IMPERIAL

Imperial College London

Department of Aeronautics

Forecasting of chaotic dynamics and extreme events with quantum reservoir computing and linear embeddings

Osama Ahmed

May 2026

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Aeronautics of Imperial College London and the Diploma of Imperial College London.

Declaration of Originality

I declare that this thesis and the work presented within it are entirely my own. Where the work of others has been used, it has been appropriately acknowledged and referenced. A complete list of references is provided in the bibliography.

Osama Ahmed
May 2026

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-NonCommercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Forecasting chaotic dynamical systems and extreme events remains a central challenge in computational science, with applications spanning weather prediction, turbulence, and nonlinear physics. Classical methods suffer from the curse of dimensionality, while data-driven approaches provide fast predictive capabilities but often lack scalability and physical interpretability. This thesis develops quantum computational approaches for modelling, forecasting, and analysing nonlinear dynamical systems to provide scalable and physically interpretable frameworks for chaotic dynamics and extreme events.

The first part of this thesis develops quantum reservoir computing (QRC) methods for forecasting high-dimensional chaotic dynamics and extreme events. Using the theory of generalised synchronisation (GS), we establish design principles for stable and learnable quantum reservoir computers and introduce recurrence-free quantum reservoir computers (RF-QRCs) as a scalable alternative, demonstrating that RF-QRCs outperform classical reservoir computers for high-dimensional forecasting. We establish practical criteria, generalised synchronisation = echo state property (GS=ESP), and derive analytical Jacobian formulations for stability analysis, investigate finite-sampling noise with denoising strategies, and demonstrate parallel implementations on quantum hardware. Depolarising and amplitude-damping channels are further analysed through GS theory to establish the noise–memory trade-off in quantum reservoir computing.

The second part of this thesis develops quantum-compatible approaches for solving nonlinear differential equations governing chaotic dynamics through linear operator embeddings based on the Fokker–Planck and Koopman–von Neumann formulations. We analyse higher-order discretisation schemes to study the effects of numerical diffusion and stability of probability-density evolution, establishing bounds linking numerical diffusion, grid resolution, and Lyapunov instability. We implement unitary block-encoding quantum algorithms for discretised Fokker–Planck dynamics, demonstrating how nonlinear chaotic evolution can be reformulated into sparse linear operator evolutions for near-term quantum computation.

Together, the complementary approaches developed in this thesis establish quantum computational frameworks for forecasting, analysing, and simulating nonlinear chaotic systems, while providing pathways toward practical quantum advantage in scientific computing.

Acknowledgements

First of all, I would like to thank my supervisor, Professor Luca Magri, for entrusting me with the opportunity to pursue this PhD, and for his guidance and support throughout this work. Under his supervision, I have developed not only technically, but also professionally and personally, gaining the confidence to explore new ideas and research directions. Given the highly multidisciplinary nature of this research, involving fields I initially knew little about, this journey often felt like a steep hill to climb. Looking back now, I can say without any doubt that I could not have asked for a better mentor.

I would like to extend my gratitude to Prof. Dr. Nikolaus A. Adams and Dr.-Ing. Josef Winter for giving me the opportunity to spend a few months at their research lab at the Technical University of Munich. It was a great experience, and I learned a lot from the team, particularly in numerical methods, which greatly helped shape the final chapter of this thesis.

I would also like to extend my sincere thanks to my collaborators, especially Felix, who joined the group as a postdoctoral researcher at the start of my PhD. His scientific guidance, constant support, and willingness to discuss new ideas played an important role throughout this work. Together, we brought complementary perspectives and skill sets to many of the challenges explored in this research. I am also grateful to the early members of the MagriLab at Imperial College London, whom I had the pleasure of working alongside when I first joined, including Nikos, Georgios, Alberto Racca, Tullio, and especially Andrea Novoa, for helping me get my research off the ground, as well as Antonio Colanera and Lloyd Fung for helpful discussions during the final phase of my thesis.

There are a few people who deserve a special mention here. My immediate colleagues and friends, Elise, Defne, Yaxin and Daniel. I was genuinely lucky to join your company. Together, you have all made these years unforgettable, and I am grateful for every moment of it. Elise, thank you for listening to my rants over lunches and coffee breaks. Defne, for lending a patient ear to my half-baked research ideas, especially about ESNs. Daniel, for motivating me to set a higher bar, especially for code writing. And Yaxin, for bringing lively and positive energy to everything, and do keep that magnet with you for your next around-the-world bike trip.

I am also thankful to the Department of Aeronautics for funding my studentship and for providing a supportive environment for research. Many thanks to Clodagh Li, Lisa Kelly, and Siân Haynes for their help and support throughout my PhD, as well as to Professor Sylvain Laizet, Dr George Rigas, and many others in the department. I am also grateful to the many brilliant and inspiring people in Aeronautics and Room 308 who made this journey memorable, including Siddhartha Sahu, Priyam Gupta, Yacine Bengana and Riccardo. Finally, I am thankful to the great friends I met in London who made life here meaningful and memorable, especially Ammad Yousuf and Sallar Ali Qazi.

The journey leading to my PhD has been nothing short of a roller coaster, and I am deeply grateful to the many people who guided me and entrusted me with opportunities along the way. Special thanks to Prof. Philippe Rigo, who recognised my potential as a researcher before anyone and invited me to the Erasmus Mundus master's programme (EMSHIP+) in Belgium. Beyond that opportunity, his support throughout COVID and his encouragement planted the seed that eventually led me here. I am also thankful to the many friends and colleagues at the University of Liège and École Centrale de Nantes who made that experience both memorable and formative. That journey later continued at Friendship Systems, where I am grateful to Stefan Harries and the talented team I had the chance to work alongside, first during my master's thesis and later as a research engineer. The experience gave me not only technical knowledge but also professional instincts that have stayed with me since.

Back in Pakistan, I am grateful to the friends and colleagues I met at the NED University of Engineering and Technology, especially Dr. Tariq Jamil, my FYP group mates, and the people of the 2015–16 batch, Section D/E. All of you are legends. I would also be remiss not to mention friends from childhood and college, including Shaheer, Junaid, Ahsan, and Moiz, among many others, who have been part of the journey long before this PhD began.

Finally, and most importantly, my family members, Papa, Mama, Bhaijan, Hina Api and Amir Bhai, Unzi, and Emaaz. Thank you for always being there for me, for being my strength through difficult times, and for encouraging me to get here. Thanks to my grandparents, aunts and uncles. Above all, Papa and Mama, words cannot express how thankful I am. Initially, I was not even sure whether I could become an engineer, but your unconditional love, sacrifices, and strength kept me going and ultimately brought me here. I promise, this is just the beginning.

Contents

1 Introduction **21**

1.1 Data-driven forecasting of chaotic dynamics and extreme events . . . 22

1.1.1 Reservoir computing and generalised synchronisation . . . 23

1.2 Quantum computing and machine learning . . . 24

1.2.1 Quantum reservoir computing . . . 24

1.2.2 Measurement problem and sampling noise . . . 25

1.3 Quantum computing for nonlinear differential equations . . . 26

1.4 Thesis objectives . . . 29

1.5 Thesis structure . . . 30

2 Learnability of chaotic systems and generalised synchronisation (GS) **31**

2.1 Chaos and predictability . . . 31

2.1.1 Largest Lyapunov exponent . . . 32

2.2 Stability theory and invariant properties . . . 33

2.2.1 Lyapunov spectrum . . . 34

2.2.2 Kaplan-Yorke dimension . . . 34

2.2.3 Covariant Lyapunov Vectors (CLVs) . . . 35

2.3 Data-driven forecasting of chaotic systems . . . 35

2.3.1 Reservoir computing . . . 35

2.3.2 Performance metrics . . . 37

2.4 Generalised synchronisation (GS) . . . 40

2.4.1 Conditional Lyapunov exponents . . . 42

2.5 Reservoir computing as a dynamical system and GS . . . 42

2.5.1 Echo state property . . . 42

2.6 Learnability conditions . . . 43

3 From synchronisation theory to quantum reservoir design **44**

3.1 Qubits and quantum states . . . 44

3.2 Leaky-integrated classical reservoir computing . . . 45

3.3 Hybrid quantum-classical reservoir computing . . . 46

3.4 Quantum reservoir computing: different designs . . . 48
3.5 Recurrence-free quantum reservoir computing (RF-QRC) . . . 50
3.6 Quantum reservoir computing and generalised synchronisation . . . 51
3.6.1 Echo state property in quantum reservoir computers . . . 52
3.6.2 Contractive maps and GS in quantum reservoir computers . . . 52
3.6.3 Jacobians . . . 53
3.7 QRC design principles from generalised synchronisation . . . 54

4 Forecasting chaotic dynamics and extreme events with quantum reservoir computers 56
4.1 Chaotic systems . . . 56
4.1.1 Three-dimensional Lorenz-63 system . . . 57
4.1.2 Higher-dimensional Lorenz-96 system . . . 57
4.1.3 Chaotic shear flow with extreme events . . . 57
4.2 Forecasting of chaotic dynamical systems . . . 58
4.2.1 Setup and metrics . . . 58
4.2.2 Lorenz-63 system . . . 59
4.2.3 Lorenz-96 system . . . 62
4.2.4 Chaotic shear flow with extreme events . . . 65
4.3 Stability and invariant properties . . . 70
4.3.1 Lorenz-63 system . . . 71
4.3.2 Lorenz-96 system . . . 72
4.4 Conclusion . . . 74

5 Noise, mitigation and hardware implementation 75
5.1 Quantum reservoir computing with finite samples . . . 75
5.1.1 Model parameters . . . 77
5.2 Noise mitigation strategies . . . 78
5.2.1 Noise suppression using SVD . . . 79
5.2.2 Noise suppression using signal filtering . . . 82
5.3 Hardware implementation . . . 84
5.3.1 Parallel training and denoising on multiple QPUs . . . 85
5.4 Conclusion . . . 88

6 Noise as a resource for robust quantum reservoir design 89
6.1 Designing robust quantum reservoir computers . . . 89
6.1.1 Lorenz-63 system . . . 90
6.1.2 Lorenz-96 system . . . 92
6.2 Influence of noise . . . 93
6.2.1 Sampling noise . . . 93
6.2.2 Incoherent noise . . . 94
6.3 Conclusion . . . 97

7 Solving nonlinear differential equations with Fokker–Planck embedding 99
7.1 Linear embedding of nonlinear dynamical systems . . . 100

7.1.1 Fokker–Planck approach . . . 100
7.1.2 Koopman–von Neumann approach . . . 102
7.2 Spatial discretisation of Fokker–Planck equation . . . 104
7.2.1 Central difference scheme . . . 105
7.2.2 First- and second-order upwinding schemes . . . 106
7.3 Numerical diffusion . . . 107
7.3.1 First-order upwind scheme . . . 107
7.3.2 Second-order upwind scheme . . . 108
7.3.3 Diffusion time scale and the Lyapunov constraint . . . 108
7.3.4 Implications for scheme selection . . . 109
7.4 Unitary block encoding for forward Euler Fokker–Planck integration . . . 110
7.5 Results . . . 112
7.5.1 Damped harmonic oscillator . . . 112
7.5.2 Lorenz-63 system . . . 118
7.6 Integrating nonlinear dynamics with the Koopman–von Neumann approach . . . 125
7.7 Block-encoding and hardware implementation . . . 126
7.7.1 Hardware implementation . . . 129
7.8 Conclusion . . . 130

8 Conclusions and future work **133**
8.1 Conclusions . . . 133
8.1.1 Quantum reservoir computing: design, robustness, and forecasting . . . 133
8.1.2 Quantum computing for differential equations governing chaotic dynamics . . . 135
8.2 Outlook and future work . . . 135

Bibliography **137**

A Appendices **150**
A.1 Open-access code repositories . . . 150
A.2 Quantum circuits visualisation . . . 150
A.3 Classical reservoir computing hyperparameters . . . 150
A.4 Memory of the reservoir networks . . . 152

List of Figures

2.1 Illustration of exponential growth of an infinitesimal perturbation between nearby trajectories, characterised by the largest Lyapunov exponent λ_1 e.g. (Frederickson et al. 1983; Nastac et al. 2017). 32

2.2 Numerical convergence of the Lyapunov exponent estimates for the continuous-time dynamical system of the Lorenz-63 system (E. N. Lorenz 1963), computed from the tangent linear dynamics using periodic QR orthonormalisation. 34

2.3 Schematic representation of a reservoir computer. The input data $\boldsymbol{u}_{in}$ is mapped to the reservoir matrix via $\boldsymbol{W}_{in}$. The reservoir neuron connections governed by $\boldsymbol{W}$ matrix allow the flow of information between neurons. The linear readout layer, with the trained output weight matrix $\boldsymbol{W}_{out}$, is used to make output predictions $\boldsymbol{u}_p$. 36

2.4 Schematic representation of reservoir computers. On the left-hand side, is the open-loop training phase. On the right side, is the closed-loop prediction phase. 38

2.5 Lorenz-63 system. Short-term time series and long-term statistical predictions via Probability Density Functions (PDFs). True trajectory obtained via Runge-Kutta integration of Lorenz-63 ODEs vs forecasted via Reservoir Computing Method (RCM) and Quantum Reservoir Computing Method (QRCM). 39

2.6 Identical and lag synchronisation visualisation for the Lorenz–63 state variable ($\boldsymbol{x}_1$). . . . 40

2.7 Reservoir computers as drive-response systems. The chaotic time series (data), $\boldsymbol{x}$, is available at discrete intervals over $[t, t+1, ..]$ as $\boldsymbol{u}(t_i) = h(\boldsymbol{x}(t_i))$, with h being the observable map. The driving input forces the reservoir system $\boldsymbol{r}$ (response system). The reservoir state evolves as $\boldsymbol{r}(t_{i+1}) = \boldsymbol{G}\big(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)\big)$. The linear map $\mathbf{W}_{out}$ is the only trainable part of the reservoir computers. 41

3.1 Schematic representation of a gate-based quantum reservoir computer. The first two unitaries are used to encode previous reservoir states $\boldsymbol{r}(t_i)$ and input time-series $\boldsymbol{u}_{in}(t_i)$, the third additional unitary provides randomisation. Each unitary consists of a specific feature map. The quantum circuit execution starts from the ground state $|\mathbf{0}\rangle$ and measures the reservoir state activation for the next time-step $\hat{\boldsymbol{r}}(t_{i+1})$, which is post-processed classically before feeding the next time-step. Each reservoir activation state $\boldsymbol{r}(t_{i+1})$ is saved classically to use for training and predictions. In RF-QRC, we remove the feedback loop and the first unitary $\boldsymbol{P}$. 47

3.2 Classical and quantum reservoir computing algorithms. 48
3.3 Quantum-classical reservoir architecture schematic representation (a) open-loop training to produce $\boldsymbol{W}_{out}$ matrix (b) closed-loop autonomous predictions starting from an arbitrary point in the unseen dataset. The dotted line in both figures indicates the recurrence involved in conventional reservoir architectures. In the RF-QRC architecture, we remove this recurrent feedback layer of reservoir states. 49
3.4 Circuit depth of different ansätze for the ten-dimensional Lorenz-96 input. 50
3.5 Quantum reservoir computers as drive-response systems. The chaotic time series (data) is available at discrete intervals over $[t, t+1, ..]$ as $\boldsymbol{u}(t_i) = h(\boldsymbol{x}(t_i))$, with h being the observable map. The driving input forces the quantum reservoir system $\boldsymbol{r}$ (response system). The reservoir state evolves as $\boldsymbol{r}(t_{i+1}) = \boldsymbol{G}\big(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)\big)$. The linear map $\mathbf{W}_{out}$ is the only trainable part of the quantum reservoir computer. 52

4.1 MFE time series kinetic energy and Probability Density Function (PDF) (a) single time series in which the Kinetic Energy exceeds the extreme event threshold k_e, highlighting the presence of an extreme event (b) probability distribution of kinetic energy calculated over an ensemble of time series 58
4.2 Lorenz-63 system. VPT vs. reservoir sizes for both classical and quantum reservoir configurations. 60
4.3 Lorenz-63 system. VPT/circuit depth vs. reservoir sizes for different quantum reservoir configurations. 60
4.4 Lorenz-63 system. Closed-loop time-series predictions. Comparison of true predictions with classical reservoir computing (CRC) and emulated quantum reservoir computing (QRC). 61
4.5 Lorenz-63 system. Closed-loop statistical predictions. The top panels present the result of the true and classical (CRC) network. The bottom panels compare the true with the QRC predictions. The shaded region in both figures indicates the uncertainty associated with different random seeds in both CRC and QRC. 62
4.6 Lorenz-96 system with ten dimensions. VPT vs reservoir sizes for both classical and quantum reservoir configurations. 63
4.7 Lorenz-96 system with ten dimensions. VPT/circuit depth vs. reservoir sizes for different quantum reservoir configurations. 64
4.8 Lorenz-96 system with ten dimensions. Closed-loop statistical predictions using CRC. The shaded region indicates the uncertainty associated with different random seeds in CRC. 64
4.9 Lorenz-96 system with ten dimensions. Closed-loop statistical predictions using QRC. The shaded region indicates the uncertainty associated with different random seeds in QRC. 65
4.10 Visualization of the prediction of extreme events for both Reservoir Computing Methods (RCMs) i.e. both CRC and QRC (PH= 5.89LT). 66
4.11 MFE system. Predictability Horizon (PH) for predictions with different reservoir sizes for the best performing CRC and QRC-C4 (RF-QRC) networks. 67
4.12 MFE system. 75^{th} percentile of Precision, Recall, F-score Vs. On the left side, Prediction Time (PT) for a constant reservoir size ($N = 1024$). On the right side, Reservoir Size for a constant Prediction Time ($PT = 3LT$). 69

4.13 MFE system. Long-term statistical predictions with classical and quantum reservoir computing. 70

4.14 MFE system. The time average of the square of the midplane velocity along x. (a) True data (b) QRC-C4 predictions (c) CRC predictions for the best-performing networks with a reservoir size of 512 (9 qubits) 70

4.15 Lorenz-63 system, RF-QRC with a reservoir of 7 qubits. Probability density functions (PDFs) of the inferred angles between the three subspaces, where U stands for unstable, N for neutral, and S for stable. 72

4.16 Lorenz-63 system. Angles between subspaces. Top: Ground truth solution. Bottom: Prediction with the RF-QRC with a reservoir of 7 qubits. 73

4.17 Lyapunov spectrum of the Lorenz-96 system with 20 dimensions. The RF-QRC has a reservoir with 13 qubits. 73

5.1 Training phase in reservoir computing. Input time series $\boldsymbol{u}_{in}(t)$ is mapped to a reservoir using the input matrix $\boldsymbol{W}_{in}$ in CRC and suitable encoding schemes in QRC. Inside the reservoir r, each neuron *echoes* with the input time series to generate a sequence of reservoir activation state signals. The reservoir activation signals are concatenated into the reservoir state matrix $\boldsymbol{R}$, which is then used to determine the optimal output weight matrix $\boldsymbol{W}_{out}$ using ridge regression. 76

5.2 Prediction of the MFE time-series on an unseen dataset with (a) noise-free probability distribution, (b) $0.5{\times}10^5$ shots, (c) $2{\times}10^5$ shots, (d) $4{\times}10^5$ shots. Classical reservoir computing (CRC) results from the best-performing reservoir of size 1024 compared with the best-performing quantum reservoir computing (QRC) of 10 qubits. QRC hyperparameters are detailed in Sec. 5.1.1 78

5.3 Averaged signal-to-noise ratios (SNR) of the reservoir states of QRC and RF-QRC with and without denoising filters. The underlying model is an MFE time series with a reservoir size of 10 qubits ($N_r = 1024$). 80

5.4 MFE system trained on a 10-qubit system results in $N_r = 1024$ reservoir activation signals each of length $N_{tr} = 5800$ time steps. A single randomly chosen reservoir activation signal is shown with 50,000 shots at each timestep for RF-QRC are used for training. Top, visualisation of the noisy reservoir activation signal (left) plotted on the ideal (noise-free) signal, and the noise component - the difference between the two signals (right). Bottom, visualisation of the denoised reservoir activation signal (left) plotted on the ideal (noise-free) signal, and the noise component, which is the difference between the two signals (right). 80

5.5 Time series prediction of the Lorenz 63 model using RF-QRC. Averaged mean-squared errors of the trajectory are shown as functions of the number of measurements S for noisy and SVD denoised protocols. a) Comparison of noisy and denoised figures with an original reservoir size of $N_r = 128$. b) Comparison of noisy and denoised figures with a reduced reservoir size of $N_r = 60$. 81

5.6 Time series prediction of the MFE model using RF-QRC. Averaged mean-squared errors of the trajectory are shown as functions of the number of measurements S for noisy and SVD denoised protocols. a) Comparison of noisy and denoised figures with an original reservoir size of $N_r = 1024$. b) Comparison of noisy and denoised figures with a reduced reservoir size of $N_r = 500$. 81
5.7 Comparison of (a) signal-to-noise ratios and (b) mean-squared errors of noisy and denoised RF-QRC trained on a time series of the Lorenz 63 system. Reservoir size $N_r = 128$ represented by 7 qubits. 83
5.8 MFE time series analysis with RF-QRC, using 10-qubit systems. Mean-squared error for noisy and denoised activation states with signal filtering according to Sec. 5.2.2. 83
5.9 Connectivity and qubit map of the ibmq_kyoto 127 qubit device. 10 of the qubits with the lowest readout assignment error are used as a reservoir for RF-QRC of the MFE model. 84
5.10 Recurrence-free quantum reservoir computing parallel training. Input data $\boldsymbol{u}_{in}(t)$ is divided into shorter signals and passed through multiple parallel quantum computers. The obtained reservoir activation signals $\hat{\boldsymbol{r}}_{in}(t)$ associated with each input time series signal are concatenated classically, where leaky integration and denoising are applied. The combined reservoir state matrix $\boldsymbol{R}$ is then used for training using ridge regression. . 86
5.11 MFE system. Training mean-squared error comparison with noisy and denoised states for both emulated and quantum processor-based RF-QRC. 86
5.12 MFE system, training mean-squared error comparison with emulated and quantum processor results. 87
5.13 MFE system, reconstructed kinetic-energy comparison with emulated and backend results. The top panel presents the results of emulated noisy and denoised reservoir activation states. The bottom panel presents the results of hardware-trained noisy and denoised reservoir activation states. 87

6.1 Lorenz-63 in closed-loop autonomous prediction phase. (a) Lyapunov exponents vs leak rate. (b) Lyapunov exponents vs maximum conditional Lyapunov exponents. 91
6.2 Maximum conditional Lyapunov exponents of quantum reservoir computers with recurrence (QRCs) and recurrence-free (RF-QRCs). Each point value is the ensemble mean over ten autonomous predictions from different initial points. 92
6.3 Lorenz-63 system autonomous prediction of Lyapunov spectrum with QRCs (a) Dynamical system's Lyapunov exponents vs Maximum conditional Lyapunov exponents of response (b) Dynamical system's Lyapunov exponents vs Leak rate. In both plots, three points on each vertical line correspond to the three inferred Lyapunov exponents. For very small leak rates, reservoir dynamics are slower, and negative LE is not inferred correctly. QRC performance is optimum for intermediate ϵ between $\{0.35,0.5\}$ but performs inaccurately with $\epsilon > 0.6$ (as opposed to RF-QRC in Fig. 6.1). 93
6.4 Lyapunov spectrum of Lorenz-96 systems with 10 dimensions. Comparison between ground truth and RF-QRC-inferred Lyapunov spectrum for a reservoir size of 10 qubits. (a) Various leak rates (b) Various maximum λ_{CLE} corresponding to leak rate in (a). . . . 94
6.5 Effect of sampling noise on the RF-QRC performance with 8 qubits. (a) Lyapunov spectrum for 1000 shots. (b) Leading Lyapunov exponents for different shots. 95

6.6 Fully-connected quantum circuit for the RF-QRC (O. Ahmed, Tennie, and Magri 2024b) architecture with noise channels η after each gate operation. The number of encoded parameters Θ for RF-QRC is equal to the dimension of the input data ($\Theta = D$) for $\Xi(\boldsymbol{u})$ unitary. When $D > n$, the feature map encoding is applied multiple times to encode all $\boldsymbol{u}$ dimensions. For a random unitary $V(\boldsymbol{u})$, the number of encoded parameters is equal to the number of qubits n, ($\Theta = n$). 95

6.7 Bloch spheres of the quantum state of a qubit for (a) noiseless unitary evolution, (b) uniform contraction of the state due to depolarising noise, and (c) non-uniform contraction of the state due to amplitude damping noise. 96

6.8 Error on the leading Lyapunov exponent with an RF-QRC with 7 qubits. Each result is the ensemble average over 3 different realisations of random seeds. (a) Effect of amplitude-damping noise. (b) Effect of depolarising noise. 97

7.1 Original dynamical equation solution of the 2-dimensional damped harmonic oscillator. On the left, phase space trajectory solution. On the right, time series plot of each variable 113

7.2 Reference phase space solution of the 2-dimensional damped harmonic oscillator obtained via matrix exponentiation. The evolution of the probability distribution function from the initial condition. 113

7.3 Integrated phase space solution of the 2-dimensional damped harmonic oscillator obtained via a first-order Euler central difference scheme. The evolution of the probability distribution function from the initial condition. 113

7.4 Error bar for the reference vs integrated phase space solution of the 2-dimensional damped harmonic oscillator for the first-order Euler central difference scheme. The colour bar indicates the absolute difference. 114

7.5 Phase space trajectory solution, reference solution vs the integrated solution obtained via first-order Euler central difference scheme. 114

7.6 Trajectory comparison for the reference vs integrated Fokker–Planck solution obtained via first-order Euler central difference, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time. 115

7.7 Error plot for the reference versus integrated phase space solution of the 2-dimensional damped harmonic oscillator using the first-order Euler upwinding scheme with zero added diffusion on a 32×32 grid. The colour bar indicates the absolute difference between the reference matrix exponential solution and the integrated Fokker–Planck solution. 115

7.8 Trajectory comparison for the reference versus integrated Fokker–Planck solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 32×32 grid, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time. 116

7.9 Phase space trajectory solution for the reference solution versus the integrated solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 32×32 grid. 116

7.10 Error plot for the reference versus integrated phase space solution of the 2-dimensional damped harmonic oscillator using the first-order Euler upwinding scheme with zero added diffusion on a 64×64 grid. The colour bar indicates the absolute difference between the reference matrix exponential solution and the integrated Fokker–Planck solution. 117

7.11 Trajectory comparison for the reference versus integrated Fokker–Planck solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 64×64 grid, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time. 117

7.12 Phase space trajectory solution for the reference solution versus the integrated solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 64×64 grid. 118

7.13 Trajectory comparison for the reference versus integrated Fokker–Planck solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 128×128 grid, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time. 118

7.14 Trajectory comparison for the SDE solution versus the integrated Fokker–Planck solution of Lorenz–63 obtained using the first-order Euler upwinding scheme with added diffusion coefficients $\mathcal{D}^{(xx)} = \mathcal{D}^{(yy)} = \mathcal{D}^{(zz)} = 0.5$ on a $100 \times 100 \times 100$ grid, compared against the original deterministic system. Time-series plots for each variable are shown together with the ensemble variances. 120

7.15 Projected probability density comparison in the (x, y) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown. 120

7.16 Projected probability density comparison in the (x, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown. 121

7.17 Projected probability density comparison in the (y, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown. 121

7.18 Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.000$, obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme and the reference SDE particle solution. 121

7.19 Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.995$, obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme and the reference SDE particle solution. 122

7.20 Trajectory comparison for the SDE solution versus the integrated Fokker–Planck solution of Lorenz–63 obtained using the second-order Euler upwinding scheme with added diffusion coefficients $\mathcal{D}^{(xx)} = \mathcal{D}^{(yy)} = \mathcal{D}^{(zz)} = 0.5$ on a $100 \times 100 \times 100$ grid, compared against the original deterministic system. Time-series plots for each variable are shown together with the ensemble variances. 122

7.21 Projected probability density comparison in the (x, y) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown. 123

7.22 Projected probability density comparison in the (x, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown. 123

7.23 Projected probability density comparison in the (y, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown. 124

7.24 Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.000$, obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme and the reference SDE particle solution. 124

7.25 Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.995$, obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme and the reference SDE particle solution. 124

7.26 Integrated phase-space solution of the two-dimensional damped harmonic oscillator obtained using the Koopman–von Neumann formulation with a first-order forward-Euler central-difference scheme and no added diffusion. The probability density is reconstructed from $\rho = |\psi|^2$. The evolution is shown over the short-time interval $t \in [0, 1]$. . . 126

7.27 Integrated phase-space solution of the two-dimensional damped harmonic oscillator obtained using the Koopman–von Neumann formulation with a first-order forward-Euler central-difference scheme and no added diffusion. The probability density is reconstructed from $\rho = |\psi|^2$. The evolution is shown over the longer-time interval $t \in [1, 40]$, where spurious oscillations become more visible. 126

7.28 Trajectory comparison between the reference Koopman–von Neumann solution, the forward-Euler Koopman–von Neumann solution, and the original system equation solution. The left panels show the time-series comparison for each variable, while the right panel shows the corresponding variance over time. 127

7.29 Comparison of the probability-density evolution obtained from the reference matrix-exponential solution, the classical forward-Euler integration, and the quantum block-encoding implementation. The exact block-encoding solution closely follows the classical evolution, while the 1000-shot reconstruction reproduces the same qualitative behaviour with small sampling fluctuations. 128
7.30 Probability-density evolution obtained from the IonQ Forte-3 trapped-ion quantum processor for the one-dimensional Fokker–Planck block-encoding implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler Fokker–Planck integration at different times. 130
7.31 Probability-density evolution obtained from the Rigetti Ankaa-3 superconducting quantum processor for the one-dimensional Fokker–Planck block-encoding implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler Fokker–Planck integration at different times. . . 130
7.32 Probability-density evolution obtained from the IQM Emerald superconducting transmon processor using 9 block-encoding integration steps for the one-dimensional Fokker–Planck implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler integration. 130
7.33 Probability-density evolution obtained from the IQM Emerald superconducting transmon processor using 20 block-encoding integration steps for the one-dimensional Fokker–Planck implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler integration. 131

A.1 Quantum circuit for linearly entangled qubits 150
A.2 Quantum circuit for fully entangled qubits 151
A.3 Quantum circuit for fully entangled symmetric qubits 151
A.4 Quantum circuit for product feature map qubits 151
A.5 Linear Memory Capacity (MC) of reservoir networks, mean and standard deviation with hyperparameters. For a summary of hyperparameters cf. Tab. 4.1. For a reservoir size $(N) = 32$. 153
A.6 Linear Memory Capacity (MC) of reservoir networks, mean and standard deviation with hyperparameters. For a summary of hyperparameters cf. Tab. 4.1. For a reservoir size $(N) = 512$. 153

List of Tables

2.1 Criteria for generalised synchronisation and differentiability. The largest conditional Lyapunov exponent of the response system is denoted $\max(\lambda_{\text{CLE}})$, and the most negative Lyapunov exponent of the driving system is denoted λ^*. All Lyapunov exponents are expressed in continuous-time units. 42

3.1 Quantum reservoir computer: different designs. 49
3.2 Connection between GS and QRCs 52

4.1 Parameters for three-dimensional Lorenz-63 system 59
4.2 Parameters for ten-dimensional Lorenz-96 system. 63
4.3 Parameters for the MFE system. 65
4.4 Parameters for the tests on the Lorenz-63 and Lorenz-96 systems. 72
4.5 Lyapunov spectrum of the Lorenz-63 system. Comparison between the ground truth (target) and RF-QRC (inferred) model. 72
4.6 Kaplan-Yorke dimensions. Comparison between the ground truth (target) and the RF-QRC (inferred) model. The % error is defined as the absolute relative error between target and inferred values. 74

5.1 Parameters for the tests on the Lorenz-63 and MFE systems with the recurrence-free quantum reservoir computing architecture. 77
5.2 Training loss comparison on emulator and backend trained networks, with and without denoising 85

6.1 Criteria for generalised synchronisation and differentiability. The largest conditional Lyapunov exponent of the response system is denoted $\max(\lambda_{\text{CLE}})$, and the most negative Lyapunov exponent of the driving system is denoted λ^*. All Lyapunov exponents are expressed in continuous-time units. 90
6.2 Types of noise analysed. 93

7.1 Overview of linear embedding approaches for nonlinear dynamical systems. 101
7.2 Input parameters used for the damped-harmonic oscillator Fokker–Planck simulation. . . 112
7.3 Input parameters used for the Lorenz–63 Fokker–Planck simulations. 119

7.4 Input parameters used for the damped harmonic oscillator, Koopman–von Neumann simulation. 125
7.5 Input parameters used for the one-dimensional Fokker–Planck block-encoding simulation. 127
7.6 Hardware configurations used for the one-dimensional Fokker–Planck block-encoding experiments. All circuits use four qubits with $\Delta t = 0.01$ and 8 grid points. Gate counts are rounded per-circuit averages over the submitted time-step runs. 129

A.1 Parameters for the Lorenz-63 and MFE system for the classical reservoir computing architecture. 152

Nomenclature

Acronyms and abbreviations

CFL	Courant–Friedrichs–Lewy number
CLE	Conditional Lyapunov exponent
CRC	Classical reservoir computer
ESP	Echo state property
GS	Generalised synchronisation
LT	Lyapunov time
MFE	Moehlis–Faisst–Eckhardt model
PH	Predictability horizon
PT	Prediction time
QRC	Quantum reservoir computer
RF-QRC	Recurrence-free quantum reservoir computer
SNR	Signal-to-noise ratio
VPT	Valid prediction time

State variables and reservoir quantities

$\hat{\boldsymbol{r}}(t)$	Reservoir activation before leaking
$\bar{\boldsymbol{r}}(t)$	Sampled reservoir activation vector with finite measurements
$\boldsymbol{p}(t)$	Discrete probability mass vector
$\boldsymbol{r}(t)$	Reservoir state vector
$\boldsymbol{u}(t)$	Observed driving signal or input time series

$\boldsymbol{u}_{in}(t)$	Input time-series signal
$\boldsymbol{x}(t)$	State vector of the driving dynamical system
$\rho(\boldsymbol{x}, t)$	Probability density over phase space
ρ_0	Initial probability density
ρ_{in}	Initial input probability density

Operators, mappings, and matrices

$\hat{\mathcal{F}}$	Linear Fokker–Planck operator
$\hat{\boldsymbol{p}}$	Momentum operator, $-i\nabla$
$\hat{H}_{\text{KvN}}$	Koopman–von Neumann Hamiltonian
$\mathcal{M}$	Attractor or synchronisation manifold
$\mathcal{T}$	Time-ordering operator
ϕ	Generalised synchronisation map from drive to response
$\boldsymbol{\mathcal{R}}$	Discretised Fokker–Planck rate operator
$\boldsymbol{F}$	Discrete-time map of the driving dynamical system
$\boldsymbol{G}$	Response map of the reservoir system
$\boldsymbol{J}_{CLE}$	Jacobian used for CLE computation
$\boldsymbol{R}$	Reservoir state matrix
$\boldsymbol{R}_S$	Upper triangular matrix obtained after QR-decomposition
$\boldsymbol{V}_Z$	Training-time average covariance of sampling noise
$\boldsymbol{W}$	Reservoir recurrent weight matrix
$\boldsymbol{W}_{in}$	Reservoir input weight matrix
$\boldsymbol{W}_{out}$	Reservoir output readout weight matrix
$\boldsymbol{W}_S$	Pseudorandom perturbation vectors containing matrix
$\boldsymbol{Z}$	Sampling-noise perturbation matrix

Stability and forecasting quantities

λ^*	Most negative Lyapunov exponent of the driving system
λ_1	Largest Lyapunov exponent
λ_{max}	Maximum Lyapunov exponent

λ_{CLE}	Largest conditional Lyapunov exponent
$\mathcal{D}_{\text{eff}}$	Effective diffusion coefficient
$\mathcal{D}_{\text{num}}$	Numerical diffusion introduced by the discretisation
τ_λ	Lyapunov time scale
τ_d	Diffusion time scale

Dimensions, hyperparameters, and numerical parameters

β	Tikhonov regularisation parameter
Δt	Time step size
Δx	Spatial grid spacing
ϵ	Leak rate of the reservoir
$\rho(\boldsymbol{W})$	Spectral radius of the reservoir matrix
a	Advection velocity
D	Phase-space dimension
dt	Numerical integration time step
N	Number of grid points
N_r	Reservoir dimension
N_u	Input dimension
N_w	Number of washout steps
N_{tr}	Number of training time steps
S	Number of measurement shots

Quantum computing symbols

$(\mathbb{C}^2)^{\otimes n}$	Hilbert space of n qubits
α, γ_q	Complex amplitudes of a qubit state
$\bar{\rho}$	Quantum density operator
$\mathbb{C}^2$	Single-qubit Hilbert space
$\mathcal{U}$	Unitary evolution operator
$\boldsymbol{\theta}$	Parameters of a quantum circuit
Ξ	Input encoding unitary

n	Number of qubits
P	Reservoir-state encoding unitary
$U_{\Delta t}$	Block-encoded unitary evolution operator
V	Randomisation or variation unitary

Dynamical-system and Fokker–Planck quantities

$\boldsymbol{\Gamma}(\boldsymbol{x}, t)$	Stochastic forcing term
$\mathcal{D}^{(i)}(\boldsymbol{x}, t)$	Drift coefficient in the Fokker–Planck equation
$\mathcal{D}^{(ij)}(\boldsymbol{x}, t)$	Diffusion coefficient matrix in the Fokker–Planck equation
$\mathcal{D}^{[k]}(x, t)$	Kramers–Moyal moment of order k
$\boldsymbol{f}(\boldsymbol{x}, t)$	Nonlinear drift vector field
$\boldsymbol{x}_0$	Initial condition
$k_{\max}$	Maximum resolved wavenumber
P_{suc}	Block-encoding success probability
T	Total integration time

Benchmark systems and extreme-event quantities

$\bar{k}$	Time average of kinetic energy
δ_{k_e}	Prediction lead time before an extreme event
γ	Nonlinear damping coefficient
$\sigma_L, \rho_L, \beta_L$	Lorenz–63 system parameters
τ_e	Prediction horizon decrement step
F_L	External forcing parameter in the Lorenz–96 system
$k(t)$	Kinetic energy in the MFE system
k_e	Extreme-event threshold for kinetic energy
x_i	State of the i-th variable in the Lorenz–96 system

Mathematical operators and distributions

$(\cdot)^\dagger$	Hermitian conjugate
$(\cdot)^T$	Transpose operator
Δ	Laplace operator

$\delta(\cdot)$	Dirac delta distribution
$\langle\cdot\rangle$	Expectation over stochastic realisations
$\mathbb{C}$	Complex numbers
$\mathbb{E}[\cdot]$	Expectation operator
$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers
$\mathcal{N}$	Gaussian distribution
$\mathcal{O}(\cdot)$	Order notation
∇	Nabla operator
$\otimes$	Tensor product
∂_t	Partial derivative with respect to time

Chapter
1

Introduction

Nonlinear dynamical systems are ubiquitous in nature and engineering. Chaotic dynamics, where small perturbations in the initial conditions lead to exponentially diverging trajectories, arise across many scientific disciplines, including astrophysics (Geisel 1985), meteorology (Selvam 2017; E. N. Lorenz 1963; Palmer 1993), chemistry (Field et al. 1993), turbulence (Manneville 2010; Srinivasan et al. 2019; Doan, Polifke, and Magri 2019a), and thermoacoustics (Huhn and Magri 2020c; Huhn and Magri 2020a). In such systems, the long-term predictability is fundamentally limited by the sensitivity to the initial conditions, making forecasting and control challenging. Important examples include weather and climate prediction (Holmstrom, D. Liu, and Vo 2016; Rolnick et al. 2022; O. Ahmed, Qazi, and Magri 2026), financial time-series forecasting (Sezer, Gudelek, and Ozbayoglu 2020), and thermoacoustic instabilities (Huhn and Magri 2020c).

Traditionally, the modelling of complex and chaotic dynamical systems has relied on solving differential equations using finite-difference, finite-element, and spectral methods. Over the past few decades, advances in computational fluid dynamics (CFD) and high-performance computing have significantly expanded the ability to simulate and analyse complex physical systems across science and engineering, yet these approaches remain computationally demanding, often requiring substantial resources to resolve the relevant scales of the dynamics faithfully. The growing availability of large observational and simulation datasets, together with rapid progress in machine learning and modern computing hardware, has accelerated the emergence of data-driven approaches for modelling and forecasting nonlinear dynamics (Brunton, Proctor, and Kutz 2016; Brunton and Kutz 2022; Pathak, B. Hunt, et al. 2018a; LeCun 2019). These methods infer the underlying system behaviour directly from data, often at a fraction of the cost of conventional physics-based simulation.

A central motivation behind the development of machine learning algorithms for scientific applications is the question: *how can the turnaround time of industrial simulations and forecasting tasks be reduced?* While early results have demonstrated promising applications of machine learning for modelling and

forecasting nonlinear dynamical systems, significant challenges remain before such methods can fully replace large-scale physics-based simulations. In particular, many existing approaches struggle with scalability to high-dimensional systems, generalisation to unseen dynamical regimes, and physical interpretability. Addressing these challenges may require looking beyond conventional computing paradigms altogether. In this context, quantum computing has emerged as a potential alternative framework, offering new computational primitives that could prove well-suited to the demands of learning and simulating complex nonlinear dynamics. This thesis explores quantum machine learning approaches for modelling and forecasting nonlinear dynamical systems, as well as quantum computational methods based on linear embedding formulations for directly solving the differential equations governing chaotic dynamical systems.

1.1 Data-driven forecasting of chaotic dynamics and extreme events

Data-driven prediction of chaotic dynamical systems has gained significant interest in the last decade (Cheng et al. 2015; Ghadami and Epureanu 2022). A data-driven model learns the solution of a dynamical system from data with the goal of predicting its temporal evolution. The use of data-driven methods for predicting nonlinear systems has been motivated mainly by two reasons: (a) the availability of large amounts of data for these systems, and (b) the difficulty of predicting temporal dynamics due to chaos.

Predicting the temporal evolution of a chaotic dynamical system is challenging because infinitesimal errors and perturbations grow exponentially in time. This makes the time-accurate prediction of chaotic systems challenging beyond the predictability time (Boffetta et al. 2002), which is a characteristic time scale of the physical system. On the other hand, chaotic solutions of ergodic systems can also be characterised by quantities that are unaffected by infinitesimal perturbations; e.g., the statistics of the solution, and invariant properties such as the Lyapunov spectrum (Shimada and Nagashima 1979; Benettin et al. 1980), the Kaplan–Yorke dimension (Frederickson et al. 1983), covariant Lyapunov vectors (Ginelli, Chaté, et al. 2013; Margazoglou and Magri 2023), and many more (Boffetta et al. 2002).

As argued in (O. Ahmed, Tennie, and Magri 2025b; Magri, Novoa, and Özalp 2024; Özalp and Magri 2025), a data-driven model for chaotic time series is "good" when it (i) predicts the evolution of the dynamics for a relatively long horizon in an autonomous way (i.e., after training), (ii) predicts the long-term statistics of the solution, and (iii) infers the invariant properties of the solution.

The forecasting of chaotic dynamics is a sequential-data problem. Recurrent Neural Networks (RNNs) are therefore a natural choice for chaotic time-series forecasting. RNNs are machine learning models that introduce recurrence in conventional neural network architectures. RNNs are employed in time-series forecasting, modelling nonlinear dynamical systems and chaos, and extreme event predictions (Vlachas, Pathak, et al. 2020; Srinivasan et al. 2019; Racca and Magri 2022b; Doan, Polifke, and Magri 2019c; Margazoglou and Magri 2023; Racca and Magri 2021; Özalp, Margazoglou, and Magri 2023a). While RNNs offer strong computational capabilities, they are difficult to train because they require backpropagation through time at each step (Werbos 1990), a process that becomes increasingly challenging for long time-series forecasting problems (Vlachas, Pathak, et al. 2020). Among the architectures that have been shown to satisfy the learnability criteria outlined above, reservoir computing has emerged as a particularly compelling approach, offering the forecasting capabilities of recurrent networks while

bypassing the need for backpropagation entirely.

1.1.1 Reservoir computing and generalised synchronisation

Reservoir computing (RC) bypasses the problem of backpropagation by introducing a fixed, randomly initialised recurrent dynamical system - the reservoir - whose internal weights are never trained, with only the linear output readout optimised via ridge regression. Reservoir computing is a unified framework introduced independently as Echo State Networks (Jaeger 2001) and Liquid State Machines (Maass, Natschläger, and Markram 2002). In this thesis, we work with Echo State Networks owing to their computational efficiency and strong performance across a range of tasks; for brevity, we refer to Echo State Networks as reservoir computers throughout.

Reservoir computers have demonstrated strong performance in short-term prediction, long-term statistical reconstruction, and stability analysis of chaotic systems (Pathak, B. Hunt, et al. 2018b; Doan, Polifke, and Magri 2019c; Vlachas, Pathak, et al. 2020; Boedecker et al. 2012). They have been employed to predict scalar invariant properties such as the Lyapunov spectrum (Pathak, Lu, et al. 2017; Vlachas, Byeon, et al. 2018; Vlachas, Pathak, et al. 2020), geometric invariant properties such as covariant Lyapunov vectors (Margazoglou and Magri 2023), and sensitivities of objective functionals to physical parameters (Ozan and Magri 2024), for systems including the Lorenz system and the Kuramoto–Sivashinsky equation. Reservoir computers have also shown promise in predicting extreme events (Doan, Polifke, and Magri 2019b; Racca and Magri 2022a; Doan, Polifke, and Magri 2021) and inferring chaotic properties in the latent space (Özalp and Magri 2025; Paine, Elfving, and Kyriienko 2023). Despite this strong performance, the capacity of reservoir computers is fundamentally limited by the reservoir size, a constraint tied directly to the memory and computational resources of classical hardware.

Beyond computational efficiency, a key advantage of reservoir computers over black-box machine learning models is that their *ansatz* is principled and theoretically interpretable, being grounded in the synchronisation theory of chaotic systems (Pecora and Carroll 1990; Pecora and Carroll 2015). The theory of synchronisation states that the synchronisation between two chaotic systems can be achieved by designing a coupling between them. In the case of one-way coupling, a driving system (the physical system) provides the forcing to a higher-dimensional response system (Pecora and Carroll 1991a; Badola, Tambe, and Kulkarni 1992). The conditions under which the response system synchronises with the driving system are provided by generalised synchronisation (GS) theory (Rulkov et al. 1995; Kocarev and Parlitz 1996; S. Yang and Duan 1998; Abarbanel, Rulkov, and Sushchik 1996), where the properties of the coupling map between the drive and response are influenced by the strength of synchronisation (B. R. Hunt, Ott, and Yorke 1997; Tang et al. 1998). Recent works have shown that, when certain conditions are met, classical reservoir computers are generalised synchronisation systems, with the training time series acting as the driving state and the reservoir state acting as the response (Lu, B. R. Hunt, and Ott 2018a; Lymburn et al. 2019; Platt, Wong, et al. 2021).

Generalised synchronisation is also closely connected to the echo state property (ESP), which must be satisfied for a reservoir computer's performance to be independent of its initial state - a necessary condition for reliable time-series forecasting (Jaeger 2001; Jaeger et al. 2007). When the ESP is fulfilled, the reservoir faithfully encodes the attractor structure of the driving system, enabling accurate reconstruction of both short-term trajectories and long-term invariant properties. The stability and learnability of

reservoir computers can therefore be rigorously analysed through the combined framework of GS and Lyapunov stability theory (Margazoglou and Magri 2023; Huhn and Magri 2020c).

The reservoir size, however, remains a fundamental bottleneck: larger reservoirs improve capacity but rapidly exhaust the memory and compute resources of classical hardware. This motivates the exploration of quantum reservoir computers, in which the exponentially large Hilbert space of a quantum system serves as the reservoir. In this thesis, we import the theory of generalised synchronisation into the quantum setting to provide a principled foundation for quantum reservoir computing.

1.2 Quantum computing and machine learning

Quantum computing offers a fundamentally different computational paradigm, leveraging superposition, entanglement, and interference to enable speed-ups over classical methods for certain classes of problems (Feynman 2018). This potential enhancement is commonly referred to as *quantum advantage*. Several quantum algorithms have theoretically demonstrated such advantages, most notably Shor's algorithm for integer factorisation, which achieves an exponential speed-up over the best known classical algorithms (Shor 1994), and quantum algorithms for linear systems and matrix inversion, which underpin many numerical and machine learning methods (Harrow, Hassidim, and Lloyd 2009; Bravo-Prieto et al. 2020).

Recent years have seen growing interest in applying quantum computing to machine learning. Current quantum hardware, however, remains limited by the relatively small number of available qubits, environmental noise, and decoherence; devices in this regime are commonly referred to as Noisy Intermediate-Scale Quantum (NISQ) systems (Bharti et al. 2022). In this context, hybrid quantum-classical methods have emerged as leading candidates for near-term quantum machine learning, combining parametrised quantum circuits with classical optimisation routines, as in Variational Quantum Algorithms and Quantum Circuit Learning (Cerezo et al. 2021; Mitarai et al. 2018). Several of these approaches are inspired by classical machine learning architectures, including recurrent neural networks and reservoir computing, and have been applied to nonlinear dynamical systems, chaotic forecasting, and extreme event prediction (Pfeffer, Heyder, and Schumacher 2022; O. Ahmed, Tennie, and Magri 2024b; Fujii and Nakajima 2020).

1.2.1 Quantum reservoir computing

Quantum reservoir computing (QRC) is a class of quantum machine learning models that combines the principles of quantum computing and reservoir computing. The central idea of QRC is to exploit quantum dynamics to enhance the reservoir implementation (Fujii and Nakajima 2017; Fujii and Nakajima 2021; Chen et al. 2022; Mujal, Martínez-Peña, Nokkala, et al. 2021). In these approaches, the exponentially large Hilbert space of a quantum system acts as the reservoir, generating high-dimensional nonlinear feature representations for temporal information processing.

Several numerical studies have shown that quantum systems with only a few qubits can exhibit computational capabilities comparable to much larger classical reservoirs (Fujii and Nakajima 2017). Previous works have proposed hybrid QRC architectures for time-series prediction, nonlinear dynamics, and chaos modelling (Pfeffer, Heyder, and Schumacher 2022; Suzuki et al. 2022; Fry et al. 2023; O. Ahmed, Tennie, and Magri 2024b). In hybrid QRC approaches, input data are encoded into quantum states through

parametrised qubit rotations, after which the quantum reservoir evolves under unitary dynamics. Reservoir states are then extracted either from quantum probability amplitudes or from measured expectation values of observables (Pfeffer, Heyder, and Schumacher 2022; Dudas et al. 2023). Quantum reservoir computers have also been applied to forecasting on both classical (Fujii and Nakajima 2021; Mujal, Martínez-Peña, Giorgi, et al. 2023a; O. Ahmed, Tennie, and Magri 2024b) and quantum data (Sornsaeng, Dangniam, and Chotibut 2023), including implementations on noisy near-term quantum hardware (Fry et al. 2023; Domingo, Carlo, and Borondo 2023). Further applications include financial forecasting (Domingo, Grande, et al. 2023) and molecular property prediction (Beaulieu et al. 2024).

Compared to classical reservoir computing approaches (Lukosevicius 2012; Jaeger 2001; Racca and Magri 2022a), QRCs offer the potential for richer feature representations through quantum superposition and entanglement, which can enhance the expressivity of the reservoir dynamics. Furthermore, because reservoir computers train only a linear readout layer, QRCs avoid backpropagation through time and are therefore less affected by vanishing gradients and barren plateau phenomena that commonly arise in variational quantum algorithms (Cerezo et al. 2021; McClean et al. 2018).

Despite these promising properties, several challenges remain. Only a limited number of studies have investigated high-dimensional chaotic systems (Tran and Nakajima 2020), long-term statistical forecasting with QRCs, or extreme-event forecasting. In addition, practical implementations are affected by finite sampling noise (O. Ahmed, Tennie, and Magri 2025a; Mujal, Martínez-Peña, Giorgi, et al. 2023a), exponential concentration effects (Xiong et al. 2023), and the need for QRC design strategies that are compatible with NISQ devices.

Among these challenges, finite measurement sampling noise is particularly important for achieving any quantum advantage. Since reservoir states are reconstructed from a finite number of quantum measurements, statistical fluctuations can affect the stability, learnability, and forecasting accuracy of QRC models. Understanding and mitigating these effects is therefore essential for developing scalable quantum reservoir computing implementations.

1.2.2 Measurement problem and sampling noise

Unlike classical reservoir computers, where reservoir states are deterministically available, quantum reservoir computers require repeated measurements to reconstruct the reservoir observables. This process introduces intrinsic statistical fluctuations due to the finite number of measurement shots available on quantum hardware, which can degrade the learnability, stability, and forecasting performance of QRC models.

Crucially, finite sampling noise represents a device-independent source of uncertainty. Since it originates from the probabilistic nature of quantum measurements themselves, it introduces a fundamental limit to learning performance (F. Hu et al. 2023; Mujal, Martínez-Peña, Giorgi, et al. 2023b) that is expected to remain relevant even for future fault-tolerant quantum computers (FTQC) (Preskill 1998a).

In any practical quantum hardware implementation, obtaining information about the density operator ρ of a quantum system requires repeated measurements on multiple copies of the quantum state, a process known as quantum state tomography (Nielsen and Chuang 2011). Full quantum state tomography scales exponentially with system size, with computational complexity $O(D^2)$, where $D = 2^n$ for a system

of n qubits. To address this limitation, (Aaronson 2018) introduced shadow tomography, showing that complete state reconstruction is often unnecessary when the objective is only to estimate specific properties efficiently. This idea was later extended through the framework of classical shadows (Huang, Kueng, and Preskill 2020), which enables the estimation of many observables from relatively few measurements.

However, many QML applications require access to a large number of expectation values for classification and regression tasks (Schuld and Petruccione 2021). In such settings, the advantages of shadow tomography become less pronounced because accurate learning still requires sufficiently precise estimation across many features.

Several approaches have been proposed to mitigate finite sampling noise in quantum machine learning systems, including weak measurements (Mujal, Martínez-Peña, Giorgi, et al. 2023b), classical shadows (Huang, Kueng, and Preskill 2020), Resolvable Expressive Capacity (REC) (F. Hu et al. 2023), and variance regularisation techniques for suppressing probabilistic noise in quantum neural networks (Kreplin and Roth 2024). Previous studies have also investigated the effect of finite sampling noise in physical QRC implementations (Khan, F. Hu, et al. 2021; Dudas et al. 2023).

The impact of finite sampling noise, however, depends strongly on the underlying QRC architecture. Gate-based recurrence-free quantum reservoir computers (RF-QRCs) (O. Ahmed, Tennie, and Magri 2024b; O. Ahmed, Tennie, and Magri 2025a) provide scalable and computationally efficient frameworks that are well-suited for systematically studying the influence of measurement noise on learning dynamics and forecasting performance. Despite recent progress, the effect of finite sampling noise on the propagation and spreading of temporal correlations in recurrent quantum machine learning systems remains relatively unexplored.

This thesis investigates the role of finite sampling noise in the stability, learnability, and forecasting performance of QRC models, examining how sampling noise affects the propagation of temporal correlations through the framework of generalised synchronisation (GS), and exploring mitigation strategies within both recurrent and recurrence-free QRC architectures alongside the hardware implementation.

1.3 Quantum computing for nonlinear differential equations

Alongside data-driven quantum machine learning approaches for forecasting chaotic dynamics, such as quantum reservoir computers and quantum neural networks, there has also been increasing interest in developing direct quantum computational methods for solving nonlinear differential equations (Preskill 1998a; Tennie, Laizet, et al. 2025; Jin, N. Liu, and Y. Yu 2023a; Bravo-Prieto et al. 2020; Gilyén et al. 2019). In contrast to data-driven forecasting models, these approaches aim to reformulate the governing equations of nonlinear dynamical systems into forms compatible with quantum computation, providing a potential alternative framework for scientific computing and numerical simulation.

The interest in developing direct quantum computational methods for differential equations existed even before the emergence of machine learning methods for scientific computing. Starting from the original idea of simulating physical systems using quantum computers (Feynman 1982), significant effort has long been devoted to developing quantum algorithms for solving partial differential equations (PDEs) and the large-scale linear systems arising from their discretisation (Harrow, Hassidim, and Lloyd 2009;

Lloyd et al. 2021). This is motivated by the fact that many classical numerical methods for PDEs suffer from the curse of dimensionality, with computational cost increasing rapidly with system size and spatial resolution.

The development of quantum linear algebra algorithms, most notably the HHL algorithm (Harrow, Hassidim, and Lloyd 2009), demonstrated that quantum computers could, in principle, provide exponential speedups for certain sparse and well-conditioned linear systems (Harrow, Hassidim, and Lloyd 2009; Childs, Kothari, and Somma 2017). Since many discretised PDEs can ultimately be expressed as large linear algebraic problems, these developments motivated extensive research into quantum methods for scientific computing. However, these theoretical advantages typically rely on restrictive assumptions regarding sparsity, condition numbers, efficient quantum state preparation, and the extraction of observables rather than the reconstruction of the full classical solution vector.

A major challenge in applying quantum methods to nonlinear systems is that quantum mechanics is fundamentally linear. The Schrödinger equation governs the evolution of quantum states through linear unitary dynamics, while most quantum algorithms are constructed around this linear structure. Nonlinear differential equations, therefore, cannot generally be encoded directly into unitary quantum evolutions, requiring reformulations that either linearise the dynamics or embed them into higher-dimensional linear systems.

One important approach is based on the Koopman operator formalism, which represents nonlinear dynamical systems as linear operators acting on an infinite-dimensional space of observables (Koopman 1931; Mezić 2005). Rather than evolving the system state directly, the Koopman operator evolves observable functions forward in time, analogous to the Heisenberg picture of quantum mechanics, where observables evolve under linear operator dynamics (Heisenberg 1949; Bondar, Gay-Balmaz, and Tronci 2019). This reformulates a nonlinear finite-dimensional dynamical system into a linear infinite-dimensional one. In practice, finite-dimensional approximations of the Koopman operator are typically constructed using methods such as Dynamic Mode Decomposition (DMD) and Extended DMD (EDMD) (Schmid 2010; Kutz et al. 2016; Schmid 2022), which introduce truncation errors and are sensitive to the choice of basis functions. The resulting operator-theoretic structure is naturally compatible with quantum linear algebra, and several quantum algorithms have been proposed for approximating Koopman spectra (Klus, Nüske, and Peitz 2022) and simulating nonlinear dynamics on quantum hardware (Bondar, Gay-Balmaz, and Tronci 2019), enabling spectral analysis and forward simulation of such systems.

The Koopman–von Neumann (KvN) formulation extends this operator-theoretic perspective by embedding classical Hamiltonian dynamics into a Hilbert-space framework analogous to quantum mechanics (Koopman 1931; Mauro 2002; Joseph 2020). In the KvN formulation, the system evolves through a linear unitary equation for a complex phase-space state function, whose modulus squared corresponds to the classical probability density. This construction provides a direct analogy with the Schrödinger and von Neumann formulations of quantum mechanics while preserving the underlying classical dynamics (Bondar, Gay-Balmaz, and Tronci 2019). Since the governing evolution remains linear and unitary, the KvN framework is well suited to quantum simulation methods.

Carleman linearisation provides another route for embedding nonlinear dynamical systems into linear representations by expanding products of state variables into an infinite hierarchy of coupled linear equations

(Carleman 1932; Kowalski and Steeb 1991). By truncating this hierarchy, nonlinear ordinary differential equations can be approximated by finite-dimensional linear systems that are more amenable to quantum linear algebra and Hamiltonian simulation methods. Several quantum algorithms based on Carleman embeddings have been proposed for solving nonlinear differential equations on quantum computers (Sanavio et al. 2024; Jin, N. Liu, and Y. Yu 2023a). However, the dimensionality of the resulting linear system typically grows rapidly with the truncation order and the number of nonlinear interaction terms, making accurate representations of strongly nonlinear or chaotic systems computationally challenging.

Despite the theoretical promise of quantum algorithms for differential equations, several works have argued that efficiently simulating nonlinear and chaotic dynamics on quantum computers remains fundamentally difficult (Jin, N. Liu, and Y. Yu 2023b; J.-P. Liu et al. 2021; Lewis et al. 2024). In chaotic systems, exponentially growing perturbations amplify discretisation, truncation, and measurement errors, often requiring exponentially increasing precision or computational resources to maintain long-time trajectory accuracy. Furthermore, many quantum algorithms provide access only to expectation values or compressed quantum states rather than the full classical solution field, limiting their practical utility for scientific computing applications where complete spatiotemporal solutions are required. However, we emphasise that many of these complexity bounds are derived for direct nonlinear trajectory simulation and do not explicitly consider formulations in which the nonlinear dynamics are first reformulated into linear operator evolution equations before quantum implementation (Lewis et al. 2024).

One possible alternative is to reformulate nonlinear dynamics in terms of probability density evolution rather than individual trajectories. In this framework, the dynamics are governed by linear evolution equations such as the Fokker–Planck equation, which describes the transport and diffusion of probability densities in phase space (Risken 1989; Tikhonenkov et al. 2013). In the deterministic limit, the Fokker–Planck equation reduces to the Liouville equation, which governs the conservative transport of probability density along system trajectories and connects directly to the Koopman–von Neumann framework discussed above. Instead of predicting a single deterministic trajectory, the evolution of the full probability distribution is solved directly, naturally connecting to ensemble forecasting and uncertainty quantification. Such formulations are particularly relevant for chaotic and turbulent systems, where long-term trajectory prediction becomes intractable due to sensitivity to initial conditions, while the statistical evolution of the system remains physically meaningful. Since the governing evolution equation remains linear, Fokker–Planck formulations are also naturally compatible with quantum linear algebra and Hamiltonian simulation methods. Previous works have proposed quantum implementations of Fokker–Planck-type dynamics for stochastic and over-damped systems (Tennie and Magri 2024; Clemente, Crippa, and Jansen 2023), motivating the extension of these approaches to nonlinear chaotic dynamical systems, which we present in this thesis.

Motivated by these developments, we investigate Fokker–Planck embedding formulations for chaotic differential equations and their quantum-compatible implementations. In contrast to trajectory-based quantum simulation methods, the objective here is to directly evolve the probability-density dynamics associated with chaotic systems, enabling probabilistic forecasting and statistical characterisation of nonlinear dynamics. This perspective is also closely connected to modern ensemble weather and climate forecasting systems, where probabilistic evolution is often more physically meaningful than deterministic long-time trajectory prediction. Furthermore, the same class of linear probability-density evolution

equations underlies diffusion-based generative models and stochastic machine learning frameworks, broadening the relevance of quantum-compatible methods to modern machine learning applications.

Despite this potential, existing quantum algorithms for differential equations have primarily focused on asymptotic complexity scaling and linear Hermitian formulations, with comparatively little attention paid to the numerical properties of discretisation schemes for chaotic and dissipative systems. In particular, the role of numerical diffusion and overdamping in suppressing fine-scale chaotic structures has remained largely unexplored within quantum-compatible discretisation frameworks. In this thesis, we address this gap by developing and evaluating higher-order upwind discretisation schemes for chaotic probability-density dynamics within sparse quantum-compatible block-encoding formulations.

1.4 Thesis objectives

The overarching goal of this thesis is to investigate quantum machine learning and quantum computational approaches for forecasting, stability analysis, and solving nonlinear dynamical systems, with particular emphasis on chaotic dynamics and extreme-event prediction. The thesis draws on nonlinear dynamics, synchronisation theory, reservoir computing, statistical mechanics, and quantum computing to develop physically grounded frameworks that connect data-driven forecasting with equation-based quantum simulation of chaotic systems. The objectives of this thesis are:

1. Establish a theoretical connection between generalised synchronisation (GS), the echo state property (ESP), and the learnability and stability of chaotic dynamics in quantum reservoir computers;

2. Derive analytical Jacobians and stability measures for recurrent and recurrence-free quantum reservoir computing (RF-QRC) architectures, enabling the inference of invariant and stability properties of chaotic attractors directly from trained quantum reservoirs;

3. Propose design principles for robust quantum reservoir computers using generalised synchronisation theory, conditional Lyapunov exponents, and contractive quantum dynamics;

4. Develop and analyse quantum reservoir computing architectures for forecasting low- and high-dimensional chaotic systems, including nonlinear systems exhibiting rare and extreme events;

5. Investigate the influence of finite sampling and hardware noise on the performance of quantum reservoir computers, develop denoising strategies suitable for noisy near-term quantum devices, and demonstrate the training of a prototypical chaotic extreme-event system on quantum hardware;

6. Develop quantum computational approaches for directly solving nonlinear differential equations using linear embedding formulations, including the Fokker–Planck and Koopman–von Neumann approaches, extend these methods to chaotic systems through the analysis of discretisation schemes and numerical diffusion, and implement unitary block-encoding formulations for simulation on near-term quantum hardware.

Collectively, this thesis contributes to both quantum machine learning methods for chaotic forecasting and operator-theoretic quantum approaches for solving nonlinear dynamical systems on near-term quantum hardware.

1.5 Thesis structure

The work presented in this thesis is based on the following journal publications (O. Ahmed, Tennie, and Magri 2024b; O. Ahmed, Tennie, and Magri 2025b; O. Ahmed, Tennie, and Magri 2025a) and conference talk (O. Ahmed, Tennie, and Magri 2024a) along with several other conference and workshop talks. The remainder of the thesis is organised as follows.

Chapter 2 focuses on the theoretical foundations of chaotic dynamical systems and presents generalised synchronisation (GS) as a framework for analysing the learnability of chaotic systems. We review Lyapunov exponents, invariant properties of chaotic attractors, and data-driven forecasting approaches, before introducing reservoir computing and the GS criteria used throughout the thesis.

Chapter 3 extends the GS framework to quantum reservoir computing (QRC). We first introduce the basic principles of quantum computing and hybrid quantum–classical reservoir computers, and then present recurrence-free quantum reservoir computers (RF-QRCs) as an efficient forecasting framework. The chapter further develops GS-based design principles for robust QRC architectures and analyses the stability of quantum reservoir updates through Jacobian formulations.

Chapter 4 applies QRC methods to forecasting chaotic dynamics and extreme events. We investigate the performance of different QRC architectures on prototypical nonlinear systems, including the Lorenz–63, Lorenz–96, and Moehlis–Faisst–Eckhardt (MFE) systems. Short-term prediction, long-term statistical forecasting, and extreme-event prediction capabilities are analysed, together with the inference of invariant quantities such as Lyapunov spectra and attractor dimensions.

Chapter 5 investigates the effects of finite sampling noise and hardware noise in QRCs. We develop theoretical descriptions of measurement noise in quantum reservoirs, analyse its impact on forecasting performance, and introduce denoising and mitigation strategies. These methods are then validated on noisy quantum hardware implementations.

Chapter 6 studies noise from the perspective of generalised synchronisation and demonstrates that noise can also act as a resource for robust reservoir design. We analyse the dependence of Lyapunov exponents and conditional Lyapunov exponents on reservoir hyperparameters and noise strength, and investigate how noise influences the stability, learnability, and robustness of QRC dynamics.

Chapter 7 presents quantum computational approaches for solving nonlinear differential equations using linear embedding formulations and their extension for chaotic systems. We investigate Fokker–Planck and Koopman–von Neumann embeddings of nonlinear dynamics, analyse the role of discretisation schemes and numerical diffusion, and implement unitary block-encoding formulations for simulations on near-term quantum hardware.

Chapter
2

Learnability of chaotic systems and generalised synchronisation (GS)

This chapter reviews the theoretical foundations for chaotic dynamical systems and establishes generalised synchronisation (GS) as a framework for their learnability. We begin by presenting dynamical systems dependence on initial conditions, and chaotic systems characterisation using the largest Lyapunov exponent (Benettin et al. 1980) in Sec. 2.1. In Sec. 2.2 we present stability theory and invariant properties for dynamical systems.

Building on this, we briefly overview data-driven approaches for forecasting chaotic systems and present reservoir computing as a dynamical system well-suited to such tasks. We then propose generalised synchronisation as a framework to analyse the learnability of chaotic systems and derive conditions under which they can be learned from data in a robust way. Overall, this chapter builds on the theoretical contributions proposed in (O. Ahmed, Tennie, and Magri 2025b) and (O. Ahmed, Tennie, and Magri 2024b).

2.1 Chaos and predictability

We consider a nonlinear autonomous dynamical system

$$\frac{d}{dt}\boldsymbol{x}(t) = f(\boldsymbol{x}(t)), \tag{2.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^D$ is the state vector; $f : \mathbb{R}^D \to \mathbb{R}^D$ is a continuously differentiable nonlinear vector function; and D is the number of degrees of freedom. Given an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, the solution of Eq. 2.1 is the trajectory $\boldsymbol{x}(t)$ in the phase space of dimension D.

The nonlinear vector function f in Eq. 2.1 governs the dynamics of the physical system. A key aspect in

analysing such systems is their behaviour in the long-time limit, $t \to \infty$. In many nonlinear dynamical systems, trajectories evolve towards invariant subsets of the phase space, known as *attractors*, which describe the geometric and statistical structure of the dynamics and remain invariant under the system's evolution. More generally, this asymptotic behaviour is characterised by invariant sets in phase space, which capture the long-time behaviour of trajectories.

As a forecasting problem, the objective is to predict the future sampled state $\boldsymbol{x}(t_{i+1})$ from its past observations $\boldsymbol{x}(t_i)$ (Although the underlying physical system evolves continuously in time, in forecasting applications it is often observed at discrete sampling times (t_i)). However, for systems that exhibit sensitivity to initial conditions, small perturbations to initial conditions grow exponentially in time, which limits accurate predictions to a finite horizon. As a result, the long-term behaviour is described in terms of statistical and invariant properties rather than a single trajectory. For chaotic systems, the exponential separation of nearby trajectories is quantified with the leading or largest Lyapunov exponent (Nastac et al. 2017).

2.1.1 Largest Lyapunov exponent

Consider two initial conditions $\boldsymbol{x}_0$ and $\boldsymbol{x}_0^*$ that are infinitesimally close to each other, with initial separation $\boldsymbol{w}(0) = \boldsymbol{x}_0 - \boldsymbol{x}_0^*$. Their corresponding trajectories $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t) + \boldsymbol{w}(t)$ evolve in time, where $\boldsymbol{w}(t)$ denotes the perturbation vector as shown in Fig. 2.1 where $\|\boldsymbol{w}(0)\| \sim O(\varepsilon)$ with $\varepsilon \to 0$

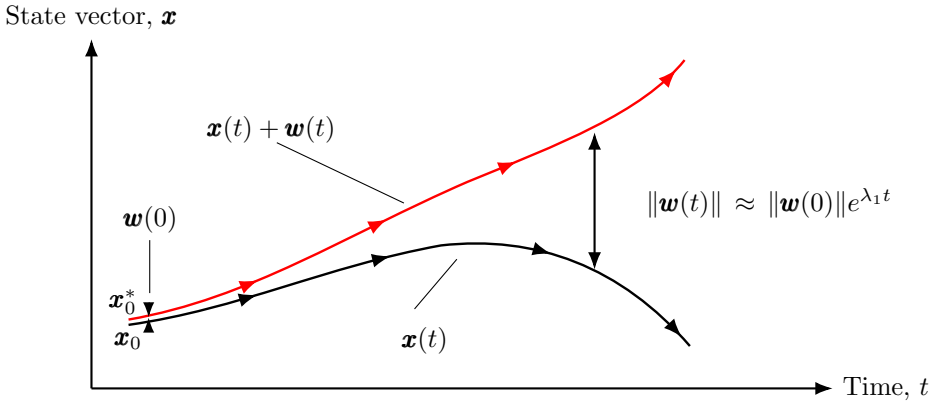


Figure 2.1: Illustration of exponential growth of an infinitesimal perturbation between nearby trajectories, characterised by the largest Lyapunov exponent λ_1 e.g. (Frederickson et al. 1983; Nastac et al. 2017).

The largest Lyapunov exponent λ_1 quantifies the average exponential rate of growth (or decay) of such infinitesimal perturbations (Boffetta et al. 2002). After a transient, the perturbation evolves as

$$\|\boldsymbol{w}(t)\| \sim \|\boldsymbol{w}(0)\| e^{\lambda_1 t}. \tag{2.2}$$

Thus, the largest Lyapunov exponent is defined as

$$\lambda_1 = \lim_{t \to \infty} \frac{1}{t} \ln \left(\frac{\|\boldsymbol{w}(t)\|}{\|\boldsymbol{w}(0)\|} \right), \tag{2.3}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Based on the value of λ_1, the long-time behaviour of the system can be classified as (Huhn and Magri 2020d; Margazoglou and Magri 2023):

- If $\lambda_1 < 0$, perturbations decay and the attractor is a fixed point,
- If $\lambda_1 = 0$, the system exhibits periodic or quasi-periodic behaviour (if there exist at least two neutral Lyapunov exponents),
- If $\lambda_1 > 0$, trajectories diverge exponentially and the system is chaotic.

The largest Lyapunov exponent also defines a characteristic predictability timescale (the Lyapunov time), $\tau_\lambda \sim 1/\lambda_1$, beyond which accurate prediction of trajectories is generally not possible. The exponential growth of perturbations, therefore, limits the predictability of chaotic systems to a finite time horizon.

Therefore, chaotic systems are often characterised by statistical and invariant properties. These include quantities such as the Lyapunov spectrum, attractor dimension, e.g. Kaplan-Yorke dimension, and covariant Lyapunov vectors.

2.2 Stability theory and invariant properties

In linear stability analysis, we analyse the evolution of an infinitesimal perturbation $\boldsymbol{w}(t)$ applied to the state $\boldsymbol{x}(t)$ as

$$\boldsymbol{x}(t) + \boldsymbol{w}(t), \qquad \varepsilon \to 0. \tag{2.4}$$

Substituting Eq. 2.4 into Eq. 2.1 and retaining the first-order terms yields

$$\frac{d}{dt}\boldsymbol{w}(t) = \mathbf{J}(\boldsymbol{x}(t))\boldsymbol{w}(t), \quad \boldsymbol{w}(0) = \boldsymbol{w}_0, \tag{2.5}$$

where $\mathbf{J}$ is the Jacobian, $J_{ij} = \partial f_i/\partial x_j$ with $i, j = 1, 2, \ldots D$. (Eq. 2.5 is known as the perturbation, or Jacobian, or tangent, or variational equation.) The perturbation, $\boldsymbol{w}$, evolves in the tangent space spanned by the Jacobian, which is evaluated around the time-varying state, $\boldsymbol{x}(t)$. The goal of stability analysis is to characterise the growth and directions of infinitesimal perturbations.

To do so, we numerically time march $K \leq D$ pseudorandom vectors, $\boldsymbol{w}_i$, cast as columns of a matrix $\boldsymbol{W}_S = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_K]$

$$\frac{d}{dt}\boldsymbol{W}_S = \mathbf{J}(\boldsymbol{x}(t))\boldsymbol{W}_S, \tag{2.6}$$

Geometrically, Eq. 2.6 describes the evolution of a basis of tangent perturbation vectors along the trajectory $\boldsymbol{x}(t)$. As discussed earlier, almost all nearby trajectories diverge exponentially fast with an average rate given by the largest Lyapunov exponent. Therefore, the tangent vectors also align exponentially fast with the leading Lyapunov vector $\boldsymbol{w}_1$. To address this numerical issue, we periodically orthonormalize the tangent space basis during time evolution using QR decomposition as $\boldsymbol{W}_S = \boldsymbol{Q}\boldsymbol{R}_S$, where $\boldsymbol{Q}$ is an orthonormal matrix, and $\boldsymbol{R}_S$ is an upper-triangular matrix (Golub et al. 1996). The calculation of stability properties for different prototypical systems is further discussed in Chap. 4.

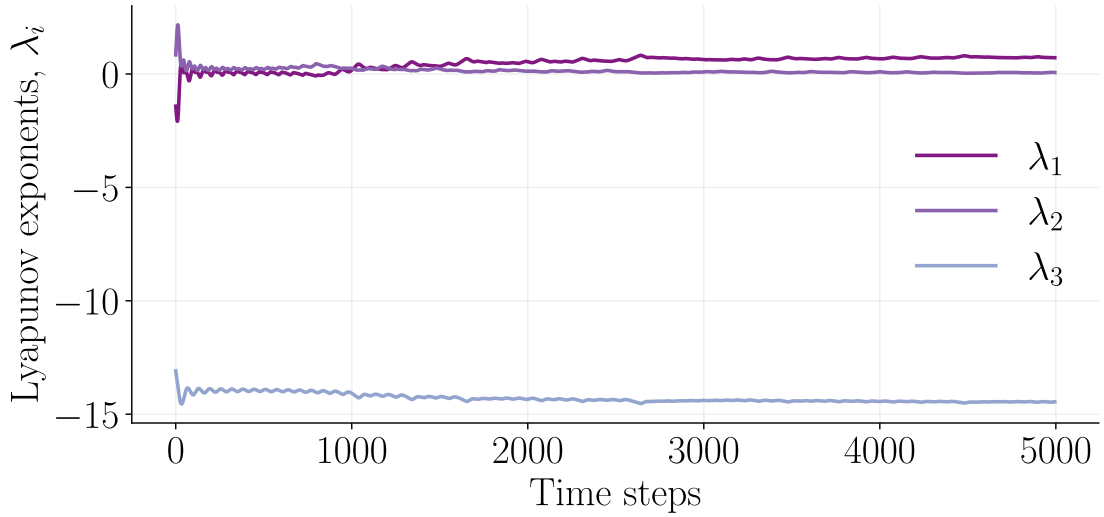


Figure 2.2: Numerical convergence of the Lyapunov exponent estimates for the continuous-time dynamical system of the Lorenz-63 system (E. N. Lorenz 1963), computed from the tangent linear dynamics using periodic QR orthonormalisation.

2.2.1 Lyapunov spectrum

Although the largest Lyapunov exponent governs the system's dynamics, a complete description of the spectral property of the Jacobian is provided by the Lyapunov spectrum. After the QR-decomposition, the columns of the resulting matrix $\boldsymbol{Q}$ constitute an orthonormal basis for the tangent space, known as Gram-Schmidt vectors (GSVs). The diagonal entries of the matrix $\boldsymbol{R}_S$ correspond to local multiplicative growth factors of the perturbations associated with each GSV. These multiplicative growth factors are the local contraction or expansion of infinitesimal perturbations, and their logarithms define the corresponding instantaneous growth rates. Time averaging of these growth rates over long times yields the Lyapunov exponents of the system.

$$\lambda_i = \lim_{T\to\infty} \frac{1}{T} \int_{t_0}^{T} \ln[\boldsymbol{R}_S(t)_{i,i}]dt. \tag{2.7}$$

Oseledets multiplicative ergodic theorem (Oseledec 1968), under mild assumptions, shows the existence of D Lyapunov exponents $\lambda_1 \geq \cdots \geq \lambda_D$.

2.2.2 Kaplan-Yorke dimension

The Lyapunov spectrum and the Kaplan-Yorke dimension are scalar invariants (measures) of the chaotic attractor. The dominant portion of the Lyapunov spectrum provides the Kaplan-Yorke dimension, which is an upper bound of the attractor dimension (Frederickson et al. 1983)

$$D_{KY} = l + \frac{\sum_{i=1}^{l} \lambda_i}{|\lambda_{l+1}|}, \tag{2.8}$$

where l is the largest integer such as $\sum_{i=1}^{l} \lambda_i > 0$ and $\sum_{i=1}^{l+1} \lambda_i < 0$.

2.2.3 Covariant Lyapunov Vectors (CLVs)

Geometric invariants, which describe the vector structure of the tangent space, are given by the covariant Lyapunov vectors (CLVs) (Ginelli, Chaté, et al. 2013; Kuptsov and Parlitz 2012; Ginelli, Poggi, et al. 2007; Consonni and Magri 2026). CLVs, by definition, are covariant with the dynamics and invariant under time reversal.

CLVs provide a (generally) non-orthogonal, local splitting of the tangent space into unstable, neutral, and stable subspaces, corresponding to positive, zero, and negative Lyapunov exponents, respectively. This decomposition allows the identification of the local expanding and contracting directions of the dynamics.

The (absolute) angle between pairs of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ of normalised CLVs is

$$\theta_{\boldsymbol{v}_i,\boldsymbol{v}_j} := \frac{180}{\pi} \cos^{-1}\left(\left|\boldsymbol{v}_i \cdot \boldsymbol{v}_j\right|\right), \quad i,j = 1,2,\ldots,D, \tag{2.9}$$

where $(\cdot)$ denotes the dot product. These angles provide information on the local structure of the CLVs, for example, distinguishing between hyperbolic and non-hyperbolic behaviour, (e.g. Huhn and Magri 2020c).

2.3 Data-driven forecasting of chaotic systems

Data-driven forecasting aims to learn the evolution of a dynamical system directly from time-series data, without explicit knowledge of the governing equations. Several machine learning approaches have been proposed for this task, including recurrent neural networks (RNNs) (Medsker, Jain, et al. 2001; Salehinejad et al. 2017; Arjovsky, Shah, and Bengio 2016), Long Short-Term Memory (LSTM) (Y. Yu et al. 2019; Holmstrom, D. Liu, and Vo 2016; Özalp, Margazoglou, and Magri 2023b) networks and reservoir computers (Platt, Penny, et al. 2022; Racca, Doan, and Magri 2023; Jaeger 2001), which are designed to model temporal dependencies in sequential data.

We focus on reservoir computing as a data-driven framework for forecasting chaotic systems (Racca and Magri 2022a; Racca and Magri 2022b; Doan, Polifke, and Magri 2019a). Reservoir computing provides an efficient alternative to traditional RNN training and has an interpretation as a driven discrete dynamical system, which allows its behaviour to be analysed using tools from dynamical systems theory (Jaeger 2001; Magri, Novoa, and Özalp 2024).

2.3.1 Reservoir computing

Reservoir computing is a type of Recurrent-Neural Network (RNN) that learns temporal correlations from the input data by mapping the low-dimensional input data to a high-dimensional reservoir. Unlike other RNNs, reservoir computing does not require backpropagation through time, which represents a major computational advantage (Jaeger 2001; Racca and Magri 2021). As a result, the training cost is substantially reduced, and the objective function is minimised through a simple linear ridge regression. A particular type of classical reservoir computing, also known as Echo State Network (Jaeger 2001) is a focus of this work. The schematic representation of classical reservoir computers is shown in Fig. 2.3.

Let $\{\hat{\boldsymbol{u}}(t_0), \hat{\boldsymbol{u}}(t_1), \ldots, \hat{\boldsymbol{u}}(t_{N_{tr}})\}$ be a training data set of a dynamical state $\hat{\boldsymbol{u}} \in \mathbb{R}^{N_u}$, which is known at

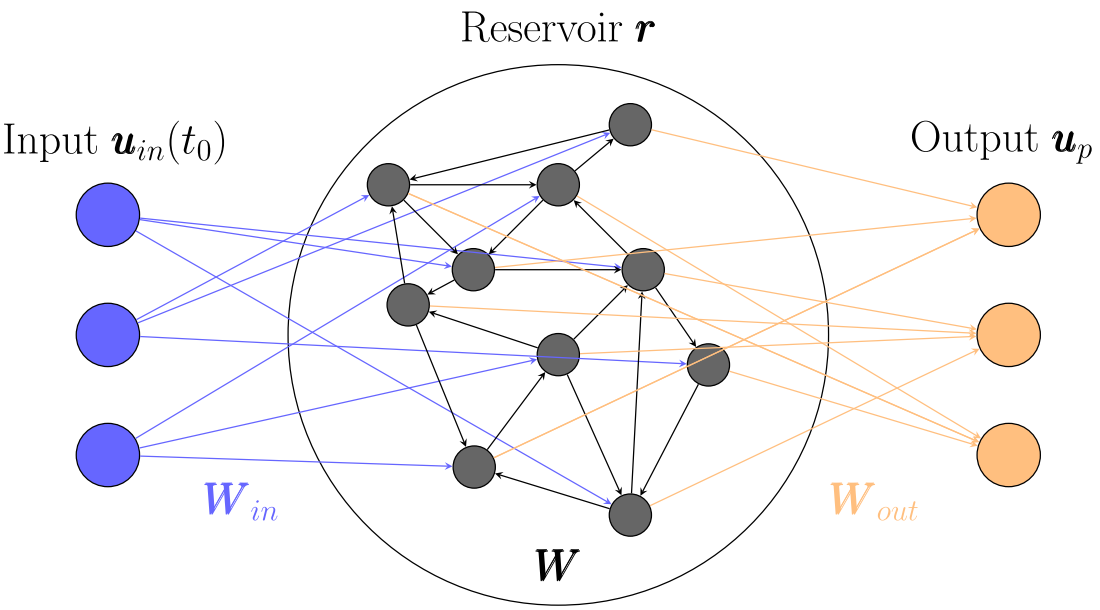


Figure 2.3: Schematic representation of a reservoir computer. The input data $\boldsymbol{u}_{in}$ is mapped to the reservoir matrix via $\boldsymbol{W}_{in}$. The reservoir neuron connections governed by $\boldsymbol{W}$ matrix allow the flow of information between neurons. The linear readout layer, with the trained output weight matrix $\boldsymbol{W}_{out}$, is used to make output predictions $\boldsymbol{u}_p$.

$N_{tr}+1$ steps in time. Generally, it is recommended to choose a set of variables that is (re-)scaled by the range component-wise, which is indicated by ($\hat{}$) on the training data set (Lukosevicius 2012). In reservoir computing, these states are mapped onto higher-dimensional reservoir states of an N_r-dimensional vector space. For a randomly chosen initial reservoir state $\boldsymbol{r}(t_0)$, we recursively compute a set of N_{tr} reservoir vectors $\boldsymbol{r}(t_1),\ldots,\boldsymbol{r}(t_{N_{tr}})$ as

$$\hat{\boldsymbol{r}}(t_{i+1}) = \tanh\left(\boldsymbol{W}_{in}\hat{\boldsymbol{u}}_{in}(t_i) + \boldsymbol{W}\boldsymbol{r}(t_i)\right), \tag{2.10}$$

$$\boldsymbol{r}(t_{i+1}) = (1-\epsilon)\,\boldsymbol{r}(t_i) + \epsilon\,\hat{\boldsymbol{r}}(t_{i+1}). \tag{2.11}$$

Here, the state $\hat{\boldsymbol{r}} \in \mathbb{R}^{N_r}$ represent a state-vector containing reservoir activation states, $\boldsymbol{W}_{in} \in \mathbb{R}^{N_r \times N_u}$ is the input matrix and $\boldsymbol{W} \in \mathbb{R}^{N_r \times N_r}$ is the reservoir weight matrix (Jaeger 2001). The tanh function is applied component-wise to provide a nonlinear activation. The matrices $\boldsymbol{W}_{in}$ and $\boldsymbol{W}$ are pseudo-randomly generated and constant throughout the training and prediction phase. The elements of $\boldsymbol{W}_{in}$ are sampled from a uniform distribution in $[-\sigma_{in}, \sigma_{in}]$, where σ_{in} is the input-scaling. The reservoir weight matrix $\boldsymbol{W}$ is an Erdös-Rényi matrix (Erdös-Rényi is a model for generating random graphs or the evolution of a random network in the field of graph theory), with average connectivity between each neuron varied by input reservoir density D. The spectral radius of the reservoir state matrix, denoted by $\rho(\boldsymbol{W})$, is defined as the maximum absolute value of its eigenvalues. The matrix $\boldsymbol{W}$ is typically rescaled such that $\rho(\boldsymbol{W}) < 1$ (Lukosevicius 2012; Jaeger 2001). This ensures that the reservoir dynamics remain contractive, leading to a decay of temporal correlations between successive reservoir states and reducing the risk of overfitting. As $\rho(\boldsymbol{W})$ approaches unity, the reservoir operates near the edge of stability, increasing memory capacity while weakening the decay of correlations. On the one hand, Eq. 2.10 is the nonlinear activation step. On the other hand, Eq. 2.11 combines the linear memory of the reservoir with the nonlinear activation, as parametrised by the leaking rate ϵ, which is known as the memory non-linearity tradeoff (Lukosevicius

2012).

Open-loop and closed-loop forecasts

Reservoir computing networks can be run in either open-loop or closed-loop configurations (Fig. 2.4). In the open-loop, we use the input data at each time step and compute the corresponding reservoir dynamics $\boldsymbol{r}(t_i)$ according to Eqs. 2.10 and 2.11. Because of the initialization process with a randomly chosen reservoir state, it is necessary to discard a small number N_w of the initial transients to satisfy the Echo State Property (ESP) (Jaeger 2001), which leads to dynamics that are less sensitive to the initial random choice. The process of discarding the initial transients is called the washout phase.

After the washout interval, reservoir dynamics at each time step $\boldsymbol{r}(t_i)$ are collected to form a reservoir state matrix $\boldsymbol{R} \in \mathbb{R}^{N_r \times (N_{tr} - N_W)}$. Reservoir computing in the closed-loop prediction mode requires an output matrix $\boldsymbol{W}_{out}$ for forecasting the dynamics of the learned chaotic system. The training of the output matrix $\boldsymbol{W}_{out}$ involves minimising the mean square error between input and output data over the training data set, consisting of N_{tr} training steps

$$E = ||\boldsymbol{u}(t_{i+1}) - \boldsymbol{r}(t_{i+1})^{\mathrm{T}} \boldsymbol{W}_{out}||^2 + \beta\, ||\boldsymbol{W}_{out}||^2, \tag{2.12}$$

where β is the non-negative Tikhonov regularisation factor, which is a hyperparameter. The simplicity of the reservoir network allows us to achieve single-shot training by solving a linear ridge regression problem

$$(\boldsymbol{R}\boldsymbol{R}^T + \beta \boldsymbol{I})\, \boldsymbol{W}_{out} = \boldsymbol{R}\, \boldsymbol{U}_d^{\mathrm{T}}, \tag{2.13}$$

Here, $\boldsymbol{R} \equiv [\boldsymbol{r}(t_1), \boldsymbol{r}(t_2), ...\boldsymbol{r}(t_{N_{tr}})] \in \mathbb{R}^{N_r \times N_{tr}}$ is a matrix of concatenated reservoir activation signals corresponding to each neuron for N_{tr} time steps of training, and $\boldsymbol{U}_d \equiv [\boldsymbol{u}_{in}(t_1), \boldsymbol{u}_{in}(t_2), ...\boldsymbol{u}_{in}(t_{N_{tr}})] \in \mathbb{R}^{N_u \times N_{tr}}$ is the matrix of concatenated input time series data used for training. Once the output matrix has been computed from Eq. 2.13, it can be used to predict the evolution of dynamical variables by

$$\boldsymbol{u}_p(t_{i+1}) = [\boldsymbol{r}(t_{i+1})]^{\mathrm{T}}\, \boldsymbol{W}_{out}. \tag{2.14}$$

In the closed-loop configuration, we recursively predict the system's dynamics from Eq. 2.14 without additional samples from a training data set. Given an initial condition, this allows for an autonomous prediction on the unseen data set. Because reservoir computers have a symmetry (Eq. 2.10), we add a constant output bias of 1 to the reservoir activation states to break the inherent symmetry of the reservoir architecture (Huhn and Magri 2020b; Herteux and Räth 2020), effectively replacing $\boldsymbol{r} \rightarrow [\boldsymbol{r}, 1]$.

2.3.2 Performance metrics

As discussed in Sec. 2.2, for chaotic systems, the exponential growth of errors due to sensitive dependence on initial conditions limits accurate long-term prediction. As a result, effective forecasting models must not only capture short-term dynamics but also reproduce the statistical and invariant properties of the underlying system. From a data-driven perspective, a model of a chaotic system is therefore considered effective if it can

1. Predict the autonomous short-term evolution of the dynamics

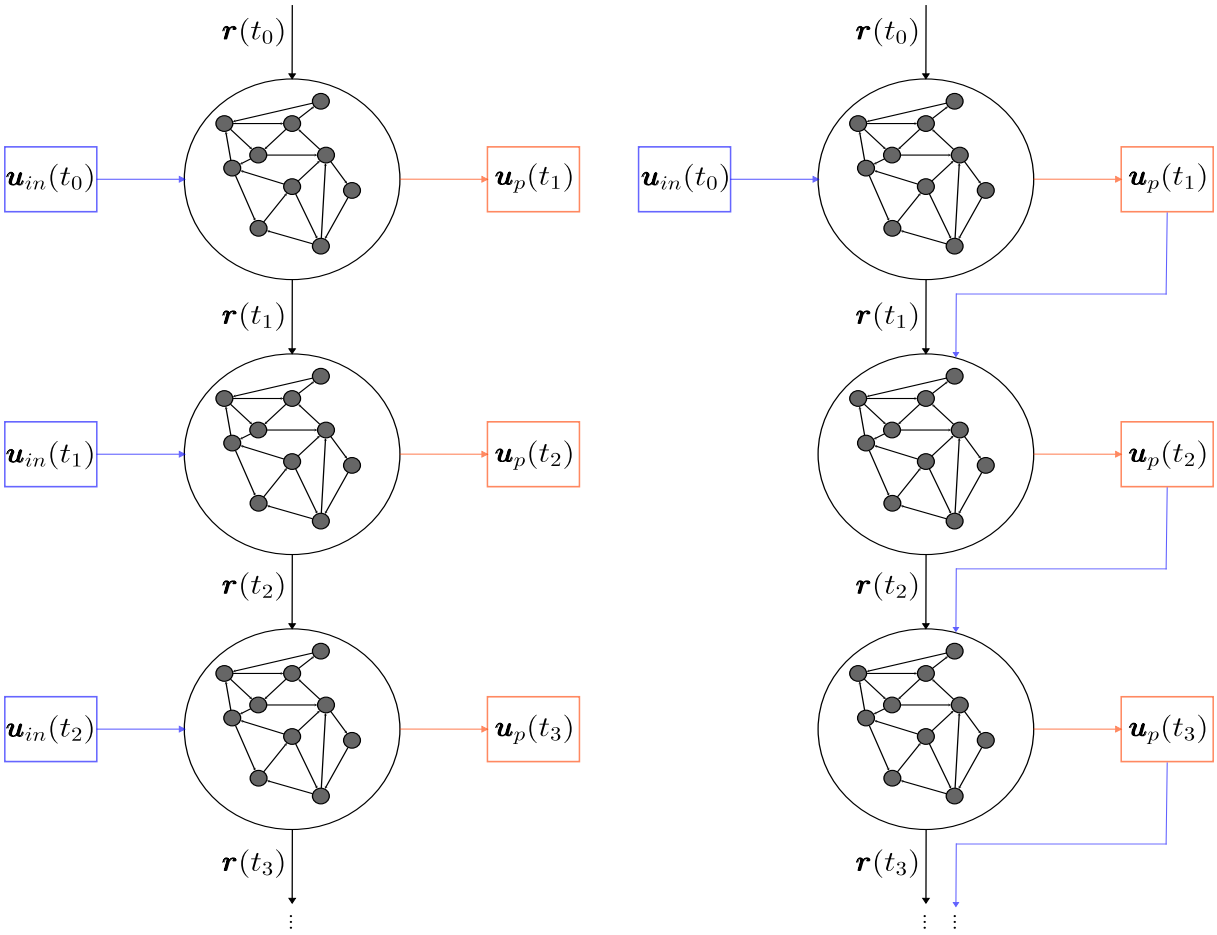


Figure 2.4: Schematic representation of reservoir computers. On the left-hand side, is the open-loop training phase. On the right side, is the closed-loop prediction phase.

2. Infer the long-term stable statistical behaviour

3. Capture the invariant properties of the system.

We define a set of metrics to quantify these characteristics of the reservoir computers. These metrics are defined as follows.

Valid Prediction Time (VPT)

We choose the Valid Prediction Time (VPT) (Vlachas, Pathak, et al. 2020) as a performance measure of short-term time-accurate predictions. For a given threshold value ν, the VPT is defined as the maximum time horizon over which the Normalised Root Mean Square Error (NRMSE) between the predicted and true values remains below ν. In this work, we take $\nu = 0.5$, as in (Vlachas, Pathak, et al. 2020).

$$\mathrm{NRMSE}(t) = \sqrt{\left\langle \frac{\left(\boldsymbol{u}(t) - \boldsymbol{u}_p(t)\right)^2}{\sigma^2} \right\rangle}, \tag{2.15}$$

$$t_f = \max\left\{t \,:\, \mathrm{NRMSE}(t) < \nu\right\}, \tag{2.16}$$

$$\mathrm{VPT} = \frac{t_f}{\tau_\lambda}, \tag{2.17}$$

where $\boldsymbol{u}(t)$ is the true value and $\boldsymbol{u}_p(t)$ is the corresponding predicted value, $\langle\cdot\rangle$ denotes averaging over the state-space dimensions, and σ is the standard deviation of each state variable computed over time. The quantity t_f is the largest time for which NRMSE(t) remains below the threshold ν. The VPT is normalised by the Lyapunov time, $\tau_\lambda = \lambda_1^{-1}$, thereby expressing predictability in units of Lyapunov time.

Higher VPT values indicate greater predictability. As an example of the forecasting problem, the prediction for a three-dimensional Lorenz–63 system is shown in Fig. 2.5. For both the Reservoir Computing Method (RCM) and the Quantum Reservoir Computing Method (QRCM), predicted trajectories diverge from the true trajectory at $\sim$ 4 Lyapunov times (LT), corresponding to a VPT of 4 LT. Since prediction errors grow exponentially at a rate governed by the largest Lyapunov exponent, the VPT provides a practical estimate of the predictability horizon of the system.

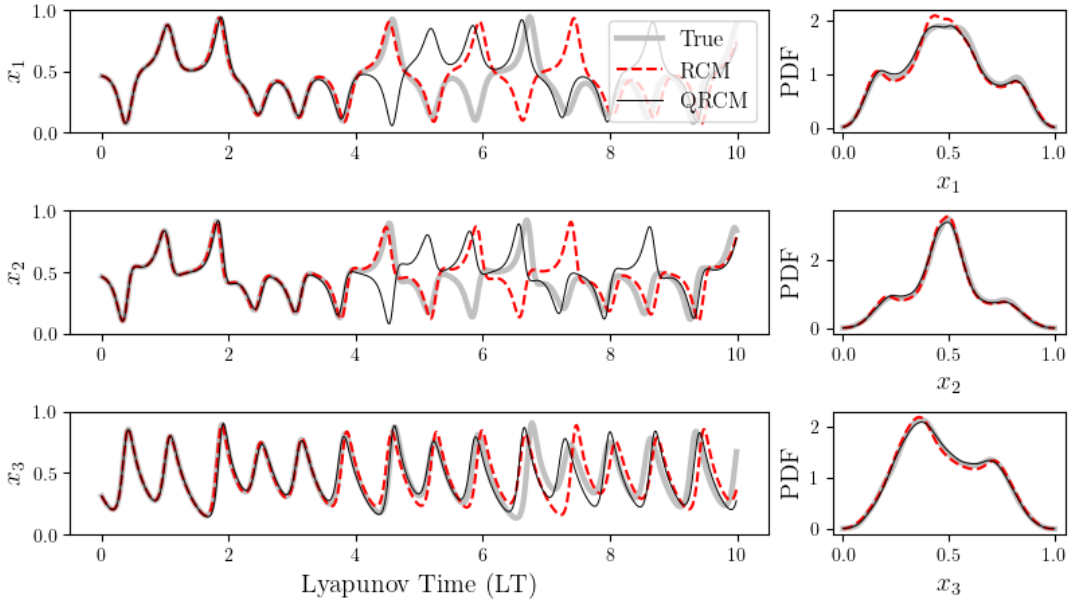


Figure 2.5: Lorenz-63 system. Short-term time series and long-term statistical predictions via Probability Density Functions (PDFs). True trajectory obtained via Runge-Kutta integration of Lorenz-63 ODEs vs forecasted via Reservoir Computing Method (RCM) and Quantum Reservoir Computing Method (QRCM).

Probability Density Functions (PDFs)

To assess the long-term statistical prediction capabilities of the reservoir networks, we compare the distributions of the predicted state variables with the corresponding true distributions. Fig 2.5 shows the PDFs of the three state variables for the predicted and true dynamics.

Invariant measures

Alongside short-term time-accurate and long-term statistical predictions, we also infer the stability properties of the underlying dynamical systems from the trained reservoir computers. We assess whether the learned dynamics reproduce the invariant properties of the underlying system, including the Lyapunov spectrum, Kaplan–Yorke dimension, and CLV angles (Sec. 2.2).

These three criteria respectively assess local trajectory accuracy, statistical consistency, and dynamical

invariance of the learned system.

2.4 Generalised synchronisation (GS)

Two identical autonomous dynamical systems with infinitesimally different initial conditions produce trajectories that diverge (almost surely) exponentially at a rate given by the largest Lyapunov exponent.

Chaotic systems can be coupled through a common driving signal to achieve synchronisation. The phenomenon of synchronisation in chaotic systems was first highlighted in Pecora and Carroll 1990; Pecora and Carroll 1991a and later confirmed experimentally in Tang et al. 1998. The key idea is to consider two dynamical systems with a suitable coupling that ensures asymptotic synchronisation (Verzelli, Alippi, and Livi 2021a). In Fig. 2.6, we illustrate different types of synchronisation.

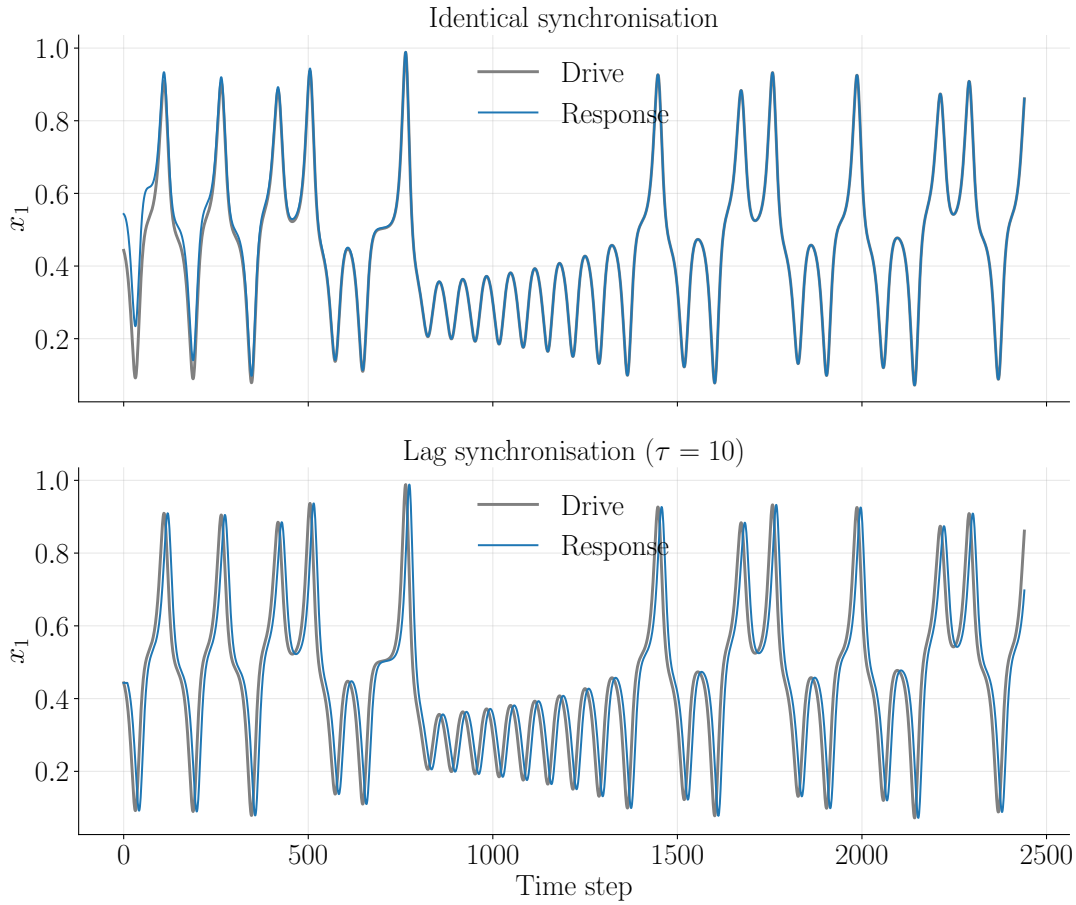


Figure 2.6: Identical and lag synchronisation visualisation for the Lorenz–63 state variable ($\boldsymbol{x}_1$).

The relationship between the states of two coupled systems may be described by a nonlinear mapping, referred to as generalised synchronisation (Kocarev and Parlitz 1996; B. R. Hunt, Ott, and Yorke 1997; Tang et al. 1998; S. Yang and Duan 1998). Generalised synchronisation has been studied in the context of classical reservoir computing to characterise the learnability of reservoir computers when forecasting chaotic dynamics (Lymburn et al. 2019; Tang et al. 1998; J. D. Hart 2024; Platt, Wong, et al. 2021). In this section, we give an overview of generalised synchronisation in reservoir computing and use these ideas to inform the design of quantum reservoir computers in Chapter 3.

Upon explicit time discretization, the continuous-in-time dynamical system in Eq. 2.1 becomes a discrete map

$$\boldsymbol{x}(t_{i+1}) = \boldsymbol{F}(\boldsymbol{x}(t_i)), \quad \boldsymbol{x}_0 = \boldsymbol{x}(0). \tag{2.18}$$

The training time series, $\boldsymbol{u}$, may, in general, be $\boldsymbol{u}(t_i) = h(\boldsymbol{x}(t_i))$. Without loss of generality, we set h to the identity operator, so $\boldsymbol{u}(t_i) = \boldsymbol{x}(t_i)$. (This means that we focus on fully observable dynamical systems.)

We consider a one-way coupled chaotic system, in which the driving state, $\boldsymbol{u}(t_i)$, one-way forces a response system described by the vector $\boldsymbol{r}(t_i) \in \mathbb{R}^N$ with $N_r \gg D$. Fig. 2.7 shows a schematic of a one-way coupled drive-response system with relevance to reservoir computers. The dynamics of the one-way coupled system are governed by (O. Ahmed, Tennie, and Magri 2025b)

$$\boldsymbol{u}(t_{i+1}) = \boldsymbol{F}\big(\boldsymbol{u}(t_i)\big), \qquad \boldsymbol{u}_0 = \boldsymbol{u}(0). \tag{2.19}$$

$$\boldsymbol{r}(t_{i+1}) = \boldsymbol{G}\big(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)\big), \qquad \boldsymbol{r}_0 = \boldsymbol{r}(0). \tag{2.20}$$

where $\boldsymbol{G}$ is a user-defined map, whose properties directly affect GS. The response state, $\boldsymbol{r}$, is in generalised synchronisation (GS) with the driving state, $\boldsymbol{u}$, when $\boldsymbol{G}$ generates a map $\phi : \boldsymbol{x} \to \boldsymbol{r}$ that guarantees asymptotic stability (e.g., Abarbanel, Rulkov, and Sushchik 1996; B. R. Hunt, Ott, and Yorke 1997)

$$\lim_{i\to\infty} \|\boldsymbol{r}(t_i) - \phi(\boldsymbol{x}(t))\| = 0, \quad \text{for every } \boldsymbol{x}(t) \in \mathcal{M}, \tag{2.21}$$

where $\mathcal{M}$ is the attractor of the driving system (also referred to as synchronisation (Verzelli, Alippi, and Livi 2021b; Platt, Penny, et al. 2022) or physical manifold). In other words, GS occurs when there exists a continuous ϕ that fulfils Eq.2.21. Equivalently, GS occurs when

- Physically, the response system converges to the attractor, $\mathcal{M}$;
- Mathematically, $\boldsymbol{G}$ is a contractive map, i.e., its largest conditional Lyapunov exponent is negative (Table 2.1); and
- From a functional analysis point of view, $\boldsymbol{r}(t_i) = \phi(\boldsymbol{x}(t_i))$ is a Cauchy sequence that tends to $\phi(\boldsymbol{x}(t))$ on the attractor $\mathcal{M}$.

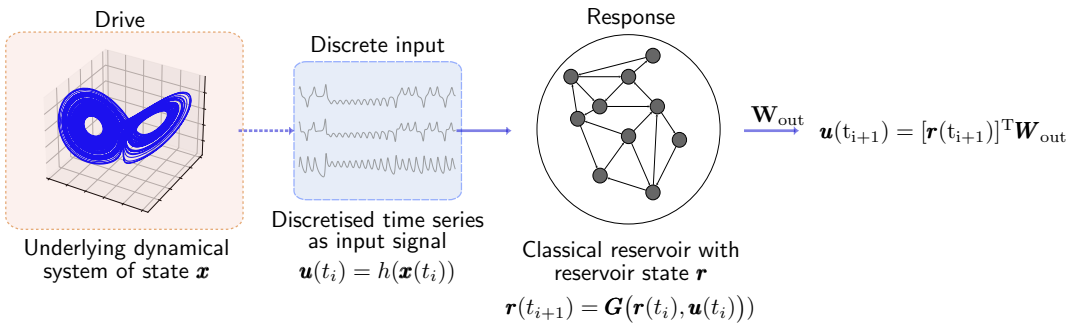


Figure 2.7: Reservoir computers as drive-response systems. The chaotic time series (data), $\boldsymbol{x}$, is available at discrete intervals over $[t, t+1, ..]$ as $\boldsymbol{u}(t_i) = h(\boldsymbol{x}(t_i))$, with h being the observable map. The driving input forces the reservoir system $\boldsymbol{r}$ (response system). The reservoir state evolves as $\boldsymbol{r}(t_{i+1}) = \boldsymbol{G}\big(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)\big)$. The linear map $\mathbf{W}_{out}$ is the only trainable part of the reservoir computers.

There might exist infinite maps, ϕ, that guarantee GS; however, only a continuously differentiable ϕ guarantees that the attractor $\mathcal{M}$ can be correctly reconstructed, thus, its invariant properties can be correctly inferred by the response system (differentiable generalised synchronisation, B. R. Hunt, Ott, and Yorke 1997).

2.4.1 Conditional Lyapunov exponents

The Conditional Lyapunov exponents (CLEs) of $\boldsymbol{G}$ (Eq. 2.20) and the Lyapunov exponents of $\boldsymbol{F}$ (Eq. 2.19) determine the differentiability of ϕ (Pecora and Carroll 1991b; Pyragas 1997; B. R. Hunt, Ott, and Yorke 1997). The CLEs are computed from the Jacobian of the response system conditioned on the driving signal $\boldsymbol{u}$, which defines the local stability of the response dynamics, as discussed in Chap. 3. As proved by (B. R. Hunt, Ott, and Yorke 1997), ϕ is continuously differentiable if condition (iii) in Tab. 2.1 is met. We use this condition to design robust reservoir computers.

A Lyapunov exponent of a discrete map, measured per iteration, is related to the corresponding continuous-time Lyapunov exponent by

$$\lambda_{\text{continuous}} = \frac{\lambda_{\text{discrete}}}{\Delta t}, \tag{2.22}$$

where Δt is the time step. Throughout this thesis, Lyapunov exponents are reported in continuous-time units.

Table 2.1: Criteria for generalised synchronisation and differentiability. The largest conditional Lyapunov exponent of the response system is denoted $\max(\lambda_{\text{CLE}})$, and the most negative Lyapunov exponent of the driving system is denoted λ^*. All Lyapunov exponents are expressed in continuous-time units.

	Condition	Property
i.	$\max(\lambda_{\text{CLE}}) \geq 0$	GS does not occur.
ii.	$\lambda^* < \max(\lambda_{\text{CLE}}) < 0$	GS occurs; ϕ is not differentiable.
iii.	$\max(\lambda_{\text{CLE}}) < \lambda^* < 0$	GS occurs; ϕ is continuously differentiable. Differentiable GS (DGS).

2.5 Reservoir computing as a dynamical system and GS

Reservoir computing can be interpreted as a driven dynamical system, where the input acts as a forcing signal (Lu, B. R. Hunt, and Ott 2018b; Platt, Wong, et al. 2021; Platt, Penny, et al. 2022). The reservoir computer ansatz 2.14 defines implicitly the local inverse of the map ϕ in GS theory (Sec. 2.4), i.e., $\phi^{-1} \equiv \boldsymbol{W}_{out}$. The trained readout $\boldsymbol{W}_{out}$ can be interpreted as an approximation of the inverse mapping ϕ^{-1} in GS theory (Sec. 2.4). This establishes a link between reservoir computing and GS, where successful learning corresponds to the existence of a functional relationship between the driving system and the reservoir states (J. D. Hart 2024; Grigoryeva, A. Hart, and Ortega 2021).

2.5.1 Echo state property

For classical reservoir computers to perform well, it is necessary that the echo state property (ESP) is fulfilled (Lukosevicius 2012; Jaeger et al. 2007; Jaeger 2001; Boedecker et al. 2012; Storm, Gustavsson,

and Mehlig 2022). The ESP is defined as the property of a reservoir computer to "forget" the effect of initial conditions (Lu, B. R. Hunt, and Ott 2018b; Platt, Wong, et al. 2021; Platt, Penny, et al. 2022; J. D. Hart 2024; Farkaš, Bosák, and Gergeľ 2016; Racca, Doan, and Magri 2023). From a GS point of view, we argue that the echo state property is equivalent to the condition of GS in Eq. 2.21.

2.6 Learnability conditions

Using tools from GS and dynamical systems theory, the performance of reservoir computers can be assessed with the following principles:

- During training, synchronisation between the drive and response systems is assessed using conditional Lyapunov exponents (CLEs).
- After training, the ability of the reservoir to reproduce invariant properties of the system is evaluated through its autonomous dynamics.

These design principles inform about the learnability conditions for a reservoir computer trained for forecasting chaotic dynamics.

In this chapter, we have introduced the learnability and forecasting problem for chaotic systems and presented reservoir computers as a framework for such tasks. We have then outlined the theory of generalised synchronisation and interpreted reservoir computers as GS systems. By defining metrics and conditions for the learning, we use these results to inform the design of stable and robust quantum reservoirs in Chap. 3.

Chapter
3

From synchronisation theory to quantum reservoir design

In this chapter, we bridge the theory of generalised synchronisation presented in Chap. 2 with quantum reservoir computers and present GS as a tool to analyse learning of QRCs. First, we give an overview of quantum computing, specifically, qubits and quantum states and derive leaky-integrated reservoir computing in Sec. 3.2. Second, in Sec. 3.3, we overview the existing hybrid quantum-classical reservoir computers and draw a comparison with classical reservoir computers in Sec. 3.4. In Sec. 3.5, we motivate the choice of QRC feature maps and present recurrence-free quantum reservoir computing (RF-QRC) as an efficient forecasting framework as proposed in O. Ahmed, Tennie, and Magri 2024b; O. Ahmed, Tennie, and Magri 2024a. In Secs. 3.6 and 3.7, we cast QRC as a GS problem, propose design principles for robust design and analytically derive the Jacobians of quantum reservoir updates and conditions for successful prediction tasks (O. Ahmed, Tennie, and Magri 2025b).

3.1 Qubits and quantum states

In quantum computers, the internal state of the machine can be represented by a complex vector in a Hilbert space. The time evolution during computation is described by a matrix multiplication of this vector with a unitary operator.

In most quantum hardware architectures, the elementary building blocks, known as qubits, are quantum degrees of freedom described by normalized complex-valued vectors in a two-dimensional Hilbert space, where the computational basis states $|0\rangle$ and $|1\rangle$ are associated with the Cartesian basis vectors, and the complex amplitudes α and ω satisfy the normalization condition $|\alpha|^2 + |\omega|^2 = 1$.

$$|\psi\rangle = \alpha|0\rangle + \omega|1\rangle \hat{=} \begin{pmatrix} \alpha \\ \omega \end{pmatrix} \in \mathbb{C}^2. \tag{3.1}$$

States of n qubits occupy the tensor-product Hilbert space of the individual two-dimensional Hilbert

spaces

$$|\psi_n\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2^n} \end{pmatrix} \in (\mathbb{C}^2)^{\otimes n}, \tag{3.2}$$

where $\otimes$ denotes the tensor product, such that $(\mathbb{C}^2)^{\otimes n}$ represents the 2^n-dimensional Hilbert space of an n-qubit quantum system. The computational basis states of this space are tensor products of the single-qubit basis states, e.g. $|00\cdots0\rangle, |00\cdots1\rangle, \ldots, |11\cdots1\rangle$.

The dimension of the state vector, therefore, scales exponentially with the number of qubits n. Unentangled pure states, known as product states, can be written as tensor products of individual qubit states

$$|\phi_n\rangle = (\alpha_0|0_0\rangle + \omega_0|1_0\rangle) \otimes (\alpha_1|0_1\rangle + \omega_1|1_1\rangle) \otimes \cdots \otimes (\alpha_{n-1}|0_{n-1}\rangle + \omega_{n-1}|1_{n-1}\rangle). \tag{3.3}$$

Most quantum states in Eq. 3.2 are entangled and therefore cannot be decomposed into such a tensor-product form of Eq. 3.3. Quantum evolution is described by unitary operators acting on the Hilbert space, such that the final quantum state after a computation is given by

$$|\psi_f\rangle = U|\psi_i\rangle. \tag{3.4}$$

Quantum hardware architectures provide elementary sets of unitary operators, known as quantum gates, which can be combined to approximate arbitrary unitary transformations. The final output of the computation is obtained through measurements of the quantum state, yielding probabilistic estimates of the modulus-squared amplitudes $(|a_1|^2, |a_2|^2, \ldots)$. A detailed introduction to the mathematical foundations of quantum computing, including tensor-product Hilbert spaces (Chap. 2), quantum circuits and universal gates (Chap. 4), and quantum algorithms such as the quantum Fourier transform and phase estimation (Chap. 5), can be found in (Nielsen and Chuang 2011).

3.2 Leaky-integrated classical reservoir computing

We derive the leaky-integrated reservoir computing equations starting from dynamical systems theory with leakage. The continuous-time dynamics of a dynamical system (from Eq. 2.1) with leaky integration can be written as

$$\tau \frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x}(t) + f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{3.5}$$

where $\boldsymbol{x}(t)$ is the state of the dynamical system, τ is a time constant determining the rate of leakage or decay, and f is a nonlinear function describing the evolution of the state $\boldsymbol{x}(t)$ influenced by an input signal $\boldsymbol{u}(t)$. For echo state networks, the function f depends on random input and reservoir weight matrices (Jaeger 2001)

$$\tau \frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x}(t) + \tanh\left(\boldsymbol{W}_{in}\boldsymbol{u}_{in}(t) + \boldsymbol{W}\boldsymbol{x}(t)\right). \tag{3.6}$$

Using Euler discretisation with stepsize Δt, we obtain a discrete-time reservoir state update equation. In practical implementations, the input signal is also sampled at discrete time intervals,

$$\boldsymbol{x}(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right)\boldsymbol{x}(t) + \frac{\Delta t}{\tau}\tanh(\boldsymbol{W}_{in}\boldsymbol{u}_{in}(t) + \boldsymbol{W}\boldsymbol{x}(t))\,. \tag{3.7}$$

After discretisation, time is re-indexed in integer steps by choosing the discrete timestep as the unit timescale, $\Delta t = 1$. The reservoir update equation, equivalent to the leaky-integrator form of the reservoir state update equations shown previously in Eqs. 2.10–2.11, then becomes

$$\boldsymbol{x}(t_{i+1}) = (1-\vartheta)\boldsymbol{x}(t_i) + \vartheta\tanh(\boldsymbol{W}_{in}\boldsymbol{u}_{in}(t_i) + \boldsymbol{W}\boldsymbol{x}(t_i))\,, \tag{3.8}$$

where $\vartheta = \frac{\Delta t}{\tau} \leq 1$ is a hyperparameter governing how much information from previous reservoir states is retained. This type of model is closely related to leaky-integrator neuron models, although it does not include the threshold-and-reset mechanism of spiking leaky integrate-and-fire (LIF) neurons, which have several applications in neuroscience (Teeter et al. 2018; Gabbiani and Cox 2010) and signal processing (Lansky and Ditlevsen 2008). The concept of leakage in reservoir computing was introduced in liquid state machines (Maass, Natschläger, and Markram 2002) and later incorporated into echo state networks (Jaeger et al. 2007; Lukosevicius 2012). Physically, leaky-integrator neurons possess intrinsic state dynamics that make them suitable for temporal learning tasks. Leaky integration can also be interpreted as a low-pass filter or exponential smoothing operation (Lukosevicius 2012), making these models appropriate for learning from noisy data, as explained in Chap. 5 and Chap. 6.

3.3 Hybrid quantum-classical reservoir computing

Hybrid quantum-classical reservoir computing (Pfeffer, Heyder, and Schumacher 2022), or QRC (termed in this work), is based on the principles of classical reservoir computing (Chap. 2). In QRC, the exponential size of n-qubit Hilbert spaces is utilised to encode and process classical reservoir state vectors, $\boldsymbol{r}(t_i)$. Thus, the classical reservoir is replaced by a quantum counterpart.

More specifically, within a single loop of the training phase, the reservoir state $\boldsymbol{r}(t_i)$ is mapped onto an n-qubit unitary operator $P(\boldsymbol{r}(t_i))$, which transforms the initial state $|0\rangle^{\otimes n}$ into the quantum state $|\psi(t_{i+1})\rangle = P(\boldsymbol{r}(t_i))|0\rangle^{\otimes n}$. The explicit form of P is determined by the quantum circuit architecture characterising the specific QRC framework. These circuits are generally parametrised by single-qubit rotation gates together with entangling CNOT gates (Appendix A.2), which together form a universal gate set for quantum computation.

For a training dataset $\{\hat{\boldsymbol{u}}(t_0), \hat{\boldsymbol{u}}(t_1), \ldots, \hat{\boldsymbol{u}}(t_{N_{tr}})\}$, the input time series is encoded into the quantum circuit using single-qubit rotation angles. With this encoding, a second unitary operator $\Xi(\boldsymbol{u}_{in}(t_i))$ is applied to the quantum state $|\psi(t_i)\rangle$, followed by a third unitary operator $V(\boldsymbol{\alpha})$ defined by a random parametrised quantum circuit. The operator $V(\boldsymbol{\alpha})$ introduces randomisation and entanglement mixing through parametrised single-qubit rotations and entangling gates. The combined action for each time step is in Fig. 3.1

$$|\psi(t_{i+1})\rangle = V(\boldsymbol{\alpha})\,\Xi(\boldsymbol{u}_{in}(t_i))\,P(\boldsymbol{r}(t_i))|0\rangle^{\otimes n}. \tag{3.9}$$

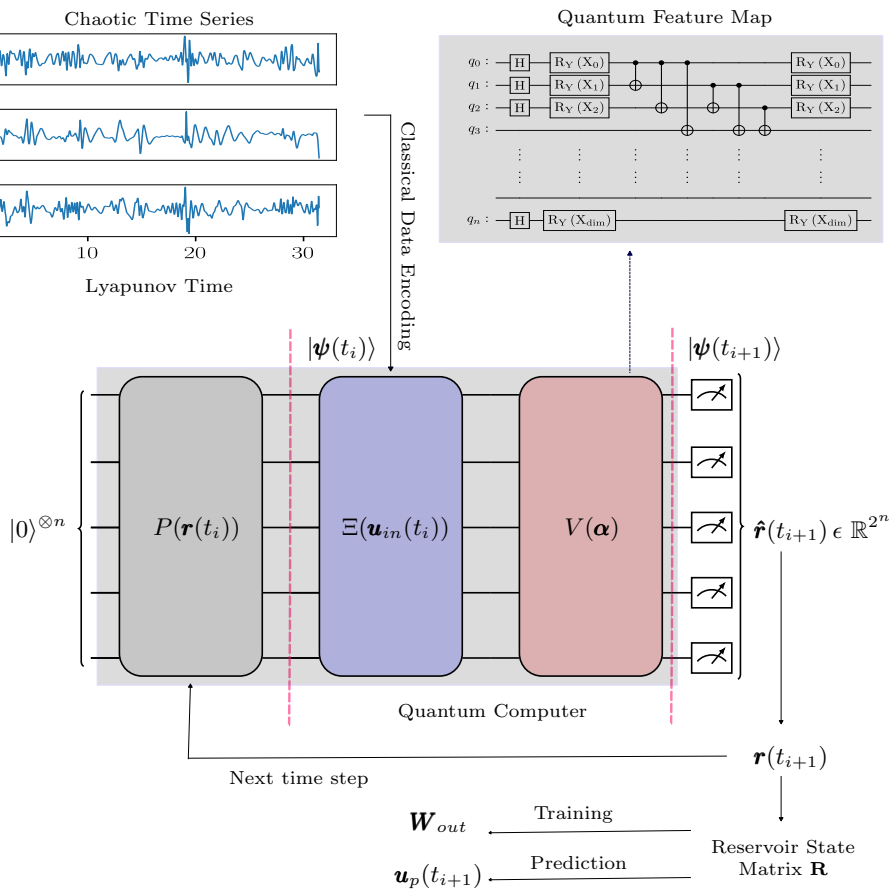


Figure 3.1: Schematic representation of a gate-based quantum reservoir computer. The first two unitaries are used to encode previous reservoir states $\boldsymbol{r}(t_i)$ and input time-series $\boldsymbol{u}_{in}(t_i)$, the third additional unitary provides randomisation. Each unitary consists of a specific feature map. The quantum circuit execution starts from the ground state $|\mathbf{0}\rangle$ and measures the reservoir state activation for the next time-step $\hat{\boldsymbol{r}}(t_{i+1})$, which is post-processed classically before feeding the next time-step. Each reservoir activation state $\boldsymbol{r}(t_{i+1})$ is saved classically to use for training and predictions. In RF-QRC, we remove the feedback loop and the first unitary $\boldsymbol{P}$.

Similar to classical reservoir computing, we derive a pre-processed reservoir state vector $\hat{\boldsymbol{r}}(t_{i+1})$ from measured probabilities of $|\psi(t_{i+1})\rangle$ in the computational basis. This requires sampling and measuring the state $|\psi(t_{i+1})\rangle$ multiple times for each time step. We compare the elementary steps of classical and quantum reservoir computing algorithms in Fig. 3.2.

Following the post-processing of $\hat{\boldsymbol{r}}(t_{i+1})$ to $\boldsymbol{r}(t_{i+1})$ (Eq. 2.2), the reservoir state matrix $\boldsymbol{R}$ is formed by concatenating the reservoir state vectors. As in classical reservoir computing, the initial reservoir state is chosen randomly, which in turn, necessitates the washout of a few initial time-stepped reservoir vectors.

The open-loop and closed-loop configurations of the hybrid quantum-classical architecture are presented in Fig. 3.3. Each reservoir update $\boldsymbol{r}(t_i)$ in both the open-loop and closed-loop requires the input of the previous reservoir state $\boldsymbol{r}(t_{i-1})$, similar to the classical reservoir update (dotted feedback loop in Fig. 3.3). In Sec. 3.5, we propose a quantum-classical architecture, which is independent of this feedback loop, thereby allowing for parallelisation of the training phase of QRC.

Algorithm 1: Classical Reservoir Computer (CRC)

1: Pre-process the input data time-series $\boldsymbol{u}_{in}(t_i)$.

2: Random and sparse input weight matrix $\boldsymbol{W}_{in}$ maps the input to high dimensional reservoir.
▷ $\boldsymbol{W}_{in}\,\boldsymbol{u}_{in}(t_i)$

3: Random reservoir weight matrix $\boldsymbol{W}$ with specified connectivity provides connections and the flow of information between different neurons in the reservoir.

4: Run reservoir update to calculate reservoir activation states $\boldsymbol{r}(t_{i+1})$ from the input $\boldsymbol{u}_{in}(t_i)$.
▷ $\tanh(\boldsymbol{W}_{in}\,\boldsymbol{u}_{in}(t_i) + \boldsymbol{W}\,\boldsymbol{r}(t_i))$

5: Compute linear readout weight $\boldsymbol{W}_{out}$ by minimizing the MSE between predicted and true output, using linear ridge regression.

6: Make predictions using the trained $\boldsymbol{W}_{out}$ matrix and the reservoir activation states $\boldsymbol{r}(t_{i+1})$. Use open-loop configuration for one-step-ahead predictions and the closed-loop configuration for the autonomous predictions from the input data.

Algorithm 2: Quantum Reservoir Computer (QRC)

1: Pre-process the input data time-series $\boldsymbol{u}_{in}(t_i)$ and encode it into the quantum circuit using rotational encoding.

2: Use the specified quantum feature maps to map the input data to the high-dimensional Hilbert space. Here, U represents the unitary evolution of the quantum state.
▷ $U(\boldsymbol{u}_{in}(t_i))$

3: Apply an additional unitary $V(\alpha)$ sampling from a uniform random distribution to provide randomization in the quantum circuit.

4: Quantum reservoir state update $|\boldsymbol{\psi}(t_{i+1})\rangle$ is achieved by running the quantum circuit with the encoded input data $\boldsymbol{u}_{in}(t_i)$, previous reservoir states $|\boldsymbol{\psi}(t_i)\rangle$ and the additional random unitary.
▷ $V(\boldsymbol{\alpha})\,\Xi(\boldsymbol{u}_{in}(t_i))\,P(\boldsymbol{r}(t_i))|\mathbf{0}\rangle$.

5: Measure the quantum circuit in a Pauli-Z basis to convert quantum reservoir state $|\boldsymbol{\psi}(t_{i+1})\rangle$ to classical reservoir state $\boldsymbol{r}(t_{i+1})$. Re-encode the measured $\boldsymbol{r}(t_{i+1})$ to P unitary for the next time-step.

6: Repeat steps 5 and 6 similarly to a classical reservoir computer, to train the network and make predictions by repeating the quantum reservoir state updates.

Figure 3.2: Classical and quantum reservoir computing algorithms.

Although QRC exploits the exponential size of the state space of the quantum register, there is a practical limitation due to the limited connectivity of the employed quantum processors (e.g., neighbour-neighbour interactions only on 2D superconducting qubit processors Tennie, Laizet, et al. 2025). The performance of QRC generally depends on the degree of entanglement present in the circuit (Götting, Lohof, and Gies 2023). Therefore, we investigate the performance of structurally different architectures.

3.4 Quantum reservoir computing: different designs

In this section, we propose ansätze for the circuit architecture in QRC. To do so, a specific quantum feature map must be chosen, and we therefore start by briefly reviewing a number of common choices. Following that, we present the Recurrence-free quantum reservoir computing (RF-QRC) architecture in Sec. 3.5.

We consider four different feature maps for QRC. These are the (a) Linearly entangling feature map, used in Pfeffer, Heyder, and Schumacher 2022 for QRC applications, (b) Product state feature map, which was proposed in Havlíček et al. 2019, and has been shown to offer a high expressivity (Abbas et al. 2021), (c) Fully entangling feature map, in which CNOT gates entangle pairs of qubits, (d) Symmetric fully entangling feature map, which only differs from (c) by an additional data encoding layer following the CNOT gates. Schematic circuit diagrams of all four feature maps are shown in Appendix A.2. When the number of encoded parameters Θ associated with an input vector exceeds the number of available

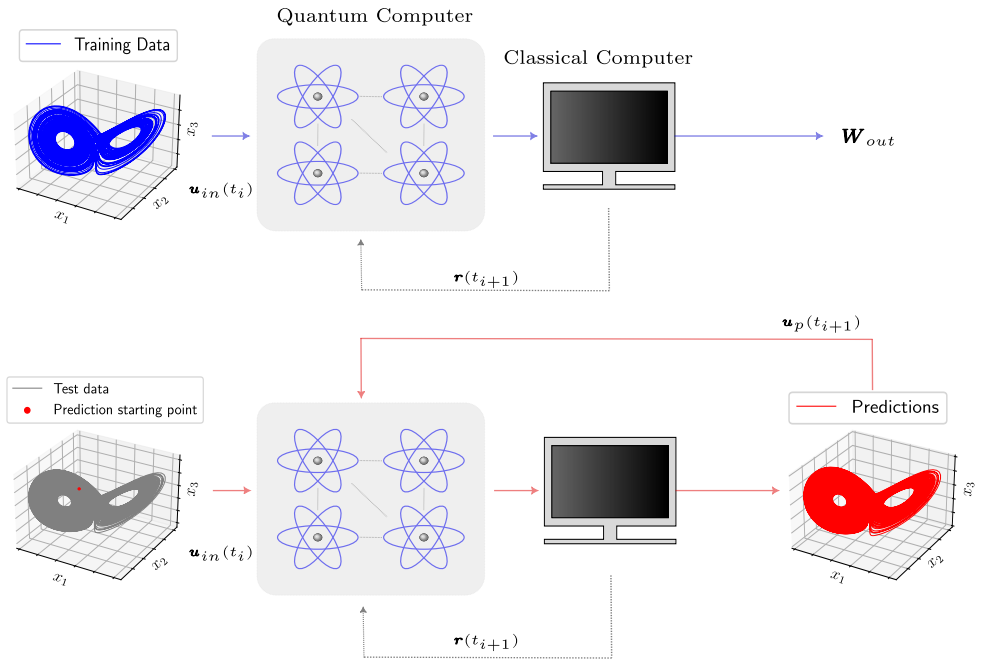


Figure 3.3: Quantum-classical reservoir architecture schematic representation (a) open-loop training to produce $\boldsymbol{W}_{out}$ matrix (b) closed-loop autonomous predictions starting from an arbitrary point in the unseen dataset. The dotted line in both figures indicates the recurrence involved in conventional reservoir architectures. In the RF-QRC architecture, we remove this recurrent feedback layer of reservoir states.

Table 3.1: Quantum reservoir computer: different designs.

	Reservoir states (P)	Input (Ξ)	Variation (V)
QRC-C1	Linearly entangled	Fully entangled	Fully entangled (symmetric)
QRC-C2	Linearly entangled	Linearly entangled	Linearly entangled
QRC-C3	–	Linearly entangled ($\times 2$)	Linearly entangled
QRC-C4 (RF-QRC)	–	Fully entangled ($\times 2$)	Fully entangled (symmetric)
QRC-C5	–	Product states ($\times 2$)	Linearly entangled

qubits, the feature-map encoding must be applied multiple times. For certain QRC architectures, such as reservoir-state encodings where $\Theta \sim 2^n$, this repeated encoding significantly increases the circuit depth of the unitary operator P. This motivates the development of more computationally efficient QRC architectures, as summarised in Tab. 3.1.

Comparison of circuit depths corresponding to the five architectures is shown in Fig. 3.4. Removing the recurrence, i.e. by setting P to the identity ($P(\boldsymbol{r}(t_i)) = \boldsymbol{I}$), and by applying Ξ twice for stimulating the reservoir dynamics, yields circuit depths that are independent of the reservoir size. The suitability of this choice is justified *a posteriori* by an improved performance of QRC in a range of prototypical model systems in Chap. 4.

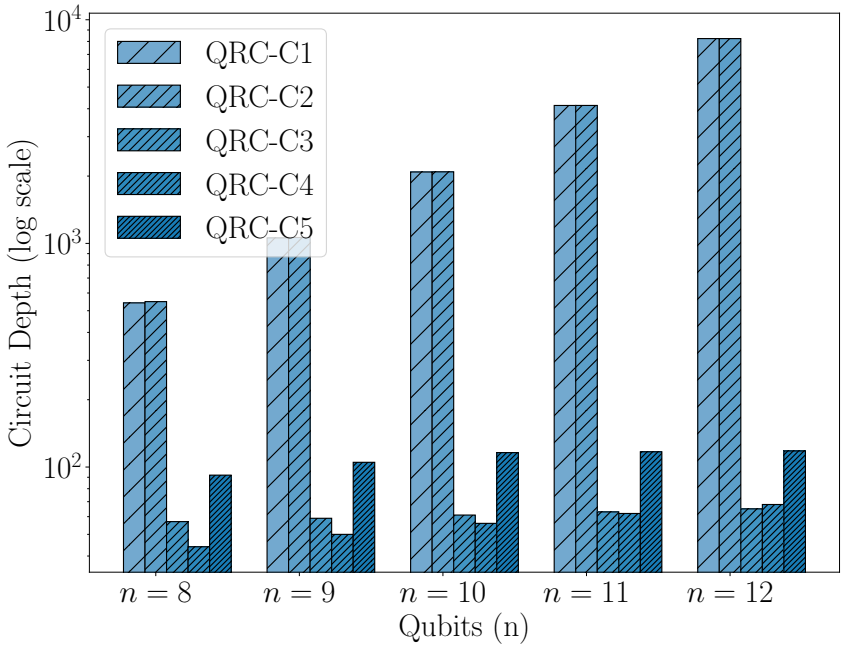


Figure 3.4: Circuit depth of different ansätze for the ten-dimensional Lorenz-96 input.

3.5 Recurrence-free quantum reservoir computing (RF-QRC)

In gate-based quantum reservoir computing, the quantum state vector is propagated by a $\boldsymbol{\theta}$-parametrised quantum circuit with unitary $\mathcal{U}(\boldsymbol{\theta})$ at each time step.

$$|\psi(t_{i+1})\rangle = \mathcal{U}(\boldsymbol{\theta})|\psi(t_i)\rangle. \tag{3.10}$$

Previous proposals of gate-based QRC involve a reservoir map that depends on input data $\boldsymbol{u}_{in}$, previous reservoir states $\boldsymbol{r}$ and a random unitary parameterised by $\boldsymbol{\alpha}$ (Eq. 3.9). By contrast, we propose a recurrence-free quantum reservoir, in which the state updating equation only evolves as a function of the input time series and a random $\boldsymbol{\alpha}$-parametrised unitary.

$$|\psi(t_{i+1})\rangle = V(\boldsymbol{\alpha})\,\Xi(\boldsymbol{u}_{in}(t_i))|0\rangle^{\otimes n}. \tag{3.11}$$

After each time stepping, a measurement in the computational basis $\{|k\rangle\}_{k=0}^{k=2^n}$ is performed, and the result is used to form a new reservoir state vector $\boldsymbol{r}(t_{i+1})$

$$r^{(k)}(t_{i+1}) = (1-\epsilon)\,r^{(k)}(t_i) + \epsilon\,|\langle\psi(t_{i+1})|k\rangle|^2 \tag{3.12}$$

where $r^{(k)}$ is the component of the reservoir state on the basis vector k, and $0 \leq \epsilon \leq 1$ is the leak rate, which is a hyperparameter.

On the one hand, QRCs (Mujal, Martínez-Peña, Nokkala, et al. 2021; Pfeffer, Heyder, and Schumacher 2022) contain recurrences in the unitary $P(\boldsymbol{r}(t_i))$ in Eq. 3.9. These recurrences keep memory of the past,

which is key to time series forecasting. On the other hand, RF-QRCs set $P(\boldsymbol{r}(t_i)) = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity. The memory of the past is kept by leaky integration in Eq. 3.12. The leaky hyperparameter, ϵ, determines the memory or leakage as discussed in Sec. 3.2 ($0 < \epsilon \ll 1$ preserves long memory, whereas the limit $\epsilon = 1$ is that of Extreme Learning Machines (Xiong et al. 2023), which have no memory). This allows RF-QRCs to capture temporal dependencies and makes them suitable for forecasting tasks.

3.6 Quantum reservoir computing and generalised synchronisation

We make a connection between quantum reservoir computing and generalised synchronisation (GS). In a quantum reservoir computing approach, a quantum system is the response system (reservoir)[a], and the driving states are the observation of a physical system, encoded in data (Fig. 3.5). The quantum ansätze with entangled qubits offer rich expressivity (Fujii and Nakajima 2017; Fujii and Nakajima 2021) to infer the mapping ϕ in the GS asymptotic relation Eq. 2.21.

Each unitary evolution in Eq. 3.11 is made of classical input-dependent single-qubit rotations R_y and two-qubit entanglement CNOT gates to encode and process the classical data for the quantum reservoir. We can rewrite Eq. 3.12 in vector notation for a recurrent QRC

$$\begin{aligned} \boldsymbol{r}(t_{i+1}) &= (1-\epsilon)\,\boldsymbol{r}(t_i) + \ldots \qquad (3.13)\\ &\ldots\epsilon\,\langle 0|^{\otimes n}\; P^{\dagger}(\boldsymbol{r}(t_i))\,\Xi^{\dagger}(\boldsymbol{u}(t_i))V^{\dagger}(\boldsymbol{\alpha})\,|k\rangle\langle k|\,V(\boldsymbol{\alpha})\,\Xi(\boldsymbol{u}(t_i))\,P(\boldsymbol{r}(t_i))|0\rangle^{\otimes n},\\ &= (1-\epsilon)\,\boldsymbol{r}(t_i) + \epsilon\,\langle 0|^{\otimes n}\; P^{\dagger}(\boldsymbol{r}(t_i))\,\Xi^{\dagger}(\boldsymbol{u}(t_i))\,\boldsymbol{\mathcal{W}}\,\Xi(\boldsymbol{u}(t_i))\,P(\boldsymbol{r}(t_i))|0\rangle^{\otimes n}, \qquad (3.14)\end{aligned}$$

where $\mathcal{W}^{(k)} = V^{\dagger}|k\rangle\langle k|V$ form a 2^n-dimensional vector. Finally, the reservoir state is mapped back onto the physical domain as

$$\hat{\boldsymbol{u}}(t_{i+1}) = \boldsymbol{r}(t_{i+1})^{\mathrm{T}}\boldsymbol{W}_{out}, \qquad (3.15)$$

Similar to classical reservoir, quantum reservoir computers are also trained by minimising a quadratic error (Eq. 2.12). The training of quantum reservoir computers is a quadratic optimisation problem, whose global minimum is found by solving a linear system (ridge regression)

$$(\boldsymbol{R}\boldsymbol{R}^{T} + \beta\boldsymbol{I})\,\boldsymbol{W}_{out} = \boldsymbol{R}\,\boldsymbol{U}_d^{\mathrm{T}}, \qquad (3.16)$$

where $\boldsymbol{R} \equiv [\boldsymbol{r}(t_1), \boldsymbol{r}(t_2), \ldots \boldsymbol{r}(t_{N_{tr}})] \in \mathbb{R}^{N_r \times N_{tr}}$ is a matrix with concatenated reservoir states corresponding to each neuron for N_{tr} time steps of training, and $\boldsymbol{U}_d \equiv [\boldsymbol{u}(t_1), \boldsymbol{u}(t_2), \ldots \boldsymbol{u}(t_{N_{tr}})] \in \mathbb{R}^{N_u \times N_{tr}}$ is the matrix of concatenated input time series (data). After training (open-loop phase) and validation, the reservoir can be deployed as an autonomous dynamical system (closed-loop phase) to predict the evolution of the physical state in the future. A "good" quantum reservoir computer accurately forecasts the time series and infers the invariant properties of the attractor.

[a] From a hardware perspective, this quantum system can be an analog (Kornjača et al. 2024) or spin qubit system (Martínez-Peña, Nokkala, et al. 2023), a superconducting quantum circuit (Suzuki et al. 2022), transverse-field Ising model (Fujii and Nakajima 2017), or more generally a gate-based quantum circuit (Pfeffer, Heyder, and Schumacher 2022; O. Ahmed, Tennie, and Magri 2024b).

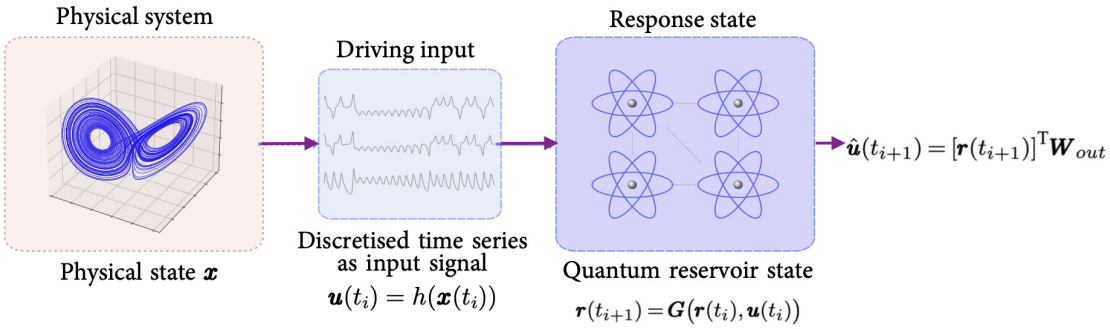


Figure 3.5: Quantum reservoir computers as drive-response systems. The chaotic time series (data) is available at discrete intervals over $[t, t+1, ..]$ as $\boldsymbol{u}(t_i) = h(\boldsymbol{x}(t_i))$, with h being the observable map. The driving input forces the quantum reservoir system $\boldsymbol{r}$ (response system). The reservoir state evolves as $\boldsymbol{r}(t_{i+1}) = \boldsymbol{G}\big(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)\big)$. The linear map $\mathbf{W}_{out}$ is the only trainable part of the quantum reservoir computer.

Table 3.2: Connection between GS and QRCs

	Generalized synchronization	Quantum reservoir computing
$\boldsymbol{x}$	Physical state (driving state)	Physical state
h	Observable map ($\boldsymbol{u} = h(\boldsymbol{x})$)	-
$\boldsymbol{u}$	Discrete driving state	Reservoirs input time series
$\boldsymbol{F}$	Evolution map of driving system	-
$\mathbf{G}$	Evolution map of response system	Evolution map of quantum reservoir computer
$\boldsymbol{r}$	Response state	Quantum reservoir state
ϕ	Defined implicitly by $\boldsymbol{G}$ and $\boldsymbol{F}$	$\phi^{-1} \equiv \boldsymbol{W}_{out}$
Learnability	Asymptotic stability	Echo state property

3.6.1 Echo state property in quantum reservoir computers

Similar to classical reservoir computing, for quantum reservoir computing to work, the reservoir feature map should be dissipative to satisfy the 'echo state' or fading memory property (Jaeger 2001) as presented in Sec. 2.5.1. This property ensures that the reservoir is determined by the driving signal irrespective of its initial state. We argue that GS and ESP are the same conditions under two different names, which originate from two different communities. We propose the criterion $GS = ESP$, which is a shorthand to mean that GS implies ESP, and vice versa, in quantum reservoir computers.

3.6.2 Contractive maps and GS in quantum reservoir computers

A summary of the connections between quantum reservoir computers and GS is shown in Tab. 3.2. A hyperparameter tuning should fulfil at least condition (ii) in Tab. 2.1 to guarantee the fulfilment of the quantum ESP.

The map $\boldsymbol{G}$ in Eq. 2.20 ensures generalised synchronisation when it is contractive

$$\|\boldsymbol{G}(\boldsymbol{r}_1, \boldsymbol{u}) - \boldsymbol{G}(\boldsymbol{r}_2, \boldsymbol{u})\| \leq \gamma \|\boldsymbol{r}_1 - \boldsymbol{r}_2\|, \tag{3.17}$$

which means that, for the same driving state $\boldsymbol{u}$, two initial conditions, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, converge to the same limit

with a rate $0 < \gamma < 1$. This implies that the reservoir state $\boldsymbol{r}$ asymptotically becomes independent of the initial state, thereby satisfying the ESP (Lu, B. R. Hunt, and Ott 2018a). In quantum reservoir computing, Eq. 3.17 can be rewritten in terms of quantum channels, i.e. completely positive and trace-preserving (CPTP) maps, T, acting on the density operator $\bar{\rho}$ (Nielsen and Chuang 2011):

$$\|T(\bar{\rho}_1, \boldsymbol{u}) - T(\bar{\rho}_2, \boldsymbol{u})\| \leq \tilde{\gamma} \|\bar{\rho}_1 - \bar{\rho}_2\| . \tag{3.18}$$

For the closed-system quantum reservoir update considered in Eq. 3.11, the map T is a unitary channel, which is a special case of a CPTP map. Unitary channels preserve distances between quantum states in trace norm, and therefore have contraction factor $\tilde{\gamma} = 1$. Hence, the unitary reservoir update is non-expansive but not contractive. As a result, in its purely unitary form, the reservoir state update in Eq. 3.11 does not satisfy the strict contraction condition required for $GS = ESP$.

We propose two methods to design a contractive quantum reservoir computer, which satisfy $GS = ESP$: (a) using leaky integration (Eq. 3.12) with $\epsilon < 1$ to ensure a contractive update; and (b) exploiting the dissipative nature of noise. Either condition (a) or (b) makes the quantum channel in quantum reservoir computers strictly contractive ($\tilde{\gamma} < 1$ in Eq. 3.18). This is good news: In contrast to classical reservoir computers, which may have expansive maps that make the machine divergent (Huhn and Magri 2022; Racca and Magri 2022b), we propose quantum reservoir computers that are contractive by design. (Further conditions on GS in quantum reservoir computers with injective generalised synchronisation were recently explored in (Martínez-Peña and Ortega 2025; Martínez-Peña and Ortega 2023).)

3.6.3 Jacobians

Key to computing the invariant properties of chaotic systems (section 2.2), such as the Lyapunov exponents, CLVs, and the KY dimension, is the Jacobian of the quantum reservoir computer update. The Jacobian of Eq. 2.11 is

$$\mathbf{J}(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)) = \frac{d\boldsymbol{r}(t_{i+1})}{d\boldsymbol{r}(t_i)} = \frac{\partial \boldsymbol{G}(\boldsymbol{u}(t_i), \boldsymbol{r}(t_i))}{\partial \boldsymbol{u}(t_i)} \frac{d\boldsymbol{u}(t_i))}{d\boldsymbol{r}(t_i)} + \frac{\partial \boldsymbol{G}(\boldsymbol{u}(t_i), \boldsymbol{r}(t_i))}{\partial \boldsymbol{r}(t_i)}. \tag{3.19}$$

By using the recurrent quantum reservoir computer ansatz, the Jacobian is

$$\begin{aligned}\mathbf{J}(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)) = (1-\epsilon)\boldsymbol{I} + \epsilon \frac{\partial}{\partial \boldsymbol{u}(t_i)} \langle 0|^{\otimes n} P^\dagger(\boldsymbol{r}(t_i)) \;\; \Xi^\dagger(\boldsymbol{u}(t_i)) \boldsymbol{\mathcal{W}} \Xi(\boldsymbol{u}(t_i)) \; P(\boldsymbol{r}(t_i)) |0\rangle^{\otimes n} \boldsymbol{W}_{out}^{\mathrm{T}} \\ + \epsilon \frac{\partial}{\partial \boldsymbol{r}(t_i)} \langle 0|^{\otimes n} \;\; P^\dagger(\boldsymbol{r}(t_i)) \, \boldsymbol{\mathcal{M}} \, P(\boldsymbol{r}(t_i)) \, |0\rangle^{\otimes n},\end{aligned} \tag{3.20}$$

where $\mathcal{M}^{(k)} = \Xi^\dagger \mathcal{W}^{(k)} \Xi$. This Jacobian can be calculated on quantum hardware with the parameter shift rule or linear addition of unitaries (Schuld, Bergholm, et al. 2019). In the RF-QRC (Sec. 3.5), $P(\boldsymbol{r}(t_i)) = \boldsymbol{I}$, which simplifies the Jacobian to

$$\mathbf{J}(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i)) = (1-\epsilon)\boldsymbol{I} + \epsilon \frac{\partial}{\partial \boldsymbol{u}(t_i)} \langle 0|^{\otimes n} \;\; \Xi^\dagger(\boldsymbol{u}(t_i)) \boldsymbol{\mathcal{W}} \Xi(\boldsymbol{u}(t_i)) \; |0\rangle^{\otimes n} \boldsymbol{W}_{out}^{\mathrm{T}}. \tag{3.21}$$

As explained in Sec. 2.4, establishing whether GS occurs (or not) also requires the computation of the conditional Lyapunov exponents (CLEs), which are defined as the Lyapunov exponents of the Jacobian

conditioned on the driving signal

$$\mathbf{J}(\boldsymbol{r}(t_i))_{|\mathrm{CLE}} \equiv \frac{\partial \boldsymbol{G}(\boldsymbol{r}(t_i), \boldsymbol{u}(t_i))}{\partial \boldsymbol{r}(t_i)}. \tag{3.22}$$

Using the recurrent quantum reservoir computer ansatz yields the Jacobian for the computation of the CLEs

$$\mathbf{J}(\boldsymbol{r}(t_i))_{|\mathrm{CLE}} = (1-\epsilon)\boldsymbol{I} + \epsilon \frac{\partial}{\partial \boldsymbol{r}(t_i)} \langle 0|^{\otimes n} \; P^{\dagger}(\boldsymbol{r}(t_i))\, \boldsymbol{\mathcal{M}}\, P(\boldsymbol{r}(t_i))\, |0\rangle^{\otimes n}, \tag{3.23}$$

which, in RF-QRCs, simplifies to

$$\mathbf{J}(\boldsymbol{r}(t_i))_{|\mathrm{CLE}} = (1-\epsilon)\boldsymbol{I}. \tag{3.24}$$

Equation 3.24 shows that the RF-QRC conditional Jacobian is a perfectly conditioned and isotropic matrix with N_r identical one-step contraction factors equal to $1-\epsilon$. The corresponding conditional Lyapunov exponents, expressed in continuous-time units, are

$$\lambda_{\mathrm{CLE}} = \frac{\ln|1-\epsilon|}{\Delta t}. \tag{3.25}$$

Therefore, $\lambda_{\mathrm{CLE}} < 0$ for $0 < \epsilon < 1$, and the RF-QRC satisfies the stability condition for GS by design (hence GS = ESP).

3.7 QRC design principles from generalised synchronisation

By interpreting quantum reservoir computing as a generalised synchronisation (GS) problem, we connect it to the learnability conditions for chaotic systems introduced in Sec. 2.6. This perspective allows us to translate these theoretical conditions into practical design principles for constructing robust quantum reservoir systems.

Drawing on classical reservoir computers, (Lu, B. R. Hunt, and Ott 2018b; Platt, Wong, et al. 2021; Platt, Penny, et al. 2022; J. D. Hart 2024; Grigoryeva, A. Hart, and Ortega 2021), we argue that the *learnability* of a dynamical system can be achieved with a quantum reservoir computer if $\boldsymbol{W}_{out}$ is the local inverse of a function that fulfills at least condition (ii) in Tab. 2.1, and, ideally, the stronger condition (iii). A good $\boldsymbol{W}_{out}$, which satisfies GS, results in an autonomous response system, which can be deployed to forecast the time series and infer the invariant properties of the driving system as shown in Chap. 4.

Using GS theory, the design of quantum reservoir computers can be guided by two key principles:

- **Stability during training:** The reservoir dynamics should synchronise with the input signal, which can be assessed using conditional Lyapunov exponents (CLEs). This ensures that the response system remains stable and admits a functional mapping from the input dynamics.
- **Consistency in autonomous evolution:** After training, the reservoir should reproduce the invariant properties of the underlying system through its autonomous dynamics. This ensures that the learned model captures not only short-term trajectories but also the long-term statistical and geometric

structure of the dynamics.

A central element in this framework is the Jacobian of the reservoir dynamics, which enables the computation of CLEs during training and invariant quantities during autonomous evolution. By analysing the Jacobian for both recurrent and recurrence-free quantum reservoir computers, we obtain explicit criteria for stability and learnability.

Finally, we relate these conditions to the echo state property (ESP), a standard requirement in reservoir computing. In this work, we adopt the criterion $GS = ESP$, meaning that generalised synchronisation of the reservoir with the input implies the fading memory property, and vice versa. This provides a unified interpretation of learnability and stability for quantum reservoir computers.

These results establish a direct link between GS theory and the practical design of quantum reservoir architectures. In Chap. 4, we construct and evaluate quantum reservoir computers for forecasting chaotic dynamics and extreme events, and assess their learnability by recovering the system's invariant properties.

Chapter
4

Forecasting chaotic dynamics and extreme events with quantum reservoir computers

In this chapter, we apply quantum reservoir computers for forecasting chaotic dynamics and extreme events on several prototypical chaotic dynamical systems. First, we give an overview of these prototypical systems, including Lorenz-63 and Lorenz-96 systems (E. N. Lorenz 1963) and the reduced-order model of chaotic shear flow that has extreme events (MFE) (Moehlis, Faisst, and Eckhardt 2004) in Sec. 4.1. Second, in Sec. 4.2, we compare the best-performing classical reservoir computers with different quantum reservoir computers, including the RF-QRC. This comparison is made against prediction tasks including (a) short-term time-accurate predictions, (b) long-term statistical predictions, and (c) extreme event prediction capabilities to identify the best-performing models (O. Ahmed, Tennie, and Magri 2024a). Finally, in Sec. 4.3, we use the best-performing quantum reservoir computer to infer the stability and invariant properties of the Lorenz-63 and Lorenz-96 systems, including the Lyapunov Spectrum, Covariant Lyapunov Vectors (CLVs) and attractor dimension via Kaplan-Yorke dimension (O. Ahmed, Tennie, and Magri 2025b), as discussed in Chap. 2.

4.1 Chaotic systems

We consider several prototypical dynamical systems exhibiting chaos and extreme events throughout this thesis, including the three-dimensional Lorenz-63 system, higher-dimensional Lorenz-96 systems ($D = 10, 20$), and the MFE system. We begin with low-dimensional systems and progressively increase the complexity to evaluate the robustness of the proposed quantum reservoir framework.

4.1.1 Three-dimensional Lorenz-63 system

The Lorenz-63 system (E. N. Lorenz 1963) is a reduced-order model of thermal convection flow governed by

$$\frac{dx_1}{dt} = \sigma_L(x_2 - x_1), \qquad \frac{dx_2}{dt} = x_1(\rho_L - x_3) - x_2, \qquad \frac{dx_3}{dt} = x_1x_2 - \beta_L x_3. \tag{4.1}$$

where σ_L , ρ_L , β_L : are system parameters. We take $[\sigma_L$, ρ_L , $\beta_L]$ = [10, 28, 8/3] to ensure chaotic behavior of the system (E. Lorenz 1995). The largest Lyapunov exponent describes the non-linear dynamics, which is $\Lambda = 0.9$ for the Lorenz-63 system and $1LT = 1/0.9$.

4.1.2 Higher-dimensional Lorenz-96 system

The Lorenz-96 system (E. Lorenz 1995) is a system of coupled ordinary differential equations that describes the large-scale behaviour of the mid-latitude atmosphere, and the transfer of a scalar atmospheric quantity, governed by

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})\, x_{i-1} - x_i + L, \qquad i = 1, \ldots, m \tag{4.2}$$

where L is the external forcing term, which we set to $L = 8$ to obtain chaotic behaviour. We impose periodic boundary conditions, such that $x_{-1} = x_{m-1}$, $x_0 = x_m$, and $x_{m+1} = x_1$. We study reduced-order Lorenz–96 models with ten and twenty dimensions, corresponding to $m = 10$ and $m = 20$, respectively.

4.1.3 Chaotic shear flow with extreme events

We consider a qualitative low-order model of turbulent shear flows, which describes a self-sustained turbulent process. This model is also known as the "Moehlis, Faisst, and Eckhardt (MFE)" model. The MFE model is nonlinear and captures relaminarisation and turbulent bursts (Moehlis, Faisst, and Eckhardt 2004). Owing to the nonlinear nature of this model, the MFE model has been employed to study turbulence transitions and chaos predictability (Doan, Polifke, and Magri 2021; Srinivasan et al. 2019). Mathematically, the MFE system can be described by the non-dimensional Navier-Stokes equations for forced incompressible flow

$$\frac{\partial \boldsymbol{v}}{\partial t} = -(\boldsymbol{v} \,.\, \boldsymbol{\nabla})\, \boldsymbol{v} \; - \boldsymbol{\nabla}\, p + \frac{1}{Re}\Delta \boldsymbol{v} + \boldsymbol{F}(y), \qquad \boldsymbol{\nabla} \cdot \boldsymbol{v} = 0 \tag{4.3}$$

where $\boldsymbol{v} = (u, v, w)$ is the three-dimensional velocity vector, p is the pressure, Re is the Reynolds number, ∇ is the gradient, and Δ is the Laplacian operator. $\boldsymbol{F}(y)$ on the right-hand side is the sinusoidal body forcing term and equals $\boldsymbol{F}(y) = \sqrt{2}\pi^2/(4Re)\sin(\pi y/2)\,\boldsymbol{e}_x$. The body force is applied along the x, y direction of the shear between the plates. We consider a three-dimensional domain of length L_x, L_y, L_z = $[4\pi, 2, 2\pi]$ and apply free slip boundary conditions at $y = L_y/2$, periodic boundary conditions at $x = [0; L_x]$ and $z = [0; L_z]$. The set of PDEs can be converted into ODEs by projecting the velocities onto Fourier modes as given by Eq. (4.4)

$$\boldsymbol{v}(\boldsymbol{x}, t) = \sum_{i=1}^{9} a_i(t)\, \hat{\boldsymbol{v}}_i\, (\boldsymbol{x}). \tag{4.4}$$

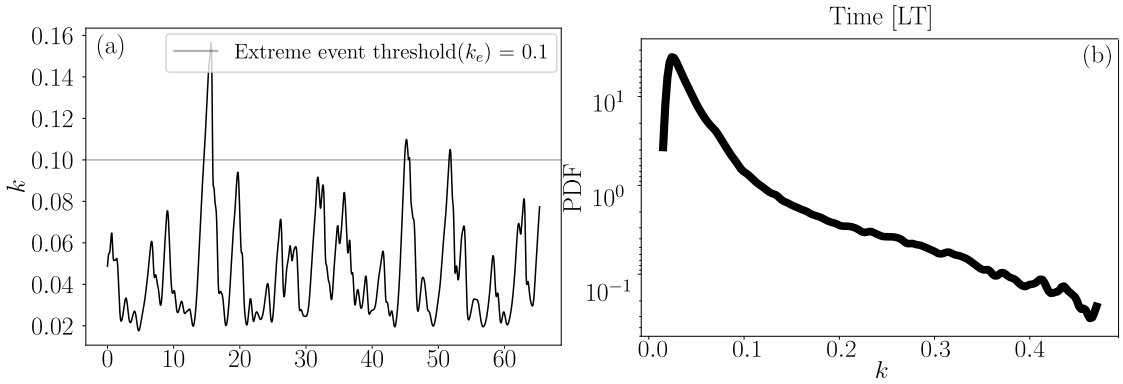


Figure 4.1: MFE time series kinetic energy and Probability Density Function (PDF) (a) single time series in which the Kinetic Energy exceeds the extreme event threshold k_e, highlighting the presence of an extreme event (b) probability distribution of kinetic energy calculated over an ensemble of time series

These nine decompositions for the amplitudes $a_i(t)$ are substituted into Eq. (4.3) to yield a set of nine ordinary differential equations as in (Moehlis, Faisst, and Eckhardt 2004). The MFE system displays a chaotic transient, which in the long term converges to a stable laminar solution. We want to predict the turbulent bursts of kinetic energy and chaotic transients, which are extreme events. Fig. 4.1 shows the evolution of kinetic energy ($k(t) = \frac{1}{2}\sum_{i=1}^{9} a_i^2(t)$) and the associated extreme event, and the long-term statistical distribution of the kinetic energy. An extreme event occurs when the kinetic energy $k(t)$ value exceeds the threshold $k(t) \geq k_e$. In this work, we take $k_e = 0.1$, as in (Racca and Magri 2022a)

4.2 Forecasting of chaotic dynamical systems

In this section, we analyse the dynamics of several chaotic systems, such as Lorenz-63, Lorenz-96, and MFE (Moehlis, Faisst, and Eckhardt), using classical and quantum reservoir computers.

4.2.1 Setup and metrics

To assess the prediction capabilities of various architectures presented in Sec. 3.3, we evaluate short-term time-accurate and long-term statistical predictions using performance measures discussed in Sec. 2.3.2. The sets of training, validation, and test data are generated by a 4^{th}-order Runge-Kutta numerical scheme.

The performance of classical reservoir computers critically depends on the set of chosen hyperparameters. We, therefore, use grid search and Bayesian optimisation to tune the hyperparameters (Snoek, Larochelle, and Adams 2012; Racca and Magri 2022a). We have used $scikit-learn$ library (Pedregosa et al. 2011) and recycle validation techniques (Racca and Magri 2021) for hyperparameter tuning of the classical reservoir. The hyperparameters and performance metrics for different chaotic systems are reported with each system. For the quantum reservoir, we use the grid-search method only to tune the hyperparameters.

To emulate the quantum circuits, we have used the *Qiskit* (Qiskit contributors 2023) and *Pennylane* (Bergholm et al. 2018) software packages to compute the noise-free evolution of the quantum register. In a physical implementation, at each time step, between 10^4 and 10^5 shots are required to reconstruct the reservoir activation signal. A number of shots between 10^4 to 10^6 is common in variational algorithms and other QML applications (Mujal, Martínez-Peña, Giorgi, et al. 2023b). The analysis of the effects of

Table 4.1: Parameters for three-dimensional Lorenz-63 system

Parameters	Symbol	Classical RC	Quantum RC
Time step	dt	0.01	0.01
Input scaling	σ_{in}	[0 , 1]	-
Spectral radius	$\rho(\boldsymbol{W})$	[0.1 , 1]	1
Tikhonov regularization	β	1×10^{-6},1×10^{-9},1×10^{-12}	1×10^{-6},1×10^{-9},1×10^{-12}
Leak rate	ϵ	[0.05, 1]	[0.05, 0.3]
Reservoir density	–	0.1,0.6,0.9	Configurations C1-C5

finite-sampling noise on the QRC's performance is the focus of the next Chap. 5.

4.2.2 Lorenz-63 system

As a first test case, we start with the analysis of the three-dimensional Lorenz-63 system (E. Lorenz 1995). The time series data set is derived by a Runge-Kutta method for $dt = 0.01$. The training time series comprises data points over a total time of 20 LT for both the classical and quantum reservoir networks. For the training, both networks are executed in an open loop to evolve the reservoir states and calculate the $\boldsymbol{W}_{out}$ matrix. For predictions, both classical and quantum networks evolve dynamically in the closed-loop configuration from ensembles of points in phase space, sampled on the attractor randomly, to quantify the time accuracy. These points correspond to the points in the unseen test data set.

In Tab. 4.1, we compare the hyperparameters of both quantum and classical reservoirs. Values inside square brackets indicate that the parameters are optimised within this range; multiple values indicate that we repeated the hyperparameter search for each of these values and then selected the best VPT out of them. The random rotation angles $\boldsymbol{\alpha}$ in the third unitary block can also be treated as a hyperparameter. Alternatively, one could derive the values of the random rotation angles from a uniform random distribution, which we do in this work. For each ensemble, we sample $V(\boldsymbol{\alpha}) \in \mathbb{R}^n$ from a uniform distribution interval $[0, 4\pi]$ with a predefined seed, which we keep fixed throughout the training and prediction of a particular realisation. This is similar to the classical input $\boldsymbol{W}_{in}$ and reservoir weight matrices $\boldsymbol{W}$, which are also pseudo-randomly generated and fixed for any specific realisation.

Time-accurate predictions

In Fig. 4.2, we present the VPT values derived with classical reservoir computing and compare them with different configurations of quantum reservoir computers from Tab. 3.1. The error bars at each value indicate the variation arising from different random seeds. We have tested five different seeds for each value. These seeds correspond to randomly generated weight matrices $\boldsymbol{W}$ and $\boldsymbol{W}_{in}$ for the classical reservoir and variation of $V(\boldsymbol{\alpha})$ values for quantum reservoir architectures. For each of these choices, after training, we initialise the closed-loop predictions from 20 different initial points for quantifying VPT and its mean value. The performance of different reservoir networks varies with the reservoir sizes. We find that CRC outperforms all the QRC architectures for the same reservoir sizes. The reservoir size corresponds to the state-vector dimension, which scales exponentially with the number of qubits (n). The best-performing quantum reservoir architecture, QRC-C4 with 9-11 qubits (n), can predict VPT values similar to the classical reservoir sizes of 500-2000.

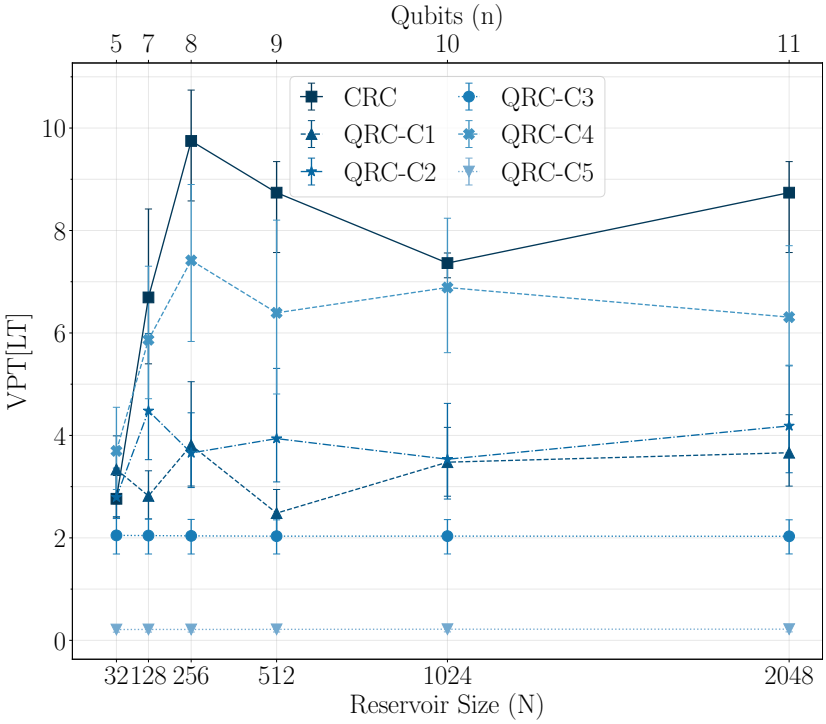


Figure 4.2: Lorenz-63 system. VPT vs. reservoir sizes for both classical and quantum reservoir configurations.

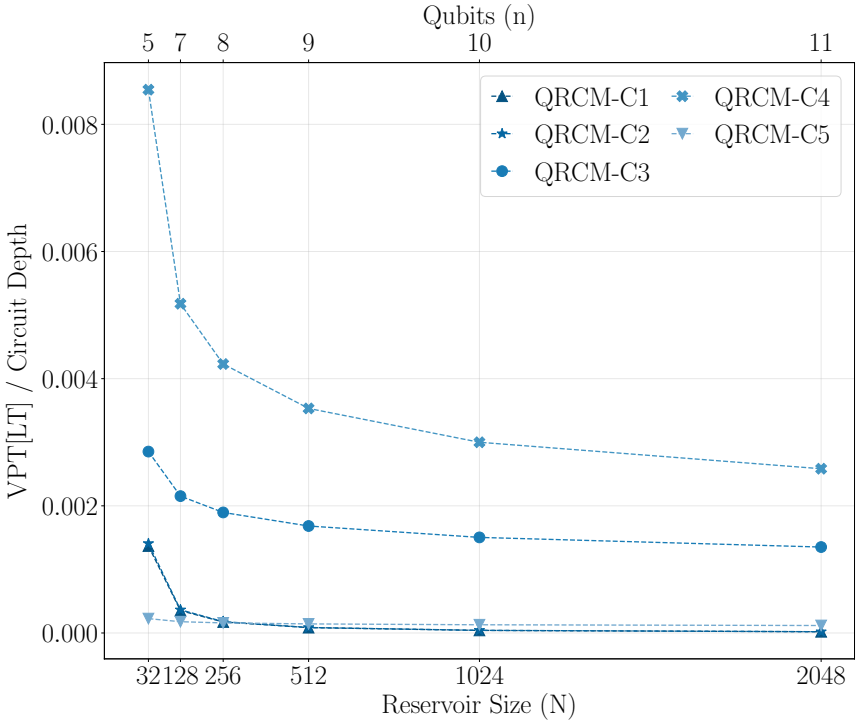


Figure 4.3: Lorenz-63 system. VPT/circuit depth vs. reservoir sizes for different quantum reservoir configurations.

We emphasise that the three QRC configurations C3-C5 do not have a feedback loop. These configurations evolve only from the input time series, and the information on reservoir activation states from previous time steps is provided via the classical leaky update Eq. 3.12. In contrast, architectures QRC-C1 and QRC-C2 involve an additional active feedback loop that encodes reservoir states to the quantum circuit at each time step, which leads to an additional overhead in the quantum-classical layer. The prediction

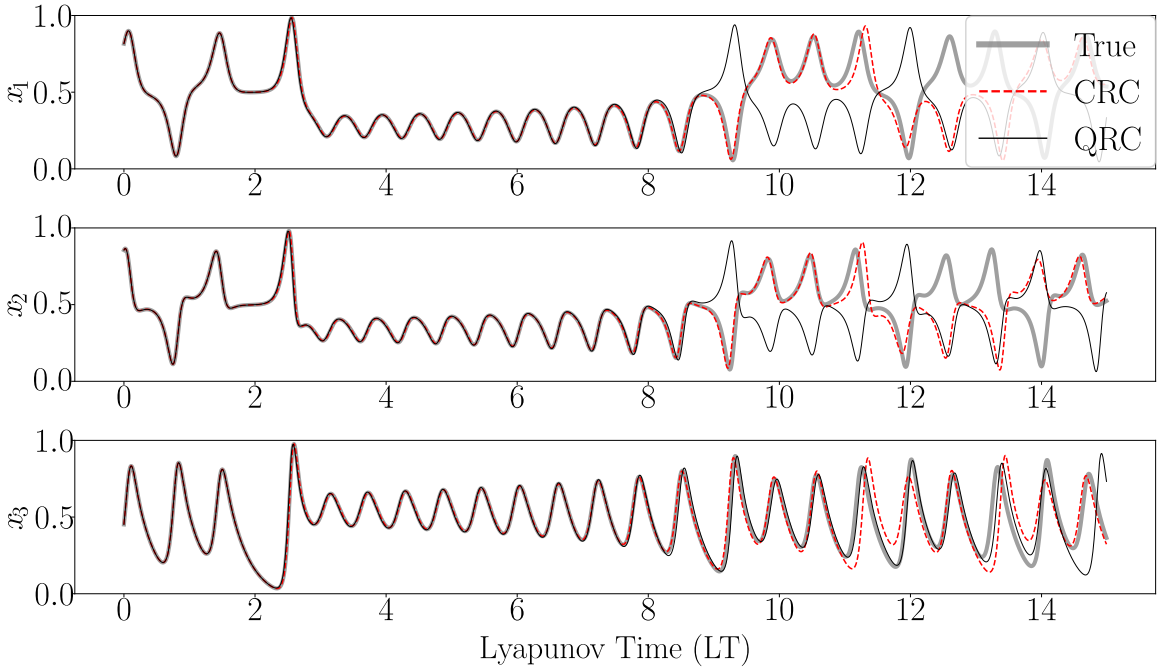


Figure 4.4: Lorenz-63 system. Closed-loop time-series predictions. Comparison of true predictions with classical reservoir computing (CRC) and emulated quantum reservoir computing (QRC).

performance of the two recurrence-Free QRC configurations, QRC-C3 and QRC-C5, does not improve with the reservoir sizes. The QRC-C4 architecture, however, shows comparable performance to the reservoirs with feedback loops (QRC-C1 and QRC-C2). These results highlight the importance of carefully chosen feature maps for QRC.

In Fig. 4.3, the VPT values normalised by the circuit depth are compared between different QRC architectures. We find that the proposed QRC-C4 architecture outperforms all other architectures by demonstrating the best prediction capabilities whilst requiring a smaller circuit depth.

Time-accurate predictions for the Lorenz-63 system are shown in Fig. 4.4. We display the best-performing CRC outcomes (reservoir size of 512), as well as predictions made by the QRC-C4 architecture using 9 qubits. Notice that both the quantum and classical reservoirs can accurately predict up to ~ 8.5 LTs. Beyond that, the prediction diverges from the true solution due to the inherent chaotic nature of the system.

Long-term statistical predictions

We select the best-performing networks for which the short-term predictions are shown in Fig. 4.4. We evolve the networks autonomously in a closed-loop for 250 LT, starting from an arbitrary point. The resulting statistics are shown in Fig. 4.5 for each state variable. Both quantum and classical reservoir

architectures are able to predict long-term statistics accurately.

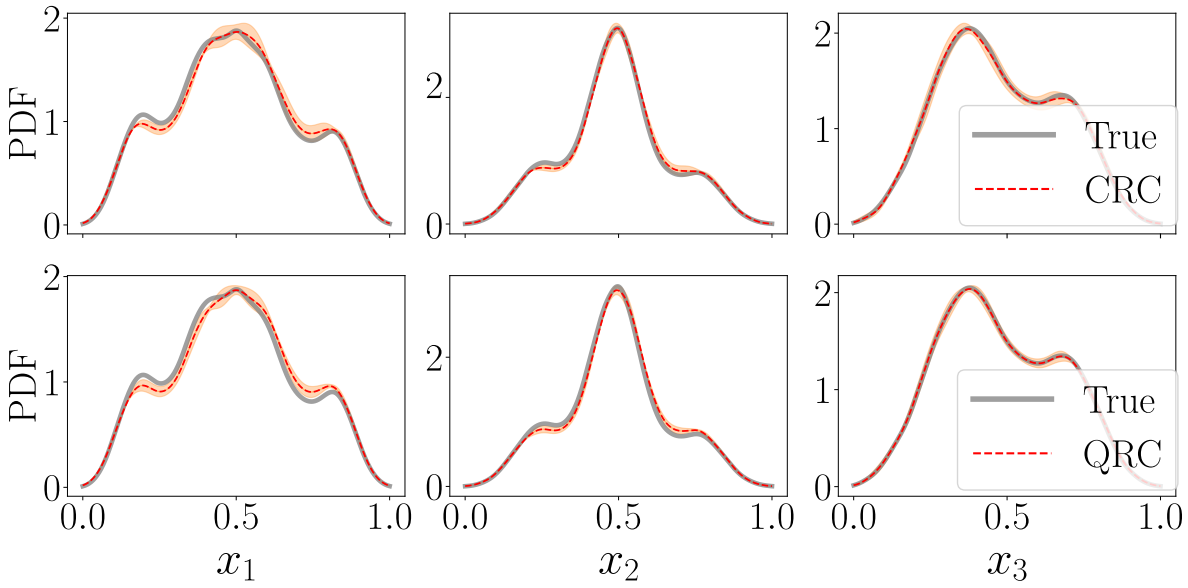


Figure 4.5: Lorenz-63 system. Closed-loop statistical predictions. The top panels present the result of the true and classical (CRC) network. The bottom panels compare the true with the QRC predictions. The shaded region in both figures indicates the uncertainty associated with different random seeds in both CRC and QRC.

4.2.3 Lorenz-96 system

In this section, we extend our comparison of classical and quantum reservoirs to higher-dimensional chaotic systems. We follow the same procedure defined in Sec. 4.2.2 and analyse the Lorenz-96 system (E. Lorenz 1995). The training set covers an evolution time of 200 LT for the Lorenz-96 system with a Leading Lyapunov exponent value of $\lambda_1 = 1.2$. In Tab. 4.2, we present the hyperparameters for the ten-dimensional Lorenz-96 system. The set of hyperparameters is derived by using the same procedure as discussed in Sec. 4.2.1.

Time-accurate predictions

We analyse the time-accurate prediction capabilities of CRC and different QRC architectures for a ten-dimensional reduced-order model of Lorenz-96. Fig. 4.6 shows the VPT values of both networks. The QRC architectures C1, C2, and C4 outperform the classical reservoir for increasing reservoir sizes. To study the robustness of the model, the results are averaged over five different sets of optimal hyper-parameters and twenty different starting points. For a particular realisation of hyperparameters and a starting point, at reservoir size 4096 the VPT value for CRC can be as large as $\sim$ 10 LT. However, the performance is not robust and degrades quickly for other hyperparameter choices and different initial points. Improved performance scaling and robustness indicate a potential quantum advantage in quantum reservoir computers.

Similar to the results in the Lorenz-63 system, the choice of a feature map and quantum reservoir architectures is critical for the performance. In Fig. 4.6, QRC-C4 and C1 have comparable VPT values at different reservoir sizes while QRC-C2 underperforms slightly. The performance of QRC-C3 and C5

Table 4.2: Parameters for ten-dimensional Lorenz-96 system.

Parameters	Symbol	Classical RC	Quantum RC
Time step	dt	0.01	0.01
Input scaling	σ_{in}	[0 , 1]	-
Spectral radius	$\rho(\boldsymbol{W})$	[0.1 , 1]	1
Tikhonov regularization	β	1×10^{-6},1×10^{-9},1×10^{-12}	1×10^{-6},1×10^{-9},1×10^{-12}
Leak rate	ϵ	[0.05, 1]	[0.05, 0.3]
Reservoir density	–	0.1,0.6,0.9	Configurations C1-C5

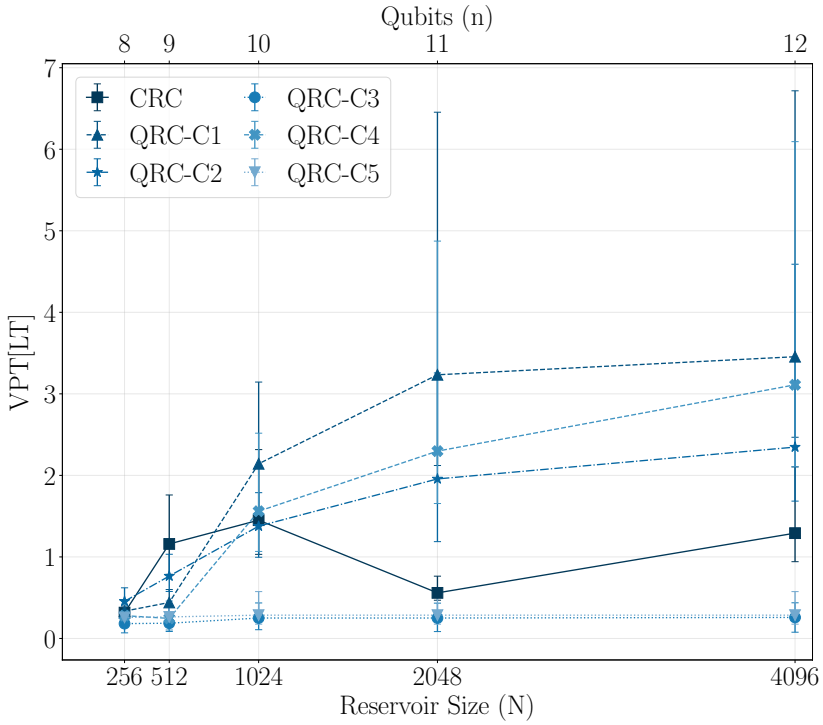


Figure 4.6: Lorenz-96 system with ten dimensions. VPT vs reservoir sizes for both classical and quantum reservoir configurations.

do not improve with the reservoir size because there is no feedback loop, and because the linear input feature map only utilises a fraction of the state space. By contrast, QRC-C4 (RF-QRC), which also does not have a recurrent feedback loop, has a fully connected input feature map that enriches the reservoir dynamics andhas a comparable VPT to QRC-C1 and C2.

As compared to QRC with recurrent connections, QRC-C4 achieves the highest VPT to circuit depth ratio for all of the reservoir sizes In Fig. 4.7. This is due to a lack of recurrence. In comparison, QRC-C1 and C2 require exponentially higher circuit depths at increasing reservoir sizes to achieve similar performance.

Long-term statistical predictions

In Figs. 4.8 and 4.9, we compare the long-term statistical predictions. The displayed results show the best-performing classical reservoir and quantum reservoir network (QRC-C4) outputs. We evolve these networks for 400 LT in a closed-loop, for a reservoir size of 1024. Both quantum and classical reservoir networks can recover long-term statistics of the chaotic model. The shaded region highlights the

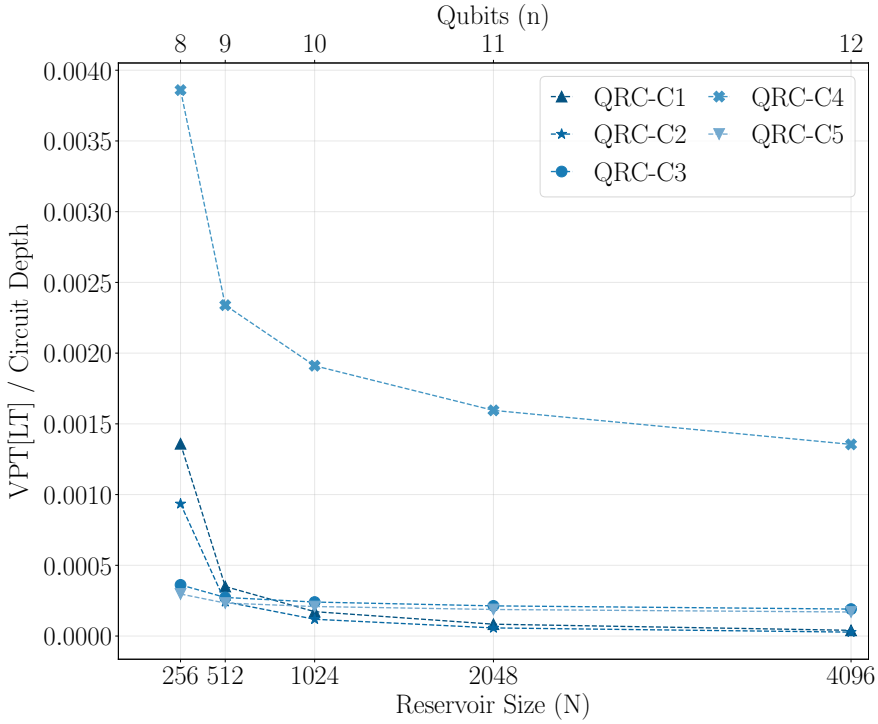


Figure 4.7: Lorenz-96 system with ten dimensions. VPT/circuit depth vs. reservoir sizes for different quantum reservoir configurations.

uncertainty of different realisations. We find that QRC-C4 predicts the statistical variables with smaller uncertainties when compared to the classical reservoir with equal reservoir sizes.

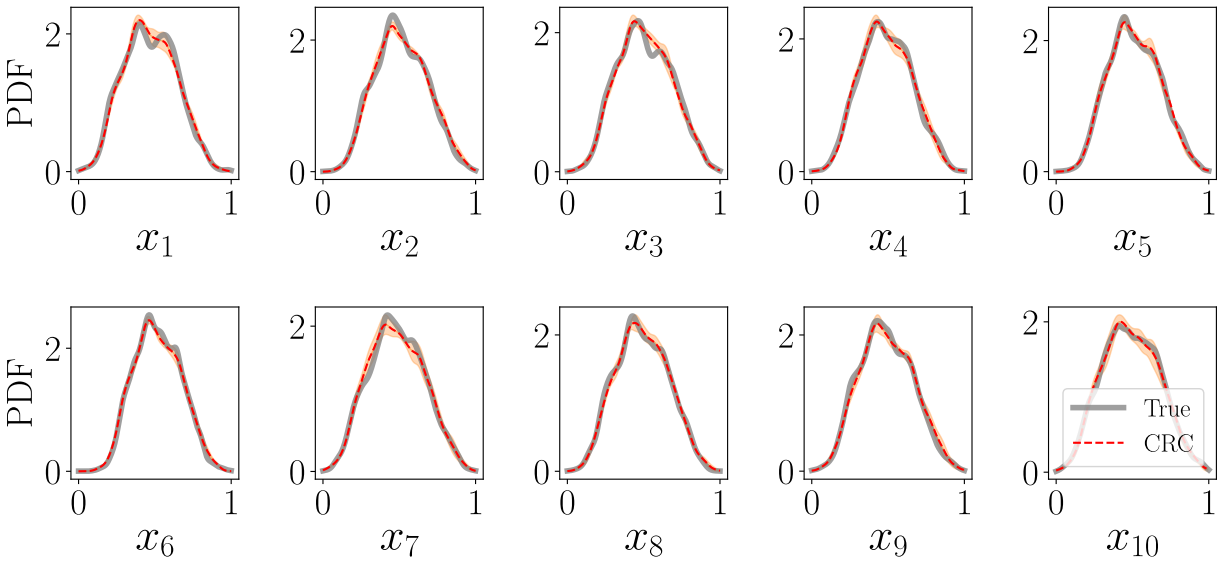


Figure 4.8: Lorenz-96 system with ten dimensions. Closed-loop statistical predictions using CRC. The shaded region indicates the uncertainty associated with different random seeds in CRC.

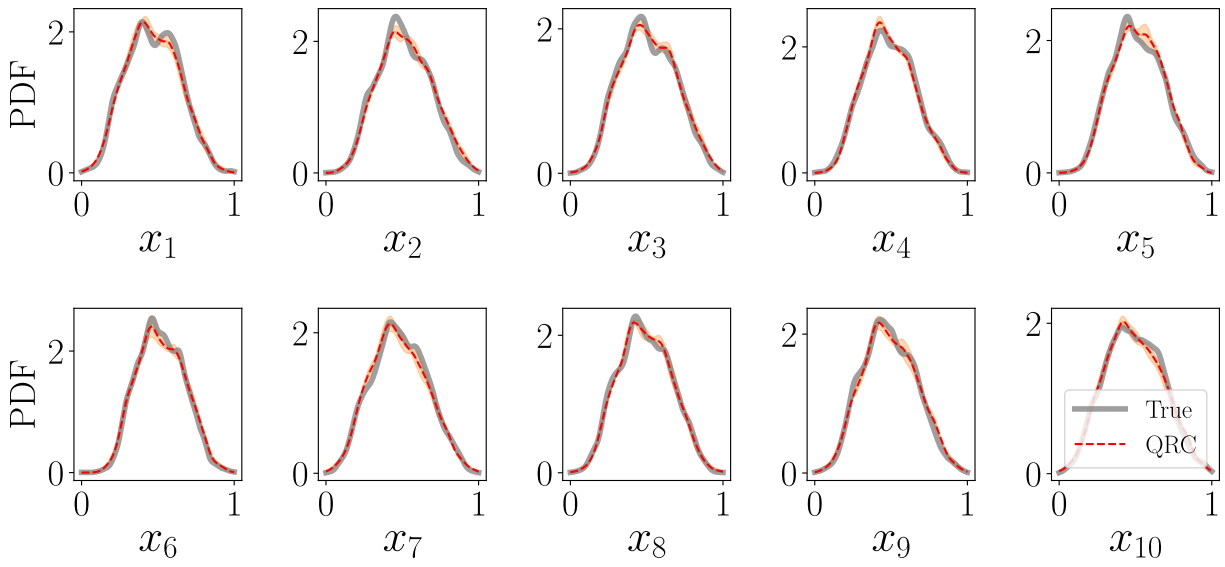


Figure 4.9: Lorenz-96 system with ten dimensions. Closed-loop statistical predictions using QRC. The shaded region indicates the uncertainty associated with different random seeds in QRC.

4.2.4 Chaotic shear flow with extreme events

We solve the MFE system ODEs using an RK4 solver with $dt = 0.25$. The leading Lyapunov Exponent is $\Lambda = 0.0163$ for the MFE model (Racca and Magri 2022a). Unlike the Lorenz-63 and Lorenz-96 systems, generating a single long-time series for washout, training, and test sets is not feasible for MFE because a single long-time series eventually laminarizes and gives limited information about the chaotic transients. To address this issue, first, we generate an ensemble of 2000 time series from different random initial points. We take the length of a single time series equal to 65 LT or 16000 time steps. Second, we discard the time series whose maximum kinetic energy is larger than the laminarization threshold $k_l = 0.48$. This threshold is selected to be close to the (asymptotic) laminarization value of $k = 0.5$. Out of the 2000 time series, 27% of the series laminarize. We divide the remaining 1441 time series into washout, training, and test sets. We train our network on 25 time series, each of length $20LT$. The test set consists of 500 time series. We compare the time-accurate and statistical predictions for the classical reservoir method against the best-performing quantum reservoir QRC-C4.

Table 4.3: Parameters for the MFE system.

Parameters	Symbol	Classical RC	Quantum RC
Time step	dt	0.25	0.25
Input scaling	σ_{in}	[0 , 1]	-
Spectral radius	$\rho(\boldsymbol{W})$	[0.1 , 1]	1
Tikhonov regularization	β	1×10^{-6},1×10^{-9},1×10^{-12}	1×10^{-6},1×10^{-9},1×10^{-12}
Leak rate	ϵ	[0.05, 1]	[0.05, 0.3]
Reservoir density	–	0.1,0.6,0.9	Configuration C4 (RF-QRC)

Time-accurate predictions

In the MFE model, we are concerned about quantifying the extreme event prediction capabilities of our reservoir networks. These capabilities are not accurately quantified by VPT, because extreme events are

difficult to predict due to the chaotic nature of the attractor. This means that the network can show good predictability in low kinetic energy regions while showing smaller VPT values near extreme events. This may produce an inaccurate representation of the predictability of our reservoir networks.

To quantify extreme event prediction capabilities, we use the Predictability Horizon (PH) (Racca and Magri 2022a; Pathak, Wikner, et al. 2018), which is defined as the time interval during which the predicted kinetic energy k_{pred} and true kinetic energy k_{true} are bounded by

$$\left| \frac{k_{pred}(t) - k_{true}(t)}{k_e - \bar{k}} \right| < 0.2, \tag{4.5}$$

where k_e is the extreme event threshold, which is equal to 0.1 here as in (Racca and Magri 2022a). $\bar{k}$ is the time average of the kinetic energy, and 0.2 is the user-defined error threshold (Racca and Magri 2022a). To quantify the PH, we sample 100 different extreme events from the unseen test set data. For each extreme event, we start the predictions 12 LT before the extreme event and discard the transients in the washout phase for 2 LT by running reservoir networks in an open loop. This gives δ_{k_e} of 10 LT. After this, we start autonomous closed-loop predictions with $\delta_{k_e} = 10$. The value of δ_{k_e} represents the time difference from the start of closed-loop predictions to the extreme event. If the PH value is greater than δ_{k_e}, we take PH equal to δ_{k_e}. If the prediction diverges before the extreme event occurs, or the PH value

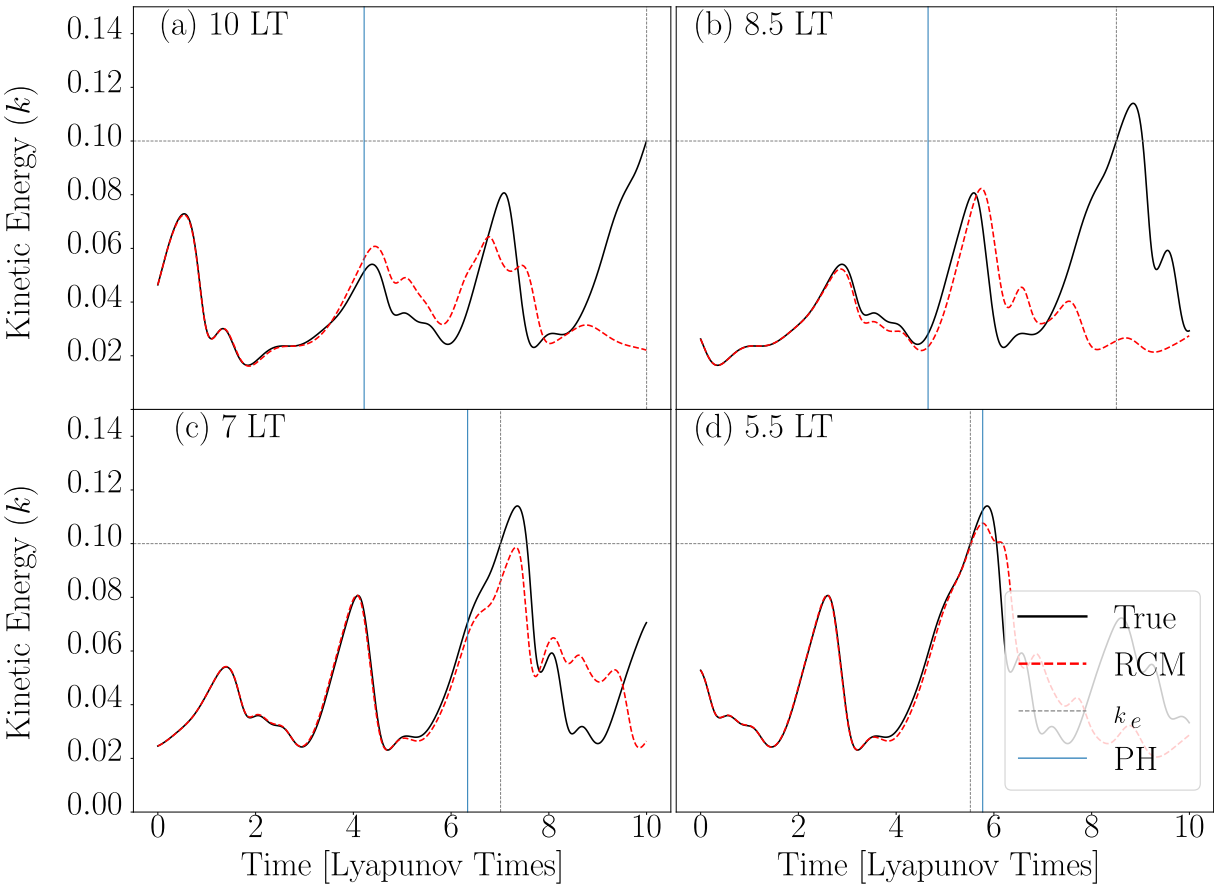


Figure 4.10: Visualization of the prediction of extreme events for both Reservoir Computing Methods (RCMs) i.e. both CRC and QRC (PH= 5.89LT).

is less than δ_{k_e}, we decrease the δ_{k_e} value by a factor of τ_e. We take τ_e = 0.5 LT and the new δ_{k_e} value equals to 9.5 LT. For the next step, we repeat the closed-loop predictions for the extreme event occurring at 9.5 LT, i.e. δ_{k_e} = 9.5 LT. We repeat the same method decreasing δ_{k_e} by τ_e until the PH is greater than δ_{k_e} (Fig. 4.10). That is, the extreme event is predicted correctly by the reservoir computers by the value of PH in the future.

We calculate the PH for a set of reservoir sizes for both quantum and classical reservoir networks. For a classical reservoir, the number of neurons is varied from $N_r = 256$ to $N_r = 2048$. We tune the hyperparameters for each reservoir and calculate the corresponding PH value. We also perform the same analysis for the quantum reservoir with qubits $n = 8$ to $n = 11$, which, in principle, corresponds to the classical reservoirs of $N_r = 256$ to $N_r = 2048$. The performance of the quantum reservoir also improves with the reservoir size. In Fig. 4.11, the comparison of classical and quantum reservoirs for various reservoir sizes is shown. Both reservoir networks have comparable performances for reservoir sizes $N_r = \{256, 512\}$, with the classical reservoir outperforming the quantum reservoir by a small margin. However, the PH value of the classical reservoir converges to a median value of around 5.5 LT. This convergence of PH value with increasing reservoir sizes is due to performance saturation in CRC, and it is consistent with the results in (Racca and Magri 2022a). In contrast, the quantum reservoir outperforms the classical reservoir for increasing reservoir sizes and has a median PH value of 8.09LT with 11 qubits.

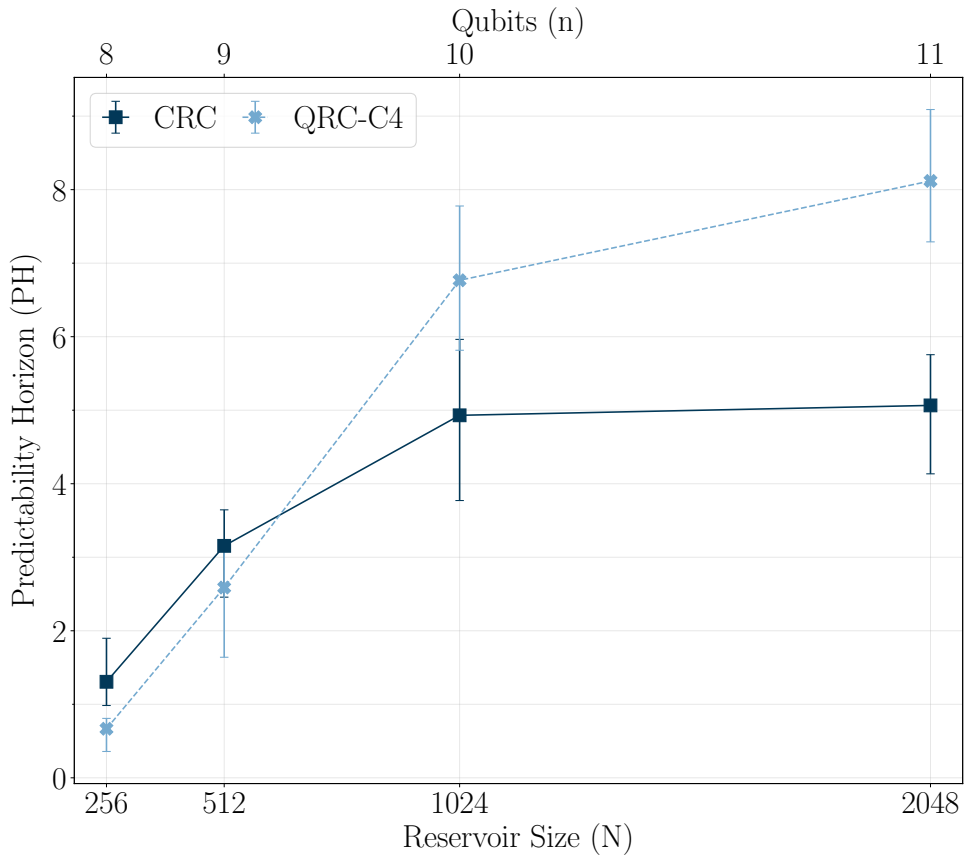


Figure 4.11: MFE system. Predictability Horizon (PH) for predictions with different reservoir sizes for the best performing CRC and QRC-C4 (RF-QRC) networks.

A second statistical measure to assess the extreme event prediction capabilities is the F-Score (Goutte and Gaussier 2005), which measures the model's accuracy by combining metrics such as precision and recall. To calculate the F-score, we measure three different occurrences: (a) an extreme event occurs in the true data set and the model also predicts the extreme event (True Positive, TP) (b) there is no extreme event in the true data but the model predicts an extreme event (False Positive, FP) (c) an extreme event occurs in the true data set but the model does not predict an extreme event, (False Negative, FN) (Racca and Magri 2022a). These three metrics are combined to calculate Precision (p), Recall (r), and F-score (F) from 1000 different starting points, which span 20 different time series

$$p = \frac{TP}{TP + FP}, \qquad r = \frac{TP}{TP + FN}, \qquad F = \frac{2}{p^{-1} + r^{-1}}, \tag{4.6}$$

We compare quantum and classical reservoirs for two different cases. First, we fix the reservoir size for $N_r = 1024$, which equals to 10 qubits for the quantum reservoir and varies the Prediction Time (PT). The Prediction Time (PT) is defined as the time interval between the extreme event and the start of the prediction. The PT value of 0 LT indicates that the prediction interval is between 0 and 1 LT, and PT = 3 indicates the prediction interval of 3 - 4 LT. The lower PT values have an F-score value closer to 1 because the network predicts nearly all the extreme events accurately, which are located within 1 LT of the start of predictions. As expected, the performance deteriorates with increasing PT value due to the inherent chaotic nature of the model. The proposed quantum reservoir architecture (QRC-C4) achieves a higher F-score as compared to the CRC for different PT intervals as shown in Fig. 4.12

Second, we fix the prediction time to 3 LT (PT = 3LT), which indicates the model is predicting extreme events between 3 and 4 LT. After fixing the PT value, we vary the reservoir sizes from $N_r = 256$ to $N_r = 2048$ and compare the performance metrics. The performances are similar for both quantum and classical reservoirs with reservoir sizes of $\{256, 512\}$. Similar to the results shown for the predictability horizon previously, the quantum architecture has a higher F-score value for 10,11 qubits as compared to the classical reservoirs of the same sizes (Fig. 4.12). All the results are computed for 5 different ensembles of networks and for 1000 different starting points.

Long-term statistical prediction

For the classical reservoir, we chose the reservoir size of $N_r = 512$. For the quantum reservoir, we chose a reservoir that corresponds to $n = 9$ qubits. We evolve the networks from different starting points in the test set of 500 time series, in which each starting point corresponds to a different time series. As previously mentioned, we perform a washout for each time series and discard the time series whose kinetic energy exceeds the laminarization threshold $k_l > 0.48$. Finally, we calculate the Probability Density Function (PDF)

Fig.4.13 presents the results for the statistical predictions against the true value. Both reservoirs can predict long-term statistics with high accuracy for the smaller kinetic energy values. Particularly, for $k < 0.15$, the prediction agrees with the true value of the statistics. This analysis provides a statistical measure of the prediction of extreme events. Both quantum and classical reservoirs can predict most of the extreme events ($k < 0.10$) in the test set. For the higher kinetic energy values, the log scale amplifies the difference between the true value and the predictions. The accuracy in the prediction of

extreme events could be improved by considering a larger reservoir size, which scales better for QRC as compared to CRC, as shown previously in Figs. 4.11 and 4.12. To further visualise the statistical predictions and compare CRC and RF-QRC, we predicted the velocity components of the flow fields from the predicted Fourier modes. In Fig. 4.14, the results of the average streamwise velocities are shown for the true data, and compared against classical and quantum reservoir predictions on the test data set. The predictions of RF-QRC approximate the true flow field better than the predictions from CRC, which have larger deviations. We conjecture that RF-QRC can be used to reconstruct flow fields with better accuracy than CRC with similar reservoir sizes. The prediction of flowfields with better accuracy also allows the RF-QRC model to predict turbulent bursts of relaminarization and thus kinetic energies (extreme events) with a better prediction horizon, as previously shown in Figs. 4.11 and 4.12

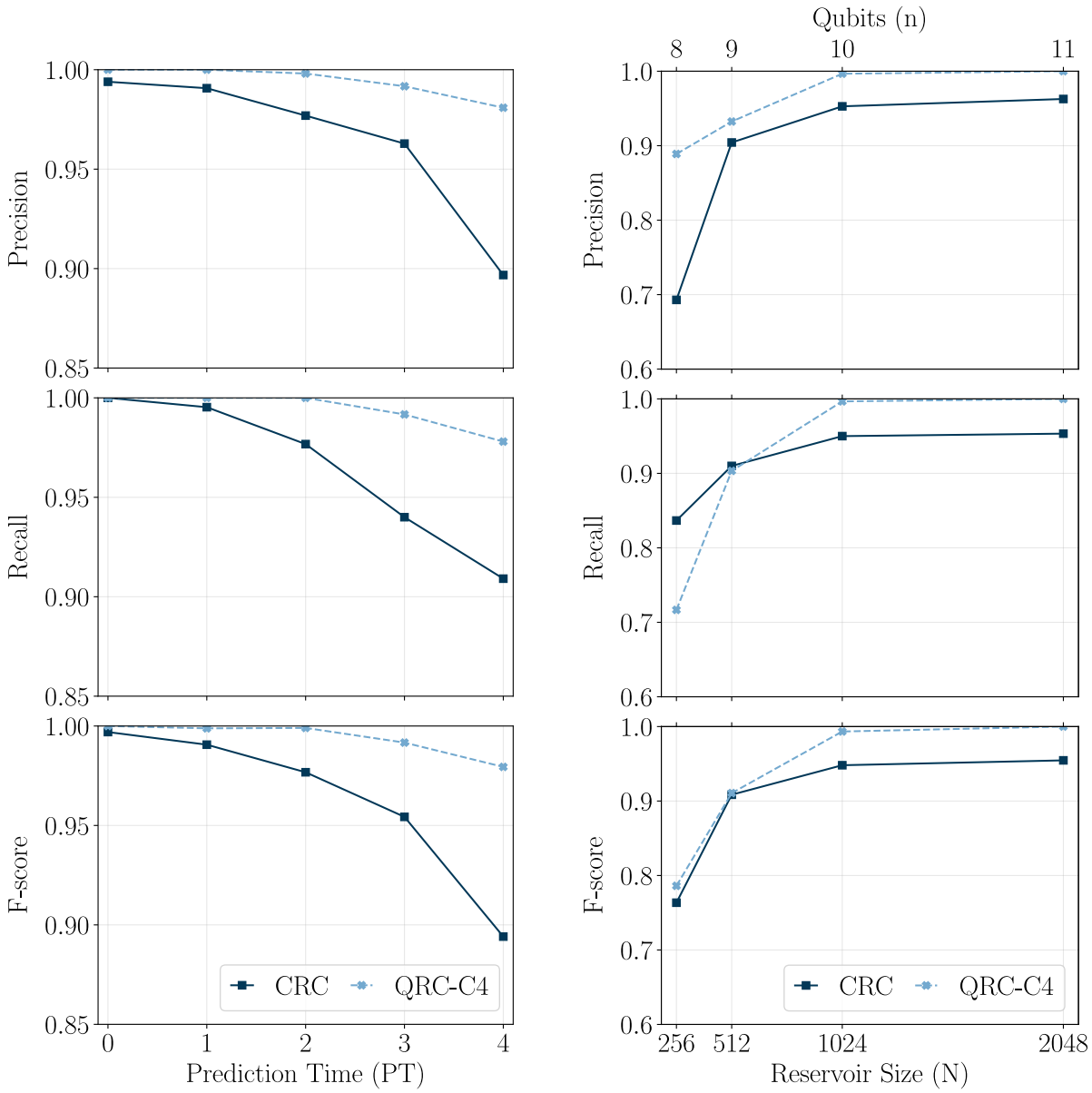


Figure 4.12: MFE system. 75^{th} percentile of Precision, Recall, F-score Vs. On the left side, Prediction Time (PT) for a constant reservoir size ($N = 1024$). On the right side, Reservoir Size for a constant Prediction Time ($PT = 3LT$).

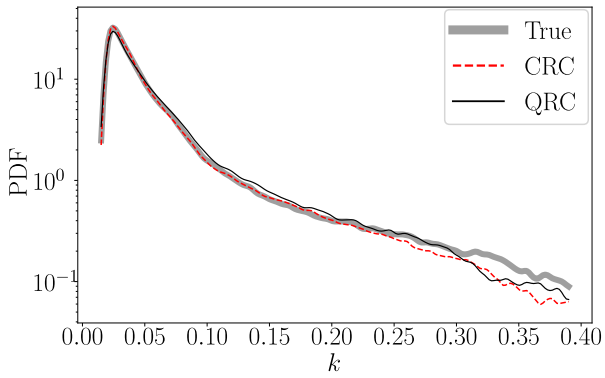


Figure 4.13: MFE system. Long-term statistical predictions with classical and quantum reservoir computing.

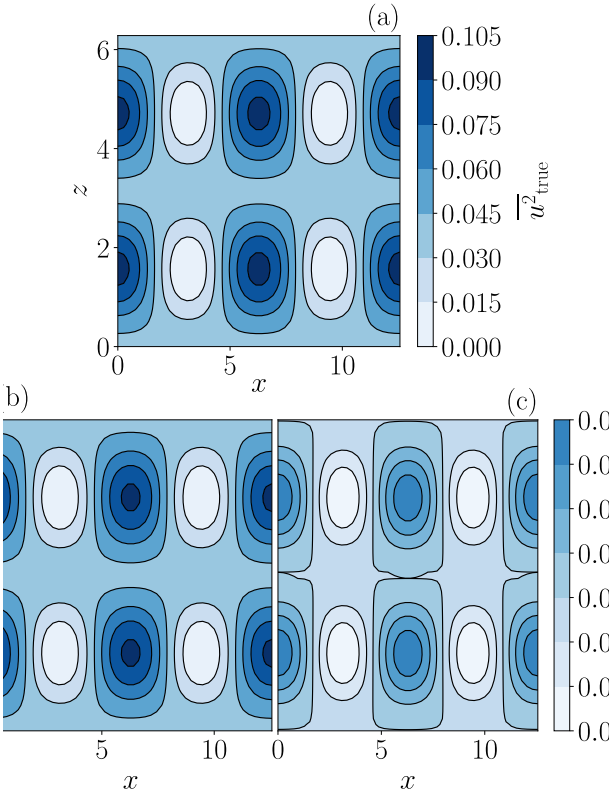


Figure 4.14: MFE system. The time average of the square of the midplane velocity along x. (a) True data (b) QRC-C4 predictions (c) CRC predictions for the best-performing networks with a reservoir size of 512 (9 qubits)

4.3 Stability and invariant properties

We extend the analysis of the best-performing quantum reservoir (RF-QRC) to analyse the stability properties of chaotic dynamical systems as discussed in Chap. 2. To compute stability properties, we follow the procedure outlined in (Huhn and Magri 2020c; Margazoglou and Magri 2023; Özalp, Margazoglou, and Magri 2023b) and the algorithm is presented in Algorithm. 1 for the RF-QRC.

4.3.1 Lorenz-63 system

We infer the invariant properties (Sec. 2.2) for a three-dimensional Lorenz-63 system. The time series data are obtained by the Runge-Kutta method with $dt = 0.01$. The numerical parameters are shown in Tab. 4.4. Each time series is divided into washout, training, and testing datasets (Brunton and Kutz 2022). The washout is the phase in which the transients are discarded to minimise the effect of initial conditions.

For the Lorenz-63 system, we employ reservoirs with 7, 8, and 9 qubits. The results are shown for the reservoir size of 7 qubits. The parameters $\boldsymbol{\alpha}$ of the random unitary $V(\boldsymbol{\alpha})$ are sampled from a uniform distribution between $[0, 4\pi]$. We train ten quantum reservoir networks, each with a different random seed, to reduce the effect of random initialisation on the model performance. The ensemble average is used in the figures. The training length is 20 LTs, where LT is the Lyapunov time (Tab. 4.4). After training, in the autonomous phase, we deploy the RF-QRC to infer the invariant properties of the physical system.

Both the ground truth and inferred Lyapunov spectra are obtained by autonomously evolving the network for 50 LTs. The quantum reservoir accurately infers the positive λ_1, neutral λ_2 and negative λ_3 Lyapunov exponents, as shown in Tab. 4.5.

Algorithm 1: An algorithm to compute LEs with RF-QRC

$\boldsymbol{W}_S \leftarrow random \in \mathbb{R}^{N_r \times D}$; /* Initialize D GSVs */
$\boldsymbol{Q}, \boldsymbol{R}_S \leftarrow QR(\boldsymbol{W}_S)$; /* Orthonormalize GSVs */
$\boldsymbol{W}_S \leftarrow \boldsymbol{Q} \in \mathbb{R}^{N_r \times D}$
$N_{QR} \leftarrow N_{test} - N_w$; /* Number of QR decompositions */
Save the time series of $\boldsymbol{R}_S$ *and* $\boldsymbol{Q}$ *for CLVs calculation*
Initialize $\tilde{\boldsymbol{R}}_S \leftarrow \boldsymbol{0} \in \mathbb{R}^{D \times D \times N_{QR}}$
Initialize $\tilde{\boldsymbol{Q}} \leftarrow \boldsymbol{0} \in \mathbb{R}^{N_r \times D \times N_{QR}}$
Initialize $\boldsymbol{\Lambda} \leftarrow \boldsymbol{0} \in \mathbb{R}^{D \times N_{QR}}$
$\boldsymbol{J}$ = jacobian(QRC) ; /* Jacobian of the quantum circuit is computed in Pennylane(Bergholm et al. 2018) as in Eq. 3.21 */
Evolve the hidden state and GSVs simultaneously.
Skip the initial transients for N_w *time steps.*
$n \leftarrow 0$; /* Increment the number of QR decompositions */
for $i = 0 : N_{test} - 1$ **do**
  $\boldsymbol{r}(t_{i+1}) = f(\boldsymbol{r}(t_i))$; /* QRC state update as in Eq. 3.12 */
  $\boldsymbol{u}(t_{i+1}) = [\boldsymbol{r}(t_{i+1})]^{\mathrm{T}} \boldsymbol{W}_{out}$
  $\boldsymbol{J} \leftarrow \boldsymbol{J}(\boldsymbol{r}(t_i))$; /* The updated Jacobian */
  $\boldsymbol{W}_S \leftarrow \boldsymbol{J}\boldsymbol{W}_S$; /* The variational equation */
  $\boldsymbol{Q}, \boldsymbol{R}_S \leftarrow QR(\boldsymbol{W}_S)$; /* QR at every time step */
  $\boldsymbol{W}_S \leftarrow \boldsymbol{Q}$
  if $i \geq N_w$ **then**
    $\boldsymbol{\Lambda}[:, n] \leftarrow \log(\mathrm{diag}[\boldsymbol{R}_S])/dt$; /* Save Finite time LEs */
    $\tilde{\boldsymbol{R}}_S[:, :, n] \leftarrow \boldsymbol{R}_S$; /* Save $\boldsymbol{R}_S$ */
    $\tilde{\boldsymbol{Q}}[:, :, n] \leftarrow \boldsymbol{Q}$; /* Save $\mathbf{Q}$ */
    $n = n + 1$;
  end
end
$\lambda_j = \frac{1}{N_{QR}} \sum_{i=0}^{N_{QR}-1} \Lambda[j, i]$; /* The jth Lyapunov exponent */

Table 4.4: Parameters for the tests on the Lorenz-63 and Lorenz-96 systems.

Parameters		3-D Lorenz-63	10-D Lorenz-96	20-D Lorenz-96
Time step	Δt	0.01	0.01	0.01
Leading Lyapunov exponent	λ_1	0.9056	1.2	1.5
Lyapunov Time	LT	1 LT = 110 steps	1 LT = 83 steps	1 LT = 66 steps
Washout steps	N_w	5 LT	10 LT	10 LT
Training steps	N_{tr}	20 LT	200 LT	200 LT
Tikhonov regularization	β	1×10^{-9},1×10^{-12}	1×10^{-9},1×10^{-12}	1×10^{-9},1×10^{-12}
Input scaling	σ	[0,1]	[0,1]	[0,1]
Leak rate	ϵ	0.21	0.15	0.12
Number of qubits	n	7	9	13
Reservoir size	N_r	128	512	8192
Reservoir connectivity	–	Fully connected	Fully connected	Fully connected

Table 4.5: Lyapunov spectrum of the Lorenz-63 system. Comparison between the ground truth (target) and RF-QRC (inferred) model.

Exponents	Target	RF-QRC
1	0.9051	0.9173
2	8.5×10^{-3}	9.6×10^{-3}
3	-14.56	-14.65

Fig. 4.15 shows the distributions of the CLV angles between the unstable-neutral subspaces, $\theta_{U,N}$, the unstable-stable subspaces, $\theta_{U,S}$, and the neutral-stable subspaces, $\theta_{U,S}$. The RF-QRC model accurately infers the long-term statistical distributions of the CLV angles, which means that the RF-QRC can accurately infer the geometric structure of the tangent space. Fig. 4.16 shows the angles between the subspaces on the chaotic attractor. When there are no tangencies, i.e., the subspaces are separated, the system is hyperbolic with structurally stable dynamics and linear response (Lucarini et al. 2014).

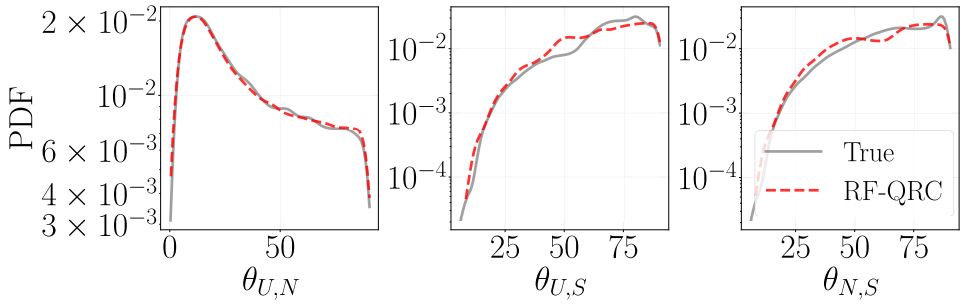


Figure 4.15: Lorenz-63 system, RF-QRC with a reservoir of 7 qubits. Probability density functions (PDFs) of the inferred angles between the three subspaces, where U stands for unstable, N for neutral, and S for stable.

4.3.2 Lorenz-96 system

We investigate the Lorenz-96 system (E. Lorenz 1995) with 10 and 20 degrees of freedom. Tab. 4.4 shows the model parameters. We train ten different networks to reduce the effect of the random initialisation due to $\boldsymbol{\alpha}$. The training data is 200 LT long. After training with different reservoir sizes, we find that a good reservoir size is made of 9 qubits for a 10-dimensional case and 13 qubits for a 20-dimensional Lorenz-96 system. In the prediction phase, the reservoir is evolved for 60 LTs. As shown in Fig. 4.17, the

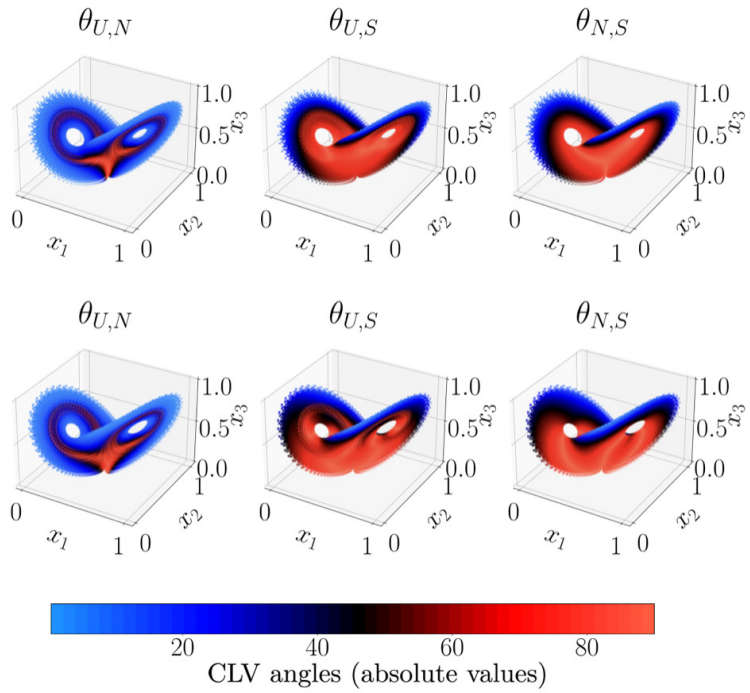


Figure 4.16: Lorenz-63 system. Angles between subspaces. Top: Ground truth solution. Bottom: Prediction with the RF-QRC with a reservoir of 7 qubits.

RF-QRC accurately learns the Lyapunov spectrum from data. Tab. 4.6 shows that the RF-QRC networks accurately infer the Kaplan-Yorke dimensions of the attractors.

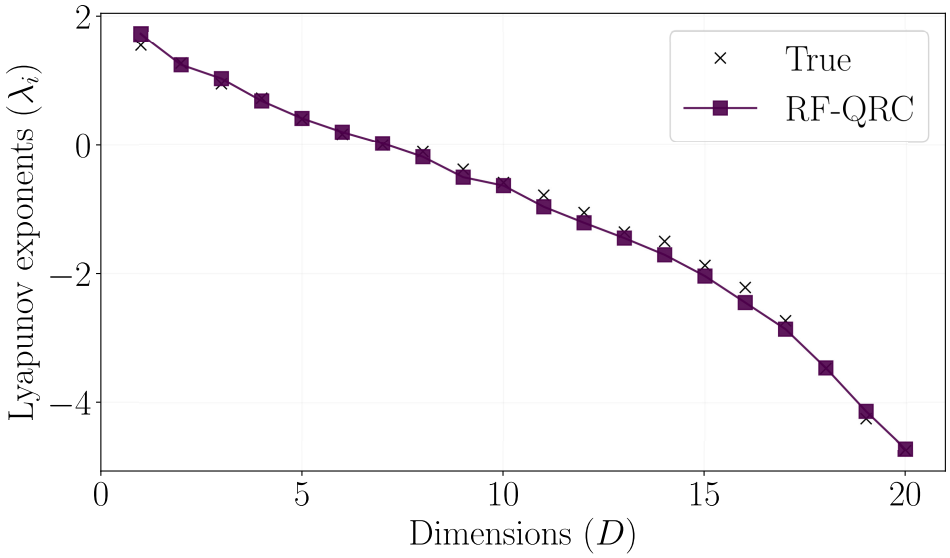


Figure 4.17: Lyapunov spectrum of the Lorenz-96 system with 20 dimensions. The RF-QRC has a reservoir with 13 qubits.

Table 4.6: Kaplan-Yorke dimensions. Comparison between the ground truth (target) and the RF-QRC (inferred) model. The % error is defined as the absolute relative error between target and inferred values.

System	Target	RF-QRC	% Error
Lorenz-63	2.06	2.06	0
Lorenz-96 (10-D)	6.52	6.58	0.92
Lorenz-96 (20-D)	13.40	13.25	1.1

4.4 Conclusion

This chapter demonstrates that quantum reservoir computers can forecast chaotic dynamics and extreme events, while also inferring the invariant properties of chaotic attractors across both low- and higher-dimensional systems.

By exploiting quantum circuit ansätze and entanglement, we design and analyse reservoir computing architectures for data-driven prediction of nonlinear dynamical systems. We show that the choice of feature map, which encodes both data and entanglement structure, plays a key role in determining the performance of the reservoir.

We introduce the recurrence-free quantum reservoir computer (RF-QRC), a scalable architecture with shallow circuit depth that does not grow with the number of qubits. Unlike standard QRCs, RF-QRC does not rely on explicit recurrence within the quantum circuit, and instead retains memory through leaky integration. This removes the need for a classical feedback loop during training and makes the approach suitable for near-term quantum hardware.

Through numerical experiments on prototypical chaotic systems, including the Lorenz-63 and Lorenz-96 systems, we show that: (i) the forecasting performance of both classical and quantum reservoirs improves with reservoir size up to saturation; (ii) for low-dimensional systems, classical reservoir computers achieve longer predictability horizons for the same degrees of freedom; (iii) for higher-dimensional systems, RF-QRC outperforms classical reservoirs in predictability for comparable resources; and (iv) RF-QRC captures extreme events while scaling more favourably with system dimension.

In addition to short-term forecasting, we show that quantum reservoir computers can recover long-term statistical behaviour and invariant quantities, including Lyapunov spectra, attractor dimensions, and covariant Lyapunov vectors. These results establish quantum reservoir computing, and in particular RF-QRC, as a promising framework for scalable prediction and analysis of chaotic dynamical systems. In Chap. 5, we extend this analysis beyond the ideal noise-free setting to account for finite-sampling noise, propose methods to mitigate its effects and implement it on real quantum hardware.

CHAPTER
5

Noise, mitigation and hardware implementation

In previous chapters, we have shown that quantum reservoir computers can forecast chaotic dynamics, extreme events and can infer the invariant properties of the physical systems. For higher-dimensional chaotic systems, we show that QRC can outperform its classical counterparts when emulated in an ideal noise-free setting. However, any prospects of achieving quantum advantage require addressing noise issues for practical reservoir implementations. The combination of leaky-integrate neurons with recurrence-free quantum reservoir dynamics provides a pathway towards noise suppression because of the individual state-dynamics of the associated neurons (Sec. 3.2).

In this chapter, we analyse the effects of finite-sampling noise on the performance of quantum reservoir computers. In Sec. 5.1, we give a theoretical description of finite sampling noise in quantum reservoir computers and use it to model the uncorrelated noise in RF-QRCs. We compare QRC and RF-QRC with finite sampling noise, and present two denoising methods based on singular value decomposition (SVD) and signal smoothing techniques in Sec. 5.2. These denoising methods are then applied to the Lorenz-63 and MFE model. Finally, in Sec. 5.3, the denoising techniques and parallel training of RF-QRCs are then tested on noisy quantum hardware. This chapter primarily contains the contributions from (O. Ahmed, Tennie, and Magri 2025a).

5.1 Quantum reservoir computing with finite samples

Before analysing the effect of finite sampling noise on RF-QRC, we first consider the effect of finite-sampling noise analytically. Due to the absence of active recurrence in RF-QRC, finite-sampling fluctuations arising from independent measurements can generally be approximated as temporally uncorrelated.

When neglecting hardware noise, the underlying reservoir state $\boldsymbol{r}(t_i)$ is obtained from repeated measure-

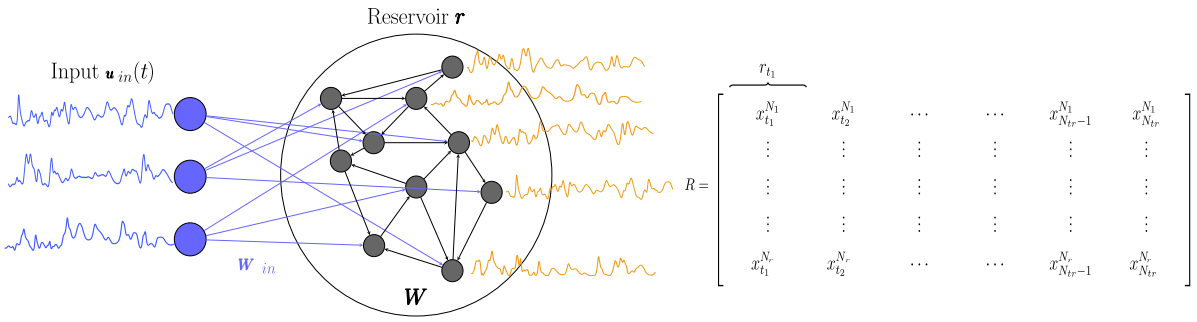


Figure 5.1: Training phase in reservoir computing. Input time series $\boldsymbol{u}_{in}(t)$ is mapped to a reservoir using the input matrix $\boldsymbol{W}_{in}$ in CRC and suitable encoding schemes in QRC. Inside the reservoir r, each neuron *echoes* with the input time series to generate a sequence of reservoir activation state signals. The reservoir activation signals are concatenated into the reservoir state matrix $\boldsymbol{R}$, which is then used to determine the optimal output weight matrix $\boldsymbol{W}_{out}$ using ridge regression.

ments of the quantum observables. For S independent measurements,

$$\boldsymbol{r}(t_i) = \lim_{S\to\infty} \frac{1}{S} \sum_{j=1}^{S} \boldsymbol{r}^{(j)}(t_i), \tag{5.1}$$

where $\boldsymbol{r}^{(j)}(t_i) \in \{0,1\}^{N_r}$ denotes the j-th single-shot measurement outcome vector in the computational basis. The empirical estimator of the reservoir activation vector is therefore

$$\bar{\boldsymbol{r}}(t_i) = \frac{1}{S} \sum_{j=1}^{S} \boldsymbol{r}^{(j)}(t_i). \tag{5.2}$$

For sufficiently large S, the effect of finite sampling noise on the reservoir activation states can be represented asymptotically as a stochastic fluctuation process $\boldsymbol{\zeta}(t)$

$$\bar{\boldsymbol{r}}(t_i) = \boldsymbol{r}(t_i) + \frac{1}{\sqrt{S}} \boldsymbol{\zeta}(t_i), \tag{5.3}$$

where the $1/\sqrt{S}$ scaling follows from the central limit theorem.

Like in classical reservoir computing, measured quantum observables or computational-basis probabilities play a role analogous to neuron activations in QRC. For an n-qubit system, the reservoir dimension scales as $N_r = 2^n$. The shot-noise fluctuations may therefore be interpreted as time-dependent noise acting on the reservoir activation signals, as illustrated in Fig. 5.1. The stochastic fluctuations in the individual reservoir activation vectors translate to fluctuations in the reservoir state matrix $\boldsymbol{R}$ with stochastic perturbations $\boldsymbol{Z}$

$$\bar{\boldsymbol{R}} = \boldsymbol{R} + \frac{1}{\sqrt{S}} \boldsymbol{Z}. \tag{5.4}$$

Quantum sampling noise of a single qubit follows a binomial distribution (Schuld and Petruccione 2021). For multiple qubits measured in the computational basis with finite samples, the sampling fluctuations are described by multinomial statistics. The lack of recurrence in RF-QRC allows these finite-sampling fluctuations to be approximated as temporally uncorrelated. This form of noise has also been considered in (F. Hu et al. 2023) in the context of Resolvable Expressive Capacity (REC).

Consequently, in Eq. 5.4, $\boldsymbol{Z}$ is a centred stochastic process with covariance determined by multinomial sampling statistics, since $\mathbb{E}[\bar{\boldsymbol{r}}(t)] = \boldsymbol{r}(t)$. The stochastic process may therefore be characterised through its second-order moments, which define the covariance matrix

$$\boldsymbol{\Sigma}_{ab}(t) = \mathrm{Cov}\left[\zeta_a(t), \zeta_b(t)\right]. \tag{5.5}$$

Averaging the covariance over the training samples yields

$$\boldsymbol{V}_Z = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \left(\mathrm{diag}[\boldsymbol{r}(t_i)] - \boldsymbol{r}(t_i)\boldsymbol{r}(t_i)^T\right), \tag{5.6}$$

which corresponds to the training-time average of the covariance matrices $\boldsymbol{\Sigma}(t_i)$.

Therefore, the covariance of the finite-shot perturbation in the observed reservoir activations is

$$\boldsymbol{V}_{\overline{R}} = \frac{1}{S}\boldsymbol{V}_Z. \tag{5.7}$$

Here, $\boldsymbol{r}(t_i)$ denotes the underlying noiseless reservoir activation vector at the i-th training timestep. In practice, this vector is not directly accessible and is estimated using the sampled activation vector $\bar{\boldsymbol{r}}(t_i)$. This substitution introduces higher-order finite-sampling corrections, which become negligible for sufficiently large S.

The outer-product term in Eq. 5.6 corresponds to the feature-correlation structure that naturally appears in the ridge-regression loss Eq. 3.16. Since the ridge-regression loss is quadratic in the reservoir activations, second-order statistics such as means and covariances are generally assumed to be sufficient for analysing the leading-order effect of sampling noise on training performance (F. Hu et al. 2023; Khan, Kaufman, et al. 2023).

5.1.1 Model parameters

In this section, we provide an overview of the parameters for the RF-QRC model implemented in this chapter to study sampling noise. We applied the RF-QRC model to study a Lorenz-63 and a MFE model. The time series data for both of these models are derived using a Runge-Kutta method. The time step, dt, for the numerical schemes and hyperparameters for RF-QRC are shown in Tab. 5.1. Each time series is divided into a washout, training, and testing dataset.

Table 5.1: Parameters for the tests on the Lorenz-63 and MFE systems with the recurrence-free quantum reservoir computing architecture.

Parameters	Symbol	Lorenz-63 system	MFE system
Time step	dt	0.01	0.25
Leading Lyapunov exponent	λ_1	0.9056	0.0163
Lyapunov Time	LT	1 LT = 110 steps	1 LT = 245 steps
Washout steps	N_w	5 LT	2 LT
Training steps	N_{tr}	20 LT	65 LT
Tikhonov regularization	β	1×10^{-9},1×10^{-12}	1×10^{-9},1×10^{-12}
Leak rate	ϵ	[0.05, 0.3]	[0.05, 0.3]
Reservoir connectivity	-	Fully connected	Fully connected

We have rescaled our time units with the inverse of the Lyapunov exponent (λ_1), which is the Lyapunov time (LT). Based on the discretisation time step, 1 LT corresponds to 110 steps for Lorenz-63 and 245 steps for the MFE system, as shown in Tab. 5.1. After prescribing quantum feature maps, the only tunable hyperparameters are the Tikhonov regularisation β and the leak-rate ϵ in RF-QRC (Chap. 4). We perform a grid search to find optimal parameters between the ranges specified in Tab. 5.1.

5.2 Noise mitigation strategies

In this section, we analyse the impact of finite sampling noise on the prediction capabilities of QRC and RF-QRC and propose methods to reduce it. After training classical and quantum reservoirs on the MFE model, we perform autonomous closed-loop predictions for both networks to compare the performances.

In Fig. 5.2, results of the time-series prediction are shown for an ideal probability distribution (assuming an infinite number of measurements, sometimes referred to as 'shots'), and for various different learning outcomes based on a variable number of shots, S. A minimum number of finite samples is required to improve the forecasting abilities of QRC beyond classical reservoir computers. To address the performance deterioration with a finite number of shots in QRC, we first outline the working of the reservoir computers.

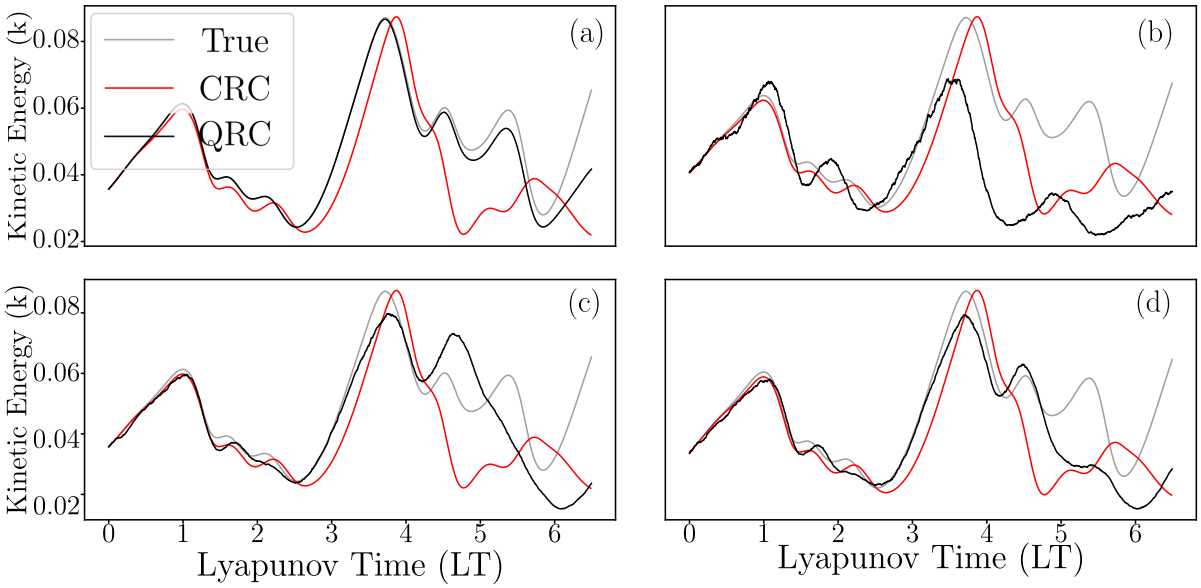


Figure 5.2: Prediction of the MFE time-series on an unseen dataset with (a) noise-free probability distribution, (b) 0.5×10^5 shots, (c) 2×10^5 shots, (d) 4×10^5 shots. Classical reservoir computing (CRC) results from the best-performing reservoir of size 1024 compared with the best-performing quantum reservoir computing (QRC) of 10 qubits. QRC hyperparameters are detailed in Sec. 5.1.1

The working principle of reservoir computing is to combine the generated reservoir activation signals for functional approximation of the dynamical systems (Fig. 5.1). This is done by minimising the loss function over the input training dataset. Reservoir signals have a large overlap with a relatively lower-dimensional manifold of active states, which is the active space (Carroll 2020). More specifically, the

singular value decomposition of common reservoir matrices reveals only a limited number of relevant singular values, and any denoising procedure should preserve the corresponding eigenspaces.

For a number of chaotic systems, the time series signals have corresponding reservoirs with low-dimensional active spaces (Carroll 2020). Often, in the classical reservoir computing framework, reservoir matrices with high-dimensional active spaces are associated with lower testing errors (Carroll and Pecora 2019). However, in QRC, we find that, although the addition of finite sampling noise in QRC reservoirs increases the dimensionality of the active space, it does not necessarily improve the learning performance. REC analysis (F. Hu et al. 2023) shows that in the presence of noise, taking more activation states than a given threshold results in poor functional approximation. Therefore, in order to perform an analysis of the impact of noise on the learning performance, we consider the signal-to-noise (SNR) ratios of individual reservoir activation signals. We also compute the training loss to assess trainability with respect to sampling noise.

In Fig. 5.3, we compare the SNR ratio for QRC and RF-QRC for the MFE system. Our results indicate that the presence of correlated noise in QRC results in more noisy estimates of the reservoir activation signals than RF-QRC because of the propagation of correlations in QRC. An example of the noisy and denoised reservoir activation state signals for RF-QRC, measured on a 10 qubits system, can be seen in Fig. 5.4. For a constant number of samples, denoising reservoir activation signals results in a better fit of the training signal with lower noise variance. We now discuss the details of the two approaches utilised for suppressing noise in RF-QRC.

5.2.1 Noise suppression using SVD

In signal processing, singular value decomposition (SVD) is a method to improve the SNR of noisy signals (Shlens 2014; Schanze 2017; Jha and Yadava 2010). The precision of unbiased estimators of expectation values in quantum computation is fundamentally limited by the Cramér–Rao bound (M. Yu et al. 2022). For an ensemble of quantum systems governed by an input time series, the resulting expectation values form a reservoir signal contaminated by finite-sampling noise. We found that the SNR of these noisy reservoir signals can be improved using classical signal-processing tools such as SVD. For a noisy reservoir-state matrix $\overline{\boldsymbol{R}} \in \mathbb{R}^{N_r \times N_{tr}}$, the compact singular value decomposition is

$$\overline{\boldsymbol{R}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T, \tag{5.8}$$

where $q = \min(N_r, N_{tr})$, $\boldsymbol{U} \in \mathbb{R}^{N_r \times q}$ and $\boldsymbol{V} \in \mathbb{R}^{N_{tr} \times q}$ have orthonormal columns, and $\boldsymbol{S} \in \mathbb{R}^{q \times q}$ is diagonal with non-negative singular values. To maximise the SNR, we construct a low-rank approximation of the noisy reservoir matrix by truncating singular values below a prescribed threshold, which are assumed to be associated primarily with the noise component.

We apply SVD to denoise reservoir activation signals in the Lorenz-63 and MFE systems, considering two cases for each system: (a) The low-rank approximation of the noisy reservoir matrix is compared to the original noisy matrix, which contains more reservoir activation signals but includes noise. This approach evaluates the impact of truncating approximately half of the singular values to reduce noise. And (b) the low-order reduced rank approximation after SVD is compared with the reduced noisy matrix of the same size, effectively having the same number of reservoir activation signals. This is relevant when

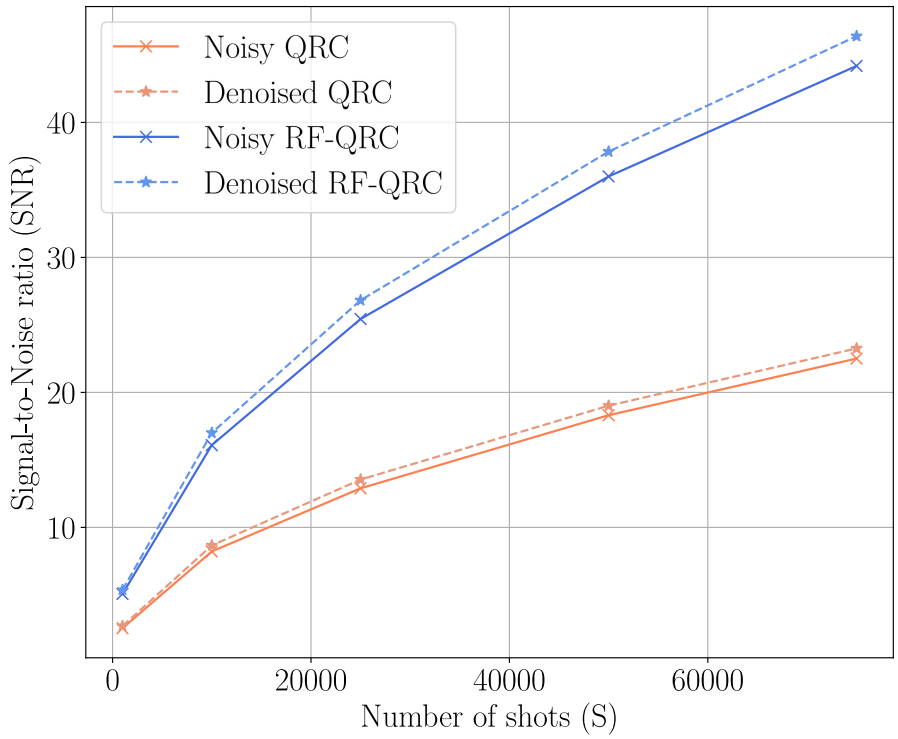


Figure 5.3: Averaged signal-to-noise ratios (SNR) of the reservoir states of QRC and RF-QRC with and without denoising filters. The underlying model is an MFE time series with a reservoir size of 10 qubits ($N_r = 1024$).

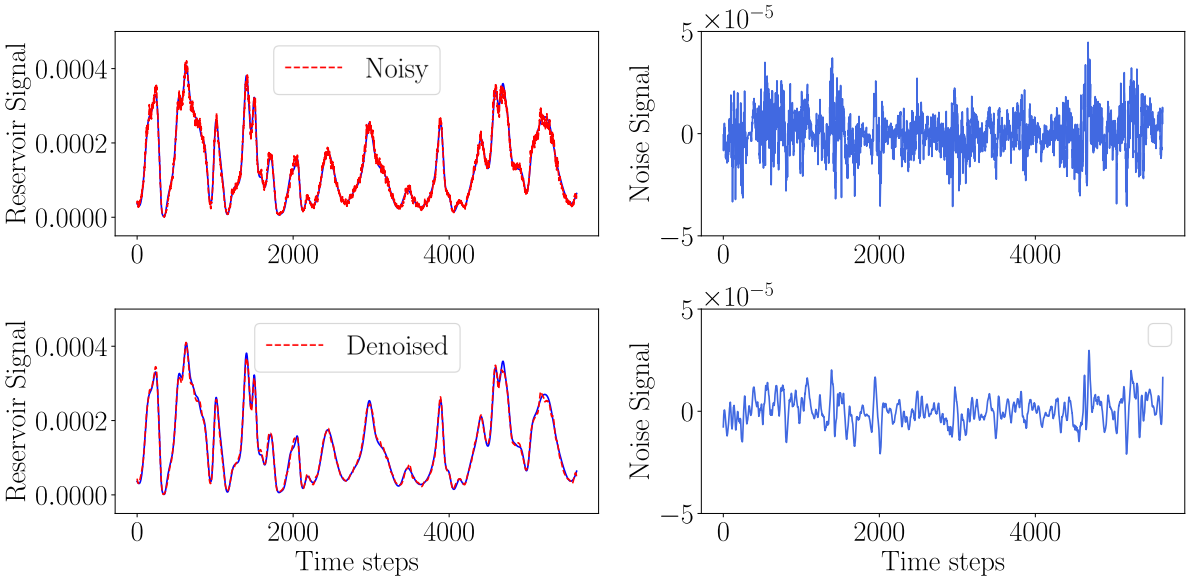


Figure 5.4: MFE system trained on a 10-qubit system results in $N_r = 1024$ reservoir activation signals each of length $N_{tr} = 5800$ time steps. A single randomly chosen reservoir activation signal is shown with 50,000 shots at each timestep for RF-QRC are used for training. Top, visualisation of the noisy reservoir activation signal (left) plotted on the ideal (noise-free) signal, and the noise component - the difference between the two signals (right). Bottom, visualisation of the denoised reservoir activation signal (left) plotted on the ideal (noise-free) signal, and the noise component, which is the difference between the two signals (right).

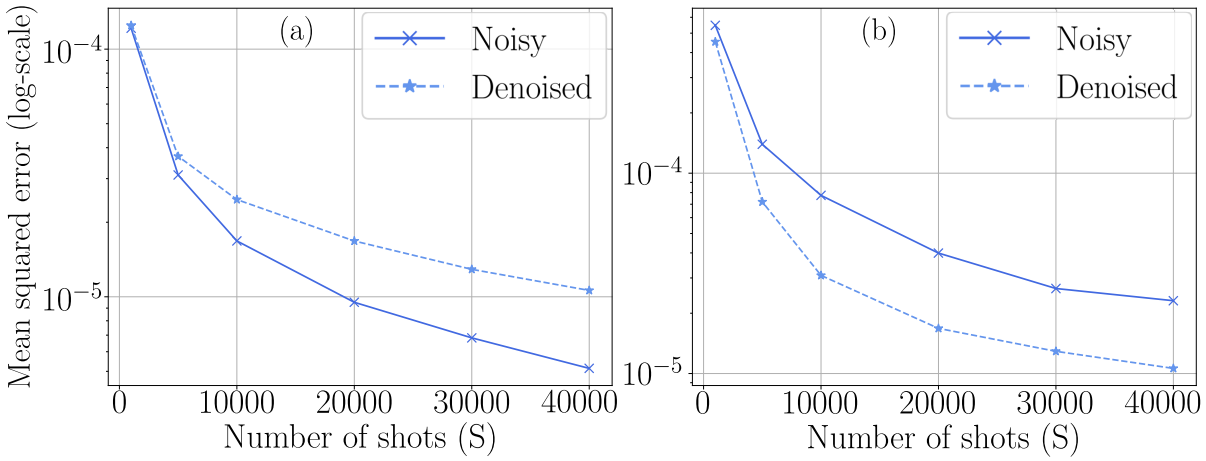


Figure 5.5: Time series prediction of the Lorenz 63 model using RF-QRC. Averaged mean-squared errors of the trajectory are shown as functions of the number of measurements S for noisy and SVD denoised protocols. a) Comparison of noisy and denoised figures with an original reservoir size of $N_r = 128$. b) Comparison of noisy and denoised figures with a reduced reservoir size of $N_r = 60$.

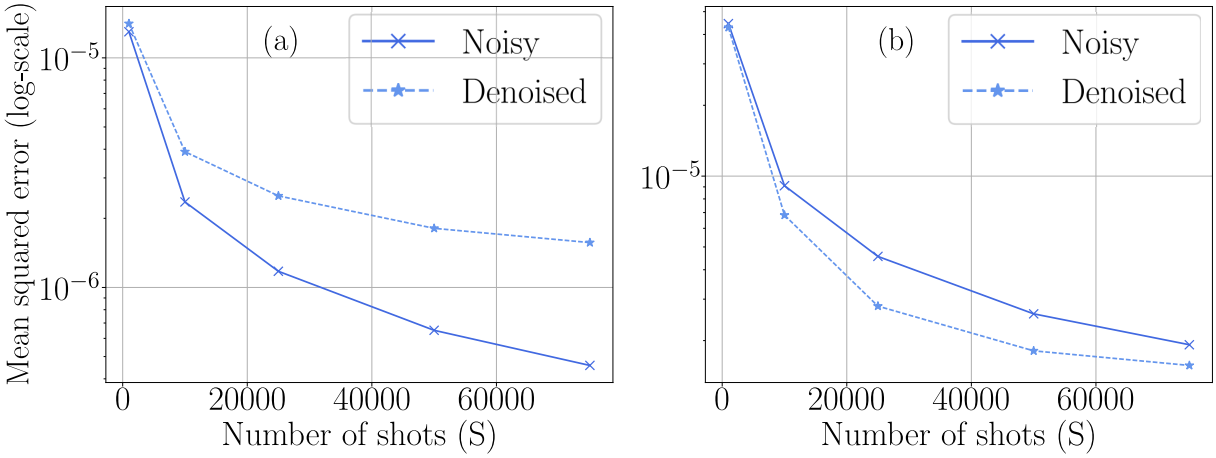


Figure 5.6: Time series prediction of the MFE model using RF-QRC. Averaged mean-squared errors of the trajectory are shown as functions of the number of measurements S for noisy and SVD denoised protocols. a) Comparison of noisy and denoised figures with an original reservoir size of $N_r = 1024$. b) Comparison of noisy and denoised figures with a reduced reservoir size of $N_r = 500$.

the reduced dimensionality of the reservoir state matrix is required to reduce computational resources before post-processing it for training.

In Fig. 5.5, we show the results of the training error for a 7-qubit ($N_r = 128$) reservoir size trained on a Lorenz-63 time series with finite quantum reservoir sampling. The results indicate that truncating half of the singular values, as shown in (Fig. 5.5, a), increases the mean-squared error compared to the original noisy reservoir matrix. This suggests that, while truncation removes noise, it may also discard important activation signals necessary for functional approximation. In contrast, (Fig. 5.5, b) demonstrates that the low-rank approximation improves training error compared to a reduced noisy matrix of the same size, making it particularly effective for dimensionality reduction of the feature matrix before training. We have found similar results for the MFE model in Fig. 5.6.

Despite the noise suppression observed in the reduced-dimensionality case, the SVD approach is generally not computationally feasible for large quantum reservoirs. The number of reservoir activation signals (N_r) in quantum reservoir computing scales exponentially with the number of qubits. Consequently, applying singular value decomposition (SVD) to the noisy reservoir activation matrix becomes computationally expensive, with a complexity of $O\big(\min(N_{tr}N_r^2,\ N_{tr}^2N_r)\big)$ where N_{tr} denotes the number of training samples. This makes classical SVD-based denoising challenging for high-dimensional reservoirs. In Sec. 5.2.2, we instead present a scalable denoising method based on signal filtering, which is more suitable for larger reservoir sizes.

5.2.2 Noise suppression using signal filtering

Using SVD for denoising requires the knowledge of the complete reservoir matrix $\boldsymbol{R}$, the dimension of which scales exponentially with the number of qubits, which limits the application of SVD analysis for RF-QRC. In this section, we propose a method of suppressing noise from the reservoir activation states by applying a denoising low-pass filter to each reservoir activation state.

We emphasise that in the case of RF-QRC and the absence of recurrence, reservoir activation states are only driven by the input time series, which are known *a priori*, as opposed to conventional quantum reservoirs with recurrence, which require the reservoir activation states from the previous time step. Thus, we can employ multiple quantum systems in parallel to generate the reservoir activation states at each time step (Fig. 5.10). Later on, we can concatenate the expectation of the estimates for each eigenbasis to form a reservoir activation signal. This signal is the noisy estimate of our reservoir signal, in which we remove the high frequencies by applying a *moving average polynomial regression* method (Schafer 2011). In principle, one could also apply a physical filter to the noisy signal estimates (Khan, Kaufman, et al. 2023); in this work, we employ digital filtering by post-processing quantum measurements on a classical computer. Specifically, we have used a *Savitsky-Golay* (Schafer 2011) type of moving-average filter and applied it to each reservoir activation state. This type of signal filtering needs the window size and the order of the polynomial to be prescribed. On the one hand, in an open loop (training), the maximum length of the window is equal to the length of the training timeseries, and there is no constraint on the minimum length. On the other hand, in a closed loop (prediction), we need to evolve the noisy reservoir as long as the moving-average window length is before denoising. Therefore, the moving-average window is a hyperparameter that requires tuning. By performing a grid search for the polynomial degree in the range $\{2,6\}$ and the window length in the range $\{10,60\}$ with a stepsize of 10, we select a 3^{rd} polynomial

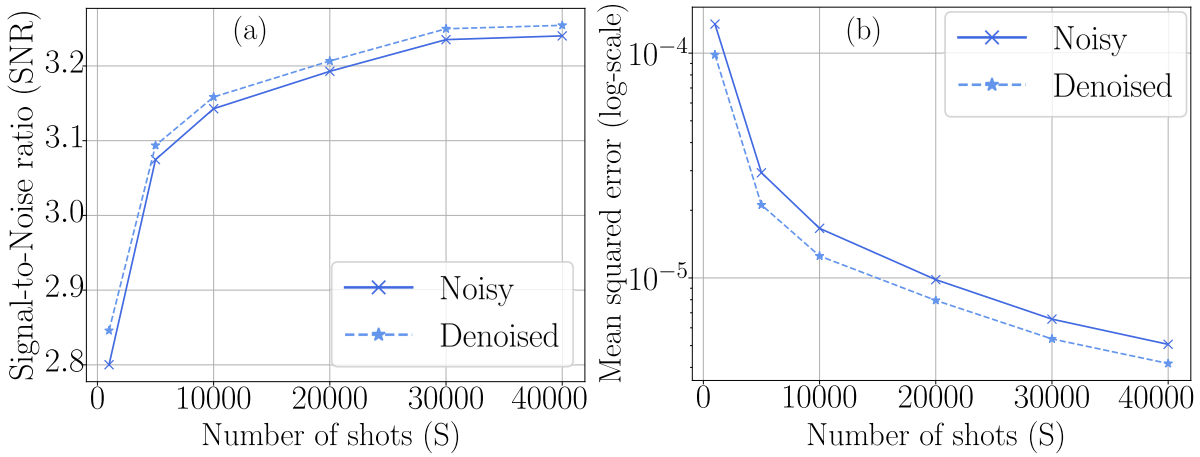


Figure 5.7: Comparison of (a) signal-to-noise ratios and (b) mean-squared errors of noisy and denoised RF-QRC trained on a time series of the Lorenz 63 system. Reservoir size N_r = 128 represented by 7 qubits.

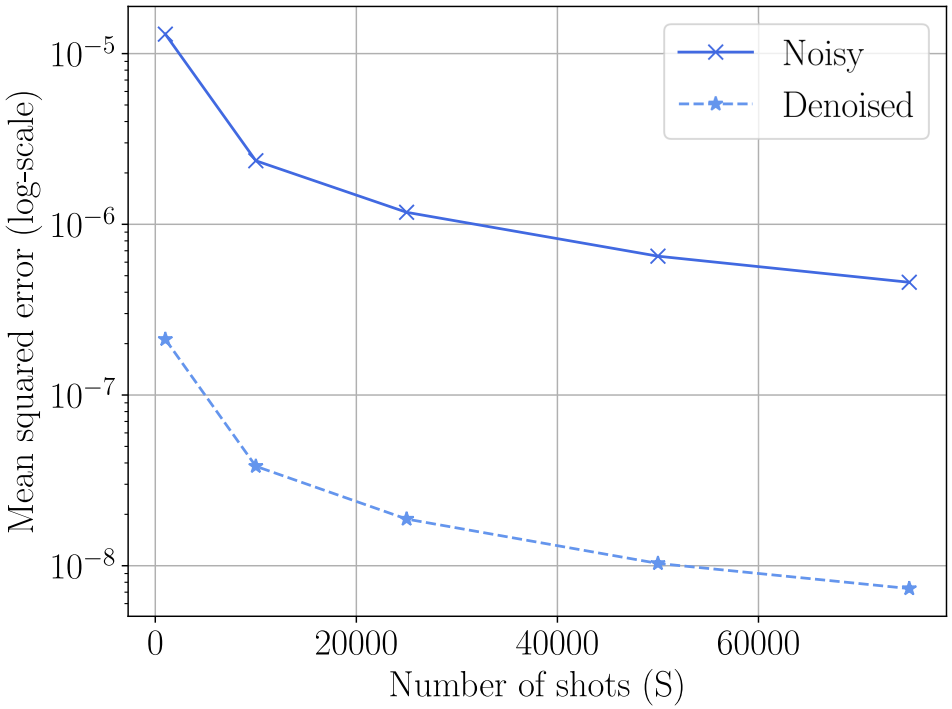


Figure 5.8: MFE time series analysis with RF-QRC, using 10-qubit systems. Mean-squared error for noisy and denoised activation states with signal filtering according to Sec. 5.2.2.

for a window length of 20 and a 4^{th} order polynomial with a window length of 30 for the Lorenz-63 and MFE systems, respectively.

In Fig. 5.7, for a time series of the Lorenz-63 system, we present the SNR and mean-squared training error for noisy and denoised reservoir activation states using polynomial regression. By contrast to denoising based on SVD, using this method for suppressing noise always results in a lower mean-squared error for both cases of reduced and complete reservoir activation state matrix representations. The results in (Fig. 5.7,(b)) show that denoising leads to a lower mean-squared error for various numbers of shots. The lower mean-squared error, therefore, gives an improved SNR ratio for the reservoir activation signals. In

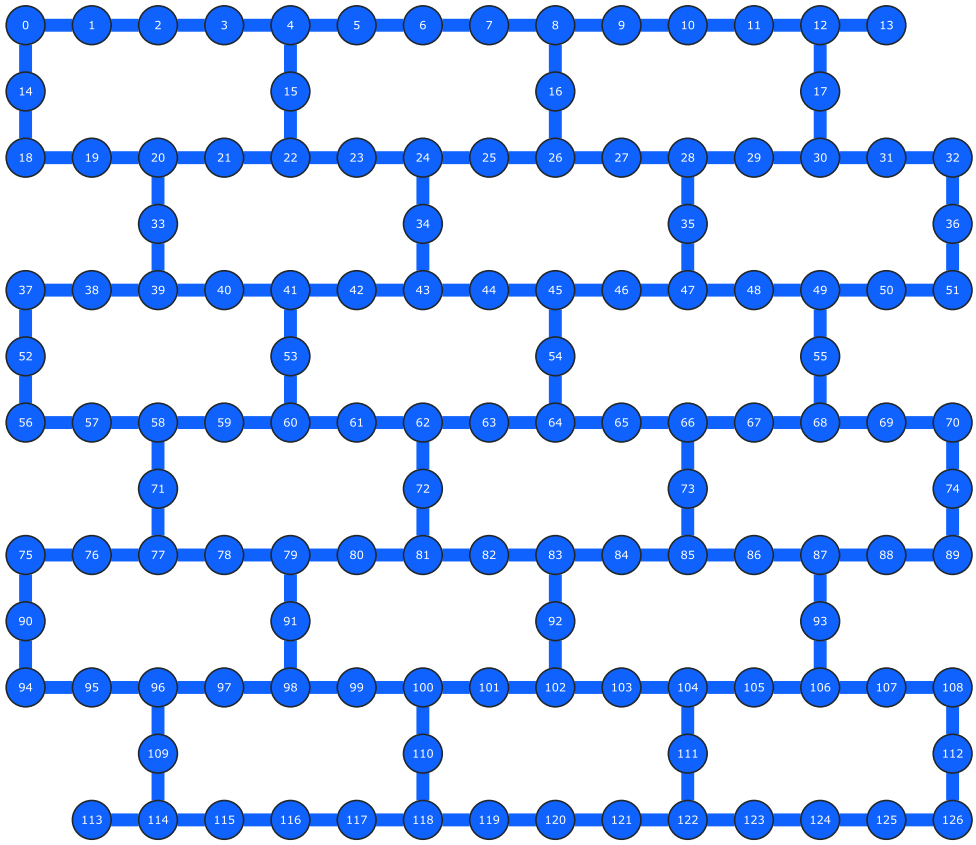


Figure 5.9: Connectivity and qubit map of the ibmq_kyoto 127 qubit device. 10 of the qubits with the lowest readout assignment error are used as a reservoir for RF-QRC of the MFE model.

Fig. 5.8, similar results are shown for the MFE model, indicating that denoising reduces the training error significantly. For the computational time, this method also requires the application of a denoising filter to each reservoir activation signal (N_r), which scales exponentially with the number of qubits. However, by comparison with SVD, denoising methods that involve individual signal smoothing are parallelizable classically for each reservoir signal, offering an efficient choice for noise suppression.

5.3 Hardware implementation

To demonstrate the feasibility of training RF-QRC on quantum hardware and test the denoising, we train the turbulent chaotic shear flow model on the IBM Quantum 127-qubit ibmq_kyoto device (2024). The qubit connectivity map of the hardware backend is shown in Fig. 5.9. From the 127-qubit system, we train the MFE model on a 10-qubit space that corresponds to a reservoir size of $N_r = 1024$.

Before proceeding with the training of RF-QRC, we simplified the ansatz from fully connected layers to linear entangling layers. This simplification is necessary because the device used has limited connectivity. Our simplified RF-QRC contains a layer of Hadamard gates applied on each qubit, followed by R_y rotation gates, parameterised by the input time series signal (9 parameters for MFE). The circuit is then followed by a linear entangling layer of CNOT gates entangling all 10 qubits. This first feature map is applied twice to enrich reservoir dynamics. The second feature map differs from the first in that the rotation gates R_y are now parameterised by random rotation angles $\boldsymbol{V}(\alpha)$ uniformly sampled from the interval $[0, 4\pi]$,

Table 5.2: Training loss comparison on emulator and backend trained networks, with and without denoising

	Noisy	Denoised
Emulator	1.24×10^{-7}	3.28×10^{-9}
Backend	1.10×10^{-6}	1.16×10^{-8}

to introduce randomisation in the reservoir. These random rotation gates are sampled once and are kept constant throughout the training.

The quantum circuit is then transpiled by the *Qiskit* runtime transpiler to achieve an optimal map of the quantum circuit to the physical qubits of the ibmq_kyoto backend. Finally, dynamic decoupling (Qiskit contributors 2023) to the idle qubits is applied to mitigate decoherence. No additional error mitigation strategies were applied.

5.3.1 Parallel training and denoising on multiple QPUs

The MFE input time series signal parameters are mapped onto the pre-processed parameterised quantum circuit. The length of the time series signal is 1200 time steps. The training procedure for RF-QRC is illustrated in Fig. 5.10. Each quantum circuit is executed in parallel, independent of the other quantum circuits, because of the lack of recurrence inside the circuit and the measurement in the computational basis is performed. For each data point, 10^4 shots are sampled, and the results are concatenated classically, followed by leaky integration and denoising techniques as outlined in Sec. 5.2.

Results in Fig. 5.11 indicate that denoising helps to improve the training accuracy for both the emulation of the state vector propagation (only subject to shot noise) and the training on the ibmq_kyoto quantum processor (subject to shot noise *and* hardware noise) for increasing reservoir states. Therefore, even in the presence of correlated hardware noises, denoising improves the performance of the model. Other results in Fig. 5.12 demonstrate that hardware noise also has a significant impact on the performance, in addition to the impact of shot noise. In particular, the model trained on the quantum processor requires more reservoir states (i.e. a larger active space of the reservoir) for the same training accuracy when compared to a model with only shot noise. Similar effects are observed in both denoised models. However, denoising can be instrumental in mitigating those errors to reduce this gap between hardware and emulated reservoirs. This suggests that denoising suppresses shot noise, and potentially other types of noise as well. The averaged training loss for each configuration with all included reservoir states ($N_r = 1024$) is listed in Tab. 5.2.

Finally, we present the results of the reconstructed time series of kinetic energy from a trained 9-mode MFE in Fig. 5.13. We do so for both emulated models with shot noise and the trained RF-QRC on the backend, including hardware noise. The reconstructed time series demonstrates that without denoising, the resulting fit for the trained signal suffers more from finite sampling and hardware noise. However, the training of RF-QRC coupled with denoising helps in reducing the shot noise as well as various hardware noises to have a better fit for optimal training.

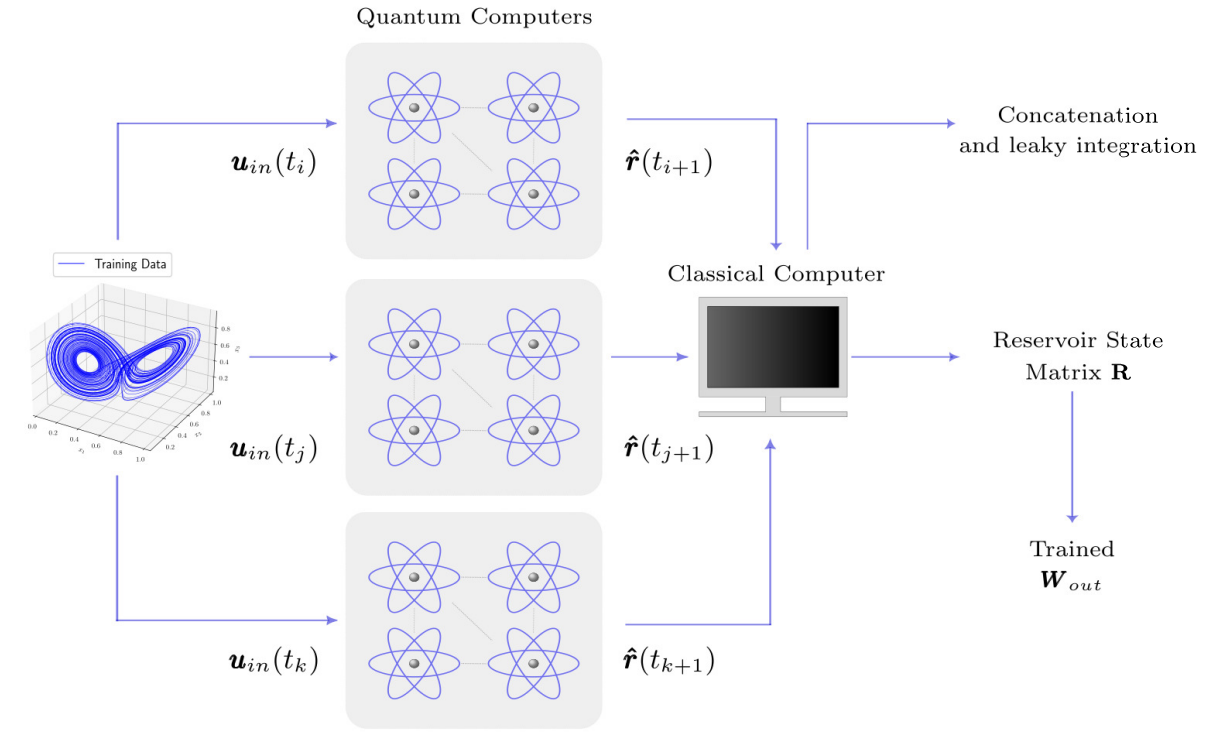


Figure 5.10: Recurrence-free quantum reservoir computing parallel training. Input data $\boldsymbol{u}_{in}(t)$ is divided into shorter signals and passed through multiple parallel quantum computers. The obtained reservoir activation signals $\hat{\boldsymbol{r}}_{in}(t)$ associated with each input time series signal are concatenated classically, where leaky integration and denoising are applied. The combined reservoir state matrix $\boldsymbol{R}$ is then used for training using ridge regression.

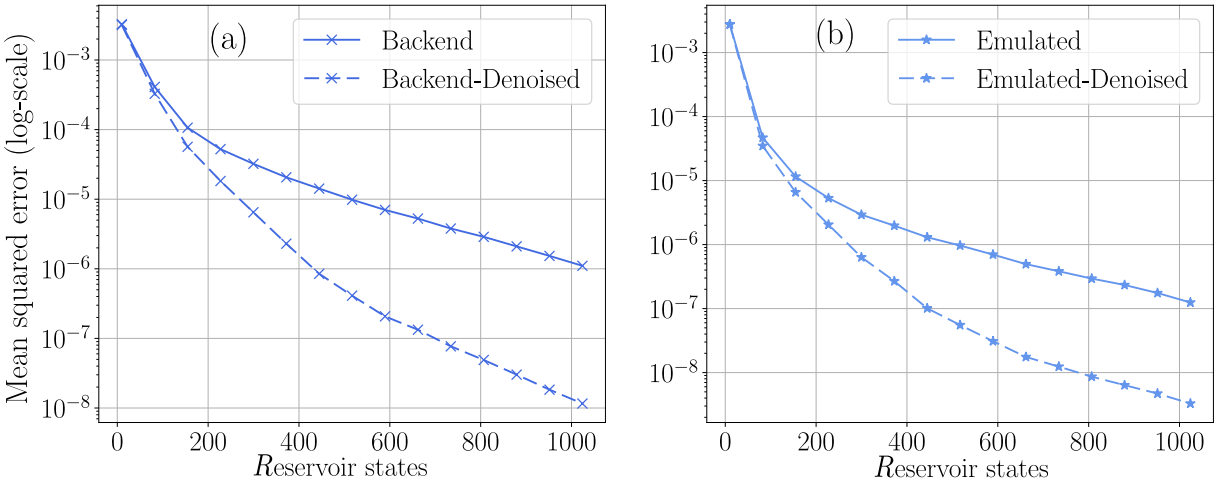


Figure 5.11: MFE system. Training mean-squared error comparison with noisy and denoised states for both emulated and quantum processor-based RF-QRC.

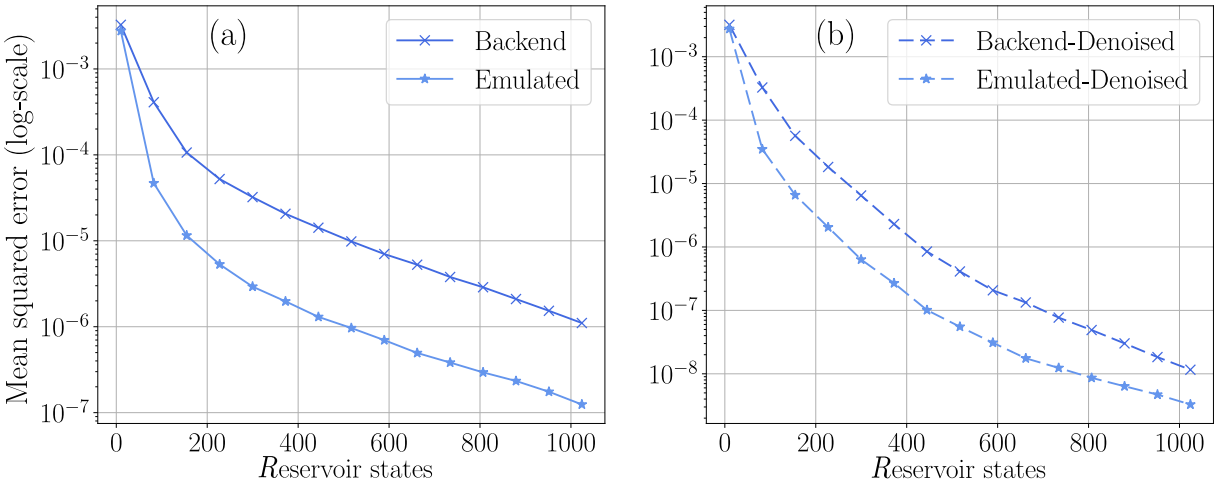


Figure 5.12: MFE system, training mean-squared error comparison with emulated and quantum processor results.

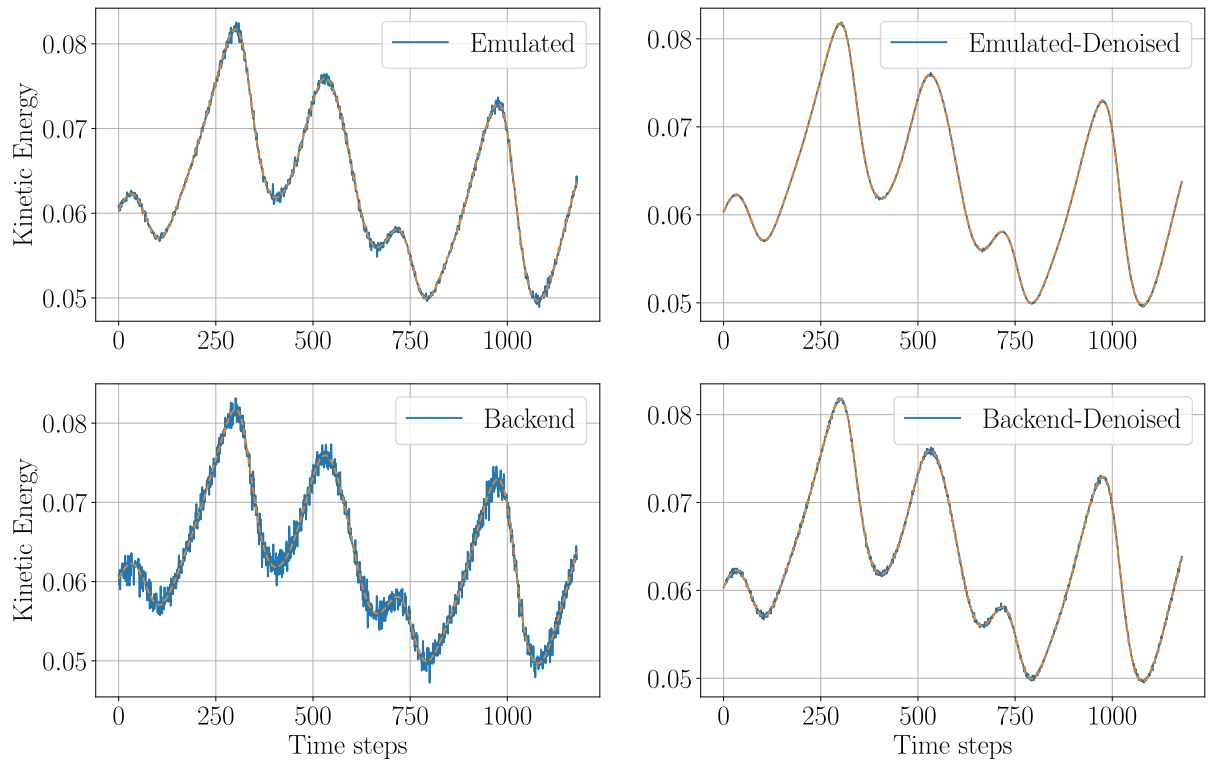


Figure 5.13: MFE system, reconstructed kinetic-energy comparison with emulated and backend results. The top panel presents the results of emulated noisy and denoised reservoir activation states. The bottom panel presents the results of hardware-trained noisy and denoised reservoir activation states.

5.4 Conclusion

In this chapter, we investigated the role of finite-sampling and hardware noise in quantum reservoir computing for chaotic time-series forecasting. While quantum reservoir computers show strong performance under ideal, noise-free assumptions, practical implementations are inherently limited by sampling noise and environmental effects on quantum hardware.

We showed that the recurrence-free quantum reservoir computer (RF-QRC), based on leaky-integrated neurons, naturally provides temporal memory and acts as an exponential smoother of noisy reservoir states. This structure makes RF-QRC both scalable and more robust to noise compared to recurrent QRC architectures.

We analysed the impact of finite-sampling noise on different quantum reservoir architectures and showed that RF-QRC exhibits higher signal-to-noise ratios due to modelling the noise as temporally uncorrelated. To further mitigate noise, we proposed two methods based on singular value decomposition (SVD) and signal filtering, which effectively suppress noise in reservoir activations. Our results show that denoising improves training performance, leading to lower mean-squared errors and further increases in signal-to-noise ratios.

Finally, we demonstrated the applicability of RF-QRC on quantum hardware by deploying the model across multiple parallel QPUs. The results indicate that denoising techniques can suppress both sampling and hardware noise. These methods can be extended to other quantum reservoir computing architectures, including recurrent QRCs and quantum extreme learning machines.

In summary, this chapter shows that sampling noise, which is inherent to quantum mechanics and will also be present in fault-tolerant quantum computers (FTQCs), can be effectively suppressed through appropriate architectural choices and classical post-processing techniques. This demonstrates that hybrid quantum-classical approaches enable the practical deployment of quantum reservoir computing on near-term quantum hardware for chaotic system prediction. In Chap. 6, we extend this analysis to include environmental and incoherent noise sources, and analyse the effect of different noise types from a generalised synchronisation (GS) perspective.

Chapter
6

Noise as a resource for robust quantum reservoir design

In this chapter, we investigate how noise itself can be exploited as a resource for designing robust quantum reservoir computers within the framework of generalised synchronisation (GS).

In Sec. 6.1, we perform a systematic hyperparameter study of Lyapunov exponents (LEs) and conditional Lyapunov exponents (CLEs). This establishes a connection between CLEs, GS, and the learnability of the Lyapunov spectrum, and provides a practical reservoir design principle based on the GS criteria summarised in Tab. 2.1 for both the Lorenz–63 and Lorenz–96 systems. We further compare the robustness of recurrent and recurrence-free QRC architectures through the computation of LEs and CLEs for the Lorenz–63 system. In Sec. 6.2, we investigate the role of noise in shaping the dynamical properties of QRCs and show that, under appropriate conditions, noise can improve the robustness and stability of the reservoir dynamics rather than simply degrading performance.

This chapter is based on the contributions from (O. Ahmed, Tennie, and Magri 2025b) and we analyse how the conditional Lyapunov exponents depend on the reservoir hyperparameters and noise strength, and use these quantities to characterise the learnability, stability, and robustness of the reservoir dynamics.

6.1 Designing robust quantum reservoir computers

In Chap. 4, we have shown that RF-QRCs can accurately forecast the chaotic dynamics whilst inferring the invariant physical properties of the physical system. In this section, we provide practical guidelines for designing robust quantum reservoirs by exploiting the connection between quantum reservoir computers and GS made in Chap. 3.

We compute the conditional Lyapunov exponents (CLEs) in the open-loop (training) phase to assess the GS criteria outlined (for brevity, we rewrite the GS and differentiability criteria from Tab. 2.1 below). We do that for different leak rates, ϵ, whilst keeping the quantum reservoir size fixed as in Tab. 4.4. The core

Table 6.1: Criteria for generalised synchronisation and differentiability. The largest conditional Lyapunov exponent of the response system is denoted $\max(\lambda_{\text{CLE}})$, and the most negative Lyapunov exponent of the driving system is denoted λ^*. All Lyapunov exponents are expressed in continuous-time units.

	Condition	Property
i.	$\max(\lambda_{\text{CLE}}) \geq 0$	GS does not occur.
ii.	$\lambda^* < \max(\lambda_{\text{CLE}}) < 0$	GS occurs; ϕ is not differentiable.
iii.	$\max(\lambda_{\text{CLE}}) < \lambda^* < 0$	GS occurs; ϕ is continuously differentiable. Differentiable GS (DGS).

idea is to analyse the CLEs in the training phase to establish whether the quantum reservoir computers are robust (or not). In the training phase, the Jacobian is provided by Eq. 3.22. The pseudo-algorithm to compute the CLEs is shown in Algorithm 2.

6.1.1 Lorenz-63 system

We start by analysing the Lorenz-63 system with RF-QRCs and compute LEs and CLEs for various leak rates. Further hyperparameters and model details are presented in Tab. 4.4.

Algorithm 2: An algorithm to compute CLEs with RF-QRC

Choose k leak rates ϵ in range [0,1]
for *each* ϵ **do**
  $\boldsymbol{W}_S \leftarrow random \in \mathbb{R}^{N_r \times D}$; /* Initialize D GSVs */
  $\boldsymbol{Q}, \boldsymbol{R}_S \leftarrow QR(\boldsymbol{W}_S)$; /* Orthonormalize GSVs */
  $\boldsymbol{W}_S \leftarrow \boldsymbol{Q} \in \mathbb{R}^{N_r \times D}$
  $N_{QR} \leftarrow N_{train} - N_w$; /* Number of retained QR decompositions */
  Save the time series of $\boldsymbol{R}_S$ *for CLE calculation*
  Initialize $\tilde{\boldsymbol{R}}_S \leftarrow \boldsymbol{0} \in \mathbb{R}^{D \times D \times N_{QR}}$
  $\boldsymbol{J}_{CLE} \leftarrow (1-\epsilon)\boldsymbol{I}$; /* Conditional Jacobian of RF-QRC as in Eq. 3.24 */
  Evolve the hidden state and GSVs simultaneously.
  Skip the initial transients for N_w *time steps.*
  $n \leftarrow 0$; /* Increment the number of QR decompositions */
  for $i = 0 : N_{train} - 1$ **do**
    $\boldsymbol{r}(t_{i+1}) = f(\boldsymbol{r}(t_i))$; /* QRC state update as in Eq. 3.12 */
    $\boldsymbol{W}_S \leftarrow \boldsymbol{J}_{CLE}\boldsymbol{W}_S$; /* The variational equation */
    $\boldsymbol{Q}, \boldsymbol{R}_S \leftarrow QR(\boldsymbol{W}_S)$; /* QR at every time step */
    $\boldsymbol{W}_S \leftarrow \boldsymbol{Q}$
    if $i \geq N_w$ **then**
      $\boldsymbol{\Lambda}[:, n] \leftarrow \log(\text{diag}[\boldsymbol{R}_S])/dt$; /* Save finite-time CLEs */
      $\tilde{\boldsymbol{R}}_S[:, :, n] \leftarrow \boldsymbol{R}_S$; /* Save $\boldsymbol{R}_S$ */
      $n \leftarrow n + 1$;
  $\lambda_j = \frac{1}{N_{QR}} \sum_{i=0}^{N_{QR}-1} \Lambda[j, i]$; /* The jth Conditional Lyapunov exponent */
  $\lambda_{CLE} = \max(\lambda_j)$; /* The largest Conditional Lyapunov exponent */

First, Fig. 6.1 (a) shows the LEs inferred in closed loop for the Lorenz-63 system for ϵ in the range $\{0,1\}$. For $\epsilon > 0.2$, the spectrum is accurately inferred. For $\epsilon < 0.2$, only the positive and zero LEs are captured correctly. This is because, when the leak rate is low, the reservoir dynamics are slow, which means that the RF-QRC struggles to capture fast-decaying perturbations associated with the negative LE.

In Fig. 6.1 (b), we show the maximum CLEs and LEs for the same ϵ as in Fig. 6.1 (a). The magnitude of the maximum CLE grows with the leak rate ϵ. For slower reservoir dynamics, when the maximum CLE is larger than the most negative LE of the driving system (max $\lambda_{CLE} > \lambda^*$, i.e., condition (ii) in Table 6.1), the reservoir is unable to capture the negative LEs accurately. This leads to an incorrect increase in the Kaplan-Yorke dimension (J. D. Hart 2024). Increasing the user-defined leak rate enables the tuning of the maximum CLE, thereby satisfying condition (iii) of Tab. 2.1, which provides an accurate prediction of the entire spectrum.

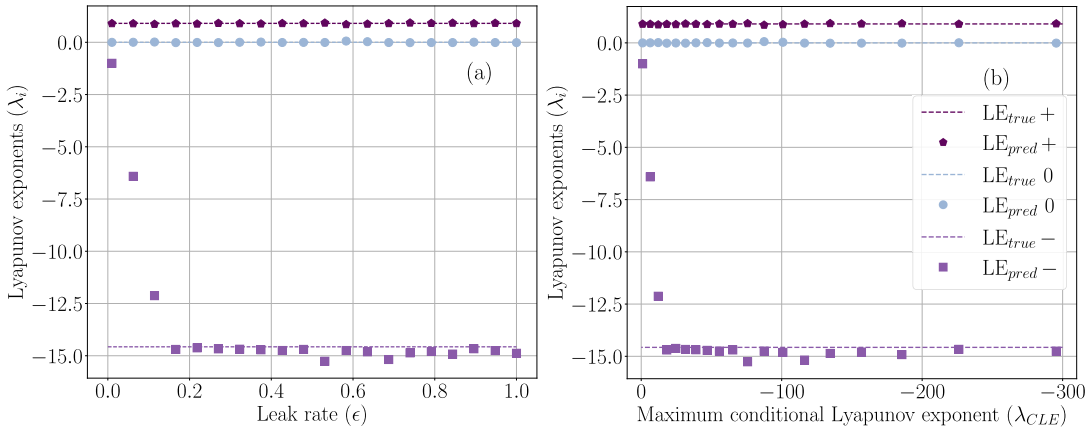


Figure 6.1: Lorenz-63 in closed-loop autonomous prediction phase. (a) Lyapunov exponents vs leak rate. (b) Lyapunov exponents vs maximum conditional Lyapunov exponents.

Recurrent vs recurrence-free quantum reservoir stability

For comparison, we perform similar studies shown in Sec. 6.1 for a recurrent QRC framework. The motivation for this work is to analyse how the addition of a recurrent layer affects the learnability of QRCs.

In Fig. 6.2 we show the maximum CLEs as functions of the leak rate for both recurrent and recurrence-free quantum reservoir computers (QRCs and RF-QRCs, respectively). The conditional Lyapunov exponents (CLEs) remain negative for all leak rates, which means that the reservoir dynamics are stable. The colour of each point in Fig. 6.2 quantifies the short-term prediction of the reservoirs with the valid prediction time (Vlachas, Pathak, et al. 2020). The results in Fig. 6.2 show that the RF-QRC performance is robust and predictable over a larger range of leak rates than that of QRCs.

In detail, in RF-QRCs, there exists an injective (monotonic) relationship between the leak rate and maximum CLE. The network approaches the behaviour of an Extreme Learning Machine (ELM) (Xiong et al. 2023) when $\epsilon = 1$, with the magnitude of maximum CLE approaching infinity (see Eqs. 3.24 and 2.22). A very large magnitude of the maximum CLE makes the reservoir unstable and sensitive to noise. On the other hand, in QRCs, the relationship between ϵ and the maximum CLE is not injective (not monotonic). The main reason for this is the addition of recurrent states in the reservoir update, which changes the spectral radius of the reservoir, and now the *effective* spectral radius governs the rate of dissipation Jaeger et al. 2007. QRCs are stable for intermediate ϵ values ($0.25 < \epsilon < 0.6$), whereas the RF-QRC is stable for a larger range, $\epsilon > 0.2$. In these regions, QRCs and RF-QRCs can be used for both short-term predictions and inference of invariant properties from chaotic time series.

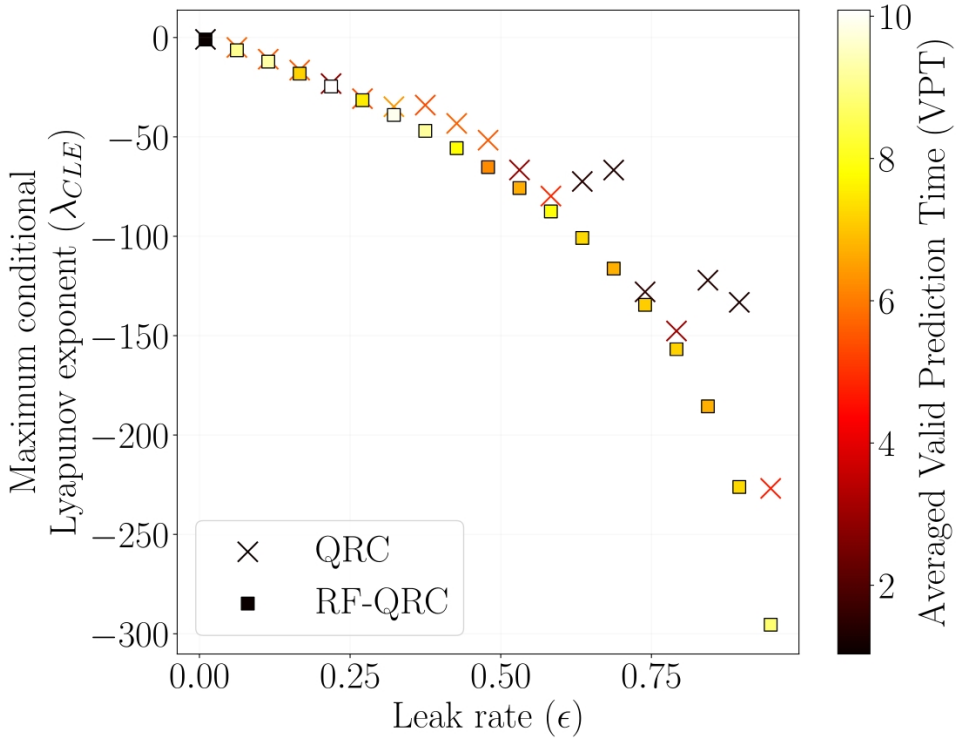


Figure 6.2: Maximum conditional Lyapunov exponents of quantum reservoir computers with recurrence (QRCs) and recurrence-free (RF-QRCs). Each point value is the ensemble mean over ten autonomous predictions from different initial points.

In Fig. 6.3, we compute the autonomous LE predictions of the Lorenz-63 system for various ϵ values, uniformly distributed within $[0, 1]$. Similarly to the results of Fig. 6.1, for small ϵ, QRCs can accurately capture the positive λ_1 and neutral λ_2 Lyapunov exponents, but cannot capture the negative λ^* Lyapunov exponent. Furthermore, in contrast to RF-QRC, which remains conditionally stable for large ϵ values, the performance of QRCs when $\epsilon > 0.6$ ($|\lambda_{CLE}| > 100$) becomes inaccurate. The addition of recurrent connections enhances the spectral radius and reservoir memory, and when the artificial dissipation with a leak rate ϵ is not strong enough, the QRCs become unstable. This analysis further informs the design of future quantum reservoir computers by showing that a very high expressivity without dissipation can result in an unstable reservoir. To address this issue, our proposed leaky integrated RF-QRC model or combining other QRCs with classical leaky-integrators to tune dissipation. Thereby, helping in the design of efficient quantum reservoir computers.

6.1.2 Lorenz-96 system

We extend the analysis about designing practical quantum reservoirs, shown in Section 6.1 for the Lorenz-63 system, to the 10-dimensional Lorenz-96 system. Specifically, we analyse the influence of the leak rate hyperparameter ϵ and the magnitude of the maximum conditional Lyapunov exponent on the ability of the reservoir to infer Lyapunov exponents on the unseen test dataset. The training procedure is the same as described in Section 4.3.2, and other model hyperparameters are shown in Tab. 4.4. In Fig. 6.4a, we vary the leak rate from $\epsilon = 0.001$ to $\epsilon = 1.0$ and compute the inferred Lyapunov exponents of the model. In Fig. 6.4 b, the colour bar indicates the maximum λ_{CLE} value associated with the corresponding leak rate in Fig. 6.4 a. These results are in line with the findings of the previous analysis from Section 6.1.

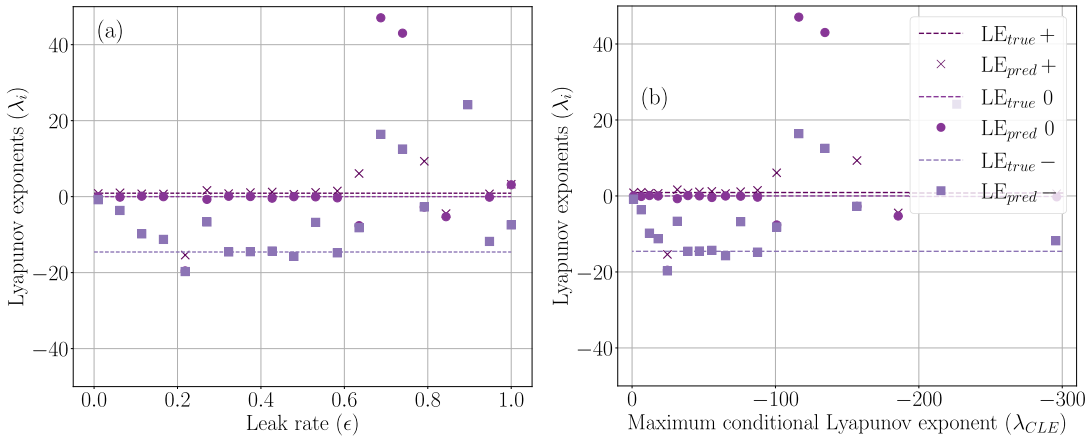


Figure 6.3: Lorenz-63 system autonomous prediction of Lyapunov spectrum with QRCs (a) Dynamical system's Lyapunov exponents vs Maximum conditional Lyapunov exponents of response (b) Dynamical system's Lyapunov exponents vs Leak rate. In both plots, three points on each vertical line correspond to the three inferred Lyapunov exponents. For very small leak rates, reservoir dynamics are slower, and negative LE is not inferred correctly. QRC performance is optimum for intermediate ϵ between $\{0.35,0.5\}$ but performs inaccurately with $\epsilon > 0.6$ (as opposed to RF-QRC in Fig. 6.1).

The maximum CLEs and their effects on the inference task are shown in Fig. 6.4 b. When the reservoir dynamics are faster and the leak rate is closer to 1, the reservoir over-predicts the positive and under-predicts the negative Lyapunov exponents. Because the reservoir dynamics are faster, the reservoir is also very sensitive to noise in this region. Therefore, depending on the learning tasks and form of noise, the rate of reservoir dynamics must be carefully tuned with the hyperparameter leak rate.

6.2 Influence of noise

In this section, we analyse the effect of different types of noise on the performance of RF-QRCs to infer the stability properties of the Lorenz-63 system. We analyse the role of finite sampling noise and also other noise types that promote dissipation (loss of information) in the reservoir update. We analyse incoherent errors due to the interaction of the quantum hardware with its environment, i.e., depolarising noise and amplitude damping channels (Nielsen and Chuang 2011). Previous works have proposed dissipation from noise as a resource to improve quantum reservoir computers (Sannia et al. 2022; Fry et al. 2023; Domingo, Carlo, and Borondo 2023). We show that tuning noise intensity helps satisfy $GS = ESP$.

Table 6.2: Types of noise analysed.

Cause	Type of noise	Parameters	Number of qubits
Finite sampling	Projective measurement	$S = \{1000,5000,10000,$ $25000,50000\}$	$n = 8$
Depolarizing channel	Unital, incoherent	$p = \{0.001,0.01,0.05,0.1\}$	$n = 7$
Amplitude damping	Non-unital, incoherent	$p = \{0.001,0.01,0.05,0.1\}$	$n = 7$

6.2.1 Sampling noise

The theoretical description of finite-sampling noise in QRCs and RF-QRCs was discussed in Chap. 5. The goal of this section is to analyse how sampling noise affects the performance of RF-QRCs in inferring

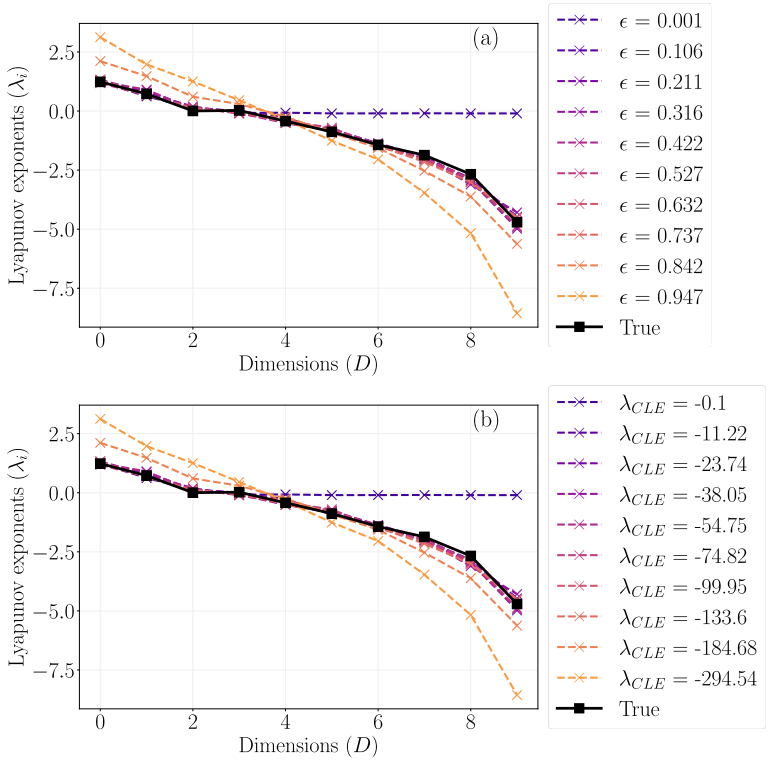


Figure 6.4: Lyapunov spectrum of Lorenz-96 systems with 10 dimensions. Comparison between ground truth and RF-QRC-inferred Lyapunov spectrum for a reservoir size of 10 qubits. (a) Various leak rates (b) Various maximum λ_{CLE} corresponding to leak rate in (a).

Lyapunov exponents for different numbers of measurement shots, S (Table 6.2).

Fig. 6.5 (a) shows the results of an RF-QRC with an 8-qubit reservoir sampled with 1000 shots at each time step during training and prediction. The results are the outcome of averaging the performances over five different random seeds for the $V(\boldsymbol{\alpha})$ unitary. Even for a number of finite samples, RF-QRC is able to accurately infer the LEs of the Lorenz-63 system when the leak rate is low (the memory is high). This means that in the presence of noise, the reservoir dynamics must be slower to infer the LEs (noise promotes dissipation, thereby requiring larger memory). However, reducing the leak rate to values close to zero, or operating near the edge of chaos ($\lambda_{CLE} = 0$ in Table 6.1), results in a loss of generalised synchronisation. In this case, in the presence of noise due to finite sampling, the hyperparameter leak rate should be tuned in the range $\{0.2, 0.3\}$. Even for finite samples (e.g., 1000 in this case), there exists a good range of leak rates for which the RF-QRC infers the stability properties accurately. Finding this optimum range can vary depending on the analysed chaotic system and the sample size. In Fig. 6.5 (b), we perform a parametric study with the number of shots (Table 6.2). The noise intensity influences the performance of the RF-QRC and the range of good hyperparameters ϵ. As expected, by increasing the number of shots, the accuracy increases.

6.2.2 Incoherent noise

A quantum hardware state, represented by a density operator $\bar{\rho}_H$, interacts with the environment, represented by $\bar{\rho}_{env}$, to form a combined system–environment state, which is initially modelled as a product state $\bar{\rho}_H \otimes \bar{\rho}_{env}$. To consider the effect of a unitary transformation, $\mathcal{U}$, acting on the combined system–

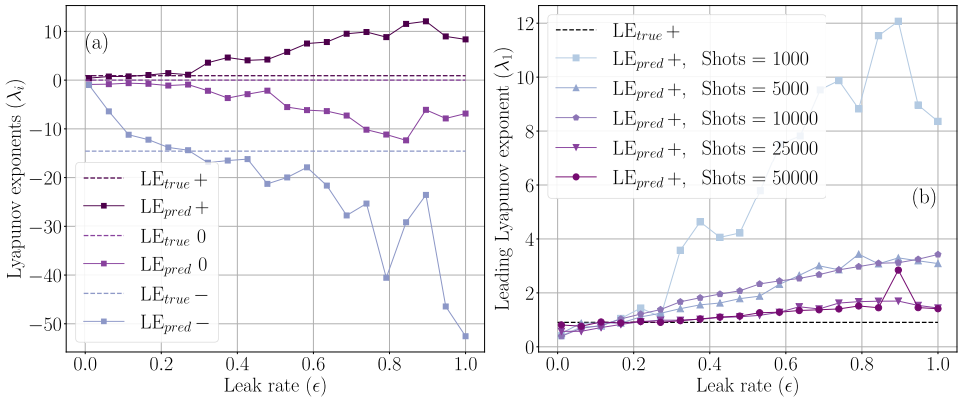


Figure 6.5: Effect of sampling noise on the RF-QRC performance with 8 qubits. (a) Lyapunov spectrum for 1000 shots. (b) Leading Lyapunov exponents for different shots.

environment state, we take a partial trace over the environment to obtain the reduced state of the system (Nielsen and Chuang 2011)

$$\mathcal{N}(\bar{\rho}_H) = \mathrm{tr}_{env}[\mathcal{U}(\bar{\rho}_H \otimes \bar{\rho}_{env})\mathcal{U}^\dagger], \tag{6.1}$$

where $\dagger$ is the conjugate transpose, and $\mathcal{N}$ is a noisy channel, which takes the quantum hardware state $\bar{\rho}_H \rightarrow \mathcal{N}(\bar{\rho}_H)$. Let $|e_m\rangle$ be an orthonormal basis for the (finite-dimensional) state space of the environment, and $\bar{\rho}_{env} = |e_0\rangle\langle e_0|$ be the initial state of the environment (assuming a pure state). Eq. 6.1 in *operator-sum* representation can be written as

$$\mathcal{N}(\bar{\rho}_H) = \sum_m \langle e_m|\mathcal{U}\ [\bar{\rho}_H \otimes |e_0\rangle\langle e_0|]\ \mathcal{U}^\dagger|e_m\rangle, \tag{6.2}$$

$$= \sum_m K_m\, \bar{\rho} K_m^\dagger, \tag{6.3}$$

where $K_m \equiv \langle e_m|\mathcal{U}|e_0\rangle$ is an operator on the state space of the quantum system. This representation is also known as the Kraus representation, in which K_m are the Kraus operators, which fulfil $\sum_m K_m K_m^\dagger = I$ (Nielsen and Chuang 2011). The maximum number of Kraus operators to model incoherent noise and,

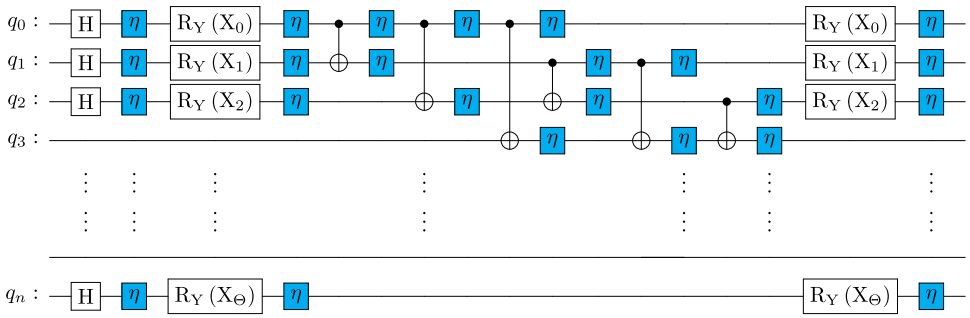


Figure 6.6: Fully-connected quantum circuit for the RF-QRC (O. Ahmed, Tennie, and Magri 2024b) architecture with noise channels η after each gate operation. The number of encoded parameters Θ for RF-QRC is equal to the dimension of the input data ($\Theta = D$) for $\Xi(\boldsymbol{u})$ unitary. When $D > n$, the feature map encoding is applied multiple times to encode all $\boldsymbol{u}$ dimensions. For a random unitary $V(\boldsymbol{u})$, the number of encoded parameters is equal to the number of qubits n, ($\Theta = n$).

consequently, the size of the noisy state, scales quadratically with the size of the Hilbert space $(2^n)^2$. We consider the effect of amplitude damping and depolarising channels (Fig. 6.7). These types of noise have been previously studied in quantum natural reservoirs (Suzuki et al. 2022; Kubota et al. 2023), using dissipation as a resource (Sannia et al. 2022) and with artificial noise channels with tunable noise parameters (Fry et al. 2023). We consider the density operator, $\bar{\rho}_H$, that represents the quantum state of one qubit

$$\bar{\rho}_H = \begin{pmatrix} \bar{\rho}_{00} & \bar{\rho}_{01} \\ \bar{\rho}_{01}^{\dagger} & \bar{\rho}_{11} \end{pmatrix}, \tag{6.4}$$

The amplitude damping channel applied to this state can be modelled with the Kraus operators K_0 and K_1

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, K_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}, \tag{6.5}$$

where $0 \leq p \leq 1$ is the noise intensity. The Kraus operators can be combined with Eq. 6.2 to build the density matrix $\mathcal{N}_{AD}(\bar{\rho}_H)$ (Nielsen and Chuang 2011). Physically, this type of noise represents the asymmetric shrinking of the Bloch sphere, thus, it is referred to as non-unital noise (Fig. 6.7 c). We vary the intensity of the noise channel with p, as shown in Table 6.2, for a 7-qubit quantum reservoir system. The visualisation of the noise channel in the quantum circuit is shown in Fig. 6.6 with η being the noise channel.

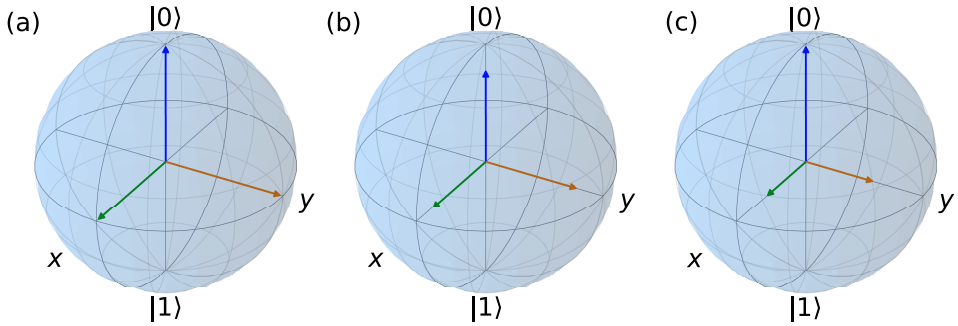


Figure 6.7: Bloch spheres of the quantum state of a qubit for (a) noiseless unitary evolution, (b) uniform contraction of the state due to depolarising noise, and (c) non-uniform contraction of the state due to amplitude damping noise.

The second type of noise analysed is the depolarizing channel, which for a single qubit can be represented by [a]

$$\mathcal{N}_{DP}(\bar{\rho}_H) = (1-p)\,\bar{\rho}_H + p\,\frac{I}{2}. \tag{6.6}$$

The depolarising noise shrinks the Bloch sphere uniformly, i.e., it is a unital type of incoherent noise (Fig. 6.7 b). With $p = 1$, the channel returns the maximally mixed state for any input state $\bar{\rho}_H$, which corresponds to the complete contraction of the Bloch sphere to a single point given by $\frac{I}{2}$. We vary

[a] The associated Kraus representation could also be derived with 4 Kraus operator $\{K_0, K_1, K_2, K_3\}$ and corresponding Pauli matrices $\{I, X, Y, Z\}$ as shown in Preskill 1998b.

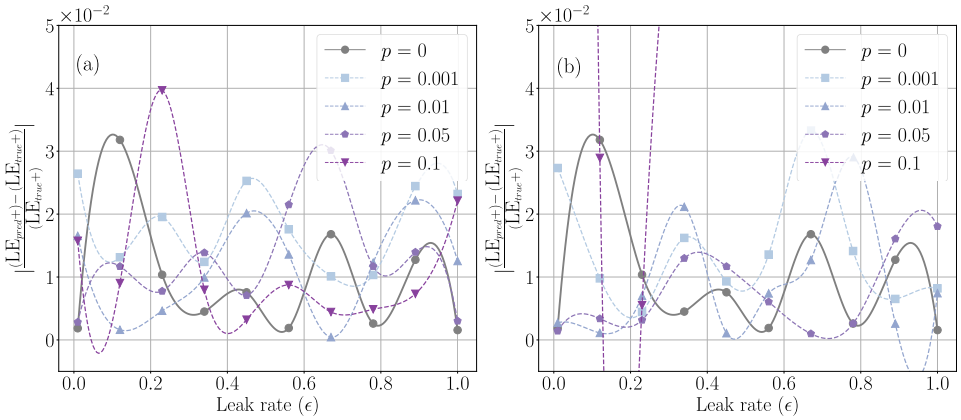


Figure 6.8: Error on the leading Lyapunov exponent with an RF-QRC with 7 qubits. Each result is the ensemble average over 3 different realisations of random seeds. (a) Effect of amplitude-damping noise. (b) Effect of depolarising noise.

the intensity of the noise channel (Table 6.2 and Fig. 6.6). First, Fig. 6.8 (a) shows the impact of the amplitude-damping noise channels on the RF-QRC performance. We show the error between the inferred and target leading Lyapunov exponents (λ_1) for different noise intensities and leak rates. Each data point in Fig. 6.8 is computed by ensemble averaging over three random seeds of the $V(\boldsymbol{\alpha})$ unitary. The case with $p = 0$ corresponds to the ideal noise-free state-vector emulation. We increase the amplitude damping noise intensity from $0.001 - 0.1$ ($0.1 - 10\%$ noise intensity). The largest error in the prediction on λ_1 is ≈ 0.04, which occurs for $p = 0.1$. For other noise intensities, when the reservoir dynamics are slower (small leak rate), the prediction performance improves over the noise-free scenario. This shows that adding amplitude-damping noise promotes dissipation, thereby strengthening GS. The addition of noise in the faster reservoir dynamics region (i.e., large leak rate) does not improve the performance beyond the ideal noise-free case. Fig. 6.8 (b) shows the results on the depolarising noise channel. For $p = 0.1$, the reservoir cannot infer the correct value of λ_1, and the maximum relative error is ≈ 0.2. As in amplitude damping, when the reservoir dynamics are slower (i.e., small leak rate), the performance improves with the addition of depolarising noise channels, i.e., noise makes the RF-QRC more robust and accurate than the ideal noise-free case. In conclusion, noise adds dissipation, which, in turn, promotes GS (and consequently ESP). This can be seen from Eq. 3.18, in which noise ensures a contractive map ($\tilde{\gamma} < 1$).

6.3 Conclusion

We introduced a practical criterion for robustness by examining the relationship between Lyapunov exponents and conditional Lyapunov exponents (CLEs), thereby identifying when the condition $GS = ESP$ holds. Using this approach, we showed that RF-QRCs are conditionally stable by construction, satisfying $GS = ESP$ across a broader range of hyperparameters compared to recurrent QRCs.

We further investigated the role of noise, including finite sampling, depolarising channels, and amplitude damping. Our results show that dissipation enhances the contraction of the reservoir dynamics, thereby improving stability. This demonstrates that noise, rather than being purely detrimental, can be leveraged to enhance robustness. In particular, RF-QRCs retain the ability to accurately infer chaotic dynamics and invariant properties even in noisy regimes, with identifiable regions of hyperparameter space where noise

improves performance relative to the noise-free case.

These findings suggest that RF-QRCs provide a practical route toward robust forecasting of complex dynamical systems. Overall, this chapter shows that noise can be exploited as a resource: by introducing dissipation, it enhances contraction and stability, and enables accurate inference of chaotic dynamics for appropriate choices of hyperparameters.

Chapter
7

Solving nonlinear differential equations with Fokker–Planck embedding

In this chapter, we investigate alternative approaches for solving nonlinear differential equations governing chaotic dynamics on quantum computers, in contrast to the data-driven quantum reservoir computing (QRC) methods discussed previously. In particular, we focus on linear embedding formulations of nonlinear dynamics, including the Fokker–Planck and Koopman–von Neumann (KvN) approaches. The primary contribution of this chapter is to extend the Fokker–Planck embedding framework of Tennie and Magri 2024 for solving nonlinear differential equations on quantum computers to chaotic dynamical systems. In doing so, we extend and analyse various numerical discretisation schemes, and analyse the effects of numerical diffusion. We also provide a comparison with the Koopman–von Neumann approach to classical mechanics, and implement block encoding of the nonlinear equations together with quantum hardware implementation results.

The objectives of this chapter are fivefold. First, we present the theoretical description of the Fokker–Planck and KvN approaches and outline related linear embedding methods from non-equilibrium statistical mechanics in Sec. 7.1. Second, in Sec. 7.2, we introduce first- and second-order central-difference and upwind schemes for discretising the Fokker–Planck operator. We investigate the relationship between explicitly added diffusion and the intrinsic numerical diffusion of these schemes, highlighting how both affect their effective behaviour. Third, in Sec. 7.4, we present existing unitary block-encoding methods (D. Camps and Beeumen 2022; Sünderhauf, Campbell, and J. Camps 2024) for integrating the discretised Fokker–Planck operator.

Fourth, in Sec. 7.5, we present numerical results for the nonlinear damped harmonic oscillator and the chaotic Lorenz system. We comment on the effects of grid size, diffusion, and numerical schemes in integrating nonlinear systems using a Fokker–Planck embedding.

Fifth, we demonstrate the block-encoding implementation of the Fokker–Planck evolution, followed by

a quantum hardware implementation of a one-dimensional nonlinear system in Sec. 7.7. These results highlight the potential of solving nonlinear chaotic differential equations on near-term quantum hardware.

7.1 Linear embedding of nonlinear dynamical systems

We recall the nonlinear autonomous dynamical system from Chap. 2

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t)), \tag{7.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^D$ is the state vector; $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^D$ is a continuously differentiable nonlinear vector function; and D is the number of degrees of freedom. Given an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, the solution of Eq. 7.1 is the trajectory $\boldsymbol{x}(t)$ in the phase space of dimension D and the resulting dynamics are deterministic.

Instead of evolving a single initial condition in the phase space, the Liouville representation introduces a probabilistic initial condition that is distributed according to $\rho_0 := \rho(\boldsymbol{x}, t = 0)$

$$\frac{\partial}{\partial t}\rho(\boldsymbol{x}, t) + \sum_{k=1}^{D} \frac{\partial}{\partial x_k} \left[f_k(\boldsymbol{x})\rho(\boldsymbol{x}, t) \right] = 0. \tag{7.2}$$

The nonlinear dynamical system in phase space is now represented by a linear partial differential equation (PDE) for the evolution of the distribution ρ given the initial distribution ρ_0. The Liouville equation is deterministic and describes the transport of a probability density function, where the probabilistic aspect arises solely from the probabilistic initial condition. In the limit where ρ_0 approaches a Dirac-δ distribution, the solution of Eq. 7.2 remains a δ-distribution that is evolved along the deterministic trajectory of Eq. 7.1.

This formulation provides a linear embedding of nonlinear dynamics, forming the basis for operator-theoretic representations such as the Perron–Frobenius (Liouville) operator, Koopman operator, Fokker–Planck equation, Hamilton–Jacobi formulations, and Carleman embeddings (Engel, Smith, and Parker 2021; Rauh, Minisini, and Aschemann 2009; Joseph 2020; Butt 2020; Ding and A. Zhou 2009; Lin et al. 2024). These approaches differ in the objects they evolve, densities, observables, value functions, or lifted state variables, but share the same principle of recasting nonlinear dynamics as a linear evolution in a function space (which is typically infinite-dimensional). We provide an overview of linear embedding approaches in Tab. 7.1. In this work, we focus on two complementary formulations: the Fokker–Planck and the Koopman–von Neumann (KvN) approaches.

7.1.1 Fokker–Planck approach

Most natural systems exhibit a separation of time scales between slow and fast degrees of freedom. The Fokker–Planck equation provides an effective description of such systems (Holubec, Kroy, and Steffenoni 2019). It governs the evolution of the probability density of the slow variables, capturing both their averaged deterministic dynamics and the fluctuations induced by the unresolved fast degrees of freedom. This description is obtained via a Markovian approximation, in which the slow variables are explicitly resolved, and the influence of the fast variables is modelled as memoryless noise. To summarise, the

Table 7.1: Overview of linear embedding approaches for nonlinear dynamical systems.

Approach	**Object Evolved**	**Representation**	**Key Idea**
Perron–Frobenius (Liouville)	Probability densities	$\partial_t \rho = -\nabla \cdot (\rho \boldsymbol{f})$	Describes the evolution of probability densities under deterministic dynamics through a linear transport operator.
Fokker–Planck Equation	Probability densities	$\partial_t \rho = -\nabla \cdot (\rho \boldsymbol{f}) + \mathcal{D}\nabla^2 \rho$	Extends the Liouville formulation to stochastic systems by incorporating diffusion and noise-driven dynamics.
Koopman–von Neumann (KvN)	Wavefunction-like states/observables	$i\partial_t \psi = \hat{L}\psi$	Provides a Hilbert-space formulation of classical dynamics analogous to quantum mechanics using linear operator evolution.
Koopman Operator	Observables	$\frac{d}{dt} g(\boldsymbol{x}(t)) = \mathcal{L} g(\boldsymbol{x}(t))$	Represents nonlinear dynamics through the linear evolution of observables in a function space.
Hamilton–Jacobi Formulation	Value functions	$\partial_t V + H(\boldsymbol{x}, \nabla V) = 0$	Recasts nonlinear control and optimisation problems through the evolution of value functions.
Carleman Embedding	Lifted state variables	$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z}$ (in a lifted space)	Transforms nonlinear dynamical systems into infinite-dimensional linear systems through polynomial lifting.

Fokker–Planck equation extends the Liouville equation by introducing diffusion terms that account for stochastic effects, such as unresolved dynamics, external forcing, and other random perturbations.

To derive Fokker–Planck equations (Shen et al. 2022; Tennie and Magri 2024), we consider a stochastic dynamical system governed by

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{\Gamma}(\boldsymbol{x}, t), \tag{7.3}$$

where $\boldsymbol{x}(t) = (x_1, x_2, \ldots, x_D)^T$ denotes the state vector of D dynamical variables, $\boldsymbol{f}$ is a generally nonlinear vector field, and $\boldsymbol{\Gamma}$ represents stochastic forcing. The stochastic term models both unresolved physical processes and intrinsic aleatoric uncertainty.

Unlike deterministic systems (Eq. 7.1), the evolution of Eq. 7.3 does not produce a single deterministic trajectory for a given initial condition. Instead, the system evolves probabilistically due to the stochastic forcing. Consequently, rather than considering a single trajectory $\boldsymbol{\xi}_{\bar{\Gamma}}(t)$ associated with one particular realisation of the noise $\bar{\Gamma}$, we consider the evolution of a probability density function $\rho(\boldsymbol{x}, t)$ over the phase space.

Given an initial distribution $\rho_{\text{in}} = \rho(\boldsymbol{x}, 0)$, the evolution of the probability density associated with Eq. 7.3 is governed by the Kramers–Moyal expansion (Risken 1989). For a single dynamical variable ($D = 1$),

the evolution equation takes the form

$$\frac{\partial}{\partial t}\rho(x,t) = \sum_{k=1}^{\infty}\left(-\frac{\partial}{\partial x}\right)^k \mathcal{D}^{[k]}(x,t)\rho(x,t), \tag{7.4}$$

where the coefficients $\mathcal{D}^{[k]}$ are the Kramers–Moyal moments,

$$\mathcal{D}^{[k]}(x,t) = \frac{1}{k!}\lim_{\tau\to 0}\frac{1}{\tau}\left\langle[\xi_\Gamma(t+\tau)-\xi_\Gamma(t)]^k\right\rangle_{|\,\xi_\Gamma(t)=x}, \tag{7.5}$$

and $\langle\cdot\rangle$ denotes the expectation over stochastic realisations.

For systems with multiple dynamical variables, the Kramers–Moyal expansion generalises to tensor-valued coefficients with multiple indices. We refer the reader to (Risken 1989) for further details.

Under the assumptions of δ-correlated Gaussian Langevin processes, all terms of order $k \geq 3$ vanish due to Pawula's theorem (Pawula 1967), reducing the Kramers–Moyal expansion to the Fokker–Planck equation,

$$\frac{\partial}{\partial t}\rho(\boldsymbol{x},t) = -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left(\mathcal{D}^{(i)}(\boldsymbol{x},t)\rho(\boldsymbol{x},t)\right) + \sum_{i,j=1}^{d}\frac{\partial^2}{\partial x_i\partial x_j}\left(\mathcal{D}^{(ij)}(\boldsymbol{x},t)\rho(\boldsymbol{x},t)\right) \equiv \hat{\mathcal{F}}\rho(\boldsymbol{x},t), \tag{7.6}$$

where $\hat{\mathcal{F}}$ denotes the linear Fokker–Planck operator.

The drift coefficients $\mathcal{D}^{(i)}$ and diffusion coefficients $\mathcal{D}^{(ij)}$ are determined by the nonlinear dynamics $\boldsymbol{f}$ and stochastic forcing amplitudes $\boldsymbol{g}_l$,

$$\mathcal{D}^{(i)}(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) + \sum_{k,l=1}^{d} g_{kl}(\boldsymbol{x},t)\frac{\partial}{\partial x_k}g_{il}(\boldsymbol{x},t), \tag{7.7}$$

$$\mathcal{D}^{(ij)}(\boldsymbol{x},t) = \sum_{k=1}^{d} g_{ik}(\boldsymbol{x},t)g_{jk}(\boldsymbol{x},t). \tag{7.8}$$

Importantly, the original nonlinear stochastic dynamical system in Eq. 7.3 has now been reformulated as a linear evolution equation acting on probability densities. This linear embedding forms the basis for the quantum implementations considered in this chapter.

In the deterministic limit $\boldsymbol{\Gamma} = 0$, the diffusion terms vanish and the Fokker–Planck equation reduces to the Liouville equation (Eq. 7.2). Hence, the Liouville equation can be interpreted as the zero-diffusion limit of the Fokker–Planck formulation. This connection highlights the close relationship between deterministic and stochastic operator-theoretic embeddings of nonlinear dynamical systems.

7.1.2 Koopman–von Neumann approach

The Fokker–Planck formulation evolves probability densities under stochastic dynamics. In the deterministic limit, this reduces to the Liouville equation, which evolves a probability density under the nonlinear flow field $\boldsymbol{f}$. The Koopman–von Neumann (KvN) formulation (Joseph 2020; Piasecki 2021) provides an alternative but closely related linear embedding of deterministic classical dynamics. Instead of evolving

the density directly, KvN introduces a complex-valued wavefunction-like object whose squared magnitude gives the classical probability density.

The total derivative of a scalar function $\rho(\boldsymbol{x}, t)$ along the flow from Eq. 7.1 is

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + \boldsymbol{f} \cdot \nabla \rho. \tag{7.9}$$

Expanding the divergence term in the Liouville equation gives

$$\frac{\partial \rho}{\partial t} + \boldsymbol{f} \cdot \nabla \rho + \rho \nabla \cdot \boldsymbol{f} = 0, \quad \frac{d\rho}{dt} = -\rho \nabla \cdot \boldsymbol{f}. \tag{7.10}$$

The KvN formulation introduces a complex wavefunction $\psi(\boldsymbol{x}, t)$ such that

$$\rho(\boldsymbol{x}, t) = |\psi(\boldsymbol{x}, t)|^2. \tag{7.11}$$

Writing the wavefunction in polar form,

$$\psi(\boldsymbol{x}, t) = \sqrt{\rho(\boldsymbol{x}, t)} e^{i\phi(\boldsymbol{x}, t)}, \tag{7.12}$$

where ϕ is a real phase, the total derivative of ψ along the classical flow is

$$\frac{d\psi}{dt} = \left[\frac{1}{2\rho} \frac{d\rho}{dt} + i \frac{d\phi}{dt} \right] \psi. \tag{7.13}$$

Using Eq. 7.10, this becomes

$$\frac{d\psi}{dt} = \left[-\frac{1}{2} \nabla \cdot \boldsymbol{f} + i \frac{d\phi}{dt} \right] \psi. \tag{7.14}$$

$$\frac{\partial \psi}{\partial t} = -\frac{1}{2} \left[\boldsymbol{f} \cdot \nabla + \nabla \cdot (\boldsymbol{f} \cdot) \right] \psi + i \frac{d\phi}{dt} \psi. \tag{7.15}$$

The final term corresponds to a phase evolution and does not affect the probability density $\rho = |\psi|^2$. Choosing this phase contribution to vanish gives the standard KvN evolution equation

$$\frac{\partial \psi}{\partial t} = -\frac{1}{2} \left[\boldsymbol{f} \cdot \nabla + \nabla \cdot (\boldsymbol{f} \cdot) \right] \psi. \tag{7.16}$$

Multiplying by i gives a Schrödinger-like equation

$$i \frac{\partial \psi}{\partial t} = \hat{H}_{\mathrm{KvN}} \psi, \tag{7.17}$$

where the KvN Hamiltonian is

$$\hat{H}_{\mathrm{KvN}} = -\frac{i}{2} \left[\boldsymbol{f} \cdot \nabla + \nabla \cdot (\boldsymbol{f} \cdot) \right]. \tag{7.18}$$

We introduce the momentum operator $\hat{\boldsymbol{p}} = -i\nabla$, and the Hamiltonian can also be written in symmetrised operator form as

$$\hat{H}_{\mathrm{KvN}} = \frac{1}{2} \left(\hat{\boldsymbol{p}} \cdot \boldsymbol{f}(\hat{\boldsymbol{x}}, t) + \boldsymbol{f}(\hat{\boldsymbol{x}}, t) \cdot \hat{\boldsymbol{p}} \right). \tag{7.19}$$

The symmetrisation ensures that $\hat{H}_{\mathrm{KvN}}$ is Hermitian under suitable boundary conditions.

The KvN formulation, therefore, provides a Hilbert-space representation of classical nonlinear dynamics, where the probability density evolves according to $\rho = |\psi|^2$. Although the underlying dynamics remain entirely classical, the evolution is recast into a linear Schrödinger-like equation with a Hermitian Hamiltonian. This operator-theoretic formulation enables the use of quantum algorithms developed for Hamiltonian simulation, making it a potentially suitable approach for nonlinear dynamics. We emphasise that, despite its resemblance to quantum mechanics, the KvN approach remains a classical method. From a numerical perspective, the KvN formulation does not include explicit diffusion and can suffer from parasitic oscillations (Piasecki 2021). Some recent proposals modify the KvN method by explicitly adding diffusion to the KvN Hamiltonian in order to suppress these oscillations (Joseph 2020; Novikau and Joseph 2025); however, the resulting equation lacks physical interpretation. We briefly analyse the original KvN equation without added diffusion in Sec. 7.5, while focusing primarily on the Fokker–Planck formulation throughout this chapter. We now present finite-difference discretisation methods for the Fokker–Planck operator.

7.2 Spatial discretisation of Fokker–Planck equation

To solve the Fokker–Planck equation numerically and enable quantum implementations, the continuous operator $\hat{\mathcal{F}}$ must first be discretised in space. The resulting semi-discretised system transforms the continuous probability density evolution into a finite-dimensional linear system that can be represented by sparse matrices and evolved using classical or quantum algorithms.

Let the spatial domain be the interval $[-l, l]$, discretised into N uniformly spaced grid points

$$x_k = -l + k\Delta x, \qquad k = 0, \ldots, N-1, \tag{7.20}$$

with grid spacing

$$\Delta x = \frac{2l}{N-1}. \tag{7.21}$$

The continuous probability density $\rho(x,t)$ is approximated on the grid by

$$\boldsymbol{\rho}(t) = (\rho(x_0,t), \rho(x_1,t), \ldots, \rho(x_{N-1},t))^T. \tag{7.22}$$

Alternatively, it is often convenient to work with the probability masses $p_k(t) = \rho(x_k,t)\Delta x$, which represent the probability of finding the system within the interval $x_k \pm \Delta x/2$.

$$\boldsymbol{p}(t) = (p_0, p_1, \ldots, p_{N-1})^T, \tag{7.23}$$

The semi-discretised Fokker–Planck equation takes the form of the master equation

$$\dot{\boldsymbol{p}} = \boldsymbol{\mathcal{R}}\boldsymbol{p}, \tag{7.24}$$

where $\boldsymbol{\mathcal{R}}$ is the discretized Fokker–Planck operator.

The formal solution is given by

$$\boldsymbol{p}(t) = \mathcal{T} \exp\left(\int_0^t \boldsymbol{\mathcal{R}}(t')dt'\right)\boldsymbol{p}(0), \tag{7.25}$$

where $\mathcal{T}$ denotes time ordering.

The discretisation of the spatial derivatives for the matrix operator $\boldsymbol{\mathcal{R}}$ is not unique. For any consistent discretisation, Eq. 7.24 must converge to the continuous Fokker–Planck equation, (Eq. 7.6), as the grid is refined. One physically consistent ansatz, considered in Tennie and Magri 2024; Holubec, Kroy, and Steffenoni 2019, is obtained by balancing the transition rates between neighbouring grid points for all non-boundary points. This construction leads to a central-difference discretisation of the Fokker–Planck operator, as discussed in Sec. 7.2.1.

Before presenting the numerical results for the integration of the Fokker–Planck equation, we introduce two discretisation schemes for the spatial derivatives: the central difference and upwinding schemes. We then discuss their equivalence in terms of numerical diffusion. While diffusion can stabilise the solution and suppress parasitic oscillations, excessive diffusion can over-dampen the probability density and distort the underlying dynamics. We demonstrate this trade-off in the results section (Sec. 7.5).

7.2.1 Central difference scheme

We give an overview of numerical schemes, including central difference and upwinding and present the corresponding discretisation ansatz for the Fokker–Planck operator. To discretise the drift terms in the Fokker–Planck operator, we first consider the one-dimensional advection equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{7.26}$$

where a denotes a constant advection velocity. Introducing a uniform spatial grid $x_j = j\Delta x$ and discrete time levels $t^n = n\Delta t$. Using the symmetric approximation of the spatial derivative

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0. \tag{7.27}$$

The stencil is second-order accurate in space and first-order accurate in time. Alternatively, substituting Taylor expansions

$$u_{j\pm1}^n = u_j^n \pm \Delta x\frac{\partial u}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u}{\partial x^2} + O(\Delta x^3), \qquad u_j^{n+1} = u_j^n + \Delta t\frac{\partial u}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + O(\Delta t^3), \tag{7.28}$$

into Eq. 7.27 recovers Eq. 7.26 in the limit $\Delta x, \Delta t \to 0$.

The central difference approximation yields the following rate equation for non-boundary grid points (Holubec, Kroy, and Steffenoni 2019; Tennie and Magri 2024)

$$\dot{p}_k = \frac{1}{\Delta x}\left(\frac{1}{2}\mathcal{D}_{k-1}^{(x)} + \frac{\mathcal{D}_{k-1}^{(xx)}}{\Delta x}\right)p_{k-1} - \frac{2\mathcal{D}_k^{(xx)}}{\Delta x^2}p_k + \frac{1}{\Delta x}\left(-\frac{1}{2}\mathcal{D}_{k+1}^{(x)} + \frac{\mathcal{D}_{k+1}^{(xx)}}{\Delta x}\right)p_{k+1}, \tag{7.29}$$

where $\mathcal{D}_j^{(x)} \equiv \mathcal{D}^{(x)}(x_j)$ and $\mathcal{D}_j^{(xx)} \equiv \mathcal{D}^{(xx)}(x_j)$ are the drift and diffusion coefficients evaluated at grid point x_j. The resulting rate matrix R is tridiagonal and sparse. For the off-diagonal entries to remain positive and $\boldsymbol{p}(t)$ to be positive probabilities, the constraint on the grid spacing is given by

$$\Delta x \leq \min_k \left| \frac{2\mathcal{D}_k^{(xx)}}{\mathcal{D}_k^{(x)}} \right|, \tag{7.30}$$

which imposes a lower bound on the number of grid points for a given domain (Tennie and Magri 2024). This tridiagonal structure is well suited to both classical and quantum implementations of the master Eq. 7.24.

7.2.2 First- and second-order upwinding schemes

For advection-dominated problems, the constraint given in Eq. 7.30 provides a practical bound on the minimum number of grid points required for the stability of the scheme. To relax this constraint, we use the upwinding schemes that address this constraint by incorporating directional information from the flow velocity into the spatial approximation.

For positive advection velocity $a > 0$, information propagates from left to right, motivating a backwards-biased stencil

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0. \tag{7.31}$$

For $a < 0$ the stencil is reversed accordingly

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_j^n}{\Delta x} = 0. \tag{7.32}$$

Substituting the Taylor expansion of u_{j-1}^n into Eq. 7.31 confirms consistency in the limit $\Delta x, \Delta t \to 0$, though the scheme is only first-order accurate in space. The leading truncation error takes the form of a numerical diffusion term proportional to $\frac{1}{2}a\Delta x\, \partial^2 u/\partial x^2$, which stabilises the solution at the cost of added dissipation.

Spatial accuracy can be further improved by considering a higher-order stencil. For $a > 0$, the second-order upwind approximation is

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{3u_j^n - 4u_{j-1}^n + u_{j-2}^n}{2\Delta x} = 0, \tag{7.33}$$

with the corresponding forward-biased stencil for $a < 0$,

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{-3u_j^n + 4u_{j+1}^n - u_{j+2}^n}{2\Delta x} = 0. \tag{7.34}$$

Both stencils are second-order accurate in space, recovering the directional stability of first-order upwinding while reducing the leading truncation error.

When applied to the drift terms of the Fokker–Planck operator, the first-order upwind scheme yields the

rate equation

$$\begin{aligned}\dot{p}_k =& \frac{1}{\Delta x}\left(\left[\mathcal{D}_{k-1}^{(x)}\right]^+ + \frac{\mathcal{D}_{k-1}^{(xx)}}{\Delta x}\right) p_{k-1} - \frac{1}{\Delta x}\left(\left[\mathcal{D}_k^{(x)}\right]^+ - \left[\mathcal{D}_k^{(x)}\right]^- + \frac{2\mathcal{D}_k^{(xx)}}{\Delta x}\right) p_k \\ &+ \frac{1}{\Delta x}\left(-\left[\mathcal{D}_{k+1}^{(x)}\right]^- + \frac{\mathcal{D}_{k+1}^{(xx)}}{\Delta x}\right) p_{k+1}.\end{aligned} \tag{7.35}$$

where $[\,\cdot\,]^+ = \max(\,\cdot\,, 0)$ and $[\,\cdot\,]^- = \min(\,\cdot\,, 0)$ denotes the positive and negative parts of the drift coefficient, respectively, ensuring the stencil correctly reflects the local flow direction at each grid point. Compared with Eq. 7.29, the first-order upwind discretisation guarantees positive off-diagonal entries by construction, relaxing the grid resolution requirement of Eq. 7.30 at the cost of introducing numerical diffusion proportional to $\frac{1}{2}|\mathcal{D}_k^{(x)}|\Delta x$. The second-order upwind scheme improves the spatial accuracy further while retaining the directional dependence of the transport operator. Extending to the second-order upwind scheme also introduces an additional dependence on p_{k-2} (or p_{k+2} for negative drift).

In the discretised Fokker–Planck equation, these upwind approximations are applied to the drift terms, producing sparse non-Hermitian matrix operators that preserve the directional structure of the underlying nonlinear flow. In both cases, the resulting rate matrix $\boldsymbol{\mathcal{R}}$ is sparse and non-Hermitian, preserving the directional structure of the underlying nonlinear flow.

In summary, the continuous Fokker–Planck equation and boundary conditions can be consistently discretised in the form of a master equation, which is linear and non-normal. We consider reflective boundary conditions in this work, and the results for various discretisation schemes and nonlinear systems are presented in Sec. 7.5.

7.3 Numerical diffusion

A key consequence of the choice of discretisation scheme is the introduction of numerical errors, including artificial diffusion and dispersion, arising from the truncation of the finite-difference approximation. To quantify this effect, we analyse the modified equation associated with each scheme by retaining higher-order terms in the Taylor expansion, and then establish a physical limitation on resolving the chaotic attractor and fine scales in the presence of diffusion.

7.3.1 First-order upwind scheme

Substituting the Taylor expansions of u_{j-1}^n and u_j^{n+1} into the first-order upwind scheme (Eq. 7.31) and retaining leading-order truncation terms yields

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \frac{a\Delta x}{2}\left(1 - \frac{a\Delta t}{\Delta x}\right)\frac{\partial^2 u}{\partial x^2} + O(\Delta x^2, \Delta t^2). \tag{7.36}$$

The right-hand side reveals that the first-order upwind scheme does not solve the original advection equation (Eq. 7.26), but rather a modified equation with an effective numerical diffusion coefficient,

$$\mathcal{D}_{\text{num}} = \frac{a\Delta x}{2}\left(1 - \frac{a\Delta t}{\Delta x}\right) = \frac{a\Delta x}{2}\left(1 - \text{CFL}\right), \tag{7.37}$$

where CFL $= a\Delta t/\Delta x$ is the Courant–Friedrichs–Lewy number. For $0 < \text{CFL} \leq 1$, $\mathcal{D}_{\text{num}} \geq 0$, so the scheme is dissipative. The numerical diffusion vanishes only when CFL $= 1$, i.e. when the discrete propagation speed exactly matches the physical advection velocity.

7.3.2 Second-order upwind scheme

Carrying out the same modified-equation analysis for the second-order upwind scheme in Eq. (7.33), the spatial approximation eliminates the leading second-derivative diffusion term present in the first-order upwind scheme. Retaining the leading temporal and spatial truncation terms gives

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = -\frac{1}{2}a^2\Delta t\,\frac{\partial^2 u}{\partial x^2} + \frac{a\Delta x^2}{3}\frac{\partial^3 u}{\partial x^3} + O(\Delta x^3, \Delta t^2). \tag{7.38}$$

The leading spatial truncation error is therefore proportional to the third spatial derivative and is predominantly dispersive, rather than diffusive. The forward-Euler temporal discretisation introduces the second-derivative term

$$\mathcal{D}_{\text{num}} = -\frac{1}{2}a^2\Delta t, \tag{7.39}$$

which has a negative coefficient and therefore corresponds to anti-diffusion rather than numerical dissipation. Consequently, the second-order upwind scheme does not contain the leading positive spatial numerical diffusion of the first-order upwind scheme, but its stability depends on the temporal discretisation and, in the present Fokker–Planck setting, on the explicitly included physical diffusion.

7.3.3 Diffusion time scale and the Lyapunov constraint

We have established that, in the Fokker–Planck formulation, the evolution contains both physical diffusion and intrinsic numerical diffusion arising from the discretisation scheme. The total effective diffusion, therefore, contains contributions from both the physical diffusion tensor $\mathcal{D}^{(xx)}$ and the numerical diffusion $\mathcal{D}_{\text{num}}$. We now compare the associated diffusion time scale against the intrinsic dynamical time scales of the dynamical system. Consider the diffusion equation

$$\frac{\partial \phi(x,t)}{\partial t} = \mathcal{D}\,\nabla^2\phi(x,t), \tag{7.40}$$

and take the Fourier transform with respect to x, yielding

$$\frac{\partial \hat{\phi}(k,t)}{\partial t} = -\mathcal{D}k^2\,\hat{\phi}(k,t), \tag{7.41}$$

whose solution is $\hat{\phi}(k,t) = e^{-\mathcal{D}k^2 t}\,\hat{\phi}(k,0)$. The diffusion time scale at wavenumber k is therefore $\tau_d = (\mathcal{D}k^2)^{-1}$, with units of time. For a spatially discretised domain with grid spacing Δx, the maximum resolved wavenumber is (Allawala and Marston 2016)

$$k_{\text{max}} = \frac{\pi}{\Delta x}, \tag{7.42}$$

so the minimum diffusion time scale, governing the fastest-decaying mode on the grid is

$$\tau_{d,\text{min}}^{-1} = \mathcal{D}k_{\text{max}}^2 = \frac{\mathcal{D}\pi^2}{\Delta x^2}. \tag{7.43}$$

This timescale sets a fundamental constraint on diffusion, which acts directly on the smallest resolved scales, and if $\tau_{d,\min}$ is too short relative to the dynamical time scales of the system, fine-scale structures in the probability density are irreversibly smoothed out.

For chaotic systems, the relevant dynamical time scale is naturally given by the Lyapunov time

$$\tau_\lambda = \lambda_{\max}^{-1}, \tag{7.44}$$

where $\lambda_{\max}$ is the maximum Lyapunov exponent. To ensure that diffusion does not erase these structures faster than they are generated, the fastest diffusive decay rate on the grid must remain smaller than the Lyapunov growth rate,

$$\tau_{d,\min}^{-1} = \mathcal{D} k_{\max}^2 \ll \lambda_{\max}. \tag{7.45}$$

Using Eq. 7.43, this gives the constraint

$$\mathcal{D} \ll \frac{\lambda_{\max} \Delta x^2}{\pi^2}. \tag{7.46}$$

Eq. 7.46 provides an upper bound on the effective diffusion that can be introduced while still resolving the Lyapunov-generated fine-scale structure of the dynamics. Here $\mathcal{D}$ should be interpreted as the total effective diffusion, including both the physical diffusion contribution $\mathcal{D}^{(xx)}$ and the numerical diffusion $\mathcal{D}_{\text{num}}$ introduced by the discretisation scheme.

If this condition is violated, diffusion begins to damp the probability density at length scales comparable to those of the chaotic attractor itself. As a result, the fine filamentary structures that characterise the chaotic invariant set are progressively smeared out. The numerical solution therefore no longer converges to the true invariant measure of the deterministic, or weakly diffusive, system, but instead approaches an artificially broadened distribution.

7.3.4 Implications for scheme selection

The Lyapunov constraint in Eq. 7.46 has direct implications for the choice of discretisation scheme. In particular, the effective diffusion introduced by the scheme must remain sufficiently small to avoid suppressing the fine-scale structure generated by the chaotic dynamics.

Comparing the schemes introduced in Secs. 7.2.1–7.2.2:

- The *central difference* scheme introduces no leading-order spatial numerical diffusion and therefore provides the most accurate representation of the dynamics when the grid resolution condition in Eq. 7.30 is satisfied. However, any physical diffusion added for stability must itself satisfy Eq. 7.46, thus requiring a large number of grid points.

- The *first-order upwind* scheme introduces numerical diffusion

$$\mathcal{D}_{\text{num}} \sim \frac{1}{2} a \Delta x (1 - \text{CFL}), \tag{7.47}$$

  which scales linearly with Δx. While this stabilises the evolution and suppresses oscillations,

excessive diffusion may smooth the probability density and violate the Lyapunov constraint.

- The *second-order upwind* scheme removes the leading-order second-derivative contribution to spatial numerical diffusion. Its leading spatial truncation error is instead dispersive, while the forward-Euler temporal discretisation introduces the signed effective coefficient

$$\mathcal{D}_{\text{num}} \sim -\frac{1}{2}a^2\Delta t. \tag{7.48}$$

  The negative sign indicates an anti-diffusive rather than dissipative effect. Under CFL-constrained time stepping, for which $\Delta t = O(\Delta x)$, the magnitude $|\mathcal{D}_{\text{num}}|$ scales as $O(\Delta x)$. Consequently, the scheme avoids the leading positive spatial numerical diffusion of the first-order upwind scheme, although its stability depends on the temporal discretisation and any explicitly added diffusion.

In all three cases, the numerical diffusion acts as an additive contribution to the physical diffusion coefficient $\mathcal{D}^{(xx)}$. The comparison between physical diffusion, numerical diffusion, and the diffusivity of a chaotic attractor is examined systematically in Sec. 7.5. We now present a unitary block encoding (Arrazola, Guala, and Soni 2023) method for integrating the Fokker–Planck equation on near-term quantum hardware.

7.4 Unitary block encoding for forward Euler Fokker–Planck integration

The semi-discretised Fokker–Planck equation introduced in Eq. 7.24 can be integrated numerically using the forward Euler method with time-step size Δt (Tennie and Magri 2024). The forward Euler update takes the form

$$\boldsymbol{p}(t + \Delta t) = (\boldsymbol{I} + \Delta t\,\boldsymbol{\mathcal{R}})\,\boldsymbol{p}(t). \tag{7.49}$$

We first define the forward-Euler evolution operator

$$\boldsymbol{A}_0 = \boldsymbol{I} + \Delta t\,\boldsymbol{\mathcal{R}}. \tag{7.50}$$

To ensure that the operator used in the block encoding is a contraction, we normalise it as

$$\boldsymbol{A} = \frac{\boldsymbol{A}_0}{\alpha}, \qquad \alpha = \|\boldsymbol{A}_0\|_2, \tag{7.51}$$

such that $\|\boldsymbol{A}\|_2 \leq 1$.

Starting from an initial probability vector $\boldsymbol{p}(0)$, repeated Euler updates require repeated applications of the non-unitary operator $\boldsymbol{A}_0$. In the quantum implementation, we instead apply its normalised form $\boldsymbol{A}$, which produces a state proportional to the same forward-Euler update after post-selection and normalisation. Let $|\boldsymbol{p}(0)\rangle$ denote a normalised quantum state proportional to the classical probability vector $\boldsymbol{p}(0)$. Since $\boldsymbol{A}$ is generally non-unitary, its action cannot be directly implemented on a quantum computer using standard unitary evolution. To overcome this, we employ unitary block encoding (Arrazola, Guala, and Soni 2023).

Introducing an ancilla qubit $|q_a\rangle$, we construct the unitary operator

$$U_{\Delta t} = \begin{pmatrix} \sqrt{\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A}} & \boldsymbol{A}^\dagger \\ \boldsymbol{A} & -\sqrt{\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^\dagger} \end{pmatrix}, \tag{7.52}$$

or, equivalently, in tensor-product form,

$$\begin{aligned} U_{\Delta t} = &|0_a\rangle\langle 0_a| \otimes \sqrt{\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A}} \\ &+ |0_a\rangle\langle 1_a| \otimes \boldsymbol{A}^\dagger \\ &+ |1_a\rangle\langle 0_a| \otimes \boldsymbol{A} \\ &- |1_a\rangle\langle 1_a| \otimes \sqrt{\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^\dagger}. \end{aligned} \tag{7.53}$$

A single forward Euler propagation step is implemented by applying $U_{\Delta t}$ to the composite state $|0_a\rangle \otimes |\boldsymbol{p}(t)\rangle$, followed by a projective measurement and post-selection of the ancilla qubit in the state $|1_a\rangle$. A subsequent Pauli-X operation may be applied to reset the ancilla to $|0_a\rangle$ before the next propagation step.

Conditioned on successful post-selection, the system register is proportional to

$$\boldsymbol{A}|\boldsymbol{p}(t)\rangle = \frac{1}{\alpha}\boldsymbol{A}_0|\boldsymbol{p}(t)\rangle. \tag{7.54}$$

The factor $1/\alpha$ does not affect the encoded state after normalisation, although it changes the post-selection success probability.

The corresponding success probability is

$$\begin{aligned} P_{\text{suc}}(t) &= \langle \boldsymbol{p}(t)|\boldsymbol{A}^\dagger \boldsymbol{A}|\boldsymbol{p}(t)\rangle \\ &= \frac{1}{\alpha^2}\Big[1 + \Delta t \langle \boldsymbol{p}(t)| \left(\boldsymbol{\mathcal{R}} + \boldsymbol{\mathcal{R}}^\dagger\right) |\boldsymbol{p}(t)\rangle + O(\Delta t^2)\Big]. \end{aligned} \tag{7.55}$$

By construction, $\|\boldsymbol{A}\|_2 \leq 1$, and therefore

$$\boldsymbol{A}^\dagger \boldsymbol{A} \preceq \boldsymbol{I}, \qquad \boldsymbol{A}\boldsymbol{A}^\dagger \preceq \boldsymbol{I}. \tag{7.56}$$

Consequently, $\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A}$ and $\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^\dagger$ are positive semidefinite. This guarantees that the block-encoded unitary operator $U_{\Delta t}$ is well defined and that the success probability satisfies $P_{\text{suc}}(t) \leq 1$.

For generic states that are not mapped entirely onto a singular subspace of $\boldsymbol{A}$ with singular value one, the success probability is strictly smaller than unity. Consequently, after n repeated Euler steps, the overall success probability is

$$P_{\text{suc}}^{(n)} = \prod_{k=0}^{n-1} \langle \boldsymbol{p}(t_k)|\boldsymbol{A}^\dagger \boldsymbol{A}|\boldsymbol{p}(t_k)\rangle, \qquad t_k = k\Delta t. \tag{7.57}$$

For small Δt, this product may decrease approximately exponentially with the total integration time $T = n\Delta t$.

Therefore, although unitary block encoding enables the implementation of non-unitary forward Euler

evolution on a quantum computer, the success probability decreases with increasing total integration time, which limits the achievable simulation horizon. In this chapter, we first present classically simulated Fokker–Planck solutions for two-dimensional nonlinear systems and three-dimensional chaotic systems in Sec. 7.5. We then implement the block-encoding framework for a one-dimensional prototypical nonlinear system and present the corresponding quantum-hardware implementation results in Sec. 7.7.

7.5 Results

We present the numerical integration solution for the Fokker–Planck equation for a nonlinear damped harmonic oscillator and the chaotic Lorenz-63 system.

7.5.1 Damped harmonic oscillator

We first consider a two-dimensional damped harmonic oscillator with drift fields

$$\boldsymbol{f}_x(x, y) = -y - \gamma x y^2, \tag{7.58}$$

$$\boldsymbol{f}_y(x, y) = x - \gamma x^2 y, \tag{7.59}$$

where $\gamma = 0.1$ controls the nonlinear damping. The original trajectory of the dynamical system is obtained using the fourth-order Runge–Kutta method with time step $\Delta t = 0.01$, starting from the initial condition $(x_0, y_0) = (0, 0.6)$. The integrated trajectory is shown in Fig. 7.1.

For the Fokker–Planck formulation, we consider the computational domain $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] = [-2, 2] \times [-2, 2]$. The initial probability density is taken to be a normalised δ -distribution centred at the initial condition. To obtain the reference Fokker–Planck solution, we compute the matrix exponential evolution of the discretised operator over a total physical time of $T = 50$. We then compare this against the forward-Euler Fokker–Planck integration using different discretisation schemes and grid resolutions.

The numerical parameters, discretisation schemes, and grid resolutions are summarised in Table 7.2.

Table 7.2: Input parameters used for the damped-harmonic oscillator Fokker–Planck simulation.

Parameter	Symbol / Setting	Central difference	Upwinding
Numerical scheme	–	Central difference	Upwind
Boundary condition	–	Reflective	Reflective
Grid points	$N_x \times N_y$	32×32	32×32, 64×64, 128×128
Added diffusion coefficient	$\mathcal{D}$	0.05	0.00
Model parameter	γ	0.1	0.1
Initial condition	(x_0, y_0)	$(0, 0.6)$	$(0, 0.6)$
Matrix exponential time step	Δt_{ref}	0.01	0.01
Forward-Euler FP time step	Δt_{FP}	0.001	0.001

Central difference with added diffusion

We first analyse the central difference discretisation of the Fokker–Planck operator for the damped harmonic oscillator system. In Fig. 7.2, the reference trajectories are shown to evolve from the initial

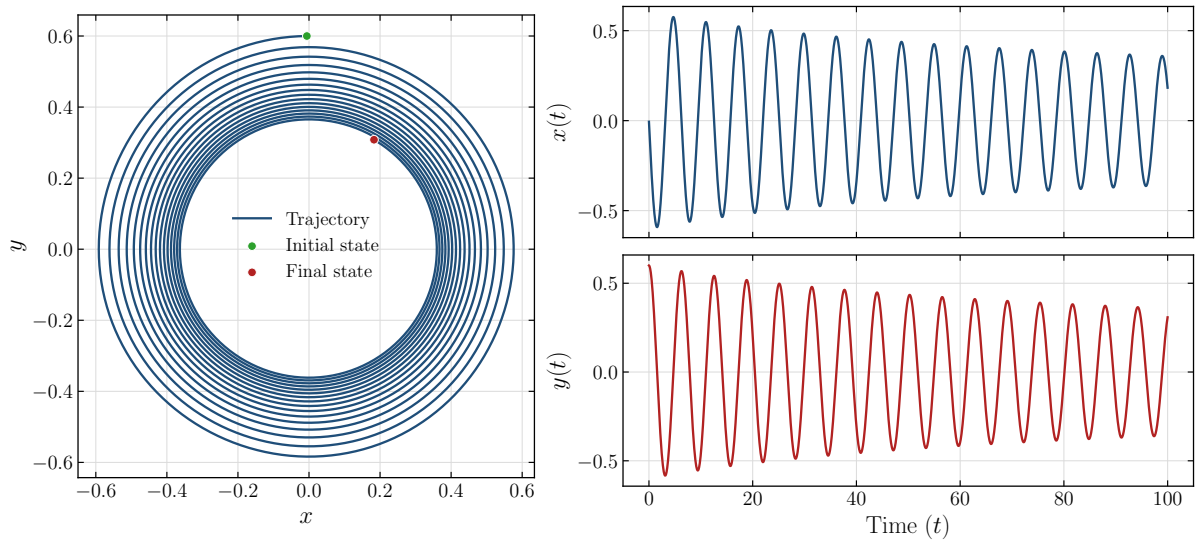


Figure 7.1: Original dynamical equation solution of the 2-dimensional damped harmonic oscillator. On the left, phase space trajectory solution. On the right, time series plot of each variable

δ-distribution forward in time. In Fig. 7.3, the Euler-integrated trajectory is shown, with similar evolution as the reference solution.

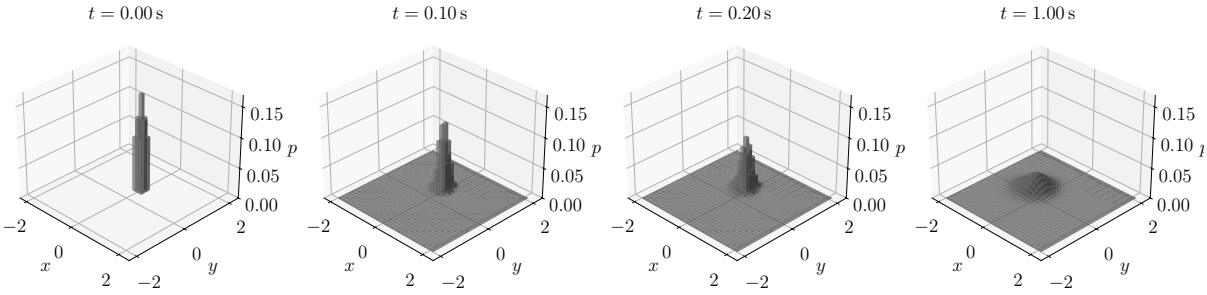


Figure 7.2: Reference phase space solution of the 2-dimensional damped harmonic oscillator obtained via matrix exponentiation. The evolution of the probability distribution function from the initial condition.

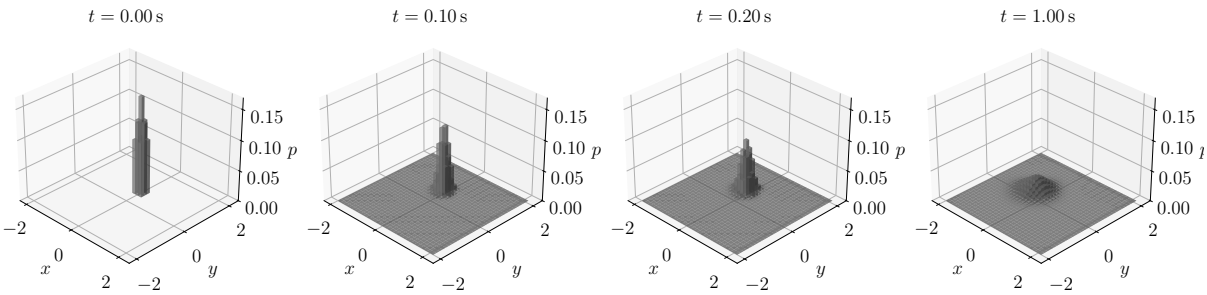


Figure 7.3: Integrated phase space solution of the 2-dimensional damped harmonic oscillator obtained via a first-order Euler central difference scheme. The evolution of the probability distribution function from the initial condition.

To make a quantitative comparison, we plot the absolute error difference between the reference and

obtained solution and present the results in Fig 7.4. The long-term solution over 5 seconds of physical time gives an error metric of $1e-4$. This can also be visualised in a phase-space trajectory comparison plot in Fig. 7.5.

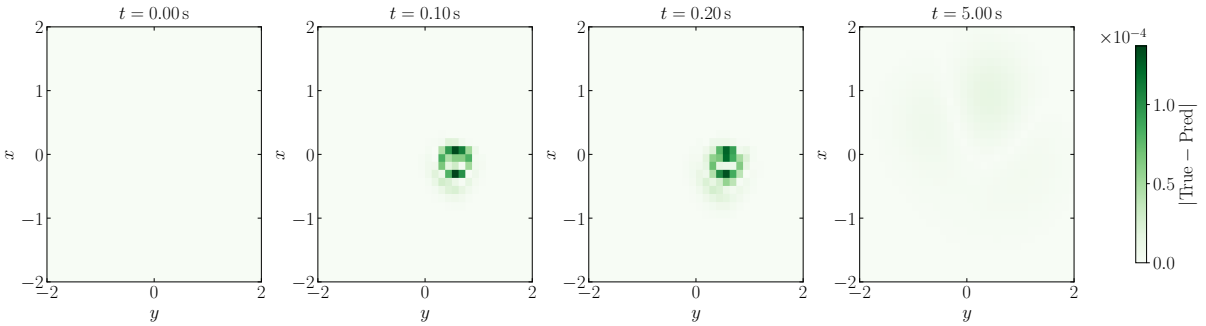


Figure 7.4: Error bar for the reference vs integrated phase space solution of the 2-dimensional damped harmonic oscillator for the first-order Euler central difference scheme. The colour bar indicates the absolute difference.

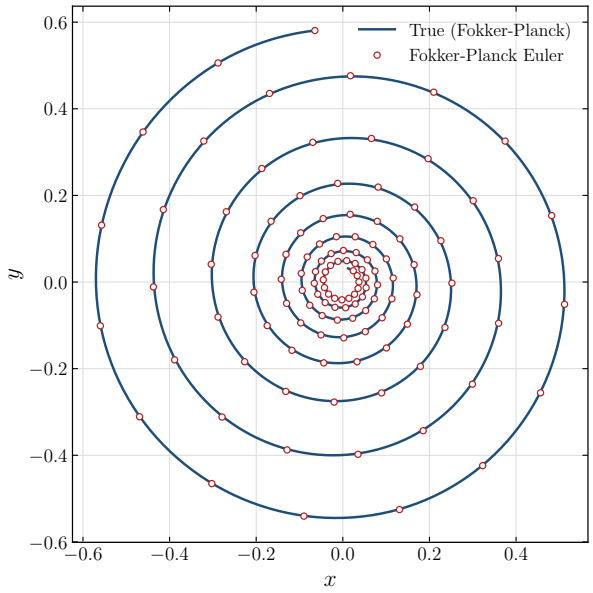


Figure 7.5: Phase space trajectory solution, reference solution vs the integrated solution obtained via first-order Euler central difference scheme.

We now compare the reference and the obtained solution with the original dynamical system solution equations in Fig. 7.28. Although the forward Euler integrated Fokker–Planck solution is close to the reference solution obtained via matrix exponential (reference solution), with similar settings. Both solutions converge to an overdamped stationary solution faster than the original solution, thus making the short-term time series prediction limited.

Upwinding with zero added diffusion

We now analyse the first-order upwinding discretisation of the Fokker–Planck operator without any explicitly added diffusion. Unlike the central difference scheme discussed previously, the upwinding formulation introduces intrinsic numerical diffusion through the directional discretisation of the advection

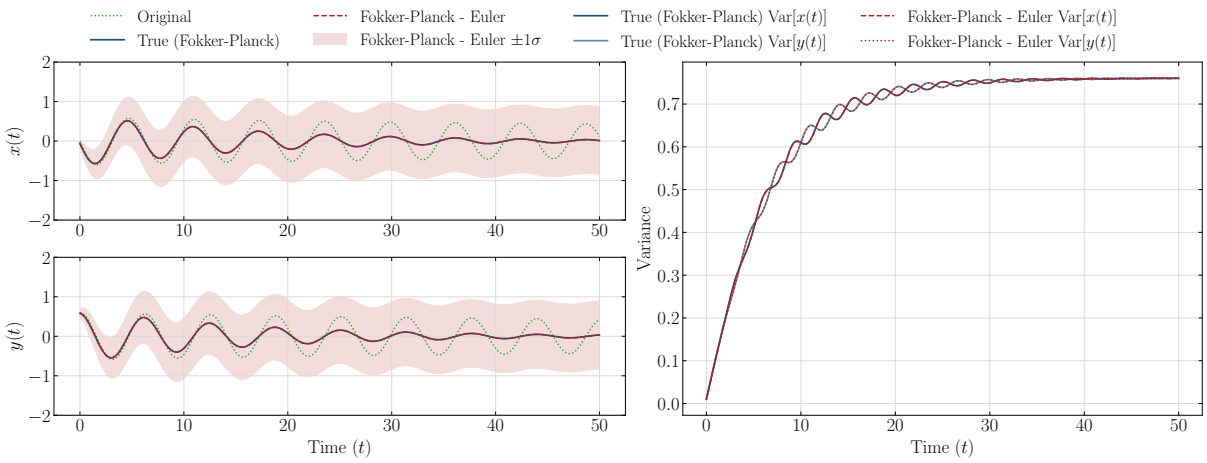


Figure 7.6: Trajectory comparison for the reference vs integrated Fokker–Planck solution obtained via first-order Euler central difference, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time.

terms. We investigate how this numerical diffusion affects the long-term stability and accuracy of the obtained probability density solution.

We first consider a coarse 32×32 phase-space discretisation. The absolute error between the reference matrix exponential solution and the forward Euler upwind-integrated solution is shown in Fig. 7.7. Compared to the central difference scheme with added diffusion, the solution remains stable despite the absence of explicit physical diffusion. This indicates that the numerical diffusion generated by the upwinding scheme is sufficient to regularise the solution at coarse resolution.

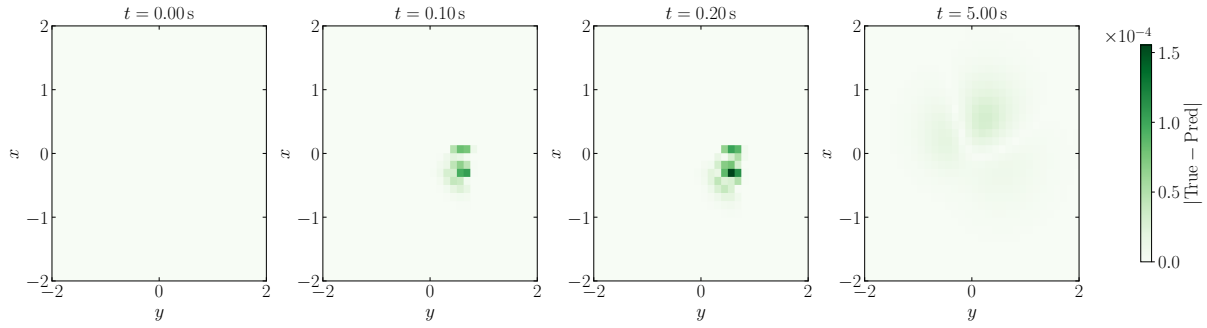


Figure 7.7: Error plot for the reference versus integrated phase space solution of the 2-dimensional damped harmonic oscillator using the first-order Euler upwinding scheme with zero added diffusion on a 32×32 grid. The colour bar indicates the absolute difference between the reference matrix exponential solution and the integrated Fokker–Planck solution.

The corresponding trajectory and variance comparison against the original dynamical system are presented in Fig. 7.8. Similar to the central difference case with added diffusion, the upwinding solution converges toward an overdamped stationary state faster than the original nonlinear dynamical system. The increased damping arises from the intrinsic numerical diffusion of the upwinding discretisation, which smooths the probability density during propagation.

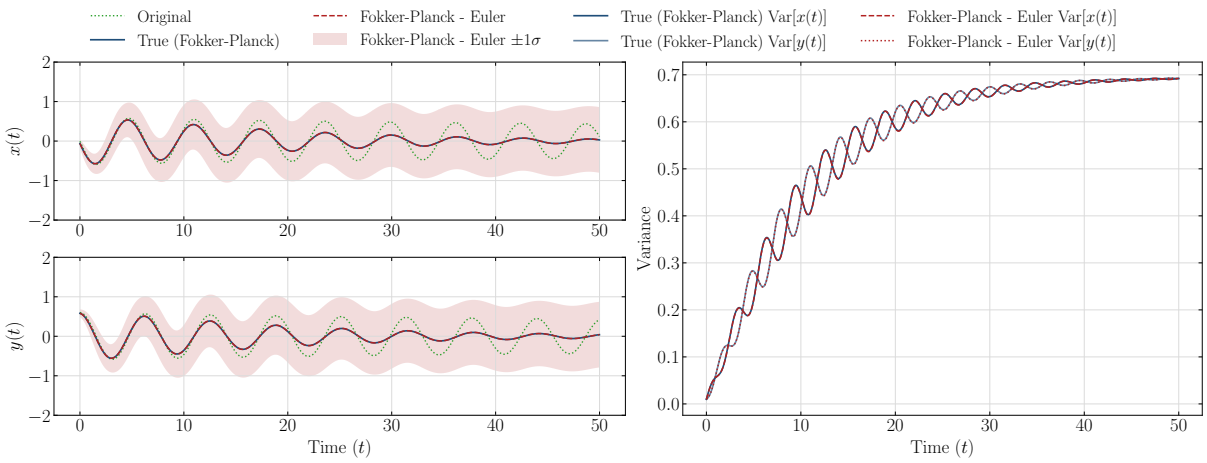


Figure 7.8: Trajectory comparison for the reference versus integrated Fokker–Planck solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 32×32 grid, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time.

This behaviour can also be visualised in the phase-space mean trajectory comparison shown in Fig. 7.9, where the integrated solution follows the reference solution closely while exhibiting stronger damping relative to the original dynamical system trajectory.

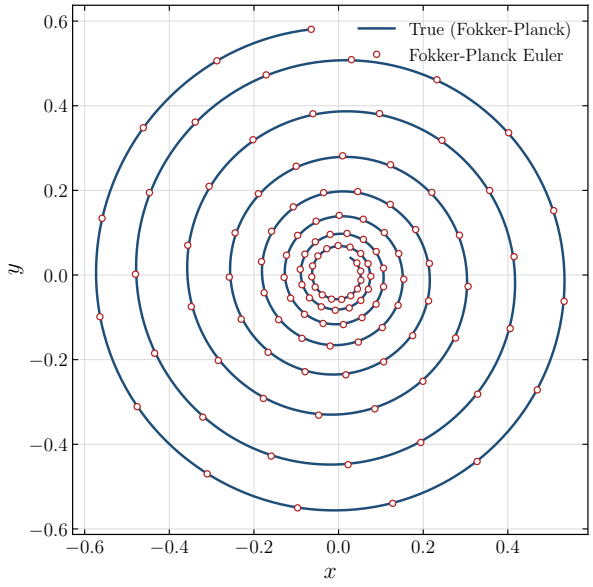


Figure 7.9: Phase space trajectory solution for the reference solution versus the integrated solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 32 × 32 grid.

We next increase the phase-space resolution to a 64 × 64 grid. The corresponding error plot is shown in Fig. 7.10. Increasing the spatial resolution reduces the discretisation error and improves the agreement between the reference and integrated solutions. The finer discretisation also reduces the effective numerical diffusion associated with the upwinding scheme.

The trajectory comparison in Fig. 7.11 shows improved agreement with the original system dynamics

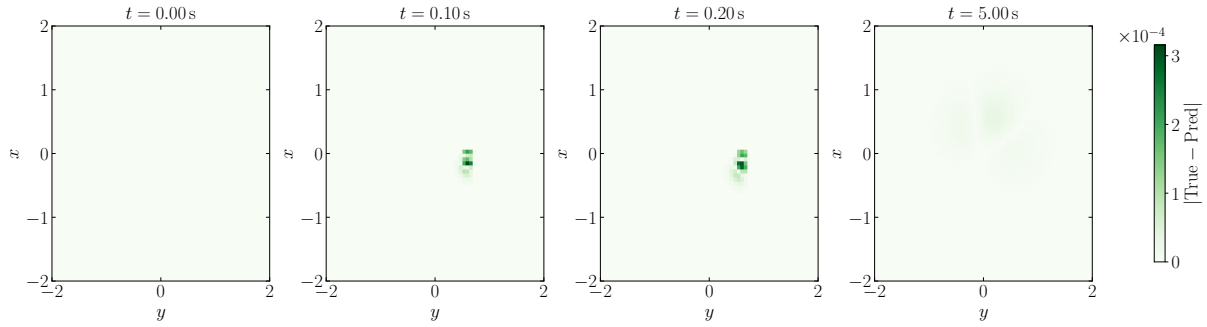


Figure 7.10: Error plot for the reference versus integrated phase space solution of the 2-dimensional damped harmonic oscillator using the first-order Euler upwinding scheme with zero added diffusion on a 64×64 grid. The colour bar indicates the absolute difference between the reference matrix exponential solution and the integrated Fokker–Planck solution.

compared to the 32×32 grid case. The variance decay remains slower relative to the coarse-grid solution, indicating that the reduced numerical diffusion allows the probability density to retain more of the underlying dynamical structure over longer integration times.

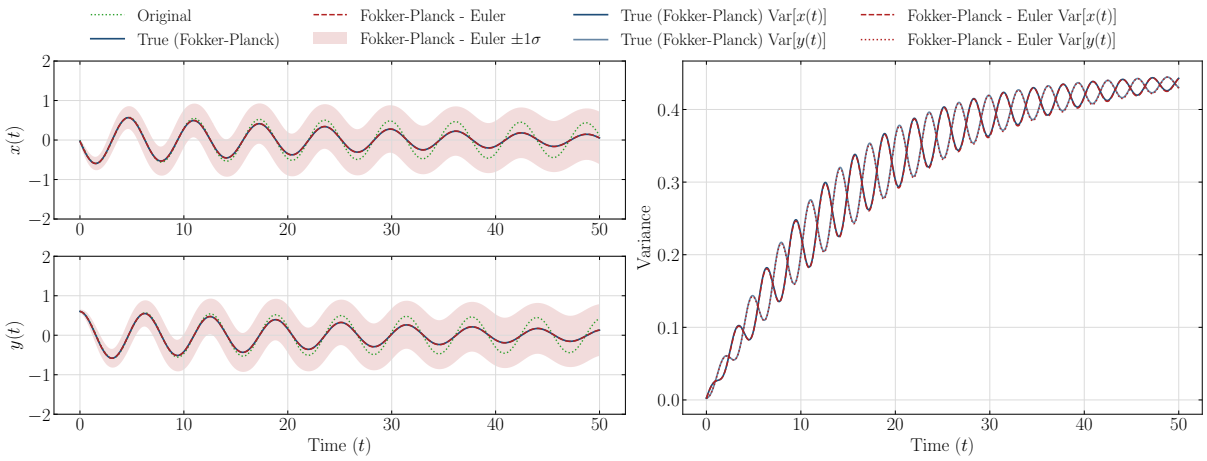


Figure 7.11: Trajectory comparison for the reference versus integrated Fokker–Planck solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 64×64 grid, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time.

The phase-space trajectory comparison shown in Fig. 7.12 further confirms the improved consistency between the reference and integrated solutions at higher spatial resolution.

Finally, we consider a 128×128 phase-space discretisation. The trajectory comparison is shown in Fig. 7.13. At this higher resolution, the numerical diffusion associated with the upwinding discretisation is further reduced, allowing the integrated Fokker–Planck solution to more accurately capture the transient oscillatory behaviour of the original dynamical system. The variance evolution also remains closer to the reference solution over longer integration times.

Overall, the results demonstrate that the upwinding scheme remains stable even without explicitly added

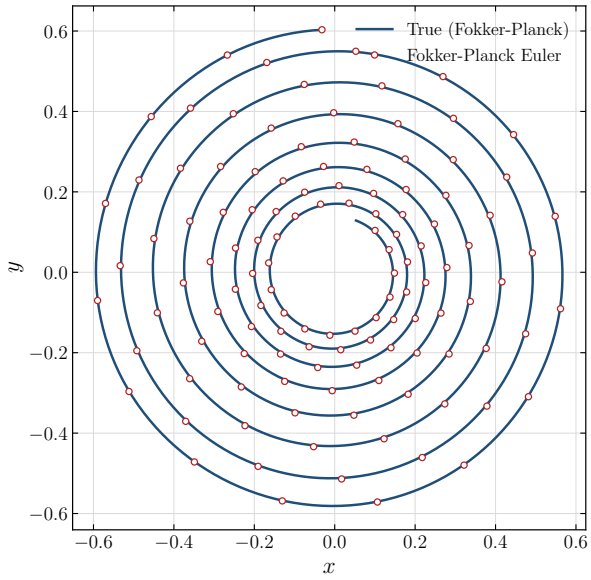


Figure 7.12: Phase space trajectory solution for the reference solution versus the integrated solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 64×64 grid.

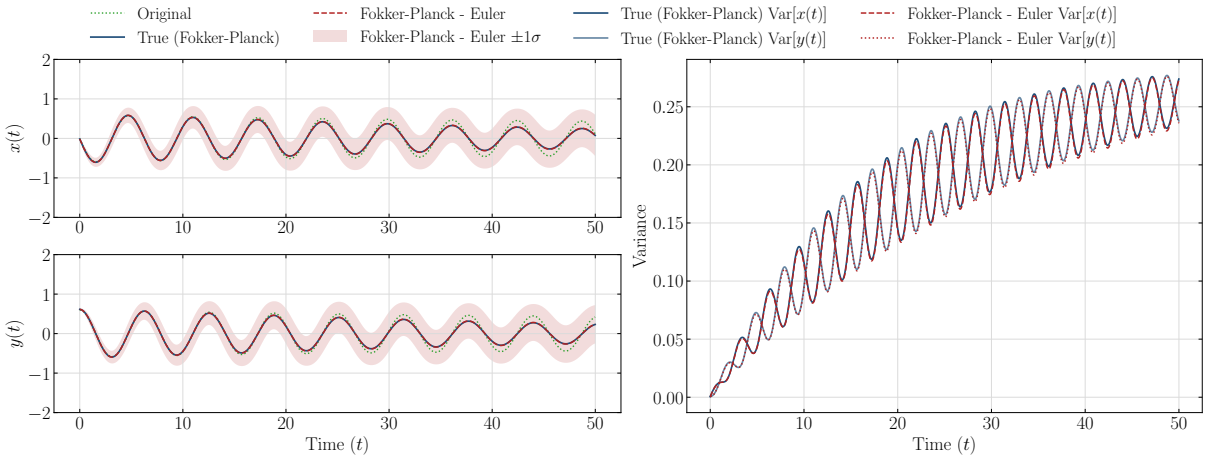


Figure 7.13: Trajectory comparison for the reference versus integrated Fokker–Planck solution obtained via the first-order Euler upwinding scheme with zero added diffusion on a 128×128 grid, against the original system equation solution. On the left, time-series plots for each variable are shown. On the right, the variance plot is shown over time.

diffusion due to its intrinsic numerical diffusion. However, this numerical diffusion introduces artificial damping that modifies the long-term dynamics. Increasing the grid resolution reduces the effective numerical diffusion and improves agreement with the original dynamical system, but results in parasitic oscillations (Fig. 7.13), highlighting the trade-off between numerical stability and dynamical accuracy in Fokker–Planck integration.

7.5.2 Lorenz-63 system

The main focus of this chapter is to extend the Fokker–Planck approach to chaotic differential equations, thereby enabling the use of quantum algorithms for the simulation of turbulent and chaotic dynamics.

As a representative test case, we consider the Lorenz–63 system introduced and analysed previously in Chap. 4.

The reference trajectory of the original nonlinear ODE system is obtained by integrating the Lorenz–63 equations forward in time using the fourth-order Runge–Kutta method with time step size $dt = 0.01$. We then reformulate the nonlinear dynamics into a linear Fokker–Planck partial differential equation and discretise the resulting operator using the upwinding schemes introduced in Sec. 7.2. The computational domain is taken as

$$[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}] = [-30, 30] \times [-40, 40] \times [-10, 60].$$

For the three-dimensional Lorenz system, direct matrix exponentiation of the Fokker–Planck operator becomes computationally prohibitive due to the larger grid size. Therefore, as a reference solution, we instead compare against the stochastic differential equation formulation integrated using the Euler–Maruyama method implemented in the `BrainPy` (Wang et al. 2023) library with time step size $\Delta t_{SDE} = 10^{-4}$ and an ensemble of 5000 particles. In Tab. 7.3, we summarise the numerical parameters and discretisation schemes for the utilised configurations.

Table 7.3: Input parameters used for the Lorenz–63 Fokker–Planck simulations.

Parameter	Symbol / Setting	Upwinding	Second-order upwinding
Numerical scheme	–	Upwind	Second-order upwind
Boundary condition	–	Reflective	Reflective
Grid points	$N_x \times N_y \times N_z$	$100 \times 100 \times 100$	$100 \times 100 \times 100$
Added diffusion coefficient	$\mathcal{D}$	$(0.5, 0.5, 0.5)$	$(0.5, 0.5, 0.5)$
Initial deterministic state	$\boldsymbol{x}_{init}$	$(7.4325, 10.0207, 29.6230)$	$(7.4325, 10.0207, 29.6230)$
SDE time step	Δt_{SDE}	10^{-4}	10^{-4}
Forward-Euler FP time step	Δt_{FP}	10^{-5}	10^{-6}

First-order upwinding scheme

For the Fokker–Planck integration of the Lorenz–63 system, we first use a first-order upwind discretisation of the drift operator together with forward Euler time integration. We use a grid size of $100 \times 100 \times 100$, a time step $\Delta t_{\mathrm{FP}} = 10^{-5}$, and added diffusion coefficients $\mathcal{D}^{(xx)} = \mathcal{D}^{(yy)} = \mathcal{D}^{(zz)} = 0.5$. Reflective boundary conditions are imposed, and the initial probability density is chosen as a localised delta distribution.

The resulting mean trajectories are shown in Fig. 7.14. Compared with the original deterministic Lorenz–63 solution and the reference SDE solution, the Fokker–Planck trajectories decay rapidly towards the stationary distribution. This behaviour is caused by excessive damping from the total effective diffusion, $\mathcal{D}_{\mathrm{eff}} = \mathcal{D} + \mathcal{D}_{\mathrm{num}}$, where the numerical diffusion introduced by the first-order upwind discretisation dominates the effective diffusion for the present grid configuration. We estimate that the numerical contribution can be as large as $O(10^2)$, leading to an over-smoothed probability density.

The overdamping and smearing of the attractor are further observed in the projected probability density functions in the (x, y), (x, z), and (y, z) planes, shown in Figs. 7.15–7.17. The comparison is made

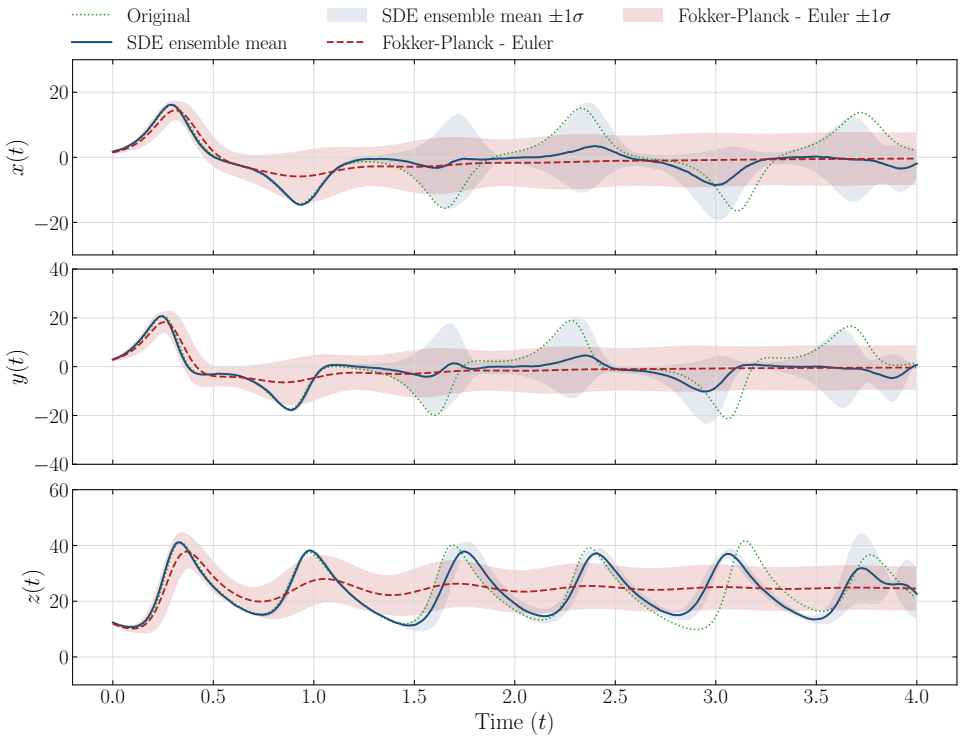


Figure 7.14: Trajectory comparison for the SDE solution versus the integrated Fokker–Planck solution of Lorenz–63 obtained using the first-order Euler upwinding scheme with added diffusion coefficients $\mathcal{D}^{(xx)} = \mathcal{D}^{(yy)} = \mathcal{D}^{(zz)} = 0.5$ on a $100 \times 100 \times 100$ grid, compared against the original deterministic system. Time-series plots for each variable are shown together with the ensemble variances.

against the reference SDE particle solution obtained using the same diffusion configuration. The first-order upwind scheme smooths the attractor structure and produces the largest signed errors near the high-density regions of the attractor. This smoothing is also visible in the one-dimensional marginal PDFs shown in Figs. 7.18 and 7.19, where the Fokker–Planck solution becomes broader and less sharply concentrated than the SDE reference.

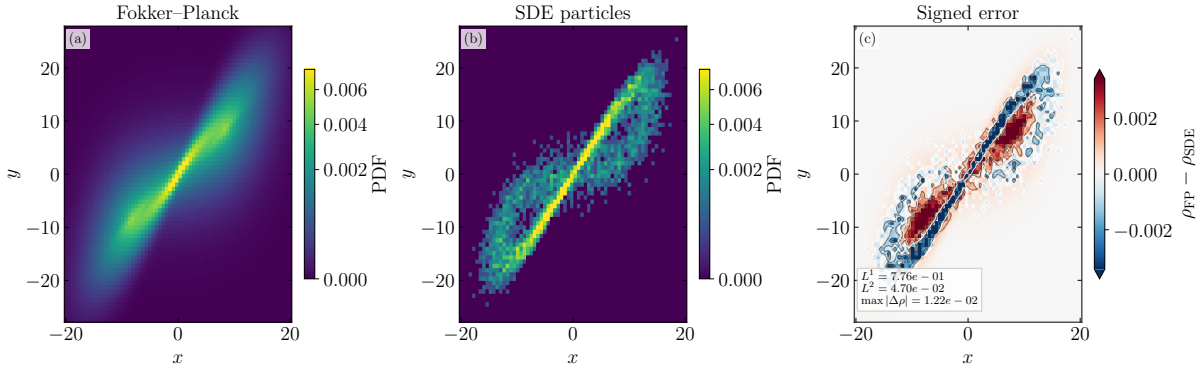


Figure 7.15: Projected probability density comparison in the (x, y) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown.

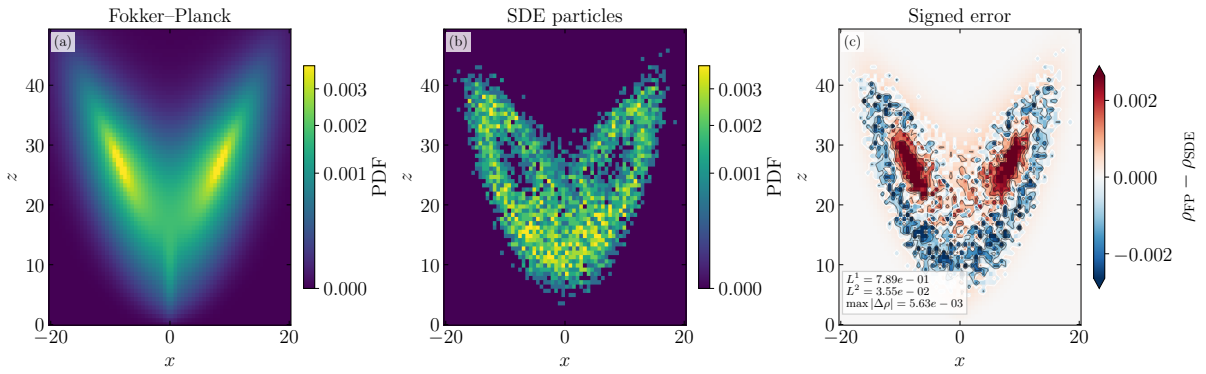


Figure 7.16: Projected probability density comparison in the (x, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown.

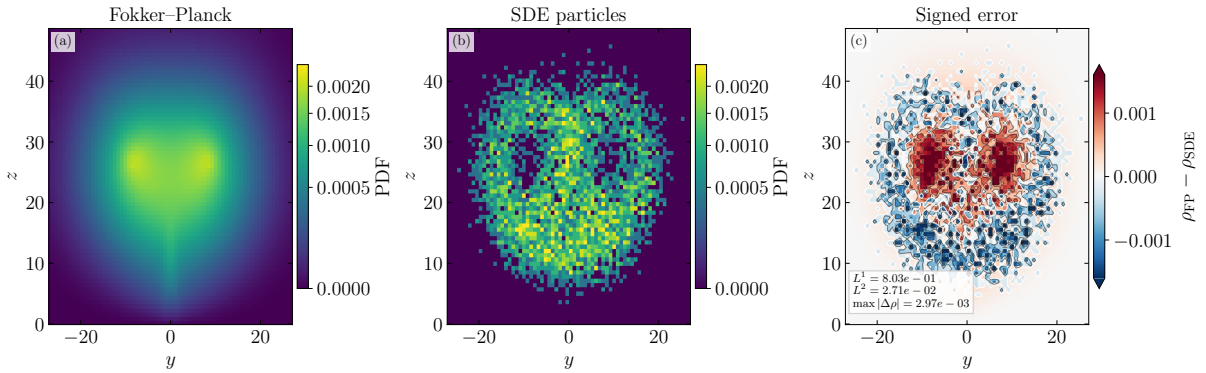


Figure 7.17: Projected probability density comparison in the (y, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown.

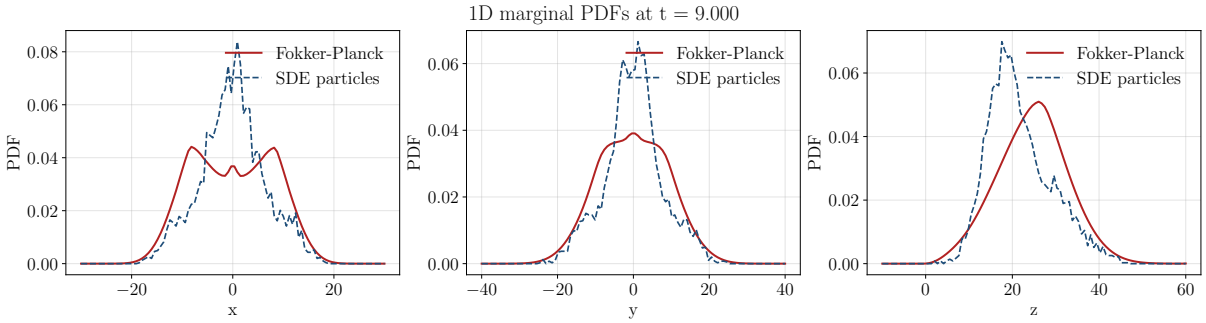


Figure 7.18: Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.000$, obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme and the reference SDE particle solution.

Second-order upwinding scheme

To reduce the overdamping observed with the first-order scheme and improve short-time predictability, we next implement a second-order upwinding scheme. As discussed in Sec. 7.3, the leading-order numerical

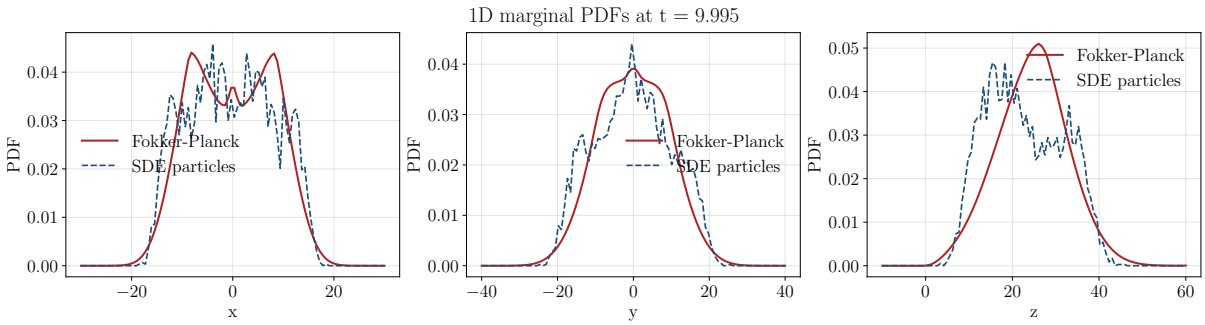


Figure 7.19: Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.995$, obtained from the integrated Fokker–Planck equation using the first-order Euler upwinding scheme and the reference SDE particle solution.

diffusion of the second-order upwind scheme no longer scales in the same way with the spatial grid spacing as in the first-order scheme. Instead, the dominant dissipative contribution is controlled mainly by the temporal discretisation. This makes the method less diffusive for the same spatial resolution.

The resulting time-series trajectories are shown in Fig. 7.20. Compared with the first-order upwind scheme, the second-order scheme preserves the transient dynamics more accurately and remains close to the deterministic and SDE reference trajectories for approximately two to three Lyapunov times. After this time, the chaotic sensitivity to initial conditions causes the mean trajectory to diverge, as expected for the Lorenz–63 system. However, unlike the first-order scheme, the solution does not immediately collapse to an overdamped stationary state.

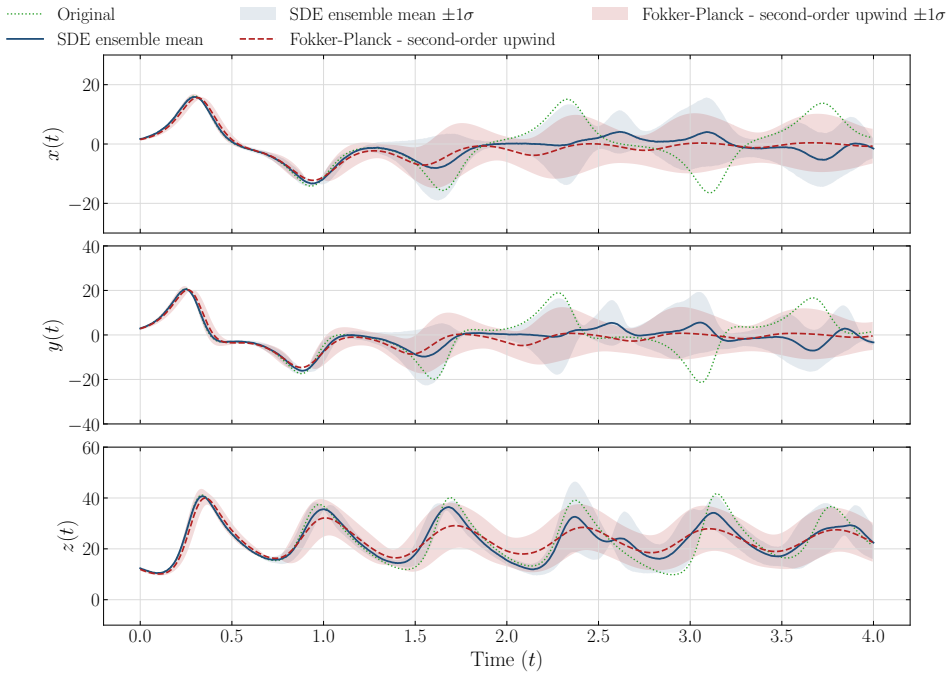


Figure 7.20: Trajectory comparison for the SDE solution versus the integrated Fokker–Planck solution of Lorenz–63 obtained using the second-order Euler upwinding scheme with added diffusion coefficients $\mathcal{D}^{(xx)} = \mathcal{D}^{(yy)} = \mathcal{D}^{(zz)} = 0.5$ on a $100 \times 100 \times 100$ grid, compared against the original deterministic system. Time-series plots for each variable are shown together with the ensemble variances.

The projected PDFs shown in Figs. 7.21–7.23 confirm that the second-order upwind scheme preserves the geometry of the Lorenz attractor more accurately than the first-order scheme. The two lobes of the attractor remain clearly visible, and the signed-error plots show a reduced smoothing bias compared with the first-order discretisation. The remaining differences are mainly concentrated around regions where the density changes rapidly, which is expected because these regions are most sensitive to both numerical diffusion and finite grid resolution.

The one-dimensional marginal distributions in Figs. 7.24 and 7.25 also show improved agreement with the SDE reference solution. At both $t = 9.000$ and $t = 9.995$, the Fokker–Planck marginals reproduce the main peaks and widths of the particle-based distributions more accurately than the first-order upwind scheme. This indicates that the second-order scheme provides a better balance between stability and accuracy for the Lorenz–63 Fokker–Planck integration.

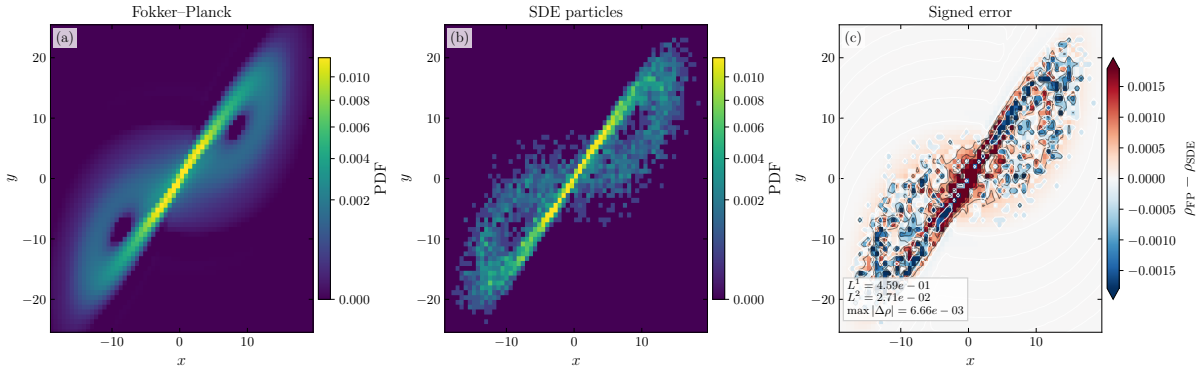


Figure 7.21: Projected probability density comparison in the (x, y) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown.

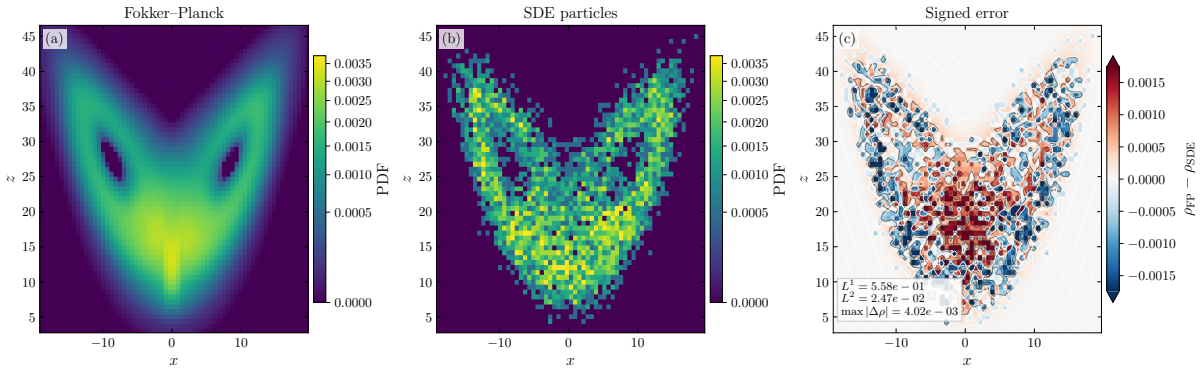


Figure 7.22: Projected probability density comparison in the (x, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown.

To summarise, the first-order upwind scheme provides a stable Fokker–Planck integration but introduces excessive numerical diffusion. For the Lorenz–63 system, this leads to rapid overdamping of the mean

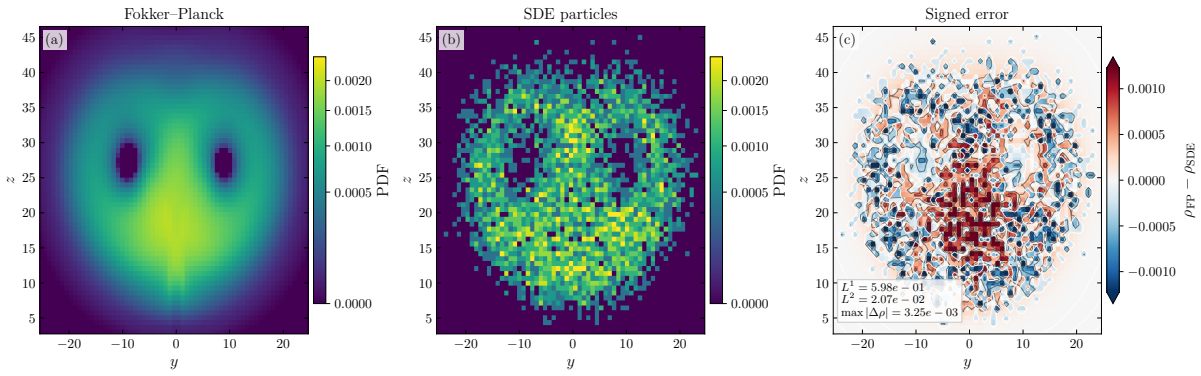


Figure 7.23: Projected probability density comparison in the (y, z) plane for the Lorenz–63 system obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme, compared against the reference SDE particle solution. The signed error between the two solutions is also shown.

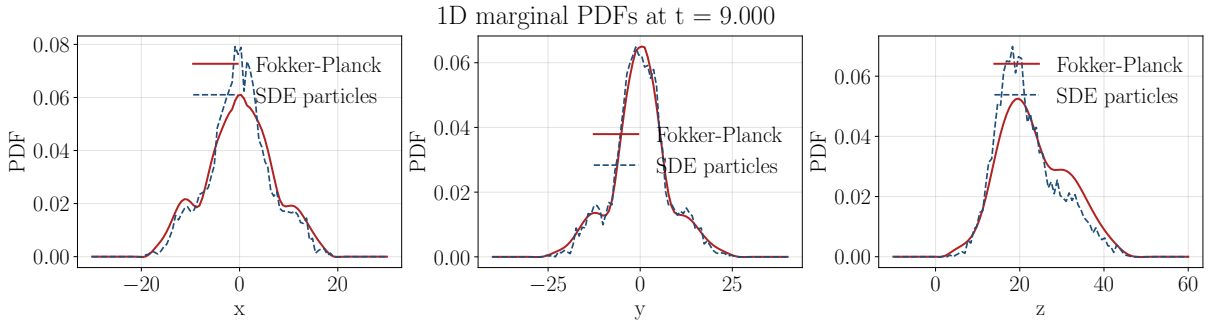


Figure 7.24: Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.000$, obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme and the reference SDE particle solution.

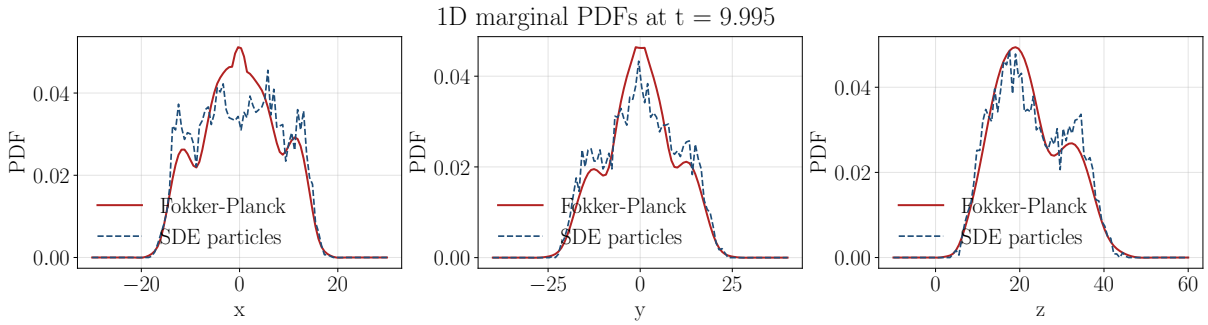


Figure 7.25: Comparison of the one-dimensional marginal probability density functions $P(x)$, $P(y)$, and $P(z)$ for the Lorenz–63 system at $t = 9.995$, obtained from the integrated Fokker–Planck equation using the second-order Euler upwinding scheme and the reference SDE particle solution.

trajectory and excessive smoothing of the probability density on the attractor. Although this stabilises the integration, it also destroys important dynamical features of the chaotic system.

In contrast, the second-order upwind scheme significantly reduces the artificial diffusion and preserves

both the transient trajectory statistics and the attractor geometry more accurately. The results show that the second-order scheme provides a better approximation to the reference SDE solution, particularly in the projected PDFs and one-dimensional marginal distributions. Therefore, for chaotic systems such as Lorenz–63, higher-order upwinding gives a more suitable compromise between numerical stability and physical accuracy.

7.6 Integrating nonlinear dynamics with the Koopman–von Neumann approach

Before presenting the numerical results for the quantum algorithms used to integrate the Fokker–Planck equation, we first test the alternative Koopman–von Neumann (KvN) approach introduced in Sec. 7.1.2. We consider the same two-dimensional damped harmonic oscillator analysed previously using the Fokker–Planck formulation in Sec. 7.5.

The purpose of this section is to investigate whether the KvN formulation avoids some of the numerical issues observed in the Fokker–Planck approach. In particular, the KvN method evolves a complex-valued state vector ψ rather than directly evolving the probability density ρ. The probability density is then recovered as $\rho = |\psi|^2$. This avoids directly imposing probability-conservation constraints on ρ during the time integration. The simulation setup follows Sec. 7.5, and the parameters are summarised in Tab. 7.4.

Table 7.4: Input parameters used for the damped harmonic oscillator, Koopman–von Neumann simulation.

Parameter	Symbol / Setting	Value
Numerical scheme	–	Central difference
Boundary condition	–	Reflective
Grid points	$N_x \times N_y$	32×32
Computational domain	$[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$	$[-2, 2] \times [-2, 2]$
Added diffusion coefficient	$\mathcal{D}$	0.00
Model parameter	γ	0.1
Initial condition	(x_0, y_0)	$(0, 0.6)$
Forward-Euler KvN time step	Δt_{KvN}	10^{-5}

For the KvN simulation, we discretise the Liouville-type KvN operator using a symmetric central-difference approximation and no added diffusion. The state vector ψ is evolved directly using a first-order forward-Euler method, after which the probability density is reconstructed from $\rho = |\psi|^2$. The results are shown in Figs. 7.26 and 7.27. Figure 7.26 shows the evolution over the short-time interval $t \in [0, 1]$, while Fig. 7.27 shows the longer-time behaviour over $t \in [1, 40]$.

Although the KvN method evolves the wavefunction rather than the probability density directly, the central-difference discretisation without diffusion still leads to spurious oscillations. These oscillations are caused by the lack of numerical dissipation in the discretised transport operator and are consistent with the instability behaviour discussed in (Novikau and Joseph 2025).

The comparison between the reference solution, the integrated KvN solution, and the original system trajectory is shown in Fig. 7.28. The time-series plots show that the reconstructed mean trajectory becomes increasingly damped over time due to the interaction between the non-diffusive central-difference

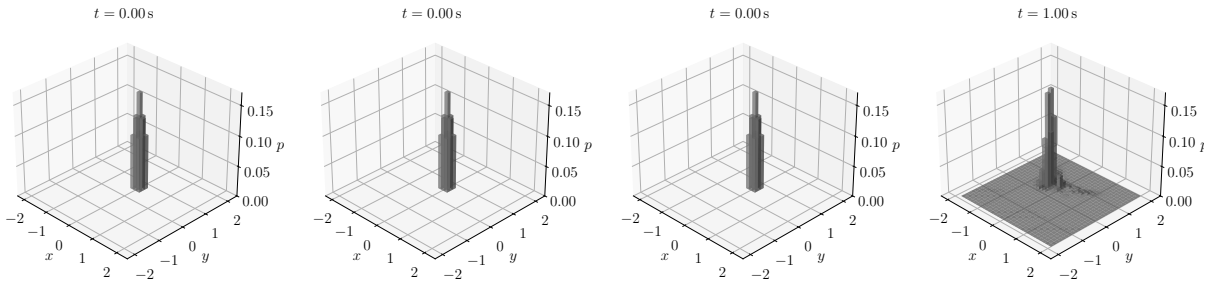


Figure 7.26: Integrated phase-space solution of the two-dimensional damped harmonic oscillator obtained using the Koopman–von Neumann formulation with a first-order forward-Euler central-difference scheme and no added diffusion. The probability density is reconstructed from $\rho = |\psi|^2$. The evolution is shown over the short-time interval $t \in [0, 1]$.

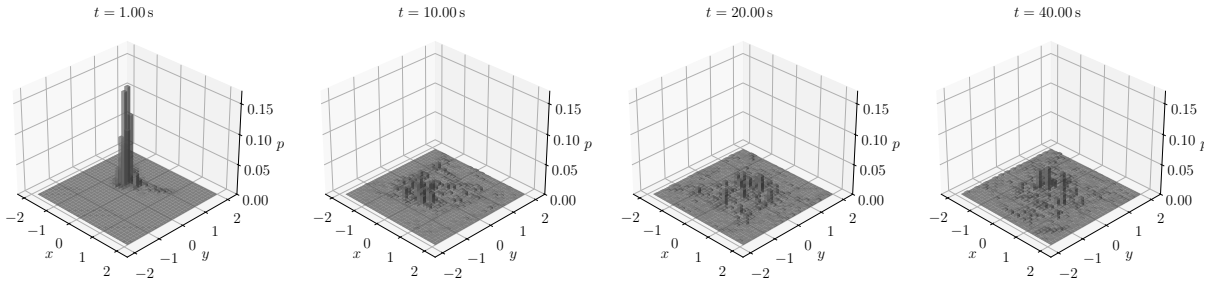


Figure 7.27: Integrated phase-space solution of the two-dimensional damped harmonic oscillator obtained using the Koopman–von Neumann formulation with a first-order forward-Euler central-difference scheme and no added diffusion. The probability density is reconstructed from $\rho = |\psi|^2$. The evolution is shown over the longer-time interval $t \in [1, 40]$, where spurious oscillations become more visible.

discretisation and the reconstruction of the probability density from $\rho = |\psi|^2$.

These results show that, although the KvN formulation provides a useful Hilbert-space representation of nonlinear classical dynamics, it does not by itself resolve the numerical issues associated with discretising transport-dominated dynamics. In the absence of added diffusion, stabilised discretisation, or more suitable time-integration schemes, the KvN approach still exhibits spurious oscillations, artificial damping, and long-time numerical degradation. Therefore, the KvN formulation does not automatically remove the numerical instability associated with discretising the Fokker–Planck operator.

7.7 Block-encoding and hardware implementation

The approaches discussed throughout this chapter provide a pathway for reformulating nonlinear differential equations into linear operator representations. However, these formulations generally suffer from the curse of dimensionality, requiring careful numerical discretisation and stabilisation strategies for efficient classical emulation. We now motivate the use of quantum algorithms for simulating these high-dimensional linearised systems more efficiently than classical approaches.

In particular, we apply the unitary block-encoding method introduced in Sec. 7.4 for the forward-Euler integration of the linearised Fokker–Planck equation. As a prototypical example, we consider the one-

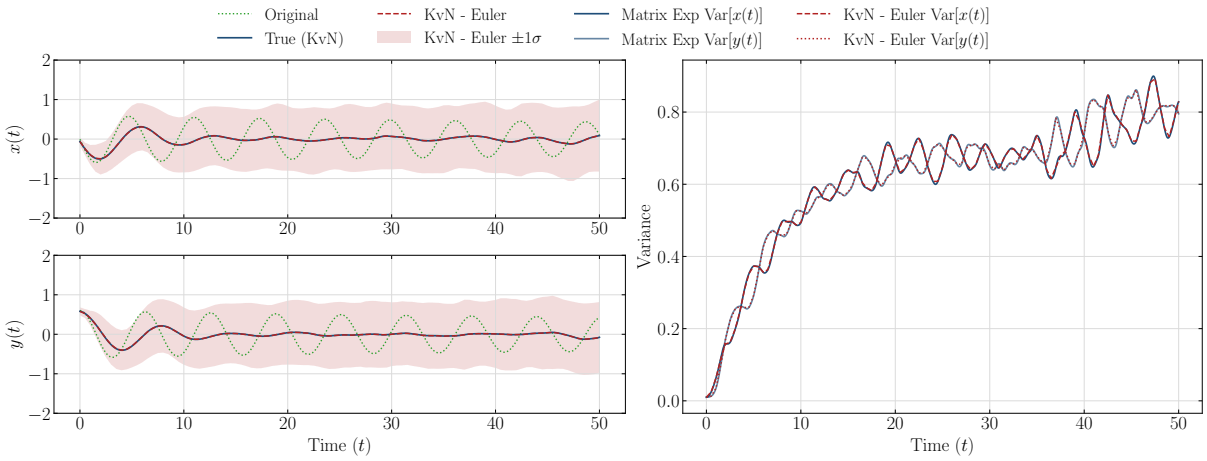


Figure 7.28: Trajectory comparison between the reference Koopman–von Neumann solution, the forward-Euler Koopman–von Neumann solution, and the original system equation solution. The left panels show the time-series comparison for each variable, while the right panel shows the corresponding variance over time.

dimensional nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{x} - \kappa \boldsymbol{x}^3, \tag{7.60}$$

with $\kappa = 0.5$. The Fokker–Planck operator is discretised using a central-difference scheme with reflective boundary conditions. The simulation parameters used for the block-encoding implementation are summarised in Tab. 7.5.

Table 7.5: Input parameters used for the one-dimensional Fokker–Planck block-encoding simulation.

Parameter	Symbol / Setting	Value
Numerical scheme	–	Central difference
Boundary condition	–	Reflective
Grid points	N_x	16
Computational domain	$[x_{\min}, x_{\max}]$	$[-2, 2]$
Added diffusion coefficient	$\mathcal{D}$	0.15
Model parameter	κ	0.5
Initial distribution	–	Normalized delta
Forward-Euler time step	Δt_{FE}	10^{-2}
Block-encoding time step	Δt_{BE}	10^{-2}
Measurement shots	–	1000

As discussed in Sec. 7.4, we use block encoding to represent the discrete one-dimensional Fokker–Planck propagator. We first define the forward-Euler evolution operator

$$\boldsymbol{A}_0 = \boldsymbol{I} + \Delta t_{\mathrm{BE}} \boldsymbol{\mathcal{R}}, \tag{7.61}$$

where $\boldsymbol{\mathcal{R}}$ is the finite-difference Fokker–Planck operator and $\Delta t_{\mathrm{BE}} = 10^{-2}$. To ensure that the operator is a contraction, we normalise it as

$$\boldsymbol{A} = \frac{\boldsymbol{A}_0}{\alpha}, \qquad \alpha = \|\boldsymbol{A}_0\|_2. \tag{7.62}$$

For the spatial discretisation $N_x = 16$, the normalised propagator $\boldsymbol{A} \in \mathbb{R}^{16\times16}$ was embedded into a unitary matrix $\boldsymbol{U} \in \mathbb{C}^{32\times32}$. This required 4 data qubits to encode the spatial probability vector together with one ancilla qubit, giving a total register size of 5 qubits.

The block-encoded unitary was constructed as

$$\boldsymbol{U} = \begin{pmatrix} \sqrt{\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A}} & \boldsymbol{A}^\dagger \\ \boldsymbol{A} & -\sqrt{\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^\dagger} \end{pmatrix}. \tag{7.63}$$

The time evolution is simulated by repeatedly applying the block-encoded unitary and postselecting the ancilla in the state $|1\rangle$. Conditional on successful postselection, the data register is proportional to $\boldsymbol{A}|p(t)\rangle$. A total of 30 block-encoding steps are applied. We also perform a finite-shot implementation using 1000 measurement shots to investigate the effects of sampling noise present in hardware implementations. After compilation, the circuit acts on 5 qubits and contains 1786 gates, with a circuit depth of 1291. The decomposition consists of 1244 single-qubit gates and 542 two-qubit CNOT gates.

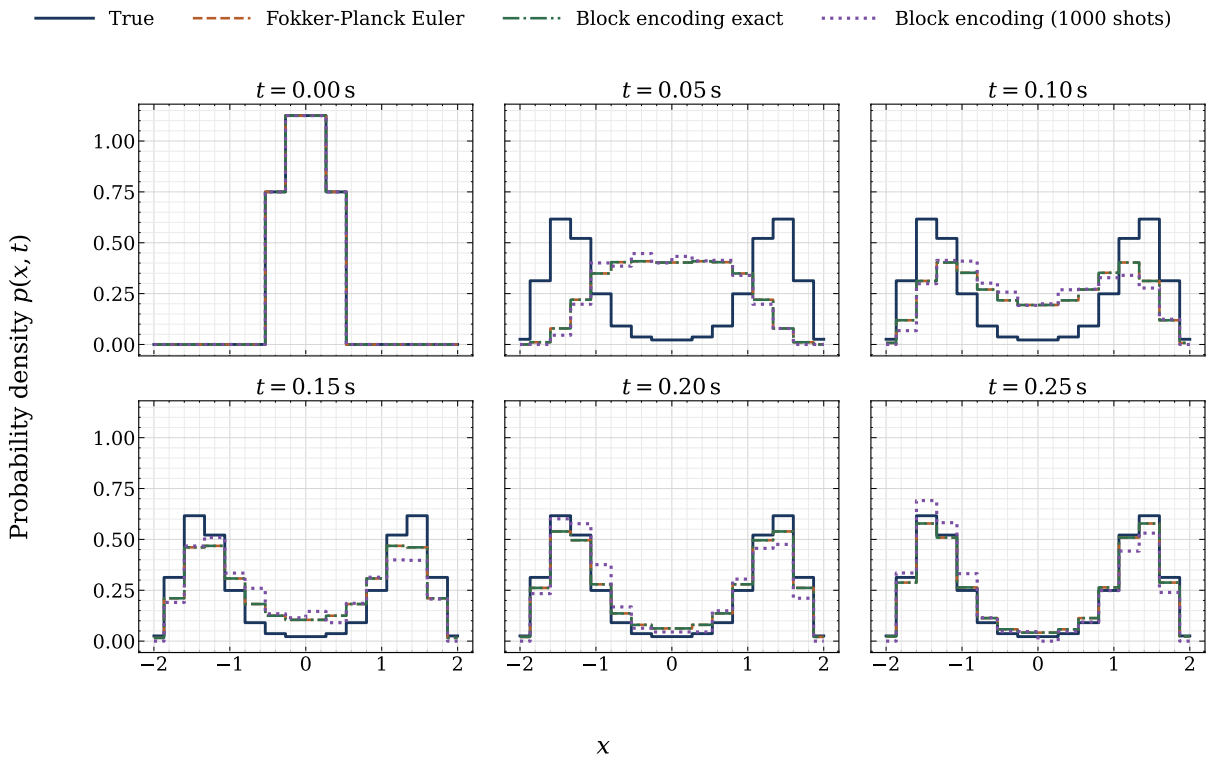


Figure 7.29: Comparison of the probability-density evolution obtained from the reference matrix-exponential solution, the classical forward-Euler integration, and the quantum block-encoding implementation. The exact block-encoding solution closely follows the classical evolution, while the 1000-shot reconstruction reproduces the same qualitative behaviour with small sampling fluctuations.

The resulting probability-density evolution is shown in Fig. 7.29. The exact state-vector block-encoding solution closely follows both the classical forward-Euler and reference matrix-exponential solutions throughout the integration. The probability density evolves away from the unstable fixed point at $x = 0$ and gradually accumulates around the two stable regions, reproducing the expected bistable behaviour of

the system. The shot-based reconstruction using 1000 measurements captures the same overall dynamics, with only small statistical fluctuations relative to the exact state-vector evolution. These results show that the block-encoding implementation successfully reproduces the dynamics of the discretised Fokker–Planck equation while remaining compatible with a unitary quantum evolution.

7.7.1 Hardware implementation

We demonstrate the implementation of the Fokker–Planck solver for the one-dimensional nonlinear system analysed in Sec. 7.7 on several quantum hardware platforms. To mitigate hardware coherence and circuit-depth limitations, we reduce the spatial discretisation from 16 to 8 grid points. A summary of the hardware configurations is given in Tab. 7.6.

Table 7.6: Hardware configurations used for the one-dimensional Fokker–Planck block-encoding experiments. All circuits use four qubits with $\Delta t = 0.01$ and 8 grid points. Gate counts are rounded per-circuit averages over the submitted time-step runs.

QPU	Device summary	Shots/run	Total runs	$\bar{p}_{\mathrm{succ}}$	$p_{\mathrm{succ}}^{\mathrm{min}}$	Single-qubit gates	Two-qubit gates	Total gates
IonQ Forte-3	36-qubit trapped-ion	100	20	0.6500	0.5100	1327	69	1400
IQM Emerald	54-qubit superconducting transmon	2000	9	0.4879	0.4650	556	181	741
Rigetti Ankaa-3	84-qubit superconducting	4000	15	0.6675	0.4753	1345	221	1570
IQM Emerald	54-qubit superconducting transmon	2000	20	0.4834	0.4610	557	182	743

The results from the hardware runs are shown in Figs. 7.30–7.33. In all cases, the hardware-reconstructed distributions diverge from the classical reference to varying degrees, with the extent of degradation reflecting each device's noise characteristics, shot budget, and circuit depth.

First, the Rigetti Ankaa-3 (Fig. 7.31) shows the most agreement with the reference solution. The two-peak structure of the evolving probability density remains visibly intact through $t = 0.09$ s, and at $t = 0.15$ s, the hardware distribution still captures the qualitative features of the true solution, including the approximate relative bin magnitudes and positions, despite amplitude errors and minor distortions throughout.

The IonQ Forte-3 (Fig. 7.30) shows initial agreement at the earliest timestep but diverges progressively from $t = 0.04$ s onward. Given the shot budget of only 100 shots, a substantial portion of this error is expected to be dominated by shot noise rather than gate errors.

The IQM Emerald results (Figs. 7.32–7.33) exhibit the most severe degradation. From as early as $t = 0.01$ s, both runs collapse to a nearly uniform probability density of approximately 0.20–0.22 across all grid points. This behaviour is consistent with the measured output approaching that of a maximally mixed state due to accumulated decoherence, gate, and readout errors.

In summary, these results indicate that, even for this simplified one-dimensional model, hardware noise dominates the evolution across all tested platforms. For a limited number of time steps and specific hardware configurations, the block-encoding implementation reproduces qualitative features of the probability transport dynamics of the discretised Fokker–Planck equation on current noisy quantum hardware. However, for a longer time horizon, accumulated decoherence, gate errors, and finite sampling increasingly dominate the reconstructed distributions.

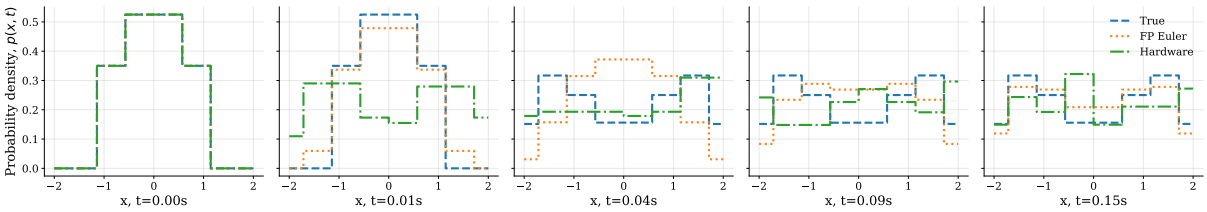


Figure 7.30: Probability-density evolution obtained from the IonQ Forte-3 trapped-ion quantum processor for the one-dimensional Fokker–Planck block-encoding implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler Fokker–Planck integration at different times.

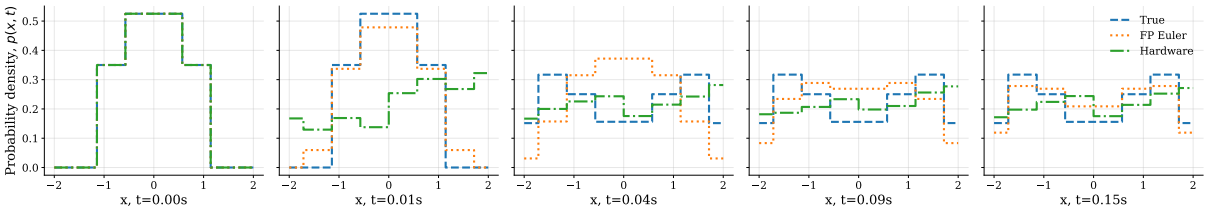


Figure 7.31: Probability-density evolution obtained from the Rigetti Ankaa-3 superconducting quantum processor for the one-dimensional Fokker–Planck block-encoding implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler Fokker–Planck integration at different times.

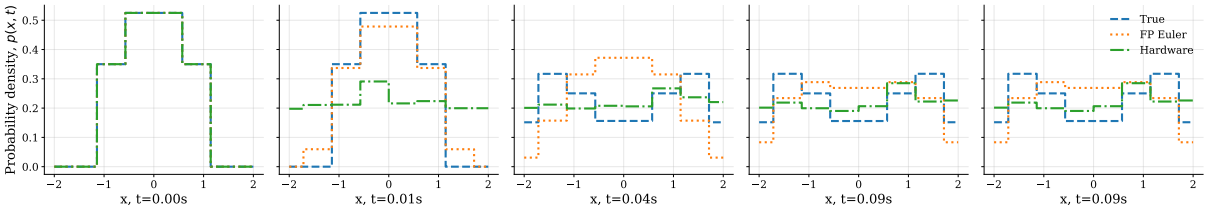


Figure 7.32: Probability-density evolution obtained from the IQM Emerald superconducting transmon processor using 9 block-encoding integration steps for the one-dimensional Fokker–Planck implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler integration.

7.8 Conclusion

In this chapter, we investigated alternative approaches for solving nonlinear differential equations governing chaotic dynamics on quantum computers, in contrast to the data-driven quantum reservoir computing methods discussed previously. In particular, we focused on linear embedding formulations of nonlinear dynamics through the Fokker–Planck and Koopman–von Neumann (KvN) approaches. The main objective of this chapter is to extend the Fokker–Planck embedding framework of Tennie and Magri 2024 to nonlinear chaotic systems, while systematically analysing the role of numerical discretisation, diffusion, and quantum implementation.

We first introduced the theoretical description of the Fokker–Planck and KvN formulations as linear

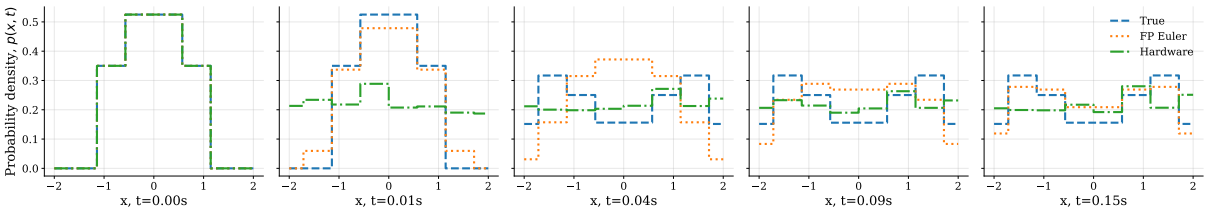


Figure 7.33: Probability-density evolution obtained from the IQM Emerald superconducting transmon processor using 20 block-encoding integration steps for the one-dimensional Fokker–Planck implementation with 8 grid points. The hardware reconstruction is compared against the exact reference solution and the classical forward-Euler integration.

operator embeddings of nonlinear dynamical systems. The Fokker–Planck equation reformulates nonlinear stochastic dynamics into a linear evolution equation acting on probability densities, while the KvN approach provides a Schrödinger-like Hilbert-space representation of deterministic classical dynamics. Although both formulations allow nonlinear systems to be represented through linear operators, we showed that the KvN formulation suffers from parasitic oscillations in the absence of explicit diffusion, motivating the primary focus on the Fokker–Planck formulation throughout this chapter.

We then introduced central difference and upwinding discretisation schemes for the Fokker–Planck operator and analysed their associated numerical diffusion. The central difference scheme provides minimal numerical diffusion but requires additional physical diffusion for stability, while the upwinding schemes introduce intrinsic numerical diffusion that stabilises the evolution at the cost of additional damping. Using modified-equation analysis, we derived effective numerical diffusion coefficients for different schemes and related them to the Lyapunov time scales of the underlying chaotic dynamics. This led to a diffusion constraint linking the effective diffusion, grid resolution, and maximum Lyapunov exponent, which provides a physical interpretation of the trade-off between numerical stability and the ability to resolve fine-scale chaotic structures.

The numerical results for the nonlinear damped harmonic oscillator demonstrated that both the central difference and upwinding schemes reproduce the reference Fokker–Planck evolution over short integration times. However, both explicitly added diffusion and intrinsic numerical diffusion lead to overdamped long-term behaviour relative to the original nonlinear dynamical system. Increasing the grid resolution reduced the effective numerical diffusion and improved agreement with the underlying dynamics, although higher resolutions also introduced parasitic oscillations. These results highlight the balance between stability, diffusion, and dynamical accuracy in Fokker–Planck integration.

We then extended the Fokker–Planck embedding approach to the chaotic Lorenz–63 system, demonstrating that nonlinear chaotic dynamics can be reformulated into sparse linear operator evolutions suitable for quantum implementation. The results showed that the numerical diffusion introduced by the discretisation schemes strongly influences the preservation of the fine-scale structure of the chaotic attractor. At the same time, the three-dimensional Fokker–Planck operator becomes computationally expensive at high resolution, motivating sparse operator formulations and quantum implementations for larger nonlinear systems. We further showed that higher-order upwind schemes better preserve the fine-scale structure of the chaotic attractor by reducing excessive numerical diffusion, thereby motivating the development of

quantum algorithms capable of efficiently implementing higher-order discretisation schemes.

Finally, we applied the unitary block-encoding framework to implement the non-unitary forward Euler evolution (Tennie and Magri 2024) of the discretised Fokker–Planck equation on quantum computers. This construction embeds the non-unitary Fokker–Planck evolution into a larger unitary operator acting on an ancilla-extended Hilbert space. We demonstrated this approach for one-dimensional systems, including finite-shot simulations and quantum hardware implementation. The results showed that repeated post-selected evolution can reproduce the forward Euler dynamics, while the success probability decreases with increasing integration time, limiting the achievable simulation horizon on near-term quantum hardware.

In summary, this chapter demonstrates that the Fokker–Planck embedding provides a physically interpretable and quantum-compatible framework for reformulating nonlinear and chaotic differential equations into linear operator evolutions. The results highlight both the opportunities and limitations of this approach on near-term quantum hardware, particularly the trade-offs between numerical diffusion, stability, spatial resolution, and post-selection success probability. Future work could investigate higher-order integration methods, adaptive discretisation strategies, and more advanced quantum algorithms, including linear combination of unitaries, quantum signal processing, and Schrödingerisation, for scalable quantum implementations of nonlinear dynamical systems. Overall, this framework opens opportunities for applications in turbulent fluid dynamics, stochastic climate systems, and probabilistic forecasting problems.

Chapter
8

Conclusions and future work

Forecasting and simulation of nonlinear dynamical systems remain fundamentally challenging due to sensitivity to initial conditions, multiscale dynamics across space and time, and the computational cost associated with accurately resolving high-dimensional chaotic systems. Moreover, many physical systems of practical and scientific interest, including turbulent and thermoacoustic flows, weather and climate dynamics, and complex industrial processes, are governed by the same nonlinear mechanisms that give rise to these computational bottlenecks. Classical numerical methods and data-driven approaches, while powerful, face fundamental limitations in scalability, long-term stability, and computational cost when applied to high-dimensional chaotic systems. This thesis demonstrates that quantum computational approaches provide a physically interpretable and scalable framework for modelling, forecasting, and analysing chaotic dynamical systems.

8.1 Conclusions

The main contributions of this thesis can be broadly categorised into two areas: (a) quantum reservoir computing (QRC) methods for data-driven prediction and stability analysis of chaotic dynamical systems, and (b) quantum-compatible formulations for directly solving nonlinear differential equations through linear embedding approaches. Together, these results establish quantum computing as a viable framework for forecasting, analysing, and solving nonlinear dynamical systems through physically interpretable and computationally scalable approaches.

8.1.1 Quantum reservoir computing: design, robustness, and forecasting

Chapter 2 reviews the forecasting and learnability problem for chaotic dynamical systems and introduces reservoir computing as a framework for such tasks. The theory of generalised synchronisation (GS) is then outlined, and reservoir computers are interpreted as GS systems. Lyapunov exponents, conditional

Lyapunov exponents, and learnability conditions are introduced as theoretical tools for characterising synchronisation, stability, and learnability within the QRC framework.

Chapter 3 establishes practical design principles for quantum reservoir computers using the framework of generalised synchronisation (GS). We analyse the stability of the driven reservoir dynamics through conditional Lyapunov exponents (CLEs), using analytical Jacobian formulations for both recurrent QRCs and RF-QRCs. The autonomous evolution of the trained reservoirs is then used to infer the invariant properties of chaotic systems. The relationships between GS, the echo state property (ESP), learnability, and stability are used to derive explicit criteria for the design of recurrent and recurrence-free quantum reservoir computers for learning and predicting chaotic systems. Collectively, these results establish GS as a framework for analysing the stability and learnability of quantum reservoir computers for chaotic systems. Furthermore, we show that RF-QRCs naturally satisfy $ESP = GS$ conditions by design, while noise and dissipation promote GS in unitary quantum feature maps, providing practical mechanisms for satisfying the ESP in QRCs.

Chapter 4 presents quantum reservoir computing architectures for forecasting chaotic dynamical systems and extreme events. In particular, we show that RF-QRC retains memory through leaky integration rather than explicit quantum recurrence, allowing the circuit depth to scale linearly with the input dimension while avoiding feedback of the exponentially large reservoir state at each time step. Through numerical experiments on the Lorenz-63 and Lorenz-96 systems, we analyse the effects of reservoir size, feature maps, and entanglement structure on predictive performance. We show that both QRCs and RF-QRCs can accurately predict chaotic dynamics, recover long-term statistical distributions, and infer invariant properties of chaotic attractors, including Lyapunov spectra, attractor dimensions, and covariant Lyapunov vectors. Furthermore, we show that RF-QRC outperforms classical reservoir computers in high-dimensional forecasting and extreme-event prediction tasks, establishing RF-QRC as a scalable framework for forecasting and analysing chaotic dynamical systems.

Chapter 5 investigates the role of finite-sampling and hardware noise in quantum reservoir computing for chaotic time-series forecasting, parallel quantum reservoir implementations, and optimal training strategies. We show that the recurrence-free quantum reservoir computer (RF-QRC) naturally suppresses temporally uncorrelated sampling noise through its leaky-integrated architecture, leading to improved robustness compared to recurrent QRCs. We analyse the effects of finite-sampling noise on different reservoir architectures through signal-to-noise ratios and training performance, and propose denoising methods based on singular value decomposition (SVD) and signal filtering to mitigate noise in reservoir activations. Furthermore, signal filtering techniques are shown to provide a robust and scalable approach for suppressing noise in parallel RF-QRC settings, in contrast to computationally expensive denoising methods such as SVD. The chapter also demonstrates the deployment of RF-QRC on quantum hardware, where hybrid quantum-classical denoising approaches effectively suppress both sampling and hardware noise for practical forecasting applications on near-term quantum devices. Although the proposed denoising methods improve robustness on near-term hardware, practical implementations remain constrained by measurement overhead, hardware noise, and limited qubit resources on current quantum devices.

Chapter 6 develops a systematic analysis of different noise sources and hyperparameter dependence in quantum reservoir computers, revealing a fundamental trade-off between noise strength and mem-

ory. Building upon the theoretical framework developed in Chap. 2 and the practical design principles established in Chap. 3, we analyse the relationship between Lyapunov exponents and conditional Lyapunov exponents (CLEs) to establish practical criteria for robustness and the condition $GS = ESP$. We then investigate the effects of finite-sampling noise, depolarising channels, and amplitude damping, demonstrating that dissipation enhances contraction and improves the stability of reservoir dynamics. Furthermore, we show that recurrence-free quantum reservoir computers (RF-QRCs) retain the ability to accurately infer chaotic dynamics and recover long-term dynamical structure in noisy regimes, identifying regions of hyperparameter space where noise improves performance relative to the noise-free case. Collectively, these results establish noise as a resource for improving robustness and stability in quantum reservoir computing, showing that dissipation and stochasticity can enhance the stability of learnable quantum representations of chaotic systems.

8.1.2 Quantum computing for differential equations governing chaotic dynamics

Chapter 7 investigates quantum-compatible formulations for solving nonlinear differential equations governing chaotic dynamics through linear embedding approaches. In particular, we analyse the Fokker–Planck and Koopman–von Neumann (KvN) formulations as linear operator representations of nonlinear dynamical systems. We show that, although both formulations reformulate nonlinear dynamics into linear operator evolutions, the KvN formulation exhibits parasitic oscillations in the absence of explicit diffusion, motivating the primary focus on the Fokker–Planck framework. We then investigate central difference and upwinding discretisation schemes for the Fokker–Planck operator, analysing the role of numerical diffusion, damping, and stability through modified equation analysis. This leads to a diffusion constraint linking effective diffusion, grid resolution, and the maximum Lyapunov exponent, where we establish quantitative bounds on the numerical diffusion required for stable integration while preserving fine-scale chaotic structures over long-time chaotic evolution. Through applications to nonlinear damped oscillators and the Lorenz–63 system, we show that nonlinear chaotic dynamics can be reformulated into sparse linear operator evolutions suitable for quantum implementation, while also demonstrating that excessive numerical diffusion suppresses fine-scale attractor structure and leads to overdamped long-term dynamics. To address this, we develop higher-order discretisation strategies that better preserve fine-scale chaotic structures while maintaining quantum-compatible linear Fokker–Planck evolution. Finally, we implement a unitary block-encoding framework for discretised Fokker–Planck dynamics on quantum hardware, showing that repeated post-selected evolution reproduces the forward Euler integration of the Fokker–Planck equation, while the post-selection success probability decreases with increasing integration time. Collectively, these results establish Fokker–Planck embedding as a physically interpretable and quantum-compatible framework for reformulating nonlinear and chaotic differential equations into linear operator evolutions suitable for near-term quantum computation.

8.2 Outlook and future work

The results presented in this thesis demonstrate that quantum computational approaches provide physically interpretable and scalable frameworks for forecasting, analysing, and simulating chaotic dynamical systems. Rather than treating chaos purely as a limitation for prediction, the methods developed here show that chaotic dynamics can be exploited through synchronisation, contraction, and operator embed-

ding frameworks naturally compatible with quantum computation. Although significant theoretical and hardware challenges remain, including scalability, noise, and limited quantum resources, the approaches developed in this thesis establish practical and theoretical foundations for quantum-enhanced modelling of nonlinear dynamical systems on both near-term and future quantum computing platforms. Several important directions remain for future work:

Scaling toward large-scale spatiotemporal systems. To demonstrate the near-term utility of quantum computational methods and potentially address the computational bottlenecks of classical approaches, future work should extend QRC architectures toward operational weather and climate forecasting, turbulence, industrial simulation, scientific foundation models, and multiscale nonlinear dynamics. This includes scalable quantum-inspired reservoir architectures, latent-space QRC methods, probabilistic forecasting approaches, and physically constrained learning frameworks for high-dimensional dynamical systems.

Optimal quantum encoding and decoding strategies. For any practically useful implementation of quantum computing for nonlinear dynamical systems, two fundamental challenges that must be addressed are the encoding problem and the measurement problem. In particular, potential quantum computational advantages can be negated by the scaling cost associated with loading high-dimensional classical data into quantum states and extracting useful observables from quantum measurements. Future work should therefore focus on the development of scalable measurement and encoding strategies for quantum reservoir computing, including classical-shadow approaches, efficient observable reconstruction methods, amplitude encoding, and amplitude amplification techniques to mitigate the measurement and readout bottlenecks associated with high-dimensional quantum reservoirs.

Advanced quantum algorithms for nonlinear dynamics. The development of advanced quantum algorithms for nonlinear dynamics will require quantum-compatible numerical schemes for solving chaotic differential equations and nonlinear partial differential equations. Similar to the role played by numerical methods in the development of computational fluid dynamics (CFD), scalable and stable discretisation strategies are likely to be essential before practical quantum methods for nonlinear scientific computing can be realised. Future work should therefore focus on operator-theoretic approaches for nonlinear dynamics, particularly the implementation of discretised Koopman, Fokker–Planck, and transfer-operator formulations on quantum computers, together with methods for probabilistic evolution, uncertainty quantification, and ensemble evolution of chaotic systems. This includes high-order integration schemes, quantum signal processing, linear-combination-of-unitaries methods, and Schrödingerisation techniques for large-scale nonlinear and chaotic systems.

Quantum hardware implementation and benchmarking. As quantum hardware continues to mature, systematic benchmarking of the methods developed in this thesis on real quantum devices will be essential for evaluating their practical utility for nonlinear dynamical systems. This includes the study of noise-resilient quantum machine learning, error mitigation strategies, and scalable hybrid quantum-classical workflows. Benchmarking on near-term hardware will also help identify which components of the proposed frameworks are most sensitive to noise and where fault-tolerant quantum resources may ultimately be required.

Bibliography

Aaronson, S. (2018). "Shadow tomography of quantum states". In: *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*, pp. 325–338.

Abarbanel, H. D., N. F. Rulkov, and M. M. Sushchik (1996). "Generalized synchronization of chaos: The auxiliary system approach". In: *Physical review E* 53.5, p. 4528.

Abbas, A., D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner (June 2021). "The power of quantum neural networks". In: *Nature Computational Science* 1.6. Number: 6 Publisher: Nature Publishing Group, pp. 403–409. ISSN: 2662-8457. DOI: 10.1038/s43588-021-00084-1. URL: https://www.nature.com/articles/s43588-021-00084-1 (visited on 09/25/2023).

Ahmed, O., S. A. Qazi, and L. Magri (2026). *Quantum-inspired machine learning for efficient and reliable weather forecasting*. Tech. rep. Copernicus Meetings.

Ahmed, O., F. Tennie, and L. Magri (2024a). "Forecasting chaotic dynamics on a quantum computer: A hardware-efficient recurrence-free quantum reservoir computer". In: *APS Division of Fluid Dynamics Meeting Abstracts*, A16–004.

Ahmed, O., F. Tennie, and L. Magri (2024b). "Prediction of chaotic dynamics and extreme events: A recurrence-free quantum reservoir computing approach". In: *Physical Review Research* 6.4, p. 043082.

Ahmed, O., F. Tennie, and L. Magri (2025a). "Optimal training of finitely sampled quantum reservoir computers for forecasting of chaotic dynamics". In: *Quantum Machine Intelligence* 7.1, pp. 1–16.

Ahmed, O., F. Tennie, and L. Magri (2025b). "Robust quantum reservoir computers for forecasting chaotic dynamics: generalized synchronization and stability". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 481.2324.

Allawala, A. and J. B. Marston (Nov. 2016). "Statistics of the stochastically-forced Lorenz attractor by the Fokker-Planck equation and cumulant expansions". In: *Phys. Rev. E* 94.5. arXiv:1604.00867 [nlin], p. 052218. ISSN: 2470-0045, 2470-0053. DOI: 10.1103/PhysRevE.94.052218. URL: http://arxiv.org/abs/1604.00867 (visited on 03/22/2025).

Arjovsky, M., A. Shah, and Y. Bengio (2016). "Unitary evolution recurrent neural networks". In: *International conference on machine learning*. PMLR, pp. 1120–1128.

Arrazola, J. M., D. Guala, and J. Soni (Oct. 2023). "Linear combination of unitaries and block encodings". en. In: *PennyLane Demos*. URL: https://pennylane.ai/qml/demos/tutorial_lcu_blockencoding (visited on 02/12/2025).

Badola, P., S. Tambe, and B. Kulkarni (1992). “Driving systems with chaotic signals”. In: *Physical Review A* 46.10, p. 6735.

Beaulieu, D., M. Kornjaca, Z. Krunic, M. Stivaktakis, T. Ehmer, S.-T. Wang, and A. Pham (2024). “Robust Quantum Reservoir Computing for Molecular Property Prediction”. In: *arXiv preprint arXiv:2412.06758*.

Benettin, G., L. Galgani, A. Giorgilli, and J.-M. Strelcyn (1980). “Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: Theory”. In: *Meccanica* 15, pp. 9–20.

Bergholm, V., J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. Akash-Narayanan, A. Asadi, et al. (2018). “Pennylane: Automatic differentiation of hybrid quantum-classical computations”. In: *arXiv preprint arXiv:1811.04968*.

Bharti, K., A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, et al. (2022). “Noisy intermediate-scale quantum algorithms”. In: *Reviews of Modern Physics* 94.1, p. 015004.

Boedecker, J., O. Obst, J. T. Lizier, N. M. Mayer, and M. Asada (Sept. 2012). “Information processing in echo state networks at the edge of chaos”. In: *Theory in Biosciences* 131.3, pp. 205–213. ISSN: 1611-7530. DOI: 10.1007/s12064-011-0146-8. URL: https://doi.org/10.1007/s12064-011-0146-8 (visited on 10/05/2023).

Boffetta, G., M. Cencini, M. Falcioni, and A. Vulpiani (2002). “Predictability: a way to characterize complexity”. In: *Physics Reports* 356.6, pp. 367–474. ISSN: 0370-1573. DOI: https://doi.org/10.1016/S0370-1573(01)00025-4. URL: https://www.sciencedirect.com/science/article/pii/S0370157301000254.

Bondar, D. I., F. Gay-Balmaz, and C. Tronci (2019). “Koopman wavefunctions and classical–quantum correlation dynamics”. In: *Proceedings of the Royal Society A* 475.2229, p. 20180879. DOI: 10.1098/rspa.2018.0879.

Boyd, S. and L. Chua (Nov. 1985). “Fading memory and the problem of approximating nonlinear operators with Volterra series”. In: *IEEE Transactions on Circuits and Systems* 32.11. Conference Name: IEEE Transactions on Circuits and Systems, pp. 1150–1161. ISSN: 1558-1276. DOI: 10.1109/TCS.1985.1085649. URL: https://ieeexplore.ieee.org/document/1085649 (visited on 10/05/2023).

Bravo-Prieto, C., R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles (2020). *Variational Quantum Linear Solver*. arXiv: 1909.05820 [quant-ph].

Brunton, S. L. and J. N. Kutz (2022). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.

Brunton, S. L., J. L. Proctor, and J. N. Kutz (2016). “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the national academy of sciences* 113.15, pp. 3932–3937.

Butt, M. M. (June 2020). *Numerical solution for Fokker-Planck equation using a two-level scheme*. arXiv:2006.11038 [math]. DOI: 10.48550/arXiv.2006.11038. URL: http://arxiv.org/abs/2006.11038 (visited on 07/14/2025).

Camps, D. and R. V. Beeumen (Sept. 2022). “FABLE: Fast Approximate Quantum Circuits for Block-Encodings”. In: *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. arXiv:2205.00081 [quant-ph], pp. 104–113. DOI: 10.1109/QCE53715.2022.00029. URL: http://arxiv.org/abs/2205.00081 (visited on 09/23/2025).

Carleman, T. (1932). “Application de la théorie des équations intégrales linéaires aux systèmes d’équations différentielles nonlinéaires”. In: *Acta Mathematica* 59, pp. 63–87.

Carroll, T. L. (2020). “Dimension of reservoir computers”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.1.

Carroll, T. L. and L. M. Pecora (2019). “Network structure effects in reservoir computers”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.8.

Carroll, T. L. (Feb. 2022). “Optimizing Memory in Reservoir Computers”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 32.2. arXiv:2201.01605 [cs], p. 023123. ISSN: 1054-1500, 1089-7682. DOI: 10.1063/5.0078151. URL: http://arxiv.org/abs/2201.01605 (visited on 09/27/2023).

Cerezo, M., A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles (Sept. 2021). “Variational quantum algorithms”. In: *Nature Reviews Physics* 3.9. Number: 9 Publisher: Nature Publishing Group, pp. 625–644. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00348-9. URL: https://www.nature.com/articles/s42254-021-00348-9 (visited on 08/23/2023).

Chen, S. Y.-C., D. Fry, A. Deshmukh, V. Rastunkov, and C. Stefanski (2022). *Reservoir Computing via Quantum Recurrent Neural Networks*. arXiv: 2211.02612 [cs.NE].

Cheng, C., A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, Z. Kong, and S. T. Bukkapatnam (2015). “Time series forecasting for nonlinear and non-stationary processes: a review and comparative study”. In: *Iie Transactions* 47.10, pp. 1053–1071.

Childs, A. M., R. Kothari, and R. D. Somma (2017). “Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision”. In: *SIAM Journal on Computing* 46.6, pp. 1920–1950. DOI: 10.1137/16M1087072.

Clemente, G., A. Crippa, and K. Jansen (2023). “Quantum simulation of the Fokker–Planck equation via Schodinger equation”. In: *arXiv preprint*. arXiv: 2301.03105.

Consonni, R. and L. Magri (2026). “Precursors of extreme events and critical transitions”. In: *arXiv preprint arXiv:2604.12869*.

Ding, J. and A. Zhou (2009). “Frobenius-Perron Operators”. In: *Statistical Properties of Deterministic Systems*. Series Title: Tsinghua University Texts. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 62–91. ISBN: 978-3-540-85366-4 978-3-540-85367-1. DOI: 10.1007/978-3-540-85367-1_4. URL: http://link.springer.com/10.1007/978-3-540-85367-1_4 (visited on 02/20/2025).

Doan, N. A. K., W. Polifke, and L. Magri (2019a). *A physics-aware machine to predict extreme events in turbulence*. arXiv: 1912.10994 [physics.flu-dyn].

Doan, N. A. K., W. Polifke, and L. Magri (Dec. 2019b). *A physics-aware machine to predict extreme events in turbulence*. arXiv:1912.10994 [nlin, physics:physics]. DOI: 10.48550/arXiv.1912.10994. URL: http://arxiv.org/abs/1912.10994 (visited on 08/24/2023).

Doan, N. A. K., W. Polifke, and L. Magri (2019c). “Physics-Informed Echo State Networks for Chaotic Systems Forecasting”. In: *Computational Science – ICCS 2019*. Ed. by J. M. F. Rodrigues, P. J. S. Cardoso, J. Monteiro, R. Lam, V. V. Krzhizhanovskaya, M. H. Lees, J. J. Dongarra, and P. M. Sloot. Cham: Springer International Publishing, pp. 192–198. ISBN: 978-3-030-22747-0.

Doan, N. A. K., W. Polifke, and L. Magri (2021). “Short-and long-term predictions of chaotic flows and extreme events: a physics-constrained reservoir computing approach”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 477.2253.

Domingo, L., M. Grande, G. Carlo, F. Borondo, and J. Borondo (2023). “Optimal quantum reservoir computing for market forecasting: An application to fight food price crises”. In: *arXiv preprint arXiv:2401.03347*.

Domingo, L., G. Carlo, and F. Borondo (May 2023). "Taking advantage of noise in quantum reservoir computing". In: *Scientific Reports* 13.1. Number: 1 Publisher: Nature Publishing Group, p. 8790. ISSN: 2045-2322. DOI: 10.1038/s41598-023-35461-5. URL: https://www.nature.com/articles/s41598-023-35461-5 (visited on 08/16/2023).

Dudas, J., B. Carles, E. Plouet, F. A. Mizrahi, J. Grollier, and D. Marković (July 2023). "Quantum reservoir computing implementation on coherently coupled quantum oscillators". In: *npj Quantum Information* 9.1. Number: 1 Publisher: Nature Publishing Group, pp. 1–7. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00734-4. URL: https://www.nature.com/articles/s41534-023-00734-4 (visited on 08/03/2023).

Engel, A., G. Smith, and S. E. Parker (June 2021). "Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms". In: *Physics of Plasmas* 28.6. arXiv:2012.06681 [physics], p. 062305. ISSN: 1070-664X, 1089-7674. DOI: 10.1063/5.0040313. URL: http://arxiv.org/abs/2012.06681 (visited on 09/05/2025).

Farkaš, I., R. Bosák, and P. Gergeľ (Nov. 2016). "Computational analysis of memory capacity in echo state networks". In: *Neural Networks* 83, pp. 109–120. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2016.07.012. URL: https://www.sciencedirect.com/science/article/pii/S0893608016300946 (visited on 09/13/2023).

Feynman, R. P. (1982). "Simulating Physics with Computers". In: *International Journal of Theoretical Physics* 21.6/7.

Feynman, R. P. (2018). "Simulating physics with computers". In: *Feynman and computation*. cRc Press, pp. 133–153.

Field, R. J. et al. (1993). *Chaos in chemistry and biochemistry*. World Scientific.

Frederickson, P., J. L. Kaplan, E. D. Yorke, and J. A. Yorke (1983). "The Liapunov dimension of strange attractors". In: *Journal of differential equations* 49.2, pp. 185–207.

Fry, D., A. Deshmukh, S. Y.-C. Chen, V. Rastunkov, and V. Markov (2023). *Optimizing Quantum Noise-induced Reservoir Computing for Nonlinear and Chaotic Time Series Prediction*. arXiv: 2303.05488 [quant-ph].

Fujii, K. and K. Nakajima (Aug. 2017). "Harnessing Disordered-Ensemble Quantum Dynamics for Machine Learning". In: *Physical Review Applied* 8.2. Publisher: American Physical Society, p. 024030. DOI: 10.1103/PhysRevApplied.8.024030. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.8.024030 (visited on 08/22/2023).

Fujii, K. and K. Nakajima (Nov. 2020). *Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices*. arXiv:2011.04890 [nlin, physics:quant-ph]. URL: http://arxiv.org/abs/2011.04890 (visited on 03/15/2023).

Fujii, K. and K. Nakajima (2021). "Quantum Reservoir Computing: A Reservoir Approach Toward Quantum Machine Learning on Near-Term Quantum Devices". In: *Reservoir Computing: Theory, Physical Implementations, and Applications*. Ed. by K. Nakajima and I. Fischer. Singapore: Springer Singapore, pp. 423–450. ISBN: 978-981-13-1687-6. DOI: 10.1007/978-981-13-1687-6_18. URL: https://doi.org/10.1007/978-981-13-1687-6_18.

Gabbiani, F. and S. J. Cox (2010). "CHAPTER 10 - Reduced Single Neuron Models". In: *Mathematics for Neuroscientists*. Ed. by F. Gabbiani and S. J. Cox. London: Academic Press, pp. 143–154. ISBN: 978-0-12-374882-9. DOI: https://doi.org/10.1016/B978-0-12-374882-9.00010-1. URL: https://www.sciencedirect.com/science/article/pii/B9780123748829000101.

Geisel, T. (1985). "Chaos and Noise". In: *Chaos in Astrophysics*. Ed. by J. R. Buchler, J. M. Perdang, and E. A. Spiegel. Dordrecht: Springer Netherlands, pp. 165–183. ISBN: 978-94-009-5468-7. DOI: 10.1007/978-94-009-5468-7_5. URL: https://doi.org/10.1007/978-94-009-5468-7_5.

Ghadami, A. and B. I. Epureanu (2022). "Data-driven prediction in dynamical systems: recent developments". In: *Philosophical Transactions of the Royal Society A* 380.2229, p. 20210213.

Gilyén, A., Y. Su, G. H. Low, and N. Wiebe (2019). "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". In: pp. 193–204. DOI: 10.1145/3313276.3316366.

Ginelli, F., H. Chaté, R. Livi, and A. Politi (2013). "Covariant lyapunov vectors". In: *Journal of Physics A: Mathematical and Theoretical* 46.25, p. 254005.

Ginelli, F., P. Poggi, A. Turchi, H. Chaté, R. Livi, and A. Politi (2007). "Characterizing dynamics with covariant Lyapunov vectors". In: *Physical review letters* 99.13, p. 130601.

Golub, G. H. et al. (1996). "Cf vanloan, matrix computations". In: *The Johns Hopkins* 113.10, pp. 23–36.

Götting, N., F. Lohof, and C. Gies (Nov. 2023). "Exploring quantumness in quantum reservoir computing". In: *Phys. Rev. A* 108 (5), p. 052427. DOI: 10.1103/PhysRevA.108.052427. URL: https://link.aps.org/doi/10.1103/PhysRevA.108.052427.

Goutte, C. and E. Gaussier (2005). "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation". In: *Advances in Information Retrieval*. Ed. by D. E. Losada and J. M. Fernandez-Luna. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 345–359. ISBN: 978-3-540-31865-1.

Grigoryeva, L., A. Hart, and J.-P. Ortega (2021). "Chaos on compact manifolds: Differentiable synchronizations beyond the Takens theorem". In: *Physical Review E* 103.6, p. 062204.

Harrow, A. W., A. Hassidim, and S. Lloyd (Oct. 2009). "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters* 103.15. DOI: 10.1103/physrevlett.103.150502. URL: https://doi.org/10.1103%2Fphysrevlett.103.150502.

Hart, J. D. (2024). "Attractor reconstruction with reservoir computers: The effect of the reservoir's conditional Lyapunov exponents on faithful attractor reconstruction". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 34.4.

Havlíček, V., A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta (Mar. 2019). "Supervised learning with quantum-enhanced feature spaces". In: *Nature* 567.7747. Number: 7747 Publisher: Nature Publishing Group, pp. 209–212. ISSN: 1476-4687. DOI: 10.1038/s41586-019-0980-2. URL: https://www.nature.com/articles/s41586-019-0980-2 (visited on 12/28/2023).

Heisenberg, W. (1949). *The physical principles of the quantum theory*. Courier Corporation.

Herteux, J. and C. Räth (2020). "Breaking symmetries of the reservoir equations in echo state networks". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.12.

Holmstrom, M., D. Liu, and C. Vo (2016). "Machine learning applied to weather forecasting". In: *Meteorol. Appl* 10, pp. 1–5.

Holubec, V., K. Kroy, and S. Steffenoni (Mar. 2019). "Physically consistent numerical solver for time-dependent Fokker-Planck equations". en. In: *Phys. Rev. E* 99.3, p. 032117. ISSN: 2470-0045, 2470-0053. DOI: 10.1103/PhysRevE.99.032117. URL: https://link.aps.org/doi/10.1103/PhysRevE.99.032117 (visited on 12/24/2024).

Hu, F., G. Angelatos, S. A. Khan, M. Vives, E. Türeci, L. Bello, G. E. Rowlands, G. J. Ribeill, and H. E. Türeci (2023). "Tackling sampling noise in physical systems for machine learning applications: Fundamental limits and eigentasks". In: *Physical Review X* 13.4, p. 041020.

Huang, H.-Y., R. Kueng, and J. Preskill (2020). "Predicting many properties of a quantum system from very few measurements". In: *Nature Physics* 16.10, pp. 1050–1057.

Huhn, F. and L. Magri (2020a). "Learning Ergodic Averages in Chaotic Systems". In: *Computational Science – ICCS 2020*. Ed. by V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira. Cham: Springer International Publishing, pp. 124–132. ISBN: 978-3-030-50433-5.

Huhn, F. and L. Magri (2020b). "Learning Ergodic Averages in Chaotic Systems". In: *Computational Science – ICCS 2020*. Ed. by V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira. Cham: Springer International Publishing, pp. 124–132. ISBN: 978-3-030-50433-5.

Huhn, F. and L. Magri (Jan. 2020c). "Stability, sensitivity and optimisation of chaotic acoustic oscillations". In: *Journal of Fluid Mechanics* 882, A24. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/jfm.2019.828. URL: https://www.cambridge.org/core/product/identifier/S0022112019008280/type/journal_article (visited on 03/15/2023).

Huhn, F. and L. Magri (2020d). "Stability, sensitivity and optimisation of chaotic acoustic oscillations". In: *Journal of Fluid Mechanics* 882, A24.

Huhn, F. and L. Magri (2022). "Gradient-free optimization of chaotic acoustics with reservoir computing". In: *Physical Review Fluids* 7.1, p. 014402.

Hunt, B. R., E. Ott, and J. A. Yorke (1997). "Differentiable generalized synchronization of chaos". In: *Physical Review E* 55.4, p. 4029.

(2024).

Jaeger, H. (2001). "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note". In: *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148.34, p. 13.

Jaeger, H., Mantas, D. Popovici, and U. Siewert (2007). "Optimization and applications of echo state networks with leaky- integrator neurons". In: *Neural Networks* 20.3. Echo State Networks and Liquid State Machines, pp. 335–352. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2007.04.016. URL: https://www.sciencedirect.com/science/article/pii/S089360800700041X.

Jafri, H. H., T. U. Singh, and K. Manchanda (2019). "Revisiting generalized synchronization: Progress and perspectives". en. In.

Jha, S. K. and R. Yadava (2010). "Denoising by singular value decomposition and its application to electronic nose data processing". In: *IEEE Sensors Journal* 11.1, pp. 35–44.

Jin, S., N. Liu, and Y. Yu (Sept. 2023a). "Quantum simulation of partial differential equations: Applications and detailed analysis". In: *Phys. Rev. A* 108.3, p. 032603. DOI: 10.1103/PhysRevA.108.032603. URL: https://link.aps.org/doi/10.1103/PhysRevA.108.032603 (visited on 03/03/2025).

Jin, S., N. Liu, and Y. Yu (Aug. 2023b). "Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations". In: *Journal of Computational Physics* 487, p. 112149. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2023.112149. URL: https://www.sciencedirect.com/science/article/pii/S0021999123002449 (visited on 01/09/2025).

Joseph, I. (Oct. 2020). "Koopman–von Neumann approach to quantum simulation of nonlinear classical dynamics". en. In: *Phys. Rev. Research* 2.4, p. 043102. ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.2.043102. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.2.043102 (visited on 10/11/2024).

Khan, S. A., R. Kaufman, B. Mesits, M. Hatridge, and H. E. Türeci (2023). "Practical trainable temporal post-processor for multi-state quantum measurement". In: *arXiv preprint arXiv:2310.18519*.

Khan, S. A., F. Hu, G. Angelatos, and H. E. Türeci (2021). "Physical reservoir computing using finitely-sampled quantum systems". In: *arXiv preprint arXiv:2110.13849*.

Klus, S., F. Nüske, and S. Peitz (June 2022). *Koopman analysis of quantum systems*. arXiv:2201.12062. URL: http://arxiv.org/abs/2201.12062 (visited on 10/11/2024).

Kocarev, L. and U. Parlitz (1996). "Generalized synchronization, predictability, and equivalence of unidirectionally coupled dynamical systems". In: *Physical review letters* 76.11, p. 1816.

Koopman, B. O. (1931). "Hamiltonian systems and transformations in Hilbert space". In: *Proceedings of the National Academy of Sciences* 17.5, pp. 315–318. DOI: 10.1073/pnas.17.5.315.

Kornjača, M., H.-Y. Hu, C. Zhao, J. Wurtz, P. Weinberg, M. Hamdan, A. Zhdanov, S. H. Cantu, H. Zhou, R. A. Bravo, et al. (2024). "Large-scale quantum reservoir learning with an analog quantum computer". In: *arXiv preprint arXiv:2407.02553*.

Kowalski, K. and W.-H. Steeb (1991). *Nonlinear Dynamical Systems and Carleman Linearization*. Singapore: World Scientific.

Kreplin, D. A. and M. Roth (2024). "Reduction of finite sampling noise in quantum neural networks". In: *Quantum* 8, p. 1385.

Kubota, T., Y. Suzuki, S. Kobayashi, Q. H. Tran, N. Yamamoto, and K. Nakajima (Apr. 2023). "Temporal information processing induced by quantum noise". In: *Physical Review Research* 5.2, p. 023057. ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.5.023057. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.5.023057 (visited on 08/03/2023).

Kuptsov, P. V. and U. Parlitz (2012). "Theory and computation of covariant Lyapunov vectors". In: *Journal of nonlinear science* 22, pp. 727–762.

Kutz, J. N., S. L. Brunton, B. W. Brunton, and J. L. Proctor (2016). *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.

Lansky, P. and S. Ditlevsen (2008). "A review of the methods for signal estimation in stochastic diffusion leaky integrate-and-fire neuronal models". In: *Biological cybernetics* 99.4, pp. 253–262.

LeCun, Y. (2019). "1.1 deep learning hardware: Past, present, and future". In: *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, pp. 12–19.

Lewis, D., S. Eidenbenz, B. Nadiga, and Y. Subaşı (Oct. 2024). "Limitations for Quantum Algorithms to Solve Turbulent and Chaotic Systems". en-GB. In: *Quantum* 8, p. 1509. DOI: 10.22331/q-2024-10-24-1509. URL: https://quantum-journal.org/papers/q-2024-10-24-1509/ (visited on 02/18/2025).

Lin, Y. T., R. B. Lowrie, D. Aslangil, Y. Subaşı, and A. T. Sornborger (July 2024). *Challenges for quantum computation of nonlinear dynamical systems using linear representations*. arXiv:2202.02188 [quant-ph]. DOI: 10.48550/arXiv.2202.02188. URL: http://arxiv.org/abs/2202.02188 (visited on 01/09/2025).

Liu, J.-P., H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs (2021). "Efficient quantum algorithm for dissipative nonlinear differential equations". In: *Proceedings of the National Academy of Sciences* 118.35, e2026805118.

Lloyd, S., B. T. Kiani, D. R. M. Arvidsson-Shukur, S. Bosch, G. D. Palma, W. M. Kaminsky, Z.-W. Liu, and M. Marvian (May 2021). *Hamiltonian singular value transformation and inverse block encoding*. arXiv:2104.01410 [quant-ph]. DOI: 10.48550/arXiv.2104.01410. URL: http://arxiv.org/abs/2104.01410 (visited on 02/12/2025).

Lorenz, E. (1995 1995). "Predictability: a problem partly solved". PhD thesis. Shinfield Park, Reading, pp. 1–18.

Lorenz, E. N. (Mar. 1963). "Deterministic Nonperiodic Flow". In: *Journal of the Atmospheric Sciences* 20.2. Publisher: American Meteorological Society Section: Journal of the Atmospheric Sciences, pp. 130–141. ISSN: 0022-4928, 1520-0469. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2. URL: https://

journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml (visited on 01/04/2024).

Lu, Z., B. R. Hunt, and E. Ott (2018a). "Attractor reconstruction by machine learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.6.

Lu, Z., B. R. Hunt, and E. Ott (June 2018b). "Attractor Reconstruction by Machine Learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.6. arXiv:1805.03362 [nlin], p. 061104. ISSN: 1054-1500, 1089-7682. DOI: 10.1063/1.5039508. URL: http://arxiv.org/abs/1805.03362 (visited on 09/18/2024).

Lucarini, V., R. Blender, C. Herbert, F. Ragone, S. Pascale, and J. Wouters (2014). "Mathematical and physical ideas for climate science". In: *Reviews of Geophysics* 52.4, pp. 809–859.

Lukosevicius, M. (2012). "A Practical Guide to Applying Echo State Networks". In: *Neural Networks: Tricks of the trade, Second Edition*. Ed. by G. Montavon, G. B. Orr, and K.-R. Muller. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 659–686. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_36. URL: https://doi.org/10.1007/978-3-642-35289-8_36 (visited on 08/25/2023).

Lymburn, T., D. M. Walker, M. Small, and T. Jüngling (2019). "The reservoir's perspective on generalized synchronization". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.9.

Maass, W., T. Natschläger, and H. Markram (Nov. 2002). "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations". In: *Neural Computation* 14.11, pp. 2531–2560. ISSN: 0899-7667. DOI: 10.1162/089976602760407955. eprint: https://direct.mit.edu/neco/article-pdf/14/11/2531/815288/089976602760407955.pdf. URL: https://doi.org/10.1162/089976602760407955.

Magri, L., A. Novoa, and E. Özalp (2024). "Machine Learning for Fluid Dynamics". In: von Karman Institute for Fluid Dynamics. Chap. Prediction of chaotic dynamics from data: An introduction.

Manneville, P. (2010). *Instabilities, chaos and turbulence*. Vol. 1. World Scientific.

Margazoglou, G. and L. Magri (May 2023). "Stability analysis of chaotic systems from data". In: *Nonlinear Dynamics* 111.9, pp. 8799–8819. ISSN: 0924-090X, 1573-269X. DOI: 10.1007/s11071-023-08285-1. URL: https://link.springer.com/10.1007/s11071-023-08285-1 (visited on 08/02/2023).

Martínez-Peña, R., J. Nokkala, G. L. Giorgi, R. Zambrini, and M. C. Soriano (2023). "Information processing capacity of spin-based quantum reservoir computing systems". In: *Cognitive Computation*, pp. 1–12.

Martínez-Peña, R. and J.-P. Ortega (2023). "Quantum reservoir computing in finite dimensions". In: *Physical Review E* 107.3, p. 035306.

Martínez-Peña, R. and J.-P. Ortega (2025). "Input-dependence in quantum reservoir computing". In: *Physical Review E* 111.6, p. 065306.

Mauro, D. (2002). "Topics in Koopman–von Neumann theory". In: *arXiv preprint*. arXiv: quant-ph/0301172.

McClean, J. R., S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven (2018). "Barren plateaus in quantum neural network training landscapes". In: *Nature communications* 9.1, p. 4812.

Medsker, L. R., L. Jain, et al. (2001). "Recurrent neural networks". In: *Design and applications* 5.64-67, p. 2.

Mezić, I. (2005). "Spectral properties of dynamical systems, model reduction and decompositions". In: *Nonlinear Dynamics* 41, pp. 309–325. DOI: 10.1007/s11071-005-2824-x.

Mitarai, K., M. Negoro, M. Kitagawa, and K. Fujii (Sept. 2018). "Quantum circuit learning". In: *Physical Review A* 98.3. DOI: 10.1103/physreva.98.032309. URL: https://doi.org/10.1103%2Fphysreva.98.032309.

Moehlis, J., H. Faisst, and B. Eckhardt (May 2004). "A low-dimensional model for turbulent shear flows". In: *New Journal of Physics* 6, pp. 56–56. ISSN: 1367-2630. DOI: 10.1088/1367-2630/6/1/056. URL: https://iopscience.iop.org/article/10.1088/1367-2630/6/1/056 (visited on 03/15/2023).

Mujal, P., R. Martínez-Peña, G. L. Giorgi, M. C. Soriano, and R. Zambrini (Feb. 2023a). "Time Series Quantum Reservoir Computing with Weak and Projective Measurements". In: *npj Quantum Information* 9.1. arXiv:2205.06809 [cond-mat, physics:quant-ph], p. 16. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00682-z. URL: http://arxiv.org/abs/2205.06809 (visited on 08/03/2023).

Mujal, P., R. Martínez-Peña, G. L. Giorgi, M. C. Soriano, and R. Zambrini (Feb. 2023b). "Time-series quantum reservoir computing with weak and projective measurements". In: *npj Quantum Information* 9.1, p. 16. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00682-z. URL: https://doi.org/10.1038/s41534-023-00682-z.

Mujal, P., R. Martínez-Peña, J. Nokkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini (Feb. 2021). *Opportunities in Quantum Reservoir Computing and Extreme Learning Machines*. Publication Title: arXiv e-prints ADS Bibcode: 2021arXiv210211831M. DOI: 10.48550/arXiv.2102.11831. URL: https://ui.adsabs.harvard.edu/abs/2021arXiv210211831M (visited on 12/27/2023).

Nastac, G., J. W. Labahn, L. Magri, and M. Ihme (2017). "Lyapunov exponent as a metric for assessing the dynamic content and predictability of large-eddy simulations". In: *Physical Review Fluids* 2.9, p. 094606.

Nielsen, M. A. and I. L. Chuang (2011). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. ISBN: 9781107002173.

Novikau, I. and I. Joseph (Apr. 2025). "Quantum algorithm for the advection-diffusion equation and the Koopman-von Neumann approach to nonlinear dynamical systems". In: *Computer Physics Communications* 309. arXiv:2410.03985 [physics], p. 109498. ISSN: 00104655. DOI: 10.1016/j.cpc.2025.109498. URL: http://arxiv.org/abs/2410.03985 (visited on 02/24/2025).

Oseledec, V. I. (1968). "A multiplicative ergodic theorem, Lyapunov characteristic numbers for dynamical systems". In: *Transactions of the Moscow Mathematical Society* 19, pp. 197–231.

Özalp, E. and L. Magri (2025). "Stability analysis of chaotic systems in latent spaces". In: *Nonlinear Dynamics*, pp. 1–16.

Özalp, E., G. Margazoglou, and L. Magri (2023a). "Reconstruction, forecasting, and stability of chaotic dynamics from partial data". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 33.9.

Özalp, E., G. Margazoglou, and L. Magri (2023b). "Reconstruction, forecasting, and stability of chaotic dynamics from partial data". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 33.9.

Ozan, D. E. and L. Magri (2024). "Data-driven computation of adjoint sensitivities without adjoint solvers: An application to thermoacoustics". In: *Physical Review Fluids* 9.10, p. 103902.

Paine, A. E., V. E. Elfving, and O. Kyriienko (Aug. 2023). *Physics-Informed Quantum Machine Learning: Solving nonlinear differential equations in latent spaces without costly grid evaluations*. arXiv:2308.01827 [cond-mat, physics:quant-ph]. DOI: 10.48550/arXiv.2308.01827. URL: http://arxiv.org/abs/2308.01827 (visited on 08/04/2023).

Palmer, T. N. (1993). "Extended-Range Atmospheric Prediction and the Lorenz Model". In: *Bulletin of the American Meteorological Society* 74.1, pp. 49–66. DOI: 10.1175/1520-0477(1993)074<0049:ERAPAT>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/bams/74/1/1520-0477_1993_074_0049_erapat_2_0_co_2.xml.

Pathak, J., B. Hunt, M. Girvan, Z. Lu, and E. Ott (Jan. 2018a). "Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach". In: *Physical Review Letters* 120.2. Publisher:

American Physical Society, p. 024102. DOI: 10.1103/PhysRevLett.120.024102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.024102 (visited on 08/14/2023).

Pathak, J., B. Hunt, M. Girvan, Z. Lu, and E. Ott (2018b). "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach". In: *Physical review letters* 120.2, p. 024102.

Pathak, J., Z. Lu, B. R. Hunt, M. Girvan, and E. Ott (2017). "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.12.

Pathak, J., A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott (Apr. 2018). "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.4, p. 041101. ISSN: 1054-1500. DOI: 10.1063/1.5028373. eprint: https://pubs.aip.org/aip/cha/article-pdf/doi/10.1063/1.5028373/10315097/041101_1_online.pdf. URL: https://doi.org/10.1063/1.5028373.

Pawula, R. (1967). "Approximation of the linear Boltzmann equation by the Fokker-Planck equation". In: *Physical review* 162.1, p. 186.

Pecora, L. M. and T. L. Carroll (1990). "Synchronization in chaotic systems". In: *Physical review letters* 64.8, p. 821.

Pecora, L. M. and T. L. Carroll (1991a). "Driving systems with chaotic signals". In: *Physical review A* 44.4, p. 2374.

Pecora, L. M. and T. L. Carroll (2015). "Synchronization of chaotic systems". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.9.

Pecora, L. M. and T. L. Carroll (Aug. 1991b). "Driving systems with chaotic signals". In: *Physical Review A* 44.4. Publisher: American Physical Society, pp. 2374–2383. DOI: 10.1103/PhysRevA.44.2374. URL: https://link.aps.org/doi/10.1103/PhysRevA.44.2374 (visited on 09/17/2024).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pfeffer, P., F. Heyder, and J. Schumacher (Sept. 2022). "Hybrid quantum-classical reservoir computing of thermal convection flow". In: *Physical Review Research* 4.3, p. 033176. ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.4.033176. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.4.033176 (visited on 03/15/2023).

Piasecki, D. (Dec. 2021). *Introduction to Koopman-von Neumann Mechanics*. arXiv:2112.05619 [quant-ph]. DOI: 10.48550/arXiv.2112.05619. URL: http://arxiv.org/abs/2112.05619 (visited on 08/21/2025).

Platt, J. A., A. Wong, R. Clark, S. G. Penny, and H. D. Abarbanel (2021). "Robust forecasting using predictive generalized synchronization in reservoir computing". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.12.

Platt, J. A., S. G. Penny, T. A. Smith, T.-C. Chen, and H. D. I. Abarbanel (Sept. 2022). "A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics". In: *Neural Networks* 153, pp. 530–552. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2022.06.025. URL: https://www.sciencedirect.com/science/article/pii/S0893608022002404 (visited on 08/04/2023).

Preskill, J. (1998a). "Fault-tolerant quantum computation". In: *Introduction to quantum computation and information*. World Scientific, pp. 213–269.

Preskill, J. (1998b). "Lecture notes for physics 229: Quantum information and computation". In: *California institute of technology* 16.1, pp. 1–8.

Pyragas, K. (1997). "Conditional Lyapunov exponents from time series". In: *Physical Review E* 56.5, p. 5183.

Qiskit contributors (2023). *Qiskit: An Open-source Framework for Quantum Computing*. DOI: 10.5281/zenodo.2573505.

Racca, A., N. A. K. Doan, and L. Magri (2023). "Predicting turbulent dynamics with the convolutional autoencoder echo state network". In: *Journal of Fluid Mechanics* 975, A2.

Racca, A. and L. Magri (2021). "Robust Optimization and Validation of Echo State Networks for learning chaotic dynamics". In: *Neural Networks* 142, pp. 252–268. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2021.05.004. URL: https://www.sciencedirect.com/science/article/pii/S0893608021001969.

Racca, A. and L. Magri (Oct. 2022a). "Data-driven prediction and control of extreme events in a chaotic flow". In: *Physical Review Fluids* 7.10, p. 104402. ISSN: 2469-990X. DOI: 10.1103/PhysRevFluids.7.104402. URL: https://link.aps.org/doi/10.1103/PhysRevFluids.7.104402 (visited on 03/15/2023).

Racca, A. and L. Magri (2022b). "Statistical Prediction of Extreme Events from Small Datasets". In: *Computational Science – ICCS 2022*. Ed. by D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot. Cham: Springer International Publishing, pp. 707–713. ISBN: 978-3-031-08757-8.

Rauh, A., J. Minisini, and H. Aschemann (Jan. 2009). "Carleman Linearization for Control and for State and Disturbance Estimation of Nonlinear Dynamical Processes". In: *IFAC Proceedings Volumes*. 14th IFAC Conference on Methods and Models in Automation and Robotics 42.13, pp. 455–460. ISSN: 1474-6670. DOI: 10.3182/20090819-3-PL-3002.00079. URL: https://www.sciencedirect.com/science/article/pii/S1474667015304821 (visited on 05/06/2025).

Risken, H. (1989). "Fokker-planck equation". In: *The Fokker-Planck equation: methods of solution and applications*. Springer, pp. 63–95.

Rolnick, D., P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, et al. (2022). "Tackling climate change with machine learning". In: *ACM Computing Surveys (CSUR)* 55.2, pp. 1–96.

Rulkov, N. F., M. M. Sushchik, L. S. Tsimring, and H. D. Abarbanel (1995). "Generalized synchronization of chaos in directionally coupled chaotic systems". In: *Physical Review E* 51.2, p. 980.

Salehinejad, H., S. Sankar, J. Barfett, E. Colak, and S. Valaee (2017). "Recent advances in recurrent neural networks". In: *arXiv preprint arXiv:1801.01078*.

Sanavio, C., R. Scatamacchia, C. de Falco, and S. Succi (May 2024). "Three Carleman routes to the quantum simulation of classical fluids". In: *Physics of Fluids* 36.5, p. 057143. ISSN: 1070-6631. DOI: 10.1063/5.0204955. URL: https://doi.org/10.1063/5.0204955 (visited on 09/13/2024).

Sannia, A., R. Martínez-Peña, M. C. Soriano, G. L. Giorgi, and R. Zambrini (Dec. 2022). *Dissipation as a resource for Quantum Reservoir Computing*. arXiv:2212.12078 [quant-ph]. DOI: 10.48550/arXiv.2212.12078. URL: http://arxiv.org/abs/2212.12078 (visited on 08/18/2023).

Schafer, R. W. (2011). "What is a savitzky-golay filter?[lecture notes]". In: *IEEE Signal processing magazine* 28.4, pp. 111–117.

Schanze, T. (2017). "Removing noise in biomedical signal recordings by singular value decomposition". In: *current directions in biomedical engineering* 3.2, pp. 253–256.

Schmid, P. J. (2010). "Dynamic mode decomposition of numerical and experimental data". In: *Journal of fluid mechanics* 656, pp. 5–28.

Schmid, P. J. (2022). "Dynamic mode decomposition and its variants". In: *Annual Review of Fluid Mechanics* 54.1, pp. 225–254.

Schuld, M., V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran (2019). “Evaluating analytic gradients on quantum hardware”. In: *Physical Review A* 99.3, p. 032331.

Schuld, M. and F. Petruccione (2021). *Machine learning with quantum computers*. Springer.

Selvam, A. M. (2017). “Nonlinear Dynamics and Chaos: Applications in Meteorology and Atmospheric Physics”. In: *Self-organized Criticality and Predictability in Atmospheric Flows: The Quantum World of Clouds and Rain*. Cham: Springer International Publishing, pp. 1–40. ISBN: 978-3-319-54546-2. DOI: 10.1007/978-3-319-54546-2_1. URL: https://doi.org/10.1007/978-3-319-54546-2_1.

Sezer, O. B., M. U. Gudelek, and A. M. Ozbayoglu (2020). “Financial time series forecasting with deep learning : A systematic literature review: 2005–2019”. In: *Applied Soft Computing* 90, p. 106181. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2020.106181. URL: https://www.sciencedirect.com/science/article/pii/S1568494620301216.

Shen, Z., Z. Wang, S. Kale, A. Ribeiro, A. Karbasi, and H. Hassani (June 2022). *Self-Consistency of the Fokker-Planck Equation*. arXiv:2206.00860 [cs]. DOI: 10.48550/arXiv.2206.00860. URL: http://arxiv.org/abs/2206.00860 (visited on 12/02/2025).

Shimada, I. and T. Nagashima (1979). “A numerical approach to ergodic problem of dissipative dynamical systems”. In: *Progress of theoretical physics* 61.6, pp. 1605–1616.

Shlens, J. (2014). “A tutorial on principal component analysis”. In: *arXiv preprint arXiv:1404.1100*.

Shor, P. (1994). “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

Snoek, J., H. Larochelle, and R. P. Adams (2012). “Practical bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems* 25.

Sornsaeng, A., N. Dangniam, and T. Chotibut (Aug. 2023). *Quantum Next Generation Reservoir Computing: An Efficient Quantum Algorithm for Forecasting Quantum Dynamics*. arXiv:2308.14239 [cond-mat, physics:physics, physics:quant-ph]. DOI: 10.48550/arXiv.2308.14239. URL: http://arxiv.org/abs/2308.14239 (visited on 09/06/2023).

Srinivasan, P. A., L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa (May 2019). “Predictions of turbulent shear flows using deep neural networks”. In: *Physical Review Fluids* 4.5, p. 054603. ISSN: 2469-990X. DOI: 10.1103/PhysRevFluids.4.054603. URL: https://link.aps.org/doi/10.1103/PhysRevFluids.4.054603 (visited on 08/02/2023).

Storm, L., K. Gustavsson, and B. Mehlig (Dec. 2022). “Constraints on parameter choices for successful time-series prediction with echo-state networks”. In: *Machine Learning: Science and Technology* 3.4. Publisher: IOP Publishing, p. 045021. ISSN: 2632-2153. DOI: 10.1088/2632-2153/aca1f6. URL: https://dx.doi.org/10.1088/2632-2153/aca1f6 (visited on 12/01/2023).

Sünderhauf, C., E. Campbell, and J. Camps (2024). “Block-encoding structured matrices for data input in quantum computing”. In: *Quantum* 8, p. 1226.

Suzuki, Y., Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto (Jan. 2022). “Natural quantum reservoir computing for temporal information processing”. In: *Scientific Reports* 12.1, p. 1353. ISSN: 2045-2322. DOI: 10.1038/s41598-022-05061-w. URL: https://www.nature.com/articles/s41598-022-05061-w (visited on 03/15/2023).

Tang, D., R. Dykstra, M. Hamilton, and N. Heckenberg (1998). “Observation of generalized synchronization of chaos in a driven chaotic system”. In: *Physical Review E* 57.5, p. 5247.

Teeter, C., R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawrylycz, et al. (2018). "Generalized leaky integrate-and-fire models classify multiple neuron types". In: *Nature communications* 9.1, p. 709.

Tennie, F., S. Laizet, S. Lloyd, and L. Magri (2025). "Quantum computing for nonlinear differential equations and turbulence". In: *Nature Reviews Physics* 7.4, pp. 220–230.

Tennie, F. and L. Magri (Jan. 2024). *Solving nonlinear differential equations on Quantum Computers: A Fokker-Planck approach*. arXiv:2401.13500 [nlin, physics:physics, physics:quant-ph]. DOI: 10.48550/arXiv.2401.13500. URL: http://arxiv.org/abs/2401.13500 (visited on 06/25/2024).

Tikhonenkov, I., A. Vardi, J. R. Anglin, and D. Cohen (Jan. 2013). "Minimal Fokker-Planck theory for the thermalization of mesoscopic subsystems". In: *Phys. Rev. Lett.* 110.5. arXiv:1210.2607 [cond-mat], p. 050401. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.110.050401. URL: http://arxiv.org/abs/1210.2607 (visited on 03/23/2025).

Tran, Q. H. and K. Nakajima (Oct. 2020). *Higher-Order Quantum Reservoir Computing*. arXiv:2006.08999 [nlin, physics:quant-ph]. URL: http://arxiv.org/abs/2006.08999 (visited on 08/18/2023).

Verzelli, P., C. Alippi, and L. Livi (2021a). "Learn to synchronize, synchronize to learn". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.8.

Verzelli, P., C. Alippi, and L. Livi (2021b). "Learn to synchronize, synchronize to learn". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.8.

Vlachas, P. R., J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos (June 2020). "Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics". In: *Neural Networks* 126, pp. 191–217. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2020.02.016. URL: https://www.sciencedirect.com/science/article/pii/S0893608020300708 (visited on 08/02/2023).

Vlachas, P. R., W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos (2018). "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2213, p. 20170844.

Wang, C., T. Zhang, X. Chen, S. He, S. Li, and S. Wu (Dec. 2023). "BrainPy, a flexible, integrative, efficient, and extensible framework for general-purpose brain dynamics programming". In: *eLife* 12. Ed. by M. Stimberg, e86365. ISSN: 2050-084X. DOI: 10.7554/eLife.86365. URL: https://doi.org/10.7554/eLife.86365.

Werbos, P. (1990). "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78.10, pp. 1550–1560. DOI: 10.1109/5.58337.

Xiong, W., G. Facelli, M. Sahebi, O. Agnel, T. Chotibut, S. Thanasilp, and Z. Holmes (2023). "On fundamental aspects of quantum extreme learning machines". In: *arXiv preprint arXiv:2312.15124*.

Yang, S. and C. Duan (1998). "Generalized synchronization in chaotic systems". In: *Chaos, Solitons & Fractals* 9.10, pp. 1703–1707.

Yu, M., Y. Liu, P. Yang, M. Gong, Q. Cao, S. Zhang, H. Liu, M. Heyl, T. Ozawa, N. Goldman, et al. (2022). "Quantum Fisher information measurement and verification of the quantum Cramér–Rao bound in a solid-state qubit". In: *npj Quantum Information* 8.1, p. 56.

Yu, Y., X. Si, C. Hu, and J. Zhang (2019). "A review of recurrent neural networks: LSTM cells and network architectures". In: *Neural computation* 31.7, pp. 1235–1270.

Chapter
A

Appendices

A.1 Open-access code repositories

The open-access code developed as part of this thesis is publicly available at:

- https://github.com/MagriLab/Stability_QRC_GS
- https://github.com/MagriLab/OptimalTraining_RFQRC

A.2 Quantum circuits visualisation

The Figs. A.1-A.4 represent the quantum circuits for different ansätze presented in Tab. 3.1. The rotation angles X_Θ represent the mapped classical data rescaled to the interval $[0, 2\pi]$.

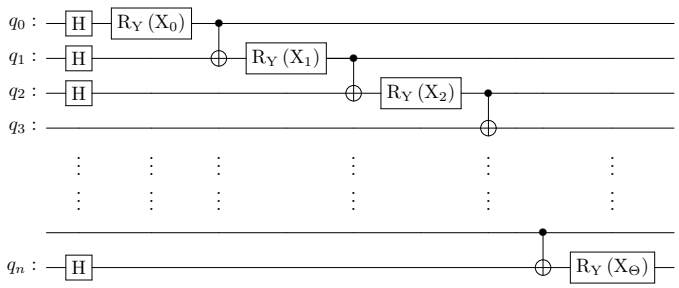


Figure A.1: Quantum circuit for linearly entangled qubits

A.3 Classical reservoir computing hyperparameters

We present the hyperparameters of the classical reservoir computer employed in Chap. 5. The performance of classical reservoir computers critically depends on the set of chosen hyperparameters (Racca and Magri

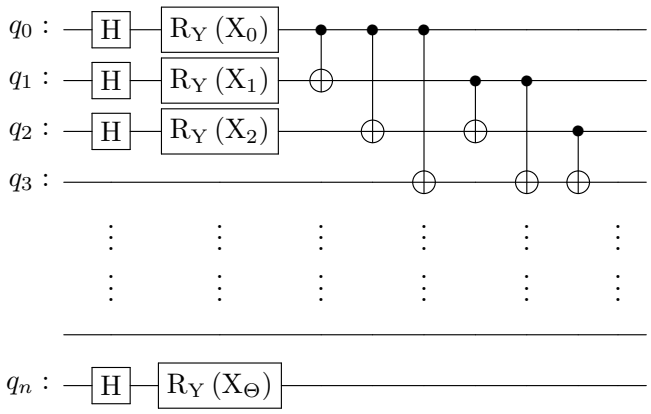


Figure A.2: Quantum circuit for fully entangled qubits

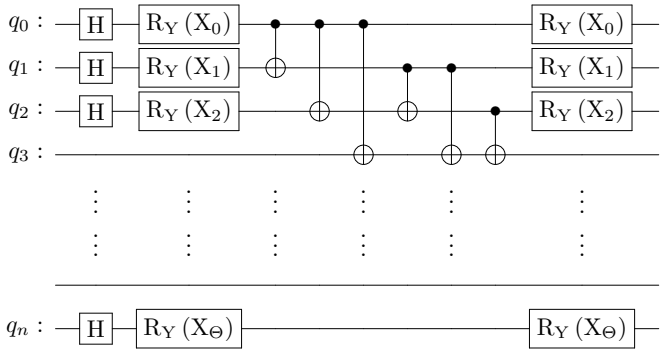


Figure A.3: Quantum circuit for fully entangled symmetric qubits

2021). As compared to QRC or RF-QRC with only two hyperparameters reported in Chap. 4, in classical reservoir computers, there are two additional hyperparameters known as the spectral radius ρ and input scaling σ_{in}. We, therefore, use grid search and Bayesian optimisation to tune the hyperparameters (Snoek, Larochelle, and Adams 2012). We have used the scikit-learn library (Pedregosa et al. 2011) and recycle validation techniques (Racca and Magri 2021) for hyperparameter tuning in classical reservoir computers. The hyperparameter ranges for different chaotic systems are reported in Table A.1.

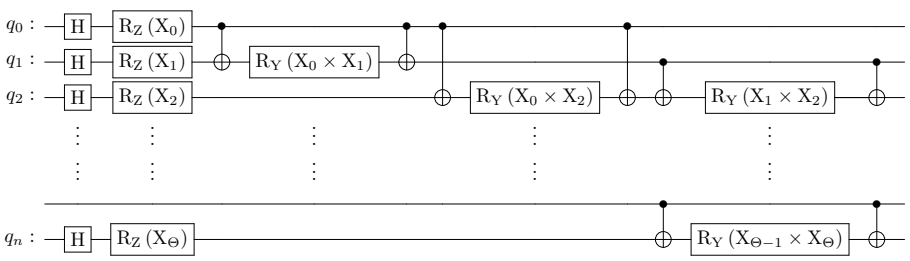


Figure A.4: Quantum circuit for product feature map qubits

Table A.1: Parameters for the Lorenz-63 and MFE system for the classical reservoir computing architecture.

Parameters	Symbol	Lorenz-63 system	MFE system
Time step	dt	0.01s	0.25s
Leading Lyapunov exponent	λ_1	0.9056	0.0163
Lyapunov Time	LT	1 LT = 110 steps	1 LT = 245 steps
Washout steps	N_w	5 LT	2 LT
Training steps	N_{tr}	20 LT	65 LT
Spectral radius	$\rho(\boldsymbol{W})$	[0.1, 1]	[0.1, 1]
Input scaling	σ_{in}	[0, 1]	[0, 1]
Tikhonov regularization	β	1×10^{-6},1×10^{-9},1×10^{-12}	1×10^{-6},1×10^{-9},1×10^{-12}
Leak rate	ϵ	[0.05, 0.3]	[0.05, 0.3]
Resevoir density	D	0.1,0.6,0.9	0.1,0.6,0.9

A.4 Memory of the reservoir networks

In this section, we discuss the empirical metrics including fading memory and linear Memory Capacity (MC) (Boyd and Chua 1985; Carroll 2022; Farkaš, Bosák, and Gergeľ 2016; Jafri, Singh, and Manchanda 2019) for different reservoir architectures. In order to efficiently reconstruct the reservoir dynamics, reservoir computers must satisfy the Echo State Property (ESP) (Jaeger 2001; Lukosevicius 2012) as discuss in Chap. 3. In classical reservoir networks, ESP is enforced by scaling the hyperparameter $\rho(\boldsymbol{W})$. In both quantum and classical reservoir computing, we perform a washout interval by discarding a limited number of initial reservoir states.

The Short-term Memory Capacity (MC) is a performance measure that is used to characterize the linear memory capacity of reservoir computers (Jaeger 2001). MC quantifies the maximally possible linear correlations between the current reservoir states and previous input data, $\boldsymbol{u}(t-d)$, and can be computed as

$$MC = \sum_{d=1}^{d_{max}} MF_d, \qquad MF_d = \frac{cov^2(\boldsymbol{u}_d(t), \boldsymbol{u}(t-d))}{\sigma^2(\boldsymbol{u}_d(t))\sigma^2(\boldsymbol{u}(t))}, \tag{A.1}$$

where $cov(\cdot,\cdot)$ is the covariance, $\sigma^2(\cdot)$ is the variance, d is the delay, $\boldsymbol{u}(t-d)$ is the delayed prediction for a particular d value and $\boldsymbol{u}_d(t)$ is the current true input value. Although the total memory capacity (MC) comprises infinitely many terms ($d_{max} = \infty$), for practical reasons it is sufficient to truncate the sum at a finite d_{max}. For a comparison of the memory capacities of CRC and QRC we choose $d_{max} = 25$. While the MC quantifies the linear memory of the reservoir, the dynamical system itself has some correlations that can increase the MC. To quantify the memory of the reservoir itself, we train our quantum and classical reservoir networks on statistically independent (i.i.d) uniform distributions. In Figs. A.5 and A.6, we compare the linear memory capacity of the classical reservoir with QRC-C1 (architecture with recurrence) and QRC-C4 (recurrence-free architecture). The global MC is bounded by the size of the reservoir and is sensitive to the choice of hyperparameters (Carroll 2022). We compute our MC for various hyperparameter values and reservoir sizes ($N = 32, 512$).

The results in Fig. A.5 and A.6 demonstrate that CRC exhibits the highest MC value, followed by QRC-C1 and then QRC-C4. A high memory capacity in CRC highlights that the reservoir contains linear

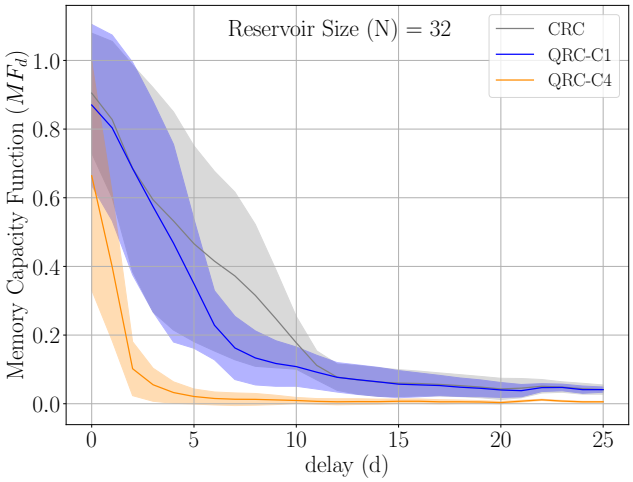


Figure A.5: Linear Memory Capacity (MC) of reservoir networks, mean and standard deviation with hyperparameters. For a summary of hyperparameters cf. Tab. 4.1. For a reservoir size $(N) = 32$.

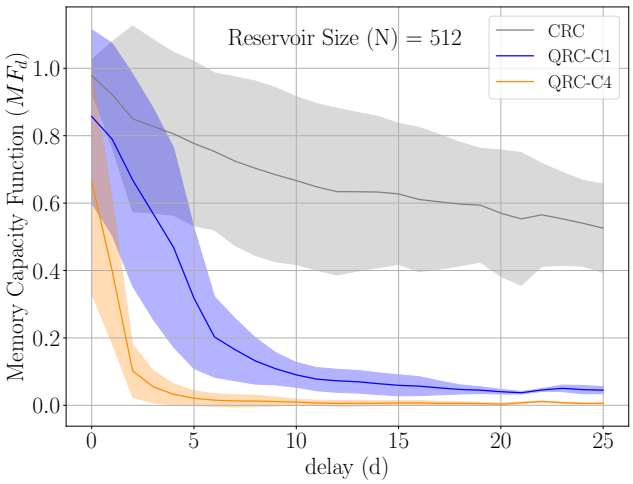


Figure A.6: Linear Memory Capacity (MC) of reservoir networks, mean and standard deviation with hyperparameters. For a summary of hyperparameters cf. Tab. 4.1. For a reservoir size $(N) = 512$.

information about the input time series for a larger delay value d. In contrast, QRC-C4 does not have any recurrent or feedback loops. Due to the removal of the recurrence reservoir memory is lost. We emphasise that the higher MC value is not indicative of better predictability. This is because of two reasons: (a) The short-term memory capacity quantifies only the linear memory of the reservoir. (b) Maximising the linear memory and nonlinear functional approximation are mutually exclusive and are referred to as the memory non-linearity trade-off (Storm, Gustavsson, and Mehlig 2022). We find comparable performances in short and long-term prediction tasks, irrespective of the MC value, for both classical and quantum reservoirs.