Imperial College London

Department of Physics

École Polytechnique Fédérale de Lausanne

Laboratory of Particle Accelerator Physics

Extrapolation of Time Evolution of Chaos Indicators Applied to Single-Particle Tracking Simulations in Circular Accelerators

Author:
Abigail Levison

Supervisors:
Tatiana Pieloni
(EPFL)
Massimo Giovannozzi
(CERN)
Carlo Emilio Montanari
(University of Manchester)

CERN-THESIS-2024-119
11/07/2024

Word Count: 12448

Thesis submitted for the degree of

MSci Physics with a Year Abroad

July 2, 2024

Abstract

Optimising lattice designs for circular colliders, such as the CERN Large Hadron Collider (LHC), is critical to maximising performance and luminosity. Given the limited computational resources for tracking particle orbits in accelerators, it is essential to extract maximum information from short-term data. Fast identification of chaotic behaviour in particle orbits is a novel approach that could lead to better investigation of diffusive behaviour, leading to better understanding of long-term beam losses. Various chaos indicators have been proposed for this purpose. This thesis explores the potential of extrapolation techniques to enhance the classification performance of chaos indicators in single-particle tracking simulations of circular accelerators.

The time evolution of the Fast Lyapunov Indicator (FLI) and the Reversibility Error Method (REM) are simulated using a modulated Hénon map over 10^8 turns. The values of these two chaos indicators at 10^8 provide a ground truth classification of regular or chaotic behaviour. Classifications based on shorter tracking times are compared to the ground truth to assess accuracy, which provides an objective benchmark assessment of the predictive power of the indicator. Extrapolation techniques using Auto-Regressive Integrated Moving Average (ARIMA) models are implemented to predict the time evolution of these indicators up to 10^8 turns.

This study provides a first approach to the use of extrapolation techniques and explores the efficacy of ARIMA models. The results show that ARIMA models, while achieving a slight improvement in accuracy, ultimately did not improve the classification accuracy for either FLI or REM. The considerations and characterisation of the data provided in this work can be transferred to the use of alternative extrapolation techniques.

Résumé

L'optimisation des conceptions de réseaux pour les collisionneurs circulaires, tels que le LHC (Large Hadron Collider) du CERN, est essentielle pour maximiser les performances et la luminosité. Étant donné les ressources informatiques limitées pour le suivi des orbites particulaires dans les accélérateurs, il est essentiel d'extraire un maximum d'informations à partir des données à court terme. L'identification rapide du comportement chaotique des orbites particulaires est une approche novatrice qui pourrait faciliter une meilleure investigation du comportement diffusif, permettant une meilleure compréhension des pertes de faisceau à long terme. Divers indicateurs de chaos ont été proposés à cet effet. Cette thèse explore le potentiel des techniques d'extrapolation pour améliorer les performances de classification des indicateurs de chaos dans les simulations de suivi des particules uniques des accélérateurs circulaires.

L'évolution temporelle de l'indicateur de Liapounov rapide (FLI) et de la méthode d'erreur de réversibilité (REM) est simulée en utilisant un attracteur de Hénon modulé sur 10^8 tours. Les valeurs de ces deux indicateurs de chaos à 10^8 tours fournissent une classification de référence du comportement régulier ou chaotique. Les classifications basées sur des temps de suivi plus courts sont comparées à cette référence pour évaluer la précision, ce qui fournit une évaluation objective de la puissance prédictive de l'indicateur. Des techniques d'extrapolation utilisant des modèles de moyenne mobile autorégressive intégrée (ARIMA) sont mises en œuvre pour prédire l'évolution temporelle de ces indicateurs jusqu'à 10^8 tours.

Cette étude propose une première approche pour l'utilisation des techniques d'extrapolation et explore l'efficacité des modèles ARIMA. Les résultats montrent que les modèles ARIMA, bien qu'ils aient légèrement amélioré la précision, n'ont finalement pas dépassé la précision de classification de référence pour le FLI ou le REM. Les considérations et la caractérisation des données ici présentées peuvent être transférées à l'utilisation de techniques d'extrapolation alternatives.

Preface

This Master's thesis project was carried out during my exchange from Imperial College London to the École Polytechnique Fédérale de Lausanne (EPFL), in affiliation with the Laboratory of Particle Accelerator Physics (LPAP) at EPFL. The work took place at CERN within the Beams Department, Accelerator and Beams Physics (ABP) group, Non-linear Dynamics and Collimation (NDC) section. This thesis work is independent but builds on a recent paper by A. Bazzani et al., which provides the state-of-the-art performance analysis of chaos indicators applied to non-linear accelerator systems. Data for the analysis in this thesis were generated using Xsuite, a Python library developed by CERN for general accelerator physics simulations, Xdyna, a submodule providing chaos indicator measurements, and C++ code written by C. E. Montanari. I prepared scripts using the functions and libraries provided to generate the data, and used GPUs managed by CERN's HTCondor system for single-particle tracking simulations. The analysis reproducing state-of-the-art performance results and the implementation of ARIMA models to extrapolate the time evolution of chaos indicators were conducted independently by me.

Contents

Chapter 1

Introduction

Optimising lattice designs for circular colliders is essential for enhancing their performance, particularly in maximising luminosity. This is particularly relevant for large superconducting colliders such as the CERN Large Hadron Collider (LHC) [1] and its high-luminosity upgrade, the HL-LHC [2], which operate high-energy beams using superconducting magnets. For example, HL-LHC beams are designed to reach a stored beam energy of 700 MJ [3]. Even a small fraction of this energy deposited on superconducting magnets can cause quenches or equipment damage, thus motivating a deep understanding of the dynamics of beam losses.

The orbit of a particle through the magnetic lattice of a circular accelerator can be modelled with a one-turn map, which represents the Poincaré section of the machine, with the various magnetic elements described with the thin-lens approximation [4]. The standard method for evaluating the quality of an accelerator lattice is through single-particle tracking: the particles' orbits are individually tracked, neglecting any interaction between the charged particles, and the volume in phase space in which particles are observed to have bounded orbits, the so-called dynamic aperture, is evaluated [5]. With unlimited computational power, the long-term performance of a realistic LHC lattice would be assessed by tracking the particle orbits for more than 10^8 turns, corresponding to the typical LHC fill time of approximately 10 hours, and the combination of initial conditions maximising the dynamic aperture would be determined. However, current computational limitations make such tasks impossible to tackle directly. Standard LHC tracking simulations consider up to 10^6 turns, which corresponds to approximately 90 seconds of LHC runtime [6]. This leads to the challenge of extrapolating the long-term performance of the accelerator lattice up to 10^8 turns with the information available at lower tracking times.

Rather than only inspecting the dynamic aperture, one could inspect the regions in phase space where particles exhibit chaotic behaviour. The presence of chaos can then be linked to possible diffusive behaviour in the beam dynamics, which can then be used to try to predict beam losses. For the fast detection of chaotic behaviour, various chaos indicators have been developed (see, e.g. [7] and references therein). Chaos indicators are functions computed over a relatively small number of turns that aim to quantify the chaotic behaviour of an orbit, i.e. its Lyapunov exponent. They have recently been applied to accelerator physics as tools to improve lattice inspection and optimisation. In this domain, finding a faster method to identify chaos in accelerator lattices can improve the investigation of the relationship between chaotic phase-space regions and beam-loss dynamics in circular accelerators. To

achieve this, we aim to use extrapolation techniques to extract as much information as possible from chaos indicators.

The first aim of this project is to simulate the time evolution of various chaos indicators over 10^8 turns, using a modulated Hénon map, which is a simplified but meaningful one-turn map that preserves most of the features of a non-linear accelerator lattice [4], [5]. Although it is not computationally feasible to perform single-particle tracking on a realistic one-turn map for up to 10^8 turns, it is possible to reach this number of turns on a modulated Hénon map. The values of a given chaos indicator at 10^8 turns, comparable to typical LHC runtime [6], provide a ground truth binary classification of regular or chaotic behaviour. The accuracy over time of the binary classification after tracking up to a smaller number of turns, compared to the ground truth classification, will serve as an objective assessment to improve upon.

The second aim is to implement Auto-Regressive Integrated Moving Average (ARIMA) [8] models to extrapolate the time evolution of various chaos indicators, tracked up to smaller numbers of turns, up to 10^8 turns. The accuracy of the classification after tracking and extrapolation will be compared with the benchmark accuracy, in the hope that the tracking with extrapolation will be accurate and more computationally efficient than single-particle tracking up to 10^8 turns.

Chapter 2

Theoretical Background

2.1 Modelling Accelerator Lattices

This section covers the fundamental elements of accelerator physics for the thesis. It explains how Hamiltonian mechanics describes an accelerator lattice, where the linear part of the system consists of dipole and quadrupole magnets, and higher-order elements introduce non-linear beam dynamics.

2.1.1 Particle Accelerators

A particle accelerator accelerates beams of charged particles in a vacuum pipe to a specific energy. Beams may collide with a fixed target or a counter-rotating beam. Particles gain momentum via an oscillating electric field (RF pulses) and travel in either a linear or circular path. In circular accelerators, dipole magnets create a circular path, while quadrupole magnets focus the beam and adjust its size. Higher-order magnets correct focusing imperfections and manage energy deviations.

2.1.2 Frenet-Serret Coordinate System

The standard coordinate system used to model circular accelerators is the Frenet-Serret coordinate system (x, y, s) [4]. An illustration of this coordinate system is shown in Fig. 2.1. The coordinate s points along the path of the ideal circular orbit of the particle with radius ρ, defined as the reference orbit. The coordinates x, which point outward in the radial direction, and y, which point vertically upward, are used to define the directions of the particle displacements around the design orbit.

This can be better understood by considering the transformation from Cartesian coordinates (X, Y, Z) to Frenet-Serret coordinates (x, y, s), with the origin in the centre of the accelerator. The angle along the ideal orbit is given by s/ρ, which can be used to define the transformation in a similar way to cylindrical polar coordinates. The transformation is given by

$$X = (x + \rho)\cos\left(\frac{s}{\rho}\right), \quad Y = y, \quad Z = (x + \rho)\sin\left(\frac{s}{\rho}\right). \tag{2.1}$$

A particle in a circular accelerator is kept in an orbit of constant radius ρ with the magnetic field of the dipoles. Equating the centripetal force $m\gamma v^2/\rho$ to the field

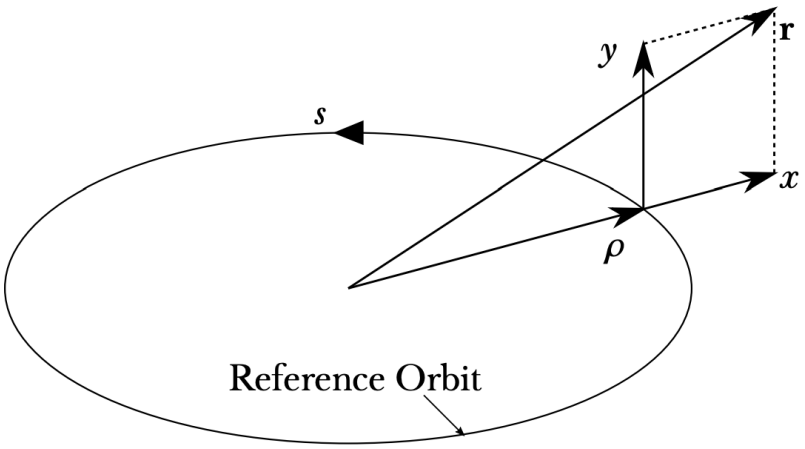


Figure 2.1: Schematic of the Frenet-Serret coordinate system, taken from [9].

provided by the modulus of the Lorentz force qvB, where v is the particle's velocity, leads to

$$B\rho = \frac{p}{q}\,, \tag{2.2}$$

where p is the particle's (relativistic) momentum, q is the particle's charge, and $B\rho$ is referred to as the magnetic rigidity. It is important to note that as the particles are accelerated, their momentum increases, and therefore it is essential to increase the magnetic field along with the particle's momentum to keep ρ constant.

2.1.3 Hamiltonian Mechanics

This subsection introduces the fundamental elements of the Hamiltonian treatment of an accelerator system, and how its magnetic field can be treated by means of a power series. For a complete derivation of the Hamiltonian and the equations of motion, please refer to [4], [9].

Starting from the Hamiltonian expression of a relativistic charged particle, that is

$$\mathcal{H} = e\Phi + \sqrt{m^2c^4 + (c\mathbf{p} - e\mathbf{A})^2}\,, \tag{2.3}$$

one can perform a change of variables into the Frenet-Serret coordinate system, and treat the coordinate s as the time variable of the Hamiltonian. When considering high-energy particle beams, it is valid to assume that the overall momentum is mostly along the direction of the orbit. Even if the particle is oscillating around the design orbit, its momentum from the oscillations is much less than the momentum of the orbit, such that $p \gg p_x$ and $p \gg p_y$. It is also assumed that there is no magnetic field in the s direction, so that $\mathbf{B} = (B_x, B_y, 0)$. Consequently, when expressing the magnetic field in terms of a magnetic vector potential using $\mathbf{B} = \nabla \times \mathbf{A}$, the only non-zero component of $\mathbf{A}$ is that in the s direction [9]. Applying these assumptions, allows the Hamiltonian to be approximated as

$$\mathcal{H} = \left(1 + \frac{x}{\rho}\right)\left[-p + \frac{1}{2p}\left(p_x^2 + p_y^2\right)\right] - eA_s\,, \tag{2.4}$$

where e is the charge of the particle, A_s is the component of the magnetic vector potential in the s direction, and ρ is the radius of the design orbit.

The first step to finding the contribution of the magnetic field eA_s to the Hamiltonian is to use Maxwell's conditions of zero divergence and zero curl in the case of our accelerator model to derive Laplace's equation $\nabla^2 \mathbf{A} = 0$. The solution to this equation can be expressed as a power series, which gives A_s. This leads to the multipolar expansion of the magnetic field given by

$$B_y(x,y;s) + iB_x(x,y;s) = B_0 \left(\rho_0 \sum_{n=1}^{M} \left[k_n(s) + ij_n(s) \right] \frac{(x+iy)^n}{n!} - \kappa(s) \right) , \quad (2.5)$$

where B_0 is the magnetic field at the reference radius ρ_0, $\kappa(s)$ is a function equal to 1 in dipoles and 0 elsewhere, and k_n and j_n are defined as the normal and skew polar coefficients of the magnetic field. They are defined in terms of the derivatives of the magnetic field and are given by

$$k_n(s) \equiv \frac{1}{B_0 \rho_0} \frac{\partial^n B_y}{\partial x^n}\bigg|_{(0,0;s)} \qquad j_n(s) \equiv \frac{1}{B_0 \rho_0} \frac{\partial^n B_x}{\partial x^n}\bigg|_{(0,0;s)} , \quad (2.6)$$

where B_x and B_y are the components of the magnetic field in the x and y directions respectively. The normal and skew coefficients can be considered as equivalent to the strength of the magnets, where k_0 corresponds to dipoles, k_1 to quadrupoles, k_2 to sextupoles, and k_3 to octupoles, and so on. For a simple accelerator model, skew magnets are not considered, so $j_n = 0$, and it is conventional to assume that each magnet generates a field corresponding to only one multipolar order. In real life, and when modelling non-linear elements, the field generated by one magnet may be expressed by several multipolar coefficients [9].

From the Hamiltonian, the equation of motion for a particle's motion in the x and y directions in an accelerator can be derived using Hamilton's equations. The equations of motion are given by

$$\begin{aligned} x'' &= \frac{1}{\rho} + \frac{x}{\rho^2} + \frac{B_y}{B\rho} \left(1 + \frac{x}{\rho} \right)^2 , \\ y'' &= \frac{B_x}{B\rho} \left(1 + \frac{x}{\rho} \right)^2 , \end{aligned} \quad (2.7)$$

where the prime denotes differentiation with respect to the coordinate s, and B is the magnetic field at the reference radius ρ_0. Considering only the linear terms in B_x and B_y and discarding the higher-order terms implies that the magnetic field is considered to be purely dipolar and quadrupolar. This is convenient because it allows the equations of motion to be expressed in the following form

$$z'' + K_z(s) z = 0 , \quad (2.8)$$

where z represents either x or y, and $K(s)$ is a function of the normal polar coefficients. Depending on the value of $K(s)$, the motion changes according to the effects of different magnetic elements. For a circular accelerator, the condition $K(s) = K(s+L)$ is imposed, where L is the accelerator length. The resulting equations then follow the Hill equation format.

The equations of motion are now decoupled, and a particle's motion can be described with the following solution to Hill's equation:

$$z(s) = A\sqrt{\beta_z(s)}\cos\left(\omega_z(s)\right) \ . \tag{2.9}$$

These harmonic oscillations are referred to in accelerator physics as betatron oscillations, with s-dependent amplitude $\beta_z(s)$ and angular phase $\omega_z(s)$, known as phase advance.

2.1.4 One-Turn Maps

A circular accelerator is modelled as a continuous Hamiltonian system, treating s as the time-like coordinate. The phase-space coordinates may be sampled at discrete values of s with the use of a Poincaré section. This is the plane in phase space in which $s = 0$, a choice made to represent the starting point of the accelerator. Particles leave the Poincaré section when travelling through the accelerator and return after one full turn with new phase-space coordinates. A one-turn map is the Poincaré map of a circular accelerator at this section [10]. It is a function that maps the phase space coordinates to the Poincaré section after one complete turn. Figure 2.2 visually illustrates the discrete sampling of the phase-space coordinates at $s = 0$ after each turn.

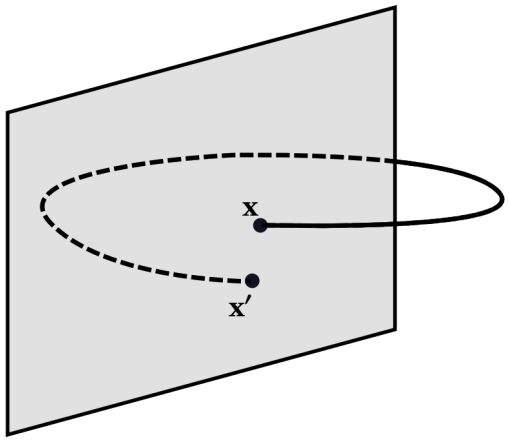


Figure 2.2: Schematic of a Poincaré section, taken from [9], adapted from [4]. A particle $\mathbf{x}$ is mapped after one full turn to new coordinates $\mathbf{x}'$.

A one-turn map is computed by multiplying the matrices that represent the different elements in the accelerator. For example, in an accelerator with L elements, the one-turn map is given by

$$\mathbf{M} = \mathbf{M}^{(L)} \circ \mathbf{M}^{(L-1)} \circ \cdots \circ \mathbf{M}^{(2)} \circ \mathbf{M}^{(1)} \, , \tag{2.10}$$

where $\mathbf{M}^{(i)}$ is the matrix representing the ith element in the accelerator. Each matrix may represent a drift space without a magnetic field or a magnet of some order [9]. Element matrices may be derived in the thin lens approximation following the principles of ray optics, treating a particle's trajectory as a ray and characterising the magnets as thin lenses [4].

When considering an accelerator lattice made of only dipoles and quadrupoles, the one-turn map corresponds to an uncoupled rotation in phase space given by

$$\begin{pmatrix} x_{n+1} \\ p_{x,n+1} \\ y_{n+1} \\ p_{y,n+1} \end{pmatrix} = \begin{pmatrix} \cos\omega_x & \sin\omega_x & 0 & 0 \\ -\sin\omega_x & \cos\omega_x & 0 & 0 \\ 0 & 0 & \cos\omega_y & \sin\omega_y \\ 0 & 0 & -\sin\omega_y & \cos\omega_y \end{pmatrix} \times \begin{pmatrix} x_n \\ p_{x,n} \\ y_n \\ p_{y,n} \end{pmatrix}, \tag{2.11}$$

where the angles $\omega_{x,n}$ and $\omega_{y,n}$ are the phase advances in the betatron oscillations in the x and y directions, respectively. Note that in accelerator physics, it is customary to build an accelerator lattice starting from the repetition of a periodic structure called a FODO cell, which consists of a focusing quadrupole, a drift space, a defocusing quadrupole, and another drift space.

In an accelerator, non-linear effects come from either magnet imperfections or from the deliberate addition of higher-order magnets such as sextupoles, for chromaticity control, or octupoles, for compensation of field errors and stabilisation of collective beam instabilities. In the case of magnetic imperfections, the magnetic field is not purely dipolar and quadrupolar, and the non-linear elements of the magnetic field, wanted and unwanted, are responsible for the non-linear motion of particles in an accelerator.

The standard approach in computer simulations to approximate one-turn maps that include non-linear elements is to use the one-kick approximation. This means considering a non-linear magnet at a set position s' in the accelerator, with all effects applied following a Dirac $\delta(s - s')$ function [4]. The one-kick approximation is applied in this study by considering a non-linear magnet at the very beginning of the accelerator, providing a kick Δ such that the phase-space coordinates after the kick are given by $\mathbf{X_n} + \Delta$. After the kick is applied, the particles undergo a rotation in phase space, such that the new phase space coordinates $\mathbf{X}_{n+1}$ are given by

$$\mathbf{X}_{n+1} = R\left(\omega_{x,n}, \omega_{y,n}\right)\left(\mathbf{X}_n + \Delta\right), \tag{2.12}$$

where $R\left(\omega_{x,n}, \omega_{y,n}\right)$ is the same rotation matrix defined in Eq. (2.11).

A generic one-turn map can then be summarised as the product of matrices, representing the FODO cell, and matrices representing the approximated thin-lens higher-order magnets.

A simple, yet effective, accelerator-like one-turn map, containing a FODO cell and a non-linear kick, reads:

$$\begin{pmatrix} x_{n+1} \\ p_{x,n+1} \\ y_{n+1} \\ p_{y,n+1} \end{pmatrix} = R\left(\omega_{x,n}, \omega_{y,n}\right) \times \begin{pmatrix} x_n \\ p_{x,n} + x_n^2 - y_n^2 + \mu\left(x_n^3 - 3x_n y_n^3\right) \\ y_n \\ p_{y,n} - 2x_n y_n + \mu\left(y_n^3 - 3y_n x_n^3\right) \end{pmatrix} \tag{2.13}$$

where μ is a variable parameter equivalent to k_3, or the octupolar strength. If $\mu = 0$, the kick is represented by the fixed quadratic part of the polynomial, corresponding to a sextupolar kick [5].

This map follows the same format as a Hénon map [11], which is generally expressed as

$$\begin{pmatrix} x_{n+1} \\ p_{n+1} \end{pmatrix} = R\left(\omega\right) \times \begin{pmatrix} x_n \\ p_n + f\left(x_n\right) \end{pmatrix} \tag{2.14}$$

A modulation in the angles of the rotation matrix can be incorporated into the Hénon map as a refinement that further enhances the model's realism. Frequency perturbations known as "ripples" occur in the power supply transmitted to the magnets in an accelerator, thereby influencing the magnetic field and subsequently impacting particle focusing. In the context of the Hénon map, this modulation is accounted for by varying ω_x and ω_y with the number of turns n, as follows:

$$\begin{aligned} \omega_{x,n} &= \omega_{x,0}\left(1+\varepsilon\sum_{k=1}^{m}\varepsilon_k\cos\left(\Omega_k n\right)\right) \\ \omega_{y,n} &= \omega_{y,0}\left(1+\varepsilon\sum_{k=1}^{m}\varepsilon_k\cos\left(\Omega_k n\right)\right) \end{aligned} \tag{2.15}$$

where ε is the modulation amplitude [12], and ε_k and Ω_k are hard-coded parameters based on data taken from observations of tune modulation due to the observed frequency ripple in the CERN Super Proton Synchrotron [5].

The Hénon map represents only one FODO cell with one non-linear magnet. For comparison, the one-turn map for a realistic LHC lattice contains over $10,000$ elements. Despite its simplicity, the Hénon map exhibits many characteristics that are expected to be similar in a realistic accelerator model. This makes it an ideal tool for testing and benchmarking scale laws and extrapolation methods to study long-term dynamics [4], [5]. It is preferable to use the modulated Hénon map to probe as much information as possible about particle behaviour in this simplified lattice before moving on to more computationally expensive simulations.

2.2 Detecting Chaos

This section introduces the concept of chaos, and how chaotic behaviour can be quantified in single-particle tracking simulations with chaos indicators.

2.2.1 Lyapunov Exponent

For two initial conditions $\mathbf{x}_0$ and $\mathbf{x}_0+\epsilon\xi$, the magnitude of their initial separation ϵ varies over time at a rate that can be expressed in the following form

$$\|(\mathbf{x}_0+\epsilon\xi)_n-\mathbf{x}_n\|\approx\epsilon e^{\lambda n}\,, \tag{2.16}$$

where $\mathbf{x}_n$ represents the condition at time n, and λ is defined as the Lyapunov exponent, which is given by

$$\lambda=\lim_{n\to\infty}\lim_{\epsilon\to 0}\frac{1}{n}\ln\frac{\|(\mathbf{x}_0+\epsilon\xi)_n-\mathbf{x}_n\|}{\epsilon} \tag{2.17}$$

where the expression inside the logarithm is the linear response of the system to the perturbation $\epsilon\xi$. In other words, it is the the rate of growth in magnitude of the initial displacement vector $\epsilon\xi$. In the case of Hamiltonian systems, such as accelerator lattice models, the Lyapunov exponent is always non-negative [13]. Each particle in an accelerator has its own orbit with a corresponding Lyapunov exponent. If $\lambda=0$, the separation between the particles maintains the same order

of magnitude, and the particle's orbit is defined as regular. Conversely, $\lambda > 0$ implies an exponential growth, and the particle's orbit is defined as chaotic.

The Lyapunov exponent has dimensions of inverse time, although in this framework it is represented in units of the number of turns, which makes it a dimensionless quantity. Its reciprocal gives the Lyapunov time, which represents the timescale over which particles exhibit chaotic behaviour.

The separation between particles in phase space may evolve at different rates depending on the direction of the initial separation vector and the nature of the orbit. This behaviour can be described by means of a spectrum of Lyapunov exponents, which has the same dimension as the phase space under consideration. The general separation rate described in Eq. (2.16) is then related to the largest Lyapunov exponent in the spectrum, also referred to as the Maximal Lyapunov Exponent.

Since it is mathematically defined in the limit of infinite time, classifying particle orbits as regular or chaotic after a finite number of turns means that the value of the Lyapunov exponent must be estimated by means of chaos indicators.

2.2.2 Fast Lyapunov Indicator (FLI)

The Fast Lyapunov Indicator (FLI) [14] is a chaos indicator that provides a computation of the Maximal Lyapunov Exponent defined for finite time. It is given by the logarithm of the linear response, as shown in Eq. (2.17), evaluated at a finite time n. For an initial condition $\mathbf{x}_0$, and a displaced initial condition $\mathbf{y}_0 = \mathbf{x}_0 + \epsilon\xi$, tracked using a one-turn map $\mathbf{M}$, the map of the displaced initial condition $\mathbf{y}_1 = \mathbf{M}(\mathbf{x}_0 + \epsilon\xi)$ can be written as a first-order Taylor expansion of the map $\mathbf{M}$, given by

$$\mathbf{M}(\mathbf{x}_0 + \epsilon\xi) = \mathbf{M}(\mathbf{x}_0) + DM(\mathbf{x}_0)\epsilon\xi \tag{2.18}$$

where DM is the Jacobian matrix, also known as a tangent map. The initial displacement vector $\epsilon\xi$ evolves in magnitude and direction after every turn according to the tangent map, evaluated at each value of n. Therefore, the map of the displaced condition after n turns $\mathbf{y}_n$ is given by:

$$\mathbf{y}_n = \mathbf{M}_n(\mathbf{x}_0) + DM\left(\mathbf{x}_{n-1}\right) \times DM\left(\mathbf{x}_{n-2}\right) \times ... \times DM\left(\mathbf{x}_1\right) \times DM\left(\mathbf{x}_0\right) \epsilon\xi \tag{2.19}$$

where $\mathbf{M}_n(\mathbf{x}_0)$ represents the one-turn map applied n times to the initial condition $\mathbf{x}_0$. The linear response, which is equal to the first-order derivative of $\mathbf{M}_n$, is therefore given by

$$\begin{aligned} \boldsymbol{\Xi}_n(\mathbf{x}) &= \lim_{\epsilon \to 0} \frac{\mathbf{y}_n - \mathbf{x}_n}{\epsilon} \\ &= DM\left(\mathbf{x}_{n-1}\right) \times DM\left(\mathbf{x}_{n-2}\right) \times ... \times DM\left(\mathbf{x}_1\right) \times DM\left(\mathbf{x}_0\right) \xi \\ &= \mathrm{L}_n(\mathbf{x_0})\xi. \end{aligned} \tag{2.20}$$

where the concatenations of DM are represented in a more compact way by $\mathrm{L}_n(\mathbf{x_0})$. Taking the logarithm of the linear response then gives the FLI as follows:

$$\mathrm{FLI}_n\left(\mathbf{x}_0, \xi\right) = \ln \left\| \mathrm{L}_n\left(\mathbf{x}_0\right) \xi \right\| \tag{2.21}$$

The logarithm of the linear response is proportional to the Maximal Lyapunov Exponent multiplied by the number of turns n. Since the Lyapunov exponent is mathematically defined as the *time average* of the logarithm of the linear response, it can

be estimated by dividing the FLI by the number of turns n. For large numbers of turns, the value of FLI_n/n tends to the Maximal Lyapunov exponent [12]. Consequently, for a regular orbit, the value of FLI_n/n tends to zero, and for chaotic orbits the value of FLI_n/n tends to a positive value.

2.2.3 Reversibility Error Method (REM)

Due to numerical rounding and truncation errors, a particle, after tracking and backtracking, will land some distance away from its exact initial condition. Reversibility Error Method (REM) [15] measures this distance and uses this inevitable feature of numerical simulation as a tool to evaluate the chaotic behaviour of a particle.

In REM, a particle is first tracked with the one-turn map from 0 to n turns, and then backtracked from n turns back to 0 turns by means of the inverse one-turn map. The Euclidean distance is then evaluated between the initial condition and the condition after tracking and backtracking. This gives the value of REM at n turns for a single particle, given by

$$\mathrm{REM} = \sqrt{(\tilde{x}_0 - x_0)^2 + (\tilde{p}_{x,0} - p_{x,0})^2 + (\tilde{y}_0 - y_0)^2 + (\tilde{p}_{y,0} - p_{y,0})^2} \tag{2.22}$$

where $\tilde{x}_0$, $\tilde{p}_{x,0}$, $\tilde{y}_0$, and $\tilde{p}_{y,0}$ are the phase space coordinates after tracking the particle for a given number of turns and backtracking to 0 turns.

A regular particle will manifest a growth in REM following a power law, while instead a chaotic particle will show a faster exponential growth. The time scale for this exponential growth is determined by the Maximal Lyapunov Exponent [15]. These two trends are visible on a logarithmic scale.

2.3 ARIMA

2.3.1 General form of an ARIMA Model

An ARIMA model is an established extrapolation tool capable of capturing multiple characteristics of a time series using a minimal number of free parameters. In the context of this thesis, it will be used to extrapolate the time evolution of various chaos indicators. It is made up of any linear combination of an autoregressive (AR) model and a moving average (MA) model, with the possible inclusion of an integrated (I) part.

It is important to note that the autoregressive and moving average components of an ARIMA model assume that the time-series data is *stationary*. This means that the mean and variance of the data do not change with time. The integrated (I) component of an ARIMA model ensures that the data is stationary before fitting the autoregressive and moving average components. A time series is made stationary by differencing the data, which means taking each data point z at time t and subtracting the previous data point, creating a new array of data z' where

$$z'_t = z_t - z_{t-1} \tag{2.23}$$

until the data is sufficiently stationary. When considering discrete data points and time intervals, differencing is equivalent to differentiation. Differencing the data two times is the analogue equivalent of taking the second derivative. The data used

for an ARIMA model may be differenced zero, one, or two times [8]. Differencing the data more than twice may amplify the noise present in the time series, which increases the variance of the data, and usually leads to overly sensitive ARIMA models, which result in over-fitting.

Once the time series is stationary, a linear combination of an autoregressive (AR) and a moving average (MA) model can be fitted to the data. This model may include only an AR component, only an MA component, both components, or neither (the latter corresponds to a random-walk model, where each data point equals the previous one plus some noise). In a stationary time series with mean μ, a data point z_t at time t depends on white noise terms a and, if an AR component is included, on past values of z, known as lags. White noise a is sampled independently at different times t from a normal distribution with a mean of zero and a variance of σ_a^2 [8].

An autoregressive (AR) model of order p assumes that a data point z at time t is a finite linear combination of p previous z values, plus some white noise. The data point z_t is given by

$$z_t = \mu + a_t + \phi_1 z_{t-1} + \phi_2 z_{t-2} + ... + \phi_p z_{t-p}\,, \tag{2.24}$$

where ϕ are independent coefficients known as weight parameters.

A moving average (MA) model of order q assumes that a data point z at time t is a finite linear combination of the previous q white noise terms. This implies that z_t is a regression in error terms and can be interpreted as the sum of q lagged forecast errors. The data point z_t is given by

$$z_t = \mu + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - ... - \theta_q a_{t-q}\,, \tag{2.25}$$

where the convention is that the independent θ parameters are defined such that the sign in front of them is negative, but the sign of θ itself can be either.

If the ARIMA model includes both an autoregressive and a moving average component, then the data point z_t at time t is given by the combination of both models, plus the mean and the white noise. The data point z_t is given by

$$z_t = \mu + a_t + \phi_1 z_{t-1} + \phi_2 z_{t-2} + ... + \phi_p z_{t-p} - \theta_1 a_{t-1} - \theta_2 a_{t-2} - ... - \theta_q a_{t-q}\,, \tag{2.26}$$

where the data point z_t depends on the past p values of z and the past q values of a. An ARIMA model of order (p, d, q) contains $p+q+2$ parameters that are optimised to fit the model to the observed data: ϕ, θ, μ, and σ_a^2.

2.3.2 Fitting an ARIMA Model

To select an optimal ARIMA order (p, d, q), the conventional method involves differencing the time series until stationary, then plotting the partial autocorrelation function (PACF) and autocorrelation function (ACF) for the stationary series. These plots are examined to determine the values of p and q based on the last significant lags showing high partial autocorrelation and high autocorrelation, respectively. The ACF and PACF plots are checked for each d to confirm the optimal differencing and avoid overfitting [8].

Once the optimal order is determined for each time series, an ARIMA model is fitted to the provided data by optimising the values of the parameters θ, ϕ, μ, and σ_a^2 with maximum likelihood estimation.

ARIMA extrapolation operates on a one-step-ahead basis, predicting the next data point based on the last p or q data points, whichever is greater. However, as the extrapolation progresses further into the future, at some point, the model input will consist purely of previously extrapolated points rather than actual data. Consequently, if there was an existing trend before the extrapolation began, ARIMA will perpetuate this trend indefinitely, without the ability to introduce a turning point.

Chapter 3

Reproduction of the State-of-the-Art Performance Analysis of Chaos Indicators

In the paper by Bazzani et al. [12], a performance comparison of chaos indicators computed for a modulated Hénon map is presented. In their analysis, a modulated Hénon map is used to track particles up to 10^8 turns, comparable to the typical LHC fill time [6].

Particles were initialised across a broad range of conditions to capture regular, chaotic, and lost cases. A binary ground truth classification of regular or chaotic behaviour is established by placing a threshold in the histogram distribution of the chaos indicator values at 10^8 turns. The position of the threshold is determined by a KDE-based algorithm which assumes that the chaos indicator values will form a clear bimodal distribution, populated by the two types of orbit. When a classification is performed after tracking up to a smaller number of turns, the ratio between the correctly classified particles and the total number of particles gives the accuracy for that number of turns [12]. In the context of this thesis, it provides a benchmark classification accuracy that can be improved with extrapolation techniques. It is important to note that only particles that have survived up to 10^8 turns are considered in any classification used in the performance analysis.

The state-of-the-art analysis considers a dynamic indicator to have a high performance if its classification accuracy is high after a relatively low number of turns [12]. The chaos indicators chosen for this study were selected as a combination of being the best performing (REM) and being suitable for extrapolation (FLI).

In this chapter, the main results for FLI and REM are reproduced, and additional considerations on methods for producing and classifying histogram chaos indicator distributions are discussed. The analysis uses $10,000$ particles initialised in a Cartesian grid from 0 to 0.4, capturing a broad range of stability conditions. About half survive up to 10^8 turns and are used for binary classifications.

3.1 Kernel Density Estimation (KDE) Classification Algorithm

Kernel Density Estimation (KDE) [16] is a statistical technique used to estimate the probability density function (PDF) of an unknown distribution. It is used in the context of this thesis to estimate the PDF of the histogram distributions of the chaos indicator value at a given number of turns. The KDE is used to determine the position of a threshold in the histogram distribution, categorising each initial condition as either regular or chaotic.

The KDE relies on an assumed kernel function, typically Gaussian, and a manually selected parameter called the bandwidth. The bandwidth determines the smoothness of the estimated density function, balancing the trade-off between bias and variance in the estimation [16]. The estimated density function is given by

$$f_h(x) = \frac{1}{n} = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{3.1}$$

where h is the bandwidth, n is the number of data points, and K is the Kernel, a function chosen based on the observed shape of the distribution.

An example of the KDE algorithm used in state-of-the-art analysis is shown in Fig. 3.1. The algorithm tests bandwidths, starting wide and decreasing iteratively, to count peaks in the KDE. It finds the first bandwidth with three peaks and selects the previous one with two peaks. The minimum amplitude between the two peaks is then set as the threshold. For the accelerator systems considered in this study, there are more regular than chaotic initial conditions, so the initial conditions with chaos indicator values on the side of the larger peak are classified as regular, and those on the side of the smaller peak are classified as chaotic.

The state-of-the-art classification algorithm is designed to evaluate how quickly a chaos indicator produces a bimodal histogram distribution. For probing the chaos indicator's mathematical information, Silverman's rule of thumb may offer a more suitable approach.

Silverman's rule of thumb allows the approximation of the bandwidth h, a method used as a compromise between accurate PDF estimation and overfitting. The approximation for h is given by

$$h = 0.9 \times \min(\sigma, \frac{\text{IQR}}{1.34}) \times n^{-1/5} \tag{3.2}$$

where σ is the standard deviation of the data and the IQR is the interquartile range. Since the standard deviation and the IQR of all the data points are calculated, Silverman's rule of thumb is especially suitable for single Gaussian distributions.

An alternative to the standard KDE algorithm was created for this study, in which only one bandwidth is computed using Silverman's rule of thumb, as outlined in Eq. (3.2). For the non-linear beam dynamics simulated in this study, the peak corresponding to the regular orbits in the distribution of a chaos indicator has a much greater amplitude than any other peak. This led to the hypothesis that Silverman's rule of thumb would be suitable for these chaos indicator histogram distributions. Since the KDE may not necessarily highlight a bimodal distribution, the threshold is placed at the minimum of the KDE between the peak with the highest amplitude and the peak after that.

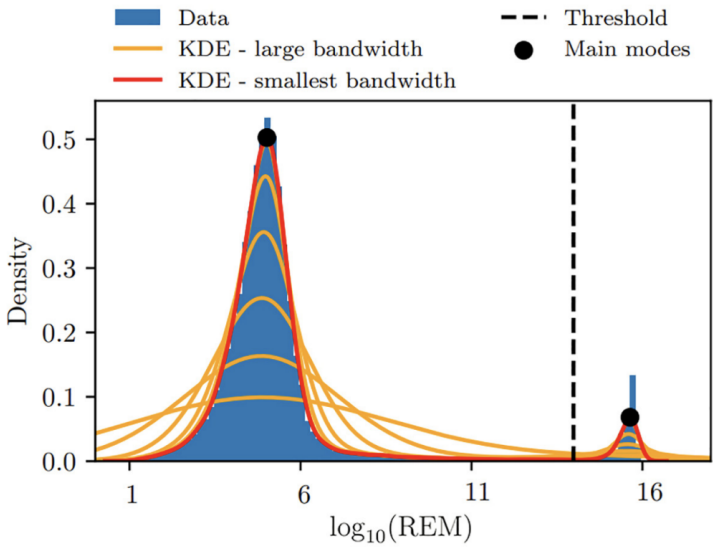


Figure 3.1: Schematic of the KDE-based algorithm used in the state-of-the-art analysis to determine the threshold for $\log_{10}(\mathrm{REM})$, taken from [12]. The algorithm finds the smallest bandwidth yielding two peaks in the histogram distribution and sets the threshold at the minimum of the KDE between these peaks. Note that $\log_{10}(\mathrm{REM})$ values are positive here due to a multiplication factor of 10^{16} used in the paper.

This adaptation resulted from the problems posed by the standard algorithm for chaos indicators such as the FLI, which has a multi-modal distribution. Keeping the bimodal assumption causes the algorithm to choose a very wide bandwidth. An example of this is shown in Fig. 3.2 for the histogram distribution of $\log_{10}(\mathrm{FLI}/n)$ at 10^8 turns. The placement of the threshold according to the KDE found with the standard algorithm produces less chaotic classifications, as shown in Fig. 3.3.

Classifying particles with a small positive Lyapunov exponent is difficult because their chaotic behaviour emerges slowly. For a long time, $\log_{10}(\mathrm{FLI}/n)$ values appear to converge to zero, but eventually they shift to a small positive value. The slow exponential growth of $e^{\lambda n}$ makes it hard to differentiate between zero and small positive Lyapunov exponents. After 10^8 turns, particles with a Lyapunov time shorter than 10^8 turns are easily identified with the FLI, but those with a longer Lyapunov time fall into a noisy region in the histogram.

3.2 Comparison of Linear Response Calculations in Single-Particle Tracking

There are two methods used to calculate the linear response: the tangent map method and the ghost particle method [9].

The linear response in Eq. (2.20) for computing the FLI requires evaluating the tangent map after each turn. For the modulated Hénon map, the tangent map can be derived analytically and used in the tangent map method, which was employed in state-of-the-art FLI analyses [12].

However, for realistic accelerator lattices with complex one-turn maps comprising thousands of element matrices, an analytical tangent map is not feasible. Therefore,

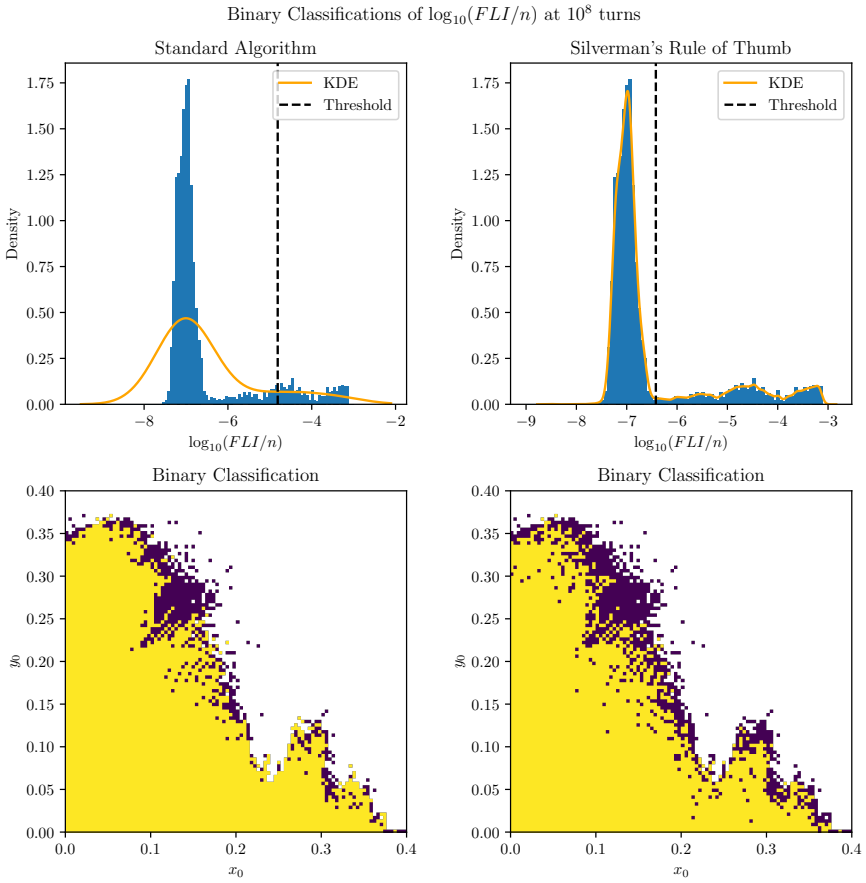


Figure 3.2: Comparison of KDE algorithms for classifying the histogram distribution of $\log_{10}(\mathrm{FLI}/n)$ values at 10^8 turns. The left column shows the binary classification, where yellow indicates a regular classification and purple chaotic, obtained using the standard algorithm, which fails to represent the features of the distribution. The right column presents the classification obtained using Silverman's Rule of Thumb to determine the bandwidth parameter, which produces a KDE that better captures the distribution features.

an approximate method, the ghost particle method [9], is necessary. Although this method is not needed for the modulated Hénon map, it is utilised in this study to evaluate noise and interference from numerical approximations, to prepare for noise in realistic accelerator lattice computations.

For each initial condition $\mathbf{x}_0$, the ghost particle method considers a displaced initial condition $\mathbf{y}_0 = \mathbf{x}_0 + \epsilon\xi$, with ϵ fixed at a small value. For the simulations in this study, ϵ is set to 10^{-8} based on the default settings offered by the code used in this study, chosen by the code author C. E. Montanari. This displaced initial condition serves as the "ghost particle". Both conditions are tracked using the one-turn map, and then the linear response from Eq. (2.20) is estimated by approximating the limit $\epsilon \to 0$ with the small value of ϵ as follows:

$$\Xi_n(\mathbf{x}) = \frac{\mathbf{y}_n - \mathbf{x}_n}{\epsilon}. \tag{3.3}$$

This estimate of the linear response can then be used to compute chaos indicators such as the FLI. When computing the FLI in single-particle tracking simulations,

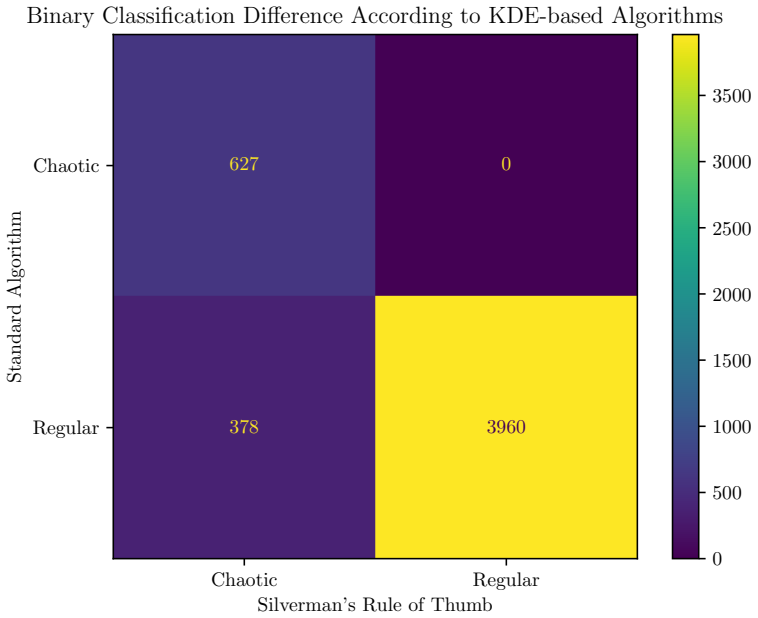


Figure 3.3: Confusion matrix of the difference in binary classifications obtained according to the KDE algorithms used to classify the histogram distribution of $\log_{10}(\mathrm{FLI}/n)$ values at 10^8 turns. In general, the standard algorithm results in less chaotic classifications than the algorithm using Silverman's rule of thumb.

for every initial condition, there is a ghost particle, so it is required to track twice the number of initial conditions compared to the tangent map method.

When approximating linear responses with numerical simulations, numerical errors due to rounding and truncation accumulate, especially when the distance between particles grows exponentially. This causes FLI values to represent the loss of memory due to numerical error pileup more than the value of the Maximal Lyapunov Exponent. To keep this as an accurate representation of the Lyapunov exponent, the distance $\|\mathbf{y}_n - \mathbf{x}_n\|$ is renormalised to ϵ every τ turns [9]. In this study, the value of τ was set at 100 as a compromise between accurate FLI values and computational cost.

The FLI was selected for this study because it provides a direct estimate of the Lyapunov exponent, effectively indicating particle chaoticity at 10^8 turns. Figure 3.4 displays the ground truth FLI data at 10^8 turns for parameters $\varepsilon = 32.0$ and $\mu = 0.5$. The comparison of FLI distributions is based on particles that survived up to 10^8 turns using both data generation methods. The top middle histogram shows that the ghost particle method results in a noisier FLI distribution. Despite this, both methods clearly distinguish between regular and chaotic conditions, with the largest peak indicating regular particles and higher FLI values representing particles with greater Lyapunov exponents.

The noise produced by the ghost particle method is illustrated in Fig. 3.5a, where the densities of each histogram are overlaid for comparison. The KDE in each distribution in the figure was found using Silverman's rule of thumb. The broader peak associated with regular particles results in an increased threshold position and a higher classification of particles as regular, although the broad regular peak

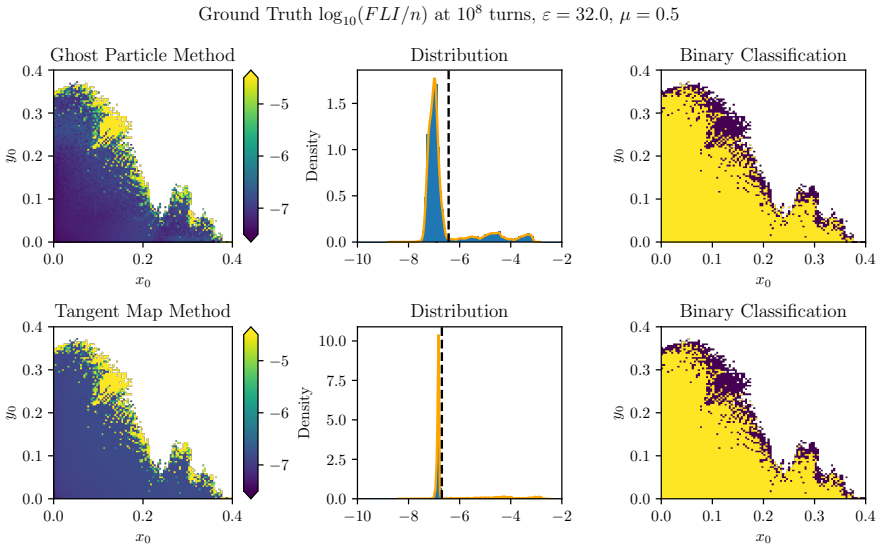


Figure 3.4: Comparison of histogram distributions and binary classifications for $\log_{10}(\text{FLI}/n)$ at 10^8 turns, using the ghost particle method (top row) and the tangent map method (bottom row). The ghost particle method introduces noise in the $\log_{10}(\text{FLI}/n)$ distribution, resulting in additional peaks and widening of the peak corresponding to regular orbits. Despite this, peaks distinguishing regular and chaotic initial conditions remain discernible.

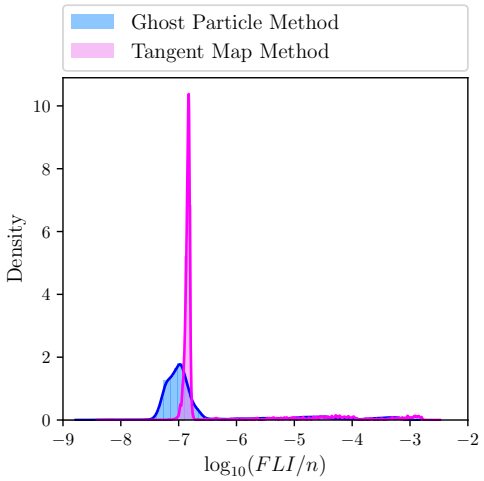
may include some particles with small Lyapunov exponents. In contrast, the FLI values generated with the tangent map method produce a narrow peak for regular particles, allowing the threshold to be placed closer to the peak. Consequently, these FLI values corresponding to small Lyapunov exponents deviate more from the peak, making it easier to identify these particles as chaotic.

Figure 3.5b shows the confusion matrix of the difference between the two ground truth FLI classifications. It confirms that the broader peak corresponding to regular particles in the ghost particle method does, in fact, lead to some particles being classified as regular in the ghost particle method but chaotic in the tangent map method.

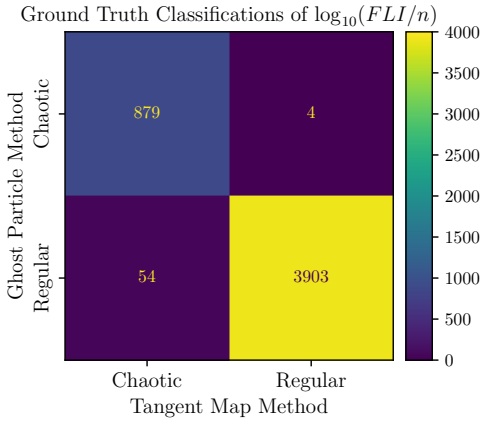
However, the percentage of particles that were classified differently according to the two methods is very small. Figure 3.5c shows the exact initial conditions that were classified differently. They are all located in the less stable region, so it makes sense that they may have a positive Lyapunov exponent.

Figure 3.5d shows a strong correlation between the two sets of FLI values after excluding outliers. Although noise is present, the minimal number of different classifications shown in Figs. 3.5b and 3.5c indicate that the ghost particle method remains viable for use in this study.

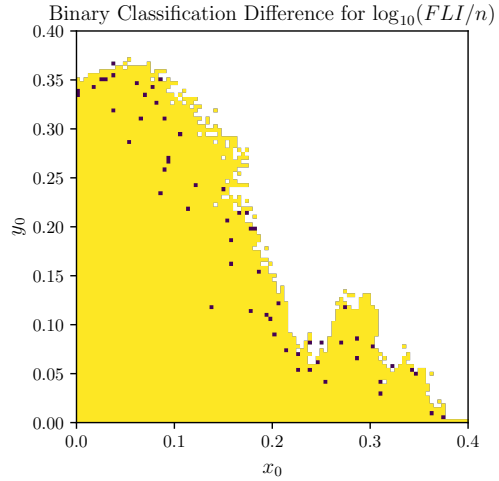
Figure 3.6 shows that the benchmark classification accuracy for FLI is similar between the two methods. Both methods exhibit the same trend, with accuracy significantly improving after 10^5 turns, when FLI values are sufficiently separated into two groups for clear classification of regular versus chaotic particles. With the standard algorithm used in state-of-the-art analysis, this improvement occurs only after 10^6 turns, which is relatively late compared to other chaos indicators [12], suggesting that FLI is not the best-performing chaos indicator. However, applying Silverman's



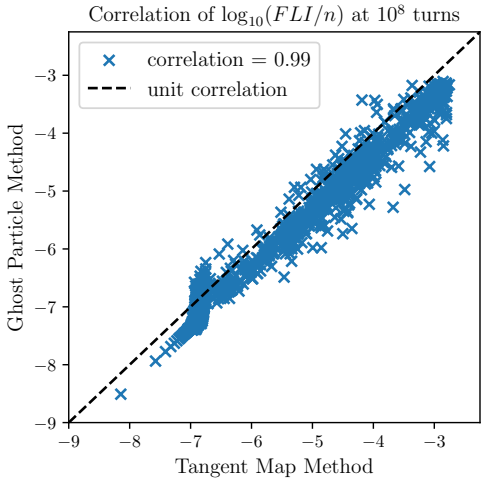
(a) Comparison of the histogram densities of $\log_{10}(\text{FLI}/n)$ at 10^8 turns for the ghost particle and tangent map methods. The KDE was found using Silverman's rule of thumb. The ghost particle method increases noise in the FLI distribution, mainly for the main peak corresponding to regular particles.



(b) Confusion matrix showing the numbers of initial conditions out of the total 4840 that were classified in different categories according to the ghost particle and tangent map methods.



(c) Display of the initial conditions for the FLI that were classified the same using the two methods (yellow), and differently (purple).



(d) Scatter plot illustrating the correlation between $\log_{10}(\text{FLI}/n)$ values at 10^8 turns using ghost particle and tangent map methods. A strong correlation (coefficient = 0.99) is observed, except near -7, where discrepancies occur at the peak of regular orbits. Outliers were excluded from the analysis.

Figure 3.5: Comparison of the $\log_{10}(\text{FLI}/n)$ histogram distributions and binary classifications obtained at 10^8 turns for the ghost particle and tangent map methods.

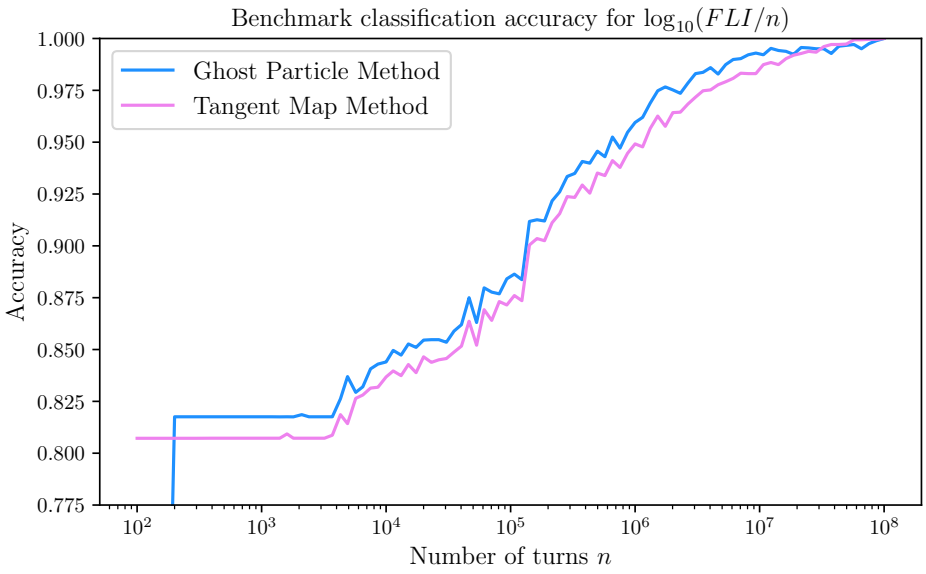


Figure 3.6: Benchmark classification accuracy established for the FLI indicator, generated with the ghost particle and tangent map methods. The two benchmark classification accuracies are comparable. Note that the classifications were performed using Silverman's rule of thumb.

rule of thumb enhances the FLI's performance. After 10^6 turns, the separation of FLI values is very clear, which supports the hypothesis that extrapolating its time evolution based on initial conditions would be effective.

3.3 REM Ground Truth and Time Evolution

Neither the tangent map nor the ghost particle method is required to compute REM, as the linear response is not required. The trend of exponential growth is captured by the numerical error pileup from tracking and backtracking the particles. REM values in this study were generated using Xsuite [17], following the method described in Section 2.2.3.

Figure 3.7 shows the time evolution of $\log_{10}$(REM). The trends of exponential and power-law increase separate after a relatively low number of turns, making REM one of the highest-performing chaos indicators. It was hypothesised that this property could lead to high-accuracy extrapolations of its time evolution based on data from a relatively low number of turns.

However, all REM values are capped at a small positive value corresponding to the diameter of the phase-space volume in which particles can exhibit stable motion. When evaluating chaos indicators, particles are considered lost if they exceed an arbitrarily set radius [12], therefore, a backtracked chaotic orbit with no memory of the initial condition can exhibit a distance at most equal to the diameter of the available area of the phase space.

As the number of turns increases, the power-law increase of regular particles eventually catches up with the capped exponential increase of chaotic particles, making it harder to distinguish between regularity and chaoticity at 10^8 turns and decreasing REM's classification reliability. Consequently, tracking beyond 10^8 turns

is not useful, as it fails to clearly distinguish between regular and chaotic particles. To address this, the ground truth binary classification for FLI is used to classify REM values in this study, following the state-of-the-art performance analysis approach [12].

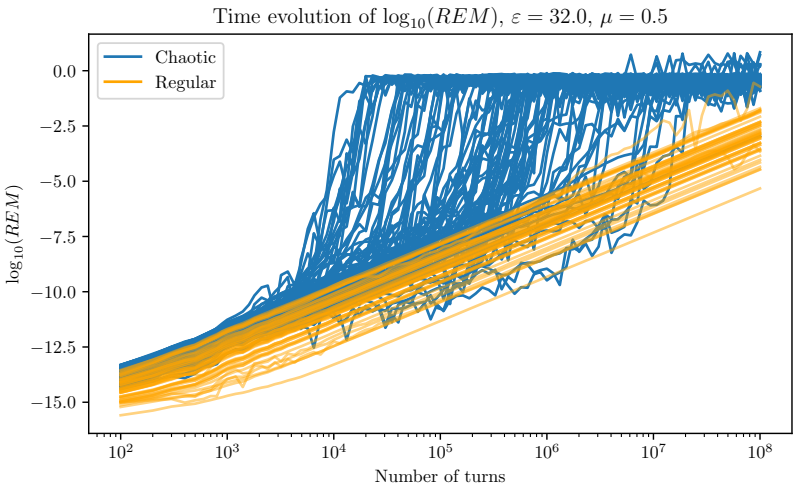


Figure 3.7: Time evolution of $\log_{10}$(REM) for a sample of chaotic and regular initial conditions, displaying the respective trends of exponential and power-law increase. The particles are coloured here according to the ground truth binary classification of the FLI (generated with the ghost particle method) at 10^8 turns.

Figure 3.8 shows a comparison of the FLI ground truth when considering the ghost particle method and the REM values at 10^8 turns. It should be noted that only particles that survived up to 10^8 turns in both FLI and REM tracking simulations were considered in this analysis.

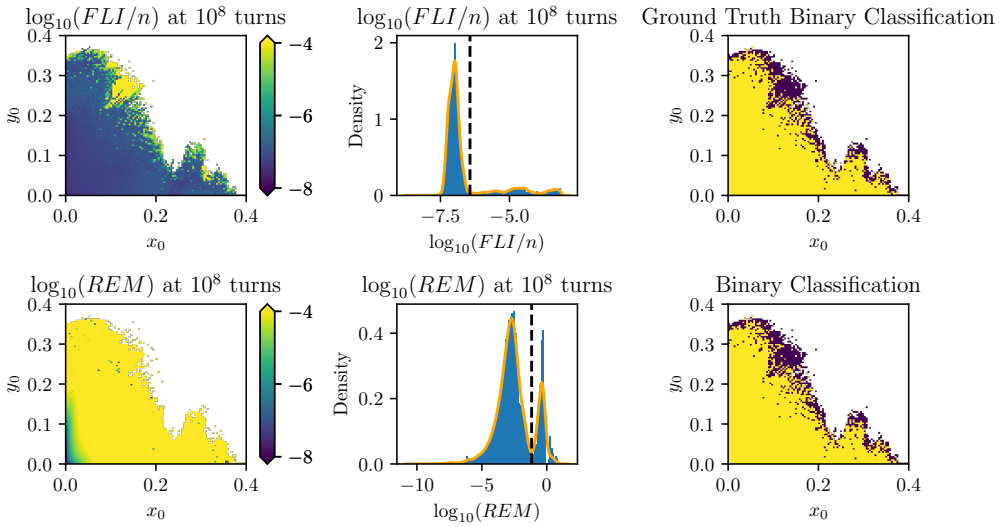


Figure 3.8: Comparison of the FLI ground truth obtained with the ghost particle method and REM values at 10^8 turns.

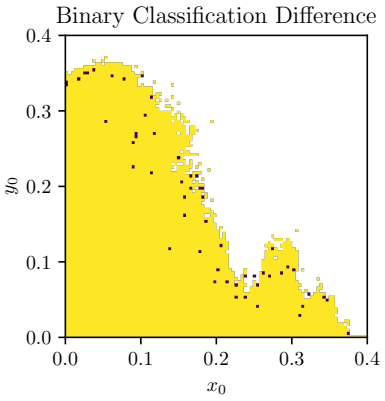
The histogram for REM values at 10^8 turns clearly illustrates the issue of the power-law increase of regular particles that catches up with saturated values of the

REM indicator for chaotic particles, since the two peaks are very close to each other. At higher numbers of turns, these peaks would merge, thereby reducing a reliable binary classification. The abrupt division of peaks in the histogram distribution at 0 is the result of fluctuations around the saturation point. The two secondary peaks correspond to the saturated particles. However, because the regular and chaotic peaks are so close, the large peak includes some of the fluctuations of saturated particles. If this histogram were used for ground truth classification, a considerable number of saturated particles would be misclassified as regular.

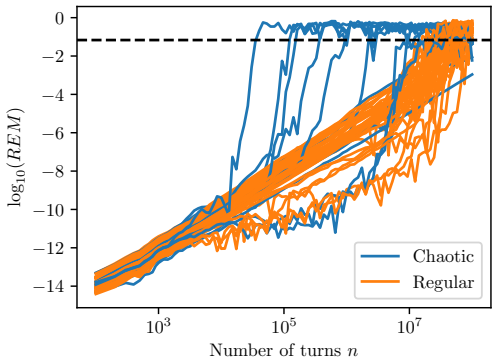
This is confirmed in Fig. 3.9, in which the exact initial conditions that are classified differently for the two indicators are shown. Figure 3.9a shows that all these initial conditions lie in the chaotic region in phase space, so it makes sense that there may be a region of uncertainty in the probing of the value of the Lyapunov exponent. Figure 3.9b shows the $\log_{10}(\text{REM})$ time evolution of these initial conditions classified differently, coloured according to the $\log_{10}(\text{FLI}/n)$ ground truth classification. The $\log_{10}(\text{REM})$ time evolution oscillates after saturation, leading to different classifications according to their final position relative to the threshold. The chaotic initial conditions shown in Fig. 3.9b happen to oscillate just below the $\log_{10}(\text{REM})$ threshold at 10^8 turns, which would have led to a regular classification. Using the FLI ground truth classification and only considering the particles that survive up to 10^8 turns in both the FLI and the REM tracking simulations therefore decreases the number of instances in which this occurs. The regular initial conditions shown saturate very late, implying that the orbits of these particles have small Lyapunov exponents, so it makes sense that they have been classified as regular. This could possibly have been circumvented by using the FLI generated with the tangent map method instead of the ghost particle method as ground truth, since there are more chaotic classifications when using the FLI generated with the tangent map method.

Figure 3.10 shows the reference classification accuracy established for REM. The classification accuracy for $\log_{10}(\text{REM})$ at 10^8 turns does not reach unity because the binary classifications of $\log_{10}(\text{FLI}/n)$ and $\log_{10}(\text{REM})$ at 10^8 turns are slightly different. Although the histograms for FLI and REM differ significantly at 10^8 turns, the FLI ground truth binary classification remains a viable target for REM classifications, because the difference in binary classifications is small.

These results indicate that the accuracy of the analysis carried out with the REM indicator could be greatly improved by performing a numerical check of the presence of saturation effects. In this way, one could verify whether the REM scales exponentially or as a power law vs the number of turns over the non-saturated part of the REM evolution.



(a) Display of the exact initial conditions that were classified the same (yellow) and differently (purple) according to the histogram distributions of $\log_{10}(\text{REM})$ and $\log_{10}(\text{FLI}/n)$ (generated with the ghost particle method) at 10^8 turns.



(b) Time evolution of $\log_{10}(\text{REM})$ for initial conditions with different classifications. Time series are coloured by $\log_{10}(\text{FLI}/n)$ (ghost particle method) ground truth at 10^8 turns. The opposite classification is observed according to the $\log_{10}(\text{REM})$ histogram distribution at 10^8 turns, with its threshold value shown by the black dashed line.

Figure 3.9: Overview of the initial conditions that were classified differently according to the histogram distributions of $\log_{10}(\text{REM})$ and $\log_{10}(\text{FLI}/n)$ (generated with the ghost particle method) at 10^8 turns.

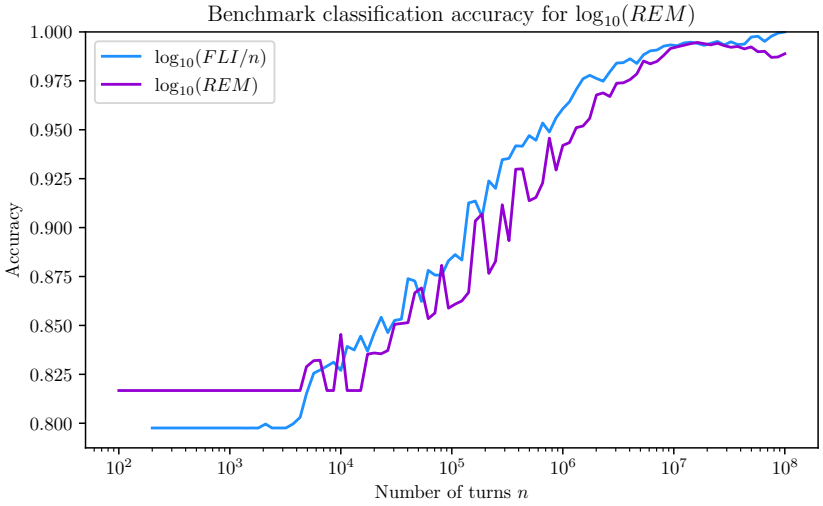


Figure 3.10: Benchmark classification accuracy for $\log_{10}(\text{REM})$ using Silverman's rule of thumb for threshold setting. Ground truth classification is based on $\log_{10}(\text{FLI}/n)$ (ghost particle method) at 10^8 turns. While benchmark accuracy is comparable for both indicators, $\log_{10}(\text{REM})$ does not achieve unit accuracy at 10^8 turns due to slight differences in binary classifications, reflecting distinct histogram distributions.

Chapter 4

Implementation of ARIMA models

In this chapter we investigate whether ARIMA models can improve the classification performance of chaos indicators from the state-of-the-art analysis [12]. We explore the process for FLI and REM, including data selection for ARIMA fitting, determining the optimal ARIMA order (p, d, q), extrapolating up to 10^8 turns, and comparing classification accuracy against the benchmark.

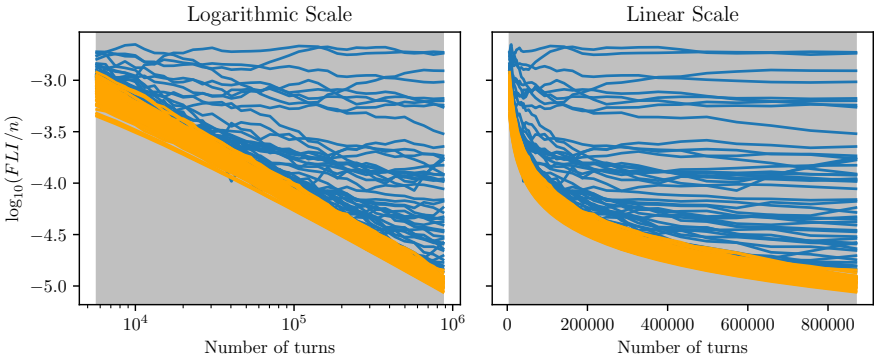
4.1 General Method of Fitting and Extrapolation

In this study, the time evolution of $\log_{10}(\text{FLI}/n)$ and $\log_{10}(\text{REM})$ is extrapolated for thousands of initial conditions. Manually inspecting each condition and determining the optimal ARIMA order is impractical, so a parameter scan was developed to find the best ARIMA order for a random sample of conditions classified as regular or chaotic. An equal number of both types was included to ensure the effectiveness of ARIMA models for different types of orbits.

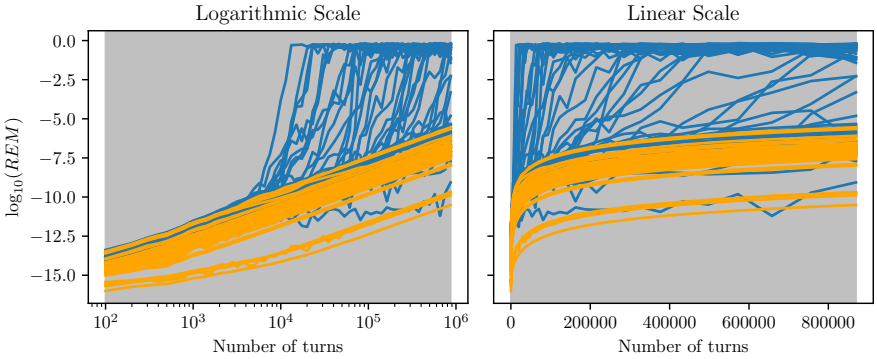
The parameter scan loops over possible combinations of orders (p, d, q). The values of p and q were chosen to be in the range of 0 to 5 and for d the chosen range was from 0 to 2. The parameter scan purposefully excludes the orders $(0, d, 0)$ because these are equivalent to a random walk model, in which the next data point equals the previous data point plus some white noise. This is not useful for extrapolating clear trends in various chaos indicators.

Logarithmic sampling up to 10^6 turns was selected for ARIMA model fitting to capture the general trend of the time evolution of chaos indicators, avoiding the emphasis on minor fluctuations. This choice of sampling up to 10^6 turns reflects the current number of turns reached in realistic LHC lattice simulations. As shown in Fig. 4.1, this approach clarifies the trend and reduces memory requirements by minimising data points while improving extrapolation accuracy.

For each potential ARIMA order (p, d, q) and each initial condition in the sample, an ARIMA model is fitted to the provided time evolution data. This model is then used to extrapolate the time series up to 10^8 turns. Given logarithmic sampling, the extrapolation from 10^6 to 10^8 turns yields a number of extrapolated samples comparable to the ones provided for the model fitting. The absolute error between the final extrapolated value and the actual final value of the time series is calculated for each initial condition. The total absolute error for all initial conditions in the sample is then summed, and the ARIMA order with the lowest total error is selected as optimal. Absolute error was chosen to ensure that the model is not penalised for



(a) $\log_{10}(\text{FLI}/n)$



(b) $\log_{10}(\text{REM})$

Figure 4.1: Example illustrating the impact of scales on trends when sampling $\log_{10}(\text{FLI}/n)$ and $\log_{10}(\text{REM})$ data. Logarithmic scaling clarifies trends for ARIMA modeling and reduces data points needed for fitting.

fluctuations, focusing instead on producing a final value close to the ground truth, which is essential for accurate classification of initial conditions as regular or chaotic.

Figure 4.2 shows an example of the absolute error calculated after a parameter scan, performed for the FLI (generated with the tangent map method). It shows that the order of the ARIMA model that produces the least absolute error after extrapolations to 10^8 turns is $(3, 1, 0)$, leading to its selection to extrapolate all initial conditions.

After the optimal ARIMA order (p, d, q) is selected, an ARIMA model with that selected order is fit to a given initial condition. The time series corresponding to each initial condition will have its own set of optimised parameters according to the ARIMA model. The time evolution of the chaos indicator for the initial condition is extrapolated from the fitting point, i.e. the number of turns up to which data is provided, up to 10^8 turns. The time evolution of the extrapolation is then saved for every initial condition that survived up to 10^8 turns according to the ground truth classification we have defined.

This process is repeated for varying numbers of turns, up to which data are provided to fit ARIMA models with the optimal order. This was done to compare whether providing more or less data would yield similar extrapolation results.

Note that some initial conditions may encounter convergence errors when fitting an ARIMA model of a given order. These errors, resulting from an inability to

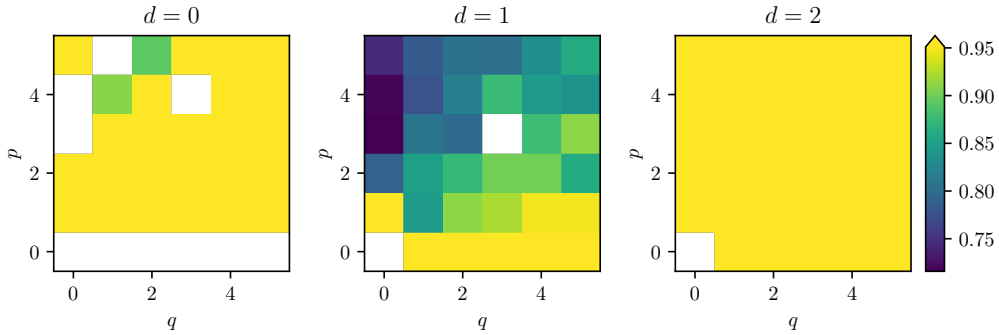


Figure 4.2: Example of $\log_{10}(\mathrm{FLI}/n)$ result (tangent map method) from parameter scan algorithm in this study. ARIMA models were fit to the time evolution of 100 initial conditions (50 regular, 50 chaotic) up to 10^6 turns. Models extrapolate to 10^8 turns; the absolute error at 10^8 turns is evaluated. The logarithm of the sum of errors for all initial conditions determines colour in heatmap squares. Optimal ARIMA order $(3, 1, 0)$ was chosen based on least absolute error. White squares denote excluded $(0, d, 0)$ random walk models and models resulting in fitting errors.

determine suitable parameters, make it impossible to fit the model for these initial conditions. In the case of REM, where this occurred for one initial condition, the condition was excluded from the binary classification comparison with the ground truth.

4.2 ARIMA Extrapolation of the FLI

4.2.1 FLI Generated with the Tangent Map Method

Figure 4.3 shows the sample FLI data provided for the parameter scan to determine the optimal ARIMA order for the extrapolations. For FLI, it was decided to start fitting ARIMA models at 5×10^3 turns to exclude initial high-amplitude fluctuations, which are not indicative of the power-law decrease trend. At low numbers of turns, the values of the FLI lack an informative value and do not differentiate between regular and chaotic particles. Including these fluctuating data would cause the ARIMA model to focus excessively on the initial fluctuations, rather than the more important power law decrease trend that follows.

The ARIMA order $(3, 1, 0)$, deduced from the data presented in Fig. 4.2, was used to extrapolate the $\log_{10}(\mathrm{FLI}/n)$ generated with the tangent map method. In the parameter scan, data up to 10^6 turns were used for fitting, providing a good basis for evaluating extrapolation performance. Figure 4.4a shows the extrapolation of $\log_{10}(\mathrm{FLI}/n)$ from 10^6 to 10^8 turns for a small sample of (10 particles each) regular and chaotic initial conditions. For regular conditions, the ARIMA model identifies and extrapolates the power-law decrease trend effectively, as seen in Fig. 4.4a. Overfitting is evident for chaotic particles that had saturated before 10^6 turns, with the model predicting horizontal lines, which is favourable as it reflects the Maximal Lyapunov Exponent at saturation.

Although the performance of the extrapolation appears promising in Fig. 4.4a, there are also some misclassified orbits, some extrapolations for which are shown in Fig. 4.4b. Some levelling off is visible for the ultrastable cases due to the initial

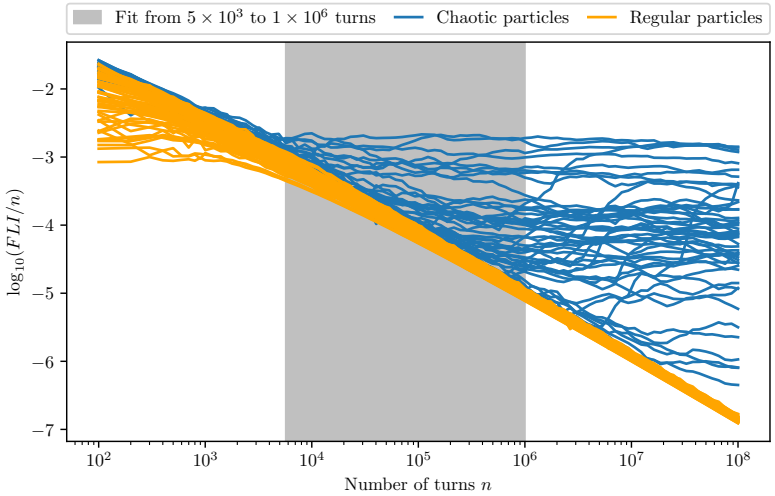


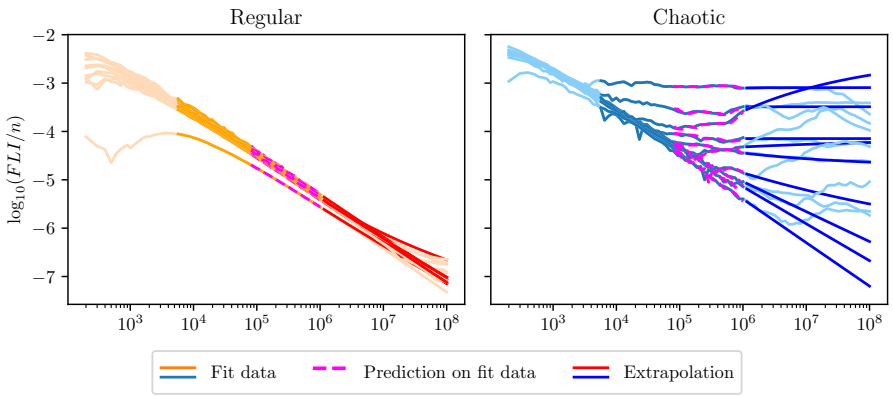
Figure 4.3: Time evolution of $\log_{10}(\mathrm{FLI}/n)$ for the sample of 100 initial conditions (50 regular and 50 chaotic) used in the ARIMA model fitting parameter scan. Data before 5×10^3 turns were intentionally omitted from ARIMA model fitting to prevent initial high-amplitude fluctuations from being misinterpreted as part of the trend.

increase of the time evolution before the decrease, which is incorporated into the ARIMA model and leads to a prediction resembling a sine curve. This likely occurred because the initial downward curves of the FLI values that appear before 10^4 turns were still incorporated into the ARIMA models. This could perhaps have been prevented by starting to fit the model at a later point in the time series than 5×10^3 turns to ensure that only the trend of power-law decrease is captured.

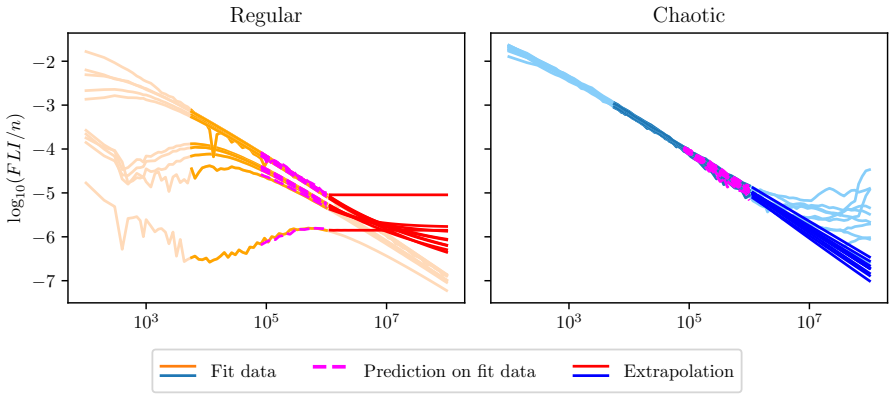
Overfitting occurs for the highly fluctuating cases shown in Fig. 4.4b. The ultrastable cases do not manifest any recognisable trend until after the fitting point, leading to the model including fluctuations and overfitting. For initial conditions that include high-amplitude fluctuations, even if the general trend still follows a power-law decrease, the prediction is a horizontal straight-line prediction with no trend at all. This shows that the fluctuations are at a sufficiently high amplitude to cause the ARIMA model to focus exclusively on the fluctuations and not the trend.

For the case of chaotic initial conditions, misclassifications stem from the inability of the ARIMA model to predict the sudden turning point that leads to saturation of $\log_{10}(\mathrm{FLI}/n)$ values, which is expected, given the suitability of ARIMA models for continuing identified trends. If the time evolution did not reach saturation during model fitting, it is expected that the ARIMA model would continue to predict a power-law decrease. This shows that the saturation for chaotic particles occurs unpredictably, since there was no feature in the trend gathered by the ARIMA model that suggested future saturation.

Figure 4.5 shows a comparison of the single-particle tracking data at 10^6 turns, the extrapolation to 10^8 turns, and the ground truth single-particle tracking data at 10^8 turns. The histogram distribution is similar to the ground truth distribution, but noisier. Despite the noise, the threshold is quite close to the narrow regular peak, which should lead to a high classification accuracy. However, the binary classification shows that there are many initial conditions in the stable region that are classified as chaotic after the extrapolation. This leads to the binary classification accuracy after extrapolation to 10^8 turns being less than the binary classification accuracy



(a)



(b)

Figure 4.4: Extrapolation of $\log_{10}(\text{FLI}/n)$ (tangent map method) from 10^6 to 10^8 turns using ARIMA $(3, 1, 0)$ models, fit from 5×10^3 to 1×10^6 turns. The plots show a sample of regular and chaotic initial conditions (mostly) correctly classified (a) and incorrectly classified (b) at 10^8 turns.

obtained after single-particle tracking up to 10^6 turns.

However, it is possible that extrapolating over a smaller number of turns may lead to higher classification accuracy, since the features emphasised by the ARIMA extrapolations have less time to develop. Fig. 4.6 shows the classification accuracy after extrapolating from each fitting point to various numbers of turns up to and including 10^8. The accuracy curves are coloured according to the maximum improvement observed after extrapolating from each fitting point up to 10^8 turns. For the case of fitting up to 10^6 turns, the classification accuracy remains approximately the same before decreasing, and it does not have a high relative improvement.

The case of providing data up to 10^5 turns is interesting because the extrapolation from 10^5 turns produces the maximum improvement of all the fitting points considered in this study. The maximum improvement is observed after extrapolating to $10^{5.64}$ turns, for which the histogram distribution and binary classification achieved are shown in Fig. 4.7.

Despite the similarity in the $\log_{10}(\mathrm{FLI}/n)$ histogram distributions before and after extrapolation, the improvement in classification accuracy is evident from the increased number of chaotic classifications. The various features of regular and chaotic time evolution were just enough emphasised to produce a histogram distribution that resulted in a binary classification with a higher accuracy. For some chaotic initial conditions, the ARIMA model extrapolated the time evolution as a power-law decrease or a slowly saturating exponential decrease. At $10^{5.64}$ turns, the power-law decrease has not advanced sufficiently to result in a regular classification of these initial conditions, while at 10^8 turns, this would certainly have resulted in a misclassification.

Figure 4.8 shows the resulting binary classification after extrapolating from 10^5 to 10^8 turns, which confirms that there are many misclassifications. Figure 4.9 presents a sample of misclassified extrapolations, showing that the same early exponential decrease of the time evolution occurs for the regular initial conditions. For chaotic initial conditions, it is visible that the ARIMA models are capable of grasping the trend after a small number of turns for $\log_{10}(\mathrm{FLI}/n)$ values that do not show much fluctuation. However, since most time series corresponding to chaotic initial conditions saturate after 10^5 turns, many chaotic initial conditions are misclassified as regular.

The classification accuracy obtained from single-particle tracking up to each fitting point and extrapolating up to 10^8 turns never exceeds the benchmark classification accuracy. Given that providing data up to 10^5 turns is insufficient for accurate extrapolation, providing less data would further degrade extrapolation performance. However, when more data are provided, the extrapolation accuracy does not increase, due to the various misclassifications shown. Figure 4.10 shows that even if data is provided up to a very high number of turns, overfitting still occurs for the regular particles, and the chaotic particles whose $\log_{10}(\mathrm{FLI}/n)$ time evolution saturate very late may still be misclassified as regular.

4.2.2 FLI Generated with the Ghost Particle Method

The extrapolation results for the ghost particle method closely mirror those for the tangent map method. Figure 4.11 displays the outcome of the parameter scan, where the optimal order chosen was $(3, 1, 0)$, consistent with the FLI generated using the

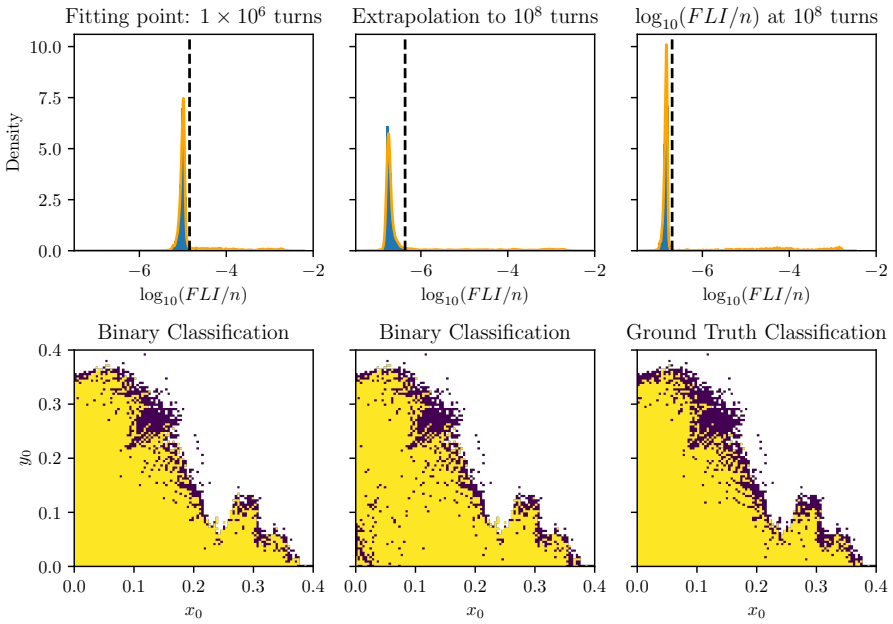


Figure 4.5: Progress of ARIMA extrapolation of $\log_{10}(\mathrm{FLI}/n)$ from 10^6 to 10^8 turns. Left column displays $\log_{10}(\mathrm{FLI}/n)$ value distribution and binary classification based on single-particle tracking up to 10^6 turns, used for ARIMA model fitting. Middle column shows extrapolated distribution and binary classification at 10^8 turns. Right column presents ground truth classification at 10^8 turns. Extrapolated binary classification at 10^8 reveals numerous misclassifications of regular particles as chaotic, particularly in stable regions.

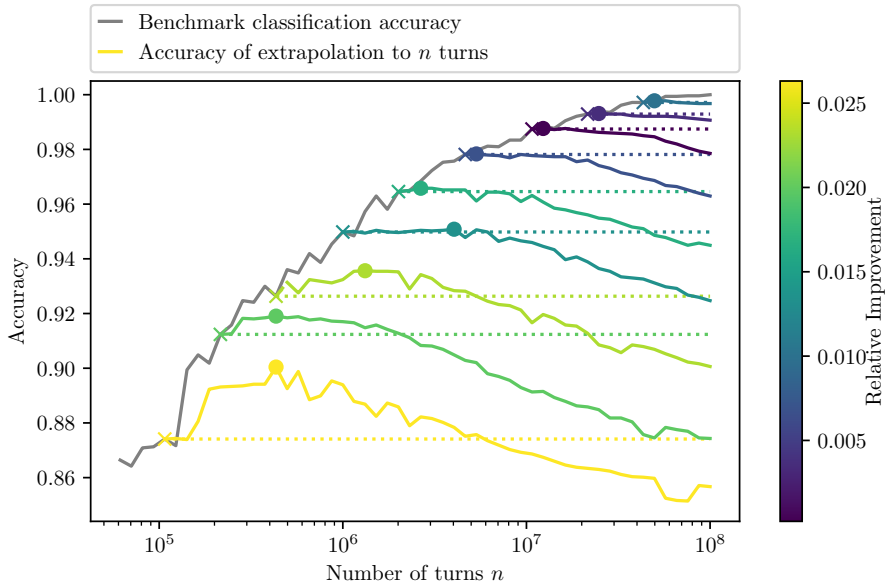


Figure 4.6: Overview of the accuracy of $\log_{10}(\mathrm{FLI}/n)$ (tangent map method) time evolution extrapolation using ARIMA order $(3, 1, 0)$ models from various fitting points (crosses) to up to 10^8 turns. Lines are coloured according to the maximum improvement observed, highlighted with a dot. Dotted lines show that all classifications after extrapolation to 10^8 turns are less accurate than single-particle tracking up to the fitting point.

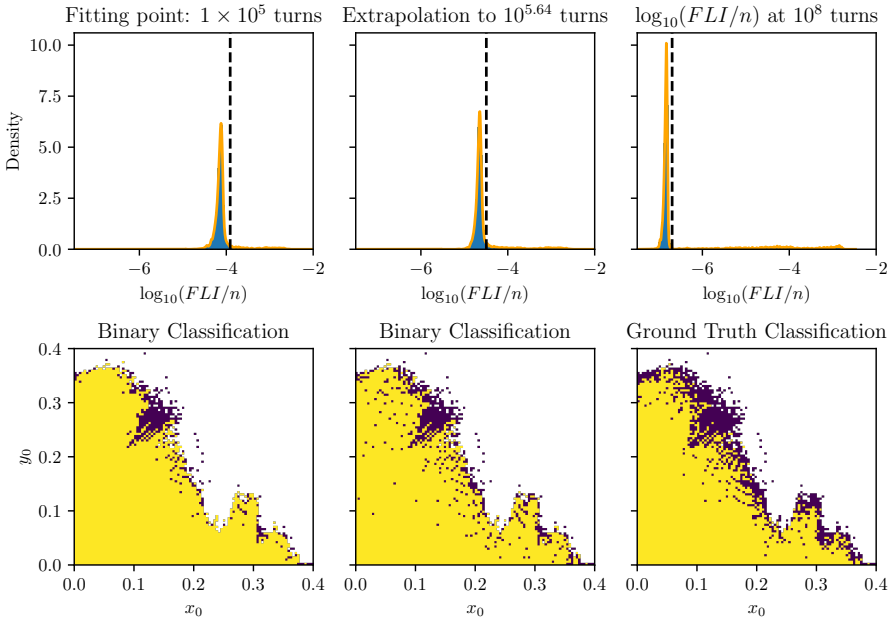


Figure 4.7: ARIMA extrapolation progress of $\log_{10}(\mathrm{FLI}/n)$ (tangent map method) from 10^5 to $10^{5.64}$ turns, which produces the maximum improvement in accuracy. Left column displays $\log_{10}(\mathrm{FLI}/n)$ value distribution and binary classification from single-particle tracking up to 10^5 turns. Middle column shows distribution and binary classification after extrapolation to $10^{5.64}$ turns. Right column presents ground truth distribution and classification at 10^8 turns. Extrapolated binary classification, over approximately 3×10^5 turns, improves accuracy notably due to increased chaotic classifications.

tangent map method. This similarity was anticipated due to the comparable time evolution of $\log_{10}(\mathrm{FLI}/n)$ between the two methods.

Figure 4.12 shows the accuracy obtained when extrapolating from the various fitting points up to 10^8 turns. As expected, the decrease in accuracy after extrapolating to 10^8 turns is comparable to the decrease seen in Fig. 4.6 for the FLI generated with the tangent map method. Figure 4.13 demonstrates that misclassifications occur for the ghost particle method for the same reasons as for the tangent map method: early saturation of regular particles and the inability of the ARIMA model to predict chaotic saturation.

Figure 4.14 compares the tangent map and ghost particle methods based on misclassification fractions of chaotic and regular initial conditions after ARIMA extrapolation to 10^8 turns. Extrapolating from 10^5 turns shows a slightly higher fraction of misclassified chaotic initial conditions and a lower fraction of misclassified regular initial conditions for the tangent map method compared to the ghost particle method. This suggests that the ghost particle method's FLI exhibits earlier saturation, leading to more regular initial conditions being misclassified as chaotic. Given the ghost particle method's slightly higher benchmark accuracy that sharply increases at 10^5 turns, fewer chaotic misclassifications imply earlier saturation in its FLI time evolution compared to the tangent map method, which was incorporated into the ARIMA models.

Both methods exhibit a higher fraction of chaotic misclassifications compared to regular ones, due to the abrupt nature of chaotic saturation. This indicates

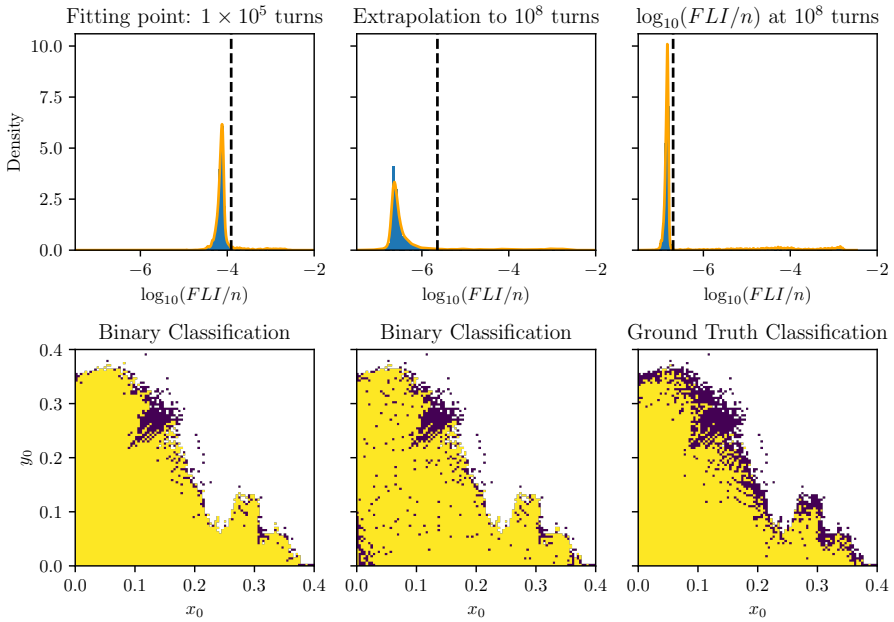


Figure 4.8: Progress of ARIMA extrapolation of $\log_{10}(\mathrm{FLI}/n)$ (tangent map method) from 10^5 to 10^8 turns, resulting in the poorest classification accuracy among all fitting points. Left column displays $\log_{10}(\mathrm{FLI}/n)$ value distribution and binary classification from single-particle tracking up to 10^5 turns, used for ARIMA model fitting. Middle column shows distribution and binary classification after extrapolation to 10^8 turns. Right column presents ground truth classification at 10^8 turns. Extrapolated binary classification at 10^8 reveals numerous misclassifications of regular particles as chaotic, particularly in stable regions.

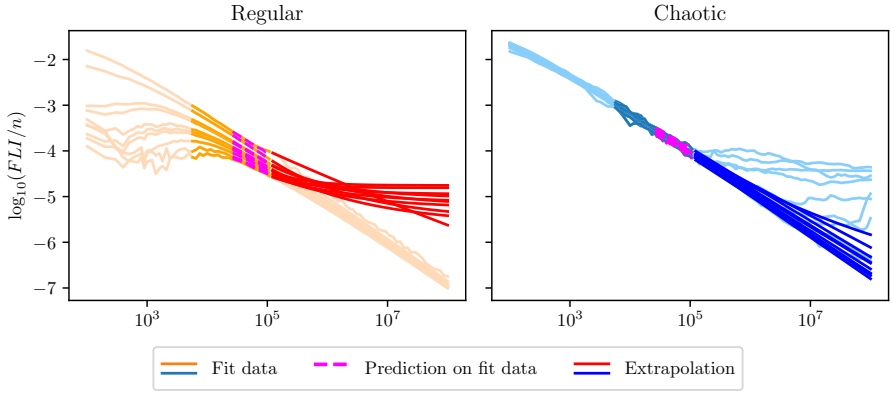


Figure 4.9: Sample of extrapolations of $\log_{10}(\mathrm{FLI}/n)$ from 10^5 turns to 10^8 turns leading to the misclassification of both regular and chaotic initial conditions at 10^8 turns. Based on the limited information provided up to 10^5 turns, the ARIMA models did not grasp the overall trends, leading to misclassifications.

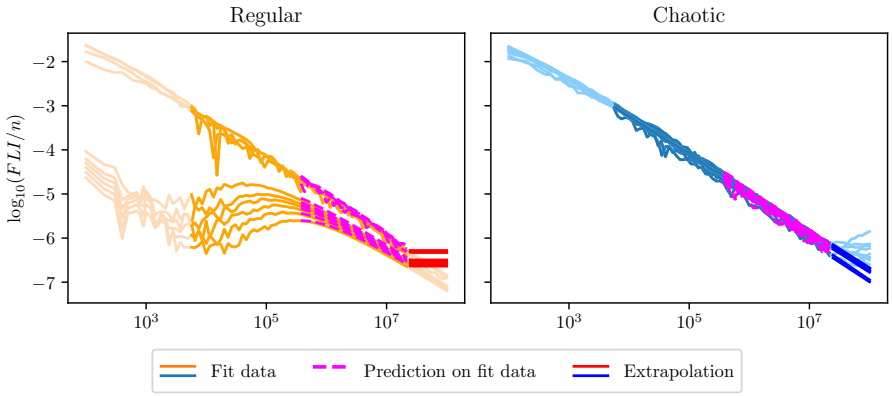


Figure 4.10: Sample of $\log_{10}(\mathrm{FLI}/n)$ extrapolations from 2×10^7 to 10^8 turns causing misclassification of regular and chaotic conditions. Regular particles suffer from overfitting due to data fluctuations, while chaotic particles face prediction failure of late saturation.

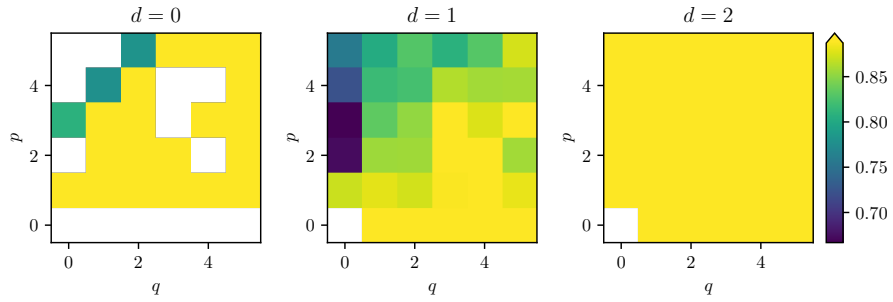


Figure 4.11: Colourmap for the parameter scan performed for $\log_{10}(\mathrm{FLI}/n)$ (ghost particle method). The selected optimal ARIMA order $(3, 1, 0)$ is the same as the one for the FLI generated with the tangent map method as shown in Fig. 4.2.

that ARIMA models are not suitable for predicting the saturation of chaotic time evolution. Generally, the misclassification fractions are comparable between the two methods, reflecting their similar time evolution trends.

Overall, there is no improvement upon the newly established benchmark accuracy using Silverman's rule of thumb with ARIMA extrapolations for the FLI, generated with the either the tangent map or the ghost particle method.

4.3 ARIMA extrapolation of REM

4.3.1 Adaptation of the Parameter Scan for REM data

Figure 4.15 depicts the time evolution of $\log_{10}(\mathrm{REM})$ for the sample of 100 initial conditions used in the ARIMA model fitting parameter scan. Model fitting for REM began from the initial data point (1×10^2 turns) due to minimal initial fluctuations, contrasting with FLI data. However, providing data up to 10^6 turns presents challenges for chaotic initial conditions whose $\log_{10}(\mathrm{REM})$ values saturate before this point. Including saturated REM values in the fitting process risks overfitting by incorporating both exponential growth and subsequent noise. Additionally, if chaotic

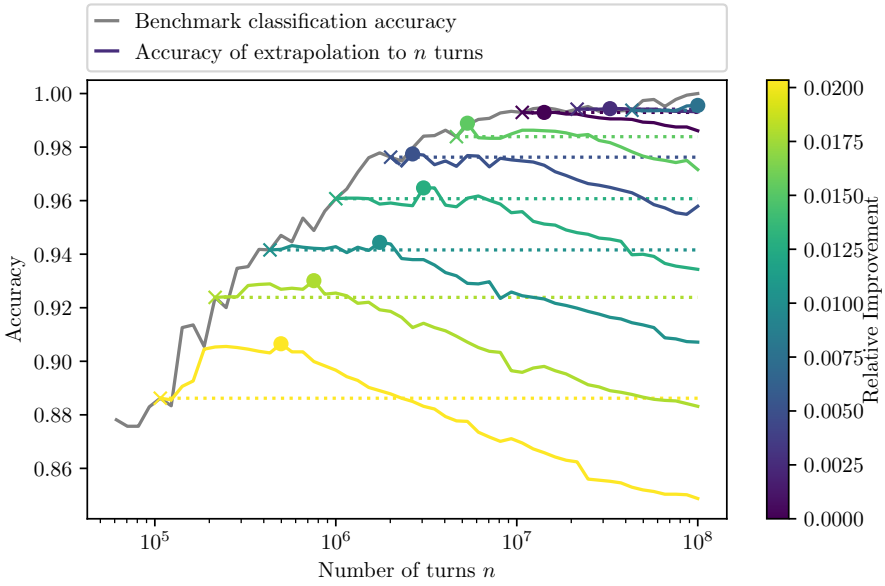


Figure 4.12: Overview of $\log_{10}(\mathrm{FLI}/n)$ (ghost particle method) time evolution extrapolation using ARIMA $(3, 1, 0)$ models from various fitting points to 10^8 turns. Coloured lines show maximum improvement, highlighted with a dot. Dotted lines indicate that extrapolation to 10^8 turns is less accurate than single-particle tracking up to the fitting point, comparable to the tangent map method in Fig. 4.6.

conditions saturate between 10^6 and 10^8 turns, extrapolation to 10^8 turns would fail to predict this saturation, resulting in high absolute error.

This led to an adjustment in the parameter scan where saturated data are excluded from fitting and absolute error calculation. Figure 4.16 illustrates these adaptations for data saturating before 10^8 turns. If saturation occurs at or before 10^6 turns, 80% of pre-saturation data is used for fitting, and extrapolation is conducted up to the saturation point. For data saturating after 10^6 turns, fitting proceeds normally up to 10^6 turns with extrapolation limited to the saturation point. The $\log_{10}(\mathrm{REM})$ evolution for regular particles, which does not saturate before 10^8 turns, is fit up to 10^6 turns, followed by extrapolation up to 10^8 turns following the standard FLI procedure. These adaptations aim to assess the ARIMA models' capability of extrapolating exponential growth trends. Figure 4.17 shows the result of this adapted parameter scan for the REM data. The optimal order selected was $(0, 2, 1)$.

4.3.2 Extrapolation results

The extrapolation curves for REM show unexpected levelling, evident even with data fitting up to 10^6 turns, as depicted in Fig. 4.19. This levelling occurs for both regular and chaotic initial conditions and is likely due to small-amplitude fluctuations in the REM data that are captured by the ARIMA model. Consequently, while ARIMA models yield optimal results based on the parameter scan, they do not fully capture the underlying trend. This effect causes regular particles to form a narrower peak around -2, while chaotic particles level off at higher values, primarily above zero. As a result, the histogram distribution exhibits a narrow peak

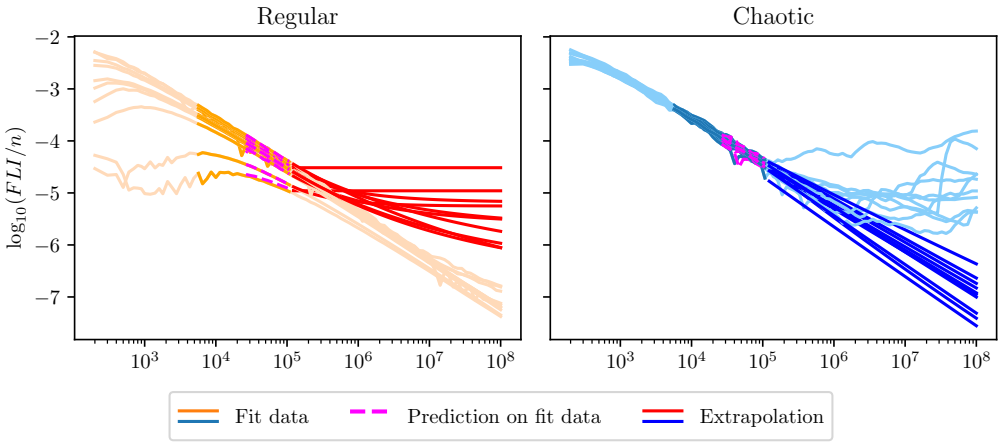
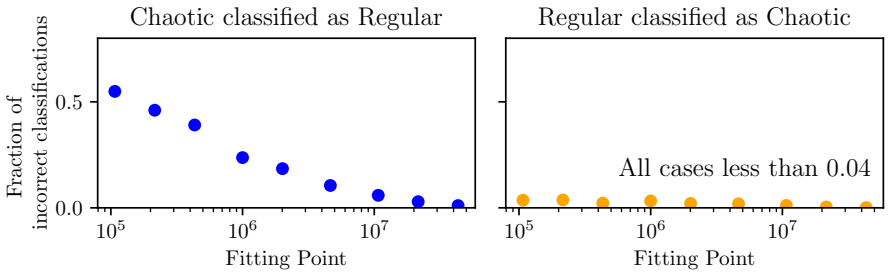
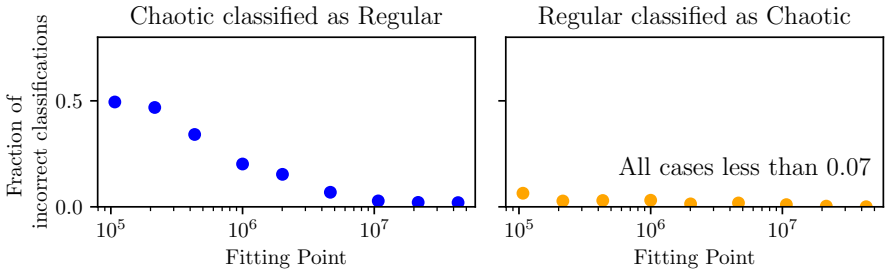


Figure 4.13: Sample of extrapolations of $\log_{10}(\text{FLI}/n)$ (generated with the ghost particle method) from 10^5 turns to 10^8 turns leading to the misclassification of both regular and chaotic initial conditions at 10^8 turns. Based on the limited information provided up to 10^5 turns, the ARIMA model continues various features such as fluctuations that are not part of the overall trend, leading to misclassifications. The same phenomenon is observed for the FLI generated with the tangent map method in Fig. 4.9.



(a) Tangent Map Method



(b) Ghost Particle Method

Figure 4.14: Fraction of incorrectly classified chaotic and regular initial conditions evaluated after ARIMA extrapolation to 10^8 turns from each fitting point, i.e. the number of turns up to which data are provided to fit the ARIMA models. The two figures provide a comparison between the tangent map (a) and the ghost particle methods (b). Though the result is comparable for the two methods, the tangent map method features slightly more chaotic misclassifications as regular, while the ghost particle method features more regular misclassifications as chaotic.

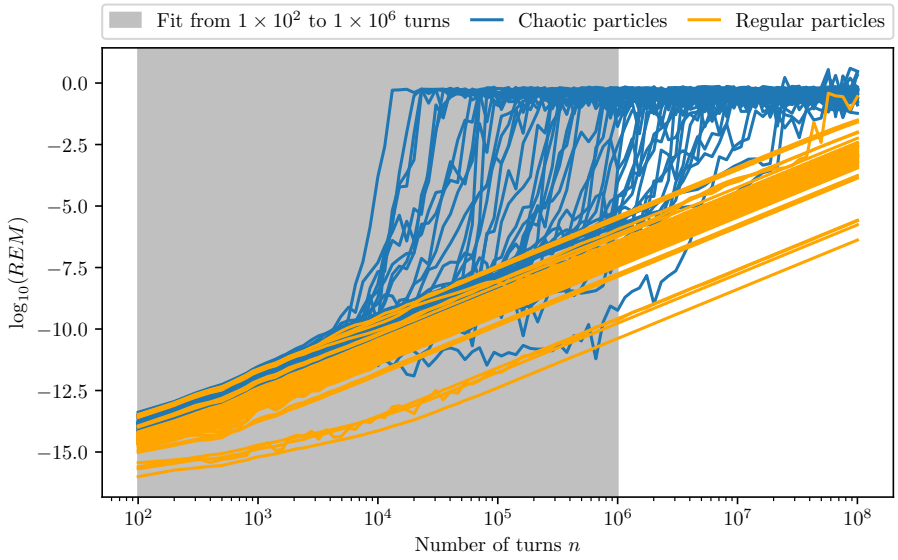


Figure 4.15: Time evolution of $\log_{10}$(REM) for the sample of 100 initial conditions (50 regular and 50 chaotic) provided for the ARIMA model fitting parameter scan.

for regular orbits and a widely spread region occupied by chaotic orbits, shown in Fig. 4.19. The distribution appears unimodal, making Silverman's rule of thumb suitable for estimation. However, the overall classification accuracy is impacted by several misclassifications.

Figure 4.18b showcases examples of extrapolations resulting in misclassifications of regular and chaotic initial conditions. Misclassification of regular particles happens when the ARIMA model predicts an exaggerated exponential increase, often due to the fitting point coinciding with an upward phase of a small fluctuation. The direction of extrapolation is sensitive to the gradient of the time series at the fitting point, reflecting the models prioritising fluctuations in $\log_{10}$(REM) values over the overall trend.

For chaotic particles, overfitting becomes apparent as the model predicts straight lines despite their highly fluctuating $\log_{10}$(REM) time evolution. The ARIMA models' sensitivity to fluctuations leads to numerous misclassifications when the fitting point coincides with a fluctuation, causing the extrapolation to continue in this direction before levelling off. This results in a final $\log_{10}$(REM) value below 0, leading to a classification as regular.

Figure 4.20 shows the binary classification accuracy obtained after extrapolating $\log_{10}$(REM) from the various fitting points to 10^8 turns. In general, the extrapolation accuracy never exceeds the benchmark classification accuracy, even if the extrapolation accuracy at 10^8 turns improves upon the single-particle tracking accuracy obtained at the fitting point. Providing data up to high numbers of turns leads to poor classification accuracy, even when extrapolating over a single data point. This suggests that fitting the chaotic $\log_{10}$(REM) time evolution only up to the saturation point and extrapolating from there results in frequent misclassifications.

The fitting point that demonstrates the highest relative improvement in accuracy is at 2×10^5 turns. Figure 4.21 compares histogram distributions and binary classifications after extrapolation from 2×10^5 turns (or various saturation points if data saturated earlier) up to 10^8 turns. While both classifications remain distant from

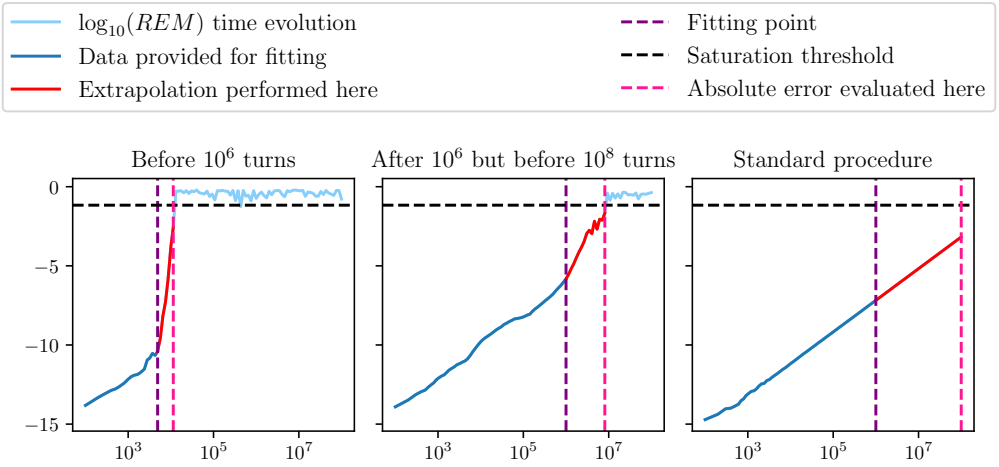


Figure 4.16: Schematic of fitting and extrapolation procedures for the parameter scan adaptation for REM. Left: Initial condition with $\log_{10}$(REM) evolution fit up to 80% before saturation, which occurs before 10^6 turns. Middle: $\log_{10}$(REM) evolution fit up to 10^6 turns, extrapolated to saturation before 10^8 turns. Right: $\log_{10}$(REM) evolution extrapolated from 10^6 to 10^8 turns without saturation before 10^8 turns.

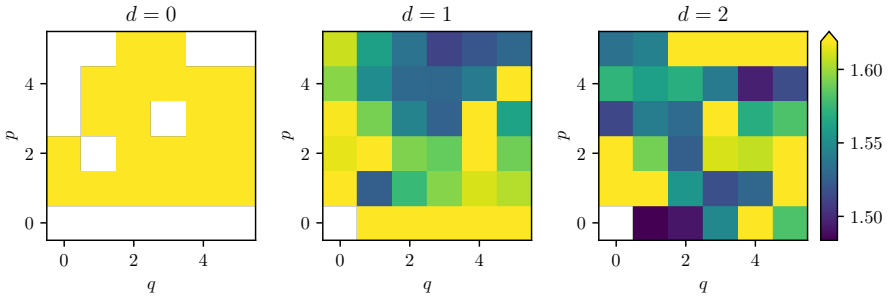
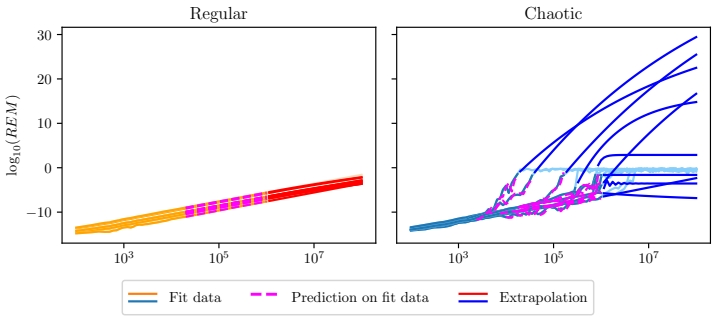


Figure 4.17: Result of the parameter scan for the REM sample, showing the optimal ARIMA order of $(0, 2, 1)$.

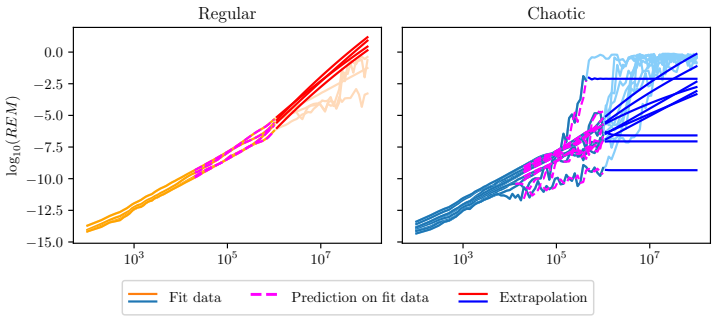
the ground truth, accuracy improves following extrapolation to 10^8 turns, primarily due to a slight increase in chaotic classifications.

Fig. 4.22 illustrates that regular particles are correctly classified due to the ARIMA model detecting a curvature in the trend, leading to a levelling off in their time evolution. This similarity in levelling locations for certain initial conditions contributes to the narrower peak observed in the histogram distribution after extrapolation to 10^8 turns seen in Fig. 4.21.

Figure 4.22a demonstrates that the time series of some chaotic initial conditions plateau above zero, resulting in a modest increase in the number of chaotic classifications compared to those at 2×10^5 turns. However, many chaotic initial conditions plateau *below* zero, leading to a regular classification. Interestingly, some chaotic time series level off even earlier than the regular ones. This indicates that the ARIMA model struggled to capture the exponential growth trend for most chaotic initial conditions, potentially due to similarities with the power-law increase seen in regular initial conditions or because the exponential growth did not start before 2×10^5 turns.



(a)



(b)

Figure 4.18: (a) Extrapolation of $\log_{10}$(REM) from 10^6 to 10^8 turns for a sample of regular and chaotic initial conditions. (b) Examples of $\log_{10}$(REM) extrapolation causing misclassification: chaotic conditions exhibit trendless extrapolations due to overfitting, while regular conditions show exaggerated exponential increases due to data fluctuations.

This suggests that the sudden exponential increase confuses the ARIMA models and leads to overfitting for chaotic particles, which is observed in Figure 4.23 when data is provided up to 2×10^6 turns. Some predictions even include fluctuations, indicating that the ARIMA models focus excessively on these fluctuations instead of the overall exponential increase trend. Some regular particles are misclassified as chaotic because they in fact saturate just before 10^8 turns, and the ARIMA model interprets their sudden exponential increase as a fluctuation, leading to incorrect chaotic classification. This either indicates a shortcoming in the ground truth classification of $\log_{10}(\text{FLI}/n)$ at 10^8 turns or that some regular $\log_{10}$(REM) time series reach saturation before 10^8 turns. Using the ground truth classification from the tangent map method instead of the ghost particle method might have mitigated this issue, as it yielded more chaotic classifications.

Figure 4.24 illustrates the fraction of incorrectly classified chaotic and regular initial conditions following ARIMA extrapolation to 10^8 turns from various fitting points. Initially, a higher fraction of chaotic initial conditions is misclassified as regular due to the ARIMA model's difficulty in predicting exponential growth. The high fraction persists across fitting points, due to overfitting that often results in horizontal predictions falling below the threshold for regular classification. Conversely, misclassifications of regular initial conditions as chaotic are minimal, one order of magnitude lower than observed for FLI in Fig. 4.14. This shows that the chosen ARIMA order $(0, 2, 1)$ was highly effective in predicting the final $\log_{10}$(REM)

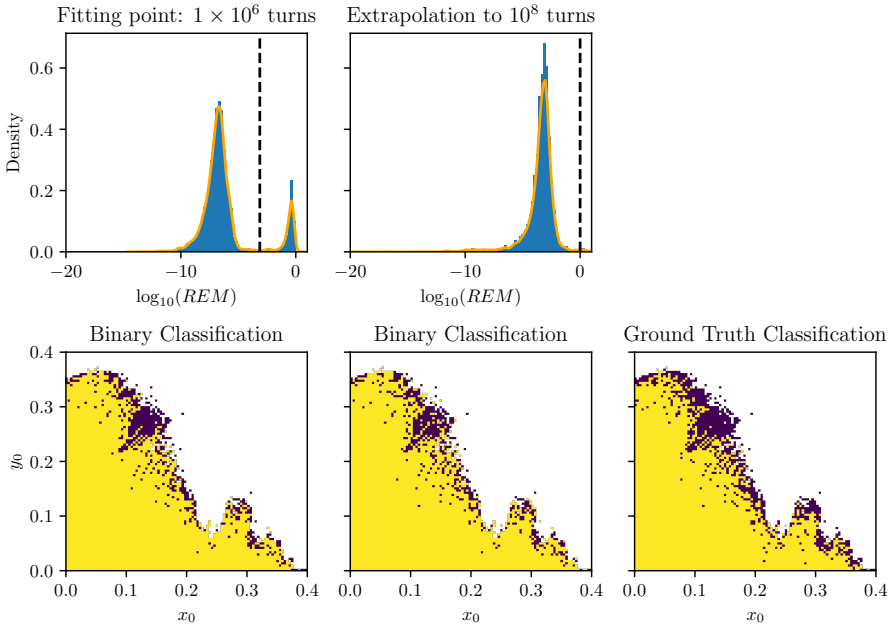


Figure 4.19: Progress of ARIMA extrapolation of $\log_{10}$(REM) from 10^6 to 10^8 turns. The left column shows the distribution and binary classification obtained with single-particle tracking up to 10^6 turns. The middle column shows the extrapolation results at 10^8 turns with fewer chaotic classifications. The right column shows the ground truth $\log_{10}(\mathrm{FLI}/n)$ binary classification used as reference.

value at 10^8 turns, despite the extrapolations not precisely following the power-law increase trend.

Given the extensive adaptations to the parameter scan and extrapolation required for REM, it is possible that a suboptimal ARIMA order was selected because of these modifications. The algorithm prioritises an accurate prediction of the final value over extrapolation, which means that the extrapolated trajectory may not align with the trends of the various time series, even if the final value at 10^8 turns is close to the true value. Many chaotic initial conditions exhibit $\log_{10}$(REM) time evolution with high-amplitude fluctuations. Incorporating these fluctuations into the ARIMA models resulted in overfitting for almost every chaotic initial condition. Consequently, the chosen ARIMA order does not produce a high improvement in classification accuracy after extrapolations, even though it was very effective at producing a similar final extrapolation value for regular initial conditions. The random sample of initial conditions chosen for the parameter scan included time series with fewer fluctuations. Consequently, time series with higher-amplitude fluctuations produced extrapolations that would have led to a high absolute error in the parameter scan. A different ARIMA order might have been selected in the parameter scan if the sample had included more highly fluctuating time series.

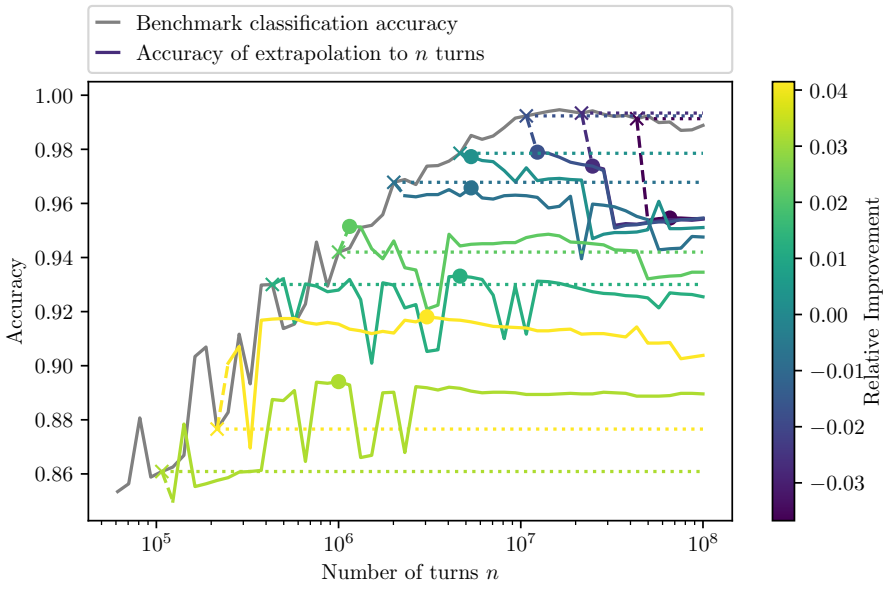


Figure 4.20: Overview of $\log_{10}(\mathrm{FLI}/n)$ (tangent map method) time evolution extrapolation accuracy using ARIMA $(3, 1, 0)$ models from various fitting points (crosses) to 10^8 turns. Lines are coloured by maximum improvement, marked with a dot. Dotted lines indicate that some classifications improve upon single-particle tracking up to the fitting point, but extrapolation accuracy never surpasses benchmark classification accuracy.

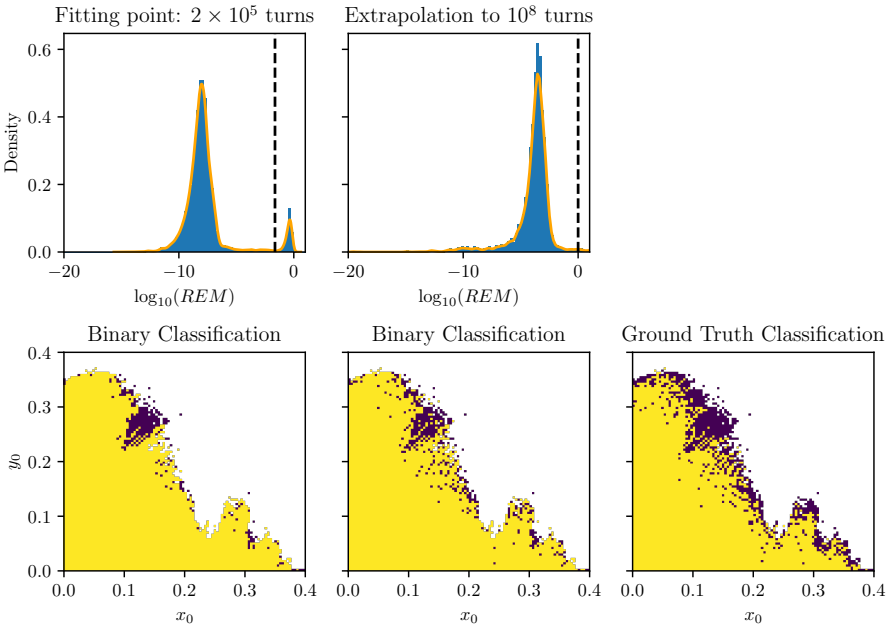
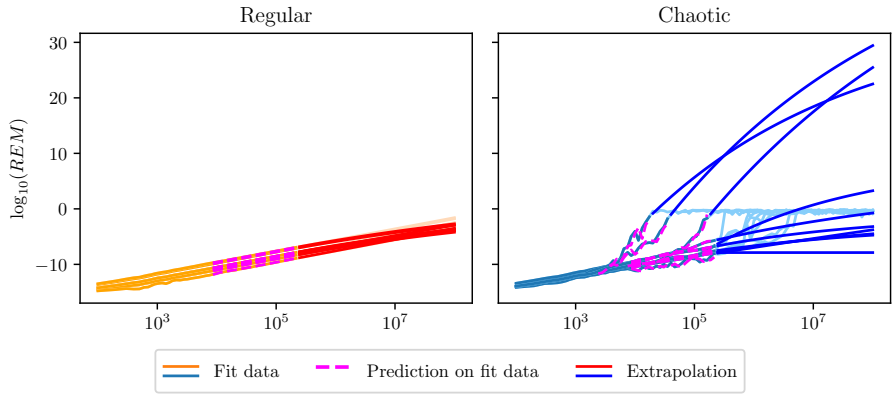
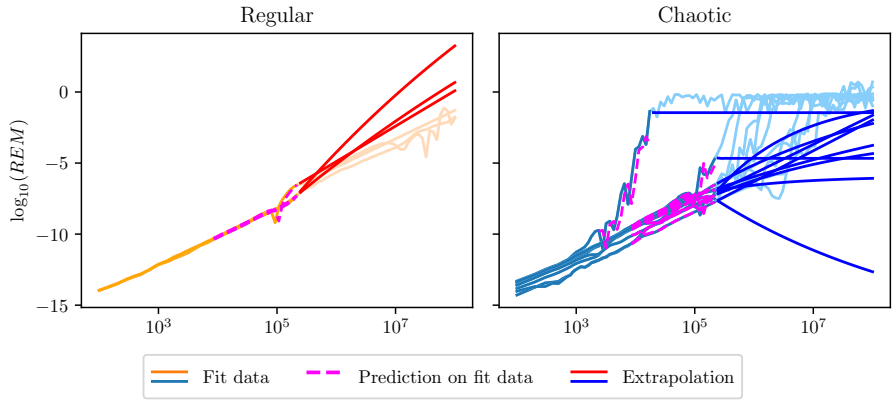


Figure 4.21: Progress of ARIMA extrapolation of $\log_{10}(\mathrm{REM})$ from 2×10^5 to 10^8 turns. The left column shows the $\log_{10}(\mathrm{REM})$ distribution and classification at 2×10^5 turns from single-particle tracking. The middle column presents the extrapolated $\log_{10}(\mathrm{REM})$ distribution and increased chaotic classifications at 10^8 turns. The right column displays the ground truth $\log_{10}(\mathrm{FLI}/n)$ classification (ghost particle method) for comparison.



(a)



(b)

Figure 4.22: Extrapolation of $\log_{10}$(REM) from 2×10^5 turns (or earlier saturation points) to 10^8 turns using ARIMA models of order $(0, 2, 1)$. The top plot (a) shows promising extrapolation performance, while the bottom plot (b) shows extrapolations leading to incorrect classifications.

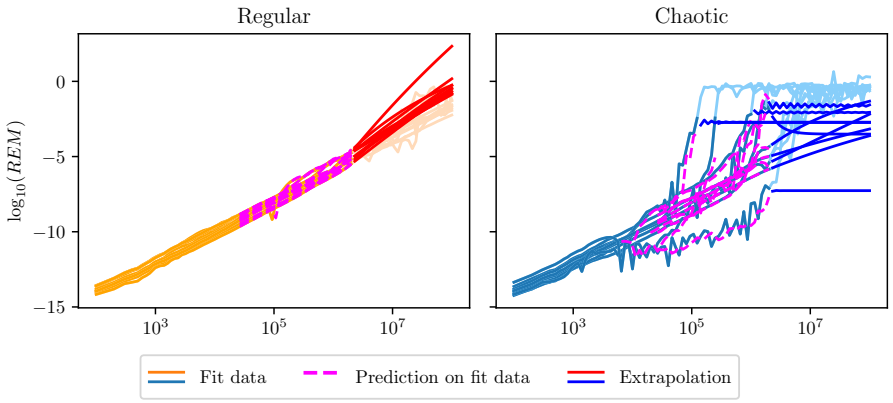


Figure 4.23: Sample of $\log_{10}$(REM) extrapolations from 2×10^6 to 10^8 turns (or earlier saturation points), causing misclassification of regular and chaotic initial conditions at 10^8 turns. Misclassifications occur due to extrapolation towards fluctuations for regular particles and overfitting to fluctuations in chaotic particles' data.

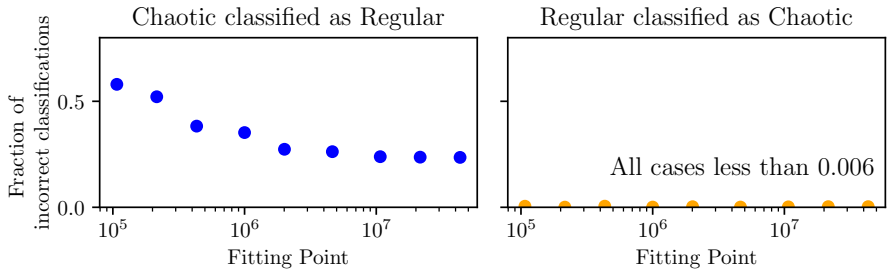


Figure 4.24: Fraction of misclassified chaotic and regular initial conditions after ARIMA extrapolation to 10^8 turns from each fitting point. Models often fail with chaotic data or overfit, but produce few regular misclassifications, proving effective in predicting $\log_{10}$(REM) values at 10^8 turns for regular conditions.

Chapter 5

Conclusions

This thesis investigates the potential of extrapolation techniques to enhance the classification performance of chaos indicators computed for Hamiltonian models of particle accelerators. The aim is to identify chaotic behaviour more rapidly, which contributes to improving the understanding of the relationship between chaotic regions in phase space and beam loss dynamics in circular accelerators.

Initially, two chaos indicators, the Fast Lyapunov Indicator (FLI) and the Reversibility Error Method (REM), are presented and discussed. Their distributions are compared at 10^8 turns, corresponding to the typical fill time of the CERN Large Hadron Collider. For a given indicator, a binary ground truth classification of regular or chaotic behaviour is established by placing a threshold in the distribution at 10^8 turns. The threshold was determined using a KDE-based algorithm designed to divide the histogram distribution into two categories that reflect the type of orbit. The accuracy for a given number of turns was calculated as the ratio of correctly classified orbits, based on the threshold at that specific number of turns, to the total number of particles that survived up to 10^8 turns. This accuracy provided a benchmark classification accuracy for each chaos indicator that could be improved through extrapolation techniques.

The standard classification algorithm used in the state-of-the-art analysis and an alternative to the algorithm created for this study, in which the KDE-based threshold is found using Silverman's rule of thumb, are presented and discussed. The adaptation using Silverman's rule of thumb proved to be more suitable for the histogram distributions of chaos indicators compared to the standard algorithm. The FLI was originally one of the lower-performing chaos indicators according to the state-of-the-art analysis, but the use of Silverman's rule of thumb increased the classification accuracy at low numbers of turns. This resulted in a benchmark classification accuracy comparable to REM, which is one of the highest-performing chaos indicators according to the state-of-the-art analysis.

In this study, the time evolution of $\log_{10}(\mathrm{FLI}/n)$ and $\log_{10}(\mathrm{REM})$ is extrapolated using ARIMA models, the orders for which were selected for each indicator by means of a parameter scanning algorithm.

For the FLI, the extrapolations from any number of turns result in a less accurate binary classification than the binary classification obtained with single-particle tracking up to that number of turns. There is some slight improvement in accuracy when extrapolating from lower numbers of turns, for example 10^5 turns, over a small number of turns within the same order of magnitude. This occurs because although

the ARIMA models are not provided with enough information to accurately capture and continue existing trends, the trends that are captured by the model, when forecasted over a small number of turns, deviate minimally from the actual data. However, when the extrapolation continues to 10^8 turns, there are many misclassifications due to the various cases of overfitting, which occurred due to the sensitivity of the ARIMA model to the initial fluctuations in the $\log_{10}(\mathrm{FLI}/n)$ time evolution, and the inability of the ARIMA model to predict sudden saturation in the time evolution for chaotic initial conditions.

Extrapolating $\log_{10}(\mathrm{REM})$ from 2×10^5 turns yielded the highest relative improvement in classification accuracy among all considered fitting points. Numerous adaptations to the fitting and extrapolation procedure were necessary due to the saturation of $\log_{10}(\mathrm{REM})$ values for chaotic initial conditions, potentially influencing the selection of a suboptimal ARIMA order. The extrapolations did not capture the true trends of the time series and the ARIMA models were sensitive to fluctuations in REM's time evolution, resulting in frequent misclassifications as extrapolations followed these fluctuations. Fitting data beyond 2×10^5 turns often led to overfitting for chaotic initial conditions, producing predictions devoid of trend and subsequent misclassifications. Consequently, ARIMA extrapolations did not significantly improve classification accuracy, failing to surpass benchmark levels. The problem of saturation in REM data could perhaps be circumvented by considering the REM time evolution on a linear scale.

In summary, the ARIMA models did not improve classification performance above the benchmark accuracy for the FLI or for REM. The FLI improved in benchmark classification accuracy with the use of Silverman's rule of thumb for binary classifications, but the extrapolation did not improve upon this newly established benchmark accuracy. If it is desired to save computation time from single-particle tracking, extrapolation from a lower number of turns, such as 10^5 turns, up to a similar order of magnitude may yield a small increase in classification accuracy. However, this approach is not recommended, as single-particle tracking up to these number of turns is already computationally feasible and produces higher benchmark accuracy than extrapolation. The data considerations and characterisations presented in this work can be extended to further studies on the time evolution of chaos indicators. It is suggested to explore machine learning-based approaches, which might more effectively capture the trends in chaos indicators.

Acknowledgements

This thesis would not have been possible without the extensive advice and support of my three supervisors. In particular, I would like to acknowledge Carlo Emilio Montanari for his extraordinary effort and dedication to supporting me and my work. He was always there to help, whether it was for advice on the next steps of my project or to fix technical issues. Thanks to his patience and encouragement and his dedication to explaining the same concepts repeatedly until I finally understood them, I was able to grasp the theoretical content of this thesis. I am deeply grateful for his relentless push for progress and for believing in me, which motivated me to achieve more than I thought possible in the short span of four months.

I am also very grateful to Massimo Giovannozzi for his meticulous attention to detail and for dissecting my results and analysis thoroughly despite his busy schedule. His guidance on how best to proceed with various aspects of the project was invaluable.

Additionally, I would like to thank Tatiana Pieloni for organising and setting up this project and for her insightful selection of supervisors. Her support, including organising a presentation that clarified the direction of the project for all of us, was essential for our progress.

I could not have had a luckier setup for this project with such supportive and dedicated supervisors. Their encouragement and expertise have been instrumental in the completion of this thesis, and I am extremely grateful for their help and perseverance. This brief acknowledgement cannot fully express my gratitude and how much they deserve recognition for their exceptional mentorship.

Bibliography

[1] O. S. Brüning, P. Collier, P. Lebrun, *et al.*, *LHC Design Report* (CERN Yellow Reports: Monographs). Geneva: CERN, 2004. DOI: 10.5170/CERN-2004-003-V-1.

[2] I. Béjar Alonso, O. Brüning, P. Fessia, L. Rossi, L. Tavian, and M. Zerlauth, *High-Luminosity Large Hadron Collider (HL–LHC): Technical design report* (CERN Yellow Reports: Monographs). Geneva: CERN, 2020. DOI: 10.23731/CYRM-2020-0010. [Online]. Available: https://cds.cern.ch/record/2749422.

[3] S. Redaelli, A. Bertarelli, R. Bruce, *et al.*, "Collimation of HL-LHC Beams," in *The High Luminosity Large Hadron Collider*. CERN, 2017, ch. Chapter 8, pp. 225–254. DOI: 10.1142/9789811278952_0008. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789811278952_0008. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/9789811278952_0008.

[4] A. Bazzani, G. Servizi, E. Todesco, and G. Turchetti, *A normal form approach to the theory of nonlinear betatronic motion* (CERN Yellow Reports: Monographs). Geneva: CERN, 1994. DOI: 10.5170/CERN-1994-002. [Online]. Available: https://cds.cern.ch/record/262179.

[5] M. Giovannozzi, W. Scandale, and E. Todesco, "Dynamic aperture extrapolation in the presence of tune modulation," *Phys. Rev. E*, vol. 57, pp. 3432–3443, 3 Mar. 1998. DOI: 10.1103/PhysRevE.57.3432. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.57.3432.

[6] A. Bazzani, M. Giovannozzi, E. H. Maclean, C. E. Montanari, F. F. Van der Veken, and W. Van Goethem, " Advances on the modeling of the time evolution of dynamic aperture of hadron circular accelerators," *Phys. Rev. Accel. Beams*, vol. 22, p. 104 003, 10 Oct. 2019. DOI: 10.1103/PhysRevAccelBeams.22.104003. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevAccelBeams.22.104003.

[7] C. Skokos, J. Laskar, and G. Gottwald, *Chaos Detection and Predictability*, 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, vol. 915, ISBN: 9783662484104. DOI: 10.1007/978-3-662-48410-4. [Online]. Available: https://library.biblioboard.com/viewer/05a8f7bd-c3cf-11ea-a5aa-0ae0aa0d175d.

[8] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis*, 5th ed. New York: Wiley, 2015, ISBN: 9781118675021. [Online]. Available: https://elib.maruzen.co.jp/elib/html/BookDetail/Id/3000065394.

[9] C. E. Montanari, *Diffusive models and chaos indicators for non-linear betatron motion in circular hadron accelerators*, Presented 2023, 2023. DOI: 10.48676/unibo/amsdottorato/10811. [Online]. Available: https://cds.cern.ch/record/2863942.

[10] C. E. Montanari, *Diffusive approach for non-linear beam dynamics in a circular accelerator*, Presented 25 Oct 2019, 2019. [Online]. Available: https://cds.cern.ch/record/2728138.

[11] M. Hénon, "Numerical study of quadratic area-preserving mappings," *Quarterly of Applied Mathematics*, vol. 27, no. 3, pp. 291–312, Jan. 1969. DOI: 10.1090/qam/253513. [Online]. Available: https://www.jstor.org/stable/43635985.

[12] A. Bazzani, M. Giovannozzi, C. E. Montanari, and G. Turchetti, "Performance analysis of indicators of chaos for nonlinear dynamical systems," *Physical Review E*, vol. 107, no. 6, -06-22 2023. DOI: 10.1103/physreve.107.064209.

[13] C. Froeschlé, R. Gonczi, and E. Lega, " The fast Lyapunov indicator: a simple tool to detect weak chaos. Application to the structure of the main asteroidal belt," *Planetary and Space Science*, vol. 45, no. 7, pp. 881–886, 1997, ID: 271836. DOI: 10.1016/S0032-0633(97)00058-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0032063397000585.

[14] E. Lega, M. Guzzo, and C. Froeschlé, "Theory and Applications of the Fast Lyapunov Indicator (FLI) Method," in *Chaos Detection and Predictability*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 35–54, ISBN: 978-3-662-48410-4. DOI: 10.1007/978-3-662-48410-4_2. [Online]. Available: https://doi.org/10.1007/978-3-662-48410-4_2.

[15] F. Panichi, K. Goździewski, and G. Turchetti, " The reversibility error method (REM): a new, dynamical fast indicator for planetary dynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 468, no. 1, pp. 469–491, Feb. 2017, ISSN: 1365-2966. DOI: 10.1093/mnras/stx374. [Online]. Available: http://dx.doi.org/10.1093/mnras/stx374.

[16] S. Weglarczyk, "Kernel density estimation and its application," *ITM Web Conf.*, vol. 23, p. 00 037, 2018. DOI: 10.1051/itmconf/20182300037. [Online]. Available: https://doi.org/10.1051/itmconf/20182300037.

[17] G. Iadarola *et al.*, "Xsuite: an integrated beam physics simulation framework," (Nashville, TN, USA, May 2024), presented at IPAC'24, Nashville, TN, USA, May 2024, paper WEPR56, unpublished, 2024.