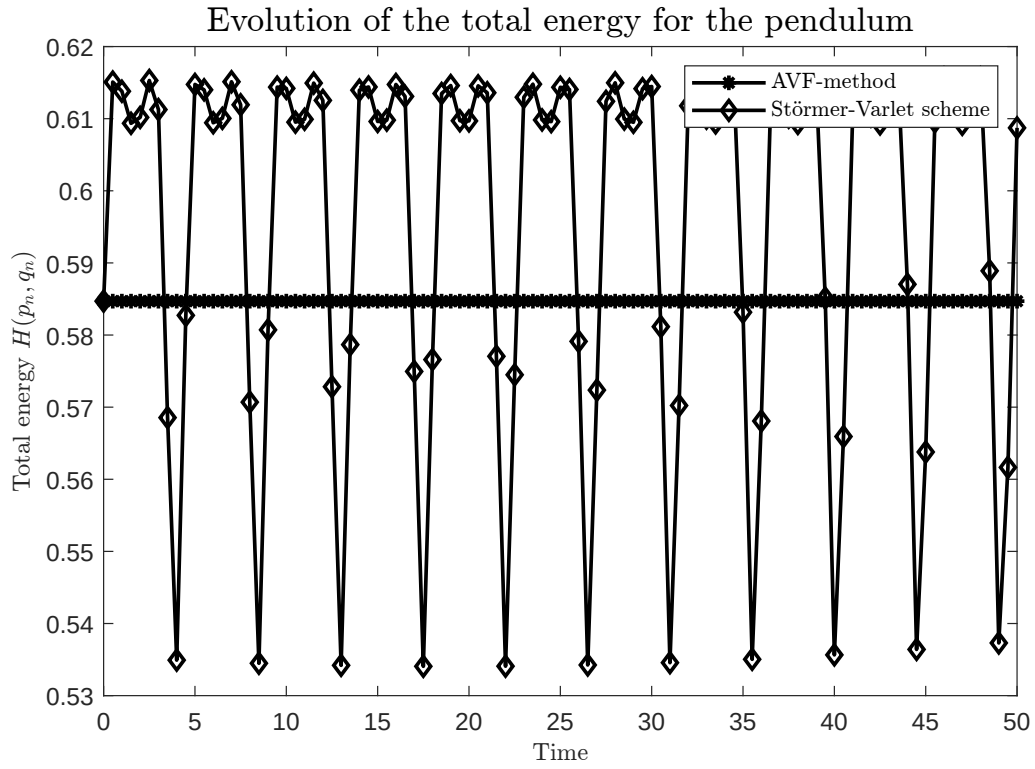


Figure 2.2: Evolution of the total energy $H(p_n, q_n)$ for the mathematical pendulum (2.4) using the AVF scheme (2.22) and the Störmer–Verlet scheme (2.10), for number of time step $k = 0.5$, discretization points $N = 100$, final time $T = 50$, initial energy $H_0 = 0.5847$, initial condition $(p_0, q_0) = (1.5, 1)$, fixed point iteration tolerance $\text{TOL} = 10^{-8}$.

3

Hamiltonian PDEs and their numerical discretizations

In this chapter, Hamiltonian Partial Differential Equations (PDEs) will be investigated, adopting the same approach as in the previous discussion on Hamiltonian ODEs. In particular, we first introduce Hamiltonian PDEs, see section 3.1. We then study the sine–Gordon equation, see section 3.1.1 and its numerical discretizations, see section 3.1.1.2. Thereafter, we numerically study the behaviour of the Hamiltonian using the Störmer–Verlet scheme (2.10) and the AVF scheme (2.12), see section 3.1.1.3. Finally, we introduce the linear time-dependent Schrödinger equation, see section 3.1.2. In particular, we study the behaviour of the L^2 –norm of solutions associated with the Schrödinger equation. After that, we show how the AVF scheme (2.12) preserves this behaviour, see section 3.1.2.2. The main reference for this chapter is [15].

3.1 Hamiltonian PDEs

Hamiltonian PDEs can be viewed as an extension of Hamiltonian ODEs to systems of infinite degrees of freedom such as systems in quantum mechanics or electrodynamics. They are mathematical models utilized to describe physical phenomena that have a continuous spatial structure and evolve over time [15].

The state of the solution of a Hamiltonian ODE is described as a point (q, p) consisting of generalized coordinates and generalized momenta in the finite dimensional phase space $\mathbb{R}^{2d}$, see previous chapter. However, in Hamiltonian PDEs, the phase space is infinite dimensional and the state is given by functions $(u, v) \in \mathcal{X} \times \mathcal{X}$, for some function space $\mathcal{X}$. The total energy of the system is then represented by a Hamiltonian functional $\mathcal{H}[u, v]$ acting on (u, v) .

In order to derive the equations of motion of Hamiltonian PDEs, we consider the variational derivative δ_z instead of the standard gradient ∇_z used in classical mechanics, see equation (2.1). That is, the expression $\dot{z} = J\nabla_z H$ becomes $\dot{z} = \mathcal{J}\delta_z \mathcal{H}$, where $\mathcal{J}$ is a structure matrix and $\delta_z \mathcal{H} = (\delta_u \mathcal{H}, \delta_v \mathcal{H})$ is the variational derivative [15]. Therefore, the corresponding equations of motion takes the form

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \text{Id} \\ -\text{Id} & 0 \end{bmatrix}}_{\mathcal{J}} \underbrace{\begin{bmatrix} \delta_u \mathcal{H} \\ \delta_v \mathcal{H} \end{bmatrix}}_{\delta_z \mathcal{H}}.$$

3.1.1 The sine–Gordon equation

An example of Hamiltonian PDEs is the sine-Gordon equation given by

$$u_{tt} = u_{xx} - \sin(u), \quad (3.1)$$

which is a second-order nonlinear PDE for the function $u = u(x, t)$. It plays an important role in mathematical physics and non-linear optics. Moreover, it arises in the modeling of several phenomena such as coupled torsion pendula and Josephson junctions, see for instance, [7]. Here, the solution $u = u(x, t)$ depends on both variables x and t and is assumed to be smooth. Furthermore, periodic boundary conditions are imposed, i.e., $u(x, t) = u(x + L, t)$, where $L > 0$ denotes the spatial period. In this setting, the total energy of (3.1) takes the form

$$\mathcal{H}[u] = \int_0^L \left(\frac{1}{2} u_t^2 + \frac{1}{2} u_x^2 - \cos(u) \right) dx, \quad (3.2)$$

which is a conserved quantity along solutions $u(x, t)$ of the sine–Gordon equation (3.1). This can be shown by computing the time derivative $\frac{d}{dt} \mathcal{H}$ (recalling $u = u(x, t)$):

$$\begin{aligned} \frac{d}{dt} \mathcal{H}[u] &= \int_0^L \left(u_t u_{tt} + u_x u_{xt} + \sin(u) u_t \right) dx, \\ &= \left[\text{integrating the second term by parts} \right], \\ &= \left[\int_0^L u_x u_{xt} dx = \underbrace{[u_x u_t]_0^L}_{=0, \text{ BCs}} - \int_0^L u_{xx} u_t dx \right], \\ &= \int_0^L u_t \underbrace{\left(u_{tt} - u_{xx} + \sin(u) \right)}_{=0, \text{ along the solutions}} dx, \\ &= 0. \end{aligned}$$

Thus, $\mathcal{H}$ remains constant as the system evolves in time along its motion.

3.1.1.1 First order system for the sine–Gordon equation

We now rewrite the second order sine–Gordon system (3.1) by introducing the auxiliary variable $v = u_t$. This reformulation enables us to view the dynamics in phase space framework represented by (u, v) , the state of the system, which further reveals

the underlying conservative structure. To make this concrete, the sine–Gordon equation (3.1) is transformed to

$$\begin{aligned} u_t &= v \\ v_t &= u_{xx} - \sin(u), \end{aligned} \tag{3.3}$$

which is a first order PDE in time and second order PDE in space [15]. In order to show that the system (3.3) is a Hamiltonian system, we need to express its dynamics in the Hamiltonian form. The phase space of the system (3.3) can be defined as $(u, v)^T \in S \times S$, where S denotes the solution space, and the corresponding Hamiltonian functional is given by

$$\mathcal{H}[u, v] = \int_0^L \left[\frac{1}{2}v^2 + \frac{1}{2}u_x^2 - \cos(u) \right] dx. \tag{3.4}$$

As before, we denote the state of the system by $\mathbf{z} = (u, v)^T$ and its variational derivative by $\boldsymbol{\delta}_\mathbf{z} = [\delta_u, \delta_v]^T$, where δ_u and δ_v define the variation in u and v respectively. Under a perturbation of both components of $\mathbf{z}$ of the form

$$u \rightarrow u + \varepsilon \delta_u, \quad v \rightarrow v + \varepsilon \delta_v,$$

where ε is a small real number, we obtain the variational derivative of (3.4), see definition 4.1 in [16], by

$$\boldsymbol{\delta}_\mathbf{z} \mathcal{H}[u, v] = \left. \frac{d}{d\varepsilon} \right|_{\varepsilon=0} \mathcal{H}[u + \varepsilon \delta_u, v + \varepsilon \delta_v]. \tag{3.5}$$

Using (3.5), one obtains

$$\begin{aligned} \left. \frac{d}{d\varepsilon} \left(\frac{1}{2}(v + \varepsilon \delta_v)^2 \right) \right|_{\varepsilon=0} &= (v + \varepsilon \delta_v) \delta_v \Big|_{\varepsilon=0} = v \delta_v, \\ \left. \frac{d}{d\varepsilon} \left(\frac{1}{2}(u_x + \varepsilon \delta_{u_x})^2 \right) \right|_{\varepsilon=0} &= (u_x + \varepsilon \delta_{u_x}) \delta_{u_x} \Big|_{\varepsilon=0} = u_x \delta_{u_x}, \\ \left. \frac{d}{d\varepsilon} \left(-\cos(u + \varepsilon \delta_u) \right) \right|_{\varepsilon=0} &= \sin(u + \varepsilon \delta_u) \delta_u \Big|_{\varepsilon=0} = \sin(u) \delta_u. \end{aligned}$$

We next combine the terms and integrate to obtain the change of $\mathcal{H}$ in the direction of both u and v as

$$\mathcal{H}[u, v][\delta_u, \delta_v] = \int_0^L \left(v \delta_v + u_x \delta_{u_x} + \sin(u) \delta_u \right) dx.$$

Integrating the second term by parts, we get

$$\int_0^L u_x \delta_{u_x} dx = \underbrace{[u_x \delta_u]_0^L}_{=0, \text{BCs}} - \int_0^L u_{xx} \delta_u dx.$$

Hence,

$$\begin{aligned}\mathcal{H}[u, v][\delta_u, \delta_v] &= \int_0^L \left(v\delta_v - u_{xx}\delta_u + \sin(u)\delta_u \right) dx, \\ &= \int_0^L \left((-u_{xx} + \sin(u))\delta_u + v\delta_v \right) dx.\end{aligned}$$

Therefore, the variational derivative is given by

$$\delta_z \mathcal{H}[z] = \begin{bmatrix} \frac{\delta \mathcal{H}}{\delta u} \\ \frac{\delta \mathcal{H}}{\delta v} \end{bmatrix} = \begin{bmatrix} -u_{xx} + \sin(u) \\ v \end{bmatrix}, \quad (3.6)$$

and (3.3) is a canonical Hamiltonian system of the form

$$\dot{z} = \mathcal{J} \delta_z \mathcal{H}[z], \quad (3.7)$$

where $\mathcal{J}$ is the canonical skew-adjoint operator given by

$$\mathcal{J} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (3.8)$$

That is the system (3.7) is defined on even dimensional Euclidean space with operator $\mathcal{J}$ as in (3.8) [15].

3.1.1.2 Numerical discretizations of the sine-Gordon equation

In this section, we study the behavior of different numerical schemes regarding energy conservation for the sine-Gordon equation (3.1). To achieve this, a finite difference spatial discretization of the domain $[0, L]$ consisting of N sub-intervals with uniform mesh Δx will be carried out according to

$$x_i = i\Delta x, \quad \text{for } i = 0, 1, \dots, N-1, \quad \Delta x = \frac{L}{N}. \quad (3.9)$$

The discrete solution at the discretization point i is denoted $(u_i(t), v_i(t)) \approx (u(x_i, t), v(x_i, t))$. Alternatively, we denote

$$\mathbf{u} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ \vdots \\ v_{N-1} \end{bmatrix},$$

in vector form, when (3.1) is reduced to (3.3). Starting by discretizing the Hamiltonian functional (3.4), denoted by $\mathcal{H}_d$; where d stands for discretized, and approximating u_x by the forward difference

$$u_x(x_i, t) \approx \frac{u_{i+1} - u_i}{\Delta x}, \quad (3.10)$$

one obtains

$$\mathcal{H}_d[\mathbf{u}, \mathbf{v}] = \Delta x \sum_{i=0}^{N-1} \left[\frac{1}{2} v_i^2 + \frac{1}{2(\Delta x)^2} (u_{i+1} - u_i)^2 - \cos(u_i) \right]. \quad (3.11)$$

By introducing the forward difference periodic operator $\mathcal{D}$, which acts on the solution vector $\mathbf{u}$, by

$$\mathcal{D} = \frac{1}{\Delta x} \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ \underbrace{1}_{\text{BCs}} & 0 & \cdots & 0 & -1 \end{bmatrix}, \quad (3.12)$$

which component-wise corresponds to

$$(\mathcal{D}\mathbf{u})_i = \frac{u_{i+1} - u_i}{\Delta x}.$$

The discretized Hamiltonian functional (3.11) can be written in matrix form as

$$\begin{aligned} \mathcal{H}_d[\mathbf{u}, \mathbf{v}] &= \Delta x \sum_{i=0}^{N-1} \left[\frac{1}{2} v_i^2 + \frac{1}{2(\Delta x)^2} (u_{i+1} - u_i)^2 - \cos(u_i) \right], \\ &= \Delta x \left(\frac{1}{2} \mathbf{v}^T \mathbf{v} + \frac{1}{2} (\mathcal{D}\mathbf{u})^T (\mathcal{D}\mathbf{u}) - \cos(\mathbf{u}) \right). \end{aligned}$$

Write $(\mathcal{D}\mathbf{u})^T (\mathcal{D}\mathbf{u}) = \mathbf{u}^T \underbrace{\mathcal{D}^T \mathcal{D}}_{\mathcal{A}} \mathbf{u} = \mathbf{u}^T \mathcal{A} \mathbf{u}$. Then:

$$\mathcal{H}_d[\mathbf{u}, \mathbf{v}] = \Delta x \left(\frac{1}{2} \mathbf{v}^T \mathbf{v} + \frac{1}{2} \mathbf{u}^T \mathcal{A} \mathbf{u} - \cos(\mathbf{u}) \right). \quad (3.13)$$

Taking into account the definition of $\mathcal{D}$ given in (3.12), $\mathcal{A}$ takes then the form

$$\begin{aligned} \mathcal{A} &= \frac{1}{(\Delta x)^2} \underbrace{\begin{bmatrix} -1 & 0 & 0 & \cdots & 1 \\ 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 & 0 \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix}}_{\mathcal{D}^T} \underbrace{\begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ 1 & 0 & \cdots & 0 & -1 \end{bmatrix}}_{\mathcal{D}}, \\ &= \frac{1}{(\Delta x)^2} \begin{bmatrix} 2 & -1 & 0 & \cdots & -1 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 2 & -1 \\ -1 & 0 & \cdots & -1 & 2 \end{bmatrix}, \end{aligned} \quad (3.14)$$

and it follows that $\mathcal{A}^T = \mathcal{A}$. One important observation is that the choice of the periodic boundary conditions, suggested in [15], is crucial along with the approximation of u_x by the forward difference scheme (3.10) in the discretized Hamiltonian (3.11).

Taking the variational derivative of the discretized Hamiltonian $\mathcal{H}_d$ in matrix form (3.13), which corresponds to computing its gradient $\nabla \mathcal{H}_d$, as shown in [5], gives

$$\nabla_{\mathbf{u}} \mathcal{H}_d = \nabla_u \Delta x \left(\frac{1}{2} \mathbf{u}^T \mathcal{A} \mathbf{u} - \cos(\mathbf{u}) \right).$$

Using the fact that the operator $\mathcal{A}$ is symmetric, one obtains

$$\nabla_{\mathbf{u}} \mathcal{H}_d = \Delta x \left(\mathcal{A} \mathbf{u} + \sin(\mathbf{u}) \right). \quad (3.15)$$

Similarly, one has

$$\nabla_{\mathbf{v}} \mathcal{H}_d = \Delta x \mathbf{v}. \quad (3.16)$$

Collecting the components (3.15) and (3.16), the gradient of the discretized Hamiltonian is given by

$$\nabla \mathcal{H}_d[\mathbf{u}, \mathbf{v}] = \begin{bmatrix} \nabla_{\mathbf{u}} \mathcal{H}_d \\ \nabla_{\mathbf{v}} \mathcal{H}_d \end{bmatrix} = \begin{bmatrix} \Delta x \left(\mathcal{A} \mathbf{u} + \sin(\mathbf{u}) \right) \\ \Delta x \mathbf{v} \end{bmatrix}. \quad (3.17)$$

Thus, the Hamiltonian equations of motions of the spatial discretized solution are

$$\begin{aligned} \dot{\mathbf{u}} &= \nabla_v \mathcal{H}_d, \\ \dot{\mathbf{v}} &= -\nabla_u \mathcal{H}_d, \end{aligned} \quad (3.18)$$

which can be written as

$$\dot{\mathbf{z}} = \mathcal{S} \nabla \mathcal{H}_d,$$

with skew-adjoint operator

$$\mathcal{S} = \frac{1}{\Delta x} \begin{bmatrix} 0 & \text{Id} \\ -\text{Id} & 0 \end{bmatrix}.$$

Based on (3.17) and (3.18), one gets the following system

$$\begin{aligned} \dot{\mathbf{u}} &= \mathbf{v}, \\ \dot{\mathbf{v}} &= -\underbrace{\mathcal{A} \mathbf{u}}_{\mathcal{L}} - \sin(\mathbf{u}), \end{aligned} \quad (3.19)$$

where $\mathcal{L}$ is the operator given by

$$\mathcal{L} = \frac{1}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & 0 & \cdots & 1 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -2 & 1 \\ 1 & 0 & \cdots & 1 & -2 \end{bmatrix}, \quad (3.20)$$

using (3.14). Furthermore, the 3-points stencil (3.20) can also be obtained by spatial discretization of the system (3.3), where the second order derivative u_{xx} is approximated by the central difference scheme

$$u_{xx} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}. \quad (3.21)$$

3.1.1.3 The AVF scheme for the sine–Gordon equation

As we have seen earlier in Proposition 2.3.1, the AVF scheme conserves the energy exactly for each time step. Therefore, the behaviour of the discretized Hamiltonian functional (3.13) will be investigated by numerically integrating the dynamics (3.19). Using the AVF scheme (2.12), we obtain the numerical approximation

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + k \left(\frac{\mathbf{v}_{n+1} + \mathbf{v}_n}{2} \right) \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + k \left[\mathcal{L} \left(\frac{\mathbf{u}_n + \mathbf{u}_{n+1}}{2} \right) - \left(\frac{\cos(\mathbf{u}_n) - \cos(\mathbf{u}_{n+1})}{\mathbf{u}_{n+1} - \mathbf{u}_n} \right) \right], \end{aligned} \quad (3.22)$$

where k is the time step, $\mathbf{u}_n$ and $\mathbf{v}_n$ denotes the vectors at discrete time n [5]. The implementation of the AVF scheme (3.22) is carried out using the fixed point method. The code used for the implementation is presented in the Appendix A.2. Figure 3.1 below, shows the evolution of the discretized Hamiltonian function (3.13) using the AVF scheme and the Störmer–Verlet scheme. As shown in proposition 2.3.1, the AVF method conserves the energy exactly, while Störmer–Verlet scheme gives a bounded approximation of the energy.

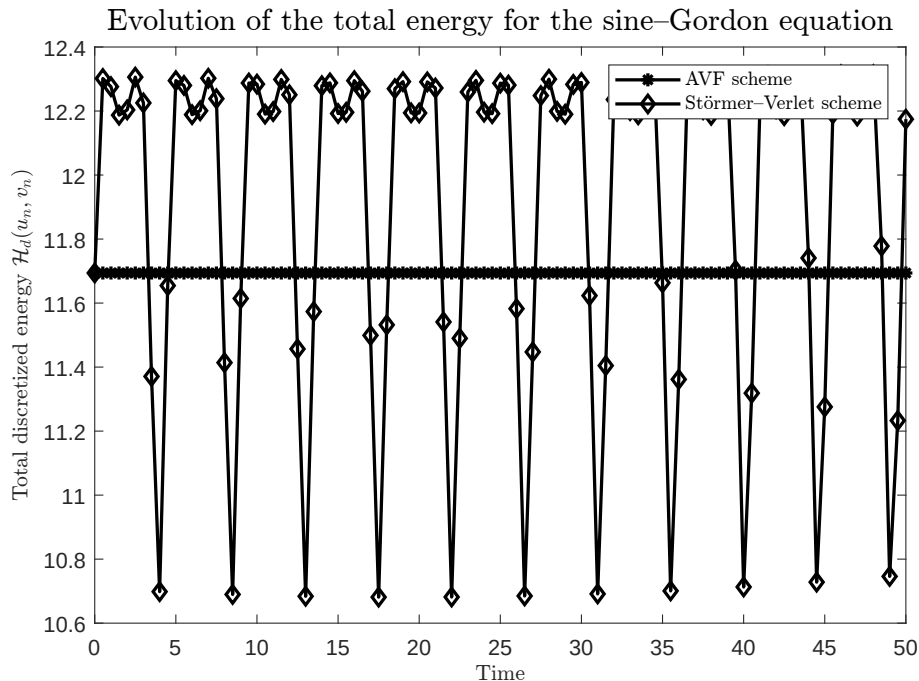


Figure 3.1: Evolution of the discretized total energy (3.13) for the sine-Gordon dynamics (3.19) using the AVF scheme and the Störmer-Verlet scheme for initial energy $\mathcal{H}_d^0 = 11.6940$, initial conditions $(\mathbf{u}_0, \mathbf{v}_0) = (1, 1.5)$, spatial domain $[0, 20]$, number of spatial discretization points $N_{\text{spatial}} = 200$, mesh size $\Delta x = 0.1$, number of time discretization points $N_{\text{time}} = 100$, time step $k = 0.5$, fixed point iteration tolerance $\text{Tol} = 10^{-8}$.

3.1.2 The time-dependent Schrödinger equation

In quantum mechanics, the state of a particle is described by a complex-valued wave function $\psi(x, t)$. This function represents the solution of the time-dependent Schrödinger equation, introduced by Schrödinger in 1926. In fact, it is considered a mathematical function rather than a physical wave, "since it is not observable" [18], containing all the information that can be derived regarding the quantum system. Based on the probabilistic interpretation of Max Born, the quantity $|\psi(x, t)|^2$ represents the probability density of finding the particle in position x at time t [18].

We consider the linear time-dependent Schrödinger equation

$$\begin{cases} i\hbar \frac{\partial}{\partial t} \psi(x, t) = \hat{\mathcal{H}} \psi(x, t), & x \in (-L, L), \quad t > 0 \\ \psi(x, 0) = \psi_0(x), & x \in [-L, L], \end{cases} \quad (3.23)$$

where i is the imaginary unit, $\hbar = \frac{h}{2\pi}$ is the reduced Planck's constant, and $\hat{\mathcal{H}}$ represents the Hamiltonian operator consisting of the total energy of a free particle in space over time, i.e. $\hat{\mathcal{H}} = \frac{-\hbar^2}{2m} \nabla^2$ [18]. Further, this equation is another class of Hamiltonian PDEs introduced as a powerful analog of Newton's second law of motion for a classical system [12]. In fact, the transition from classical to quantum mechanics is nothing but a shift made in the interpretation of the states, from

classical observable and deterministic states to quantum states whose measurement outcomes are probabilistic. In particular, $\hat{\mathcal{H}}$ would correspond to the kinetic energy in classical mechanics.

A significant characteristic of the Schrödinger equation which is shown in Proposition 3.1.1 below is the conservation of the L^2 -norm. In other words, the overall probability

$$\|\psi(x, t)\|_{L^2}^2 = \int |\psi(x, t)|^2 dx \quad (3.24)$$

must remain unchanged during evolution, i.e.

$$\|\psi(x, t)\|_{L^2}^2 = \|\psi(x, 0)\|_{L^2}^2, \quad \forall t \geq 0.$$

Proposition 3.1.1. *Assume that the Hamiltonian operator $\hat{\mathcal{H}}$ for the time-dependent Schrödinger equation is self-adjoint. Then, the time dependent Schrödinger equation has a unique probability conserving solution for all times. That is*

$$\int_{-L}^L \bar{\psi}(x, t) \psi(x, t) dx = 1$$

holds for all t [1].

Proof. Let $\eta(t) = \int_{-L}^L \bar{\psi}(x, t) \psi(x, t) dx$. We need to show $\frac{d}{dt}\eta(t) = 0$.

$$\begin{aligned} \frac{d}{dt}\eta(t) &= \int_{-L}^L \frac{\partial}{\partial t} (\bar{\psi}(x, t) \psi(x, t)) dx \\ &\quad \left[\text{using notation } \psi = \psi(x, t) \right] \\ &= \int_{-L}^L \left(\frac{\partial}{\partial t} \bar{\psi} \psi + \bar{\psi} \frac{\partial \psi}{\partial t} \right) dx. \end{aligned}$$

Plug $\left[\frac{\partial}{\partial t} \psi = -\frac{i}{\hbar} \hat{\mathcal{H}} \psi \right]$ and $\left[\frac{\partial}{\partial t} \bar{\psi} = \frac{i}{\hbar} \hat{\mathcal{H}} \bar{\psi} \right]$ into the previous expression and obtain

$$\frac{d}{dt}\eta(t) = \int_{-L}^L \frac{i}{\hbar} (\hat{\mathcal{H}} \bar{\psi} \psi - \bar{\psi} \hat{\mathcal{H}} \psi) dx = 0.$$

□

3.1.2.1 The Schrödinger equation as a Hamiltonian system

For notational simplicity, consider the wave function $u(x, t) \equiv \psi(x, t)$ such that $u : [-L, L] \times \mathbb{R}_+ \rightarrow \mathbb{C}$ and recall its norm and energy as

$$\|u(x, t)\|_{L^2}^2 = \int_{-L}^L \bar{u}(x, t) u(x, t) dx \quad (3.25)$$

$$\mathcal{H} = \int_{-L}^L \bar{u}(x, t) \hat{\mathcal{H}} u(x, t) dx \quad (3.26)$$

where $\bar{u}$ is the complex conjugate of u , see [2].

For ease of presentation, we set $\hbar = m = 1$, resulting in the Hamiltonian operator $\hat{\mathcal{H}} = -\frac{1}{2}\nabla^2 = -\frac{1}{2}\Delta$. This gives the free-particle Schrödinger equation

$$iu_t = -\frac{1}{2}\Delta u, \quad (3.27)$$

where $\Delta u = \frac{\partial^2}{\partial x^2}$. Since the wave function $u(x, t)$ is a complex function, we can decompose it into two parts, real and complex, as follows

$$u(x, t) = u_{\text{real}}(x, t) + iu_{\text{im}}(x, t). \quad (3.28)$$

Inserting (3.28) in (3.27) gives

$$iu_{t_{\text{real}}} - u_{t_{\text{im}}} = -\frac{1}{2}\Delta u_{\text{real}} - \frac{1}{2}\Delta iu_{\text{im}}.$$

Rearranging terms, the dynamics of the free Schrödinger equation (3.27) can be written as

$$\begin{aligned} u_{t_{\text{real}}} &= -\frac{1}{2}\Delta u_{\text{im}} \\ u_{t_{\text{im}}} &= \frac{1}{2}\Delta u_{\text{real}}. \end{aligned} \quad (3.29)$$

One important observation is that the complex valued wave function and its conjugate can be constructed using u_{real} and u_{im} .

Using (3.26), the Hamiltonian functional for the free Schrödinger equation (3.27) can be derived as follows

$$\mathcal{H}[u(x, t)] = \int_{-L}^L \bar{u} \hat{\mathcal{H}} u dx.$$

Inserting $\hat{\mathcal{H}} = -\frac{1}{2}\Delta$, yields

$$\mathcal{H}[u(x, t)] = -\frac{1}{2} \int_{-L}^L \bar{u} \Delta u(x, t) \, dx.$$

Applying Green's formula and imposing periodic BCs, gives

$$\mathcal{H}[u(x, t)] = \frac{1}{2} \int_{-L}^L \nabla \bar{u} \nabla u dx.$$

This can be written in terms of u_{real} and u_{im} as

$$\mathcal{H}[u(x, t)] = \frac{1}{2} \int_{-L}^L \left[|\nabla u_{\text{real}}|^2 + |\nabla u_{\text{im}}|^2 \right] dx. \quad (3.30)$$

Now, let δ_{real} and δ_{im} denote the variation in u_{real} and u_{im} respectively. Denote the state of the system (3.29) by $\mathbf{z} = (u_{\text{real}}, u_{\text{im}})$. Then, the variational derivative of the Hamiltonian functional $\mathcal{H}$ (3.30), see definition 4.1 in [16], is given by

$$\delta_{\mathbf{z}} \mathcal{H}[u_{\text{real}}, u_{\text{im}}] = \left. \frac{d}{d\varepsilon} \right|_{\varepsilon=0} \mathcal{H}[u_{\text{real}} + \varepsilon \delta_{\text{real}}, u_{\text{im}} + \varepsilon \delta_{\text{im}}]. \quad (3.31)$$

To compute the RHS of (3.31), the terms $|\nabla u_{\text{real}} + \varepsilon \nabla \delta_{\text{real}}|^2$ and $|\nabla u_{\text{im}} + \varepsilon \nabla \delta_{\text{im}}|^2$ are first expanded. Thereafter, the derivative with respect to ε for each term is taken. Finally, setting ε to zero, yields

$$\delta_{\mathbf{z}} \mathcal{H}[u_{\text{real}}, u_{\text{im}}] = \frac{1}{2} \int_{-L}^L \left(2 \nabla u_{\text{real}} \nabla \delta_{\text{real}} + 2 \nabla u_{\text{im}} \nabla \delta_{\text{im}} \right) dx. \quad (3.32)$$

Applying Green's formula to (3.32) together with periodic BCs, gives

$$\delta_{\mathbf{z}} \mathcal{H}[u_{\text{real}}, u_{\text{im}}] = \int_{-L}^L \left((-\Delta u_{\text{real}}) \delta_{\text{real}} + (-\Delta u_{\text{im}}) \delta_{\text{im}} \right) dx.$$

Thus, the variational derivative is given by

$$\begin{bmatrix} \frac{\delta \mathcal{H}}{\delta_{\text{real}}} \\ \frac{\delta \mathcal{H}}{\delta_{\text{im}}} \end{bmatrix} = \begin{bmatrix} -\Delta u_{\text{real}} \\ -\Delta u_{\text{im}} \end{bmatrix},$$

and the system (3.29) is a Hamiltonian system in the form

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{u}_{\text{real}} \\ \dot{u}_{\text{im}} \end{bmatrix} = \frac{1}{2} \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\mathcal{J}} \begin{bmatrix} -\Delta u_{\text{real}} \\ -\Delta u_{\text{im}} \end{bmatrix}. \quad (3.33)$$

Writing the Schrödinger equation as a Hamiltonian system is yet another indicator of having the property of conservative quantities. That is, the solutions of these differential equations are level curves for $\mathcal{H}$, see chapter 2 for ODEs.

3.1.2.2 The AVF scheme for the Schrödinger equation

In section 3.1.2.1 above, we have shown that the system (3.29) is a Hamiltonian system and can be written in the form (3.33). In this section we show that the AVF scheme (2.12) conserves the L^2 -norm of the system (3.29), see Proposition 2.3.1. By defining the spatial domain $[-20, 20]$ which is then discretized in $N = 200$ discretization points with uniform mesh $\Delta x = 0.2$. The dynamics of the free Schrödinger (3.29) can be spatially discretized using the central finite difference scheme

$$u_{xx}(x_i, t) \approx \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{(\Delta x)^2}, \quad \text{for } i = 1, \dots, N. \quad (3.34)$$

This leads to the following dynamics written in vector form

$$\begin{cases} \dot{\mathbf{u}}_{\text{real}}(t) &= -\frac{1}{2} \mathcal{L} \mathbf{u}_{\text{im}}(t), \\ \dot{\mathbf{u}}_{\text{im}}(t) &= \frac{1}{2} \mathcal{L} \mathbf{u}_{\text{real}}(t), \end{cases} \quad (3.35)$$

where $\mathcal{L} \in \mathbb{R}^{N \times N}$ is given by

$$\mathcal{L} = \frac{1}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & 0 & \cdots & 1 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -2 & 1 \\ 1 & 0 & \cdots & 1 & -2 \end{bmatrix}. \quad (3.36)$$

In order to numerically integrate the system of ODEs (3.35), the AVF method is used as in the previous chapters. In this case, we have

$$\begin{aligned} \mathbf{u}_{\text{real}}^{n+1} &= \mathbf{u}_{\text{real}}^n - \frac{\Delta t}{4} \mathcal{L} \left(\mathbf{u}_{\text{im}}^n + \mathbf{u}_{\text{im}}^{n+1} \right), \\ \mathbf{u}_{\text{im}}^{n+1} &= \mathbf{u}_{\text{im}}^n + \frac{\Delta t}{4} \mathcal{L} \left(\mathbf{u}_{\text{real}}^n + \mathbf{u}_{\text{real}}^{n+1} \right). \end{aligned} \quad (3.37)$$

Let us recall that the L^2 -norm of the exact solution

$$\|u(\cdot, t)\|_{L^2}^2 = \int_{-L}^L |u(\cdot, t)|^2 dx = \int_{-L}^L \left(u_{\text{real}}^2(\cdot, t) + u_{\text{im}}^2(\cdot, t) \right) dx, \quad (3.38)$$

is discretized as

$$\|\mathbf{u}^n\|_{L^2}^2 = \Delta x \left[(\mathbf{u}_{\text{real}}^n)^T \mathbf{u}_{\text{real}}^n + (\mathbf{u}_{\text{im}}^n)^T \mathbf{u}_{\text{im}}^n \right]. \quad (3.39)$$

Figure 3.2 below, illustrates the evolution of the discretized L^2 -norm squared (3.39) using the AVF scheme (3.37). The results show that the AVF scheme preserves the L^2 -norm exactly. The code used to implement the AVF scheme (2.12) for the Schrödinger equation (3.29) is presented in Appendix A.6.

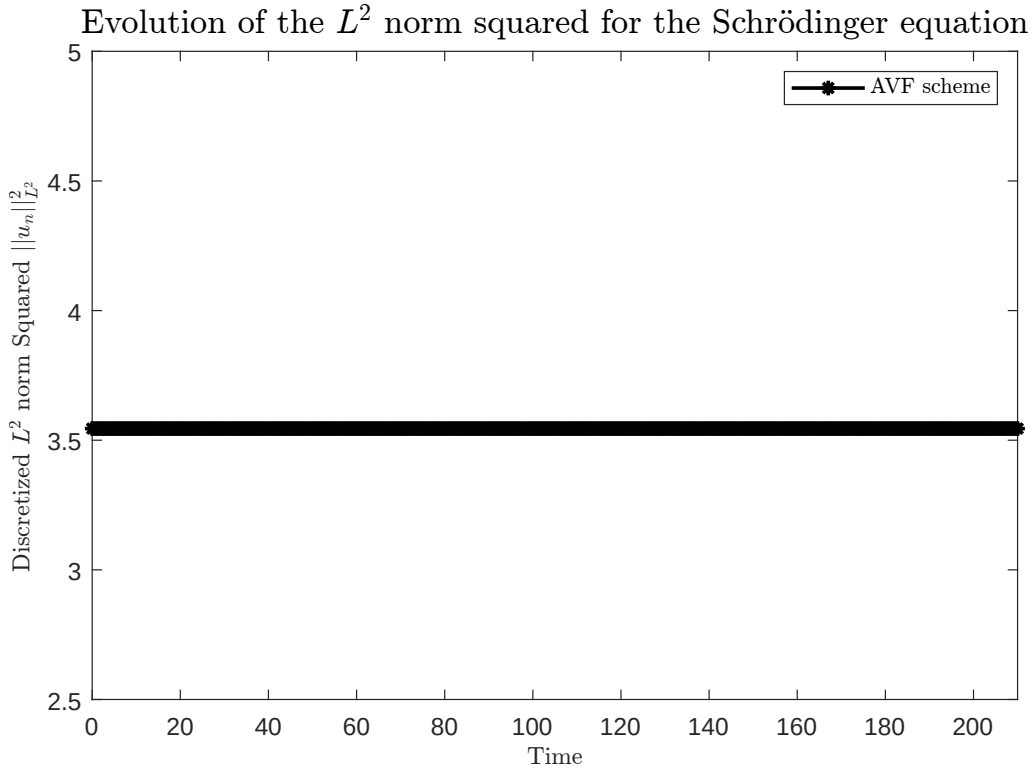


Figure 3.2: Evolution of the L^2 norm squared for the Schrödinger equation (3.23) using the AVF scheme for spatial domain $[-20, 20]$, number of discretization points $N_{\text{spatial}} = 200$, spatial step size $\Delta x = 0.2$, number of discretization points in time $N_{\text{time}} = 300$, time step $\Delta t = 0.7$, initial conditions $(\mathbf{u}_{\text{real}}^0, \mathbf{u}_{\text{im}}^0) = (e^{-\frac{(x-1)^2}{2}}, e^{-\frac{x^2}{2}})$, initial L^2 norm = 3.5449, fixed point iteration tolerance $\text{TOL} = 10^{-6}$.

4

Stochastic differential equations

In the previous chapters, we have examined deterministic Hamiltonian ODEs and PDEs which normally describe predictable systems where energy is conserved. However, many real world scenarios are lead by uncertainty, appearing as random forces or some kind of perturbation. The assets prices for example possess a random behaviour, markets are in a constant change, prices movement are irregular and fundamentally random [20]. In this context, stochastic models tend to play a significant role as they attempt to deal with the unknown, i.e. factors that we cannot predict beforehand. These factors appear also in many other scientific fields such as chemistry and epidemiology, where uncertainty is crucial and requires an educated guess to investigate [8].

This motivates the introduction of stochastic differential equations (SDEs) which lays the foundation needed to extend the deterministic Hamiltonian framework for ODEs and PDEs to Hamiltonian SDEs and Hamiltonian stochastic partial differential equations (SPDEs) in the next chapters. In this chapter, Brownian motion is first defined in section 4.1, which is a fundamental continuous stochastic process that models the cumulative effect of pure noise [13]. We then present the Itô integral and its properties, which is the main building block for defining SDEs, see section 4.2. After that, we give a rigorous definition of SDEs in section 4.3. Thereafter, a numerical method that approximates the solution of these SDEs is introduced, namely the Euler–Maruyama method, see section 4.4. Finally, in section 4.5, an example of SDEs is given where the strong solution is derived using Itô formula and then compared to the numerical solution obtained using the Euler–Maruyama method. For this chapter, we follow the structure presented in [10].

4.1 Brownian Motion

Let T denote the final time of the time domain $[0, T]$. The Brownian motion $B(t)$ is a stochastic process defined for $t \in [0, T]$ equipped with the following properties:

- Its initial value is zero, i.e. $B(0) = 0$ a.s.
- The increments $B(t) - B(s)$ are normally distributed with mean zero and variance $t - s$. That is,

$$B(t) - B(s) \sim \mathcal{N}(0, t - s) \quad \text{for } 0 \leq s < t \leq T. \quad (4.1)$$

- The two random variables $B(t) - B(s)$ and $B(w) - B(v)$ are independent for

$$0 \leq s \leq t \leq v \leq w \leq T.$$

Furthermore, the observed paths of Brownian motion are continuous a.s. throughout the interval $[0, T]$, although they are not differentiable at any point. Moreover, every sample path has quadratic variation on $[0, T]$ equal to T [13].

In numerical simulations, one considers a discretized Brownian motion where $B(t)$ is specified for discrete time values. Let N be the number of discretization points, the time domain $[0, T]$ is then partitioned with time step $\delta t = \frac{T}{N}$. The value of Brownian motion at a discretized time point $t_j = j\delta t$ is given by $B(t_j)$, which we denote B_j for convenience [10]. Therefore, given the definition and properties of the standard Brownian motion mentioned above, one can simulate a Brownian motion path based on

$$B_j = B_{j-1} + dB_j, \quad \text{for } j = 1, 2, \dots, N, \quad (4.2)$$

where $B_0 = 0$ and dB_j is the Brownian motion increment. An important observation is that each Brownian motion increment dB_j in (4.2) is an independent random variable that has the distribution $\sqrt{\delta t}\mathcal{N}(0, 1)$ [11]. Figure 4.1 below, shows a simulation of one path of discretized Brownian motion generated in Matlab using a standard random number generator **randn** as suggested in [10]. The code used is presented in Appendix A.3.

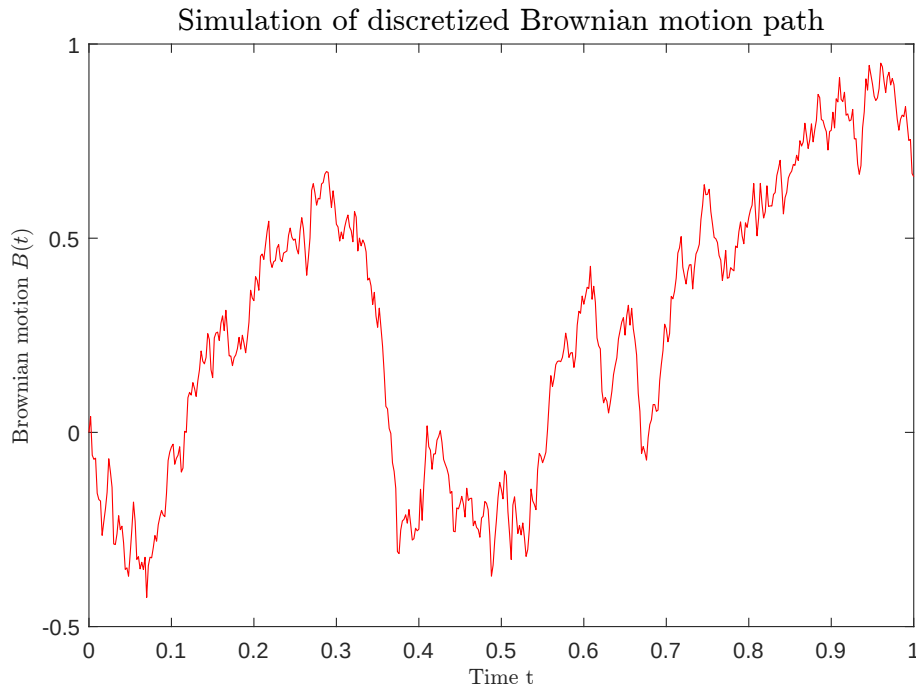


Figure 4.1: Simulation of discretized Brownian motion path using (4.2) for final time $T = 1$ and number of discretization points $N = 500$.

4.2 Itô Integral

The study of the drift and diffusion coefficients of a diffusion process by Itô laid the foundation of stochastic integration and provided a meaningful formulation of

stochastic differential equations [14]. The stochastic integrals are integrals driven by a Brownian motion which is the reason we cannot operate on these integrals within the traditional framework of the classical Riemann calculus. Accordingly, we work within the framework of what is called Itô stochastic calculus.

For ease of presentation, let $h(t)$ be a deterministic function and N the number of discretization points of the time domain $[0, T]$. Using the Riemann sum along partitions of the interval $[0, T]$, the integral $\int_0^T h(t)dt$ can be approximated by

$$\int_0^T h(t)dt \approx \sum_{i=0}^{N-1} h(t_i)(t_{i+1} - t_i). \quad (4.3)$$

The discretized time point t_i in (4.3) is given by $i\delta t$, where $\delta t = \frac{T}{N}$ is the step size. Moreover, the area on each small interval $[t_i, t_{i+1}]$ is approximated by a rectangle with width $t_{i+1} - t_i$ and height $h(t_i)$ that is evaluated at the left end point t_i . Thus, the integral (4.3) is defined by taking $\delta t \rightarrow 0$ in the limit [11].

Following the same analogy, the Itô integral

$$\int_0^T h(t)dB(t), \quad (4.4)$$

can be approximated using

$$\sum_{i=0}^{N-1} h(t_i)(B(t_{i+1}) - B(t_i)). \quad (4.5)$$

Since the Brownian motion increment in (4.5) is a random variable, it follows that the whole expression (4.5) is also a random variable.

In order to show the difference between deterministic calculus and Itô calculus, we now consider the case where the integrand in the Itô integral is a stochastic process. More specifically, a Brownian motion process, that is

$$\int_0^T B(t)dB(t). \quad (4.6)$$

Using (4.5), the integral (4.6) can be approximated as

$$\sum_{i=0}^{N-1} B(t_i)(B(t_{i+1}) - B(t_i)). \quad (4.7)$$

Denoting the Brownian motion increment in (4.7) by dB_{i+1} , applying the definition (4.2) and taking the square of $B(t_{i+1})$, one obtains

$$B(t_{i+1})^2 = B(t_i)^2 + 2B(t_i)dB_{i+1} + (dB_{i+1})^2. \quad (4.8)$$

Rearranging the terms in (4.8) yields

$$B(t_i)dB_i = \frac{1}{2} \left(B(t_{i+1})^2 - B(t_i)^2 - (dB_{i+1})^2 \right). \quad (4.9)$$

Using (4.9), the expression (4.7) can be written as

$$\frac{1}{2} \sum_{i=0}^{N-1} \left[B(t_{i+1})^2 - B(t_i)^2 - (dB_{i+1})^2 \right] = \frac{1}{2} \sum_{i=0}^{N-1} \left(B(t_{i+1})^2 - B(t_i)^2 \right) - \frac{1}{2} \sum_{i=0}^{N-1} (dB_{i+1})^2. \quad (4.10)$$

The first term in the RHS of (4.10) gives $\frac{1}{2}B(T)^2$, given the fact that $B(0) = 0$. Moreover, since the Brownian motion increment has quadratic variation T on $[0, T]$, we obtain in the limit

$$\int_0^T B(t)dB(t) = \frac{1}{2}B(T)^2 - \frac{1}{2}T,$$

which has the extra term $-\frac{1}{2}T$ compared to the deterministic calculus case.

An important property that will be used to derive the expected total energy of a system in the next two chapters is the zero expectation property.

Given that the integrand $h(t)$ depends only on the information available up to time t and is square integrable, i.e. $\int_0^T |h(t)|^2 dt < \infty$, the Itô integral therefore has the necessary assumptions to be well defined. From this, we conclude that the expectation of the Itô integral (4.4)

$$\mathbb{E} \left(\int_0^T h(t)dB(t) \right) = 0,$$

for more details, see [13].

4.3 Stochastic Differential Equations

We recall that ODEs are used to describe and predict the evolution of a deterministic system. An important class of such systems is the Hamiltonian system, which we studied in chapter 2. SDEs incorporate randomness, they arise when coefficients in ODEs are perturbed with white noise process $\xi(t)$ that can be interpreted as the "derivative" of the Brownian motion $B(t)$ [13]. However, since the Brownian motion $B(t)$ is nowhere differentiable, $\xi(t)$ does not exist as a function of time in the usual sense. Instead, we consider $\xi(t)dt$ to be an expression interpreted as Itô integrals w.r.t $B(t)$.

Thus, given a Brownian motion process $B(t)$, an SDE driven by Brownian motion $B(t)$ has the form

$$dX(t) = f(X(t))dt + g(X(t))dB(t), \quad (4.11)$$

where $X(t)$ is an unknown process, f is the drift coefficient and g is the diffusion coefficient. A strong solution of (4.11) is given by

$$X(t) = X(0) + \int_0^t f(X(s))ds + \int_0^t g(X(s))dB(s), \quad \text{a.s.} \quad (4.12)$$

if for all times $t > 0$, both integrals exists [13]. When the diffusion coefficient in the SDE (4.11) is constant, that is, $g(X(t)) = \sigma$, the SDE is said to have additive noise [11].

Having established the necessary background on SDEs, we are now ready to introduce a method for approximating their solutions, namely the Euler–Maruyama method.

4.4 The Euler–Maruyama method

The Euler–Maruyama method is a method that approximates the solution to SDEs. It is an extension of the Euler scheme in which a random kick is added at each iteration [8].

We recall from section 4.3 that a scalar autonomous SDE has the integral form

$$X(t) = X_0 + \int_0^t f(X(s))ds + \int_0^t g(X(s))dB(s), \quad \text{for } 0 \leq t \leq T, \quad (4.13)$$

where the solution $X(t)$ is a random variable for each time t . The SDE (4.13) can be written in differential form as

$$dX(t) = f(X(t))dt + g(X(t))dB(t), \quad \text{for } 0 \leq t \leq T \quad \text{and} \quad X(0) = X_0. \quad (4.14)$$

To apply the Euler–Maruyama method, let N be the number of discretization points and T be the final time. The time domain $[0, T]$ can then be discretized with uniform step size $\Delta t = T/N$, where each discretized step is given by $\tau_j = j\Delta t$.

Let X_j denote the numerical approximation at each discretized time step, the Euler–Maruyama scheme is given by

$$X_j = X_{j-1} + f(X_{j-1})\Delta t + g(X_{j-1})(B(\tau_j) - B(\tau_{j-1})). \quad (4.15)$$

The Brownian motion increment in (4.15) can be computed as shown earlier in section 4.1 by simulating discretized Brownian motion paths [10]. Moreover, the update (4.15) is derived from the fact that if the SDE in integral form (4.13) is taken over the partitioned time domain, that is

$$X(\tau_j) = X(\tau_{j-1}) + \int_{\tau_{j-1}}^{\tau_j} f(X(s))ds + \int_{\tau_{j-1}}^{\tau_j} g(X(s))dB(s), \quad (4.16)$$

then each term on the RHS of (4.16) is approximated by the corresponding term in (4.15) [10].

4.5 Application: Black–Scholes–Merton model

To give an example of an SDE in which the Euler–Maruyama scheme (4.15) is applied to, we consider the Black–Scholes–Merton model. A model for growth with uncertain rate of return, given by the SDE (4.11) with $f(x) = rx$ and $g(x) = \sigma x$, that is,

$$dX(t) = rX(t)dt + \sigma X(t)dB(t). \quad (4.17)$$

In the SDE (4.17) $X(t)$ is the stock price at time t , r is the interest rate and σ is the stock volatility [20].

The solution to (4.17) is given by the following geometric Brownian motion

$$X(t) = X(0) \exp\left(\left(r - \frac{1}{2}\sigma^2\right)t + \sigma B(t)\right), \quad (4.18)$$

which can be obtained by Itô's formula, see Theorem 4.16 in [13]. To derive the solution (4.18), we consider the auxiliary function $w(x) = \ln(x)$, which is twice continuously differentiable function. This allows us to define the following Itô process $Y(t) = w(X(t)) = \ln(X(t))$, which has the following stochastic differential

$$dw(X(t)) = w'(X(t))dX(t) + \frac{1}{2}w''(X(t))d[X, X](t). \quad (4.19)$$

Taking the quadratic variation $d[X, X](t) = (dX(t))^2$ of (4.17), one obtains intuitively

$$(dX(t))^2 = r^2 X(t)^2 \underbrace{dt^2}_0 + 2r\sigma X(t)^2 \underbrace{dt dB(t)}_0 + \sigma^2 X(t)^2 \underbrace{dB(t)^2}_{dt}.$$

Therefore, the stochastic differential (4.19) can be written as

$$dw(X(t)) = w'(X(t))dX(t) + \frac{1}{2}w''(X(t))\sigma^2 X(t)^2 dt. \quad (4.20)$$

Inserting (4.17) into (4.20), one obtains

$$dw(X(t)) = w'(X(t))\left(rX(t)dt + \sigma X(t)dB(t)\right) + \frac{1}{2}w''(X(t))\sigma^2 X(t)^2 dt. \quad (4.21)$$

Rearranging the terms in (4.21), yields

$$\begin{aligned} dw(X(t)) &= \left(w'(X(t))rX(t) + \frac{1}{2}w''(X(t))\sigma^2 X(t)^2\right)dt \\ &\quad + w'(X(t))\sigma X(t)dB(t). \end{aligned} \quad (4.22)$$

For the auxiliary function $w(x) = \ln(x)$, the first and second derivatives are given by $w'(x) = \frac{1}{x}$ and $w''(x) = -\frac{1}{x^2}$. Substituting these into (4.22) gives

$$\begin{aligned} dw(X(t)) &= \left(\frac{1}{X(t)}rX(t) - \frac{1}{2}\frac{1}{X(t)^2}\sigma^2 X(t)^2\right)dt \\ &\quad + \frac{1}{X(t)}\sigma X(t)dB(t). \end{aligned}$$

Thus, the Itô process $Y(t)$ has the following differential form

$$dY(t) = \left(r - \frac{1}{2}\sigma^2\right)dt + \sigma dB(t),$$

with the following integral representation

$$Y(t) - Y(0) = \int_0^t (r - \frac{1}{2}\sigma^2)ds + \int_0^t \sigma dB(s). \quad (4.23)$$

Recalling that $Y(t) = \ln(X(t))$, taking the exponential on both sides of (4.23) and using that $B(0) = 0$ gives the solution (4.18).

Next, the Euler–Maruyama method is used to numerically solve the SDE (4.17) where $r = 2$, $\sigma = 1$, and $X(0) = 1$. The discretized Brownian motion path is computed in the time domain $[0, 1]$ with 2^8 number of discretization points and increment size $\delta t = 2^{-8}$, which is then used to evaluate the exact solution (4.18). Finally, the Euler–Maruyama scheme (4.15) is carried out with step size $\Delta t = 4\delta t$ as done in [10]. Figure 4.2 below, shows the results of the numerical approximation obtained using the Euler–Maruyama scheme (4.15) compared to the exact solution (4.18). The code used in this experiment is presented in Appendix A.3.

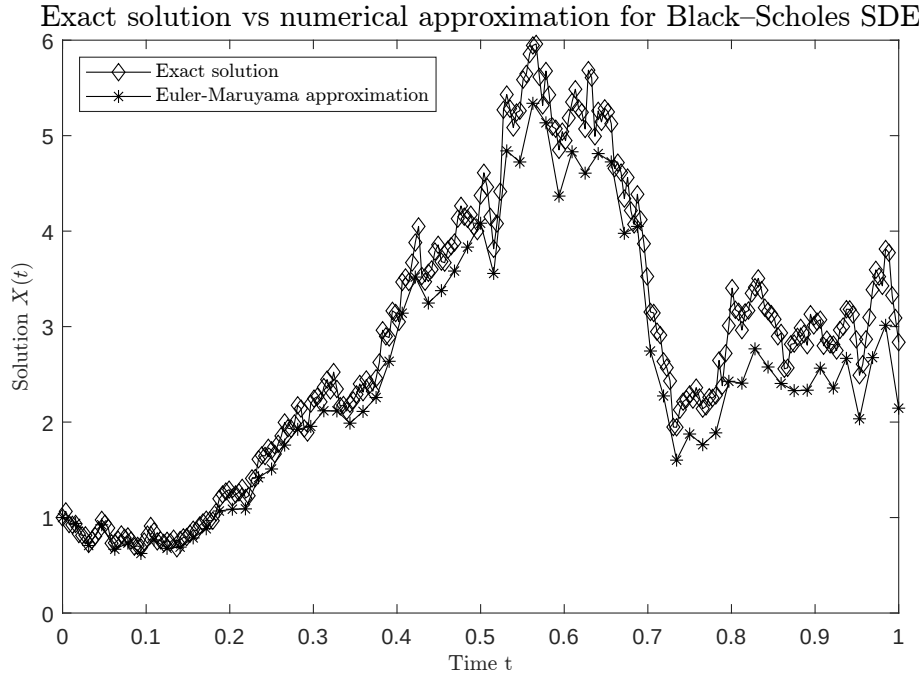


Figure 4.2: Numerical approximation of the SDE (4.17) using the Euler–Maruyama scheme (4.15) compared to the exact solution (4.18), which is evaluated using a discretized Brownian motion path.

5

Hamiltonian stochastic differential equations

In chapter 2, we have seen that the total energy of Hamiltonian ODEs is a conserved quantity, see (2.2). In particular, we studied using the explicit Euler scheme (2.8) and the Störmer–Verlet scheme (2.10) the behaviour of the total energy for the mathematical pendulum as a study case, see figure 2.1. Moreover, we showed that the AVF scheme (2.12) conserves the total energy for each time step, see Proposition 2.3.1.

Using the tools in chapter 4 as a foundation, we can incorporate randomness into the dynamics of Hamiltonian ODEs. This can be done using an additive Brownian motion $\sigma dB(t)$, where σ is a constant representing the noise scale. In the present chapter, we exemplify this by deriving the dynamics of the stochastic mathematical pendulum as system of SDE, see section 5.1. We then study the behaviour of the expected total energy of the stochastic mathematical pendulum, see section 5.2. Finally, in section 5.3, we construct a numerical integrator that exhibits the correct long–time energy behaviour.

5.1 The stochastic pendulum

We introduce randomness into the pendulum system (2.4) as an external force term. Let $\sigma \in \mathbb{R}$ be the diffusion coefficient. By adding an additive Brownian motion $\sigma dB(t)$, the stochastic pendulum can be written as the following system of two SDEs

$$\begin{cases} dq = p dt, & (5.1a) \\ dp = -\sin(q) dt + \sigma dB(t), & (5.1b) \\ q(0) = q_0, \\ p(0) = p_0, \end{cases}$$

where $p(t)$ and $q(t)$ are two unknown stochastic processes.

5.2 The expected total energy of the pendulum

Now that the stochastic dynamics of the mathematical pendulum is derived, We are interested in the behavior of the total energy, that is, the Hamiltonian function

$$H(p(t), q(t)) = \frac{1}{2}p(t)^2 - \cos(q(t)). \quad (5.2)$$

Observe that since $p(t)$ and $q(t)$ are stochastic processes, the Hamiltonian function (5.2) inherits a stochastic differential with two variables. Therefore, we differentiate it using Itô's formula for functions of several variables, see Theorem 4.17 in [13]. The Itô formula for the Hamiltonian function (5.2) can thus be written as

$$dH(p(t), q(t)) = H_p dp + H_q dq + \frac{1}{2}H_{pp}dpdp + \frac{1}{2}H_{qq}dqdq + H_{pq}dpdq. \quad (5.3)$$

Computing the derivative terms in (5.3) gives the following:

$$H_p = p, H_q = \sin(q), H_{pp} = 1, H_{qq} = \cos(q), H_{pq} = 0.$$

Using (5.1a) and (5.1b), one then obtains informally

$$\begin{aligned} (dp(t))^2 &= \sin(q(t))^2 dt^2 - 2 \sin(q(t)) dt dB(t) + \sigma^2 dB(t)^2, \\ (dq(t))^2 &= p(t)^2 dt^2, \\ dp(t)dq(t) &= -p \sin(q(t)) dt^2 + p dt dB(t). \end{aligned}$$

Further, we intuitively use Itô calculus "identities" $dt^2 = 0$, $dt dB(t) = 0$ and $dB(t)^2 = dt$. For more details on how these identities are derived, see [13] and [11].

Substituting all the above into Itô's formula (5.3), yields the following:

$$dH(p(t), q(t)) = \frac{1}{2}\sigma^2 dt + p(t)\sigma dB(t), \quad (5.4)$$

where on the right hand side, the first term is the drift term and the second term is the diffusion term. This shows that the total energy is not exactly conserved for the stochastic pendulum, the thing that we did not encounter when the system contained only a deterministic part, i.e., where

$$dH = H_q dq + H_p dp = 0.$$

However, one could predict that a random change in the Hamiltonian function would alter the settings somehow, and thus the energy conservation property. Therefore, to study the evolution of the total energy under random perturbation we consider the evolution of the expected total energy $\mathbb{E}[H(p(t), q(t))]$.

Integrating (5.4) yields the function

$$H(p(t), q(t)) = H(p_0, q_0) + \frac{1}{2}\sigma^2 \int_0^t ds + \sigma \int_0^t p(s) dB(s), \quad (5.5)$$

where the second integral is an Itô integral. The presence of a Brownian motion implies that each path gives rise to different energy curves and that is why examining the behaviour of this function consists of taking the expectation $\mathbb{E}[H(p(t), q(t))]$, i.e. the average energy over many realizations of the noise.

Taking the expectation of (5.5) and using the zero expectation property presented earlier in chapter 4, one finally obtains

$$\mathbb{E}[H(p(t), q(t))] = \mathbb{E}[H(p_0, q_0)] + \frac{1}{2}\sigma^2 t, \quad (5.6)$$

for all times $t > 0$. This implies that the expected total energy for the stochastic pendulum (5.1) grows linearly with time. Moreover, it is worth mentioning that to incorporate randomness into the system, one might also consider adding an external stochastic force to the two equations in (2.4). However, if an additive Brownian motion is added directly to both the position and the momentum equations, the nice formula (5.6) cannot be derived.

5.3 The Strang–AVF scheme for the stochastic pendulum

In the deterministic ODE settings, we have seen that the AVF scheme (2.12), presented in chapter 2 section 2.3, preserves the total energy exactly for the pendulum (2.3). In the previous section, we have studied the expected total energy for the stochastic pendulum (5.1). It has been shown, using Itô's formula that the expected total energy (5.6) grows linearly in time, which is known as the trace formula for the energy [6].

In this section, we construct a novel numerical integrator scheme, using a combination of the AVF scheme (2.12) and a splitting strategy [9], [3]. We call this scheme the Strang–AVF scheme. We will later see that this method exactly preserves the total energy (5.6) over long-time intervals for our stochastic pendulum (5.1).

The idea behind splitting methods is to decompose the vector field of a given dynamical system into smaller integrable parts, thereby yielding smaller sub-problems that are easier to solve separately [9], [3]. The solutions for those sub-problems are then combined in a certain way, which provides a numerical approximation to the original dynamical system.

5.3.1 Derivation of the Strang–AVF scheme

Recall that the Hamiltonian equations of motion with an additive noise for the stochastic mathematical pendulum are given by

$$\begin{cases} dq = p dt, \\ dp = -\sin(q) dt + \sigma dB(t), \\ q(0) = q_0, \\ p(0) = p_0. \end{cases} \quad (5.7)$$

The vector field in (5.7) can be split into the following two parts

$$f^{[1]} = \begin{bmatrix} p dt \\ -\sin(q) dt \end{bmatrix}, \quad (5.8a)$$

$$f^{[2]} = \begin{bmatrix} 0 \\ \sigma dB(t) \end{bmatrix}, \quad (5.8b)$$

which implies that the system (5.7) can be decomposed into two smaller sub-problems. Those sub-problems can then be solved independently with respect to either the vector field (5.8a) or the vector field (5.8b).

One possible splitting idea is to solve three sub-problems instead of directly solving the system (5.7). The first sub-problem consists of solving exactly the system for the vector field (5.8b), using half the step size $\frac{h}{2}$ and given an initial condition (q_0, p_0) . The obtained solution is then used as an initial condition to solve numerically the second sub-problem for the vector field (5.8a) using the full step size h . Similarly, the obtained solution is then used as an initial condition to solve the third sub-problem, which consists of solving the first sub-problem again for half the step size $\frac{h}{2}$. This method is symmetric and is known as the Strang splitting method [9],[3].

In practice, this means that the following sub-problem

$$dp = \sigma dB(t), \quad (5.9)$$

which is an SDE, is first solved for half the step size $\frac{h}{2}$ using the initial condition p_0 . The SDE (5.9) has the exact solution (after time $\frac{h}{2}$)

$$p_{\frac{h}{2}}^{[2]} = p_0 + \sigma \Delta B_1,$$

where the Brownian motion is simulated as shown earlier in chapter 4, using a discretized Brownian motion path for the time step $\frac{h}{2}$. The obtained solution $(q_{\frac{h}{2}}^{[2]}, p_{\frac{h}{2}}^{[2]})$, where $q_{\frac{h}{2}}^{[2]} = q_0$, is then used as an initial condition to solve the following sub-problem

$$\begin{aligned} \frac{dq}{dt} &= p, \\ \frac{dp}{dt} &= -\sin(q). \end{aligned} \quad (5.10)$$

The second sub-problem (5.10), is deterministic and can be numerically solved using the AVF scheme for the full step size h , see (2.12), according to

$$\begin{aligned} q_h^{[1]} &= q_{\frac{h}{2}}^{[2]} + \frac{h}{2}(p_{\frac{h}{2}}^{[2]} + p_h^{[1]}), \\ p_h^{[1]} &= p_{\frac{h}{2}}^{[2]} - h \left(\frac{\cos(q_{\frac{h}{2}}^{[2]}) - \cos(q_h^{[1]})}{q_h^{[1]} - q_{\frac{h}{2}}^{[2]}} \right). \end{aligned}$$

¹The numbers in square brackets [1] and [2] in the superscript will be used to indicate which part of the partitioned vector field (5.8) a subproblem is being solved for. Similarly, h and $\frac{h}{2}$ in the subscript will be used to indicate which size of the time step is used.

Finally, the obtained solution $(q_h^{[1]}, p_h^{[1]})$ is used as an initial condition to solve the third sub-problem. The third sub-problem consists of solving (5.9) again for half the step size $\frac{h}{2}$, which yields the following exact solution

$$p_{\frac{h}{2}}^{[2]} = p_h^{[1]} + \sigma \Delta B_2.$$

The obtained solution $(q_{\frac{h}{2}}^{[2]}, p_{\frac{h}{2}}^{[2]})$, where $q_{\frac{h}{2}}^{[2]} = q_h^{[1]}$, concludes that a full update of the Strang–AVF scheme

$$\begin{aligned} p_{\frac{h}{2}}^{[2]} &= p_n + \sigma \Delta B_1, \\ q_{\frac{h}{2}}^{[2]} &= q_n, \\ q_h^{[1]} &= q_{\frac{h}{2}}^{[2]} + \frac{h}{2} \left(p_{\frac{h}{2}}^{[2]} + p_h^{[1]} \right), \\ p_h^{[1]} &= p_{\frac{h}{2}}^{[2]} - h \left(\frac{\cos(q_{\frac{h}{2}}^{[2]}) - \cos(q_h^{[1]})}{q_h^{[1]} - q_{\frac{h}{2}}^{[2]}} \right), \\ q_{n+1} &= q_h^{[1]}, \\ p_{n+1} &= p_h^{[1]} + \sigma \Delta B_2, \end{aligned} \tag{5.11}$$

is now completed.

One important observation is that the stochastic sub-problem (5.9) is solved twice for each iteration of the numerical scheme. This means that two independent Brownian motion increments need to be simulated such that

$$\Delta B_1, \Delta B_2 \sim \mathcal{N}(0, \frac{h}{2}),$$

which implies a total increment of $\sim \mathcal{N}(0, h)$ for each iteration of the numerical scheme.

5.3.2 Numerical experiments

To investigate the energy behaviour of the Strang–AVF scheme (5.11) derived earlier in section 5.3.1, we compute the total energy over M numerical realizations of the stochastic pendulum (5.7). For each realization, the stochastic dynamics are evolved using the Strang–AVF scheme (5.11) and the corresponding Hamiltonian function (5.2) is evaluated at each time step. The expected total energy is then approximated by the sample average over M realizations

$$\mathbb{E}[H(p(t_n), q(t_n))] \approx \frac{1}{M} \sum_{m=1}^M H^{(m)}(p_n, q_n). \tag{5.12}$$

This approximation is then used to compare the numerical energy behaviour with the exact result (5.6) using a sufficiently large M to reduce the Monte Carlo error. Figure 5.1 below shows the approximation of the expected total energy for the

stochastic pendulum (5.1). The dynamics is evolved using the constructed Strang–AVF scheme (5.11) and the Euler–Maruyama scheme (4.15), and then the results are compared with the exact expected total energy (5.6).

One observes that, while the Euler–Maruyama scheme (4.15) seems to show a much stronger drift with time, the constructed Strang–AVF scheme (5.11) seems to preserve the behaviour of the exact expected total energy (5.6). The implementation of the constructed Strang–AVF scheme (5.11) and the Euler–Maruyama scheme (4.15) for the stochastic pendulum (5.1) is presented in Appendix A.4.

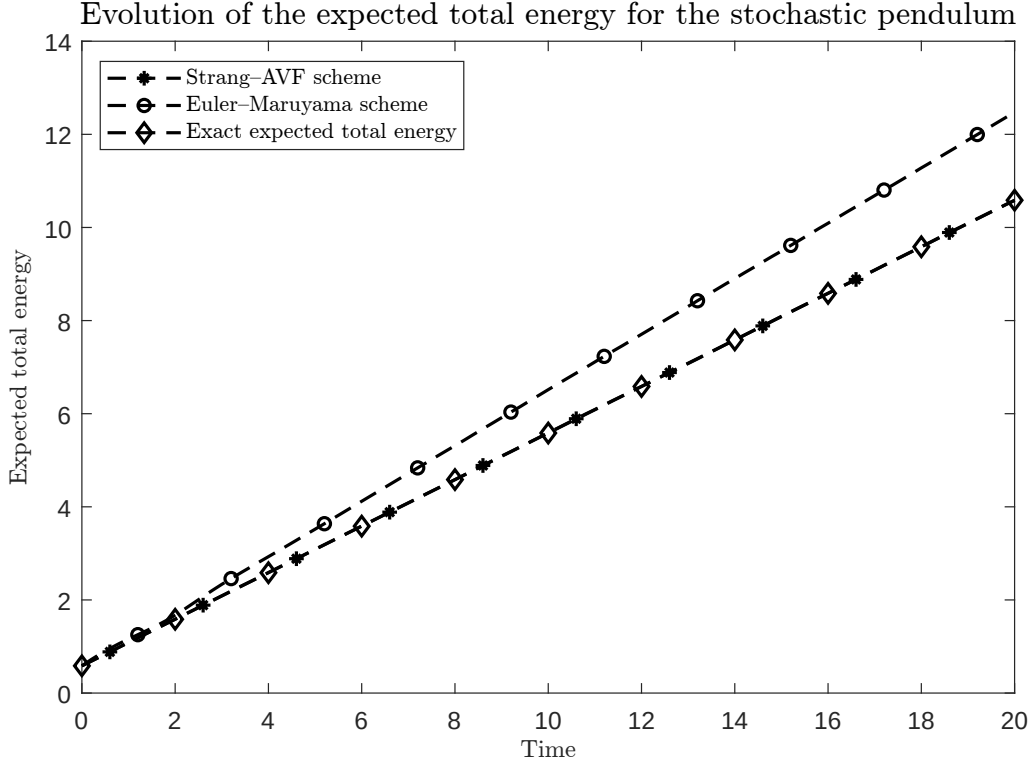


Figure 5.1: Evolution of the expected total energy for the stochastic pendulum (5.7) using the constructed Strang–AVF scheme (5.11) and the Euler–Maruyama scheme (4.15). For final time $T = 20$, number of discretization points $N = 100$, diffusion coefficient $\sigma = 1$, initial condition $(q_0, p_0) = (1, 1.5)$, initial exact total energy $H_0 = 0.5847$, step size $h = \frac{T}{N}$, tolerance for fixed point iteration $\text{TOL} = 10^{-8}$, and number of realizations $M = 10^6$, run time ≈ 2.29 minutes.

6

Hamiltonian stochastic partial differential equations

In this chapter, we build on the framework developed for Hamiltonian SDEs in chapter 5, and extend it by incorporating randomness into the dynamics of Hamiltonian PDEs as an external force. In this setting, we consider a stochastic version of the sine–Gordon equation (3.1), studied in chapter 3, by introducing an additive noise term, namely $\sigma dB(t)$, where σ is the intensity of the noise. A natural question that arises in this context is whether the classical conservation properties, as the ones we discussed earlier in chapter 3, can persist under random perturbation. Therefore, a more detailed discussion regarding this question will be presented in section 6.1.1. We then study the behaviour of the constructed Strang–AVF scheme with respect to the expected total energy, see section 6.1.2. Finally, we analyze the stochastic Schrödinger equation in section 6.2. The chapter concludes with an investigation of the behaviour of the Strang–AVF method applied for the expected value of the L^2 -norm, see section 6.2.1.

6.1 The stochastic sine–Gordon equation

Recall from chapter 3, that the sine–Gordon equation is given by

$$\begin{cases} u_{tt} = u_{xx} - \sin(u), & x \in [0, L], \ t > 0, \\ u(x, 0) = u_0(x), & x \in [0, L], \\ u_t(x, 0) = v_0(x), & x \in [0, L]. \end{cases}$$

where the solution $u = u(x, t)$ is assumed to be smooth in both dependent variables x and t . Furthermore, periodic boundary conditions are imposed.

By introducing the variable $v = u_t$, the stochastic sine–Gordon equation can be obtained as a system of SPDEs

$$du = vdt, \tag{6.1a}$$

$$dv = u_{xx}dt - \sin(u)dt + \sigma dB(t), \tag{6.1b}$$

where $u(x, t)$ and $v(x, t)$ are stochastic processes, $\sigma \in \mathbb{R}$ is the diffusion coefficient and $B(t)$ is the Brownian motion as seen in chapter 4.

6.1.1 Expected total energy for the sine–Gordon equation

We recall that, the Hamiltonian functional for the sine–Gordon equation is given by

$$\mathcal{H} = \mathcal{H}[u, v] = \int_0^L \left[\frac{1}{2}v^2 + \frac{1}{2}u_x^2 - \cos(u) \right] dx. \quad (6.2)$$

In order to study the behaviour of the expected Hamiltonian (6.2), we informally apply Itô’s formula, for more details, see Theorem 4.17 in [13]. Thus, we obtain

$$d\mathcal{H} = \mathcal{H}_u du + \mathcal{H}_v dv + \frac{1}{2}\mathcal{H}_{uu}dudv + \frac{1}{2}\mathcal{H}_{vv}dv dv + \mathcal{H}_{uv}dudv. \quad (6.3)$$

Let du defined in (6.1a) be the variation in u , dv defined in (6.1b) be the variation in v and ε be a small real number. Therefore, under the following perturbation

$$u \rightarrow u + \varepsilon du, \quad v \rightarrow v + \varepsilon dv,$$

the first two terms in the RHS of (6.3) can be computed using the first variation of (6.2). That is,

$$\mathcal{H}_u du + \mathcal{H}_v dv = \left. \frac{d}{d\varepsilon} \right|_{\varepsilon=0} \mathcal{H}[u + \varepsilon du, v + \varepsilon dv].$$

The first variation gives (details are omitted, as this has been done before in section 3.1.1.1)

$$\mathcal{H}_u du + \mathcal{H}_v dv = \int_0^L \left((-u_{xx} + \sin(u))du + v dv \right) dx.$$

Substituting (6.1a) and (6.1b) into the above, yields intuitively

$$\mathcal{H}_u du + \mathcal{H}_v dv = \int_0^L v \sigma dx dB(t). \quad (6.4)$$

Now take the next two terms of (6.3) and compute using the second variation of (6.2). That is,

$$\mathcal{H}_{uu}dudv + \mathcal{H}_{vv}dv dv = \left. \frac{d^2}{d\varepsilon^2} \right|_{\varepsilon=0} \mathcal{H}[u + \varepsilon du, v + \varepsilon dv]. \quad (6.5)$$

Using (6.4), one obtains

$$\begin{aligned} \frac{d^2}{d\varepsilon^2} \left(\frac{1}{2}(v + \varepsilon dv)^2 \right) &= (dv)^2, \\ \frac{d^2}{d\varepsilon^2} \left(\frac{1}{2}(u_x + \varepsilon du_x)^2 \right) &= (du_x)^2, \\ \frac{d^2}{d\varepsilon^2} \left(-\cos(u + \varepsilon du) \right) &= \cos(u + \varepsilon du)(du)^2 \Big|_{\varepsilon=0} = \cos(u)(du)^2. \end{aligned}$$

Combining the results, we have the expression

$$\mathcal{H}_{uu}dudv + \mathcal{H}_{vv}dvdu = \int_0^L \left((dv)^2 + (du_x)^2 + \cos(u)(du)^2 \right) dx.$$

Using the Itô calculus "identities" $(dt)^2 = 0$, $dt dB(t) = 0$ and $(dB(t))^2 = dt$, one obtains intuitively

$$\begin{aligned} (dv)^2 &= \sigma^2 dt, \\ (du_x)^2 &= 0, \\ \cos(u)(du)^2 &= 0. \end{aligned}$$

Therefore, the second two terms in the Itô formula (6.3) contribute with

$$\frac{1}{2} \int_0^L \sigma^2 dt dx. \quad (6.6)$$

Finally, the last term $\mathcal{H}_{uv}dudv$ in the RHS of (6.3) computes to zero due to the Itô calculus "identities" $dt^2 = 0$ and $dt dB(t) = 0$. Combining all the computed terms above (6.4) and (6.6), the equation (6.3) can thus be written as

$$d\mathcal{H} = \sigma \int_0^L v dx dB(t) + \frac{1}{2} \int_0^L \sigma^2 dx dt. \quad (6.7)$$

Integrating (6.7) in time, one obtains

$$\mathcal{H}[u(x, t), v(x, t)] = \mathcal{H}[u(x, 0), v(x, 0)] + \sigma \int_0^t \left(\int_0^L v(x, s) dx \right) dB(s) + \frac{1}{2} L \sigma^2 t, \quad (6.8)$$

where the integral is an Itô integral. Taking the expectation of (6.8) and using the zero expectation property of Itô's integral (for more details see, [13]), yields

$$\mathbb{E}[\mathcal{H}[u(x, t), v(x, t)]] = \mathbb{E}[\mathcal{H}[u(x, 0), v(x, 0)]] + \frac{1}{2} L \sigma^2 t. \quad (6.9)$$

This expression implies that the expected total energy of the sine-Gordon equation (6.1) grows linearly in time. This is the same behaviour that we encountered in chapter 5, see (5.6), for SDE.

6.1.2 The Strang–AVF scheme for the stochastic sine–Gordon equation

In section 5.3, we constructed a numerical integrator scheme based on Strang splitting method [3], [9], and the AVF scheme (2.12) for the stochastic pendulum (5.1). In this section, the same numerical integrator scheme is used to numerically investigate the expected total energy for the stochastic sine–Gordon equation (6.1).

We use the finite spatial discretization from chapter 3 and we find that the semi-discrete form of the stochastic sine–Gordon equation is given by

$$\begin{cases} d\mathbf{u} = \mathbf{v}dt, \\ d\mathbf{v} = (\mathcal{L}\mathbf{u} - \sin(\mathbf{u}))dt + \Sigma dB(t), \\ \mathbf{u}(x, 0) = \mathbf{u}_0, \\ \mathbf{v}(x, 0) = \mathbf{v}_0. \end{cases} \quad (6.10)$$

Σ in (6.10), is modeled as

$$\Sigma = \sigma \mathbf{1}_N = \sigma \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \vdots \\ \sigma \end{pmatrix} \in \mathbb{R}^N. \quad (6.11)$$

$\mathcal{L} \in \mathbf{R}^{N \times N}$ is the 3-point stencil operator given by (3.20). In order to define the Strang-splitting scheme, we decompose the vector field in (6.10) into

$$f^{[1]} = \begin{bmatrix} \mathbf{v}dt \\ (\mathcal{L}\mathbf{u} - \sin(\mathbf{u}))dt \end{bmatrix}, \quad (6.12a)$$

$$f^{[2]} = \begin{bmatrix} 0 \\ \Sigma dB(t) \end{bmatrix}. \quad (6.12b)$$

Therefore, the Strang-AVF scheme is used to solve two smaller sub-problems instead of directly solving the system (6.10). The following sub-problem

$$d\mathbf{v} = \Sigma B(t), \quad (6.13)$$

is first solved exactly for half the step size $\frac{h}{2}$ using the initial condition $\mathbf{v}_0$. The SDE (6.13) has the exact solution (after time $\frac{h}{2}$)

$$\mathbf{v}_{\frac{h}{2}}^{[2]} = \mathbf{v}_0 + \Sigma \Delta B_1,$$

where the Brownian increment ΔB_1 is simulated as shown earlier in section 4.1 s.t. $\Delta B_1 \sim \mathcal{N}(0, \frac{h}{2})$. The obtained solution $\mathbf{v}_{\frac{h}{2}}^{[2]}$ is then used together with the initial condition $(\mathbf{u}_0, \mathbf{v}_{\frac{h}{2}}^{[2]})$ to solve numerically the following deterministic sub-problem

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{v}, \\ \frac{d\mathbf{v}}{dt} &= \mathcal{L}\mathbf{u} - \sin(\mathbf{u}). \end{aligned} \quad (6.14)$$

The sub-problem (6.14), is then approximated for the full step size h using the AVF scheme (2.12)

$$\begin{aligned} \mathbf{u}_h^{[1]} &= \mathbf{u}_0 + h \left(\frac{\mathbf{v}_h^{[1]} + \mathbf{v}_{\frac{h}{2}}^{[2]}}{2} \right) \\ \mathbf{v}_h^{[1]} &= \mathbf{v}_{\frac{h}{2}}^{[2]} + h \left[\mathcal{L} \left(\frac{\mathbf{u}_0 + \mathbf{u}_h^{[1]}}{2} \right) - \left(\frac{\cos(\mathbf{u}_0) - \cos(\mathbf{u}_h^{[1]})}{\mathbf{u}_h^{[1]} - \mathbf{u}_0} \right) \right]. \end{aligned}$$

The obtained solution $(\mathbf{u}_h^{[1]}, \mathbf{v}_h^{[1]})$ is used as an initial condition to solve the third sub-problem with half the step size $\frac{h}{2}$. This third sub-problem consists of solving the stochastic sub-problem (6.13) again, which yields the following exact solution (after time $\frac{h}{2}$)

$$\mathbf{v}_{\frac{h}{2}}^{[2]} = \mathbf{v}_h^{[1]} + \Sigma \Delta B_2,$$

where ΔB_2 is the Brownian motion increment that is simulated as shown earlier in section 4.1 s.t. $\Delta B_2 \sim \mathcal{N}(0, \frac{h}{2})$. Finally, the obtained solution $(\mathbf{u}_{\frac{h}{2}}^{[2]}, \mathbf{v}_{\frac{h}{2}}^{[2]})$, where $(\mathbf{u}_{\frac{h}{2}}^{[2]}, = \mathbf{u}_h^{[1]})$, concludes that a full update of the Strang–AVF scheme

$$\begin{aligned} \mathbf{u}_{\frac{h}{2}}^{[2]} &= \mathbf{u}_n, \\ \mathbf{v}_{\frac{h}{2}}^{[2]} &= \mathbf{v}_n + \Sigma \Delta B_1, \\ \mathbf{u}_h^{[1]} &= \mathbf{u}_{\frac{h}{2}}^{[2]} + h \left(\frac{\mathbf{v}_h^{[1]} + \mathbf{v}_{\frac{h}{2}}^{[2]}}{2} \right), \\ \mathbf{v}_h^{[1]} &= \mathbf{v}_{\frac{h}{2}}^{[2]} + h \left[\mathcal{L} \left(\frac{\mathbf{u}_{\frac{h}{2}}^{[2]} + \mathbf{u}_h^{[1]}}{2} \right) - \left(\frac{\cos(\mathbf{u}_{\frac{h}{2}}^{[2]}) - \cos(\mathbf{u}_h^{[1]})}{\mathbf{u}_h^{[1]} - \mathbf{u}_{\frac{h}{2}}^{[2]}} \right) \right], \\ \mathbf{u}_{n+1} &= \mathbf{u}_h^{[1]}, \\ \mathbf{v}_{n+1} &= \mathbf{v}_h^{[1]} + \Sigma \Delta B_2, \end{aligned} \tag{6.15}$$

is now completed. The same observation regarding the two independent Brownian motion increment ΔB_1 , ΔB_2 that are used holds here as well. Namely, the total increment for each Strang–AVF iteration should be $\sim \mathcal{N}(0, h)$.

6.1.3 Numerical experiments

To investigate the behaviour of the Strang–AVF scheme (6.15) derived earlier in section 6.1.2, the total energy over M realizations of the stochastic sine–Gordon equation (6.1) is computed. This is done by first evolving the stochastic dynamics of the sine–Gordon equation (6.1a) and (6.1b) using the Strnag–AVF scheme (6.15). Thereafter, the total energy is evaluated at each time step. Finally the expected total energy is then approximated by the sample average over M realizations. Figure 6.1 below, shows that the Euler–Maruyama scheme (4.15) shows a faster increase in the expected total energy. While, the constructed Strang–AVF scheme (6.15) seems to preserve the behaviour of the exact total energy for the sine–Gordon equation (6.9). The code used to implement the constructed Strang–AVF scheme (6.15) and Euler–Maruyama scheme (4.15) for the semi-discrete sine–Gordon equation (6.10) is presented in Appendix A.5.

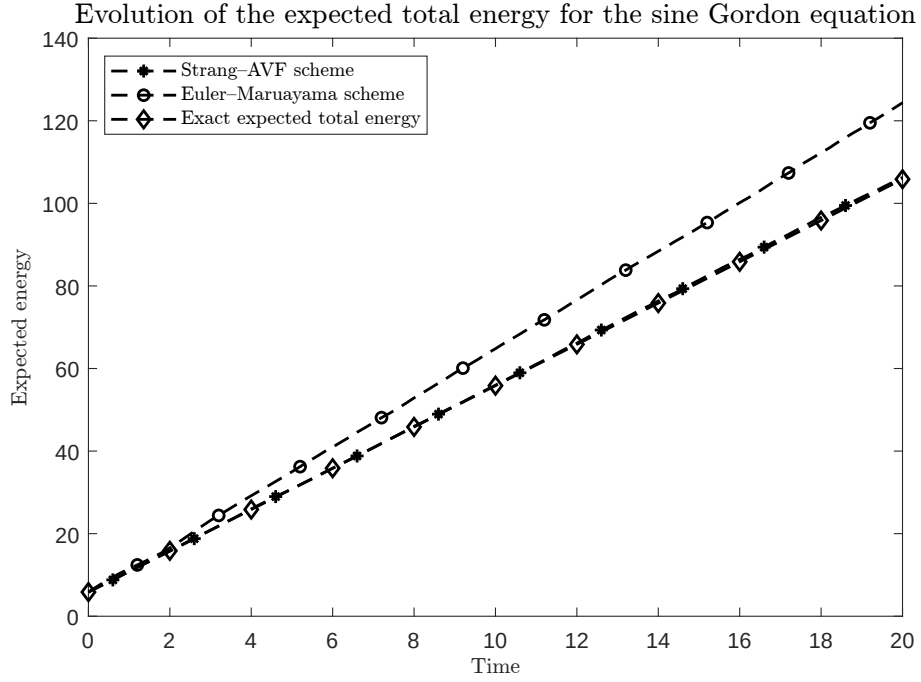


Figure 6.1: Evolution of the expected total energy for the stochastic sine-Gordon equation (6.1) using the constructed Strang-AVF scheme (6.15) and Euler-Maruyama scheme (4.15). For final time $T = 20$, number of discretization points of the time domain $N_{\text{time}} = 100$, time step size $h = \frac{T}{N_{\text{time}}}$, spatial domain $[0, L] = [0, 10]$, number of discretization points of spatial domain $N_{\text{spatial}} = 150$, spatial mesh $\Delta x = \frac{L}{N_{\text{spatial}}}$, initial condition $(\mathbf{u}_0, \mathbf{v}_0) = (1, 1.5)$, initial energy $\mathcal{H}_0 = 5.8470$, tolerance for fixed point iteration $\text{TOL} = 10^{-8}$, number of realizations $M = 10^5$, run time ≈ 27.18 minutes.

6.2 Long-time behaviour of the stochastic Schrödinger equation

We consider the stochastic Schrödinger equation

$$\begin{cases} i\partial_t u = -\frac{1}{2}\Delta u dt + \sigma dB(t), & x \in [-L, L], t > 0, \\ u(x, 0) = u_0(x), & x \in [-L, L]. \end{cases} \quad (6.16)$$

where σ is the noise intensity and $B(t)$ is a Brownian motion process. Decomposing $u(x, t)$ into real and complex parts, as illustrated in section 3.1.2.1, yields the stochastic system

$$\begin{aligned} du_{\text{real}} &= -\frac{1}{2}\Delta u_{\text{im}} dt \\ du_{\text{im}} &= \frac{1}{2}\Delta u_{\text{real}} dt - \sigma dB(t). \end{aligned} \quad (6.17)$$

Denoting the L^2 -norm given in (3.38) by $\eta(t)$ and informally applying Itô's formula, we obtain

$$\begin{aligned} d\eta(t) &= \eta_{u_{\text{real}}} du_{\text{real}} + \eta_{u_{\text{im}}} du_{\text{im}} \\ &\quad + \frac{1}{2} \eta_{u_{\text{real}} u_{\text{real}}} du_{\text{real}} du_{\text{real}} \\ &\quad + \frac{1}{2} \eta_{u_{\text{im}} u_{\text{im}}} du_{\text{im}} du_{\text{im}} \\ &\quad + \eta_{u_{\text{real}} u_{\text{im}}} du_{\text{real}} du_{\text{im}}. \end{aligned} \quad (6.18)$$

Computing the derivatives $\eta_{u_{\text{real}}}$, $\eta_{u_{\text{im}}}$, $\eta_{u_{\text{im}} u_{\text{im}}}$ and inserting the stochastic dynamics (6.17) in (6.18) yields

$$\begin{aligned} d\eta(t) &= -\int_{-L}^L u_{\text{real}} \Delta u_{\text{im}} dx dt \\ &\quad + \int_{-L}^L u_{\text{im}} \Delta u_{\text{real}} dx dt - 2u_{\text{im}} \sigma dx dB(t) \\ &\quad + \int_{-L}^L \sigma^2 dx dt. \end{aligned} \quad (6.19)$$

Rearranging the terms in (6.19) gives

$$\begin{aligned} d\eta(t) &= \left[\int_{-L}^L (-u_{\text{real}} \Delta u_{\text{im}} + u_{\text{im}} \Delta u_{\text{real}} + \sigma^2) dx \right] dt \\ &\quad - 2 \int_{-L}^L u_{\text{im}} \sigma dx dB(t). \end{aligned} \quad (6.20)$$

Using Green's formula together with periodic BCs, (6.20) can be reduced to

$$d\eta(t) = \int_{-L}^L \sigma^2 dx dt - 2 \int_{-L}^L u_{\text{im}}(x, t) \sigma dx dB(t),$$

which can be written in integral form as

$$\eta(t) = \eta(0) + \int_0^t \int_{-L}^L \sigma^2 dx ds - 2 \int_0^t \left(\int_{-L}^L u_{\text{im}}(x, s) \sigma dx \right) dB(s). \quad (6.21)$$

Finally, taking the expectation of (6.21) gives the relation

$$\mathbb{E}[\eta(t)] = \mathbb{E}[\eta(0)] + 2L\sigma^2 t \quad (6.22)$$

since the expectation of the Itô integral is zero and σ^2 is a constant. The expression (6.22) shows that the expected value of the L^2 -norm of the stochastic Schrödinger (6.17) grows linearly with time. This behaviour can be compared to the behaviour of the energy (6.9) of the stochastic sine-Gordon equation (6.1).

6.2.1 Strang–AVF scheme for the stochastic Schrödinger

In this section we aim to study the behaviour of the Strang–AVF scheme regarding the property (6.22). Using the finite spatial discretization from section 3.1.2.2, the stochastic Schrödinger (6.17) can be written in vector form as

$$\begin{cases} d\mathbf{u}_{\text{real}} = -\frac{1}{2}\mathcal{L}\mathbf{u}_{\text{im}}dt, \\ d\mathbf{u}_{\text{im}} = \frac{1}{2}\mathcal{L}\mathbf{u}_{\text{real}}dt - \Sigma dB(t), \\ \mathbf{u}_{\text{real}}(x, 0) = \mathbf{u}_{\text{real}}^0, \\ \mathbf{u}_{\text{im}}(x, 0) = \mathbf{u}_{\text{im}}^0, \end{cases} \quad (6.23)$$

where Σ is modeled as in (6.11), $\mathcal{L} \in \mathbf{R}^{N \times N}$ is the 3-point stencil operator given by (3.20). Decomposing the vector field in (6.23) yields

$$f^{[1]} = \begin{bmatrix} -\frac{1}{2}\mathcal{L}\mathbf{u}_{\text{im}}dt \\ \frac{1}{2}\mathcal{L}\mathbf{u}_{\text{real}}dt \end{bmatrix}, \quad (6.24a)$$

$$f^{[2]} = \begin{bmatrix} 0 \\ -\Sigma dB(t) \end{bmatrix}. \quad (6.24b)$$

Therefore, using the vector field (6.24b) the following stochastic subproblem

$$d\mathbf{u}_{\text{im}} = -\Sigma dB(t), \quad (6.25)$$

is first solved exactly for half the step size $\frac{h}{2}$. Given two independent Brownian motion increments $\Delta B_1, \Delta B_2 \sim \mathcal{N}(0, \frac{h}{2})$, the initial condition $\mathbf{u}_{\text{im}}^0$, the exact solution reads

$$\mathbf{u}_{\text{im}, \frac{h}{2}}^{[2]} = \mathbf{u}_{\text{im}}^0 - \Sigma \Delta B_1.$$

The solution obtained $\mathbf{u}_{\text{im}, \frac{h}{2}}^{[2]}$ is then used as an initial condition to solve the second subproblem together with $\mathbf{u}_{\text{real}, \frac{h}{2}}^{[2]} = \mathbf{u}_{\text{real}}^0$. That is, the initial condition $(\mathbf{u}_{\text{real}, \frac{h}{2}}^{[2]}, \mathbf{u}_{\text{im}, \frac{h}{2}}^{[2]})$ is used to solve numerically the deterministic subproblem for the vector field (6.24a), for the full step size h , using the AVF scheme

$$\begin{aligned}
 \mathbf{u}_{\text{real},h}^{[1]} &= \mathbf{u}_{\text{real},\frac{h}{2}}^{[2]} - \frac{h}{4} \mathcal{L} \left(\mathbf{u}_{\text{im},h}^{[1]} + \mathbf{u}_{\text{im},\frac{h}{2}}^{[2]} \right), \\
 \mathbf{u}_{\text{im},h}^{[1]} &= \mathbf{u}_{\text{im},\frac{h}{2}}^{[2]} + \frac{h}{4} \mathcal{L} \left(\mathbf{u}_{\text{real},\frac{h}{2}}^{[2]} + \mathbf{u}_{\text{real},h}^{[2]} \right).
 \end{aligned} \tag{6.26}$$

The obtained solution $(\mathbf{u}_{\text{real},h}^{[1]}, \mathbf{u}_{\text{im},h}^{[1]})$ is then used as initial condition to solve exactly the stochastic subproblem (6.25) again for half the step size $\frac{h}{2}$, the exact solution in this case reads

$$\mathbf{u}_{\text{im},\frac{h}{2}}^{[2]} = \mathbf{u}_{\text{im},h}^{[1]} - \Sigma \Delta B_2. \tag{6.27}$$

Finally, the obtained solution $(\mathbf{u}_{\text{real},\frac{h}{2}}^{[2]}, \mathbf{u}_{\text{im},\frac{h}{2}}^{[2]})$, where $(\mathbf{u}_{\text{real},\frac{h}{2}}^{[2]} = \mathbf{u}_{\text{real},h}^{[1]})$, concludes that a full update of the Strang–AVF scheme

$$\begin{aligned}
 \mathbf{u}_{\text{real},\frac{h}{2}}^{[2]} &= \mathbf{u}_{\text{real}}^n, \\
 \mathbf{u}_{\text{im},\frac{h}{2}}^{[2]} &= \mathbf{u}_{\text{im}}^n - \Sigma \Delta B_1, \\
 \mathbf{u}_{\text{real},h}^{[1]} &= \mathbf{u}_{\text{real},\frac{h}{2}}^{[2]} - \frac{h}{4} \mathcal{L} \left(\mathbf{u}_{\text{im},h}^{[1]} + \mathbf{u}_{\text{im},\frac{h}{2}}^{[2]} \right), \\
 \mathbf{u}_{\text{im},h}^{[1]} &= \mathbf{u}_{\text{im},\frac{h}{2}}^{[2]} + \frac{h}{4} \mathcal{L} \left(\mathbf{u}_{\text{real},\frac{h}{2}}^{[2]} + \mathbf{u}_{\text{real},h}^{[2]} \right), \\
 \mathbf{u}_{\text{real}}^{n+1} &= \mathbf{u}_{\text{real},h}^{[1]}, \\
 \mathbf{u}_{\text{im}}^{n+1} &= \mathbf{u}_{\text{im},h}^{[1]} - \Sigma \Delta B_2,
 \end{aligned} \tag{6.28}$$

is completed. The same observation as before holds here regarding the two independent Brownian motion increment that are used, namely the total increment for each Strang–AVF iteration should be $\sim \mathcal{N}(0, h)$.

6.2.2 Numerical experiments

After evolving the dynamics of the stochastic Schrödinger (6.23) using the Strang–AVF scheme (6.28) derived earlier in section 6.2.1, the average of the discretized L^2 -norm (3.39) over M number of realizations is computed and compared to the exact expectation of the L^2 -norm (6.22) for $L = 20$. Figure 6.2 below shows the evolution of the expected value of the L^2 -norm squared. It shows that the constructed Strang–AVF scheme (6.28) preserves the exact expectation of the L^2 -norm (6.22). The code used to implement the constructed Strang–AVF scheme (6.28) for the stochastic Schrödinger equation (6.23) is presented in Appendix A.6.

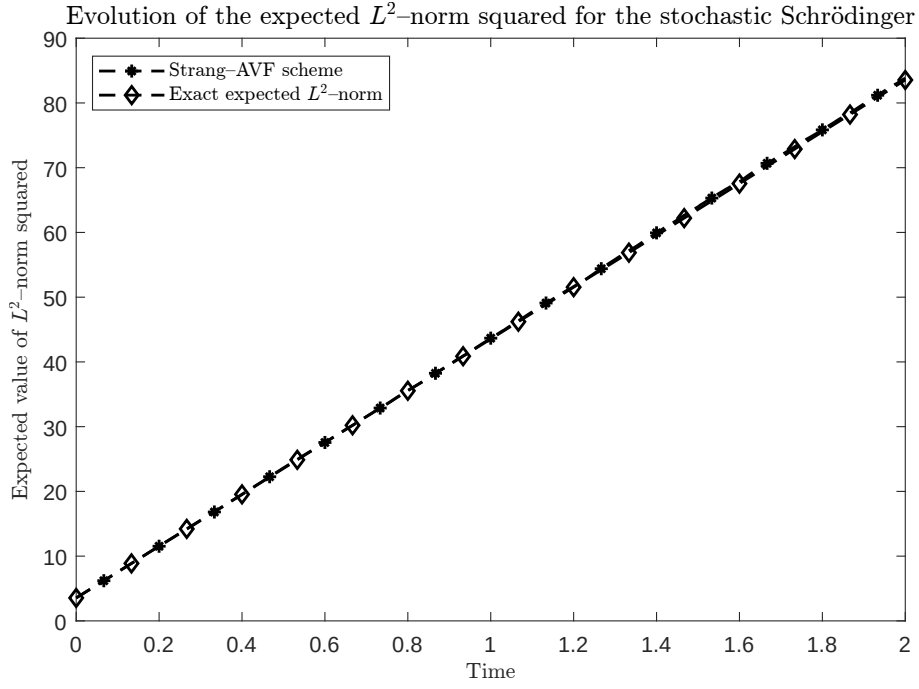


Figure 6.2: Evolution of the expected value of the discretized L^2 norm squared (3.39) for the Stochastic Schrödinger equation (6.17) equation using Strang-AVF scheme (6.28), for final time $T = 2$, number of discretization points of the time domain $N_{\text{time}} = 300$, spatial domain $[-20, 20]$, number of discretization points of spatial domain $N_{\text{spatial}} = 200$, initial conditions $(\mathbf{u}_{\text{real}}^0, \mathbf{u}_{\text{im}}^0) = (e^{-\frac{(x-1)^2}{2}}, e^{-\frac{x^2}{2}})$, initial L^2 norm squared $M_0 = 3.5449$, time step size $h = \frac{T}{N_{\text{time}}}$, spatial step size $\Delta x = \frac{40}{N_{\text{spatial}}}$, $M = 10^5$ number of realizations, run time ≈ 52.70 minutes.

7

Conclusion

As our research has demonstrated, a consistent long-time behaviour of Hamiltonian SPDEs can be achieved. We have seen that the property of the Hamiltonian function, being a conserved quantity, within the deterministic framework of Hamiltonian ODEs and Hamiltonian PDEs has been studied analytically and examined numerically. The AVF numerical scheme is shown to exactly preserve this property for longer time intervals. See, figures 2.2 and 3.1 as well as the L^2 -norm property, see figure 3.2. Having this as a foundation together with some mathematical tools from stochastic calculus, we have succeeded in bridging the gap between the deterministic Hamiltonian framework and the stochastic framework of Hamiltonian SDEs and Hamiltonian SPDEs. In particular, we have shown that the expected value of the Hamiltonian function for the stochastic pendulum as well as the stochastic sine-Gordon equation drifts linearly in time, see equations (5.6) and (6.9). A similar behaviour for the stochastic linear Schrödinger equation regarding the L^2 -norm has also been shown, see equation (6.22). Moreover, we were able to derive a numerical integrator scheme that we called Strang-AVF scheme which successfully conserves the long-time behaviour of the Hamiltonian function along with the long-time behaviour of the L^2 -norm.

Although this thesis has achieved its objective. Future work could explore many directions. In this project, we studied the evolution of Hamiltonian PDEs driven by an additive one dimensional Brownian motion. Another alternative approach would be to investigate these systems using multiplicative Brownian motion instead. Future research could also investigate the long-time behaviour of Hamiltonian PDEs driven by alternative noise processes. In addition, we could also try to work with different versions of equations. As for the Schrödinger equation, it would be interesting to examine the Schrödinger equation with potential, the many particle Schrödinger equation, or even the non-linear Schrödinger equation. Our work was carried out within the Itô framework, however, our objective could be investigated within other stochastic framework, such as the Stratonovich framework. Regarding our numerical experiment with the Strang splitting method, a different splitting strategy would also be something to.

Bibliography

- [1] V. S. Araujo, F. A. B. Coutinho, and F. Toyama. The time-dependent Schrödinger equation: the need for the Hamiltonian to be self-adjoint. *Brazilian Journal of Physics*, 38(1):178–187, 2008.
- [2] S. Blanes and F. Casas. *A concise introduction to geometric numerical integration*. Monographs and Research Notes in Mathematics. CRC Press, Boca Raton, FL, 2016.
- [3] S. Blanes, F. Casas, and A. Murua. Splitting methods for differential equations. *Acta Numer.*, 33:1–161, 2024.
- [4] R. L. Burden, J. D. Faires, and A. C. Reynolds. *Numerical analysis*. Prindle, Weber & Schmidt, Boston, MA, 1978.
- [5] E. Celledoni, V. Grimm, R. I. McLachlan, D. I. McLaren, D. O’Neale, B. Owren, and G. R. W. Quispel. Preserving energy resp. dissipation in numerical PDEs using the “average vector field” method. *J. Comput. Phys.*, 231(20):6770–6789, 2012.
- [6] D. Cohen and G. Vilmart. Drift-preserving numerical integrators for stochastic Poisson systems. *Int. J. Comput. Math.*, 99(1):4–20, 2022.
- [7] J. Cuevas-Maraver, P. G. Kevrekidis, and F. Williams, editors. *The sine-Gordon model and its applications*, volume 10 of *Nonlinear Systems and Complexity*. Springer, Cham, 2014. From pendula and Josephson junctions to gravity and high-energy physics.
- [8] D. F. Griffiths and D. J. Higham. *Numerical methods for ordinary differential equations*. Springer Undergraduate Mathematics Series. Springer-Verlag London, Ltd., London, 2010. Initial value problems.
- [9] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.
- [10] D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Rev.*, 43(3):525–546, 2001.
- [11] D. J. Higham and P. E. Kloeden. *An introduction to the numerical simulation of stochastic differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, [2021] ©2021.
- [12] X. Huang. How did Schrödinger obtain the Schrödinger equation. *arXiv preprint arXiv:1206.0055*, 2012.
- [13] F. C. Klebaner. *Introduction to stochastic calculus with applications*. Imperial College Press, London, third edition, 2012.

- [14] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1992.
- [15] B. Leimkuhler and S. Reich. *Simulating Hamiltonian dynamics*, volume 14 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2004.
- [16] P. J. Olver. *Applications of Lie groups to differential equations*, volume 107 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1993.
- [17] G. R. W. Quispel and D. I. McLaren. A new class of energy-preserving numerical integration methods. *J. Phys. A*, 41(4):045206, 7, 2008.
- [18] M. Saleem. *Quantum Mechanics*. IOP Publisher, 2015.
- [19] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian problems*, volume 7 of *Applied Mathematics and Mathematical Computation*. Chapman & Hall, London, 1994.
- [20] S. E. Shreve. *Stochastic calculus for finance. II*. Springer Finance. Springer-Verlag, New York, 2004. Continuous-time models.
- [21] S. H. Strogatz. *Nonlinear dynamics and chaos—with applications to physics, biology, chemistry, and engineering*. CRC Press, Boca Raton, FL, third edition, 2024.

A

Numerical implementation

In this appendix, we present the code used to produce the numerical experiments discussed in this thesis. In addition, an important observation regarding the implementation of the AVF method is presented.

A.1 Code used for the pendulum experiments

```
1 %
2 % parameters:
3 k = 0.2; % step-size Explicit Euler, SV
4 N = 500; % Number of discretization points
5 final_time = k * N;
6
7 % initialize arrays:
8 % Explicit Euler
9 pEulerExplicit = zeros(1,N+1);
10 qEulerExplicit = zeros(1,N+1);
11 tEulerExplicit = zeros(1,N+1);
12 energy_Explicit = zeros(1,N+1);
13
14 % Störmer-Verlet
15 qSV = zeros(1,N+1);
16 pSV = zeros(1,N+1);
17 tSV = zeros(1,N+1);
18 energy_SV = zeros(1,N+1);
19
20 % initial values
21 % explicit Euler
22 pEulerExplicit(1) = 1.5;
23 qEulerExplicit(1) = 1;
24 energy_Explicit(1) = 0.5*pEulerExplicit(1)^2-cos(qEulerExplicit(1))
25 ;
26 tEulerExplicit(1) = 0;
27
28 % Störmer-Verlet
29 qSV(1) = 1;
30 pSV(1) = 1.5;
31 energy_SV(1) = 0.5*pSV(1)^2-cos(qSV(1));
32 tSV(1) = 0;
33
34 % Explicit Euler update
35 for n = 1:N
36     tEulerExplicit(n+1) = tEulerExplicit(n) + k;
```

```

36     pEulerExplicit(n+1) = pEulerExplicit(n) - k * sin(
qEulerExplicit(n));
37     qEulerExplicit(n+1) = qEulerExplicit(n) + k * pEulerExplicit(n)
;
38     energy_Explicit(n+1) = 0.5*pEulerExplicit(n+1)^2-cos(
qEulerExplicit(n+1));
39 end
40
41
42 % Störmer-Verlet
43 % one step formulation
44 for n = 1:N
45     tSV(n+1) = tSV(n) + k;
46     pHalfStep = pSV(n) - (k/2) * sin(qSV(n));
47     qSV(n+1) = qSV(n) + k*pHalfStep;
48     pSV(n+1) = pHalfStep - (k/2) * sin(qSV(n+1));
49     energy_SV(n+1) = 0.5*pSV(n+1)^2-cos(qSV(n+1));
50 end
51
52
53 % Plotting the different behaviour:
54 figure
55 plot(tEulerExplicit, energy_Explicit, 'r', 'LineWidth', 1.5)
56 hold on
57 plot(tSV, energy_SV, 'm', 'LineWidth', 1.5)
58
59 xlabel('Time', 'Interpreter','latex','FontSize',10)
60 xlim([0,final_time])
61 ylabel('Total energy  $H(p_n,q_n)$ ', 'Interpreter','latex','FontSize',
10)
62
63 legend('Explicit Euler', 'Störmer-Verlet', 'Interpreter', 'latex
', Location='northwest');
64 title('Evolution of the total energy for the pendulum', '
Interpreter','latex','FontSize',14)
65
66 set(gcf,'Color','w')
67
68 ax = gca;
69 ax.LooseInset = max(ax.TightInset, 0.02);
70
71 set(gcf, 'Units', 'centimeters');
72 set(gcf, 'Position', [2 2 12 8]);
73 exportgraphics(gcf,'Evolution of the total Energy for the Pendulum
using explicit Euler and SV.pdf','ContentType','vector')

```

Listing A.1: Implementation of the explicit Euler scheme (2.8) and the Störmer–Verlet scheme (2.10) for the mathematical pendulum (2.4).

```

1 clear all
2
3 % Implementation of AVF for the Mathematical Pendulum
4 % the update consist of a non-linear system with 2 unknowns
5 % we use fix-point iteration
6
7 % parameters:
8
9 h = 0.2; % time step
10 N = 500; % Numer of discretization points
11 final_time = h*N;
12 K = 150; % fix-point iteration iterations
13 Tol = 1e-8;
14
15 % initialization:
16
17 % arrays
18 q_AVF = zeros(1,N+1);
19 p_AVF = zeros(1,N+1);
20 t_AVF = zeros(1,N+1);
21 energy_AVF = zeros(1,N+1);
22
23 % Störmer-Verlet
24 qSV = zeros(1,N+1);
25 pSV = zeros(1,N+1);
26 tSV = zeros(1,N+1);
27 energy_SV = zeros(1,N+1);
28
29 % values
30 q_AVF(1) = 1;
31 p_AVF(1) = 1.5;
32 t_AVF(1) = 0;
33 energy_AVF(1) = 0.5*p_AVF(1)^2-cos(q_AVF(1));
34
35 % Störmer-Verlet
36 qSV(1) = 1;
37 pSV(1) = 1.5;
38 energy_SV(1) = 0.5*pSV(1)^2-cos(qSV(1));
39 tSV(1) = 0;
40
41 for n = 1:N
42     t_AVF(n+1) = t_AVF(n) + h;
43
44     % initial guess (one step of explicit Euler)
45     p_AVF_fp_old = p_AVF(n) - h * sin(q_AVF(n));
46     q_AVF_fp_old = q_AVF(n) + h * p_AVF(n);
47
48     for k = 1:K
49         % quotient (sum to product identity) yields the following
50         if abs(q_AVF_fp_old - q_AVF(n)) < 1e-12
51             AVF_quotient = sin((q_AVF_fp_old + q_AVF(n))/2);
52         else
53             AVF_quotient = (cos(q_AVF(n))-cos(q_AVF_fp_old))/(
54                 q_AVF_fp_old - q_AVF(n));
55         end

```

```

56     % AVF update
57     p_AVF_fp_new = p_AVF(n) - h*AVF_quotient;
58     q_AVF_fp_new = q_AVF(n) + (h/2) * (p_AVF(n) + p_AVF_fp_new)
    ;
59
60     if norm(p_AVF_fp_new - p_AVF_fp_old) < Tol && norm(
q_AVF_fp_new - q_AVF_fp_old) < Tol
61         fprintf('Fixed_point converged at iteration %d\n',k);
62         p_AVF(n+1) = p_AVF_fp_new;
63         q_AVF(n+1) = q_AVF_fp_new;
64         break
65     else
66         p_AVF_fp_old = p_AVF_fp_new;
67         q_AVF_fp_old = q_AVF_fp_new;
68
69     end
70
71 end
72
73 % Energy update
74 energy_AVF(n+1) = 0.5*p_AVF(n+1)^2-cos(q_AVF(n+1));
75 end
76
77 % Störmer-Verlet
78 % one step formulation
79 for n = 1:N
80     tSV(n+1) = tSV(n) + h;
81     pHalfStep = pSV(n) - (h/2) * sin(qSV(n));
82     qSV(n+1) = qSV(n) + h*pHalfStep;
83     pSV(n+1) = pHalfStep - (h/2) * sin(qSV(n+1));
84     energy_SV(n+1) = 0.5*pSV(n+1)^2-cos(qSV(n+1));
85 end
86
87 % Plotting
88 figure
89 plot(t_AVF, energy_AVF, 'b', 'LineWidth', 1.5)
90 hold on
91 plot(tSV, energy_SV, 'm', 'LineWidth', 1.5)
92
93 xlabel('Time', 'Interpreter','latex','FontSize',10)
94 xlim([0,final_time])
95 ylabel('Total energy $H(p_n,q_n)$', 'Interpreter','latex','FontSize',
    ,10)
96
97 legend('AVF-method', 'Störmer-Verlet scheme', 'Interpreter', '
    latex', Location='northeast');
98 title('Evolution of the total energy for the pendulum', '
    Interpreter','latex','FontSize',14)
99
100 set(gcf,'Color','w')
101
102 ax = gca;
103 ax.LooseInset = max(ax.TightInset, 0.02);
104
105 set(gcf, 'Units', 'centimeters');
106 set(gcf, 'Position', [2 2 12 8]);

```

```
107 exportgraphics(gcf,'Evolution of Energy for the Pendulum using AVF  
and St\''ormer-Verlet.pdf','ContentType','vector')
```

Listing A.2: Implementation of the AVF scheme (2.22) for the mathematical pendulum (2.4).

A.2 Code used for the sine–Gordon experiments

```
1 % Implementation of AVF for the Hamiltonian PDE (using Fixed-point iteration)
2 % sine-Gordon equation:  $u_{tt} = u_{xx} - \sin(u)$  call it (1), periodic BCs are imposed
3 % We rewrite (1) into:
4 %  $u_t = v$ ,
5 %  $v_t = u_{xx} - \sin(u)$ ,
6 % Spatial dsicretization gives the ciculant matrix L (3-point stencil)
7
8 % spatial dsicretization parameters:
9 N_spatial = 200; % number of discretization points
10 L_domain = 20;
11 Delta_x = L_domain/N_spatial; % spatial mesh
12
13 % times discretization parameters:
14 N_time = 300;
15 Delta_t = 0.7; % time step size
16 final_time = Delta_t * N_time;
17
18 % FP method parameters
19 K = 100;
20 Tol = 1e-8;
21
22 % initializations:
23
24 % AVF
25 u_AVF = ones(N_spatial,1);
26 v_AVF = 1.5 * ones(N_spatial,1);
27 energy_AVF = zeros(N_time + 1,1);
28 t_AVF = zeros(N_time + 1,1);
29 t_AVF(1) = 0;
30
31 % Sörmer-Verlet
32 u_SV = ones(N_spatial,1);
33 v_SV = 1.5 * ones(N_spatial,1);
34 energy_SV = zeros(N_time+1,1);
35 t_SV = zeros(N_time + 1,1);
36 t_SV(1) = 0;
37
38 % circulant Matrix L (3-point stencil)
39 L = zeros(N_spatial,N_spatial);
40
41 for index = 1 : N_spatial
42     L(index,index) = -2; % diagonal
43     if index > 1
44         L(index,index-1) = 1; %lower-diagonal
45     end
46     if index < N_spatial
47         L(index,index + 1) = 1; %upper-diagonal
48     end
49 end
50 % adding coeffs due to periodic BCs
51 L(1,N_spatial) = 1;
52 L(N_spatial,1) = 1;
53 % multiplying by  $(1/Delta_x)^2$ 
```

```

54 L = (1/Delta_x)^2 * L;
55
56 % Matrix A used in discrete Hamiltonian
57 A = -L;
58
59 % initial energy:
60 energy_AVF(1) = Delta_x*(0.5*(v_AVF.'*v_AVF) + 0.5*(u_AVF.'*A*u_AVF
    ) - sum(cos(u_AVF)));
61
62 % AVF update
63 for n = 1:N_time
64     t_AVF(n+1) = t_AVF(n) + Delta_t;
65
66     % initial guess: (explicit Euler one-step)
67     u_fp_old = u_AVF + Delta_t * v_AVF;
68     v_fp_old = v_AVF + Delta_t * (L*u_AVF - sin(u_AVF));
69
70     % Start fp iteration
71     for k = 1:K
72         % quotient and matrix-vector term checking if du is very close to zero:
73         du = u_fp_old - u_AVF;
74
75         AVF_quotient = zeros(N_spatial,1);
76
77         small = abs(du) < 1e-12;
78         large = ~small;
79
80         AVF_quotient(small) = sin((u_fp_old(small) + u_AVF(small))/2);
81         AVF_quotient(large) = (cos(u_AVF(large)) - cos(u_fp_old(large))) ./ du(large);
82
83         AVF_matrix_vector = L*((u_AVF + u_fp_old)./2); %Nx1
84
85         % AVF update
86         u_AVF_fp_new = u_AVF + (Delta_t/2) .* (v_fp_old + v_AVF); %
Nx1
87         v_AVF_fp_new = v_AVF + Delta_t .* (AVF_matrix_vector -
AVF_quotient); %Nx1
88
89         if norm(u_AVF_fp_new - u_fp_old) < Tol && norm(v_AVF_fp_new
- v_fp_old) < Tol
90             fprintf('Fixed_point converged at iteration %d\n',k);
91             u_AVF = u_AVF_fp_new;
92             v_AVF = v_AVF_fp_new;
93             break
94         else
95             u_fp_old = u_AVF_fp_new;
96             v_fp_old = v_AVF_fp_new;
97         end
98     end
99
100     % update energy
101     energy_AVF(n+1) = Delta_x*(0.5*(v_AVF.'*v_AVF) + 0.5*(u_AVF.'*A
*u_AVF) - sum(cos(u_AVF)));
102

```

```
103 end
104
105 % Störmer-Verlet update:
106 energy_SV(1) = Delta_x*(0.5*(v_SV.'*v_SV) + 0.5*(u_SV.'*A*u_SV) -
    sum(cos(u_SV)));
107
108 for n = 1:N_time
109     t_SV(n+1) = t_SV(n) + Delta_t;
110
111     v_SV_half_step = v_SV + (Delta_t/2).* (L*u_SV - sin(u_SV));
112     u_SV_step = u_SV + Delta_t.* v_SV_half_step;
113     v_SV_step = v_SV_half_step + (Delta_t/2).* (L*u_SV_step - sin(
        u_SV_step));
114
115     u_SV = u_SV_step;
116     v_SV = v_SV_step;
117
118     energy_SV(n+1) = Delta_x*(0.5*(v_SV.'*v_SV) + 0.5*(u_SV.'*A*
        u_SV) - sum(cos(u_SV)));
119 end
120
121 % Plotting
122 figure
123 plot(t_AVF, energy_AVF, 'b', 'LineWidth', 1.5)
124 hold on
125 plot(t_SV, energy_SV, 'g', 'LineWidth', 1.5)
126
127 xlabel('Time', 'Interpreter','latex','FontSize',10)
128 xlim([0,final_time])
129 ylabel('Total discretized energy  $\mathcal{H}_d(u_n, v_n)$ ',
    'Interpreter','latex','FontSize',10)
130
131 legend('AVF scheme','Störmer-Verlet scheme','Interpreter','
    'latex', Location='northeast');
132 title('Evolution of the total energy for the sine-Gordon equation',
    'Interpreter','latex','FontSize',14)
133
134 set(findall(gcf,'Type','axes'),'LooseInset',max(get(gca,'TightInset')
    ),0.02))
135 set(gcf,'Color','w')
136
137 exportgraphics(gcf,'Evolution of the discretized total energy for
    the sine Gordon equation AVF vs SV.pdf','ContentType','vector');
```

Listing A.3: Implementation of the AVF scheme (3.22) for sine-Gordon equation (3.19)

A.3 Code used for the simulation of Brownian motion path and the Black–Scholes–Merton model

```

1 % Brownian path simulation (vectorized)
2 randn('state',100);
3 T = 1; N = 500; dt = T/N;
4
5 dW = sqrt(dt) * randn(1,N);
6 W = cumsum(dW);
7
8 % plot
9 plot([0:dt:T], [0,W], '-r')
10
11 xlabel('Time t', 'Interpreter','latex','FontSize',10)
12 ylabel('Brownian motion $B(t)$', 'Interpreter','latex','FontSize',10)
13
14 title('Simulation of discretized Brownian motion path', 'Interpreter','latex','FontSize',14)
15
16 set(findall(gcf,'Type','axes'),'LooseInset',max(get(gca,'TightInset'),0.02))
17 set(gcf,'Color','w')
18 exportgraphics(gcf,'Simulation of discretized Brownian motion path.pdf','ContentType','vector')

```

Listing A.4: Simulation of a Brownian motion path based on (4.2).

```

1 % Euler-Maruyama method for linear SDE:
2 %  $dX(t) = r * X(t) dt + \mu * X(t) * dW(t)$ ,  $X(0) = Xzero$ 
3 %  $r = 2$ ,  $\mu = 1$ ,  $Xzero = 1$ ;
4 % we first start by computing the Brownian path over  $[0,1]$  with  $dt = 2^{-8}$ :
5 %  $2^{-8}$ :
6 randn('state',100)
7 r = 2;
8 mu = 1;
9 Xzero = 1;
10 T = 1; N = 2^8; dt = T/N;
11 dW = sqrt(dt)*randn(1,N); % Brownian increment
12 W = cumsum(dW);
13
14 % evaluate the exact solution given by
15 %  $X(t) = X(0)\exp((r - (1/2)\sigma^2)t + \sigma * B(t))$ 
16 X_exact = Xzero*exp((r - (1/2)*mu^2)*(dt:dt:T) + mu*W);
17
18 % Euler-Maruyama update
19 R = 4; Dt = R*dt; L = N/R;
20 X_EM = zeros(1,L);
21 X_temp = Xzero;
22
23 for j = 1:L
24     W_inc = sum(dW(R*(j-1)+1:R*j));
25     X_temp = X_temp + Dt*r*X_temp + mu*X_temp*W_inc;

```

```

26     X_EM(j) = X_temp;
27 end
28
29
30 % plotting
31 plot(0:dt:T, [Xzero,X_exact], 'm-')
32 hold on
33 plot(0:Dt:T, [Xzero,X_EM], 'r--*')
34 hold off
35
36 xlabel('Time t', 'Interpreter','latex','FontSize',10)
37 ylabel('Solution  $X(t)$ ', 'Interpreter','latex','FontSize',10)
38
39 title('Exact solution vs Numerical approximation for Black-scholes
      SDE', 'Interpreter','latex','FontSize',14)
40 legend('Exact solution', 'Euler-Maruyama approximation', '
      Interpreter', 'latex', Location='northwest');
41
42 set(findall(gcf,'Type','axes'),'LooseInset',max(get(gca,'TightInset
      '),0.02))
43 set(gcf,'Color','w')
44 exportgraphics(gcf,'Numerical solution of SDE using Euler Maruyama.
      pdf','ContentType','vector')

```

Listing A.5: Implementation of the Euler–Maruyama scheme (4.15) versus the strong solution (4.18) for the Black–Scholes–Merton SDE (4.17).

A.4 Code used for the stochastic pendulum experiments

```

1 % Numerical scheme based on strang splitting and AVF scheme:
2 % Idea: decompose the problem into two subproblems, one deterministic and
3 % one Stochastic. Use AVF for the deterministic subproblem and
4 % exactly solve the Stochastic subproblem
5 %
6 % Use IC ( $p_0, q_0$ ) to solve the stochastic sub-problem first for step size h/2,
7 % then the obtained solution is used as IC to solve the deterministic
8 % sub-problem using AVF scheme for full step size h.
9 % Finally, the obtained solution is used
10 % as IC to solve the stochastic sub-problem again for step size h/2.
11
12 % parameters:
13 N = 100; % Numer of discretization points
14 T = 20; % final time
15 h = T/N; % step size
16 K = 50; % Number of fix-point iteration
17 Tol = 1e-8; % tolerance used in fix-point iteration
18
19 randn('state',100) % seed the random number generator
20 dt = h/2; % time step for the discretized BM path
21 sigma = 1; % difusion coefficient used in stochastic sub-problem
22 M = 1e6; % number of realizations
23
24 % to accumulate the energy for each sample m
25 accumm_sum_energy_SA = zeros(1,N+1);
26 accumm_sum_energy_EM = zeros(1,N+1);
27
28 % time grid
29 time_grid = linspace(0,T,N+1);
30
31 % Initial Values for Strang-AVF scheme
32 q_0_SA = 1;
33 p_0_SA = 1.5;
34 energy_0_SA = 0.5*p_0_SA^2 - cos(q_0_SA);
35
36 % Initial values for Euler-Maruyama scheme
37 q_0_EM = 1;
38 p_0_EM = 1.5;
39 energy_0_EM = 0.5*p_0_EM^2 - cos(q_0_EM);
40
41 tic
42
43 % simulate Brownian motion (vectorized for efficiency)
44 dB_SA = sqrt(dt) * randn(M,N,2); % BM-increment used in Strang-AVF scheme
45 dB_EM = sqrt(h) * randn(M,N); % BM-increment used in Euler-Maruyama scheme
46
47
48 for m = 1:M
49
50     % Initialization for each realization m:
51

```

```

52 % Strang-AVF scheme
53 q_SA = zeros(1,N+1);
54 p_SA = zeros(1,N+1);
55 energy_SA = zeros(1,N+1);
56
57 q_SA(1) = q_0_SA;
58 p_SA(1) = p_0_SA;
59 energy_SA(1) = energy_0_SA;
60
61 % Euler-Maruyama scheme
62 q_EM = zeros(1,N+1);
63 p_EM = zeros(1,N+1);
64 energy_EM = zeros(1,N+1);
65
66 q_EM(1) = q_0_EM;
67 p_EM(1) = p_0_EM;
68 energy_EM(1) = energy_0_EM;
69
70 for n = 1:N
71
72     % Euler-Maruyama scheme:
73
74     p_EM(n+1) = p_EM(n) - h * sin(q_EM(n)) + sigma * dB_EM(m,n)
;
75     q_EM(n+1) = q_EM(n) + h * p_EM(n);
76
77
78     % Strang-AVF scheme:
79     % first subproblem
80     % Stochastic case solved exactly (step size h/2)
81     % pendulum; solving dp = sigma * dB(t), using exact update
82     % scheme  $p_n = p_{n-1} + \sigma * (B(\tau_{t_j}) - B(\tau_{t_{j-1}}))$ 
83     % q stays the same.
84
85     p_sub1 = p_SA(n) + sigma .* dB_SA(m,n,1);
86     q_sub1 = q_SA(n);
87
88     %second subproblem
89     % Deterministic case AVF using f.p. iteration (full step size h)
90     % IC taken as the obtained solution from the first sub-problem
91
92     p_tmp = p_sub1;
93     q_tmp = q_sub1;
94
95     % initial guess for f.p. (one-step of explicit euler)
96     p_AVF_fp_old = p_tmp - (h) * sin(q_tmp);
97     q_AVF_fp_old = q_tmp + (h) * p_tmp;
98
99     % start of fixed point iteration
100    for k = 1:K
101
102        % quotient (sum to product identity)
103        if abs(q_AVF_fp_old - q_tmp) < 1e-12
104            AVF_quotient = sin((q_AVF_fp_old + q_tmp)/2);
105        else
106            AVF_quotient = (cos(q_tmp) - cos(q_AVF_fp_old))/(

```

```

q_AVF_fp_old - q_tmp);
107     end
108
109     % AVF update
110     p_AVF_fp_new = p_tmp - (h) * AVF_quotient;
111     q_AVF_fp_new = q_tmp + (h/2) * (p_tmp + p_AVF_fp_new);
112
113     % check convergence
114     if norm(p_AVF_fp_new - p_AVF_fp_old) < Tol && norm(
q_AVF_fp_new - q_AVF_fp_old) < Tol
115
116         p_sub2 = p_AVF_fp_new;
117         q_sub2 = q_AVF_fp_new;
118         break
119     else
120         p_AVF_fp_old = p_AVF_fp_new;
121         q_AVF_fp_old = q_AVF_fp_new;
122
123     end
124
125 end
126
127 % third sub-problem
128 % solve the stochastic sub-problem again for step size h/2
129 % we use the obtained solution from the second subproblem as IC
130
131 p_sub3 = p_sub2 + sigma .* dB_SA(m,n,2);
132 q_sub3 = q_sub2;
133
134 % Denote the solution as  $q_{SA}(n+1)$  and  $p_{SA}(n+1)$  to indicate that a
135 % full step of Strang splitting method is done.
136 q_SA(n+1) = q_sub3;
137 p_SA(n+1) = p_sub3;
138
139 % energy for each realization
140 energy_SA(n+1) = 0.5*p_SA(n+1).^2 - cos(q_SA(n+1));
141 energy_EM(n+1) = 0.5*p_EM(n+1).^2 - cos(q_EM(n+1));
142
143 end
144
145 % accumulated sum of energy over the M realizations
146 acumm_sum_energy_SA = acumm_sum_energy_SA + energy_SA;
147 acumm_sum_energy_EM = acumm_sum_energy_EM + energy_EM;
148
149 end
150
151 time = toc;
152 fprintf('Monte Carlo runtime: %.2f seconds\n', time);
153
154 % compute the average of accumulated sum of energy
155 mean_energy_SA = 1/M * acumm_sum_energy_SA;
156 mean_energy_EM = 1/M * acumm_sum_energy_EM;
157
158 % exact expected energy
159 exact_expected_energy = energy_0_SA + 0.5 * sigma^2 * time_grid;
160

```

```
161 % Plotting
162 figure
163 plot(time_grid, mean_energy_SA, 'k--*', 'LineWidth', 1.3, '
    MarkerSize', 5, 'MarkerIndices', 4:10:length(time_grid))
164 hold on
165 plot(time_grid, mean_energy_EM, 'k--o', 'LineWidth', 1.3, '
    MarkerSize', 5, 'MarkerIndices', 7:10:length(time_grid))
166 hold on
167 plot(time_grid, exact_expected_energy, 'k--d', 'LineWidth', 1.3, '
    MarkerSize', 6, 'MarkerIndices', 1:10:length(time_grid))
168
169 xlabel('Time', 'Interpreter', 'latex', 'FontSize', 10)
170 xlim([0, T])
171 ylabel('Expected total energy', 'Interpreter', 'latex', 'FontSize', 10)
172
173 legend('Strang--AVF scheme', 'Euler--Maruyama scheme', 'Exact
    expected total energy', 'Interpreter', 'latex', Location='
    northwest');
174 title('Evolution of the expected total energy for the stochastic
    pendulum', 'Interpreter', 'latex', 'FontSize', 13)
175
176 set(findall(gcf, 'Type', 'axes'), 'LooseInset', max(get(gca, 'TightInset
    '), 0.02))
177 set(gcf, 'Color', 'w')
178 filename = sprintf("Evolution of Expected Energy for the Pendulum
    using Strang AVF and Euler Maruyama for %d realizations.pdf", M);
179 exportgraphics(gcf, filename, 'ContentType', 'vector')
```

Listing A.6: Implementation of the constructed Strang–AVF scheme and the Euler–Maruyama scheme (4.15) for the stochastic pendulum (5.1).

A.5 Code used for the stochastic sine–Gordon experiments

```

1 % Numerical scheme based on strang splitting and AVF:
2 % Idea: decompose the problem into two subproblems, one deterministic and
3 % one Stochastic. Use AVF for the deterministic subproblem and
4 % Solve the stochastic sub-problem exactly
5 %
6 % Use IC  $(v_0, u_0)$  to integrate the stochastic sub-problem for step size  $h/2$ ,
7 % then the obtained solution is used as IC to solve the
8 % parameters:
9 % Spatial discretization
10 N_spatial = 150; % Numer of discretization points
11 L = 10;
12 Delta_x = L/N_spatial; % step size
13
14 % Time discretization
15 N_time = 100;
16 T = 20;
17 h = T/N_time; % step size
18 % time grid
19 time_grid = linspace(0,T,N_time+1);
20
21 % Fixed-point iteration
22 K = 50; % Number of fix-point iteration
23 Tol = 1e-8; % tolerance used in fix-point iteration
24
25 randn('state',100) % seed the random number generator
26 dt = h/2; % time step for the discretized BM path (half step size)
27 M = 1e6; % number of realizations
28
29 % simulate Brownian motion (vectorized for efficiency)
30 dB_SA = sqrt(dt) * randn(M,N_time,2); % BM-increment used in Strang-AVF
31 dB_EM = sqrt(h) * randn(M,N_time); % BM-increment used in Euler-Maruyama
32 sigma = 1; % diffusion coefficient
33 Sigma = sigma * ones(N_spatial,1);
34
35 % circulant Matrix L (3-point stencil)
36 %
37 L = zeros(N_spatial,N_spatial);
38
39 for index = 1 : N_spatial
40     L(index,index) = -2; % diagonal
41     if index > 1
42         L(index,index-1) = 1; %lower-diagonal
43     end
44     if index < N_spatial
45         L(index,index + 1) = 1; %upper-diagonal
46     end
47 end
48 % adding coeffs due to periodic BCs
49 L(1,N_spatial) = 1;
50 L(N_spatial,1) = 1;
51 % multiplying by  $(1/Delta_x)^2$ 

```

```

52 L = (1/Delta_x)^2 * L;
53
54 % Matrix A used in discrete Hamiltonian
55 A = -L;
56
57 % to accumulate the energy for each sample m
58 accum_sum_energy_SA = zeros(1,N_time+1);
59 accum_sum_energy_EM = zeros(1,N_time+1);
60
61 % Initial values for Strang-AVF scheme
62 u_SA = ones(N_spatial,1);
63 v_SA = 1.5 * ones(N_spatial,1);
64 energy_0_SA = Delta_x*(0.5*(v_SA.'*v_SA) + 0.5*(u_SA.'*A*u_SA) -
    sum(cos(u_SA)));
65
66 % Initial values for Euler-Maruyama scheme
67 u_EM = ones(N_spatial,1);
68 v_EM = 1.5 * ones(N_spatial,1);
69 energy_0_EM = Delta_x*(0.5*(v_EM.'*v_EM) + 0.5*(u_EM.'*A*u_EM) -
    sum(cos(u_EM)));
70 tic
71
72 for m = 1:M
73
74     % Initialization for each realization m
75
76     % Strang-AVF scheme
77     u_SA = ones(N_spatial,1);
78     v_SA = 1.5 * ones(N_spatial,1);
79     energy_SA = zeros(1,N_time+1);
80     energy_SA(1) = energy_0_SA;
81
82     % Euler-Maruyama scheme
83     u_EM = ones(N_spatial,1);
84     v_EM = 1.5 * ones(N_spatial,1);
85     energy_EM = zeros(1,N_time+1);
86     energy_EM(1) = energy_0_EM;
87
88     for n = 1:N_time
89
90         % Euler-Maruyama scheme
91
92         u_old_EM = u_EM;
93         v_old_EM = v_EM;
94
95         u_EM = u_old_EM + h * v_old_EM;
96         v_EM = v_old_EM + h * (L*u_old_EM - sin(u_old_EM)) + Sigma
    * dB_EM(m,n);
97
98         % first subproblem
99         % Integrating the SDE using half-step size
100
101         u_sub1 = u_SA;
102         v_sub1 = v_SA + Sigma * dB_SA(m,n,1);
103
104         % second subproblem

```

```

105 % Deterministic case AVF using f.p. iteration (full step size h)
106 % IC taken as the obtained solution from the first sub-problem
107
108 u_tmp = u_sub1;
109 v_tmp = v_sub1;
110
111 % initial guess: (explicit Euler one-step)
112 u_fp_old = u_tmp + h * v_tmp;
113 v_fp_old = v_tmp + h * (L*u_tmp - sin(u_tmp));
114
115 % Start fp iteration
116 for k = 1:K
117     % quotient and matrix-vector term checking if du is very close to
118     % zero:
119     du = u_fp_old - u_tmp;
120
121     AVF_quotient = zeros(N_spatial,1);
122
123     small = abs(du) < 1e-12;
124     large = ~small;
125
126     AVF_quotient(small) = sin((u_fp_old(small) + u_tmp(
small))/2);
127     AVF_quotient(large) = (cos(u_tmp(large)) - cos(u_fp_old
(large))) ./ du(large);
128
129     AVF_matrix_vector = L*((u_tmp + u_fp_old)./2); % Nx1
130
131     % AVF update
132     u_AVF_fp_new = u_tmp + (h/2) .* (v_fp_old + v_tmp); %Nx1
133     v_AVF_fp_new = v_tmp + h .* (AVF_matrix_vector -
AVF_quotient); %Nx1
134
135     if norm(u_AVF_fp_new - u_fp_old) < Tol && norm(
v_AVF_fp_new - v_fp_old) < Tol
136         u_sub2 = u_AVF_fp_new;
137         v_sub2 = v_AVF_fp_new;
138         break
139     else
140         u_fp_old = u_AVF_fp_new;
141         v_fp_old = v_AVF_fp_new;
142     end
143 end
144
145 % third sub-problem
146 % stochastic sub-problem again (step size h/2)
147 % using IC obtained as solution to deterministic subproblem
148
149 v_SA = v_sub2 + Sigma * dB_SA(m,n,2);
150 u_SA = u_sub2;
151
152 % energy for each simulated path
153 energy_SA(n+1) = Delta_x*(0.5*(v_SA.'*v_SA) + 0.5*(u_SA.'*A*
u_SA) - sum(cos(u_SA)));
154 energy_EM(n+1) = Delta_x*(0.5*(v_EM.'*v_EM) + 0.5*(u_EM.'*A*
u_EM) - sum(cos(u_EM)));

```

```
155     end
156
157     % accumulated sum of energy over the smulated M BM paths
158     accumm_sum_energy_SA = accumm_sum_energy_SA + energy_SA;
159     accumm_sum_energy_EM = accumm_sum_energy_EM + energy_EM;
160 end
161
162 time = toc;
163 fprintf('Monte Carlo runtime: %.2f seconds\n', time);
164
165 % compute the average of accumulated sum of energy
166 mean_energy_SA = 1/M * accumm_sum_energy_SA;
167 mean_energy_EM = 1/M * accumm_sum_energy_EM;
168
169 % exact expected energy
170 exact_expected_energy = energy_0_SA + 0.5 * (Delta_x * N_spatial) *
    sigma^2 * time_grid;
171
172 % Plotting
173 figure
174 plot(time_grid, mean_energy_SA, 'k-*', 'LineWidth', 1.3, '
    MarkerSize', 5, 'MarkerIndices', 4:10:length(time_grid))
175 hold on
176 plot(time_grid, mean_energy_EM, 'k-o', 1.3, 'MarkerSize', 5, '
    MarkerIndices', 7:10:length(time_grid))
177 hold on
178 plot(time_grid, exact_expected_energy, 'k-d', 'LineWidth', 1.3, '
    MarkerSize', 6, 'MarkerIndices', 1:10:length(time_grid))
179
180 xlabel('Time', 'Interpreter','latex','FontSize',10)
181 xlim([0,T])
182 ylabel('Expected energy','Interpreter','latex','FontSize',10)
183
184 legend('Strang--AVF scheme', 'Euler--Maruyama scheme', 'Exact
    expected total energy','Interpreter', 'latex', Location='
    northwest');
185 title('Evolution of the expected total energy for the sine Gordon
    equation', 'Interpreter','latex','FontSize',14)
186
187 set(findall(gcf,'Type','axes'),'LooseInset',max(get(gca,'TightInset
    '),0.02))
188 set(gcf,'Color','w')
189 filename = sprintf("Evolution of Expected Energy for the Sine
    Gordon equation using Strang AVF and Euler Maruyama for %d
    realizations.pdf",M);
190 exportgraphics(gcf,filename,'ContentType','vector')
```

Listing A.7: Implementation of the constructed Strang–AVF scheme and the Euler–Maruyama scheme (4.15) for the stochastic sine–Gordon equation (6.1).

A.6 Code used for the time-dependent Schrödinger equation

```

1 % Implementation of AVF for the Hamiltonian PDE (using Fixed-point iteration)
2 % Schrödinger equation:  $iu_t = -(1/2)\nabla^2 u$  call it (1), periodic BCs are imposed
3 % We rewrite (1) into:
4 %  $d/dtu_{real} = -(1/2)\nabla^2 u_{im}$ ,
5 %  $d/dtu_{im} = (1/2)\nabla^2 u_{real}$ ,
6 % Spatial dsicretization gives the ciculant matrix L (3-point stencil)
7
8 % spatial discretization parameters
9 N_spatial = 200;
10 x_left = -20;
11 x_right = 20;
12 L_domain = x_right - x_left;
13
14 Delta_x = L_domain / N_spatial;
15
16 % periodic grid: do not include the right endpoint
17 x = x_left + (0:N_spatial-1).' * Delta_x;
18
19 % time discretization parameters
20 N_time = 300;
21 Delta_t = 0.7;
22 final_time = Delta_t * N_time;
23
24 % FP method parameters
25 K = 100;
26 Tol = 1e-8;
27
28 % initializations
29
30 % AVF
31 u_real_AVF = exp(-((x - 1).^2)/2);
32 u_im_AVF = exp(-(x.^2)/2);
33
34 L2_norm_squared_AVF = zeros(N_time + 1, 1);
35 t_AVF = zeros(N_time + 1, 1);
36 t_AVF(1) = 0;
37
38 % Störmer-Verlet
39 u_real_SV = exp(-((x - 1).^2)/2);
40 u_im_SV = exp(-(x.^2)/2);
41
42 L2_norm_squared_SV = zeros(N_time+1,1);
43 t_SV = zeros(N_time + 1,1);
44 t_SV(1) = 0;
45
46 % circulant Matrix L (3-point stencil)
47 L = zeros(N_spatial,N_spatial);
48
49 for index = 1 : N_spatial
50     L(index,index) = -2; % diagonal
51     if index > 1

```

```

52     L(index,index-1) = 1; %lower-diagonal
53     end
54     if index < N_spatial
55         L(index,index + 1) = 1; %upper-diagonal
56     end
57 end
58 % adding coeffs due to periodic BCs
59 L(1,N_spatial) = 1;
60 L(N_spatial,1) = 1;
61 % multiplying by (1/Delta_x)^2
62 L = (1/Delta_x)^2 * L;
63
64 % initial square L2-norm:
65 L2_norm_squared_AVF(1) = Delta_x*((u_real_AVF.' * u_real_AVF) + (
    u_im_AVF.' * u_im_AVF));
66
67 % AVF update
68 for n = 1:N_time
69     t_AVF(n+1) = t_AVF(n) + Delta_t;
70
71     % initial guess: (explicit Euler one-step)
72     u_real_fp_old = u_real_AVF - (1/2) * Delta_t * L * u_im_AVF;
73     u_im_fp_old = u_im_AVF + (1/2) * Delta_t * L * u_real_AVF;
74
75     % Start fp iteration
76     for k = 1:K
77
78         AVF_matrix_vector_u_im = L*((u_im_AVF + u_im_fp_old)./2);
79         AVF_matrix_vector_u_real = L*((u_real_AVF + u_real_fp_old)
    ./2);
80
81         % AVF update
82         u_real_AVF_fp_new = u_real_AVF - (Delta_t/2) .*
    AVF_matrix_vector_u_im; %Nx1
83         u_im_AVF_fp_new = u_im_AVF + (Delta_t/2) .*
    AVF_matrix_vector_u_real; %Nx1
84
85         if norm(u_real_AVF_fp_new - u_real_fp_old) < Tol && norm(
    u_im_AVF_fp_new - u_im_fp_old) < Tol
86             fprintf('Fixed_point converged at iteration %d\n',k
    );
87             u_real_AVF = u_real_AVF_fp_new;
88             u_im_AVF = u_im_AVF_fp_new;
89             break
90         else
91             u_real_fp_old = u_real_AVF_fp_new;
92             u_im_fp_old = u_im_AVF_fp_new;
93         end
94     end
95
96     % Update square L2-norm
97     L2_norm_squared_AVF(n+1) = Delta_x*((u_real_AVF.' * u_real_AVF)
    + (u_im_AVF.' * u_im_AVF));
98 end
99
100 % Plotting

```

```

101 figure
102 plot(t_AVF, L2_norm_squared_AVF, 'k-*', 'LineWidth', 1.5)
103
104 xlabel('Time', 'Interpreter', 'latex', 'FontSize', 10)
105 xlim([0, final_time])
106 ylabel('Discretized  $L^2$  norm Squared  $||u_n||^2_{L^2}$ ',
107         'Interpreter', 'latex', 'FontSize', 10)
108 legend('AVF scheme', 'Interpreter', 'latex', Location='northeast');
109 title('Evolution of the  $L^2$  norm squared for the Schrödinger
110       equation', 'Interpreter', 'latex', 'FontSize', 14)
111 set(findall(gcf, 'Type', 'axes'), 'LooseInset', max(get(gca, 'TightInset')
112             , 0.02))
112 set(gcf, 'Color', 'w')
113
114 exportgraphics(gcf, 'Evolution of the discretized L2 norm squared
115               using AVF.pdf', 'ContentType', 'vector');

```

Listing A.8: Implementation of the AVF scheme (2.12) for the linear Schrödinger equation (3.29).

```

1 % Numerical scheme based on strang splitting:
2 % Idea: decompose the problem into two subproblems, one deterministic and
3 % one Stochastic. Use AVF for the deterministic subproblem and
4 % Solve the stochastic sub-problem exactly
5 %
6 % Use IC ( $u_{real}^0, u_{im}^0$ ) to integrate the stochastic sub-problem for step size h/2,
7 % then the obtained solution is used as IC to solve the deterministic
8 % sub-problem using AVF scheme
9 % Finally, the obtained solution is used
10 % as IC to solve the stochastic sub-problem again for step size h/2.
11
12 % parameters:
13 % spatial discretization parameters
14 N_spatial = 200;
15 x_left = -20;
16 x_right = 20;
17 L_domain = x_right - x_left;
18
19 Delta_x = L_domain / N_spatial;
20
21 % periodic grid: do not include the right endpoint
22 x = x_left + (0:N_spatial-1).' * Delta_x;
23
24 % Time discretization
25 N_time = 300;
26 T = 2;
27 h = T/N_time; % step size
28 % time grid
29 time_grid = linspace(0,T,N_time+1);
30
31 % Fixed-point iteration
32 K = 50; % Number of fix-point iteration
33 Tol = 1e-8; % tolerance used in fix-point iteration
34
35 randn('state',100) % seed the random number generator
36 dt = h/2; % time step for the discretized BM path (half step size)
37 M = 1e5; % number of realizations
38
39 % simulate Brownian motion (vectorized for efficiency)
40 dB = sqrt(dt) * randn(M,N_time,2); % increment
41 sigma = 1; % diffusion coefficient
42 Sigma = sigma * ones(N_spatial,1);
43
44 % circulant Matrix L (3-point stencil)
45 L = zeros(N_spatial,N_spatial);
46
47 for index = 1 : N_spatial
48     L(index,index) = -2; % diagonal
49     if index > 1
50         L(index,index-1) = 1; %lower-diagonal
51     end
52     if index < N_spatial
53         L(index,index + 1) = 1; %upper-diagonal
54     end
55 end
56 % adding coeffs due to periodic BCs

```



```
108         if norm(u_real_AVF_fp_new - u_real_fp_old) < Tol &&
norm(u_im_AVF_fp_new - u_im_fp_old) < Tol
109             u_real_sub2 = u_real_AVF_fp_new;
110             u_im_sub2 = u_im_AVF_fp_new;
111             break
112         else
113             u_real_fp_old = u_real_AVF_fp_new;
114             u_im_fp_old = u_im_AVF_fp_new;
115         end
116     end
117
118     % third sub-problem
119     % stochastic sub-problem again (step size h/2)
120     % using IC obtained as solution to deterministic subproblem
121
122     u_im_SS = u_im_sub2 + Sigma * dB(m,n,2);
123     u_real_SS = u_real_sub2;
124
125     % energy for each simulated path
126     L2_norm_squared_SS(n+1) = Delta_x*((u_real_SS.'*u_real_SS) + (
u_im_SS.'*u_im_SS));
127
128     end
129
130     % accumulated sum of energy over the smulated M BM paths
131     acumm_sum_L2_norm_squared = acumm_sum_L2_norm_squared +
L2_norm_squared_SS;
132
133 end
134
135 time = toc;
136 fprintf('runtime: %.2f seconds\n', time);
137
138 % compute the average of accumulated sum of energy
139 mean_L2_norm_squared = 1/M * acumm_sum_L2_norm_squared;
140
141 % exact expected energy
142 exact_expected_L2_norm_squared = L2_norm_squared_0 + 2*(20) * sigma
^2 * time_grid;
143
144 % Plotting
145 figure
146 plot(time_grid, mean_L2_norm_squared, 'k-*', 'LineWidth', 1.3, '
MarkerSize', 5, 'MarkerIndices', 11:20:length(time_grid))
147 hold on
148 plot(time_grid, exact_expected_L2_norm_squared, 'k--d', 'LineWidth',
1.3, 'MarkerSize', 6, 'MarkerIndices', 1:20:length(time_grid))
149
150 xlabel('Time', 'Interpreter','latex','FontSize',10)
151 xlim([0,T])
152 ylabel('Expected value of L2 norm squared','Interpreter','latex','
FontSize',10)
153
154 legend('Strang--AVF scheme', 'Exact expected L2 norm','Interpreter'
, 'latex', Location='northwest');
155 title('Evolution of the expected value for the L2 norm squared for
```

```
156     the stochastic Schr\{"{o}dinger equation', 'Interpreter','latex',  
157     'FontSize',13)  
157 set(findall(gcf,'Type','axes'),'LooseInset',max(get(gca,'TightInset'  
    '),0.02))  
158 set(gcf,'Color','w')  
159 filename = sprintf("Evolution of Expected value of L2 norm squared  
    for %d realizations.pdf",M);  
160 exportgraphics(gcf,filename,'ContentType','vector')
```

Listing A.9: Implementation of the constructed Strang–AVF scheme for the stochastic Schrödinger equation (6.23).

A.7 The AVF quotient

The AVF quotient in the AVF update (2.22a) derived in section 2.3 needs to be handled carefully, specially when the dominator is very close to zero which may lead to numerical stability issues. Therefore, we use the following sum to product identity as suggested in [5]

$$\cos(a) - \cos(b) = -2 \sin\left(\frac{a+b}{2}\right) \sin\left(\frac{a-b}{2}\right).$$

The AVF quotient then reads

$$\frac{\cos(a) - \cos(b)}{b - a} = \frac{-2 \sin\left(\frac{a+b}{2}\right) \sin\left(\frac{a-b}{2}\right)}{b - a}.$$

Writing $\sin(\frac{a-b}{2})$ as $-\sin(\frac{b-a}{2})$, the quotient becomes

$$\frac{2 \sin\left(\frac{a+b}{2}\right) \sin\left(\frac{b-a}{2}\right)}{b - a},$$

which can be written as

$$\sin\left(\frac{a+b}{2}\right) \cdot \frac{2 \sin\left(\frac{b-a}{2}\right)}{b - a}.$$

Substituting $z = \frac{b-a}{2}$ and using the standard limit

$$\lim_{z \rightarrow 0} \frac{\sin(z)}{z} = 1,$$

gives that the quotient can be written as

$$\frac{\cos(a) - \cos(b)}{b - a} = \sin\left(\frac{a+b}{2}\right),$$

when the difference $b - a$ is very close to zero.

DEPARTMENT OF Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY