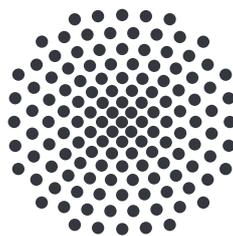


Application of neural networks to determine the transition state trajectory in periodically driven systems with two or more degrees of freedom

Master's thesis
by
Martin Tschöpe

September 21st, 2018

Examiner: Prof. Dr. Jörg Main
Co-examiner: Prof. Dr. Siegfried Dietrich



1. Institut für Theoretische Physik
Universität Stuttgart
Pfaffenwaldring 57, 70550 Stuttgart

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 5 |
| 1.1 | Motivation | 5 |
| 1.2 | Structure of the Thesis | 6 |
| 2 | Theory | 7 |
| 2.1 | Analytics and Definitions | 7 |
| 2.1.1 | Geometric Structures in Phase Space | 8 |
| 2.1.2 | Floquet Analysis of the NHIM | 11 |
| 2.1.3 | Neural Networks | 12 |
| 2.2 | Numerical Methods | 14 |
| 2.2.1 | Calculation of Discrete Points of the NHIM | 14 |
| 2.2.2 | Modified Numerical Integrator | 16 |
| 3 | Results | 19 |
| 3.1 | Phase Space Structures of the normally hyperbolic invariant manifold (NHIM) | 19 |
| 3.2 | Determination of the Transition State Trajectory | 24 |
| 3.2.1 | Centroid Search Algorithm | 24 |
| 3.2.2 | Friction Search Algorithm | 27 |
| 3.3 | Stability Eigenvalues and Rate Constants | 27 |
| 3.3.1 | Floquet Analysis of NHIM Trajectories | 29 |
| 3.3.2 | Comparison with Rate Constants of the Ensemble Method | 32 |
| 3.4 | Algorithmic Optimizations | 35 |
| 3.4.1 | Ensemble Rate Calculation | 35 |
| 3.4.2 | Hyper-Parameter Dependency | 39 |
| 3.4.3 | Optimizer Comparison | 44 |
| 3.4.4 | Regularisations | 45 |
| 4 | Conclusion and Outlook | 49 |
| 4.1 | Conclusion | 49 |
| 4.2 | Outlook | 50 |

Contents

| | |
|---------------------------------|-----------|
| Deutsche Zusammenfassung | 53 |
| Bibliography | 57 |
| Acknowledgements | 63 |

1 Introduction

The basic idea of this thesis is to find a way to observe trajectories in an unstable, invariant subspace of a *time-dependent* system at the paradigm of reaction dynamics. The procedure is visualized in Fig. 1.1. The crucial point here is the time-dependence of the system. On the one hand, many chemical reactions in nature are influenced by external fields. On the other hand, chemical reactions in technical applications would benefit if the reaction rate could be changed by external fields.

In the first step, a model system supplies a potential surface with a rank-1-saddle from which the unstable, invariant subspace, called *NHIM* [1–8] can be determined. The latter is calculated on discrete points and is approximated continuously with a machine learning method called *neural networks (NNs)*. Alternated application of a numerical integrator and a projection via the NN prevents the trajectories from leaving the NHIM due to an unstable degree of freedom. In this thesis, two possible applications of this stabilization are presented. Both applications are visualized in Fig. 1.1. On the one hand, the phase space structure of the NHIM itself can be studied. On the other hand, when combined with *Floquet analysis*, it enables to calculate a reaction rate constant.

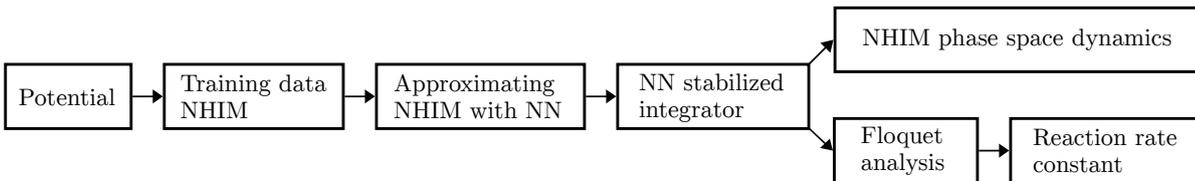


Figure 1.1: Visualisation of all interim stages for reaching the two primary goals of this thesis.

1.1 Motivation

Gathering knowledge about the phase space structure of the NHIM promises an understanding of the stability of this nonlinear system. In detail, it should be possible to

determine which *potential parameters* decide about a regular or chaotic behaviour of the system. This could allow to apply the already existing framework of nonlinear dynamics to gain more insight into this particular system.

The determination of reaction rates has a very wide range of applications from economically driven research (e.g. in chemical industry) to basic research (e.g. in astrophysics) [4, 9–21]. For instance, the time-dependency in the latter example emerges from varying electric fields which are virtually omnipresent in space. However most of the existing reaction rate calculation methods only allow to investigate static systems. The remaining techniques require the computational-time intensive propagation of large ensembles or are limited to systems with one degree of freedom. In contrast, the method introduced in this thesis has none of these fundamental limitations.

1.2 Structure of the Thesis

The modular structure of this thesis allows to address three types of readers. Those who are already familiar with machine learning and are primarily interested in the results of this may skip the mathematical part of NNs and focus on the general design in sections 2.2.2 and the detailed analysis in the sections 3.4.2 to 3.4.4.

The second reader group is assumed to have knowledge of reaction dynamics as well as reaction rates and seeks to enhance the same. For those it is advisable to look at the definitions of the NHIM and the transition state trajectory (TSt) which are used in this thesis (section 2.1.1) and possibly the very short section about Floquet analysis. In the *Results* chapter 3, it should be focused on section 3.3 about the application of Floquet analysis and especially subsection 3.3.2 that compares the rate constants obtained by Floquet analysis to the one's of the ensemble method. For more technical details sections 3.2 (about search algorithms for the TSt) and 3.4.1 (with improvements for the ensemble method algorithm used in this thesis) are recommended.

For the last group, that is exclusively interested in the phase space structures, the focus should be on sections 2.1.1 about the theory and 3.1 with the general results. The comparison of TSts for different systems is shown in section 3.3.1.

2 Theory

The theoretical chapter of this thesis is split in two parts: the first one covers the analytical basics like the definition of the *NHIM*, an introduction to the *Floquet analysis* and the basic idea behind *Neural Networks*. The second part focuses on more technical topics like specific hyper-parameters of the NN, the method that generates training data, and the modification for numerical integrators introduced in the thesis.

2.1 Analytics and Definitions

The first part of the following section introduces the NHIM and describes its properties. Furthermore, the transition state trajectory (TSt) is defined to prevent confusion with otherwise defined objects with the same name. The second part of the section shows a *Floquet analysis* for systems with more than one degree of freedom (DoF). This is important for the rate calculation. The chapter ends with a brief introduction to neural networks.

One of the two goals in this thesis is to calculate a *reaction rate*. A common way to introduce reaction rates (i. e. Ref. [22]) is by considering the number of reactants $N_r(t)$ at the time t , of a unimolecular reaction. The number of products at the same point in time t follow as

$$N_p(t) = N_r(0) - N_r(t). \quad (2.1)$$

For the rate law, this results in

$$\frac{\partial N_p(t)}{\partial t} = -\frac{\partial N_r(t)}{\partial t}. \quad (2.2)$$

As described in Ref. [23] a rate law of this form for a unimolecular reaction leads to a single rate constant k . The corresponding rate law has the form

$$\frac{\partial N_p(t)}{\partial t} = kN_r(t). \quad (2.3)$$

In combination with Eq. 2.2 the number of reactants N_r follows an exponential function of the form

$$N_r(t) - N_r(\infty) = N_{r,0} \exp(-kt). \quad (2.4)$$

Here, $N_{r,0}$ describes the number of initial reactants that are going to react. Positive rate constants k correspond to an exponential decay of the reactants in the system. In the following, a distinction is made between reaction rates obtained by Floquet analysis k_f and reaction rates obtained by ensemble propagation k_e .

2.1.1 Geometric Structures in Phase Space

In contrast to many chemical works, this thesis uses phase space coordinates instead of configuration space coordinates. This is for the following reason:

On the one hand there are systems, which allow to define a recrossing-free dividing surface (DS) [24–33] in phase space but not in configuration space [34]. This is true especially for *time-dependent* systems as discussed in this thesis. The reason why the phase space is more general than the configuration space is the higher dimensionality. This is the reason why on the other hand, every representation of a recrossing-free DS in configuration space can be extended to the phase space.

For a system with one saddle, it is possible to assign every point in the phase space according to the behaviour of particles at this position in one of the following five classes:

1. reactants who react forwards to products
2. products who stay products
3. products who react backwards to reactants
4. reactants who stay reactants
5. particles, that are neither reactants nor products (NHIM, null set in the x - v_x plane of the phase space)

This is shown in Fig. 2.1 for a Poincaré surface of section in one coordinate and its corresponding velocity. The intersection between the first and the second area as well as the intersection between the third and the fourth section form two unstable manifolds \mathcal{W}_u . Stable manifolds \mathcal{W}_s correspond to the intersections between the second and third area as well as the intersection between the first and fourth. It has to be mentioned

that the interval boundaries of the manifolds are open at their quasi-intersection. This means that \mathcal{W}_u and \mathcal{W}_s actually do not cross. Neither the two stable nor the two unstable manifolds are connected, but have an excluded point at the intersection between the manifolds. In a one-DoF system this point is called normally hyperbolic invariant manifold (NHIM) [1–8]. For a higher-dimensional system — with n DoFs — the NHIM is the unification of all intersections between \mathcal{W}_s and \mathcal{W}_u . Therefore the NHIM is $(2n - 2)$ -dimensional. Furthermore, it can be seen that the manifolds $\mathcal{W}_{s,u}$ are included in the five aforementioned classes of particles. The two manifolds on the left hand side of the NHIM correspond to the fourth class, because the represented particles will never react. Analogously, the second class includes the two manifolds on the right hand side of the NHIM.

The abbreviation NHIM comes about as follows [35]: The word *manifold* describes that this topological space is not necessarily an Euclidean space, i. e. it can be curved. *Hyperbolic* means that there is at least one unstable DoF. This thesis considers only systems with a rank-1-saddle, which ensures that there is at least one unstable degree of freedom. Since the NHIM is *normally* hyperbolic there is one unstable DoF that dominates all unstable motion. Last but not least the NHIM is *invariant*, which means that any particle on the NHIM will never leave this subspace of the full phase space.

In *time-independent* systems the Lyapunov exponents normal to the NHIM are positive or negative [34]. In contrast to that the Lyapunov exponents tangential to the NHIM are close to zero [34]. The latter is not necessarily true for *time-dependent* systems as studied in this thesis. Positive eigenvalues correspond to trajectories that approach from the NHIM exponentially over time. These trajectories form the stable manifold \mathcal{W}_s . The unstable manifold consist of trajectories, which leave the NHIM exponentially fast and have negative Lyapunov exponents.

A time-dependent, recrossing-free DS attached to the NHIM divides reactants from products. This will be used for the comparison with previous calculations of the reaction rate of our system in section 3.3.2. Furthermore, the NHIM is used to define the TSt. In a periodically driven system — as studied in this thesis — the TSt is that very trajectory on the NHIM with a periodicity equal to the system periodicity and the lowest energy. The TSt promises for the calculation of a reaction rate by an alternative method to the often used ensemble propagation.

A common way to calculate rate constants is by using statistical reaction theory. This is done for example in the *transition state theory (TST)* [36–53] where the particle flux through a DS is used to determine a reaction rate. A problem is that the basic assumption of statistical distributed ensembles is not fulfilled in some systems. For example, in

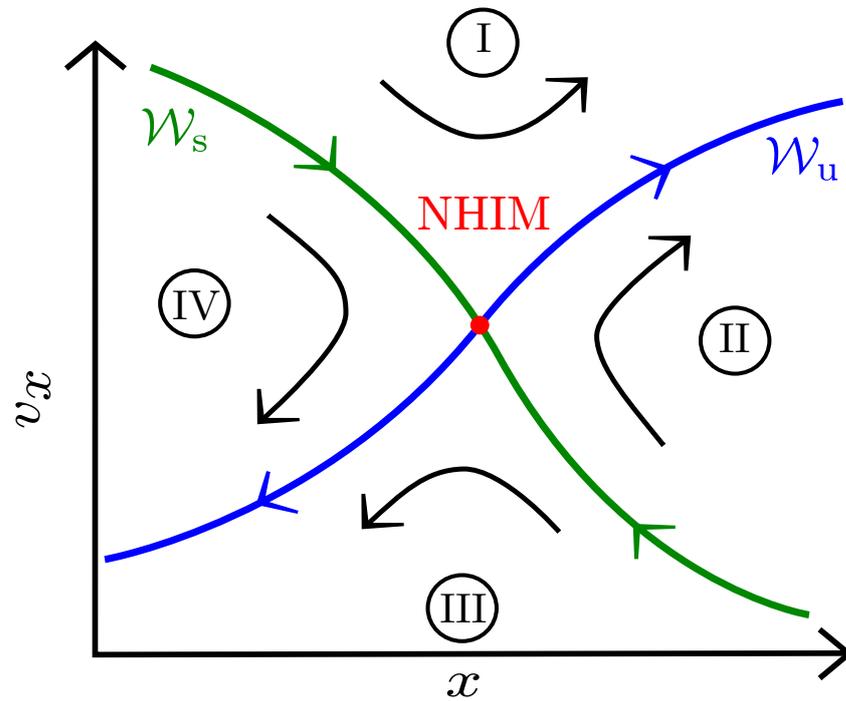


Figure 2.1: The four classes of particles with stable and unstable manifolds and the NHIM at the intersection of the manifolds (adapted from [22]).

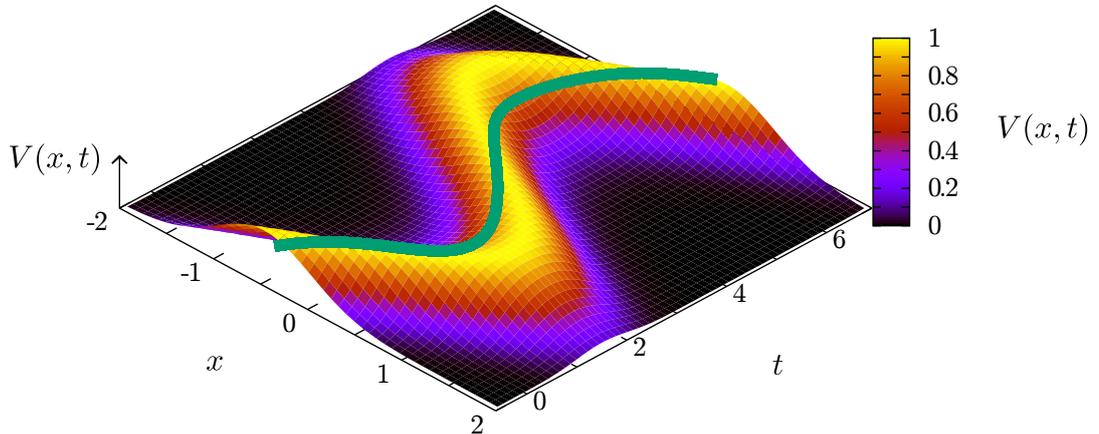


Figure 2.2: Time-dependent 1-DoF system, with a gaussian barrier in x -direction. The position of the barrier maximum is time-dependent and follows a sine oscillation. The NHIM is shown as a solid line in green and follows the sine oscillation with reduced amplitude, due to the inertia of the particles forming the ensemble.

systems with flat wells the particle distribution is not necessary statistical [34]. A method that does not need an ensemble is a promising way to circumvent this problem. This comes in addition to the aforementioned decrease in computational time.

2.1.2 Floquet Analysis of the NHIM

This section motivates how *Floquet analysis* might be used to obtain a reaction rate constant of a system from the stability properties of a trajectory. The section starts with a short review of *monodromy matrices* in a Hamiltonian systems, following Ref. [54]. For simplicity a generalisation of the configuration space coordinates x_i (for the i -th DoF, in a system with d DoF) and the momentum coordinates p_i to the phase space coordinate is used

$$\boldsymbol{\gamma} := (x_1, x_2, \dots, x_d, p_1, \dots, p_d)^T. \quad (2.5)$$

With this notation the equations of motion transform to (Ref. [54])

$$\dot{\boldsymbol{\gamma}} = \mathbf{J} \frac{\partial H}{\partial \boldsymbol{\gamma}} \quad \text{with} \quad \mathbf{J} := \begin{pmatrix} \mathbf{0}_d & \mathbf{1}_d \\ -\mathbf{1}_d & \mathbf{0}_d \end{pmatrix}, \quad (2.6)$$

where H denotes the Hamiltonian of the system, $\mathbf{1}_d$ and $\mathbf{0}_d$ describe the d -dimensional identity-matrix respectively zero-matrix and d corresponds to the degrees of freedom of the system [54]. Based on this, a stability or monodromy matrix can be defined as

$$\mathbf{M}_{ij}[\boldsymbol{\gamma}(0), t] := \frac{\partial \gamma_i(t)}{\partial \gamma_j(0)}, \quad (2.7)$$

which describes the propagation of a trajectory starting from $\boldsymbol{\gamma}(0)$. By considering two initially neighbouring trajectories, it follows by chain rule from Eq. 2.6 and Eq. 2.7 the differential equation

$$\dot{\mathbf{M}} = \frac{\partial^2 H}{\partial \boldsymbol{\gamma}(t)^2} \mathbf{M}(0), \quad \mathbf{M}(0) = \mathbf{1}_{2d}. \quad (2.8)$$

The latter is the initial condition that simply follows from the property that a particle that gets propagated for zero time steps does not move at all. For this thesis the important fact about the monodromy matrix \mathbf{M} is that it is symplectic. This means for every eigenvalue λ of \mathbf{M} are $1/\lambda$ and their complex conjugates $\bar{\lambda}$ and $1/\bar{\lambda}$ also eigenvalues. Since the monodromy matrix has only real entries, the eigenvalues are either complex conjugated or of the form λ and $1/\lambda$ [54].

The Floquet analysis of Ref. [29] for 1-DoF-systems shows that the difference between the two eigenvalues is connected to the reaction rate. In the following m_1 denotes the larger

eigenvalue and $m_s = 1/m_1$ denotes the smaller one. The so called *Floquet exponents* are defined by

$$\mu_{1,s} := \frac{1}{T} \ln |m_{1,s}|, \quad (2.9)$$

where T corresponds to the period of the trajectory. Physically, the Floquet exponent describes the factor by which two neighbouring factors separate exponentially. In Ref. [29] it was shown for a 1-DoF-system that the Floquet exponents can be used to calculate the reaction rate k_f in the following way:

$$k_f = \mu_1 - \mu_s \quad (2.10)$$

For systems with more DoFs there are more eigenvalues. If the system has a rank-1-saddle, two eigenvalues should correspond to the unstable degree of freedom and all other eigenvalues should correspond to the stable degrees of freedom. It seems plausible that the unstable degree of freedom gives a pair of real eigenvalues which are inverse to each other, but not equal to 1. This is due to the fact that $m \neq 1$ gives $\mu \neq 0$ according to Eq. 3.2. This means that neighbouring trajectories divide exponentially from each other in this DoF. On the other hand, all stable degrees of freedom should give pairs of complex eigenvalues with absolute value 1, since this implies a Floquet exponent of zero. Hence, neighbouring trajectories do not divide exponentially by each other in this degree of freedom. In section 3.1 we want to validate this assumption for a 2-DoF-system with a rank-1-saddle.

2.1.3 Neural Networks

The main topic in this paper is to use a NN to modify numerical integrators. Therefore it is recommended to recap the idea of NNs. Here, we will give a brief summary about NNs and refer the interested reader to the literature e. g. [55, 56].

In general, NN consists of *layers* which correspond to vectors in a mathematical point of view, see Fig. 2.3a. Every layer is composed of *neurons*, which correspond to the entries of the vector and can be seen in Fig. 2.3b. A simple form of NNs are *feed-forward* neural nets, which propagate information only in one direction. The values for the $(l + 1)$ -th layer are obtained as follows:

1. Multiply the l -th *weight matrix* $w_{ij}^{(l)}$ with the values of the l -th layer $c_j^{(l)}$.
2. Add a *bias vector* $b_i^{(l)}$ to the result of step 1.

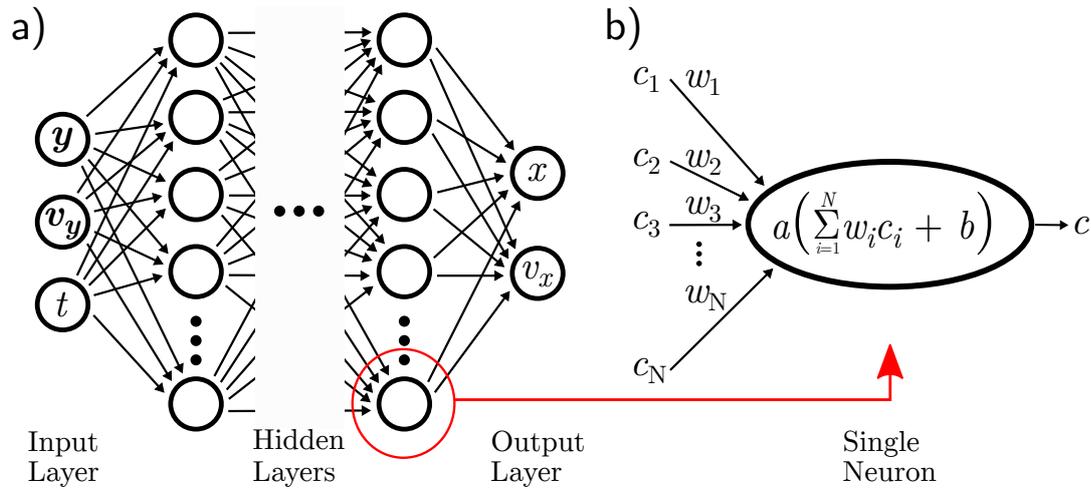


Figure 2.3: Basic construction of a feedforward neural net and neuron. (Adapted from [57])

3. Apply the (usually) nonlinear *activation function* a componentwise.

Mathematically, this corresponds to

$$c_i^{(l+1)} = a \left[\left(\sum_{j=1}^N w_{ij}^{(l)} c_j^{(l)} \right) + b_i^{(l)} \right]. \quad (2.11)$$

These three steps get applied layer by layer, starting from the input layer and ending in the output layer. Usually every neuron in one layer has the same activation function.

The so called *loss* or *cost function* gives a measure for the quality of the NN. A very common loss function is the mean-squared error

$$C_{\mathbf{w}, \mathbf{b}}(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{2n} \sum_{i=1}^n \|\mathbf{y}_i - \tilde{\mathbf{y}}(\mathbf{x}_i)\|^2. \quad (2.12)$$

Thereby \mathbf{x}_i names the input vector of the NN, $\tilde{\mathbf{y}}(\mathbf{x}_i)$ is the corresponding output vector, \mathbf{y}_i is the value that the output should be according to the *test data* or *trainings data*. Both the test and the training data get calculated with the binary contraction method, which is briefly described in section 2.2.2 and in more detail in Ref. [22]. The number of trainings points or the batch size is denoted by n and the subscripts \mathbf{w} , \mathbf{b} show the

dependence on the weight matrices and the bias vectors. During the training of a NN the weights, i.e. the values of the weight matrices and the bias vectors, are adjusted such that the loss is minimized. The specific properties of NNs, e.g. activation functions, net topology and hyper-parameters, used in this thesis are described in section 2.2.2.

2.2 Numerical Methods

In this thesis we consider time-dependent systems with a single rank-1-saddle, i.e. with at least one unstable degree of freedom. As mentioned before, there is one unstable degree of freedom, that dominates the instability of the system. The latter is the major reason why trajectories with a small distance δx to the NHIM diverge exponentially from the NHIM. A visualization of this divergence process is shown in Fig. 2.4 for four trajectories with different initial distances $\delta x_{\text{initial}}$ to the NHIM. Since the initial position of the NHIM in numerical calculations can only be given up to *machine precision*, it is challenging to study the long term behaviour of the NHIM with a regular integrator.

This thesis introduces a method that prevents an exponential increase of δx . Since this requires a method to calculate the NHIM for discrete points, such a method is introduced in the next subsection. The second subsection describes how these discrete NHIM points can be used to modify an arbitrary numerical integrator, such that δx does not diverge.

2.2.1 Calculation of Discrete Points of the NHIM

This section describes the *binary contraction (BC) method* [58], that enables to determine single points on the NHIM. More precisely, in a time-dependent n -DoF-system the BC allows to calculate the x and v_x coordinate of the NHIM for a given set of the $2(n - 1)$ other phase space coordinates. Since the BC is described precisely in Ref. [58], this thesis will in the following only give a brief overview of this method.

As shown in Fig. 2.5 the BC starts with four starting points forming a rectangle. Each of the four corners (1-4) of the rectangle corresponds to one of the following classes of particles:

1. reactants who react forwards to products
2. products who stay products
3. products who react backwards to reactants
4. reactants who stay reactants

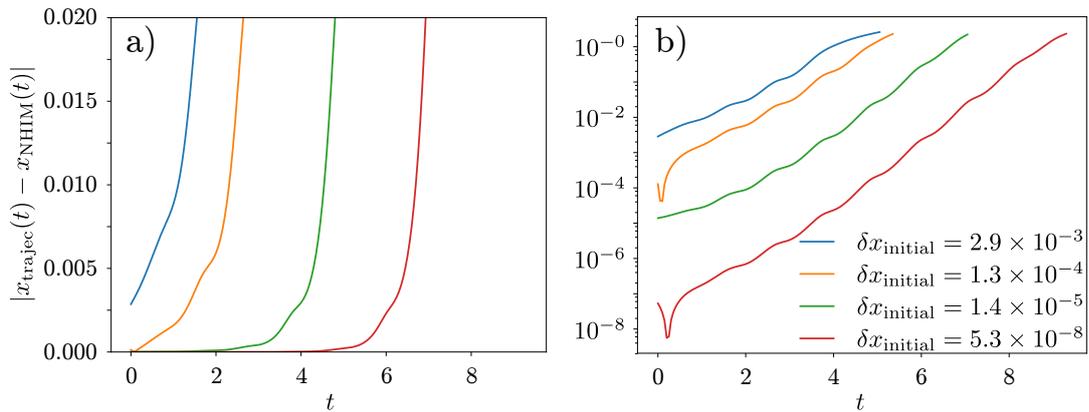


Figure 2.4: Distance between the NHIM and trajectories starting close to the NHIM over time. The four trajectories start with different initial distances to the NHIM as given in the key [58].

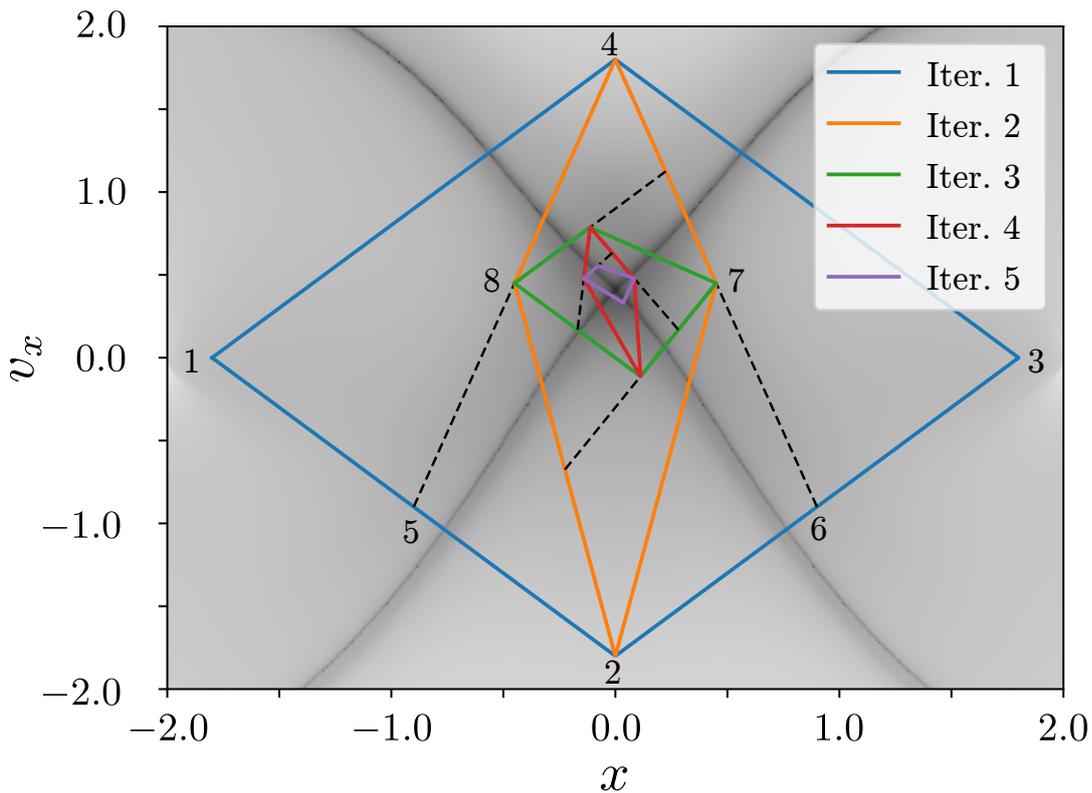


Figure 2.5: Binary contraction algorithm to determine one point on the NHIM. The algorithm starts in the first iteration with four points forming a rectangle. Each corner corresponds to one of the first four particle classes introduced in section 2.1.1. (Adapted from [58])

Section 2.1.2 introduces these four types of particles together with the only remaining fifth class, which is the NHIM. In the first iteration the midpoint (5 in Fig. 2.5) between two corners (1 and 2) gets also assigned to one of the four classes. Thus there are two points of the same class. Moving the initial quadrangle corner to the new representative of this class *contracts* the included area of the quadrangle. This is done iteratively and counter clockwise for every edge. This method ensures that for every iteration step the NHIM lies inside the quadrangle. The algorithm stops if the desired determination precision for the NHIM is reached.

It can be seen that the algorithm has a logarithmic runtime in the determination precision. This follows from the fact, that the length of the edges is divided by a factor of 2 in every iteration. Nevertheless, the computational time for a single point is in the order of 1 s, for my system. In comparison the propagation of a short and small NN takes in the same system about 5 ms. For this reason the BC gets only used to determine the test and training data for the NN. A major disadvantage of using NNs is that it only allows precision of about 1×10^{-4} phase space units. In contrast to that, the BC would allow to determine the NHIM up to 1×10^{-15} phase space units.

2.2.2 Modified Numerical Integrator

Due to the unstable degree of freedom, particles separate exponentially from the NHIM. To prevent this, a stabilization for numerical integrators gets invented in this thesis. The stabilization is done by alternated application of a numerical integrator and a projection on the NHIM. Applying the aforementioned BC after every integrator step would slow down the numerical integrator by magnitudes. Therefore a NN is used to approximate the NHIM. The following subsection discusses the specific properties of the NNs used in this thesis, e. g. the activation functions, the net topology and the hyper-parameters.

Training of the Neural Network

The NN-design used in this thesis is based on previous results in [59]. The so far used NN was constructed to approximate the DS. As mentioned before, the DS is a $(2n - 1)$ -dimensional object for a time-dependent system with n DoFs. Therefore, in a two-DoF system the DS is three-dimensional. Hence, for a set of coordinates in the form y , v_y and t the resulting x coordinate of the DS is uniquely determined, i. e. the DS is a function

of the form $x^{\text{DS}}(y, v_y, t)$. For this reason a NN that approximates the DS gets trained with input and output vectors of the form

$$\mathbf{v}_{\text{input}} = \begin{pmatrix} y \\ v_y \\ t \end{pmatrix}, \quad \mathbf{v}_{\text{output}} = x. \quad (2.13)$$

Since the NNs in this thesis approximate the NHIM they use the same input, but have in addition a second output neuron, which gives the corresponding v_x value.

To simplify the description of NN topologies, the notation

$$(\alpha_1, \dots, \alpha_i, \alpha_{i+1}, \dots, \alpha_n) \quad (2.14)$$

is introduced, for an n -layer NN with α_1 input neurons, α_n output neurons and α_i neurons in the i -th layer.

The previous work used a NN of the form (3, 40, 10, 1) to approximate the DS. Studies in this thesis use a (3, 100, 100, 30, 2)-net as a NHIM approximation, since two outputs require more fit-parameter in form of matrix- and bias-weights. In addition to that (3, 40, 40, 40, 10, 1)-nets are used in this thesis to approximate the DS. This is necessary for the ensemble propagation, in order to compare the reaction rates of the Floquet method with the reaction rates of the ensemble method. The enlargement of the NN in comparison to the previous work allows for a shorter training time to reach the same loss, or to reach a lower loss with the same training time. This aspect is described in more detail in section 3.3.

Furthermore the previous work used $\text{atan}(x)$ as activation function for every layer, besides the last one. The last layer requires an activation function that has no upper or lower bounds, since the x coordinate is unbounded. Therefore a linear activation function is used. In contrast to the previous work this thesis uses the tangent hyperbolicus $\tanh(x)$ as activation function in every layer besides the last one. The investigations show that this is also works as an activation function.

The training of the NNs is done with a machine learning library for *Python* [60] called *Tensorflow* [61]. This is in contrast to the previous work, where a C++ [62] custom implementation of the NN was used. One reason to use a library is that it allows to test a large variety of functionalities, like different optimizers, the use of batches and different initializers. A detailed analysis of there parameter dependencies can be found in section 3.3.

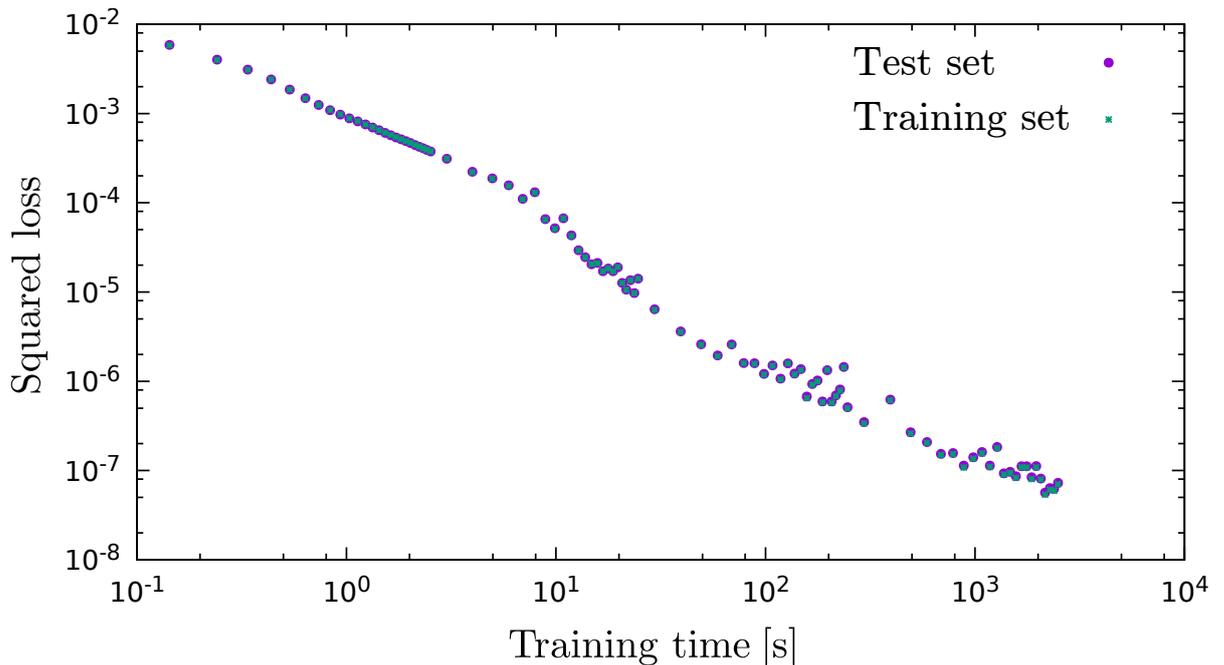


Figure 2.6: Typical training progress for about 5×10^4 trainings data for 1.5 periods of potential oscillation. The green crosses denote the training data, from which the gradient during learning get calculated and minimized. The test data (violet dots) have never been used for training and hence can be seen as an independent statistical metric for the quality of the neural net.

The NN are mostly trained for 5×10^4 epochs with 5×10^4 training points. Therefore during training $(5 \times 10^4) \cdot (5 \times 10^4) = 2.5 \times 10^9$ gradients are calculated. This is nearly equal to the previous work, which had 2.5×10^9 gradient calculations due to the 2×10^3 training points used in 1×10^6 epochs of training. A typical training behaviour for a NN that approximates the NHIM is shown in Fig. 2.6. It can be seen that the loss does not saturate until this point in the double logarithmic plot. However, it is not necessary to continue the training since the required precision is reached.

It is obvious that on the one hand this thesis puts some effort in the optimization of the NNs, but on the other hand an even more detailed analysis of the hyper-parameters would be possible. However, it has to be kept in mind that it is in principle possible to stabilize a numerical integrator on the NHIM without a NN by using solely the BC. A less frequent use of the BC might solve the aforementioned runtime problem for the numerical integrator stabilized with BC. Further investigation of the hyper-parameters was not conducted due to the possible use of this method in future works.

3 Results

A fundamental concept in physics is to divide large and complex problems in smaller and simpler subproblems. For this reason, model systems with specific properties are introduced to investigate a certain aspect.

Model System

In this thesis, the NHIM [1–8] should be time-dependent. Hence our model system has to be time-dependent, since this is the only way to get a time-dependent NHIM.

Furthermore we want to have at least two independent DoF, because in a one-DoF system there is no motion within the NHIM, since it is a zero-dimensional object for every point in time. The model system from Ref. [63]

$$V(x, y, t) = E_b \exp(-a[x - \hat{x} \sin(\omega_x t)]^2) + \frac{\omega_y^2}{2} \left[y - \frac{2}{\pi} \arctan(2x) \right]^2 \quad (3.1)$$

fulfills both these requirements and is visualized in Fig. 3.1.

It contains a Gaussian barrier with height E_b and reciprocal width a and is oscillating along the x axis with frequency ω_x and amplitude \hat{x} . The y direction shows a harmonic potential with frequency ω_y . A nonlinear coupling between the coordinates follows from the term $(2/\pi) \arctan(2x)$ which corresponds to the minimum energy path. For simplicity, all variables are dimensionless and in accordance to Ref. [63] set to $E_b = 2$, $a = 1$, $\omega_x = \pi$, $\omega_y = 2$ and $\hat{x} = 0.4$. The latter allows for a comparison with their results and gives us some knowledge about the system.

3.1 Phase Space Structures of the NHIM

To visualize trajectories on the NHIM in the following we use Poincaré surface of sections by showing only the y and v_y coordinate. Furthermore, we use a stroboscopic view, which

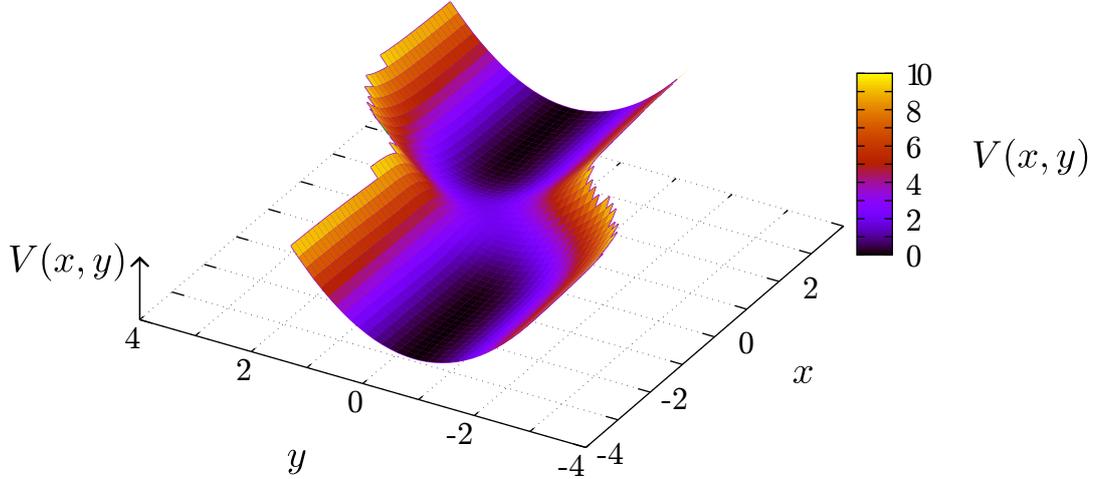


Figure 3.1: The potential surface with the arctan in x - y -direction and the parabola in y -direction, see Eq. 3.1.

means that we separate the periodic driving in the figures by showing an individual trajectory only after a full period.

Some exemplary trajectories in this stroboscopic view can be seen in Fig. 3.2 for varying \hat{x} . Remarkably, all trajectories lie primarily on ellipses, which means that the phase spaces of these four systems exhibit torus structures. Due to this fact, the systems are regular, i. e. they are non-chaotic.

Furthermore, torus structures imply that the system is *near-integrable* and has a *conserved quantity*. For reasons of simplification, the conserved quantity is in the following called χ . It should be mentioned that it is unknown what physical quantity corresponds to χ .

Note that it is not obvious that no dissipation occurs. The projection with the neural net during integration could lead to systematic errors, which could cause a divergence of the particle energy. Of course it could be that for long integration times (i.e. 50 system periods) both effects occur in a weak form, but cancel each other. However, the studies of the TSt (see section 3.3.1) as an exemplary trajectory show that for one period the modified integrator conserves χ , if the used neural network is trained “good enough”. What this precisely means is described at the end of this section. Before that, we want to discuss the potential parameter dependency of the TSt start point and the tori.

Comparing the four plots of Fig. 3.2 shows that the χ conservation holds for every of the chosen \hat{x} . Furthermore it can be seen that for increasing \hat{x} the v_y coordinate of the TSt (and hence the center of the tori) decreases. A larger \hat{x} implies a larger motion of the barrier top, which is related but not equal to the motion of the NHIM. The nonlinear

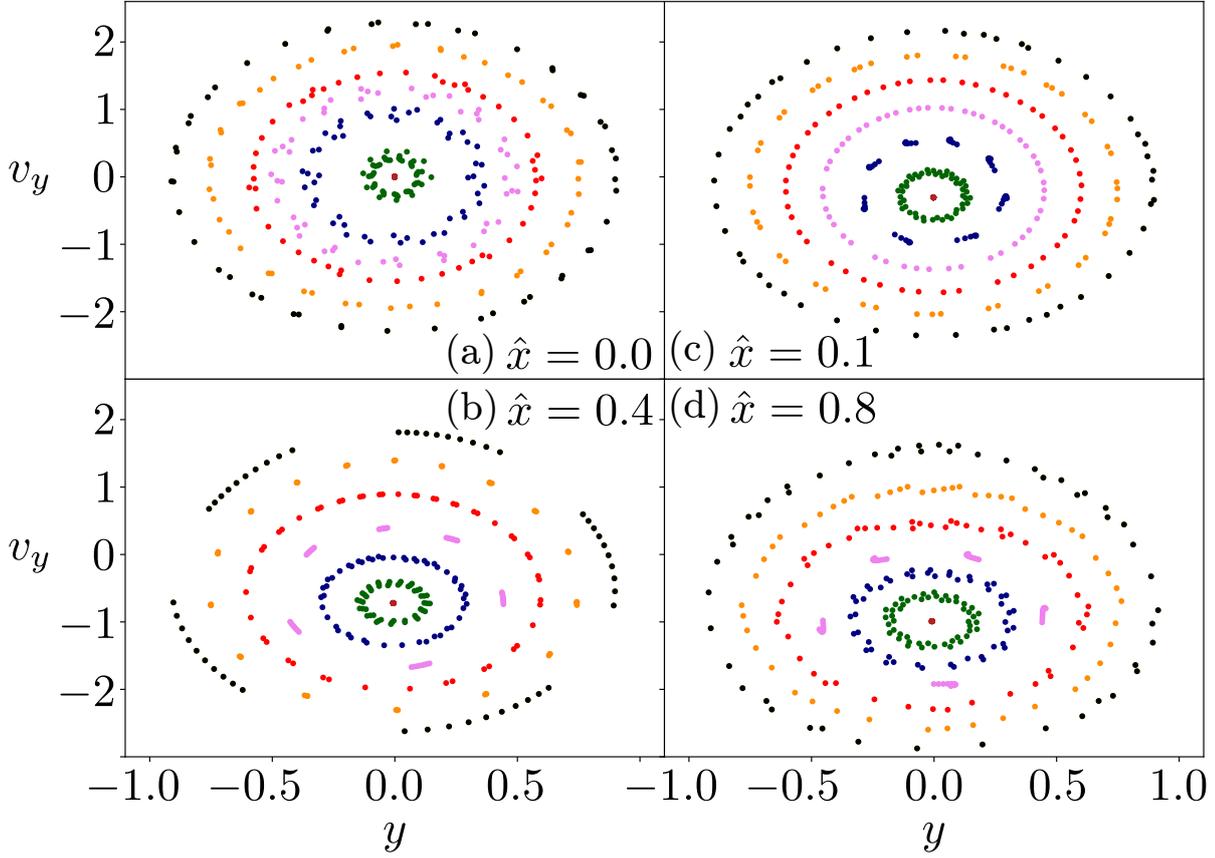


Figure 3.2: Poincaré surface of section in stroboscopic view for trajectories with different initial conditions and varying \hat{x} . All system parameters besides the oscillation amplitude \hat{x} correspond to the standard values, introduced at the beginning of the chapter. For the upper left figure it is $\hat{x} = 0.0$. In the upper right it is $\hat{x} = 0.1$, for the lower left it is $\hat{x} = 0.4$ (standard value of our group) and the lower right figure has $\hat{x} = 0.8$. The trajectories were determined by the aforementioned stabilized numerical integrator and sampled around the TSt. Integration happened for 50 system periods. Nets for the stabilized integrator are trained for 5×10^4 epochs with batch sizes of 500 and have three layers with (100, 100, 40) hidden neurons. All training data were generated with integrator step size of 1×10^{-5} .

coupling of the potential by the arctan causes an increasing motion in the v_y - y subspace. Since the phase of the potential in Fig. 3.2 was chosen such that $t = 2\pi n$ with $n \in \mathbb{N}$, it follows that v_y increases for this particular TSt-starting point. For $t = 2\pi \cdot n + \pi$, v_y would increase with \hat{x} and for $t = 2\pi \cdot n + 1/2 \cdot \pi$ it would be $v_y \approx 0$ but $|y|$ would reach a maximum.

For the analysis of the phase space structures it is an important question if the chaoticity depends on \hat{x} . This question gets even more difficult due to the varying quality of the neural nets. The latter causes a slight noise within the ring structures in the upper left ($\hat{x} = 0.0$) and the lower right ($\hat{x} = 0.8$) figure of Fig. 3.2. Thus the other systems seem to be less chaotic, although none of the figures show (in the estimate of the author) irregular/chaotic behavior. In further analysis it could be interesting to see if very high values of \hat{x} would change this. In addition to this, influence of other system parameters could be researched.

As mentioned before, the modified numerical integrator is only χ preserving if the “quality” of the used neural net is sufficient. In the following, this will be analyzed with the aid of Fig. 3.3. Here, the selected training data area influences the integrator behaviour. The NNs for the two upper plots were trained with equidistant training and test data on a grid with 40 values in the range $-4 < y < +4$, 40 values in the range $-8 < v_y < +8$ and 50 values in the range $-0.25T < t < 1.25T$. T denotes the period of the potential. The time interval is chosen larger than a full period to ensure continuity at $t = nT$ with $n \in \mathbb{N}$.

The data for the plots in the middle of Fig. 3.3 form the same grid, with the only difference that the y -direction is scanned from -0.5 to $+0.5$ and in v_y -direction from -1 to $+1$. Since the amount of data points stays the same, the grid is smaller but denser. For the lower plot, a neural network was trained on the combined data.

It can be seen that the larger data range yields good results for trajectories on a wider area around the TSt (upper left) and the finer data on a small range yield good results for the narrow area around the TSt (middle right). Nevertheless both fail in the opposite cases. Furthermore, it can be seen that strong interpolation (upper right) still works better than extrapolation (middle left). This meets the expectation that approximation methods do not handle extrapolation well.

Obviously, the goal should be to get a neural net that behaves well in both the small and the large area around the TSt. To achieve this, the neural net for the two lower plots were trained on both data sets at once. The result is a single net, that is indeed competitive with the combination of both previous nets. It should be noticed that combination of

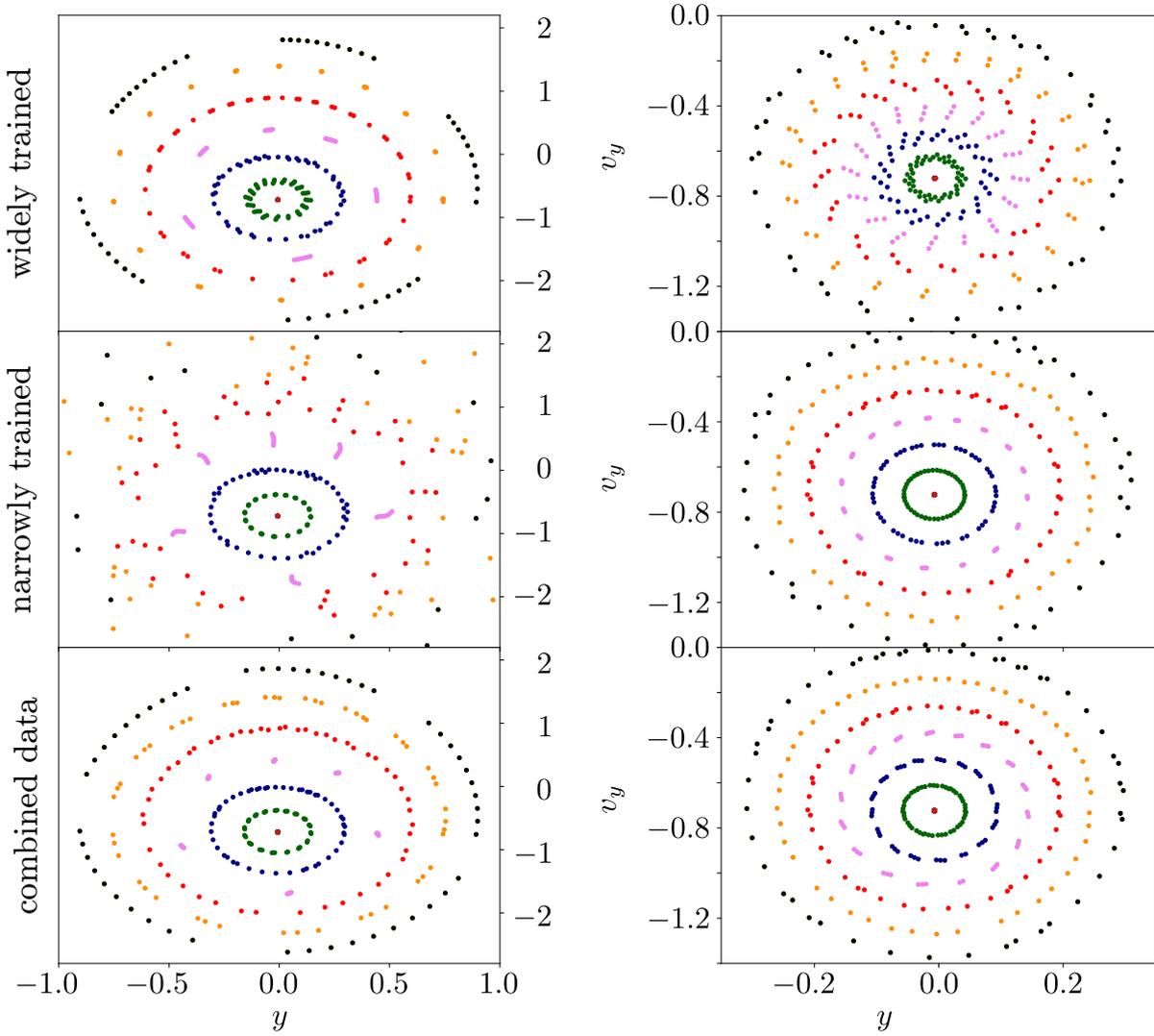


Figure 3.3: Poincaré surface of section in stroboscopic view for trajectories with different initial conditions. The trajectory in center of every figure corresponds to the TSt. Six more trajectories are sampled with a y gap of 0.15 in left figures and a gap of 0.05 in the right figures. The trajectories were integrated for 50 periods and sampled around the TSt. The neural net used in the upper figures was trained on a training set that represents a large but coarse phase space area. The net for the figures in the middle uses the same amount of training data but in a smaller phase space area (hence they are more dense). The bottom figures use both of the aforementioned training sets. Every net was trained for 5×10^4 epochs, with batch sizes of 500 and has three layers with (100, 100, 40) hidden neurons. All training data were generated with integrator step size of 1×10^{-5} .

training data in this way requires knowledge about the position of the TSt. This is one reason why the next section presents two algorithms to find a point on the TSt.

3.2 Determination of the Transition State Trajectory

One of the goals of this thesis is to apply the Floquet analysis to the transition state trajectory (TSt). For this reason, the localisation of the latter is an important interim step. In the following, we present two algorithms to find a point on the TSt from which the trajectory itself can be obtained by numerical integration with our NHIM stabilized integrator. The idea of both algorithms is to iteratively minimize the phase space volume that is enclosed by an arbitrary initial trajectory. This is legitimate for us, since we define the TSt as the periodic trajectory on the NHIM with the lowest average energy over one period.

3.2.1 Centroid Search Algorithm

The first algorithm (*centroid search*) uses the fact, that the system is periodic, so the potential is identical after a full period in time. Therefore, we consider the trajectory only for points in time where the phase of the periodic driving (*barrier phase*) is identical. The results are shown in a stroboscopic view in Fig. 3.4 as red curves. The v_y and y position of the starting point for the next trajectory are equal to the centroid or geometric center of the previous trajectory. The v_x and x position are given by the corresponding coordinates on the NHIM.

Fig. 3.4 also shows that the centroid search converges relatively fast (within 80 system periods) for all of the four different system parameter but depends on the same. In particular, the rate of convergence decreases with increasing \hat{x} because a large \hat{x} leads to strong nonlinearity in the NHIM. If this is the case, then the v_x and x coordinates of the next iteration are far away from the centroid in this direction. The opposite case can be seen in the upper right plot of Fig. 3.5 where $\hat{x} = 0$. Here is no barrier motion; hence, the centroid in v_x - x direction is nearly perfectly aligned with the NHIM-position of the centroid in v_y - y direction. For $\hat{x} = 0.1$ the barrier motion is small, so the x - y -dependency of the NHIM is mostly linear. Therefore the centroid and the NHIM-position are at least close together. This can be seen in the upper right plot of Fig. 3.5. In the two remaining cases the rate of convergence is significantly slower due to the strong nonlinearity of the NHIM.

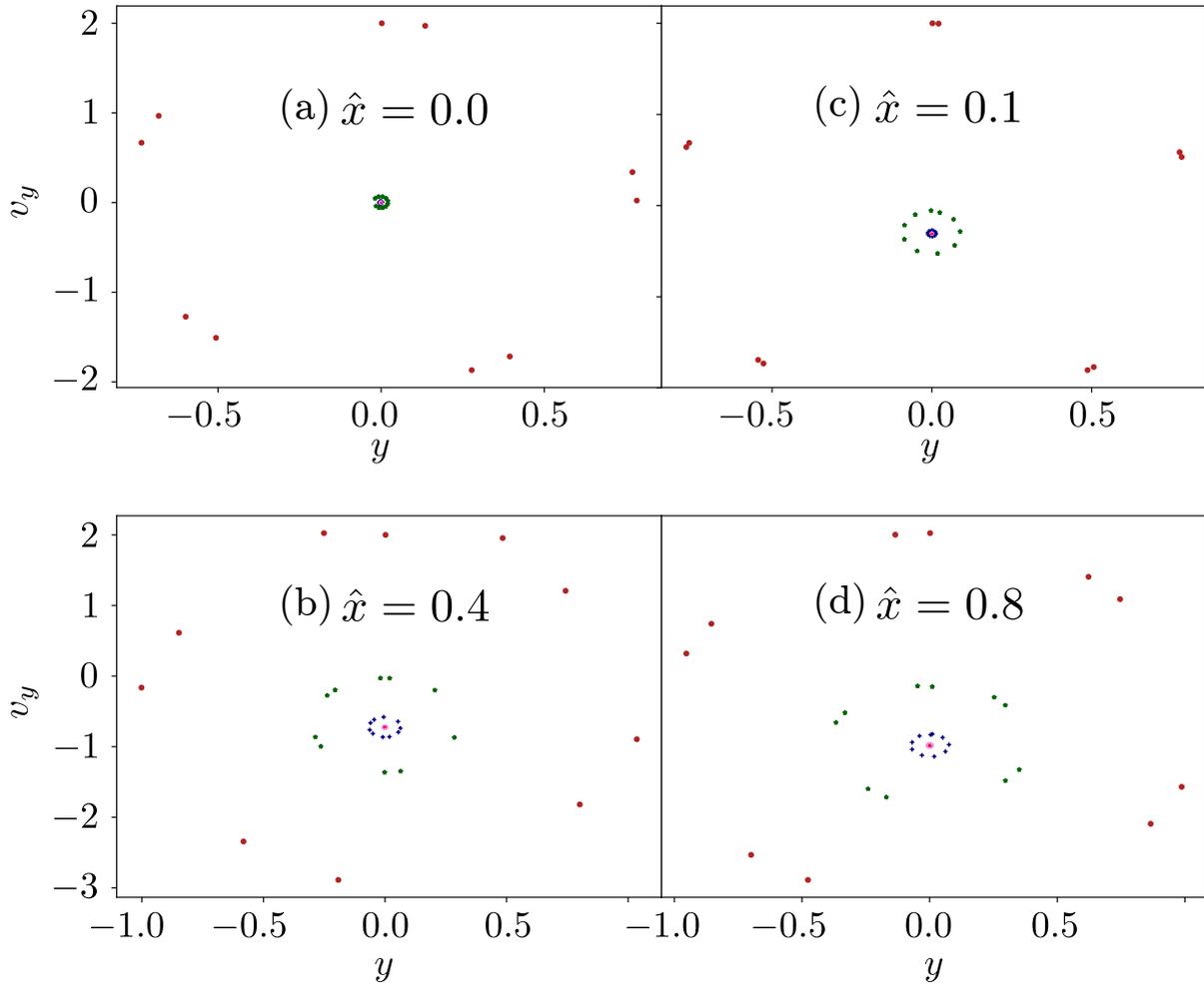


Figure 3.4: Stroboscopic view of trajectories for 10 periods determined by the aforementioned modified integrator forced on the NHIM. In the four single figures the system parameter \hat{x} (corresponding to the driving amplitude) is varied. The centroid of the n -th trajectory in v_y - y -direction corresponds to the starting point of the $(n + 1)$ -st trajectory. The algorithm converges to the periodic trajectory at the center.

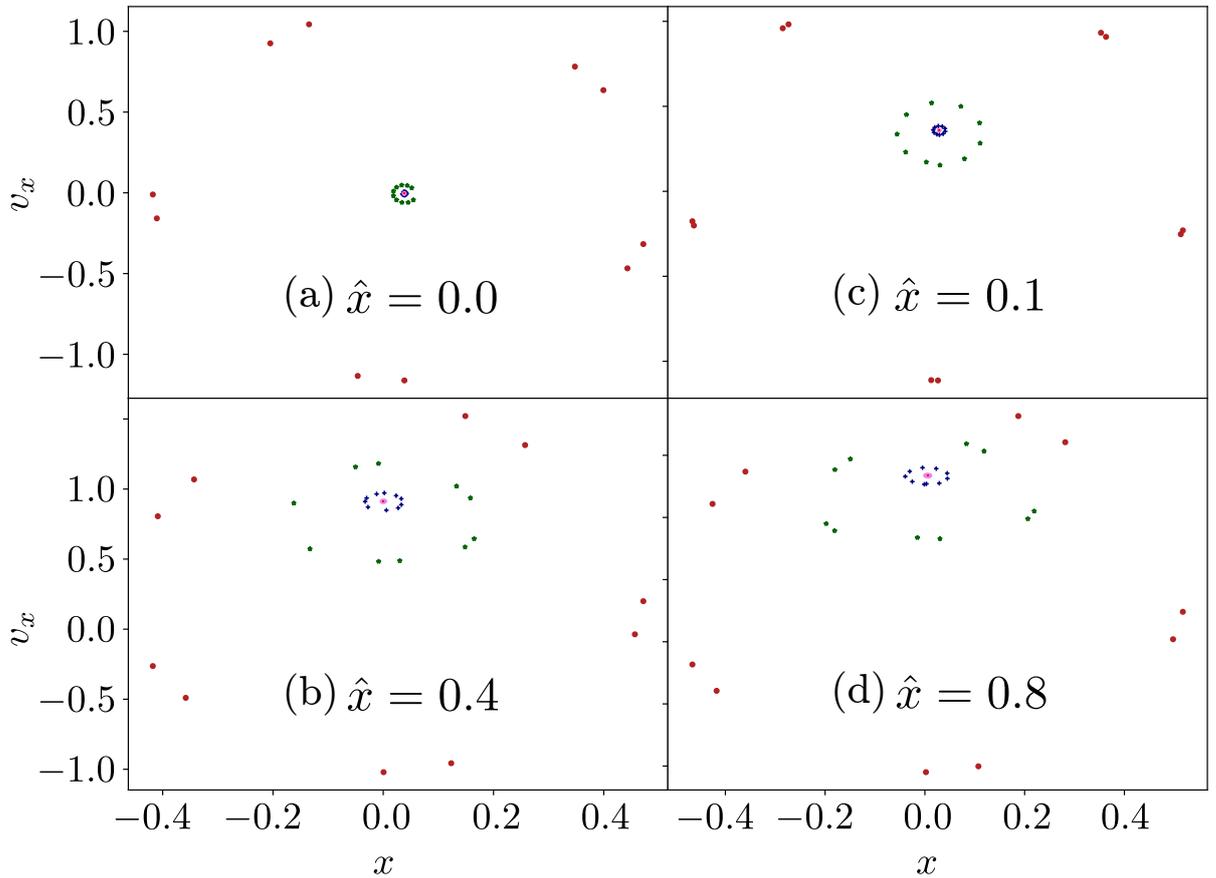


Figure 3.5: Stroboscopic view of trajectories for 10 periods determined by the aforementioned modified integrator forced on the NHIM. In contrast to Fig. 3.4, it shows v_x over x and not v_y over y . In the four single figures the system parameter \hat{x} (corresponding to the driving amplitude) is varied. The centroid of the n -th trajectory in v_y - y -direction corresponds to the starting point of the $(n + 1)$ -st trajectory. However the starting points in v_x - x -direction are determined by the NHIM and therefore do not necessarily correspond to the centroids in this direction.

The result after 9 iterations for $\hat{x} = 0.4$ is shown in Fig. 3.8 and is discussed in section 3.3.1. The major advantage of this algorithm is its fast convergence, which results in short computation times. Since not every system is periodic, the stroboscopic view is not always possible and therefore the centroid can not be calculated. This lead to the development of an algorithm that works even in this case and is the topic of the next section.

3.2.2 Friction Search Algorithm

As mentioned before, *centroid search* only works for periodic time dependent systems. Hence, another algorithm is developed for this task. In the following, it gets called *friction search algorithm*, since it introduces friction to the system. It has to be mentioned that this is not a physically correct friction, i. e. the friction does not obey the *Fluctuation-Dissipation-Theorem* [64].

The friction allows to reduce the energy of an arbitrary trajectory. Thus the latter converges to that specific periodic trajectory with the lowest possible energy on the NHIM. Fig. 3.6 shows that the friction search converges to a similar value as the centroid search does. The latter is marked with a blue cross. A large friction parameter in the algorithm leads to a fast convergence, since the energy dissipation is high, but the error in the determined TSt position is large in this case. Therefore, the algorithm starts with a large friction parameter and decreases it iteratively to get a precise point on the TSt. This process is shown in the left part of Fig. 3.6 for the v_y and y coordinates and in more detail for the x -coordinate in the right part of Fig. 3.6. The slow energy dissipation for small frictions is the reason why friction search converges slower than the centroid search. However, even for periodic systems friction search could be the better choice for extremely nonlinear systems if its rate of convergence is independent of the nonlinearity of the NHIM.

To give a magnitude for the computational time both algorithms take about 10s for the pure determination of the TSt on an *Intel(R) Core(TM) i5-3470 CPU* with 3.20GHz.

3.3 Stability Eigenvalues and Rate Constants

This section requires the knowledge of some of the previous sections. To analyze the stability properties of trajectories on the NHIM, the previously introduced modified integrator is needed, see section 2.2.2. The main focus is on TSts since their periodicity is crucial for Floquet analysis, see section 2.1.2. Although both TSt search algorithms

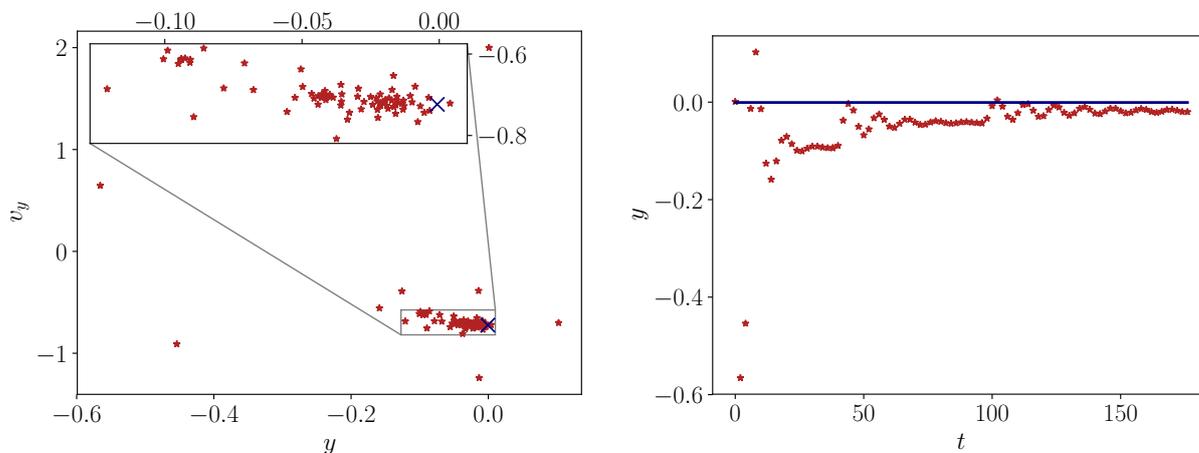


Figure 3.6: **Left:** Stroboscopic view of the convergence behaviour of the friction search algorithm (red dotted) for the TSst starting point. The trajectory forms a spiral, starting at $y = 0$ and $v_y = 2$ and ending at about $y = -0.02$ and $v_y = -0.7$, as shown in the inset. During the integration the friction was reduced twice. The blue crosses mark the points to which the centroid search converges. **Right:** Iterative convergence of the friction search (red dotted) for the starting point of the TSst to the result of the centroid search (blue line). The friction parameter was reduced twice, once after $t = 40$ and once after $t = 100$. High friction causes a decreasing oscillation amplitude ending in a plateau with a characteristic distance to the exact TSst starting point. Lowering the friction reduces this distance, but induces new oscillations.

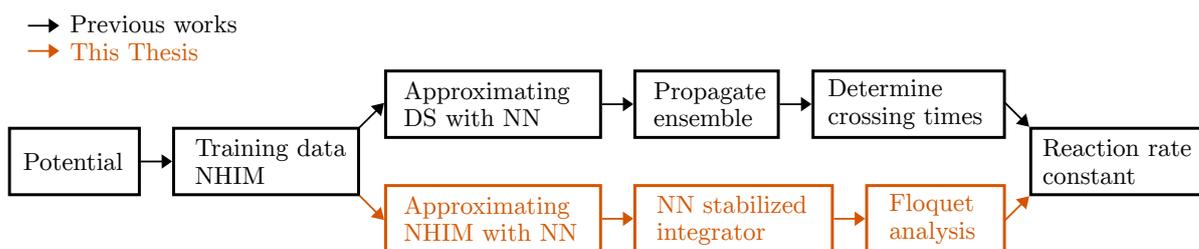


Figure 3.7: Sketch of two possible paths to determine the reaction rate beginning from the potential of a chemical system. In the upper path, a NN is used to approximate the DS and an ensemble is propagated. In the lower path, a NN is used to approximate the NHIM and Floquet analysis is used to determine the stability matrix and thus a reaction rate constant for the TSst.

(section 3.2) converge against almost the same point, it should be mentioned that the exact coordinates used in the further investigations are obtained by the centroid search algorithm.

In Ref. [29] it was shown for a 1-DoF-system that the Floquet exponents can be used to calculate a reaction rate k_f for a chemical systems. One reason to use this method is that it is a promising way for a computational efficient calculation of the reaction rate. A visualisation of the way from the potential to the reaction rate constant is shown in Fig. 3.7. In both cases, the rate calculation starts with the application of the binary contraction on the potential in order to obtain the training data for the NNs. In the upper case (*ensemble method*), the NN has one output, since it corresponds to the three-dimensional DS. In the lower case (*Floquet method*), it has two outputs for the two-dimensional NHIM. The DS-NN allows to determine the crossing times while the NHIM-NN is used to prevent the numerical integrator from leaving the NHIM.

The upper path in illustration 3.7 corresponds to the previously established path, by propagating a large ensemble of particles in a given potential. The major disadvantage of this method is that the computational time is proportional to the number of particles in the ensemble. Hence, a statistically significant result can only be obtained after a long computational time. In the lower path, instead of propagating an ensemble it is necessary to determine the TSt and calculate its stability eigenvalues. As mentioned before, the TSt determination takes about 10 s. Including the computation of the monodromie matrix, the reaction rate constant can in my case be determined in less than 30 s.

3.3.1 Floquet Analysis of NHIM Trajectories

In this section we use an initial point on the TSt (obtained using a centroid search) to determine the full TSt by integrating this initial point with both our NHIM stabilized numerical integrator and a regular integrator. The results are shown in a non stroboscopic view in the left part of Fig. 3.8. As can be seen that there is good agreement between the two different results for at least one period. The distance between the two trajectories in x -direction in the right part of Fig. 3.8 shows nearly exponential separation. This is due to the fact that the rank-1 saddle has one unstable degree of freedom that causes trajectories, which are close to the NHIM but not directly on the NHIM, to separate from the latter.

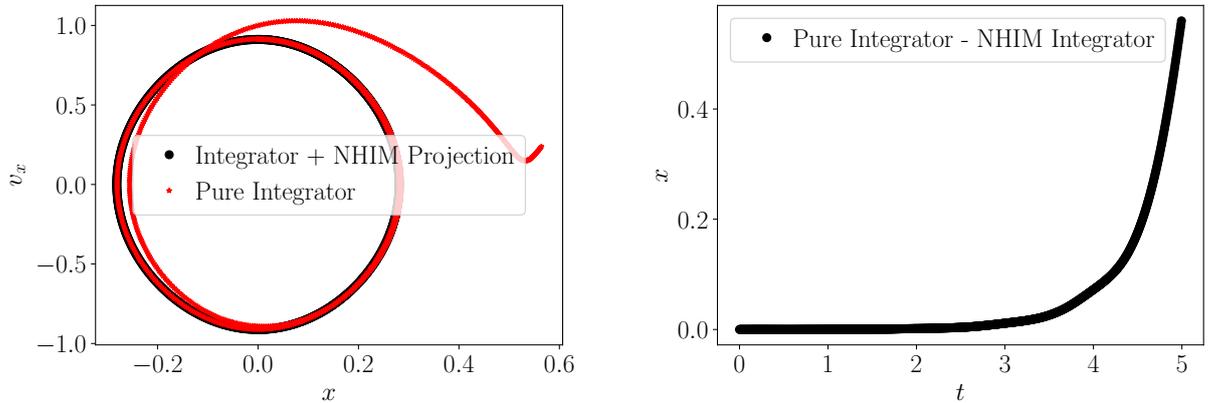


Figure 3.8: Results for integrating the TSt starting point, obtained by centroid search over 2.5 periods in the system with $\hat{x} = 0.4$. One period in time is equivalent to 2 time units. **Left:** Poincaré surface of section of the TSt, determined by centroid search. Trajectories are plotted for both, pure integrator and the integrator forced to the NHIM. **Right:** Distance between the x -coordinates of the two different integrators. Within one period, the separation is magnitudes smaller than the actual motion in x direction.

The good agreement for at least one period on the right side of Fig. 3.8 shows that:

1. Withing one system period, the projection does not change the underlying physical behaviour of the integrator, e. g. by creating artefacts in the trajectory.
2. The projection prevents the trajectory from leaving its perfectly periodic orbit.

A plausibility check can be done by comparing the amplitude of the TSt in x -direction $A_{\text{trajec}} \approx 0.25$ with the amplitude of the barrier oszillation $\hat{x} = 0.4$. The fact that A_{trajec} is smaller than \hat{x} confirms the assumption that the TSt can have no larger amplitude than the barrier tip.

The trajectory displayed in Fig. 3.8 is used for the Floquet analysis. Tab. 3.1 shows the eigenvalues m of the monodromy matrix and the Floquet exponents

$$\mu = \frac{1}{T} \ln |m| \quad (3.2)$$

(with period time T) of the TSt. It can be seen that the two real eigenvalues are inverse to each other. The two complex eigenvalues with absolute value of 1 are complex conjugated to each other. This shows that the monodromy matrix of our TSt is indeed symplectic, which is in coincidence with the theory. The corresponding Floquet exponents result in a reaction rate of $k_{\text{Floquet}} = 3.806$, see Eq. 2.10. Tab. 3.2 shows the results for all four rate

Table 3.1: Eigenvalues m and Floquet exponents μ of the TSt (determined with centroid search) for the potential barrier amplitude $\hat{x} = 0.4$.

| | m | μ |
|-------------------|------------------|--------|
| λ_1 | 44.976 | +1.903 |
| $1/\lambda_1$ | 0.0222 | -1.903 |
| λ_2 | $0.437 + 0.899i$ | 0 |
| $\bar{\lambda}_2$ | $0.437 - 0.899i$ | 0 |

Table 3.2: Reaction rate constants k_{Floquet} obtained by Floquet analysis for different barrier oscillation amplitudes \hat{x} . For the first four cases, the TSt trajectory was used to determine the rate constant. In the last case, a trajectory with periodicity of $6T$ was used for Floquet analysis.

| \hat{x} | Trajectory | k_{Floquet} |
|-----------|------------|----------------------|
| 0.0 | TSt | 2.762 |
| 0.1 | TSt | 2.980 |
| 0.4 | TSt | 3.806 |
| 0.8 | TSt | 4.017 |
| 0.4 | 6 Periods | 3.795 |

constants for the TSts and (as a comparison) the result for the trajectory that is closed after 6 system periods. The latter was found accidentally during the centroid search. Therefore, this rate constant is only determined for one specific system parameter.

The TSts in every of the four investigated systems show the structure of one pair of real eigenvalues and one pair with absolute value of 1. This is not necessary for symplectic systems, since all four eigenvalues could be complex without absolute values of 1 and behave like λ , $1/\lambda$, $\bar{\lambda}$ and $1/\bar{\lambda}$. Indeed, some trajectories on the NHIM show such eigenvalues. So far, no detailed investigation of this behaviour was done. Not only the TSts, but also the fifth trajectory in Tab. 3.2 shows two real eigenvalues and two with absolute values of 1, while some non-periodic trajectories do not. To be certain if this is coincidence or if every *closed* trajectory has this two eigenvalue couples and every *open* trajectory has not, more research is needed.

A comparison of the TSts for the four different systems in a Poincaré surface of section can be seen in Fig. 3.9. As expected, without motion of the barrier there is no motion

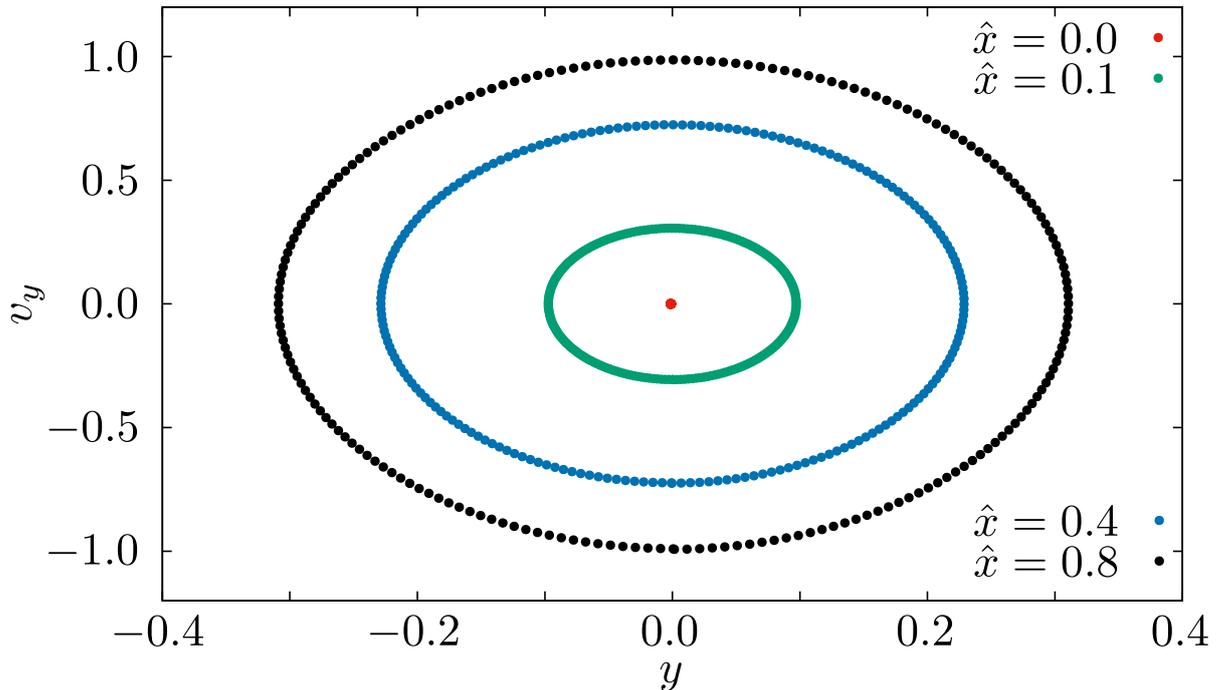


Figure 3.9: Poincaré surface of section of the TSs determined by centroid search for four different system parameter values of \hat{x} . The trajectories were only determined by the modified integrator forced to stay on the NHIM and show one period in time.

in TSt, which corresponds to the static border case. Increasing motion of the barrier in x -direction (increasing \hat{x}), leads to a higher amplitude of the TSs in y -direction. Furthermore, the maximum velocity in y -direction is also increased. Both these effects have been expected, since the x and y directions are coupled. Once again, it should be mentioned that these trajectories do neither show unphysical artefacts, nor have an offset between the start and the end.

3.3.2 Comparison with Rate Constants of the Ensemble Method

To set these reaction rates in context, we compare them with results obtained by ensemble propagation recording the crossing times. The results are visualized in Fig. 3.10 and can be found in Tab. 3.3. A more detailed description of this method can be found in section 3.4.1 in this thesis or for example in the master's thesis of Philippe Schraft [59] or the bachelor's thesis of Melissa Lober [65]. For the ensemble, two different initialisations are used. This results in two slightly different rate constants, which can be understood as

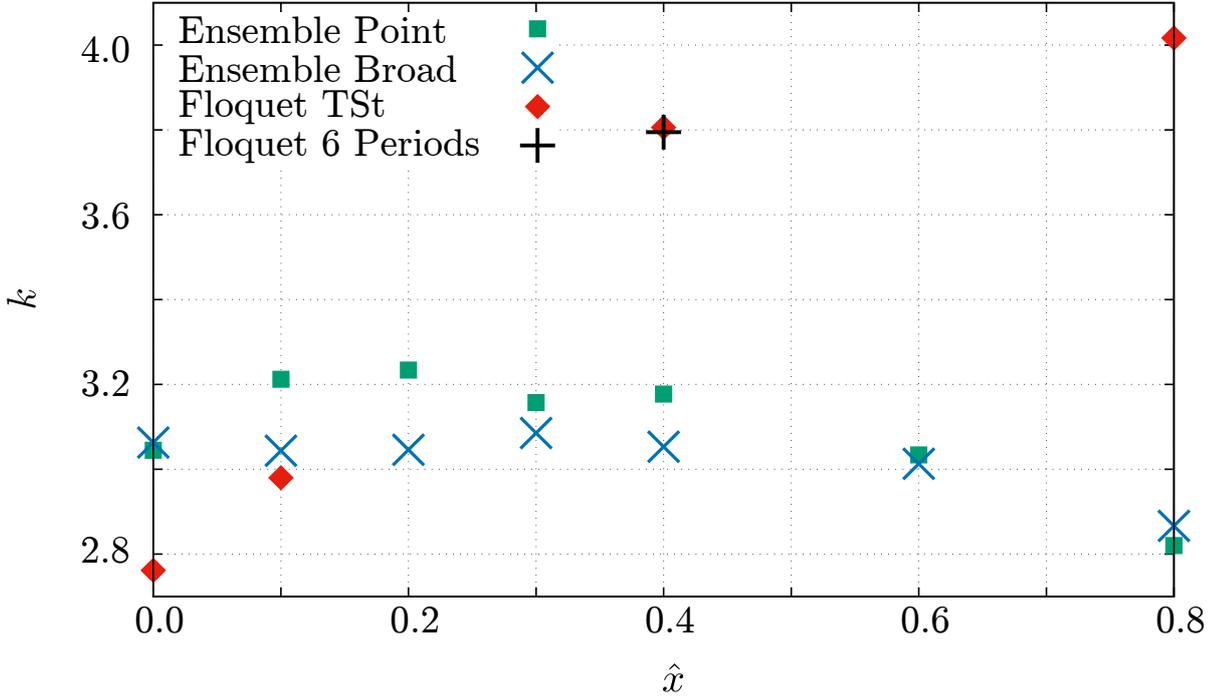


Figure 3.10: Comparison of the reaction rates for the two different ensembles and the trajectory analysed with the Floquet method. There are only a small number of Floquet method data points, since the parameter dependency of this method is only an incidental aspect of this thesis. The periodic trajectory that is no TSt was discovered unintentionally. Therefore, there is only one data point for this type.

Table 3.3: Reaction rate constant k_{Floquet} obtained by Floquet analysis for the TSts and the single trajectory with periodicity of 6 and obtained by ensemble propagation k_{Ensemble} for different barrier oscillation amplitudes \hat{x} . Every system is simulated with both an ensemble forming a single point in space but varying velocities v_x and v_y and with an ensemble that has a broad distribution in the initial coordinates of x .

| \hat{x} | $k_{\text{Floquet, TSt}}$ | $k_{\text{Floquet, 6 Periods}}$ | $k_{\text{Ensemble, Point}}$ | $k_{\text{Ensemble, Broad}}$ |
|-----------|---------------------------|---------------------------------|------------------------------|------------------------------|
| 0.0 | 2.762 | - | 3.045 | 3.063 |
| 0.1 | 2.980 | - | 3.212 | 3.044 |
| 0.4 | 3.806 | 3.795 | 3.177 | 3.053 |
| 0.8 | 4.017 | - | 2.820 | 2.867 |

a lower bound for the uncertainty of this method, or the magnitude of the statistical error of the latter. It has to be kept in mind that additional systematic errors are possible.

The results of the ensemble method show only small variations in the rate constant around $k = 3.0$. Only for $\hat{x} = 0.8$ is k significant smaller, which could be an outlier. In contrast to that, the rate constant of the Floquet method increases with the barrier amplitude \hat{x} . This seems counter-intuitive if it is assumed that the barrier motion hampers the particle reaction. So it might be that this is not the case but on the contrary the motion is helping the particles to pass the barrier. Alternatively, another effect could overlay this interpretation. In fact, all ensembles are initialized with the same energy, but the average energy of the studied TSts is an intrinsic property that cannot be changed. Therefore, TSts for different \hat{x} could simply have a systematically different average energy. This could result in a variation of the reaction rates. A final answer to this effect can only be given after further investigations. The trajectory with a period of 6 system oscillations has about the same reaction rate as the corresponding TSt. This could be seen as a hint that spatially close trajectories also have similar reaction rates. However more research is needed for a definite conclusion.

So far, it cannot be said which results are more trustworthy. The main problem of the ensemble method is that the reaction rate highly depends on the initialization of the ensemble. Although this dependency is physically comprehensible, it hampers the comparison with the Floquet method if the latter is only used for single trajectories. A possible solution for this problem could be the following:

1. Search many periodic trajectories on the NHIM,
2. calculate the eigenvalues of their monodromy matrices and
3. use these matrices to determine many rate constants for one set of system parameters.

A linear combination of these rate constants – according to a distribution that is determined by a certain ensemble – should give the same rate constant as a simulation with the ensemble method for the same ensemble. So far, neither a method to determine periodic trajectories of a certain periodicity is known, nor is it clear how to obtain the prefactors for the linear combination.

The periodicity of the trajectories is demanded because the derivation of the reaction rate constant uses the periodicity to eliminate a *limes*. Possibly replacing the Floquet exponents by Liapunov exponents and integrating over very long times could solve this

problem. In this case, the periodicity requirement could be dropped. Before that, an analytic validation of this idea is necessary.

3.4 Algorithmic Optimizations

This chapter describes some of the algorithmic problems, which occurred during this thesis and explains how they can be tackled. It starts with the previously existing algorithm of propagating particles and comparing their position with the DS [24–33] in order to obtain the crossing times. After that, there are sections about the choice of hyperparameters and the optimizer as well as the result of introducing a regularization. All of the optimizations of neural networks are done for nets with one output, corresponding to the approximation of the position of the DS and instead of the NHIM. This is for the following reason: As discussed in subsection 2.2.2, a stabilized integrator is possible without a NN by only using the BC. Therefore it is not sure if the NHIM-NN will be necessary in future. In contrast to that, it is more unlikely that the ensemble method will in future work without a NN that approximates the DS.

3.4.1 Ensemble Rate Calculation

In this thesis the calculation of the reaction rate via the *ensemble method* was used as a reference to compare the results of the Floquet method with. The ensemble method works in the following way:

- Define an ensemble, whose reaction rate should be calculated.
- Propagate the ensemble with an arbitrary numerical integrator, which is explicitly not our aforementioned modified integrator, until the reaction is finished.
- Compare the particle trajectories with the position of the DS in order to obtain the crossing times.
- Plot the decay in the ensemble population over time to get the rate constant for this specific ensemble.

The reason why this method is mentioned in more detail here is that at the beginning it took about eight hours of computational time to get the reaction rate for a single ensemble with 1×10^6 particles. Furthermore, some of the late reacting particles (which are the most important ones) are crossing the DS more than once. This unphysical

behaviour is called *recrossings* and excepts these particles from the crossing time statistics. More precisely, a subroutine removes all recrossing particles from the ensemble after the propagation has finished. Since several of these ensemble calculations were needed, some effort was put in optimizations of this algorithm.

Computational Time

Empirical tests have shown that about 99% of the computational time is spend for inferring, i.e. to evaluate the NN (e.g. 8 sec for particle propagation and 600 sec for NN evaluation). Obviously this problem can be tackled in general in two different ways. Either reduce the evaluation rate, or reduce the time for a single evaluation of the NN.

At the beginning of this thesis, inferring happened after every integrator step. This divided the whole integration time in 2×10^4 equidistant intervals. The result is a very high precision in time but a very low precision in particle numbers of late reacting particles. This is because of the long computational time per particle and therefore small ensemble size. A naive approach is a reduction in the crossing time resolution by a factor of 10 to 1×10^{-2} , which allows to increase the ensemble size by the same factor of 10 without significantly increasing the computational time.

A more sophisticated way to tackle the problem of too many NN evaluations per trajectory is the implementation of a root-finding algorithm for the crossing time. As it is usual for root search, it scales logarithmic in the number of function evaluations and has therefore a logarithmic run-time. The main problem with this is that graphics cards handle badly with some types of *if-conditions*. This is related to the hardware structure of the graphics card. The used *graphics processing units (GPUs)* are clustered in *blocks*, which are anon clustered in *hardware thread (HT)* or processing unit. The reason why *if-conditions* can cause problems on *GPUs* is that every HT of the same block has to execute the same code as the other HTs do, or idle. In case of *if-clauses* every HT evaluates the if-condition for its data. After that, the HTs for which the condition was *true* execute the *then-instruction* on their data while the other HTs do nothing. After that the other HTs execute the *else-instruction* on their data while the others idle. For two different branches, this does not seem to be a problem. However, if it comes to *for-* or *while-loops* the number of branches can diverge. In the worst case, every HT has its own branch, which means that there is no parallelization at all.

A regular bisection search in the trajectory data does not show this *branch-divergence*. However, it requires to record the full trajectories for all in parallel treated particles. Since this would exceed the size of the fast GPU memory, the writing would slow down

the algorithm. Therefore a combination of *linear search* and *bisection search* is used. The linear search is used for the coarse determination of the root and bisection allows for the precise root-finding in the short part of the trajectory, where the crossing happens.

This combined algorithm allows a reduction of the computational time by about 80% from 6 min to about 1 min and 20-30 calls of the NN instead of 2×10^3 . The major disadvantage is that it is no longer possible to find out the number of recrossings. This is due to the fact that the regular bisection search can not find multiple roots of a polynomial, even if it is a continuous function. The number of recrossings is used as a measure for the quality of the NN. Therefore the previous algorithm is used on a small ensemble to determine the magnitude of recrossings for a particular NN. For a sufficiently low number of recrossings in this test, the calculation with the large ensemble follows.

It should be mentioned that something like branch divergence still appears, but in a weak form. This is due to the fact that for early reacting particles the propagation stops early, after the reacted particle is far enough behind the barrier. But if one non-reacting particle is calculated in parallel with many early reacting particles, then the former is blocking the GPU for all particles. Therefore a future improvement could be to sort the ensemble by reacting and non-reacting particles.

Recrossings

It was mentioned before that during the numerical propagation of the particles unphysical recrossings through the DS can happen. Recrossings are in every case a numerical problem, since the DS is by definition recrossing free. A recrossing means that the particle is crossing our approximation of the DS more than once, i. e. it gets for example classified in the following sequence: reactant, product, reactant, product. According to the algorithm, this sequence has three points in time when crossings happen, while in fact only one point in time can be defined as the *reaction time point*.

If mapping the number of not reacted particles over time, then recrossings can appear as a non-monotonic decay. In case of early recrossing particles, non-monotonic decays get mostly compensated by regular reacting particles, but for late reacting particles there are not enough particles to compensate the recrossings. Therefore only the latter causes a non-monotonic decay. Since the reaction rate constant follows directly from this decay, it is important to reduce this type of recrossings.

In principle, there are three different sources where this numerical problem could emerge:

- Numerical integrator during propagation
- Training data for the DS from the binary contraction
- Approximation of the DS by the NN

If the integrator step size is chosen below a certain value (between 1×10^{-2} and 1×10^{-3}), then the number of recrossings is independent of the integrator step size. Furthermore, the binary contraction allows for the determination of the DS up to *machine precision* and is usually done up to 1×10^{-7} . Since the NN reaches an accuracy of only 1×10^{-4} , it is concluded that this is by far the most important source of recrossings.

The first approach to reduce the number of recrossings was to increase the number of parameters in our NN. It is intuitive that the nonlinear activation function — tanh in our case — is the most time consuming part of the NN evaluation and empirical experiments confirmed this assumption. This implicates that the ideal NN needs only a small number of activation function evaluations. This is possible by either using short but wide NNs or long but narrow ones. Furthermore, it is intuitive that longer nets allow to approximate more complexly curved surfaces than shorter nets, since they have more consecutive nonlinear activation functions. Therefore the design was changed from (3, 40, 10, 1) to (3, 40, 40, 40, 10, 1), i. e. two additional layers with tanh as activation function were added. The number of activation functions calculated per NN evaluation rises from 50 to 130 and hence the calculation time increases by a factor of 2.6. This reduced the number of recrossing in an ensemble of 1×10^6 particles by about 50%. Actually, the more important fact is that only 5 particles recross outside the ranges $Tn - \epsilon < t < Tn + \epsilon$ with $n \in \mathbb{N}$ and T corresponding to the barrier period.

The reason for this is that the NNs get trained in the time period $-0.25T < t < 1.25T$ and the evaluation only happens for $t \bmod T$, i. e. in the interval $0.0 < t < T$. Therefore, even very smooth NNs do not necessarily provide a smooth approximation of the DS at $t = T$. This is despite the fact that we use an overlap for the training data of about 50% of the time interval. Previous works have shown that this reduces this problem. Concretely, with the longer NN it has been shown, that about 99% of the recrossings appear around $t = T$ and $t = 2T$. An ensemble with 1×10^7 particles shows about 1000 recrossings but only 8 are not at these particular times. Since the DS is a three-dimensional object in a four-dimensional space, a simple interpolation between the crossing times does not describe the physical reality correctly.

To tackle this problem, three different possible solutions were implemented:

1. Introduce a *cost factor* that gets minimized, if the NN has the same value for $t = 0$ and $t = 2$.
2. Increasing the *overlap* in time of the training data, e. g. to $-0.5T < t < 1.5T$.
3. Use training data in the interval $0 < t < 3T$, because none of the particles reacted later than this. It also means that *no modulo* operation is used in the important part of the time interval.

The first approach decreased the number of recrossings by a factor of 2 to 9, but also increased the number of recrossings at all other time points disproportionately. It was tried to compensate this by extending the training time by a factor of 10. In the end NNs with the additional cost factor never reached less than 100 recrossings in an ensemble of 1×10^7 particles.

Empirical tests for the second approach show that increasing the overlap as mentioned above could decrease the number of recrossings by about 30% (700 recrossings in 1×10^7 particles). Our next step in increasing the overlap would have been $-1T < t < 2T$, which was not tried. The reason is that in the third approach, the NN has to learn an identical interval length, but with the advantage that it is by construction impossible to have the periodicity problem; since the periodicity is not used. Consequently the last approach shows no recrossings at the time points $t = nT$ and has overall 8 recrossings in 1×10^7 particles. This is by far the best result. As a consequence the training time of the NN is increased by a factor of two, because the trained time interval is twice as long, it gets sampled with the same grid and the number of epochs stays constant at 5×10^4 . Empirical tests show that this number of recrossings varies for different potential parameters. However, every tested system showed less than 100 recrossings in 1×10^7 particles.

In short, by adding two layers, doubling the trained time interval and omitting the modulo operation in the important time frame the number of recrossings get reduced from 2000 to 8 in an ensemble of 1×10^7 particles.

3.4.2 Hyper-Parameter Dependency

The next parts of this chapter discuss different optimizations of the neural nets, which were implemented to reduce the learning time, increase the prediction accuracy or increase the probability of training a neural network successfully.

In the following, all figures do not show the square loss, but the root of the square loss. The root of the square loss corresponds directly to the phase space distance between the position of the NHIM predicted by the NN and the actual position according to the training and test data.

Batch size

In the previous work [66], the absolute batch size n_B was always chosen as $n_B = 1$. This means that after every propagation of a training point, the gradient was calculated and every weight was adjusted. A better method is to calculate the weight changes for every point but do not apply them, until a certain number of training points. After that all weight adjustments get added up and applied at once. This is for two reasons a better strategy:

First, the weight adjustments have a distinct direction with random noise for every single point. This is because every data point requires its own specific adjustments in the extreme high-dimensional space of matrix and bias weights. Using a batch leads to an averaging effect over this noise. Therefore the training is more stable.

Second, using batches for training is faster since the cost intensive writing process is rarer applied. A disadvantage of too large batch sizes is that for two very similar training points, no progress happens between them, since the weight changes are calculated but not applied. Therefore the training progress per epoch decreases, which harms the computational efficiency. So a trade-off between the two effects, influencing the computational time, has to be found.

In the previous work [66], the training set size was $n_T = 2000$ and the batch size n_B was $n_B = 1$. Therefore the relative batch size p_B was

$$p_B = \frac{n_B}{n_T} = \frac{1}{2000} = 0.5\% \quad (3.3)$$

The search for an optimal batch size is shown in Fig. 3.11. The relative batch size p_B is measured relative to the size of the training data. For this figure, the training set has 177 000 data points. For $p_B = 0.01\%$ this leads for example to 18 data points. Increasing the relative batch size by a factor of 10 reduces the training time roughly by a factor of 9, due to the reduction in writing processes. Since the number of epochs is shown on the horizontal axis of this figure, the red curve is faster in computational time than the blue one, even if it takes some more epochs. For the black curve, both effects roughly neglect each other. Obviously it would be possible to investigate this behaviour in more detail,

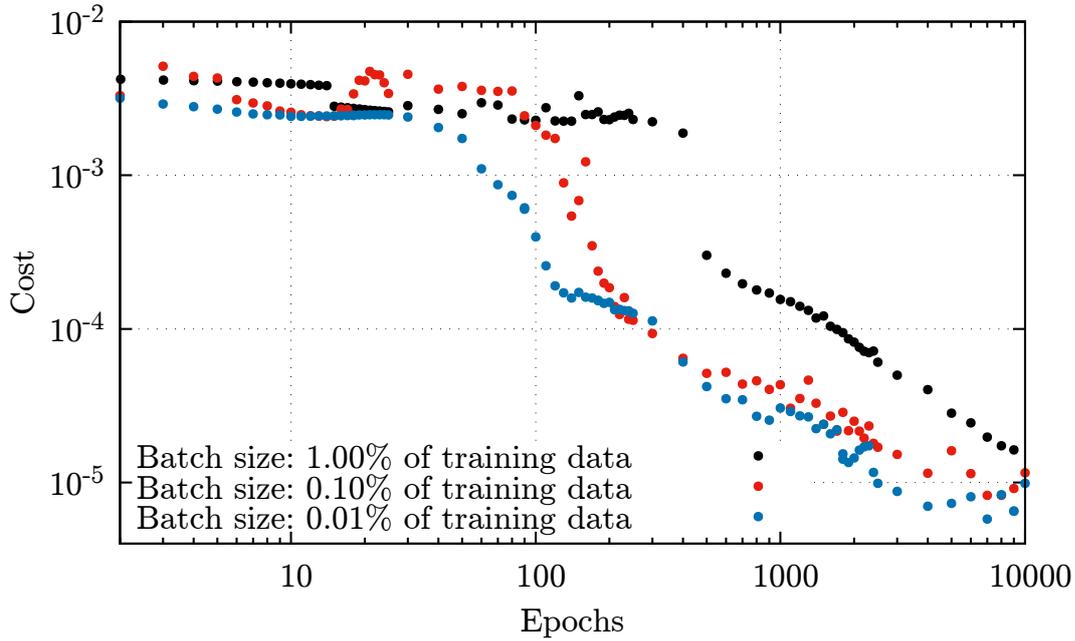


Figure 3.11: Learning progress for different batch sizes relative to the full training set. The optimizer used is stochastic gradient descent (SGD), the learning rate $\eta = 0.1$ and the NN has the shape (3,40,10,1).

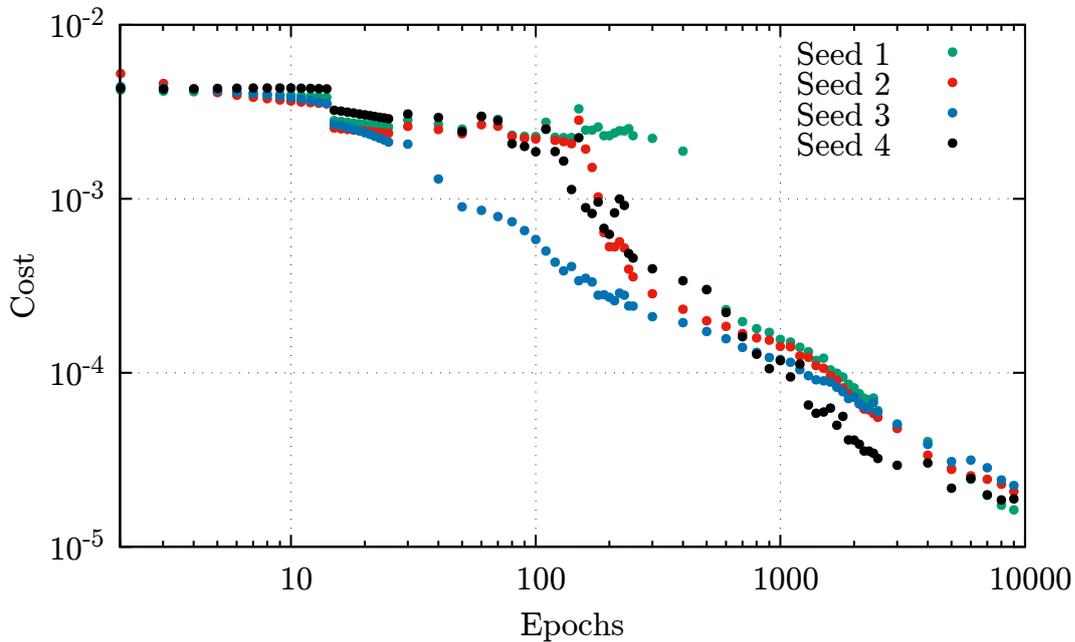


Figure 3.12: Learning progress for different seeds. The optimizer used is stochastic gradient descent (SGD), the learning rate $\eta = 0.1$, the relative batch size $p_B = 1\%$ and the NN has the shape (3,40,10,1).

to find the optimal relative batch size. However, this was not done because of the low promised benefit. Furthermore it can be seen in Fig. 3.11 that the black curve shows less noise, than the other two. Less noise implies that training is more stable, e. g. depends less on random factors like the weight initialisation seed. This is for us more important than a slightly faster convergence of the training. In every further investigation, the relative batch size was set to $p_B = 1\%$.

Seed Dependency

In this section the influence of different initialisation seeds is analysed. This addresses the question: How reproducible are the results of the NNs. Training can be understood as an optimization problem in a very high-dimensional potential surface, which has the weights of the matrices and bias vectors as axis. For different seeds the training (and therefore the weight optimization) starts at different spots on this potential surface.

A comparison for four different seeds is shown in Fig. 3.12. It can be seen that the four cost functions are very similar until the 30th epoch and after the 600th epoch. Even if the exact seed dependency of the learning progress is not important, it is a good measure for the stability of the optimizer and can be crucial for termination criterion. Fig. 3.12 shows that the optimizer SGDs is not really stable. Considering the 1st seed, it can be seen that nearly no progress happens until the 400th epoch, where the cost function drops by a factor of 10. Furthermore, it should be noticed that the 1st seed is the worst at the beginning but the best at the end. This led to experiments with different optimizers. Nevertheless, all four seeds converge to roughly the same cost values, if training is stopped after a sufficiently large number of epochs.

Learning Rate

The previous paragraph has shown that the learning progress depends on random factors like the selected seed, but converges in general to the same final cost. This paragraph investigates how the learning progress depends on the learning rate and if these dependence is stronger than the one of the seed.

In Fig. 3.13, it can be seen that increasing the learning rate η by a factor of 10 leads to a 10 times faster learning progress. This is logical, since the learning rate directly influences the step size of the optimizer. According to Fig. 3.13 $\eta = 0.1$ is the best choice. Furthermore, one would expect that a higher learning rate would increase the

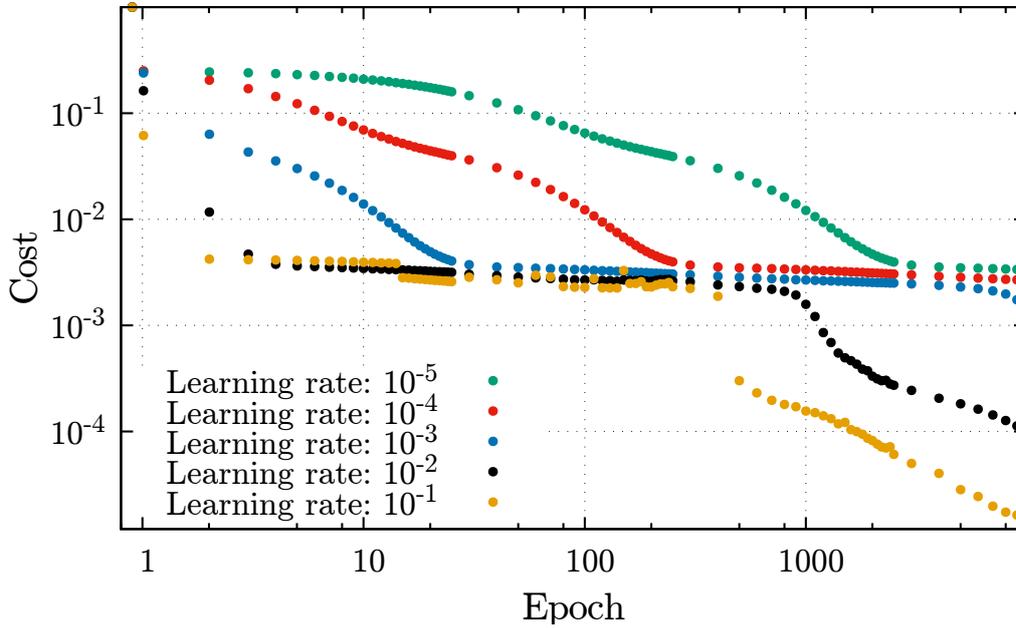


Figure 3.13: Learning progress for different learning rates. The optimizer used is stochastic gradient decent (SGD), the relative batch size $p_B = 1\%$ and the NN has the shape (3,40,10,1).

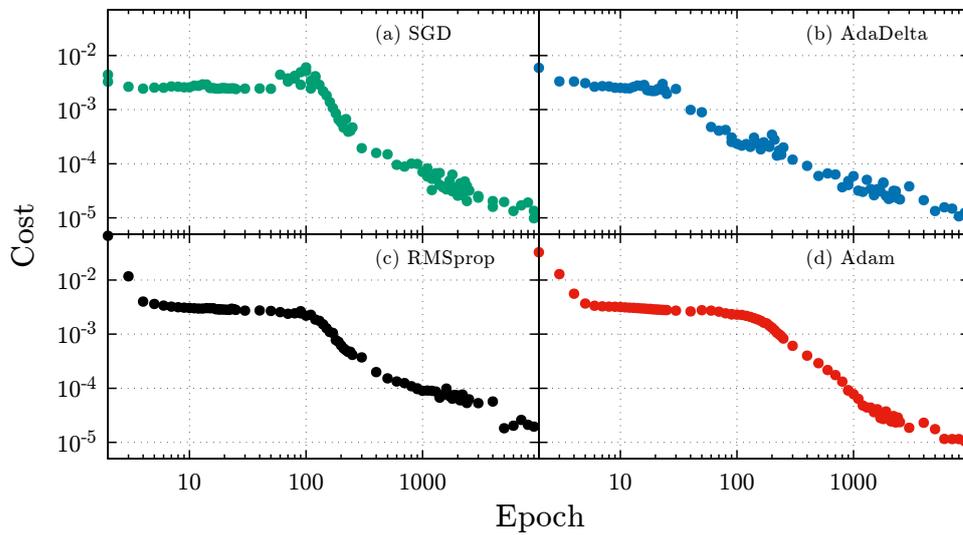


Figure 3.14: Comparison of the learning progress for four different optimizer. The relative batch size is $p_B = 1\%$ and the NN has the shape (3,40,10,1).

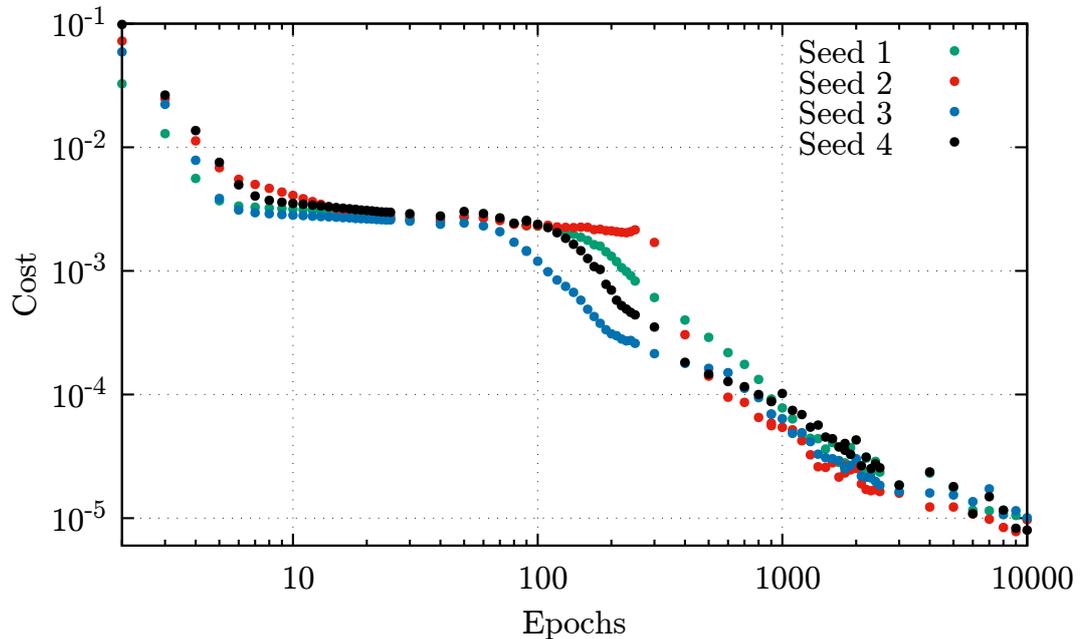


Figure 3.15: Learning progress for different seeds. The optimizer used is Adam, the learning rate $\eta = 0.1$, the relative batch size $p_B = 1\%$ and the NN has the shape (3,40,10,1).

learning progress. However for $\eta = 1$ the training terminates due to divergence of the cost function. This shows the instability of the optimizer for this hyper-parameter.

3.4.3 Optimizer Comparison

It has been seen that the training involuntarily depends on random factors like the initialisation seed and on hyper-parameters, e.g. the learning rate. This paragraph studies, if the training behaviour can be improved by other optimization methods. First of all, we have to clarify which requirements we impose on the optimizer. Essentially the following three abilities are important for us:

1. The learning progress should not depend on the initialisation seeds, so that the NN quality is independent of random factors.
2. The learning progress curve should be as smooth and less noisy as possible, to prevent training from terminating at the wrong time.

3. The resulting NN should not be sensitive in the selected hyper-parameter, such as the learning rate.

The first and the third point are already discussed in section 3.4.2. The reason why smoothness is an important attribute for the learning progress is that it makes the heuristic for the training termination easier.

In the following, the four different optimizers *SGD* [67], *AdaDelta* [68], *RMSprop* [69], and *Adam* [70] get studied. All of them are based on stochastic gradient decent, but adding a momentum with pre-factors and or step size control with pre-factors. For theoretical details, we refer to the aforementioned secondary literature.

Fig. 3.14 compares the optimizers for identical training data with identical relative batch sizes and the same shape of the NNs. As expected from section 3.4.2, SGD shows strong noise and a harsh curve including local maxima. AdaDelta has a bit less noise, no local maxima and is in this example the fastest of the four algorithms. Even less noise can be seen for RMSprop, which has also a very smooth learning progress. In Fig. 3.14 there is no big difference between the Adam optimizer and RMSprop. As a short summary, the second requirement gets best fulfilled by Adam and RMSprop.

In a comparison of five different learning rates with four different seeds each and three different relative batch sizes neither Adam nor RMSprop failed in any of the parameter sets, i. e. producing a *NaN*. Also AdaDelta had no diverging loss, but ten systems where practically no progress happened. SGD failed in 12 cases, resulting in *NaNs*. In the seed dependency of the optimizer, there is no significant difference between Adam and RMSprop. Thus, Adam was arbitrarily chosen for the further studies. Fig. 3.15 shows that with Adam the learning progress depends on the specific seed too. However, the curves are slightly more close to each other, than with SGD.

3.4.4 Regularisations

The sections 3.4.2 and 3.4.3 discuss results for a (3,40,10,1)-net with one output neuron, to approximate the DS. In this section, it gets focused on a problem that emerged for nets with the shape (3,100,100,30,2), which is used for the NHIM approximation. As shown in Fig. 3.16, the curve without regularisation (green circles) is noisy and has outliers, e. g. at epoch number 12000. This is due to the fact that Adam is used as optimizer. To prevent this behaviour, a regularisation term was added to the cost function.

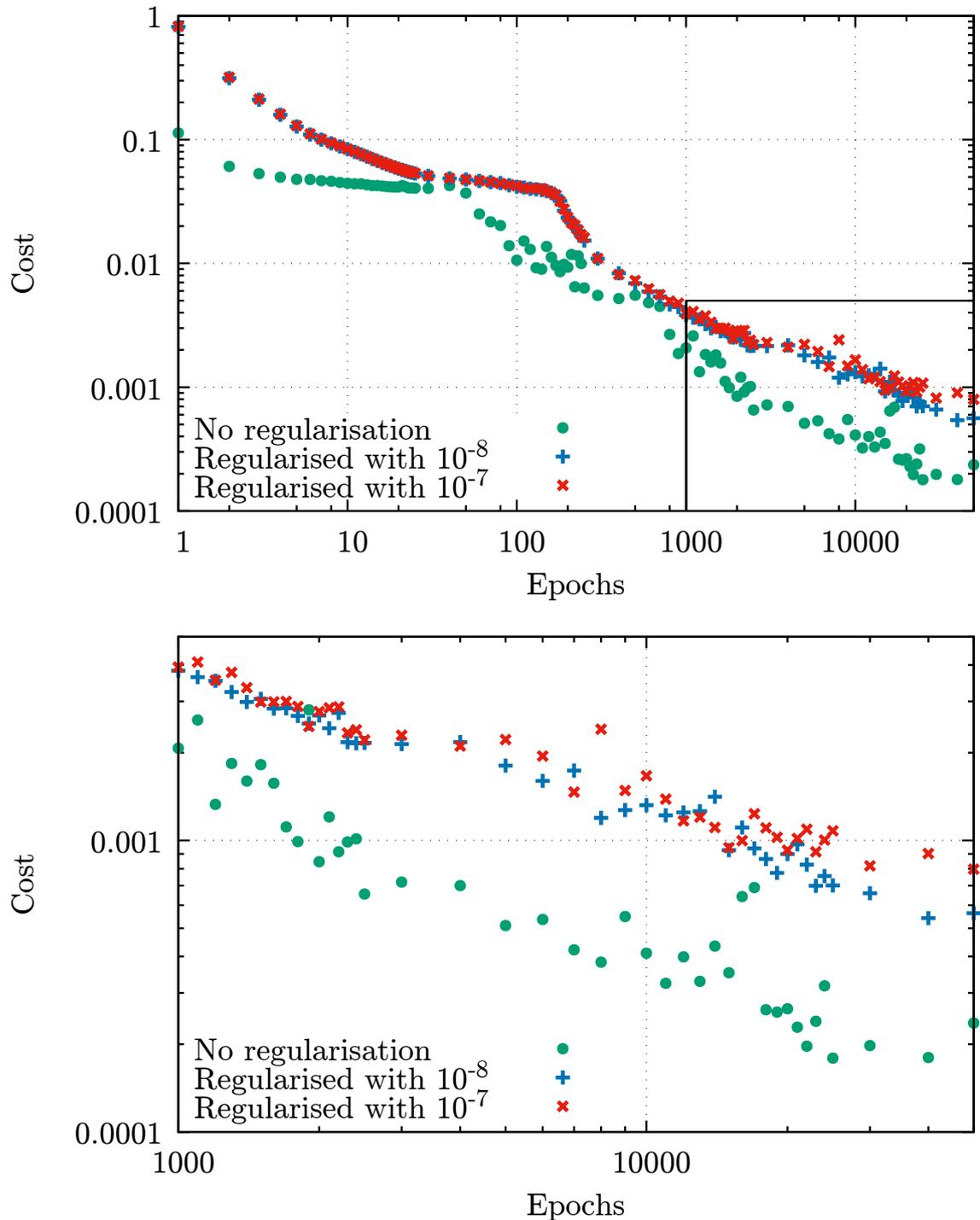


Figure 3.16: Comparison of the learning progress without regularisation, and with two different regularisation pre-factors 1×10^{-7} and 1×10^{-8} . The optimizer used is Adam, the relative batch size is $p_B = 1\%$ and the NN has the shape (3,100,100,30,2) to approximate the NHIM. **Top:** Whole training over 50 000 epochs. **Bottom:** Detailed view without the first 1000 epochs.

The idea behind regularisation is often to penalise overfitting by an additional term in the cost function. However it can also be used to reduce noisy training behaviour. In the further it is concentrated on the L_2 -regularisation, even if there are several different ways to introduce a regularisation. The L_2 -regularisation is implemented by simply adding the L_2 -norm of all weights with a pre-factor to the cost term. In Fig. 3.16, it can be seen that the cost function for the first 2500 epochs is smoothed for both pre-factors. Furthermore the cost function is significantly higher, if regularisation is used. A more detailed view in the lower graphic of Fig. 3.16 illustrates that a larger regularisation factor leads to higher costs. In detail, without regularisation the minimal cost is about 2×10^{-4} , compared to 6×10^{-4} for the small pre-factor and 8×10^{-4} with the larger pre-factor. Since the smoothing is weaker than expected, but the increase in cost is larger than expected, the regularisation was not used for the training of the NNs, used in section 3.3.

4 Conclusion and Outlook

In the following, a summary of the thesis' results is given. Afterwards possible topics for future works get discussed.

4.1 Conclusion

In this thesis it was shown that NNs can be used to implement a stabilized integrator whose trajectories stay on the NHIM for long integration times, despite an unstable DoF. Based on this, it was possible to study the phase space dynamics on one hand and determine a reaction rate constant on the other hand.

In terms of phase space dynamics it turned out that the model system shows regular behaviour on the NHIM for all analysed system parameters. A further result is that the system has a *conserved quantity*, which is for example implied by the torus structures in the stroboscopic views of the *Poincaré surface of sections*.

The *Floquet analysis* revealed, that in periodic trajectories of the 2-DoF-system the eigenvalues of the *monodromie matrix* appear in two pairs. In one pair, each eigenvalue has an absolute value of 1 and is complex conjugated to each other. In the other pair each eigenvalue is strictly real and both are inverse to each other.

In addition, two algorithms to determine the TSt were presented. One algorithm uses the fact that in conservative systems a reduction of the phase space volume that is included by a trajectory corresponds to a reduction of the energy of the trajectory. The other algorithm introduces an artificial friction into the system, to relax an arbitrary trajectory on the NHIM into the lowest energy periodic trajectory, that is still on the NHIM. It was shown that both algorithms converge against the same trajectory, called the TSt.

A comparison of the reaction rate constants obtained by the Floquet method with the rate constants of the *ensemble method* showed significant differences. The investigations have shown that the rate constant depend on that specific trajectory studied with the

Floquet analysis. So far this dependency seems to be relatively small, e. g. $k_f = 3.806$ instead of $k_f = 3.795$. However further research in this particular problem is needed. In addition to that, the reaction rate of the ensemble method strongly correlates with the specific ensemble used.

One of the main results of section 3.4 is, that a binary search of the reaction times in parallel on the GPU is possible. However, this algorithm does no longer determine the number of recrossings. Further results are the optimal *batch size* of 1% of the training data and that *Adam* as well as *RMSprop* are the most stable and least noisy optimizers.

4.2 Outlook

During this thesis by far more interesting questions emerged, than the limited amount of time allowed to tackle. Both, the phase space structures and the reaction rates were only researched for four different values of the one system parameter \hat{x} . However there are six more system parameters, whose influence on the torus structures and reaction rates should be determined. Even the parameter \hat{x} was only sampled for four different values which is not enough for a statistical reliable statement. In particular, for the phase space structures it would be interesting to find a parameter that decides about the chaoticity of the system. Furthermore, the behaviour in a larger area of phase space should be investigated.

For the reaction rate constant, it should be analysed what causes the discrepancy between the Floquet and the ensemble method. As mentioned in section 3.3.2, there are many ideas how this topic can be tackled. On the one hand, this requires a deeper understanding how the chosen ensemble influences the reaction rate in the ensemble method. On the other hand, it is necessary to determine reaction rates for other trajectories on the NHIM besides the TSt with the Floquet method.

However some major improvements in the algorithms are needed to investigate such a large variety of problems efficiently. Fig. 4.1 presents possible improvements for both the Floquet and the ensemble method.

The Floquet method presented in this thesis has an important disadvantage in comparison to the method in lower path in Fig. 4.1: Every set of system parameters requires to generate new training data with the *binary contraction method* and the time consuming training of a neural network. However in principle it would be possible to implement a stabilized integrator without a NN, but by directly using the binary contraction to

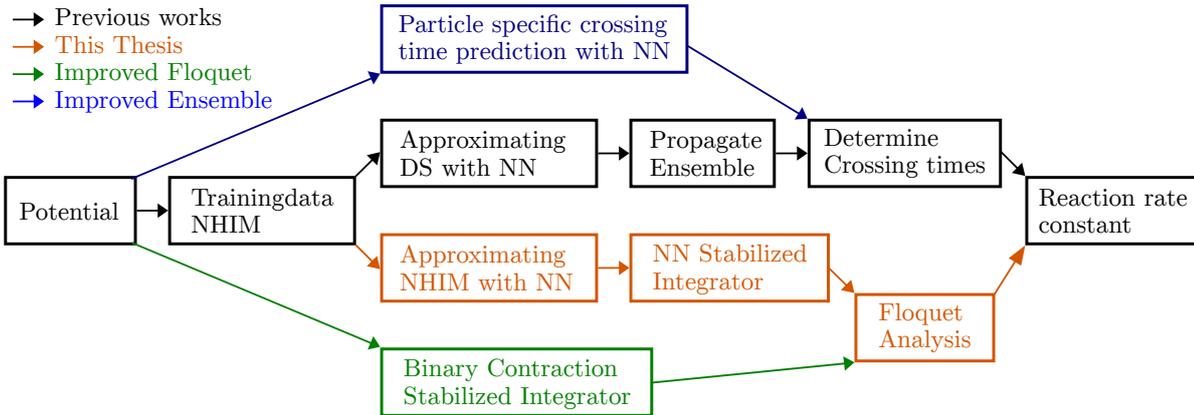


Figure 4.1: Schematic of the already existing ensemble algorithm (upper middle), the Floquet method introduced in this thesis (lower middle) that uses a NN to approximate the NHIM, as well as the proposed Floquet method, that only uses binary contraction instead of a NN (bottom) and the proposed improved ensemble method (top), that uses a NN to directly predicts the crossing times.

project the particle back in the NHIM. In Fig. 4.1 this is called *Improved Floquet*. Binary contraction has a logarithmic runtime in the determination accuracy δ of the NHIM. Therefore it is only logical to set δ to machine precision and apply binary contraction as rare as possible between the integrator steps. This approach promises to reduce the calculation time for a Floquet rate to less than 1 min for an arbitrary system.

Since this modification directly affects the stabilized integrator, it can also be used for the phase space dynamics studies. A further advantage of the method without NN is that it ensures that occurring structures are no artefacts of the NN. Last but not least, the NN are only trained for one period of time plus a small margin and get evaluated with the modulo. This results in a small gap in the trajectory at the end of every period. This will no longer happen by only using binary contraction for the projection.

Since so far, the ensemble method is by far more time consuming, it is more promising to introduce an alternative to that. As shown in Fig. 4.1 the improved version of this method bypasses not only the whole training data generation and the NN training, but also ensemble propagation itself. The idea is to try a neural network that gets an encoding of the potential and phase space coordinates for a specific particle position. Based on this inputs, it predicts the crossing time of this particle. The generalisation respectively approximation happens in two different ways: First, the NN should be able predict the crossing times for particle positions, which were not part of the training data,

i. e. spatial interpolation. Second, the NN should be able to predict crossing times for system parameter, that were not explicitly trained, i. e. system parameter interpolation. For example, if the NN is trained for $\hat{x} \in \{0.2, 0.4, 0.6, 0.8\}$ it could predict the behaviour of a system with $\hat{x} = 0.5$.

Deutsche Zusammenfassung

Diese Arbeit führt, am Beispiel der chemischen Reaktionsdynamik, eine Modifikation für numerische Integratoren ein. Diese ermöglicht es Trajektorien über lange Zeiten in einem instabilen aber invarianten Unterraum des Phasenraumes zu bestimmen. Die wesentlichen Aspekte beim betrachteten Modellsystem sind dabei die *Zeitabhängigkeit* und der enthaltene *Rank-1-Sattel*, welcher sicherstellt, dass es mindestens einen instabilen Freiheitsgrad gibt. Die besondere Bedeutung der Zeitabhängigkeit liegt zum Einen darin, dass einige natürlich ablaufende chemische Reaktionen beispielsweise Wechselfeldern unterliegen, welche das System periodisch treiben. Zum Anderen besteht die Möglichkeit, dass chemische Reaktionen in der technischen Anwendung durch externe Felder beschleunigt werden könnten.

Die Modifikation des numerischen Integrators besteht darin, nach jedem Integratorschritt eine Projektion in den instabilen, invarianten Unterraum, der *normally hyperbolic invariant manifold (NHIM)* genannt wird, durchzuführen. Die NHIM wird dabei durch ein künstliches *Neuronales Netz* angenähert. Es werden zwei Anwendungen für diesen stabilisierten Integrator in dieser Arbeit vorgestellt. Zum Einen ermöglicht der Integrator die Untersuchung der Phasenraumdynamik für lange Zeiten. Zum Anderen kann eine Floquet-Analyse verwendet werden, um eine chemische Reaktionsrate für das System zu bestimmen.

Im Kapitel 2 wird die Grundlagen für das Verständnis der Phasenraumstrukturen gelegt, die Floquet-Analyse vorgestellt und eine Einführung in Neuronale Netze gegeben. Bei letzterem wird zwischen den allgemeinen Grundlagen zum Aufbau und der Funktionsweise von Neuronalen Netzen und den Design-Details des, in dieser Arbeit verwendeten Netzes unterschieden.

Kapitel 3 beschreibt detailliert die Resultate für die zwei zuvor erwähnten Anwendungen des stabilisierten Integrators. In Bezug auf die Phasenraumdynamik zeigt sich, dass das Modellsystem ein reguläres Verhalten für alle analysierten Systemparameter aufweist. Ein weiteres Ergebnis besteht darin, dass das betrachtete System *fast integrabel* ist. Dies

wird beispielsweise durch Torusstrukturen in den stroboskopischen *Poincaré-Abbildung* impliziert.

Die Floquet-Analyse zeigt, dass in periodischen Bahnen von Systemen mit zwei Freiheitsgraden Eigenwerte der *Monodromie-Matrix* in zwei Paaren auftreten. Eines der Paare zeichnet sich dadurch aus, dass die Eigenwerte jeweils einen absoluten Wert von 1 besitzen und durch komplexe Konjugation ineinander übergehen. Die Eigenwerte des zweiten Paares sind rein reell und invers zueinander.

Außerdem werden in der Arbeit zwei Algorithmen zur Bestimmung der *transition state trajectory* (*TSt*) vorgestellt. Für ein periodisch getriebenes System entspricht die TSt der periodische Bahn, deren mittlere Energie minimal ist und deren Periodendauer mit der des Systems übereinstimmt. Der als erstes vorgestellte Algorithmus verwendet die Tatsache, dass in konservativen Systemen das Phasenraumvolumen, welches von einer Trajektorie eingeschlossen wird im Zusammenhang mit der Energie der Trajektorie steht. Aus diesem Grund wird beim ersten Algorithmus iterativ das Phasenraumvolumen reduziert. Der zweite Algorithmus führt eine künstliche Reibung in das System ein, um von einem beliebigen Punkt auf der NHIM in einen Orbit zu konvergieren, dessen Energie minimal ist. Es zeigt sich, dass beide Algorithmen auf den gleichen Punkt im Phasenraum konvergieren. Die Trajektorie, die sich aus diesem Punkt ergibt, wurde für die Untersuchung mittels Floquet-Analyse verwendet.

Anschließend erfolgt ein Vergleich, zwischen der Reaktionsrate, die durch Ensemblepropagation bestimmt wurde (z.B. $k_e \approx 3,1$) und der Rate, die sich mit Hilfe der Floquet-Analyse ergeben hat (z.B. $k_f = 3,806$). Es zeigen sich signifikante Unterschiede zwischen den beiden Werten. In einem Beispielfall wurde untersucht inwiefern die Floquet-Rate von der verwendeten Trajektorie abhängt. Für die TSt ergibt sich beispielsweise eine Rate von $k_f = 3,806$, während man für die andere Bahn eine Reaktionsrate von $k_f = 3,795$ erhält. Dieser Unterschied ist relativ klein im Vergleich zur Abweichung zwischen Ensemble- und Floquet-Methode. Da bisher lediglich eine periodische Trajektorie untersucht wurde, die keine TSt darstellt, kann daraus noch keine allgemeine Aussage getroffen werden. Darüber hinaus zeigt sich bei der Reaktionsrate, die mittels Ensemblepropagation bestimmt wurde, eine starke Abhängigkeit vom verwendeten Ensemble.

Der Abschnitt 3.4 beschäftigt sich einerseits mit Optimierungen bei der Ensemblepropagation und andererseits mit der Untersuchung möglicher Verbesserungen am Neuronalen Netz. Im ersten Teil hat sich gezeigt, dass eine binäre Suche der Reaktionszeitpunkte mit Hilfe von Grafikkarten parallelisiert werden kann. Dies ermöglicht implizit ein selteneres Aufrufen der Methode, die die Trennfläche zwischen Reaktanten und Produkten approximiert. Dadurch lässt sich die Laufzeit der Ensemblepropagation auf dem verwendeten

Computer um 99 % reduzieren. Dieser Algorithmus ist jedoch nicht mehr in der Lage zu bestimmen, ob die approximierte Trennfläche zwischen Reaktanten und Produkten mehrfach von einem Teilchen durchquert wird. Bezüglich des Trainings der Neuronalen Netze hat sich gezeigt, dass die optimale *Batchsize* etwa 1 % der Trainingsdaten entspricht. Von den vier getesteten Optimierungsverfahren sind *Adam* und *RMSprop* am wenigsten abhängig von Zufallsfaktoren, wie dem verwendeten Initialisierungsseed.

Bibliography

1. Pollak, E. & Pechukas, P. Transition States, Trapped Trajectories, and Classical Bound States Embedded in the Continuum. *J. Comput. Phys.* **69**, 1218–1226 (1978).
2. Pechukas, P. & Pollak, E. Classical Transition State Theory is Exact if the Transition State is Unique. *J. Comput. Phys.* **71**, 2062–2068 (1979).
3. Uzer, T., Jaffé, C., Palacián, J., Yanguas, P. & Wiggins, S. The Geometry of Reaction Dynamics. *Nonlinearity* **15**, 957–992 (2002).
4. Jaffé, C. *et al.* Statistical Theory of Asteroid Escape Rates. *Phys. Rev. Lett.* **89**, 011101 (2002).
5. Teramoto, H., Toda, M. & Komatsuzaki, T. Dynamical Switching of a Reaction Coordinate to Carry the System Through to a Different Product State at High Energies. *Phys. Rev. Lett.* **106**, 054101 (2011).
6. Li, C.-B., Shoujiguchi, A., Toda, M. & Komatsuzaki, T. Definability of No-Return Transition States in the High-Energy Regime above the Reaction Threshold. *Phys. Rev. Lett.* **97**, 028302 (2006).
7. Waalkens, H. & Wiggins, S. Direct Construction of a Dividing Surface of Minimal Flux for Multi-Degree-of-Freedom Systems that Cannot be Recrossed. *J. Phys. A* **37**, L435–L445 (2004).
8. Çiftçi, Ü. & Waalkens, H. Reaction Dynamics Through Kinetic Transition States. *Phys. Rev. Lett.* **110**, 233201 (2013).
9. Jaffé, C., Farrelly, D. & Uzer, T. Transition State Theory without Time-Reversal Symmetry: Chaotic Ionization of the Hydrogen Atom. *Phys. Rev. Lett.* **84**, 610–614 (2000).
10. Jacucci, G., Toller, M., DeLorenzi, G. & Flynn, C. P. Rate Theory, Return Jump Catastrophes, and the Center Manifold. *Phys. Rev. Lett.* **52**, 295 (1984).
11. Komatsuzaki, T. & Berry, R. S. Regularity in chaotic reaction paths. I. Ar₆. *J. Comput. Phys.* **110**, 9160–9173 (1999).

12. Komatsuzaki, T. & Berry, R. S. Chemical Reaction Dynamics: Many-Body Chaos and Regularity. *Adv. Chem. Phys.* **123**, 79–152 (2002).
13. Toller, M., Jacucci, G., DeLorenzi, G. & Flynn, C. P. Theory of classical diffusion jumps in solids. *Phys. Rev. B* **32**, 2082 (1985).
14. Voter, A. F., Montalenti, F. & Germann, T. C. Extending the time scale in atomistic simulations of materials. *Annu. Rev. Mater. Res.* **32**, 321–346 (2002).
15. De Oliveira, H. P., Ozorio de Almeida, A. M., Damião Soares, I. & Tonini, E. V. Homoclinic chaos in the dynamics of a general Bianchi type-IX model. *Phys. Rev. D* **65**, 083511/1–9 (Apr. 2002).
16. H. Waalkens, A. Burbanks & S. Wiggins. Escape from planetary neighborhoods. *Mon. Not. R. Astron. Soc.* **361**, 763 (2005).
17. Huepe, C., Métens, S., Dewel, G., Borckmans, P. & Brachet, M. E. Decay rates in attractive Bose-Einstein condensates. *Phys. Rev. Lett.* **82**, 1616 (1999).
18. Huepe, C., Tuckerman, L. S., Métens, S. & Brachet, M. E. Stability and decay rates of nonisotropic attractive Bose-Einstein condensates. *Phys. Rev. A* **68**, 023609 (2003).
19. A. Junginger, J. Main, G. Wunner & M. Dorwarth. Transition state theory for wave packet dynamics. I. Thermal decay in metastable Schrödinger systems. *J. Phys. A: Math. Theor.* **45**, 155201 (2012).
20. A. Junginger, M. Dorwarth, J. Main & G. Wunner. Transition state theory for wave packet dynamics. II. Thermal decay of Bose-Einstein condensates with long-range interaction. *J. Phys. A: Math. Theor.* **45**, 155202 (2012).
21. A. Junginger, M. Kreibich, J. Main & G. Wunner. Transition states and thermal collapse of dipolar Bose-Einstein condensates. *Phys. Rev. A* **88**, 043617 (2013).
22. R. Bardakcioglu. *Time-Dependent Transition State Theory to Determine Dividing Surfaces and Reaction Rates in Multidimensional Systems* Masterthesis (University of Stuttgart, 2018).
23. Krenos, John. *Chemical Kinetics and Reaction Dynamics* doi:10.1021/ed078p1466 (Journal of Chemical Education, Houston, 2001).
24. Bartsch, T., Hernandez, R. & Uzer, T. Transition state in a noisy environment. *Phys. Rev. Lett.* **95**, 058301 (2005).
25. Bartsch, T., Uzer, T. & Hernandez, R. Stochastic transition states: Reaction geometry amidst noise. *J. Comput. Phys.* **123**, 204102 (2005).

-
26. Bartsch, T., Uzer, T., Moix, J. M. & Hernandez, R. Identifying Reactive Trajectories Using a Moving Transition State. *J. Comput. Phys.* **124**, 244310 (2006).
 27. Kawai, S. & Komatsuzaki, T. Dynamic pathways to mediate reactions buried in thermal fluctuations. I. Time-dependent normal form theory for multidimensional Langevin equation. *J. Comput. Phys.* **131**, 224505 (2009).
 28. Craven, G. T., Bartsch, T. & Hernandez, R. Persistence of Transition State Structure in Chemical Reactions Driven by Fields Oscillating in Time. *Phys. Rev. E* **89**, 040801(R) (2014).
 29. Craven, G. T., Bartsch, T. & Hernandez, R. Communication: Transition State Trajectory Stability Determines Barrier Crossing Rates in Chemical Reactions Induced by Time-Dependent Oscillating Fields. *J. Comput. Phys.* **141**, 041106 (2014).
 30. Craven, G. T., Bartsch, T. & Hernandez, R. Chemical Reactions Induced by Oscillating External Fields in Weak Thermal Environments. *J. Comput. Phys.* **142**, 074108 (2015).
 31. Junginger, A. & Hernandez, R. Uncovering the Geometry of Barrierless Reactions Using Lagrangian Descriptors. *J. Phys. Chem. B* **120**, 1720 (2016).
 32. Junginger, A. *et al.* Transition state geometry of driven chemical reactions on time-dependent double-well potentials. *Phys. Chem. Chem. Phys.* **18**, 30270 (2016).
 33. Junginger, A. & Hernandez, R. Lagrangian Descriptors in Dissipative Systems. *Phys. Chem. Chem. Phys.* **18**, 30282 (2016).
 34. Kawai, S., Teramoto, H., Li, C.-B., Komatsuzaki, T. & Toda, M. in *Advancing Theory for Kinetics and Dynamics of Complex, Many-Dimensional Systems: Clusters and Proteins* 123–169 (Wiley-Blackwell, 2011). doi:10.1002/9781118087817.ch4.
 35. Wiggins, S. *Normally Hyperbolic Invariant Manifolds in Dynamical Systems* (Springer, New York, 1994).
 36. Pitzer, K. S., Smith, F. T. & Eyring, H. *The Transition State* 53 (Chemical Society, London, 1962).
 37. P. Pechukas. Transition state theory. *Ann. Rev. Phys. Chem.* **32**, 159 (1981).
 38. B. C. Garrett & D. G. Truhlar. Generalized Transition State Theory. *J. Phys. Chem.* **83**, 1052–1079 (1979).
 39. D. G. Truhlar, A. D. Issacson & B. C. Garrett. in (ed M. Baer) 65–137 (CRC Press, Boca Raton, FL, 1985).

40. Hynes, J. T. Chemical reaction dynamics in solution. *Annu. Rev. Phys. Chem.* **36**, 573–597 (1985).
41. Berne, B. J., Borkovec, M. & Straub, J. E. Classical and modern methods in reaction rate theory. *J. Phys. Chem.* **92**, 3711–3725 (1988).
42. Nitzan, A. Activated rate processes in condensed phases: The Kramers theory revisited. *Adv. Chem. Phys.* **70**, 489–555 (1988).
43. Hänggi, P., Talkner, P. & Borkovec, M. Reaction-rate theory: Fifty years after Kramers. *Rev. Mod. Phys.* **62**, and references therein, 251–341 (1990).
44. Natanson, G. A., Garrett, B. C., Truong, T. N., Joseph, T. & Truhlar, D. G. The Definition of Reaction Coordinates for Reaction-Path Dynamics. *J. Chem. Phys.* **94**, 7875–7892 (1991).
45. Truhlar, D. G., Garrett, B. C. & Klippenstein, S. J. Current Status of Transition-State Theory. *J. Phys. Chem.* **100**, 12771–12800 (1996).
46. Truhlar, D. G. & Garrett, B. C. Multidimensional Transition State Theory and the Validity of Grote-Hynes Theory. *J. Phys. Chem. B* **104**, 1069–1072 (2000).
47. Komatsuzaki, T. & Berry, R. S. Dynamical hierarchy in transition states: Why and how does a system climb over the mountain? *Proc. Natl. Acad. Sci. USA* **98**, 7666–7671 (2001).
48. Pollak, E. & Talkner, P. Reaction rate theory: What it was, where it is today, and where is it going? *Chaos* **15**, 026116 (2005).
49. Waalkens, H., Schubert, R. & Wiggins, S. Wigner’s dynamical transition state theory in phase space: classical and quantum. *Nonlinearity* **21**, R1 (2008).
50. Bartsch, T., Moix, J. M., Hernandez, R., Kawai, S. & Uzer, T. Time-dependent transition state theory. **140**, 191–238 (2008).
51. Kawai, S. & Komatsuzaki, T. Robust Existence of a Reaction Boundary to Separate the Fate of a Chemical Reaction. *Phys. Rev. Lett.* **105**, 048304 (4 2010).
52. Hernandez, R., Bartsch, T. & Uzer, T. Transition State Theory in Liquids Beyond Planar Dividing Surfaces. **370**, 270–276 (2010).
53. Sharia, O. & Henkelman, G. Analytic dynamical corrections to transition state theory. *New J. Phys.* **18**, 013023 (2016).
54. Goldstein, H. *Classical mechanics* ISBN: 9780201029697. <<https://books.google.de/books?id=I1JKjwEACAAJ>> (Addison-Wesley, 1980).
55. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* <http://www.deeplearningbook.org> (MIT Press, 2016).

-
56. Nielsen, M. A. *Neural Networks and Deep Learning* <http://neuralnetworksanddeeplearning.com/> (Determination Press, 2015).
 57. Feldmaier, M. *et al.* Invariant Manifolds and Rate Constants in Driven Chemical Reactions (in prep.).
 58. Bardakcioglu, R., Junginger, A., Feldmaier, M., Main, J. & Hernandez, R. Binary contraction method for the construction of time-dependent dividing surfaces in driven chemical reactions. *Phys. Rev. E* **98**, 032204 (2018).
 59. P. Schraft. *Neural Networks for the Approximation of Time-Dependent Dividing Surfaces in Transition State Theory* Masterthesis (University of Stuttgart, 2017).
 60. Van Rossum, G. *Python tutorial* tech. rep. CS-R9526 (Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995).
 61. Martin Abadi *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from [tensorflow.org](https://www.tensorflow.org/). <<https://www.tensorflow.org/>> (2015).
 62. Stroustrup, B. *A C++ Tutorial* tech. rep. (National Communications Forum, Sept. 1984).
 63. Feldmaier, M., Junginger, A., Main, J., Wunner, G. & Hernandez, R. Obtaining time-dependent multi-dimensional dividing surfaces using Lagrangian descriptors. *Chem. Phys. Lett.* **687**, 194 (2017).
 64. Kubo, R. The Fluctuation-Dissipation Theorem. *Rep. Prog. Phys.* **29**, 255–284 (1966).
 65. M. Lober. *Parameter Dependent Rates in Transition State Theory for Periodically Driven Systems* Bachelorthesis (University of Stuttgart, 2018).
 66. Schraft, P. *et al.* Neural network approach to time-dependent dividing surfaces in classical reaction dynamics. *Phys. Rev. E* **97**, 042309 (2018).
 67. Robbins, H. & Monro, S. A Stochastic Approximation Method. *The Annals of Mathematical Statistics* **22**, 400–407. ISSN: 00034851 (1951).
 68. Zeiler, M. D. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).
 69. Hinton, G., Srivastava, N. & Swersky, K. A separate, adaptive learning rate for each connection. *Lecture 6d - Slides of Lecture Neural Networks for Machine Learning* (2012).

70. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *CoRR* **abs/1412.6980**. arXiv: 1412.6980. <<http://arxiv.org/abs/1412.6980>> (2014).

Acknowledgements

First of all I want to thank my supervisor *Prof. Dr. Jörg Main*, who gave me the chance to work on this interesting topic. Furthermore I would like to acknowledge *Matthias Feldmaier* for his guidance through this subject. I also want to thank *Philippe Schraft* for stimulating conversations about machine learning and artificial intelligence. Additionally I want to thank my office mate *Johannes Reiff* for inspiring discussions about transition state theory and software engineering in general.

Last but not least, I would like to express my gratitude to my family. I am very thankful that my parents *Gaby* and *Michael* gave me the chance to let my dream to become a physicist come true. Furthermore, I am deeply grateful for their mental encouragement and financial support during the last eight years, which enabled me to obtain a bachelor's degree in physical engineering as well as both a bachelor's and a master's degree in physics. Moreover, my brother *Matthias* was always the best motivation and inspiration with his study of theoretical computer science. He deserves my special thanks for many hours of exciting discussions over both of our passions.

Erklärung

Ich versichere,

- dass ich diese Masterarbeit selbständig verfasst habe,
- dass ich keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe,
- dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist,
- und dass das elektronische Exemplar mit den anderen Exemplaren übereinstimmt.

Stuttgart, den 21. September, 2018

Martin Tschöpe