Randomized Numerical Schemes: Generalization and Applications

Xinheng Xie

Department of Mathematics and Statistics

University of Strathclyde, Glasgow

June 26, 2026

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Abstract

This thesis focuses on the theoretical analysis, numerical implementation, and practical applications of randomized numerical schemes for solving differential equations characterized by low regularity and temporal irregularities. Randomized schemes, which introduce stochastic perturbations into traditional deterministic numerical methods, can offer advantages in convergence analysis and error estimation in certain low-regularity settings, especially when dealing with non-smooth coefficient functions.

The research is structured into three primary components. First, we study delay differential equations (DDEs) with Carathéodory-type coefficients that are measurable in time and continuous in the state variables, and establish existence and uniqueness under appropriate assumptions. We then propose a novel randomized two-stage Runge-Kutta scheme and establish convergence rates under strengthened low-regularity assumptions—Lipschitz in the state and Hölder in delay and time—extending the class of DDEs tractable beyond standard deterministic methods. Numerical experiments validate these theoretical results and demonstrate the scheme's effectiveness.

Second, the thesis explores stochastic delay differential equations (SDDEs) with drift coefficients satisfying Carathéodory-type conditions. By constructing and analysing a randomized Euler scheme tailored for these equations, we derive rigorous error bounds and confirm convergence properties theoretically. Implementation details and extensive numerical experiments illustrate the practical performance and robustness of the proposed scheme.

Lastly, the thesis introduces a randomized numerical scheme within neural ordinary differential equation (NODE) frameworks for image denoising applications. A new deep learning model, NODE-ImgNet, is developed by embedding a randomized Euler solver

within a NODE architecture. Evaluated on multiple image denoising benchmarks, the proposed model demonstrates competitive and, in many cases, superior empirical performance relative to several strong baseline methods.

Overall, this thesis establishes randomized numerical methods as powerful tools for handling differential equations and data-driven models with limited regularity. The work not only provides rigorous theoretical insights but also broadens the practical applicability of numerical methods, opening pathways for future research into higher-order randomized schemes and deeper connections with machine learning techniques.

Contents

List of Figures

List of Figures

List of Tables

List of Tables

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Yue Wu, for her invaluable guidance, support, and patience throughout my doctoral studies. Her rigorous academic standards, generous mentorship, and open-minded attitude toward life have profoundly influenced both my research and personal growth.

I am also sincerely grateful to my examiners, Dr Mateusz Majka and Dr Mohammud Foondun, for their careful reading of the thesis, constructive comments, and valuable suggestions during the viva and correction process. I would also like to thank Professor Chris Robertson for organising and coordinating the viva process.

I am also grateful to all my collaborators: Cuiyu, Hao, Fabio, Paweł, Kureha, Margaux, Simon, Varun and Victoria. Thank you for your insights, ideas, and the many fruitful discussions—it has been a privilege to work with each of you.

Special thanks to Henglei Xu for taking the time to proofread this thesis. Your feedback and effort are deeply appreciated.

I would also like to thank my family—my parents, Fangming Xie and Mei Zhou—for their unconditional love, understanding, and unwavering support. Your belief in me has always been my greatest source of strength and motivation.

To my partner, Xingyu Zhao—thank you for walking this journey with me, with love, patience, and constant encouragement.

Finally, I am sincerely grateful to everyone who has supported me in any way throughout this PhD journey. Thank you all.

Chapter 0. Acknowledgements

Chapter 1

Introduction

Numerical schemes are fundamental tools in modern mathematical research, offering effective solutions to a wide range of complex problems through discretization and computer simulation. In the context of differential equations, numerical schemes are indispensable, as most such equations do not admit explicit analytical solutions or are computationally intractable. Leveraging high-performance computing and discretization techniques, researchers can obtain numerical approximations that meet practical accuracy requirements. Traditional *deterministic numerical schemes* for ordinary differential equations (ODEs) and stochastic differential equations (SDEs), such as Euler schemes [13], Runge-Kutta schemes [30], the Euler-Maruyama scheme [59], and their implicit variants [2,40,106], achieve this by discretizing the temporal domain into finite steps and propagating solutions iteratively. These schemes have been the cornerstone of dynamic system analysis, enabling accurate trajectory predictions in classical mechanics [41], financial modelling [59], and biological systems [78], where smooth coefficients and well-defined initial conditions prevail.

However, many real-world models involve low-regularity or non-smooth coefficient functions, posing serious challenges to traditional deterministic schemes. In recent years, a new class of *randomised schemes* has emerged to address such difficulties [4,7,61,64,77,87,89]. These methods introduce controlled random perturbations—such as random evaluation points or discretization nodes—within classical discretization frameworks. In certain low-regularity settings, this randomisation can help avoid patho-

logical alignment with irregular features of the problem and may remain effective where some deterministic point-evaluation-based methods fail to converge. Moreover, such schemes often admit probabilistic error estimates, which enable rigorous convergence analysis under weaker regularity assumptions. This thesis studies the development of randomised numerical schemes for low-regularity delay differential equations and stochastic delay differential equations, and further explores their extension to a neural ODE framework for image denoising.

1.1 Background and Literature Review

We begin by reviewing the main ideas behind randomised schemes and the existing literature on their convergence advantages in low-regularity settings. To this end, we first consider the prototypical framework of ordinary differential equations (ODEs). Consider the initial value problem for an ODE: for a given $T \in (0, +\infty)$, find a function $u : [0, T] \to \mathbb{R}^d$ $(d \in \mathbb{N})$ such that

$$\begin{cases} u'(t) = f(t, u(t)), \quad t \in [0, T] \\ u(0) = u_0 \end{cases} \tag{1.1}$$

where $u_0 \in \mathbb{R}^d$ is the initial condition.

Traditional deterministic numerical schemes typically discretize the ODE using finitely many point evaluations to produce approximate solutions. For instance, the classical explicit Euler scheme updates the solution via

$$U^j = U^{j-1} + hf(t_{j-1}, U^{j-1}), \quad j = 1, \dots, N_h,$$

where $h > 0$ denotes a fixed step size, $t_j = jh$, $N_h = \lfloor T/h \rfloor$, and $U^0 = u_0$.

As a probabilistic extension, the *randomised Euler scheme* [64] incorporates randomness while maintaining the original structure. Let $(\gamma_j)_{j \geq 1}$ be a sequence of i.i.d. $\mathcal{U}(0, 1)$ random variables (uniformly distributed on $(0, 1)$). The numerical solution is

then iteratively generated as

$$U^j = U^{j-1} + h\, f\left(t_{j-1} + \gamma_j h,\ U^{j-1}\right), \quad j = 1, \ldots, N_h.$$

This approach embeds random perturbations into the classical Euler scheme and can be interpreted as a special case of Runge-Kutta-type Monte Carlo schemes (see [53, 99, 100]). The resulting sequence (U^j) forms a discrete-time stochastic process on the underlying probability space.

In low-regularity settings, randomised schemes can exhibit advantages in convergence and stability. For initial value problems (1.1) with low-regularity coefficient functions f, particularly Carathéodory-type problems where coefficients are only time-integrable and Lipschitz continuous in the state (as in autonomous SDEs with additive noise or rough differential equations), deterministic single-step schemes based on fixed point evaluations may fail to converge, or may require stronger regularity assumptions [64, 77]. In contrast, Monte Carlo-based randomised algorithms can converge under mere integrability assumptions. If f possesses slightly stronger temporal regularity—specifically, γ-continuity in time for $\gamma \in (0, 1]$—classical Runge-Kutta or linear multistep schemes do converge, but the best possible error rate is $\mathcal{O}(N^{-\gamma})$ due to the lack of differentiability in f [48]. However, the randomised algorithms proposed in [24, 48, 53, 55, 99] achieve improved error rates in the L^p-sense, specifically of order $\mathcal{O}(N^{-\gamma-\frac{1}{2}})$, under the same regularity conditions. Note that almost-sure convergence is attained for sufficiently small step sizes, with a convergence order of $\mathcal{O}(N^{-\gamma-\frac{1}{2}+\epsilon})$ for any small $\epsilon > 0$ [60].

The applicability of randomised schemes for ODEs has also been further extended. For example, [25, 45] applied them to initial value problems in Banach spaces; [89] studied randomised Euler–Maruyama schemes for stochastic ODEs; and [19] considered related quasi-random methods. More recently, attention has focused on ODEs with time-irregular vector fields (see, e.g., [8, 9, 27, 47, 52, 64]). These works provide a foundation for the analysis of randomised numerical schemes for differential equations with low temporal regularity.

Building on the insights from randomised numerical schemes for ODEs, this method-

ology has been naturally extended to SDEs, exploiting similar probabilistic structures to handle the complexities introduced by stochastic perturbations. These advances have been particularly important in addressing the challenges arising in engineering, physical modelling, and computational finance, where classical schemes (e.g., [58, 72, 73]) often become inadequate. A fundamental challenge in SDEs stems from the non-smooth nature of solutions, inherently caused by stochastic perturbations from noise terms. For instance, the celebrated Milstein scheme [70, 71] can achieve first-order accuracy via Itô-Taylor expansion when coefficient functions are sufficiently smooth. However, standard Euler-Maruyama and Milstein schemes may diverge in both strong and weak convergence senses when handling coefficients with super-linear growth or lacking basic differentiability [49].

To overcome this theoretical barrier, recent advances have introduced Monte Carlo-inspired randomised algorithms (e.g., [23, 46, 62, 74, 75, 88]). These approaches are well suited to temporally irregular coefficients. Taking non-autonomous SDEs as an example, whereas the classical Milstein scheme requires differentiability in both temporal and spatial variables for first-order convergence, its randomised counterpart [62] attains the same accuracy under spatial Lipschitz continuity and temporal Hölder continuity. This improvement is mainly due to the randomised quadrature strategy adapted for Hölder continuous processes, which reduces the need for the stronger smoothness assumptions used in traditional Itô-Taylor expansions. Building on this theoretical foundation, randomised Euler-Milstein schemes have been developed for SDE models including additive noise models [4], multiplicative noise systems [61], jump-diffusion SDEs [87] and McKean–Vlasov-type equations [7].

1.2 Overview of the Main Results

Despite this progress for ODEs and SDEs, important gaps remain. In particular, delay equations introduce memory effects that substantially complicate both well-posedness analysis and error propagation, and the available theory for low-regularity DDEs and SDDEs remains much more limited than in the non-delay case. Moreover, although neural ODEs have become an important modelling framework in machine learning,

the use of randomised numerical solvers in this setting remains largely unexplored. In this thesis, we address these gaps by developing and analysing randomised numerical schemes for delay differential equations (DDEs) and stochastic delay differential equations (SDDEs), and by extending the use of randomised solvers to neural ordinary differential equation (NODE) models.

Existence and uniqueness of delay equations have been studied in several classical and modern settings. For deterministic delay differential equations, standard well-posedness theory under regularity and growth assumptions can be found, for example, in [42, 43], while Carathéodory-type delay equations with low regularity in the time variable are treated in [27, 28]. For stochastic delay differential equations, classical existence and uniqueness results under Lipschitz and growth assumptions are available in, for example, [68], with related stability questions studied in [69]. The added perspective in this thesis is to combine these well-posedness ideas with the low-regularity randomised numerical framework considered here, and to record the quantitative moment, Hölder-regularity, and error estimates needed for the subsequent analysis of the proposed randomised DDE and SDDE schemes.

The first core part of this thesis (Chapter 3) builds on the preprint [29] and presents a refined version of that work, focusing on the existence, uniqueness, and numerical approximation of solutions to DDEs of the form

$$\begin{cases} x'(t) = f(t, x(t), x(t-\tau)), & t \in [0, (n+1)\tau], \\ x(t) = \varphi(t), & t \in [-\tau, 0], \end{cases}$$

with fixed delay $\tau > 0$. Existence and uniqueness are established in a Carathéodory-type setting, where $t \mapsto f(t, x, z)$ is measurable for fixed (x, z) and $(x, z) \mapsto f(t, x, z)$ is continuous for fixed t. For the error analysis of the randomised two-stage Runge–Kutta scheme, we further assume global Lipschitz continuity in x and Hölder continuity in z and t. Under these conditions we derive $L^p(\Omega)$-error bounds, validated by numerical experiments.

The second core part of this thesis (Chapter 4) builds on the peer-reviewed work [90] and studies SDDEs with Carathéodory-type drift coefficients. In particular, we address

the existence, uniqueness, and approximation of solutions to equations of the form

$$\begin{cases} \mathrm{d}X(t) = f(t, X(t), X(t-\tau))\,\mathrm{d}t + g(t, X(t), X(t-\tau))\,\mathrm{d}W(t), & t \in [0, (n+1)\tau], \\ X(t) = x_0, & t \in [-\tau, 0], \end{cases}$$

where constant time-lag $\tau > 0$, fixed time horizon $n \in \mathbb{N}$, f is Borel measurable in t and continuous in (x, z), and g is continuous in all its arguments. We develop a randomised Euler scheme for these SDDEs, derive upper error bounds for the approximations, and validate our theoretical findings through numerical experiments.

The third core part of this thesis (Chapter 5) builds on the peer-reviewed paper [107] and explores the application of randomised Euler schemes within the NODE framework for image denoising. Based on this approach, we propose NODE-ImgNet, a novel neural network architecture that integrates neural ODEs with convolutional neural network (CNN) blocks. By leveraging the continuous-time formulation of NODEs, the model learns the denoising process directly from data and provides a flexible alternative to traditional hand-crafted dynamical models. Another important contribution is our integration of a randomised Euler scheme into the NODE solver, which is motivated by challenges posed by irregular or low-regularity dynamics and input data, while also introducing beneficial stochastic perturbations that may enhance model robustness and generalization. In the NODE setting considered here, the randomised Euler scheme is adopted primarily as a computational design choice, and its effectiveness is assessed empirically. More detailed background and literature review for each main topic are provided in the respective chapters.

1.3 Structure of the Thesis

The structure of this thesis is as follows. Chapter 2 presents the necessary preliminary knowledge and mathematical background. Chapters 3 to 5 contain the main research contributions of the thesis: Chapter 3 studies DDEs and develops a randomised Runge–Kutta scheme, Chapter 4 investigates SDDEs and develops a randomised Euler scheme, and Chapter 5 explores the use of randomised Euler solvers in NODEs for image de-

noising through the NODE-ImgNet model. Finally, Chapter 6 summarises the main findings, discusses limitations, and outlines possible directions for future work.

Chapter 2

Preliminaries

In this chapter, the main definitions and auxiliary results (lemmas and theorems) required throughout this thesis are collected. Notation and definitions specific to individual topics or applications will be introduced in the relevant chapters as needed. Since the introduction and error analysis of randomised schemes form a central theme of this work, some fundamental concepts from probability theory and stochastic analysis are also reviewed here. For a broader introduction to these topics, the reader is referred to standard texts such as [56, 57, 80], while measure-theoretic background can be found in [5, 18].

2.1 Basic Notation and Concepts for Stochastic Analysis

As usual, we denote by $\mathbb{N}$ the set of all positive integers, and set $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For an integer k, we write $[k] := \{1, \ldots, k\}$ and $[k]_0 := \{0\} \cup [k]$. The set of real numbers is denoted by $\mathbb{R}$. The notation $|\cdot|$ refers to the Euclidean norm on $\mathbb{R}^d$, or to the Frobenius norm on $\mathbb{R}^{d\times m}$ when appropriate. Recall that a *complete probability space* is a triple $(\Omega, \Sigma, \mathbb{P})$, where Ω is the sample space, Σ is a σ-algebra of subsets of Ω that contains all subsets of $\mathbb{P}$-null sets, and $\mathbb{P}$ is a probability measure defined on Σ with $\mathbb{P}(\Omega) = 1$. For any event $A \in \Sigma$, the value $\mathbb{P}(A) \in [0, 1]$ represents the probability of occurrence of A.

A function $X : \Omega \to \mathbb{R}^d$ is called a *random variable* if it is measurable with respect

to the σ-algebra Σ on Ω and the Borel σ-algebra $\mathcal{B}(\mathbb{R}^d)$ on $\mathbb{R}^d$. In other words, for any Borel set $B \in \mathcal{B}(\mathbb{R}^d)$, the preimage

$$X^{-1}(B) = \{\omega \in \Omega : X(\omega) \in B\}$$

belongs to Σ. Every random variable X induces a probability measure μ_X on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, where $\mu_X(B) = \mathbb{P}(X^{-1}(B))$ for all $B \in \mathcal{B}(\mathbb{R}^d)$. This measure is called the *distribution* (or law) of X.

For the randomised schemes in this thesis, a critical tool is the family of $\mathcal{U}(a,b)$-distributed random variables $(\gamma_k)_{k\in\mathbb{N}}$. Specifically, for each $k \in \mathbb{N}$, the mapping $\gamma_k : \Omega \to \mathbb{R}$ is a random variable that is *uniformly distributed* on the interval (a,b), where $a, b \in \mathbb{R}$ with $a < b$. The distribution μ_{γ_k} of γ_k is then given by

$$\mu_{\gamma_k}(A) = \frac{1}{b-a}\,\lambda(A \cap (a,b))$$

for all $A \in \mathcal{B}(\mathbb{R})$, where λ denotes the Lebesgue measure on $\mathbb{R}$. Furthermore, in the context of DDEs or SDDEs, we will also work with families of independent random variables $(\gamma_k^j)_{j\in\mathbb{N}_0,\, k\in\mathbb{N}}$, where the index j indicates the stage or segment in the delay process (i.e., corresponding to different intervals after successive delays). Each γ_k^j is also uniformly distributed on (a,b).

Recall that a random variable $X : \Omega \to \mathbb{R}^d$ is called *integrable* if $\int_\Omega |X(\omega)|\,\mathrm{d}\mathbb{P}(\omega) < \infty$. The *expectation* (or mean) of X is defined as

$$\mathbb{E}[X] := \int_\Omega X(\omega)\,\mathrm{d}\mathbb{P}(\omega) = \int_{\mathbb{R}^d} x\,\mathrm{d}\mu_X(x).$$

For $p \in [1,\infty)$, we write $X \in L^p(\Omega;\mathbb{R}^d)$ if $\int_\Omega |X(\omega)|^p\,\mathrm{d}\mathbb{P}(\omega) < \infty$, and define the L^p-norm by

$$\|X\|_{L^p(\Omega;\mathbb{R}^d)} := (\mathbb{E}[|X|^p])^{1/p} = \left(\int_\Omega |X(\omega)|^p\,\mathrm{d}\mathbb{P}(\omega)\right)^{1/p}.$$

The space $L^p(\Omega;\mathbb{R}^d)$ is a Banach space when random variables that differ only on sets of probability zero are identified. For notational simplicity, we may also write $\|X\|_{L^p(\Omega)}$

or $\|X\|_p$ when the underlying space is clear from context. Further notation and conventions will be introduced as needed in the subsequent chapters. This construction is analogous to the classical L^p-spaces of p-integrable functions, such as $L^p([0,T];\mathbb{R}^d)$, consisting of measurable functions $f : [0,T] \to \mathbb{R}^d$ satisfying

$$\int_0^T |f(t)|^p \, \mathrm{d}t < \infty,$$

and equipped with the norm

$$\|f\|_{L^p([0,T];\mathbb{R}^d)} := \left(\int_0^T |f(t)|^p \, \mathrm{d}t \right)^{1/p}.$$

If we view the index m as representing time, then a collection of $\mathbb{R}^d$-valued random variables $(X_m)_{m\in\mathbb{N}}$ is called a discrete-time *stochastic process*. A particularly important class of such processes is that of *martingales*. Formally, let $(\mathcal{F}_n)_{n\in\mathbb{N}_0}$ be a *filtration*, that is, an increasing sequence of σ-algebras,

$$\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F},$$

where $\mathcal{F} = \sigma\left(\bigcup_{n\in\mathbb{N}_0} \mathcal{F}_n\right)$ is the σ-algebra generated by the union of all $\mathcal{F}_n$ (i.e., the smallest σ-algebra containing all sets in $\bigcup_n \mathcal{F}_n$). For a continuous-time filtration $(\mathcal{F}_t)_{t\geq 0}$, right-continuity means that $\mathcal{F}_t = \bigcap_{s>t} \mathcal{F}_s$ for every $t \geq 0$, while completeness means that all $\mathbb{P}$-null sets in $\mathcal{F}$ belong to $\mathcal{F}_0$ and hence to every $\mathcal{F}_t$. When we say that a filtration satisfies the *usual conditions*, we mean that it is both right-continuous and complete. For the discrete-time filtrations used below, the relevant condition is completeness; right-continuity is only relevant in the continuous-time setting. A stochastic process $(X_n)_{n\in\mathbb{N}_0}$ is called a martingale (with respect to the filtration $(\mathcal{F}_n)$) if it is integrable, adapted to the filtration (i.e., X_n is $\mathcal{F}_n$-measurable for each n), and satisfies

$$\mathbb{E}[X_{n+1} \mid \mathcal{F}_n] = X_n, \quad \text{for all } n \in \mathbb{N}_0.$$

For our purposes, it is particularly relevant to note that if $(Y_m)_{m\in\mathbb{N}}$ is a sequence

of independent, integrable random variables with $\mathbb{E}[Y_m] = 0$ for all $m \in \mathbb{N}$, then the sequence of partial sums

$$S_n := \sum_{m=1}^{n} Y_m, \quad n \in \mathbb{N},$$

defines a discrete-time martingale with respect to the natural filtration $\mathcal{F}_n := \sigma(Y_1, \ldots, Y_n)$. This property enables us to utilize powerful inequalities for martingales, most notably the discrete-time Burkholder–Davis–Gundy (BDG) inequality (see, for example, [12, Theorem 3.2]), which plays a key role in our analysis.

Theorem 2.1.1 (BDG inequality)**.** *For each $p \in (1, \infty)$, there exist positive constants c_p and C_p such that for any discrete-time martingale $(X_n)_{n\in\mathbb{N}}$ and any $n \in \mathbb{N}$,*

$$c_p \|[X]_n^{1/2}\|_{L^p(\Omega;\mathbb{R})} \leq \left\| \max_{1\leq j\leq n} |X_j| \right\|_{L^p(\Omega;\mathbb{R})} \leq C_p \|[X]_n^{1/2}\|_{L^p(\Omega;\mathbb{R})},$$

where the quadratic variation $[X]_n$ *is defined as* $[X]_n := |X_1|^2 + \sum_{k=2}^{n} |X_k - X_{k-1}|^2$.

2.2 Error Estimates for randomised Quadrature Schemes

In this section, we introduce the error and convergence properties of randomised quadrature schemes for numerical integration. The classical deterministic quadrature rules typically rely on certain smoothness conditions of the integrand to achieve satisfactory accuracy. When the integrand is only of low regularity or even discontinuous, deterministic numerical schemes based on such rules may fail to converge or only provide suboptimal error rates, as discussed in the introduction. In contrast, randomised quadrature schemes can provide convergence and improved error bounds for the classes of low-regularity integrands considered below. The inclusion of these established error estimates helps highlight the key advantages of randomised quadrature and forms the theoretical basis for the application of such methods in subsequent chapters.

The basic intuition can be seen from the Dirichlet-type function

$$g(t) = \mathbf{1}_{\mathbb{Q}}(t), \qquad t \in [0, 1].$$

Its Lebesgue integral is zero, but the deterministic left-endpoint rule with $h = 1/N$ gives $h\sum_{j=0}^{N-1} g(jh) = 1$, since all grid points are rational. By contrast, a randomised rule samples $jh + h\gamma_{j+1}$, which belongs to $\mathbb{Q}$ with probability zero when γ_{j+1} is uniformly distributed on $(0, 1)$. Thus the randomised sum matches the integral almost surely in this toy example, illustrating how randomisation can avoid artefacts caused by a fixed deterministic grid.

Randomised quadrature schemes, first introduced in [38, 39], are typically the randomised analogues of classical deterministic quadrature rules. The following results, adapted from [62], provide sharp error estimates for the randomised Riemann sum (i.e., randomised quadrature) rule under different smoothness conditions on the integrand. To apply these results in our setting, we now present the precise construction of the randomised Riemann sum used to approximate the integral $\int_0^{t_n} g(s)ds$. Fix a terminal time $T \in (0, \infty)$. Given a step size $h \in (0, 1)$, define $N_h \in \mathbb{N}$ as the unique integer satisfying $N_h h \leq T < (N_h + 1)h$. For every measurable function $g : [0, T] \to \mathbb{R}^d$ with $\|g\|_{L^p([0,T];\mathbb{R}^d)} < \infty$ for some $p \in [2, \infty)$, the *randomised Riemann sum approximation* of $\int_0^{t_n} g(s)\,\mathrm{d}s$ is given by

$$Q^n_{\gamma,h}[g] := h\sum_{j=1}^{n} g(t_{j-1} + h\gamma_j), \quad n \in \{1, \dots, N_h\}, \tag{2.1}$$

where $t_j = jh$ and $(\gamma_j)_{j\in\mathbb{N}}$ is a family of independent random variables, each uniformly distributed on $(0, 1)$, defined on a probability space $(\Omega, \Sigma, \mathbb{P})$.

We begin with an error estimate in the $L^p(\Omega; \mathbb{R}^d)$-norm for the randomised Riemann sum. To formulate the result, we first recall the definition of the Hölder space $\mathcal{C}^\gamma([0, T])$ for $\gamma \in (0, 1]$. For every $\gamma \in (0, 1]$, we denote by $\mathcal{C}^\gamma([0, T]) := \mathcal{C}^\gamma([0, T]; \mathbb{R}^d)$ the set of all γ-Hölder continuous mappings $g\colon [0, T] \to \mathbb{R}^d$, also called Hölder continuous of order γ (or having Hölder regularity γ), that is, the set of all functions satisfying

$$\|g\|_{\mathcal{C}^\gamma([0,T])} := \sup_{t\in[0,T]} |g(t)| + \sup_{\substack{t,s\in[0,T]\\ t\neq s}} \frac{|g(t) - g(s)|}{|t - s|^\gamma} < \infty.$$

Equipped with this norm, $\mathcal{C}^\gamma([0, T])$ is a Banach space. In particular, every $g \in$

$\mathcal{C}^\gamma([0,T])$ satisfies

$$|g(t)-g(s)| \le \|g\|_{\mathcal{C}^\gamma([0,T])}|t-s|^\gamma, \quad \text{for all } t,s\in[0,T]. \tag{2.2}$$

More generally, if $(E,\|\cdot\|_E)$ is a Banach space, then $\mathcal{C}^\gamma([0,T];E)$ denotes the space of all mappings $Y:[0,T]\to E$ such that

$$\|Y\|_{\mathcal{C}^\gamma([0,T];E)} := \sup_{t\in[0,T]}\|Y(t)\|_E + \sup_{\substack{s,t\in[0,T]\\ s\neq t}} \frac{\|Y(t)-Y(s)\|_E}{|t-s|^\gamma} < \infty.$$

In particular, the notation $\mathcal{C}^\gamma([0,T];L^p(\Omega;\mathbb{R}^d))$ used below is understood with $E = L^p(\Omega;\mathbb{R}^d)$.

Theorem 2.2.1. *Let $g\colon[0,T]\to\mathbb{R}^d$ be a measurable mapping satisfying $\|g\|_{L^p([0,T];\mathbb{R}^d)} < \infty$ for some $p\in[2,\infty)$. Then, for every $h\in(0,1)$ and $n\in\{1,\dots,N_h\}$ the randomised Riemann sum $Q^n_{\gamma,h}[g]\in L^p(\Omega;\mathbb{R}^d)$ is an unbiased estimator for the integral $\int_0^{t_n} g(s)\,\mathrm{d}s$, i.e., $\mathbb{E}[Q^n_{\gamma,h}[g]] = \int_0^{t_n} g(s)\,\mathrm{d}s$. Further, for all $h\in(0,1)$ we have*

$$\Big\| \max_{n\in\{1,\dots,N_h\}} \Big| \int_0^{t_n} g(s)\,\mathrm{d}s - Q^n_{\gamma,h}[g]\Big| \Big\|_{L^p(\Omega;\mathbb{R})} \le 2C_p T^{\frac{p-2}{2p}} \|g\|_{L^p([0,T];\mathbb{R}^d)} h^{\frac12}. \tag{2.3}$$

In addition, if $g\in\mathcal{C}^\gamma([0,T])$ for some $\gamma\in(0,1]$, then for all $h\in(0,1)$ we have

$$\Big\| \max_{n\in\{1,\dots,N_h\}} \Big| \int_0^{t_n} g(s)\,\mathrm{d}s - Q^n_{\gamma,h}[g]\Big| \Big\|_{L^p(\Omega;\mathbb{R})} \le C_p\sqrt{T}\|g\|_{\mathcal{C}^\gamma([0,T])} h^{\frac12+\gamma}. \tag{2.4}$$

The proof first establishes that the randomised Riemann sum is unbiased. Indeed, by the independence and uniform distribution of γ_j on $(0,1)$, one has

$$\mathbb{E}\big[h\, g(t_{j-1}+h\gamma_j)\big] = \int_{t_{j-1}}^{t_j} g(s)\,ds,$$

and summing over j yields $\mathbb{E}[Q^n_{\gamma,h}[g]] = \int_0^{t_n} g(s)\,ds$. Consequently, the associated error process has increments with zero (conditional) expectation and therefore forms a discrete-time martingale with respect to the natural filtration. This property makes it possible to apply the BDG inequality to estimate the L^p-norm of the maximal er-

ror. Further estimates are obtained via Hölder's inequality and the properties of L^p and Hölder spaces. A complete proof can be found in [60]. It should be emphasized that the application of the BDG inequality critically depends on the randomness and independence in the construction; such an approach is not available for deterministic quadrature rules.

Remark 2.2.2 (Technique behind [60], Theorem 5.2)**.** The proof of the randomised Runge–Kutta error estimate in [60, Theorem 5.2] combines the preceding randomised-quadrature mechanism with a deterministic stability argument. For an ODE solution u and the randomised Runge–Kutta approximation, the global error is decomposed into three parts: a randomised quadrature error for $t \mapsto f(t, u(t))$, a local consistency error caused by replacing $u(\theta_j)$ by the Runge–Kutta stage based at t_{j-1}, and a propagation error depending on the previous grid error. The first part is controlled as above: the random time evaluations make the quadrature increments centred and independent, so the maximal L^p-error is estimated by the martingale/BDG argument, with the Hölder regularity in time yielding the $h^{1/2+\gamma}$ contribution. The local consistency term is estimated from the Hölder continuity in time and the regularity of the exact solution, while the propagation term is controlled by the global Lipschitz condition in the state variable and a discrete Gronwall argument. In the DDE chapter, the proof of Theorem 3.4.3 uses this same decomposition on each delay interval; the additional DDE-specific work is the induction over delay intervals and the auxiliary scheme used to separate the inherited delay error from the local randomised Runge–Kutta error.

Remark 2.2.3 (Deterministic Riemann sums)**.** In contrast, the standard deterministic Riemann sum defined by

$$Q_h^n[g] := h \sum_{j=1}^{n} g(t_{j-1})$$

may fail to converge if g is merely in $L^p([0,T])$ without additional continuity assumptions. Even under the stronger assumption $g \in \mathcal{C}^\gamma([0,T])$, the deterministic quadrature only achieves an error rate of order $O(h^\gamma)$ [48].

The error analysis for randomised quadrature can be extended to integrals of stochastic processes, which are central in the numerical treatment of SDEs and related prob-

lems. The following result, adapted from [62], provides L^p-error bounds for the randomised Riemann sum when the integrand is itself a stochastic process.

Here and below, Bochner measurability of an E-valued mapping means measurability with respect to the Borel σ-algebra of the Banach space E; equivalently, in the separable finite-dimensional setting used in this thesis, it is the usual joint measurability of the component functions. Let $Y : [0,T] \times \Omega_W \to \mathbb{R}^d$ be a stochastic process defined on a probability space $(\Omega_W, \Sigma^W, \mathbb{P}_W)$, and assume that Y is Bochner-measurable with $\|Y\|_{L^p([0,T]\times\Omega_W;\mathbb{R}^d)} < \infty$ for some $p \in [2,\infty)$. The goal turns to approximating the integral $\int_0^{t_n} Y(s)\,ds$. For the randomization, let $(\gamma_j)_{j\in\mathbb{N}}$ be a sequence of independent random variables, each uniformly distributed on $(0,1)$ and independent of Y, defined on an auxiliary probability space $(\Omega_\gamma, \Sigma^\gamma, \mathbb{P}_\gamma)$.

We consider the product probability space $(\Omega_{\text{tot}}, \Sigma^{\text{tot}}, \mathbb{P}_{\text{tot}}) = (\Omega_W \times \Omega_\gamma,\ \Sigma^W \otimes \Sigma^\gamma,\ \mathbb{P}_W \otimes \mathbb{P}_\gamma)$, where $\otimes$ denotes the product operation on σ-algebras and measures, defined by $\mathcal{A}\otimes\mathcal{B} := \sigma(\{A \times B : A \in \mathcal{A},\ B \in \mathcal{B}\})$. On this space, for each $n \in \{1,\dots,N_h\}$, we approximate the random variable $\int_0^{t_n} Y(s)\,\mathrm{d}s \in L^p(\Omega_W;\mathbb{R}^d)$ by the randomised Riemann sum

$$Q^n_{\gamma,h}[Y] := h\sum_{j=1}^{n} Y(t_{j-1} + \gamma_j h), \quad n \in \{1,\dots,N_h\}. \tag{2.5}$$

Theorem 2.2.4. *For $p \in [2,\infty)$ let $Y\colon [0,T]\times\Omega_W \to \mathbb{R}^d$ be a stochastic process with $Y \in L^p([0,T]\times\Omega_W;\mathbb{R}^d)$. Then, for every $h \in (0,1)$ and $n \in \{1,\dots,N_h\}$, the randomised Riemann sum approximation $Q^n_{\gamma,h}[Y] \in L^p(\Omega_{tot};\mathbb{R}^d)$ defined in (2.5) is an unbiased estimator for the integral $\int_0^{t_n} Y(s)\,\mathrm{d}s$ in the sense that*

$$\mathbb{E}_\gamma[Q^n_{\gamma,h}[Y]] = \int_0^{t_n} Y(s)\,\mathrm{d}s \in L^p(\Omega_W;\mathbb{R}^d). \tag{2.6}$$

Moreover, it holds that

$$\begin{aligned}\Big\| \max_{n\in\{1,\dots,N_h\}} \Big| Q^n_{\gamma,h}[Y] - \int_0^{t_n} Y(s)\,\mathrm{d}s \Big| \Big\|_{L^p(\Omega_{tot};\mathbb{R})} \\ \le 2C_p T^{\frac{p-2}{2p}} \|Y\|_{L^p([0,T]\times\Omega_W;\mathbb{R}^d)} h^{\frac{1}{2}},\end{aligned} \tag{2.7}$$

where C_p is a constant only depending on $p \in [2, \infty)$.

In addition, if $Y \in \mathcal{C}^\gamma([0,T], L^p(\Omega_W; \mathbb{R}^d))$ for some $\gamma \in (0,1]$, then we have

$$\begin{aligned} &\Big\| \max_{n \in \{1,\dots,N_h\}} \Big| Q^n_{\gamma,h}[Y] - \int_0^{t_n} Y(s)\,\mathrm{d}s \Big| \Big\|_{L^p(\Omega_{tot};\mathbb{R})} \\ &\quad \le C_p \sqrt{T} \|Y\|_{\mathcal{C}^\gamma([0,T],L^p(\Omega_W;\mathbb{R}^d))} h^{\frac{1}{2}+\gamma}, \end{aligned} \tag{2.8}$$

where C_p is the same constant as in (2.7).

2.3 A Short Hierarchy of Randomised Time-Stepping Schemes

The quadrature estimates above explain the basic source of randomisation: a fixed deterministic evaluation time is replaced by a random point inside the current time step. For clarity, the following schematic overview gives a formula-level guide from classical deterministic schemes to the randomised schemes used later, especially the delay schemes in Chapters 3 and 4. The notation in this section is schematic: the complete assumptions, grids, measurability statements, and implementable definitions are given in the relevant chapters. In particular, for delay equations the superscript j below indicates the jth delay interval, while the subscript k indicates the grid point inside that interval.

Consider first an ODE

$$u'(t) = F(t, u(t)), \qquad u(0) = u_0,$$

on a grid $t_k = kh$. The deterministic Euler method uses the left endpoint of each interval (see, for example, [13]),

$$U_{k+1} = U_k + hF(t_k, U_k).$$

The randomised Euler method replaces this deterministic evaluation time by $\theta_{k+1} = t_k + h\gamma_{k+1}$, where γ_{k+1} is uniformly distributed on $(0,1)$:

$$U_{k+1} = U_k + hF(\theta_{k+1}, U_k).$$

This is the basic mechanism studied, for example, in [64]; it is useful when $t \mapsto F(t,x)$ has low regularity, because the update no longer relies on a fixed grid of point evaluations.

Runge–Kutta methods add intermediate stages [13, 40]. A deterministic explicit two-stage method first forms a stage value and then evaluates the vector field at that stage, for instance

$$\widehat{U}_{k+1} = U_k + chF(t_k, U_k), \qquad U_{k+1} = U_k + hF(t_k + ch, \widehat{U}_{k+1}),$$

with a prescribed stage parameter $c \in [0,1]$. In a randomised Runge–Kutta method, the stage time is randomised. With the same notation $\theta_{k+1} = t_k + h\gamma_{k+1}$ and $h_{k+1} = h\gamma_{k+1}$, the corresponding two-stage randomised update has the form

$$\widehat{U}_{k+1} = U_k + h_{k+1}F(t_k, U_k), \qquad U_{k+1} = U_k + hF(\theta_{k+1}, \widehat{U}_{k+1}).$$

This is the ODE-level idea behind the randomised Runge–Kutta analysis of [60]: the random stage time gives a randomised quadrature error, while the stage value gives the Runge–Kutta local consistency correction.

For SDEs, the solution is already random because of the driving Wiener process, but the coefficient evaluation times may still be randomised. For the non-autonomous SDE

$$\mathrm{d}X(t) = F(t, X(t))\,\mathrm{d}t + G(t, X(t))\,\mathrm{d}W(t)$$

the deterministic Euler–Maruyama method evaluates both coefficients at the left endpoint [59]:

$$U_{k+1} = U_k + hF(t_k, U_k) + G(t_k, U_k)(W(t_{k+1}) - W(t_k)).$$

The randomised Euler–Maruyama method keeps the diffusion coefficient adapted to the left endpoint but replaces the drift evaluation time by a random point in the step:

$$U_{k+1} = U_k + hF(\theta_{k+1}, U_k) + G(t_k, U_k)(W(t_{k+1}) - W(t_k)).$$

Here the randomisation is used for the drift integral, while the stochastic term is estimated through martingale inequalities such as BDG. Randomised schemes for time-irregular SDEs and related stochastic equations are analysed, for example, in [62,74,86].

Delay equations require one further layer of notation. On the interval $[j\tau, (j+1)\tau]$, the delayed value belongs to the preceding interval. For a DDE

$$x'(t) = f(t, x(t), x(t-\tau)),$$

a deterministic Euler update on this interval, in line with standard numerical treatments of delay equations [6, 66], would have the schematic form

$$y_{k+1}^j = y_k^j + hf(t_k^j, y_k^j, y_k^{j-1}),$$

where y_k^j approximates the solution at $j\tau + kh$, and y_k^{j-1} represents the delayed value. Randomising the time evaluation gives the corresponding randomised Euler DDE update

$$y_{k+1}^j = y_k^j + hf(\theta_{k+1}^j, y_k^j, y_k^{j-1}),$$

which is the randomised Euler structure used for Carathéodory DDEs in [28]. The two-stage randomised Runge–Kutta method developed in Chapter 3, based on [29], can then be viewed as the next layer: it adds not only a current-stage value but also a delayed-stage value:

$$\tilde{y}_{k+1}^j \quad \text{and} \quad \tilde{y}_{k+1}^{j-1,j}.$$

The first quantity predicts the current state at the random stage time in the interval $[j\tau, (j+1)\tau]$; the second predicts the corresponding delayed state using data from the previous interval. This explains why the DDE scheme in Chapter 3 contains two intermediate terms rather than the single intermediate stage seen in the ODE formula.

The SDDE scheme in Chapter 4, based on [90], follows the same delay-interval idea, but with an additional Brownian increment. Its randomised Euler update has the

schematic form

$$y_{k+1}^j = y_k^j + hf(\theta_{k+1}^j, y_k^j, y_k^{j-1}) + g(t_k^j, y_k^j, y_k^{j-1})(W(t_{k+1}^j) - W(t_k^j)).$$

Thus the drift is evaluated at a random time point, while the diffusion term remains adapted to the left endpoint. The error analysis then combines the randomised quadrature idea for the drift, BDG estimates for the stochastic integral, and induction over delay intervals to propagate the delayed error.

2.4 Further Notation and Useful Inequalities

We collect here some further notation, basic concepts, and useful lemmas that will be used throughout this thesis. $C(A; B)$ denotes the set of all continuous mappings from A to B. The indicator function of a set A is denoted by $\mathbf{1}_A(t)$, which equals 1 if $t \in A$ and 0 otherwise. A mapping $f : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ is referred to as a *Carathéodory function* if, for each fixed $x \in \mathbb{R}^d$, the function $t \mapsto f(t, x)$ is measurable, and for almost every $t \in [0, T]$, the mapping $x \mapsto f(t, x)$ is continuous. In this thesis, we refer to problems with a right-hand side f satisfying the above conditions as *Carathéodory-type problems*. The Carathéodory setting is useful here because it allows low-regularity time dependence, including jumps or rapidly oscillating coefficients, while keeping enough continuity in the state variable for standard well-posedness arguments under suitable growth and Lipschitz-type assumptions. More singular state dependence or genuinely path-dependent coefficients require different stability estimates and are therefore outside the scope of this thesis.

A mapping $f : D \to \mathbb{R}^m$ is said to satisfy the *Lipschitz condition* on D if there exists a constant $L > 0$ such that

$$|f(x) - f(y)| \leq L|x - y|, \quad \forall\, x, y \in D.$$

The condition above corresponds to the special case $\gamma = 1$ of Hölder continuity. The following inequality will be used repeatedly in the error analysis.

Lemma 2.4.1 (Hölder's inequality)**.** *Let $\{a_j\}, \{b_j\}$ be real or complex numbers, or let f, g be measurable functions. Let $p, q \in [1, \infty]$ satisfy $\frac{1}{p} + \frac{1}{q} = 1$. Then*

$$\sum_j |a_j b_j| \leq \left(\sum_j |a_j|^p \right)^{1/p} \left(\sum_j |b_j|^q \right)^{1/q}$$

in the discrete (finite sum) case, and

$$\int |f(x)g(x)|\, dx \leq \left(\int |f(x)|^p dx \right)^{1/p} \left(\int |g(x)|^q dx \right)^{1/q}$$

in the integral (continuous) case.

The proof appears in standard textbooks; see, for instance, [57, Theorem 2.6] or [35, Theorem 6.8]. Jensen's inequality will also play a key role in various estimates later in this thesis.

Lemma 2.4.2 (Jensen's Inequality)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let X be an integrable real-valued random variable, and let $\varphi : \mathbb{R} \to \mathbb{R}$ be a convex function. Then*

$$\varphi\big(\mathbb{E}[X]\big) \leq \mathbb{E}\big[\varphi(X)\big]. \tag{2.9}$$

In particular, for $p \geq 1$ and any random variable X,

$$\mathbb{E}\big[|X|^p\big] \geq \big|\mathbb{E}[X]\big|^p. \tag{2.10}$$

A proof can be found in [57, Proposition 5.1].

Remark 2.4.3. The formulation of Jensen's inequality in Lemma 2.4.2 is the probabilistic one. In applications later in this thesis we will also make use of its integral and discrete analogues. For $f \in L^q([0, \tau])$ with $q \geq 1$,

$$\Big| \int_0^\tau f(s)\, ds \Big|^q \;\leq\; \tau^{q-1} \int_0^\tau |f(s)|^q\, ds, \tag{2.11}$$

which corresponds to applying Lemma 2.4.2 on the measure space $([0, \tau], \frac{ds}{\tau})$ with the

convex function $x \mapsto |x|^q$. Similarly, for a finite sequence $(x_j)_{j=1}^N \subset \mathbb{R}$ and $q \geq 1$,

$$\Big|\sum_{j=1}^{N} x_j\Big|^q \;\leq\; N^{\,q-1} \sum_{j=1}^{N} |x_j|^q, \tag{2.12}$$

that is, Jensen's inequality applied to the uniform measure on the finite set $\{1, \dots, N\}$.

The following continuous version of the BDG inequality will be used later in the analysis of SDDEs. It is stated here in the form used below; see, for example, [93, Chapter IV, Theorem 48].

Lemma 2.4.4 (Continuous Burkholder–Davis–Gundy inequality)**.** *Let $(\Omega, \Sigma, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$ be a filtered probability space satisfying the usual conditions, and let $M = (M_t)_{t\geq 0}$ be a continuous local martingale with $M_0 = 0$. For every $p \in (0, \infty)$ there exist constants $c_p, C_p > 0$ (depending only on p and the dimension) such that for all $T \geq 0$,*

$$c_p\, \mathbb{E}([M]_T^{p/2}) \;\leq\; \mathbb{E}\Big(\sup_{0\leq t\leq T} |M_t|^p\Big) \;\leq\; C_p\, \mathbb{E}([M]_T^{p/2}),$$

where $[M]_T$ denotes the quadratic variation of M up to time T.

Corollary 2.4.5. *Let W be an m-dimensional Wiener process with respect to $(\mathcal{F}_t)$ and let $G : [0,T] \times \Omega \to \mathbb{R}^{d\times m}$ be a predictable process satisfying $\int_0^T |G(s)|^2 \,\mathrm{d}s < \infty$ a.s. (so that the Itô integral $\int_s^t G(r)\,\mathrm{d}W(r)$ is well-defined for all $0 \leq s \leq t \leq T$). Then for every $p \in (0, \infty)$ there exists a constant $C_p > 0$ such that for all $0 \leq s \leq t \leq T$,*

$$\mathbb{E}\left[\sup_{u\in[s,t]} \Big|\int_s^u G(r)\,\mathrm{d}W(r)\Big|^p\right] \leq C_p\, \mathbb{E}\left[\Big(\int_s^t |G(r)|^2\,\mathrm{d}r\Big)^{p/2}\right].$$

In particular, if $p \geq 2$, then by applying Hölder's and Jensen's inequalities with respect to the time integral (that is, with respect to the Lebesgue variable in the stochastic integral, not with respect to the probability expectation) one further has

$$\mathbb{E}\Big|\int_s^t G(r)\,\mathrm{d}W(r)\Big|^p \leq C_p\,(t-s)^{\frac{p}{2}-1} \int_s^t \mathbb{E}|G(r)|^p\,\mathrm{d}r.$$

Lemma 2.4.6 (SDEs with progressively measurable Lipschitz coefficients)**.** *Let $T \in (0,\infty)$, $p \geq 2$, and let W be an m-dimensional Wiener process on a filtered probability space satisfying the usual conditions. Consider*

$$X(t) = \xi + \int_0^t b(s, X(s))\,\mathrm{d}s + \int_0^t \sigma(s, X(s))\,\mathrm{d}W(s), \qquad t \in [0,T],$$

where ξ is $\mathcal{F}_0$-measurable with $\mathbb{E}|\xi|^p < \infty$. Assume that, for each $x \in \mathbb{R}^d$, the processes $b(\cdot, x)$ and $\sigma(\cdot, x)$ are progressively measurable, and that there is a constant $L > 0$ such that, for all $x, y \in \mathbb{R}^d$ and almost all (t,ω),

$$|b(t,x) - b(t,y)| + |\sigma(t,x) - \sigma(t,y)| \leq L|x-y|.$$

Assume moreover that the growth at the origin is p-integrable, for instance

$$\mathbb{E}\Big(\int_0^T |b(t,0)|\,\mathrm{d}t\Big)^p + \mathbb{E}\Big(\int_0^T |\sigma(t,0)|^2\,\mathrm{d}t\Big)^{p/2} < \infty.$$

Then the SDE has a unique strong solution with continuous adapted trajectories and

$$\mathbb{E}\Big(\sup_{0\leq t\leq T} |X(t)|^p\Big) < \infty.$$

This is the form of the classical result from [81, Proposition 3.28, p. 187] used in Chapter 4; the corresponding moment estimate is given in [81, Remark 3.29, p. 188].

The subsequent inequality provides an effective means to estimate the error arising in numerical approximations. A detailed proof, along with several generalizations, can be found in [31, Proposition 4.1].

Lemma 2.4.7 (Discrete Gronwall's inequality)**.** *Consider two non-negative sequences $(u_n)_{n\in\mathbb{N}}$ and $(w_n)_{n\in\mathbb{N}}$ which for some given $a \in [0,\infty)$ satisfy*

$$u_n \leq a + \sum_{j=1}^{n-1} w_j u_j, \qquad \text{for all } n \in \mathbb{N},$$

then for all $n \in \mathbb{N}$ *it also holds true that*

$$u_n \leq a \exp\Big(\sum_{j=1}^{n-1} w_j\Big).$$

The following result on the properties of solutions to Carathéodory-type ODEs will be used in the chapter on DDEs. In particular, it provides key existence, uniqueness, and regularity estimates that will be instrumental in establishing well-posedness and further analytical properties of DDEs discussed later in this thesis. It follows from [60, Proposition 4.2].

Lemma 2.4.1. *Let us consider the following ODE*

$$z'(t) = g(t, z(t)), \quad t \in [a, b], \quad z(a) = \xi, \tag{2.13}$$

where $-\infty < a < b < +\infty$, $\xi \in \mathbb{R}^d$ *and* $g : [a, b] \times \mathbb{R}^d \to \mathbb{R}^d$ *satisfies the following conditions*

(G1) *for all* $t \in [a, b]$ *the function* $g(t, \cdot) : \mathbb{R}^d \to \mathbb{R}^d$ *is continuous,*

(G2) *for all* $y \in \mathbb{R}^d$ *the function* $g(\cdot, y) : [a, b] \to \mathbb{R}^d$ *is Borel measurable,*

(G3) *there exists* $K \in (0, \infty)$ *such that for all* $t \in [a, b]$, $y \in \mathbb{R}^d$

$$|g(t, y)| \leq K(1 + |y|),$$

(G4) *there exists* $L \in (0, \infty)$ *such that for all* $t \in [a, b]$, $x, y \in \mathbb{R}^d$

$$|g(t, x) - g(t, y)| \leq L|x - y|.$$

Then (2.13) *has a unique absolutely continuous solution* $z : [a, b] \to \mathbb{R}^d$ *such that*

$$\sup_{t \in [a,b]} |z(t)| \leq (|\xi| + K(b - a))e^{K(b-a)}. \tag{2.14}$$

Moreover, for all $t, s \in [a, b]$

$$|z(t) - z(s)| \leq \bar{K}|t - s|, \tag{2.15}$$

where $\bar{K} = K \cdot \Big(1 + (|\xi| + K(b-a))e^{K(b-a)}\Big)$.

Remark 2.4.8. Assumption (G4) implies the continuity requirement in Assumption (G1), since Lipschitz continuity in the state variable is stronger than continuity. However, Assumption (G4) does not by itself imply the linear growth condition in Assumption (G3): one also needs a uniform bound on $|g(t, 0)|$ over $t \in [a, b]$. Indeed, if such a bound is available, then

$$|g(t, y)| \leq |g(t, 0)| + L|y| \leq (M + L)(1 + |y|),$$

where $M := \sup_{t\in[a,b]} |g(t, 0)|$.

Proof. The proof for our proposition follows the approach outlined in Lemma 7.1 of [28]. This is feasible because our assumptions (G3) and (G4) are stronger than [28, Assumptions (G3) and (G4)]. Particularly, [28, Eqn. (7.8) and Eqn. (7.9)] will undergo modifications as follows due to the strengthened conditions:

$$|z(t) - z(s)| \leq K(1 + \sup_{a\leq t\leq b} |z(t)|)|t - s|. \tag{2.16}$$

Consequently, this gives rise to our refined conclusions. □

Chapter 3

A randomised Runge-Kutta Scheme for time-irregular delay differential equations

3.1 Introduction

We explore the application of randomised numerical schemes to the approximation of solutions for delay differential equations (DDEs) of the following type:

$$\begin{cases} x'(t) = f(t, x(t), x(t-\tau)), & t \in [0, (n+1)\tau], \\ x(t) = \varphi(t), & t \in [-\tau, 0], \end{cases} \tag{3.1}$$

with a constant time-lag $\tau \in (0, +\infty)$, a fixed time horizon $n \in \mathbb{N}$, a right-hand side function $f : [0, (n+1)\tau] \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, and initial-value function: $\varphi(t) : [-\tau, 0] \to \mathbb{R}^d$.

We assume $t \mapsto f(t, x, z)$ is measurable and $(x, z) \mapsto f(t, x, z)$ is continuous, i.e., a Carathéodory setting under which existence and uniqueness of solutions are ensured. For error bounds we will strengthen these assumptions to Hölder continuity in time and suitable regularity in (x, z).

Building upon the concepts presented in [28], our objective in this study is to introduce a randomised Runge-Kutta scheme tailored specifically for time-irregular

DDEs. This novel scheme differentiates itself from existing methods, including those analysed in the literature.

The motivation behind exploring these DDEs is multifaceted. One aspect stems from the need to model switching systems with memory, as expounded upon in [42] and [43]. Another facet is rooted in practical engineering applications, as evidenced by discussions in [21] and [20]. For instance, in scenarios like the phase change of metallic materials, a DDE becomes crucial due to the time delay in response to changes in processing conditions. In addition, low regularity in the time variable may arise naturally when the system is perturbed by external signals with discontinuities or rapid oscillations. Such perturbations can lead to Carathéodory-type right-hand sides $f(t, x, z)$, where $t \mapsto f(t, x, z)$ is merely integrable.

When classical assumptions, such as C^r-regularity of $f = f(t, x, z)$ concerning all variables t, x, z, are imposed on the right-hand side function, errors for deterministic schemes have been documented in [6]. Furthermore, in [21], the error of the Euler scheme has been explored for a certain class of non-linear DDEs under non-standard assumptions like the one-side Lipschitz condition and local Hölder continuity. However, limited knowledge exists regarding the approximation of solutions for DDEs with less regular Carathéodory right-hand side functions. As discussed above, for Carathéodory ODEs, deterministic algorithms encounter convergence issues, necessitating the use of randomised algorithms. In this context, we propose a randomised version of the Runge-Kutta scheme tailored for DDEs of the form (3.1).

In the landscape of existing literature, several relevant contributions are worth noting. [32] provides an analysis of multistep Runge-Kutta schemes for DDEs, while [66] offers comprehensive insights into numerical methods for DDEs. The work by [10] introduces a general-purpose implicit-explicit Runge-Kutta integrator for DDEs, and [97] proposes a novel explicit two-stage Runge-Kutta scheme for DDEs with constant delay. Each of these works contributes valuable perspectives to the broader understanding of numerical methods for DDEs. As mentioned earlier, despite the extensive research on randomised algorithms for ODEs, to the best of our knowledge there is no previous systematic study of a randomised Runge–Kutta scheme for time-irregular (Carathéodory-

type in time) DDEs. In particular, this chapter introduces such a scheme and establishes convergence and error bounds under strengthened regularity assumptions on the drift, including Hölder continuity in time and appropriate regularity in the delay variable.

The structure of this chapter is as follows. In Section 3.2, we introduce the model assumptions and describe the construction of the randomised two-stage Runge-Kutta scheme in detail. Section 3.3 is devoted to properties of solutions to DDEs under assumptions stated in Section 3.2. Section 3.4 contains detailed error analysis of the randomised Runge-Kutta scheme for DDEs. Finally, details of the Python implementation together with extensive numerical experiments are reported in Section 3.5.

3.2 Assumptions and Algorithm

We consider a complete probability space $(\Omega, \Sigma, \mathbb{P})$. For a random variable $X : \Omega \to \mathbb{R}$, we denote its $L^p(\Omega)$-norm by $\|X\|_p = (\mathbb{E}|X|^p)^{1/p}$, where $p \in [2, +\infty)$. Let us fix the *horizon parameter* $n \in \mathbb{N}$. On the right-hand side function f and initial-value function: $\varphi(t)$, we impose the following assumptions:

(A1) $f(t, \cdot, \cdot) \in C(\mathbb{R}^d \times \mathbb{R}^d; \mathbb{R}^d)$ for all $t \in [0, (n+1)\tau]$ and $\varphi(t) \in C([-\tau, 0]; \mathbb{R}^d)$ for $t \in [-\tau, 0]$.

(A2) $f(\cdot, x, z)$ is Borel measurable for all $(x, z) \in \mathbb{R}^d \times \mathbb{R}^d$,

(A3) There exists $K \in (0, \infty)$ such that for all $t \in [0, (n+1)\tau]$, $x, z \in \mathbb{R}^d$

$$|f(t, x, z)| \leq K(1 + |x|)(1 + |z|), \tag{3.2}$$

(A4) There exists $L \in (0, \infty)$ such that for all $t \in [0, (n+1)\tau]$, $x_1, x_2, z \in \mathbb{R}^d$

$$|f(t, x_1, z) - f(t, x_2, z)| \leq L(1 + |z|)|x_1 - x_2|, \tag{3.3}$$

Remark 3.2.1. Assumption (A4) implies continuity of f with respect to the current state variable x, for each fixed pair (t, z). It does not imply Assumption (A3), since

(A3) is a growth condition on the value of $f(t, x, z)$ itself. To derive such a bound from (A4), one would additionally need a suitable bound on $f(t, 0, z)$, uniformly in t and with controlled growth in z.

In Section 3.3, under the assumptions (A1)-(A4), we investigate existence and uniqueness of solution for (3.1). Next, in Section 3.4 we investigate error of the *randomised two-stage Runge-Kutta scheme* under slightly stronger assumptions. Namely, we impose Hölder continuity assumptions on $f = (t, x, z)$ with respect to (t, z).

The construction below follows the hierarchy of time-stepping schemes recalled in Section 2.3: starting from randomised Euler time evaluation, we add a Runge–Kutta intermediate current state and, because of the delay, a matching intermediate delayed state.

Before introducing the two-stage method, we recall the simpler randomised Euler-type baseline used for comparison with Carathéodory DDEs in [28]. With the notation below, this first-order baseline is obtained by setting $y_k^{(-1)} = \varphi(t_k^{(-1)})$ and, for $j = 0$,

$$\begin{aligned} y_0^0 &= y_N^{(-1)}, \\ y_{k+1}^0 &= y_k^0 + hf(\theta_{k+1}^0, y_k^0, y_k^{(-1)}), \qquad k = 0, 1, \ldots, N-1, \end{aligned}$$

while for $j \geq 1$ it is given by

$$\begin{aligned} y_0^j &= y_N^{j-1}, \\ y_{k+1}^j &= y_k^j + hf(\theta_{k+1}^j, y_k^j, y_k^{j-1}), \qquad k = 0, 1, \ldots, N-1. \end{aligned}$$

Thus this baseline evaluates the drift at a random time point in each step, but it does not use an intermediate Runge–Kutta stage. The randomised two-stage Runge–Kutta scheme introduced next can be viewed as a higher-stage variant of this randomised time-evaluation idea, and it will be compared numerically with this Euler baseline in Section 3.5.

The definition of the two-stage randomised Runge-Kutta scheme for DDEs goes as follows. Let $(\gamma_k^j)_{j\in\mathbb{N}_0, k\in\mathbb{N}}$ be i.i.d. from $\mathcal{U}(0,1)$, $N \in \mathbb{N}$, $h = \tau/N$, $t_k^j = j\tau + kh$,

$\theta_{k+1}^j = t_k^j + h\gamma_{k+1}^j$ for $k = 0, 1, \ldots, N$, $j = -1, 0, 1, \ldots, n$. The parenthesised superscript (-1) is used only to label the initial-history interval $[-\tau, 0]$, so as to distinguish it from ordinary delay indices such as $j-1$. Also let $h_{k+1}^j := h\gamma_{k+1}^j$ and define $y_k^{(-1)} = \varphi(t_k^{(-1)})$.

For $j = 0$, define

$$\begin{aligned} y_0^0 &= y_N^{(-1)}, \\ \tilde{y}_{k+1}^{(-1,0)} &= \varphi(t_k^{(-1)} + h_{k+1}^0) \\ \tilde{y}_{k+1}^0 &= y_k^0 + h_{k+1}^0 f(t_k^0, y_k^0, y_k^{(-1)}), \\ y_{k+1}^0 &= y_k^0 + hf(\theta_{k+1}^0, \tilde{y}_{k+1}^0, \tilde{y}_{k+1}^{(-1,0)}), \end{aligned} \tag{3.4}$$

and for $j \geq 1$,

$$\begin{aligned} y_0^j &= y_N^{j-1}, \\ \tilde{y}_{k+1}^{j-1,j} &= y_k^{j-1} + h_{k+1}^j f(t_k^{j-1}, y_k^{j-1}, y_k^{j-2}), \\ \tilde{y}_{k+1}^j &= y_k^j + h_{k+1}^j f(t_k^j, y_k^j, y_k^{j-1}), \\ y_{k+1}^j &= y_k^j + hf(\theta_{k+1}^j, \tilde{y}_{k+1}^j, \tilde{y}_{k+1}^{j-1,j}). \end{aligned} \tag{3.5}$$

Note that for $j = 1$, the delay term y_k^{j-2} in the second line of (3.5) is exactly the evaluation of the initial condition $\varphi(t_k^{(-1)})$.

Adapted from the schematic presentation in [29], Figures 3.1 and 3.2 illustrate the information flow in the two-stage randomised Runge–Kutta construction. The first figure shows the initial interval, where the delayed value is supplied by the history φ, while the second shows a later interval, where both current and delayed intermediate values are computed from previously available numerical data.

Let us call $\tilde{y}_{k+1}^{j-1,j}$ and $\tilde{y}_{k+1}^j$ *the intermediate delay term* and *the intermediate term* respectively. A key component here is the intermediate delay term $\tilde{y}_{k+1}^{j-1,j}$: we do not recycle the intermediate term $\tilde{y}_{k+1}^{j-1}$ computed from preceding τ-interval $[(j-1)\tau, j\tau]$ to replace $\tilde{y}_{k+1}^{j-1,j}$ for a simplified computation, because $\tilde{y}_{k+1}^{j-1}$ is generated on γ_{k+1}^{j-1}, a different random resource from γ_{k+1}^j. Using the random variable from the current interval for both the current intermediate term and its delayed counterpart keeps the approximation aligned at the same random quadrature point. If one recycled the in-

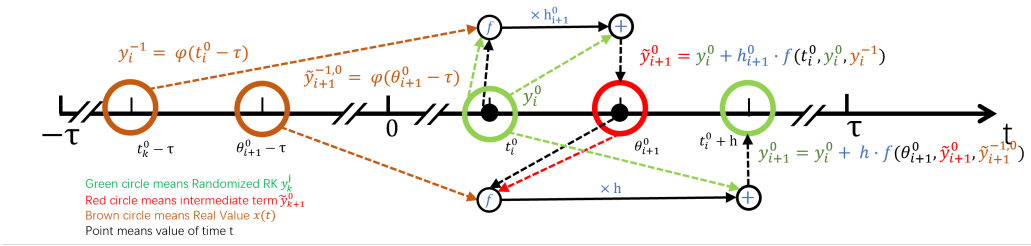


Figure 3.1: Schematic illustration of the randomised Runge–Kutta construction on the initial delay interval $j = 0$, adapted from [29].

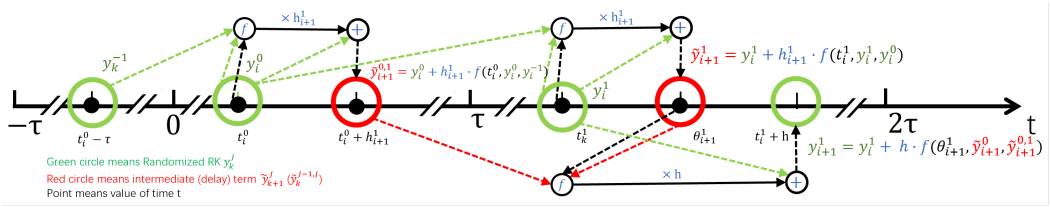


Figure 3.2: Schematic illustration of the randomised Runge–Kutta construction on a later delay interval, shown for $j = 1$, adapted from [29].

termediate value generated with γ_{k+1}^{j-1} from the previous interval, the delayed argument would correspond to a different random time than the current stage, which would make the measurability structure and the error decomposition less transparent. It is easy to see that each random vector y_k^j, $j = 0, 1, \ldots, n$, $k = 0, 1, \ldots, N$, is measurable with respect to the σ-field generated by the following family of independent random variables

$$\left\{\theta_1^0, \ldots, \theta_N^0, \ldots, \theta_1^{j-1}, \ldots, \theta_N^{j-1}, \theta_1^j, \ldots, \theta_k^j\right\}. \tag{3.6}$$

Equivalently, one may write the same σ-field in terms of the variables γ_i^r, because $\theta_i^r = t_{i-1}^r + h\gamma_i^r$ and the grid points and h are deterministic. Hence $\sigma(\theta_i^r) = \sigma(\gamma_i^r)$ for each fixed pair (r, i). We use the notation with θ_i^r to emphasise that the scheme evaluates the coefficient at random time points.

As the horizon parameter n is fixed, the randomised two-stage Runge-Kutta scheme uses $O(N)$ evaluations of f (with a constant in the O notation that only depends on n but not on N).

3.3 Properties of solutions to Carathéodory DDEs

This section investigates the existence and uniqueness of solutions to (3.1), assuming conditions (A1)–(A4) hold.

In the sequel we use the following equivalent representation of the solution of (3.1), that is very convenient when proving its properties and when estimating the error of the randomised Runge-Kutta scheme. For $j = 0, 1, \ldots, n$ and $t \in [0, \tau]$ it holds

$$x'(t + j\tau) = f(t + j\tau, x(t + j\tau), x(t + (j - 1)\tau)). \tag{3.7}$$

Hence, we take $\phi_{-1}(t) := \varphi(t - \tau)$, $\phi_j(t) := x(t + j\tau)$ and for $j = 0, 1, \ldots, n$ we consider the following sequence of initial-value problems

$$\begin{cases} \phi_j'(t) = g_j(t, \phi_j(t)), \quad t \in [0, \tau], \\ \phi_j(0) = \phi_{j-1}(\tau), \end{cases} \tag{3.8}$$

with $g_j(t, x) = f(t + j\tau, x, \phi_{j-1}(t))$, $(t, x) \in [0, \tau] \times \mathbb{R}^d$. Then the solution of (3.1) can be written as

$$x(t) = \sum_{j=-1}^{n} \phi_j(t - j\tau) \cdot \mathbf{1}_{[j\tau,(j+1)\tau]}(t), \quad t \in [-\tau, (n + 1)\tau]. \tag{3.9}$$

We prove the following result about existence, uniqueness and Hölder regularity of the solution of the DDE (3.1). We will use this theorem in the next section when proving error estimate for the randomised Runge-Kutta algorithm.

Theorem 3.3.1. *Let $n \in \mathbb{N} \cup \{0\}$, $\tau \in (0, +\infty)$ and let f, φ satisfy the assumptions (A1)-(A4). Then there exists a unique absolutely continuous solution x to* (3.1) *such that for $j = 0, 1, \ldots, n$ we have*

$$\sup_{0 \le t \le \tau} |\phi_j(t)| \le K_j, \tag{3.10}$$

where $K_{-1} := \max_{t\in[-\tau,0]} |\varphi(t)|$ *and*

$$K_j = (1 + K_{j-1})(1 + K\tau) \cdot \exp\Big((1 + K_{j-1})K\tau\Big). \tag{3.11}$$

Then for all $j = 0, 1, \ldots, n$, $t, s \in [0, \tau]$ *it holds*

$$|\phi_j(t) - \phi_j(s)| \leq (1 + K_{j-1})(1 + K_j)K|t - s|. \tag{3.12}$$

Remark 3.3.2 (Relation to Theorem 3.1 in [28])**.** *The proof of Theorem 3.3.1 uses the same interval-by-interval strategy as Theorem 3.1 in [28]. The key idea there is to split the delay equation into successive intervals of length* τ*. On the first interval, the delayed term is fully determined by the initial condition, so one obtains an ordinary Carathéodory initial-value problem. Once the solution has been constructed on one interval, it is treated as a known delayed input on the next interval, and the next interval is again reduced to a Carathéodory initial-value problem. Repeating this argument gives existence and uniqueness on the whole interval, while Gronwall-type estimates provide the interval-wise bounds.*

There are two differences in the present theorem. First, [28, Theorem 3.1] is stated for a constant initial condition $\varphi(t) = x_0$*, whereas here the initial history* φ *may depend on time on* $[-\tau, 0]$*. This only changes the first interval through the function* $g_0(t, x) = f(t, x, \varphi(t - \tau))$*. Secondly, [28] uses time-dependent linear-growth and Lipschitz assumptions on* f *([28], Assumptions A3 and A4), whereas Assumptions (A3) and (A4) impose stronger uniform constants. Hence the same induction applies, but the bounds are written here with the explicit constants* K_j *and the Lipschitz estimate* (3.12)*.*

Proof. We use the interval-by-interval induction described in the preceding remark. We start with the case when $j = 0$ and consider the following initial-value problem.

$$\begin{cases} \phi_0'(t) = g_0(t, \phi_0(t)), \quad t \in [0, \tau], \\ \phi_0(0) = \varphi(0), \end{cases} \tag{3.13}$$

with $g_0(t,x) = f(t,x,\phi_{-1}(t)) = f(t,x,\varphi(t-\tau))$. It is obvious that for all $t \in [0,\tau]$, the function $g_0(t,\cdot)$ is continuous and for all $x \in \mathbb{R}^d$, the function $g_0(\cdot,x)$ is Borel measurable. Moreover, by (3.2) we have

$$|g_0(t,x)| \leq K(1+|\varphi(t-\tau)|)(1+|x|) \leq K(1+K_{-1})(1+|x|), \tag{3.14}$$

where $K_{-1} := \max_{t\in[-\tau,0]} |\varphi(t)|$, for all $(t,x) \in [0,\tau] \times \mathbb{R}^d$, and by (3.3) there exists $L \in (0,\infty)$ such that for all $t \in [0,\tau]$, $x,y \in \mathbb{R}^d$ we have

$$|g_0(t,x) - g_0(t,y)| \leq L(1+|\varphi(t-\tau)|)|x-y| \leq L(1+K_{-1})|x-y|. \tag{3.15}$$

Therefore, by Lemma 2.4.1 we have that there exists a unique absolutely continuous solution $\phi_0 : [0,\tau] \to \mathbb{R}^d$ of (3.13) that satisfies (3.10) with $j = 0$. In addition, by Lemma 2.4.1 we obtain that ϕ_0 satisfies (3.12) for $j = 0$. For the inductive step from j to $j+1$, we can simply follow the approach provided in [28], since our assumptions (A3) and (A4) are stronger than [28, Assumptions A3 and A4]. It is crucial to note that [28, Equation (3.14)] will be modified to

$$|\phi_{j+1}(t) - \phi_{j+1}(s)| \leq (1+K_j)(1+K_{j+1})K|t-s|,$$

due to the differences between our Lemma 2.4.1 and [28, Lemma 7.1]. This concludes the inductive proof. □

Remark 3.3.3. *Theorem 3.3.1 already allows a general time-dependent initial history φ on $[-\tau,0]$. A further possible extension would be to replace the uniform linear growth and Lipschitz constants in Assumptions (A3) and (A4) by time-dependent functions. This is natural, but it changes the estimates in a non-trivial way: the constants K_j and the Lipschitz bounds for the interval problems* (3.8) *would have to depend on integrals of the time-dependent growth and Lipschitz functions over successive delay intervals. Consequently, the induction over the delay intervals and the later numerical error analysis would require weighted Gronwall-type estimates rather than the uniform bounds used here. For this reason, the thesis keeps time-independent constants in the main analy-*

sis, while treating the time dependence of the equation itself through the Carathéodory and Hölder-in-time assumptions on f.

3.4 Error analysis of the randomised Runge-Kutta scheme

In this section we derive error bounds for the randomised two-stage Runge–Kutta scheme under strengthened assumptions—global Lipschitz in x, Hölder continuity in z and in time, and a Lipschitz history. Concretely, we replace Assumptions (A3)–(A4) by (A3') below.

(A3') there exist $\bar{K} \in [0, +\infty)$ such that for all $t \in [0, (n+1)\tau]$,

$$|f(t, 0, 0)| \leq \bar{K}, \tag{3.16}$$

and there exist $L \in [0, +\infty)$, $\alpha, \gamma \in (0, 1]$ such that:

1. for all $t \in [0, (n+1)\tau]$, $x_1, x_2, z_1, z_2 \in \mathbb{R}^d$,

$$|f(t, x_1, z_1) - f(t, x_2, z_2)| \leq L\Big(|x_1 - x_2| + |z_1 - z_2|^\alpha\Big), \tag{3.17}$$

2. for all $t_1, t_2 \in [0, (n+1)\tau]$, $x, z \in \mathbb{R}^d$,

$$|f(t_1, x, z) - f(t_2, x, z)| \leq L(1 + |x| + |z|)|t_1 - t_2|^\gamma. \tag{3.18}$$

3. for all $s, t \in [-\tau, 0]$,

$$|\varphi(t) - \varphi(s)| \leq L|t - s|, \quad s, t \in [-\tau, 0]. \tag{3.19}$$

Remark 3.4.1. *Note that the assumptions (A1), (A2), (A3') are stronger than the assumptions (A1)-(A4). To see this, note that if f satisfies (A1), (A2), (A3') then we get for all $t \in [0, (n+1)\tau]$ and $x, x_1, x_2, z \in \mathbb{R}^d$ that*

$$|f(t, x, z)| \leq (\bar{K} + L)(1 + |x|)(1 + |z|), \tag{3.20}$$

$$|f(t,x_1,z)-f(t,x_2,z)|\le L|x_1-x_2|, \tag{3.21}$$

and

$$|f(t_1,x_1,z_1)-f(t_2,x_2,z_2)|\le L\big((1+|x_1|+|z_1|)|t_1-t_2|^\gamma+|x_1-x_2|+|z_1-z_2|^\alpha\big), \tag{3.22}$$

since $1+|x|+|z|\le(1+|x|)(1+|z|)$ *and* $|z|^\alpha\le 1+|z|$ *for all* $x,z\in\mathbb{R}^d$. *Hence, the assumptions (A1)-(A4) are satisfied with* $K=\bar{K}+L<\infty$, *and under the assumptions (A1), (A2), (A3') Theorem 3.3.1 holds. The reason for imposing the stronger assumption (A3') in the error analysis is that the proof must compare the drift at different time points, current states, and delayed states. The global Lipschitz bound in the current state gives the stability estimates and the discrete Gronwall step. The Hölder continuity in the delayed variable controls how the numerical error from the previous delay interval is transferred to the next one, which is the origin of the factor* α^j *in Theorem 3.4.3. The Hölder continuity in time is used in the randomised quadrature and local consistency estimates. These assumptions are therefore not needed merely for well-posedness, but for obtaining a quantitative convergence rate. They are satisfied, for example, by coefficients with bounded Hölder time dependence, globally Lipschitz dependence on the current state, and Hölder dependence on the delayed state, such as finite sums of terms of the form* $a(t)+Bx+c\,\psi(z)$ *where* a *is bounded and Hölder continuous and* ψ *is bounded or Hölder continuous of order* α.

We now address the regularity of g_j:

Lemma 3.4.2. *Assume that (A1), (A2), and (A3') hold. For* g_j *defined in* (3.8), *the function* $[0,\tau]\ni t\mapsto g_j(t,\phi_j(t))$ *is bounded in* $L^p([0,\tau])$ *norm for each* $j\in\mathbb{N}$, *and is* β*-Hölder continuous, where* $\beta:=\min\{\gamma,\alpha\}$. *In particular, if the initial condition* $\varphi(t)=x_0$ *for* $t\in[-\tau,0]$, *then* $g_0(\cdot,\phi_0(\cdot))$ *is* γ*-Hölder continuous.*

Proof. Since the function $[0,\tau]\ni t\mapsto g_j(t,\phi_j(t))$ is Borel measurable and by (3.10) in Theorem 3.3.1 and (3.20) we get

$$\|g_j(\cdot,\phi_j(\cdot))\|_{L^p([0,\tau])}\le(\bar{K}+L)(1+K_{j-1})(1+K_j)\tau^{1/p}<+\infty. \tag{3.23}$$

Then, by (3.22) and (3.12) we get

$$\begin{aligned}
&|g_j(s,\phi_j(s)) - g_j(t,\phi_j(t))| \\
\leq& L((1+|\phi_j(s)|+|\phi_{j-1}(s)|)|s-t|^\gamma + |\phi_j(s)-\phi_j(t)| + |\phi_{j-1}(s)-\phi_{j-1}(t)|^\alpha) \\
\leq& L((1+|K_j|+|K_{j-1}|)|s-t|^\gamma + |(1+K_{j-1})(1+K_j)K|t-s| \\
&+ (|(1+K_{j-2})(1+K_{j-1})K|t-s|)^\alpha) \\
\leq& C_{g,j}|s-t|^\beta, \qquad (3.24)
\end{aligned}$$

where $C_{g,j}$ is a generic constant, for any $s,t \in [0,\tau]$ and $j \in \mathbb{N}$. □

The main result of this section is given as follows.

Theorem 3.4.3. *Let $n \in \mathbb{N}_0$, $\tau \in (0,+\infty)$, and let f, φ satisfy the assumptions (A1), (A2), (A3') for some $p \in [2,+\infty)$ and $\alpha,\gamma \in (0,1]$. There exist $C_{p,0}, C_{p,1}, \ldots, C_{p,n} \in (0,+\infty)$ such that for all $N \geq \lceil \tau \rceil$ it holds for $j = 0,1,\ldots,n$,*

$$\Big\| \max_{0\leq i\leq N} |x(t_i^j) - y_i^j| \Big\|_p \leq C_{p,j} h^{\alpha^j \rho}, \qquad (3.25)$$

where $\rho := \frac{1}{2} + \beta$.

Remark 3.4.4 (Discussion of the convergence rate)**.** *The rate in* (3.25) *has two distinct components. The factor $\rho = \frac{1}{2} + \beta$ is the one-step regularity contribution, where $\beta = \min\{\gamma,\alpha\}$ is the effective Hölder regularity of the integrand $t \mapsto g_j(t,\phi_j(t))$ from Lemma 3.4.2. Thus, the rate improves when the time dependence of f and the delayed argument are more regular. The factor α^j reflects propagation through the delay intervals. Indeed, the error on $[j\tau,(j+1)\tau]$ depends on the numerical error from the previous delay interval through the delayed variable, and the assumption* (3.17) *transfers this previous error with Hölder exponent α. Consequently the theoretical order may deteriorate as j increases. In the Lipschitz case $\alpha = 1$, this loss disappears and the same order h^ρ is retained on each fixed delay interval, although the constants $C_{p,j}$ may still depend on j.*

Remark 3.4.5 (Proof roadmap for Theorem 3.4.3)**.** *The proof is organised as an induction over the delay intervals. On the first interval, the delayed value is supplied by the prescribed initial history, so the equation reduces to an ordinary Carathéodory initial-value problem and the randomised quadrature estimate from [60, Theorem 5.2] gives the initial error bound.*

For the induction step, the delayed argument in the implementable scheme is numerical rather than exact. We therefore introduce an auxiliary randomised Runge–Kutta scheme in which the delayed path is still the exact solution. The error is then decomposed into the propagated delay error and the new local discretisation error, and both parts are closed by stability and discrete Gronwall estimates. The Hölder dependence on the delayed variable transfers the previous error with exponent α, which is why the final rate on the jth interval contains the factor α^j.

Proof. We start with $j = 0$ and consider the initial-vale problem (3.13). From Lemma 3.4.2, it has been shown that g_0 is β-Hölder continuous in time. Moreover, by (A1), (A2) and (A3') we have that g_0 is Borel measurable, and for all $t \in [0, \tau]$ and $x, y \in \mathbb{R}^d$

$$|g_0(t, x) - g_0(t, y)| \leq L|x - y|. \tag{3.26}$$

Hence, by Theorem 3.3.1 and using the randomised Runge–Kutta error mechanism recalled in Remark 2.2.2, which is based on the proof of [60, Theorem 5.2], we obtain (3.25), where the constant $C_{p,0}$ is independent of N.

Recall that $\phi_j(t) := x(t + j\tau)$ for $t \in [0, \tau]$ and $t_i^j = j\tau + t_i^0$ with $t_i^0 = ih$. In particular, $\phi_j(t_i^0) = x(t_i^j)$. Let us now assume that there exists $l \in \{0, 1, \ldots, n-1\}$ for which there exists $C_{p,l} \in (0, +\infty)$ such that for all $N \geq \lceil \tau \rceil$

$$\left\| \max_{0 \leq i \leq N} |\phi_l(t_i^0) - y_i^l| \right\|_p \leq C_{p,l} h^{\alpha^l \rho}. \tag{3.27}$$

We consider the following initial-value problem

$$\begin{cases} \phi'_{l+1}(t) = g_{l+1}(t, \phi_{l+1}(t)), \quad t \in [0, \tau], \\ \phi_{l+1}(0) = \phi_l(\tau), \end{cases} \tag{3.28}$$

with

$$g_{l+1}(t,x) = f(t+(l+1)\tau, x, \phi_l(t)). \tag{3.29}$$

From Lemma 3.4.2 we know that $g_{l+1}(\cdot, \phi_{l+1}(\cdot))$ is β-Hölder continuous.

In the proof we use the following auxiliary notations: $\delta_{k+1}^j = h(k+\gamma_{k+1}^j)$ and $h_{k+1}^j = h\gamma_{k+1}^j$. Then δ_{k+1}^j is uniformly distributed in (t_k^0, t_{k+1}^0) and $\theta_{k+1}^j = \delta_{k+1}^j + j\tau$ is uniformly distributed in (t_k^j, t_{k+1}^j). The latter one h_{k+1}^j is a random step size uniformly distributed in $[0,h]$. We define the auxiliary randomised Runge-Kutta scheme as

$$\begin{aligned}
&\bar{y}_0^{l+1} = y_0^{l+1} = y_N^l, \\
&\bar{y}_{k+1}^{l+1} = \bar{y}_k^{l+1} + h g_{l+1}(\delta_{k+1}^{l+1}, \bar{y}_k^{l+1} + h_{k+1}^{l+1} g_{l+1}(t_k^0, \bar{y}_k^{l+1})),\ k = 0,1,\dots,N-1.
\end{aligned} \tag{3.30}$$

To express the auxiliary scheme directly in terms of f we have:

$$\bar{y}_{k+1}^{l+1} = \bar{y}_k^{l+1} + h f(\theta_{k+1}^{l+1}, \bar{y}_k^{l+1} + h_{k+1}^{l+1} f(t_k^{l+1}, \bar{y}_k^{l+1}, \phi_l(t_k^0)), \phi_l(\delta_{k+1}^{l+1})) \tag{3.31}$$

From the definition, it follows that $\bar{y}_k^j$, for $j = 0,1,\dots,n$ and $k = 0,1,\dots,N$, is measurable with respect to the σ-field generated by (3.6), as is y_k^j. Note that this auxiliary scheme serves only as a theoretical construction. Moreover, $\bar{y}_i^{l+1}$ approximates ϕ_{l+1} at t_i^0, although the sequence $\{\bar{y}_i^{l+1}\}_{i=0}^N$ is not implementable in practice, since the delay term involves the exact history $\phi_l(\cdot)$. We use $\bar{y}_i^{l+1}$ solely to estimate the error (3.25) of y_i^{l+1}, since it holds that

$$\begin{aligned}
\Big\| \max_{0\le i\le N} |\phi_{l+1}(t_i^0) - y_i^{l+1}| \Big\|_p \le{}& \Big\| \max_{0\le i\le N} |\phi_{l+1}(t_i^0) - \bar{y}_i^{l+1}| \Big\|_p \\
&+ \Big\| \max_{0\le i\le N} |\bar{y}_i^{l+1} - y_i^{l+1}| \Big\|_p.
\end{aligned} \tag{3.32}$$

Firstly, we estimate $\Big\| \max_{0\le i\le N} |\bar{y}_i^{l+1} - y_i^{l+1}| \Big\|_p$. For $k \in \{1, \dots, N\}$ we get

$$
\begin{aligned}
&\bar{y}_k^{l+1} - y_k^{l+1} \\
&= \sum_{j=1}^{k} (\bar{y}_j^{l+1} - \bar{y}_{j-1}^{l+1}) - \sum_{j=1}^{k} (y_j^{l+1} - y_{j-1}^{l+1}) \\
&= h \sum_{j=1}^{k} \Big(f(\theta_j^{l+1}, \bar{y}_{j-1}^{l+1} + h_j^{l+1} f(t_{j-1}^{l+1}, \bar{y}_{j-1}^{l+1}, \phi_l(t_{j-1}^0)), \phi_l(\delta_j^{l+1})) \\
&\quad - f(\theta_j^{l+1}, y_{j-1}^{l+1} + h_j^{l+1} f(t_{j-1}^{l+1}, y_{j-1}^{l+1}, y_{j-1}^l), y_{j-1}^l + h_j^{l+1} f(t_{j-1}^l, y_{j-1}^l, y_{j-1}^{l-1})) \Big),
\end{aligned}
$$

which, by using (3.17) twice, gives

$$
\begin{aligned}
&|\bar{y}_k^{l+1} - y_k^{l+1}| \\
&\le hL \sum_{j=1}^{k} |\bar{y}_{j-1}^{l+1} + h_j^{l+1} f(t_{j-1}^{l+1}, \bar{y}_{j-1}^{l+1}, \phi_l(t_{j-1}^0)) - y_{j-1}^{l+1} - h_j^{l+1} f(t_{j-1}^{l+1}, y_{j-1}^{l+1}, y_{j-1}^l)| \\
&\quad + hL \sum_{j=1}^{k} |\phi_l(\delta_j^{l+1}) - y_{j-1}^l - h_j^{l+1} f(t_{j-1}^l, y_{j-1}^l, y_{j-1}^{l-1})|^\alpha \\
&\le hL \sum_{j=1}^{k} \left((1 + hL) |\bar{y}_{j-1}^{l+1} - y_{j-1}^{l+1}| + hL |\phi_l(t_{j-1}^0) - y_{j-1}^l|^\alpha \right) \\
&\quad + hL \sum_{j=1}^{k} |\phi_l(\delta_j^{l+1}) - y_{j-1}^l - h_j^{l+1} f(t_{j-1}^l, y_{j-1}^l, y_{j-1}^{l-1})|^\alpha \\
&\le hL(1 + L) \sum_{j=1}^{k} |\bar{y}_{j-1}^{l+1} - y_{j-1}^{l+1}| + h^2 L^2 \sum_{j=1}^{k} |\phi_l(t_{j-1}^0) - y_{j-1}^l|^\alpha \\
&\quad + hL \sum_{j=1}^{k} \Big| \phi_l(t_{j-1}^0) - y_{j-1}^l \\
&\qquad + \int_{t_{j-1}^l}^{t_{j-1}^l + h_j^{l+1}} \left(f(s, \phi_l(s - l\tau), \phi_{l-1}(s - l\tau)) - f(t_{j-1}^l, y_{j-1}^l, y_{j-1}^{l-1}) \right) \mathrm{d}s \Big|^\alpha,
\end{aligned}
$$

where for the last term we use the expression of the initial-value problem (3.28). Note that $|a + b|^\alpha \le |a|^\alpha + |b|^\alpha$ for $\alpha \in (0, 1]$. Now using (3.17), and (3.24) to estimate the last term, it follows that

$$
\begin{aligned}
&\Big|\phi_l(t_{j-1}^0) - y_{j-1}^l + \int_{t_{j-1}^l}^{t_{j-1}^l + h_j^{l+1}} (f(s, \phi_l(s-l\tau), \phi_{l-1}(s-l\tau)) - f(t_{j-1}^l, y_{j-1}^l, y_{j-1}^{l-1}))\mathrm{d}s\Big|^\alpha \\
&\leq |\phi_l(t_{j-1}^0) - y_{j-1}^l|^\alpha + h^\alpha |f(t_{j-1}^l, \phi_l(t_{j-1}^0), \phi_{l-1}(t_{j-1}^0)) - f(t_{j-1}^l, y_{j-1}^l, y_{j-1}^{l-1})|^\alpha \\
&\quad + \Big| \int_{t_{j-1}^l}^{t_{j-1}^l + h_j^{l+1}} (f(s, \phi_l(s-l\tau), \phi_{l-1}(s-l\tau)) - f(t_{j-1}^l, \phi_l(t_{j-1}^0), \phi_{l-1}(t_{j-1}^0))\mathrm{d}s\Big|^\alpha \\
&\leq |\phi_l(t_{j-1}^0) - y_{j-1}^l|^\alpha + L^\alpha h^\alpha (|\phi_l(t_{j-1}^0) - y_{j-1}^l|^\alpha + |\phi_{l-1}(t_{j-1}^0) - y_{j-1}^{l-1}|^{\alpha^2}) \\
&\quad + |C_{g,l} h^{\beta+1}|^\alpha
\end{aligned}
\tag{3.33}
$$

Thus, overall,

$$
\begin{aligned}
&|\bar{y}_k^{l+1} - y_k^{l+1}| \\
&\leq hL(1+L) \sum_{j=1}^k |\bar{y}_{j-1}^{l+1} - y_{j-1}^{l+1}| + Lh(1 + L^\alpha h^\alpha + Lh) \sum_{j=1}^k |\phi_l(t_{j-1}^0) - y_{j-1}^l|^\alpha \\
&\quad + L^{1+\alpha} h^{1+\alpha} \sum_{j=1}^k |\phi_{l-1}(t_{j-1}^0) - y_{j-1}^{l-1}|^{\alpha^2} + hL \sum_{j=1}^k |C_{g,l} h^{\beta+1}|^\alpha \\
&\leq hL(1+L) \sum_{j=1}^k \max_{0 \leq i \leq j-1} |\bar{y}_i^{l+1} - y_i^{l+1}| + L\tau(1+2L) \max_{0 \leq i \leq N} |\phi_l(t_i^0) - y_i^l|^\alpha \\
&\quad + L^{1+\alpha} \tau h^\alpha \max_{0 \leq i \leq N} |\phi_{l-1}(t_i^0) - y_i^{l-1}|^{\alpha^2} + \tau L C_{g,l} h^{\alpha(\beta+1)}.
\end{aligned}
$$

Taking the L_p-norm on both sides gives that

$$
\begin{aligned}
&\mathbb{E}\Big(\max_{0 \leq i \leq k} |\bar{y}_i^{l+1} - y_i^{l+1}|^p\Big) \\
&\leq c_p h^p L^p \mathbb{E}\Big(\sum_{j=1}^k \max_{0 \leq i \leq j-1} |\bar{y}_i^{l+1} - y_i^{l+1}|\Big)^p + c_p L^p \tau^p (1+2L)^p \mathbb{E}\Big[\max_{0 \leq i \leq N} |\phi_l(t_i^0) - y_i^l|^{\alpha p}\Big] \\
&\quad + c_p L^{(1+\alpha)p} \tau^p h^{\alpha p} \mathbb{E}\Big[\max_{0 \leq i \leq N} |\phi_{l-1}(t_i^0) - y_i^{l-1}|^{\alpha^2 p}\Big] + c_p \tau^p L^p C_{g,l}^p h^{\alpha(\beta+1)p}.
\end{aligned}
\tag{3.34}
$$

By Jensen's inequality (see Lemma 2.4.2) and (3.27) we have

$$\mathbb{E}\Big[\max_{0\le i\le N}|\phi_l(t_i^0)-y_i^l|^{\alpha p}\Big]\le\left(\mathbb{E}\Big[\max_{0\le i\le N}|\phi_l(t_i^0)-y_i^l|^{p}\Big]\right)^{\alpha}\le C_{p,l}^{\alpha p}h^{\alpha^{l+1}\rho p},$$

and

$$\mathbb{E}\Big[\max_{0\le i\le N}|\phi_{l-1}(t_i^0)-y_i^{l-1}|^{\alpha^2 p}\Big]\le C_{p,l-1}^{\alpha^2 p}h^{\alpha^{l+1}\rho p}.$$

Note that via the Hölder inequality (see Lemma 2.4.1) we get, for $k=1,2,\dots,N$ and all positive numbers $a_1,a_2,\dots,a_k$, that

$$h^p\Big(\sum_{j=1}^{k}a_j\Big)^p\le\tau^{p-1}h\sum_{j=1}^{k}a_j^p, \tag{3.35}$$

and hence

$$h^p\mathbb{E}\Big[\sum_{j=1}^{k}\max_{0\le i\le j-1}|\bar{y}_i^{l+1}-y_i^{l+1}|\Big]^p\le\tau^{p-1}h\sum_{j=1}^{k}\mathbb{E}\Big[\max_{0\le i\le j-1}|\bar{y}_i^{l+1}-y_i^{l+1}|^p\Big]. \tag{3.36}$$

Finally, substituting all the estimates above into (3.34) gives

$$\begin{aligned}&\mathbb{E}\Big(\max_{0\le i\le k}|\bar{y}_i^{l+1}-y_i^{l+1}|^p\Big)\\&\le c_p\tau^{p-1}L^ph\sum_{j=1}^{k}\mathbb{E}\Big[\max_{0\le i\le j-1}|\bar{y}_i^{l+1}-y_i^{l+1}|^p\Big]+\bar{C}_{\mathrm{Part1},p,l+1}h^{\alpha^{l+1}\rho p}.\end{aligned} \tag{3.37}$$

where $\bar{C}_{\mathrm{Part1},p,l+1}$ is a generic constant, for any $l\in\{0,1,\dots,n-1\},p\in[2,\infty)$.

Now applying Gronwall inequality (see Lemma 2.4.7) to (3.37) gives that

$$\mathbb{E}\Big(\max_{0\le i\le k}|\bar{y}_i^{l+1}-y_i^{l+1}|^p\Big)\le C_{\mathrm{Part1},p,l+1}h^{\alpha^{l+1}\rho p}. \tag{3.38}$$

We now establish an upper bound on $\Big\|\max_{0\le i\le N}|\phi_{l+1}(t_i^0)-\bar{y}_i^{l+1}|\Big\|_p$. For $k\in\{1,2,\dots,N\}$

we have

$$
\begin{aligned}
&\phi_{l+1}(t_k^0) - \bar{y}_k^{l+1} \\
=&\phi_{l+1}(0) - \bar{y}_0^{l+1} + (\phi_{l+1}(t_k^0) - \phi_{l+1}(t_0^0)) - (\bar{y}_k^{l+1} - \bar{y}_0^{l+1}) \\
=&(\phi_l(t_N^0) - y_N^l) + \sum_{j=1}^{k}(\phi_{l+1}(t_j^0) - \phi_{l+1}(t_{j-1}^0)) - \sum_{j=1}^{k}(\bar{y}_j^{l+1} - \bar{y}_{j-1}^{l+1}) \\
=&(\phi_l(t_N^0) - y_N^l) + \sum_{j=1}^{k}\Big(\int_{t_{j-1}^0}^{t_j^0} g_{l+1}(s, \phi_{l+1}(s))\mathrm{d}s \\
&\qquad - h g_{l+1}(\delta_j^{l+1}, \bar{y}_{j-1}^{l+1} + h_j^{l+1} g_{l+1}(t_{j-1}^0, \bar{y}_{j-1}^{l+1}))\Big) \\
=&(\phi_l(t_N^0) - y_N^l) + S_{1,l+1}^k + S_{2,l+1}^k + S_{3,l+1}^k,
\end{aligned}
\tag{3.39}
$$

where

$$
\begin{aligned}
S_{1,l+1}^k &:= \sum_{j=1}^{k}\left(\int_{t_{j-1}^0}^{t_j^0} g_{l+1}(s, \phi_{l+1}(s))\mathrm{d}s - h g_{l+1}(\delta_j^{l+1}, \phi_{l+1}(\delta_j^{l+1}))\right), \\
S_{2,l+1}^k &:= h\sum_{j=1}^{k}\Big(g_{l+1}(\delta_j^{l+1}, \phi_{l+1}(\delta_j^{l+1})) \\
&\qquad - g_{l+1}(\delta_j^{l+1}, \phi_{l+1}(t_{j-1}^0) + h_j^{l+1} g_{l+1}(t_{j-1}^0, \phi_{l+1}(t_{j-1}^0)))\Big), \\
S_{3,l+1}^k &:= h\sum_{j=1}^{k}\Big(g_{l+1}(\delta_j^{l+1}, \phi_{l+1}(t_{j-1}^0) + h_j^{l+1} g_{l+1}(t_{j-1}^0, \phi_{l+1}(t_{j-1}^0))) \\
&\qquad - g_{l+1}(\delta_j^{l+1}, \bar{y}_{j-1}^{l+1} + h_j^{l+1} g_{l+1}(t_{j-1}^0, \bar{y}_{j-1}^{l+1}))\Big).
\end{aligned}
$$

Since $g_{l+1}(\cdot, \phi_{l+1}(\cdot))$ is β-Hölder continuous from Lemma 3.4.2 and by (3.23) and (3.24) we get by Theorem 3.1 in [60] that

$$
\Big\| \max_{1\le k\le N} |S_{1,l+1}^k| \Big\|_p \le c_p\sqrt{\tau}\, \|g_{l+1}(\cdot, \phi_{l+1}(\cdot))\|_{C^\beta([0,\tau])} h^\rho =: C_{S_1,l+1} h^\rho. \tag{3.40}
$$

Then, by inequity (3.21) and (3.24) we get

$$\begin{aligned}
|S^k_{2,l+1}| &\le Lh\sum_{j=1}^{k}\Big|\phi_{l+1}(\delta_j^{l+1}) - \phi_{l+1}(t^0_{j-1}) - h_j^{l+1}g_{l+1}(t^0_{j-1},\phi_{l+1}(t^0_{j-1}))\Big| \\
&\le Lh\sum_{j=1}^{k}\int_{t^0_{j-1}}^{t^0_{j-1}+h_j^{l+1}}\Big|g_{l+1}(s,\phi_{l+1}(s)) - g_{l+1}(t^0_{j-1},\phi_{l+1}(t^0_{j-1}))\Big|\mathrm{d}s \\
&\le Lh\sum_{j=1}^{k}\int_{t^0_{j-1}}^{t^0_{j-1}+h_j^{l+1}}\Big|C_{g,l+1}|s-t^0_{j-1}|^{\beta}\Big|\mathrm{d}s \\
&\le C_{g,l+1}Lh\sum_{j=1}^{k}\beta^{-1}(h_j^{l+1})^{\beta+1} \\
&\le C_{g,l+1}\beta^{-1}L\tau h^{\beta+1} \\
&=: C_{S_2,l+1}h^{\beta+1}. \qquad (3.41)
\end{aligned}$$

Moreover, for the last term $S^k_{3,l+1}$, by using (3.21) twice, we get

$$\begin{aligned}
S^k_{3,l+1} &\le hL\sum_{j=1}^{k}\Big(\phi_{l+1}(t^0_{j-1}) + h_j^{l+1}g_{l+1}(t^0_{j-1},\phi_{l+1}(t^0_{j-1})) - \bar{y}^{l+1}_{j-1} - h_j^{l+1}g_{l+1}(t^0_{j-1},\bar{y}^{l+1}_{j-1})\Big) \\
&\le hL\sum_{j=1}^{k}\Big(\phi_{l+1}(t^0_{j-1}) - \bar{y}^{l+1}_{j-1} + h_j^{l+1}L(\phi_{l+1}(t^0_{j-1}) - \bar{y}^{l+1}_{j-1})\Big) \\
&\le hL\sum_{j=1}^{k}\Big((h_j^{l+1}L+1)(\phi_{l+1}(t^0_{j-1}) - \bar{y}^{l+1}_{j-1})\Big) \\
&\le C_{S_3,l+1}h\sum_{j=1}^{k}\Big((\phi_{l+1}(t^0_{j-1}) - \bar{y}^{l+1}_{j-1})\Big). \qquad (3.42)
\end{aligned}$$

Therefore, by (3.27), (3.40), (3.41), (3.42), and (3.35), for all $k \in \{1,2,\dots,N\}$ the

following inequality holds

$$\begin{aligned}
&\mathbb{E}\Big[\max_{0\le i\le k}|\phi_{l+1}(t_i^0)-\bar{y}_i^{l+1}|^p\Big] && (3.43)\\
\le&\mathbb{E}\Big[\max_{0\le i\le k}|(\phi_l(t_N^0)-y_N^l)+S_{1,l+1}^k+S_{2,l+1}^k+S_{3,l+1}^k|^p]\\
\le&c_p\Big(\mathbb{E}|\phi_l(t_N^0)-y_N^l|^p+\mathbb{E}\Big[\max_{1\le k\le N}|S_{1,l+1}^k|^p\Big]+\mathbb{E}\Big[\max_{1\le k\le N}|S_{2,l+1}^k|^p\Big]\Big)\\
&+c_p\mathbb{E}\Big[\max_{1\le k\le N}|S_{3,l+1}^k|^p\Big]\\
\le&c_p\Big(C_{p,l}^p h^{\alpha^l\rho p}+C_{S_1,l+1}^p h^{p\rho}+C_{S_2,l+1}^p h^{p(\beta+1)}\Big)\\
&+c_p\mathbb{E}\Big[(C_{S_3,l+1}h\sum_{j=1}^N|\phi_{l+1}(t_{j-1}^0)-\bar{y}_{j-1}^{l+1}|)^p\Big] && (3.44)\\
\le&\bar{C}_{\mathrm{Part2},p,l+1}h^{\alpha^l\rho p}+c_pC_{S_3,l+1}^p\tau^{p-1}h\sum_{j=1}^N\mathbb{E}\Big[|\phi_{l+1}(t_{j-1}^0)-\bar{y}_{j-1}^{l+1}|^p\Big]. && (3.45)
\end{aligned}$$

where $\bar{C}_{\mathrm{Part2},p,l+1}$ is a generic constant, for any $l\in\{0,1,\dots,n-1\}, p\in[2,\infty)$.

By using Gronwall's Lemma 2.4.7, we get for all $k\in\{1,2,\dots,N\}$

$$\begin{aligned}
\mathbb{E}\Big[\max_{0\le i\le k}|\phi_{l+1}(t_i^0)-\bar{y}_i^{l+1}|^p\Big]&\le\bar{C}_{\mathrm{Part2},p,l+1}h^{\alpha^l\rho p}\exp(c_pL^p(L+1)^p\tau^{p-1}hN)\\
&\le\bar{C}_{\mathrm{Part2},p,l+1}h^{\alpha^l\rho p}\exp(c_pL^p(L+1)^p\tau^p),
\end{aligned}$$

which gives

$$\Big\|\max_{0\le i\le N}|\phi_{l+1}(t_i^0)-\bar{y}_i^{l+1}|\Big\|_p\le C_{\mathrm{Part2},p,l+1}h^{\alpha^l\rho}. \tag{3.46}$$

Combining (3.32), (3.38), and (3.46) we finally obtain

$$\Big\|\max_{0\le i\le N}|\phi_{l+1}(t_i^0)-y_i^{l+1}|\Big\|_p\le C_{p,l+1}h^{\alpha^{l+1}\rho}, \tag{3.47}$$

which ends the inductive part of the proof. Finally, $\phi_{l+1}(t_i^0)=\phi_{l+1}(ih)=x(ih+(l+1)\tau)=x(t_i^{l+1})$ and the proof of (3.25) is finished. □

3.5 Numerical experiments

3.5.1 The implementation

The implementation of the randomised Runge-Kutta scheme is not straightforward. To evaluate the solution at time point $t_i^j = ih + j\tau$ within the interval $[j\tau, (j+1)\tau]$ for $j \geq 1$, we need four steps:

Step (j1): First simulate $\gamma \sim \mathcal{U}(0,1)$ and set a random step size $\gamma h \in [0, h]$ and a random time point $t_{i-1}^j + \gamma h \in [t_{i-1}^j, t_i^j]$.

Step (j2): Following the second line of (3.5), compute the intermediate delay term $\tilde{y}_i^{j-1,j}$ via the random step size, the time grid point and evaluations from the preceding τ-intervals, namely, t_{i-1}^{j-1} and y_{i-1}^{j-1} over $[(j-1)\tau, j\tau]$, and y_{i-1}^{j-2} within $[(j-2)\tau, (j-1)\tau]$.

Step (j3): Following the third line of (3.5), compute the intermediate term $\tilde{y}_i^j$ via the random step size, the time grid point and previous evaluations from the preceding or the current τ-interval, namely, t_{i-1}^j and y_{i-1}^j within $[j\tau, (j+1)\tau]$, and y_{i-1}^{j-1} over $[(j-1)\tau, j\tau]$.

Step (j4): Following the last line of (3.5), compute evaluation y_i^j via the random time point, the preceding evaluation y_{i-1}^j, the intermediate term $\tilde{y}_i^j$ and the intermediate delay term $\tilde{y}_i^{j-1,j}$ from **Step (j1)-Step (j3)**.

Equivalently, for $j \geq 1$ these four steps are given by the following update formulas:

$$\begin{aligned} h_i^j &= \gamma_i^j h, \qquad \theta_i^j = t_{i-1}^j + h_i^j, \\ \tilde{y}_i^{j-1,j} &= y_{i-1}^{j-1} + h_i^j f(t_{i-1}^{j-1}, y_{i-1}^{j-1}, y_{i-1}^{j-2}), \\ \tilde{y}_i^j &= y_{i-1}^j + h_i^j f(t_{i-1}^j, y_{i-1}^j, y_{i-1}^{j-1}), \\ y_i^j &= y_{i-1}^j + h f(\theta_i^j, \tilde{y}_i^j, \tilde{y}_i^{j-1,j}). \end{aligned} \tag{3.48}$$

For $j = 0$, the same structure is used, except that the delay value is supplied by the

initial history φ:

$$\begin{aligned} h_i^0 &= \gamma_i^0 h, \qquad \theta_i^0 = t_{i-1}^0 + h_i^0, \\ \tilde{y}_i^{(-1,0)} &= \varphi(t_{i-1}^{(-1)} + h_i^0), \\ \tilde{y}_i^0 &= y_{i-1}^0 + h_i^0 f(t_{i-1}^0, y_{i-1}^0, y_{i-1}^{(-1)}), \\ y_i^0 &= y_{i-1}^0 + h f(\theta_i^0, \tilde{y}_i^0, \tilde{y}_i^{(-1,0)}). \end{aligned} \tag{3.49}$$

Listing 3.1: A sample implementation of (3.4) and (3.5) in PYTHON

```python
import numpy as np

def rand_rrk_dde(tau, phi, h, f, n_taus):
    N = int(round(tau / h))
    y = np.zeros((N + 1, n_taus + 1))

    delay_grid = -tau + h * np.arange(N + 1)
    y[:, 0] = np.array([phi(t) for t in delay_grid])

    for j in range(1, n_taus + 1):
        y[0, j] = y[N, j - 1]
        for i in range(1, N + 1):
            t = (j - 1) * tau + (i - 1) * h
            gamma = np.random.rand()
            h_gamma = h * gamma
            theta = t + h_gamma

            if j == 1:
                z_left = phi(t - tau)
                z_stage = phi(theta - tau)
            else:
                z_left = y[i - 1, j - 1]
                z_prev = y[i - 1, j - 2]
                z_stage = z_left + h_gamma * f(t - tau, z_left, z_prev)

            x_left = y[i - 1, j]
            x_stage = x_left + h_gamma * f(t, x_left, z_left)
            y[i, j] = x_left + h * f(theta, x_stage, z_stage)

    return y
```

Remark 3.5.1 (Relation of the DDE experiments to Theorem 3.4.3)**.** *The two DDE examples below play different roles. Example 1 contains jump discontinuities in the time-dependent factor $g(t)$, and hence the Hölder-in-time part of Assumption (A3') is not satisfied. Figure 3.3 is therefore used as a robustness and comparison test for a*

time-irregular DDE, rather than as a direct verification of Theorem 3.4.3. Example 2, by contrast, is designed so that the drift and the initial history satisfy the assumptions of Theorem 3.4.3; Figure 3.4 and Table 3.1 are the theorem-backed numerical tests illustrating the dependence of the rate on α, γ and the delay interval.

3.5.2 Example 1

We modify [63, Example 6.2] as follows:

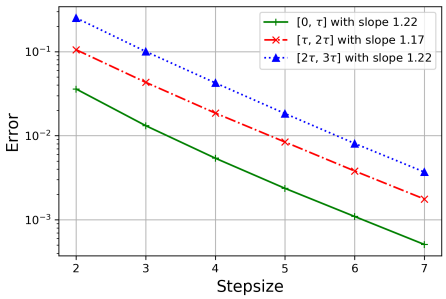
$$\begin{cases} \dot{u}(t) &= g(t)\big(u(t) + (1 + |u(t-\tau)|)^{\alpha}\big), \quad t \in [0, 3\tau], \\ u(t) &= 1, \quad t \in [-\tau, 0], \end{cases} \tag{3.50}$$

where $g(t) := \Big[-\frac{1}{10}\mathrm{sgn}(\frac{1}{4}T - t) - \frac{1}{5}\mathrm{sgn}(\frac{1}{2}T - t) - \frac{7}{10}\mathrm{sgn}(\frac{3}{4}T - t)\Big]$ and

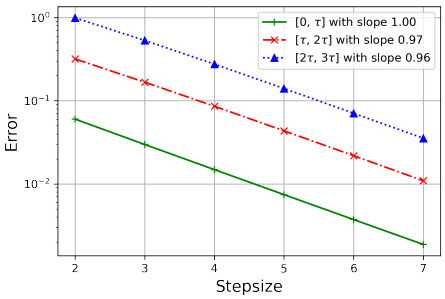
$$\mathrm{sgn}(t) := \begin{cases} -1, & \text{if } t < 0, \\ 0, & \text{if } t = 0, \\ 1, & \text{if } t > 0. \end{cases} \tag{3.51}$$

Here we have three jump points at $t = \frac{1}{4}T$, $t = \frac{1}{2}T$ and $t = \frac{3}{4}T$. Note that when $\alpha = 0$, our proposed scheme can recover the performance shown in [63, Example 6.2], with an order of convergence roughly 1.5.

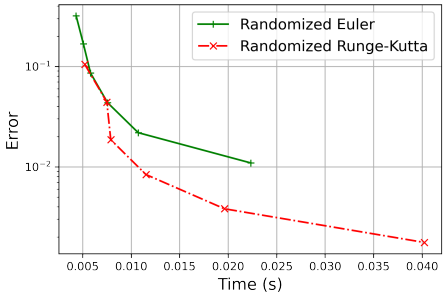
In this example, we choose $\alpha = 0.5$ and investigate the performance of our proposed randomised Runge-Kutta scheme for solving such a time-irregular DDE. We compare it with the randomised Euler scheme proposed in [28] and recalled in Section 3.2, which is designed for Carathéodory DDEs, in terms of convergence rate, accuracy, and computational efficiency. The reference solution is computed through the randomised Runge-Kutta scheme at step size $h_{\mathrm{ref}} = 2^{-15}$. We test the performance via 1000 experiments for each $h = 2^{-l}$, $l = 2, \ldots, 7$, and record the mean-square error (MSE) and time cost against MSE for both methods respectively in Figure 3.3. It can be observed from Figure 3.3, and in particular from Figure 3.3a and Figure 3.3b, that the randomised Runge-Kutta scheme outperforms consistently over all intervals in terms of



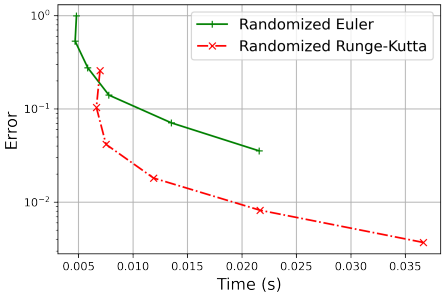
(a) MSE of randomised Runge-Kutta.



(b) MSE of randomised Euler.



(c) Time cost against MSE over $[\tau, 2\tau]$.



(d) Time cost against MSE over $[2\tau, 3\tau]$.

Figure 3.3: The performance of solving DDE (3.50) via randomised Runge-Kutta and randomised Euler.

accuracy and the order of convergence. Clearly, due to the additional computation of $\tilde{y}_{k+1}^{j}$ and $\tilde{y}_{k+1}^{j-1,j}$ in (3.5), the randomised Runge-Kutta scheme is approximately three times as expensive as the randomised Euler scheme with the same step size. But due to its better accuracy, the randomised Runge-Kutta scheme is superior for all the step sizes smaller than 2^{-3}.

3.5.3 Example 2

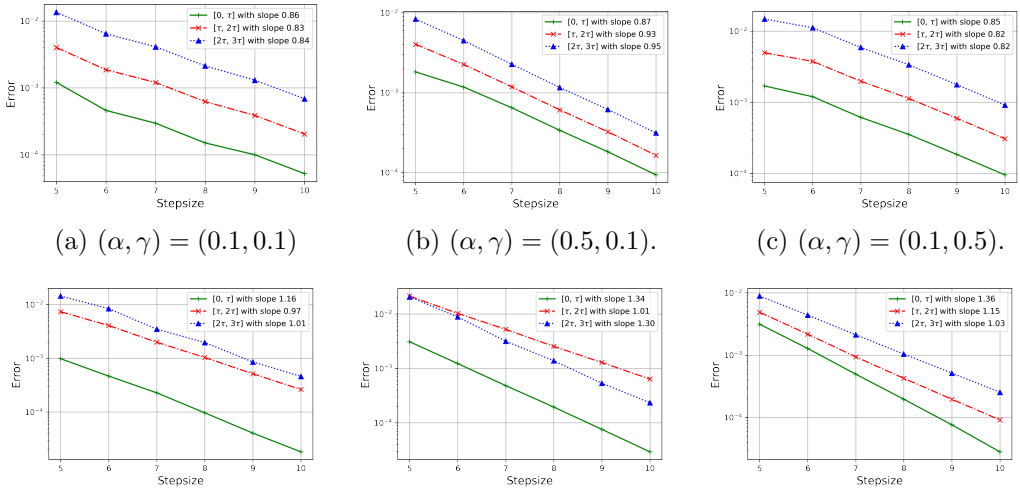
To verify (3.25) in Theorem 3.4.3, we test the sensitivity of the performance of the randomised Runge-Kutta scheme with respect to parameters $\alpha, \gamma \in (0, 1]$ on the following DDE:

$$\begin{cases} \dot{u}(t) &= u(t) - |u(t-\tau)|^{\alpha} + |t|^{\gamma}, \quad t \in [0, 3\tau], \\ u(t) &= t + \tau, \quad t \in [-\tau, 0]. \end{cases} \tag{3.52}$$

Clearly DDE (3.52) satisfies all the assumptions of Theorem 3.4.3. We choose the combinations

$$(\alpha, \gamma) \in \{(0.1, 0.1), (0.1, 0.5), (0.5, 0.1), (0.5, 0.5), (0.5, 1), (1, 0.5)\}$$

For each pair, the reference solution is computed through the randomised Runge-Kutta scheme at step size $h_{\text{ref}} = 2^{-16}$. We test the performance via 1000 experiments for each $h = 2^{-l}$, $l = 5, \ldots, 10$, and record the mean-square error (MSE) and the negative MSE slope in Figure 3.4 and Table 3.1 respectively. One can observe from Table 3.1 that for each pair of (α, γ), the order of convergence over $[j\tau, (j+1)\tau]$ is consistently higher than the theoretical one for $j = 0, 1, 2$, where the latter one is $\left(\frac{1}{2} + (\gamma \wedge \alpha)\right)\alpha^j$ in (3.25). In particular, Figure 3.4a, Figure 3.4b and Figure 3.4c show that, although the observed orders from the three cases are similar over $[0, \tau]$, those over $[\tau, 2\tau]$ and $[2\tau, 3\tau]$ in Figure 3.4b are slightly higher than the other two, due to the larger value of α. This behaviour is consistent with Theorem 3.4.3.



(a) $(\alpha, \gamma) = (0.1, 0.1)$ (b) $(\alpha, \gamma) = (0.5, 0.1)$. (c) $(\alpha, \gamma) = (0.1, 0.5)$.

(d) $(\alpha, \gamma) = (0.5, 0.5)$. (e) $(\alpha, \gamma) = (0.5, 1)$. (f) $(\alpha, \gamma) = (1, 0.5)$.

Figure 3.4: The performance of solving DDE (3.52) via randomised Runge-Kutta.

Table 3.1: Performance of solving DDE (3.52) using randomised Runge-Kutta: negative MSE slopes for varying α and γ.

α	γ	$[0, \tau]$	$[1\tau, 2\tau]$	$[2\tau, 3\tau]$	Figure
0.1	0.1	0.86	0.83	0.84	Figure 3.4a
0.5	0.1	0.87	0.93	0.95	Figure 3.4b
0.1	0.5	0.85	0.82	0.82	Figure 3.4c
0.5	0.5	1.16	0.97	1.01	Figure 3.4d
0.5	1	1.34	1.01	1.30	Figure 3.4e
1	0.5	1.36	1.15	1.03	Figure 3.4f

Chapter 4

On approximation of solutions of stochastic delay differential equations via randomised Euler scheme

4.1 Introduction

In this chapter, we investigate the efficiency of randomised numerical scheme for simulating the stochastic delay differential equations (SDDEs) by considering the SDDEs of the following form

$$\begin{cases} \mathrm{d}X(t) = f(t, X(t), X(t-\tau))\,\mathrm{d}t + g(t, X(t), X(t-\tau))\,\mathrm{d}W(t), & t \in [0, (n+1)\tau], \\ X(t) = x_0, & t \in [-\tau, 0], \end{cases} \tag{4.1}$$

with the constant time-lag $\tau \in (0, +\infty)$, fixed time horizon $n \in \mathbb{N}$, $f : [0, (n+1)\tau] \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, $g : [0, (n+1)\tau] \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times m}$, and $x_0 \in \mathbb{R}^d$. Here, we consider the drift coefficient f to satisfy Carathéodory type conditions, which means that we assume drift coefficient $f = f(t, x, z)$ is Borel measurable with respect to t, and (at least) continuous with respect to (x, z). For the diffusion coefficient $g(t, x, z)$ we assume (at

least) continuity with respect to all variables (t, x, z).

Though SDEs can be regarded as a special case of SDDEs, the extension of the *randomised* numerical approaches from SDEs to SDDEs is non-trivial. On the one hand, compared to SDEs, the existence, uniqueness and L^p-Hölder regularity of the strong solution to (4.1) is currently not known in the literature under conditions for f and g considered in this chapter. On the other hand, the time-delays may induce instabilities in the basic SDDEs [69], and the presence of time-delays may further influence the convergence speed [14].

In this chapter, we resolve these challenges one by one. For the study of the exact solution to (4.1), instead of considering the SDDE over the entire interval all at once, we follow a different way, known for DDEs form in [27]. The entire time interval will be divided into multiple subintervals of the length τ and the solution of the SDDE will be considered at each subinterval separately. This will allow the SDDE to be converted into a sequence of iterative SDEs with random coefficients and analysed by induction, where the delay term is treated as a random resource that has been given (see (4.12)). The randomised Euler-Maruyama scheme is defined in the same manner, i.e., iteratively. We keep the same grid for all the subintervals of length τ so that the simulation obtained from the preceding subinterval will directly be the delay input for the current subinterval. To assist the error analysis, we introduce the auxiliary randomised Euler-Maruyama scheme. In the case when f or g are Hölder continuous with respect to z we observe that the numerical error accumulates over the subintervals. In particular, this may suggest that this analysis is not valid for the infinite time horizon cases.

To summarise, the main contributions of the study are as follows:

- We show existence, uniqueness, and Hölder regularity of the strong solution to (4.1) when both the drift $f(t, x, z)$ and diffusion $g(t, x, z)$ coefficients are Borel measurable with respect to t, satisfy a global Lipschitz condition with respect to x and global linear growth condition with respect to (x, z), but are continuous with respect to the delay variable z (see Theorem 4.3.1).

- We perform rigorous error analysis of the randomised Euler scheme applied to

(4.1) when both drift coefficient $f(t,x,z)$ and diffusion coefficient $g(t,x,z)$ satisfy a global Lipschitz condition with respect to x and a global Hölder condition with respect to z. Still, we assume that f is only Borel measurable with respect to t, but for g we assume now that it is also Hölder continuous with respect to the time variable t. (See Theorem 4.4.4).

- We present an implementation of the randomised Euler-Maruyama scheme in Python code and report results of numerical experiments that show stable error behaviour as stated in Theorem 4.4.4.

The structure of this chapter is as follows. Basic notions, definitions together with assumptions and the construction of the randomised Euler-Maruyama scheme are given in Section 4.2. Section 4.3 is devoted to the issue of existence and uniqueness of strong solutions of the Carathéodory type SDDEs (4.1). Section 4.4 contains the proof of the main error result for the randomised Euler-Maruyama scheme. In Section 4.5 we report results of numerical experiments with an exemplary Python implementation.

4.2 Assumptions and Algorithm

We consider a complete probability space $(\Omega, \Sigma, \mathbb{P})$. Recall from Chapter 2 that for a random variable $X : \Omega \to \mathbb{R}^d$ we write $\|X\|_{L^p(\Omega)}$ for its L^p-norm, where $p \in [2, +\infty)$; throughout this chapter, c_p denotes a generic constant that depends on p. We denote by $(\Sigma_t)_{t\geq 0}$ a filtration, satisfying the usual conditions. Let $W = (W(t))_{t\geq 0}$ is m-dimensional standard Wiener process on $(\Omega, \Sigma, \mathbb{P})$ with respect to $(\Sigma_t)_{t\geq 0}$. Let $\Sigma_\infty = \sigma\Big(\bigcup_{t\geq 0} \Sigma_t\Big)$. For two sub-$\sigma$-fields $\mathcal{A}$, $\mathcal{B}$ of Σ we denote by $\mathcal{A} \vee \mathcal{B} = \sigma(\mathcal{A} \cup \mathcal{B})$. Let $Y : [0,T] \times \Omega \to \mathbb{R}^d$ be a stochastic process and $\xi : \Omega \to [0,T]$ a random variable.

Let us fix the *horizon parameter* $n \in \mathbb{N}$. On the drift coefficient $f : [0,(n+1)\tau] \times \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$ we impose the following assumptions:

(A1) $f(t,\cdot,\cdot) \in C(\mathbb{R}^d \times \mathbb{R}^d; \mathbb{R}^d)$ for all $t \in [0,(n+1)\tau]$,

(A2) $f(\cdot,x,z) : [0,(n+1)\tau] \to \mathbb{R}^d$ is Borel measurable for all $(x,z) \in \mathbb{R}^d \times \mathbb{R}^d$,

Symbol	Meaning
γ_k^j	uniformly distributed sample from $[0,1]$ for $j \in \mathbb{N}_0$ and $k \in \mathbb{N}$
t_k^j	$t_k^j = j\tau + kh$ for $k \in [N]_0$ and $j \in [n]_0$
θ_{k+1}^j	$\theta_{k+1}^j = t_k^j + h\gamma_{k+1}^j$
δ_{k+1}^j	$\delta_{k+1}^j = kh + h\gamma_{k+1}^j$

Table 4.1: Notations.

(A3) there exist $K_f \in (0,\infty)$ such that for all $t \in [0,(n+1)\tau]$, $x, x_1, x_2, z \in \mathbb{R}^d$

$$\begin{aligned} |f(t,x,z)| &\le K_f(1+|x|+|z|), \\ |f(t,x_1,z) - f(t,x_2,z)| &\le K_f|x_1 - x_2|. \end{aligned} \tag{4.2}$$

For the diffusion coefficient $g : [0,(n+1)\tau] \times \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^{d\times m}$ we impose the following assumptions:

(B1) $g(t,\cdot,\cdot) \in C(\mathbb{R}^d \times \mathbb{R}^d; \mathbb{R}^{d\times m})$ for all $t \in [0,(n+1)\tau]$,

(B2) $g(\cdot,x,z) : [0,(n+1)\tau] \to \mathbb{R}^{d\times m}$ is Borel measurable for all $(x,z) \in \mathbb{R}^d \times \mathbb{R}^d$,

(B3) there exists $K_g \in (0,\infty)$ such that for all $t \in [0,(n+1)\tau]$, $x, x_1, x_2, z \in \mathbb{R}^d$

$$\begin{aligned} |g(t,x,z)| &\le K_g(1+|x|+|z|), \\ |g(t,x_1,z) - g(t,x_2,z)| &\le K_g|x_1 - x_2|. \end{aligned} \tag{4.3}$$

In Section 4.3 we show that, under the assumptions (A1)-(A3), (B1)-(B3), the SDDE (4.1) has an unique strong solution. Next, in Section 4.4 we investigate error of the *randomised Euler scheme* under slightly stronger assumptions than (A1)-(A3), (B1)-(B3). The scheme defined below is the SDDE instance of the hierarchy described in Section 2.3: the drift is evaluated at a random time point, while the diffusion coefficient is kept at the left endpoint so that the Brownian increment remains adapted. The aforementioned randomised Euler scheme is defined as follows. Fix the *discretization parameter* $N \in \mathbb{N}$ and set

$$t_k^j = j\tau + kh, \quad k \in [N]_0,\ j \in [n]_0, \tag{4.4}$$

where

$$h = \frac{\tau}{N}. \tag{4.5}$$

Note that for each j the sequence $\{t_k^j\}_{k=0}^N$ provides uniform discretization of the subinterval $[j\tau, (j+1)\tau]$. Let $\{\gamma_k^j\}_{j\in\mathbb{N}_0, k\in\mathbb{N}}$ be an i.i.d. sequence of random variables, defined on the complete probability space $(\Omega, \Sigma, \mathbb{P})$, where every γ_k^j is uniformly distributed on $[0,1]$. We assume that the σ-fields $\sigma(\{\gamma_k^j\}_{j\in\mathbb{N}_0,k\in\mathbb{N}})$ and Σ_∞ are independent. Then $W = (W(t))_{t\geq 0}$ is also the Wiener process with respect to the extended filtration

$$\tilde{\Sigma}_t = \Sigma_t \vee \sigma(\{\gamma_k^j\}_{j\in\mathbb{N}_0,k\in\mathbb{N}}), \quad t \geq 0. \tag{4.6}$$

Here, as in Chapter 3, the superscript (-1) labels the prescribed initial-history interval $[-\tau, 0]$ rather than a further computed delay interval. We define randomised Euler scheme as setting $y_0^{(-1)} = \ldots = y_N^{(-1)} = x_0$ and then for $j \in [n]_0$, $k \in [N-1]_0$ taking

$$y_0^j = y_N^{j-1}, \tag{4.7}$$

$$y_{k+1}^j = y_k^j + h \cdot f(\theta_{k+1}^j, y_k^j, y_k^{j-1}) + g(t_k^j, y_k^j, y_k^{j-1})(W(t_{k+1}^j) - W(t_k^j)), \tag{4.8}$$

where $\theta_{k+1}^j = t_k^j + h\gamma_{k+1}^j$. As the output we obtain the sequence of $\mathbb{R}^d$-valued random vectors $\{y_k^j\}_{k\in[N]_0, j\in[n]_0}$ that provides a discrete approximation of the values $\{X(t_k^j)\}_{k\in[N]_0,j\in[n]_0}$. By induction we get that for all $j \in [n]_0$, $k \in [N]_0$

$$\begin{aligned}\sigma(\{y_k^j\}) \subset &\sigma\Big(\{\gamma_1^0 \ldots, \gamma_N^0, \ldots, \gamma_1^{j-1}, \ldots, \gamma_N^{j-1}, \gamma_1^j, \ldots, \gamma_k^j\}\Big) \\ &\vee \sigma\Big(\{W(t_0^0), \ldots, W(t_N^0), \ldots, W(t_0^{j-1}), \\ &\qquad \ldots, W(t_N^{j-1}), W(t_0^j), \ldots, W(t_k^j)\}\Big),\end{aligned} \tag{4.9}$$

as the horizon parameter n is fixed, the randomised Euler scheme uses $O(N)$ evaluations of f, g and W (with a constant in the 'O' notation that depends on n but not on N).

The aim is to establish upper bounds on the $L^p(\Omega)$-error

$$\Big\| \max_{0\leq k\leq N} |X(t_k^j) - y_k^j| \Big\|_{L^p(\Omega)} \tag{4.10}$$

for all $j = 0, 1, \ldots, n$.

4.3 Properties of solutions to Carathéodory SDDEs

We take into account the semi-flow property holds for SDDE (4.1). Namely, we define

$$\phi_l(t) := X(t + l\tau), \quad t \in [0, \tau], \quad l = -1, 0, 1, \ldots, n \tag{4.11}$$

and, in particular, $\phi_{-1}(t) = x_0$ for $t \in [0, \tau]$. Using change of variable formula for Lebesgue and Itô integrals we arrive at the following SDE with random coefficients

$$\begin{cases} \mathrm{d}\phi_l(t) = f_l\big(t, \phi_l(t)\big)\,\mathrm{d}t + g_l\big(t, \phi_l(t)\big)\,\mathrm{d}W_l(t), \quad t \in [0, \tau], \\ \phi_l(0) = \phi_{l-1}(\tau), \end{cases} \tag{4.12}$$

where the random fields $f_l : [0, \tau] \times \mathbb{R}^d \to \mathbb{R}^d$ and $g_l : [0, \tau] \times \mathbb{R}^d \to \mathbb{R}^{d\times m}$ are defined follows

$$\begin{aligned} f_l(t, x) &:= f\big(t + l\tau, x, \phi_{l-1}(t)\big), \\ g_l(t, x) &:= g\big(t + l\tau, x, \phi_{l-1}(t)\big), \end{aligned} \tag{4.13}$$

and $W_l(t) := W(t + l\tau)$, $t \in [0, \tau]$, $x \in \mathbb{R}^d$. We also set

$$\Sigma_t^{(-1)} := \{\emptyset, \Omega\}, \tag{4.14}$$

$$\Sigma_t^j := \Sigma_{t+j\tau}, \tag{4.15}$$

for $t \in [0, \tau]$, $j \in [n]_0$. The solution X of (4.1) can be written as

$$X(t) = \sum_{j=-1}^{n} \phi_j(t - j\tau) \cdot \mathbf{1}_{[j\tau,(j+1)\tau]}(t), \quad t \in [-\tau, (n + 1)\tau]. \tag{4.16}$$

We state L^p-boundedness and regularity of the solution X in the following Theorem 4.3.1. Theorem 3.1 on pages 156–157 of [68] gives an existence and uniqueness result for stochastic delay differential equations under classical Lipschitz and growth assump-

tions. Our use of this result here is therefore not to claim a new existence theorem in that classical setting, but to adapt the interval-by-interval construction to the present formulation and, more importantly, to record the explicit L^p estimates and L^p-Hölder regularity of the solution that are needed later in the numerical error analysis. These quantitative estimates are not stated in Theorem 3.1 of [68].

Theorem 4.3.1. *Let $n \in \mathbb{N}_0$, $\tau \in (0, +\infty)$, $x_0 \in \mathbb{R}^d$ and let f, g satisfy the assumptions (A1)-(A3), (B1)-(B3). Then there exists a unique strong solution $X = X(x_0, f, g)$ to (4.1) such that for $j \in [n]_0$ we have*

$$\mathbb{E}\Big(\sup_{0\le t\le\tau} |\phi_j(t)|^p\Big) \le K_{j,p}, \tag{4.17}$$

where $K_{-1,p} := |x_0|^p$, $K = \max\{K_f, K_g\}$,

$$K_{j,p} = c_p\Big(K_{j-1,p} + \tau^{p/2}K^p(\tau^{p/2}+1)(1+K_{j-1,p})\Big)\exp\Big(c_p\tau^pK^p(1+K^p)\Big), \tag{4.18}$$

and for all $j \in [n]_0$, $t, s \in [0, \tau]$ it holds

$$\|\phi_j(t) - \phi_j(s)\|_{L^p(\Omega)} \le c_pK(\tau^{1/2}+1)(1 + \sqrt[p]{K_{j-1,p}} + \sqrt[p]{K_{j,p}})|t-s|^{1/2}. \tag{4.19}$$

Proof. We proceed by induction. We start with the case when $j = 0$ and consider the following SDE

$$\begin{cases} \mathrm{d}\phi_0(t) = f_0(t, \phi_0(t))\,\mathrm{d}t + g_0(t, \phi_0(t))\,\mathrm{d}W_0(t), \quad t \in [0, \tau], \\ \phi_0(0) = x_0, \end{cases} \tag{4.20}$$

with $f_0(t, x) = f(t, x, \phi_{-1}(t)) = f(t, x, x_0)$, $g_0(t, x) = g(t, x, \phi_{-1}(t)) = g(t, x, x_0)$. Moreover, by (4.2), (4.3) we have for all $t \in [0, \tau]$, $x, y \in \mathbb{R}^d$ that

$$\max\{|f_0(t, x)|, |g_0(t, x)|\} \le \max\{K_f, K_g\}(1 + |x| + |x_0|), \tag{4.21}$$

and

$$\max\{|f_0(t,x)-f_0(t,y)|,|g_0(t,x)-g_0(t,y)|\}\leq \max\{K_f,K_g\}|x-y|. \tag{4.22}$$

Therefore, by Lemma 2.4.6 we have that there exists a unique strong solution $\phi_0 : [0,\tau]\to\mathbb{R}^d$ of (4.20) that is adapted to $(\Sigma_t^0)_{t\in[0,\tau]}$. By the moment estimate recalled in that lemma, the solution ϕ_0 satisfies (4.17) with $j=0$ and

$$K_{0,p}=c_p\Big[|x_0|^p+\tau^{p/2}K^p(1+\tau^{p/2})(1+|x_0|^p)\Big]\exp\Big(c_p\tau^pK^p(1+K^p)\Big), \tag{4.23}$$

with $K=\max\{K_f,K_g\}$. In addition, ϕ_0 satisfies (4.19) for $j=0$.

Let us now assume that for some $j\in[n-1]_0$, there exists a unique strong solution $\phi_j : [0,\tau]\times\Omega\to\mathbb{R}^d$ adapted to $(\Sigma_t^j)_{t\in[0,\tau]}$ of the following equation:

$$\begin{cases}\mathrm{d}\phi_j(t)=f_j(t,\phi_j(t))\,\mathrm{d}t+g_j(t,\phi_j(t))\,\mathrm{d}W_j(t), \quad t\in[0,\tau],\\ \phi_j(0)=\phi_{j-1}(\tau),\end{cases} \tag{4.24}$$

that satisfies (4.17), (4.19), where $f_j(t,x)=f(t+j\tau,x,\phi_{j-1}(t))$, $g_j(t,x)=g(t+j\tau,x,\phi_{j-1}(t))$. We consider the following SDE

$$\begin{cases}\mathrm{d}\phi_{j+1}(t)=f_{j+1}(t,\phi_{j+1}(t))\,\mathrm{d}t+g_{j+1}(t,\phi_{j+1}(t))\,\mathrm{d}W_{j+1}(t), \quad t\in[0,\tau],\\ \phi_{j+1}(0)=\phi_j(\tau),\end{cases} \tag{4.25}$$

with $f_{j+1}(t,x)=f(t+(j+1)\tau,x,\phi_j(t))$, $g_{j+1}(t,x)=g(t+(j+1)\tau,x,\phi_j(t))$. Since the process ϕ_j is adapted to $(\Sigma_t^j)_{t\in[0,\tau]}$, $\Sigma_t^j\subset\Sigma_t^{j+1}$ and has continuous trajectories, for all $x\in\mathbb{R}^d$ the processes $(f_{j+1}(t,x))_{t\in[0,\tau]}$, $(g_{j+1}(t,x))_{t\in[0,\tau]}$ are $(\Sigma_t^{j+1})_{t\in[0,\tau]}$-progressively measurable. Moreover, by (4.2), (4.3) for all $(t,x)\in[0,\tau]\times\mathbb{R}^d$

$$\max\{|f_{j+1}(t,x)|,|g_{j+1}(t,x)|\}\leq K(1+|\phi_j(t)|)+K|x|, \tag{4.26}$$

and for all $t \in [0,\tau]$, $x, y \in \mathbb{R}^d$ it holds

$$\max\{|f_{j+1}(t,x) - f_{j+1}(t,y)|, |g_{j+1}(t,x) - g_{j+1}(t,y)|\} \leq K|x-y|. \tag{4.27}$$

Hence, by Lemma 2.4.6 there exists a unique strong solution $\phi_{j+1} : [0,\tau] \times \Omega \to \mathbb{R}^d$ of (4.25). By the inductive assumption, the moment estimate recalled in Lemma 2.4.6, and Jensen's Inequality (see Lemma 2.4.2) we get that

$$\begin{aligned}
&\mathbb{E}\Big(\sup_{0\leq t\leq\tau} |\phi_{j+1}(t)|^p\Big) \\
&\leq c_p\bigg(\mathbb{E}|\phi_j(\tau)|^p + \mathbb{E}\Big(\int_0^\tau K(1+|\phi_j(s)|)\,\mathrm{d}s\Big)^p + \mathbb{E}\Big(\int_0^\tau K^2(1+|\phi_j(s)|)^2\,\mathrm{d}s\Big)^{p/2}\bigg) \\
&\qquad \times \exp\Big(c_p\tau^{p-1}\int_0^\tau (K^p + K^{2p})\,\mathrm{d}s\Big) \\
&\leq c_p\bigg(\mathbb{E}|\phi_j(\tau)|^p + \tau^{p-1}\int_0^\tau \mathbb{E}\left[K^p(1+|\phi_j(s)|))^p\right]\mathrm{d}s + \tau^{p/2-1}\int_0^\tau \mathbb{E}[K^p(1+|\phi_j(s)|)^p]\,\mathrm{d}s\bigg) \\
&\qquad \times \exp\Big(c_p\tau^p K^p(1+K^p)\Big) \\
&\leq c_p\Big(K_{j,p} + \tau^{p/2}K^p(\tau^{p/2}+1)(1+K_{j,p})\Big)\exp\Big(c_p\tau^p K^p(1+K^p)\Big) \\
&= K_{j+1,p},
\end{aligned} \tag{4.28}$$

and therefore, by the Hölder inequality (see Lemma 2.4.1) and BDG inequality (see Corollary 2.4.5), we get for $s,t \in [0,\tau]$, $s<t$

$$\begin{aligned}
&\mathbb{E}|\phi_{j+1}(t) - \phi_{j+1}(s)|^p \\
&\leq c_p\left(\mathbb{E}\left|\int_s^t f_{j+1}(u,\phi_{j+1}(u))du\right|^p + \left|\int_s^t g_{j+1}(u,\phi_{j+1}(u))dW(u)\right|^p\right) \\
&\leq c_p\Big[(t-s)^{p-1}\int_s^t \mathbb{E}|f_{j+1}(u,\phi_{j+1}(u))|^p\,\mathrm{d}u + (t-s)^{\frac{p-2}{2}}\int_s^t \mathbb{E}|g_{j+1}(u,\phi_{j+1}(u))|^p\,\mathrm{d}u\Big] \\
&\leq \bar{K}_{j+1,p}(t-s)^{p/2},
\end{aligned} \tag{4.29}$$

where $\bar{K}_{j+1,p} = c_p^p K^p(\tau^{p/2}+1)(1+K_{j,p}+K_{j+1,p})$. This ends the inductive proof. □

4.4 Error analysis for the randomised Euler algorithm

In this section we provide error analysis for randomised Euler scheme under slightly stronger assumptions than those stated in Section 4.2. Namely, for the drift coefficient we assume that instead of (A3) it satisfies what follows:

(A3') There exist $\alpha_1 \in (0,1]$, $\bar{K}_f, L_f \in (0,\infty)$ such that for all $t \in [0,(n+1)\tau]$

$$|f(t,0,0)| \leq \bar{K}_f, \tag{4.30}$$

and for all $t \in [0,(n+1)\tau]$, $x_1, x_2, z_1, z_2 \in \mathbb{R}^d$

$$|f(t,x_1,z_1) - f(t,x_2,z_2)| \leq L_f(|x_1 - x_2| + |z_1 - z_2|^{\alpha_1}). \tag{4.31}$$

For the diffusion coefficient we impose the following assumption, that is stronger than (B3):

(B3') There exist $\alpha_2, \varrho \in (0,1]$, $\bar{K}_g, L_g \in (0,\infty)$ such that for all $t_1, t_2 \in [0,(n+1)\tau]$, $x, z \in \mathbb{R}^d$

$$|g(t_1,x,z) - g(t_2,x,z)| \leq \bar{K}_g(1 + |x| + |z|)|t_1 - t_2|^{\varrho}, \tag{4.32}$$

and for all $t \in [0,(n+1)\tau]$, $x_1, x_2, z_1, z_2 \in \mathbb{R}^d$

$$|g(t,x_1,z_1) - g(t,x_2,z_2)| \leq L_g(|x_1 - x_2| + |z_1 - z_2|^{\alpha_2}). \tag{4.33}$$

Note that for any f that satisfies the assumption (A3') it holds for all $t \in [0,(n+1)\tau]$, $x, z \in \mathbb{R}^d$

$$|f(t,x,z)| \leq K_f(1 + |x| + |z|), \tag{4.34}$$

with $K_f = \bar{K}_f + L_f$, while for any g satisfying (B3') we have for all $t \in [0,(n+1)\tau]$, $x, z \in \mathbb{R}^d$

$$|g(t,x,z)| \leq K_g(1 + |x| + |z|), \tag{4.35}$$

with $K_g = |g(0,0,0)| + L_g + \bar{K}_g((n+1)\tau)^{\varrho}$. Note that under **(A3')** and **(B3')**, f and

g automatically satisfy (4.2) and (4.3).

The strengthened assumptions above are used only for the error analysis, not for the basic existence result. The weaker assumptions in Section 4.2 are sufficient to construct the SDDE solution and obtain moment estimates. In contrast, the randomised Euler error proof requires quantitative control of the drift and diffusion coefficients when the time argument, the current state, and the delayed state are perturbed. The Lipschitz dependence on the current state is used in the stability estimates after the error decomposition, while the Hölder dependence on the delayed state controls the propagation of the previous interval's error through the delay term. For the diffusion coefficient, the Hölder continuity in time in **(B3')** is also needed to estimate the local stochastic consistency error and to identify the regularity contribution $\min\{\varrho, 1/2\}$ in Theorem 4.4.4. These assumptions still allow time-irregular coefficients; differentiability in time is not required. Typical examples are coefficients of the form

$$f(t,x,z) = a_f(t) + B_f x + \psi_f(z), \qquad g(t,x,z) = a_g(t) + B_g x + \psi_g(z),$$

where a_g is ϱ-Hölder continuous, the time-dependent part of f is bounded as required in **(A3')**, B_f, B_g are bounded linear maps, and ψ_f, ψ_g are Hölder continuous in the delayed variable with exponents α_1 and α_2, respectively. Thus the strengthened assumptions describe the regularity needed to prove rates, while remaining compatible with the low-regularity time-dependent coefficients considered in this thesis.

Remark 4.4.1 (Relation to assumptions in related SDDE approximation work)**.** The assumptions used in this chapter should be interpreted according to the part of the analysis in which they are used. Classical well-posedness results for SDDEs, such as the result recalled from [68], are typically based on Lipschitz and growth conditions which guarantee existence, uniqueness, and moment bounds. The assumptions (A1)–(A3) and (B1)–(B3) play this role here, with the delay equation treated interval by interval.

The strengthened assumptions **(A3')** and **(B3')** are imposed for a different purpose. They are not meant to claim a weakest possible existence theory for SDDEs. Rather,

they provide the quantitative regularity needed for the strong error analysis of the randomised Euler scheme. In the non-delay SDE setting, randomised approximation results such as [74] already show that the convergence rate depends on the available time regularity and on martingale estimates for the stochastic integral. In the present SDDE setting the same type of local error control must be combined with an additional delay-interval induction: errors made on one interval enter the delayed argument on the next interval. This is why Hölder continuity in the delayed variable is tracked explicitly through the exponents α_1 and α_2.

Thus the contribution here is not a blanket weakening of all assumptions in the SDDE literature. It is a rate-oriented framework for Carathéodory-type SDDEs which combines standard moment estimates, randomised quadrature ideas for low-regularity drift terms, BDG-based estimates for the stochastic part, and delay-error propagation. This explains the form of the final rate in Theorem 4.4.4, where both the regularity of the coefficients and the repeated propagation through delay intervals are visible.

Before proceeding with the error analysis, we introduce the following auxiliary lemma regarding measurability properties of stochastic processes evaluated at random times, which will be crucial for our subsequent proofs.

Lemma 4.4.1. *Let $Y = (Y(t))_{t\geq 0}$ is $(\Sigma_t)_{t\geq 0}$-progressively measurable stochastic process and let $\xi : \Omega \mapsto [a, b]$, $0 \leq a < b < +\infty$, is a random variable on $(\Omega, \Sigma, \mathbb{P})$. Then $Y(\xi)$ is $\sigma(\xi) \vee \Sigma_b$-measurable.*

Proof. Since Y is progressively measurable, its restriction to $[0, b] \times \Omega$ is $\mathcal{B}([0, b]) \otimes \Sigma_b$-to-$\mathcal{B}(\mathbb{R}^d)$ measurable. In particular, the mapping $[a, b] \times \Omega \ni (t, \omega) \mapsto Y(t, \omega) \in \mathbb{R}^d$ is $\mathcal{B}([a, b]) \otimes \Sigma_b$-to-$\mathcal{B}(\mathbb{R}^d)$ measurable. Define

$$H : \Omega \to [a, b] \times \Omega, \qquad H(\omega) := (\xi(\omega), \omega),$$

and let $\mathcal{F}_0 := \{B \times F : B \in \mathcal{B}([a, b]),\ F \in \Sigma_b\}$, so that $\mathcal{B}([a, b]) \otimes \Sigma_b = \sigma(\mathcal{F}_0)$. For any $B \times F \in \mathcal{F}_0$ we have

$$H^{-1}(B \times F) = \{\omega : \xi(\omega) \in B\} \cap F = \xi^{-1}(B) \cap F \in \sigma(\xi) \vee \Sigma_b.$$

Hence, by Proposition 2.3. in [17], the map H is $\sigma(\xi) \vee \Sigma_b$-to-$\mathcal{B}([a,b]) \otimes \Sigma_b$ measurable. Therefore, for $\omega \in \Omega$,

$$Y(\xi(\omega), \omega) = (Y \circ H)(\omega),$$

where $\circ$ denotes composition, i.e. $(Y \circ H)(\omega) := Y(H(\omega))$. For any $A \in \mathcal{B}(\mathbb{R}^d)$,

$$(Y \circ H)^{-1}(A) = H^{-1}(Y^{-1}(A)) \in \sigma(\xi) \vee \Sigma_b,$$

which shows that $Y(\xi)$ is $\sigma(\xi) \vee \Sigma_b$-measurable. □

Recall the expression of randomised Euler-Maruyama scheme in (4.7)-(4.8) as well as the expressions for f_l and g_l in (4.13). In order to perform error analysis, we define the auxiliary randomised Euler-Maruyama scheme with solutions $\{\bar{y}_k^l\}$ as follows:

$$\bar{y}_0^0 := y_0^0 = y_N^{(-1)} = x_0, \tag{4.36}$$

$$\bar{y}_{k+1}^0 := \bar{y}_k^0 + h \cdot f_0(\theta_{k+1}^0, \bar{y}_k^0) + g_0(t_k^0, \bar{y}_k^0)\Delta_k^0 W, \ k = 0, 1, \ldots, N-1, \tag{4.37}$$

and for $l = 0, 1, \ldots, n-1$

$$\bar{y}_0^{l+1} := y_0^{l+1} = y_N^l, \tag{4.38}$$

$$\begin{aligned} \bar{y}_{k+1}^{l+1} :=& \bar{y}_k^{l+1} + h \cdot f_{l+1}(\delta_{k+1}^{l+1}, \bar{y}_k^{l+1}) \\ &+ g_{l+1}(t_k^0, \bar{y}_k^{l+1})\Delta_k^{l+1} W, \ k = 0, 1, \ldots, N-1, \end{aligned} \tag{4.39}$$

where $\delta_{k+1}^{l+1} = kh + \gamma_{k+1}^{l+1} h$ and $\Delta_k^{l+1} W := W(t_{k+1}^{l+1}) - W(t_k^{l+1})$ for $l = -1, 0, \ldots, n-1$.

The auxiliary randomised Euler-Maruyama scheme differs from the randomised Euler-Maruyama one over each $[l\tau, (l+1)\tau]$ in that the delay terms of f and g evaluated in the former scheme are the exact values while the ones in the latter scheme are numerical approximations from $[(l-1)\tau, l\tau]$.

Note that δ_{k+1}^{l+1} is uniformly distributed in (t_k^0, t_{k+1}^0) and $\theta_{k+1}^{l+1} = \delta_{k+1}^{l+1} + (l+1)\tau$ is uniformly distributed in $(t_k^{l+1}, t_{k+1}^{l+1})$. For $N \geq 2$ we have that $h \in (0, \tau)$, which gives that $t_k^{l+1} > t_{k+1}^l$ and $\Sigma_{t_{k+1}^l} \subset \Sigma_{t_k^{l+1}}$. This fact and Lemma 4.4.1 imply that for

$\phi_l(\delta_{k+1}^{l+1}) = X(t_k^l + h\gamma_{k+1}^{l+1})$, where ϕ_l is defined in (4.11), we have

$$\sigma(\phi_l(\delta_{k+1}^{l+1})) \subset \sigma(\{\gamma_{k+1}^{l+1}\}) \vee \Sigma_{t_{k+1}^l} \subset \sigma(\{\gamma_{k+1}^{l+1}\}) \vee \Sigma_{t_k^{l+1}}, \tag{4.40}$$

and hence, $\phi_l(\delta_{k+1}^{l+1})$ is independent of $\Delta_k^{l+1}W$. Moreover, it follows from (4.13), (4.39) that

$$\begin{aligned}\bar{y}_{k+1}^{l+1} =& \bar{y}_k^{l+1} + h \cdot f(t_k^{l+1} + h\gamma_{k+1}^{l+1}, \bar{y}_k^{l+1}, \phi_l(\delta_{k+1}^{l+1})) \\ &+ g(t_k^{l+1}, \bar{y}_k^{l+1}, \phi_l(t_k^0)) \cdot \Delta_k^{l+1}W.\end{aligned} \tag{4.41}$$

Hence, by (4.41), (4.40), and induction we get for $j \in [n]_0$ and $k \in [N]_0$ that

$$\sigma(\bar{y}_k^j) \subset \Sigma_{t_k^j} \vee \sigma(\{\gamma_1^0 \ldots, \gamma_N^0, \ldots, \gamma_1^{j-1}, \ldots, \gamma_N^{j-1}, \gamma_1^j, \ldots, \gamma_k^j\}). \tag{4.42}$$

Therefore, $\bar{y}_k^j$ is measurable with respect to larger σ-field than y_k^j (see (4.9)). Note that $\bar{y}_k^l$ is not implementable. However, we use $\bar{y}_k^l$ only in order to estimate the error (4.10) of (implementable scheme) y_k^l.

Before turning to the error analysis, we note that both the randomised Euler scheme and its auxiliary counterpart admit finite second moments on each subinterval. The following lemma collects this elementary estimate.

Lemma 4.4.2. *Assume that f, g satisfy (A1)–(A3) and (B1)–(B3). Let $\{y_k^j\}$ be given by* (4.7)–(4.8) *and let $\{\bar{y}_k^j\}$ be given by* (4.36)–(4.39)*. Then for every $j \in [n]_0$ it holds*

$$\max_{0\le k\le N} \mathbb{E}|y_k^j|^2 < \infty, \qquad \max_{0\le k\le N} \mathbb{E}|\bar{y}_k^j|^2 < \infty.$$

Proof. We first show the assertion for $\{y_k^j\}$ by induction over j. For $j = -1$ we have $y_k^{(-1)} = x_0$, hence $\max_k \mathbb{E}|y_k^{(-1)}|^2 = |x_0|^2 < \infty$. Fix $j \in [n]_0$ and assume that $\max_{0\le k\le N} \mathbb{E}|y_k^{j-1}|^2 < \infty$. Since $y_0^j = y_N^{j-1}$, we have $\mathbb{E}|y_0^j|^2 < \infty$. For $k \in [N-1]_0$, by

(4.2), (4.3), (4.8), $\mathbb{E}|\Delta_k^j W|^2 = h$, and $|a+b+c|^2 \le 3(|a|^2+|b|^2+|c|^2)$, we have

$$\begin{aligned}
\mathbb{E}|y_{k+1}^j|^2 &= \mathbb{E}\Big|y_k^j + hf(\theta_{k+1}^j, y_k^j, y_k^{j-1}) + g(t_k^j, y_k^j, y_k^{j-1})\Delta_k^j W\Big|^2 \\
&\le 3\mathbb{E}|y_k^j|^2 + 3h^2\mathbb{E}|f(\theta_{k+1}^j, y_k^j, y_k^{j-1})|^2 + 3\mathbb{E}|g(t_k^j, y_k^j, y_k^{j-1})\Delta_k^j W|^2 \\
&\le 3\mathbb{E}|y_k^j|^2 + 9h^2K_f^2\,\mathbb{E}(1+|y_k^j|^2+|y_k^{j-1}|^2) + 9hK_g^2\,\mathbb{E}(1+|y_k^j|^2+|y_k^{j-1}|^2) \\
&\le (1+Ch)\mathbb{E}|y_k^j|^2 + Ch\Big(1+\mathbb{E}|y_k^{j-1}|^2\Big),
\end{aligned}$$

where we used $h^2 \le \tau h$ (since $h = \tau/N \le \tau$) and absorbed all constants into $C < \infty$.

Taking maximum over k and applying a discrete Gronwall argument (see Lemma 2.4.7) yields $\max_{0\le k\le N}\mathbb{E}|y_k^j|^2 < \infty$. This completes the induction in j.

We now prove the bound for $\{\bar{y}_k^j\}$. For $j = 0$ we have $\bar{y}_k^0 = y_k^0$, hence $\max_k \mathbb{E}|\bar{y}_k^0|^2 < \infty$. Let $j \ge 1$. Using (4.41) and repeating the above estimate with y_k^j replaced by $\bar{y}_k^j$ and y_k^{j-1} replaced by ϕ_{j-1}, we obtain

$$\mathbb{E}|\bar{y}_{k+1}^j|^2 \le (1+Ch)\,\mathbb{E}|\bar{y}_k^j|^2 + Ch\Big(1+\mathbb{E}|\phi_{j-1}(t_k^0)|^2\Big).$$

By Theorem 4.3.1, $\sup_{t\in[0,\tau]}\mathbb{E}|\phi_{j-1}(t)|^2 < \infty$, and thus $\max_{0\le k\le N}\mathbb{E}|\phi_{j-1}(t_k^0)|^2 < \infty$. Another discrete Gronwall argument (see Lemma 2.4.7) yields $\max_{0\le k\le N}\mathbb{E}|\bar{y}_k^j|^2 < \infty$. □

To bound the error between the implementable randomised Euler scheme (y_k^l) and the SDDE's exact solution (X), we will analyse the error chain $|X-y| \le |X-\bar{y}|+|\bar{y}-y|$. The following lemma quantifies key component errors between the decomposed exact solution ϕ_l and $\bar{y}_k^l$, laying the groundwork for total error aggregation.

Lemma 4.4.3. *Let $n \in \mathbb{N}_0$, $\tau \in (0,+\infty)$, $x_0 \in \mathbb{R}^d$, and let f, g satisfy the assumptions (A1), (A2), (A3') and (B1), (B2), (B3') for some $\varrho, \alpha_1, \alpha_2 \in (0,1]$. Consider the initial-value problem* (4.12) *for ϕ_l with $l \in [n]_0$. For any $N \ge \lceil\tau\rceil$, take the grid points $\{t_k^l\}_{k=0}^N$ defined in* (4.4) *for $[l\tau,(l+1)\tau]$ with δ_j^l defined in* (4.39)*, then the following*

estimates hold

$$\begin{aligned}&\Big\|\max_{1\le k\le N}\Big|\sum_{j=1}^{k}\Big(\int_{t_{j-1}^0}^{t_j^0} f_l(s,\phi_l(s))\,\mathrm{d}s - hf_l(\delta_j^l,\phi_l(\delta_j^l))\Big)\Big|\Big\|_{L^p(\Omega)}\\&\le c_p\tau^{1/2}K_f(1+\sqrt[p]{K_{l-1,p}})(1+\sqrt[p]{K_{l,p}})h^{1/2}.\end{aligned}\tag{4.43}$$

$$\begin{aligned}&\Big\|\max_{1\le k\le N}\Big|h\sum_{j=1}^{k}\Big(f_l(\delta_j^l,\phi_l(\delta_j^l)) - f_l(\delta_j^l,\phi_l(t_{j-1}^0))\Big)\Big|\Big\|_{L^p(\Omega)}\\&\le c_pL_f\tau K(\tau^{1/2}+1)(1+\sqrt[p]{K_{l-1,p}}+\sqrt[p]{K_{l,p}})h^{1/2},\end{aligned}\tag{4.44}$$

$$\begin{aligned}&\Big\|\max_{1\le k\le N}\Big|\int_0^{t_k^0}\sum_{j=1}^{N}\big(g_l(s,\phi_l(s)) - g_l(t_{j-1}^0,\phi_l(s))\big)\mathbf{1}_{[t_{j-1}^0,t_j^0)}(s)\,\mathrm{d}W_l(s)\Big|\Big\|_{L^p(\Omega)}\\&\le \mathbf{1}_{\{l=0\}}(l)\tilde{c}_{l,p}h^{\varrho} + \mathbf{1}_{\{l\ne 0\}}(l)\tilde{c}_{l,p}h^{\min\{\varrho,\alpha_2/2\}},\end{aligned}\tag{4.45}$$

and

$$\begin{aligned}&\Big\|\max_{1\le k\le N}\Big|\int_0^{t_k^0}\sum_{j=1}^{N}\big(g_l(t_{j-1}^0,\phi_l(s)) - g_l(t_{j-1}^0,\phi_l(t_{j-1}^0))\big)\mathbf{1}_{[t_{j-1}^0,t_j^0)}(s)\,\mathrm{d}W_l(s)\Big|\Big\|_{L^p(\Omega)}\\&\le \tilde{c}_{l,p}h^{1/2},\end{aligned}\tag{4.46}$$

where $\tilde{c}_{l,p}$ is a generic constant that depends on both l and p.

Proof. We consider the initial-value problem (4.12) for ϕ_l. By (A1), (A2), (A3') and (B1), (B2), (B3') we have that f_l and g_l are Borel measurable, and for all $t, t_1, t_2 \in [0,\tau]$, $x, y \in \mathbb{R}^d$

$$\begin{aligned}&|f_l(t,x)| \le K_f(1+|x|+|\phi_{l-1}(t)|),\\&|f_l(t,x) - f_l(t,y)| \le L_f|x-y|.\end{aligned}\tag{4.47}$$

and

$$\begin{aligned}
&|g_l(t,x)| \le K_g(1+|x|+|\phi_{l-1}(t)|),\\
&|g_l(t,x)-g_l(t,y)| \le L_g|x-y|,\\
&|g_l(t_1,x)-g_l(t_2,x)|\\
&\le \bar{K}_g(1+|x|+|\phi_{l-1}(t_1)|)|t_1-t_2|^{\varrho} + \mathbf{1}_{\{l\neq 0\}}(l)L_g|\phi_{l-1}(t_1)-\phi_{l-1}(t_2)|^{\alpha_2}.
\end{aligned} \tag{4.48}$$

To show (4.43), we define

$$\begin{aligned}
&\sum_{j=1}^{k}\Big(\int_{t_{j-1}^0}^{t_j^0} f_l(s,\phi_l(s))\,\mathrm{d}s - h f_l(\delta_j^l,\phi_l(\delta_j^l))\Big)\\
&:= \int_0^{t_k^0} Y_l(s)\,\mathrm{d}s - h\sum_{k=1}^{k} Y_l(\delta_j^l)\\
&:= \int_0^{t_k^0} Y_l(s)\,\mathrm{d}s - Q_{\gamma,h}^{k,l}(Y_l),
\end{aligned} \tag{4.49}$$

where $\delta_j^l = t_{j-1}^0 + h\gamma_j^l$, $Y_l(t) = f_l(t,\phi_l(t)) = f\big(t+l\tau,\phi_l(t),\phi_{l-1}(t)\big)$, $Q_{\gamma,h}^{k,l}(Y_l) := h\sum\limits_{j=1}^{k} Y_l(\delta_j^l)$, and $\sigma((Y(t))_{t\in[0,\tau]}) \subset \Sigma_\infty$. Hence, the process Y is independent of the σ-field $\sigma(\{\gamma_k^j\}_{j\in\mathbb{N}_0,k\in\mathbb{N}})$. Moreover, by Theorem 4.3.1, (4.47) and Jensen's inequality (see Lemma 2.4.2), we arrive at

$$\begin{aligned}
&\|Y_l\|_{L^p([0,\tau]\times\Omega;\mathbb{R}^d)} = \|f_l(\cdot,\phi_l(\cdot))\|_{L^p([0,\tau]\times\Omega;\mathbb{R}^d)}\\
&\le K_f(\|1\|_{L^p([0,\tau]\times\Omega;\mathbb{R}^d)} + \|\phi_l(\cdot)\|_{L^p([0,\tau]\times\Omega;\mathbb{R}^d)} + \|\phi_{l-1}(\cdot)\|_{L^p([0,\tau]\times\Omega;\mathbb{R}^d)}\\
&\le \tau^{1/p} c_p K_f (1+K_{l-1,p}+K_{l,p})^{1/p} < +\infty.
\end{aligned} \tag{4.50}$$

Therefore, by Theorem 4.1 in [62] and using the estimate in (4.50) we obtain

$$\begin{aligned}
&\Big\| \max_{1\le k\le N} \Big| \sum_{j=1}^{k} \Big(\int_{t_{j-1}^0}^{t_j^0} f_l(s,\phi_l(s))\,\mathrm{d}s - h f_l(\delta_j^l, \phi_l(\delta_j^l)) \Big) \Big| \Big\|_{L^p(\Omega)} \\
&= \Big\| \max_{1\le k\le N} \Big| \int_0^{t_k^0} Y_l(s)\,\mathrm{d}s - Q_{\gamma,h}^{k,l}(Y_l) \Big| \Big\|_{L^p(\Omega)} \\
&\le c_p \tau^{\frac{p-2}{2p}} \|Y_l\|_{L^p([0,\tau]\times\Omega;\mathbb{R}^d)} h^{1/2} \\
&\le c_p \tau^{1/2} K_f (1 + K_{l-1,p} + K_{l,p})^{1/p} h^{1/2}. \\
&\le c_p \tau^{1/2} K_f (1 + \sqrt[p]{K_{l-1,p}})(1 + \sqrt[p]{K_{l,p}}) h^{1/2}.
\end{aligned} \tag{4.51}$$

To get (4.44), by (4.47) and Jensen's inequality (see Lemma 2.4.2) we have

$$\begin{aligned}
&\max_{1\le k\le N} \Big| h \sum_{j=1}^{k} \Big(f_l(\delta_j^l, \phi_l(\delta_j^l)) - f_l(\delta_j^l, \phi_l(t_{j-1}^0)) \Big) \Big|^p \\
&\le \max_{1\le k\le N} \Big| L_f h \sum_{j=1}^{k} \Big(\phi_l(\delta_j^l) - \phi_l(t_{j-1}^0) \Big) \Big|^p \\
&\le \max_{1\le k\le N} \left(L_f h \sum_{j=1}^{k} \Big| \phi_l(\delta_j^l) - \phi_l(t_{j-1}^0) \Big| \right)^p \\
&\le \max_{1\le k\le N} \left((L_f h)^p k^{p-1} \sum_{j=1}^{k} \Big| \phi_l(\delta_j^l) - \phi_l(t_{j-1}^0) \Big|^p \right) \\
&\le \tau^{p-1} L_f^p h \sum_{j=1}^{N} |\phi_l(\delta_j^l) - \phi_l(t_{j-1}^0)|^p.
\end{aligned}$$

Thus, by Theorem 4.3.1

$$\begin{aligned}
&\Big\| \max_{1\le k\le N} \Big| h \sum_{j=1}^{k} \Big(f_l(\delta_j^l, \phi_l(\delta_j^l)) - f_l(\delta_j^l, \phi_l(t_{j-1}^0)) \Big) \Big| \Big\|_{L^p(\Omega)} \\
&\le c_p L_f \tau K (\tau^{1/2} + 1)(1 + \sqrt[p]{K_{l-1,p}} + \sqrt[p]{K_{l,p}}) h^{1/2}.
\end{aligned} \tag{4.52}$$

Regarding (4.45), define for all $t \in [0, t_N^0]$ the following stochastic process

$$M^l(t) := \int_0^t \sum_{j=1}^{N} \left(g_l(s, \phi_l(s)) - g_l(t_{j-1}^0, \phi_l(s))\right) \mathbf{1}_{[t_{j-1}^0, t_j^0)}(s)\, \mathrm{d}W_l(s). \tag{4.53}$$

Since the process

$$\Big(\sum_{j=1}^{N} \left(g_l(s, \phi_l(s)) - g_l(t_{j-1}^0, \phi_l(s))\right) \mathbf{1}_{[t_{j-1}^0, t_j^0)}(s)\Big)_{t \in [0, t_N^0]} \tag{4.54}$$

has càdlàg trajectories (recall that càdlàg means right-continuous with left limits; hence the trajectories are also in $L^2([0, t_N^0])$) and is adapted, the stochastic integral $M^l(t)$ is well-defined for all $t \in [0, t_N^0]$ (see, for example, page 30 in [68]). By the basic property of Itô integrals (see, e.g., [68]), for $t \in [0, t_N^0]$ the quadratic variation of the martingale M^l is given by

$$\langle M^l \rangle(t) = \int_0^t \sum_{j=1}^{N} \left|g_l(s, \phi_l(s)) - g_l(t_{j-1}^0, \phi_l(s))\right|^2 \mathbf{1}_{[t_{j-1}^0, t_j^0)}(s)\, \mathrm{d}s. \tag{4.55}$$

Now we are able to conclude the L^q boundedness for $\langle M^l \rangle$ for any $q \geq 2$ by using

Jensen's inequality (see Lemma 2.4.2), Theorem 4.3.1 and (4.48) on g_l:

$$
\begin{aligned}
&\mathbb{E}\Big[|\langle M^l\rangle(t_N^0)|^q\Big] \\
&= \mathbb{E}\bigg(\int_0^{t_N^0}\sum_{j=1}^{N}\big|g_l(s,\phi_l(s)) - g_l(t_{j-1}^0,\phi_l(s))\big|^2\mathbf{1}_{[t_{j-1}^0,t_j^0)}(s)\,\mathrm{d}s\bigg)^q \\
&= \mathbb{E}\bigg(\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}\big|g_l(s,\phi_l(s)) - g_l(t_{j-1}^0,\phi_l(s))\big|^2\,\mathrm{d}s\bigg)^q \\
&\le (\frac{\tau}{h})^{q-1}\sum_{j=1}^{N}\mathbb{E}\bigg(\int_{t_{j-1}^0}^{t_j^0}\big|g_l(s,\phi_l(s)) - g_l(t_{j-1}^0,\phi_l(s))\big|^2\,\mathrm{d}s\bigg)^q \\
&\le \tau^{q-1}\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}\mathbb{E}|g_l(s,\phi_l(s)) - g_l(t_{j-1}^0,\phi_l(s))|^{2q}\,\mathrm{d}s \\
&\le \tau^{q-1}\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}\mathbb{E}\Big(\bar{K}_g|s-t_{j-1}^0|^{\varrho}(1+|\phi_l(s)|+|\phi_{l-1}(s)|) \\
&\qquad + \mathbf{1}_{\{l\neq 0\}}(l)L_g|\phi_{l-1}(s)-\phi_{l-1}(t_{j-1}^0)|^{\alpha_2}\Big)^{2q}\,\mathrm{d}s \qquad (4.56)\\
&\le c_{2q}\tau^{q-1}\Big(\bar{K}_g^{2q}\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}|s-t_{j-1}^0|^{2q\varrho}\mathbb{E}(1+|\phi_l(s)|+|\phi_{l-1}(s)|)^{2q}\,\mathrm{d}s \\
&\qquad + \mathbf{1}_{\{l\neq 0\}}(l)L_g^{2q}\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}\mathbb{E}|\phi_{l-1}(s)-\phi_{l-1}(t_{j-1}^0)|^{2q\alpha_2}\,\mathrm{d}s\Big) \\
&\le c_{2q}\tau^{q-1}\bigg(\bar{K}_g^{2q}(1+K_{l-1,2q})(1+K_{l,2q})\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}|s-t_{j-1}^0|^{2q\varrho}\,ds \\
&\qquad\qquad + \mathbf{1}_{\{l\neq 0\}}(l)\,L_g^{2q}\,\bar{K}_{l-1,2q}\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}|s-t_{j-1}^0|^{q\alpha_2}\,ds\bigg), \\
&\le c_{2q}\tau^q\bar{K}_g^{2q}(1+K_{l-1,2q})(1+K_{l,2q})h^{2q\varrho} + \mathbf{1}_{\{l\neq 0\}}(l)c_{2q}\tau^q\bar{K}_{l-1,2q}L_g^{2q}h^{q\alpha_2} \\
&< \infty.
\end{aligned}
$$

By now we have shown that M^l is a martingale and we shall apply BDG inequality

(Corollary 2.4.5) and reuse the estimate in (4.56) with $q = p/2$ to get

$$\begin{aligned}
&\mathbb{E}\bigg[\max_{1\le k\le N}\Big|\int_0^{t_k^0}\sum_{j=1}^{N}\big(g_l(s,\phi_l(s))-g_l(t_{j-1}^0,\phi_l(s))\big)\mathbf{1}_{[t_{j-1}^0,t_j^0)}(s)\,\mathrm{d}W_l(s)\Big|^p\bigg]\\
&\le c_p\mathbb{E}\bigg(\int_0^{t_N^0}\sum_{j=1}^{N}\big|g_l(s,\phi_l(s))-g_l(t_{j-1}^0,\phi_l(s))\big|^2\mathbf{1}_{[t_{j-1}^0,t_j^0)}(s)\,\mathrm{d}s\bigg)^{p/2}\\
&\le c_p\tau^{p/2}\bar{K}_g^p(1+K_{l-1,p})(1+K_{l,p})h^{p\varrho}+\mathbf{1}_{\{l\neq 0\}}(l)c_p\tau^{p/2}\bar{K}_{l-1,p}L_g^p h^{p\alpha_2/2}\\
&\le \mathbf{1}_{\{l=0\}}(l)\tilde{c}_{l,p}h^{p\varrho}+\mathbf{1}_{\{l\neq 0\}}(l)\tilde{c}_{l,p}h^{p\min\{\varrho,\alpha_2/2\}}.
\end{aligned}\tag{4.57}$$

Similarly, we can show the integral in (4.46) is well-defined and is a martingale. Utilising BDG inequality (Corollary 2.4.5), Theorem 4.3.1 and (4.48) on g_l we get

$$\begin{aligned}
&\mathbb{E}\Big[\max_{1\le k\le N}\Big|\int_0^{t_k^0}\sum_{j=1}^{N}\big(g_l(t_{j-1}^0,\phi_l(s))-g_l(t_{j-1}^0,\phi_l(t_{j-1}^0))\big)\mathbf{1}_{[t_{j-1}^0,t_j^0)}(s)\,\mathrm{d}W_l(s)\Big|^p\Big]\\
&\le c_p\mathbb{E}\bigg(\int_0^{t_N^0}\sum_{j=1}^{N}\big|g_l(t_{j-1}^0,\phi_l(s))-g_l(t_{j-1}^0,\phi_l(t_{j-1}^0))\big|^2\mathbf{1}_{[t_{j-1}^0,t_j^0)}(s)\,\mathrm{d}s\bigg)^{p/2}\\
&\le c_pL_g^p\mathbb{E}\bigg(\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}\big|\phi_l(s)-\phi_l(t_{j-1}^0)\big|^2\,\mathrm{d}s\bigg)^{p/2}\\
&\le c_pL_g^p\Big(\frac{\tau}{h}\Big)^{\frac{p}{2}-1}\sum_{j=1}^{N}\mathbb{E}\bigg(\int_{t_{j-1}^0}^{t_j^0}\big|\phi_l(s)-\phi_l(t_{j-1}^0)\big|^2\,\mathrm{d}s\bigg)^{p/2}\\
&\le c_pL_g^p\tau^{\frac{p}{2}-1}\sum_{j=1}^{N}\int_{t_{j-1}^0}^{t_j^0}\mathbb{E}\big[\big|\phi_l(s)-\phi_l(t_{j-1}^0)\big|^p\big]\,\mathrm{d}s\le\tilde{c}_{l,p}h^{p/2}.
\end{aligned}\tag{4.58}$$

□

Theorem 4.4.4. *Let $n\in\mathbb{N}_0$, $\tau\in(0,+\infty)$, $x_0\in\mathbb{R}^d$, and let f,g satisfy the assumptions (A1), (A2), (A3') and (B1), (B2), (B3') for some $\varrho,\alpha_1,\alpha_2\in(0,1]$. For any $p\ge 2$ $p\in\mathbb{N}$, there exist $C_{0,p},C_{1,p},\dots,C_{n,p}\in(0,+\infty)$ such that for all $N\ge\lceil\tau\rceil$ and $l=$*

$0, 1, \ldots, n$ *it holds*

$$\left\| \max_{0\le k\le N} |X(t_k^l) - y_k^l| \right\|_{L^p(\Omega)} \le C_{l,p} h^{\min\{\varrho,\frac{1}{2}\}\alpha^l}, \tag{4.59}$$

where $\alpha := \min\{\alpha_1, \alpha_2\}$*, and, in particular, if* $\alpha = 1$ *then*

$$\left\| \max_{0\le k\le N} |X(t_k^l) - y_k^l| \right\|_{L^p(\Omega)} \le C_{l,p} h^{\min\{\varrho,\frac{1}{2}\}}. \tag{4.60}$$

Remark 4.4.5 (Discussion of the convergence rate)**.** *The rate in* (4.59) *combines the temporal regularity of the stochastic equation with the propagation of delay errors. On the first interval, where* $l = 0$*, the exponent is* $\min\{\varrho, \frac{1}{2}\}$*. Here* ϱ *comes from the Hölder regularity in time of the diffusion coefficient in* (4.32)*, while the exponent* $\frac{1}{2}$ *is the natural strong-order limitation associated with Brownian increments and the stochastic integral estimates. For later intervals, the delayed argument introduces the factor* α^l*, where* $\alpha = \min\{\alpha_1, \alpha_2\}$ *is the weakest Hölder exponent in the delayed variable for the drift and diffusion coefficients. Hence each passage to the next delay interval can reduce the theoretical order by a factor* α*. If* $\alpha = 1$*, the delayed dependence is Lipschitz and the order remains* $\min\{\varrho, \frac{1}{2}\}$ *on every fixed interval, as stated in* (4.60)*; nevertheless, the constants* $C_{l,p}$ *may still grow with* l*.*

Before giving the proof, we outline its structure. The argument first treats the initial interval $[0, \tau]$, where the delayed value is the deterministic initial datum and the auxiliary and implementable schemes coincide. On this interval, the error is decomposed into drift quadrature terms, stochastic integral terms, and stability terms, which are estimated using the randomised quadrature estimate, the BDG inequality, Jensen's inequality, and a discrete Gronwall argument. For the later intervals, the proof proceeds by induction over the delay interval index l. The auxiliary scheme $\bar{y}^{l+1}$ is used to separate the new local discretisation error on $[l\tau, (l+1)\tau]$ from the error propagated from the previous interval. The total error is then bounded through the decomposition

$$\|X - y\| \le \|X - \bar{y}\| + \|\bar{y} - y\|,$$

and the Hölder dependence on the delayed variable transfers the previous interval error into the exponent α^l. This explains why the convergence order in (4.59) deteriorates with l unless $\alpha = 1$.

Proof. The proof uses the error-decomposition and martingale-estimate strategy from [74], with additional induction over the delay intervals and with the auxiliary scheme introduced above to handle the delayed arguments. We start with the initial-vale problem (4.12) with $l = 0$. Since $\bar{y}_0^0 = y_0^0 = x_0$, $f_0(t,x) = f(t,x,x_0)$ and $g_0(t,x) = g(t,x,x_0)$, we have that $\bar{y}_k^0 = y_k^0$ for all $k = 0,\dots,N$. For $k \in [N]$ we have by the facts $\phi_0(0) = y_0^0 = x_0$ and $\bar{y}_k^0 = y_k^0$ for all $k = 0,\dots,N$ that

$$
\begin{aligned}
&\phi_0(t_k^0) - y_k^0 = \phi_0(0) - y_0^0 + (\phi_0(t_k^0) - \phi_0(t_0^0)) - (y_k^0 - y_0^0) \\
&= \sum_{j=1}^{k} (\phi_0(t_j^0) - \phi_0(t_{j-1}^0)) - \sum_{j=1}^{k} (y_j^0 - y_{j-1}^0) \\
&= \sum_{j=1}^{k} \left(\int_{t_{j-1}^0}^{t_j^0} f_0(s, \phi_0(s))\,\mathrm{d}s - h f_0(\delta_j^0, y_{j-1}^0) \right) \\
&\quad + \sum_{j=1}^{k} \left(\int_{t_{j-1}^0}^{t_j^0} g_0(s, \phi_0(s))\,\mathrm{d}W_0(s) - g_0(t_{j-1}^0, y_{j-1}^0) \Delta_{j-1}^0 W \right) \\
&=: \sum_{i=1}^{6} S_{i,0}^k,
\end{aligned}
\tag{4.61}
$$

where

$$S^k_{1,0} := \sum_{j=1}^{k} \left(\int_{t^0_{j-1}}^{t^0_j} f_0(s, \phi_0(s))\, ds - h\, f_0(\delta^0_j, \phi_0(\delta^0_j)) \right),$$

$$S^k_{2,0} := h \sum_{j=1}^{k} \Big(f_0(\delta^0_j, \phi_0(\delta^0_j)) - f_0(\delta^0_j, \phi_0(t^0_{j-1})) \Big),$$

$$S^k_{3,0} := h \sum_{j=1}^{k} \Big(f_0(\delta^0_j, \phi_0(t^0_{j-1})) - f_0(\delta^0_j, y^0_{j-1}) \Big),$$

$$S^k_{4,0} := \int_0^{t^0_k} \sum_{j=1}^{N} \Big(g_0(s, \phi_0(s)) - g_0(t^0_{j-1}, \phi_0(s)) \Big) \mathbf{1}_{[t^0_{j-1}, t^0_j)}(s)\, dW_0(s),$$

$$S^k_{5,0} := \int_0^{t^0_k} \sum_{j=1}^{N} \Big(g_0(t^0_{j-1}, \phi_0(s)) - g_0(t^0_{j-1}, \phi_0(t^0_{j-1})) \Big) \mathbf{1}_{[t^0_{j-1}, t^0_j)}(s)\, dW_0(s),$$

$$S^k_{6,0} := \int_0^{t^0_k} \sum_{j=1}^{N} \Big(g_0(t^0_{j-1}, \phi_0(t^0_{j-1})) - g_0(t^0_{j-1}, y^0_{j-1}) \Big) \mathbf{1}_{[t^0_{j-1}, t^0_j)}(s)\, dW_0(s).$$

For $S^k_{1,0}, S^k_{2,0}, S^k_{4,0}$ and $S^k_{5,0}$, by Lemma 4.4.3 we have that

$$\Big\| \max_{1\le k\le N} |S^k_{1,0}| \Big\|_{L^p(\Omega)} \le c_p \tau^{1/2} K_f (1 + \sqrt[p]{K_{-1,p}})(1 + \sqrt[p]{K_{0,p}}) h^{1/2}, \tag{4.62}$$

$$\Big\| \max_{1\le k\le N} |S^k_{2,0}| \Big\|_{L^p(\Omega)} \le c_p L_f \tau K (\tau^{1/2} + 1)(1 + \sqrt[p]{K_{-1,p}} + \sqrt[p]{K_{0,p}}) h^{1/2}, \tag{4.63}$$

$$\Big\| \max_{1\le k\le N} |S^k_{4,0}| \Big\|_{L^p(\Omega)} \le \tilde{c}_{0,p} h^{\varrho}, \tag{4.64}$$

and

$$\Big\| \max_{1\le k\le N} |S^k_{5,0}| \Big\|_{L^p(\Omega)} \le \tilde{c}_{0,p} h^{1/2}. \tag{4.65}$$

Moreover, by (4.47) and Jensen's inequality (see Lemma 2.4.2) it holds that

$$\begin{aligned}
&\mathbb{E}\Big[\max_{1\le i\le k}|S^i_{3,0}|^p\Big]\\
&\le L_f^p\mathbb{E}\Big[\max_{1\le i\le k}\Big(\sum_{j=1}^{i}h|\phi_0(t^0_{j-1})-y^0_{j-1}|\Big)^p\Big]\\
&\le L_f^p\tau^{p-1}h\sum_{j=1}^{k}\mathbb{E}\Big[|\phi_0(t^0_{j-1})-y^0_{j-1}|^p\Big]\\
&\le L_f^p\tau^{p-1}h\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\phi_0(t^0_i)-y^0_i|^p\Big].
\end{aligned}\tag{4.66}$$

Regarding the last term $S^k_{6,0}$, it is convenient to introduce a notation that will also be used in the induction step. For any fixed $l\in[n]_0$ and $k\in[N]$, define

$$S^k_{6,l}:=\int_0^{t^0_k}\sum_{j=1}^{N}\Big(g_l(t^0_{j-1},\phi_l(t^0_{j-1}))-g_l(t^0_{j-1},\bar y^l_{j-1})\Big)\mathbf{1}_{[t^0_{j-1},t^0_j)}(s)\,\mathrm{d}W_l(s).\tag{4.67}$$

In the present case $l=0$ we have $\bar y^0=y^0$, hence (4.67) coincides with the term $S^k_{6,0}$ appearing in (4.86). Now set, for $j\in[N]$,

$$Q^l_{j-1}:=g_l(t^0_{j-1},\phi_l(t^0_{j-1}))-g_l(t^0_{j-1},\bar y^l_{j-1}),\tag{4.68}$$

and define the associated simple process

$$H_l(s):=\sum_{j=1}^{N}Q^l_{j-1}\,\mathbf{1}_{[t^0_{j-1},t^0_j)}(s),\qquad s\in[0,\tau].$$

Then (4.67) can be written as $S^k_{6,l}=\int_0^{t^0_k}H_l(s)\,\mathrm{d}W_l(s)$. Note that by (4.68), (4.9), and (4.42) we have

$$\sigma(Q^l_{j-1})\subset\tilde\Sigma_{t^l_{j-1}}=\Sigma^l_{t^0_{j-1}}\vee\sigma(\{\gamma^j_k\}_{j\in\mathbb{N}_0,k\in\mathbb{N}}).\tag{4.69}$$

Hence the process $H_l(s)$ is simple and $\tilde\Sigma_t$-predictable. Moreover, by (4.33) we have

$|Q^l_{j-1}| \le L_g|\phi_l(t^0_{j-1}) - \bar{y}^l_{j-1}|$. Hence, by Theorem 4.3.1 and Lemma 4.4.2,

$$\mathbb{E}\int_0^{t^l_N} |H_l(s)|^2\, ds = h\sum_{j=1}^{N} \mathbb{E}|Q^l_{j-1}|^2 < \infty.$$

Therefore, the Itô integral $S^k_{6,l} = \int_0^{t^0_k} H_l(s)\,\mathrm{d}W_l(s)$ is well-defined. Consequently, we may apply the BDG inequality (Corollary 2.4.5), together with Jensen's inequality (Lemma 2.4.2) and (4.33), to obtain

$$\begin{aligned}
\mathbb{E}\Big[\max_{1\le i\le k}|S^i_{6,l}|^p\Big] &\le \mathbb{E}\Big[\sup_{0\le t\le t^0_k}\Big|\int_0^t \sum_{j=1}^{k} Q^l_{j-1}\mathbf{1}_{[t^0_{j-1},t^0_j)}(s)\,\mathrm{d}W_l(s)\Big|^p\Big] \\
&\le c_p\,\mathbb{E}\Big(\int_0^{t^0_k}\sum_{j=1}^{k}|Q^l_{j-1}|^2\mathbf{1}_{[t^0_{j-1},t^0_j)}(s)\,\mathrm{d}s\Big)^{p/2} \\
&= c_p\,\mathbb{E}\Big(h\sum_{j=1}^{k}|Q^l_{j-1}|^2\Big)^{p/2} \\
&\le c_p\,\tau^{\frac{p}{2}-1}h\sum_{j=1}^{k}\mathbb{E}|Q^l_{j-1}|^p \\
&\le c_p\,L_g^p\,\tau^{\frac{p}{2}-1}h\sum_{j=1}^{k}\mathbb{E}|\phi_l(t^0_{j-1}) - \bar{y}^l_{j-1}|^p \\
&\le c_p\,L_g^p\,\tau^{\frac{p}{2}-1}h\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\phi_l(t^0_i) - \bar{y}^l_i|^p\Big].
\end{aligned} \tag{4.70}$$

Therefore,

$$\mathbb{E}\Big[\max_{1\le i\le k}|S^i_{6,0}|^p\Big] \le c_p\,L_g^p\,\tau^{\frac{p}{2}-1}h\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\phi_0(t^0_i) - y^0_i|^p\Big], \tag{4.71}$$

Hence, from (4.61), (4.66) and (4.71) we have for all $k \in [N]$

$$\begin{aligned}
&\mathbb{E}\Big[\max_{0\le i\le k}|\phi_0(t^0_i) - y^0_i|^p\Big] \\
&\le c_p\sum_{m\in\{1,2,4,5\}}\mathbb{E}\Big[\max_{1\le i\le N}|S^i_{m,0}|^p\Big] \\
&\quad + c_p(L_f^p\tau^{p-1} + L_g^p\tau^{p/2-1})h\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\phi_0(t^0_i) - y^0_i|^p\Big].
\end{aligned}$$

Now by using Gronwall's lemma (see, Lemma 2.4.7), and (4.62) to (4.71) we get for all $k \in [N]$

$$\begin{aligned}
&\mathbb{E}\Big[\max_{0\le i\le k}|\phi_0(t_i^0)-y_i^0|^p\Big]\\
&\le e^{c_p(L_f^p\tau^{p-1}+L_g^p\tau^{p/2-1})\tau}\sum_{m\in\{1,2,4,5\}}\mathbb{E}\Big[\max_{1\le i\le N}|S_{m,0}^i|^p\Big]\\
&\le \tilde{C}_{0,p}h^{p\min\{\varrho,\frac{1}{2}\}}.
\end{aligned}$$

Consequently,

$$\Big\|\max_{0\le i\le N}|\phi_0(t_i^0)-y_i^0|\Big\|_{L^p(\Omega)}\le C_{0,p}\,h^{\min\{\varrho,\frac{1}{2}\}}. \tag{4.72}$$

Since $\phi_0(t_k^0)=X(t_k^0)$ we get (4.59) for $l=0$.

Let us now assume that there exists $l\in\{0,1,\dots,n-1\}$ for which there exists $C_{l,p}\in(0,+\infty)$ such that for all $N\ge\lceil\tau\rceil$

$$\Big\|\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|\Big\|_{L^p(\Omega)}\le C_{l,p}h^{\min\{\varrho,\frac{1}{2}\}\alpha^l}. \tag{4.73}$$

The following error decomposition holds

$$\begin{aligned}
\Big\|\max_{0\le k\le N}|\phi_{l+1}(t_k^0)-y_k^{l+1}|\Big\|_{L^p(\Omega)}\le{}&\Big\|\max_{0\le k\le N}|\phi_{l+1}(t_k^0)-\bar{y}_k^{l+1}|\Big\|_{L^p(\Omega)}\\
&+\Big\|\max_{0\le k\le N}|\bar{y}_k^{l+1}-y_k^{l+1}|\Big\|_{L^p(\Omega)}.
\end{aligned} \tag{4.74}$$

Firstly, we estimate $\Big\|\max_{0\le k\le N}|\bar{y}_k^{l+1}-y_k^{l+1}|\Big\|_{L^p(\Omega)}$. For $k\in[N]$, by taking into account the fact that $y_0^{l+1}=\bar{y}_0^{l+1}$ from (4.38) we get

$$\begin{aligned}
\bar{y}_k^{l+1}-y_k^{l+1}&=\sum_{j=1}^k(\bar{y}_j^{l+1}-\bar{y}_{j-1}^{l+1})-\sum_{j=1}^k(y_j^{l+1}-y_{j-1}^{l+1})\\
&=h\sum_{j=1}^k\Big(f(\theta_j^{l+1},\bar{y}_{j-1}^{l+1},\phi_l(\delta_j^{l+1}))-f(\theta_j^{l+1},y_{j-1}^{l+1},y_{j-1}^l)\Big)\\
&\quad+\sum_{j=1}^k\Big(g(t_{j-1}^{l+1},\bar{y}_{j-1}^{l+1},\phi_l(t_{j-1}^0))-g(t_{j-1}^{l+1},y_{j-1}^{l+1},y_{j-1}^l)\Big)\Delta_{j-1}^{l+1}W,
\end{aligned}$$

where $\Delta_{j-1}^{l+1}W = W(t_j^{l+1}) - W(t_{j-1}^{l+1})$. This gives by the assumption (4.31) that

$$|\bar{y}_k^{l+1} - y_k^{l+1}|^p \le C_p L_f^p \Big(h\sum_{j=1}^{k} |\bar{y}_{j-1}^{l+1} - y_{j-1}^{l+1}| + h\sum_{j=1}^{k} |\phi_l(\delta_j^{l+1}) - y_{j-1}^{l}|^{\alpha_1} \Big)^p + c_p \Big| \sum_{j=1}^{k} \Big(g(t_{j-1}^{l+1}, \bar{y}_{j-1}^{l+1}, \phi_l(t_{j-1}^0)) - g(t_{j-1}^{l+1}, y_{j-1}^{l+1}, y_{j-1}^{l}) \Big) \Delta_{j-1}^{l+1} W \Big|^p.$$

Let us denote by

$$G_{j-1}^{l+1} = g(t_{j-1}^{l+1}, \bar{y}_{j-1}^{l+1}, \phi_l(t_{j-1}^0)) - g(t_{j-1}^{l+1}, y_{j-1}^{l+1}, y_{j-1}^{l}). \tag{4.75}$$

Hence, by Jensen's inequality (see Lemma 2.4.2), we get that for $k \in [N]$:

$$\begin{aligned} &\mathbb{E}\Big[\max_{0\le i\le k} |\bar{y}_i^{l+1} - y_i^{l+1}|^p\Big] \\ &\le L_f^p c_p \tau^{p-1} h \sum_{j=1}^{k} \mathbb{E}\Big[\max_{0\le i\le j-1} |\bar{y}_i^{l+1} - y_i^{l+1}|^p\Big] \\ &\quad + c_p L_f^p \tau^{p-1} h \sum_{j=1}^{k} \mathbb{E}\Big[|\phi_l(\delta_j^{l+1}) - y_{j-1}^{l}|^{\alpha_1 p}\Big] \\ &\quad + c_p \mathbb{E}\Big[\max_{1\le i\le k} \Big| \sum_{j=1}^{i} G_{j-1}^{l+1} \cdot \Delta_{j-1}^{l+1} W\Big|^p\Big]. \end{aligned} \tag{4.76}$$

Moreover,

$$\begin{aligned} \mathbb{E}\Big[|\phi_l(\delta_j^{l+1}) - y_{j-1}^{l}|^{\alpha_1 p}\Big] &= \mathbb{E}\Big[|\phi_l(\delta_j^{l+1}) - \phi_l(t_{j-1}^0) + \phi_l(t_{j-1}^0) - y_{j-1}^{l}|^{\alpha_1 p}\Big] \\ &\le \mathbb{E}\Big[2^{\alpha_1 p - 1}\Big(|\phi_l(\delta_j^{l+1}) - \phi_l(t_{j-1}^0)|^{\alpha_1 p} + |\phi_l(t_{j-1}^0) - y_{j-1}^{l}|^{\alpha_1 p}\Big)\Big] \\ &\le c_p\, \mathbb{E}\Big[|\phi_l(\delta_j^{l+1}) - \phi_l(t_{j-1}^0)|^{\alpha_1 p}\Big] + c_p\, \mathbb{E}\Big[\max_{0\le k\le N} |\phi_l(t_k^0) - y_k^{l}|^{\alpha_1 p}\Big]. \end{aligned}$$

Then, by Theorem 4.3.1 and Jensen's inequality (see Lemma 2.4.2) we have

$$\mathbb{E}[|\phi_l(\delta_j^{l+1}) - \phi_l(t_{j-1}^0)|^{\alpha_1 p}] \le \Big(\mathbb{E}[|\phi_l(\delta_j^{l+1}) - \phi_l(t_{j-1}^0)|^p]\Big)^{\alpha_1} \le (\bar{K}_{l,p})^{\alpha_1} h^{\alpha_1 p/2}, \tag{4.77}$$

and

$$\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^{\alpha_1 p}\Big]\le\left(\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^{p}\Big]\right)^{\alpha_1}. \tag{4.78}$$

Now define for all $t\in[0,t_N^{l+1}]$ the following stochastic process

$$M(t):=\int_0^t\sum_{j=1}^N G_{j-1}^{l+1}\cdot\mathbf{1}_{[t_{j-1}^{l+1},t_j^{l+1})}(s)\,\mathrm{d}W(s).$$

Note that by (4.75), (4.9), and (4.42) we have that

$$\sigma(G_{j-1}^{l+1})\subset\tilde{\Sigma}_{t_{j-1}^{l+1}}=\Sigma_{t_{j-1}^0}^{l+1}\vee\sigma(\{\gamma_k^j\}_{j\in\mathbb{N}_0,k\in\mathbb{N}}). \tag{4.79}$$

Moreover, we have

$$\mathbb{E}\int_0^{t_N^{l+1}}\Big|\sum_{j=1}^N G_{j-1}^{l+1}\mathbf{1}_{[t_{j-1}^{l+1},t_j^{l+1})}(s)\Big|^2 ds=h\sum_{j=1}^N\mathbb{E}|G_{j-1}^{l+1}|^2. \tag{4.80}$$

By (4.33) we have

$$|G_{j-1}^{l+1}|\le L_g\Big(|\bar{y}_{j-1}^{l+1}-y_{j-1}^{l+1}|+|\phi_l(t_{j-1}^0)-y_{j-1}^l|^{\alpha_2}\Big).$$

Hence, by Theorem 4.3.1 and Lemma 4.4.2, $\mathbb{E}|G_{j-1}^{l+1}|^2<\infty$ and therefore (4.80) is finite. Therefore, the stochastic Itô integral above is well-defined as a stochastic integral of the simple process

$$\Big(G_{j-1}^{l+1}\cdot\mathbf{1}_{[t_{j-1}^{l+1},t_j^{l+1})}(s)\Big)_{s\ge0} \tag{4.81}$$

that is adapted to the filtration (4.6), with respect to which W is also a Wiener process. Moreover, the quadratic variation of the martingale M for $t\in[0,t_N^{l+1}]$ is as follows

$$\langle M\rangle(t)=\int_0^t\sum_{j=1}^N|G_{j-1}^{l+1}|^2\cdot\mathbf{1}_{[t_{j-1}^{l+1},t_j^{l+1})}(s)\,\mathrm{d}s$$

and then for $k \in [N]$

$$\langle M \rangle (t_k^{l+1}) = h \sum_{j=1}^{k} |G_{j-1}^{l+1}|^2.$$

Following the martingale estimate used in [74], from BDG inequality (Corollary 2.4.5), Jensen's inequality (Lemma 2.4.2) and the assumption (4.33) on g, we have that

$$\begin{aligned}
&\mathbb{E}\left[\max_{1\le i\le k}\left|\sum_{j=1}^{i} G_{j-1}^{l+1}\cdot\Delta_{j-1}^{l+1}W\right|^p\right] \\
&\le c_p\mathbb{E}\left[\left(\sum_{j=1}^{k}\left|G_{j-1}^{l+1}\cdot\Delta_{j-1}^{l+1}W\right|^2\right)^{p/2}\right] \\
&\le c_p\mathbb{E}\left[\left(\sum_{j=1}^{k}|G_{j-1}^{l+1}|^2\cdot|\Delta_{j-1}^{l+1}W|^2\right)^{p/2}\right] \\
&\le c_p\tau^{\frac{p}{2}-1}h\sum_{j=1}^{k}\mathbb{E}\left[|G_{j-1}^{l+1}|^p\right] \\
&\le c_p\tau^{\frac{p}{2}-1}L_g^p h\sum_{j=1}^{k}\mathbb{E}\left[\max_{0\le i\le j-1}|\bar{y}_i^{l+1}-y_i^{l+1}|^p\right] \\
&\quad + c_p\tau^{\frac{p}{2}}L_g^p\Big(\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^p\Big]\Big)^{\alpha_2}.
\end{aligned} \tag{4.82}$$

Combining (4.76), (4.77), (4.78) and (4.82) we arrive at for $k \in [N]$

$$\begin{aligned}
&\mathbb{E}\Big[\max_{0\le i\le k}|\bar{y}_i^{l+1}-y_i^{l+1}|^p\Big] \\
&\le c_p(L_f^p\tau^{p-1}+L_g^p\tau^{p/2-1})h\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\bar{y}_i^{l+1}-y_i^{l+1}|^p\Big] \\
&\quad + (\bar{K}_{l,p})^{\alpha_1}h^{\alpha_1 p/2} + c_p\left(\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^p\Big]\right)^{\alpha_1} \\
&\quad + c_p\tau^{p/2}L_g^p\left(\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^p\Big]\right)^{\alpha_2}.
\end{aligned} \tag{4.83}$$

By the discrete version of Gronwall's lemma (see Lemma 2.4.7) we get

$$\begin{aligned}
&\mathbb{E}\Big[\max_{0\le k\le N}|\bar{y}_k^{l+1}-y_k^{l+1}|^p\Big]\\
&\le \tilde{c}_{l+1,p}e^{c_p(L_f^p\tau^{p-1}+L_g^p\tau^{p/2-1})\tau}\Bigg[h^{\alpha_1 p/2}+\Big(\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^p\Big]\Big)^{\alpha_1}\\
&\qquad+\Big(\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^p\Big]\Big)^{\alpha_2}\Bigg].
\end{aligned}\tag{4.84}$$

We now establish an upper bound on $\Big\|\max_{0\le i\le N}|\phi_{l+1}(t_i^0)-\bar{y}_i^{l+1}|\Big\|_{L^p(\Omega)}$. For $k\in[N]$ by definition (4.86) we have

$$\begin{aligned}
&\phi_{l+1}(t_k^0)-\bar{y}_k^{l+1}\\
&=\phi_{l+1}(0)-\bar{y}_0^{l+1}+(\phi_{l+1}(t_k^0)-\phi_{l+1}(t_0^0))-(\bar{y}_k^{l+1}-\bar{y}_0^{l+1})\\
&=(\phi_l(t_N^0)-y_N^l)+\sum_{j=1}^k(\phi_{l+1}(t_j^0)-\phi_{l+1}(t_{j-1}^0))-\sum_{j=1}^k(\bar{y}_j^{l+1}-\bar{y}_{j-1}^{l+1})\\
&=(\phi_l(t_N^0)-y_N^l)+\sum_{j=1}^k\Bigg(\int_{t_{j-1}^0}^{t_j^0}f_{l+1}(s,\phi_{l+1}(s))\,\mathrm{d}s-hf_{l+1}(\delta_j^{l+1},\bar{y}_{j-1}^{l+1})\Bigg)\\
&\quad+\sum_{j=1}^k\Bigg(\int_{t_{j-1}^0}^{t_j^0}g_{l+1}(s,\phi_{l+1}(s))\,\mathrm{d}W_{l+1}(s)-g_{l+1}(t_{j-1}^0,\bar{y}_{j-1}^{l+1})\Delta_{j-1}^{l+1}W\Bigg)\\
&=(\phi_l(t_N^0)-y_N^l)+\sum_{i=1}^6 S_{i,l+1}^k.
\end{aligned}\tag{4.85}$$

where

$$
\begin{aligned}
S^k_{1,l+1} &= \sum_{j=1}^{k}\Bigg(\int_{t^0_{j-1}}^{t^0_j} f_{l+1}(s,\phi_{l+1}(s))\,\mathrm{d}s - hf_{l+1}(\delta^{l+1}_j,\phi_{l+1}(\delta^{l+1}_j))\Bigg),\\
S^k_{2,l+1} &= h\sum_{j=1}^{k}\Big(f_{l+1}(\delta^{l+1}_j,\phi_{l+1}(\delta^{l+1}_j)) - f_{l+1}(\delta^{l+1}_j,\phi_{l+1}(t^0_{j-1}))\Big),\\
S^k_{3,l+1} &= h\sum_{j=1}^{k}\Big(f_{l+1}(\delta^{l+1}_j,\phi_{l+1}(t^0_{j-1})) - f_{l+1}(\delta^{l+1}_j,\bar{y}^{l+1}_{j-1})\Big),\\
S^k_{4,l+1} &= \int_0^{t^0_k}\sum_{j=1}^{N}\left(g_{l+1}(s,\phi_{l+1}(s)) - g_{l+1}(t^0_{j-1},\phi_{l+1}(s))\right)\mathbf{1}_{[t^0_{j-1},t^0_j)}(s)\,\mathrm{d}W_{l+1}(s),\\
S^k_{5,l+1} &= \int_0^{t^0_k}\sum_{j=1}^{N}\left(g_{l+1}(t^0_{j-1},\phi_{l+1}(s)) - g_{l+1}(t^0_{j-1},\phi_{l+1}(t^0_{j-1}))\right)\mathbf{1}_{[t^0_{j-1},t^0_j)}(s)\,\mathrm{d}W_{l+1}(s),\\
S^k_{6,l+1} &= \int_0^{t^0_k}\sum_{j=1}^{N}\left(g_{l+1}(t^0_{j-1},\phi_{l+1}(t^0_{j-1})) - g_{l+1}(t^0_{j-1},\bar{y}^{l+1}_{j-1})\right)\mathbf{1}_{[t^0_{j-1},t^0_j)}(s)\,\mathrm{d}W_{l+1}(s).
\end{aligned}
\tag{4.86}
$$

For $S^k_{1,l+1}, S^k_{2,l+1}, S^k_{4,l+1}$ and $S^k_{5,l+1}$, by Lemma 4.4.3 we have that

$$
\Big\|\max_{1\le k\le N}|S^k_{1,l+1}|\Big\|_{L^p(\Omega)} \le c_p\tau^{1/2}K_f(1+K^{1/p}_{l,p})(1+K^{1/p}_{l+1,p})h^{1/2}, \tag{4.87}
$$

$$
\Big\|\max_{1\le k\le N}|S^k_{2,l+1}|\Big\|_{L^p(\Omega)} \le c_pL_f\tau K(\tau^{1/2}+1)(1+K^{1/p}_{l,p}+K^{1/p}_{l+1,p})h^{1/2}, \tag{4.88}
$$

$$
\Big\|\max_{1\le k\le N}|S^k_{4,l+1}|\Big\|_{L^p(\Omega)} \le \tilde{c}_{l+1,p}h^{\min\{\varrho,\alpha_2/2\}}, \tag{4.89}
$$

and

$$
\Big\|\max_{1\le k\le N}|S^k_{5,l+1}|\Big\|_{L^p(\Omega)} \le \tilde{c}_{l+1,p}h^{1/2}. \tag{4.90}
$$

Moreover by (4.31) and Jensen's Inequality (see Lemma 2.4.2) we have,

$$\begin{aligned}
&\mathbb{E}\Big[\max_{1\le i\le k}|S^i_{3,l+1}|^p\Big]\\
&\le L_f^p\mathbb{E}\Big[\max_{1\le i\le k}\Big(\sum_{j=1}^{i}h|\phi_{l+1}(t^0_{j-1})-\bar y^{l+1}_{j-1}|\Big)^p\Big]\\
&\le L_f^p\tau^{p-1}h\sum_{j=1}^{k}\mathbb{E}\Big[|\phi_{l+1}(t^0_{j-1})-\bar y^{l+1}_{j-1}|^p\Big]\\
&\le L_f^p\tau^{p-1}h\Big(\mathbb{E}|\phi_l(t^0_N)-y^l_N|^p+\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\phi_{l+1}(t^0_i)-\bar y^{l+1}_i|^p\Big]\Big).
\end{aligned}\tag{4.91}$$

For the last term $S^k_{6,l+1}$, by (4.70) (applied with l replaced by $l+1$), we obtain for $k\in[N]$ that

$$\begin{aligned}
\mathbb{E}\Big[\max_{1\le i\le k}|S^i_{6,l+1}|^p\Big]&\le c_pL_g^p\,\tau^{\frac{p}{2}-1}h\sum_{j=1}^{k}\mathbb{E}|\phi_{l+1}(t^0_{j-1})-\bar y^{l+1}_{j-1}|^p\\
&=c_pL_g^p\,\tau^{\frac{p}{2}-1}h\,\mathbb{E}|\phi_{l+1}(t^0_0)-\bar y^{l+1}_0|^p\\
&\quad+c_pL_g^p\,\tau^{\frac{p}{2}-1}h\sum_{j=2}^{k}\mathbb{E}|\phi_{l+1}(t^0_{j-1})-\bar y^{l+1}_{j-1}|^p\\
&\le c_pL_g^p\,\tau^{\frac{p}{2}-1}h\,\mathbb{E}|\phi_l(t^0_N)-y^l_N|^p\\
&\quad+c_pL_g^p\,\tau^{\frac{p}{2}-1}h\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\phi_{l+1}(t^0_i)-\bar y^{l+1}_i|^p\Big].
\end{aligned}\tag{4.92}$$

Hence, from (4.85), (4.91) and (4.92) we have for $k\in\{1,2,\dots,N\}$

$$\begin{aligned}
&\mathbb{E}\Big[\max_{0\le i\le k}|\phi_{l+1}(t^0_i)-\bar y^{l+1}_i|^p\Big]\\
&\le c_p(L_f^p\tau^{p-1}+L_g^p\tau^{p/2-1})h\mathbb{E}|\phi_l(t^0_N)-y^l_N|^p\\
&\quad+c_p\sum_{m\in\{1,2,4,5\}}\mathbb{E}\Big[\max_{1\le k\le N}|S^k_{m,l+1}|^p\Big]\\
&\quad+c_p(L_f^p\tau^{p-1}+L_g^p\tau^{p/2-1})h\sum_{j=1}^{k-1}\mathbb{E}\Big[\max_{0\le i\le j}|\phi_{l+1}(t^0_i)-\bar y^{l+1}_i|^p\Big].
\end{aligned}$$

Now by using Gronwall's lemma (see Lemma 2.4.7), and (4.87) to (4.90) we get for all

$k \in [N]$

$$\begin{aligned}
&\mathbb{E}\Big[\max_{0\le i\le k}|\phi_{l+1}(t_i^0)-\bar{y}_i^{l+1}|^p\Big]\\
&\le c_p(L_f^p\tau^{p-1}+L_g^p\tau^{p/2-1})e^{c_p(L_f^p\tau^{p-1}+L_g^p\tau^{p/2-1})\tau}\\
&\qquad\times\Big(\mathbb{E}|\phi_l(t_N^0)-y_N^l|^p+\sum_{m\in\{1,2,4,5\}}\mathbb{E}\Big[\max_{1\le k\le N}|S_{m,l+1}^k|^p\Big]\Big),
\end{aligned}$$

which gives

$$\Big\|\max_{0\le i\le N}|\phi_{l+1}(t_i^0)-\bar{y}_i^{l+1}|\Big\|_{L^p(\Omega)}^p\le\tilde{c}_{l+1,p}\Big(\mathbb{E}\Big[\max_{0\le k\le N}|\phi_l(t_k^0)-y_k^l|^p\Big]+h^{p\min\{\varrho,\alpha_2/2\}}\Big).\quad(4.93)$$

Combining (4.74), (4.84), and (4.93) we obtain

$$\begin{aligned}
&\Big\|\max_{0\le i\le N}|\phi_{l+1}(t_i^0)-y_i^{l+1}|\Big\|_{L^p(\Omega)}\\
&\le\tilde{c}_{l+1,p}\Big(h^{\min\{\varrho,\alpha_1/2,\alpha_2/2\}}+\Big\|\max_{0\le i\le N}|\phi_l(t_i^0)-y_i^l|\Big\|_{L^p(\Omega)}\\
&\qquad+\Big\|\max_{0\le i\le N}|\phi_l(t_i^0)-y_i^l|\Big\|_{L^p(\Omega)}^{\alpha_1}+\Big\|\max_{0\le i\le N}|\phi_l(t_i^0)-y_i^l|\Big\|_{L^p(\Omega)}^{\alpha_2}\Big),
\end{aligned}$$

and therefore by (4.73) we have,

$$\Big\|\max_{0\le i\le N}|\phi_{l+1}(t_i^0)-y_i^{l+1}|\Big\|_{L^p(\Omega)}\le C_{l+1,p}h^{\min\{\varrho,\frac{1}{2}\}\alpha^{l+1}},\quad(4.94)$$

which ends the inductive part of the proof. Finally, $\phi_{l+1}(t_i^0)=\phi_{l+1}(ih)=X(ih+(l+1)\tau)=X(t_i^{l+1})$ and the proof of (4.59) is finished. □

Remark 4.4.6. *We briefly comment on optimality of the defined algorithm in the Information-Based Complexity sense, see [54]. In this context, Information-Based Complexity studies the best possible convergence rate that any numerical method can achieve when it is allowed to use only a prescribed amount of information about the coefficients, such as finitely many point evaluations. Thus, optimality means that the error rate cannot in general be improved by another method using the same type and order of available information. In the special case $\alpha_1=\alpha_2=1$ we have the optimal bound $\Theta(h^{\min\{\varrho,1/2\}})$, which follows from Proposition 5.1 in [86].*

Remark 4.4.7. *In general, for fixed τ and p, the coefficient $C_{l,p}$ in* (4.59) *strictly increases with l while the convergence order* $\min\{\varrho, \frac{1}{2}\}\alpha^l$ *decreases. Such a trend is a consequence of the error propagation used in the proof of Theorem 4.4.4. For a larger but finite l, the error can grow extremely large over $[l\tau, (l+1)\tau]$, where the error bound may become too large to be informative. With this nature, the theoretical result (Theorem 4.4.4) can not be extended to infinite time horizon case.*

4.5 Numerical experiments

In order to illustrate our theoretical findings we perform several numerical experiments. We chose the following exemplary right-hand side functions for $i \in \{1, 2\}$:

$$f_i(t, x, z) = k_i(t)\Big(x + 0.01|z|^{\alpha_1} + \sin(10x) \cdot \cos(100|z|^{\alpha_1})\Big), \tag{4.95}$$

and the diffusion is either additive

$$g_1(t, x, z) = 0.5|\cos(2^5 \pi t)|, \tag{4.96}$$

or multiplicative

$$g_2(t, x, z) = k(t)\Big(x + 0.01|z|^{\alpha_2} + \cos(10x) \cdot \cos(100|z|^{\alpha_2})\Big), \tag{4.97}$$

where, $\alpha_1, \alpha_2 \in (0, 1]$, k_1 is the following periodic function

$$k_1(t) = \sum_{j=0}^{n} \Big((j+1)\tau - t\Big)^{-1/\gamma_1} \cdot \mathbf{1}_{[j\tau,(j+1)\tau]}(t), \quad \gamma_1 > 2, \tag{4.98}$$

which belongs to $L^p([0, (n+1)\tau])$, k_2 is a step function satisfying

$$k_2(t) = \sum_{j=0}^{n} 0.1 \cdot (j+1) \cdot \mathbf{1}_{[j\tau,(j+1)\tau]}(t), \tag{4.99}$$

and k is given by

$$k(t) = t^{\gamma_2}, \quad \gamma_2 \in (0, 1], \tag{4.100}$$

which is γ_2-Hölder continuous. Note that formally f_1 does not satisfy our assumptions; nevertheless, the numerical evidence suggests that the result may extend to settings where the time-dependent bound in the drift is integrable rather than uniformly bounded.

Remark 4.5.1 (Relation of the SDDE experiments to Theorem 4.4.4)**.** *The SDDE experiments below have two roles. Example 4.5.2 and the final multiplicative-noise example use the singular time factor k_1 and are therefore outside Assumption **(A3')**; they are included as stress tests for irregular time dependence. The second experiment uses the drift formula* (4.95) *with the bounded step factor k_2 and the Hölder continuous diffusion factor $k(t) = t^{\gamma_2}$, and is the numerical test directly related to Theorem 4.4.4.*

By fixing the value of $N \in \mathbb{N}$, we implement randomised Euler-Maruyama scheme (4.7)-(4.8) at step size $h = \tau/N \in (0, 1)$ using Python programming language. Moreover, since for the right-hand side functions (4.95) plus (4.96) or (4.95) plus (4.97) we do not know the exact solution $X(t)$, the reference solution $\tilde{y}$ is obtained via randomised Euler-Maruyama scheme at a refined step size $\tilde{h} = h/m$ for $m \gg 1$, $m \in \mathbb{N}$. We approximate the mean square error (4.10) for each $0 \leq j \leq n$ with

$$\left\| \max_{0\leq i\leq N} |\tilde{y}_i^j - y_i^j| \right\|_{L^2(\Omega)} \approx \left(\frac{1}{K} \sum_{k=1}^{K} \max_{0\leq i\leq N} |\tilde{y}_i^j(\omega_k) - y_i^j(\omega_k)|^2 \right)^{\frac{1}{2}},$$

where $K \in \mathbb{N}$, $\{\omega_k\}_{k=1}^K$ represents the kth realisation from the complete probability space, y_i^j is the output of the randomised Euler-Maruyama scheme on the initial mesh $t_i^j := j\tau + ih$ for $i = 0, \ldots, N-1$, while $\tilde{y}_i^j$ is the reference solution evaluated at t_i^j. Note that $t_i^j = j\tau + im\tilde{h}$ on the refined mesh.

The implementation of the randomised Euler-Maruyama scheme is straightforward. To evaluate the solution at time point $t_i^j = ih + j\tau$ within the interval $[j\tau, (j+1)\tau]$, we need two steps:

Step (j1): First simulate $\gamma \sim \mathcal{U}(0, 1)$ and set random time point $t_{i-1}^j + \gamma h \in [t_{i-1}^j, t_i^j]$.

Step (j2): Compute y_i^j as defined in (4.7) and (4.8).

Listing 4.1 shows an implementation of scheme (4.7) and (4.8) in the case of a 1-dimensional Wiener process ($m = 1$) in PYTHON.

Listing 4.1: A sample implementation of (4.7) and (4.8) in PYTHON

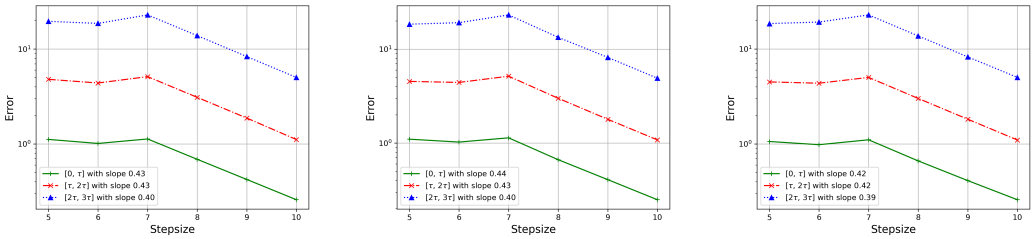
```
import numpy as np

def f(t,x,z):
    return [...]

def g(t,x,z):
    return [...]



def randEM_full(tau, X0, h, f,g,n_taus):

    # input:     delay lag tau, stepsize h, initials X0,
    #             drift and diffusion functions f and g
    #             number of intervals of length tau n_taus
    # output:
    #  one trajectory of the randomized Euler-Maruyama method

    ###to get numerical evaluation:

    #number of steps within one interval with length tau
    N=int(tau/stepsize)

    #setting initial conditions
    sol = np.zeros((N+ 1,n_tau+1))+X0

    #for each interval [j*tau, (j+1)*tau]
    for j in range(1, n_taus+1):
```

```
        sol[0,j] = sol[-1,j-1]

        # collect the grid points over [j*tau, (j+1)*tau]
        grid = j* tau + h *np.arange(0, N)


        for i in range(1,N+1):

            # step (j1):
            gamma=np.random.rand()
            rand_time=grid[i-1]+gamma*h

            # step (j2):
            drift=h*f(rand_time, sol[i-1,j], sol[i-1,j-1])
            diff=g(grid[i-1], sol[i-1,j], sol[i-1,j-1]) \
                    *np.sqrt(h)*np.random.normal()

            sol[i,j] = sol[i-1,j]+drift+diff


    return sol
```

Example 4.5.2 (Additive noise)**.** In the following numerical test we use (4.95) with (4.98) and (4.96), generalised from [27, Example 5.1] of an ordinary DDE case. In this example, we formally have $\alpha_2 = 1$, which allows us to examine in the stochastic setting the numerical effect of α_1 alone, i.e., the regularity of the delay term in the drift. Due to the irregularity of f with respect to t, Assumption **(A3')**, and hence Theorem 4.4.4, does not apply, though such a condition is covered in [27, Assumption **(A3')**] for the ordinary DDE case. Through this example, we illustrate the behaviour of the

Table 4.2: Negative mean square error slopes and corresponding figures for (4.95) plus (4.96).

γ_1	α_1	$[0,\tau]$	$[1\tau,2\tau]$	$[2\tau,3\tau]$	Figure
3	0.1	0.43	0.43	0.40	Figure 4.1a
	0.5	0.44	0.43	0.40	Figure 4.1b
	1	0.42	0.42	0.39	Figure 4.1c
5	0.1	0.39	0.40	0.39	Figure 4.2a
	0.5	0.39	0.41	0.40	Figure 4.2b
	1	0.40	0.41	0.39	Figure 4.2c



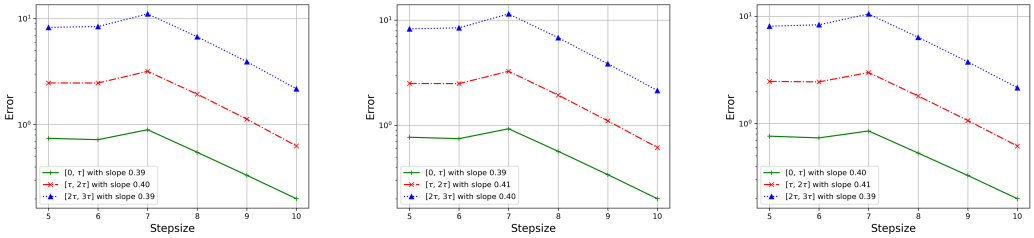
(a) $\alpha_1 = 0.1$. (b) $\alpha_1 = 0.5$. (c) $\alpha_1 = 1$.

Figure 4.1: Mean square error slopes for $\gamma_1 = 3$ and values of $\alpha_1 = 0.1, 0.5, 1$ for (4.95) plus (4.96).

proposed scheme outside the assumptions of the present theorem. We fix the number of experiments $K = 1000$ for each $N = 2^l$, $l = 5, \ldots, 10$, and the reference solution is computed with step size 2^{-17}; also, the horizon parameter is $n = 3$. We present the results in Table 4.2, which illustrate the negative mean square error slopes for different combinations of $\gamma_1 = 3, 5$ and $\alpha_1 = 0.1, 0.5, 1$.

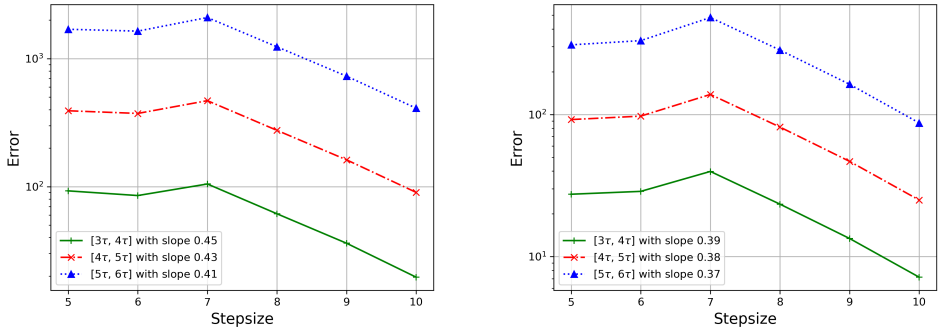
All the figures show that the error of $[j\tau, (j+1)\tau]$ is worse than the error of $[(j-1)\tau, j\tau]$ for $j \in \{1, 2\}$, although the deterioration is not as pronounced as the delay-interval deterioration predicted by Theorem 4.4.4 in the theorem-covered setting.

To illustrate that the accumulated error over later delay intervals becomes more visible, we plot in Figure 4.3 the order of convergence over $[3\tau, 4\tau]$, $[4\tau, 5\tau]$ and $[5\tau, 6\tau]$ for the case when $\alpha_1 = 0.1$ and $\gamma_1 = 3, 5$. One can observe that the convergence rate further reduces compared to the corresponding convergence rate over $[0, \tau]$, $[\tau, 2\tau]$ and $[2\tau, 3\tau]$ (shown in Figure 4.1a and Figure 4.2a), and the error grows to a scale of $10^2 - 10^3$ over $[5\tau, 6\tau]$.



(a) $\alpha_1 = 0.1$. (b) $\alpha_1 = 0.5$. (c) $\alpha_1 = 1$.

Figure 4.2: Mean square error slopes for $\gamma_1 = 5$ and values of $\alpha_1 = 0.1, 0.5, 1$ for (4.95) plus (4.96).



(a) $\gamma_1 = 3$. (b) $\gamma_1 = 5$.

Figure 4.3: Mean square error slopes over $[3\tau, 4\tau]$, $[4\tau, 5\tau]$ and $[5\tau, 6\tau]$ for $\gamma_1 = 3, 5$ and $\alpha_1 = 0.1$ for (4.95) plus (4.96).

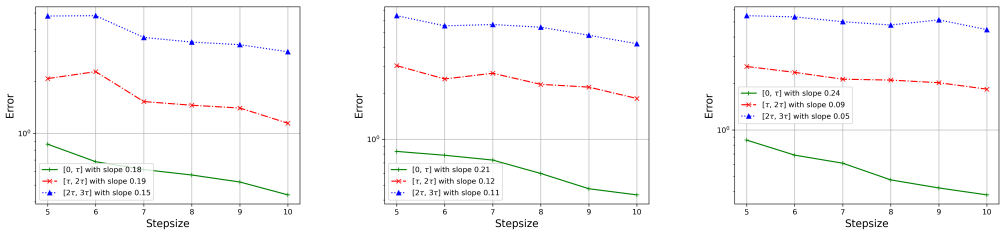
Table 4.3: Negative mean square error slopes for (4.95) with (4.99) plus (4.97) at $\gamma_2 = 0.1, 0.5, 1$ and values of $\alpha_1 = \alpha_2 = \alpha = 0.1, 0.5, 1$.

γ_2	α	$[0, \tau]$	$[1\tau, 2\tau]$	$[2\tau, 3\tau]$	Figure
	0.1	0.18	0.19	0.15	Figure 4.4a
0.1	0.5	0.21	0.12	0.11	Figure 4.4b
	1	0.24	0.09	0.05	Figure 4.4c
	0.1	0.21	0.19	0.12	Figure 4.4d
0.5	0.5	0.24	0.11	0.08	Figure 4.4e
	1	0.28	0.09	0.07	Figure 4.4f
	0.1	0.23	0.17	0.23	Figure 4.4g
1	0.5	0.24	0.16	0.14	Figure 4.4h
	1	0.22	0.01	0.09	Figure 4.4i

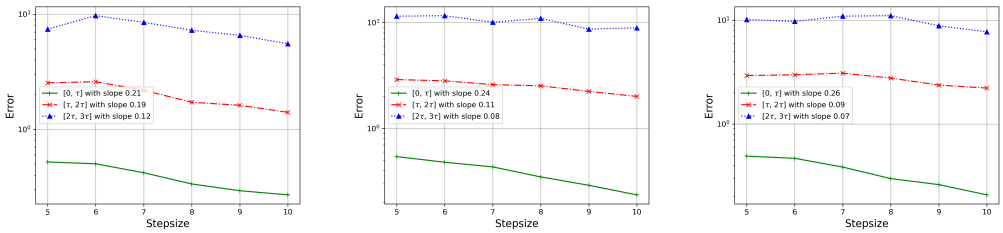
Example 4.5.3 (Multiplicative noise)**.** To illustrate the rate behaviour predicted by Theorem 4.4.4 in the general setting where α_1 and α_2 are both nontrivial, in the following numerical test we use (4.95) with (4.99) and (4.97). We fix the number of experiments $K = 1000$ for each $N = 2^l$, $l = 5, \ldots, 10$, and the reference solution is computed with step size 2^{-17}; also, the horizon parameter is $n = 3$. We vary γ_2, α_1, but allow the same value of α_2 as α_1, i.e., $\alpha_1 = \alpha_2 = \alpha$. We present the results in Table 4.3, which illustrate the negative mean square error slopes for different combinations of $\gamma_2 = 0.1, 0.5, 1$ and $\alpha_1 = \alpha_2 = \alpha = 0.1, 0.5, 1$.

One can compare the results horizontally or vertically in Figure 4.4. For each fixed (α_1, α_2) pair and for each fixed γ_2, the negative slope in general decreases with j. Horizontally, each fixed γ_2 and for each fixed $[j\tau, (j+1)\tau]$ interval, the error decreases significantly with increasing α_1. Vertically, for each fixed (α_1, α_2) pair and for each fixed $[j\tau, (j+1)\tau]$ interval, the numerical error decreases with γ_2. These observations are broadly consistent with the qualitative dependence predicted by Theorem 4.4.4.

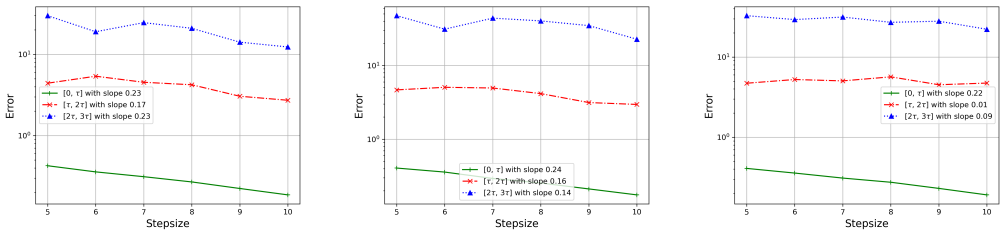
Example 4.5.4 (Multiplicative noise)**.** Finally, we extend Example 4.5.2 to the multiplicative noise case by replacing (4.96) with (4.97). As in Example 4.5.2, Assumption **(A3')**, and hence Theorem 4.4.4, does not apply here. We use this example to test the behaviour of the proposed scheme in this more irregular setting. We fix the number of experiments $K = 1000$ for each $N = 2^l$, $l = 5, \ldots, 10$, and the reference solution is



(a) $\gamma_2 = 0.1$, and $\alpha = 0.1$. (b) $\gamma_2 = 0.1$ and $\alpha = 0.5$. (c) $\gamma_2 = 0.1$ and $\alpha = 1$.



(d) $\gamma_2 = 0.5$, and $\alpha = 0.1$. (e) $\gamma_2 = 0.5$ and $\alpha = 0.5$. (f) $\gamma_2 = 0.5$ and $\alpha = 1$.



(g) $\gamma_2 = 1$ and $\alpha = 0.1$. (h) $\gamma_2 = 1$ and $\alpha = 0.5$. (i) $\gamma_2 = 1$ and $\alpha = 1$.

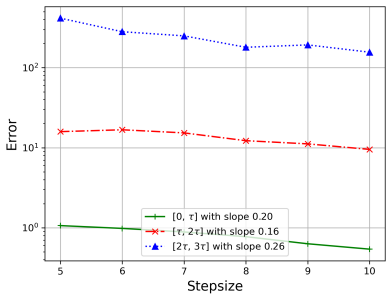
Figure 4.4: Mean square error slopes for (4.95) with (4.99) plus (4.97) at $\gamma_2 = 0.1, 0.5, 1$ and values of $\alpha_1 = \alpha_2 = \alpha = 0.1, 0.5, 1$.

Table 4.4: Negative mean square error slopes for $\gamma_1 = 5$ and $\gamma_2 = 0.1, 0.5, 1$ and values of $(\alpha_1, \alpha_2) = (0.1, 0.1), (1, 0.1), (0.1, 1), (0.5, 0.5)$ for (4.95) plus (4.97).

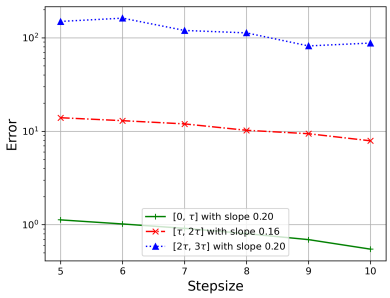
γ_1	γ_2	α_1	α_2	$[0, \tau]$	$[1\tau, 2\tau]$	$[2\tau, 3\tau]$	Figure
5	0.1	0.1	0.1	0.20	0.16	0.26	Figure 4.5a
		1	0.1	0.20	0.16	0.20	Figure 4.5b
		0.1	1	0.21	0.07	0.21	Figure 4.5c
		0.5	0.5	0.20	0.19	0.23	Figure 4.5d
5	0.5	0.1	0.1	0.18	0.18	0.17	Figure 4.6a
		1	0.1	0.21	0.17	0.23	Figure 4.6b
		0.1	1	0.17	0.14	0.17	Figure 4.6c
		0.5	0.5	0.19	0.15	0.16	Figure 4.6d
5	1	0.1	0.1	0.19	0.23	0.27	Figure 4.7a
		1	0.1	0.21	0.33	0.25	Figure 4.7b
		0.1	1	0.18	0.18	0.21	Figure 4.7c
		0.5	0.5	0.21	0.27	0.36	Figure 4.7d

computed with step size 2^{-17}; also, the horizon parameter is $n = 3$. We fix $\gamma_1 = 5$ and vary γ_2, α_1 and α_2. We present the results in Table 4.4, which illustrate the negative mean square error slopes for different combinations of $\gamma_1 = 5$, $\gamma_2 = 0.1, 0.5, 1$ and $(\alpha_1, \alpha_2) = (0.1, 0.1), (1, 0.1), (0.1, 1), (0.5, 0.5)$.

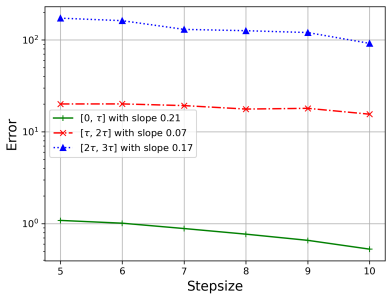
It can be observed that for each fixed (α_1, α_2) pair and for each fixed $[j\tau, (j+1)\tau]$ interval, the order of convergence increases with γ_2; for each fixed γ_2 and for each fixed $[j\tau, (j+1)\tau]$ interval, the negative slopes for $(\alpha_1, \alpha_2) = (0.1, 0.1), (1, 0.1), (0.1, 1)$ are almost the same, and are slightly lower than the negative slope for $(\alpha_1, \alpha_2) = (0.5, 0.5)$; for each fixed (α_1, α_2) pair and for each fixed γ_2, the negative slope decreases with j. Although this example is outside the assumptions of Theorem 4.4.4, these observations mirror, at an empirical level, the same qualitative dependence on temporal regularity, Hölder exponents and the delay interval.



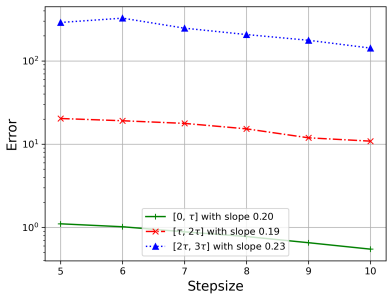
(a) $\alpha_1 = 0.1$ and $\alpha_2 = 0.1$.



(b) $\alpha_1 = 1$ and $\alpha_2 = 0.1$.

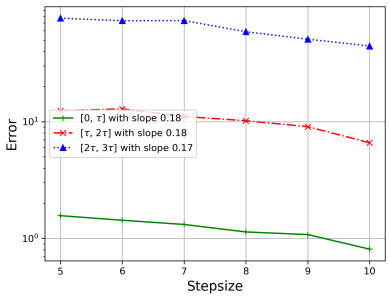


(c) $\alpha_1 = 0.1$ and $\alpha_2 = 1$.

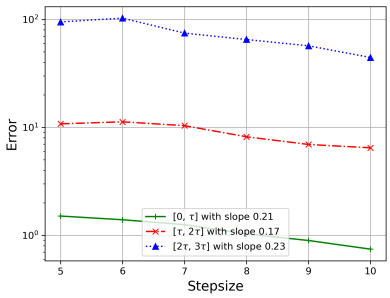


(d) $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$.

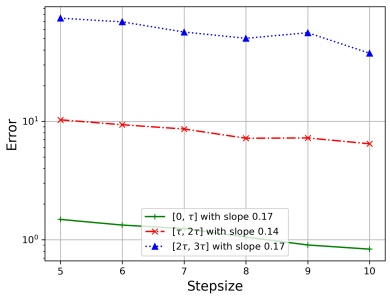
Figure 4.5: Mean square error slopes for $\gamma_1 = 5$ and $\gamma_2 = 0.1$ and values of $(\alpha_1, \alpha_2) = (0.1, 0.1), (1, 0.1), (0.1, 1), (0.5, 0.5)$ for (4.95) plus (4.97).



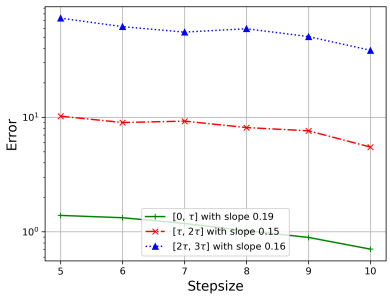
(a) $\alpha_1 = 0.1$ and $\alpha_2 = 0.1$.



(b) $\alpha_1 = 1$ and $\alpha_2 = 0.1$.

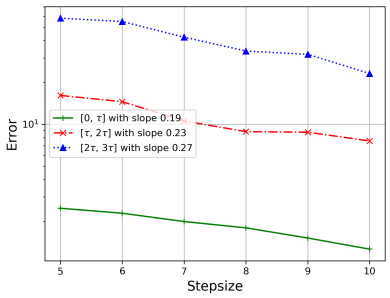


(c) $\alpha_1 = 0.1$ and $\alpha_2 = 1$.

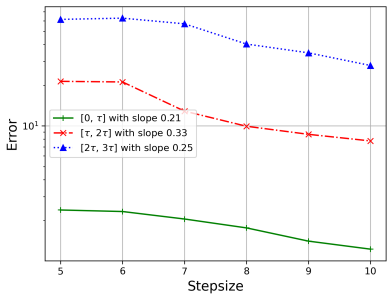


(d) $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$.

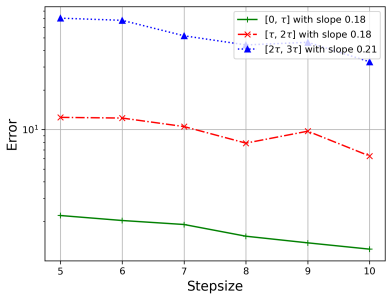
Figure 4.6: Mean square error slopes for $\gamma_1 = 5$ and $\gamma_2 = 0.5$ and values of $(\alpha_1, \alpha_2) = (0.1, 0.1), (1, 0.1), (0.1, 1), (0.5, 0.5)$ for (4.95) plus (4.97).



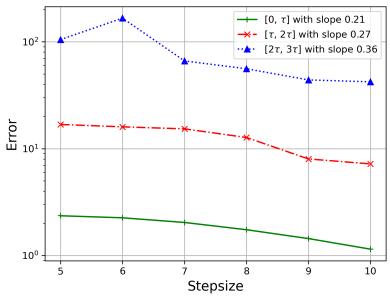
(a) $\alpha_1 = 0.1$ and $\alpha_2 = 0.1$.



(b) $\alpha_1 = 1$ and $\alpha_2 = 0.1$.



(c) $\alpha_1 = 0.1$ and $\alpha_2 = 1$.



(d) $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$.

Figure 4.7: Mean square error slopes for $\gamma_1 = 5$ and $\gamma_2 = 1$ and values of $(\alpha_1, \alpha_2) = (0.1, 0.1), (1, 0.1), (0.1, 1), (0.5, 0.5)$ for (4.95) plus (4.97).

Chapter 5

NODE-ImgNet: A Robust Image Denoising Model via a Randomised Euler Scheme

5.1 Introduction

In this work, we enhance image denoising by integrating randomised numerical schemes into the neural ordinary differential equation (NODE) framework, aiming to improve denoising performance. The image denoising problem can be formulated as follows:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}$ is the noisy image, $\boldsymbol{x}$ is the ground truth image and $\boldsymbol{\epsilon}$ is the noise. Given the noisy image $\boldsymbol{y}$, the task is to recover the ground truth $\boldsymbol{x}$ by eliminating the noise $\boldsymbol{\epsilon}$.

Inspired by traditional differential equation-based image denoising techniques, we propose a novel neural network architecture, NODE-ImgNet, which combines NODEs with convolutional neural network (CNN) modules. In solving the NODE component, we employ a randomised Euler scheme, which improves robustness to irregularities in the data. To the best of our knowledge, this is the first work to incorporate a randomised numerical scheme as the solver of a neural ODE within a deep-learning architecture.

It should be noted that the observed improvements are not solely attributable to the use of the randomised Euler scheme, but also largely depend on the intrinsic structure of NODEs.

It is well known that noise in images can arise from various sources. The most common types are sensor noise [110], compression noise [79], and optical noise [65]. Image denoising refers to the process of removing unwanted noise or artefacts from an image. It is critical for improving the quality, accuracy, and usefulness of images in a wide range of applications. Image denoising techniques have been extensively explored both in the industry and academia in the last several decades, see e.g., [33, 50, 76] for a brief review.

In general, image denoising methods can be categorized into two main groups: classical methods and machine learning-based methods. Classical methods are rooted in mathematical models and signal processing techniques, such as median filtering [34], wavelet denoising [15], total variation denoising [94], non-local means filtering [11], to list a few. On the other hand, machine learning-based methods use artificial neural networks or other machine learning algorithms to learn the mapping between noisy and clean images. In recent years, many machine learning-based image denoising models have achieved remarkable success. Popular deep learning methods in this area include image denoising CNNs (DnCNN) [111], residual dense networks (RDN) [113], non-local neural networks (N3Net) [85], Batch renormalization denoising network (BRDNet) [102], to list a few.

Among classical methods, differential equation-based approaches have been widely used to remove noise from digital images, with representative examples including the Perona–Malik model [84], total variation (TV) method [94], and fast marching methods (FMM) [96]. While these traditional methods, which do not involve neural networks, are effective in preserving edges and textures, they are often computationally intensive due to the need to solve non-linear differential equations numerically, and each fixed model typically has limited capacity for handling complex or varying noise. These drawbacks have motivated ongoing research into more flexible and efficient alternatives for image denoising.

Recently several neural network methods have been proposed to learn differential equations for various imaging tasks [3, 26]. One advantage of using learning-based differential equations over traditional ones is that they do not require human intervention, as they are learned directly from training data. In contrast, the traditional approach often involves human input and expertise in formulating mathematical equations. In this article, we also take a similar learning-based approach. Such an approach allows us not to propose any specific dynamic system in the first instance, and instead learn it through a general ODE system modelled by the neural ODE (NODE) method [16].

More precisely, we assume that the image denoising task can be modelled by the following dynamic system:

$$\boldsymbol{v}_t = F(t, \boldsymbol{v}), \quad \boldsymbol{v}(t=0) = \boldsymbol{y}, \quad \boldsymbol{v}(t=T) = \boldsymbol{x}, \tag{5.1}$$

where the functional $F(t, \boldsymbol{v})$ is to be learned. This formulation models the denoising process as a continuous evolution governed by the dynamics defined by F, transitioning from the initial noisy image $\boldsymbol{v}(0) = \boldsymbol{y}$ to the target denoised image $\boldsymbol{v}(T) = \boldsymbol{x}$ over the variable t.

Since our model in Equation 5.1 represents a general ODE system, it is then natural to utilize a NODE-based neural network [16] to model the system. In addition, NODE is also closely related to the Deep Residual Network (ResNet) [44] as ResNets can be viewed as a discrete-time approximation of a Neural ODE. The authors in [16] demonstrated that NODEs are capable of diverse tasks, such as image classification, density estimation, and generative modelling, and show that they can outperform traditional neural networks on certain tasks while using fewer parameters.

While deterministic numerical solvers can be employed within the NODE framework, they may encounter difficulties when the underlying dynamics or the input data exhibit irregularity or low regularity. To address these challenges, we adopt a randomised Euler scheme for solving the NODE system. Randomised numerical schemes have been shown to be effective in certain challenging ODE settings in the numerical analysis literature, particularly for differential equations with non-smooth coefficients or solutions [60,62,64]. Moreover, in the context of deep learning, explicitly introducing

noise is known to improve the robustness and generalization of models [36, 98]. The stochastic perturbations arising in randomised numerical schemes introduce such noise without an additional noise-injection module, thereby potentially improving the stability and performance of NODE-based architectures for image denoising. These advantages motivate our integration of the randomised Euler scheme within the NODE-based architecture for enhanced image denoising performance.

This chapter is organized as follows: Section 5.2 introduces the proposed NODE-ImgNet model in detail. In particular, Section 5.2.1 presents the NODE framework, followed by the randomised Euler scheme used as the numerical solver. In Section 5.3, we present the results of our proposed method on multiple benchmark datasets, which include two greyscale image datasets (BSD68 and Set12) and three colour-scale image datasets (CBSD68, Kodak24 and McMaster). Our method achieves state-of-the-art performance in terms of peak signal-to-noise ratio (PSNR) across multiple scales of additive Gaussian noise. Moreover, our proposed model surpasses other highly competitive baseline models in real image denoising tasks.

5.2 NODE-ImgNet for Image Denoising

5.2.1 Preliminaries on the NODE Framework

We first recall the NODE framework that serves as the foundation of NODE-ImgNet. A standard feed-forward neural network transforms an input through a finite sequence of layers. A NODE instead regards the hidden representation as a continuous-time state and lets a learned vector field describe how that state evolves. Thus the network output is obtained by following the flow of an ordinary differential equation from an initial time to a final time. This viewpoint is useful in the present chapter because image denoising can naturally be interpreted as an evolution from a noisy state towards a cleaner state.

More concretely, a NODE models a continuous map from $\mathbb{R}^d$ to $\mathbb{R}^d$ as follows. For an input $\mathbf{h}_0 \in \mathbb{R}^d$, the output is the final-time value $\mathbf{h}(T)$ of the solution trajectory

$\mathbf{h}(t)_{t\in[0,T]}$ satisfying

$$\frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} = \mathcal{N}_{\boldsymbol{\theta}}^{\mathrm{vec}}(\mathbf{h}(t), t), \quad \mathbf{h}(0) = \mathbf{h}_0. \tag{5.2}$$

Here the vector field $\mathcal{N}_{\boldsymbol{\theta}}^{\mathrm{vec}}$ is a neural network with trainable parameters $\boldsymbol{\theta}$. In the image denoising model, it is natural to take $\mathbf{h}_0$ as the initial noisy image $\boldsymbol{y}$ and $\mathbf{h}(T)$ as the denoised image, ideally close to the clean image $\boldsymbol{x}$. This is the standard initial-value formulation used in NODEs; in practice the vector field is represented by a neural network and the flow is computed numerically [16].

A residual neural network (ResNet) is built from residual blocks of the form

$$h_{k+1} = h_k + \mathcal{R}_{\theta_k}(h_k),$$

where $\mathcal{R}_{\theta_k}$ is a trainable nonlinear transformation. The skip connection $h_k \mapsto h_k + \mathcal{R}_{\theta_k}(h_k)$ makes the block resemble one forward Euler step. Equivalently, if a residual update is written in the scaled form

$$h_{k+1} = h_k + \Delta t\, F(t_k, h_k; \theta_k),$$

then it has the same algebraic structure as a forward Euler discretisation of an ODE. In this sense, the NODE model can be viewed as a continuous-depth analogue of a ResNet: instead of stacking finitely many residual blocks, it specifies an ODE vector field and obtains the output by solving the corresponding initial-value problem. The practical computation is still discrete, since an ODE solver evaluates the learned vector field at selected time points and returns an approximation to $\mathbf{h}(T)$; these solver evaluations reuse the same trainable vector field rather than adding new trainable layers. Residual blocks have been widely used in many state-of-the-art image denoising neural networks, e.g., DnCNN [111], FFDNet [112], DudeNet [103], and batch renormalization denoising network (BRDNet) [102]. It is important to note that we have the flexibility to utilize various state-of-the-art image denoising network models for the vector field $\mathcal{N}_{\boldsymbol{\theta}}^{\mathrm{vec}}$.

With the recent increase in computational power, state-of-the-art image denoising

models have become more complex, with millions of tunable parameters [37, 82, 109]. However, this increased complexity often requires a large amount of training data to achieve optimal results. In many applications where images are obtained at a high cost, there may be insufficient training data, leading to over-fitting with large iterations. One of our aims is to enable the network to handle different training data sizes, thereby providing greater flexibility. Our guidance is to use a simpler model for smaller training data sets and a more complex model for larger ones through a simple adjustment of time integration steps used in the ODE numerical solver.

More specifically, the complexity of our NODE-based neural network arises from two factors: the vector field $\mathcal{N}_{\boldsymbol{\theta}}^{\text{vec}}(\mathbf{h}(t), t)$ on the right-hand side and the number of time integration steps N used in the numerical ODE solver. As aforementioned, we have the flexibility to construct our vector network $\mathcal{N}_{\boldsymbol{\theta}}^{\text{vec}}(\mathbf{h}(t), t)$ using many state-of-the-art networks as a basis. Notably, when the number of time integration steps is set to $N = 1$, the NODE model reduces to a single residual-style update based on the vector field.

However, unlike classical residual blocks, increasing the number of time integration steps in the NODE network does not increase the number of model parameters; it mainly increases the computational cost of training and evaluation. This is because the same vector field $\mathcal{N}_{\boldsymbol{\theta}}^{\text{vec}}$ is reused at every numerical time step: changing N changes how many times the learned vector field is evaluated, but it does not introduce a new set of trainable weights at each step. To balance the two components of complexity, we propose using a structure-similar yet much simpler analogue of state-of-the-art neural networks for the vector field. The model can then compensate for its simpler vector field by choosing an appropriate number of time integration steps. Since our model uses fewer parameters, it tends to be more effective for small training sets. Furthermore, by increasing the number of time integration steps, we can increase the effective integration depth of the learned flow, which may be beneficial when larger training sets are available.

5.2.2 Randomised Euler Scheme

In practice, solving the NODE system for high-dimensional image denoising tasks requires numerical methods that are both computationally efficient and well-suited to parallel implementation on modern hardware. While classical adaptive ODE solvers such as Runge-Kutta-Fehlberg or Dormand-Prince methods are well-established in scientific computing, they are less commonly used in large-scale deep learning frameworks due to their sequential nature and the substantial memory overhead associated with storing intermediate states, particularly in the context of backpropagation and automatic differentiation.

Consequently, most implementations of neural ODEs rely on fixed-step solvers, such as the forward Euler or Runge-Kutta schemes, which are more compatible with mini-batch training and GPU acceleration. In our framework, we go further by introducing a randomised Euler scheme as the solver for the NODE. Our motivation is twofold: first, randomised numerical schemes have been shown to be effective in certain challenging ODE settings in the numerical analysis literature [64]; second, the resulting stochasticity provides a practical and flexible mechanism that may improve robustness during training [36, 98]. In the present learning-based setting, we do not claim that existing convergence theory directly applies to the NODE framework considered here. Rather, we adopt the randomised Euler scheme as a computational design choice and assess its value empirically in the NODE-ImgNet architecture. A rigorous theoretical analysis of this randomised Euler scheme within the present learning-based NODE framework remains an interesting direction for future research.

The analysis in Section 3.4 suggests a possible route towards such a theory: one may first fix the trained vector field $\mathcal{N}_{\boldsymbol{\theta}}^{\mathrm{vec}}$ and study the randomised Euler discretisation as a numerical approximation of the corresponding NODE flow. Under suitable regularity and growth assumptions on this learned vector field, estimates similar in spirit to the low-regularity ODE analysis could be sought for the solver error. A complete theory for NODE-ImgNet, however, would also have to account for training and optimisation effects, the high-dimensional convolutional parametrisation, and generalisation beyond the training distribution. For this reason, the present chapter focuses on the empiri-

cal performance of the randomised solver, while a full convergence and generalisation analysis for randomised NODE solvers is left as future work.

We now define the randomised Euler scheme employed in this work.

Definition 5.2.1 (Randomised Euler Scheme for NODE-ImgNet)**.** *Let* $(\Omega, \mathcal{F}, \mathbb{P})$ *be a complete probability space, and let* $[0, T]$ *be partitioned uniformly into* N *subintervals with step size* $\Delta t = T/N$*:*

$$0 = t_0 < t_1 < \cdots < t_N = T, \qquad t_k = k\Delta t, \quad k = 0, 1, \ldots, N.$$

For each $k = 1, \ldots, N$*, let* τ_k *be an independent random variable uniformly distributed on* $[t_{k-1}, t_k]$*, i.e.*

$$\tau_k = t_{k-1} + \Delta t\, \gamma_k, \qquad \gamma_k \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0, 1).$$

Consider the NODE system

$$\frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} = \mathcal{N}_{\boldsymbol{\theta}}^{\text{vec}}(\mathbf{h}(t), t), \qquad \mathbf{h}(0) = \boldsymbol{y}.$$

The randomised Euler approximation is defined recursively by

$$\mathbf{h}_k = \mathbf{h}_{k-1} + \Delta t\, \mathcal{N}_{\boldsymbol{\theta}}^{\text{vec}}(\mathbf{h}_{k-1}, \tau_k), \qquad k = 1, 2, \ldots, N,$$

with initial value $\mathbf{h}_0 = \boldsymbol{y}$*. The numerical approximation to* $\mathbf{h}(T)$ *is then given by* $\mathbf{h}_N$*.*

5.2.3 Our proposed NODE-ImgNet

We use (5.2) to model the dynamics of the image denoising process. In particular, we choose to use a shallow CNN with batch normalization and dilation as our *vector field* for image denoising. Such a structure has been used as a basic block in many state-of-the-art networks for image denoising tasks, e.g., the BRDNet [102], DilatedConv-BN-ReLU [105], FFDNet [112], DudeNet [103].

The structure of the shallow CNN vector field consists of a 9-layer CNN, with integrated batch normalization (BN) and dilated convolution. Batch normalization and dilated convolution have been proven effective and widely used in the construction

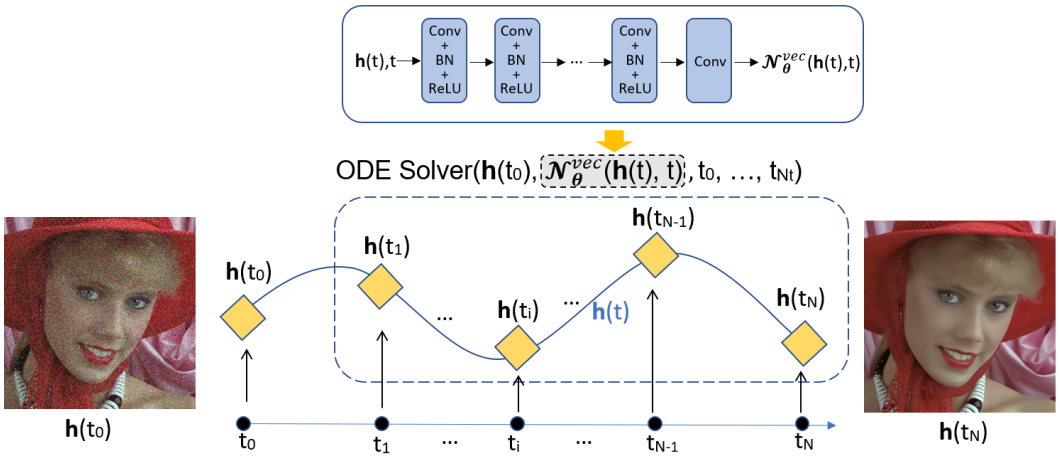


Figure 5.1: Architecture of the proposed NODE-ImgNet network.

of deep CNN models [51, 105]. BN has been proven effective in removing exploding or vanishing gradients, as well as resolving the internal covariate shift problem caused by convolutional operations [51]. The dilation factor and the number of dilated convolution layers directly determine the receptive field size of the network. We refer to Figure 5.1 for a graphical illustration of NODE-ImgNet. The detailed structure of the 9-layer vector field is comprised of two types of structures: Conv + BN + ReLU and Conv. The term 'Conv' refers to a dilated convolutional layer with 3×3 filters and ReLU refers to rectified linear units.

Layers 1-8 are of the Conv + BN + ReLU type and the 9-th layer is Conv. For all layers except the first and the last, the in-channel and out-channel numbers are both set to 128, unless otherwise specified. The in-channel number of the first layer is set to $c+1$ where c is the channel of the image with the extra channel being the time variable. For the final layer, the out-channel number is set to be c.

For the first and last two layers, the dilation factor is set to be 1, which is equivalent to the normal convolution without dilation. The dilation factor of 4 is applied in the convolution layers 2-7. Applying such dilated convolutions gradually increases the receptive field size of the network. More precisely, the initial receptive field size is 3 in the first layer, the same as the kernel size. For each subsequent layer, the size of the receptive field is increased by (kernel size $-$ 1) $\times$ dilation factor. Therefore,

the receptive field sizes of our model in the nine layers are $3, 11, 19, 27, 35, 43, 51, 53$ and 55, respectively. We note that the dilation factors are chosen such that the final receptive field size is close to the size of the image patches. Recall that one of the key features of our model is the flexibility to control its depth by adjusting the number of time integration steps, N. In the experiments described in the chapter, we set N to be between 2 and 8, which allowed us to balance the training time and the expected performance of the model effectively. As mentioned earlier, for the numerical integration, we use the randomised (forward) Euler scheme [64]. Finally, we choose the final time parameter $T = 1$.

5.2.4 Loss function

In our work, we choose the mean-square error (MSE). Define

$$l(\boldsymbol{\theta}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\boldsymbol{f}(\boldsymbol{y}_i, \boldsymbol{\theta}) - \boldsymbol{x}_i\|_2^2, \tag{5.3}$$

where N_p is the number of noisy image patches, $\boldsymbol{\theta}$ denotes the set of all model parameters, and $\boldsymbol{f}(\boldsymbol{y}_i, \boldsymbol{\theta})$ denotes the network output at the final time $t = T$ for the input noisy image $\boldsymbol{y}_i$, i.e.,

$$\boldsymbol{f}(\boldsymbol{y}_i, \boldsymbol{\theta}) = \boldsymbol{h}(T), \qquad \text{with } \boldsymbol{h}(0) = \boldsymbol{y}_i$$

in (5.2). Here, $\boldsymbol{x}_i$ is the ground truth image corresponding to $\boldsymbol{y}_i$. The norm $\|\cdot\|_2$ denotes the Euclidean norm.

5.3 Experimental results

This section presents the experimental results of NODE-ImgNet, together with comparisons against competitive baseline models with similar CNN-based structures. The datasets and experimental settings used in the experiments are documented in Section 5.3.1 and 5.3.2. We compare the model performance on the synthetic Gaussian noise denoising for both gray and colour images in Section 5.3.3 and 5.3.3 respectively. We also perform experiments on the real images and the results are presented

in Section 5.3.3. After that, we investigate the scalability of our proposed method on small-scale partial datasets in Section 5.3.4. In Section 5.3.5, we present our ablation experiments and demonstrate the flexibility of our model. Lastly, we discuss the computational costs in the subsequent Section 5.3.6. The codes are available at `https://github.com/xiexinheng/NODE-ImgNet`.

5.3.1 Datasets

Training dataset

For the experiments of Gaussian noise denoising, we train our model on the Waterloo Exploration Database, which contains 4744 colour images. The full database is used for both colour and gray image denoising tasks, where for the latter we first convert the original colour images to gray images. To test the effectiveness of our model in handling data of various sizes, we also train the model based on 2%, 10%, 20%, and 50% portion of the full image set with random sampling. In all experiments, each image from the dataset is cropped into 300 small 60×60 sized patches. The choice of 60×60 as the patch size is made as a trade-off between computational efficiency and performance. The Gaussian noises are added as independent and identically distributed (i.i.d.) random variables with a mean of 0 and a standard deviation of σ (either fixed or to be specified in a range) to each patch in the training dataset.

For the real image denoising, we train our model based on a dataset of 40 image pairs collected by Xu et al. [108], as well as the SIDD Medium dataset [1], which includes 320 image pairs. Each image pair contains a ground truth image and a corresponding noisy image. Since the images in the first dataset are relatively small, we augment the dataset by applying classical image augmentation techniques such as rotations of 90°, 180°, and 270°, resulting in 120 additional images. We also apply horizontal flipping to each of the 160 images, resulting in an additional 160 images. Combining the augmented dataset with the SIDD dataset, we obtain a total of 640 images. Finally, each image is cropped into 662 patches of size 60×60 pixels, resulting in a total of $423,680$ training patches.

Test dataset

For a fair comparison, we follow other Gaussian image denoising models [102, 111, 112] and test our model on datasets BSD68 and Set12 for the trained gray-scale image models, and on CBSD68, Kodak24 and McMaster for the trained colour image models. BSD68 consists of 68 natural images with dimensions of either 481×321 or 321×481; Set12 consists of 12 gray-scale images with a size of either 256×256 or 512×512. CBSD68 includes 68 colour images with the same background as BSD68. Kodak24 is composed of 24 natural colour images, each with a size of 500×500. McMaster consists of 18 colour images, each with a size of 500×500.

It is known that real noisy images are usually captured by cameras of different types with different ISO values [22]. Motivated by this fact, we choose cc [79] as the test dataset for real noise task. The cc dataset has 15 noisy images captured by three different cameras, i.e., Nikon D800, Nikon D600, and Canon 5D Mark III with different ISO values (1600, 3200, and 6400), shown in Figure 5.2.

Figure 5.2: 15 images in the cc dataset.

5.3.2 Other experimental settings

We use the Adam optimizer with a learning rate of 5×10^{-4}, β_1 of 0.9, β_2 of 0.999, and epsilon of 1×10^{-8}. Here, β_1 and β_2 control the exponential decay rates for the first and second moment estimates of the gradient, respectively, while ϵ is a small value added to the denominator of the Adam update formula to prevent division by zero. The batch size is set to 40, and the maximum number of epochs for training NODE-ImgNet is 50. During the training, the learning rate is halved if the training loss does not decrease for three consecutive epochs. Additionally, we employ early stopping techniques to reduce training time and avoid overfitting. In particular, the loss on a specified test set is computed after each epoch. For gray and colour image denoising tasks, we use the BSD68 and McMaster test sets, respectively. The early stopping is activated if the loss of test set does not decrease for five consecutive epochs.

We train our proposed NODE-ImgNet model under the PyTorch framework [83]. All experiments are conducted on a server running the Linux-5.4.0 operating system and Python 3.9 environment. The server has a memory capacity of 1.4 TiB and is powered by an Intel Xeon(R) Gold 6240 CPU @ 2.60GHz x 72G processor. The server is equipped with an NVIDIA GeForce RTX 3080 Ti GPU, which has 12 GB of GPU RAM. To accelerate the computational performance of the GPU, we use the NVIDIA CUDA 11.8 version. We use the 0.2.3 version of the torchdiffeq package to solve NODEs. This package is optimized for GPU usage and offers a way to reduce memory consumption through backpropagation. To provide the reader with a point of reference, the training of a colour Gaussian denoising NODE-ImgNet model, employing the structure presented in Section 5.2.3 and utilizing $N = 8$, requires an estimated duration of around 82 hours.

5.3.3 Compared results

In this section, we provide both quantitative and qualitative analyses of the performance of NODE-ImgNet. We compare NODE-ImgNet with other competitive baseline image denoising methods, including DnCNN [111], Dilated Conv-BN-ReLU(DC-B-R) [105], FFDNet [112], complex-valued denoising network (CDNet) [91], DudeNet [103], batch

renormalization denoising network(BRDNet) [102], ADNet [101], MWDCNN [104] and GradNet [67]. We select these models because they are primarily constructed using Convolution (Conv), Batch Normalization (BN), Residual Learning (RL), and Rectified Linear Unit (ReLU) technologies, or their variations such as complex-valued and batch renormalization. Comparing our model to these existing models allows for a clear demonstration of the improvements made by our model.

In this section, we use $N = 8$ by default as the number of time integration steps in the ODE solver for NODE-ImgNet, unless otherwise specified. This specific hyperparameter selection is based on finding the optimal balance between accuracy and training cost, ensuring an effective trade-off between the two.

In the following, we first present a comparison of NODE-ImgNet's denoising performance on gray images using the BSD68 and Set12 test sets in Section 5.3.3. Next, in Section 5.3.3, we evaluate the denoising capabilities of NODE-ImgNet on colour images of three test datasets: CBSD68, Kodak24, and McMaster. We also compare NODE-ImgNet's denoising ability on real images using the cc dataset in Section 5.3.3.

NODE-ImgNet for gray Gaussian image denoising

For gray Gaussian image denoising, we train multiple NODE-ImgNet models in two modes. In the first mode, following literature [102, 111, 112] we train three separate models with fixed noise level $\sigma = 15, 25, 50$, respectively. We refer to these three models as NODE-ImgNet.

In the second mode, we employ a blind denoising approach, where the noise level σ for each image patch is randomly sampled based on a uniform distribution between 0 and 55 during the training process. We refer to this model as NODE-ImgNet-B, which aims to provide a robust image denoising model that can deal with a wide range of noise levels in real-world scenarios.

The test results of our trained models along with comparisons based on the BSD68 dataset are presented in Table 5.1, where the highest and second-highest PSNR values for each noise level are highlighted in red and blue, respectively. The results show that NODE-ImgNet outperforms all the compared benchmark models in terms of PSNR.

Moreover, we observe a more significant improvement in our model as the noise level increases. In particular, compared to the baseline model DnCNN, whose structure is similar to the vector field of NODE-ImgNet, NODE-ImgNet achieves average improvements of $0.09, 0.13, 0.22$ (in dB) for $\sigma = 15$, 25 and 50, respectively. NODE-ImgNet-B also shows competitive performance. Notably, when tested at $\sigma = 50$, NODE-ImgNet-B performs better than all other models except NODE-ImgNet.

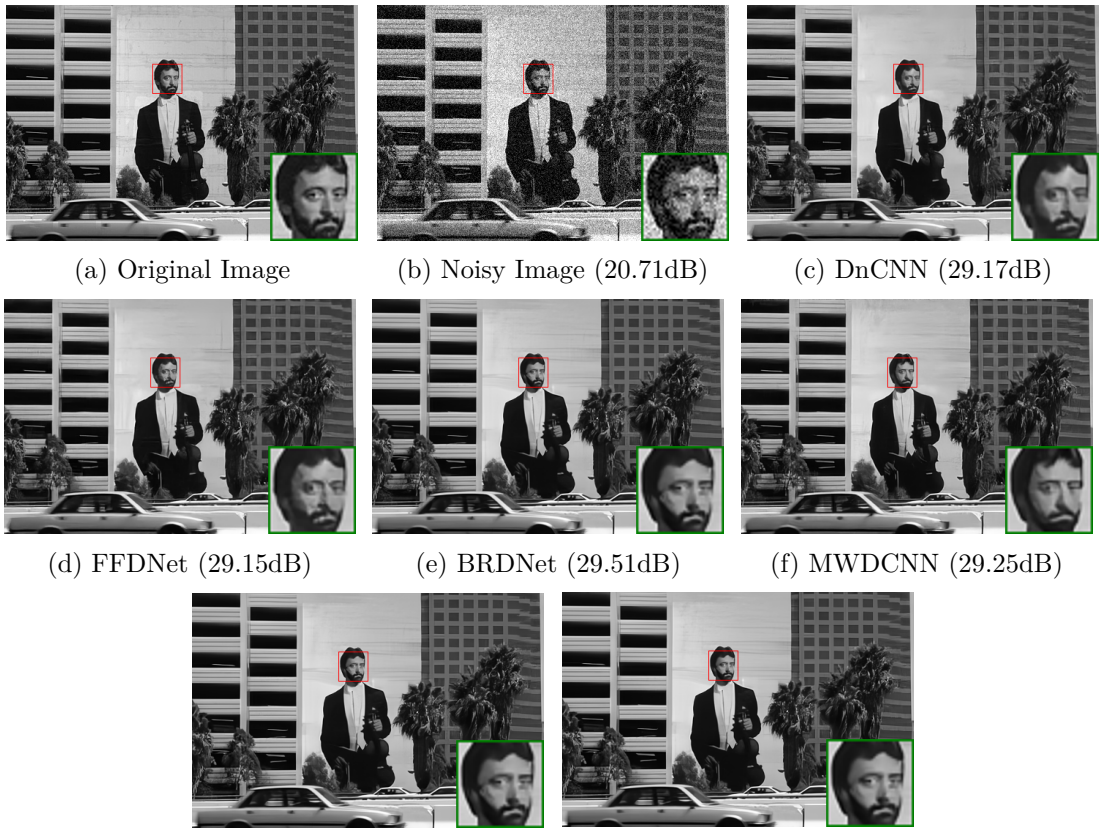
Methods / σ	**DnCNN [111]**	**DC-B-R [105]**	**FFDNet [112]**	**ADNet [101]**	**DudeNet [103]**	**CDNet [91]**	MWDCNN [104]	**BRDNet [102]**	**NODE-ImgNet**	**NODE-ImgNet-B**
$\sigma = 15$	31.72	31.68	31.63	31.74	31.78	31.74	31.77	31.79	31.81	31.66
$\sigma = 25$	29.23	29.18	29.19	29.25	29.29	29.28	29.28	29.29	29.36	29.27
$\sigma = 50$	26.23	26.21	26.29	26.29	26.31	26.36	26.29	26.36	26.45	26.41

Table 5.1: Average PSNR (dB) results of various methods for Gaussian gray image denoising on the BSD68 Dataset at various noise levels.

In Figure 5.3, we compare the results through visualisation of the same zoomed region from denoised images produced by various methods when $\sigma = 25$. We observe that the image denoised by NODE-ImgNet better preserves edges and fine textures.

Figure 5.4 visualises all test images from Set12. In Table 5.2, we present the comparison results for each individual image from Set12 on various noise levels. As before, the highest and second-highest PSNR results for each setting are marked in red and blue, respectively. We again observe that NODE-ImgNet on average outperforms all other models. In particular, compared to the baseline model DnCNN, NODE-ImgNet achieves average improvements of $0.18, 0.30, 0.45$ (in dB) for $\sigma = 15, 25$ and 50, respectively. Clearly, NODE-ImgNet demonstrates superior performance compared to other models across a wide range of image settings. In particular, when evaluated on the Set12 dataset with noise levels of $\sigma = 15, 25$, and 50, NODE-ImgNet achieves the highest PSNR results for 58%, 71%, and 92% of the images, respectively. Similar to previous observations, as the noise level increases, our model achieves more significant improvement.

In Figure 5.5, we compare the results through visualisation of the same zoomed region from denoised images produced by various methods when $\sigma = 50$. These visual results suggest that NODE-ImgNet is better at preserving edges and complex textures.



(a) Original Image (b) Noisy Image (20.71dB) (c) DnCNN (29.17dB)

(d) FFDNet (29.15dB) (e) BRDNet (29.51dB) (f) MWDCNN (29.25dB)

(g) NODE-ImgNet-B (29.52dB) (h) NODE-ImgNet (29.56dB)

Figure 5.3: Denoising results for Gaussian gray noisy image from the BSD68 dataset with $\sigma = 25$.

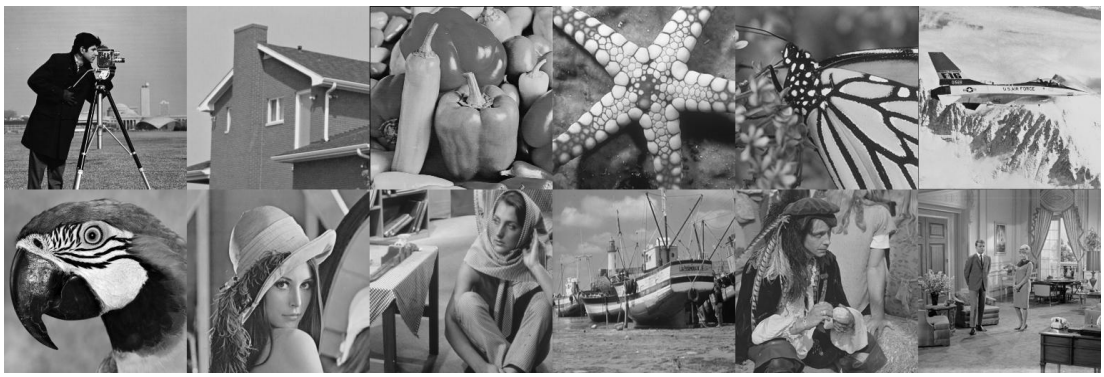


Figure 5.4: The 12 images in the Set12 dataset.

In summary, we conclude that NODE-ImgNet provides a highly competitive model for image denoising on gray images with Gaussian noise. Moreover, as noise level increases, we observe that the NODE-ImgNet achieves more significant improvement

Methods \ Images	C.man	House	Peppers	Starfish	Monarch	Airplane	Parrot	Lena	Barbara	Boat	Man	Couple	Average
Noise Level $\sigma = 15$													
DnCNN [111]	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.86
FFDNet [112]	32.43	35.07	33.25	31.99	32.66	31.57	31.81	34.62	32.54	32.38	32.41	32.46	32.77
ADNet [101]	32.81	35.22	33.49	32.17	33.17	31.86	31.96	34.71	32.80	32.57	32.47	32.58	32.98
DudeNet [103]	32.71	35.13	33.38	32.29	33.28	31.78	31.93	34.66	32.73	32.46	32.46	32.49	32.94
MWDCNN [104]	32.53	35.09	33.29	32.28	33.20	31.74	31.97	34.64	32.65	32.49	32.46	32.52	32.91
BRDNet [102]	32.80	35.27	33.47	32.24	33.35	31.85	32.00	34.75	32.93	32.55	32.50	32.62	33.03
NODE-ImgNet	32.68	35.44	33.48	32.35	33.29	31.90	31.97	34.76	32.87	32.53	32.52	32.65	33.04
NODE-ImgNet-B	32.55	35.28	33.39	32.22	33.10	31.78	31.88	34.70	32.35	32.44	32.44	32.58	32.89
Noise Level $\sigma = 25$													
DnCNN [111]	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.43
FFDNet [112]	30.10	33.28	30.93	29.32	30.08	29.04	29.44	32.57	30.01	30.25	30.11	30.20	30.44
ADNet [101]	30.34	33.41	31.14	29.41	30.39	29.17	29.49	32.61	30.25	30.37	30.08	30.24	30.58
DudeNet [103]	30.23	33.24	30.98	29.53	30.44	29.14	29.48	32.52	30.15	30.24	30.08	30.15	30.52
MWDCNN [104]	30.19	33.33	30.85	29.66	30.55	29.16	29.48	32.67	30.21	30.28	30.10	30.13	30.55
BRDNet [102]	31.39	33.41	31.04	29.46	30.50	29.20	29.55	32.65	30.34	30.33	30.14	30.28	30.61
NODE-ImgNet	30.33	33.68	31.12	29.82	30.57	29.31	29.55	32.80	30.56	30.41	30.21	30.43	30.73
NODE-ImgNet-B	30.22	33.61	31.07	29.64	30.51	29.23	29.51	32.74	29.90	30.33	30.19	30.35	30.61
Noise Level $\sigma = 50$													
DnCNN [111]	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.18
FFDNet [112]	27.05	30.37	27.54	25.75	26.81	25.89	26.57	29.66	26.45	27.33	27.29	27.08	27.32
ADNet [101]	27.31	30.59	27.69	25.70	26.90	25.88	26.56	29.59	26.64	27.35	27.17	27.07	27.37
DudeNet [103]	27.22	30.27	27.51	25.88	26.93	25.88	26.50	29.45	26.49	27.26	27.19	26.97	27.30
MWDCNN [104]	26.99	30.58	27.34	25.85	27.02	25.93	26.48	29.63	26.60	27.23	27.27	27.11	27.34
BRDNet [102]	27.44	30.53	27.67	25.77	26.97	25.93	26.66	29.73	26.85	27.38	27.27	27.17	27.45
NODE-ImgNet	27.32	31.10	27.73	26.05	27.18	26.03	26.67	29.90	27.32	27.46	27.41	27.37	27.63
NODE-ImgNet-B	27.30	31.03	27.68	25.98	27.17	26.00	26.68	29.88	27.08	27.41	27.36	27.33	27.57

Table 5.2: PSNR (dB) results of various methods for Gaussian gray image denoising on the Set12 Dataset at various noise levels.

than baseline models.

NODE-ImgNet for colour Gaussian image denoising

For colour Gaussian image denoising, we also train multiple NODE-ImgNet models in two modes. In the first mode, we train 5 separate NODE-ImgNet models with fixed noise levels $\sigma = 15, 25, 35, 50, 75$, respectively. In the second mode, we employ a similar blind denoising approach as in section 5.3.3, i.e., the noise level σ for each image patch is uniformly distributed between 0 and 55 during the training process. This model is also referred to as NODE-ImgNet-B. We assess the denoising performance of our models with other benchmark models on datasets CBSD68, Kodak24, and McMaster. The results are presented in Table 5.3. NODE-ImgNet consistently demonstrates superior performance for colour Gaussian image denoising in comparison to other baseline models. In addition, it is worth noting that our NODE-ImgNet-B model also outperforms other baseline models at $\sigma = 25, 35, 50$ for all test datasets, demonstrating superior blind noise reduction capabilities.

Figures 5.6, 5.7 and 5.8 visualize the resulting denoised colour images from various denoising models at different Gaussian noise, i.e., $\sigma = 25$, 50 and 75. It is evident from

(a) Original image (b) Noisy image (14.77dB)

(c) DnCNN (26.22dB) (d) FFDNet (26.45dB) (e) BRDNet (26.85dB)

(f) NODE-ImgNet-B (27.08dB) (g) NODE-ImgNet (27.32dB)

Figure 5.5: Denoising results for Gaussian gray noisy image "Barbara" from the Set12 dataset with $\sigma = 50$.

each of the figures that the denoised image produced by NODE-ImgNet successfully restores the most texture and details depicted in the clean image.

In summary, NODE-ImgNet stands out as a competitive model for denoising colour images affected by Gaussian noise. Once again, as the noise level escalates, NODE-ImgNet exhibits a considerably more substantial improvement in performance in con-

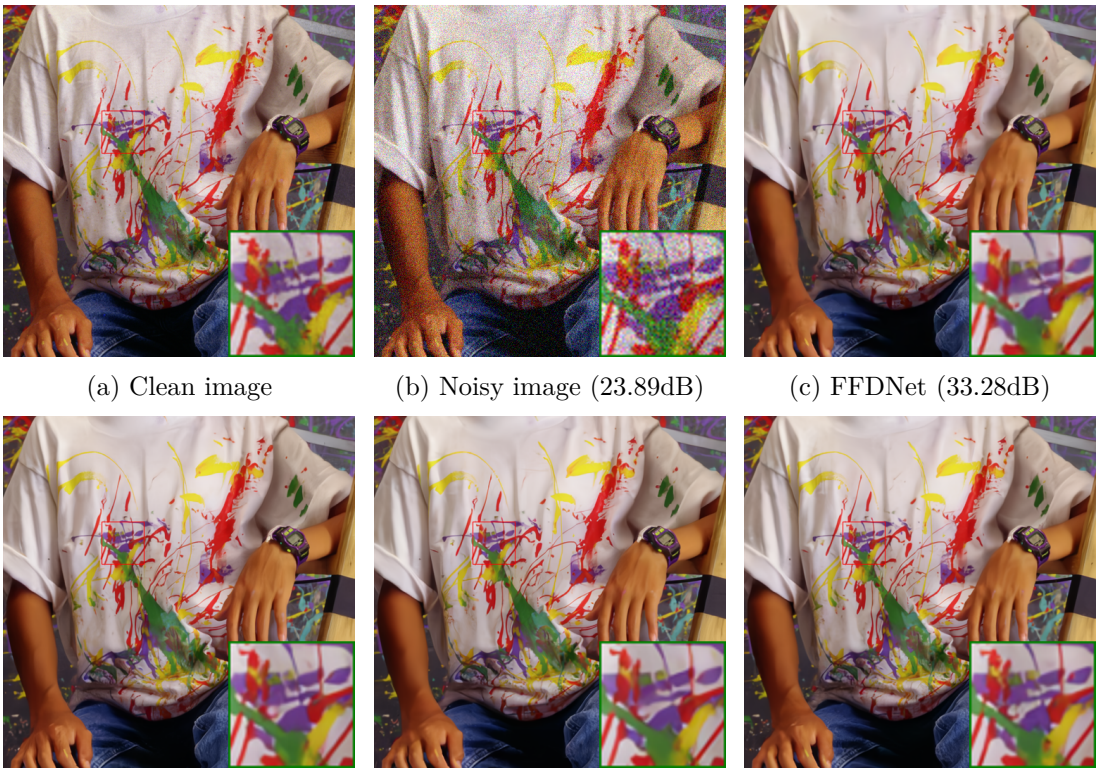
Datasets	**Methods**	$\sigma = 15$	$\sigma = 25$	$\sigma = 35$	$\sigma = 50$	$\sigma = 75$
CBSD68	DnCNN [111]	33.98	31.31	29.65	28.01	-
	DC-B-R [105]	31.68	-	29.18	26.21	-
	FFDNet [112]	33.80	31.18	29.57	27.96	26.24
	GradNet [67]	34.07	31.39	-	28.12	-
	ADNet [101]	33.99	31.31	29.66	28.04	26.33
	DudeNet [103]	34.01	31.34	29.71	28.09	26.40
	BRDNet [102]	34.10	31.43	29.77	28.16	26.43
	NODE-ImgNet	34.19	31.49	29.88	28.29	26.60
	NODE-ImgNet-B	34.05	31.45	29.86	28.24	-
Kodak24	DnCNN [111]	34.73	32.23	30.64	29.02	-
	FFDNet [112]	34.55	32.11	30.56	28.99	27.25
	GradNet [67]	34.85	32.35	-	29.23	-
	ADNet [101]	34.76	32.26	30.68	29.1	27.4
	DudeNet [103]	34.81	32.26	30.69	29.1	27.39
	BRDNet [102]	34.88	32.41	30.8	29.22	27.49
	NODE-ImgNet	34.96	32.51	31.00	29.47	27.78
	NODE-ImgNet-B	34.87	32.47	30.96	29.38	-
McMaster	DnCNN [111]	34.80	32.47	30.91	29.21	-
	FFDNet [112]	34.47	32.25	30.76	29.21	27.29
	GradNet [67]	34.81	32.45	-	29.39	-
	ADNet [101]	34.93	32.56	31	29.36	27.53
	BRDNet [102]	35.08	32.75	31.15	29.52	27.72
	NODE-ImgNet	35.11	32.83	31.35	29.79	27.97
	NODE-ImgNet-B	35.01	32.78	31.31	29.72	-

Table 5.3: Average PSNR (dB) results of various methods for Gaussian colour image denoising at various noise levels on the CBSD68, Kodak24, and McMaster datasets.

trast to the baseline models.

NODE-ImgNet for real image denoising

In this subsection, we compare the performance of NODE-ImgNet on real image denoising with other competitive baseline methods including DnCNN [111], ADNet [101], DudeNet [103], MWDCNN [104] and BRDNet [102]. The trained model is tested on the cc [79] dataset with 15 noisy images. The results presented in Table 5.4 show that NODE-ImgNet outperforms other models for the majority of images in the data set. In particular, when the ISO values are larger, indicating higher levels of noise, our model



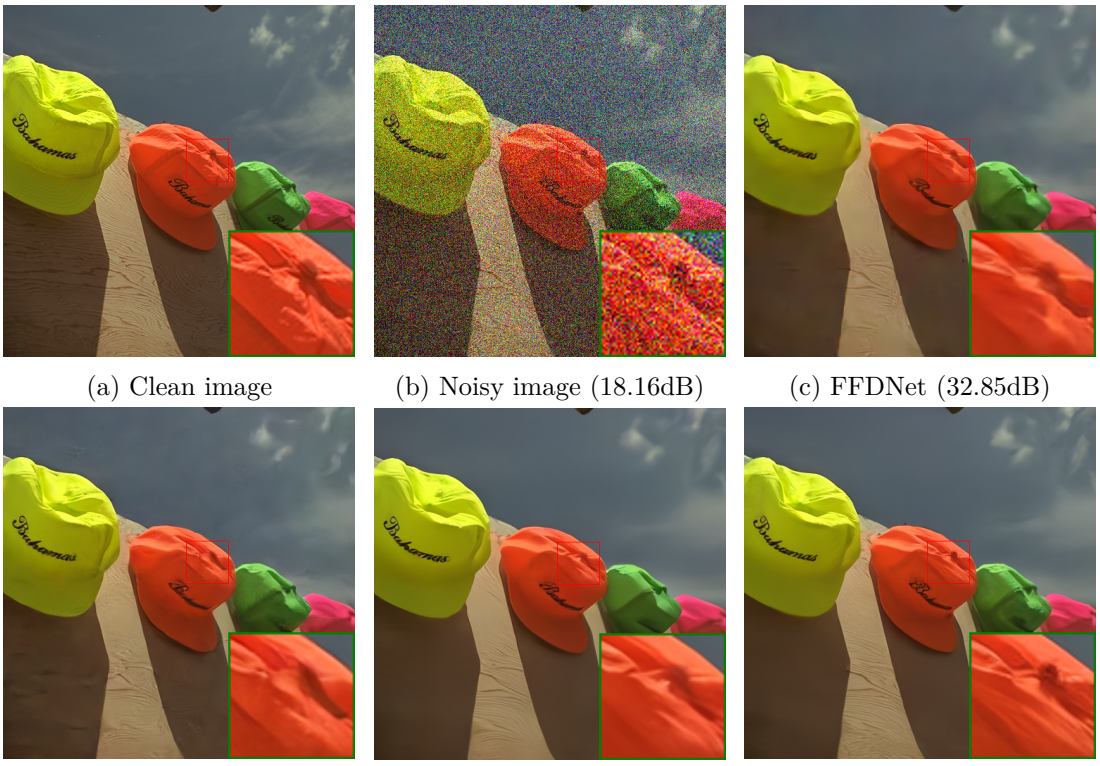
(a) Clean image (b) Noisy image (23.89dB) (c) FFDNet (33.28dB)

(d) BRDNet (33.54dB) (e) NODE-ImgNet-B (33.59dB) (f) NODE-ImgNet (33.65dB)

Figure 5.6: Denoising results for a Gaussian colour noisy image from the McMaster dataset with $\sigma = 25$.

performs uniformly better with higher improvement. This along with former results indicates that our NODE-ImgNet has strong denoising capabilities for more complex noise types, such as real-world noise.

It is important to note that for real image denoising, the improvement of NODE-ImgNet over other baseline networks, e.g., BRDNet, is more significant than the synthetic Gaussian denoising tests. More precisely, for the colour Gaussian denoising experiments, the average improvement range in PSNR compared to BRDNet across each test set is between 0.03 to 0.29 dB. Whereas, for the real image denoising, the improvement is 0.60 in dB which is more than twice as large.



(a) Clean image (b) Noisy image (18.16dB) (c) FFDNet (32.85dB)

(d) BRDNet (33.05dB) (e) NODE-ImgNet-B (33.35dB) (f) NODE-ImgNet (33.50dB)

Figure 5.7: Denoising results for a Gaussian colour noisy image from the Kodak24 dataset with $\sigma = 50$.

Camera settings	DnCNN [111]	ADNet [101]	DudeNet [103]	MWDCNN [104]	BRDNet [102]	NODE-ImgNet
Canon 5D ISO=3200	37.26	35.96	36.66	36.97	37.63	37.09
	34.13	36.11	36.70	36.01	37.28	36.78
	34.09	34.49	35.03	34.80	37.75	35.58
Nikon D600 ISO=3200	33.62	33.94	33.72	33.91	34.55	34.93
	34.48	34.33	34.70	34.88	35.99	36.43
	35.41	38.87	37.98	37.02	38.62	42.05
Nikon D800 ISO=1600	35.79	37.61	38.10	37.93	39.22	37.92
	36.08	38.24	39.15	37.49	39.67	39.81
	35.48	36.89	36.14	38.44	39.04	38.14
Nikon D800 ISO=3200	34.08	37.20	36.96	37.10	38.28	40.35
	33.70	35.67	35.80	36.72	37.18	36.76
	33.31	38.09	37.49	37.25	38.85	40.93
Nikon D800 ISO=6400	29.83	32.24	31.94	32.24	32.75	34.79
	30.55	32.59	32.51	32.56	33.24	34.14
	30.09	33.14	32.91	32.76	32.89	34.34
Average	33.86	35.69	35.72	35.74	36.73	37.33

Table 5.4: PSNR (dB) Results of Various Methods on Real Noisy Images

5.3.4 Efficiency with Small-Scale Partial Datasets

As aforementioned, the inherent flexibility in NODE-ImgNet enables its efficacy on smaller datasets, which is critical for tasks with limited access to training data. To

(a) Clean image (b) Noisy image (15.48dB)

(c) FFDNet (30.68dB) (d) BRDNet (30.94dB) (e) NODE-ImgNet (31.12dB)

Figure 5.8: Denoising results for a Gaussian colour noisy image from the CBSD68 dataset with $\sigma = 75$.

better support the statement, we train our NODE-ImgNet and BRDNet models based on each dataset that comprises $50\%, 25\%, 10\%$, and 2% of the Waterloo Exploration Database with random sampling, respectively. The resulting number of training images for the four experiments is, therefore, $2372, 1186, 474$, and 95, respectively. Each selected image is also cropped into 300 patches of size 60×60, following the same procedure as before. The models are trained with fixed noise level $\sigma = 50$.

The test results on the McMaster, Kodak24, and CBSD68 dataset are presented in Table 5.5. We observe that NODE-ImgNet is less negatively affected when decreasing the amount of training data. To better visualize the data, we also organize them in Figure 5.9 for the McMaster test dataset. With linear regression, the decrease rates for NODE-ImgNet and BRDNet are -0.168 and -0.246 in dB/percent, respectively. Therefore, Figure 5.9 further confirms that our model can achieve better performance on small-scale partial datasets compared to BRDNet and, therefore, can effectively handle tasks with limited training data.

5.3.5 Ablation Experiments and Demonstration of Model Flexibility

In this subsection, we conducted ablation experiments. The results highlight the efficacy

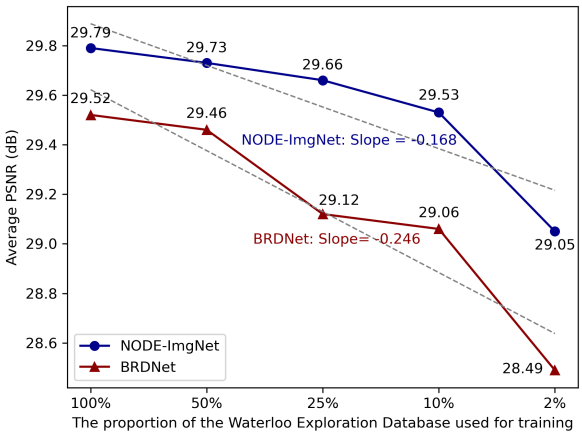


Figure 5.9: Performance of NODE-ImgNet and BRDNet with decreasing size of training Dataset. Shown results are obtained through the McMaster test set.

	Training Data Ratio	NODE-ImgNet	BRDNet	Relative Decrease Value (dB)		Relative Decrease Percentage	
McMaster	100%	29.79	29.52	NODE-ImgNet	BRDNet	NODE-ImgNet	BRDNet
	50%	29.73	29.46	-0.06	-0.06	-0.19%	-0.19%
	25%	29.66	29.12	-0.13	-0.40	-0.45%	-1.37%
	10%	29.53	29.06	-0.26	-0.47	-0.87%	-1.60%
	5%	29.29	29.02	-0.50	-0.50	-1.72%	-1.74%
	2%	29.05	28.49	-0.74	-1.03	-2.56%	-3.61%
Kodak24	100%	29.47	29.22	NODE-ImgNet	BRDNet	NODE-ImgNet	BRDNet
	50%	29.46	29.20	0.01	-0.02	-0.03%	-0.08%
	25%	29.35	28.95	-0.13	-0.27	-0.42%	-0.94%
	10%	29.26	28.95	-0.21	-0.27	-0.72%	-0.94%
	5%	29.06	28.90	-0.41	-0.32	-1.39%	-1.09%
	2%	28.94	28.50	-0.53	-0.72	-1.81%	-2.52%
CBSD68	100%	28.29	28.16	NODE-ImgNet	BRDNet	NODE-ImgNet	BRDNet
	50%	28.28	28.10	-0.01	-0.06	-0.04%	-0.20%
	25%	28.22	27.95	-0.07	-0.22	-0.24%	-0.77%
	10%	28.17	27.93	-0.12	-0.23	-0.44%	-0.83%
	5%	28.03	27.90	-0.26	-0.26	-0.91%	-0.93%
	2%	27.93	27.58	-0.36	-0.59	-1.30%	-2.12%

Table 5.5: Performance of NODE-ImgNet and BRDNet with decreasing size of training Dataset. Evaluations are shown on CBSD68, Kodak24, and McMaster test sets.

of our approach. Concurrently, these ablation studies also reveal a significant advantage of our NODE structure: it offers flexibility in choosing the hyper-parameter N, thereby facilitating a balance between training time and accuracy.

The novelty of our model stems from the combination of the NODE framework with the CNN-based vector field. This leads us to consider two aspects in the ablation study: first, removing the NODE structure to assess the denoising capability of the underlying vector-field CNN; and second, varying the number of time integration steps N to evaluate the behaviour of NODE-ImgNet under different integration depths. For

the former, we design two baseline models. In the first, we completely remove the NODE structure and retain only the vector-field CNN. We refer to this model as *Vector field ($N = 0$)*. In the second, we again remove the NODE structure but introduce a residual connection between the input and output to strengthen the original CNN. We refer to this model as *Vector field ($N = 1$)*. This notation is used only for convenience: when $N = 1$, the discretisation reduces to a single residual-style update, so the residual-augmented vector-field model serves as the closest non-NODE comparator. Accordingly, rather than reporting a separate NODE-ImgNet configuration at $N = 1$, we use this residual-augmented baseline for comparison. For the latter approach, we test the denoising capability of NODE-ImgNet by solely adjusting the N from 2 to 8, examining its performance under various time integration steps.

For a fair comparison, all of the aforementioned models are trained based on the full Waterloo dataset with a fixed noise level of $\sigma = 50$. The final experimental results are shown in Figure 5.10. On the x-axis, the labels $N = 0$ and $N = 1$ correspond to the Vector field ($N = 0$) and Vector field ($N = 1$) models respectively, while labels $N = 2$ to $N = 8$ represent the NODE-ImgNet with the number of time integration steps from $N = 2$ to $N = 8$. In the figure, the model's performance is depicted by a blue curve, while the training time is represented by a red curve.

Comparing the results of Vector field ($N = 0$, 1) with NODE-ImgNet $N = 2 - 8$ underscores the efficacy of incorporating NODE with an appropriate number of steps, thus validating the effectiveness of our structure. Further, when observing the results of NODE-ImgNet from $N = 2$ to $N = 8$, it's evident that both the training time (for 100 batches) and the PSNR value exhibit an almost linear relationship with the value of N. This demonstrates the flexibility of our model, as the trade-off between performance and training time can be adjusted simply by changing N.

However, it's crucial to point out that as the value of N increases, the growth rate for PSNR is relatively more gradual than that for the training time. Nonetheless, it's noteworthy that with $N = 2$, NODE-ImgNet already surpasses BRDNet under an identical noise setting.

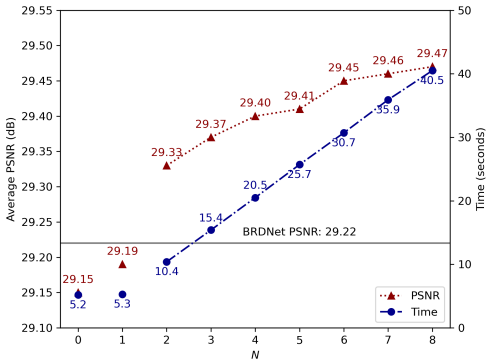


Figure 5.10: Average PSNR for Kodak24 test set and training time for 100 batches under varying N values.

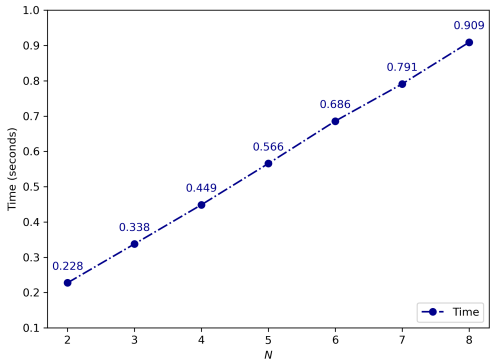


Figure 5.11: Test time versus N values on a 1024×1024 colour image using NODE-ImgNet.

5.3.6 Computational Costs

Testing time and the number of parameters are also important metrics for evaluating models [95]. In this subsection, we compare our model NODE-ImgNet with various other baseline models including FFDNet, DnCNN, MWDCNN, and BRDNet in terms of evaluation/testing time and the number of parameters. The testing times are obtained on testing colour images of size 256×256, 512×512, and 1024×1024 with Gaussian fixed noise level $\sigma = 50$. To ensure fair comparison, we rebuild these models and test them on the same device mentioned in section 5.3.2. The results are shown in Table 5.6. We also plot the testing time for denoising the same 1024×1024 colour image versus N, as shown in Figure 5.11. We observe that when $N = 2$, the testing time of NODE-ImgNet is comparable with BRDNet whereas NODE-ImgNet performs better and has less number of parameters.

Methods		**DnCNN**	**ADNet**	**DudeNet**	**MWDCNN**	**BRDNet**	**NODE-ImgNet**		
							N=2, 128 channels	**N=8, 128 channels**	**N=8, 64 channels**
Parameters		0.56M	0.52 M	1.03M	0.50M	1.11M	1.04M	1.04M	0.26M
size	**256 × 256**	0.007s	0.012s	0.014	0.027s	0.012s	0.017s	0.072	0.052s
	512 × 512	0.024s	0.032s	0.047s	0.147s	0.047s	0.060s	0.254s	0.148s
	1024 × 1024	0.103s	0.142s	0.221s	0.877s	0.197s	0.228s	0.887s	0.4763s

Table 5.6: Comparison of testing time for various models using different input image sizes.

It is worth mentioning that BRDNet utilizes a channel number of 64 in its interme-

diate layers. To provide a comparison point, we also train NODE-ImgNet models with the same intermediate channel number of 64 and $N = 8$.

In this scenario, the vector field network employed by NODE-ImgNet is a proper subset of BRDNet, resulting in NODE-ImgNet utilizing only 23% of the parameters found in BRDNet. The performance of NODE-ImgNet, with a channel number of 64, is presented in Table 5.7. We also include a comparison to other baseline models for three colour image test datasets. Upon examining the results, our model demonstrates superior performance compared to all the baseline models for the chosen high levels of noise.

The results suggest that NODE-ImgNet exhibits significant potential for image denoising on portable devices like smartphones, drones, embedded systems, and cameras.

Datasets	Noise Level	DnCNN [111]	FFDNet [112]	GradNet [67]	ADNet [101]	DudeNet [103]	BRDNet [102]	NODE-ImgNet-S
CBSD68	$\sigma = 50$	28.01	27.96	28.12	28.04	28.09	28.16	28.19
	$\sigma = 75$	-	26.24	-	26.33	26.40	26.43	26.54
Kodak24	$\sigma = 50$	29.02	28.99	29.23	29.1	29.1	29.22	29.30
	$\sigma = 75$	-	27.25	-	27.4	27.39	27.49	27.67
McMaster	$\sigma = 50$	29.21	29.21	29.39	29.36	-	29.52	29.57
	$\sigma = 75$	-	27.29	-	27.53	-	27.72	27.81

Table 5.7: Comparison of NODE-ImgNet (channel number 64) and various other models for Gaussian colour image denoising at two high noise levels.

Chapter 6

Conclusion

This thesis focuses on the development and analysis of randomized numerical schemes for differential equations with low regularity, and further demonstrates their application in data-driven contexts. First, for DDEs with time dependence that is only Hölder continuous, a randomized Runge-Kutta scheme is proposed and its convergence and error bounds are rigorously established under suitable strengthened low-regularity assumptions, thereby extending the class of low-regularity Carathéodory-type DDEs that can be treated within a rigorous numerical analysis framework. Second, for SDDEs with Carathéodory-type drift coefficients, a randomized Euler scheme is constructed and analysed, providing a rigorous convergence analysis together with numerical validation for this class of low-regularity SDDEs. Finally, the randomized numerical schemes are extended to data-driven domains by introducing the randomized Euler scheme into NODE solvers, resulting in the NODE-ImgNet model for image denoising, which shows competitive and, in many cases, superior empirical performance on both synthetic and real noisy-image benchmarks, while also performing well under limited training data.

A central theme throughout this thesis is the usefulness of randomized numerical schemes for differential equations with low regularity. Compared to traditional fixed-grid approaches, randomized evaluation points within each time step can reduce the adverse impact of pathological alignment between the discretization and irregular features of the problem. Both theoretical analysis and numerical experiments demonstrate that, for problems where classical deterministic schemes may fail or experience a sig-

nificant loss in accuracy, randomized schemes can still provide controlled error bounds and, in some settings, improved convergence rates relative to classical deterministic schemes. In other words, while randomization does not improve the inherent regularity of the underlying problem, it enables effective control over the numerical error induced by non-smoothness. This perspective relaxes the strict smoothness requirements typically associated with classical numerical schemes and makes it possible to approximate a broader class of low-regularity problems with high accuracy.

From a practical standpoint, randomized numerical integrators broaden the range of low-regularity problems that can be treated effectively, but they also introduce new considerations. Unlike deterministic schemes, randomized algorithms typically provide statistical forms of convergence, such as convergence in expectation, in probability, or in suitable L^p-senses. Careful control of random seeds and sampling procedures is helpful for reproducibility in practical implementations. Additionally, the accumulation of random perturbations must be monitored in high-dimensional or long-time simulations, though theoretical L^p error bounds provide quantitative control of the approximation error in an average sense. In data-driven contexts such as deep learning, integrating randomized solvers (as in NODE-ImgNet) requires balancing solver-induced stochasticity with training stability in practical implementation. Despite these additional considerations, randomized schemes provide a promising framework for problems that are challenging for classical approaches.

For future work, several promising avenues emerge from this research. One direction is the development of higher-order randomized schemes: while this thesis focused on a first-order Euler scheme and a basic two-stage Runge-Kutta scheme, one could design multistage or adaptive randomized integrators to achieve greater accuracy when the problem's regularity permits. Another promising avenue is to extend the randomized integration approach to partial differential equations (PDEs); for example, by applying random time stepping to time-dependent PDEs or introducing randomness into spatial discretization to handle irregular coefficients in space. Such generalizations would enable the application of similar Monte Carlo-inspired techniques to solve PDEs and stochastic PDEs with low regularity.

Additionally, further integration with deep learning presents an exciting frontier. The empirical performance of NODE-ImgNet suggests that further exploration of neural architectures embedding advanced numerical solvers is a promising direction. Examples include physics-informed neural networks (PINNs) [92] that incorporate randomized time integrators, as well as frameworks that exploit solver-induced stochasticity to improve robustness and support uncertainty-aware learning. It would also be worthwhile to pursue theoretical refinements, such as establishing almost-sure convergence results or reducing the variance in the random sampling, to strengthen the reliability of these algorithms. By pursuing these directions, future research can build upon the foundations laid in this thesis, applying the power of randomized numerical schemes to an even broader class of high-dimensional, low-regularity, and data-driven problems.

Bibliography

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Adam Andersson and Raphael Kruse. Mean-square convergence of the BDF2-Maruyama and backward Euler schemes for SDE satisfying a global monotonicity condition. *BIT Numerical Mathematics*, 57(1):21–53, 2017.

[3] Fatemeh Ashouri and Mohammad R Eslahchi. A new PDE learning model for image denoising. *Neural Computing and Applications*, 34(11):8551–8574, 2022.

[4] Jianhai Bao and Yue Wu. Randomised Euler-Maruyama method for SDEs with Hölder continuous drift coefficient. *arXiv preprint arXiv:2501.15527*, 2025.

[5] Heinz Bauer. *Measure and Integration Theory*, volume 26. Walter de Gruyter, 2001.

[6] A. Bellen and M. Zennaro. *Numerical Methods for Delay Differential Equations*. Oxford University Press, Oxford, 2003.

[7] S. Biswas, C. Kumar, Neelima, G. D. Reis, and C. Reisinger. An explicit milstein-type scheme for interacting particle systems and mckean-vlasov sdes with common noise and non-differentiable drift coefficients. *The Annals of Applied Probability*, 34(2):2326–2363, 2024.

[8] T. Bochacik, M. Goćwin, P. M. Morkisz, and P. Przybyłowicz. Randomized Runge-Kutta method–Stability and convergence under inexact information. *J. Complex.*, 65:101554, 2021.

[9] T. Bochacik and P. Przybylowicz. On the randomized Euler schemes for ODEs under inexact information. *Numer. Algor.*, 91:1205–1229, 2022.

[10] H. Brunner and E. Hairer. A general-purpose implicit-explicit Runge-Kutta integrator for delay differential equations. *BIT*, 52(2):293–314, 2012.

[11] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

[12] Donald L. Burkholder. Distribution function inequalities for martingales. *The Annals of Probability*, 1(1):19–42, 1973.

[13] John Charles Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016.

[14] W. Cao, Z. Zhang, and G. E.M. Karniadakis. Numerical methods for stochastic delay differential equations via the Wong–Zakai approximation. *SIAM J. Sci. Comput.*, 37(1):A295–A318, 2015.

[15] S. Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.

[16] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[17] E. Cinlar. *Probability and Stochastics*. Springer Science & Business Media, 2011.

[18] Donald L Cohn. *Measure Theory*, volume 2. Springer, 2013.

[19] Ibrahim Coulibaly and Christian Lécot. A quasi-randomized Runge-Kutta method. *Mathematics of Computation*, 68(226):651–659, 1999.

[20] N. Czyżewska, P. Morkisz, J. Kusiak, P. Oprocha, M. Pietrzyk, P. Przybyłowicz, Ł. Rauch, and D. Szeliga. On mathematical aspects of evolution of dislocation density in metallic materials. *IEEE Access*, 10:86793–86811, 2022.

[21] N. Czyżewska, P. M. Morkisz, and P. Przybyłowicz. Approximation of solutions of DDEs under nonstandard assumptions via Euler scheme. *Numer. Algor.*, 91:1829–1854, 2022.

[22] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I–313. IEEE, 2007.

[23] T. Daun and S. Heinrich. Complexity of Banach space valued and parametric stochastic Itô integration. *J. Complexity*, 40:100–122, 2017.

[24] Thomas Daun. On the randomized solution of initial value problems. *Journal of Complexity*, 27(3-4):300–311, 2011.

[25] Thomas Daun and Stefan Heinrich. Complexity of parametric initial value problems in Banach spaces. *Journal of Complexity*, 30(4):392–429, 2014.

[26] Juan Carlos De los Reyes and Carola-Bibiane Schönlieb. Image denoising: learning the noise model via nonsmooth PDE-constrained optimization. *Inverse Problems & Imaging*, 7(4):1183, 2013.

[27] F. V. Difonzo, P. Przybyłowicz, and Y. Wu. Existence, uniqueness and approximation of solutions to Carathéodory delay differential equations. *to appear in J. Comp. Appl. Math., https://doi.org/10.1016/j.cam.2023.115411*, 2023.

[28] Fabio V. Difonzo, Paweł Przybyłowicz, and Yue Wu. Existence, uniqueness and approximation of solutions to Carathéodory delay differential equations. *J. Comput. Appl. Math.*, 436:115411, 2024. Preprint version: arXiv:2204.02016.

[29] Fabio V Difonzo, Paweł Przybyłowicz, Yue Wu, and Xinheng Xie. A Randomized Runge-Kutta Method for time-irregular delay differential equations. *arXiv preprint arXiv:2401.11658*, 2024.

[30] John R Dormand and Peter J Prince. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[31] Etienne Emmrich. *Discrete Versions of Gronwall's Lemma and Their Application to the Numerical Analysis of Parabolic Problems*. Techn. Univ., 1999.

[32] C. A. Eulalia and C. Lubich. An analysis of multistep Runge-Kutta methods for delay differential equations. *Math. Comput. Model.*, 34(10-11):1197–1213, 2001.

[33] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1):1–12, 2019.

[34] Behrouz Farhang-Boroujeny. *Adaptive Filters: Theory and Applications*. John Wiley & Sons, 2013.

[35] Gerald B Folland. *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, 1999.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[37] Javier Gurrola-Ramos, Oscar Dalmau, and Teresa E Alarcón. A residual dense u-net neural network for image denoising. *IEEE Access*, 9:31742–31754, 2021.

[38] Seymour Haber. A modified Monte-Carlo quadrature. *Mathematics of Computation*, 20(95):361–368, 1966.

[39] Seymour Haber. A modified Monte-Carlo quadrature. II. *Mathematics of Computation*, 21(99):388–397, 1967.

[40] E Hairer, G Wanner, and O Solving. Ii: Stiff and differential-algebraic problems. *Springer*, 1991.

[41] Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006.

[42] J. K. Hale. *Theory of Functional Differential Equations*. Applied Mathematical Sciences. Springer New York, New York, 1977.

[43] J. K. Hale and S. M. V. Lunel. *Introduction to Functional Differential Equations*. Springer New York, New York, 1993.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[45] S Heinrich. Complexity of initial value problems in Banach spaces. *Journal of Mathematical Physics, Analysis, Geometry*, 9(1):73–101, 2013.

[46] S. Heinrich. Complexity of stochastic integration in Sobolev classes. *J. Math. Anal. Appl.*, 476:177–195, 2019.

[47] S. Heinrich and B. Milla. The randomized complexity of initial value problems. *J. Complex.*, 24:77–88, 2008.

[48] Stefan Heinrich and Bernhard Milla. The randomized complexity of initial value problems. *Journal of Complexity*, 24(2):77–88, 2008.

[49] M. Hutzenthaler, A. Jentzen, and P. E. Kloeden. Strong and weak divergence in finite time of euler's method for stochastic differential equations with non-globally lipschitz continuous coefficients. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 467(2130):1563–1576, 2011.

[50] Ademola E Ilesanmi and Taiwo O Ilesanmi. Methods for image denoising using convolutional neural network: a review. *Complex & Intelligent Systems*, 7(5):2179–2198, 2021.

[51] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[52] A. Jentzen and A. Neuenkirch. A random Euler scheme for Carathéodory differential equations. *Journal of Computational and Applied Mathematics*, 224:346–359, 2009.

[53] Arnulf Jentzen and Andreas Neuenkirch. A random Euler scheme for Carathéodory differential equations. *Journal of computational and applied mathematics*, 224(1):346–359, 2009.

[54] H. Woźniakowski J.F. Traub, G.W. Wasilkowski. *Information-Based Complexity*. Academic Press, New York, 1988.

[55] Bolesław Kacewicz. Almost optimal solution of initial-value problems by randomized and quantum algorithms. *Journal of Complexity*, 22(5):676–690, 2006.

[56] Olav Kallenberg and Olav Kallenberg. *Foundations of Modern Probability*, volume 2. Springer, 1997.

[57] Achim Klenke. *Probability Theory: a Comprehensive Course*. Springer Science & Business Media, 2013.

[58] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, Berlin, 3 edition, 1999.

[59] Peter E Kloeden, Eckhard Platen, Peter E Kloeden, and Eckhard Platen. *Stochastic Differential Equations*. Springer, 1992.

[60] R. Kruse and Y. Wu. Error analysis of randomized Runge–Kutta methods for differential equations with time-irregular coefficients. *Comput. Methods Appl. Math.*, 17:479–498, 2017.

[61] R. Kruse and Y. Wu. A randomized milstein method for stochastic differential equations with non-differentiable drift coefficients. *Discrete & Continuous Dynamical Systems-Series B*, 24(8):479–498, 2019.

[62] R. Kruse and Y. Wu. A randomized Milstein method for stochastic differential equations with non-differentiable drift coefficients. *Discrete and Continuous Dynamical Systems - B*, 24(8):3475–3502, 2019.

[63] Raphael Kruse and Yue Wu. A randomized Milstein method for stochastic differential equations with non-differentiable drift coefficients. *arXiv preprint arXiv:1706.09964*, 2017.

[64] Raphael Kruse and Yue Wu. Error analysis of randomized Runge–Kutta methods for differential equations with time-irregular coefficients. *Computational Methods in Applied Mathematics*, 17(3):479–498, 2017.

[65] Xiaodong Kuang, Xiubao Sui, Yuan Liu, Qian Chen, and GU Guohua. Single infrared image optical noise removal using a deep convolutional neural network. *IEEE photonics Journal*, 10(2):1–15, 2017.

[66] J. Kuehn. *Numerical Methods for Delay Differential Equations*. World Scientific Publishing Company, ?, 2004.

[67] Yang Liu, Saeed Anwar, Liang Zheng, and Qi Tian. Gradnet image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 508–509, 2020.

[68] X. Mao. *Stochastic Differential Equations and Applications, 2nd. ed.* Woodhead Publishing, 2007.

[69] X. Mao and L. Shaikhet. Delay-dependent stability criteria for stochastic differential delay equations with Markovian switching. *Stability and Control: Theory and Applications*, 3(2):88–102, 2000.

[70] G. N. Milstein. Approximate integration of stochastic differential equations. *Teor. Verojatnost. i Primenen.*, 19:583–588, 1974. in Russian.

[71] G. N. Milstein. Approximate integration of stochastic differential equations. *Theory Probab. Appl.*, 19(3):557–562, 1975. translated by K. Durr.

[72] G. N. Milstein and M. V. Tretyakov. *Stochastic Numerics for Mathematical Physics*. Scientific Computation. Springer-Verlag, Berlin, 2004.

[73] G.N. Milstein. *Numerical Integration of Stochastic Differential Equations*, volume 313 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1995. Translated and revised from the 1988 Russian original.

[74] P. M. Morkisz and P. Przybyłowicz. Optimal pointwise approximation of SDE's from inexact information. *J. Comp. Appl. Math.*, 324:85–100, 2017.

[75] P. M. Morkisz and P. Przybyłowicz. Randomized derivative-free Milstein algorithm for efficient approximation of solutions of SDEs under noisy information. *J. Comp. Appl. Math.*, 383, 2021. 113112.

[76] Mukesh C Motwani, Mukesh C Gadiya, Rakhi C Motwani, and Frederick C Harris. Survey of image denoising techniques. In *Proceedings of GSPX*, volume 27, pages 27–30. Proceedings of GSPX, 2004.

[77] Thomas Müller-Gronbach, Erich Novak, and Klaus Ritter. *Monte Carlo-Algorithmen*. Springer-Verlag, 2012.

[78] James D Murray. *Mathematical Biology: I. An Introduction*, volume 17. Springer Science & Business Media, 2007.

[79] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1683–1691, 2016.

[80] Bernt Oksendal. *Stochastic Differential Equations: an Introduction with Applications*. Springer Science & Business Media, 2013.

[81] E. Pardoux and A. Rascanu. *Stochastic Differential Equations, Backward SDEs, Partial Differential Equations*. Stochastic Modelling and Applied Probability. Springer Cham, Switzerland, 2014.

[82] Bumjun Park, Songhyun Yu, and Jechang Jeong. Densely connected hierarchical network for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[83] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[84] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[85] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018.

[86] P. Przybyłowicz and P. M. Morkisz. Strong approximation of solutions of stochastic differential equations with time-irregular coefficients via randomized Euler algorithm. *Appl. Numer. Math.*, 2014.

[87] P. Przybyłowicz, V. Schwarz, and M. Szölgyenyi. Randomized milstein algorithm for approximation of solutions of jump-diffusion sdes. *Journal of Computational and Applied Mathematics*, 440:115631, 2024.

[88] P. Przybyłowicz, M. Sobieraj, and Ł. Stępień. Efficient approximation of SDEs driven by countably dimensional Wiener process and Poisson random measure. *SIAM Journal on Numerical Analysis*, 60(2):824–855, 2022.

[89] Paweł Przybyłowicz and Paweł Morkisz. Strong approximation of solutions of stochastic differential equations with time-irregular coefficients via randomized Euler algorithm. *Applied Numerical Mathematics*, 78:80–94, 2014.

[90] Paweł Przybyłowicz, Yue Wu, and Xinheng Xie. On approximation of solutions of stochastic delay differential equations via randomized euler scheme. *Applied Numerical Mathematics*, 197:143–163, 2024.

[91] Yuhui Quan, Yixin Chen, Yizhen Shao, Huan Teng, Yong Xu, and Hui Ji. Image denoising using complex-valued deep CNN. *Pattern Recognition*, 111:107639, 2021.

[92] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[93] Daniel Revuz and Marc Yor. *Continuous Martingales and Brownian Motion*, volume 293. Springer Science & Business Media, 2013.

[94] Leonid I. Rudin, Stanley. Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[95] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2774–2781, 2014.

[96] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[97] Y. Song and X. Yang. A novel explicit two-stage Runge-Kutta method for delay differential equations with constant delay. *Applied Mathematics and Computation*, 272:317–322, 2015.

[98] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[99] Gilbert Stengle. Numerical methods for systems with measurable coefficients. *Applied Mathematics Letters*, 3(4):25–29, 1990.

[100] Gilbert Stengle. Error analysis of a randomized numerical method. *Numerische Mathematik*, 70(1):119–128, 1995.

[101] Chunwei Tian, Yong Xu, Zuoyong Li, Wangmeng Zuo, Lunke Fei, and Hong Liu. Attention-guided CNN for image denoising. *Neural Networks*, 124:117–129, 2020.

[102] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep CNN with batch renormalization. *Neural Networks*, 121:461–473, 2020.

[103] Chunwei Tian, Yong Xu, Wangmeng Zuo, Bo Du, Chia-Wen Lin, and David Zhang. Designing and training of a dual CNN for image denoising. *Knowledge-Based Systems*, 226:106949, 2021.

[104] Chunwei Tian, Menghua Zheng, Wangmeng Zuo, Bob Zhang, Yanning Zhang, and David Zhang. Multi-stage image denoising with the wavelet transform. *Pattern Recognition*, 134:109050, 2023.

[105] Tianyang Wang, Mingxuan Sun, and Kaoning Hu. Dilated deep residual network for image denoising. In *2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI)*, pages 1272–1279. IEEE, 2017.

[106] Yue Wu. Backward Euler–Maruyama method for the random periodic solution of a stochastic differential equation with a monotone drift. *Journal of Theoretical Probability*, 36(1):605–622, 2023.

[107] Xinheng Xie, Yue Wu, Hao Ni, and Cuiyu He. Node-imgnet: A pde-informed effective and robust model for image denoising. *Pattern Recognition*, 148:110176, 2024.

[108] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018.

[109] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[110] Jiachao Zhang and Keigo Hirakawa. Improved denoising via poisson mixture modeling of image sensor noise. *IEEE Transactions on Image Processing*, 26(4):1565–1578, 2017.

[111] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[112] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[113] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481. IEEE, 2018.