

Efficiency of micro-macro acceleration for scale-separated stochastic differential equations

Hannes Vandecasteele

Thesis voorgedragen tot het behalen
van de graad van Master of Science in
de ingenieurswetenschappen:
wiskundige ingenieurstechnieken

Promotoren:

Prof. dr. ir. G. Samaey
dr. P. Zieliński

Assessoren:

Prof. dr. ir. S. Vandewalle
Prof. dr. ir. D. Nuyens

© Copyright KU Leuven

Without written permission of the thesis supervisors and the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to the Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 or by email info@cs.kuleuven.be.

A written permission of the thesis supervisors is also required to use the methods, products, schematics and programmes described in this work for industrial or commercial use, and for submitting this publication in scientific contests.

Zonder voorafgaande schriftelijke toestemming van zowel de promotoren als de auteur is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot het Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 of via e-mail info@cs.kuleuven.be.

Voorafgaande schriftelijke toestemming van de promotoren is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Preface

I would like to thank everyone who contributed to my thesis. First and foremost, I would like to thank my promoter, Giovanni Samaey, for accepting me as a thesis student even though he was on sabbatical during the first semester. He was always available for regular meetings or Skype calls and had a broader mindset and stressed the importance of my thesis. I'd also like to thank my supervisor, Przemysław Zieliński, for the intense collaboration, stressing mathematical rigour and for giving some insights in functional analysis. Finally I would like to thanks my parents who gave me the opportunity to pursue a Master's degree in engineering.

Hannes Vandecasteele

Contents

Preface	i
Abstract	iv
Abstract	v
List of Figures	vi
List of Symbols	ix
1 Introduction	1
1.1 Background	1
1.2 Techniques to overcome stiffness	1
1.3 A micro-macro acceleration method	3
1.4 Goal and contributions	4
2 The micro-macro acceleration method	7
2.1 A General Micro-Macro Acceleration Algorithm	7
2.2 Properties of the restriction and matching operators	10
2.3 Convergence Theorem	11
2.4 Matching in f -divergence	12
3 Implementation of the micro-macro acceleration algorithm	15
3.1 Practical Implementation Issues	15
3.2 Object-oriented structure of the code	18
3.3 Numerical Verification: FENE Dumbbells	21
4 Linear slow-fast SDEs: Convergence & Stability	25
4.1 A decomposition for matching with only slow state variables	26
4.2 Analytic results for matching with a Gaussian initial distribution	28
4.3 Convergence of the micro-macro acceleration method with slow mean extrapolation	33
4.4 Stability of micro-macro acceleration	36
5 Effect of the choice of macroscopic state variables	43
5.1 The Fokker-Planck equation and closure relations	43
5.2 Estimating the number of moments for linear SDEs	45
5.3 A non-linear case: FENE-dumbbells	47

6 Accuracy of the extrapolation step size	53
6.1 A natural averaging strategy	53
6.2 A linear driven process	55
6.3 Efficiency and accuracy of the micro-macro scheme	56
7 Applications	61
7.1 A bistable system	61
7.2 A tri-atom molecule	66
8 Conclusion and Outlook	73
8.1 Results	73
8.2 Outlook to future work	74
A Finite volume method for the Fokker-Planck equation	77
A.1 Derivation of a finite volume scheme	77
A.2 Boundary conditions	79
A.3 Order test to verify a correct implementation	79
B Derivation of a closure model for linear SDEs	81
B.1 Derivation of the closure relations	81
B.2 Analytical solution of the closure relations	82
B.3 Convergence test for the forward Euler implementation	83
C Poster	85
Bibliography	87

Abstract

We investigate a new micro-macro acceleration algorithm for stochastic differential equations (SDEs) with a time-scale separation. Standard explicit Monte Carlo time steppers are computationally too expensive for highly stiff systems due to a very small stability domain. The micro-macro acceleration scheme is able to take larger time steps by a general four-step procedure. (i) We perform a short explicit Monte Carlo simulation of the complete differential equation. (ii) At each microscopic time step we record some macroscopic state variables of interest. (iii) We then extrapolate these variables over a much larger time step. (iv) Finally, we build a new microscopic probability density that is consistent with the extrapolated states, and deviates the least from the final distribution obtained after the microscopic steps. The fourth step is called *matching* and is the hardest part of the micro-macro acceleration method.

The goal of the thesis is to perform an efficiency analysis of the micro-macro acceleration scheme. It is already known that the micro-macro acceleration scheme converges to the full microscopic dynamics when the number of state variables increases to infinity and the extrapolation and microscopic step sizes decrease to zero. In contrast, it is also known that the stability domain for extrapolation is independent of the time-scale separation, which allows for large extrapolation time steps.

We discuss both convergence and stability and proposes an object-oriented structure for an efficient implementation. The first contribution of this thesis is a new convergence result of the micro-macro acceleration scheme for linear SDEs with only slow-mean extrapolation. This theorem is in contrast to the general convergence theorem that requires an infinite hierarchy of state variables. The second contribution is an extensive efficiency analysis of the acceleration scheme, presented in three parts. The first part is an a priori estimation of how many state variables are required for accurate simulations, without performing too many computations. This text discusses the effect of state selection for a linear SDE and also considers a non-linear example. The second part investigates how large the extrapolation time step can become before the micro-macro acceleration method becomes less accurate than an approximate macroscopic model. We show that the micro-macro acceleration scheme can simultaneously be more efficient than a pure microscopic simulation and more accurate than an approximate macroscopic model. In the third part, we consider two practical applications where the macroscopic approximations make a steady-state modelling error. The micro-macro acceleration scheme is able to remove the modelling error while extrapolating further than the microscopic time stepper.

Abstract

We onderzoeken een nieuw micro-macro acceleratie algoritme om de simulatie van stochastische differentiaalvergelijken met een tijdsschaal separatie te versnellen. Gewone expliciete Monte Carlo methods zijn computationeel te duur voor stijve problemen door hun kleine stabiliteitsgebied. De micro-macro acceleratie methode kan grotere tijdsstappen nemen via een generieke procedure, in vier stappen. (i) Eerst doen we een korte Monte Carlo simulatie van het stijve systeem. (ii) Na elke stap berekenen we een aantal macroscopische toestanden. (iii) Deze toestanden extrapoleren we vervolgens over een groter tijdsinterval. (iv) Tot slot construeren we een nieuwe microscopische distributie die consistent is met de geëxtrapolerde toestanden, *matching* genoemd. De nieuwe verdeling is zodanig dat ze minimaal afwijkt van de laatst gekende verdeling na de korte Monte Carlo simulatie. Matching is de duurste stap van het algoritme.

Het doel van deze thesis is een onderzoek naar de efficiëntie van het micro-macro acceleratie algoritme. Er is reeds convergentie aangetoond wanneer het aantal toestanden naar oneindig gaat en de tijdsstappen naar nul convergeren. Het stabiliteitsdomein van de extrapolatie is ook onafhankelijk van de stijve component van het proces, waardoor grotere stappen mogelijk zijn dan bij een microscopische simulatie.

We bespreken convergentie en stabiliteit en stelt een efficiënte object-geörienteerde implementatie voor. De eerste nieuwe bijdrage van de thesis is een nieuwe convergentiestelling van de micro-macro acceleratie methode voor lineaire SDEs, met enkel extrapolatie van het gemiddelde van de trage mode. Deze stelling staat in scherp contrast tot de algemene convergentiestelling die een hiërarchie van oneindig veel toestanden veronderstelt. De tweede bijdrage is een efficiëntie-analyse van de acceleratie methode, in drie delen. Het eerste is een schatting van het aantal toestanden nodig voor een nauwkeurige simulatie. Deze tekst bestudeert het effect van de toestandskeuze voor lineaire processen en beschouwt ook een niet-linear systeem. Het tweede luik behandelt de grootste extrapolatie stap mogelijk van de micro-macro acceleratie methode, vooraleer benaderende modellen voor de trage mode accurater worden. Deze tekst voert een die analyse uit op een aangedreven linear systeem. We vinden dat de maximale extrapolatiefactor stijgt naarmate de tijdsschaal separatie groter wordt. Tot slot beschouwt het derde deel van de efficiëntie-analyse twee modellen uit de literatuur: een bimodaal proces en een molecule bestaande uit drie atomen. De macroscopische benaderingen van de trage mode maken een steady-state fout. Het micro-macro acceleratie algoritme echter volgt de microscopische dynamica goed, terwijl grotere tijdsstappen mogelijk zijn dan de microscopische simulatie.

List of Figures

2.1	The four stages of the micro-macro acceleration algorithm	8
3.1	Object-oriented implementation for the micro-macro acceleration algorithm	20
3.2	Exact FENE densities after 0, 1 and 1.1 seconds	22
3.3	Different matching strategies for FENE with prior after 1 second and target distribution after 1.1 seconds.	23
3.4	Relative error of higher order moments of the FENE process	24
3.5	Relative error of the stress tensor as a function of Δt	24
4.1	Analytical results for mean-only extrapolation on a linear SDE	33
4.2	Analytic results for mean-variance extrapolation on a linear SDE	34
4.3	Stability of the explicit Euler-Maruyama scheme	38
4.4	Stability of mean-only extrapolation and the average extrapolation step	40
4.5	Invariance of the stability threshold with respect to ε with slow mean extrapolation	41
4.6	Stability of the micro-macro acceleration scheme with mean-variance extrapolation	42
5.1	Agreement between several numerical techniques for linear SDEs	45
5.2	Impact of the number slow moments on the slow mode of a linear SDE	46
5.3	Impact of the number of fast moments on the slow mode of a linear SDE	47
5.4	Impact of the number of slow moments on a linear SDE with uniform initial condition	47
5.5	Hysteresis of the $\tau - M_1$ phase diagram for the FENE-model	49
5.6	Accuracy of the first moment hierarchy for FENE-dumbbells	49
5.7	Accuracy of the second moment hierarchy for FENE-dumbbells	50
5.8	Accuracy of the third moment hierarchy for FENE-dumbbells	51
6.1	Approximate macroscopic model for the periodic model	56
6.2	Convergence of the approximate macroscopic periodic model	57
6.3	Micro-macro acceleration solutions for the periodically driven linear system	57
6.4	Convergence order of the micro-macro acceleration method	58
6.5	Crossover curves when the micro-macro acceleration scheme is more accurate than the approximate macroscopic model	59

6.6	Maximal extrapolation factor of the micro-macro acceleration scheme before becoming less accurate than the approximate macroscopic model	59
7.1	Bimodal potential and the associated invariant distribution	62
7.2	Particle moving through the double well and the influence on the slow mode .	63
7.3	Steady-state error of the approximate macroscopic model	64
7.4	Limiting cases where the approximate macroscopic model seems accurate . .	65
7.5	Improvement by the micro-macro acceleration method on the bimodal model	65
7.6	Visualisation of the planar tri-atom molecule	67
7.7	Drift and diffusion term of reaction coordinate $(A - C)^2$	68
7.8	Accurate approximations of the approximate macroscopic model with reaction coordinate $\xi_1 = \theta$	69
7.9	Steady-state error of the approximate macroscopic model with reaction coordinate $\xi_1 = \theta$	69
7.10	Accurate approximation of the micro-macro acceleration scheme with extrapolating ξ_1	70
7.11	Accurate approximation of the micro-macro acceleration scheme with extrapolating ξ_2	71
7.12	Accurate approximation of the micro-macro acceleration scheme with extrapolating the first two moments of x_c and y_c	72
A.1	Finite volume grid	78
A.2	Convergence Fokker-Planck	80
B.1	Linear closure convergence order	83
C.1	Poster	86

List of Symbols

δt	Microscopic time step
Δt	Extrapolation time step
K	Number of microscopic time steps
L	Number of macroscopic state variables
N	Number of Monte Carlo particles
k	Microscopic stepper index
n	Macroscopic time index
l	Macroscopic state variable index
j	Monte Carlo particle index
\mathcal{X}	Monte Carlo particle ensemble
X, Y	Random variables
S	Microscopic time stepper
\mathcal{R}_L	Restriction operator
\mathcal{E}	Extrapolation operator
\mathcal{P}_L	Matching operator
π	Prior density
φ	Matched density
W	Brownian motion
\mathcal{N}	Normal distribution
\mathcal{U}	Uniform distribution
\mathbb{E}	Expected value
Var	(Co-) variance matrix
\mathcal{H}	Kullback-Leibler divergence
ε	Time-scale separation
λ	Lagrange multiplier
β	Inverse temperature
\mathbb{R}	Set of real numbers
\mathbb{N}	Set of natural numbers, including 0

Chapter 1

Introduction

1.1 Background

Many processes in nature can be described by a system of differential equations (ordinary, partial or stochastic), in which a large time-scale separation is present. For example, the relevant time scales for an individual particle in a fusion reactor relate to the time between particle collisions, whereas the plasma as a whole evolves on a much slower time scale [12]. Similarly, individual atoms in a polymer network vibrate on a time scale that is much faster than the motion of the entire polymer molecule [32]. The difference in time scales makes simulating such processes very costly. We need to capture the fast dynamics accurately as they influence the slower time scales. In practice however, we are only interested in the evolution of some slow functions of interest. Hence, not only we need to use small time steps to capture the fast dynamics, we also need to compute a lot of these time steps to simulate the slow evolution over long enough time intervals. This is an inherent problem with stiff systems of equations.

In many cases, the fast variables are modelled using a stochastic process. This is the case if some missing knowledge leads to some inherent stochasticity during the modelling stage. Alternatively, if the fast dynamics is modelled via a high-dimensional partial differential equation, deterministic grid-based numerical schemes soon become intractable. As a result, Monte Carlo techniques are very attractive to solve inherent stochastic or high dimensional problems with a reasonable accuracy. This thesis deals with an algorithm to speed up the simulation of stochastic differential equations with a large time-scale separation, while remaining truthful to the fast microscopic dynamics.

1.2 Techniques to overcome stiffness

There are many techniques that are able to overcome the issue of stiffness, both for deterministic and stochastic problems. In this paragraph, we review implicit methods, explicit methods with an extended stability domain, schemes that make use of approximate macroscopic models, equation-free techniques and a new slow-fast acceleration method

recently developed at KU Leuven, which will be the focus of this thesis.

Implicit time steppers Implicit time integration methods have proven to be very successful in the context of stiff ODEs [7]. These methods allow for much larger time steps than their explicit counterparts because implicit methods generally have a much larger stability domain. The computational cost of an implicit method is however higher than for an explicit solver: at every time step a system of equations needs to be solved. Nevertheless, implicit methods still are a major improvement over the only small steps explicit methods can take.

A problem arises however when employing implicit techniques to SDEs [6]. The authors of [24] have shown that these methods are unable to capture the probability distribution of the fast modes. Implicit methods on stochastic problems yield very inaccurate results, unless the time steps are very small. In other words, the big advantage of implicit methods, e.g. larger time steps, is exactly what prevents them from working efficiently on stochastic differential equations. For SDEs, there is no reason to choose implicit methods over the simpler and more accurate explicit solvers.

S-ROCK It is possible to systematically enlarge the stability domain of explicit solvers by the so-called S-ROCK (Stochastic Orthogonal Runge-Kutta Chebyshev) method [1, 2, 20]. S-ROCK increases the stability domain by using a Runge-Kutta like stage-wise scheme with Chebyshev polynomials. The method can be useful since a small stability domain is one of the main drawbacks of explicit methods for slow-fast stochastic systems. A drawback is that S-ROCK schemes do not attain a high convergence order [21].

Approximate macroscopic models In specific situations, it is sometimes possible to derive an approximate macroscopic model for the slow modes. Such an approximation typically holds, in the limit when the time-scale separation goes to infinity [13, 29, Chapter 11]. Examples of such approximate macroscopic models occur in the field of molecular dynamics [22], bacterial chemotaxis [8] and many others. Using an approximate macroscopic model, one can take much larger time steps to simulate the slow variables of interest, while the averaged effect of the microscopic dynamics is correctly taken into account.

Equation-free techniques In many situations however, an approximate macroscopic model is not available in closed form. Methods for these kinds of problems are called *equation-free* techniques, since there is no direct macroscopic model to simulate [18]. Another paradigm in this context is called *heterogeneous multiscale methods* (HMM) [3] but this technique is equivalent to the equation-free formulation. Usually in an equation-free context, one is only interested in a few macroscopic quantities, or state variables, that describe the slow modes accurately enough.

A general four-way framework then exists to simulate the unavailable macroscopic model over larger time intervals, explained in much detail in [19]. First, a short Monte Carlo

simulation of the microscopic model is performed with small time steps, attuned to the fast dynamics. Second, at every microscopic step some macroscopic state variables of interest are recorded, what is called *restriction*. Third, these macroscopic variables are extrapolated over a much larger time step to accelerate the simulation. This step is also called *projective integration* in the literature [19]. Finally, a new microscopic state or distribution is built from the extrapolated macroscopic quantities in the *lifting* step. Lifting obtains a unique distribution that is both consistent with and in conditional equilibrium with respect to the extrapolated macroscopic states. The complete four-step algorithm tries to accelerate the simulation of the complete stiff system by taking larger time steps than the microscopic time stepper, while still maintaining accuracy.

The method with lifting however exhibits two drawbacks. First, it is only possible to prove convergence of projective integration when the time-scale separation tends to infinity [3]. In this limit, the fast variables settle almost immediately to their conditional invariant distribution making larger extrapolation steps possible. If however the time-scale separation is not that large (order 10, 100), a modelling error is introduced since the fast modes are not modelled accurately enough. As these fast modes also influence the slow modes, a lack of accuracy in the former variables induces an inaccurate approximation of the latter. The second drawback of that lifting typically initializes the new microscopic distribution in such a way that there is a large deviation from the exact microscopic distribution. In practice, lifting usually precedes a few constrained simulation steps that reduces the deviation from the distribution obtained after the microscopic time steps. See [32] for more details. These constrained simulation steps are however computationally very expensive.

1.3 A micro-macro acceleration method

Recently, a new micro-macro acceleration scheme was developed at KU Leuven [9] that tries to alleviate the two drawbacks from the equation-free context. This new method focusses on processes modelled by general stochastic differential equations in Itô sense

$$d\mathbf{X}(t) = a(t, \mathbf{X}(t))dt + b(t, \mathbf{X}(t))d\mathbf{W}(t), \quad \mathbf{X}(t) \in G \quad (1.1)$$

on the time interval $[0, T]$, where $a(t, \mathbf{x})$ is the drift vector, $b(t, \mathbf{x})$ the diffusion matrix and \mathbf{W} is an n -dimensional Wiener process. The set $G \subset \mathbb{R}^d$ is the domain of the random variable $\mathbf{X}(t)$ and the initial distribution $X(0)$ is independent from the Brownian motion \mathbf{W} . The proposed new micro-macro acceleration method replaces the expensive lifting procedure by a new *matching* algorithm. Matching picks the probability density that is closest to a prior distribution and is consistent with the extrapolated macroscopic state variables. The prior is again the final distribution obtained after the microscopic steps in the first stage. The L macroscopic state variables are defined as the expectation $\mathbb{E}[R_l(\mathbf{X}(t))]$ of given state functions R_l , of the process,

$$m_l(\mu) = \mathbb{E}[R_l(\mathbf{X}(t))] = \int_G R_l(\mathbf{x})d\mu(\mathbf{x}) \quad l = 1, \dots, L \quad (1.2)$$

where μ is the associated probability law at every step. The restriction *operator*, defined as the vector of states at every time step, then reads

$$\mathcal{R}_L(\mu) = (m_1(\mu), \dots, m_1(\mu)) = \mathbf{m}(\mu). \quad (1.3)$$

The expected improvement over lifting is twofold. First, matching does not require a constrained simulation afterwards, since the new matched distribution is already close to the previous computed distribution. Second, the authors of [9] are able to prove convergence of projective integration with matching to the exact microscopic distribution, even if there is a finite time-scale separation. This convergence result is in sharp contrast to equation-free methods, where there is only convergence when the time-scale separation tends to infinity. As a consequence, projective integration with matching does not make a modelling error for larger time-scale separations. The expected improvements make the matching algorithm both faster and more accurate than projective integration with lifting.

The acceleration algorithm converges to the exact microscopic dynamics when the step sizes decrease to zero and when the number of state variables L increases to infinity [9]. There also exist stability bounds on the extrapolation step size above which the numerical scheme diverges to infinity [10], more clearly explained in Chapter 4. However, no tangible results exist on efficiency of the micro-macro acceleration method, i.e. the minimal computational effort required to attain a certain accuracy during the simulation. The computational cost of the micro-macro acceleration method decreases when the number of state variables to match decreases. The cost also diminishes when the extrapolation step increases so that we can take larger steps at the time. In both situations, the error however increases.

1.4 Goal and contributions

The objective of the text is twofold. First we try to assess the choice of the number of macroscopic states so that the acceleration method is accurate, and that adding an extra state variable does not yield a significant improvement anymore. Chapter 5 discusses the choice of states on a linear and non-linear SDE. Second, for many practical problems it is possible to derive an approximate macroscopic model for the slow modes of (1.1). We want to find how large an extrapolation step one can take above which the micro-macro acceleration scheme becomes less accurate than the macroscopic approximation to the inherently stiff SDE. In this situation, the micro-macro acceleration scheme is more efficient than a microscopic simulation and more accurate than the approximate macroscopic model. This way, the micro-macro acceleration scheme can outperform current state-of-the-art acceleration methods, thereby proving the merit of the method.

The text is organized as follows:

- Chapter 2 goes in depth into the micro-macro acceleration algorithms and defines all the four steps rigorously. It also contains the convergence theorem and introduces the notion of relative-entropy matching. Chapter 2 is part of the literature study.

- Chapter 3 presents the implementation of matching and discusses some practical implementation issues. The chapter also contains an efficient object-oriented structure of the code and a case study to verify the correctness of the implementation.
- Chapter 4 is the beginning of the efficiency analysis and we discuss the micro-macro acceleration algorithm on a linear SDE. More specifically, it contains novel proofs on matching with a Gaussian prior and also presents a new convergence result for linear SDEs [4](#). The chapter ends with a summary on stability. The new convergence theorem is important since general convergence of the micro-macro acceleration scheme requires an infinite hierarchy of macroscopic state variables, while the new theorem only requires the slow mean. The downside is that it only works for linear SDEs.
- Chapter 5 assesses what the effect of the number of extrapolated state variables is for a linear scale-separated SDE. The chapter discovers an adequate number of moments of the linear process that yields accurate simulation results, while avoiding superfluous computational work.
- Chapter 6 studies the error as a function of the extrapolation time step and the inherent time-scale separation of the process. This analysis is done for a periodically driven linear system, for which we possess an approximate macroscopic model in the limit of infinite time-scale separation. The chapter finds that the associated maximal extrapolation factor for the micro-macro acceleration method gets larger when the time-scale separation gets stronger, while maintaining an accuracy that is higher than the limiting approximate macroscopic model. This is a novel result, proving the merit of the micro-macro acceleration scheme.
- Finally, Chapter 7 presents two practical applications where the micro-macro acceleration method outperforms macroscopic models in terms of accuracy and microscopic models in terms of computation time. The first is a slow-fast bimodal problem while the other is a set of equations that describe the behaviour of a three-atom molecule, arising from molecular dynamics [\[22\]](#). In both cases, the approximate macroscopic models make a steady-state error. The micro-macro acceleration method is able to eliminate the steady-state error, while still allowing for larger time steps than a complete microscopic simulation.
- Chapter 8 contains the final conclusion.

Chapter 2

The micro-macro acceleration method

As mentioned in the introduction, the micro-macro acceleration method approximates the true microscopic dynamics through some macroscopic state variables of interest, in a four-way scheme. (i) A short burst Monte Carlo simulations of the SDE with a particle ensemble. (ii) At every microscopic step, we record the state variables, also called restriction. (iii) We then extrapolate these state variables over a much larger time step. (iv) Finally, we compute a new distribution that is consistent with the extrapolated states and minimizes some distance function, with respect to the prior distribution. The final step is called matching and the prior distribution is the final distribution obtained during the Monte Carlo simulation of step (i).

In this chapter, we discuss the micro-macro acceleration scheme in more mathematical depth, as introduced in [9], with similar notation. Section 2.1 introduces the complete algorithm with the exact definition of the restriction, extrapolation and matching operator. In Section 2.2, we set forth some properties these operators need to have. Based on these properties, in Section 2.3 we state the exact convergence theorem of the micro-macro acceleration method without proof. Finally, Section 2.4 introduces so-called relative-entropy matching as an example of a practical matching operator that has all the required properties. Relative-entropy matching will also be used throughout the text. We also touch lightly on other possible matching strategies in the final section.

2.1 A General Micro-Macro Acceleration Algorithm

Figure 2.1 depicts a visual representation of one cycle of the acceleration method and Algorithm 1 describes the four steps in more detail. Suppose we are given a weighted Monte Carlo ensemble of J particles $\mathcal{X}^n = (w_j, X_j^n)_{j=1}^J$ at time $t^n = n\Delta t$. The K inner steps of the microscopic simulation are carried out with a time stepper $S^{n,k}$ that is in principle time dependent. This text only considers the well-known Euler-Maruyama

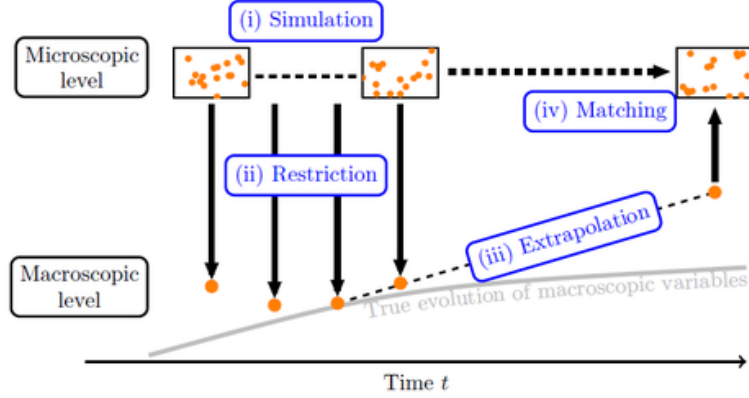


Figure 2.1: Visual representation of the four stages of the micro-macro acceleration algorithm 1. This figure was taken from [9].

method [31], which is a discretization of the SDE (1.1) with time step δt

$$X_j^{n,k+1} = X_j^{n,k} + a(t^n + k\delta t, X_j^{n,k})\delta t + b(t^n + k\delta t, X_j^{n,k})\sqrt{\delta t}\delta W, \quad \delta W \sim \mathcal{N}(0, 1), \quad (2.1)$$

where $\mathcal{X}^{n,k} = (w_j, X_j^{n,k})_{j=1}^J$ is the weighted Monte Carlo ensemble at time $t^{n,k} = n\Delta t + k\delta t$.

The restriction operator \mathcal{R}_L was already defined in (1.3). In the context of Monte Carlo simulations, a discrete version of the restriction operator is required that acts on a particle ensemble, instead of a continuous probability density. The discrete operator reads

$$\mathcal{R}_L(\mathcal{X}) = (\mathbb{E}_J[R_1(\mathcal{X})], \dots, \mathbb{E}_J[R_L(\mathcal{X})]). \quad (2.2)$$

where $\mathcal{X} = (w_j, X_j)_{j=1}^J$ is a weighted particle ensemble and \mathbb{E}_J the finite weighted expected value over this ensemble. The restricted macroscopic states at every microscopic step are then given by $\mathbf{m}^{n,k} = \mathcal{R}_L(\mathcal{X}^{n,k})$. The extrapolation operator \mathcal{E} extrapolates the sequence of macroscopic state vectors $(\mathbf{m}^{n,k})_{k=0}^K$ over a time interval Δt , to obtain a new macroscopic state vector, written as

$$\mathbf{m}^{n+1} = \mathcal{E}((\mathbf{m}^{n,k})_{k=0}^K, \delta t, \Delta t).$$

We will only consider linear extrapolation in this text, so the extrapolation operator becomes

$$\mathbf{m}^{n+1} = \mathbf{m}^{n,0} + \frac{\Delta t}{K\delta t}(\mathbf{m}^{n,K} - \mathbf{m}^{n,0}), \quad (2.3)$$

where $\mathbf{m}^{n,0} = \mathbf{m}^n$. Finally, the matching operator $\mathcal{P}_L(\mathbf{m}, \pi)$ computes the distribution is consistent with the macroscopic states \mathbf{m} and minimizes some distance function

$$\mathcal{P}_L(\mathbf{m}, \pi) = \arg \min_{\nu \in \mathcal{R}_L^{-1}(\mathbf{m})} d(\nu, \pi), \quad (2.4)$$

with π the prior distribution and $\mathcal{R}_L^{-1}(\mathbf{m})$ the set of all densities that are consistent with the macroscopic states \mathbf{m} . The function d does not have to be a true metric as long as it bears some meaning of similarity between two probability measures. In fact, the similarity function for relative-entropy matching in Section 2.4 is not a metric.

In the context of Monte Carlo simulations however, we need a discrete variant of the matching operator that takes a particle ensemble as prior distribution and outputs a particle representation of the matched distribution. Section 2.4 discusses three such examples of discrete matching operators. With a slight abuse of notation, we will write the particle ensemble as the second argument of the matching operator instead of the underlying probability density. For the (weighted) Monte Carlo ensemble $\mathcal{X}^{n,K}$, the fourth step of the micro-macro acceleration scheme can then be written as

$$\mathcal{X}^{n+1} = \mathcal{P}_L(\mathbf{m}^{n+1}, \mathcal{X}^{n,K}). \quad (2.5)$$

Throughout the text, we use superscript for temporal indices like n and k and subscript for counting indices like j and l .

Algorithm 1 The general micro-macro acceleration algorithm for stiff SDEs.

Given a microscopic ensemble \mathcal{X}^n at time t^n , a microscopic step size δt , the extrapolation step size Δt and a number of microscopic steps K such that $K\delta t \leq \Delta t$. Let T be the end time of the simulation and L the number of macroscopic state variables. The algorithm produces the microscopic ensemble \mathcal{X}^{n+1} in four steps:

(i) **Monte Carlo simulation:** simulate the microscopic ensemble \mathcal{X}^n over K small inner steps of size δt by using an inner microscopic discrete time stepper

$$\mathcal{X}^{n,k+1} = \mathcal{S}^{n,k}(\mathcal{X}^{n,k}, \delta t)$$

(ii) **Restriction:** compute the macroscopic states corresponding to these microscopic ensembles

$$\mathbf{m}^{n,k} = \mathcal{R}_L(\mathcal{X}^{n,k}).$$

(iii) **Extrapolation:** approximate the macroscopic states over a time interval Δt using the extrapolation operator

$$\mathbf{m}^{n+1} = \mathcal{E}((\mathbf{m}^{n,k})_{k=0}^K, \delta t, \Delta t)$$

(iv) **Matching:** compute a new ensemble consistent with these extrapolated macroscopic states using the matching operator with prior distribution $\mathcal{X}^{n,K}$

$$\mathcal{X}^{n+1} = \mathcal{P}_L(\mathbf{m}^{n+1}, \mathcal{X}^{n,K})$$

and advance time with Δt until the end time T is reached.

2.2 Properties of the restriction and matching operators

To prove convergence of the micro-macro acceleration method, the restriction operator \mathcal{R}_L cannot be built from just any set of macroscopic state functions. Every additional macroscopic state function should add new information about the distribution it represents. In other words, a new state function has to be linearly independent from the already present functions. Also, in the limit as L goes to infinity, the functions R_l should uniquely represent μ . In this limit, the modelling error in the restriction and matching step, for finite L is completely removed. That is what the next assumption states. The statement of the assumption is almost identical to the one in [9].

Assumption 1. *The macroscopic state functions R_l , $l \geq 1$ need to satisfy the following conditions:*

1. *The state functions $(R_l)_{l=1}^L$ are linearly independent on every non-null subset of G , the domain of all state vectors*
2. *There is a one-to-one correspondence between the infinite sequence of macroscopic states $(m_l(\mu))_{l=1}^\infty$ and the distribution μ .*

Another requirement of the Theorem 1 is that the restriction and matching operators are inverses of each other in some sense. Applying the matching operator with a prior distribution π and a macroscopic state \mathbf{m} such that $\mathbf{m} = \mathcal{R}_L(\pi)$ should yield the prior distribution π again. The converse also needs to be true: the restriction operator applied to a matched distribution $\mathcal{P}_L(\mathbf{m}, \pi)$ should give the exact same macroscopic states \mathbf{m} that the matched density is consistent with. A pair of restriction and matching operators that has these properties is called a *restriction-matching pair*. This definition was also taken from [9].

Definition 1. *Restriction-Matching pair. Suppose \mathcal{R}_L and \mathcal{P}_L satisfy*

1. $\mathcal{R}_L(\mathcal{P}_L(\mathbf{m}, \pi)) = \mathbf{m}$ for all $(\mathbf{m}, \pi) \in \text{dom}(\mathcal{P}_L)$
2. $\mathcal{P}_L(\mathcal{R}_L(\mu), \mu) = \mu$ for all $(\mathcal{R}_L(\mu), \mu) \in \text{dom}(\mathcal{P}_L)$

then the pair $(\mathcal{R}_L, \mathcal{P}_L)$ is called a Restriction-Matching pair.

Remark 1. Condition 1 implicitly states that the state vector \mathbf{m} should always be attainable by the restriction operator. In practice however, it sometimes happens that the vector of extrapolated macroscopic state variables \mathbf{m} fall outside the domain of the matching operator and no density can correspond to these states. In practice, condition 1 is not always met and forms the basis for an adaptive time-stepping strategy, based on matching failures. When the extrapolation step is smaller, the extrapolated states will lie closer to the exact dynamics. There then is a higher probability that the extrapolated states lie in the domain of \mathcal{P}_L so that the restriction operator can attain these states. We discuss adaptive time stepping in more depth in Chapter 3. Condition 2 on the other

hand, states that the function $\mathcal{P}_L(\mathbf{m}, \cdot)$ is a projection. Indeed, let $\mathcal{P}_L(\mathbf{m}, \pi) = \mu$, then $\mathcal{P}_L^2(\mathbf{m}, \pi) = \mathcal{P}_L(\mathbf{m}, \mu) = \mu = \mathcal{P}_L(\mathbf{m}, \pi)$ since μ is consistent with \mathbf{m} .

2.3 Convergence Theorem

All elements are now in place to state the convergence theorem for the micro-macro acceleration algorithm 1, first formulated in [9]. The theorem is stated in terms of general restriction and matching operators, but assumes linear extrapolation. The proof also requires some very technical assumptions that are explained in [9], but omitted here for clarity.

Theorem 1. *Consider an SDE of the form (1.1) with initial condition $\mathbf{X}(0) = \mathbf{X}_0$. Let R_l , $l \geq 1$, be a sequence of macroscopic state functions, fulfilling Assumption 1. Let $(R_L, P_L)_{L=1}^\infty$ be a sequence of restriction-matching pairs fulfilling Definition 1 and two technical conditions in [9]. Furthermore, consider a microscopic time discretization scheme of order $p_S \geq 1$ with time step δt . Finally, let \mathcal{E} denote linear extrapolation (2.3) with step size Δt and let $K \in \mathbb{N}$ be a number of microscopic steps with $K\delta t \leq \Delta t$. If we denote the solution of Algorithm 1 with L macroscopic state variables at time T as \mathbf{X}_L^N , then for any smooth enough function g we have*

$$\mathbb{E}(|g(\mathbf{X}_L^N) - g(\mathbf{X}(T))|) \leq C_L + \tilde{C}_L((\delta t)^{p_S} + \Delta t), \quad (2.6)$$

in which C_L and \tilde{C}_L are constants that depend also on T, g and \mathbf{X}_0 , with $C_L \rightarrow 0$ as $L \rightarrow \infty$.

Loosely speaking, Theorem 1 says that the micro-macro acceleration Algorithm 1 converges to the exact solution, when both the microscopic and extrapolation step size decrease to zero and the number of macroscopic states L goes to infinity. One of the downsides of the theorem is that it gives no precise expression for C_L , the parameter that contains information of how many states are needed to keep part of the error small. The number of macroscopic state variables is application specific, and part of the thesis is to discover what a good number of states is. We discuss the selection of adequate state variables more in Chapter 5.

Remark 2. A few remarks about Theorem 1. First, the theorem omits some technical conditions that the SDE and restriction-matching pair should have but these conditions are nicely explained in [9]. The function g also needs to belong to a certain class of functions, but the exact requirements are out of scope for this thesis. Every example in this text obeys these conditions. Second, an advantage of Theorem 1 is that the micro-macro acceleration method converges even when there is only a finite time scale separation, in contrast to other multiscale techniques. As mentioned, in the equation-free context, there is only converge when the scale separation goes to infinity. This is an asset compared to other multiscale approaches. Finally, Theorem 1 only proves convergence and does not say anything on the maximal extrapolation step Δt possible to keep the solution finite. This is related to stability and Chapter 4 goes deeper into this problem.

2.4 Matching in f -divergence

In the remainder of the text, we will always use so-called relative-entropy matching, an instance of matching in ‘ f -divergence’, introduced in [9]. Relative-entropy matching is also called Kullback-Leibler matching, or KLD in short. In general, given a prior distribution π and a vector of macroscopic state variables \mathbf{m} that the matched distribution needs to comply with, the objective of matching in f -divergence is to minimize the functional \mathcal{I}_f

$$\mathcal{P}_L(\mathbf{m}, \pi) = \underset{\varphi \in \mathcal{R}_L^{-1}(\mathbf{m})}{\operatorname{argmin}} \mathcal{I}_f(\varphi|\pi) \quad (2.7)$$

where

$$\mathcal{I}_f(\varphi|\pi) = \begin{cases} \int_G f\left(\frac{\varphi(\mathbf{x})}{\pi(\mathbf{x})}\right) \pi(\mathbf{x}) d\mathbf{x} & \text{when } \operatorname{supp} \varphi \subset \operatorname{supp} \pi \\ +\infty & \text{otherwise,} \end{cases} \quad (2.8)$$

with $f : [0, \infty) \rightarrow [0, \infty)$ a convex function such that $f(1) = 0$. The condition $\operatorname{supp} \varphi \subset \operatorname{supp} \pi$ is very technical and states that φ should be absolute continuous with respect to π . We will not go into further technical details, see [9] for more information.

A problem with (2.8) is that the solution of the optimization problem (2.7) does not necessarily have to be a probability density. There is no guarantee that the solution is positive over the domain G and that it has unit mass. To make it a density, we have to make sure the solution of (2.7) has both properties. The first property can be readily dealt with by considering the extension f_+ of f such that $f_+(x) = +\infty$ when $x < 0$. Using this extension, negative values for φ will have an infinite penalty so that these negative values will not occur on a non-null subset of G . For the second property, we can extend the restriction operator \mathcal{R}_L with an additional state function $R_0(\mathbf{x}) = 1$, such that the matched density needs to have unit mass. For the remainder of this text, the restriction operator will always contain the extra unit state, as zeroth component in \mathcal{R}_L .

The solution of the optimization problem (2.7) is given by [9]

$$\mathcal{P}_L(\mathbf{m}, \pi) = (f_+^*)' \left(\sum_{l=0}^L \lambda_l R_l \right) \pi, \quad (2.9)$$

where f_+^* is the *convex conjugate* of f_+ , defined as $f_+^*(s) = \sup_{t \geq 0} (st - f_+(t))$. The numbers $\lambda_0, \dots, \lambda_L$ are the Lagrange multipliers associated to the optimization problem. These multipliers are the solution of the non-linear system

$$\mathcal{R}_L \left((f_+^*)' \left(\sum_{l=0}^L \lambda_l R_l \right) \right) = \mathbf{m}, \quad (2.10)$$

The Lagrange multipliers are in practice computed by a Newton-Raphson scheme, explained in Chapter 3. The solution to (2.10) is unique, if it exists, because the optimization problem is convex.

Relative-entropy matching Relative-entropy matching, an instantiation of matching in f -divergence, is defined by choosing

$$f(t) = \begin{cases} t \ln t - t + 1, & t > 0 \\ 0, & t = 0. \end{cases} \quad (2.11)$$

Note that f is indeed convex and $f(1) = 0$. The minimization problem becomes

$$\mathcal{P}_L(\mathbf{m}, \pi) = \arg \min_{\varphi \in \mathcal{R}_L^{-1}(\mathbf{m})} \int_G \ln \left(\frac{\varphi(\mathbf{x})}{\pi(\mathbf{x})} \right) \pi(\mathbf{x}) d\mathbf{x}, \quad (2.12)$$

so that we are effectively minimizing the Kullback-Leibler divergence (KLD) between the functions φ and π . The KLD is also denoted by $H(\varphi|\pi)$ in the context of information theory and has some very important applications in image processing and machine learning, in which we will not go deeper. The matched density looks as follows,

$$\mathcal{P}_L(\mathbf{m}, \pi) = \exp \left(\sum_{l=0}^L \lambda_l R_l \right) \pi \quad (2.13)$$

where the Lagrange multipliers λ_l satisfy the non-linear system

$$\int_G R_l(\mathbf{x}) \exp \left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x}) \right) \pi(\mathbf{x}) d\mathbf{x} = m_l, \quad l = 0, \dots, L. \quad (2.14)$$

The authors of [23] have proven that relative entropy has all the required properties for the convergence Theorem 1, such that the micro-macro acceleration method with relative-entropy matching converges to the exact microscopic dynamics.

Other matching strategies Besides relative-entropy matching, there also exist different strategies. We mention here L2D and L2N matching, introduced in [9]. L2D-matching is also a special form of matching in f -divergence with the function f defined as

$$f(t) = \frac{1}{2}(t - 1)^2 \quad (2.15)$$

such that the matching operator looks like

$$\mathcal{P}_L(\mathbf{m}, \pi) = \max \left(0, \sum_{l=0}^L \lambda_l R_l \right) \pi, \quad (2.16)$$

see [9] for more details. Finally, there also exists a matching operator that is not based on the notion of f -divergence. Matching in L_2 -norm is defined as

$$\begin{aligned} \min_{\varphi \in L_2(G)} \quad & \|\varphi - \pi\|_2^2 \\ \text{subject to} \quad & \mathcal{R}_L(\varphi) = \mathbf{m}, \end{aligned} \quad (2.17)$$

The solution of the optimization problem is given by

$$\varphi(\mathbf{m}, \pi) = \left(\frac{\sum_{l=0}^L \lambda_l R_l}{\pi} + 1 \right) \pi \quad (2.18)$$

where the Lagrange multipliers $\lambda_0, \dots, \lambda_L$ are the solution of the linear system $H\boldsymbol{\Lambda} = \mathbf{m} - \mathcal{R}_L(\pi)$. The matrix H is defined as $(H_L)_{j,k} = \int_G R_j(\mathbf{x}) R_k(\mathbf{x}) d\mathbf{x}$. There is no guarantee in the optimization problem itself that the matched density φ is positive everywhere. The authors of [9] expand on this problem a bit more, but the only guarantee for positivity is taking small enough extrapolation steps.

The original paper [9] explains L2N and L2D-matching in more detail and demonstrates that L2N has all the required properties for the convergence proof. We will only use these two strategies in Section 3.4 for numerical verifications.

Chapter 3

Implementation of the micro-macro acceleration algorithm

In Chapter 1 and 2, we present all the mathematical concepts, formulas and insights that are necessary to understand the micro-macro acceleration method. However, the high-level formulation of Algorithm 1 is not detailed enough to implement readily. There are three implementation details not mentioned in Chapter 2: an efficient formulation of the Newton-Raphson solver, reweighting and resampling of the Monte Carlo particles and adaptivity of the extrapolation time step. We discuss these practical aspects in Section 3.1. We then go on to present an object-oriented structure of the code for an efficient and comprehensible implementation in Section 3.2. Finally in Section 3.3, we illustrate and validate the code on a practical example called FENE-dumbbells, which is also discussed in [9]

3.1 Practical Implementation Issues

3.1.1 Newton-Raphson for relative-entropy matching

Matching is inherently an optimization problem and requires the solution of a system of (non-)linear equations to find the Lagrange multipliers. We discuss here relative-entropy matching where the system of equations to solve (2.14) is non-linear. A standard Newton-Raphson iteration usually converges quite quickly to an accurate solution of these systems.

In general, the Newton-Raphson procedure tries to approximate the zeros of a differentiable function $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by iterating

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - (\nabla g(\boldsymbol{\lambda}^k))^{-1} g(\boldsymbol{\lambda}^k) \tag{3.1}$$

with a well-chosen initial value $\boldsymbol{\lambda}^0$, that should be close enough to the exact solution $\boldsymbol{\lambda}^*$. Equation (3.1) uses the variable $\boldsymbol{\lambda}$ to make the link with Lagrange multipliers clear. The iterative procedure converges quadratically to the true solution [33].

In the case of relative-entropy matching, the non-linear system $g(\boldsymbol{\lambda}) = 0$ to solve is (2.14) and reads

$$g(\boldsymbol{\lambda})_l = m_l - \int_G R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \pi(\mathbf{x}) d\mathbf{x} = 0, \quad l = 0, \dots, L. \quad (3.2)$$

Differentiating $g(\boldsymbol{\lambda})$ results in the Jacobian of the system

$$\nabla g(\boldsymbol{\lambda})_{k,l} = - \int_G R_k(\mathbf{x}) R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \pi(\mathbf{x}) d\mathbf{x}. \quad (3.3)$$

The involved integrals can be interpreted as expectations of some functions, relative to the prior density $\pi(\mathbf{x})$. Approximating the expected values by the weighted ensemble $(w_j, X_j)_{j=1}^J$ of Monte Carlo particles from the microscopic simulation, gives

$$\nabla g(\boldsymbol{\lambda})_{k,l} \approx - \sum_{j=1}^N R_k(X_j) R_l(X_j) \exp\left(\sum_{l=0}^L \lambda_l R_l(X_j)\right) w_j, \quad (3.4)$$

with the weights w_j normalized such that $\sum_{j=1}^N w_j = 1$. The Jacobian matrix is symmetric, so in practice only half of its entries needs to be computed.

The final choice that remains is the initial value $\boldsymbol{\lambda}^0$ of the Newton-Raphson procedure. A good choice is to take the $\boldsymbol{\lambda}^0 = 0 \in \mathbb{R}^{L+1}$. When there is no extrapolation, i.e. $\Delta t = \delta t$, the prior distribution is equal to the matched distribution. Formula (3.2) then requires all λ_l to be zero. If $\boldsymbol{\lambda}^0$ is not zero, there is a chance that the iterative scheme won't be able to converge to the exact zero multipliers, even in the simplest case without extrapolation.

3.1.2 Particle reweighting and resampling

Implementing matching by particle reweighting Matching produces a new probability distribution after every extrapolation step and sampling these densities can be an expensive part of the algorithm. Standard sampling techniques like the accept-reject method however, are not necessary. It is possible to sample the matched distribution in linear time, using a reweighting method. Looking at the formulas for relative-entropy matching (2.13), the matched density is the product of a function of the Lagrange multipliers and the prior density.

$$\mathcal{P}_L(\mathbf{m}, \pi) = \exp\left(\sum_{l=0}^L \lambda_l R_l\right) \pi, \quad (3.5)$$

therefore, a weighted ensemble $(w_j, X_j)_{j=1}^J$, representing the prior distribution π , can be reweighted during matching by

$$w'_j = \exp\left(\sum_{l=0}^L \lambda_l R_l(X_j)\right) w_j \quad (3.6)$$

such that $(w'_j, X_j)_{j=1}^J$ now represents the matched distribution obtained by relative-entropy matching. Reweighting can also be performed *in-place*, because once a weight w_i has been updated to w'_i , it is never required again for reweighting other particles X_j , $j > i$.

Particle resampling Due to the multiplicative behaviour of matching an ensemble of particles, it may happen that some weights become very large and others very small, causing a big spread. As a consequence, computing the expectations in the restriction operator may become very inaccurate. In practice, it is therefore necessary to resample the Monte Carlo particles every now and then so that all new particles have an equal weight and still represent the same probability distribution.

One possible criterion to estimate a large spread, is by comparing the entropy of the weights with the entropy of a set of all weights equal to $1/J$, proposed in [9]. In the case of relative-entropy matching, the relative entropy of a set of weights $(w_j)_{j=1}^J$ is

$$\sum_{j=1}^J w_j \ln(Jw_j) \in [0, \ln J].$$

When the relative entropy of the weights is larger than a certain threshold, it is necessary to resample the particles such that they resemble the same density but with equal weights. In practice, to resample quickly enough, we set the threshold to one tenth of the maximum entropy, $\ln J$. To reduce the overhead of computing entropies, the resampling criterion is checked every five extrapolation steps.

One popular resampling method is so called *stratified* resampling [15], also discussed more thoroughly in [9]. The idea is to duplicate certain particles n_j times such that all of these copies have weight $1/J$. A practical implementation generates random numbers

$$u_k = \frac{(k-1) + \tilde{u}_k}{J}, \quad \tilde{u}_k \sim \mathcal{U}(0, 1)$$

and takes $n_j = \#\{u_k | u_k \in [\sum_{i=1}^{j-1} w'_i, \sum_{i=1}^j w'_i]\}$. The number of particles is chosen such that it yields an unbiased resampled distribution, in expectation.

3.1.3 Resolving matching failures by extrapolation time step adaptivity

If the extrapolation time step Δt is too large, the extrapolated macroscopic states \mathbf{m} may fall out of the domain of the matching operator \mathcal{P}_L . In this situation, there is

no density φ such that $\mathcal{R}_L(\varphi) = \mathbf{m}$, and matching will fail. One situation where this happens often is when the slow dynamics changes relatively fast so that the macroscopic states are extrapolated too far and hence deviate considerably from the exact states. The micro-macro acceleration algorithm should be robust against matching failures. One way of handling failures is by reducing the extrapolation step size when matching fails. Such failures are very costly since they consume a lot of resources without adding any value. A strong reduction of the extrapolation step size is required to prevent more costly failures around this simulation step. Therefore the authors of [9] propose to reduce Δt with a factor of two when a matching failure occurs, and to increase the step with a factor of 1.2 when matching succeeds. The result is an adaptive acceleration algorithm that can both quickly fall back when an matching error occurs and gently increase the step size when everything works fine.

Defining a matching failure in practice is not straightforward. One possible way, proposed in [9], is by looking at the number of Newton-Raphson iterations when finding the Lagrange multipliers associated to matching. If the solver does not converge within a reasonable amount of steps, we can safely assume that the solver will probably not converge rapidly. Such a behaviour can be an indication of a matching failure. In practice, a maximum number of iterations of 5 is usually sufficient. Even if the solver could converge with more iterations than the predefined maximum number of iterations, it is a good idea to reduce the extrapolation step size. The Newton-Raphson iterations are the most expensive part and constitute a bottleneck in the algorithm. Reducing the extrapolation step size will reduce the number of Newton-Raphson iterations and could provide a speed up, compared to larger step sizes.

3.2 Object-oriented structure of the code

Now all practical details are in place for a robust implementation of the micro-macro acceleration algorithm in `Python`. However, a decent object-oriented structure of the code is also very important. There are three different matching strategies that can be used, many possible microscopic time steppers besides a simple Euler-Maruyama method and also many iterative solvers to solve the non-linear system besides the standard Newton-Raphson method. A BFGS strategy can also be used and could speed up the iterative solver. Besides these alternatives, there could be variants of the coarse projective integration method using lifting instead of matching. All these four different parts have to play well with each other in a carefully designed object-oriented structure.

One popular method to combine all these parts is called `dependency injection` [28]. The main idea is that an object can take other another object that complies with an interface, as input. The containing object is then able to call certain methods defined in the interface, without knowing what the exact type the contained object is. For example, the coarse projective integration class `MicroMacroCPI` implements the micro-macro acceleration algorithm with matching. This class only requires a `MatchingStrategy` object on which it can call certain methods, but the projective integrator should not

care whether it is KLD, L2D or L2N matching. The `simulate` method only requires the particle ensemble, the start and end time and time step size so that it is able to perform the acceleration algorithm. The complete object-oriented structure is depicted in Figure 3.1. Dependency injection is relatively easy to implement in Python since each containing object assumes that the injected objects have a certain interface, without actively checking this. Other languages like C++ or Java require to define explicit virtual objects or interfaces. A drawback of dependency injection is that it requires extensive documentation of the code so that the user perfectly knows how each object is interfaced. We now discuss the important classes in the object-oriented diagram.

Matching Strategies All the matching strategies have in common that they need to compute the Lagrange multipliers and weights based on these values. It is therefore possible to define a super class with abstract `match` and `weights` methods. The `match` method takes as argument the current weighted Monte Carlo ensemble and the extrapolated macroscopic state variables. An additional argument is a `lambda expression` representing the macroscopic state functions R_i . The method then computes the Lagrange multipliers associated to the input arguments. It also returns whether the solver converged or not. The `weights` method takes the computed Lagrange multipliers as input, together with previous weights and computes the new weights of the particles after matching.

The three different matching methods discussed above are subclasses of `MatchingStrategy`. KLD and L2N for instance, take an iterative solver by dependency injection as a constructor argument. They also requires the maximum number of iterations, in order to stop the solver timely.

Iterative Solvers This thesis considers only one iterative solver for the non-linear systems arising from matching in f -divergence: a Newton-Raphson solver. The constructor of the Newton-Raphson class requires an object that has a `function` and `jacobian` method. These functions are called every step and take only the current Lagrange multiplies as argument. KLD implements such an interface as it requires an iterative solver. The Newton-Raphson solver also takes a dictionary of parameters, like the tolerance and maximum number of iterations. Finally it has a `solve` method, required by KLD, that only takes the initial condition as input argument and returns whether the solver converged and the final iteration value.

Microscopic Time steppers The base class for the microscopic time integrators is `MonteCarloIntegrator`. The `integrate` method takes an ensemble of particles, a start and stop time and a time step as arguments. The method advances every particle over the time step by calling its `step` method. Each subclass implements its own `step` method. For instance, the standard `EulerMaruyama` class has no boundary conditions but `GridEulerMaruyama` does as it has to reflect the particles at the boundary. The FENE example from next section requires an accept-reject strategy near the boundary of the domain and has a special subclass that implements the accept-reject method.

3. IMPLEMENTATION OF THE MICRO-MACRO ACCELERATION ALGORITHM

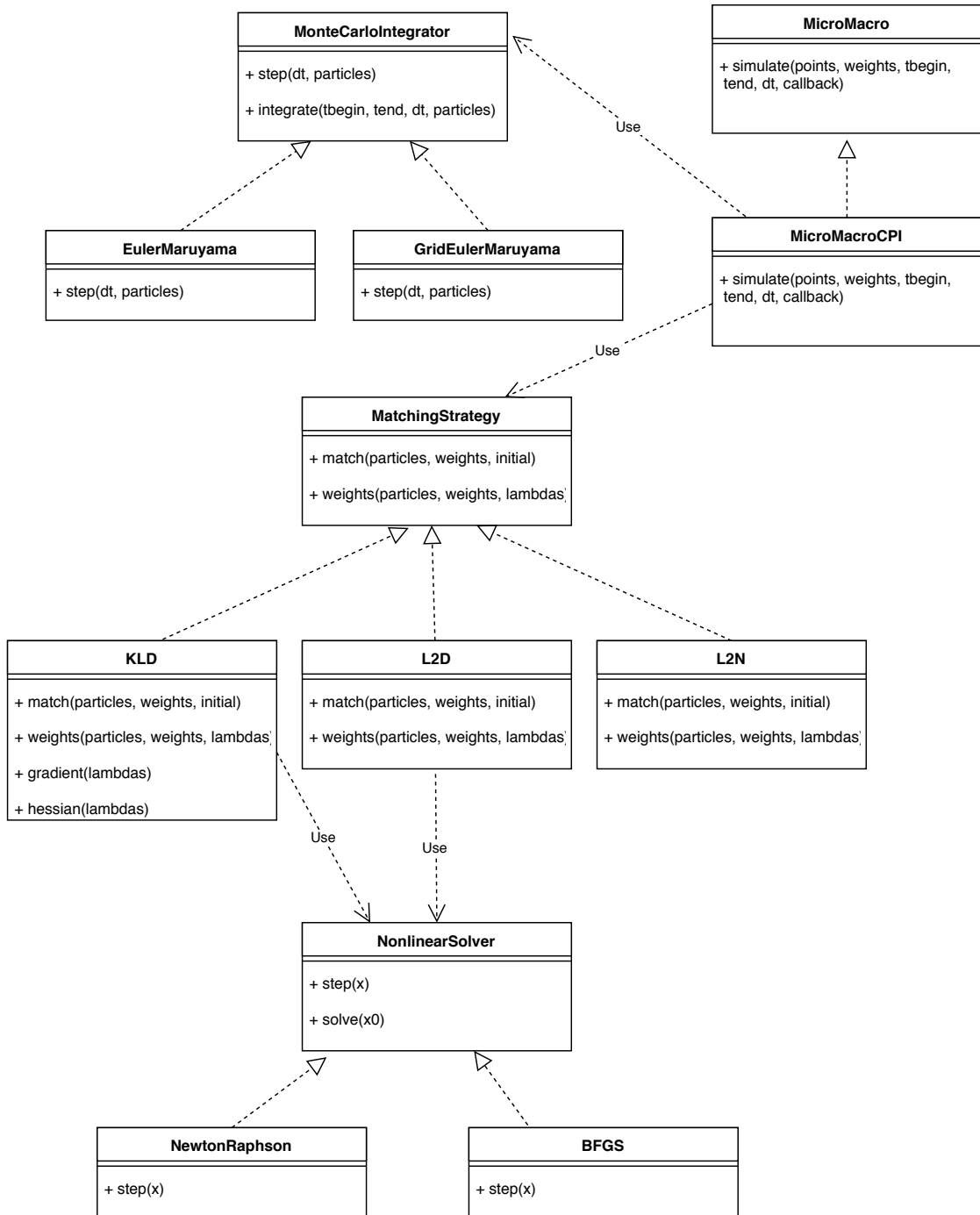


Figure 3.1: Class diagram of all relevant classes and subclasses for an efficient and reusable implementation of the micro-macro acceleration algorithm. The full arrowheads represent inheritance while the arrows with ‘use’ on represent object dependencies.

3.3 Numerical Verification: FENE Dumbbells

This section serves as a validation of the object-oriented implementation of the micro-macro acceleration algorithm. The sample problem is a one-dimensional system, called FENE-dumbbells. The FENE system is also extensively studied in the context of lifting [32]. The experiments in this section reproduce the numerical tests from [9] and serve as a confirmation of object-oriented implementation.

FENE stands for ‘Finitely Extensible Non-linear Elastic’ dumbbells, which are polymer chains moving through a solvent in which they are immersed. The stochastic variable $X(t)$ represents the length of the polymer, which is modelled as two beads connected by a non-linear spring. The spring force is modelled as

$$F : B(\sqrt{b}) \rightarrow \mathbb{R}, x \mapsto F(x) = \frac{b}{b - x^2}x \quad (3.7)$$

where \sqrt{b} is the maximum length the dumbbells. The spring force diverges to infinity near $x \approx \pm\sqrt{b}$, which causes stiffness near the edges of the domain of the polymer lengths. The explicit Euler-Maruyama integrator then requires small time steps to simulate the behaviour of the Monte Carlo particles accurately. Besides the spring force, the polymers also experience Stokes’ drag and Brownian motion. The complete SDE for this problem is

$$dX(t) = \left(\kappa(t)X(t) - \frac{1}{2W_e}F(X(t)) \right) dt + \frac{1}{\sqrt{W_e}}dW(t), \quad t \in [0, T] \quad (3.8)$$

where $\kappa(t)$ is the time-dependent velocity field due to Stokes drag and W_e is the Weissenberg number. An important quantity in practice is the stress tensor

$$\tau = \frac{1}{W_e} (\mathbb{E}[XF(X)] - 1), \quad (3.9)$$

which we want to approximate as accurately as possible.

The following experiments assume a constant velocity field $\kappa(t) = 2$ and a Weissenberg number of 1. All the exact densities are obtained by performing an Euler-Maruyama integration with time step $\delta t = 2 \cdot 10^{-4}$, where the initial condition is the invariant distribution with zero velocity field and. The initial distribution then reads

$$X_0(x) = \frac{1}{Z} \exp(2W_e U(x)) \quad (3.10)$$

where $U(x)$ is the potential energy function associated with the spring force $F(x) = \nabla U(x)$. A standard accept-reject sampling strategy suffices to sample the initial distribution with $N = 10^5$ independent particles.

Euler-Maruyama simulation A standard Euler-Maruyama simulation is not consistent with the stochastic model. Due to the Brownian increments, some particles may slip out the domain $[-\gamma, \gamma]$, $\gamma = \sqrt{b}$. Therefore, the authors of [9] present an accept-reject

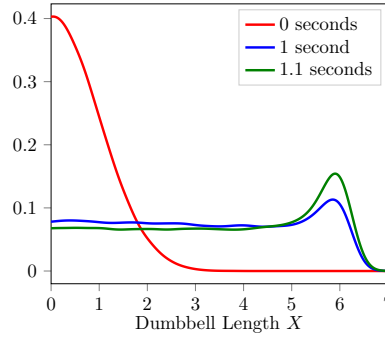


Figure 3.2: Kernel density estimation of the probability densities for the FENE model (3.8) after 0, 1 and 1.1 seconds, with bandwidth 0.2 and $N = 10^5$ particles. The initial condition is given by (3.10). There clearly forms a peak in the density near the edges, which will lower the acceptance ratio in the Euler-Maruyama particle sampler.

strategy to ensure that the Monte Carlo particles remain in the feasible domain. In general, the method increments time for each particle X_j^k of an ensemble $\mathcal{X}^k = (X_j^k)_{j=1}^J$ at time $t^k = k\delta t$ by

$$X_j^{k+1} = X_j^k + (\kappa(t^k)X_j^k - \frac{1}{2W_e}F(X_j^k))\delta t + \sqrt{\delta t}\xi^k, \quad \xi^k \sim \mathcal{N}(0, 1) \quad (3.11)$$

and accepts or rejects the new sample X_j^{k+1} . The sampler rejects a particle when $|X_j^{k+1}| > \alpha\sqrt{b}$ with $\alpha < 1$, to avoid very large spring forces that will result in rejections the next time step. The value of α should be close to 1 to get a consistent numerical scheme. The smaller the time step δt , the closer the particles may go to the boundaries since the variance of the Brownian motion is also smaller in that case. The authors of [9] therefore propose to take $\alpha = 1 - \sqrt{\delta t}$. Upon rejection of a particle X_j^{k+1} , the Euler-Maruyama step is repeated until the truncation accepts the particle.

Figure 3.2 shows the densities of the model after 0, 1 and 1.1 seconds. The distribution was obtained by kernel density estimation with a bandwidth of 0.01γ . The plots are insensitive to the exact value of the bandwidth as long as it is not orders of magnitude higher or lower. As time increases, there clearly forms a peak close to the maximal polymer length, so that many particle reside near the boundary of the domain of the polymer length. The equilibrium distribution originates from the interaction between the spring force that keeps the particles inside the domain and the velocity fields that elongates the polymers.

Comparison of matching strategies Figure 3.3 compares the three different matching strategies from previous chapter. The experiment takes three macroscopic states into account. The prior distribution is the exact (obtained by Euler-Maruyama) distribution after 1 second, and the states for the matching operator are the exact state variables

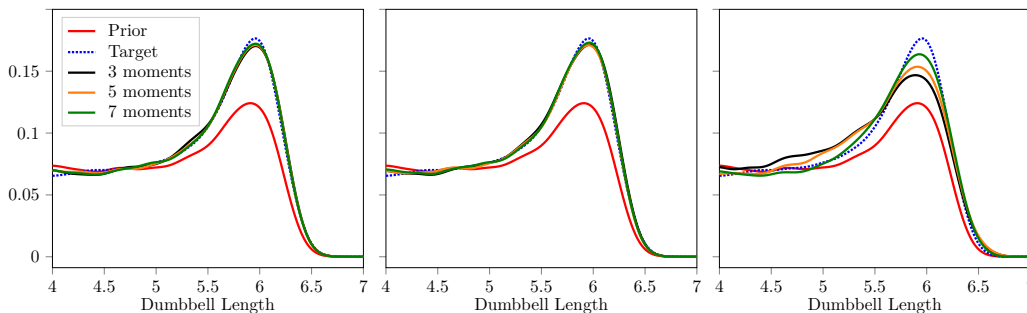


Figure 3.3: Matching results for a varying number of moments for KLD (left), L2D (middle) and L2N (right). Matching in L_2 norm clearly has the poorest accuracy, compared to matching in f -divergence.

after 1.1 seconds. The macroscopic states are the first three normalized even moments

$$R_l(x) = \left(\frac{x}{\gamma}\right)^{2l}, \quad l = 1 \dots 3. \quad (3.12)$$

Odd moment functions should yield zero in expectation because every transient distribution of the FENE model is symmetric around zero, because the initial condition is symmetric. There also is the zero-th state $R_0(x) = 1$ to make sure the matched density has unit mass.

Figure 3.3 shows that L2N-matching yields a density that lies far from the exact density, especially where they peak. L2N hence needs more moments than matching in f -divergence before it reaches an acceptable approximation of the exact density. Matching in f -divergence yields a very good approximation, even with only one state variable. This thesis will from now on only focus on relative-entropy matching as it is very accurate.

Moment accuracy for a varying number of state variables The second experiment again uses the exact distribution after 1 second as prior and takes the exact normalized even moments at time 1.1 seconds as extrapolated states. We now look at the relative error

$$\frac{|m_l - m_l^*|}{m_l^*} \quad (3.13)$$

between the moments m_l^* of the exact and those of the matched distribution m_l at 1.1 seconds. Figure 3.4 presents the experimental results, averaged over 20 i.i.d. runs, for the three matching strategies and a varying number of even moments (3.12). The results indicate that for a moment m_l with $l \leq L$, the error in moments is beneath the tolerance of the Newton-Raphson solver, indicating convergence of the iterative solver. For higher order moments however, the error suddenly increases very rapidly since the higher order moments are not taken into account while matching. The error for moments $l > L$ however decreases with increasing L , indicating that adding more moments to the matching operator has a positive impact on the accuracy of higher moments.

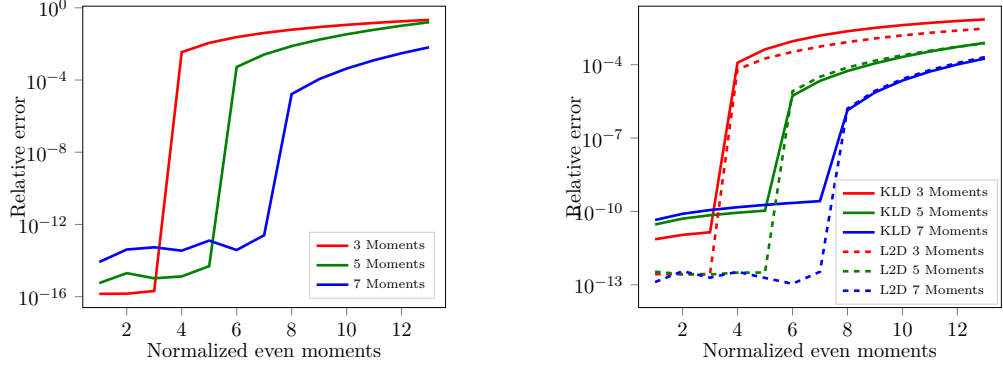


Figure 3.4: Relative error of even normalized moments as a function L for L2N (left) and matching in f -divergence (right). For $l \leq L$ the errors are below the Newton-Raphson tolerance, while for $l > L$ the errors decrease with increasing L .

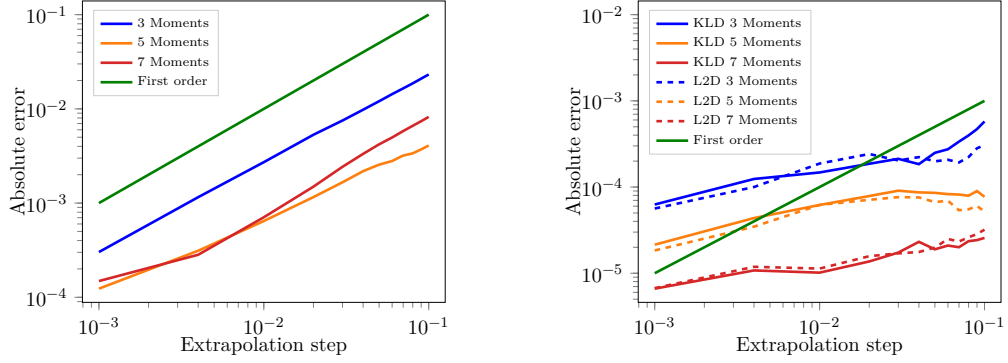


Figure 3.5: Relative error of the stress tensor as a function of the extrapolation time step Δt for L2N (left) and matching in f -divergence (right). L2N matching attains a convergence order of 1, while matching in f -divergence has a lower order.

Accuracy dependency on the extrapolation step Finally, in Figure 3.5 we look at how the matching error depends on the extrapolation time step Δt for a fixed microscopic step $\delta t = 2 \cdot 10^{-4}$ and a varying number of states L . By Theorem 1, the error of a well-behaved function should decrease linearly with Δt if this is the dominant term in the error bound. The following experiment considers the relative error of the stress tensor $|\tau(t) - \hat{\tau}(t)|/\tau(t)$ between the matched $\hat{\tau}(t)$ and the exact stress $\tau(t)$, with an extrapolation step $\Delta t \in [5\delta t, 500\delta t]$. The convergence of matching in f -divergence is slower than first order. The authors of [9] also came to this conclusion and further investigation is required. For fixed Δt , the error decreases with increasing L , due to the fact that the constant C_L decreases uniformly with increasing L . L2N does attain first order, confirming Theorem 1 but the errors are almost an order of magnitude lower for KLD and L2D matching compared to L2N.

Chapter 4

Linear slow-fast SDEs: Convergence & Stability

In the previous chapters, we focussed on the introduction of the micro-macro acceleration method, as it was already studied in [9]. In this thesis, we are interested in the efficiency of the method, the speed-up that we can attain for a given accuracy. This chapter is the start of the efficiency analysis of the micro-macro acceleration method. Since this speed-up will depend on the time-scale separation that is present in the system, we study in this chapter how matching performs on a two dimensional linear slow-fast equation

$$\begin{aligned}dX(t) &= (Y(t) - X(t))dt + D_x dW_x(t) \\dY(t) &= -\frac{1}{\varepsilon}Y(t)dt + \frac{1}{\sqrt{\varepsilon}}D_y dW_y(t)\end{aligned}\tag{4.1}$$

with a time-scale separation ε . The fast mode $Y(t)$ is independent from the slow mode X , while the latter depends on the former. In practice, we are only interested in the slow process $X(t)$, and we would hence like to only extrapolate states of $X(t)$. In this context, there are three questions that need to be answered. The first is about convergence. Theorem 1 states that the micro-macro acceleration algorithm converges if the number of state variables in X and Y goes to infinity. For linear processes however, we will show that only the mean of X as state variable is required for convergence. This is a new result and the main point of this chapter. A second fundamental question concerns stability: how large the extrapolation step can become before the micro-macro acceleration diverges to infinity. This has been extensively studied in [10] and this chapter gives a summary of the stability results. The last question is how many state variables of X and Y are required to have an as accurate simulation as possible. An a priori estimate of the number of state variables has also not been studied extensively in the literature and these results are also new. We discuss the selection of state variables for a linear scale-separated process in Chapter 5.

In Section 4.1 we start with an analytical decomposition of the joint density between the slow and fast variables, when the matching algorithm only uses slow state variables

for extrapolation. This decomposition forms the basis of Section 4.2, where analytical formulas are derived for matching with slow mean or slow mean and variance extrapolation with a Gaussian prior distribution. This section also contains numerical experiments confirming these theorems. In Section 4.3 we state and prove the new convergence theorem with only slow state variables, based on the analytical formulas of matching. Finally, in Section 4.4 we summarize stability of micro-macro acceleration.

4.1 A decomposition for matching with only slow state variables

In practice, we are usually only interested in the behaviour of the slow stochastic process $X(t)$, so it is useful to perform matching with only states of the slow process. The reasons are two-fold. First, it is intuitive to assume that the fast variables will quickly relax to an equilibrium distribution after only a few microscopic steps. This allows for healing of the fast density after matching with only slow states. The healing happens on faster time scales than the movement of the slow modes. If the fast modes can relax quickly, there is no stringent reason to include its state variables in the restriction operator. Second, adding more states increases the computational burden of matching and will result in a slower Newton-Raphson procedure. This is often the computational bottleneck and minimizing this when possible will yield a faster acceleration algorithm with the same accuracy of the slow values.

For relative-entropy matching there exists a decomposition formula that represents the complete microscopic matched distribution as a function of the matched marginal distribution of the slow variables, and a conditional distribution of the fast variables, given the slow. This decomposition effectively decouples the fast from the slow dynamics and is a useful result for further proofs on the linear SDEs, but the proof however holds for any n -dimensional process. The derivation that follows was taken from [10] and reworked in the notation of this text.

Suppose the joint density between the slow stochastic variables $\mathbf{X} \in \mathbb{R}^s$ and fast variables $\mathbf{Y} \in \mathbb{R}^f$ is given by $\rho(\mathbf{z})$ with $\mathbf{z} = (\mathbf{x}, \mathbf{y})^T \in \mathbb{R}^{s+f}$. Further assume that the restriction operator \mathcal{R}_L only contains state functions of the slow modes \mathbf{X} ,

$$\mathcal{R}_L(\mu) = (\mathbb{E}_\mu[R_0(\mathbf{X})], \mathbb{E}_\mu[R_1(\mathbf{X})], \mathbb{E}_\mu[R_2(\mathbf{X})], \dots, \mathbb{E}_\mu[R_L(\mathbf{X})]) = \mathbf{m} \in \mathbb{R}^{L+1}.$$

Also denote for shortness of notation $\rho^*(\mathbf{z}) = \mathcal{P}_L(\mathbf{m}, \rho)(\mathbf{z})$ the density of matching with (extrapolated) macroscopic states \mathbf{m} and prior distribution $\rho(\mathbf{z})$. For relative-entropy matching, formula (2.13) states that the matched density can be written as

$$\rho^*(\mathbf{z}) = \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{z}).$$

The Lagrange multipliers λ_l are the solution of the set of non-linear equations given by

$$\int_{\mathbb{R}^{s+f}} R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{z}) d\mathbf{z} = m_l. \quad (4.2)$$

Factorizing the joint density as $\rho(\mathbf{z}) = \rho(\mathbf{x})\rho(\mathbf{y}|\mathbf{x})$, into a marginal density of the slow variables $\rho(\mathbf{x})$ and a conditional distributions of the fast variables, given the slow $\rho(\mathbf{y}|\mathbf{x})$, gives a way of integrating out the fast variables

$$\begin{aligned} \int_{\mathbb{R}^{s+f}} R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{z}) d\mathbf{z} &= \int_{\mathbb{R}^s} \int_{\mathbb{R}^f} R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{y}|\mathbf{x}) \rho(\mathbf{x}) d\mathbf{y} d\mathbf{x} \\ &= \int_{\mathbb{R}^s} R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{x}) \int_{\mathbb{R}^f} \rho(\mathbf{y}|\mathbf{x}) d\mathbf{y} d\mathbf{x} \\ &= \int_{\mathbb{R}^s} R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{x}) d\mathbf{x}. \end{aligned}$$

Finally, we can reduce formula (4.2) to

$$\int_{\mathbb{R}^s} R_l(\mathbf{x}) \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{x}) d\mathbf{x} = m_l, \quad (4.3)$$

or said differently: the Lagrange multipliers corresponding to the matching problem with fast and slow variables are exactly the same as the multipliers for the slow matching problem only. An interesting consequence of this reduction is that the matched microscopic density $\rho^*(\mathbf{z})$ is also decomposable as the matched density of the marginal distribution of the slow variables $\rho^*(\mathbf{x})$ and the prior conditional density of the fast given the slow variables $\rho(\mathbf{y}|\mathbf{x})$:

$$\rho^*(\mathbf{z}) = \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{y}|\mathbf{x}) \rho(\mathbf{x}) = \rho^*(\mathbf{x}) \rho(\mathbf{y}|\mathbf{x}) \quad (4.4)$$

where the matched marginal distribution of the slow variables equals

$$\rho^*(\mathbf{x}) = \exp\left(\sum_{l=0}^L \lambda_l R_l(\mathbf{x})\right) \rho(\mathbf{x}).$$

Equation (4.4) is an important result in its own right and is also useful to deduce the exact formulas of relative-entropy matching in the context of linear equations with a Gaussian prior.

4.2 Analytic results for matching with a Gaussian initial distribution

If the initial distribution of a linear system of SDEs (4.1) is Gaussian, then all intermediate steps in the micro-macro acceleration algorithm are Gaussian too. This is a well-known result for the Euler-Maruyama microscopic time stepper, but the same is true for relative-entropy matching. If the prior distribution is Gaussian then the matched density turns out to be Gaussian too. This result is independent of whether only the slow mean, variance or both are extrapolated. This section contains a proof in case of mean-variance extrapolation. The proof is similar to mean-only extrapolation, stated in [10], and serves as an extension of that result. The proof relies heavily on decomposition (4.4) and another lemma stated first. The discussion ends with a numerical verification of these analytical results.

4.2.1 Minimizing relative entropy

The following lemma is a building block for the theorems concerning matching with a Gaussian prior. It concerns minimizing the relative entropy in the context of relative-entropy matching where the new distribution needs to have a predefined (extrapolated) mean and/or variance and where the prior is normally distributed. The objective is thus to find the minimizing distribution Q^* of the Kullback-Leibler divergence

$$\mathcal{H}(Q|P) = \int_{\mathbb{R}} q(\mathbf{x}) \ln \left(\frac{q(\mathbf{x})}{p(\mathbf{x})} \right) d\mathbf{x}$$

where P is a Gaussian distribution and Q needs to have certain moments fixed. The following proof only holds when the matched distribution needs to have a certain mean, but an extension to extrapolated mean and variance is straightforward.

Lemma 1. *Given a prior Gaussian distribution $P = \mathcal{N}(\mu, \Sigma)$ and a new mean μ^* , the minimizer Q^* of the constrained relative-entropy minimization problem*

$$\min_Q \mathcal{H}(Q|P) = \int_{\mathbb{R}} q(x) \ln \left(\frac{q(x)}{p(x)} \right) dx \quad \text{subject to} \quad \mathbb{E}[Q] = \mu^* \quad \text{and} \quad Q \ll P \quad (4.5)$$

is again a Gaussian distribution with mean μ^ and variance Σ .*

Proof. For simplicity of notation, we work with one-dimensional distributions and use the dummy variable x inside the integrals to suggest ‘slow’ variables. A generalization to more dimensions is straightforward. Using the fact that $p(x) \sim \mathcal{N}(\mu, \Sigma)$, we can expand the logarithm term to

$$\mathcal{H}(Q|P) = \int_{\mathbb{R}} q(x) \ln(q(x)) dx - \int_{\mathbb{R}} q(x) \ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{(x-\mu)^2}{2\sigma^2} \right) \right) dx. \quad (4.6)$$

The first term is the negative of the Shannon entropy of a distribution $q(x)$, also denoted by $-\mathcal{H}(Q)$. The second term in (4.6) can be further expanded using properties of the

logarithm

$$\mathcal{H}(Q|P) = -\mathcal{H}(Q) + \frac{1}{2} \ln(2\pi\sigma^2) \int q(x) dx + \frac{1}{2\sigma^2} \int_{\mathbb{R}} q(x)(x - \mu)^2 dx. \quad (4.7)$$

Since $q(x)$ is a probability density with mean μ^* we can finally write the relative entropy as

$$-\mathcal{H}(Q) + \frac{1}{2} \ln(2\pi\sigma^2) + \frac{1}{2\sigma^2} (\text{Var}[q] - (\mu - \mu^*)^2). \quad (4.8)$$

A basic result from information theory [16, 17] states that for a given variance σ_*^2 , the distribution that maximizes the Shannon entropy is exactly a Gaussian with this variance, regardless the mean. The value of the entropy then is $\mathcal{H}(Q) = \ln(\sigma_* \sqrt{2\pi e})$. We will first fix σ_* so that the first term in equation (4.8) is hence minimized by this Gaussian distribution. Plugging in the value of $\mathcal{H}(Q)$ reads

$$\mathcal{H}(Q|P) = \frac{1}{2} + \frac{1}{2} \ln\left(\frac{\sigma^2}{\sigma_*^2}\right) + \frac{\sigma_*^2}{2\sigma^2} + \frac{(\mu - \mu^*)^2}{2\sigma^2}. \quad (4.9)$$

and it minimizes the value of (4.8) over all distributions with variance σ_*^2 . Finally, we vary σ_* . The function (4.9) is convex in σ_* and attains its global minimum in $\sigma_* = \sigma$, thereby proving the lemma. \square

Remark 3. A similar lemma where the new distribution Q^* needs to have a certain mean and variance, besides only a predefined mean, is simpler to prove. The variance of Q^* then is already fixed so it is not necessary to minimize over the variance of Q . The minimizer is also a Gaussian with the predefined mean and variance.

4.2.2 Matching with extrapolated slow mean

Now all elements are in place to prove that relative-entropy matching with a Gaussian prior and extrapolated slow moments results in a distribution that is also Gaussian. The theorem here is specific for slow-mean extrapolation, but an analogous result holds for variance-only or mean-variance extrapolation, with more complicated formulas. The proof of the theorem can be found in [10], only the theorem is stated here. Next section contains a proof of mean-variance extrapolation.

Theorem 2. *Suppose P is the prior Gaussian distribution with mean μ covariance matrix Σ ,*

$$\mu = \begin{bmatrix} \mu_s \\ \mu_f \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_s & C \\ C^T & \Sigma_f \end{bmatrix},$$

where ‘s’ indicates the slow variables and ‘f’ the fast. The distribution Q^ that minimizes the Kullback-Leibler divergence, constrained to $\mathbb{E}[Q]_s = \mu_s^*$ is a normal distribution $\mathcal{N}(\mu^*, \Sigma)$ with mean $\mu^* = [\mu_s^*, \mu_f^*]^T$ where $\mu_f^* = \mu_f + C^T \Sigma_s^{-1} (\mu_s^* - \mu_s)$.*

The distribution obtained by relative-entropy matching with a Gaussian prior and extrapolated slow mean is again a normal distribution with the same covariance matrix but a different expression for the mean.

4.2.3 Matching with extrapolated slow mean and variance

A similar result holds when the restriction operator also contains the slow variance. The proof here is inspired on the proof on mean-only extrapolation in [10], but it is a new result since it does not appear in anywhere in the literature.

Theorem 3. *Suppose P is the prior Gaussian distribution with mean μ covariance matrix Σ ,*

$$\mu = \begin{bmatrix} \mu_s \\ \mu_f \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_s & C \\ C^T & \Sigma_f \end{bmatrix}.$$

The distribution Q^ that minimizes the Kullback-Leibler divergence, constrained to $\mathbb{E}[Q]_s = \mu_s^*$ and $\text{Var}[Q]_s = \Sigma_s^*$ is a normal distribution $\mathcal{N}(\mu^*, \Sigma^*)$ with parameter values*

$$\begin{cases} \mu_f^* = \mu_f + C^T \Sigma_s^{-1} (\mu_s^* - \mu_s) \\ C^{*T} = C^T \Sigma_s^{-1} \Sigma_s^* \\ \Sigma_f^* = \Sigma_f - C^T \Sigma_s^{-1} (\Sigma_s - \Sigma_s^*) \Sigma_s^{-1} C. \end{cases} \quad (4.10)$$

Proof. Denote again $z = (x, y)^T$. It is a well-known fact that we can decompose $\mathcal{N}_{\mu, \Sigma}$ as the product of the marginal normal density of the slow variables and a conditional normal density of the fast variables given the slow [10]

$$\mathcal{N}_{\mu, \Sigma}(z) = \mathcal{N}_{\mu_s, \Sigma_s}(x) \mathcal{N}_{\mu_{f|s}(x), \Sigma_{f|m}}(y).$$

with

$$\begin{cases} \mu_{f|s}(x) = \mu_f + C^T \Sigma_s^{-1} (x - \mu_s) \\ \Sigma_{f|s} = \Sigma_f - C^T \Sigma_s^{-1} C. \end{cases} \quad (4.11)$$

By Remark 3, the minimizer of the Kullback-Leibler divergence with predefined mean μ_s^* and variance Σ_s^* and prior a Gaussian is again a Gaussian with the given mean and variance, so

$$\mathcal{P}_L((\mu_s^*, \Sigma_s^*), \mathcal{N}_{\mu_s, \Sigma_s}) = \mathcal{N}_{\mu_s^*, \Sigma_s^*}.$$

As a consequence by (4.4) the full matched density reads

$$\mathcal{P}_L((\mu_s^*, \Sigma_s^*), \mathcal{N}_{\mu, \Sigma})(z) = \mathcal{N}_{\mu_s^*, \Sigma_s^*}(x) \mathcal{N}_{\mu_{f|s}(x), \Sigma_{f|s}}(y). \quad (4.12)$$

We would like that this matched density is also a Gaussian distribution with mean μ^* and variance Σ^* given as

$$\mu^* = \begin{bmatrix} \mu_s^* \\ \mu_f^* \end{bmatrix} \quad \Sigma^* = \begin{bmatrix} \Sigma_s^* & C^* \\ C^{*T} & \Sigma_f^* \end{bmatrix}.$$

Suppose there exists a similar factorization for the matched density as for the prior distribution:

$$\mathcal{P}_L((\mu_s^*, \Sigma_s^*), \mathcal{N}_{\mu, \Sigma})(z) = \mathcal{N}_{\mu^*, \Sigma^*}(z) = \mathcal{N}_{\mu_s^*, \Sigma_s^*}(x) \mathcal{N}_{\mu_{f|s}(x), \Sigma_{f|s}^*}(y) \quad (4.13)$$

Where the parameters $\mu_{f|s}^*(x)$ and $\Sigma_{f|s}^*$ are defined similarly as in equation (4.11):

$$\begin{cases} \mu_{f|s}^*(x) &= \mu_f^* + C^{*T}\Sigma_s^{*-1}(x - \mu_s^*) \\ \Sigma_{f|s}^* &= \Sigma_f^* - C^{*T}\Sigma_s^{*-1}C^*. \end{cases} \quad (4.14)$$

The first factor in equation (4.13) already equals the marginal normal density of the slow variables in (4.12), so that the second conditional factors also need to be equal. Equalling the mean and covariance matrices of the two second factors in (4.11) and (4.14) yields

$$\begin{cases} \mu_f + C^T\Sigma_s^{-1}(x - \mu_s) = \mu_f^* + C^{*T}\Sigma_s^{*-1}(x - \mu_s^*) \\ \Sigma_f - C^T\Sigma_s^{-1}C = \Sigma_f^* - C^{*T}\Sigma_s^{*-1}C^*. \end{cases} \quad (4.15)$$

The first equation is a linear polynomial in x , implying that $C^{*T} = C^T\Sigma_s^{-1}\Sigma_s^*$. Equality of the constant terms in the first equation then gives $\mu_f^* = \mu_f + C^T\Sigma_s^{-1}(\mu_s^* - \mu_s)$ and finally the second equation gives the value for $\Sigma_f^* = \Sigma_f - C^T\Sigma_s^{-1}(\Sigma_s - \Sigma_s^*)\Sigma_s^{-1}C$. If we use these values for the unknowns, then equation (4.12) defines a conditional factorization of a normal distribution, meaning that $\mathcal{P}_L([\mu_s^*, \Sigma_s^*]^T, \mathcal{N}_{\mu, \Sigma})(z)$ is also normally distributed with mean μ^* and covariance Σ^* . \square

4.2.4 Numerical experiments

The formulas for the mean and variance of the Gaussian distribution after matching with a Gaussian prior are straightforward to implement and make it possible to check against numerical experiments. This section contains several experiments on SDE (4.1) with mean and mean-variance extrapolation to see if they agree with the theoretical results given by Theorems 2 and 3.

Propagation of mean and variance with the Euler-Maruyama scheme

To verify the implementation of the micro-macro acceleration scheme for correctness over larger time scales, it is necessary to know how the mean and variance propagate through the K inner microscopic steps. This thesis exclusively uses the Euler-Maruyama scheme to perform these microscopic steps. The propagation formulas for mean and variance are relatively simple in this case. We will compute the propagation formulas for a general two-dimensional SDE

$$d \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} dt + \begin{bmatrix} D_x & 0 \\ 0 & D_y \end{bmatrix} dt \begin{bmatrix} dW_x \\ dW_y \end{bmatrix}.$$

A particle (X^n, Y^n) propagates through the linear SDE by

$$\begin{aligned} X^{n+1} &= (1 + a\delta t)X^n + b\delta tY^n + D_x\delta W_x^n, & \delta W_x^n &\sim \mathcal{N}(0, \delta t) \\ Y^{n+1} &= c\delta tX^n + (1 + d\delta t)Y^n + D_y\delta W_y^n, & \delta W_y^n &\sim \mathcal{N}(0, \delta t). \end{aligned} \quad (4.16)$$

There exists a useful property, called the *martingale* property, that states that the expectation of Brownian increments are zero. Put mathematically,

$$\mathbb{E}\left[\int_0^T (\phi(t)dW(t))dt\right] = 0, \text{ or } \mathbb{E}[dW(t)] = 0, \quad (4.17)$$

for any function $\phi(t)$ and end time T . Taking expectations from both sides and employing the martingale property gives how the means μ_x^n and μ_y^n propagate

$$\begin{aligned} \mu_x^{n+1} &= (1 + a\delta t)\mu_x^n + b\delta t\mu_y^n \\ \mu_y^{n+1} &= c\delta t\mu_x^n + (1 + d\delta t)\mu_y^n. \end{aligned} \quad (4.18)$$

Subtracting (4.18) from (4.16), squaring both sides and taking expectations finally gives the propagation of the variances σ_x^n , σ_y^n and the covariance $\sigma_{x,y}^n$

$$\begin{aligned} \sigma_x^{n+1} &= (1 + a\delta t)^2\sigma_x^n + b^2\delta t^2\sigma_y^n + D_x^2\delta t + 2b\delta t(1 + a\delta t)\sigma_{x,y}^n \\ \sigma_y^{n+1} &= (1 + d\delta t)^2\sigma_y^n + c^2\delta t^2\sigma_x^n + D_y^2\delta t + 2c\delta t(1 + d\delta t)\sigma_{x,y}^n \\ \sigma_{x,y}^{n+1} &= c\delta t(1 + a\delta t)\sigma_x^n + b\delta t(1 + d\delta t)\sigma_y^n + ((1 + a\delta t)(1 + d\delta t) + bc\delta t^2)\sigma_{x,y}^n. \end{aligned} \quad (4.19)$$

Equations (4.18) and (4.19) completely describe how the mean and variances of a Gaussian distribution propagate through one step of the Euler-Maruyama scheme with microscopic step size δt . Together with the formulas for matching with a Gaussian prior, we can simulate how the exact means and variances should propagate through micro-macro acceleration for linear SDEs. The following experiments test the numerics against these formulas.

Experiment 1: extrapolating the slow mean

In the first experiment, we plot the mean and variance for both the slow and fast variables, given by the micro-macro acceleration algorithm and the analytical expressions stated in (4.18) and (4.19). This experiment only extrapolates the slow mean before matching, such that theorem 2 gives the exact expression of the matching operator with a Gaussian prior. The variance of both variables should remain constant during matching.

The initial condition of the experiment is a Gaussian with mean $[1, 2]^T$ and the identity matrix for the covariance. The inner time stepper is a standard Euler-Maruyama scheme on the domain $[-6, 6] \times [-6, 6]$ with reflective boundary conditions and $N = 10^5$ particles. In practice, very few particles reach these boundaries as the distribution should converge to the origin, meaning the boundary condition has no impact. The numerical results for the mean are shown in Figure 4.1.

The blue lines are the exact mean and variance of SDE (4.1). The red lines represent the analytical solutions of the micro-macro acceleration method by Theorem 2 and (4.18), (4.19). The green points are the means and variances computed by the acceleration algorithm at the points in time when the numerical solution is known, i.e. after matching and each microscopic step. Furthermore, the orange line represents what the extrapolated

4.3. Convergence of the micro-macro acceleration method with slow mean extrapolation

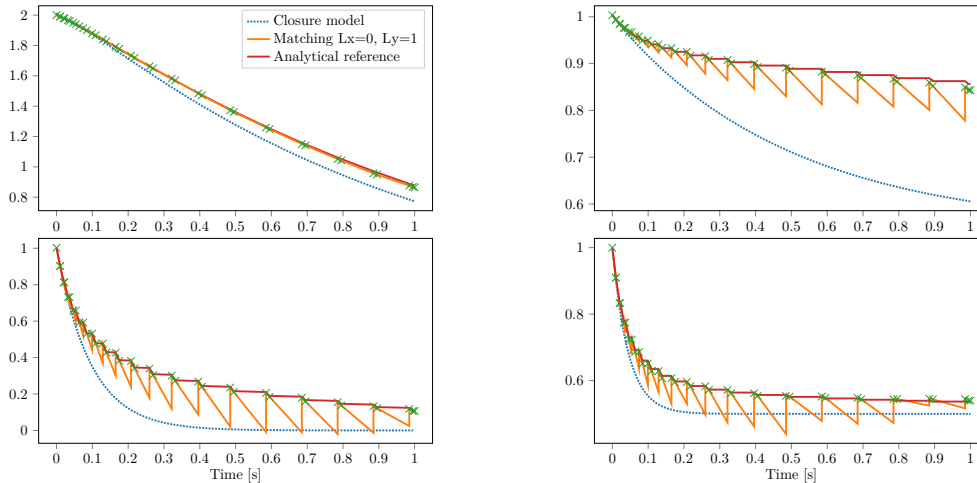


Figure 4.1: Mean (top left) and variance (top right) of the slow variable of (4.1) with mean-only extrapolation. (Bottom) the same plots for the fast mode. The blue lines are the exact solution, and the red lines represent the analytical formulas by Theorem 2 and (4.18),(4.19). The latter curve lies exactly on the orange lines, the numerical results of the micro-macro acceleration algorithm, confirming the analytical expressions.

means and variances would look like if all of these were extrapolated. The analytical solution lies almost exactly on the numerical results of the micro-macro acceleration algorithm at the known times, given by the green crosses. The discontinuities after extrapolation arise when a certain quantity is not part of the extrapolation operator and the matching operator corrects these values at each step by Theorems 2 and 3. For example, the red lines in the figures on the right, displaying the variance, are indeed flat during extrapolation as Theorem 2 predicts. The experiment confirms that the analytic expressions and formulas are indeed equal to the numerical results by the acceleration algorithm up to statistical noise, which confirms a correct implementation.

Experiment 2: extrapolating slow mean and variance

Figure 4.2 shows similar plots as the previous experiment but with extrapolated slow mean and variance. Both figures again demonstrate that the analytical results agree with the numerical output of the micro-macro acceleration algorithm.

4.3 Convergence of the micro-macro acceleration method with slow mean extrapolation

The main result of this chapter is the convergence of the micro-macro acceleration scheme on the linear SDE (4.1), with only slow-mean extrapolation. Theorem 4 differs from the main convergence Theorem 1, as the latter requires a hierarchy of macroscopic state

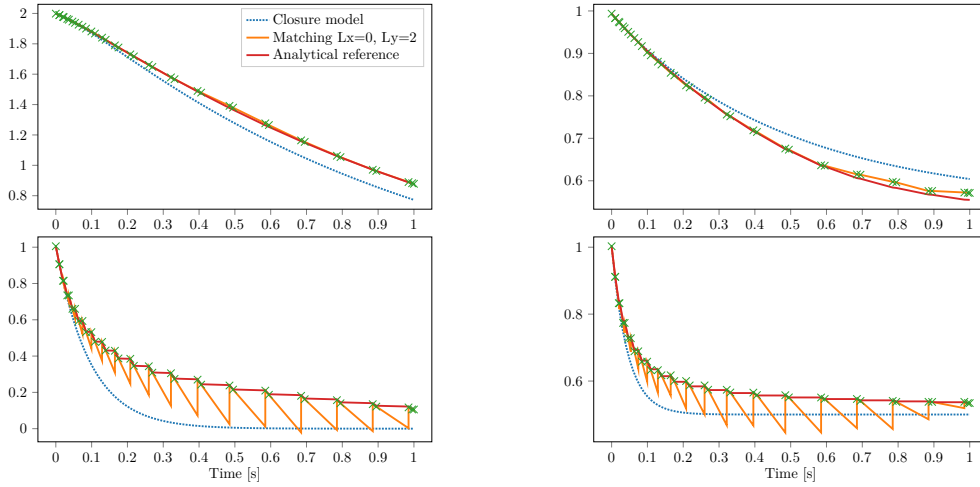


Figure 4.2: Mean (top left) and variance (top right) of the slow mode with mean-variance extrapolation. (bottom) the same plots for the fast mode. The numerical results of the micro-macro algorithm lie exactly on the analytical expression by Theorem 3.

variables to form a complete description of the density it represents. The slow mean never forms a complete description of the underlying density, as for example the variance is not known and nor are the fast variables. Theorem 4 is a new result and relies on Theorem 2. An extension of the proof to non-linear SDEs is non-trivial.

4.3.1 An iterative formula for slow mean-only extrapolation

The convergence proof relies on an iterative formula that defines how the complete mean and variance propagate through one full step of the micro-macro acceleration scheme. The formula is also fundamental to the derivation of a stability bound with slow-mean extrapolation in Section 4 of this chapter. The derivation here assumes only one Euler-Maruyama inner step of size δt for simplicity, but it can easily be extended to K inner steps. The derivation of the iterative formula was first done in [10] and is stated here with a slight change in notation to be consistent with the rest of the text.

Suppose that at time $t^n = n\Delta t$, the mean vector μ^n , the covariance matrix Σ^n and the matrix A in block-diagonal form are given:

$$\mu^n = \begin{bmatrix} \mu_s^n \\ \mu_f^n \end{bmatrix}, \quad \Sigma^n = \begin{bmatrix} \Sigma_s^n & C^n \\ (C^n)^T & \Sigma_f^n \end{bmatrix}, \quad A = \begin{bmatrix} A_s & 0 \\ 0 & A_f \end{bmatrix}.$$

The subscript ‘s’ again has the meaning of ‘slow’ variable. After one Euler-Maruyama step, a particle X^n propagates as

$$X^{n,1} = (I + A\delta t)X^n + \sqrt{\delta t}B\xi^n, \quad \xi^n \sim \mathcal{N}(0, 1),$$

so that, by taking expectations and employing the martingale property (4.17),

$$\mu^{n,1} = (I + A\delta t)\mu^n, \quad \Sigma^{n,1} = (I + \delta tA)\Sigma^n(I + \delta tA)^T + \delta tBB^T$$

4.3. Convergence of the micro-macro acceleration method with slow mean extrapolation

Extrapolating the slow mean yields

$$\mu_s^{n+1} = \mu_s^n + \frac{\Delta t}{\delta t}(\mu_s^{n,1} - \mu_s^n) = (I + \Delta t A)\mu_s^n.$$

By Theorem 2 we know that the covariance matrix Σ^n stays constant during matching, and that the fast mean is given by

$$\mu_f^{n+1} = (I_f + \delta t A_f)\mu_f^n + (\Delta t - \delta t)(C^{n,1})^T(\Sigma^{n,1})^{-1}A_s\mu_s^n.$$

Bundling the propagation of the complete mean vector in one equation finally gives,

$$\mu^{n+1} = \left(I_{s+f} + \begin{bmatrix} \Delta t A_s & 0 \\ (\Delta t - \delta t)C_{n,1}^T(\Sigma^{n,1})^{-1}A_s & \delta t A_f \end{bmatrix} \right) \mu^n \quad (4.20)$$

4.3.2 Convergence Theorem

The following theorem shows convergence of the micro-macro acceleration scheme on a linear SDE with only slow-mean extrapolation.

Theorem 4. *Given a linear SDE with a normal initial distribution, and consider the micro-macro acceleration algorithm with relative-entropy matching and only slow-mean extrapolation. Also fix an end time $T > 0$. Denote by \mathbf{X}^t the exact distribution of the linear SDE at time $t \in [0, T]$, and by $\mathbf{X}^{n_{\Delta t}(t)}$ the distribution obtained by the micro-macro acceleration scheme at that same time, where $n_{\Delta t}(t) = t/\Delta t$. Then the Kullback-Leibler divergence $\mathcal{H}(\mathbf{X}^{n_{\Delta t}(t)}|\mathbf{X}^t)$ converges uniformly to zero over the interval $[0, T]$ as δt goes to zero and Δt goes to δt ,*

$$\lim_{\delta t \rightarrow 0} \lim_{\Delta t \rightarrow \delta t} \max_{t \in [0, T]} \mathcal{H}(\mathbf{X}^{n_{\Delta t}(t)}|\mathbf{X}^t) = 0. \quad (4.21)$$

As a consequence, micro-macro acceleration distribution also converges uniformly to the exact distribution in the same limits.

Proof. If the initial condition of a linear equation is Gaussian, all intermediate distributions of the exact solution, the Euler-Maruyama method and the micro-macro acceleration algorithm are Gaussian too. By a well known expression [14], the Kullback-Leibler divergence between the approximation of the micro-macro acceleration scheme and the exact distribution reads

$$\begin{aligned} \mathcal{H}(\mathbf{X}^{n_{\Delta t}(t)}|\mathbf{X}^t) &= \frac{1}{2} \left(\ln \left(\frac{|\Sigma^t|}{|\Sigma^{n_{\Delta t}(t)}|} \right) - s - f + \text{trace}((\Sigma^t)^{-1}\Sigma^{n_{\Delta t}(t)}) \right) \\ &\quad + \frac{1}{2}(\mu^t - \mu^{n_{\Delta t}(t)})^T(\Sigma^t)^{-1}(\mu^t - \mu^{n_{\Delta t}(t)}), \end{aligned}$$

where μ^t and Σ^t are the mean and variance of the exact solution at time t . Likewise, $\mu^{n_{\Delta t}(t)}$ and $\Sigma^{n_{\Delta t}(t)}$ are the mean and variance of the approximation by the micro-macro

acceleration scheme at time t . First, we fix $\delta t \leq \Delta t$ and expand iteration (4.20) as

$$\begin{aligned}
 \mu_s^{n_{\Delta t}(t)} &= (I_s + A_s \Delta t)^{n_{\Delta t}(t)} \mu_s^0 \\
 \mu_f^{n_{\Delta t}(t)} &= (I_f + \delta t A_f) \mu_f^{n_{\Delta t}(t)-1} + (\Delta t - \delta t) (C^{n_{\Delta t}(t)-1,1})^T (\Sigma^{n_{\Delta t}(t)-1,1})^{-1} A_s \mu_s^{n_{\Delta t}(t)-1} \\
 &= (I_f + A_f \delta t)^{n_{\Delta t}(t)} \mu_f^0 + (\Delta t - \delta t) \sum_{k=0}^{n_{\Delta t}(t)-1} (I_f + A_f \delta t)^{n_{\Delta t}(t)-k-1} (C^{k,1})^T (\Sigma^k)^{-1} A_s \mu_s^k.
 \end{aligned} \tag{4.22}$$

As Δt decreases to δt , $n_{\Delta t}(t)$ will increase to $n_{\delta t}(t)$ and the number of terms in the sum will remain finite. The contribution of the large sum then becomes zero. As a result, the mean vector $\mu^{n_{\Delta t}(t)}$ uniformly approaches his respective mean $\mu^{n_{\delta t}(t)}$ of the Euler-Maruyama scheme. Similarly, the variance stays constant during matching, thus as $n_{\Delta t}(t)$ goes to $n_{\delta t}(t)$, the variances by the micro-macro acceleration scheme converge uniformly to the variance $\Sigma^{n_{\delta t}(t)}$ of the Euler-Maruyama scheme. The expression in (4.21) hence reduces to

$$\begin{aligned}
 \lim_{\delta t \rightarrow 0} \max_{t \in [0, T]} & \frac{1}{2} \left(\ln \left(\frac{|\Sigma^t|}{|\Sigma^{n_{\delta t}(t)}|} \right) - s - f + \text{trace}((\Sigma^t)^{-1} \Sigma^{n_{\delta t}(t)}) \right) \\
 & + \frac{1}{2} (\mu^t - \mu^{n_{\delta t}(t)})^T (\Sigma^t)^{-1} (\mu^t - \mu^{n_{\delta t}(t)}),
 \end{aligned}$$

which is the Kullback-Leibler divergence between the Euler-Maruyama scheme and the exact solution. The latter expression converges uniformly to zero because the mean and variance of the Euler-Maruyama method converges to their respective exact values, as δt decreases to zero. This concludes the proof. \square

Theorem 4 is quite remarkable since it does not require the number of state variables to diverge to infinity at all, just the slow mean is enough. The downside is that it only holds for linear SDEs with a Gaussian initial as the proof relies heavily on iteration (4.20). At the moment, no proof exists on convergence with slow mean and variance extrapolation, because the formulas for matching in this setting (4.10) are a lot more involved.

4.4 Stability of micro-macro acceleration

Besides convergence of a numerical method, it is also important to know what the largest step size is that one can take before the numerical scheme diverges to infinity. This concept is called stability and the authors of [10] perform a complete analysis of stability for the micro-macro acceleration scheme. Stability is also an important aspect of the efficiency analysis of the micro-macro acceleration scheme, which is the topic of this and the following three chapters. This section serves as a summary of the results in [10], with one small additional experiment. In the context of micro-macro acceleration, there are effectively two time steps at play: the microscopic stepper δt and the extrapolation Δt . A stability analysis for both steppers is required.

4.4.1 Stability of the Euler-Maruyama scheme with additive noise

Stability for ODEs and PDEs is usually studied on a linear test equation since any non-linear ODE or PDE can be linearised at every time step in a first-order approximation. In the context of SDEs there is, however, no clear connection between non-linear and linearised SDEs to study stability [10]. Nevertheless, we can study a stochastic linear test equation in its own right and derive stability bounds on the time step. We will consider a general linear test equation, where the matrix A is again decomposed in a block-diagonal form

$$d\mathbf{X}(t) = A\mathbf{X}(t)dt + Bd\mathbf{W}(t), \quad A = \begin{bmatrix} A_s & 0 \\ 0 & A_f \end{bmatrix}. \quad (4.23)$$

The process governed by (4.23) has a normal invariant distribution with mean 0 and a finite variance when A is negative definite. A numerical scheme is stable in this context when the intermediate distributions \mathcal{P}^n at times $t^n = n\delta t$ also converge to some invariant distribution \mathcal{P}^∞ as time increases, not necessarily equal to the invariant distribution of the continuous SDE. For instance, for the Euler-Maruyama scheme we can derive the stability bound by propagating a particle X^n

$$X^{n+1} = (I_{s+f} + \delta t A)X^n + \sqrt{\delta t}B\xi^n, \quad \xi^n \sim \mathcal{N}(0, 1) \quad (4.24)$$

such that by taking expectations and employing the martingale property, the mean μ^n and variance Σ^n at time t^n

$$\begin{aligned} \mu^n &= (I_{s+f} + \delta t A)^n \mu^0 \\ \Sigma^n &= (I_{s+f} + \delta t A)^n \Sigma^0 (I_{s+f} + \delta t A^T)^n + \sum_{k=1}^n (I_{s+f} + \delta t A)^k B B^T (I_{s+f} + \delta t A^T)^{n-k}, \end{aligned} \quad (4.25)$$

where I_{s+f} is the $s + f$ -dimensional unit matrix. The Euler-Maruyama scheme is stable when these quantities remain finite as $n \rightarrow \infty$. This is the case when the eigenvalues of $I_{s+f} + \delta t A$ lie inside the unit ball centred at the origin, $\mathcal{B}(0, 1)$.

For instance, consider the time-scale separated linear equation (4.1) with $\varepsilon = 0.1$. The smallest eigenvalue of A is $-1/\varepsilon$ so the stability bound is $\delta t \leq 0.2$. Figure 4.3 shows the slow distribution after $T = 30$ for different microscopic steps δt and computed with $N = 10^5$ particles. For $\delta t < 0.2$ the numerical distribution approximates the exact invariant distribution, obtained by computing (4.25) up to a tolerance of 10^{-6} . When $\delta t > 0.2$, the variance diverges to infinity so the numerical scheme is unstable.

4.4.2 Stability of micro-macro acceleration with mean extrapolation

Stability of the micro-macro acceleration algorithm works in the same setting with additive noise and a Gaussian initial condition since then there exist analytical formulas for matching 2. Stability again means that the intermediate distributions computed by the micro-macro acceleration algorithm converge to an invariant distribution, or put mathematically $\mathcal{P}_t \rightarrow \mathcal{P}_\infty$ as $t \rightarrow \infty$. The goal of acceleration is that Δt can be much

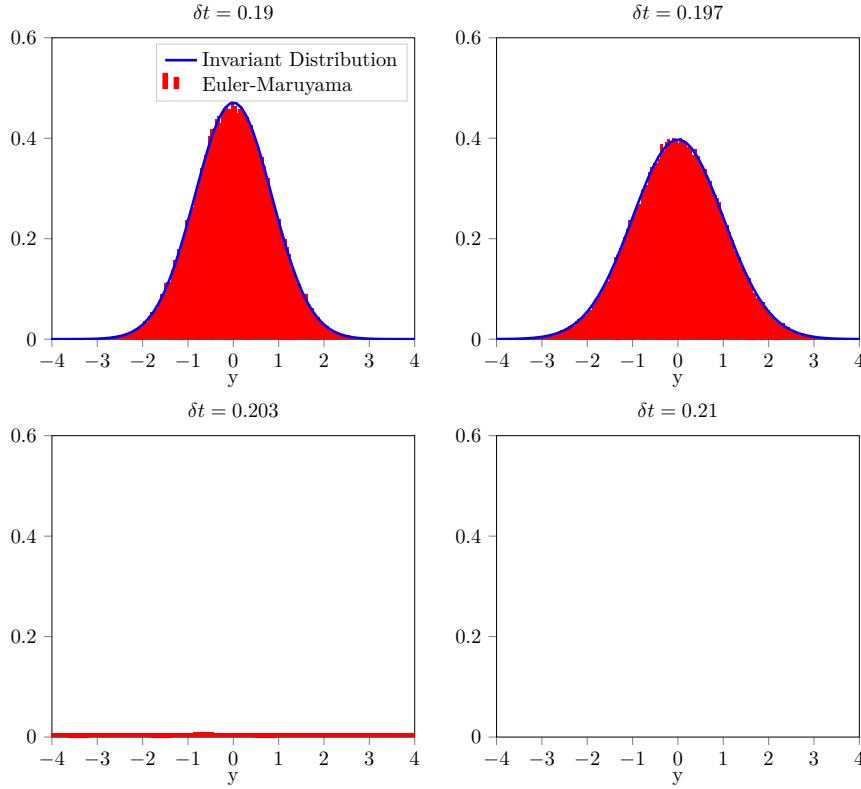


Figure 4.3: Invariant distributions of the slow mode of the Euler-Maruyama scheme (4.25) for linear SDE (4.1) for different step sizes δt with end time $T = 30$ seconds. The stability of the method clearly breaks if $\delta t > 0.2$.

larger than the microscopic time step δt while still having a stable numerical scheme. A very attractive property of micro-macro stability is that the stability bound for Δt only relies on the slow eigenvalues in the case of mean-only extrapolation. A large spectral gap is hence favourable for stable extrapolation. The derivation demonstrates stability for mean-only extrapolation while next section discusses mean-variance extrapolation. The derivations were taken from [10] but adapted to the notation of this thesis.

We know from previous section that the mean vector propagates as

$$\mu^{n+1} = \left(I_{s+f} + \begin{bmatrix} \Delta t A_s & 0 \\ (\Delta t - \delta t)(C^{n,1})^T (\Sigma^{n,1})^{-1} A_s & \delta t A_f \end{bmatrix} \right) \mu^n \quad (4.26)$$

through the micro-macro acceleration scheme and that the variance stays constant during matching. The mean vector is thus stable when the time-variant matrix D^n has all eigenvalues within the unit circle. As D^n is upper block diagonal, its eigenvalues are union of the eigenvalues of $I_f + \delta t A_f$ and $I_s + \Delta t A_s$. The fast components are stable when the Euler-Maruyama scheme is stable and the extrapolation is stable when $\text{spec}(I_s + \Delta t A_s) \in \mathcal{B}(0, 1)$. The fast and slow components are hence effectively decomposed.

For instance, consider again the test equation (4.1) with a finite time-scale separation ε . The eigenvalues of A are -1 and $-1/\varepsilon$ such that the Euler-Maruyama scheme is stable when $\delta t \leq 0.2$. By condition (4.26) the acceleration algorithm is stable when $\Delta t \leq 2$, which is a factor of 10 larger than the microscopic integrator. As ε decreases this spectral gap will become larger but the extrapolation in the micro-macro acceleration scheme will still be stable independent of ε , i.e. $\Delta t \leq 2$.

Crossing the stability boundary A first experiment shows the influence of crossing the stability of extrapolation with $\varepsilon = 0.1$. Figure 4.4 displays the distributions of the micro-macro acceleration method after $T = 300$ seconds, as an approximation to the invariant distributions. On the Figures we indeed see that the micro-macro acceleration scheme nicely reaches the invariant distribution of the Euler-Maruyama scheme for $\Delta t \leq 2$, while for $\Delta t > 2$ the adaptive strategy is turned on. The adaptive strategy halves Δt when a matching failure occurs and increases it with a factor of 1.2 otherwise. The maximum number of Newton-Raphson iterations is 8. It happens that the mean of the extrapolation step size is always below the stability boundary in these experiments although there is no theoretical ground for this behaviour [10]. The exact mean of Δt depends heavily on simulation parameters and such an analysis is not in the scope of this thesis.

Independence of ε As a second experiment, Figure 4.5 shows that for a different value of ε , here $\varepsilon = 0.01$, the macroscopic stability bound is exactly the same as for $\varepsilon = 0.1$ and that the adaptive procedure is only triggered when $\Delta t_{\max} \geq 2$. The end time again is $T = 300$ to make sure that the distributions have reached the invariant regime. When $\Delta t_{\max} \leq 2$, the micro-macro acceleration algorithm attains the invariant distribution of the Euler-Maruyama scheme while for larger step sizes, the micro-macro acceleration algorithm is unstable, confirming the stability bound (4.26) with slow mean extrapolation.

4.4.3 Stability with extrapolated mean and variance

There also exist analytical formulas for the stability bound when extrapolating slow mean and variance, proven in [10]. We will only state the stability criterion and perform a numerical experiment as a deep stability analysis is not the focus of this text.

In the context of mean-variance extrapolation, the mean vector also evolves through a time dependent linear system

$$\mu^{n+1} = D^n \mu^n. \quad (4.27)$$

The covariance matrix can however not remain constant anymore. The authors of [10] were able to prove that the slow covariance matrix propagates through one step of the micro-macro acceleration schemes by

$$\Sigma_s^{n+1} = (I_{s^2} + \Delta t(A_s \oplus A_s + \delta t(A_s \otimes A_s))). \Sigma_s^n + \delta t(BB^T)^s \quad (4.28)$$

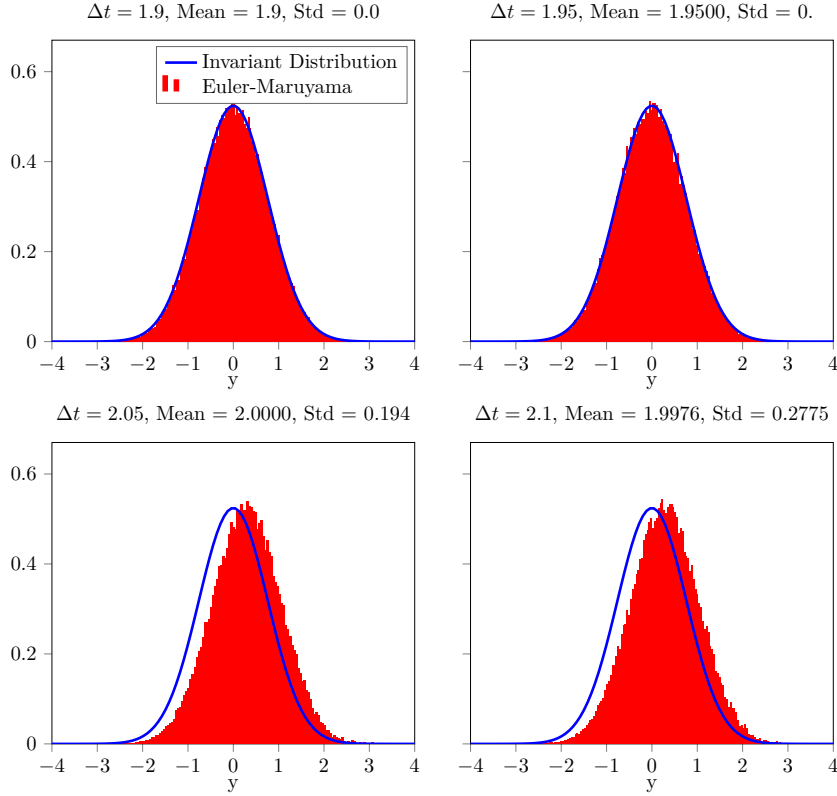


Figure 4.4: Histograms of the slow mode, computed by the micro-macro acceleration scheme with mean-only extrapolation, on the slow-fast model (4.1) with $\varepsilon = 0.1$ after $T = 300$ seconds, with adaptive time stepping, for several values Δt_{\max} . When $\Delta t_{\max} \geq 2$, stability breaks and the adaptivity is activated due to matching failures.

where the operator $A.B$ denotes the inverse vectorization of $\text{Avec}(B)$, when B is a square matrix. The vectorization operator $\text{vec}(A)$ on a matrix puts the columns of A vertically on top of each other. Expression (4.28) also makes use of the Kronecker product (\otimes) and the Kronecker sum (\oplus), which are defined in [30].

Stability of the micro-macro acceleration scheme hence depends on the spectrum of the matrix $L_s^{\delta t} = A_s \oplus A_s + \delta t(A_s \otimes A_s)$ and micro-macro acceleration is stable when $\text{spec}(I_{s^2} + \Delta t(L_s^{\delta t})) \subset \mathcal{B}(0, 1)$. Note that, in contrast to mean-only extrapolation, the stability bound now depends on the microscopic step size δt . If $\delta t \ll \rho(A_s)$, the largest negative eigenvalue of A_s , the term $\delta t(A_s \otimes A_s)$ acts as a small perturbation on the matrix $I_{s^2} + \Delta t(A_s \oplus A_s)$, which describes the extrapolation of the covariance matrix. The perturbation may change the exact stability bound of the micro-macro acceleration algorithm slightly by making it larger or smaller than the macroscopic stability requirement. The authors of [10] investigate the eigenvalues of $L_s^{\delta t}$ further but this is out of scope for this text. We end this chapter with a numerical example showing

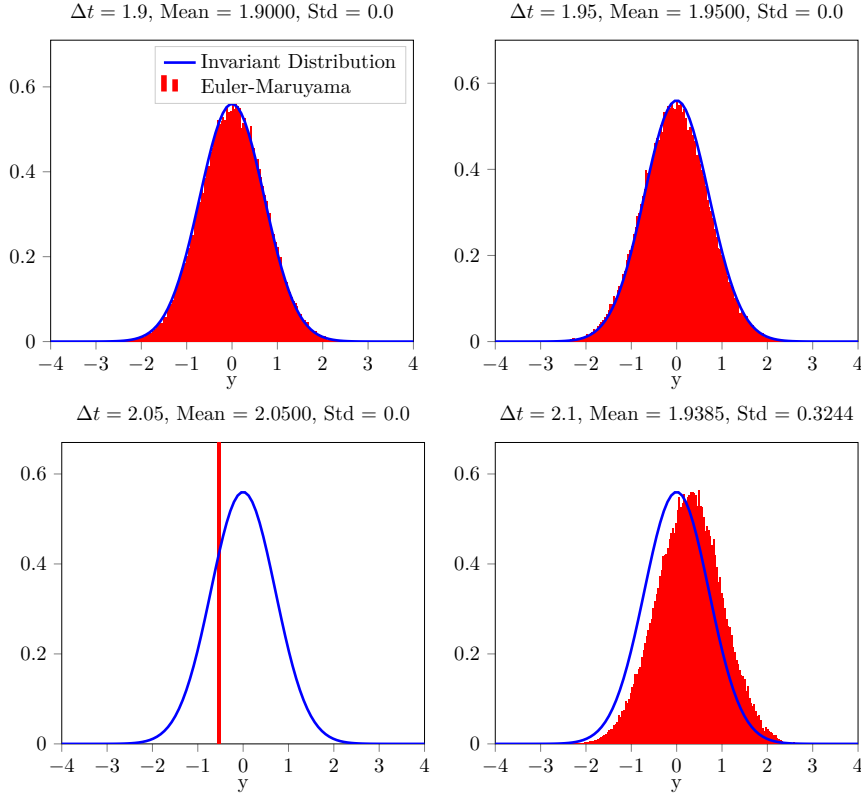


Figure 4.5: Histograms by the micro-macro acceleration scheme with mean-only extrapolation, on the slow-fast model (4.1) with $\varepsilon = 0.1$ after $T = 300$ seconds, with adaptive time stepping, for several values Δt_{\max} . The adaptivity activates when the stability breaks, i.e. $\Delta t_{\max} > 2$, indicating that the stability threshold is independent from ε .

the effect of the microscopic perturbation.

Numerical Experiment Take again the linear test system (4.1) with $\varepsilon = 0.1$ so that $A_s = -1$. The stability bound for mean-variance extrapolation requires that $|1 + \Delta t(-2 + \delta t)| \leq 1$ which is equivalent to $\Delta t \leq \frac{2}{2-\delta t}$. For any δt , the perturbation ensure that the micro-macro stability bound is larger than the deterministic bound for the variance. Figure 4.6 depicts the effect of the perturbation for $\delta t = \varepsilon = 0.1$ after $T = 210$ seconds, with stability bound $\Delta t \leq 1.053$. For $\Delta t \leq 1.053$ the numerical invariant distribution of the micro-macro acceleration scheme approximates the invariant distribution of the Euler-Maruyama scheme very well. When Δt crosses the stability boundary, the adaptive strategy activates due to matching failures. The maximum number of Newton iterations in this experiment is also 8.

Conclusion In this chapter, we proved a new convergence result of the micro-macro acceleration algorithm on a linear scale-separated process wit only slow-mean extrapolation.

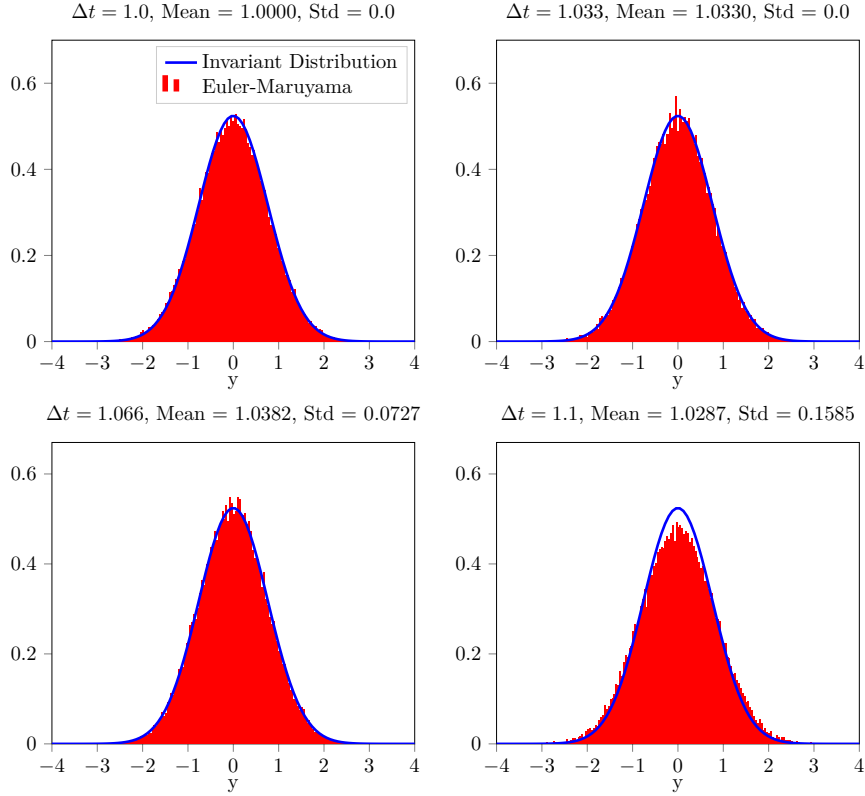


Figure 4.6: Histograms of the slow mode, by the micro-macro acceleration algorithm with mean-variance extrapolation of the linear system (4.1) with $\varepsilon = 0.1$ after $T = 210$ seconds for several values of Δt_{\max} , below and above the stability bound. Below the stability bound, the micro-macro acceleration scheme approximates the invariant distribution of the Euler-Maruyama scheme well, while above the stability bound, the step-size adaptivity activates. The averaged Δt_{\max} is always beneath the stability bound of 1.033.

Theorem 4 is an important result in the light of efficiency, since the number of states doesn't need to tend to infinity. In Section 4.4, we presented a summary of the stability results, a second milestone in the efficiency analysis. The stability analysis gives us the largest extrapolation time step we can take before the micro-macro acceleration scheme diverges to infinity. Chapter 5 takes the next step and presents a detailed study on how many macroscopic state variables are required for an accurate simulation for linear processes, and also considers a non-linear example. Chapter 6 then fixes the number of state variables for a linear driven process and studies how large the extrapolation step can be, before existing approximate macroscopic models become more accurate. Finding a good number of state variables and the maximal extrapolation time step for a given accuracy is the goal of the efficiency analysis. Chapter 7 ends with two practical examples where the micro-macro acceleration method outperforms existing approximate macroscopic models in accuracy and takes larger steps than the microscopic simulation.

Chapter 5

Effect of the choice of macroscopic state variables

An important parameter when using the micro-macro acceleration scheme is the number of macroscopic state functions L to extrapolate and match with. Choosing an inadequate number of states can be devastating for the performance of the algorithm. If L is too low, the state functions may not capture most of the dynamics of the probability density. If L is too high however, the computational cost increases rapidly due to the computation of the Jacobian in the Newton-Raphson procedure. The micro-macro acceleration algorithm is very inefficient in this case. If L is too high, the extrapolation stage may even introduce small modelling errors that can have a big impact with large extrapolation time steps. More details on this later in Section 5.2 on this matter.

Section 5.1 starts with a short explanation of the so called Fokker-Planck equation, which is a PDE representation of a general stochastic process (1.1). We also discuss a closure method for linear SDEs that serves as a reference solution for the remainder of this chapter. In Section 5.2, we investigate the effect of macroscopic state selection for a linear scale-separated SDE, where we choose the state variables as moments of the stochastic process. Finally, in Section 3 we look at different states hierarchies for the non-linear FENE-dumbbells problem.

5.1 The Fokker-Planck equation and closure relations

The exact intermediate distributions of the linear SDE (4.1) are almost never readily available, unless the initial condition is Gaussian. To estimate the accuracy of the micro-macro acceleration algorithm, it is necessary to compare the numerical results with different methods that simulate the same problem. This chapter considers a deterministic simulation of the Fokker-Planck equation and simulation of a closure model for the mean vector and covariance matrix. Since the two methods agree up to their respective discretization errors, any of the two can serve as an accurate substitution of the exact continuous solution.

The Fokker-Planck equation The Fokker-Planck equation associated to an SDE of the form (1.1) is a deterministic partial differential equation (PDE) that describes how the probability density $\rho(\mathbf{x}, t)$ evolves over time, under influence of the SDE. The probability density function is defined such that for any region $A \subset G$

$$\mathbb{P}(\mathbf{X}(t) \in A) = \int_A \rho(\mathbf{x}, t) d\mathbf{x}.$$

For a general n -dimensional SDE, the Fokker-Planck equation reads

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \nabla \cdot (a(\mathbf{x}, t)\rho(\mathbf{x}, t)) = \frac{1}{2} \sum_{i,j=1}^n \frac{\partial^2}{\partial x_i \partial x_j} (b_{i,j}(\mathbf{x})^2 \rho(\mathbf{x}, t)). \quad (5.1)$$

Simulating this equation instead of the stochastic representation gives equivalent results, but the computational cost of simulating the PDE directly increases exponentially with the dimensionality. This is one of the reasons to consider a Monte Carlo simulation of the SDE in the first place.

When simulating PDE (5.1), it is important to ensure that the numerical solution also has unit mass. This property could be lost with many simulations techniques. Hence, to discretize (5.1), we consider a finite volume approach with standard central differences for the second order term. Appendix A contains the complete mathematical derivation of the finite volume scheme for the two-dimensional linear Fokker-Planck equation, together with an order test to verify the correctness.

Closure relations For a general SDE (1.1) there also exists a (possibly infinite) system of ODEs that describe the evolution of all the moments $\mathbb{E}[m_n(\mathbf{X})]$, where $m_n(x) = x^n$, $n \in \mathbb{N}$ as a function of time. One can derive such a system by first writing down the evolution equation for $\mathbb{E}[\mathbf{X}]$, which by the martingale property reads

$$d\mathbb{E}[\mathbf{X}] = \mathbb{E}[a(\mathbf{X}, t)]dt.$$

Usually, the right hand side can be reduced to an expression of higher moments of \mathbf{X} . For an accurate simulation we need an evolution law for these higher moments as well, which will induce even higher moments in their right hand side, and so on. In practice, it is not feasible to simulate the infinite system of moment equations. However, as higher and higher moments add less information of the exact distribution, we can simulate only the evolution laws for the first M moments and make some approximations. In the case of a linear SDE, it is possible to prove that only the evolutions for the mean and covariance are required, as no higher order moments pop up in the derivation of the closure relations. Appendix B contains a derivation of the closure relations for a general n -dimensional linear SDE.

Comparison of both methods Figure 5.1 depicts the means and variances of the slow and fast variables for a Fokker-Planck implementation on a $[-6, 6] \times [-6, 6]$ grid with 48 cells in each direction and reflective boundary conditions. It also depicts the simulation

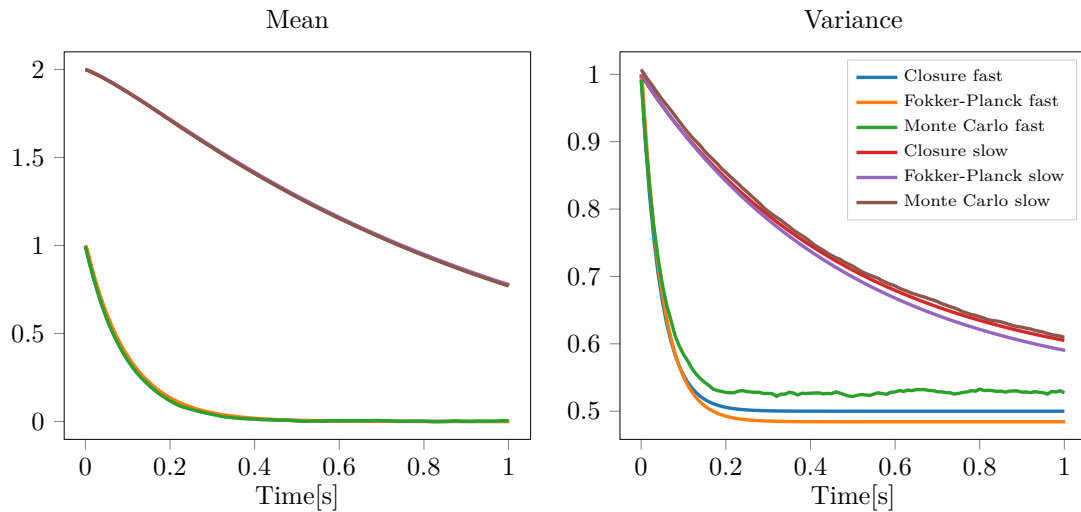


Figure 5.1: Means (Left) and variances (Right) of the fast and slow modes using the deterministic Fokker-Planck equations, the closure relations and a standard microscopic Euler-Maruyama method. The three methods agree up to their respective discretization errors.

of the closure relations with a step size of 0.1 and the result of an Euler-Maruyama simulation. The initial condition is a normal distribution with mean $[1, 2]$ and unit covariance matrix. The three methods agree up to their respective discretization errors, meaning that we can substitute any of these as exact solution in the efficiency plots. The blue lines in the experiments from Section 4.3 represent the simulation of the closure relations as substitute of the exact solution.

5.2 Estimating the number of moments for linear SDEs

A linear SDE, for instance (4.1), with a Gaussian initial condition is completely described by its mean and covariance. For a two-dimensional process, these are five parameters. In the context of slow-fast processes, however, we would like to extrapolate only the slow mean and slow variance, as the slow marginal distribution is completely described by these two moments. Adding more moments is not necessary, for three reasons. First, adding moments increases the computational cost of matching. Second, by the stability condition [10], the extrapolation of the fast moments is bounded by the stiff mode of the process, resulting in an expensive time stepper. Third, if ε is small, the fast modes should equilibrate quickly to their conditional equilibrium, given the slow modes, which renders extrapolating fast modes obsolete. The only upside of extrapolating fast modes is a possible increase in accuracy, although this is not visible.

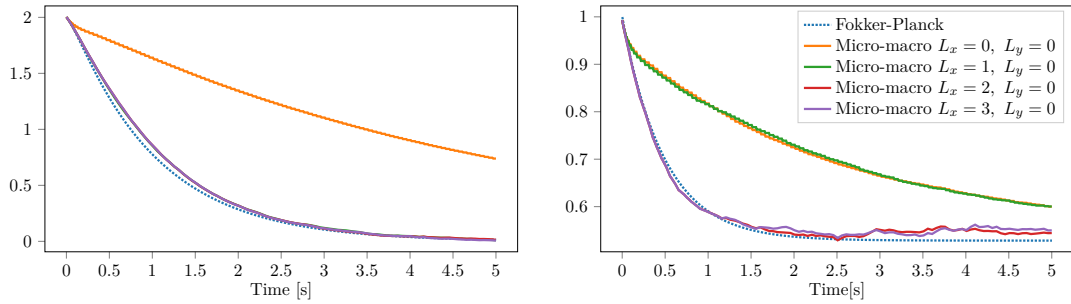


Figure 5.2: The mean (left) and variance (right) of the slow variable for different numbers of extrapolated slow moments L_x . For $L_x = 1$ the mean of the micro-macro acceleration algorithm follows the exact mean well and adding moments does not improve its accuracy. Similarly, two slow moments are required for the slow variance.

Effect of the number of slow moments Figure 5.2 shows the means and variances of (4.1), obtained by the micro-macro acceleration scheme with several values for the number of extrapolated slow moments L_x , without fast moments, i.e. $L_y = 0$. The number of Monte Carlo particles is $N = 10^5$. We initialise the process with the same Gaussian distribution as Section 4.2. From the experiments in Chapter 4, we already know that extrapolating only the slow mean yields inaccurate results for the slow variance, while extrapolating slow mean and slow variance yields accurate results. This result is simple to explain, as the exact Gaussian solution is completely determined by its mean and variance. The numerical simulation gives very accurate results for the slow mean and variance. When adding the third slow moment, the numerical approximation is almost the same, confirming the former reasoning. There could however arise a problem with three slow moments. After K microscopic inner steps, the third moment is a function of the slow mean and variance, called *slaving relations*, but after linear extrapolation, these slaving relations may not hold anymore. The matched distribution will hence not be Gaussian anymore, while the exact solution is. This effect is rather small.

Adding fast moments to the extrapolation operator The following experiment uses the same equation (4.1) with $L_x = 2$ slow moments: the mean and variance, and adds a few fast moments L_y to the extrapolation operator. Figure 5.3 depicts the slow and fast means and variances. The figures show that the slow mean and variance are not more accurate when adding fast moments. To the contrary, they even seem to lie a bit further away from the exact solution, although the differences are very small. This experiment confirms that for linear scale-separated processes, two slow moments are enough to obtain an accurate simulation.

A uniform initial distribution The above experiments demonstrate the situation where the initial condition is a normal distribution. The same adequate moment selection, however, also holds with a non-normal initial condition. In this setting, all intermediate distributions are not Gaussian but they converge to an invariant measure that is normally

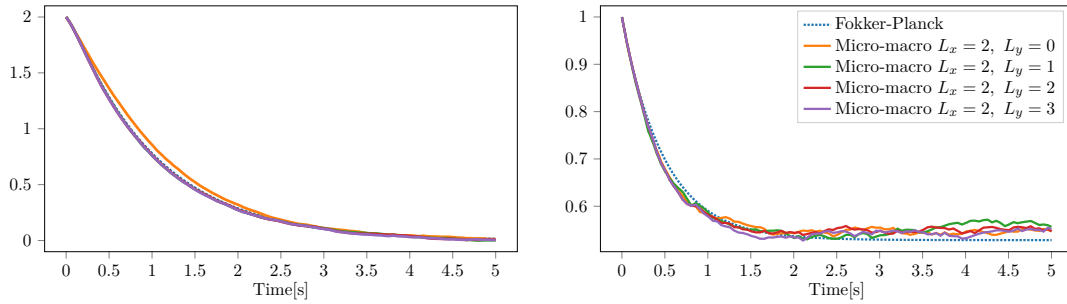


Figure 5.3: The mean (left) and variance (right) of the slow variable for different numbers of extrapolated fast moments L_y . Adding fast moments to the extrapolation operator does give more accurate simulations than without fast moments.

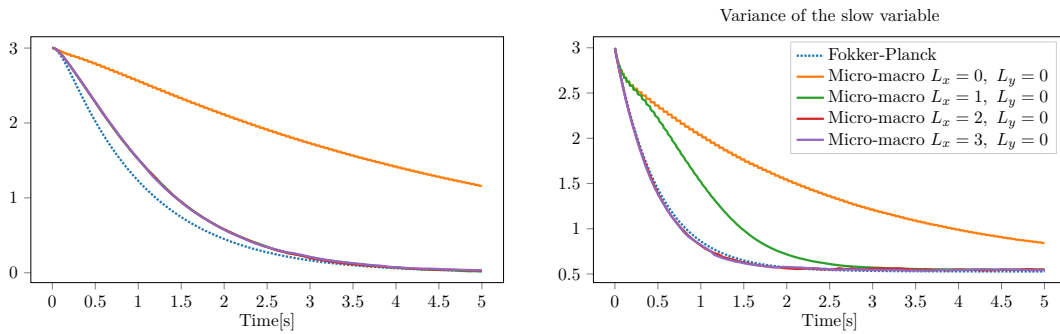


Figure 5.4: The mean (left) and variance (right) of the slow variable for different numbers L_x of extrapolated slow moments. The conclusion is the same as in Figure 5.2: two slow moments gives the most accurate simulation results by the micro-macro acceleration scheme.

distributed. Over time, the slow mean and slow variance hence become more important. Only in the case of a Gaussian initial condition, the intermediate distributions are too. As an example, we consider a uniform density over the square $[0, 6] \times [0, 6]$ as initial condition. Figure 5.4 shows the slow means and variances with several number of slow moments. Indeed, two slow moments gives the most accurate solution, even with a non-Gaussian initial.

5.3 A non-linear case: FENE-dumbbells

In practice, many stochastic processes are non-linear in nature. An a priori estimation of an adequate number of state variables is a lot more difficult than in the linear case and depends heavily on the macroscopic quantities of interest. In this section, we look at the problem of an accurate estimation of the number of macroscopic states in case of FENE-dumbbells. Section 3.3 already introduced the FENE model for polymers. The authors of [32] already performed an analysis of an adequate number of macroscopic

states in the context of lifting with an additional constrained simulation. This section performs the same experiments for matching, which has not been done yet.

5.3.1 FENE macroscopic state hierarchies

An important quantity for FENE-dumbbells is the stress tensor (3.9). The FENE SDE usually occurs in conjunction with the Navier-Stokes fluid equations, which are coupled through the stress tensor [25]. Hence, we want to approximate the stress tensor as accurately as possible. The authors of [32] propose three different macroscopic state hierarchies for approximating the stress tensor:

- Strategy 1: The first L even moments with $R_l(x) = x^{2l}$, $l = 1 \dots L$ from Section 3.3
- Strategy 2: Start with the first $L - 1$ even moments with $R_l(x) = x^{2l}$, $l = 1 \dots L - 1$, and the stress tensor (3.9) as additional final state function $R_L(x)$. This choice of state variables stems from a set of evolution equations that form a closure for FENE-dumbbells, clearly explained in [32]. We will not go deeper in the mathematical details.
- Strategy 3: Here we start again with the first even moment $R_1(x) = x^2$ and as additional state variables we add the first $L - 1$ term in the Taylor expansion of the stress tensor, by Itô's lemma. The first four state functions then are [32]:

$$R_1(x) = x^2, R_2(x) = \frac{x^2}{1 - x^2/\gamma^2} - 1, R_3(x) = \frac{x^2}{(1 - x^2/\gamma^2)^2}, R_4(x) = \frac{x^4}{(1 - x^2/\gamma^2)^3}. \quad (5.2)$$

For each of these three state hierarchies, the authors of [32] plot the τ - M_1 phase diagram together with the temporal evolution of both states. M_1 is the first state variable in every of the three strategies, i.e. $M_1 = \mathbb{E}[R_1(X)]$. When the velocity field $\kappa(t)$ is chosen as

$$\kappa(t) = 100t(1 - t)e^{-4t}, \quad (5.3)$$

the τ - M_1 phase diagram exhibits hysteresis, depicted in Figure 5.5. The goal of the following experiments is to analyse how many state variables of each hierarchy are required to approximate the hysteresis curve accurately. All experiments in this section use the same initial distribution as Section 3.3, which is the invariant distribution for $\kappa = 0$ (3.10). The parameter values are also the same, i.e. $\gamma = 7$, $W_e = 1$.

5.3.2 Numerical experiments

Figures 5.6 to 5.8 show the hysteresis curves and time evolution of τ and M_1 , computed by the micro-macro acceleration algorithm with relative-entropy matching, for the first, second and third hierarchy of state variables. The maximal extrapolation step is $\Delta t = 5\delta t$ and the relative-entropy matching operator allows for a maximum of three Newton-Raphson iterations, above which it reports a matching failure and halves the extrapolation

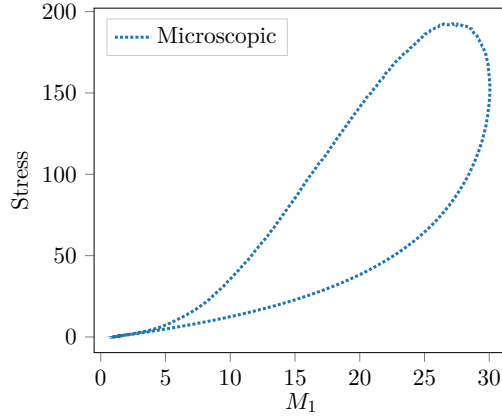


Figure 5.5: The τ - M_1 phase diagram for the FENE-process (3.8) with time dependent velocity field (5.3), simulated with the Euler-Maruyama scheme with time step $\delta t = 2 \cdot 10^{-4}$ up to 2 seconds.

step. Both parameters are small to clearly see the effect of the choice of macroscopic states. Otherwise, the micro-macro acceleration results may be cluttered by a large extrapolation error. The number of Monte Carlo particles is $N = 5 \cdot 10^4$.

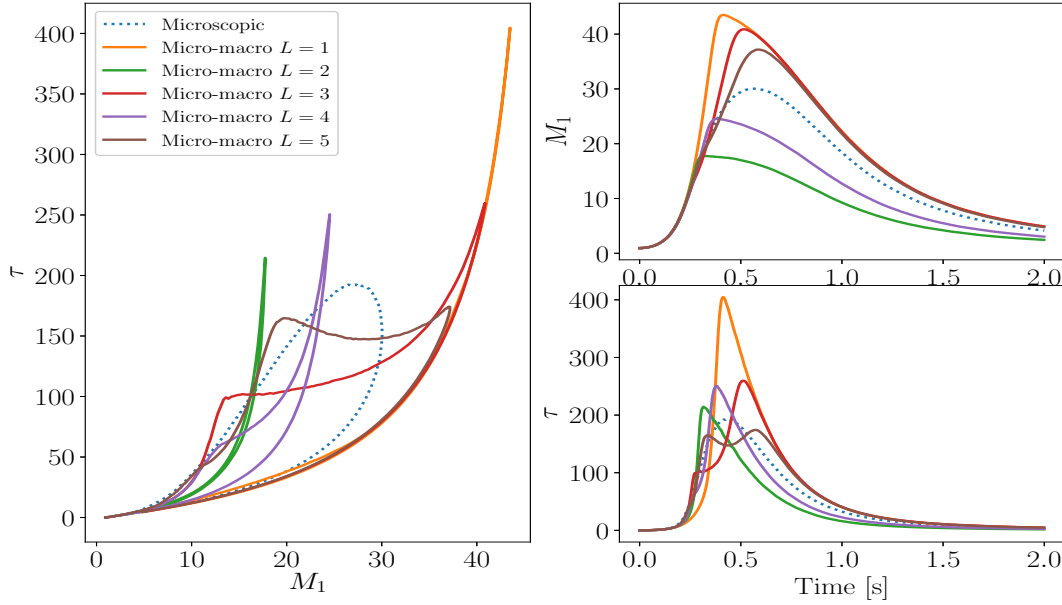


Figure 5.6: The τ - M_1 phase diagram for the first hierarchy of macroscopic state variables (left) and the evolution of M_1 (top right) and of the stress tensor τ (Bottom right). Only for $L = 5$ the first hierarchy starts to approximate the hysteresis curve well, although there still is room for improvement.

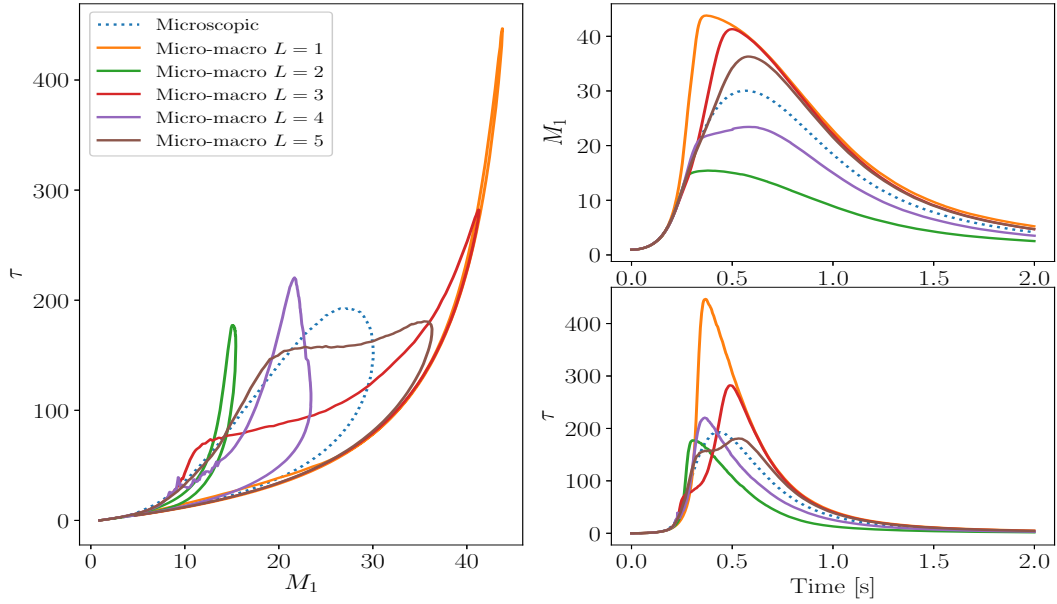


Figure 5.7: The τ - M_1 phase diagram for the first hierarchy of macroscopic state variables (left) and the evolution of M_1 (top right) and of the stress tensor τ (Bottom right). The conclusion is the same as for the first hierarchy of states: only for $L = 5$ the approximation of the hysteresis curve is better with a lot of room for improvement. The approximation here however is already better than for the first hierarchy of states.

For the first hierarchy of state variables in Figure 5.6, the micro-macro acceleration scheme is not able to capture the hysteresis effect when $2 \leq L \leq 4$. Only when $L = 5$, the approximation by the micro-macro acceleration scheme looks better although there still is a lot of room for improvements. When turning to the second state hierarchy in Figure 5.7, the approximation of the micro-macro acceleration algorithm with $L = 5$ is better than with the first hierarchy. Adding the stress tensor as a state variable, has a positive effect on the accuracy of the micro-macro acceleration scheme. The approximations for $L \leq 4$ are also a bit closer to the hysteresis curve but they all fail to capture the complete microscopic dynamics. The third hierarchy of state variables however yields the best results, visible in Figure 5.8. Already when $L = 4$, the micro-macro acceleration method almost exactly approximates the microscopic evolution. Strategy 3 hence requires fewer state variables and makes better approximations to the exact microscopic dynamics. As a result, out of the three proposed hierarchies of states, the third one is the best. This conclusion is very similar to the one obtained in [32] but now for matching instead of lifting, which is a new result.

5.3.3 Conclusion

Choosing an a priori number of macroscopic state variables is an important aspect of an efficiency analysis of the micro-macro acceleration algorithm. The numerical simulation

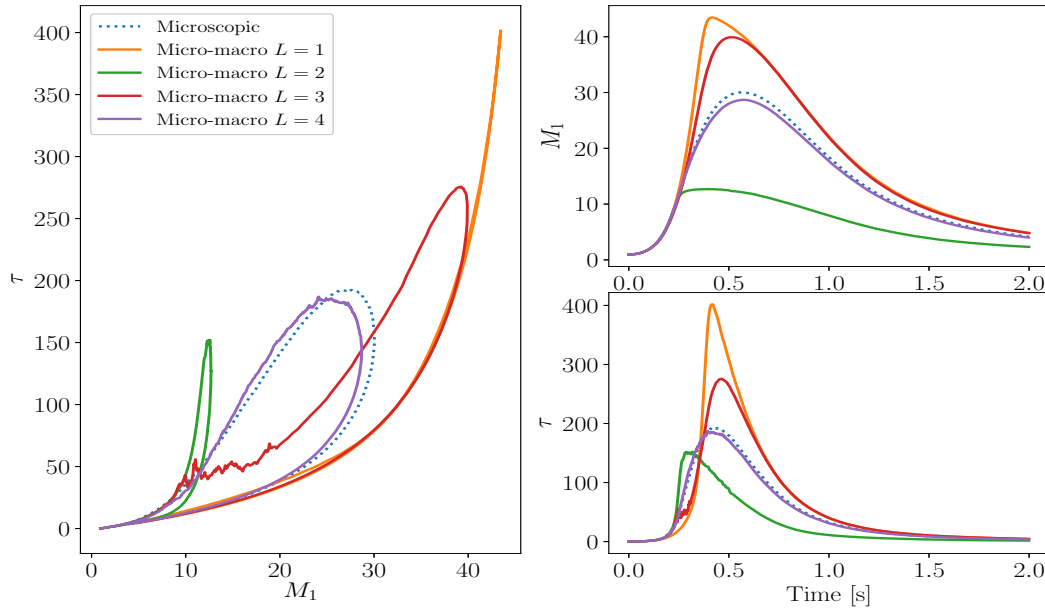


Figure 5.8: The τ - M_1 phase diagram for the first hierarchy of macroscopic state variables (left) and the evolution of M_1 (top right) and of the stress tensor τ (Bottom right). For the third hierarchy of states, $L = 4$ already gives much closer approximations to the hysteresis curve than the two previous hierarchies.

can either be very inaccurate when the number of states is too low, or too costly for a certain precision when the number is too high. In the case of a scale-separated linear process, two slow moments is enough for a very accurate simulation without performing too much work. For non-linear processes however, selecting good states is a much more difficult task and in many cases there are different state hierarchies to consider. In the context of FENE-dumbbells, choosing the state variables as terms in the stochastic Taylor expansion yields very accurate results for a modest number of moments. An algorithm that adaptively selects the number of states, based on some error estimate could be of great value, but is a topic for further research.

Chapter 6

Accuracy of the extrapolation step size

Until now, we discussed that the micro-macro acceleration algorithm converges with Δt decreasing to zero and stays stable with Δt depending only on the slow variables. The micro-macro error depends on the extrapolation time step and the number of macroscopic state variables. In the previous chapter, we looked at the effect of the number of states, here we look at extrapolation time step. We first investigate how the error grows as a function of increasing extrapolation time step. Second, to assess efficiency, we need to decide what the maximal error is that we are willing to tolerate. In situations where one can derive an approximate macroscopic model, the computational effort of micro-macro acceleration is wasted if we can obtain an equally accurate result with the approximate macroscopic model. Hence, we choose to define efficiency as “faster than microscopic method” and “more accurate than approximate macroscopic model”.

In Section 6.1 we introduce an averaging technique to derive an approximate macroscopic model for the slow variables of slow-fast SDEs. We will investigate the question of efficiency in the linear setting, where we do not need to care about the number of moments. Section 6.2 presents the linear driven process. We use a periodically driven system as it is easier to compute errors than for a standard linear process where the approximate macroscopic model reaches the same steady-state as the true solution. A similar periodic system was already studied in [24] to prove that implicit methods with large time steps are of no use for SDEs. In Section 6.3, we show that the method is generally more accurate than the approximate macroscopic model when the time-scale separation is moderate (as then, the approximate macroscopic model is insufficiently accurate). However, also in the limit of an infinite time-scale separation, the micro-macro acceleration method can be efficient when choosing Δt appropriately as a function of ε .

6.1 A natural averaging strategy

Suppose we have a slow-fast system of SDEs with a scale separation $\varepsilon > 0$

$$\begin{aligned} dX &= f_1(X, Y)dt + \alpha(X, Y)dW_x(t), \quad X(0) = X_0 \\ dY &= \frac{1}{\varepsilon}g(X, Y)dt + \frac{1}{\sqrt{\varepsilon}}\beta(X, Y)dW_y(t), \quad Y(0) = Y_0 \end{aligned} \quad (6.1)$$

where W_x and W_y are two independent Brownian motions. Note that the linear system (4.1) from Chapters 4 and 5 fit in this general framework. For very small values of ε , the system becomes stiff very quickly and the microscopic time steps are then prohibitively small. If we are only interested in the slow variable X , there exist an approximate macroscopic model that only describes the evolution of the slow mode by averaging out the fast mode. An approximate macroscopic model typically allows for much larger time steps when there is a large spectral gap between the eigenvalues of X and Y in (6.1), while attaining a very good accuracy when ε is small.

The idea of averaging is the following: when ε decreases to 0, the fast component $Y(t)$ settles very quickly to a conditional equilibrium where $\mathbb{E}[dY] \approx 0$ such that $\mathbb{E}(g(X(t), Y(t))) \approx 0$. As a consequence, it is possible to derive an equivalent SDE for the slow variable $X(t)$ alone, in the limit when ε decreases to 0. It is then usually possible to write $\mathbb{E}[Y]$ as a function of X so that we can plug the fast equilibrium in the slow equation of (6.1). Theorem 5 makes this intuitive reasoning rigorous. The theorem was taken from [29].

Theorem 5. *Given a system of SDEs (6.1) and a fixed end time T . The solution of the approximate macroscopic SDE*

$$d\bar{X} = F(\bar{X})dt + A(\bar{X})dW, \quad \bar{X}(0) = X_0 \quad (6.2)$$

approximates the solution of (6.1) in the limit as ε decreases to 0. The drift vector $F(x)$ and diffusion matrix $A(x)$ are given by

$$\begin{aligned} F(x) &= \int_Y f_1(x, y)\rho^\infty(y; x)dy \\ A(x)A(x)^T &= \int_Y \alpha(x, y)\alpha(x, y)^T \rho^\infty(y; x)dy, \end{aligned}$$

where $\rho^\infty(y; x)$ is the conditional invariant distribution of the fast variables of (6.1), given a value of the slow variable x .

Remark 4. Note that the diffusion matrix is inherently not uniquely defined, only $A(x)A(x)^T$ is. As a consequence, there are many SDEs of the form (6.2) that are a limiting form of (6.1). When $X(t)$ is one dimensional $A(x)$ can have any sign but will still generate the same macroscopic solution.

The approximate macroscopic model is only accurate for very small values of ε so that in general, the approximate macroscopic SDE will generate a modelling error for moderate and larger values of ε . The goal of micro-macro acceleration is to get rid of this modelling error but still be able to take larger time steps that exceed the stability bound of a full

microscopic simulation. For moderate values of ε , the micro-macro acceleration scheme could hence be more accurate than the approximate macroscopic model, but on the other hand still faster than a complete microscopic simulation due to stiffness. In the range of moderate ε , the micro-macro acceleration scheme could hence be more efficient than the approximate macroscopic model (7.3) and a microscopic simulation of (6.1). We investigate this phenomenon in more depth on the following example.

6.2 A linear driven process

As an example in this chapter, we consider a linear driven SDE of a form similar to the one in [24],

$$\begin{cases} dX = -\lambda(X + Y)dt + E \sin(at)dt + dW_x \\ dY = \frac{1}{\varepsilon}(X - Y)dt + \frac{1}{\sqrt{\varepsilon}}dW_y. \end{cases} \quad (6.3)$$

with $\lambda = 2$, $E = 10$, $a = 2\pi$. In practice, we are only interested in $\mathbb{E}[X]$. From Section 5.2 we know that only extrapolating the slow mean is already accurate for the mean and adding extra slow moments does not improve accuracy. The evolution for $\mu_X(t) = \mathbb{E}[X(t)]$ and $\mu_Y(t) = \mathbb{E}[Y(t)]$ is by the martingale property given by

$$\begin{cases} \frac{d\mu_X}{dt} = -\lambda(\mu_X + \mu_Y) + E \sin(at) \\ \frac{d\mu_Y}{dt} = \frac{1}{\varepsilon}(\mu_X - \mu_Y), \end{cases} \quad (6.4)$$

which has an analytical solution given by

$$\begin{pmatrix} \mu_X(t) \\ \mu_Y(t) \end{pmatrix} = e^{tM} \begin{pmatrix} \mu_{X_0} - A \\ \mu_{Y_0} - C \end{pmatrix} + \begin{pmatrix} A \\ C \end{pmatrix} \cos(at) + \begin{pmatrix} B \\ D \end{pmatrix} \sin(at), \quad (6.5)$$

where $M = \begin{pmatrix} -\lambda & -\lambda \\ \frac{1}{\varepsilon} & -\frac{1}{\varepsilon} \end{pmatrix}$ and $\begin{pmatrix} \mu_{X_0} \\ \mu_{Y_0} \end{pmatrix}$ is the initial condition to the equation. The constants A, B, C and D are the solution of the linear system

$$\begin{pmatrix} -a & \lambda & 0 & \lambda \\ \lambda & a & \lambda & 0 \\ 0 & -\frac{1}{\varepsilon} & -a & \frac{1}{\varepsilon} \\ -\frac{1}{\varepsilon} & 0 & \frac{1}{\varepsilon} & a \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \begin{pmatrix} E \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The linear driven process (6.3) fits in the framework of Theorem 5 with $f_1(x, y) = -2(x + y)$, $\alpha(x, y) = 1$, $g(x, y) = x - y$ and $\beta(x, y) = 1$. The approximate macroscopic model for the driven problem then is

$$d\bar{X} = -2\bar{X}dt + 10 \sin(2\pi t)dt + dW. \quad (6.6)$$

Note that the external driving force $10 \sin(2\pi t)$ can be ignored in the computation of the approximate macroscopic model as it is independent from X and Y . The equation for

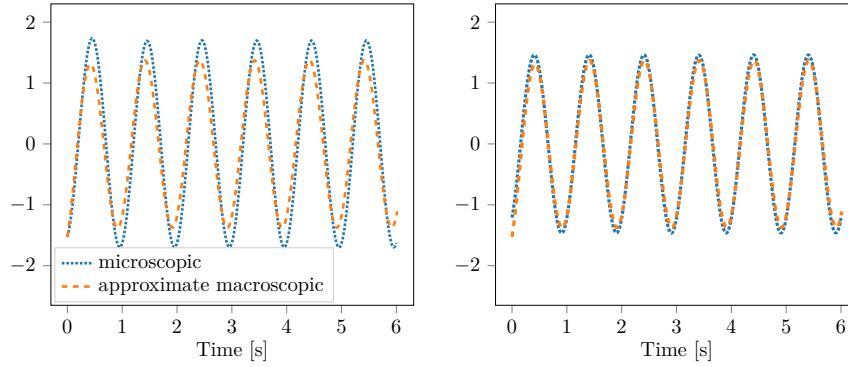


Figure 6.1: The solutions of the slow variables with the microscopic and approximate macroscopic model with initial condition close to the periodic orbit, for $\varepsilon = 0.5$ (left) and $\varepsilon = 0.03$ (right). The approximate macroscopic model lies closer to the exact microscopic dynamics when ε decreases to 0, as Theorem 5 predicts.

$\bar{\mu}(t) = \mathbb{E}[\bar{X}(t)]$ of the approximate macroscopic model (6.6) is by the martingale property

$$\frac{d\bar{\mu}}{dt} = -2\lambda\bar{\mu} + E \sin(at), \quad (6.7)$$

with analytical solution

$$\bar{\mu}(t) = \left(\mu_0 + \frac{aE}{a^2 + 4\lambda^2}\right)e^{-2\lambda t} - \frac{aE}{a^2 + 4\lambda^2} \cos(at) + \frac{2\lambda E}{a^2 + 4\lambda^2} \sin(at). \quad (6.8)$$

Figure 6.1 shows the solution of the microscopic and approximate macroscopic model for two values of the scale-separation, $\varepsilon = 0.5$ and $\varepsilon = 0.03$ and $\delta t = \varepsilon/20$. The approximate macroscopic model lies already quite close to the exact microscopic solution and the approximation gets better as ε decreases to 0. To measure the discrepancy in the slow variable between the microscopic and approximate macroscopic model, we use the L_2 -error between the two over one period. The L_2 -norm is a good choice as both solutions have the same phase but a different amplitude. The errors in L_2 -norm of the micro-macro acceleration scheme in this chapter are always approximated by a second order trapezoidal scheme [34], while for the approximate macroscopic model the analytical expressions are used. Figure 6.2 depicts this L_2 -error as a function of ε . The Figure shows that the error by the approximate macroscopic model decreases linearly with ε , which is an important result for the next section.

6.3 Efficiency and accuracy of the micro-macro scheme

A fundamental quantity in the context of the micro-macro acceleration scheme in this chapter is the extrapolation factor M , i.e. $\Delta t = M\delta t$. Figure 6.3 shows the solutions of the slow mode of the micro-macro acceleration scheme to the linear driven process for

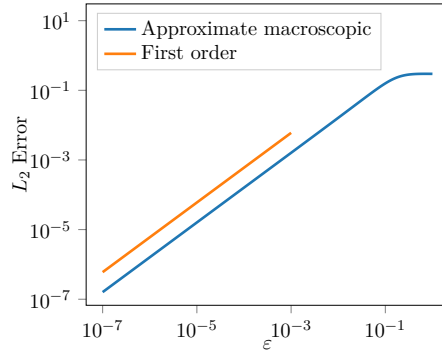


Figure 6.2: L_2 -error of the approximate macroscopic model compared to the exact solution of the microscopic model, as a function of ε . The error decreases linearly with ε .

several values of M and only slow-mean extrapolation, with an intermediate of $\varepsilon = 0.1$. The number of Monte Carlo particles is $N = 10^5$. One difference between the approximate macroscopic model and the micro-macro acceleration method is that the former always has a lower amplitude than the exact solution, while the latter always overshoots the true solution. The overshooting behaviour is due to the extrapolation of the slow mean, because the micro-macro acceleration scheme follows the steep slope for longer time steps than a full microscopic simulation.

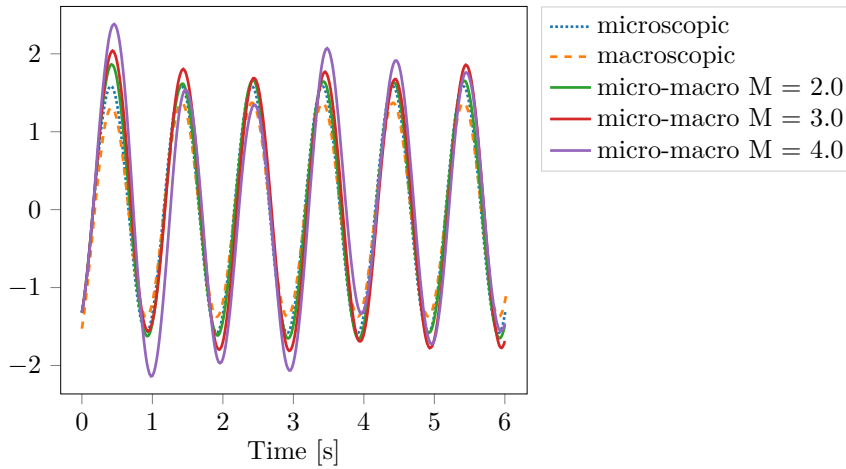


Figure 6.3: Solutions of the slow variable for the micro-macro acceleration scheme for several values of M and $\varepsilon = 0.1$. The smaller M , the better the approximation of the micro-macro acceleration scheme. For larger M , the approximate macroscopic model is more accurate than the micro-macro acceleration scheme, while for small M it is the other way around.

When $M = 2, 3$, the micro-macro acceleration scheme follows the microscopic dynamics very well, while if $M = 4$, the amplitude of the micro-macro acceleration solution is

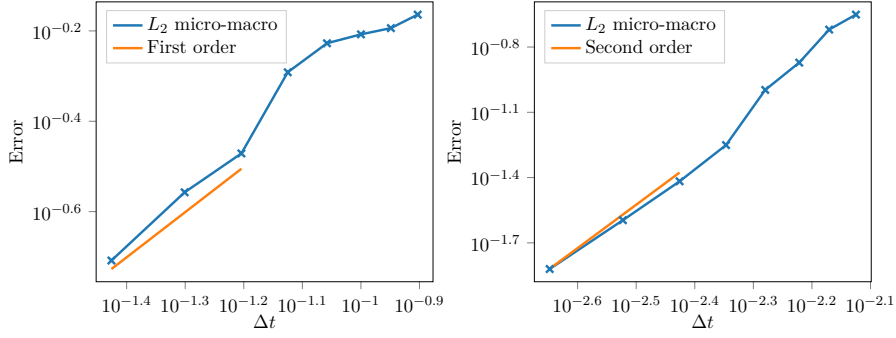


Figure 6.4: First order convergence in Δt of the micro-macro acceleration error of the slow variable for large $\varepsilon = 0.5$ (left). Second order convergence for small $\varepsilon = 0.03$ (right).

different in every peak. For small M , the micro-macro acceleration method is more accurate than the approximate macroscopic model, while for larger M it is the other way around. The fundamental question is how far we can extrapolate before the approximate macroscopic model for the slow mode is more accurate than the micro-macro acceleration method, measured in the L_2 -norm over one interval. In other words, for a given ε , we are interested in the maximal $M(\varepsilon)$.

To quantify $M(\varepsilon)$, we first need to know how the micro-macro acceleration error depends on the extrapolation step size Δt . Figure 6.4 depicts the convergence order of the micro-macro acceleration error for a larger ($\varepsilon = 0.2$) and smaller ($\varepsilon = 0.03$) time-scale separation. If ε is large, the error of the micro-macro acceleration scheme decrease linearly with Δt , as Theorem 1 predicts. However, for smaller ε , hyper convergence sets in and the error decreases quadratically. There is, however, no direct explanation for hyper convergence in the context of micro-macro acceleration. For small ε , the error of the micro-macro acceleration scheme can be written as

$$E_{mM} = \alpha(\varepsilon)(M\delta t)^2 = \alpha(\varepsilon)(M'\varepsilon)^2, \quad M' = CM \quad (6.9)$$

because $\delta t = C\varepsilon$ with C a constant as the microscopic stability domain scales with ε . The function $\alpha(\varepsilon)$ is for each ε the proportionality constant between the error of the micro-macro acceleration method and the extrapolation time step. Putting it all together, we are able to deduce a formula when the micro-macro acceleration scheme is more accurate than the approximate macroscopic model. Figure 6.2 demonstrated that the error of the approximate macroscopic model decreases linearly, i.e. $E_{avg} = \beta\varepsilon$. The micro-macro acceleration scheme and approximate macroscopic model have the same accuracy when

$$\alpha(\varepsilon)(M'\varepsilon)^2 = \beta\varepsilon, \quad (6.10)$$

which gives $M' = \sqrt{\frac{\beta}{\alpha(\varepsilon)\varepsilon}}$. The only unknown is the function $\alpha(\varepsilon)$, which is hard to deduce theoretically.

As a numerical illustration, Figure 6.5 shows, for a range of ε -values, where the micro-macro acceleration error (solid lines) crosses the error of the approximate macroscopic

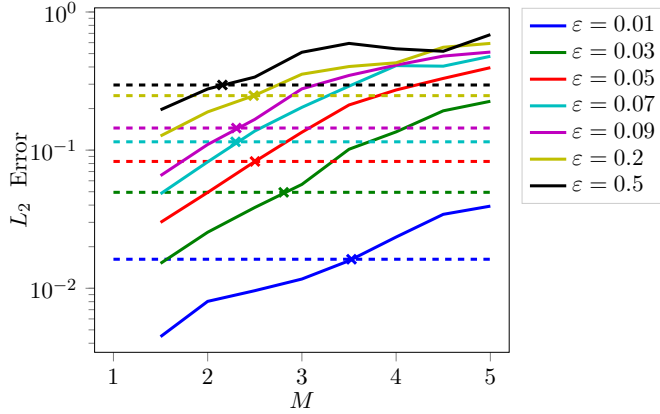


Figure 6.5: The L_2 -errors of the micro-macro acceleration scheme (solid lines) for several values of ε and the L_2 errors of the approximate macroscopic model (dashed lines). The crosses indicate when the micro-macro acceleration method becomes as accurate as the approximate macroscopic model. The maximal extrapolation factor increases when ε decreases.

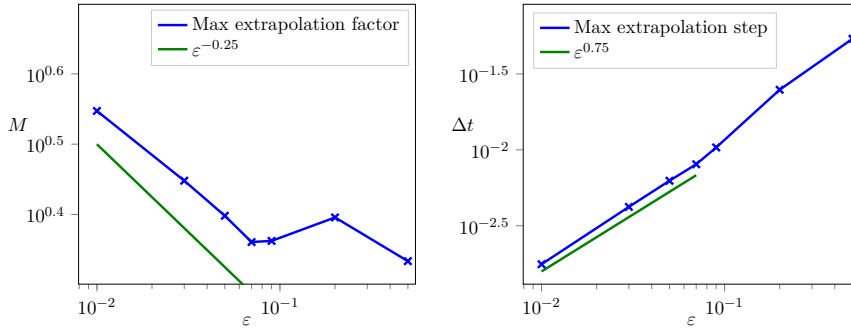


Figure 6.6: The maximal extrapolation factor $M(\varepsilon)$ before the approximate macroscopic model is more accurate than the micro-macro acceleration method, as a function of ε (left), and the corresponding maximal Δt (right). For small ε , $M(\varepsilon)$ scales with ε which is an important result in the efficiency analysis.

model (dashed lines). The former are the full decreasing lines while the latter are the constant dashed lines. The points where the error of the micro-macro acceleration method and approximate macroscopic cross each other are depicted in Figure 6.6, on the left. These points are exactly $M(\varepsilon)$. Figure 6.6 (left) shows these crossover points. For high ε the gain of the micro-macro acceleration scheme is almost constant, while for smaller ε , in the regime with hyper convergence, the extrapolation factor seems to increase like $\varepsilon^{-1/4}$. The corresponding Δt hence decreases as $\varepsilon^{3/4}$, which is better than the scaling law for a microscopic simulation.. There is, however, no theoretical explanation for the exact scaling law for the micro-macro acceleration scheme since $\alpha(\varepsilon)$ is unknown. For small ε , we can hence take larger extrapolation steps than a complete microscopic simulation,

while attaining an accuracy that is better than the approximate macroscopic model.

Conclusion The stability result in Section 4.4 states that the maximal extrapolation step is independent of ε , at least for block-diagonal linear processes. However, for accuracy, the maximal time step Δt allowed before the approximate macroscopic model is more accurate than the micro-macro acceleration scheme depends on ε . The main result of this chapter is that for a wide range of ε , the micro-macro acceleration scheme gains efficiency over the microscopic method as it is able to take larger steps. For small ε , the micro-macro acceleration gain even increases. The micro-macro acceleration method is as accurate as the approximate macroscopic model, when $\Delta t = \mathcal{O}(\varepsilon^{3/4})$. Moreover, any accuracy level between the microscopic and approximate macroscopic model can be achieved by tuning Δt , while taking larger steps than the microscopic model. This is a new result and demonstrates the merit of the micro-macro acceleration scheme for small ε . These results, however, have not been fully analysed yet and more experiments with smaller values of ε are required.

Chapter 7

Applications

In Chapters 2 and 3, we dealt with convergence and an implementation of the micro-macro acceleration scheme while in Chapter 4, we introduced the linear slow-fast system as the first step in the efficiency analysis. Chapter 4 also contained a summary of the stability of the micro-macro acceleration method and presented a new convergence result. In Chapter 5, we discussed an adequate number of state variables that is required for an accurate simulation, both for the linear slow-fast model and FENE-dumbbells. Chapter 6 presented a scaling law for the extrapolation factor on a linear driven process and this factor increases when the time-scale separation get stronger. This final chapter opens up the application domain of the micro-macro acceleration scheme to important areas such as molecular dynamics.

This chapter contains two numerical examples, a bistable system in Section 7.1 and a model for a molecule consisting of three atoms in Section 7.2. For both models, there exist approximate macroscopic models for the slow dynamics, but these models make a large steady-state error for some parameter settings. The micro-macro acceleration method will be able to alleviate these modelling errors while allowing for larger time steps than a complete microscopic simulation.

7.1 A bistable system

A first case that depicts the power of the micro-macro acceleration scheme is given by a bistable system of SDEs

$$\begin{aligned}dX &= -(2X + Y)dt + AdW_x \\dY &= \frac{1}{\varepsilon}(Y - Y^3)dt + \frac{1}{\sqrt{\varepsilon}}dW_y = -\frac{1}{\varepsilon}\nabla V(Y)dt + \frac{1}{\sqrt{\varepsilon}}dW_y.\end{aligned}\tag{7.1}$$

The bistable process is inspired from the more complex bistable problem, introduced in [4], while retaining its critical features. The constant A determines the noise on the slow variable. The fast variable Y is autonomous and governed by a double-well potential $V(y)$, plus the effect of Brownian motion. The potential energy function is defined as

$V(y) = \frac{1}{4\varepsilon}y^4 - \frac{1}{2\varepsilon}y^2$. The invariant distribution of the fast variable is in general given as a function of the potential energy,

$$\rho^\infty(y) = \frac{1}{Z}e^{-V(y)} \quad (7.2)$$

with $Z = \int_{\mathbb{R}} e^{-V(y)} dy$ the normalization constant. Figure 7.1 depicts the potential energy function on the left and the associated invariant distribution on the right. The points $y = \pm 1$ are stable point and $y = 0$ is unstable.

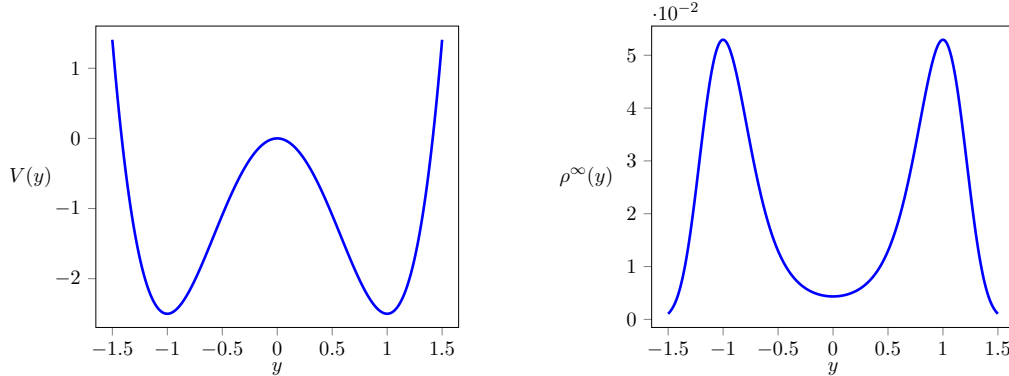


Figure 7.1: The double well potential for $\varepsilon = 0.1$ corresponding to (7.1) (left) and the associated invariant measure according to (7.2) (right).

As an effect, if a particle starts in one well, it will oscillate around the stable point due to the inherent random Brownian increments and will occasionally switch wells when the Brownian increment is suddenly large. The particle then oscillates a while in the other well before it moves again to the first well. This switching process repeats constantly. Figure 7.2 shows the motion of one particle in the potential well and the influence on the slow particle, for $\varepsilon = 0.1$, $A = 0.1$ and for $\varepsilon = 0.01$, $A = 0.1$. When ε is small, the fast variable (blue) switches quickly between the wells while the slow variable (red) stays relatively constant. For large ε however, the slow variable feels the switching behaviour of the fast variable. The slow mode increases when the fast part is in the left well and decreases when it resides in the other well, because the fast variable stays for a longer time in one of the wells. This switching behaviour is explained in much detail in [4], but we will not go deeper into details.

7.1.1 The modelling error in the approximate macroscopic model

For small ε , it is possible to derive an approximate macroscopic model for the slow variables as these do not change under the influence of the fast variables. The bimodal equations (7.1) fit in the framework of (6.1) with $f_1(x, y) = -2x - y$, $g(x, y) = -V(y)$, $\alpha(x, y) = A$ and $\beta(x, y) = 1$. According to Theorem 5 the bistable system is well approximated by the approximate macroscopic system

$$d\bar{X} = -2\bar{X}dt + AdW_x, \quad (7.3)$$

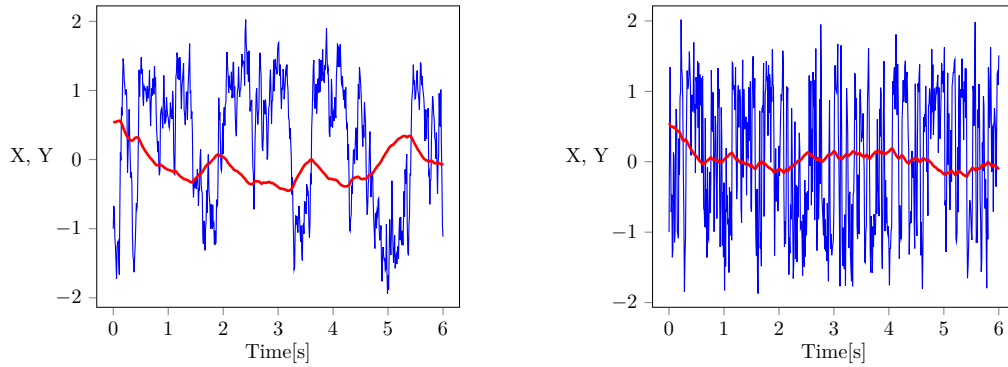


Figure 7.2: Behaviour of a fast particle through the double well potential in blue and the effect of well switching on the slow variable in red for $\varepsilon = 0.1$ (left) and $\varepsilon = 0.01$ (right). For large ε , the fast particle resides longer in one well and its effect on the slow variable is visible, while this effect is averaged out for small ε .

because

$$F(x) = \int_Y (-2x - y) \frac{1}{Z} e^{-V(y)} dy = -2x - \mathbb{E}[Y] = -2x$$

$$A(x)^2 = \int_Y A^2 \frac{1}{Z} e^{-V(y)} dy = A^2.$$

Hence, the double well potential is completely eliminated and replaced by a constant value $Y = 0$ in the approximate macroscopic model. The approximate macroscopic model introduces a modelling error by ignoring the motion of the fast variable, which has an effect on the slow variable, as seen in Figure 7.2 on the left. Figure 7.3 displays the slow and fast means, variances and histograms for the bimodal problem with $\varepsilon = 0.1$ and $A = 0.1$, all computed with $N = 10^5$ particles. The initial condition is chosen so that all fast particles reside in the left well and the slow particles are in equilibrium, so that $X = -Y/2$. The slow steady-state variance in the microscopic and approximate macroscopic models are different, indicating a severe modelling error by the approximate macroscopic model. The slow histogram in Figure 7.3 indeed depicts that the approximate macroscopic model fails to capture the variance of the microscopic model.

The approximate macroscopic model does, however, approximate the slow microscopic behaviour better in two cases: when the time-scale separation is stronger and when the noise level A on the slow variables is higher. The former case is due to Theorem 5, while in the latter case, the Brownian motion in the slow equation of (7.1) starts to dominate. The slow distribution will start looking more like a Gaussian, as the Brownian motion is larger in amplitude than the bimodal behaviour of the fast variables. Figure 7.4 depicts the variances and histograms for both cases.

To sum up, for moderate scale separations ε and small noise levels A the macroscopic model makes a non-negligible modelling error on the microscopic distribution. The approximate macroscopic model is hence of little use in this regime.

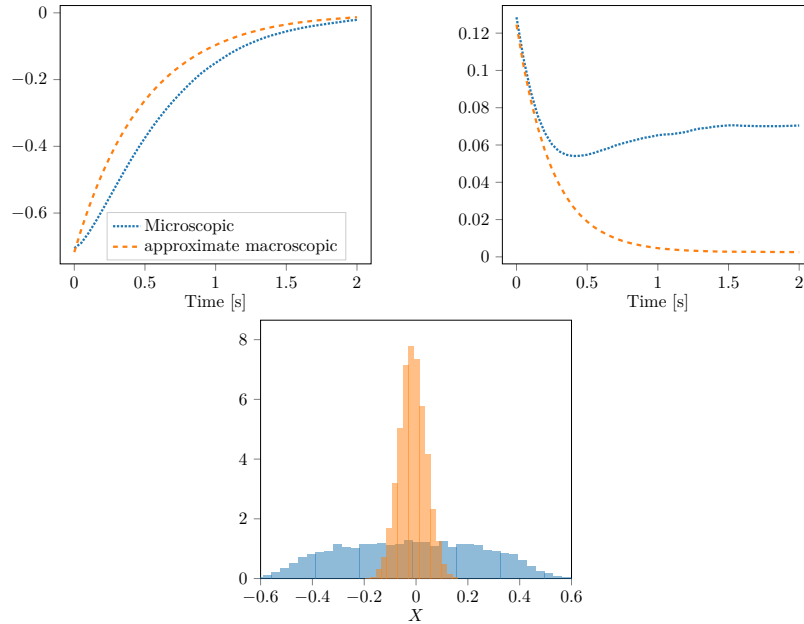


Figure 7.3: Means (top left), variances (top right) and histograms (bottom) after 2 seconds of the microscopic and approximate macroscopic model (7.3) of the slow variable of (7.1) for the bimodal problem. The model parameters are $\varepsilon = 0.1$ and $A = 0.1$. There is a clear steady-state error by the approximate macroscopic model in the slow variance, also visible in the histogram.

7.1.2 Improvements by the micro-macro acceleration scheme

The offset of the micro-macro acceleration scheme is that it interleaves microscopic simulations of the complete SDE with extrapolations of only slow state variables. This way, the acceleration scheme is also truthful to the underlying microscopic behaviour, in contrast to the approximate macroscopic model from previous section. Figure 7.5 depicts the variances and histograms of the slow variables for the same parameter values as in Figure 7.3. The micro-macro acceleration scheme does not make a modelling error, while allowing for larger time steps. For larger extrapolation factors however, the micro-macro acceleration method becomes less accurate in the transient state. It does however always converge to the exact steady-state solution of the slow variance. An adaptive step size strategy would be of great value in this setting, taking smaller steps in the transient and larger steps when the steady-state is almost reached.

Conclusion In the example problem from this section, the micro-macro acceleration scheme does not make a modelling error in the mean and variance of the slow mode. This result is clearly visible in case of larger ε and modest noise levels and is an improvement over the approximate macroscopic models for the slow variables. Also when the process has an inherent large scale separation or a large noise level, the micro-macro

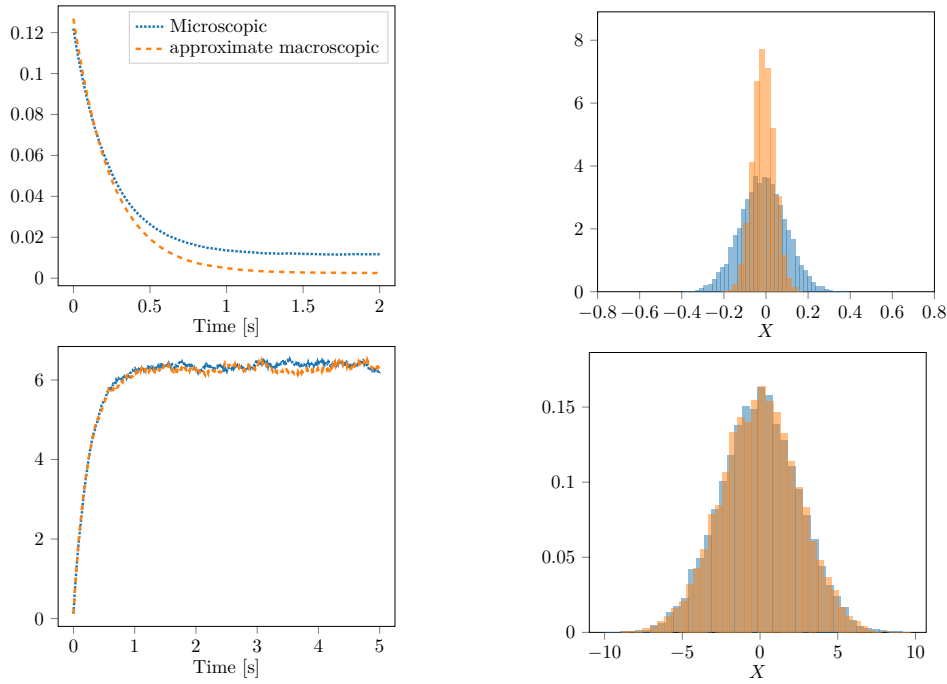


Figure 7.4: Variance (top left) and histogram (top right) of the slow variables using the microscopic and approximate macroscopic model (7.3) with model parameters $\varepsilon = 0.01$ and $A = 0.1$. The bottom two figures are similar with parameters $\varepsilon = 0.1$ and $A = 5$. In both cases, the approximate macroscopic model lies closer to the microscopic dynamics.

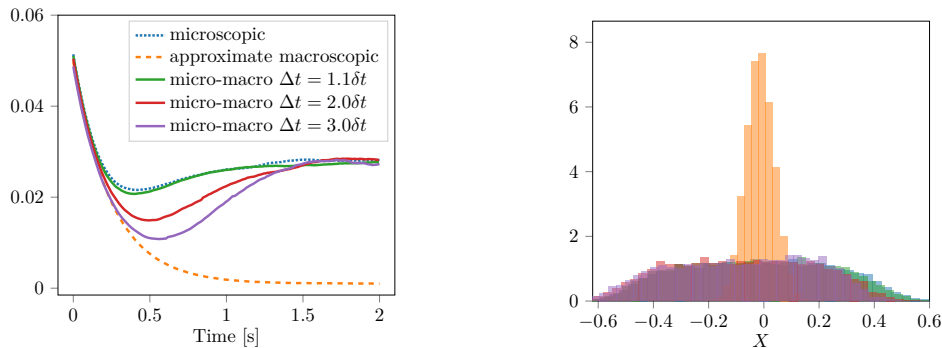


Figure 7.5: The slow variances (left) and histograms after 2 seconds (right) of the microscopic and approximate macroscopic model, together with the micro-macro acceleration scheme with extrapolating the first two moments and several given extrapolation step sizes. The model parameters are $\varepsilon = 0.1$, $A = 0.1$. The micro-macro acceleration scheme does not make a steady-state error, improving greatly over the approximate macroscopic model. The histograms of the microscopic and by the acceleration scheme are also indistinguishable.

acceleration method does not make a modelling error either. However, in these two cases, the approximate macroscopic model is very accurate and the improvements by the micro-macro acceleration scheme are not clearly visible. To conclude, the micro-macro acceleration scheme gains in accuracy by getting rid of the modelling errors made by the approximate macroscopic model. At the same time, the scheme increases the efficiency while allowing larger time steps than a full microscopic simulation.

7.2 A tri-atom molecule

A second practical application arises from the field of molecular dynamics and describes a planar molecule consisting of three atoms, A , B and C . The authors of [22] studied this problem in detail. The atoms move under influence of deterministic potential energy function V and Brownian motion depending on the temperature T . For a unique solution, we fix molecule $B = (0, 0)$ in the origin of the plane and $A = (x_a, 0)$ can only move on the positive x-axis, i.e. $x_a > 0$. Molecule C has coordinates $C = (x_c, y_c)$. Figure 7.6 depicts the conformation of the molecule and its motion is described by the over-damped Langevin dynamics, taken from [5]

$$\begin{cases} dx_a = -\frac{\partial V}{\partial x_a} dt + \sqrt{2\beta^{-1}} dW_{x_a} \\ dx_c = -\frac{\partial V}{\partial x_c} dt + \sqrt{2\beta^{-1}} dW_{x_c} \\ dy_c = -\frac{\partial V}{\partial y_c} dt + \sqrt{2\beta^{-1}} dW_{y_c}, \end{cases} \quad (7.4)$$

where $\beta = 1/T$ is the inverse temperature. The potential energy $V(x_a, x_c, y_c)$ is given by

$$V(x_a, x_c, y_c) = \frac{1}{2\varepsilon}(x_a - l_{\text{eq}})^2 + \frac{1}{2\varepsilon}(\sqrt{x_c^2 + y_c^2} - l_{\text{eq}})^2 + W(\theta), \quad (7.5)$$

with l_{eq} the equilibrium distance between molecules A and B and B and $B = C$. The potential energy in the angle between A and C is bimodal and reads

$$W(\theta) = \frac{k}{2}((\theta - \theta_{\text{saddle}})^2 - \delta\theta^2). \quad (7.6)$$

The parameter values in this section are $\varepsilon = 10^{-3}$, $l_{\text{eq}} = 1$, $k = 208$, $\theta_{\text{saddle}} = \frac{\pi}{2}$ and $\delta\theta = \theta_{\text{saddle}} - 1.1187$. Note that the parts of x_a and $r_c = \sqrt{x_c^2 + y_c^2}$ are stiff, while $W(\theta)$ is bimodal and represents the slow motion in the molecule. The peaks of the bimodal distribution are located in $\theta = 1.187$ and $\theta = \pi - 1.187$. In practice, the solution of (7.4) is expensive to simulate while only the bimodal behaviour of θ is of interest. The angle determines which shapes the molecule can take, which is a generally important topic in chemistry [27]. There however exists a closed model with only one equation that tries to model the bimodality of the system, while allowing for larger time steps. We discuss such a closure model in the following section.

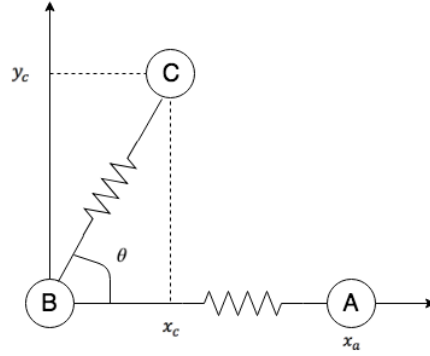


Figure 7.6: The planar tri-atom molecule with B fixed at the origin and A constrained to the x -axis.

7.2.1 Approximate macroscopic models for over-damped Langevin dynamics

In the context of Langevin dynamics, there is no $\frac{1}{\sqrt{\varepsilon}}$ present in the stiff equations, only an inverse temperate that does not scale with ε . As a result, Theorem 5 does not hold for the tri-atom molecule. There however exist general approximate macroscopic models for over-damped Langevin problems. The authors of [22] study this problem extensively, and the results are summarized shortly here, with the same notation. Suppose we have a general system of SDEs described a potential function V ,

$$d\mathbf{X} = -\nabla V dt + \sqrt{2\beta^{-1}} d\mathbf{W}, \quad \mathbf{X} \in \mathbb{R}^n \quad (7.7)$$

and we are interested in the behaviour of a scalar variable $z = \xi(\mathbf{X})$, also called a *reaction coordinate* [22]. The approximate macroscopic dynamics of z is given by the SDE

$$dz = b(z)dt + \sqrt{2\beta^{-1}}\sigma(z)dW, \quad (7.8)$$

where $b(z)$ and $\sigma(z)$ are defined by averaging out the fast variables

$$\begin{aligned} b(z) &= \int (-\nabla V(\mathbf{x}) \cdot \nabla(\mathbf{x})\xi + \beta^{-1}\Delta\xi(\mathbf{x}))\Psi_\infty(\mathbf{x})\delta_{\xi(\mathbf{x})=z}d\mathbf{x} \\ \sigma^2(z) &= \int |\nabla\xi(\mathbf{x})|^2\Psi_\infty(\mathbf{x})\delta_{\xi(\mathbf{x})=z}d\mathbf{x}, \end{aligned} \quad (7.9)$$

with $\Psi(\mathbf{x}) = \frac{1}{Z} \exp V(\mathbf{x})$ the invariant distribution of (7.7). The authors of [22] have proven that the approximate macroscopic model converges in path-wise sense to the microscopic dynamics when $\varepsilon \rightarrow 0$, which validates the use of the approximate model (7.8). The integrals in (7.9) however, are only tractable to compute for very simple reaction coordinates. Usually, numerical integration in some points z is required, with linear interpolation between these values.

The authors of [22] consider two reaction coordinates for the tri-atom molecule. The first is $\xi_1(X) = \theta$ and the second $\xi_2(X) = \|A - C\|^2 = (x_a - x_c)^2 + y_c^2$, which is similar to the first reaction coordinate at first sight. Note that the first reaction coordinate appears readily in (7.4) and is purely slow. It is possible to show by calculating the integrals in (7.9) that (7.8) for ξ_1 reduces to

$$d\theta = -W'(\theta)dt + \sqrt{2\beta^{-1}}dW \quad (7.10)$$

The derivation is tedious but elementary and not given here. The dynamics is however logical since θ appears readily in the expression for the potential energy (7.5). The reaction coordinate ξ_2 however, contains both a slow and fast component due to the presence of θ and r_c in $\|A - C\|^2$. The analytical derivation of a closed model for $\xi_2(t)$ is not immediately possible, since we need to compute the integrals (7.9) over a infinite skew cylinder $(x_a - x_c)^2 + y_c^2 = z = \text{constant}$. For this non-trivial reaction coordinate, we compute the integrals numerically in 100 uniformly distributed grid points z in the interval $[0, 5]$ by a Monte Carlo procedure with $x_a \in [-5, 5]$, $\theta \in [0, 2\pi]$ sampled uniformly with $N = 10^5$ particles. The resulting functions $b(z)$, $\sigma(z)$ are depicted in Figure 7.7. The critical points where $b(z) = 0$ correspond exactly to the critical points of $W(\theta)$.

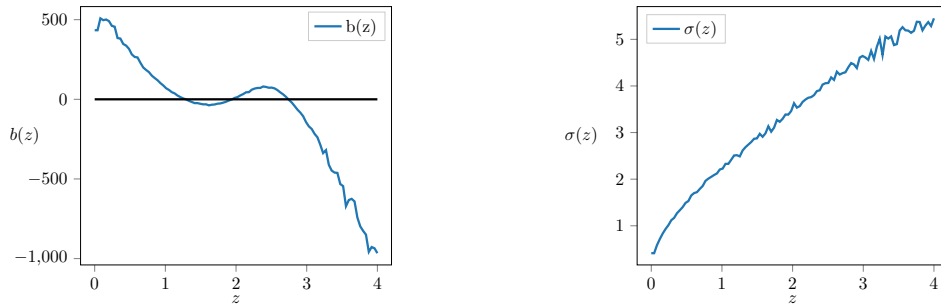


Figure 7.7: The drift term $b(z)$ (left) and diffusion term $\sigma(z)$ (right) of the approximate macroscopic model with reaction coordinate ξ_2 (7.9), computed by numerical integration and linear interpolation.

Suppose we have an initial condition that is very far from equilibrium, where x_a is uniformly distributed between 0.4 and 0.5, θ is chosen around one well, and the position of atom C is defined by $x_c = 2 \cos(\theta)$, $y_c = 2 \sin(\theta)$. Figure 7.8 shows the evolution of the mean and histogram of θ after six seconds using a time step of $\delta t = \varepsilon$ and $N = 5 \cdot 10^4$ particles, for both the microscopic model and the approximate macroscopic model with reaction coordinate ξ_1 . The approximate macroscopic model follows the exact dynamics of θ very well and the histograms are indistinguishable. For the bad reaction coordinate ξ_2 however, the approximate macroscopic model makes a modelling error in the mean of $\|A - C\|^2$, depicted in Figure 7.9. As a consequence, the peaks in the histograms have a different magnitude. The authors of [22] ascribed the modelling error to the fact that ξ_2 is not completely slow. The bimodality of $\|A - C\|^2$ is hence biased by the second reaction coordinate. Also note that with reaction coordinate ξ_2 , the mean changes very

abruptly in the beginning. This phenomenon has been studied extensively in [35] and is also due to the fact that ξ_2 is not entirely slow.

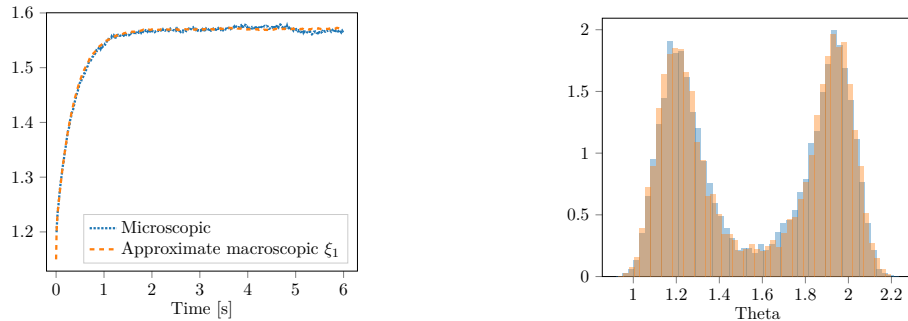


Figure 7.8: Evolution of the mean of θ (left) and the histogram of θ at time $T = 6$ seconds (right) for the microscopic and approximate macroscopic model with reaction coordinate ξ_1 (7.10). The approximate macroscopic model has no modelling error.

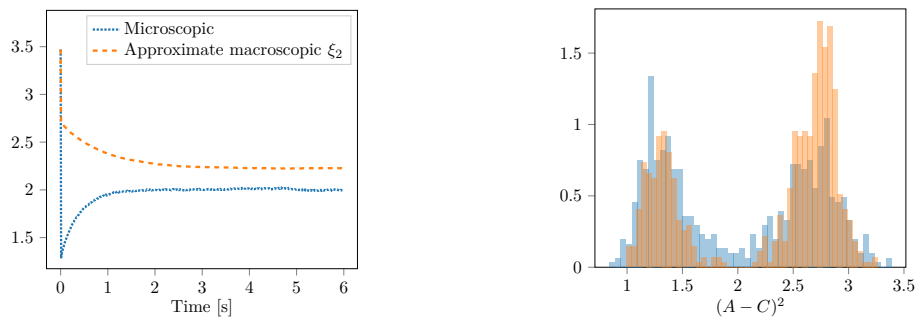


Figure 7.9: Evolution of the mean of $\|A - C\|^2$ (left) and the histogram of $\|A - C\|^2$ at time $T = 6$ seconds (right) for the microscopic and approximate macroscopic model with reaction coordinate ξ_1 (7.10). The approximate macroscopic model makes a modelling error in the mean, also visualized by the difference in amplitude in the two peaks.

To conclude, it is important to take a good reaction coordinate to capture the bimodality in (7.4). However, choosing a decent reaction coordinate in advance is not always straightforward. The micro-macro acceleration scheme will reduce this difficulty.

7.2.2 Improvements by micro-macro acceleration

For the micro-macro acceleration scheme it is important to select adequate state variables in advance for accuracy. What states to select also depends on the quantities of interest. In the following experiments, we will choose three different states to extrapolate: the mean θ , the mean of $\|A - C\|^2$ and the first two moments of x_c together with the first two moments of y_c . All three choices are a big improvement over the bad reaction coordinate

ξ_2 and compete with the good reaction coordinate ξ_1 , but some choices are still more accurate than others.

Extrapolating the reaction coordinates Figure 7.10 depicts the means and histograms of θ and $\|A - C\|^2$ obtained by the micro-macro acceleration scheme with only extrapolating the mean of θ , for several extrapolation step sizes. Micro-macro follows the exact microscopic dynamics very well for the angle as well as the distance, regardless of the extrapolation step. The acceleration method gets completely rid of the modelling error made by reaction coordinate ξ_2 while attaining the same accuracy in θ as reaction coordinate ξ_1 . This is a remarkable result since the distance is not part of the extrapolation, $\|A - C\|^2$ is only a function of θ when x_a and r_c are slaved to l_{eq} . Even if we only extrapolate the mean of $\|A - C\|^2$ in Figure 7.11, the micro-macro acceleration method again approximates the mean of θ and $\|A - C\|^2$ very well, for rather large extrapolation steps. This is another strong result, since ξ_2 is a bad reaction coordinate for the approximate macroscopic model, but not for the micro-macro acceleration technique. There is hence no a priori choice of one reaction coordinate of the two over the other, in the context of the acceleration scheme.

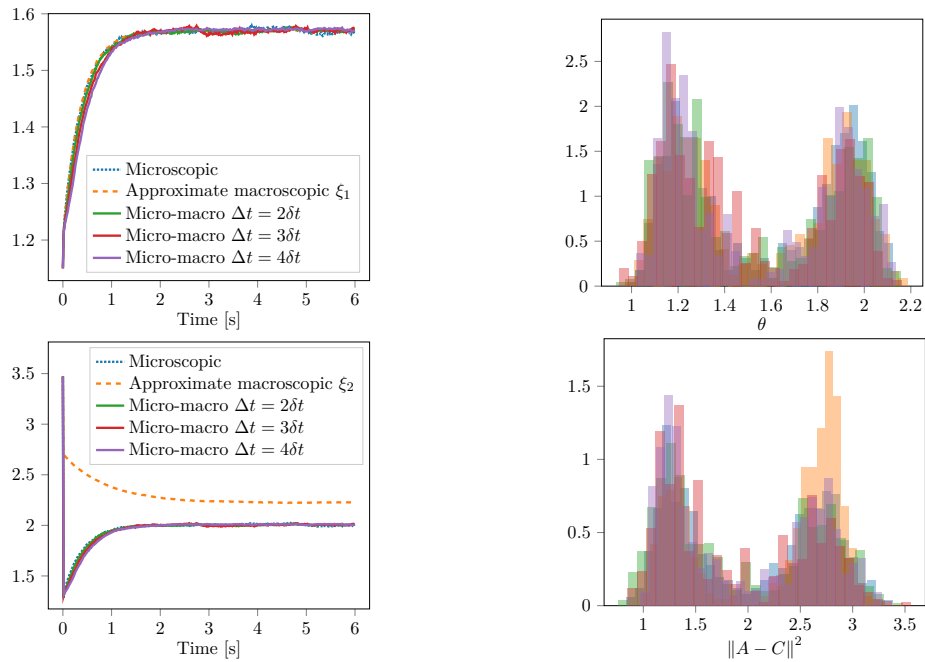


Figure 7.10: The evolution of the mean of θ (top left) and the histogram of θ after 6 seconds (top right) for the microscopic model, approximate macroscopic model with ξ_1 and the micro-macro acceleration method with only extrapolating the mean of ξ_1 , for several extrapolation steps. The bottom plots are the same but show the evolution and histograms of $\|A - C\|^2$ with the approximate macroscopic model with reaction coordinate ξ_2 . The micro-macro acceleration scheme does not make a modelling error.

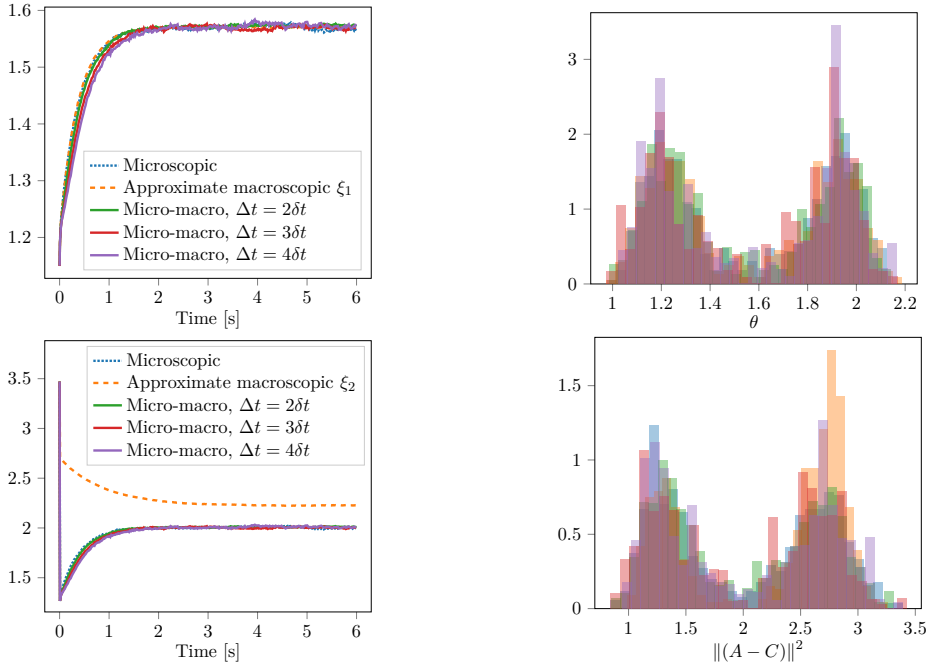


Figure 7.11: The evolution of the mean of θ (top left) and the histogram of θ after 6 seconds (top right) for the microscopic model, approximate macroscopic model with reaction coordinate ξ_2 and the micro-macro acceleration method with only extrapolating the mean of ξ_1 , for several extrapolation steps. The bottom plots are the same but show the evolution and histograms of $\|A - C\|^2$ with the approximate macroscopic model with reaction coordinate ξ_2 . The micro-macro acceleration scheme again does not make a modelling error.

Extrapolating moments of x_c and y_c Besides one of the reaction coordinates, it is also possible to extrapolate moments of x_c and y_c since these two variables determine the angle θ completely. Figure 7.12 shows the results of the micro-macro acceleration scheme with the first two moments of x_c , y_c after six seconds and 10^4 particles. Again, the acceleration scheme approximates the exact mean and histogram of θ and $\|A - C\|^2$ very well, but the error increases faster with the extrapolation step than during the previous experiments. We already saw for the FENE-dumbbells model in Section 5.3 that extrapolating raw moments may not give the most accurate simulation results. In case of the tri-atom molecule, the variable y_c is purely fast and extrapolating this moment further may yield less accurate results. As a result, the micro-macro acceleration method performs slightly worse than the approximate macroscopic model on the variable θ but still removes the modelling error of ξ_2 on $\|A - C\|^2$. An adaptive time stepping strategy based on accuracy could alleviate the small error of the micro-macro acceleration method. Note that in this experiment, the noise amplitude is also larger, which is an effect for further investigation.

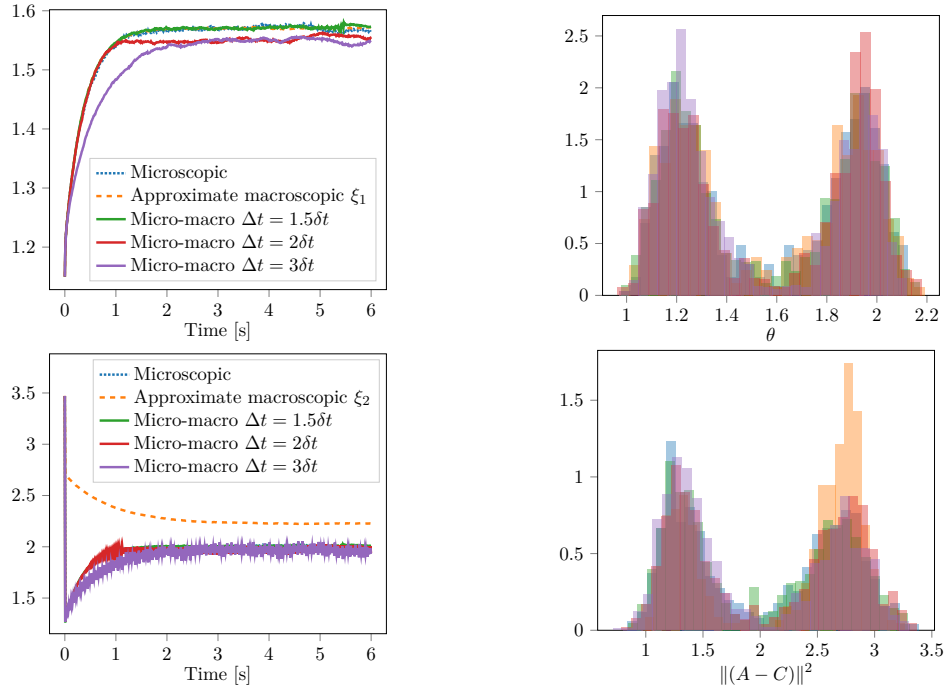


Figure 7.12: The evolution of the mean of θ (top left) and the histogram of θ after 6 seconds (top right) for the microscopic model, approximate macroscopic model with ξ_1 and the micro-macro acceleration method with the first two moments of x_c and y_c , for several extrapolation steps. The bottom plots are similar but display the evolution and histograms of $\|A - C\|^2$ with the approximate macroscopic model with reaction coordinate ξ_2 .

Conclusion The micro-macro acceleration scheme is a big improvement over the approximate macroscopic dynamics. Regardless of the state variables in the extrapolation operator, the acceleration algorithm approximates the microscopic dynamics very accurately. The micro-macro acceleration method does not make a modelling error when compared to the approximate macroscopic model with the bad reaction coordinate, and has the same accuracy in case of a good reaction coordinate. There is hence a priori no best choice of which states to include in the extrapolation operator. For some choices of macroscopic states however, the error increases more rapidly with the extrapolation step. The third experiment is an example of this behaviour. An adaptive time stepping procedure could be of great value for the third experiment to reduce the error. The micro-macro acceleration algorithm is hence a great substitute for the approximate macroscopic models, while being more efficient than the microscopic simulation.

Chapter 8

Conclusion and Outlook

The objective of this thesis consists of two main parts. The first part introduces the micro-macro acceleration algorithm consisting of four steps and discusses its convergence, matching operators and proposes an efficient object-oriented implementation. Stability of the micro-macro acceleration scheme is also discussed. The second main part of the thesis is original material and presents an extensive analysis on the efficiency of micro-macro acceleration. One aspect of the efficiency analysis is the effect of the number of macroscopic states to extrapolate. The second aspect investigates how large the extrapolation step of the micro-macro acceleration method can be before approximate macroscopic models become more accurate, as a function of the time-scale separation.

8.1 Results

In Chapter 4, we prove a new convergence result for the micro-macro the acceleration algorithm in the case of linear SDEs, with only slow-mean extrapolation. The theorem is a new result since the general convergence theorem 1 requires an infinite hierarchy of macroscopic state variables that completely describe the underlying distribution. In the new result however, there is only one slow moment. Also for linear SDEs, in Chapter 5 we discuss the most accurate selection of macroscopic states for micro-macro acceleration approximation, without performing superfluous work. Extrapolating the slow mean and variance gives most accurate results with the least computational work possible. For non-linear examples, choosing an efficient number of state variables is much more involved and Chapter 5 presents a case-study on FENE-dumbbells. A similar study was already performed for lifting, but not for matching. For FENE-dumbbells, the stress tensor is of great importance and choosing the state variables as states in the evolution equation of the stress yields good results for a modest number of states.

For the second part of the efficiency analysis, Chapter 6 finds that the maximal extrapolation factor M increases with decreasing ε , for a linear driven SDE. For moderate time-scale separations, M stays relatively constant. This is an interesting result since it relies on the hyper-convergence of the micro-macro acceleration scheme. In Chapter 6 we

prove that the micro-macro acceleration scheme can take larger steps than a microscopic simulation, while being more accurate than the approximate macroscopic model.

Finally, in Chapter 7 we present two more cases where with a similar conclusion. For the bimodal problem, there is an intermediate regime of intermediate ε and large slow noise levels where the macroscopic model makes a large modelling error and the micro-macro acceleration method does not. For smaller ε or larger noise variances, the macroscopic model is very accurate, making it hard to see the gain of the micro-macro acceleration scheme. For the tri-atom molecule, the micro-macro acceleration scheme follows the exact transient dynamics and steady-state of the two considered reaction coordinates up to discretization errors. This is the case even when the extrapolation operator only contains the mean of the bad reaction coordinate. In case of a bad reaction coordinate, the approximate macroscopic model makes a modelling error and the micro-macro acceleration scheme does not, clearly proving the merit of the new acceleration algorithm. Only when extrapolating the first two moments of atom C , the micro-macro acceleration scheme makes small transient errors and exhibits larger noise amplitudes, which is the topic of further research. All three applications from Chapter 6 and 7 prove that there are important problems where the micro-macro acceleration algorithm is more accurate than current state-of-the-art approximate macroscopic models, while being more efficient than a brute-force microscopic simulation.

8.2 Outlook to future work

Although the thesis proves the merit of the micro-macro acceleration algorithm, a couple of important questions need further attention. In the experiments on the bimodal problem and FENE-dumbbells, there is a larger transient deviation between the microscopic simulation and the micro-macro acceleration scheme. A possible solution is to devise an adaptive procedure, based on accuracy. An algorithm that adaptively changes the extrapolation step size or the number of macroscopic state variables, could greatly improve the transient approximations. Until now, adaptivity only activates when there is a matching failure.

Second, an analysis of how noise propagates through the micro-macro acceleration scheme could deepen the understanding of the algorithm. In case of the tree-atom molecule, when extrapolating the first and second moment of the position of molecule C , there is a much larger noise variance than with the other macroscopic state variables.

Finally, there are many more important problems in science and industry that are high dimensional, where brute-force simulations take too much time. An example of such a situation occurs when simulating the kinetic equations in fusion reactors [11]. Applying the acceleration in similar contexts could be of great value.

Appendices

Appendix A

Finite volume method for the Fokker-Planck equation

Section 5.1 (5.1) uses the results of simulating the Fokker-Planck equation (5.1) associated to a linear SDE. This appendix contains the derivation of a finite volume scheme for the two-dimensional Fokker-Planck PDE. The content is largely based on [26].

Consider a general linear two-dimensional SDE with additive noise:

$$\begin{aligned}dX &= (aX + bY)dt + D_x dW_x \\dY &= (cX + dY)dt + D_y dW_y\end{aligned}$$

where $W_x(t)$ and $W_y(t)$ are Wiener processes. The Fokker-Planck equation describing the joint density $\rho(x, y, t)$ is then given by

$$\partial_t \rho + \partial_x((ax + by)\rho) + \partial_y((cx + dy)\rho) = \frac{1}{2}(D_x \partial_{xx} \rho + D_y \partial_{yy} \rho)$$

We want that the total mass is numerically conserved, being equal to one. Therefore we will use a finite-volume method to discretize the drift term, while using a finite difference scheme for the second order diffusion term. The following section contains a derivation of a first order scheme for the Fokker-Planck with a general drift term.

$$\partial_t \rho + \partial_x F(\rho) + \partial_y G(\rho) = \frac{1}{2}(D_x \partial_{xx} \rho + D_y \partial_{yy} \rho)$$

A.1 Derivation of a finite volume scheme

Suppose for simplicity that we are using a rectangular grid $C_{i,j}$ with box dimensions $(\Delta x, \Delta y)$, and central points $(x_i, y_j) = ((i - \frac{1}{2})\Delta x, (j - \frac{1}{2})\Delta y)$ in the middle of each rectangle, as in Figure A.1. Finite volume methods are based on the integral form of the PDE. Integrating the Fokker-Planck equation over a small rectangle $C_{i,j}$ yields

$$\frac{d}{dt} \int_{C_{i,j}} \rho(x, y, t) dx dy + \int_{C_{i,j}} \nabla \cdot (F(\rho), G(\rho)) dx dy = \frac{1}{2} \int_{C_{i,j}} D_x \partial_{xx} \rho(x, y, t) + D_y \partial_{yy} \rho(x, y, t) dx dy.$$

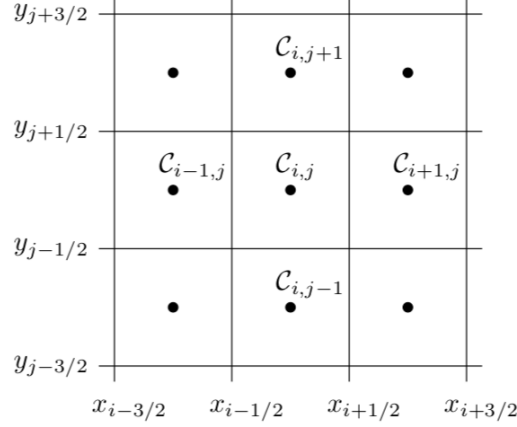


Figure A.1: The two-dimensional finite volume grid. This figure was taken from [26].

By Green's theorem, the surface integral of the divergence of $(F(\rho), G(\rho))$ equals the line integral around $C_{i,j}$, counter clockwise, of the vector field $(F(\rho), G(\rho))$ itself. Filling this in yields

$$\begin{aligned} \frac{d}{dt} \int_{C_{i,j}} \rho(x, y, t) dx dy + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} F(\rho(x_{i+\frac{1}{2}}, y, t)) - F(\rho(x_{i-\frac{1}{2}}, y, t)) dy + \\ \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} G(\rho(x, y_{j+\frac{1}{2}}, t)) - G(\rho(x, y_{j-\frac{1}{2}}, t)) dx = \frac{1}{2} \int_{C_{i,j}} D_x \partial_{xx} \rho(x, y, t) + D_y \partial_{yy} \rho(x, y, t) dx dy. \end{aligned} \quad (\text{A.1})$$

Before discretizing the integrals, let's first introduce the cell average of $\rho(x, y, t)$ over a cell $C_{i,j}$ as

$$\bar{\rho}_{i,j}(t) = \frac{1}{\Delta x \Delta y} \int_{C_{i,j}} \rho(x, y, t) dx dy.$$

We can now approximate each of different terms occurring in (A.1). For the diffusion we use standard finite differences, reading

$$\partial_{xx} \bar{\rho}_{i,j} \approx \frac{\bar{\rho}_{i-1,j} - 2\bar{\rho}_{i,j} + \bar{\rho}_{i+1,j}}{\Delta x^2}$$

and similarly

$$\partial_{yy} \bar{\rho}_{i,j} \approx \frac{\bar{\rho}_{i,j-1} - 2\bar{\rho}_{i,j} + \bar{\rho}_{i,j+1}}{\Delta y^2}$$

while for the drift terms we introduce the numerical fluxes on the edges of $C_{i,j}$

$$\begin{aligned} F_{i\pm\frac{1}{2},j}(t) &\approx \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} F(\rho(x_{i\pm\frac{1}{2}}, y, t)) dy \\ G_{i,j\pm\frac{1}{2}}(t) &\approx \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} G(\rho(x, y_{j\pm\frac{1}{2}}, t)) dx \end{aligned}$$

where a implicit midpoint rule suffices for an accurate computation of the flux integrals

$$\begin{aligned} F_{i\pm\frac{1}{2},j}(t) &= F(\rho(x_{i\pm\frac{1}{2}}, y_j, t)) \\ G_{i,j\pm\frac{1}{2}}(t) &= G(\rho(x_i, y_{j\pm\frac{1}{2}}, t)) \end{aligned}$$

Putting it all together, the finite volume approximation of the Fokker-Planck equation becomes

$$\begin{aligned} \frac{d}{dt}\bar{\rho}_{i,j} &= -\frac{1}{\Delta x}(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) - \frac{1}{\Delta y}(G_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}) \\ &\quad + D_x \frac{\bar{\rho}_{i-1,j} - 2\bar{\rho}_{i,j} + \bar{\rho}_{i+1,j}}{\Delta x^2} + D_y \frac{\bar{\rho}_{i,j-1} - 2\bar{\rho}_{i,j} + \bar{\rho}_{i,j+1}}{\Delta y^2} \end{aligned}$$

Right now the Fokker-Planck PDE is reduced to a system of ODEs that can be integrated over time using any standard technique for ODEs like Runge-Kutta methods or alike. A simple forward Euler will work nicely too. The complete discrete finite volume approximation thus reads

$$\begin{aligned} \bar{\rho}_{i,j}^{n+1} &= \bar{\rho}_{i,j}^n - \frac{\Delta t}{\Delta x}(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) - \frac{\Delta t}{\Delta y}(G_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}) \\ &\quad + D_x \frac{\Delta t}{\Delta x^2}(\bar{\rho}_{i-1,j} - 2\bar{\rho}_{i,j} + \bar{\rho}_{i+1,j}) + D_y \frac{\Delta t}{\Delta y^2}(\bar{\rho}_{i,j-1} - 2\bar{\rho}_{i,j} + \bar{\rho}_{i,j+1}). \end{aligned}$$

A.2 Boundary conditions

In principle, the invariant distribution of a linear model is a two-dimensional Gaussian that stretches out all over the plane \mathbb{R}^2 . The bell-curve decays very rapidly so that we can approximate the plane \mathbb{R}^2 by a square grid $[x_1, x_2] \times [y_1, y_2]$ so that the transient and invariant distributions are very close to zero on the boundary of the rectangular grid. A numerical computation on this finite grid also requires boundary conditions. Mass conservation is very important in the context of probability distributions, which demands for no-flux boundary conditions. No flux on the boundaries requires that the numerical fluxes $F_{i\pm\frac{1}{2},j}$ and $G_{i,j\pm\frac{1}{2}}$ are zero on the boundary. Also for second order derivatives near the boundary, the no flux condition implies that $G_{i,j\pm\frac{1}{2}} = 0$ on the lower and upper boundary and $F_{i\pm\frac{1}{2},j} = 0$ on the left and right boundary.

A.3 Order test to verify a correct implementation

One way to verify the correctness of the finite volume implementation is by computing the error between the numerical solution and analytical solution for different grid sizes and temporal steps. The error should decrease with a certain order, given by the numerical scheme. We will use the standard forward Euler method so the error should decrease linearly in time and quadratically in space, due to the central implicit midpoint scheme to approximate the fluxes and central differences for the second order derivatives. Both

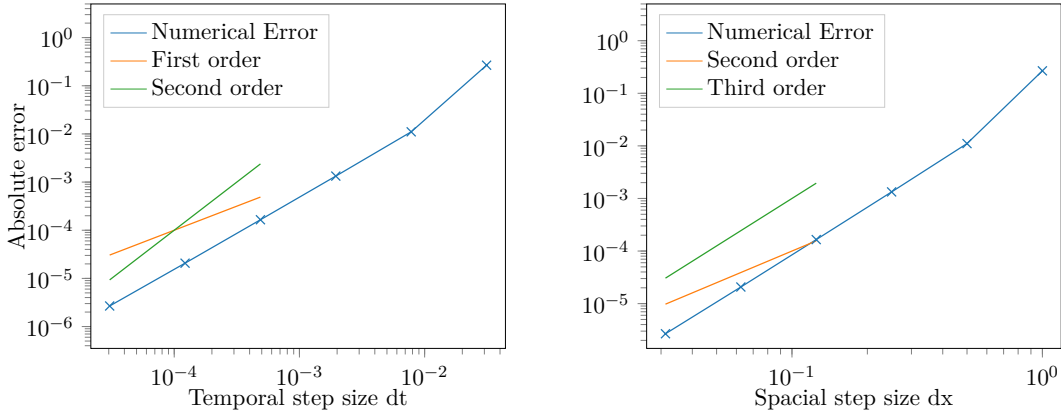


Figure A.2: Convergence of the error as a function of the temporal and spatial step size for a fixed end time for the Fokker-Planck equation. The figure on the left shows more than first order convergence in time and the graph on the right depicts more than second order in space, as it theoretically should be.

approximations are second order in space. Figure A.2 depicts the error as a function of the temporal and spatial step for a Gaussian initial condition with mean $\mu_x, \mu_y = 1, 2$ and covariances $\Sigma_x, \Sigma_y, \Sigma_{x,y} = 1, 1, 0$ and final simulation time 0.5 seconds. The time step dt scales quadratically with the spatial grid size dx to keep the numerical scheme stable. Figure A.2 also shows better than second order convergence in space, it even is third order, and similarly the convergence in time is better than first order. The higher-order convergence is called *hyper-convergence* but more research is needed on why hyper-convergence manifests on a linear PDE. Both figures confirm a correct implementation of the finite volume approximation of the Fokker-Planck equation.

Appendix B

Derivation of a closure model for linear SDEs

B.1 Derivation of the closure relations

Section 5.1 explains the concept of closure relations for SDEs. For a general linear process, there exist closed equation for the means and variances alone, without higher order moments appearing. Consider again an n -dimensional linear SDE with additive noise. We can write this as

$$dX = AXdt + DdW$$

where $X \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $D = \text{diag}(D_1, \dots, D_n)$ and $W(t)$ an n -dimensional Wiener process. Instead of simulating the SDE directly, it is possible to derive a system of ODEs describing the propagation of means $\mathbb{E}[X]$ and covariance $\mathbb{E}[(X - \mathbb{E}(X))(X - \mathbb{E}(X))^T]$ of the components of the SDE. In the two dimensional case for instance, the SDE reduces to a system of five ODEs because the covariance matrix is symmetric. The system has a multidimensional Gaussian invariant distribution. In this setting, the invariant mean and covariance describe the invariant distribution completely.

For the mean, we can proceed by taking the expectation of both sides

$$d\mathbb{E}[X] = A\mathbb{E}[X]dt + D\mathbb{E}[dW].$$

By the martingale property in Itô-calculus, the final term is zero and we get the first two relations as $\frac{d\mathbb{E}[X]}{dt} = A\mathbb{E}[X]$. For the covariance, a similar derivation is possible, but Itô's lemma is required. The derivative of the covariance matrix using Itô's lemma reads

$$d(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T = dX(X - \mathbb{E}[X])^T + (X - \mathbb{E}[X])dX^T + \frac{1}{2}DD^T dt.$$

Writing the former expression out further gives us

$$\begin{aligned} d(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T &= \left(AX(X - \mathbb{E}[X])^T + (X - \mathbb{E}[X])X^T A^T + \frac{1}{2}DD^T \right) dt \\ &\quad + dW(X - \mathbb{E}[X])^T + (X - \mathbb{E}[X])dW \end{aligned}$$

Again taking expectations from both sides, the terms with Brownian motion will disappear due to the martingale property, so we end up with

$$d\text{Cov}[X] = \left(A\mathbb{E}[X(X - \mathbb{E}[X])^T] + \mathbb{E}[(X - \mathbb{E}[X])X^T]A^T + \frac{1}{2}DD^T \right) dt.$$

The first two expectation-terms equal the covariance matrix, so the system of ODEs describing the covariance matrix is

$$\frac{d\text{Cov}(t)}{dt} = A\text{Cov}(t) + \text{Cov}(t)A^T + \frac{1}{2}DD^T.$$

In general, higher order terms pop up in the equations describing the covariance matrix. For non-linear SDEs, this is usually the case so that in principle an infinite number of equations are required to describe the intermediate distributions completely. For example, the two dimensional linear process in Chapter 4 case where

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad D = \begin{pmatrix} D_x & 0 \\ 0 & D_y \end{pmatrix}$$

the equations for means and covariance reduce to

$$\frac{d}{dt} \begin{pmatrix} \mu_X \\ \Sigma_X \\ \mu_Y \\ \Sigma_Y \\ \text{Cov}(X, Y) \end{pmatrix} = \begin{pmatrix} a & 0 & b & 0 & 0 \\ 0 & 2a & 0 & 0 & 2b \\ c & 0 & d & 0 & 0 \\ 0 & 0 & 0 & 2d & 2c \\ 0 & c & 0 & b & a + d \end{pmatrix} \begin{pmatrix} \mu_X \\ \Sigma_X \\ \mu_Y \\ \Sigma_Y \\ \text{Cov}(X, Y) \end{pmatrix} + \begin{pmatrix} 0 \\ D_x^2 \\ 0 \\ D_y^2 \\ 0 \end{pmatrix}$$

which can be solved directly or simulated using any standard ODE solver.

B.2 Analytical solution of the closure relations

The system of ODEs from above has a simple analytical solution. Denote the system as

$$\frac{dx(t)}{dt} = Mx + b, \quad x(0) = x_0. \tag{B.1}$$

The steady-state solution is given by $x_\infty = -M^{-1}b$ and if we denote $y(t) = x(t) - x_\infty$, the evolution law for $y(t)$ is simply

$$\frac{dy(t)}{dt} = My(t) \tag{B.2}$$

such that $y(t) = e^{Mt}y_0$ and hence

$$x(t) = x_\infty + e^{Mt}(x_0 - x_\infty). \tag{B.3}$$

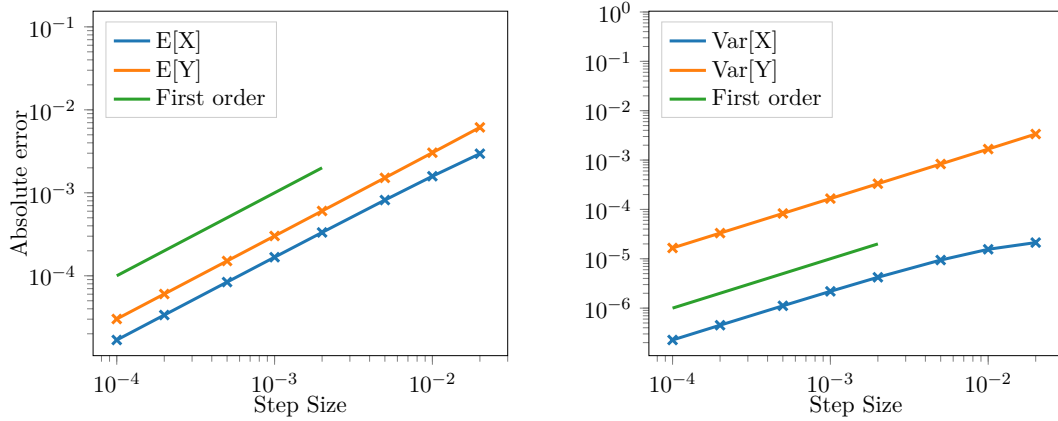


Figure B.1: First order convergence in time for the error mean and variance of the slow and fast variables.


B.3 Convergence test for the forward Euler implementation

Chapter 4 uses a forward Euler implementation of the two-dimensional closure model to compare the accuracy of the matching algorithms with the true solution. It is therefore necessary to be sure that the implementation is correct by carrying out an order test. The following test compares the analytical solution given by equation (B.3) at time $t = 0.5$ seconds with initial condition $[\mu_x, \Sigma_x, \mu_y, \Sigma_y, \Sigma_{x,y}] = [1., 1., 2., 1., 0.]$ with the numerical solution at the same time as a function of the temporal step size. Figure B.1 shows the error of the mean and variance of the fast and slow variable relative to the analytic solution as a function of the step size. The plot shows first order convergence, which is exactly the order of the forward Euler method. This is sufficient proof that the numerical implementation is trustworthy and if the step size is small enough, the closure model is close to the exact solution.

Appendix C

Poster

Figure C.1 shows the poster that was presented at the Master's thesis fair on the 23rd of April at KU Leuven.



**KATHOLIEKE UNIVERSITEIT
LEUVEN**
FACULTEIT
INGENIEURSWETENSCHAPPEN

Master
Wiskundige
Ingenieurstechnieken

Masterproef
Hannes
Vandecesteele

Promotor
Prof. dr. Ir.
Giovanni
Sarmaey

Academiejaar
2017-2018

Efficiency of micro-macro acceleration for scale-separated SDEs

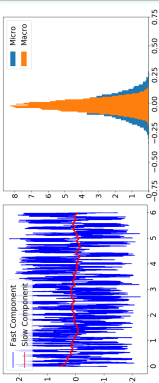
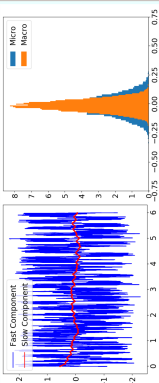
Context

Many systems are inherently stochastic with a strong scale separation ε

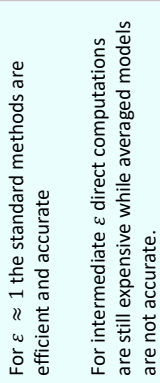
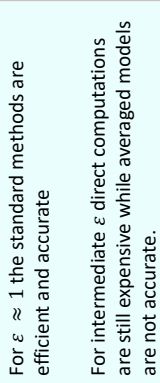
$$dX = -(2X + Y)dt + AdW_x$$

$$dY = -\frac{1}{\varepsilon}(Y^3 - Y)dt + \frac{1}{\sqrt{\varepsilon}}dW_y$$

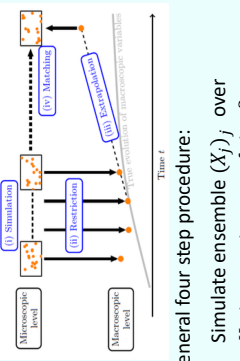
- For $\varepsilon \ll 1$ direct computations are intractable but averaged models exist [2]

- For $\varepsilon \approx 1$ the standard methods are efficient and accurate
- For intermediate ε direct computations are still expensive while averaged models are not accurate.

Micro-macro acceleration [1]



General four step procedure:

- Simulate ensemble $(X_j)_j$ over K microscopic steps of size δt

$$X_j^{n,k+1} = X_j^{n,k} + a(X_j^{n,k})\delta t + b(X_j^{n,k})dW$$

- Record macroscopic moment functions of interest at every step

$$m_l^{n,k} = \mathbb{E}_\mu[R_l(x)]$$

- Extrapolate these moments over a larger step Δt

$$m_l^{n+1} = m_l^n + \frac{\Delta t}{\delta t}(m_l^{n,k} - m_l^n)$$

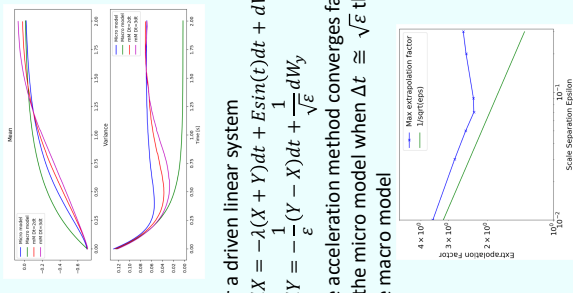
- Find a new distribution at t^{n+1} that is consistent with $(m_l^{n+1})_l$ while minimizing

$$\varphi(x) = \arg \min_{\varphi} \int_G \varphi(x) \ln \left(\frac{\varphi(x)}{\pi(x)} \right) dx$$

It is known that this method converges and the extrapolation stability bound is independent of δt .

Focus of the thesis: Efficiency

- Bimodal model: Micro-macro acceleration removes modelling error



- For a driven linear system

$$dX = -\lambda(X + Y)dt + E \sin(t)dt + dW_x$$

$$dY = -\frac{1}{\varepsilon}(Y - X)dt + \frac{1}{\sqrt{\varepsilon}}dW_y$$

the acceleration method converges faster to the micro model when $\Delta t \approx \sqrt{\varepsilon}$ than the macro model

While for the microscopic model $\delta t \approx \varepsilon$.

- [1] DEBRABANT, Kristian; SAMMEY, Giovanni; ZIELINSKI, Przemyslaw. Micro-macro acceleration for stochastic differential equations. *SIAM Journal on Numerical Analysis*, 2017, 55.6: 2745-2786.
- [2] PAVLIOTIS, Grigorios; STUART, Andrew. *Multiscale methods: averaging and homogenization*. Springer Science & Business Media, 2008.

Figure C.1: The poster.

Bibliography

- [1] Assyr Abdulle and Stéphane Cirilli. Stabilized methods for stiff stochastic systems. *Comptes Rendus Mathématique*, 345(10):593–598, 2007.
- [2] Assyr Abdulle and Stéphane Cirilli. S-ROCK: Chebyshev methods for stiff stochastic differential equations. *SIAM Journal on Scientific Computing*, 30(2):997–1014, 2008.
- [3] Assyr Abdulle, Weinan E, Björn Engquist, and Eric Vanden-Eijnden. The heterogeneous multiscale method. *Acta Numerica*, 21:1–87, 2012.
- [4] Maria Bruna, S Jonathan Chapman, and Matthew J Smith. Model reduction for slow–fast stochastic systems with metastable behaviour. *The Journal of chemical physics*, 140(17):174107, 2014.
- [5] S Burov and E Barkai. Fractional langevin equation: Overdamped, underdamped, and critical behaviors. *Physical Review E*, 78(3):031112, 2008.
- [6] Kevin Burrage, Pamela Burrage, and Tianhai Tian. Numerical methods for strong solutions of stochastic differential equations: an overview. *Proceedings of The Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 460(2041):373–402, 2004.
- [7] George D Byrne and Alan C Hindmarsh. Stiff ode solvers: A review of current and coming attractions. *Journal of Computational physics*, 70(1):1–62, 1987.
- [8] Fabio Chalub, Yasmin Dolak-Struss, Peter Markowich, Dietmar Oelz, Christian Schmeiser, and Alexander Soreff. Model hierarchies for cell aggregation by chemotaxis. *Mathematical Models and Methods in Applied Sciences*, 16(supp01):1173–1197, 2006.
- [9] Kristian Debrabant, Giovanni Samaey, and Przemysław Zieliński. A micro-macro acceleration method for the Monte Carlo simulation of stochastic differential equations. *SIAM Journal on Numerical Analysis*, 55(6):2745–2786, 2017.
- [10] Kristian Debrabant, Giovanni Samaey, and Przemysław Zieliński. Study of micro-macro acceleration schemes for linear slow-fast stochastic differential equations with additive noise. *arXiv:1805.10219*, 2018.

- [11] Werner Ebeling and M Yu Romanovsky. Microfields, kinetic equations and fusion rates in exploding ion clusters. *Contributions to Plasma Physics*, 49(7-8):477–487, 2009.
- [12] Kristel Ghoois, Wouter Dekeyser, Giovanni Samaey, Petra Börner, and Martine Baelmans. Accuracy and convergence of coupled finite-volume/monte carlo codes for plasma edge simulations of nuclear fusion reactors. *Journal of Computational Physics*, 322:162–182, 2016.
- [13] Dror Givon, Raz Kupferman, and Andrew Stuart. Extracting macroscopic dynamics: model problems and algorithms. *Nonlinearity*, 17(6):R55, 2004.
- [14] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.
- [15] Jeroen D Hol, Thomas B Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.
- [16] Edwin T Jaynes. *Probability theory: the logic of science*. Cambridge university press, 2003.
- [17] Jagat Narain Kapur and Hiremaglur K Kesavan. Entropy optimization principles and their applications. In *Entropy and energy dissipation in water resources*, pages 3–20. Springer, 1992.
- [18] Ioannis G Kevrekidis and Panagiotis G Kevrekidis. Equation-free, coarse-grained multiscale computation: Enabling mocrosopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.
- [19] Ioannis G Kevrekidis and Giovanni Samaey. Equation-free multiscale computation: Algorithms and applications. *Annual review of physical chemistry*, 60:321–344, 2009.
- [20] Yoshio Komori and Kevin Burrage. Weak second order S-ROCK methods for stratonovich stochastic differential equations. *Journal of Computational and Applied Mathematics*, 236(11):2895–2908, 2012.
- [21] Yoshio Komori and Kevin Burrage. Strong first order S-ROCK methods for stochastic differential equations. *Journal of Computational and Applied Mathematics*, 242:261–274, 2013.
- [22] Frédéric Legoll and Tony Lelièvre. Effective dynamics using conditional expectations. *Nonlinearity*, 23(9):2131, 2010.

-
- [23] Tony Lelièvre, Giovanni Samaey, and Przemysław Zieliński. Analysis of a micro-macro acceleration method with minimum relative entropy moment matching. *arXiv preprint arXiv:1801.01740*, 2018.
- [24] Tiejun Li, Assyr Abdulle, et al. Effectiveness of implicit methods for stiff stochastic differential equations. 2008.
- [25] Nader Masmoudi. Well-posedness for the fene dumbbell model of polymeric flows. *Communications on Pure and Applied Mathematics*, 61(12):1685–1714, 2008.
- [26] Ward Melis. Projective integration for hyperbolic conservation laws and multiscale kinetic equations. *Phd thesis, Faculty of Engineering Sciences, KU Leuven*, 2017.
- [27] Paul G Mezey. *Shape in chemistry: an introduction to molecular shape and topology*. Wiley-VCH, 1993.
- [28] Dhananjay Nene. A beginners guide to Dependency Injection. URL: <http://www.theserverside.com/news/1321158/A-beginners-guide-to-Dependency-Injection>, Online; last accessed 22-May-2018, 2005.
- [29] Grigoris Pavliotis and Andrew Stuart. *Multiscale methods: averaging and homogenization*. Springer Science & Business Media, 2008.
- [30] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 2008.
- [31] Yoshihiro Saito and Taketomo Mitsui. Stability analysis of numerical schemes for stochastic differential equations. *SIAM Journal on Numerical Analysis*, 33(6):2254–2267, 1996.
- [32] Giovanni Samaey, Tony Lelièvre, and Vincent Legat. A numerical closure approach for kinetic models of polymeric fluids: exploring closure relations for FENE dumbbells. *Computers & Fluids*, 43(1):119–133, 2011.
- [33] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.
- [34] Shi-Tao Yeh et al. Using trapezoidal rule for the area under a curve calculation. *Proceedings of the 27th Annual SAS® User Group International (SUGI02)*, 2002.
- [35] Antonios Zagaris, Christophe Vandekerckhove, C William Gear, Tasso J Kaper, and Ioannis G Kevrekidis. Stability and stabilization of the constrained runs schemes for equation-free projection to a slow manifold. *Discrete and continuous dynamical systems-Series A*, 32(8):2759–2803, 2012.

Fiche masterproef

Student: Hannes Vandecasteele

Titel: Efficiency of micro-macro acceleration for scale-separated stochastic differential equations

Nederlandse titel: Efficiëntie van micro-macro acceleratie voor stochastische differentiaalvergelijkingen met een tijdsschaal separatie

UDC: 621.3

Korte inhoud:

We investigate a new micro-macro acceleration algorithm to simulate stochastic differential equations with an inherent time-scale separation. The method consists of four steps: (i) short Monte Carlo simulation. (ii) The computation of several macroscopic state variables of interest (called restriction). (iii) Extrapolation of these states over a larger time interval. (iv) Finding a new distribution that is consistent with the latter states and deviates the least from the final distribution obtained after the Monte Carlo steps from (i). The final step is called matching. This thesis performs an efficiency analysis of the micro-macro acceleration scheme. It is already known that the scheme converges when the number of state variables tends to infinity and the present time steps decrease to zero. It is also known that the stability bound is independent from the present time-scale separation. In the text, we discuss both convergence and stability and present an object-oriented implementation. For the efficiency analysis, we first look at how the micro-macro acceleration scheme performs on a linear SDE. We present a new convergence theorem in case of scale-separated linear SDEs with only slow-mean extrapolation. We also find that two moments of the slow component of the process gives the most accurate simulation results, without performing too much computational work. We also consider a non-linear example where a priori state selection is more involved. For the second part of the efficiency analysis, we look at what the maximal extrapolation step is before the micro-macro acceleration scheme becomes less accurate than approximate macroscopic models. We find that the maximal extrapolation factor increases for stronger time-scale separations for a linear driven process, which is a new result. Finally, we discuss two practical examples where approximate macroscopic models make a modelling error and where the micro-macro acceleration method follows the exact microscopic dynamics accurately. The micro-macro acceleration scheme is also able to take larger steps than a microscopic simulation, proving the merit of the method.

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: wiskundige ingenieurstechnieken

Promotoren: Prof. dr. ir. G. Samaey
dr. P. Zieliński

Assessoren: Prof. dr. ir. S. Vandewalle
Prof. dr. ir. D. Nuyens

Begeleider: