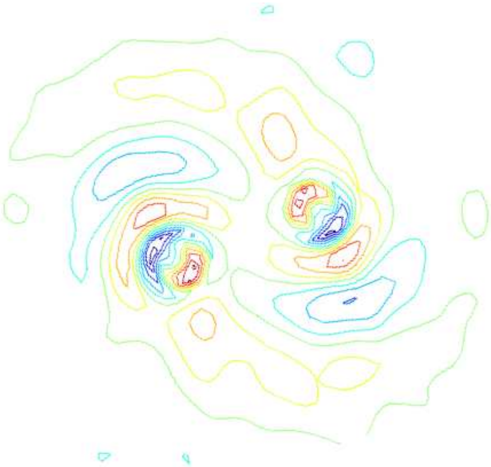
Theoretische Physik **Kondensierte Materie**

Density Matrix Renormalisation Applied to Nonlinear Dynamical Systems



Dissertation
zur Erlangung des Doktorgrades
an der Fakultät für Physik
der Universität Bielefeld
vorgelegt von

Thorsten Bogner

August 2007

Printed on paper ∞ ISO 9706

Contents

Glossary

Notation	**Description**	
$i : j$	Integer numbers from i to j	142
$\cdot^{\dagger}$	Complex conjugated (number), adjungated (matrix)	143
$\cdot^{*}$	Complex conjugated	5
Δ	Laplace operator	20
$[A, B]$	Matrix formed from the columns of matrix A and matrix B. The number of rows in A and B have to be equal.	69
$\lvert \cdot \rvert$	Absolute value	13
$\alpha_i(t)$	Time dependent coefficient	26
$\chi(x)$	Characteristic polynomial	9
δ_{ij}	Kronecker delta	142
Δt	Time step size	17
Δx	Spatial step size	21
$\frac{d}{dt}$	Total derivative with respect to t	13
$\frac{d^n}{dt^n}$	n-th total derivative with respect to t	16
$\frac{\partial^n}{\partial x^n}$	n-th partial derivative with respect to x	16
$\frac{\partial}{\partial x}$	Partial derivative with respect to x	13
$\mathcal{V}$	Enstrophy	103
ϵ_M	Machine precision	126
ϕ	Scalar field	13
$\lvert i \rangle$	Microscopic basis state i	36
$\cdot^{-1}$	Inverse (multiplicative)	8
λ	General scalar	140
μ	Dynamic fluid viscosity	103
∇	Nabla operator	26
$\mathcal{O}(x)$	Landau notation for asymptotic behaviour as x	18
$\cdot^{\perp}$	Orthogonal complement	142
Φ	Scalar field	26
$\hat{\Phi}$	Discretised field	26
ϕ_i	Global ansatz-function	26
ϕ_i^{elm}	Element ansatz-function	27
$\tilde{\Phi}$	Fourier transformed field	30

Notation	**Description**	
π	Ratio of circumference and diameter of a circle in Euclidean space [10, 93] (also known as Archimedes' constant)	31
Ψ	Stochastic system state	37
$\langle \cdot, \cdot \rangle$	Scalar product	141
$\langle \cdot \vert$	Dual of a state vector	40
$\vert \cdot \rangle$	State vector	40
$\sum'$	Sum in which the first and last summand is multiplied by $\frac{1}{2}$	30
$\mathbb{1}$	Unity matrix	142
$\mathbb{1}_k$	$k \times k$ unity matrix	142
$\partial\mathcal{V}$	Boundary of region $\mathcal{V}$	26
ω	Pseudo-scalar vorticity	103
$\boldsymbol{\omega}$	Vorticity	103
A	General matrix	143
a	Creation operator	38
B	General basis	142
$\mathcal{B}$	Basis of a floating point number	125
B^i	Reduced basis after i-th iteration	69
$B^{i\prime}$	Extended basis in the variational POD method	69
$B^{i\prime\prime}$	Orthonormalised extended basis in the variational POD method	69
B_{new}	Test basis in the variational POD method	69
C	Spatial correlation matrix	51
$\mathbb{C}$	Set of all complex numbers	125
CA	Cellular automaton	35
$\mathbf{cos}(x)$	Cosine of x	32
$\mathbf{cot}(x)$	Cotangens of x	32
D	Diagonal matrix	9
$\mathcal{D}_{l/r}$	Left/right diffusion operator	38
$D^2_{d,inhom}$	Operator for the second derivative with inhomogeneous Dirichlet condition	26
$D^{2D}_{x/y}$	Operator for the second derivative with respect to the x/y coordinate in two dimensions	24
DE	Differential equation	13
$\mathbf{det}(A)$	Determinant of a matrix A	9
DFT	Discrete Fourier Transform	30
DMRG	Density matrix renormalisation group	53

Notation	**Description**	
$D_{d,inhom}$	Operator for the derivative with inhomogeneous Dirichlet condition	26
D_x^2	Second derivative with respect to x	21
$D_{x,d}$	Downwind derivative with respect to x	21
$D_{x,s}$	Centred derivative with respect to x	21
$D_{x,u}$	Upwind derivative with respect to x	21
D_d^2	Matrix for the second derivative	22
D_d	Matrix for downwind derivative	22
D_s	Matrix for centred, symmetric derivative	22
D_u	Matrix for upwind derivative	22
E	Exponent bias of a floating point number	125
$\mathcal{E}$	Kinetic energy	103
$\boldsymbol{e}$	Exponent of a floating point number	125
F	General function in phase and parameter space	15
$F_k(t)$	Fourier mode	31
$f(y)$	Function of y	13
FFT	Fast Fourier Transform	30
f_τ	Test function	26
G	Generator of time evolution	16
$g(x)$	General function of x	26
h_t	Time step size	91
i	Imaginary unit, $i = \sqrt{-1}$	5
$\mathcal{J}$	Jacobi operator	20
K	General field	140
$\mathbf{K}$	Forcing matrix	29
k_B	Boltzmann constant	56
$\mathbf{Kern}(\cdot)$	Kernel, null space of an operator	143
KPZ	Kardar-Parisi-Zhang (equation)	1
$\mathbf{kron}(\cdot)$	Kronecker product	24
L	Lower triangular matrix	8
$\mathcal{L}$	General linear transformation	142
l	Litre	35
$\mathbf{ln}(x)$	Natural logarithm of x	32

Notation	**Description**	
M	Mantissa of a floating point number	125
$\mathcal{M}$	Master operator	35
$\boldsymbol{M}$	Mass matrix	29
m	Integer number	10
MR	Model reduction	2
μ	General scalar	142
n_a	Algebraic multiplicity	6
n_g	Geometric multiplicity	6
N	Dimensionality of a vector space, number of degrees of freedom	141
$\hat{\mathbf{n}}$	Normal vector with length 1 perpendicular to some plane	26
n	Number operator	38
N_1	Number of nodes in x-direction	24
N_2	Number of nodes in y-direction	24
N_a	Avogadro number	35
ODE	Ordinary differential equation	13
ONB	Orthonormal basis	2
P	General projection operator	7
PDE	Partial differential equation	1
P_i	Probability for state i	40
$p(\|i\rangle)$	Probability for the i-th microscopic configuration	36
POD	Proper orthogonal decomposition	3
POD-DMRG	Proper orthogonal decomposition density matrix renormalisation group	65
P_s	Standard atmosphere pressure at sea level	35
R^*	Gas constant	35
$\mathbb{R}$	Set of all real numbers	125
Range$(\cdot)$	Range of an operator	143
$\Re\cdot$	Real part	37
Re	Reynolds number	103
$\mathcal{S}$	Source operator	38
$\boldsymbol{s}$	Sign bit	125
s	Line element	15
SAD	Source-annihilation-diffusion (process)	1
Σ	Diagonal matrix	6
sin(x)	Sine of x	32
S_N	Periodic sinc function	32

Notation	**Description**	
$\mathcal{S}^+$	Sink operator	38
$\mathbf{span}(\cdot)$	Subspace spanned by the listed vectors	6
t	Time variable	13
$\mathbf{tan}(x)$	Tangens of x	32
T_s	Standard atmosphere temperature at sea level	35
U	Upper triangular matrix	8
$\boldsymbol{U}$	Orthogonal matrix	6
$U(t)$	Time evolution operator	16
V	Vector space	140
$\mathcal{V}$	Region in space or in phase space	13
$\boldsymbol{V}$	Orthogonal matrix	6
v	Vacancy operator	38
W	General subspace	142
X	Space domain	22

1. Introduction

Dynamical systems arise in most fields of physics, as well as in mathematics, biology [86], economy [81] and essentially in all problems for which a quantitative description of a time evolution is considered. In many cases the dynamical systems are nonlinear, to the effect that linear combinations of solutions are no solutions of the system any more. This is a severe restriction and often makes the use of numerical approaches unavoidable. Closed form solutions are typically only known (if at all) for special cases, which are in addition often not of any practical interest.

The problem of nonlinear dynamics is old and for many decades much work in very different fields such as mathematics, physics and engineering has been devoted to it. This was also motivated by the great practical impact of the results, e.g. for aero/hydrodynamics applications in meteorology, aviation, marine engineering, architecture, etc. Despite the many successes obtained in these fields still no complete understanding of nonlinear dynamical systems is reached. Many questions are still unanswered today, e.g. it is not known whether the Euler-equation (describing inviscid flow) exhibits singularities for finite times [94, 9]. For numerical approaches, e.g. the possibility of chaotic behaviour is a severe restriction for the simulation of such systems. The exponential deviation between two initially close trajectories, indicated by a positive Lyapunov exponent renders predictive long term simulations practically useless due to unavoidable discretisation and rounding errors. In fact such problems led to the quite late discovery of deterministic chaos [80]. In many practical applications such as the mixing of fluids or boundary layer problems, the phenomenon of turbulence is of great importance, but is yet very difficult to describe. In practice, even something like stirring a cup of coffee is still a challenging task to be simulated properly. The progress of computer hardware has also lead to a demand for efficient numerical procedures for simulating nonlinear dynamical systems.

The analysis starts with the study of models that arise from a microscopical description. Many physical systems that are described by partial differential equations (PDEs) are in fact derived from much more complicated microscopic dynamics. The detailed model has in general far too many degrees of freedom to be treated directly and the details are often even irrelevant for the macroscopic behaviour. An example for this is an ideal gas. I will consider simplified models of a reaction diffusion process and surface deposition. The reaction diffusion process was investigated as analytically solvable model system for fundamental research on non-equilibrium statistical mechanics [77, 106]. For the surface deposition process a modified version of a lattice model [71] for the Kardar-Parisi-Zhang (KPZ) equation [66] is considered. To describe the microscopic dynamics a stochastic approach is followed, which has some parallels to quantum mechanics. In this approach the state vectors have a probabilistic interpretation and obey a dynamics described by the so called master equation. The reason for this ansatz is that then all systems are inherently linear, as in quantum mechanics. On the other hand they are also, even in cases where the classical system is finite-dimensional, typically infinite-dimensional. In a way that will be described later the linearity is achieved by a drastic increase of the phase space dimensionality. High-dimensional linear descriptions are obtained in a similar way for dynamical systems that are not linear. In the one-dimensional case density matrix methods allow an analysis of the long term dynamics despite the large dimensionality of the models. This approach has been previously pursued by Carlon, Henkel and Schollwöck [27, 28] and Rodriguez-Laguna [38] for calculating the ground-state of a source-annihilation-diffusion(SAD) process. I propose the real Schur extension to access also transient states whose calculation has proven to be problematic in recent investigations [27].

In the second part of this work model systems are studied, which are based on partial differential equations (PDEs). These equations constitute an own topic in mathematics. In physics most dynamic processes are (or can be) described in the 'language' of PDEs which are basically a cornerstone in quantitative descriptions of the physical world. They occur in quantum mechanics, e.g. the Schrödinger equation [33, 103] as well as the Dirac equation [15]. Further examples are the Einstein-Hilbert equation [84] in general relativity. Also continuum mechanics [75], the hydro/aerodynamics already mentioned above and the electromagnetic field are described by partial differential equations. From a PDE it is in general not possible to reconstruct the microscopic dynamics, even if the equation are derived from them. Consequently, getting a stochastic description out of a PDE is difficult, if not impossible. Therefore the master equation approach cannot easily be extended systematically to a model determined by a PDE. It is merely a tool for modelling the other way around, namely starting from the microscopic description as exemplified in Chapter 8. Thus it is of little use for practical applications. One important and often the only feasible way to treat partial differential equations is a numerical analysis. This most general approach is followed, thus exclusively finite-dimensional systems are treated, which are discretised versions of PDEs. Discretisation is unavoidable for numerical treatment.

Numerical simulations of nonlinear dynamical systems are a topic of its own right. One particular difficulty is the high dimensionality often required to obtain reasonable accurate results. This problem can be mitigated by model reduction (MR) methods, which aim at finding a model system that - while describing some features of the original system sufficiently accurately - has a much smaller dimensionality [4]. Beside an effective reduction, the success and failure of MR-methods can also contribute to a better understanding of the particular dynamical system to be investigated. The MR-methods considered here use an orthogonal projection of the original system's phase space. This is motivated by the fact, that in this case the reduced system is determined by the original system and a suitably chosen orthonormal basis (ONB) of the original phase space only.

To find such an orthonormal basis a newly devised approach is followed. The new method is also motivated by DMRG, but is based on the proper orthogonal decomposition (POD) [4], which is a standard MR-method. The proper orthogonal decomposition uses sample trajectories to calculate a reduced basis. Therefore, it is necessary to simulate the unreduced system in order to obtain these sample trajectories. My new approach uses analogies between a DMRG single-particle algorithm and the POD to calculate the reduced basis without ever simulating the full system. While the original POD is of order $\mathcal{O}(N^3)$ in the system size N my method is principally of order $\mathcal{O}(N)$. In this part of the work the generality of the resulting algorithm is emphasised concerning the model equations to be modelled as well as the numerical methods for integrating these equations. The new method is derived in one-dimensional version which is closely related to single-particle DMRG methods. This algorithm is already quite flexible in the model equations to be processed. Also the discretisation and numerical solution methods can be chosen relatively freely. Consequently, several model equations are considered, ranging from the linear diffusion equation(for pedagogical reasons) to the Burgers and nonlinear diffusion equation.

The extension to higher dimensions is based on the DMRG algorithm proposed by Delgado et al. [87]. The resulting method can be described best as a variational method to calculate a proper orthogonal decomposition. In this work only the two-dimensional, incompressible Navier-Stokes equation is considered. Variational methods have already been applied to the steady, two-dimensional Euler equation [7]. The two-dimensional Navier-Stokes equations shows already a high complex behaviour and its numerical treatment is nontrivial. On the other hand much work has already been done on this problem and the restriction to two dimensions reduces the numerical effort significantly. Therefore this example is well suited to exemplify the new algorithm. However, the method is by no means restricted to two-dimensional systems. As in the previous case it is also very easy to alter the basic equations, the discretisation or the numerical solution method.

Outline of the thesis

The thesis can roughly be divided into two parts. In the first one some notation and known methods are presented. In the second part three new methods are introduced. In Chapter 2 a short summary of some matrix decomposition methods is given. Also some notations are introduced. An overview on partial differential equations is given in Chapter 3. Numerical approaches to ordinary and partial differential equations are also treated there and the discretisation techniques used in this work are presented. Due to the high complexity of this field only the most relevant points can be mentioned. This includes the basic terminology and an outline of those questions that are relevant for the methods developed later on.

Chapter 4 gives an introduction to a stochastic description based on microscopic dynamics. There the dynamics of the system is characterised by the master equation, which is linear. The linearity is obtained to the cost of a huge increase of the dimensionality of the phase space. Also model reduction leads to randomness within this approach which is in agreement with an interpretation of random behaviour by missing information.

Chapter 5 addresses dynamical systems and model reduction methods for dynamical systems. Concerning dynamical systems only a few conventions and definitions are presented, as no explicit use of dynamic system theory is made. The focus here lies on model reduction methods applied to dynamical systems. In particular orthogonal projection methods are of interest, which will be used exclusively throughout this work. This includes the proper orthogonal decomposition (POD) as projection method for nonlinear systems.

The density matrix renormalisation group is motivated and presented in Chapter 6. Here the basic concepts and language are introduced.

The second part of the thesis starts with Chapter 7. There, three new applications are introduced. First the Schur DMRG approach to the master equation for microscopic dynamics is exemplified. The basic idea there is to consider Schur vectors which are orthonormal instead of eigenvectors. The second novel method is the DMRG approach to the proper orthogonal decomposition. It is derived from standard single particle DMRG methods and allows to study also nonlinear (discretised) PDEs. Finally, a variational POD algorithm is presented. It has the advantage to be applicable also to higher dimensional problems or more complicated discretisation schemes. These methods are applied to some appropriate model problems in the following three chapters. Each chapter starts with a short introduction to the particular model, followed by a presentation of the results. For the real Schur DMRG method the models treated in Chapter 8 are the source-diffusion-annihilation(SAD) process and a lattice model for surface deposition related to the Kardar-Parisi-Zhang-equation. In addition to the steady state long living transitional states are calculated. In Chapter 9 the DMRG POD method is used to study several one-dimensional systems as the diffusion equation, the Burgers equation and a Fisher-type equation. The variational POD algorithm is studied in Chapter 10. As model the 2D incompressible Navier-Stokes equations are used, which are shortly presented in advance. Chapter 11 summarises the effectiveness of the different approaches we pursue and discusses further applications of the methods.

In Appendix A the topic of finite numerical precision is touched briefly. A proof for the sufficient conditions for an optimal reduction in the short and long time limit is contained in Appendix B for the special case of linear dynamical systems. Details on the ordering algorithm for real Schur forms is given in Appendix C. Appendix D finally summarises some facts from linear algebra.

2. Linear Matrix Decompositions

This chapter is devoted to a recapitulation of some facts from the field of linear algebra. In particular, some basic decompositions which will become important in this work are presented here. The details of this decompositions are to some extent crucial for the argumetation in the following chapters. Further, they cannot be fully ranked among the canonical physical knowledge. Concerning e.g. matrix diagonalisation, one would assume the matrix to be Hermitian in most physical exmples. Additional information on linear algebra and the notation and conventions in this work can be found in Appendix D.

I give no proof of the statements below, they are only listed to recall and summarise some important facts. A good treatment of these points including proofs and also some numerical remarks are given in the book of Golub and Van Loan [50].

2.1. Diagonalisation of a Matrix

An eigenvector/state v with corresponding eigenvalue μ of a matrix L is defined by

$$Lv = \mu v. \tag{2.1}$$

The eigenvalues are the zeros of the characteristic polynomial $\chi(x)$, the determinant (see Sec. 2.6) of $L - x\mathbb{1}$. For a $N \times N$-matrix this is of order N, so maximal N distinct eigenvalues can exist. Whenever the characteristic polynomial factorises to linear factors, it is possible to find such an eigensystem. Then the eigenvectors constitute a basis B_e in which the matrix L is diagonal, i.e.

$$\hat{L} = B_e^\dagger L B_e = \begin{pmatrix} \mu_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mu_N \end{pmatrix}. \tag{2.2}$$

The eigenvalues can be ordered arbitrarily. The eigenbasis is orthonormal if L is normal $L^\dagger L = LL^\dagger$ [50]. This is in particular true for hermitian matrices H for which $H = H^\dagger$ holds. For hermitian matrices all eigenvalues are real.

Non-symmetric real matrices are in general not diagonalisable in $\mathbb{R}$. Then the characteristic polynomial still decomposes to linear factors over $\mathbb{C}$. In this case and if L is non-defective[1], a complex diagonalisation of the form of Eq.(2.2) exists.

[1] For defective matrices only a decomposition to Jordan block [50] form is possible, which will not be considered here.

Then the eigenvalues occur in complex conjugated pairs $\lambda = \gamma + i\mu$, $\lambda^\dagger = \gamma - i\mu$. The eigenvector is $v = y + iz$. In this case one can obtain an 'almost' diagonal real decomposition if y and z instead of v and v^* [2] are used in the decomposition Eq.(2.2). The decomposed matrix is then block diagonal with 1×1 or 2×2 blocks. The 2×2 blocks arise from the complex eigenvalues and have the form

$$\begin{pmatrix} \gamma & \mu \\ -\mu & \gamma \end{pmatrix}. \tag{2.3}$$

This relation between complex and real arithmetics is also used in the real Schur decomposition.

For a defective matrix at least one eigenvalue is degenerated, i.e. is a multiple root of the characteristic polynomial. The grade of degeneration is termed the algebraic multiplicity n_a. Further, the dimensionality of the corresponding eigenspace is termed the geometric multiplicity n_g. The matrix A is defective in case of $n_g < n_a$ for some eigenvalue. Then, the range of $A \ \in \mathbb{R}^{n\times n}$ is not the full space $\mathbb{R}^n$. Concerning model reduction one could analyse $\hat{A} := Q^\dagger A Q$ instead of A, where Q contains the orthonormal bases of all eigenspaces of A. By construction $\hat{A}$ is non-defective. Thus, also defective matrix fit in the framework of this thesis. However, we will not encounter such matrices in the following.

2.2. Singular Value Decomposition

More general than the diagonalisation above is the so called singular value decomposition SVD. For a rectangular matrix $A \in \mathbb{R}^{m\times n}$ there exist orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ such that

$$\boldsymbol{U}^\dagger A \boldsymbol{V} = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \end{pmatrix} =: \Sigma, \;\; p = \min\{m,n\}. \tag{2.4}$$

$\boldsymbol{U}$ and $\boldsymbol{V}$ can be chosen so that

$$\sigma_1 \geq \ldots \geq \sigma_p \geq 0 \tag{2.5}$$

which will be assumed in the following. The σ_i are termed singular values of A. If σ_r is the smallest nonzero singular value one obtains

$$\begin{aligned} \mathrm{Kern}(A) &= \mathrm{span}(\mathbf{v}^{r+1}, \ldots, \mathbf{v}^n), \qquad (2.6)\\ \mathrm{Range}(A) &= \mathrm{span}(\mathbf{u}^1, \ldots, \mathbf{u}^r), \qquad (2.7) \end{aligned}$$

where $\mathbf{u}^i$ and $\mathbf{v}^i$ are the i-th column vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively.

[2] v^* denotes here the complex conjugated of v, while $v^\dagger$ would be the transposed and complex conjugated.

One application of the SVD is most relevant for this work.[3] Assume to maximise the following expression

$$c = \max_{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^m} \frac{\mathbf{y}^\dagger A\mathbf{x}}{||\mathbf{x}||_2||\mathbf{y}||_2}. \tag{2.8}$$

Representing $\mathbf{y}$ and $\mathbf{x}$ in the basis U and V respectively as

$$\mathbf{y} = \sum_{i=1}^{m} \mu_i \mathbf{u}^i, \quad \mathbf{x} = \sum_{j=1}^{n} \nu_j \mathbf{v}^j, \tag{2.9}$$

leads to

$$c = \max_{\nu\in\mathbb{R}^n,\mu\in\mathbb{R}^m} \frac{\mu U^\dagger U\Sigma V^\dagger V\nu}{||\nu||_2||\mu||_2} = \max_{\nu\in\mathbb{R}^n,\mu\in\mathbb{R}^m} \frac{\mu\Sigma\nu}{||\nu||_2||\mu||_2}. \tag{2.10}$$

From Eq.(2.4) and Eq.(2.5) it is now obvious that the correct choice to maximise c is

$$\mu_i = \delta_{1i}, \;\; \nu_i = \delta_{1i} \text{ leading to } c = \sigma_1. \tag{2.11}$$

Consequently, one obtains

$$\mathbf{y} = \mathbf{u}_1, \;\; \mathbf{x} = \mathbf{v}_1. \tag{2.12}$$

2.3. Orthogonal Projections

If $W \subset \mathbb{R}^n$ is a subspace, an orthogonal projection to W is given by $P \in \mathbb{R}^{n\times n}$ if

$$\text{Range}(P) = W, \tag{2.13}$$
$$P^2 = P, \tag{2.14}$$
$$P^\dagger = P. \tag{2.15}$$

Then one has automatically $\text{Kern}(P) = \text{Range}(\mathbb{1} - P) = W^\perp$. Starting with an orthonormal basis B for W then the projector is $P = BB^\dagger$. While P is unique for a given W, B is not.

2.4. Gram-Schmidt Orthonormalisation

Consider an arbitrary basis $B = \{v^1, \ldots, v^N\}_N$ of the vector space V. This basis can be orthonormalised as follows. First, normalise v^1:

$$v^{1'} = \frac{1}{||v^1||_2} v^1. \tag{2.16}$$

[3] Although actually it will not be necessary to calculate a SVD explicitely.

The second basis vector is calculated by subtracting the parallel component of v^1 and v^2 from v^2.

$$v^{2'} = \frac{1}{||v^2 - \langle v^1, v^2 \rangle \, v^2||_2} (v^2 - \langle v^1, v^2 \rangle \, v^2). \tag{2.17}$$

This is successively repeated for all column vectors of B. While this algorithm always works analytically, finite numerical precision can lead to very inaccurate results if some v^i, v^j are nearly parallel. A more stable variant is described in [50]. Note that the Gram-Schmidt orthonormalisation is one way to calculate a QR-decomposition as described below in Section 2.8.

2.5. The LU Decomposition

A decomposition of a $m \times n$ matrix A of the form

$$A = LU \tag{2.18}$$

with a lower triangular[4] matrix L and an upper triangular matrix U is termed LU-decomposition. The LU-decomposition exists if the first k leading $k \times k$ sub-matrices are nonsingular, with $k = \min(m, n)$. For this work the case $m = n$ is relevant. This decomposition is useful for solving the system of linear equations

$$A\mathbf{x} = \mathbf{b}. \tag{2.19}$$

This is done in two steps that exploit the structure of U and L.

$$L\mathbf{y} = \mathbf{b}, \tag{2.20}$$

$$U\mathbf{x} = \mathbf{y} \tag{2.21}$$

Eq.(2.20) is solved by so called forward elimination:

$$y_i = \frac{1}{L_{ii}} \left(b_i - \sum_{j=1}^{(i-1)} L_{ij} y_j \right) \qquad i = 1 : n \tag{2.22}$$

Eq.(2.21) is solved by back substitution:

$$x_{(n-i)} = \frac{1}{U_{(n-i)(n-i)}} \left(y_{(n-i)} - \sum_{j=n-i+1}^{n} U_{(n-i)j} x_j \right) \qquad i = 0 : n - 1 \tag{2.23}$$

The LU-decomposition can be viewed as a formal description of the Gaussian elimination. Practical implementations can be found e.g. in [96].

[4] A lower triangular matrix has vanishing matrix elements above the main diagonal. Likewise, an upper triangular matrix has no nonzero entries below the main diagonal.

2.5.1. Matrix Inversion by LU Decomposition

The inverse A^{-1} of a matrix A is defined by

$$A^{-1}A = \mathbb{1}. \tag{2.24}$$

One possibility to compute A^{-1} is to use the LU-decomposition and solve Eq.(2.19) for each colum of A^{-1}, choosing the canonical basis vectors for $\mathbf{b}$.

2.6. Determinant of a Matrix and Characteristic Polynomial

The determinant of a square $n \times n$ matrix A is a map $K^{n \times n} \to K$. It describes the volume change of a unit hypercube under A. A formal definition can be found in the literature [45]. Most useful for the purposes in this work is its following property

$$\det(A) = 0 \quad \Leftrightarrow \quad A \text{ is singular.} \tag{2.25}$$

The characteristic polynomial $\chi(x)$ of a square $n \times n$ matrix A is defined by

$$\chi(x) := \det(A - x\mathbb{1}). \tag{2.26}$$

It is a polynomial of degree N, its zeros are the eigenvalues of A which are considered in the following section.

2.7. Inverse and Pseudo-inverse of a Matrix

A diagonal matrix can be inverted most easily. This is performed by

$$D^{-1} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{pmatrix}^{-1} = \begin{pmatrix} \lambda_1^{-1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n^{-1} \end{pmatrix} \tag{2.27}$$

This is clearly only possible if $\det(D) = \prod_i \lambda_i \neq 0$. For singular matrices D_s a unique pseudo-inverse D_s^+ can be calculated by

$$D_s^+ := \begin{pmatrix} \hat{\lambda}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \hat{\lambda}_n \end{pmatrix} \tag{2.28}$$

with $\hat{\lambda}_i := \begin{cases} \frac{1}{\lambda_i} & \lambda_i \neq 0 \\ 0 & \text{else} \end{cases}$. This pseudo-inverse satisfies the Moore-Penrose conditions which are itself sufficient to determine D_s^+.

$$D_s D_s^+ D_s = D_s \tag{2.29}$$

$$(D_s D_s^+)^\dagger = D_s D_s^+ \tag{2.30}$$

$$D_s^+ D_s D_s^+ = D_s^+ \tag{2.31}$$

$$(D_s^+ D_s)^\dagger = D_s^+ D_s \tag{2.32}$$

2.8. The QR decomposition

For a $m \times n$ matrix A $(m > n)$ the QR-decomposition is given by

$$A = QR, \tag{2.33}$$

with an orthonormal $m \times m$ matrix Q and upper triangular $m \times n$ matrix R. If A has full rank and one requires the diagonal elements of R to be positive, the QR-decomposition is also unique. Several methods can be used to calculate the QR-decomposition. The Householder approach [50] requires that A has full rank which is not true for the applications in this work. Another possibility are Givens-rotations. A rotation $\tilde{Q}$ affects in general only a two-dimensional subspace. Here this is chosen to be a space spanned by two basis vectors. Then only one row and column of A, e.g. row i and column j, are affected. The rotation angle can be chosen so that $(QA)_{ij} = 0$. Repeating this for the lower triangular part of $R := \tilde{Q}A$ leads to the desired form. As all $\tilde{Q}^k$ are rotations and thus orthonormal also the product $Q = \left(\prod_k \tilde{Q}^k\right)^\dagger$ is.

2.9. The Schur decomposition

For a general real matrix it can be shown that the Schur decomposition exists [50]. For $A \in \mathbb{R}^n \times \mathbb{R}^n$ there exists an orthogonal $Q \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$Q^\dagger A Q = \begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ 0 & R_{22} & \cdots & R_{2m} \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & R_{mm} \end{pmatrix} \tag{2.34}$$

with R_{ii} either 1×1 real matrices or 2×2 real matrices with complex conjugated eigenvalues. The number of blocks is m where $m \leq n$ holds. Further, the ordering of the R_{ii} can be arbitrary.

This decomposition always provides a real orthonormal decomposition basis with the property that the first M Schur vectors span A-invariant subspaces, provided M is chosen so that no 2×2 blocks are broken up. Note, that the spaces spanned by the Schur vectors are not mutually invariant.

2.9.1. Ordering of the Schur decomposition

It has been stated above, that the R_{ii} can be ordered arbitrarily. One possible algorithm for such an ordering has been given by Brandts [20]. There, the ordering is obtained by successive swapping of neighbouring blocks R_{ii}. A description of the version used by us is given in Appendix C.

3. Partial Differential Equations

In this chapter I will give a short overview on the topic of partial differential equations. Some basic terminology and background will be given. With regard to the intention of this thesis I will exemplify some discretisation approaches and numerical solution techniques. Some references for partial and ordinary differential equations are [48, 36, 115]. For numerical integration, see [65, 26, 74], also [96] might be helpful. The finite element method is detailed presented in [32], [73] provides some numerical issues. The spectral method is presented in [52, 114], the latter providing a very simple and easy accessible introduction.

Differential equations (DEs) are, as indicated by the name, equations that involve differential operators. The highest order of derivative that occurs in the equation determine the *order of a DE*. In the simplest case only a single independent variable occurs. This results in so called ordinary differential equations (ODEs). For ODEs some rigorous mathematical results exist, e.g. the theorem of Peano or the theorem of Picard and Lindelöf [59]. The first states the existence while the later guarantees existence and uniqueness for the solution of the initial value problem

$$\frac{d}{dt}y = f(y,t), \qquad y(t_0) = y_0 \tag{3.1}$$

under certain preconditions. The theorem of Peano guarantees existence of a solution if $f(y,t)$ is continuous. If $f(y,t)$ is further Lipschitz-continuous, i.e. there exists a constant L so that $|f(y,t) - f(y',t)| \leq L|y - y'|$, existence and uniqueness are guaranteed by the Picard Lindelöf theorem. In Eq.(3.1) it is already indicated that the ODE is not sufficient to determine a solution. Here initial conditions have to be provided additionally, i.e. the value of y_0. Beside from ODEs being simple examples of differential equations they typically result also from the discretisation methods and are thus relevant for the following work.

The focus is set on differential equations that contain partial derivatives, i.e. partial differential equations (PDEs). To be distinguished non-trivially from ODEs at least two independent variables, say x and t have to occur. A simple linear example would be

$$\alpha\frac{\partial}{\partial t}\phi + \beta\frac{\partial}{\partial x}\phi = f(x,t), \qquad (x,t) \in \mathcal{V} \tag{3.2}$$

The equation describes the behaviour of a field variable $\phi(x,t)$. The solution of Eq.(3.2) describes the field for all points $(x,t) \in \mathcal{V}$ for some space-time region $\mathcal{V}$. Analogous to the initial conditions for ODEs, also for PDEs additional information is necessary to determine a solution. Here initial conditions typically comprise a function instead of a single value. Also boundary conditions have to be provided. What data are needed to specify a particular solution is a nontrivial problem that depends on the problem at hand. Prescribing initial conditions in form of a function $\phi(x, t_0) = \Phi_0(x)$ and boundary conditions for ϕ on the boundary of $\mathcal{V}$, i.e. $\phi(x, t) = g(x, t)\ \forall t, x \in \partial\mathcal{V}$, is usually necessary and sufficient to specify a solution. More insight in this problem provides the section on the method of characteristics, Section. 3.2. However, for nonlinear PDEs no general proof of existence of solutions exist. Also the methods to obtain a solution depend much stronger on the equation at hand, as for ODEs. In the following one of many usual classifications for linear PDEs is reproduced.

3.1. Classifications

For second order linear PDEs, i.e. PDEs of the form

$$\mathrm{A}\frac{\partial^2}{\partial x^2}\phi + 2\mathrm{B}\frac{\partial}{\partial t}\frac{\partial}{\partial x}\phi + \mathrm{C}\frac{\partial^2}{\partial t^2}\phi + \mathrm{D}\frac{\partial}{\partial x}\phi + \mathrm{E}\frac{\partial}{\partial t}\phi + \mathrm{F}\phi + \mathrm{G} = 0 \tag{3.3}$$

the classification into **hyperbolic, parabolic and elliptic PDEs** is common. The classification is determined by the determinant of the matrix $\mathbf{M}$

$$\mathbf{M} = \begin{pmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B} & \mathrm{C} \end{pmatrix}. \tag{3.4}$$

In case of a positive definite $\mathbf{M}$, i.e. $\det(\mathbf{M}) = \mathrm{AC} - \mathrm{BB} > 0$, the PDE is called elliptic, for a negative definite $\mathbf{M}$, i.e. $\det(\mathbf{M}) < 0$, hyperbolic. For a parabolic PDE $\det(\mathbf{M}) = 0$.[1] A pictorial interpretation of one difference between these classes is as follows. The definiteness of the matrix $\mathbf{M}$, i.e. the classification, determines the way how disturbances ('information') are propagated by the equations. This is relevant e.g. for the method of characteristics, see Section 3.2. For nonlinear PDEs such a classification is usually not possible. The matrix $\mathbf{M}$ will then typically depend on $\phi(x, t)$ and thus, on the type of the equation. Consequently, the system may act e.g. in some regions of the phase space as hyperbolic but in others as an elliptic system. As a physical example consider the Navier-Stokes equations

$$\frac{\partial \mathbf{v}}{\partial t} = \nu\nabla^2\mathbf{v} - (\mathbf{v}\nabla)\mathbf{v} - \nabla p, \tag{3.5}$$

$$\nabla\mathbf{v} = 0. \tag{3.6}$$

Where $\mathbf{v}$ is the velocity and p the pressure. Here the fluid is assumed to be a Newtonian fluid, i.e. the shear stress is proportional to the shear velocity. Often additional thermodynamic quantities have to be coupled to the Navier-Stokes

[1] This notation is in analogy to the definition of a hyperbola, parabola and ellipse from a general quadratic algebraic equation. The derivatives in Eq.(3.3) just have to be replaced by corresponding powers of that variable.

equations. Examples of this type will be considered in Chapter 10. The Navier-Stokes equations are elliptic for subsonic flow. For supersonic flow they become hyperbolic. Since (macroscopically[2]) the flow velocity at a boundary has to vanish, this generically leads to problems where elliptic and hyperbolic behaviour occurs in different regions for the same problem. So such classifications are naturally only meaningful in a local sense, i.e. for a linearisation in a given point.

3.2. Method of characteristics

The basic idea of the method of characteristics is to determine so called characteristic lines which have the property that along these lines the PDE reduces to an ODE. For a first order PDE e.g. the characteristic lines can be obtained by comparing

$$\alpha \frac{\partial}{\partial t}\phi + \beta \frac{\partial}{\partial x}\phi = F(\phi, x, t) \tag{3.7}$$

and

$$\frac{d}{ds}\phi(x(s), t(s)) = \frac{dt}{ds}\frac{\partial}{\partial t}\phi + \frac{dx}{ds}\frac{\partial}{\partial x}\phi = F(\phi(x(s), t(s)), x(s), t(s)). \tag{3.8}$$

The characteristic line is determined by $\frac{dt}{ds}$ and $\frac{dx}{ds}$. As mentioned above the treatment of the ODE is much simpler and for integration only one initial value is required. Due to this fact the characteristic lines also indicate the 'flow of information'. This indicates also what boundary conditions are necessary to determine a solution of the PDE at hand for a particular region $\mathcal{V}$ in $x \times t$-space. Generalising the requirement for ODEs, each point (x, t) should lie on a characteristic line for which an initial condition is given. It should be noted that in general not all points in $\mathcal{V}$ will be reached by characteristic lines. Then still weak solutions (described later in Section 3.5.3) may exist. Consideration of the information flow is also crucial for stable integration schemes.

In the following this will be exemplified with the classification of the previous section. Hyperbolic PDEs have real characteristics. The wave equation e.g. has straight lines as characteristics. A single point is therefore influenced only by the points on a cone (and within, since these points influence the points on the cone) in the past (or equivalently in the future). Physically, this reflects the finite propagation speed of disturbances. If one for example considers a rectangular space time region $\mathcal{V} = [x_a, x_e] \times [t_0, t_1]$, it is necessary to prescribe $\phi(x, t_0)$, i.e. the initial conditions, and the boundary conditions $\phi(x_a, t), \phi(x_b, t)$.

Parabolic PDEs have one real characteristic. For the typical example, the dif-

[2] For systems, where the Knudsen number, describing the ratio of mean free path length and characteristic length scale of the flow problem is sufficiently high this has to be modified.

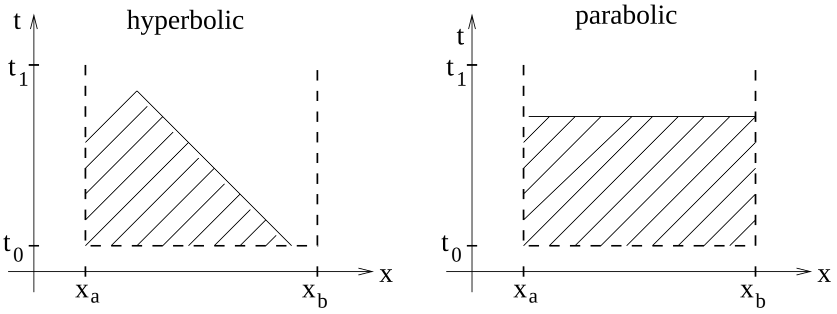


Figure 3.1.: Characteristic lines for hyperbolic and parabolic PDEs.

fusion equation[3](see also Section 9.1),

$$\frac{\partial}{\partial t}\Phi(x) = d\Delta\Phi(x) \quad x \in [0,1], \tag{3.9}$$

the characteristics are just constants, i.e. the limit of the previous case with the slope of the characteristics going to zero (infinite propagation speed). One point can therefore influence the whole system at all later times. This corresponds to the hyperbolic case in the limit of infinite propagation speed. Consequently, the initial- and boundary conditions have to be provided for the past of the whole system in order to specify a solution. This can also be seen in Fig. 3.1.

For elliptic PDEs the characteristics are complex and do therefore not have a straightforward meaning as in the previous cases. A single point can influence all other points. Typically such equations arise in steady state problems. To allow for a unique solution, the boundary conditions have to be specified on a closed boundary.

3.3. Linearity

Differential operators are linear which follows trivially from the derivation rules for sums and constant multiplicative factor:

$$\frac{d^n}{dt^n}\left(\mu f + \nu g\right) = \mu\frac{d^n}{dt^n}f + \nu\frac{d^n}{dt^n}g, \ \forall n \in \mathbb{N} \tag{3.10}$$

Linear equations will occur in the following mainly as motivation and simple example where already methods or mathematical proofs exist or additional facts, as the optimal reduction, are known. In the special linear case the considered PDEs are typically of the form

$$\frac{\partial}{\partial t}\phi(\mathbf{x},t) = G\phi(\mathbf{x},t) \tag{3.11}$$

[3] The diffusion equation describes diffusive transport of a scalar field Φ, e.g. heat transport, in a medium. For homogeneous media it is given by Eq.(3.9) with the diffusion constant d.

where G is a linear operator. The operator G is called the generator of the time evolution. Interpreting one dimension as time most PDE can be brought to this form. However, for equations with higher order time derivatives, dependent variables have to be introduced. Further, it will be assumed that G is independent of t, as this will be always the case later on. Then Eq.(3.11) can be formally integrated giving for the time evolution operator

$$U(t) := e^{tG}. \tag{3.12}$$

This scenario can be generalised by allowing G to be dependent on ϕ. The resulting equation is a nonlinear PDE and is given by

$$\frac{\partial}{\partial t}\phi(\mathbf{x},t) = G(\phi(\mathbf{x},t))\phi(\mathbf{x},t). \tag{3.13}$$

The linearisation for a given ϕ_{fix} is the linear operator $G(\phi_{\text{fix}})$.

3.4. Well posed problems

In mathematics the PDE together with initial and boundary conditions is referred to as a well posed problem if a unique solution exists that depends continuously on the initial and boundary conditions [48]. For nonlinear PDEs it is however not always possible to prove the existence of such a solution. In many cases it is further known that the system can become chaotic. The famous Lorenz model, a standard example for deterministic chaos e.g. was derived from the three-mode spectral Galerkin approximation of the Bernard-Reighley flow [80]. The details will be described later. For general numerical treatment, one has to rely on discretised models. Therefore it is necessary to go one step further and discretise the PDE of interest. In order for the resulting description to have any meaning in giving a predictive simulation of the systems dynamical behaviour, one has to assume the problem of interest to be well posed.

Nevertheless, the points mentioned above should also hold for numerical algorithms. For linear systems such conditions are even necessary and sufficient for convergence of the algorithm due to the Lax theorem [98]. In the nonlinear case this is not that simple but in the following uniqueness and continuous dependence on initial and boundary conditions are still presumed for the algorithm. However, one should still keep in mind that such requirements may be violated in physical solutions.

3.5. Numerical Treatment

3.5.1. Integration of ODEs

The focus of this thesis is the presentation of new numerical methods for model reduction of nonlinear dynamical systems. Now a short introduction to numerical techniques will be given, starting with the solution of ODEs. Only first order ODEs will be considered. Higher order ODEs can be brought to adequate form by introducing new variables. Again explicit time dependence is excluded and the following generic ODE is considered

$$\frac{d}{dt}y = f(y), \qquad y(t_s) = y_0. \tag{3.14}$$

To solve Eq.(3.14) one would integrate Eq.(3.14) formally as

$$y(t) = \int_{t_s}^{t} f(y(t))dt + y_0. \tag{3.15}$$

A simple method to calculate a numerical approximation to Eq.(3.15) is to discretise the time interval of interest $[t_s, t_e]$ with equidistant points t_i, $i = 1 : N$ of distance Δt. Thus $t_s = t_1$, $t_i = t_s + (i-1)\Delta t$ and $t_e = t_s + (N-1)\Delta t$.

If one approximates $y(t)$ by a piece-wise constant function $\hat{y}(t)$, one can evaluate

$$\hat{y}_t = y_0 + \delta t \sum_{i=1}^{t-1} f(\hat{y})_i, \tag{3.16}$$

see Fig. 3.2. Now the value of $\hat{y}(t_i)$ is completely determined by the previous time step, giving the following explicit equation

$$\hat{y}(t_i) = \hat{y}(t_{i-1}) + \Delta t f(\hat{y}(t_{i-1})). \tag{3.17}$$

This method is called Euler forward method or explicit Euler method. The same integration scheme can be obtained if one expands $y(t_i)$ in a Taylor series

$$y(t_i + \Delta t) = \sum_{i=0}^{m} \frac{(\Delta t)^i}{i!} \left.\frac{dy}{dt}\right|_{y(t_i)} + \mathcal{O}((\Delta t)^{m+1}). \tag{3.18}$$

If only terms up to first order in Δt are considered Eq.(3.17) is reproduced. It can further be obtained by approximating $\frac{dy}{dt}$ via finite differences. Finite differencing makes use of the difference quotients that give in the limit of arbitrary fine discretisation the derivative. One could write

$$\frac{y(t_i + 1) - y(t_i)}{\Delta t} = f(y(t_i)), \tag{3.19}$$

which is again equivalent to Eq.(3.17).

To calculate the numerical solution, the summing of Eq.(3.17) is simply repeated iteratively starting by $y(t_s) = y_0$. This is simple, easy to implement and fast, but not very accurate. This method will be used for some of the toy models. Due to its limitations it is however not well suited for practical use. In particular it is not stable for so called stiff problems. A stability condition for the explicit Euler method is

$$|1 + \lambda \Delta t| \leq 1, \tag{3.20}$$

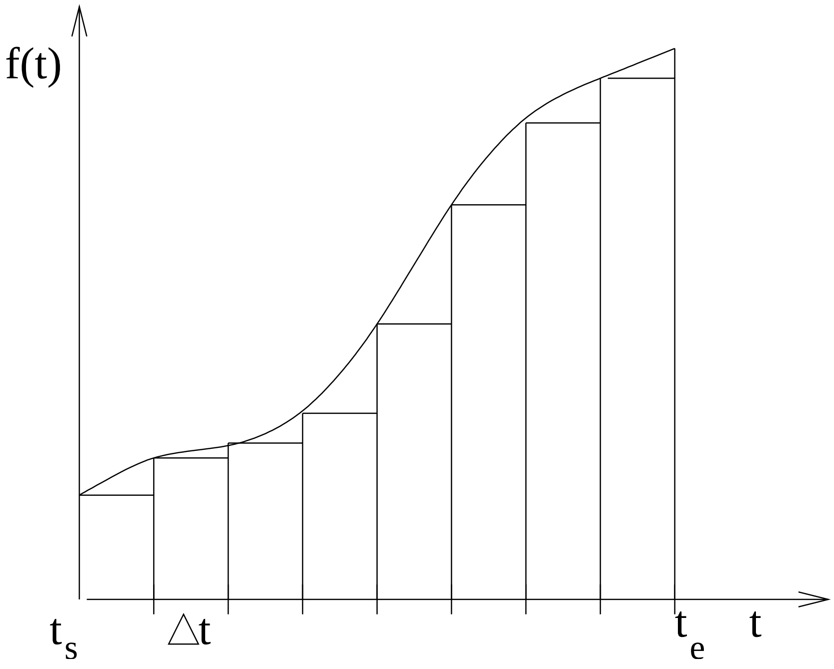


Figure 3.2.: Lower sum approximation of a function f(t)

where λ is the largest absolute eigenvalue of the generator of evolution [50]. This restriction will become relevant later throughout this work. Although convergence can be achieved by reducing the time step Δt, this can require an unacceptable small Δt. This is the case for $|\lambda| \gg 1$, the ODE is then termed stiff. The solution is changing on small time scales, nevertheless these are often changes along directions in phase space which can be considered irrelevant, see the next chapter. Now I will present some alternatives to circumvent this problem.

Note, that Eq.(3.19) is not symmetric. Likewise, one could have used the form

$$\frac{y(t_i) - y(t_{i-1})}{\Delta t} = f(y(t_i)), \tag{3.21}$$

the limit $\Delta t \to 0$ is the same. Now the value of $y(t_i)$ is *implicitly* given by

$$y(t_i) - \Delta t f(y(t_i)) = y(t_{i-1}). \tag{3.22}$$

The integration is now done by solving Eq.(3.22) for each time step. This implicit Euler method is unconditionally stable, i.e. for all λ. The time integration implies now the solution of a linear system of equations, while the explicit Euler method requires only a matrix vector product which is much faster to compute.

The error of the method presented above is of order $(\Delta t)^2$. This is only one order of Δt smaller than the integration terms. Although useful for pedagogical purposes the methods presented above are not well suited for practical applications. To increase the accuracy several approaches exist. One possibility is to calculate additional terms to fit the Taylor expansion Eq.(3.18) to a higher order in Δt in a similar manner as above. This ansatz is pursued by Runge-Kutta schemes [96] which enjoy a certain popularity. The simplest form, the explicit mid-point rule, calculates first a trial step $y' = y(t_i) + \Delta t f(y(t_i))$ via the explicit Euler method. From the mid-point $\frac{y(t_i)+y'}{2}$ an additional explicit Euler step is performed which yields

$$y(t_{i+1}) = y(t_i) + \Delta t f(y') \tag{3.23}$$

for the whole time step. The error is of order $(\Delta t)^3$ and higher order schemes can be devised in a similar way. Most often used is the explicit fourth-order Runge-Kutta method with error of order $(\Delta t)^5$.

An alternative way to increase the accuracy is to use a higher order approximation for the discretised time derivative, i.e. the left hand side of Eq.(3.19) or Eq.(3.21). This leads to the so called multi-step methods. Such methods are termed Adams methods after John Couch Adams(1819-1892) and can be explicit (Adams-Bashforth) or implicit (Adams-Moulton). In Chapter 10 the incompressible 2D Navier-Stokes equations will be considered. There, the general variational POD algorithm is presented. For the 2D Navier-Stokes equations the friction term leads to a stiff ODE, while the interesting physics is due to the nonlinear part of the generator of evolution. The latter can be treated sufficiently by explicit methods for the problem at hand. Therefore so called operator splitting methods are a good choice. In particular an ansatz proposed by Karniadakis et al. will be used. More exactly, it is a version of the third order scheme, given in [67]. Here the term 'order' indicates the number of previous time-steps used to calculate the actual configuration. Its advantage is to calculate the contribution from the nonlinear part of the generator of evolution explicitely. In our example, the Navier-Stokes equations, this is the Jacobi operator or Jacobian J, in Cartesian coordinates

$$\mathcal{J}(\phi, \psi) := \frac{\partial \phi}{\partial x}\frac{\partial \psi}{\partial y} - \frac{\partial \phi}{\partial y}\frac{\partial \psi}{\partial x}. \tag{3.24}$$

This operator is relevant for the advection term in the Navier-Stokes equations and contains first derivatives of the field. However, the linear part of the generator of evolution, i.e. the Laplace operator Δ for the Navier-Stokes equations is treated implicitly by the scheme. The scheme is reproduced briefly for the first three orders.

The derivative is approximated by

$$\frac{1}{\Delta t}\sum_{i=0}^{J-1} \alpha_i \left(y(t_{n+1}) - y(t_{n-i})\right) = \frac{1}{\Delta t}\sum_{i=0}^{J-1} \alpha_i \int_{t_{n-i}}^{t_{n+1}} \frac{\partial y}{\partial t} dt \tag{3.25}$$

$$= \frac{1}{\Delta t}\sum_{i=0}^{J-1} \alpha_i \int_{t_{n-i}}^{t_{n+1}} f(y) dt. \tag{3.26}$$

Here the left hand side is the approximation of the time derivative. The coefficients are given in Tab. 3.5.1. The right hand side is transformed by inserting already the

Coefficient	1st Order	2nd Order	3rd Order
γ_0	1	3/2	11/6
α_0	1	2	3
α_1	0	-1/2	-3/2
α_2	0	0	1/3
β_0	1	2	3
β_1	0	-1	-3
β_2	0	0	1

Table 3.1.: Stiffly stable coefficients from [67].

ODE of interest, see Eq.(3.14). The last integral can now be split as convenient, in present case into the linear part $f_1(\cdot)$ and the nonlinear part $f_2(\cdot)$ of $f(\cdot) = f_1(\cdot) + f_2(\cdot)$. In practice, for a scheme of order J first the explicit part is evaluated by

$$\hat{k} = \sum_{q=0}^{J-1} \alpha_q y(t_{n-q}) + \Delta t \sum_{q=0}^{J-1} \beta_q f_2(y(t_{n-q})). \tag{3.27}$$

For the complete time step one obtain including the implicit part

$$y(t_{n+1}) = \frac{1}{\gamma_0}\left(\hat{k} + \Delta t f_1(y(t_{n+1}))\right), \tag{3.28}$$

which gives by definition of $f_1(\cdot)$ a system of linear equations. In Chapter 10 this is actually solved by the LU-decomposition.

The coefficients α_i, β_i are given in Tab. 3.5.1. The coefficient γ_0 is simply $\gamma_0 = \sum_{i=0}^{J-1} \alpha_i$.

3.5.2. Finite differencing for PDEs

We have already encountered the idea of finite differencing in the previous subsection. Now we come to the basic ideas for performing the spatial discretisation for a PDE in the x, t domain. To this end, for the space coordinate a grid with nodes x_i is constructed. The spatial discretisation step Δx is constant in the following for clarity, although this is no necessity. The two ways to approximate the first derivative shown in Eq.(3.19) and Eq.(3.21) are also possible here and are commonly known as *upwind* and *downwind* derivatives. This nomenclature is motivated from the use in problems where a preferred direction exist [73]. For a particular space point they are given by

$$D_{x,d}\big|_{x_i} \phi = \frac{\phi(x_i) - \phi(x_{i-1})}{\Delta x} \quad , \quad D_{x,u}\big|_{x_i} \phi = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x}. \tag{3.29}$$

If a Taylor series expansions around x_i for the next neighbouring nodes is performed, one obtains

$$\phi(x_{i+1}) = \phi(x_i) + \frac{\partial\phi(x_i)}{\partial x}\Delta x + \frac{\partial^2\phi(x_i)}{\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3) \tag{3.30}$$

$$\phi(x_{i-1}) = \phi(x_i) - \frac{\partial\phi(x_i)}{\partial x}\Delta x + \frac{\partial^2\phi(x_i)}{\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3) \tag{3.31}$$

From this it is clear that the two discretisation schemes, the downwind and upwind derivative, are accurate only to first order in Δx. However, if Eq.(3.31) is subtracted from Eq.(3.30) and this divided by two one gets a second order accurate approximation of the first derivative, i.e. the centred differencing scheme

$$D_{x,s}\big|_{x_i}\,\phi = \frac{\phi(x_{i+1}) - \phi(x_{i-1})}{2\Delta x}. \tag{3.32}$$

To approximate the second derivative Eq.(3.31) and Eq.(3.30) are added. After a rearrangement one obtains

$$D_x^2\big|_{x_i}\,\phi = \frac{\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1})}{\Delta x^2}. \tag{3.33}$$

From Eq.(3.30) and Eq.(3.31) it is clear that this approximation is again of second order accuracy in Δx.

Boundary conditions

If the space domain X has a boundary, e.g. with node x_1 the derivative will depend on the boundary conditions. Some types of boundary conditions can be included into the derivation operator.

One simple possibility is to assume the topology of a ring in this space direction which corresponds to *periodic boundary conditions*. This describes an infinite but periodic system. Practically this is implemented by simply identifying the fictive node x_0 with the existing node x_N if N is the dimensionality from discretising X. If the value at the boundary is prescribed, this is called a *Dirichlet boundary condition.*The boundary node will then not be included in the grid, since the calculation of the corresponding values is trivial. For homogeneous Dirichlet boundary conditions, i.e $\phi(x_0) = 0$ the derivatives are calculated as without boundary, the fictive node x_0 for which the value is prescribed is not on the grid, so contributions are absent. For the inhomogeneous case additional terms are necessary, which cannot be included in the derivation operator.

Likewise, the derivative on the boundary can be prescribed. Setting it to zero results in homogeneous Neumann boundary conditions. The first derivative can still be calculated for all nodes by using the upwind or downwind scheme at the boundaries. To implement homogeneous Neumann boundary conditions for the second derivative one can start from the definition of the upwind or downwind derivative, Eq.(3.29). This requires for the fictive node $-\phi(x_0) = \phi(x_1)$ to be valid. The second derivative for x_1 reduces then to $\frac{-\phi(x_i)+\phi(x_{i+1})}{\Delta x^2}$.

Matrix notation

The discretisation above leads to a finite vector $\phi(\mathbf{x}_i)$ which describes the field at each node of the grid. The grid does not have to be one-dimensional. From the previous section it is known how to calculate derivatives for certain grid nodes. Obtaining the derivative for a field vector ϕ corresponds in the discrete case as well as in the original problem to the application of a linear operator. Thus it can be described by Matrix multiplication for finite systems. The form of these matrices is now presented for one-dimensional grids. One benefit is the possibility of using standard tools from linear algebra to manipulate and solve the resulting equations. Assuming an equally spaced grid the downwind, upwind and centred derivative described above translate to

$$D_d = \begin{pmatrix} \ddots & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & & \ddots \end{pmatrix}, \tag{3.34}$$

$$D_u = \begin{pmatrix} \ddots & & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & -1 & 1 \\ & & & & \ddots \end{pmatrix}, \tag{3.35}$$

$$D_s = \begin{pmatrix} \ddots & & & & \\ -1 & 0 & 1 & & \\ & -1 & 0 & 1 & \\ & & -1 & 0 & 1 \\ & & & & \ddots \end{pmatrix}. \tag{3.36}$$

For the second order derivative - the one-dimensional Laplace operator - one obtains

$$D_d^2 = \begin{pmatrix} \ddots & & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & & \ddots \end{pmatrix}. \tag{3.37}$$

These matrices have a simple tridiagonal structure, i.e. only the main diagonal and the next two diagonals are non zero (for $D_{d,u,s}$ even only two diagonals are

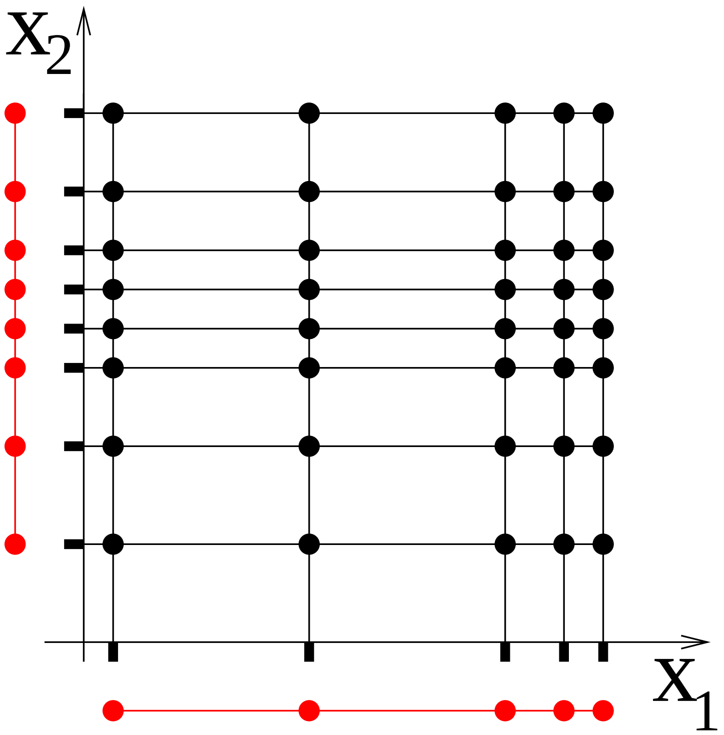


Figure 3.3.: Two-dimensional grid as tensor product from two one-dimensional grids.

nonzero). [4] The structure can be exploited to improve the efficiency of numerical algorithms. Generalisation to unequal grid spacing is possible but not used in this work. For complex geometries the finite element methods below are much more flexible.

For generalisations to higher-dimensional finite differencing methods tensor product spaces are considered. In two dimensions e.g. they are generated by grids with N_1 and N_2 nodes as shown in Fig. 3.3. Practically for each coordinate x_1 all N_2 possible combinations (x_1, x_2) exist. If one chooses as ordering of the nodes

$$k := i + N_1(j-1) \quad , \quad i = 1:N_1, \;\; j = 1:N_2, \tag{3.38}$$

the derivation operators for the two-dimensional system can be constructed by the Kronecker product, e.g.

$$D_x^{2D} = \mathrm{kron}(D^{1D}, \mathbb{1}_{N_2}), \quad D_y^{2D} = \mathrm{kron}(\mathbb{1}_{N_1}, D^{1D}). \tag{3.39}$$

The Kronecker product of a $m \times n$ matrix A and a $\alpha \times \beta$ matrix B is the $m\alpha \times n\beta$ matrix defined by

$$\mathrm{kron}(A, B)_{i+m(\nu-1),j+n(\mu-1)} := A_{ij}B_{\nu\mu} \quad i = 1:m, j = 1:n, \nu = 1:\alpha, \mu = 1:\beta. \tag{3.40}$$

[4] Boundary conditions may add additional off-diagonal contributions.

The Laplace operator is always represented by a single matrix. This matrix is symmetric so that always an ONB of eigenvectors and a real spectrum exists. The centred first order operator is antisymmetric and has a purely imaginary spectrum.

Boundary conditions and Matrix notation

Until now the matrices were only defined for regions that do not belong directly to the boundary of the domain where the PDE (or ODE) has to be solved. For the existence of a well defined solution often boundary conditions have to be specified.[5] This is reflected, for the discrete version, by the fact that without the boundary terms the number of equations is smaller than the number of lattice sites. For this work a solution of an equation of the form

$$A(\phi)\phi = f(\phi), \tag{3.41}$$

is of interest. Here A is a rectangular matrix, if no boundary conditions are specified. The solution is then not unique. This is resolved by appropriate boundary conditions. For example, in one dimension and n lattice sites there are only $n-1$ up- or downwind derivatives possible. To get n equations, one boundary condition has to be specified.[6] This is done by adding additional lines to A until A is a square matrix. For the Laplacian, two values cannot be calculated according to scheme Eq.(3.37). The corresponding lines are used to express two boundary condition. Of course this is in agreement with standard calculus for differential equations, which requires the same numbers of boundary conditions to fully specify a solution.

If the value of e.g the function ϕ or its derivative etc. is prescribed to be zero at the boundaries this is termed a homogeneous boundary condition. The generalisation to some value $x_{\text{BC}} \neq 0$ is made by adding a contribution to the right side of Eq.(3.41), which is the inhomogeneity.

1. **Dirichlet conditions:** Here the function value is prescribed on the boundary. The case with zero valued function on the boundary is called homogeneous. In the finite difference scheme this is expressed by inserting a canonical basis vector, corresponding to the given boundary site, as additional line in the matrix. Alternatively, one can exclude the boundary site from the sites to be calculated (since it is already known) and then set the corresponding link(s) to this site to zero without changing the diagonal entry. In the latter case, the dimensionality of the system is decreased without loss of information.

[5] Although existence of a solution will not always be guaranteed thereby.

[6] Contrary to the term boundary condition this specification has not to be at the boundary, also more general constraints are possible.

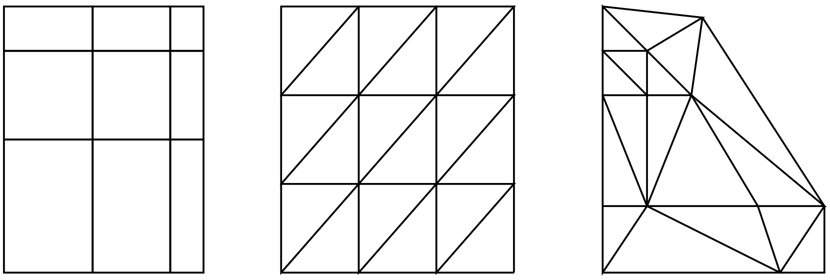


Figure 3.4.: Two-dimensional finite element grids. From left to right a rectangular and a triangular regular grid and a irregular triangular grid.

Eq.(3.42) shows the homogeneous Dirichlet conditions for a left boundary for the upwind differencing and the Laplacian

$$D_d = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & 1 & \\ & & & \ddots \end{pmatrix}, \quad D_d^2 = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & & & \ddots \end{pmatrix}. \tag{3.42}$$

If the left boundary is fixed to $a \neq 0$, i.e. inhomogeneous Dirichlet conditions, one has to add a corresponding inhomogeneous term, see Eq.(3.43), to the homogeneous expression.

$$D_{d,inhom} = D_d + \begin{pmatrix} -a \\ 0 \\ \vdots \end{pmatrix} \quad , \quad D^2_{d,inhom} = D_d^2 + \begin{pmatrix} a \\ 0 \\ \vdots \end{pmatrix} \tag{3.43}$$

2. **Neumann conditions:** This signifies boundary conditions in which the derivative of the function is prescribed at the boundary. The condition can be incorporated by adding a finite difference term for the appropriate site as line to the matrix. Inhomogeneous boundary conditions are achieved in the same way as in the previous case.

Within this framework **periodic boundary conditions** can also easily be implemented. To connect the system boundaries at minimal $i = 1$ and maximal $i = N$ value of a space index, one simply has to add the contribution A_{1N} and A_{N1} according to the desired differencing scheme.

3.5.3. Finite Element Methods for PDEs

Finite element methods have been developed in mathematics as well as engineering sciences since the 1970' years. They provide a very general description for PDEs and can also be implemented very flexibly for numerical solutions of complex problems. Basically, the framework allows to obtain a finite system of ODEs from

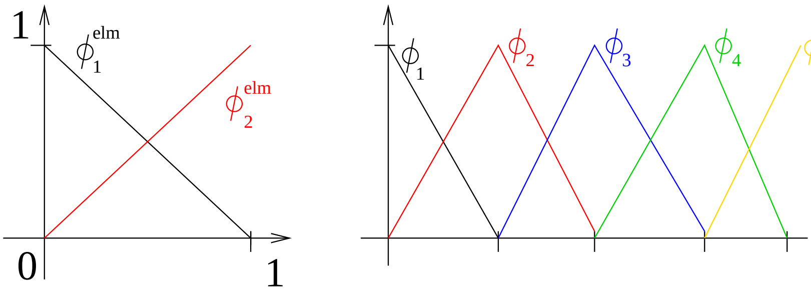


Figure 3.5.: One-dimensional finite element with linear basis function. Right the assembled ansatz-functions for a grid with 5 elements.

a given PDE problem. Starting with a general description of the basic idea also some selected details relevant for this work are introduced.

Consider a PDE of the form

$$\frac{\partial}{\partial t}\Phi(\mathbf{x},t) = F(\Phi(\mathbf{x},t))\Phi(\mathbf{x},t)\ , \quad \mathbf{x} \in \mathcal{V}, \tag{3.44}$$

$$\begin{array}{lrcll} \text{with} & \Phi(\mathbf{x},t=0) & = & \Phi_0(\mathbf{x}), & \\ & \Phi(\mathbf{x},t) & = & g(\mathbf{x}) & \text{on } \partial\mathcal{V}_D, \\ & (\hat{\mathbf{n}}\nabla)\Phi(\mathbf{x},t) & = & f(\mathbf{x}) & \text{on } \partial\mathcal{V}_N. \end{array} \tag{3.45}$$

Here $\mathcal{V}$ is a domain in space, $\partial\mathcal{V} = \partial\mathcal{V}_N + \partial\mathcal{V}_D$ is the boundary, on $\partial\mathcal{V}_D$ Dirichlet conditions are applied while $\partial\mathcal{V}_N$ is subjected to Neumann boundary conditions. The normal vector to $\partial\mathcal{V}_N$ is denoted with $\hat{\mathbf{n}}$. For the following the so called weak solution of Eq.(3.44) is now of interested. The weak solution of Eq.(3.44) is a solution for

$$\left\langle f_\tau, \frac{\partial}{\partial t}\Phi(\mathbf{x},t) - F(\Phi(\mathbf{x},t))\Phi(\mathbf{x},t)\right\rangle = 0 \quad \forall \text{ test functions } f_\tau, \tag{3.46}$$

where the scalar product is the scalar product of the vector space in which Φ lies, e.g.L^2. Typical choices of test functions are from the space of all infinitely often continously differentiable functions with compact support or Schwartz spaces. A whole mathematical theory on this topic exists [16]. For most physical problems Eq.(3.44) and Eq.(3.46) are equivalent with these choices. To obtain a discretisation now a finite set of test functions is chosen.

For discrete calculations the solution is also restricted to a finite (say N) dimensional subspace, spanned by the ansatz-functions ϕ_i, $i = 1 \dots N$. The ansatz for the solution is then

$$\hat{\Phi}(\mathbf{x},t) := \sum_{i=1}^{N} \alpha_i(t)\phi_i(\mathbf{x}). \tag{3.47}$$

For classical finite elements these are typical piecewise linear or low order polynomial functions. An important feature is that the ansatz-functions are defined for so called elements, smaller regions of the physical space with simple geometry. The whole physical space is segmented into such elements by a grid. The complete set of ansatz-functions is give by the ansatz-functions of all elements. For a single element the ansatz-functions are typically orthonormalised. Since the elements do not overlap the whole set is then orthonormal. Further, the element-wise ansatz-functions are chosen such that for convenience the coefficient $\alpha_i(t)$ is identical to the field $\hat{\Phi}(\mathbf{x_n^i}, t)$ for some particular point $\mathbf{x}_n^i$, typically a grid node. The flexibility can be further increased by allowing simple linear transformations on the element geometry. Also a combination of different elements is possible. Fig. 3.4 shows some possible finite element grids. For one-dimensional finite elements the element functions are defined on a finite interval, e.g. $[0, 1]$. Linear element functions, i.e. a piecewise linear ansatz, can be then defined as

$$\begin{array}{rclr} \phi_1^{elm} & = & 1 - x & x \in [0, 1], \\ \phi_2^{elm} & = & x & x \in [0, 1]. \end{array} \tag{3.48}$$

On the interval $[0, 1]$ ϕ_1^{elm} and ϕ_2^{elm} are orthonormal and the nodes are the points $x = 0$ and $x = 1$. This is depicted in Fig. 3.5 together with the ansatz-function for a grid with several elements.

Galerkin Methods

An important class of finite element methods is constituted by Galerkin methods [32]. Here the same functions are chosen as ansatz-functions as well as test functions. Then Eq.(3.46) reduces with Eq.(3.47) to a finite system of equations, namely

$$\sum_{i=1}^{N} \left(\frac{\partial}{\partial t} \alpha_i(t) \int_{\mathcal{V}} \phi_i \phi_j d\mathbf{x} - \alpha_i(t) \int_{\mathcal{V}} F \left(\sum_{k=1}^{N} \alpha_k(t) \phi_k \right) \phi_i \phi_j d\mathbf{x} \right) = 0 \qquad \forall j = 1 : N. \tag{3.49}$$

The term

$$\boldsymbol{M}_{ij} = \int_{\mathcal{V}} \phi_i \phi_j d\mathbf{x} \tag{3.50}$$

is also termed mass matrix. As practical example the diffusion equation

$$\frac{\partial}{\partial t} \Phi(x) = d \Delta \Phi(x) \quad x \in [0, 1],$$

is considered. Then F is the Laplace operator Δ. Using integration by parts one obtains

$$\sum_{i=1}^{N} \left(\boldsymbol{M}_{ij} \frac{\partial}{\partial t} \alpha_i(t) + \mathbf{K}_{ij} \alpha_i(t) \right) = 0 \quad \forall j = 1 : N, \tag{3.51}$$

$$\text{with } \mathbf{K}_{ij} := \int_{\mathcal{V}} \nabla \phi_i \nabla \phi_j d\mathbf{x}. \tag{3.52}$$

Eq.(3.51) is a set of N ODEs which can be solved by the methods presented in Section 3.5.1 to yield the coefficient vector α.

If further a one-dimensional system is considered and the ansatz-functions are chosen to be piecewise linear, one can directly calculate the matrices $\boldsymbol{M}$ and $\mathbf{K}$ exemplarily. In particular one gets for ϕ_i and $\nabla\phi_j$ (compare also with Fig. 3.5)

$$\phi_i = \begin{cases} x & , \quad x_{i-1} < x < x_i \\ -x & , \quad x_i < x < x_{i-1} \\ 0 & , \qquad \text{else.} \end{cases} \tag{3.53}$$

$$\nabla\phi_i = \begin{cases} 1 & , \quad x_{i-1} < x < x_i \\ -1 & , \quad x_i < x < x_{i+1} \\ 0 & , \qquad \text{else.} \end{cases} \tag{3.54}$$

Here x_i denotes the position of the i-th lattice node. This gives for $\boldsymbol{M}$ and $\mathbf{K}$

$$\boldsymbol{M}_{ij} = \delta_{ij}, \tag{3.55}$$

$$\mathbf{K}_{ij} = \begin{cases} -1 & , \quad i = \pm j \\ 2 & , \quad i = j \\ 0 & , \quad \text{else.} \end{cases} \tag{3.56}$$

The resulting dynamical system is exactly the same as one would obtain by a finite differencing scheme with the second order accurate Laplace operator from Eq.(3.37). This is a result of the particular choice of the ansatz-functions. Considering different ansatz-functions clearly gives a different system of equation.

Implementation of Boundary Conditions

Eq.(3.45) and Eq.(3.45) state the boundary condition for Eq.(3.44) above. Dirichlet boundary conditions were considered (Eq.(3.45)) as well as von Neumann boundary conditions (Eq.(3.45)). The boundary of $\mathcal{V}$, i.e. $\partial\mathcal{V}$ is split into parts, where these boundary conditions apply $\partial\mathcal{V} = \partial\mathcal{V}_D + \partial\mathcal{V}_N$.

For homogeneous Dirichlet conditions, i.e. $g(\mathbf{x}) = 0$ in Eq.(3.45) one can require the element ansatz-functions ϕ_i^{elm} to satisfy the condition in the elements that contain $\partial\mathcal{V}$. Then the same is true for the ansatz-functions $\phi_i(\mathbf{x})$. The element functions are typically constructed in a way so that just a few contributions have to be omitted.

Inhomogeneous Dirichlet conditions can be considered in a similar manner. Then an additional function ϕ^D is introduced, which satisfies the inhomogeneous Dirichlet conditions $\phi^D(\mathbf{x}, t) = g(\mathbf{x})$ on $\partial\mathcal{V}_D$. In the interior of $\mathcal{V}$ ϕ^D can be arbitrary. The new ansatz for the solution is then

$$\hat{\Phi}(\mathbf{x}, t) := \phi^D(\mathbf{x}) + \sum_{i=1}^{N} \alpha_i(t)\phi_i(\mathbf{x}). \tag{3.57}$$

Neumann conditions can typically be treated using integration by parts. This is exemplified for the Poisson equation with homogeneous von Neumann boundary conditions

$$\begin{array}{rrcll} & \Delta\Phi(\mathbf{x}) & = & F(\mathbf{x}) & \mathbf{x} \in \mathcal{V}, \\ \text{with} & (\hat{\mathbf{n}}\nabla)\Phi(\mathbf{x}) & = & 0 & \text{on } \partial\mathcal{V}. \end{array} \tag{3.58}$$

The Galerkin method together with the ansatz Eq.(3.47) results in

$$\sum_{i=1}^{N} \alpha_i \int_{\mathcal{V}} (\Delta\phi_i)\phi_j d\mathbf{x} - \int_{\mathcal{V}} \phi_j d\mathbf{x} = 0 \quad \forall j = 1 : N. \tag{3.59}$$

Application of Green's formula to Eq.(3.59) leads to

$$\sum_{i=1}^{N} \alpha_i \int_{\partial\mathcal{V}} ((\hat{\mathbf{n}}\nabla)\phi_i)\, \phi_j ds - \sum_{i=1}^{N} \alpha_i \int_{\mathcal{V}} \nabla\phi_i \nabla\phi_j d\mathbf{x} - \int_{\mathcal{V}} \phi_j d\mathbf{x} = 0 \qquad \forall j = 1 : N. \tag{3.60}$$

Including the boundary condition Eq.(3.58) this reduces to

$$\sum_{i=1}^{N} \alpha_i \int_{\mathcal{V}} \nabla\phi_i \nabla\phi_j d\mathbf{x} + \int_{\mathcal{V}} \phi_j d\mathbf{x} = 0 \quad \forall j = 1 : N, \tag{3.61}$$

where the homogeneous von Neumann boundary conditions are included implicitly. Inhomogeneous von Neumann boundary conditions as from Eq.(3.45) can be accounted for by an additional term $\int_{\partial\mathcal{V}} f(\mathbf{x})\phi_j d\mathbf{x}$ in Eq.(3.60). Due to the straightforward application of von Neumann boundary conditions for this type of problems these are also termed *natural* boundary conditions. In a similar spirit the Dirichlet boundary conditions above are termed *essential* boundary conditions.

3.5.4. Spectral Methods

Spectral methods can provide very accurate results or allow for reduced numerical effort for given accuracy. One reason for this is that discrete derivatives can be evaluated with higher precision since information from the whole space domain is used to calculate the derivative in one point. Further, differential operator have a very simple structure in Fourier space which can be exploited. The Fast Fourier Transform (FFT) allows an efficient application of these techniques.

Analogue to the Fourier transform of integrable functions a discrete function $\Phi(t_i)$, with equidistant $t_i \in [0, 2\pi]$, $i = 1 : N$, defined as $t_i = \frac{i2\pi}{N+1}$, can be decomposed by the Discrete Fourier Transform (DFT)

$$\tilde{\Phi}(k) := \frac{1}{\sqrt{N}} \sum_{j=1}^{N} e^{-ikt_j} \Phi(t_j), \quad k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}. \tag{3.62}$$

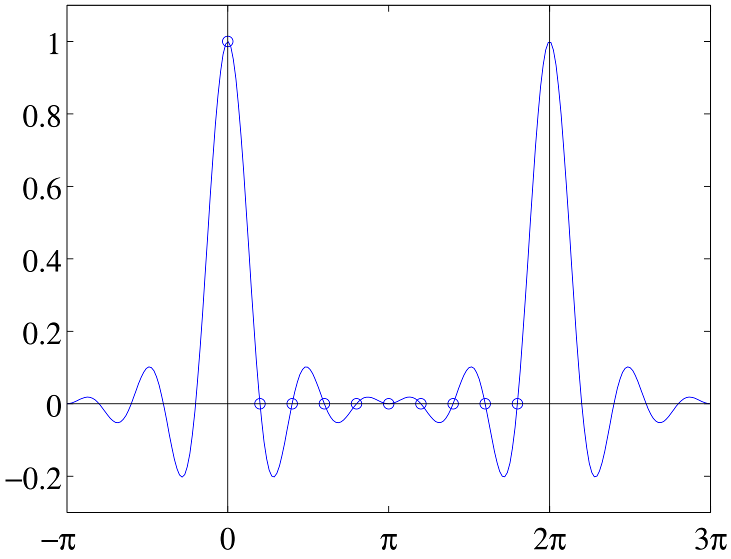


Figure 3.6.: Periodic sinc function for N=10.

The inverse discrete Fourier transform is given by

$$\Phi(t_j) := \frac{1}{\sqrt{N}} \sum_{k=-N/2+1}^{N/2} e^{ikt_j}\tilde{\Phi}(k), \quad j = 1, \ldots, N. \tag{3.63}$$

According to Eq.(3.63) the highest frequency mode (given by a sawtooth pattern) would have a nonzero derivative at the grid points. To mend this Eq.(3.63) is symmetrised according to the highest frequency to

$$\Phi(t_j) := \frac{1}{\sqrt{N}} {\sum_{k=-N/2}^{N/2}}' e^{ikt_j}\tilde{\Phi}(k) \quad j = 1, \ldots, N, \tag{3.64}$$

where the prime indicates to multiply the first and last summand by $\frac{1}{2}$.

The discrete Fourier transform is in fact a unitary basis transform.

Boundary conditions

The basis functions for the discrete Fourier transform are the Fourier modes $F_k(t_i) = \frac{1}{\sqrt{N}}e^{-ikt_i}$. These can be extended to the complete real axis and are 2π-periodic for all $k \in \mathbb{N}$. Consequently, each discrete $\Phi(t_i)$ can be extended to a 2π-periodic function $\Phi(t)$ on $\mathbb{R}$. $\Phi(t)$ is termed the band limited interpolant. Thus the functions are treated effectively as 2π-periodic. For derivatives one obtains therefore straightforward periodic boundary conditions.

Derivation

The derivative can simply be calculated by transforming to Fourier space, where derivation reduces to a multiplication, and transforming back.

$$\frac{d}{dt}\Phi(t)\bigg|_{t_j} = \frac{1}{\sqrt{N}} \sum_{k=-N/2}^{N/2}{}' ike^{ikt_i}\tilde{\Phi}(k) \quad j = 1, \dots, N \tag{3.65}$$

The use of the Fast Fourier Transform has contributed to make this approach attractive for efficient and accurate calculations. The number of multiplications required by the FFT is only $\mathcal{O}(N \ln N)$ compared to $\mathcal{O}N^2$ for general matrix multiplication. However, no use is made of the FFT in this work. A practically oriented description is given in [96].

For the following considerations obtaining an explicit matrix notation for the derivation operator D as in the sections before is of interest. To this end one simply has to calculate the derivative of the (orthonormal) basis vectors. These constitute the column vectors of the desired matrix representation D. Here the canonical basis $\mathbb{1}_{ij} = \delta_{ij}$ is chosen.

The transformed basis vectors are simply the Fourier modes $F_k(t)$. Transforming back yields for the approximant

$$\delta_j(t) = \frac{1}{N} \sum_{k=-N/2}^{N/2}{}' e^{ik(t-t_j)} \tag{3.66}$$

$$= \frac{1}{N}\left(\frac{1}{2}\sum_{k=-N/2}^{N/2-1} e^{ik(t-t_j)} + \frac{1}{2}\sum_{k=-N/2+1}^{N/2} e^{ik(t-t_j)}\right) \tag{3.67}$$

$$= \frac{1}{N}\cos\left(\frac{t-t_j}{2}\right)\sum_{k=-N/2+1/2}^{N/2-1/2} e^{ik(t-t_j)} \tag{3.68}$$

$$= \frac{1}{N}\cos\left(\frac{t-t_j}{2}\right)\frac{e^{i(-N/2+1/2)(t-t_j)} - e^{i(N/2+1/2)(t-t_j)}}{1-e^{i(t-t_j)}} \tag{3.69}$$

$$= \frac{1}{N}\cos\left(\frac{t-t_j}{2}\right)\frac{e^{-i(N/2)(t-t_j)} - e^{i(N/2)(t-t_j)}}{e^{-i(t-t_j)/2} - e^{i(t-t_j)/2}} \tag{3.70}$$

$$= \frac{1}{N}\cos\left(\frac{t-t_j}{2}\right)\frac{\sin\left(\frac{N(t-t_j)}{2}\right)}{\sin\left(\frac{t-t_j}{2}\right)}. \tag{3.71}$$

The function

$$S_N := \frac{\sin\left(\frac{Nt}{2}\right)}{N\tan\left(\frac{t}{2}\right)}, \tag{3.72}$$

is also termed *periodic sinc function.* It is presented in Fig. 3.6. On the grid points it reproduces the canonical basis vector, but shows additional oscillations due to the finite bandwidth of the DFT. For the derivative in the grid points for the first canonical basis vector one obtains

$$\left.\frac{d}{dt}S_N(t)\right|_{t_j} := \begin{cases} 0, & j \mod N = 0 \\ (-1)^j \frac{1}{2}\cot(j\pi/N), & j \mod N \neq 0. \end{cases} \tag{3.73}$$

All other columns of D_S can be calculated by simply translating the index j. Thus the derivation matrix for the spectral method D_S is a Töplitz matrix, determined by the first column. Higher derivatives can be calculated in a similar manner. Only the second derivative D_S^2 will be used in addition. It is also a Töplitz matrix with first column

$$\left.\frac{d^2}{dt^2}S_N(t)\right|_{t_j} := \begin{cases} -\frac{N^2}{12} - \frac{1}{6}, & j \mod N = 0 \\ -\frac{(-1)^j}{2\sin^2(j\pi/N)}, & j \mod N \neq 0. \end{cases} \tag{3.74}$$

As in the methods before the first derivation operator D_S is antisymmetric and the second derivation operator D_S^2 is symmetric. Beside other simplifications that are possible due to the special structure of a Töplitz matrix also the matrix product can be calculated within $\mathcal{O}(N \ln N)$ multiplications.

Note that the derivation operators are now dense matrices. In the spectral method the maximally available information on the discrete function is used to determine the derivative in one point. Due to this fact spectral methods provide a high accuracy for smooth functions. Unfortunately, they cannot easily applied to problems with more complex geometry. Also the choice of boundary conditions is restricted. For non-periodic geometries extensions as e.g. Tschebyschev[7] methods exist but no use of them is made in the following.

[7] In literature some alternative translations exist as Tschebyschow, Tschebyscheff, Tschebycheff, Tschebyschew or Chebychev

4. Master Equation Description

Modelling physical (or other) systems by microscopic or pseudo-microscopic models can have various reasons. First, it is not always possible to find closed form equations. It can be easier to construct microscopic rules from the knowledge of a system. Closed form equations also usually have a limited validity range. The Navier-Stokes equations e.g. cannot describe the flow on a molecular level. Thermal fluctuations are not included and transport coefficients have to be derived from experiments etc. [57]. With the advance of computer technology simulations of highly complex systems (also regarding their constituents), e.g. from biology have become possible. There, microscopic or pseudo-microscopic models can reproduce the behaviour of such systems [104]. It is often only necessary to capture some relevant properties to find a useful model of a complex system. The gas-liquid phase transition e.g. can already be observed in a system of hard spheres. Polymers can be simulated by random walks or chains of spheres [39, 108]. Complex biological systems like membranes can be studied by strongly simplified bilipid molecules [105]. Some relevant behaviour of protein folding and molecular recognition can be studied on discrete lattice systems with a simple nearest neighbour interaction [18, 11]. Even continuous systems like the Navier-Stokes flow can be modelled by pseudo microscopic or mesoscopic models as dissipative particle dynamics, or by lattice Boltzmann methods [109].

In this chapter an ansatz is presented which does not deal with PDEs directly. As stated above many PDEs arise from systems with a microscopic dynamics which is far to complicated to be treated explicitely. The microscopic details are also often irrelevant for the macroscopic variables of interest. One example are the Navier-Stokes equations, which describe the behaviour of a fluid or gas consisting of a huge number of molecules. One litre of air at standard atmosphere conditions at sea level contains e.g. approx. $2.54697 \cdot 10^{22} = l\frac{N_a P_s}{R^* T_s}$ molecules [1].

However, the progress in Density Matrix Renormalisation Group (DMRG) techniques has made the treatment of systems with an extremely high number of degrees of freedom possible, at least in the spatially one dimensional case. Now a stochastic description of microscopic models will be introduced that includes also models for which the generator of evolution depends on the state of the system. Such a model can be considered to be nonlinear. Chapter 8 includes an explicit example of a lattice model for a reaction diffusion process and a KPZ-type partial differential equation.

[1] Avogadro number $N_a = 6.022169 \cdot 10^{23} \frac{1}{\text{mol}}$, gas constant $R^* = 8.31432 \frac{\text{J}}{\text{molK}}$ [63, 64] Standard atmosphere conditions at sea level: $P_s = 1013.25\text{hPa}$, $T_s = 288.15°\text{K}$.

Dimensionality versus Linearity

Considering a system on a one-dimensional lattice with N sites and one scalar variable on each site, the phase space would classically be N-dimensional. As explained in the previous chapters the generator of evolution depends on the state of the system in the nonlinear case. An alternative is to treat every point in phase space as a degree of freedom of its own. The system becomes then automatically infinite-dimensional if continuous values are permitted at the grid nodes. This is due to the fact that already a one-dimensional finite interval contains an uncountably infinte set of numbers. For discrete valued sites, e.g. two possible states (spin up/spin down), the system stays finite dimensional. This is the case for cellular automata (CA). In the high-dimensional description the dynamics can simply be determined by transition rates. The generator of evolution in this space can be written down once for all (excluding explicit time dependence) and is thus linear. It is called master operator $\mathcal{M}$ and contains the rates for all possible transitions between two states. It is in general not symmetric.

The master operator acts on a high-dimensional space, which is spanned by all possible microscopic states. This space is m^N-dimensional, with m the number of states per site. In this vector space state vectors Ψ can be defined. The components of the state vector describe the probability to find the system in the corresponding microscopic state. If a microscopic configuration is denoted with $|i\rangle$ and the probability for this configuration with $p(|i\rangle)$ this gives

$$\Psi_i = p(|i\rangle). \tag{4.1}$$

Following this probabilistic interpretation, physical relevant states have to be normalised. The natural normalisation would be

$$\sum_i \Psi_i = 1, \tag{4.2}$$

instead of a L^2-norm of 1 as in quantum mechanics. Note, that while the description is probabilistic also deterministic behaviour can be included in this framework.

This approach has some similarities with quantum mechanics. Considering a single classical particle on an one-dimensional grid, its phase space is two-dimensional. If the grid has N nodes, the quantum mechanical system has a $2N$-dimensional real phase space. Now superpositions of different classical states are allowed. While the classical equations of motion can be nonlinear, the quantum mechanical description by the Schrödinger equation is always linear. However, unlike quantum mechanics the master equation ansatz deals with probabilities directly instead of probability amplitudes. Consequently, no interference effects can occur.

More quantitatively, a system with finite phase space V, i.e. the phase space is a finite set $V = \{|i\rangle\}_{i=1:N}$ is compared with the corresponding master equation approach.

	Direct description	Master equation approach
System state	$\psi(t) = \lvert i\rangle$ for some $i = 1 : N$	$\Psi(t) = (p(\lvert 1\rangle), \ldots, p(\lvert N\rangle))$
Time evolution	Updating rule depending on ψ	Master operator $\mathcal{M}$
Interpretation	Single realisation of ensemble	Probability distribution for all physical states

Stochasticity from Missing Information

In the description above also deterministic processes can be included. Then the rates in the master equation are either ±1 or 0 depending on whether the transition occurs under the dynamics or not. Even in such a case a reduction as described in Chapter 5 in general always leads to a probabilistic model. This is due to the fact that a simple structure of a matrix is not necessarily preserved under a projection. This is a practical example for creation of randomness through missing information. In a more general sense randomness in every day's life is often due to missing information and this is one basic motivation for stochastic models.

Time Evolution

The stochastic description explained above makes sense for continuous as well as discrete time evolution. For continuous time the dynamics is given by the master equation

$$\frac{\partial}{\partial t}\Psi = \mathcal{M}\Psi. \tag{4.3}$$

A discrete version could be obtained by the explicit Euler method, Eq.(3.17). For the evolution only the previous time step (which would be infinitesimal in the continuous case) is relevant. This is also true for the microscopic processes considered here. The stochastic process they describe is thus termed to be a Markov process. The formal solution to the dynamic equation Eq.(4.3) is given by

$$\Psi(t) = e^{-(t-t_0)\mathcal{M}}\Psi(t = t_0). \tag{4.4}$$

This is a linear equation, so it is convenient to consider it in the eigenvector representation[2]. Since $\mathcal{M}$ is non-hermitian the eigenvalues can be complex. As the matrix representation of $\mathcal{M}$ is real valued, the complex eigenvalues are always accompanied by their complex conjugate. Further, $\mathcal{M}$ is in general not normal, therefore the eigenvectors need not to be orthogonal. The evolution is then simply an exponential decay of all modes with life time $\frac{1}{\Re\lambda}$ with λ being the eigenvalue. The imaginary part results in an oscillation within the subspace spanned by the complex conjugated eigenvector pair. It is obvious that for relevant physical systems the condition $\Re\lambda \leq 0$ must hold to prevent unbounded increase of $||\Psi(t)||$. The problem of complex eigenvectors can be circumvented by finding a real basis

[2] Even if $\mathcal{M}$ would be defective this poses no problem. The orthogonal complement of all eigenspaces is then simply not affected by the dynamics, which is thus a trivial case.

for each subspace spanned by a complex conjugated pair of eigenvalues/states, which is always possible.

4.1. Constructing the Master Operator

In all models considered in this work the dynamics is defined by processes that affect only a single site or two nearest neighbour sites locally. Not adjacent sites cannot affect each other in one time step. The interactions are thus local. To define the master operator explicitely it is convenient to use single site creation $a^\dagger$ and destruction a operators. They are defined as

$$a = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad a^\dagger = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{4.5}$$

Within this representation $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ denotes an empty site, while an occupied site is represented by $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Together with the occupation operator n and the vacancy operator v

$$n = a^\dagger a = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad v = a a^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.6}$$

they constitute a basis for $\mathbb{R}^{2\times 2}$. All single site operators can be constructed as linear combinations of a, $a^\dagger$, n and v. Two site operators are constructed similarly from products of single site operators.

Source and Sink Operator

The source and the sink operators are examples for single site operators. Their effect is to bring the site from an empty to an occupied state (source) or vice versa. The source operator is defined by

$$\mathcal{S} = a^\dagger - v = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}. \tag{4.7}$$

The sink operator is given by

$$\mathcal{S}^+ = a - n = \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix}. \tag{4.8}$$

Note that each colum sums up to zero. With this as constraint S and S^+ are the only possible single site operators.

Diffusion Operator

The diffusion operator is a two site operator. The left diffusion and right diffusion operators are defined as $\mathcal{D}_l$, $\mathcal{D}_r$. Applied to two adjacent sites it brings one site (the right for $\mathcal{D}_l$) from an occupied to an empty state, while simultaneously bringing the left site (for $\mathcal{D}_l$) from an empty to an occupied state. This leads effectively to a transport to the left. For the right diffusion operator the sides are reversed. More explicitely this can be written as

$$\mathcal{D}_l = a_i a_j^\dagger - n_i v_j, \quad \mathcal{D}_r = a_i^\dagger a_j - v_i n_j. \tag{4.9}$$

Again the columns add up to zero.

Annihilation Operator

Pair annihilation is a strongly simplified model for a reaction of two particles. The corresponding two site operator is defined by

$$\mathcal{A} = a_i a_j - v_i v_j. \tag{4.10}$$

4.1.1. Probability Conservation and Steady State

All operators[3] described above have the property that the columns sum up to zero. This is equivalent to probability conservation and is necessary for any physically relevant description. A matrix with this property is also called a stochastic matrix. Note that there is typically no particle conservation (interpreting the state $\binom{0}{1}$ as particle).

As the master operator $\mathcal{M}$ is in general represented by a non-symmetric, non-hermitian matrix, different left and right eigenstates to an eigenvalue generally exist. For stochastic matrices one left eigenstate is trivially always known from the probability conservation property. Writing this constraint explicitely for the master operator one gets

$$0 = \sum_i \mathcal{M}_{i,j} = \langle s| \mathcal{M} \quad \text{with} \quad \langle s|_i = 1. \tag{4.11}$$

Here $\langle s|$, also called summation state, is always a left eigenstate of $\mathcal{M}$ to the eigenvalue 0. Therefore always a steady state exist.

While the existence of the steady state is alway guaranteed by construction, this steady state can be degenerated. In such cases the initial conditions decide which steady state is reached so they can influence the system for all later times. In statistical mechanics the concept of ergodicity is relevant [14, 119, 120]. It states basically whether the probability to reach any state in finite time from an arbitrary state by the dynamics is strictly positive. It plays an important role to

[3] Apart from a, $a^\dagger$, v and n which do not describe physical processes.

justify the description of complex microscopic dynamics by stochastic models. In the ergodic case the steady state is unique [106, 77], which identifies all systems with degenerated steady state as non-ergodic.

An even stronger condition than stationarity as for the steady state is the so called detailed balance. If the probability for a state $|i\rangle$ is P_i, then the detailed balance condition is

$$\mathcal{M}_{ij}P_j = \mathcal{M}_{ji}P_i. \tag{4.12}$$

This states simply that the probability to be in state j and jump to state i is equal to the probability to be in state i and jump to state j. While for a steady state only $\sum_j \mathcal{M}_{ij}P_j = 0$ is required, in general Eq.(4.12) is not satisfied [68].

5. Dynamical Systems and Model Reduction

Though this be madness, yet there's method in't.

Hamlet, William Shakespeare

The aim of this work is to devise and present algorithms which allow to calculate effective models for large dynamical systems. In the literature this approach is counted to the field of model reduction [4]. I will consider finite-dimensional systems exclusively. These are either obtained from microscopical descriptions of lattice models directly or from discretised partial differential equations.

5.1. Problem Setup

Dynamical systems arise wherever time dependent processes have to be described. They can be based on heuristic descriptions or on complex theories. They can also be derived from empirical data. A description on a high level is given by PDEs which typically arise from e.g. physical theories. However, it is also possible to derive models directly from the microscopical behaviour of a system.

In case of a PDE description, a system of ODEs is obtained by discretisation of the spatial coordinates, as described above in Chapter 3. The dimensionality N of the system of ODEs can be principally chosen arbitrarily large but it is practically limited by the ability to process the resulting system. The discretisation also leads in general to a discretisation error. Typically, the discretisation error decreases with increasing dimensionality of the ODE system and vanishes in the infinite limit. The system of ODEs are given in the following form

$$\frac{d}{dt}\phi(t) = G(\phi(t), t)\phi(t) + \mathbf{F}(t). \tag{5.1}$$

Here ϕ is a N-dimensional vector, representing the internal degrees of freedom. $\mathbf{F}$ is an external forcing, but in the following this term will be zero. This reduces Eq.(5.1) to a homogeneous ODE. G is the generator of evolution and has the form of a $N \times N$ matrix. Nevertheless G can be dependent on both $\phi(t)$ and t explicitely. Throughout this work the explicit t-dependence will always be excluded. In the literature the case in which the time dependence of the right hand side of Eq.(5.1) is only due to $\phi(t)$ directly is also termed to be the autonomous case [61]. Often

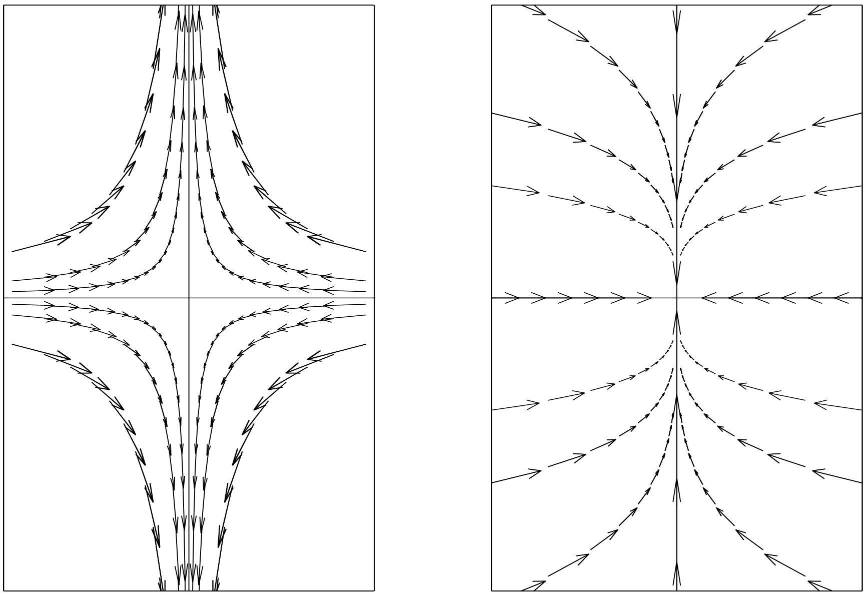


Figure 5.1.: Phase space diagram of a saddle point $\lambda_1 < 0 < \lambda_2$ and a stable node $\lambda_2 \leq \lambda_1 \leq 0$. The horizontal direction is the eigenspace corresponding to λ_1. For an unstable node only the arrows in the right sketch have to be reversed.

only finite powers of the variable ϕ occurs. Thus Eq.(5.1) can be described by some tensors of increasing order, e.g. up to third order as

$$\frac{d}{dt}\phi = L_{ij}\Phi_j + Q_{ijk}\Phi_j\Phi_k + K_{ijkl}\Phi_j\Phi_k\Phi_l, \tag{5.2}$$

where the contributions L, Q and K represent the linear, the quadratic and the cubic part, respectively.

In dynamical systems theory not the whole state of the system is considered further to be accessible but only some observables defined by linear maps [4]

$$\mathbf{y} = C\phi(t) + DF(t). \tag{5.3}$$

This description is very appropriate in control theory [4] for existing dynamical systems. In contrast, each component of ϕ is assumed to be assessable in this work for simplification. Only in Chapter 8 some selected observables are presented instead of the system state itself which is very high-dimensional and of poor intuitive comprehensibility.

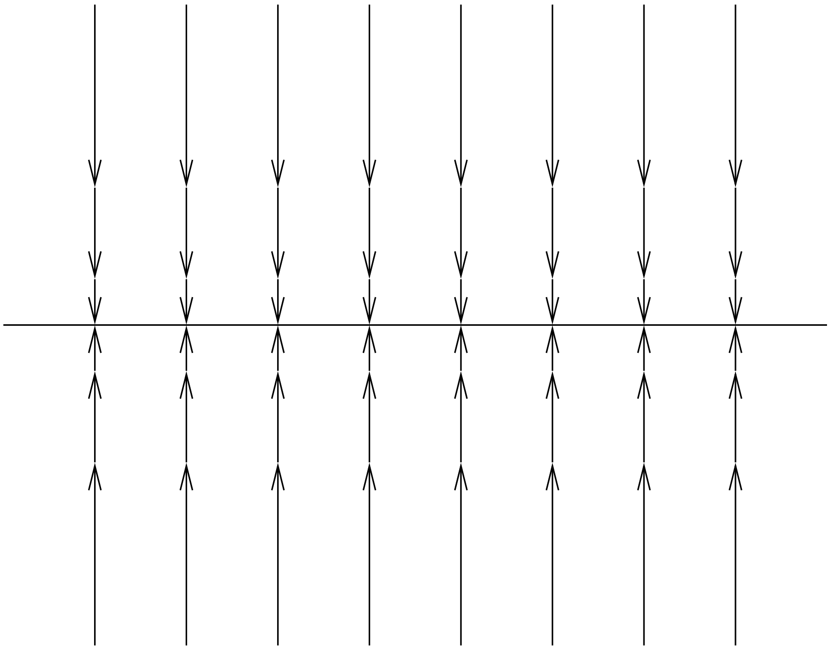


Figure 5.2.: Phase space diagram for the special case of a stable node with $\lambda_2 < \lambda_1 = 0$. The horizontal direction is the eigenspace corresponding to λ_1.

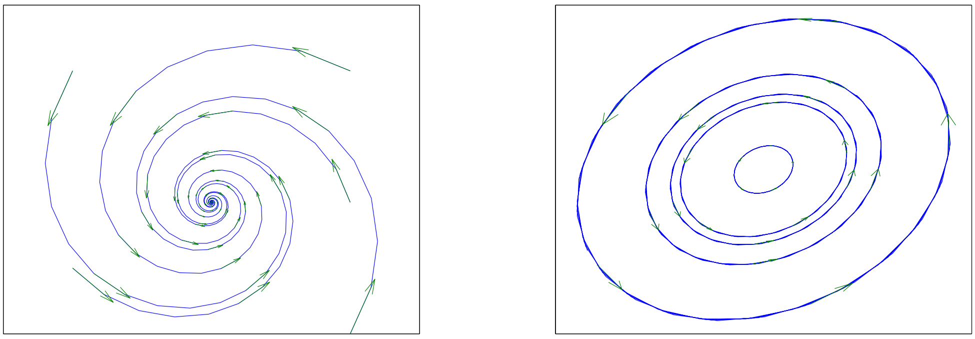


Figure 5.3.: Phase space diagrams, left : For a so called sink for a complex conjugated pair of eigenvalues. The real part is negative $\Re\lambda < 0$. A positive real part is unphysical. Right : Phase space diagram for a purely imaginary pair of eigenvalues.

Linear Systems

Homogeneous linear systems are encountered in Chapter 8 extensively and also later as trivial test problems. There Eq.(5.1) reduces to

$$\frac{d}{dt}\phi(t) = G\phi(t), \tag{5.4}$$

which can be integrated formally. The solution is [61]

$$\phi(t) = e^{tG}\phi(t=0). \tag{5.5}$$

Depending on the eigenvalues λ_i of G some scenarios are possible. Since the eigenspaces decouple, it is sufficient to consider, e.g. two-dimensional subspaces. One possibility is the occurrence of real eigenvalues. Depending on the sign of λ_i the dynamics in the corresponding eigenspaces is stable $\lambda_i \leq 0$ or unstable $\lambda_i > 0$. In phase space this gives rise to a node (stable), saddle or unstable node, as depicted in Fig. 5.1. A saddle point is obtained for $\lambda_1 < 0 < \lambda_2$. The exponential growth is unphysical[1] and no systems with such properties occur in this work. It is nevertheless a trivial example for sensitivity to initial conditions in linear systems. The same holds for an unstable node defined by $0 < \lambda_2$, $0 \geq \lambda_1$. The stable node with $\lambda_2 \leq \lambda_1 \leq 0$ is the generic case for the problems at hand. Fig. 5.1 presents the case for $\lambda_2 \neq \lambda_1$. The Figure suggests also that the eigenspace with largest eigenvalue is more relevant. Trajectories approach this subspace before approaching the origin. The special case of $\lambda_2 = \lambda_1$ is also termed focus and has rotational symmetry around the origin. Situations in which the kernel of the generator of evolution is non-trivial will occur in this work. Then some eigenvalues are zero and along the corresponding eigenspaces no dynamics occurs. This is the only case for which the initial conditions determine the long time solution significantly and those eigenvectors must not being projected out.

In general complex conjugated pairs of eigenvalues can (and will) occur. For the stability again only the real part of the eigenvalues is relevant. Fig. 5.3 (left) shows a so called sink with negative real part. The trajectories oscillate in the corresponding two-dimensional invariant subspace and converge in the origin. The convergence is the faster the smaller (more negative) the real part of the eigenvalue is. A special case is again a zero real part, shown in Fig. 5.3 (right). Then the trajectories are ellipses around the origin. As in the case of a zero eigenvalue the initial conditions influence the system state to all later times significantly. Also the corresponding sub-space has to be included in any meaningful reduction. Thus it can be stated that - concerning model reduction - only the real part of an eigenvalue determines the relevance of its invariant sub-space. From Fig. 5.3it is also evident that pairs of eigenvectors for complex conjugated eigenvalues must not be separated by a reduction, i.e. either projecting out both or none.

[1] For linear systems the exponential increase of some mode is not restricted to a certain region in phase space, which can occur in nonlinear systems. An exponential increase, indicated by a strictly positive real part of an eigenvalue, always leads to a divergence of the solution, i.e. also the norm of the solution diverges. This is considered to be unphysical.

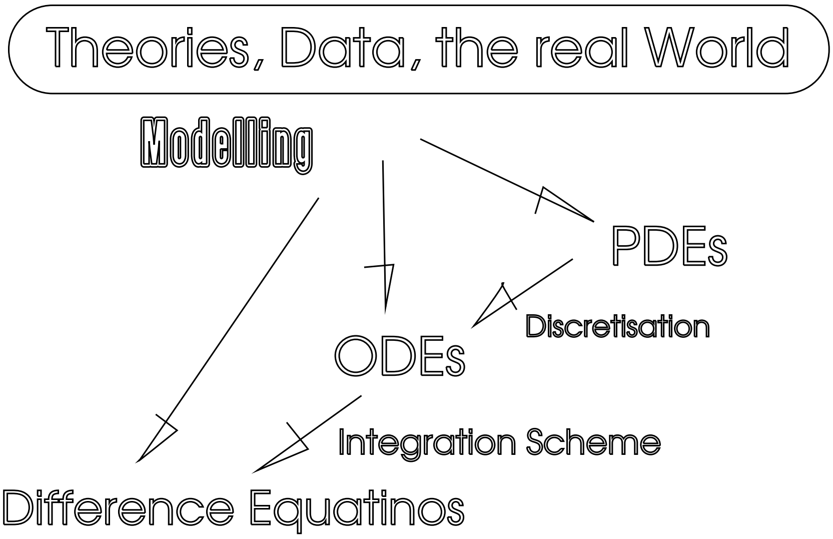


Figure 5.4.: Relation between several formulations

Time Discrete Systems

The following considerations will be restricted to systems with discrete time. The discretisation of the system of ODEs is also described in Chapter 3, compare e.g. Eq.(3.25). This leads to a set of difference equations:

$$\phi_t = \sum_i \left(\alpha_i \phi_{t-i} + \beta_i G(\phi_{t-i})\phi_{t-i}\right) \tag{5.6}$$

The index i is usual positive $i \geq 0$. Then the evolution is termed *causal*. Typically only a few coefficients are nonzero, in this work at maximum three in the multi step method in Chapter 10. In Eq.(5.6) $t \in \mathbb{N}$ denotes a discrete time index and the α_i, β_i determine the ODE integration scheme to be used, see Tab. 3.5.1. As indicated in Eq.(5.6) the generator of evolution $G(\phi_{t-i})$ depends itself on ϕ, thus the dynamics can be nonlinear.

5.2. Model Reduction

The are many possible motivations to determine a reduced model for a dynamical system. One reason is certainly to reduce computational effort or memory requirement. Whether this can be achieved is also implementation or problem dependent. Further the reduced model can be possibly analysed directly. The modes calculated by the proper orthogonal decomposition, e.g. can give a descriptive picture of the relevant processes during the time evolution. A study of certain 'modes' to analyse data is also of interest in very different fields as medical image processing [76].

5.2.1. Choice of error functional

In the previous Section 5.1 it has been considered how dynamical systems can be modelled and solved, see Chapter 3. For practical applications such models can be highly complex, see [4] for an overview. Model reduction can provide simplified descriptions which still reproduce the behaviour of the original system. To measure the quality of a reduction some error functional has to be specified. Then the reduction is chosen to minimise this error. As stated above, the values of the fields at the nodes are of interest. A natural choice for the error would then be the L^2-error

$$E_2(t) = ||\Phi(t) - B\hat{\Phi}(t)||_2. \tag{5.7}$$

Here $\hat{\Phi}$ is the reduced field which is defined on the smaller reduced phase space. To calculate the error $\hat{\Phi}$ needs to be embedded into the original phase space. This is denoted by $B\hat{\Phi}$. Later this will be indeed a matrix-vector product, but for now this notation has to be understood as a bit more general. The error defined in Eq.(5.7) is not the only meaningful choice. The derivative of the field or other observables are possibly poorly reproduced by requiring $E_2(t)$ to be minimal. In practical applications often only a few observables as lift and drag coefficients and some other momenta in aerodynamics or stress and displacement at some particular points in structural mechanics calculations have to be determined with good accuracy. Then also alternative approaches are useful [83].

5.2.2. Reduced Model

The reduced model should have a significantly lower complexity as the original system. Typically the complexity of the reduced model is prescribed by the available computer power and storage. Ideally, also the dynamics of a nonlinear system should lead to an invariant manifold in phase space. Trajectories approach such a manifold (typically exponentially fast) and are bound to it for all later times. In the linear case these manifolds are the tensor products of eigenspaces. The eigenspaces can be treated independently. The relevant criteria whether a subspace should be projected out or is relevant for a reduction have been considered above. For nonlinear systems on the other hand the invariant manifolds are in general no subspaces. Even whether such manifolds exist is not clear a priori. This complicates the description, since a suitable parametrisation has to be found. In practice even a very low-dimensional manifold can lead to a significant amount of complexity.

5.2.3. Linear Projection

This is arguably the simplest ansatz. Already the discretisation of a PDE itself is a reduction of this type. The reduced phase space is a subspace of the original phase space. It is completely determined by a basis for the reduced phase space and the complexity of the reduction is determined by the dimensionality M of the reduced phase space. For convenience and also to avoid unnecessary numerical inaccuracy an orthonormal basis B should be chosen with $B^\dagger B = BB^\dagger = \mathbb{1}$. The projection operator to the reduced phase space is then $P = BB^\dagger$. The reduced dynamics is given by

$$P\phi_t = \sum_i \left(\alpha_i P\phi_{t-i} + \beta_i G(P\phi_{t-i})P\phi_{t-i}\right). \tag{5.8}$$

Practically the dynamics of the M-dimensional reduced model is considered. This work is restricted to systems which have a polynomial dependence on the field ϕ so that Eq.(5.2) holds and the generator of evolution G is given by the tensors L, Q and K. Their transformation properties are known so that one can introduce the reduced entities

$$\begin{aligned} \hat{\Phi} &= B^\dagger \Phi, && (5.9)\\ \hat{L} &= B^\dagger L B, && (5.10)\\ \hat{Q}_{i,j,k} &= \sum_{a,b,c} B^\dagger_{i,a} Q_{a,b,c} B_{b,j} B_{c,k}, && (5.11)\\ \hat{K}_{i,j,k,l} &= \sum_{a,b,c,d} B^\dagger_{i,a} K_{a,b,c,d} B_{b,j} B_{c,k} B_{d,l}. && (5.12) \end{aligned}$$

The reduced model dynamics can be written as

$$\partial_t \hat{\Phi} = \hat{L}_{ij}\hat{\Phi}_j + \hat{Q}_{ijk}\hat{\Phi}_j\hat{\Phi}_k + \hat{K}_{ijkl}\hat{\Phi}_j\hat{\Phi}_k\hat{\Phi}_l. \tag{5.13}$$

Note, that the simple structure of the operators (band-diagonal or Töplitz matrix, etc.) is now in general lost. For the time dependent error one obtains

$$\mathbf{E}(t) = \Phi(t) - B\hat{\Phi}(t) = (\mathbb{1} - P)\Phi(t). \tag{5.14}$$

The L^2-error is also time dependent and given by

$$||\mathbf{E}(t)||_2 = ||(\mathbb{1} - P)\Phi(t)||_2. \tag{5.15}$$

5.2.4. Nonlinear Reductions

Nonlinear reductions cannot be described as uniformly as linear methods. This approach will not be pursued in the following. The efficiency of this approach depends on how the invariant manifold is parametrised. Gorban e.g. proposed a numerical description by so called invariant grids [51]. Also the complexity of the invariant manifold depends on the system at hand.

A further problem is to find a projection to this manifold. Starting at an arbitrary point of phase space, it is not clear which the corresponding start point for the reduced description is.

An other approach starts from a linear reduction and tries to improve it by introducing contributions from the irrelevant subspaces as function of the relevant degrees of freedom [82]. This is an application of the so called slaving principle [54].

Summarising, the nonlinear methods depends much stronger on the particular problem as the linear approach. In particular the complexity of the reduced system cannot be controlled very systematically. This work is restricted to linear projections. Thus the minimal dimensionality of the reduced model of given accuracy is sacrificed for a simple description of this model.

5.2.5. Optimal Model Reduction for Linear Dynamics

In case of a linear generator of evolution the optimal linear orthogonal projection (in the L^2 sense) can be determined directly. For optimality the L^2-error of full and reduced dynamics is required to be minimal. This means, B_{opt} is chosen so that

$$\langle E(t)\rangle_t = \left\langle ||\Phi(t) - B_{opt}B_{opt}^\dagger \Phi(t)||_2 \right\rangle_t \overset{!}{=} \text{minimal.} \tag{5.16}$$

Since the error in Eq.(5.15) is time dependent, the time averaged error is considered as indicated by $\langle\cdot\rangle_t$. At least for sufficiently long times the arguments from Appendix B hold. There it is shown that the range of the projector should contain the eigenstates corresponding to the lowest (absolute real part) eigenvalues and it should be invariant under G. For symmetric matrixes this is a well defined problem for which many algorithms exist [96, 53, 47, 99].

Also for general matrices these eigenstates exist but they can be complex for non-symmetric G.[2] They can even be non-orthonormal if G is not normal i.e. $G^\dagger G \neq GG^\dagger$. This situation will be encountered in Section 8. The most straightforward approach would be choosing the relevant eigenvectors of G as column vectors of B. However, this is suboptimal. For numerical reasons one should always choose B orthonormal which is always possible. The requirements above only concern the spectrum of the reduced generator of evolution $A := B^\dagger G B$. To meet the above requirements, only the following form with an orthonormal basis B_F is needed

$$B_F^\dagger G B_F = \begin{pmatrix} A & C \\ 0 & D \end{pmatrix}. \tag{5.17}$$

Here the first columns of B_F constitute B. The spectra of A and D constitute the spectrum of G and $A = B^\dagger G B$. An optimal reduction is then obtained if the spectra of A and D contain the appropriate eigenvalues of G. The off-diagonal contribution C is irrelevant for the reduced model. The subspace spanned by the columns of B is still G-invariant. Of course, the subspace spanned by the rest of B_F is not G-invariant which in turn is not necessary.

[2] G is assumed to be non-defective. In case of a defective G the eigenvectors do not span the whole space. Thus the orthogonal complement of the eigenspaces is not affected by the dynamics and has to be treated as a zero eigenvalue eigenspace.

Here we recall that any real matrix A can be orthonormally transformed to a quasi upper triangular matrix S with 1×1 or 2×2 blocks on the diagonal and zero below this diagonal.

$$Q^{\dagger}AQ = \begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ 0 & R_{22} & \cdots & R_{2m} \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \cdots & R_{mm} \end{pmatrix} \tag{5.18}$$

The 2×2 blocks arise form pairs of complex conjugated eigenvalues and can only occur for non-symmetric matrices. Non-normality is indicated by a nonzero upper diagonal part. If one denotes the block diagonal part of A with D this means

$$A^{\dagger}A = AA^{\dagger} \Leftrightarrow A - D = 0. \tag{5.19}$$

In the non-normal case one can either choose a non-orthonormal basis for which A becomes block diagonal, or alternatively an orthonormal basis O where the upper triangular part of $O^{\dagger}AO$ is non-zero. Provided no 2×2 blocks are cut apart Eq.(5.18) has the form of Eq.(5.17) for several possible segmentations. Now it becomes obvious that ordering the Schur form Eq.(5.18) leads to a nested sequence of invariant subspaces which all provide an optimal approximation with a given complexity. More quantitatively

$$V_i := \operatorname{span}(v_1, \dots, v_i) \quad , \quad i = 1, \dots, m \tag{5.20}$$

$$V_i \subset V_j \quad , \quad \text{for } i < j \tag{5.21}$$

$$V_i \ A\text{-invariant} \quad , \quad \forall i = 1, \dots, m. \tag{5.22}$$

The v_i are either single Schur vectors or pairs of Schur vectors, m is the number of diagonal Blocks in A.

5.2.6. Optimising Model Reduction for Nonlinear Dynamics

For the more general nonlinear systems as defined by equation 3.13 the choice of the projected subspace is a priori not clear. Since $G(\Phi(\mathbf{x}, t))$ is not constant any more, the decomposition into invariant subspaces also depends on $\Phi(\mathbf{x}, t)$.

Linearisation

As long as only the neighbourhood of a phase space point Φ_{fix} is of interest, e.g. an equilibrium point, the linearisation in that point, i.e. $G(\Phi_{\text{fix}})$ can be investigated. This linearised system

$$\frac{\partial}{\partial t}\phi(\mathbf{x}, t) = G(\Phi_{\text{fix}})\phi(\mathbf{x}, t) \tag{5.23}$$

can be treated as described before. It should be assured that the solution stays close to Φ_{fix}, otherwise the error can increase uncontrollably.

Minimisation Approach

The minimisation approach has been proposed by Degenhard et al. [37]. Here the Frobenius norm of the error operator Eq.(5.8) is minimised for some ansatz state v. Starting by $P = \mathbb{1}$ the next projection operator is given by $P = \mathbb{1} - vv^\dagger$. Iteratively more and more states are calculated and thus the dimensionality of the model reduced. The minimisation is carried out numerically by a sequential quadratic programming method [49]. The numerical scheme was provided by a commercial software library [92]. The ansatz is plagued by various problems. First all ansatz states are normalised. As stated above the columns of B and consequently those of $\mathbb{1} - B$ should be orthonormal. Such a restriction on the other hand during the minimisation of the error leads to ignoring almost the whole phase space. Further the parameters were chosen in such a way that the influence of the nonlinearity was negligible. The KPZ-equation was studied which is known to become unstable by finite differencing schemes [34]. The computational effort for the minimisation was astronomical (1 hour on a SUN-SPARK Ultra 10 workstation for N=16) compared to a proper orthogonal decomposition (POD) (simulation of some trajectories plus diagonalisation of a symmetric $N \times N$ matrix, see Section 5.2.7 below). Further for the POD optimality (in some sense) can be proven, as is exemplified in the following.

5.2.7. Proper Orthogonal Decomposition

In order to find an optimal linear projection for nonlinear systems one certainly has to incorporate information from the nonlinearity, i.e. from $G(\Phi(\mathbf{x}, t))$ for the whole phase space. Systematically this is done by the proper orthogonal decomposition (POD). The proper orthogonal decomposition is a linear projection method which is widely used in model reduction. An extensive literature on this topic exists. Some examples are [79, 107, 12, 100, 91]. A short explanation of POD together with the method of snapshots is also given in [23]. One of the advantages of this method is the possibility to incorporate information from the nonlinear dynamics to obtain a linear reduction. In practice the basic idea is to generate sample trajectories by simulating the dynamical system of interest.

From the dynamic equations Eq.(5.1), one can define a time average, if a set of trajectories is specified. Formally this could comprise even all possible trajectories. The time average of some observable A is

$$\langle A \rangle_{\mathbf{T}} := \frac{1}{n} \sum_{k=1}^{n} \int A(\Phi^k(\mathbf{x}, t)) dt, \tag{5.24}$$

where n is the number of sample trajectories and Φ^k are the state vectors for the k-th trajectory. A common optimality condition [107, 4] is requiring the average least square truncation error being minimal. Although this is not the only sensible choice, see Section 5.2.1, this condition is used, i.e.

$$\epsilon := \left\langle ||\Phi(x, t) - P\Phi(x, t)||_2^2 \right\rangle_{\mathbf{T}} \overset{!}{=} \text{ minimal.} \tag{5.25}$$

Here P is the projection operator defined by the reduced orthonormal basis B as

$$P := BB^{\dagger}. \tag{5.26}$$

Instead of minimising the error ϵ which is the time average of

$$\langle \Phi - P\Phi, \Phi - P\Phi \rangle = ||\Phi||_2^2 - 2\,\langle \Phi, P\Phi \rangle + ||P\Phi||_2^2, \tag{5.27}$$

one can maximise the time average of $\langle \Phi, P\Phi \rangle$, i.e. the average projection onto Φ. The brackets denote the scalar product. Therefore one has to maximise the functional

$$\begin{aligned} \text{maximal} \overset{!}{=} & \left\langle \sum_{i=1}^{N} \Phi(x_i, t) \sum_{j=1}^{N} P_{ij} \Phi(x_j, t) \right\rangle_{\mathbf{T}} & (5.28) \\ = & \sum_{i=1}^{N} \sum_{j=1}^{N} \langle \Phi(x_i, t) \Phi(x_j, t) \rangle_{\mathbf{T}} P_{ij} =: c & (5.29) \end{aligned}$$

Here the so called spatial correlation matrix C defined by $C_{ij} = \langle \Phi(x_i, t)\Phi(x_j, t) \rangle_{\mathbf{T}}$ which is a discrete version of the spatial correlation function. It is symmetric and positive semi-definite. After calculating the eigenbasis $\{\phi^i\}_{i=1:N}$ for C one gets

$$c = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{\alpha=1}^{N} \phi_k^i B_{i\alpha} \lambda_k B_{\alpha j}^{\dagger} \phi_k^j, \tag{5.30}$$

where ϕ_k^i denotes the k-th component of the i-th eigenvector of C. Since B and the ϕ^i are orthonormal and the λ_i are positive, Eq.(5.30) is maximised if the eigenvectors ϕ^i for the largest eigenvalues of C are chosen as columns of B.

With this method an optimal basis B can be obtained. However, it is necessary to calculate the correlation matrix C. This is done by simulating the system and evaluating Eq.(5.24) numerically. The system dynamics is then characterised by an ensemble of snap-shots. Accumulating the data in a matrix $D_{ij} = \Phi(x_j, t_i)$ the time-averaging and calculation of the correlation matrix could be written as

$$C = \frac{1}{N_t} D^{\dagger} D, \tag{5.31}$$

where the number of time steps is N_t. If several trajectories are considered also an averaging over these different trajectories has to be performed. Decomposing D via the SVD gives orthogonal U and V and a diagonal Σ with $D = U\Sigma V^{\dagger}$. The eigenbasis of C is V as is obvious from $D^{\dagger}D = V\Sigma U^{\dagger}U\Sigma V^{\dagger} = V\Sigma^2 V^{\dagger}$. In some cases the number of time steps N_t is much smaller than the spatial dimensionality N. Then it is easier to calculate the eigenbasis of $DD^{\dagger}$ which is U. The matrix V can be obtained via $U^{\dagger}D = \Sigma V$. Since Σ is diagonal one only has to normalise the columns of ΣV to get V which is the eigenbasis of $D^{\dagger}D$ and consequently also of the correlation matrix $C = \frac{1}{N_t} D^{\dagger}D$. In the literature this is also known as *method of snapshots* [107].

6. Density Matrix Renormalisation Group

Du mußt verstehn!
Aus Eins mach Zehn,
Und Zwei laß gehn,
Und Drei mach gleich,
So bist du reich.
Verlier die Vier!

Aus Fünf und Sechs,
So sagt die Hex,
Mach Sieben und Acht,
So ist's vollbracht:
Und Neun ist Eins,
Und Zehn ist keins.
Das ist das Hexen-Einmaleins!

Faust I, Johann Wolfgang von Goethe

This section aims to give a very brief overview to density matrix renormalisation group methods. Doing so it is difficult to present the basic ideas without confusing the reader with merely technical details. As DMRG provides numerical techniques, some technicalities are inevitable. However, in many points freedom of choice for a particular scheme exists. One should not confuse what are (to some extent) arbitrary details with what are the essentials of DMRG.

The density matrix renormalisation group (DMRG) was first proposed by Steven R. White in 1991 [122], see e.g. [55] for a review. The aim was an extension of the renormalisation group theory which has been applied successfully to describe the critical behaviour of spin lattice systems by Wilson [125, 126, 13]. One of the simplest examples of such systems, the Ising model with nearest neighbour interaction [22, 13], is described by the following Hamiltonian

$$H = -J \sum_{<ij>} \sigma_i \sigma_j - K \sum_i \sigma_i. \tag{6.1}$$

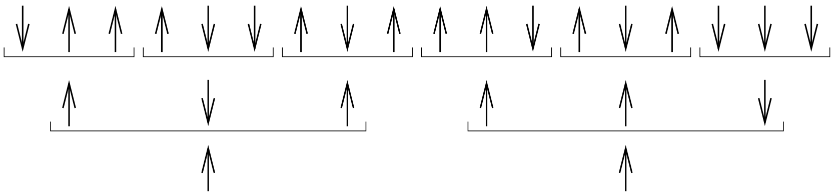


Figure 6.1.: Three block-spin renormalisation steps for a part of a one-dimensional spin lattice.

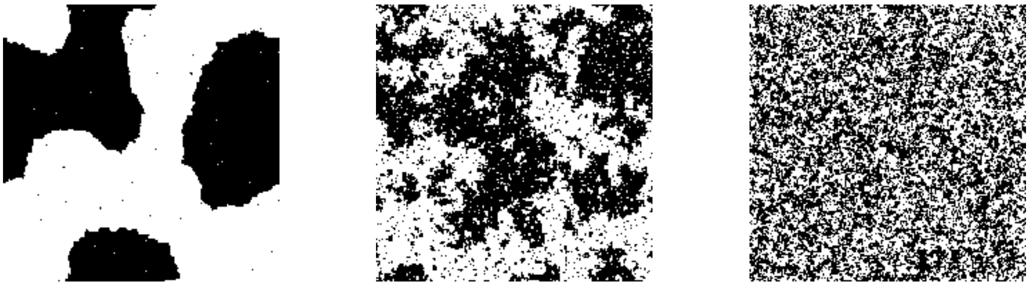


Figure 6.2.: The 2D-Ising model with 200×200 spins and periodic boundary conditions below (left) around (centre) and above the critical temperature. The sample configurations were obtained a standard Monte-Carlo-method with $1000 \cdot 200^2$ updating steps.

The σ_i are the spin variable on the lattice site i, being either $\sigma_i = 1$, i.e. up or down $\sigma_i = -1$. An external field is represented by K while J is the nearest neighbour interaction energy. The renormalisation of the model is performed by combining several spins to a block which is then represented by a single effective spin. For the lattice of the effective spins the model parameters have to be adjusted to represent the same system. This process is termed renormalisation and is also of importance in high energy physics [101, 128, 121, 111]. The new effective model describes the system on a lager scale. For the Ising model this process is drafted in Fig. 6.1. The renormalisation group transform should have the properties of an semi group, i.e.

$$R_{ik} = R_{ij} R_{jk}, \tag{6.2}$$

explaining the term renormalisation group. The success of this approach was in the description of critical properties. One characterisation of critical behaviour is the divergence of the typical length scale within a system. In case of the Ising model this is the divergence of the typical domain size of aligned spins. This is visualised by a typical configuration of the 2D-Ising model below, at and above the critical temperature, as shown in Fig 6.2. Since the renormalisation group transform acts as a 'zooming out' the system at the critical point is invariant under the renormalisation group transform. Consequently a critical point is a fix-point of the renormalisation group transform. The renormalisation group can therefore give information on the critical coupling J_c at which a phase transition occurs.

Despite these successes of the renormalisation group method this Ansatz fails at amazingly simple systems. One example problem is the one-dimensional quantum mechanical particle in a box. The system is described by the Schrödinger equation (setting $\hbar = 1$ and $m = \frac{1}{2}$)

$$i\frac{\partial}{\partial t}\psi(x,t) = \Delta\psi(x,t) = H\phi(x,t) \ , \quad x \in [0,1] \tag{6.3}$$

with Dirichlet boundary conditions (representing impenetrable walls). The energy

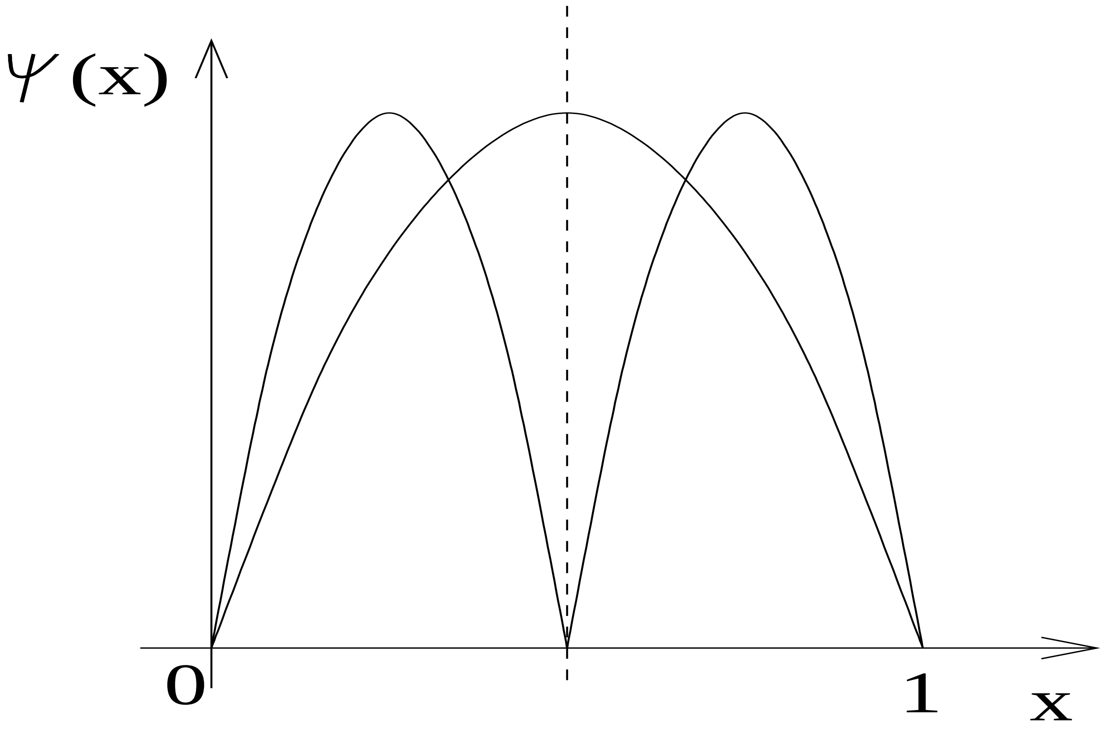


Figure 6.3.: Ground state of the particle in a box (broad maximum) and ground states of the subblocks (narrow maxima).

eigenstates, i.e. the eigenstates of $H = \Delta$ are the Fourier modes $\psi_k(x) = e^{ikx}$ satisfying the eigenvalue equation

$$E_k \psi_k = \Delta \psi_k. \tag{6.4}$$

For a finite box size the eigenvalues of the Laplace operator Δ are discrete. The Fourier modes $\psi_k(x)$ are mutually orthogonal. Expanding Eq.(6.3) in the Fourier modes results in

$$\sum_k \psi_k(x) i \frac{\partial}{\partial t} \alpha_k(t) = \sum_k E_k \alpha_k(t) \psi_k(x) \ , \ \ x \in [0, 1]. \tag{6.5}$$

Thus one easily obtains the time dependence of the coefficients $\alpha_k(t) = e^{-iE_k t}$. The ground state, i.e. the state with lowest energy E_k and the low lying spectrum is often of special interest. This is due to the fact that a small perturbation of the Hamiltonian $H = H_0 + H_1$ can lead effectively to additional transitions between different eigenstates of H. The eigenstates of H are not the eigenstates of H_0 but the system given by H_0 is usually much easier to describe and also a good approximation to H. The transitions could be introduced by the environment of the system defined by H_0. The transitions often lead to stationary distribution e.g. to a Boltzmann distribution

$$P(E_k) \propto e^{-\frac{E_k}{k_B T}}, \tag{6.6}$$

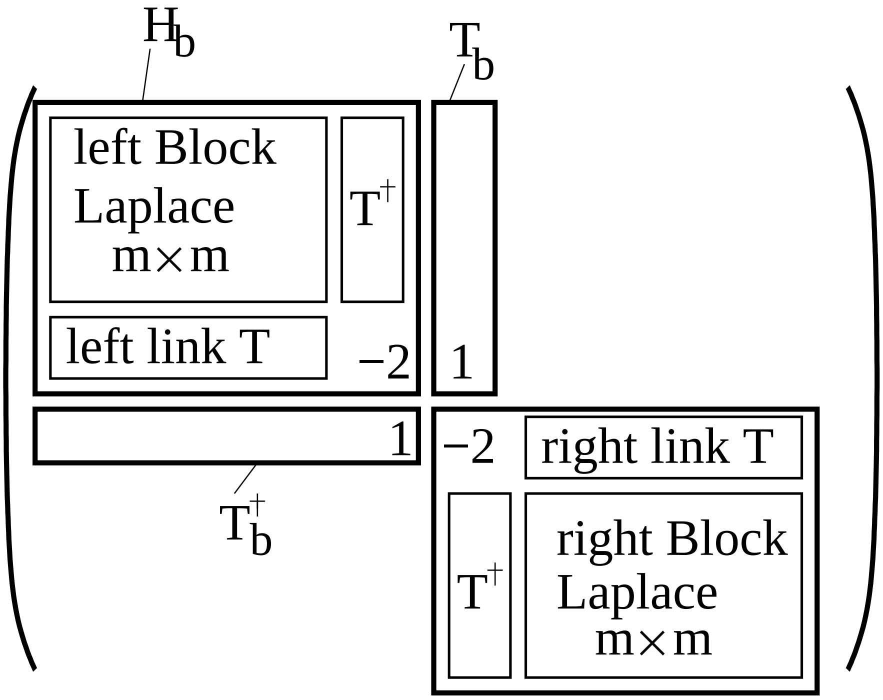


Figure 6.4.: Assembly of the superblock Hamiltonian.

where T is the temperature of the system. The Boltzmann constant k_B defines the relation of energy and temperature units, for SI-units it is $k_B = 1.3806508 \cdot 10^{-23} \frac{J}{K}$ [33], in theoretical descriptions one usually sets $k_B = 1$. From Eq.(6.6) it is obvious that the low lying spectrum is most relevant for the low temperature behaviour. In the limit $T \rightarrow 0$ the ground-state is even the only occupied state. In the following, the states of interest will be denoted as 'target states'. In the following this will be the ground state.

In analogy to the block-spin approach one can construct a larger system from identical sub-systems. However, the state constructed from the ground-states of the sub systems is in general no low energy state for the complete system. In Fig. 6.3 the situation is pictured for the particle in a box showing the ground state of the whole system as well as for the two subblocks. The mismatch accountable for the failure of the renormalisation group approach is obvious. It is also clear that the problem is caused by the inappropriate boundary conditions for the two blocks. Their mutual interaction has been neglected. DMRG removes this constraint and includes this interaction. The above models are local, i.e. each site is only affected by its nearest neighbours. In such cases DMRG is most efficient. The basic ingredient for including block interactions is the superblock concept which will be explained in the following. For numerical treatment, a discretised version of Eq.(6.4) has to be considered. The corresponding techniques are described in Section 3.5, finite differencing is used.

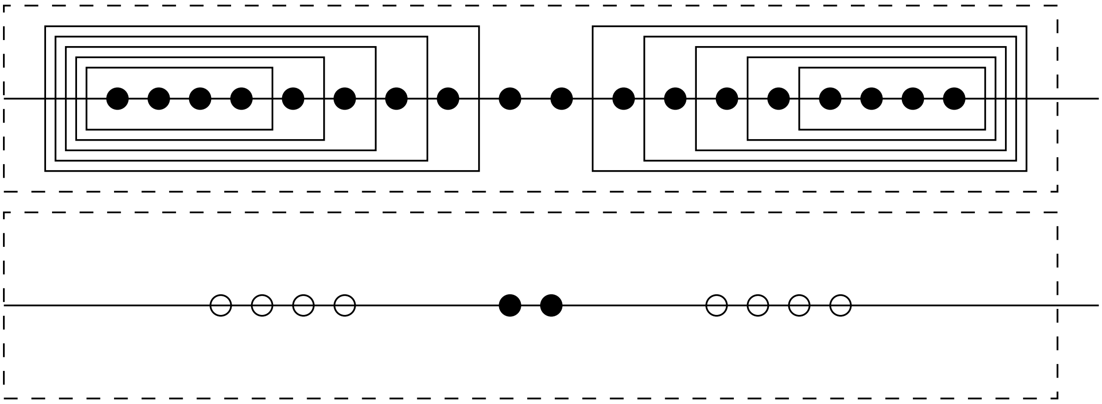


Figure 6.5.: The Russian doll scheme for the superblock. Above the effective sites described by the superblock, below the actual degrees of freedom, contained in the superblock. The open circles do not describe single sites.

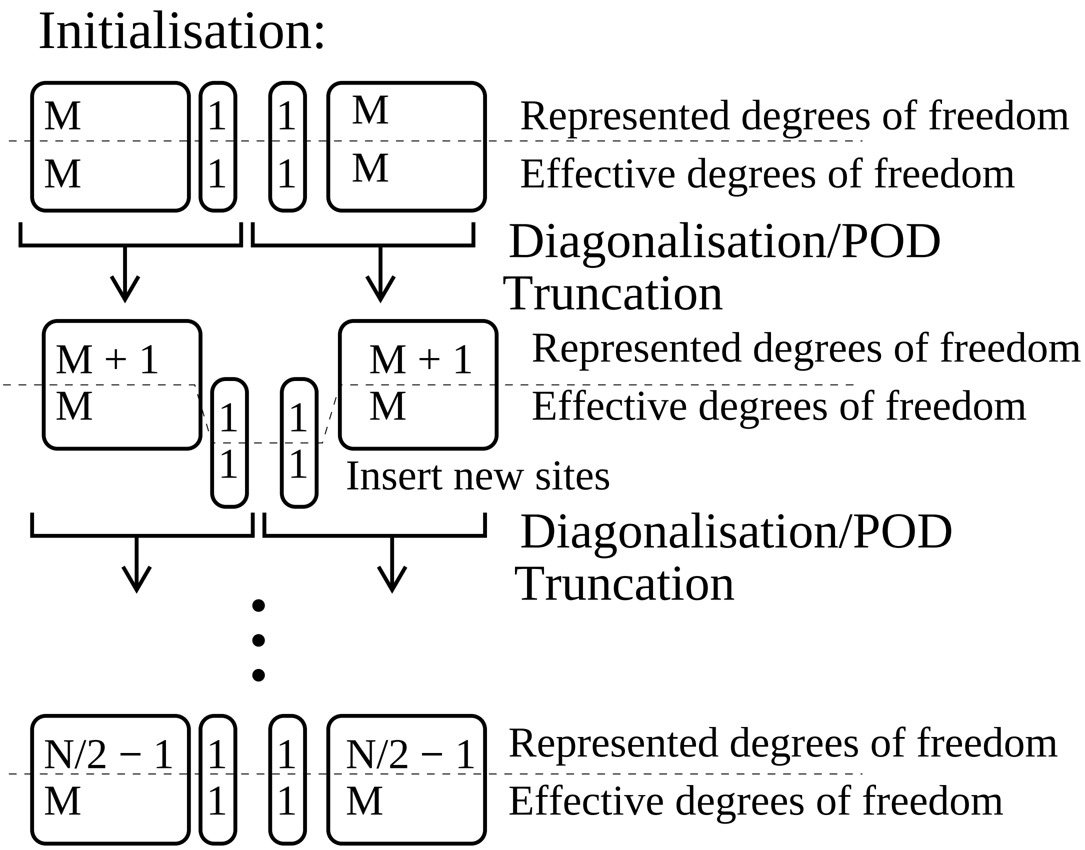

Initialisation:
M
M
1
1
1
1
M
M
Represented degrees of freedom
Effective degrees of freedom
Diagonalisation/POD
Truncation
M + 1
M
1
1
1
1
M + 1
M
Represented degrees of freedom
Effective degrees of freedom
Insert new sites
Diagonalisation/POD
Truncation
N/2 − 1
M
1
1
1
1
N/2 − 1
M
Represented degrees of freedom
Effective degrees of freedom

Figure 6.6.: Graphical illustration of the DMRG initialisation (or warmup) scheme.

6.1. Infinite System Method

The infinite[1] system method increases the effective size of a system successively. The numerical size, determining the size of the vectors and matrices used in the calculation is thereby kept constant. This is achieved by first splitting the system in (typically) two blocks. For each block, the block Hamiltonian H_b and the interaction terms T with its environment are known. Then two additional sites are introduced. They are described by the full systems equations, i.e. Eq.(6.4). The system composed of the two blocks and the additional sites is the superblock. It is still small enough to be treated with conventional methods. The actual assembly of the superblock Hamiltonian (here the discrete Laplace operator) from these data is sketched in Fig. 6.4 for single particle systems. For many body problems the construction involves Kronecker products, but apart from some technicalities the superblock Hamiltonian H_b can be obtained in both cases. For the problem at hand now a diagonalisation of the superblock Hamiltonian $\tilde{H}$ is performed. Now e.g. the ground state is of interest, although other choices are possible. The ground state is then the target state. As in the renormalisation group method above each block is now combined together with its adjacent site to a new block. If the superblock is in the target state ψ_t these new blocks are typically not in a pure state, but described by density matrices [44]. Denoting the inner-block degrees of freedom by α and those of the superblock excluding the block by β, the block density matrix ρ is defined as

$$\rho(\alpha, \alpha') = \sum_{\beta\beta'} \psi_t(\alpha, \beta)\psi_t(\alpha', \beta'). \tag{6.7}$$

This matrix is always symmetric and positive semidefinite. The eigenvalues describe the probability to find the block in the referring eigenstate of ρ.[2] From this interpretation it is obvious that one should use the most probable normalised density matrix eigenstates as columns for the truncation matrix. More quantitatively it can be proven that this choice satisfies an error minimisation criterion [123] similar to the one used in section 5. Generalising to n_t target states a natural choice for the density matrix to be diagonalised would be

$$\rho = \frac{1}{n_t} \sum_{i=1}^{n_t} \rho^i, \tag{6.8}$$

where the ρ^i are the density matrices for the i-th target state. If the blocks initially had contained m degrees of freedom, the m most probable eigenstates of ρ are selected to form columnwise the block truncation matrix R. From this one can construct an effective block which contains still m degrees of freedom but describes a by one site larger block. The effective block Hamiltonian $\tilde{H}_b$ is given

[1] This notation can be found e.g. at [123]

[2] The target states have to be normalised. Then also $Tr(\rho) := \sum_i \rho_{ii} = 1$ holds.

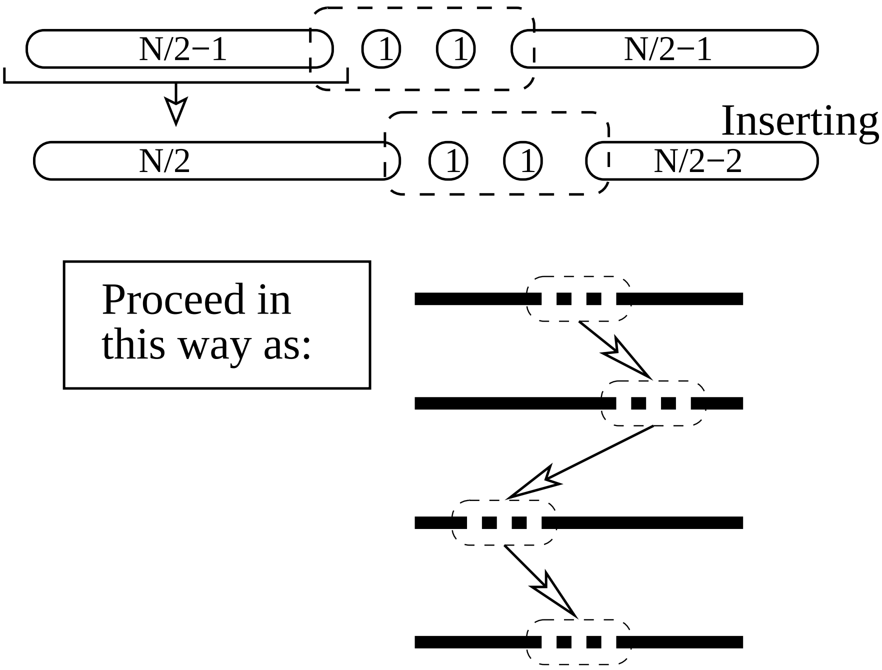


Figure 6.7.: Graphical illustration of the DMRG iteration (or sweeping) scheme.

by

$$\tilde{H}_b = R^{\dagger} H_b R. \tag{6.9}$$

Likewise, the effective interactions $\tilde{T}$ are determined by

$$\tilde{T} = R^{\dagger} T. \tag{6.10}$$

If one is interested in reconstructing the eigenstates instead of just calculating the eigenvalues the reduction matrices R have to be stored as well.

Repeating this procedure leads to a superblock with describes the full system in a Russian doll like scheme, see Fig. 6.5. After a sufficient high number of iterations the effective size of the superblock is eventually large enough to neglect finite size effects. Typically this method is the start of DMRG algorithms where it is used to get a first approximation. A pictorially description of the scheme is given in Fig. 6.6. After several steps, the superblocks describe a large system in which finite size effects become less important.

6.2. Finite System Iteration

While the infinite system method provides a description of increasingly large effective systems one is often interested to get an accurate model for a system of given finite size. Further it should be possible to control the deviation from the correct, also finite description, in a systematical way. This aim is achieved by finite system iterations, in literature also termed finite system sweeps.

In contrast to the infinite system method, the finite system iterations improve already calculated subblocks. This data is usually generated by the infinite system method as indicated before. Again the growth procedure for a subblock is used, increasing the effective block size while keeping the numerical block size constant. However, the effective as well as the numerical size of the superblock is kept constant. This is achieved by applying the subblock growth to only one side of the superblock. The subblock on the other side is replaced by a pre-calculated block with *smaller* effective size. In this way the active region of the superblock is moved through the physical system. Since the inserted sites are always derived from the correct dynamics information is added to adjust the blocks for the finite system. DMRG can be interpreted as a variational method [88, 95] and yields usually results with a high numerical accuracy. These concepts will be applied to non-hermitian systems where numerics are much more demanding. For nonlinear problems even no rigorous results exist in this context.

6.3. Reconstruction of States

DMRG algorithms can be formulated as implicit methods. It is not necessary to store the truncation matrices. The part of the spectrum of interest can still be calculated if only the effective block operators and links are known. The truncation matrices are only required for the reconstruction of the state of the full system. Once a superblock state is given, the expansion to a corresponding full system state is performed by successively multiplying parts of the state vector with the truncation matrices which are rectangular. This is sketched on basis of the subblocks in Fig. 6.8. There the situation for subblocks of equal effective size is shown, although more general partitions can occur. The whole reconstruction from a M-dimensional superblock state to a N-dimensional full system state can also be described as a matrix multiplication with a $N \times M$ matrix having orthonormal columns. For large systems this can be quite inefficient.

It is also possible to revert this process and embed a full system state into the superblock system. Then of course information is lost and a whole subspace of the full phase space leads to the same effective superblock state. This embedding will be required for the DMRG-POD methods in section 10 and 10.

6.4. Single Particle vs. Many Particle DMRG

Single Particle Systems

In quantum mechanics one choice for the phase space of a single particle could be e.g. $L^2([-1.0, 1.0])$ depending on the system at hand, boundary conditions, etc. More generally consider the vector space V, which can be infinite-dimensional, but for numerical considerations V has to be finite-dimensional. The dynamics of the system is determined by a Hermitian linear operator H, i.e. the Hamiltonian

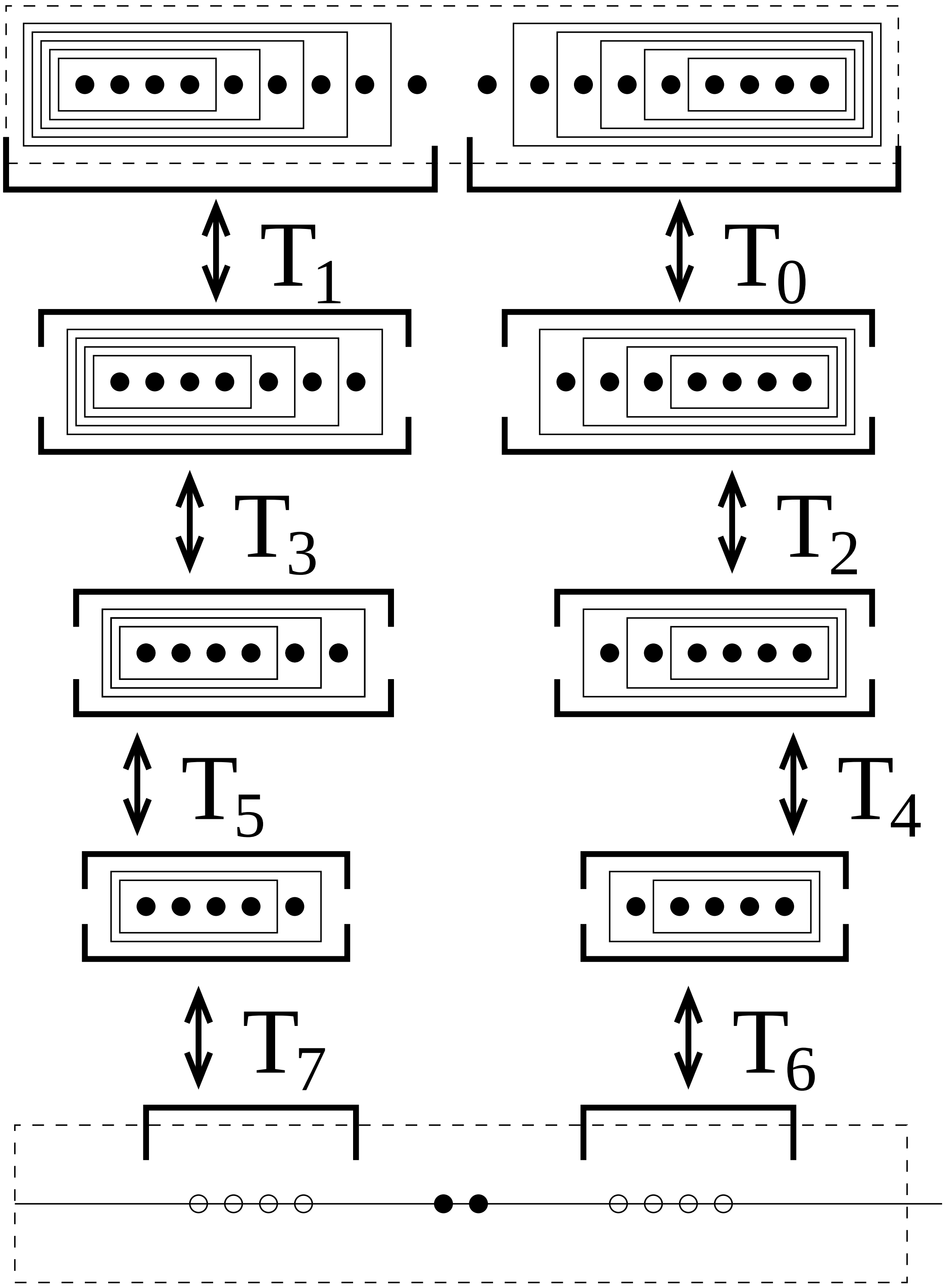


Figure 6.8.: Relation of a superblock state (below) with the corresponding full system state (above) via the truncation matrices. The truncation matrices only act on the particular subblocks. This examples shows the situation for subblocks of equal effective size.

via the Schrödinger equation Eq.(6.3). Discretising the system e.g. by finite differencing to a N-dimensional system each lattice site represents a single degree of freedom. If the system is decomposed into subblocks the phase space of the full system is the direct sum of the phase spaces of the subblocks. This means that if W_i, $i = 1, 2$ are the phase spaces of two subblocks constituting the system with bases B_i, one has $W_1 \cap W_2 = \emptyset$ and the whole phase space is spanned by (B_1, B_2). Consequently, the number of degrees of freedom is proportional to the number of lattice sites within a system.

Due to the property described above, the density matrix e.g.for the first m sites of a larger block of n sites depends non-trivially only upon the first m entries of a target vector. If one has n_t target vectors the relevant data is contained in the $m \times n_t$ matrix D^t consisting of these first m entries of the n_t target vectors. The density matrix ρ would be proportional to $D^t D^{t\dagger}$. Due to this special form only n_t eigenvalues can be non zero since D^t can be maximally of rank n_t.The eigenvectors corresponding to these nonzero eigenvalues all lie in the range of D^t, which is spanned by the columns of D^t. Instead of diagonalising the density matrix ρ one can likewise orthonormalise D^t thus also obtaining an orthonormal basis for the relevant subspace.

An operator defined for a subblock can be extended to the full system simply by defining the additional matrix elements to be δ_{ij}. The superblock operator e.g. can be directly obtained via the insertion scheme already sketched above.

Many Body Systems

For many particle systems the phase space is still a - possibly infinite-dimensional - vector space V. The system will be described on the basis of single particle states. But in contrast to the single particle problem now also an entanglement between the particles can occur. Given two single particle states ψ_1, ψ_2, an entangled state would be e.g. $\Psi = \frac{1}{\sqrt{2}}(\psi_1^1\psi_2^2 + \psi_2^1\psi_1^2)$. Here ψ_i^j denotes the j-th particle to be in the i-th state. The state Ψ is already of a product form which is used in the Hartree and Hartree-Fock Ansatz. Since particles are usually indistinguishable physical states have to be either symmetric (Bosons) or antisymmetric (Fermions) under exchange of two particles.[3] The latter case can be described by the so called Slater determinant, which is the determinant of the matrix where the (i, j)-th entry is the i-th single particle state of the j-th particle. The definition of the determinant guarantees the desired properties. However, these methods will not be detailed further, but the structure of the phase space composed of two subblocks W_1, W_2 is relevant for us. In contrast to the single particle systems this phase space is now the tensor product $V = W_1 \otimes W_2$. If $\{b_j^i\}_{j=1:N}$ is the basis of W_i the whole phase space V is spanned by $\{b_i^1 b_j^1\}_{i,j=1:N}$ which is thus N^2-dimensional.

Obtaining full system operators from operators defined on subblocks now involves the Kronecker product. Given the splitting above an operator A_1 defined

[3] I.e. exchanging two particles only leads to an additional factor of $+1$ or -1, respectively.

on W_1 can be extended to V simply by $\mathrm{kron}(A_1, \mathbb{1}_N)$, where $\mathbb{1}_N$ is the identity on W_2. The construction scheme is thus also very clear.

7. Proposed Methods

In the following Chapter I introduce the methods that were devised during the work on this thesis. Consequently the current Chapter can be viewed as the most important part of the thesis. The new methods are presented in a separate Chapter to outline them from already existing approaches. In particular the new work comprises the Schur variant of the non-symmetric many-body DMRG, the proper orthogonal decomposition DMRG (POD-DMRG) and the variational proper orthogonal decomposition. The corresponding applications are presented in the next three Chapters. There, also the models that will be studied are introduced. They serve as a sort of testing ground for the methods. These models have merely exemplary character, as the new methods can also be applied to other problems.[1]

7.1. Real Schur DMRG

The aim of this algorithm is to find a few ordered Schur vectors for a dynamical system determined by a master equation, see Eq.(4.3), with non-hermitian master operator $\mathcal{M}$. The master operator is extremely high-dimensional due to the particular stochastic approach, see Chapter 4. For one-dimensional systems DMRG methods are known to be effective in calculating a small set of target vectors even for systems with a high-dimensional phase space. The target vectors are usually some of the eigenvectors, but different choices are also possible, e.g. in finite temperature DMRG [25, 85].

Calculations for non-symmetric matrices are often much more complicated and numerically demanding than for symmetric matrices [50]. This is especially true for very large systems. While DMRG has already been applied to calculate the steady state for such systems the calculations of transient states have led to problems [27]. Reasons are the spurious emergence of non-vanishing imaginary parts due to finite numerical precision and the non-orthogonality of the eigenstates.

For a systematical model reduction, which is the physical idea behind these [27, 38] calculations, the Schur vectors of a real ordered Schur decomposition are much better suited. Therefore I propose a many-body DMRG method that uses these Schur vectors as target vectors. To this end the calculation and ordering of a real Schur decomposition and a management of the target vectors considering their occurrence in pairs for complex conjugated eigenvalues are necessary.

[1] All algorithms were programmed in C++ using public accessible libraries and tools. The classes for vectors and matrices were provided by Javier Rodríguez Laguna [99] unless the gsl implementations were used.

7.1.1. Technical Implementation

The physical system is a one-dimensional chain of N lattice sites each containing n degrees of freedom, see Chapter 4. This means that each site can assume n different states and the number of different states for the complete system is n^N. Consequently the phase space is also of dimension n^N for the master equation approach.[2] The master operator $\mathcal{M}$ acts on this space, so in matrix representation this would result in a $n^N \times n^N$ matrix which can be considered only for very small systems. Thus the master operator is given in terms of single site operators. Technically this is implemented by a formated string which describes the action of $\mathcal{M}$ upon a state. A parser has been programmed to interpret this format.[3] Only nearest neigbour sites give rise to interaction terms, but due to the stochastic descripion the master operator has no simple band structure.

For the blocks a class was defined containing all relevant block informations. Further, a superblock class was defined to administrate the assembly of a superblock from blocks. The superblock scheme with two additional sites in each initialisation/iteration step was chosen. Denoting the number of sites within a subblock with m the superblock has dimensionality

$$n^m \cdot n \cdot n \cdot n^m = n^{2m+2}. \tag{7.1}$$

Initialising the first pair of subblocks from scratch with information on the full master operator the initialisation of the other blocks is done in a standard warm-up scheme. The necessary calculation steps are divided into appropriate functions.

The calculation of the target states is done by first constructing the explicit matrix representation of the superblock operator. Then a real ordered Schur decomposition is performed by a gsl routine [53, 47]. The ordering of the Schur form is done in the way described in Appendix C. The number of target states is n_s. Due to numerical inaccuracies it is possible that this choice separates a conjugated pair of eigenvectors. To avoid this the number of target states kept is adaptive either n_s or $n_s + 1$.

The truncation of a block is performed by calculating the reduced density matrices ρ^i for each target state and forming an average density matrices $\rho = \frac{1}{\hat{n_s}} \sum_{i=1}^{\hat{n_s}} \rho^i$ where $\hat{n_s} = n_s$ or $\hat{n_s} = n_s + 1$. Diagonalisation of ρ gives the desired truncation matrix as in standard DMRG methods.

For the iteration or sweeps basically the same actions are necessary. The truncation matrices are stored, since we are interested in a reconstruction of the full system state. Principally it is possible to extract expectation values of observables via the summation state which is the left eigenstate to the eigenvalue zero. However, this state is not kept and expectation values of observables are evaluated in the canonical way.

[2] Considering the normalisation condition $\sum_i \Psi_i = 1$ for a state vector Ψ this reduces to a phase space of dimension $n^N - 1$.

[3] This procedure was proposed by Javier Rodríguez Laguna [99, 38].

After the initialisation and the desired number of sweeps the n_s target states are reconstructed.

7.2. Proper Orthogonal Decomposition DMRG

The calculation of a proper orthogonal decomposition inevitably requires the simulation of the complete system at hand. For linear systems the optimal modes are known from analytical considerations. These modes can also be calculated by single particle DMRG methods for one-dimensional systems.

The aim of this ansatz is to circumvent the necessity for a simulation of the complete system for calculating a POD. This is achieved by applying a blocking scheme similar to DMRG. Physically a blocking should make sense as adjacent regions in a spatially one-dimensional system can only interact at their interfaces. With the new approach the POD for a general[4] nonlinear system can be calculated. Considering also the reduction in computational effort due to the reduced size of the correlation matrix which has to be diagonalised, the advantage of the method is even more significant although other methods as the method of snapshots, see Section 5.2.7, can reduce this benefit.

As model systems the diffusion equation (for pedagogical reasons), the Burgers equation and a nonlinear diffusion equation were chosen. The numerical simulation of this systems is comparatively simple and requires no involved algorithms. Also the system size was chosen relatively low since large systems had not led to qualitative different results. The application of the POD-DMRG method to these model systems is presented in Chapter 9.

7.2.1. Technical Implementation

Due to the close analogy of the new method to the single particle DMRG, which is also the simplest of the DMRG applications, the algorithm is explained on this footing. Three modifications are necessary to obtain the POD-DMRG method from single particle standard DMRG presented in Chapter 6. The required modifications are:

First, instead of a diagonalisation of the superblock operator, a POD on the superblock system has to be performed. This is composed of first, a simulation of the superblock system, as defined in Eq.(5.13). Then the superblock correlation matrix from the generated data has to be diagonalised. This gives an orthonormal set of vectors which are the target vectors in the context of DMRG.

[4] In fact the nonlinear terms are restricted to a finite power of the state variables. This is a problem for nonlinearities e.g. of the form of $\ln(\Phi)$. For most physical relevant systems this is effectively no restriction.

Second, to each subblock there exists not only a linear sub-block operator but also higher order operators, given by third and higher order tensors, see Eq.s(5.9,5.10,5.11,5.12). These have to be updated in a similar way.

Third, for the POD the initial states for the sample trajectories are a crucial point. The initial states are defined for the full system. They have to be projected onto the superblock system which requires all truncation matrices explicitely.

Concerning the first point, this is no great difference, since the POD (simulation and diagonalisation of the correlation matrix) returns also an orthonormal set of 'relevant' states (POD modes) that serve as target states, as described above.

Beside the linear operator (L in Eq.(5.2)) which is assembled identically as the superblock operator in single particle DMRG, the higher order operators have to be assembled as well. This is principally possible, but complex. Here a simple trick is used. For all models systems it is sufficient to know the component-wise squaring operator $\Omega_{i,j,k} := \delta_{ij}\delta_{ik}$. (And in some cases the derivative operator which is linear and is also assembled like the superblock Laplace operator.) Ω is purely diagonal, so no links have to be stored and assembled. The reduction with a truncation matrix R is straightforward:

$$\hat{\Omega}_{i,j,k} = \sum_{a,b,c} R_{i,a}\Omega_{a,b,c}R_{b,j}R_{c,k}. \tag{7.2}$$

From this the higher order tensors can be calculated directly, e.g. for the Burgers equation ([24], for more details, see section 9.)

$$\frac{\partial}{\partial t}\Phi = d\Delta\Phi + \nu(\Phi\nabla)\Phi, \tag{7.3}$$

one obtains the following quadratic terms

$$\begin{aligned}\hat{Q}_{\text{Burgers},i,j,k} &:= \nu\sum_{l}\delta_{ij}\delta_{lj}\hat{D}_{x,N,l,k}\\ &= \nu\sum_{l}\hat{\Omega}_{j,i,l}\hat{D}_{x,N,l,k}. \end{aligned}\tag{7.4}$$

This the one-dimensional analogon to the convective derivative, see Section 9.2 for details. For fourth and higher order operators this procedure is a bit memory consuming. E.g. for calculating Φ^3 it is more efficient to calculate first $\Phi_{\text{tmp}} := \hat{\Phi}^2 = \Omega\Phi\Phi$ and then $\hat{\Phi}^3 = \hat{\Phi}^2\hat{\Phi} = \Omega\Phi_{\text{tmp}}\Phi$. Note, that only nonlinearities of a finite power of the state variables can be treated in this way which is fully sufficient for most problems.

The third point may be a small disadvantage, since the projection operators have all to be stored, which is not necessary in DMRG if only the energy values are of interest. However, here as well as in DMRG it is possible to expand a superblock state to a state of the original system as well as project down a system state to the superblock if all truncation matrices are stored. The down-projection of the N-dimensional state is in particular done by iteratively contracting the $m+1$ outermost sites of e.g. Φ with the corresponding block truncation matrix R. Apart from the memory requirement this is simply a book keeping problem.

It should be noted that only $m+1$ most relevant states from the POD are used as target states. Thus only $m+1$ relevant states of the superblock are optimised although it represents $2m+2$ degrees of freedom. This has to be considered when comparing the results in Chapter 9. However, the POD-DMRG is nevertheless faster than the full POD, see Section 9.4.

To summarise: Apart from the POD itself, which is a standard technique, no fundamental changes have to be implemented to get a POD-DMRG method from the simple toy model DMRG. The assembly of linear operators has to be performed in any case, only the new method requires several operators. The assembly of the Ω operator is even simpler, since all links vanish. The reconstruction of full system states is also possible in DMRG. In contrast to standard DMRG it is mandatory for the method presented here in order to evaluate the correct initial conditions.

7.3. General Variational Method for Proper Orthogonal Decomposition

The POD-DMRG method of the previous section is restricted to spatially one-dimensional systems by construction. For physical applications this is a severe restriction. This problem is typical for DMRG applications and no complete solution to it is known until now. However, some approaches to higher spatial dimensions exist for quantum mechanics [87]. I extend the POD-DMRG method in a similar way to higher-dimensional systems. The resulting ansatz is best described as variational POD method. It is also very general and even conceptually simple. For the numbers of spatial dimensions no principial limit exists and also an application to higher order finite element methods should be possible.

Since calculations on higher-dimensional systems and their evaluation are much more demanding I choose a relatively simple model system, the two-dimensional, incompressible Navier-Stokes equations. The physics of this model is still far from being trivial and also numerical it is the most complicated system studied in this thesis. The results are presented in Chapter 10.

7.3.1. Technical Implementation

For the N-dimensional phase space a reduced basis of dimension M is searched for by a variational method. First, an ansatz basis B^0 is chosen. Principally, this could be a random but orthonormal basis. This is inefficient but works e.g. for the diffusion equation. For the Navier-Stokes equations instabilities arise, so one would start typically with Fourier modes providing a low wavenumber. The

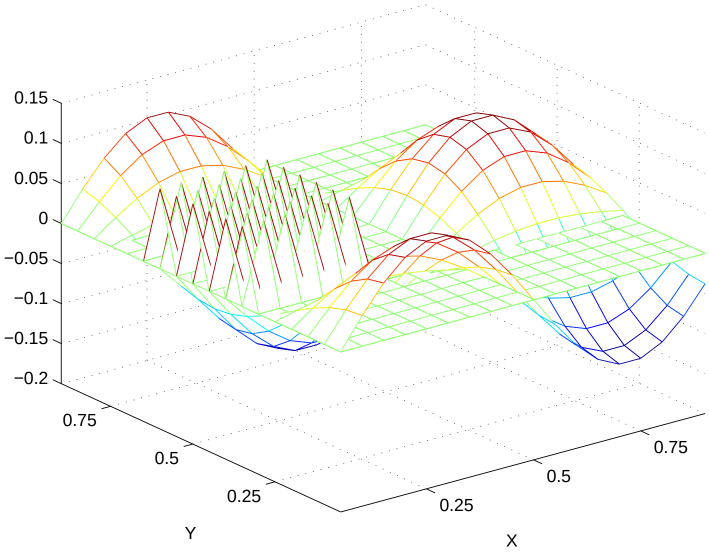


Figure 7.1.: Illustration of one low wavenumber Fourier mode and a set of delta states that make up one particular choice of B_{new}. The delta states are not shown normalised here.

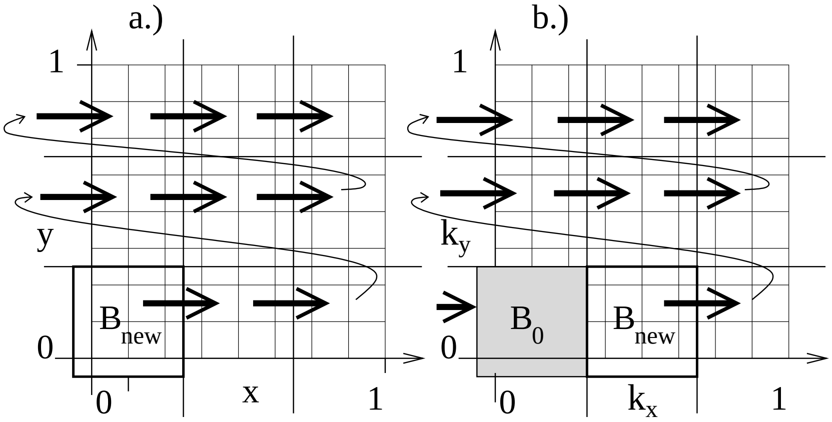


Figure 7.2.: Scheme for choosing the inserted basis B_{new}. For the real space method a.) choosing delta functions, one for each grid node in the current patch. For the spectral variant in Fourier space b.). Since then B^0 is initialised with the lowest wave number vectors, starting with an adjacent patch is necessary.

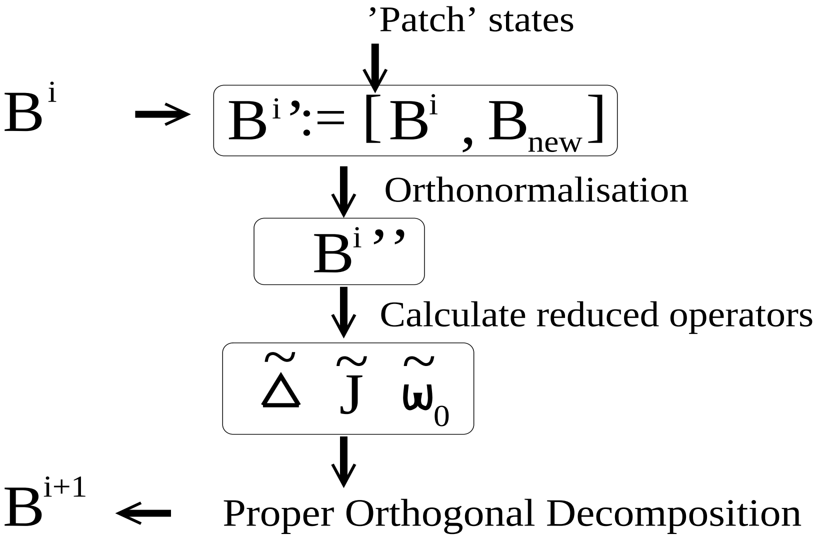


Figure 7.3.: Flow chart diagram for a single step in the variational method. An iteration step consists of several sub steps, until the inserted B_{new} together span the whole phase space. Here the reduced entities are - exemplarily for the Navier Stokes equations - the reduced Laplace operator $\tilde{\Delta}$, the reduced Jacobi operator $\hat{\mathcal{J}}$ and the reduced initial condition $\tilde{\omega}_0$. The choice of these reduced entities has to be adapted to the equation of interest, if necessary.

ansatz basis B^0 is then extended by a test basis B_{new}, which should be linearly independent of B^0. In the work of Delgado et al. [87] delta states for a particular 'patch' region in the physical space are chosen but this is not mandatory. As an example a set of delta states together with a Fourier mode is represented in Fig. 7.1 graphically. The resulting basis $B^{0\prime} := [B_0, B_{new}]$ has size $N \times (M + M_{patch})$ and full range. Via an orthonormalisation procedure, e.g. Gram-Schmidt [50], the $N \times (M + M_{patch})$ orthonormal matrix $B^{0\prime\prime}$ is obtained. The effective system is now determined by $B^{0\prime\prime}$ via Eq.(5.9) - Eq.(5.12). For the following iterarations the construction of the corresponding matrices $B^{i\prime}, B^{i\prime\prime}$ (for the i-th iteration) is similar. As described in Section 5.2.7 an orthonormal POD basis of the effective system $\tilde{B}_{POD}$ is obtained. This in turn is used to calculate the new, improved ansatz basis

$$B^{i+1} = B^{i\prime\prime\dagger}\tilde{B}_{POD}. \tag{7.5}$$

The basic step described above is now repeated with different choices for B_{new}. In the case when B_{new} is composed of delta states one typically moves the 'patch' through the physical space. This is also done in [87] and is depicted in Fig. 7.2a. A single iteration step is completed when the full system has been covered by the patch, or more generally, when all used matrices B_{new} (from all steps of the iteration) span the whole phase space. To improve the reduction several iterations can be performed. In Fig. 7.3 a flow chart of the above proposed method is presented.

7.3.2. Spectral variant

The choice of delta states for B_{new} in the method above results in a local inhomogeneous description of the field evolution. This is in particular problematic for the study of the incompressible Navier Stokes equations as it enforces the occurrence of instabilities. A smoother approximation is obtained if one chooses B_{new} to be composed of Fourier modes instead. The 'patching' occurs then in Fourier space, see Fig. 7.2b. Since B^i and B_{new} are required to be linearly independent, one has to choose the first initialisation accordingly.

8. Microscopic Models

In this chapter I apply the Schur DMRG method, presented in Section 7.1, to some simple one-dimensional models that are described by a microscopic dynamics. Technically these are modelled via the master equation as described in Chapter 4. The focus here is set on a systematic treatment of in principle nonlinear models. By the conversion to a high-dimensional linear model one can rely on the known results for such systems and need not to treat the nonlinearity approximatively. All models in this section are stochastic already by construction.

For the study of stochastic non-equilibrium systems DMRG has been previously employed [28, 38, 27]. There the focus was on the steady state. The calculation of the long living transient states have proved to be a numerically demanding task. In the following the sorted Schur vectors will be calculated, which give also a description for the long time transient behaviour, but are more appropriate for a description as explained in Chapter 5.

8.1. Reaction Diffusion System

This model has been studied previously e.g. by [1, 27, 38] and is also termed the pair annihilation process. This model can also be found in [106], Section 9.5. In the literature it has been used as a simple example for a non-equilibrium process. The physical space is a one-dimensional lattice, whose sites are either free characterised by 0 or occupied 1. The dynamics is determined by three processes

Annihilation

$$11 \to 00 \quad \text{with rate } K_a \text{ for nearest neighbours,} \tag{8.1}$$

Diffusion

$$10 \leftrightarrow 01 \quad \text{with rate } K_d \text{ for nearest neighbours,} \tag{8.2}$$

Source term

$$0 \to 1 \quad \text{with rate } K_s. \tag{8.3}$$

The source process occurs only at the first site. This results in a system where particles are inserted at one boundary of the spatial domain, can diffuse around and can annihilate pairwise. Under this special conditions the model was investigated e.g. in [38]. Although a mean field approximation for the time evolution of the average density is given in [106] the model has no direct correspondence

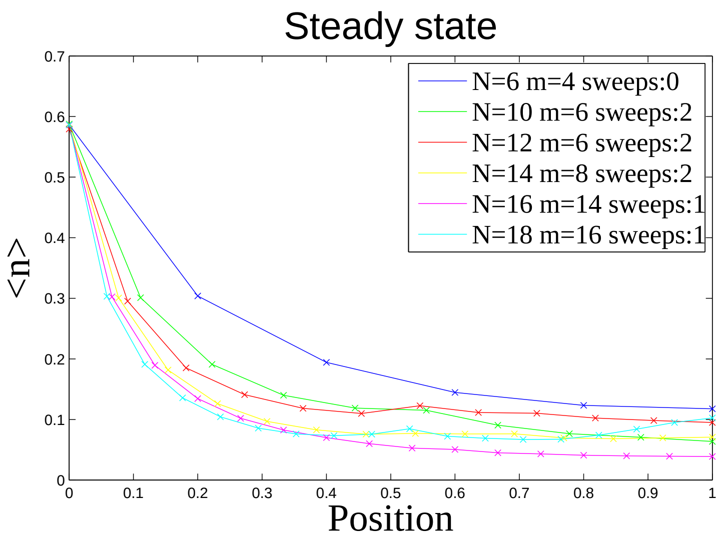


Figure 8.1.: Rescaled average occupation number for various lattice sizes in the steady state of the reaction diffusion system. The state was normalised according to Eq.(4.2), i.e. $\sum \Psi_i = 1$.

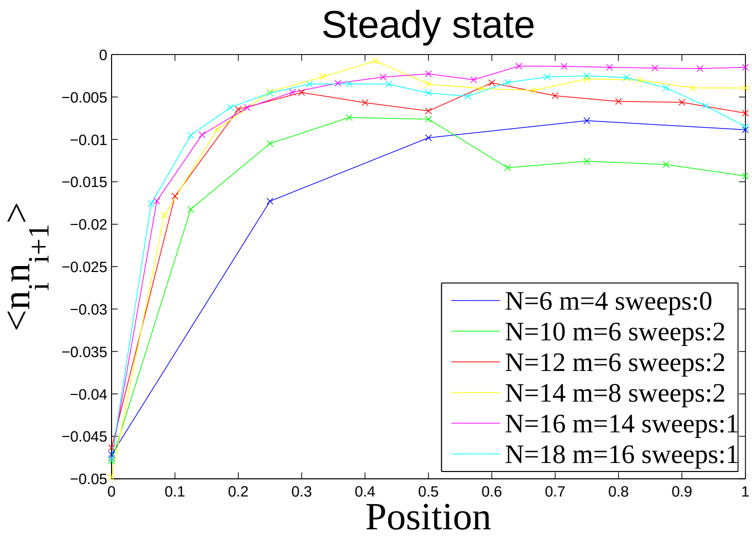


Figure 8.2.: Rescaled nearest neighbour density correlation for various lattice sizes in the steady state of the reaction diffusion system. The state was normalised according to Eq.(4.2), i.e. $\sum \Psi_i = 1$.

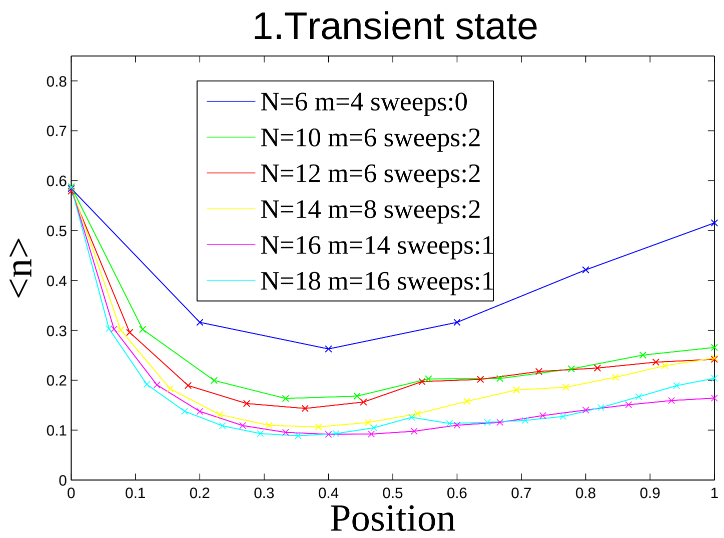


Figure 8.3.: Rescaled average occupation number for various lattice sizes in the first transient state of the reaction diffusion system. The state was normalised according to Eq.(4.2), i.e. $\sum \Psi_i = 1$.

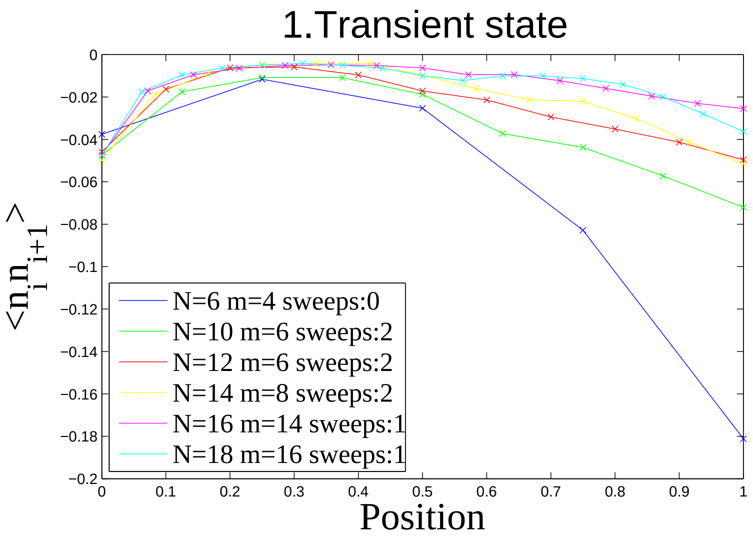


Figure 8.4.: Rescaled nearest neighbour density correlation for various lattice sizes in the first transient state of the reaction diffusion system. The state was normalised according to Eq.(4.2), i.e. $\sum \Psi_i = 1$.

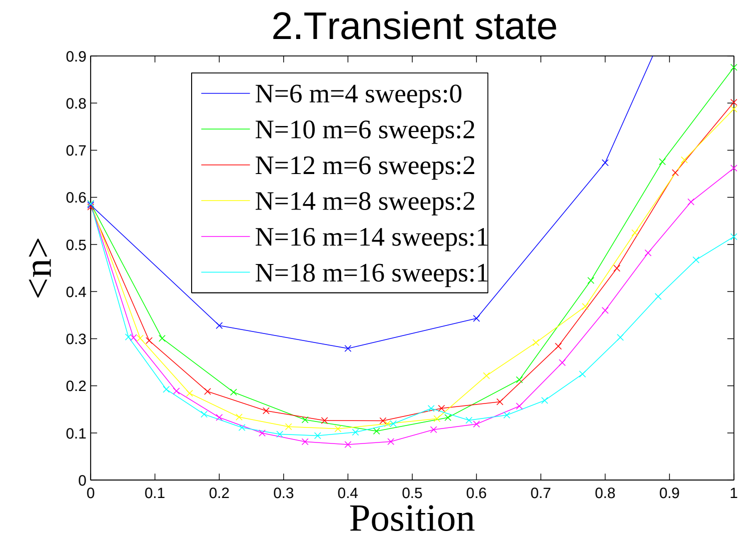


Figure 8.5.: Rescaled average occupation number for various lattice sizes in the second transient state of the reaction diffusion system. The state was normalised according to Eq.(4.2), i.e. $\sum \Psi_i = 1$.

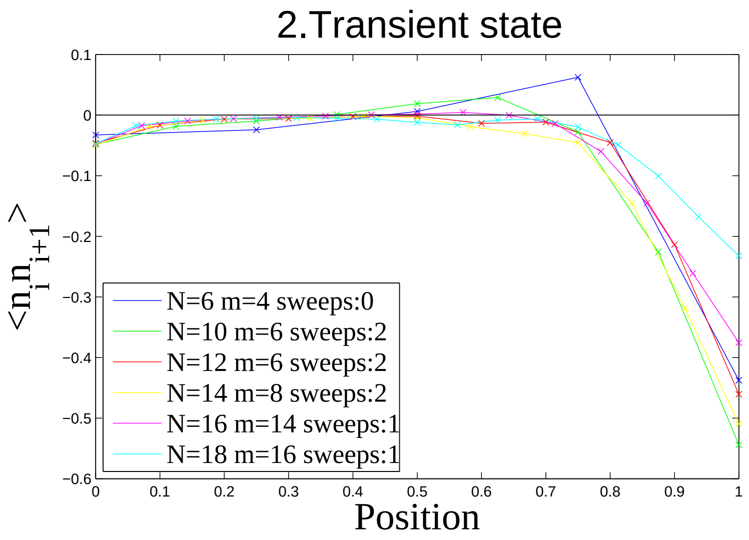


Figure 8.6.: Rescaled nearest neighbour density correlation for various lattice sizes in the second transient state of the reaction diffusion system. The state was normalised according to Eq.(4.2), i.e. $\sum \Psi_i = 1$.

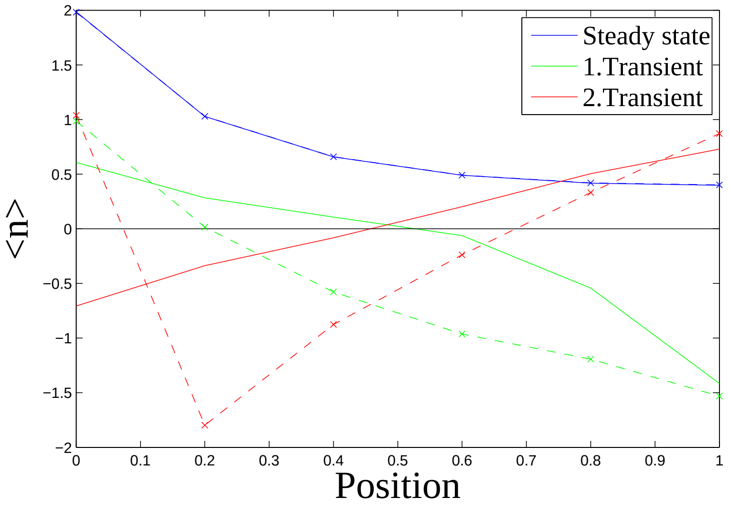


Figure 8.7.: Rescaled average occupation number for the reaction diffusion system and the 6-site lattice in the first eigenstates. Solid lines: Results from the Schur-DMRG method, dashed lines: POD modes. The curves for the steady state lie on top of each other. Note that the transient states can be interpreted as corrections to the steady state. Therefore negative values are permitted.

to a PDE. By mapping the model onto a quantum chain analytical results were obtained in [1].

Without the source term the system tends to an empty lattice for large times for all initial conditions [27]. It is known that for low spatial dimensions the diffusion is efficient for mixing [1].

8.1.1. Numerical Results

The first three Schur vectors are calculated for the parameters $K_s = 1$, $K_a = 1$ and $K_d = 1$. To visualise the results, the average density profile and the nearest neighbour density correlation are evaluated.

Normalisation of the Results

For the reaction diffusion model the normalisation condition

$$\sum_i \Psi_i = 1, \tag{8.4}$$

is adopted for all states Ψ. This choice accounts for the probabilistic interpretation of the state vector Ψ. The transient states can be considered as corrections to the steady state so that these states can have negative components. In principle then also the norm $\sum_i \Psi_i$ can be zero. However, this feature is not present in the calculations on the reaction diffusion model so one can use the normalisation Eq.(8.4). Nevertheless the situation $\sum_i \Psi_i = 0$ will be encountered in later sections.

Average Density Profile

The density profiles for lattice sizes from 6 to 18 for the steady state are shown in Fig 8.1. The corresponding phase spaces were 64 to 262144-dimensional. The profile shows fast decrease near the source site, due to the annihilation and a long tail. The profiles also show signs for numerical inaccuracies. For the system of size 6 (with $m = 4$) the Schur-DMRG procedure is equivalent to a direct real Schur decomposition. The same data for the first transient state is presented in Fig. 8.3. Note that while the Schur vectors are orthogonal, the density profiles for different states are not. Comparing Fig. 8.1 and Fig. 8.3 one sees that the long time corrections are most important for the region which has a low average occupation. The second transient state gives a very similar correction to the density, although all Schur vectors are mutually orthogonal. For larger system sizes also the correction to the steady state are smaller. The correct eigenstates are derived from the Schur vectors by diagonalising the effective master operator M as

$$B^{\dagger}\mathcal{M}B\mathcal{V} = \mathcal{V}M, \tag{8.5}$$

where B contains the Schur vectors and $\mathcal{V}$ is a matrix with normalised columns. The entries for one column of $\mathcal{V}$ are the expansion coefficients for the eigenstates in the Schur vectors. The resulting density profiles are shown in Fig. 8.7.

To evaluate the results a direct simulation of the model has been performed. In order to calculate an average density, an ensemble of 2000 random (uniformly distributed) initialised states were evolved under the stochastic time evolution for 10^6 time steps. For the resulting time dependent density profile a proper orthogonal decomposition was performed. The results are shown in Fig. 8.7 as dashed lines. The agreement for the steady state is excellent. For the transient states this is clearly not the case. This is due to the fact, that the density profiles of the Schur vectors as well as of the eigenvectors are not orthogonal. On the other hand, the POD-modes are by construction orthonormal. The density profile further does not contain all information on the stochastic process. Therefore the failure of this comparison does not question the Schur DMRG results.

Nearest Neighbour Density Correlation

The nearest neighbour density correlation c_i is defined by

$$c_i =< n_i n_{i+1} > - < n_i >< n_{i+1} > \ , \ \ i = 1 : N - 1. \tag{8.6}$$

This function has been evaluated for lattice sizes from 6 to 18. For the steady state one observes a negative correlation for all positions, see Fig. 8.2. The absolute

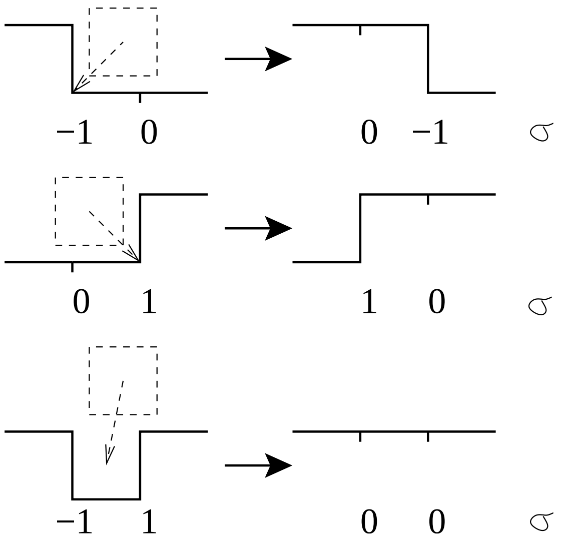


Figure 8.8.: Dynamics for the deposition model in the surface step picture.

value decreases rapidly with increasing distance to the source site. This is in analogy to the decrease in the average density. Annihilation should lead to a negative correlation. For lower average density the influence of the annihilation process also decreases. The corresponding results are shown in Fig. 8.4 for the first transient state. Again the correlations are negative and the behaviour of the absolute value is very similar to the average density. For the second transient state the nearest neighbour density correlation becomes positive for intermediate distances from the source site. At least for the $N = 6$ lattice this is unlikely due to numerical inaccuracies since here the Schur DMRG method is equivalent to a direct Schur decomposition. At the boundary without source the nearest neighbour density correlation becomes negative with comparatively large absolute value. This is also in agreement with the average density. In all cases the absolute value of the correlation decreases with increasing lattice size. The effect is small directly at the source site and increases with the distance.

8.2. Surface Deposition Model

Continuous Equation

The Kardar Parisi Zhang (KPZ) equation is one possible model for surface growth [66]. In [71] a generalisation of the form

$$\frac{\partial}{\partial t} h(x,t) = d\Delta h(x,t) - \nu |\nabla h(x,t)|^{\beta} + \xi(x,t) \tag{8.7}$$

was considered. Here $h(x,t)$ is the surface profile, d is the diffusion constant and ν determines the strength of the nonlinear term. The last term $\xi(x,t)$ is a Gaussian noise with zero mean and variance $\sqrt{<\xi^2>}$. The original KPZ equation is

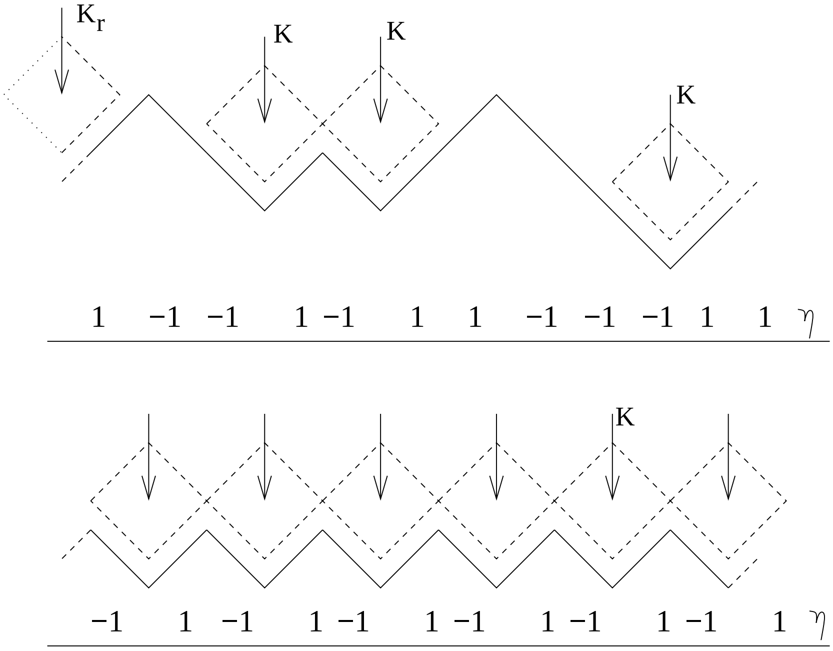

Kr
K
K
K
1 −1 −1 1 −1 1 1 −1 −1 −1 1 1 η
K
−1 1 −1 1 −1 1 −1 1 −1 1 −1 1 η

Figure 8.9.: Illustration of one particular state of the deposition model, together with the updating step. Below the same picture for the steady state. More exactly the steady state is a superposition of the pictured stated and its updated version.

recovered for $\beta = 2$. For $\sqrt{< \xi^2 >} \rightarrow 0$ one obtains the deterministic version. Note that Eq.(8.7) is invariant under translations. The case $\beta = 1$ will be considered which is also discussed in [71]. There, periodic boundary conditions were applied. Free boundary conditions are employed in the following. For long times Eq.(8.7) leads to a steady growth and the surface profile becomes flat (on average for the stochastic version). Discretisations of the KPZ-equations show numerical instabilities [34] that are not present in the continuous description which can be mapped onto the diffusion equation by the Hopf-Cole transformation [56] and can be solved exactly.

Microscopic Model

The microscopic models described here are also derived in [71]. The connection to the continuous model is merely qualitative. This is no limitation since in practice microscopic models can be much more realistic than continuous models.

σ-Model

To derive the model for the master equation approach, the space is discretised. Instead of the height profile h_i itself the surface steps

$$\sigma_i := h_i - h_{i-1}, \tag{8.8}$$

are considered. Further, the steps are restricted to $\sigma_i = 0, \pm 1$. The dynamics is defined as follows. Particle can adsorb only at the corners of an existing surface step. This gives for the surface height the following updating rule

$$h_i(t+1) = \max(h_{i-1}(t), h_i(t), h_{i+1}(t)). \tag{8.9}$$

Within this model the surface heights are integer numbers which allows principally for an infinite number of possible values for each lattice site. This is significantly reduced by considering the surface steps from Eq.(8.8). Their dynamics is given by the simple rules

$$-10 \rightarrow 0-1 \quad \text{with rate } K, \tag{8.10}$$
$$01 \rightarrow 10 \quad \text{with rate } K, \tag{8.11}$$
$$-11 \rightarrow 00 \quad \text{with rate } K, \tag{8.12}$$

i.e. the single site state 1 denotes a particle which can diffuse to the left, while the state -1 describes its anti-particle which can diffuse to the right. These processes are depicted in Fig. 8.8. Also annihilation is possible, while 0 is the neutral, empty state.

η-Model

In [71] the relation of this model to an even simpler system was proposed. There each lattice site can assume only two states either a step up, $\eta_i = 1$, or a step down, $\eta_i = -1$. Particles can adsorb on this surface only at local minima, i.e. for two neighbouring sites which have the configuration $\eta_i \eta_{i+1} = (1-1)$ a transition to (-11) occurs with rate K. The corresponding microscopic rule is

$$1-1 \rightarrow -11 \quad \text{with rate } K. \tag{8.13}$$

A graphical illustration is given in Fig. 8.9. This model is fully equivalent to the deposition model above and will be used in the following. The relation between the surface steps σ_i and the new variables η_i is

$$\sigma_i = \eta_i + \eta_{i+1} - 1. \tag{8.14}$$

Thus each surface step σ is defined by a nearest neighbour pair of the variables η. A system with N lattice sites in the η-model is equivalent to a system with $N-1$ lattice site in the σ-model. The big advantage of the η-model is that the dimensionality of the phase space is 2^N, instead of 3^{N-1} for the equivalent σ-model.

In [71] the deterministic model with rate $K = 1$ and a model with random updating was considered. The description with a master equation is significantly more general and keeps track of the whole ensemble of all possible pathways. This lattice model can also be interpreted differently, e.g. as describing a lattice gas, by interpreting the state 1 as a gas-particle and the state -1 with a vacancy. Considering periodic boundary conditions, the model has some trivial steady states, e.g. the filled or empty states do not change under this dynamics. The boundary conditions are chosen as follows. In the lattice gas picture the microscopic rule Eq.(8.13) describes the hopping of a gas-particle to the right with rate K. It would be obvious to add a source term at the left and a sink term at the right boundary of the lattice. This gives the two additional boundary terms

$$-1(\cdots) \to 1(\cdots) \quad \text{with rate } K_r \tag{8.15}$$
$$(\cdots)1 \to (\cdots)-1 \quad \text{with rate } K_r. \tag{8.16}$$

In the deposition picture for η this corresponds to an adsorption on the interface of the system and the continuing surface. It is not known whether this is actually a local minimum, in general it will be not. Therefore an additional rate K_r was introduced. Since one has no information from outside of the system it would be reasonable to assume equal probability for a minimum localised at the interface leading for K_r to

$$K_r = \frac{K}{2}. \tag{8.17}$$

The master operator for the η-model can be constructed as in the previous section and contains only the terms for diffusion to the right together with the boundary terms

$$\mathcal{M} = K \sum_{<ij>} \mathcal{D}^R_{ij} + K_r \mathcal{S}_1 + K_r \mathcal{S}^\dagger_N, \tag{8.18}$$

with the right diffusion operator $\mathcal{D}^R$

$$\mathcal{D}^R_{ij} := a_i a^\dagger_j - n_i v_i. \tag{8.19}$$

Here use of the notations and operators from Section 4 is made.

If Eq.(8.17) is satisfied, the steady state is the state with equal probability for each microscopic state. In Chapter 4 it was already shown that this state is always a left eigenstate of every stochastic matrix. For the deposition model it is also a right eigenstate to the same eigenvalue 0 although $\mathcal{M}$ is not normal. This is due to the fact that also the sum of the entries for each row of $\mathcal{M}$ is zero. Thus the same argument as for the summation state in Chapter 4 holds for the left eigenstate of $\mathcal{M}^\dagger$ which is a right eigenstate of $\mathcal{M}$. More quantitatively this is exemplified for the two site system. The source and annihilation operators according to the deposition model for η are

$$\mathcal{S}_1 + \mathcal{S}^+_2 = \mathrm{kron}(\mathcal{S}, 1\!\!1_2) + \mathrm{kron}(1\!\!1_2, \mathcal{S}) = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix}. \tag{8.20}$$

Thus the sums of the rows of $\mathcal{S}_1 + \mathcal{S}^+_2$ are $(0, -2, 2, 0)$. The diffusion operator $\mathcal{D}_1$ is

$$\mathcal{D}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{8.21}$$

The master operator $\mathcal{M}$ in the two site case is $K\mathcal{D}_1 + K_r(\mathcal{S}_1 + \mathcal{S}_2^+)$. One easily sees that all row sums vanish exactly in the case $K_r = \frac{K}{2}$ which is adopted to obtain free boundary conditions.

Since the probability for a step up is equal to a probability for a step down at all lattice sites in the steady state, the expectation value for the surface steps is constant zero. This agrees with the deterministic continuous KPZ-equation whose solutions also tend to a flat surface for long times.

Further at least for small systems (N=6) the steady state is not degenerated. As it is known that the longest living transient states have low spatial frequency and increasing N leads to including small scale phenomena, one can argue that the non degeneracy of the steady state also holds for large N.

8.2.1. Numerical Results

The Schur DMRG algorithm has been applied to the deposition model described above, using the η-representation. As the actual phase space vectors are high-dimensional and not intuitively accessible some observables are considered in the analysis. In particular have the expectation value of the surface step variable σ_x and the correlation function

$$G(x) := \langle \sigma_0 \sigma_x \rangle \tag{8.22}$$

have been evaluated.

For very small systems the Schur decomposition can be performed exactly. Here the system with $N = 6$ is treated in this way. Choosing $m = 4$ for the number of retained degrees of freedom in each subblock, the superblock master operator is identical to the full master operator.

Systems of the size $N = 6$ to $N = 16$ were considered. The corresponding phase spaces are of dimension 64 to 65536. The degrees of freedom denoted by m were referred to directly, since also values of m were used which do not correspond to a particular physical sub-system, i.e. m does not have to be a power of two.

Normalisation of the Results

The entries of a state vector Ψ have a probabilistic interpretation in that each entry gives the probability to find the system in the corresponding microscopic state. Thus the correct normalisation would be

$$\sum_i \Psi_i = 1. \tag{8.23}$$

By construction the entries of the steady state vector are all positive (or zero). The transient states are Schur vectors and as such orthonormal. However, their physical interpretation is to be a correction to the steady state vector for long but finite times. Therefore they can have negative entries and typically their average $\sum_i \Psi_i^{\text{trans}}$ is even zero. The magnitude of the transient states depends only on

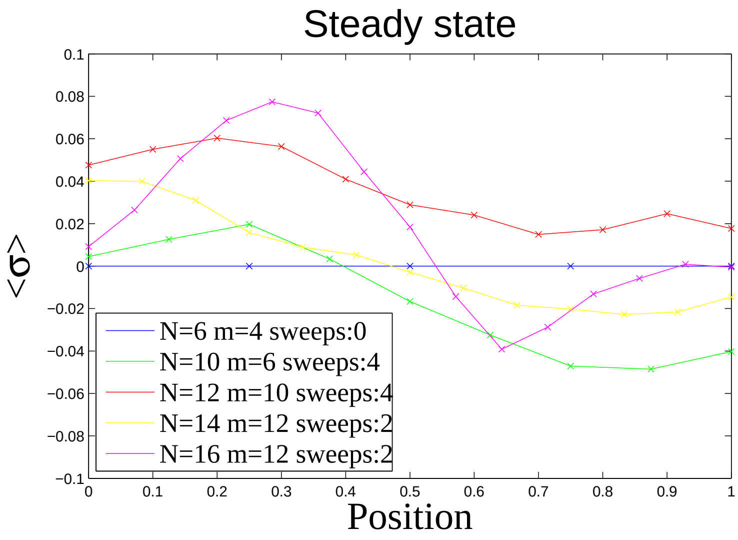


Figure 8.10.: Expectation value for the surface step variable σ for the steady state.

the initial conditions, but the analysis only considers the temporal decay of the modes. For comparison one has to choose an appropriate normalisation for the transient states.

To do so a normalisation factor is chosen c so that $\sum_i \sigma_i = 1$ for the first transient state. This is meaningful since it is known that for this state the average surface step size is always positive. This is not true for the second transient mode. Therefore c is chosen so that $\sum_i |\sigma_i| = 1$.

Average Surface Step Size

As discussed above the average surface step size for the steady state is zero at all lattice sites. The numerical results for the steady state are shown in Fig. 8.10. The deviations are exclusively due to numerical inaccuracies and not due to finite size effects [1], since the steady state has zero average surface step size for all values of N. By increasing m these inaccuracies can be reduced. Also increasing the number of sweeps has a positive effect, but if m is chosen too small this cannot be compensated. The calculations took some seconds up to a few hours. More extensive calculations were not carried out, since also the working memory was a limiting factor. Beside efficiency issues the algorithm can further be improved, e.g. by using full pivoting in solving the Sylvester equation or the QR-decomposition in the Schur ordering. These 'canned' routines were used if possible to avoid errors and get results relatively quickly. The results presented here can only provide a qualitative study in any case, due to the oversimplified models.

[1] This is merely an analogy for a finite size effect since increasing lattice size does not directly correspond to a more detailed description of an ideally infinite system. However it should be clear what is meant with the term in this context.

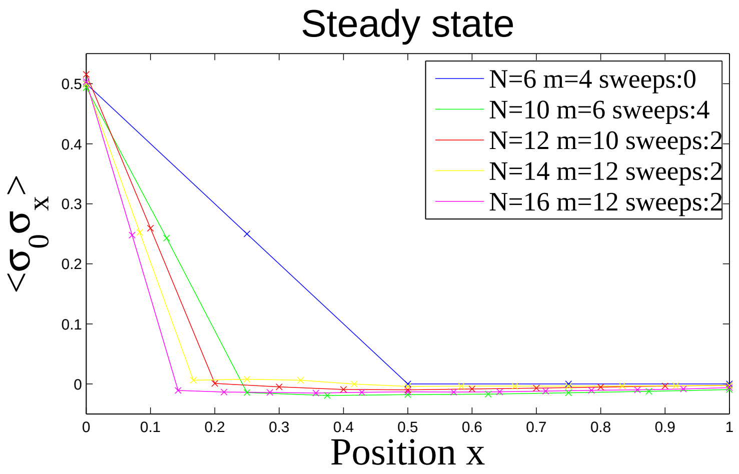


Figure 8.11.: Expectation value for the correlation function $\langle\sigma_0\sigma_x\rangle$ of the surface step variable σ for the steady state.

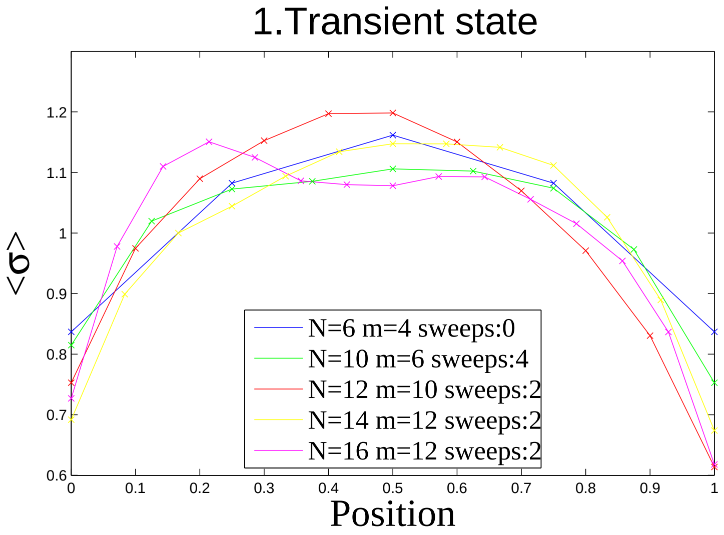


Figure 8.12.: Expectation value for the surface step variable σ for the first transient state. The normalisation is chosen so that $\int \sigma_x dx = 1$.

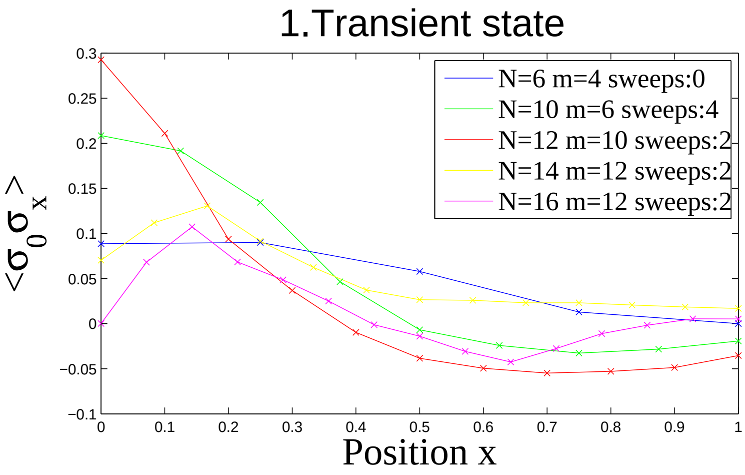


Figure 8.13.: Expectation value for the correlation function $\langle\sigma_0\sigma_x\rangle$ of the surface step variable σ for the first transient state. The normalisation is chosen so that $\int\sigma_x dx = 1$.

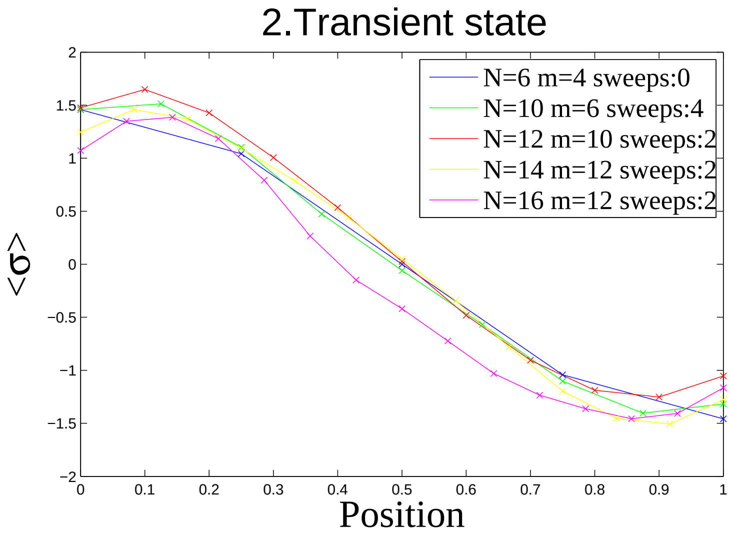


Figure 8.14.: Expectation value for the surface step variable σ for the second transient state. The normalisation is chosen so that $\int|\sigma_x|dx = 1$.

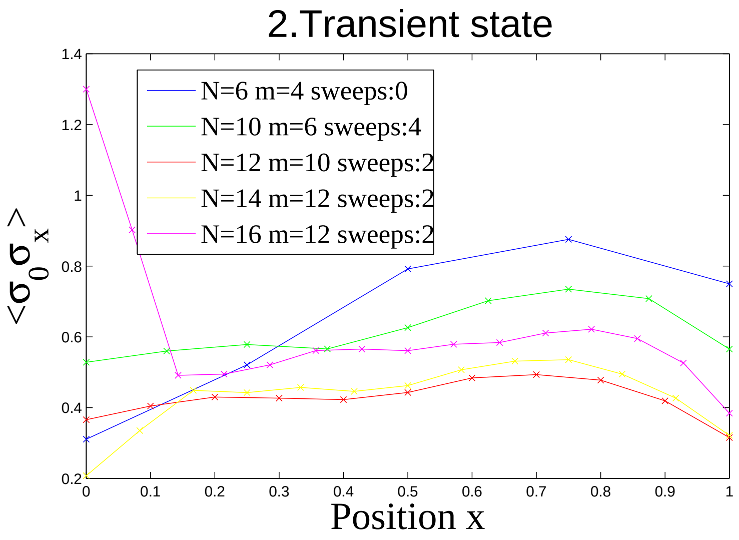


Figure 8.15.: Expectation value for the correlation function $\langle\sigma_0\sigma_x\rangle$ of the surface step variable σ for the second transient state. The normalisation is chosen so that $\int|\sigma_x|dx = 1$.

The first excited state is a symmetric parabola like function. This is reproduced by all calculations qualitatively, although also here significant noise is present. For increasing N the average surface step size seems to tend to zero at the boundaries. This could be attributed to a finite size effect. Although the data are noisy this interpretation is further supported by the results presented below in Section 8.2.2.

The third excited state is antisymmetric under reflection of the spatial coordinate. Again one observes significant noise, and also here the states tend to zero at the boundaries for large N. This is also supported by the results in Section 8.2.2.

Correlation of Surface Steps

The correlation function of the surface steps is defined by

$$G(x) := \langle\sigma_0\sigma_x\rangle\,. \tag{8.24}$$

Some scaling arguments were given in [66] for the infinite system and $\beta = 1$. They obtain for large correlation length $L := 2t$ and t

$$G(x,t) := \frac{1}{\pi\sqrt{2tx}}f(x/L), \tag{8.25}$$

The function f is given by

$$f(y) = \begin{cases} \frac{1}{1+y} & y \leq 1, \\ 0 & \text{else.} \end{cases} \tag{8.26}$$

The normalisation constant for the states in this analysis are the same as above. For the steady state a vanishing correlation is obtained for more than two sites away from the origin, see Fig 8.11. There the correlation decays linearly. The inaccuracies are acceptable and Fig. 8.11 supports the interpretation of the nonzero part as finite size effects. A zero correlation for $t \to \infty$ is also predicted by Eq.(8.25). For the first transient state the correlation shown in Fig 8.13 is larger over long distances. Near the origin the inaccuracies are significant and allow no definite statement. However, the shape of the correlation decay is different from those proposed by Eq.(8.11) in particular for the exact $N = 6$ system. The correlation decreases even towards the origin for the second excited state, see Fig 8.13, which is not in agreement with Eq.(8.11). The inaccuracies are even higher so it cannot be confirmed that this is due to finite size effects.

8.2.2. POD results

Stochastic Simulation

As for the reaction diffusion model the master equation can also be simulated directly. Then the surface profile $h(x,t)$ and consequently the surface steps $\sigma_x(t)$ underly a stochastic time evolution. The transition rates for this evolution are determined by the master operator. In this way sample trajectories can be constructed. On this basis of course also a proper orthogonal decomposition is possible. This has been evaluated and the resulting first three POD modes are shown in Fig. 8.16. It can be stated that there is a qualitative agreement of the data. All modes tend to zero at the boundaries which is not present in the calculations above, but for increasing N there is a tendency for this effect. One reason for the qualitative agreement here is that for the deposition model the height profile determines the surface steps completely and all information is available for the POD.

Simulation of the KPZ Equation

Simulating Eq.(8.7) for $\beta = 1$ directly and performing a POD gives the modes presented in Figs. 8.17. As stated above the relation of Eq.(8.7) to the microscopic models is merely quantitative. It is not known a priori how to relate the parameters in Eq.(8.7) to those of the deposition model. Therefore just the qualitative behaviour can be compared. The nonlinear coefficient ν has been varied, as for the dynamics only the ratio of d to ν is relevant upon a rescaling. The simulations were carried out using a spectral method and explicit Euler integration. The impact of the nonlinear constant ν on the POD models is small. The excited modes look like the real and imaginary part of the lowest Fourier mode. Considering that in this case the subspaces spanned by these mode are identical the influence of ν is more or less nonexistent. Comparing the results from this section with those from the microscopic model one could assume homogeneous Dirichlet boundary conditions were applied. However this is not true. The explicit simulation of the master equation allows for a direct visualisation of the surface profile evolution.

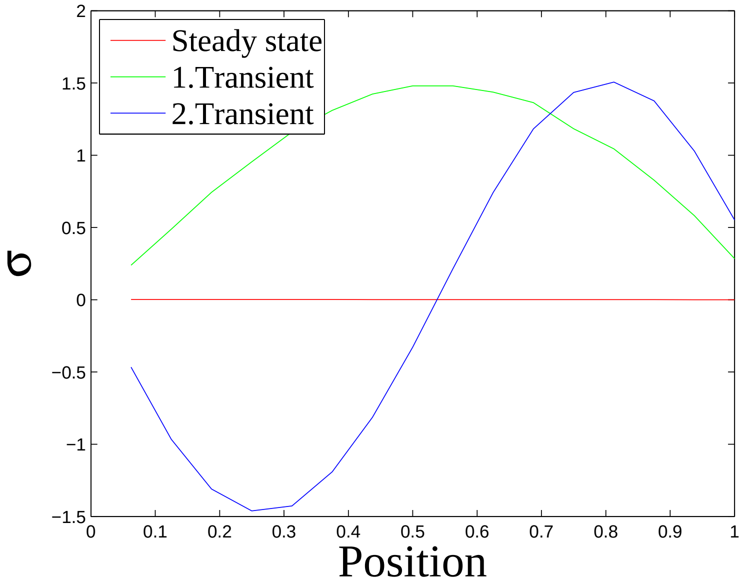


Figure 8.16.: Expectation value for the surface step variable obtained from an explicit simulation of the deposition model and a proper orthogonal decomposition.

These visualisations show the violation of homogeneous Dirichlet boundary conditions. Graphical illustrations of the stochastic surface profile evolution are not presented here, since they provide no deeper insight to the problem.

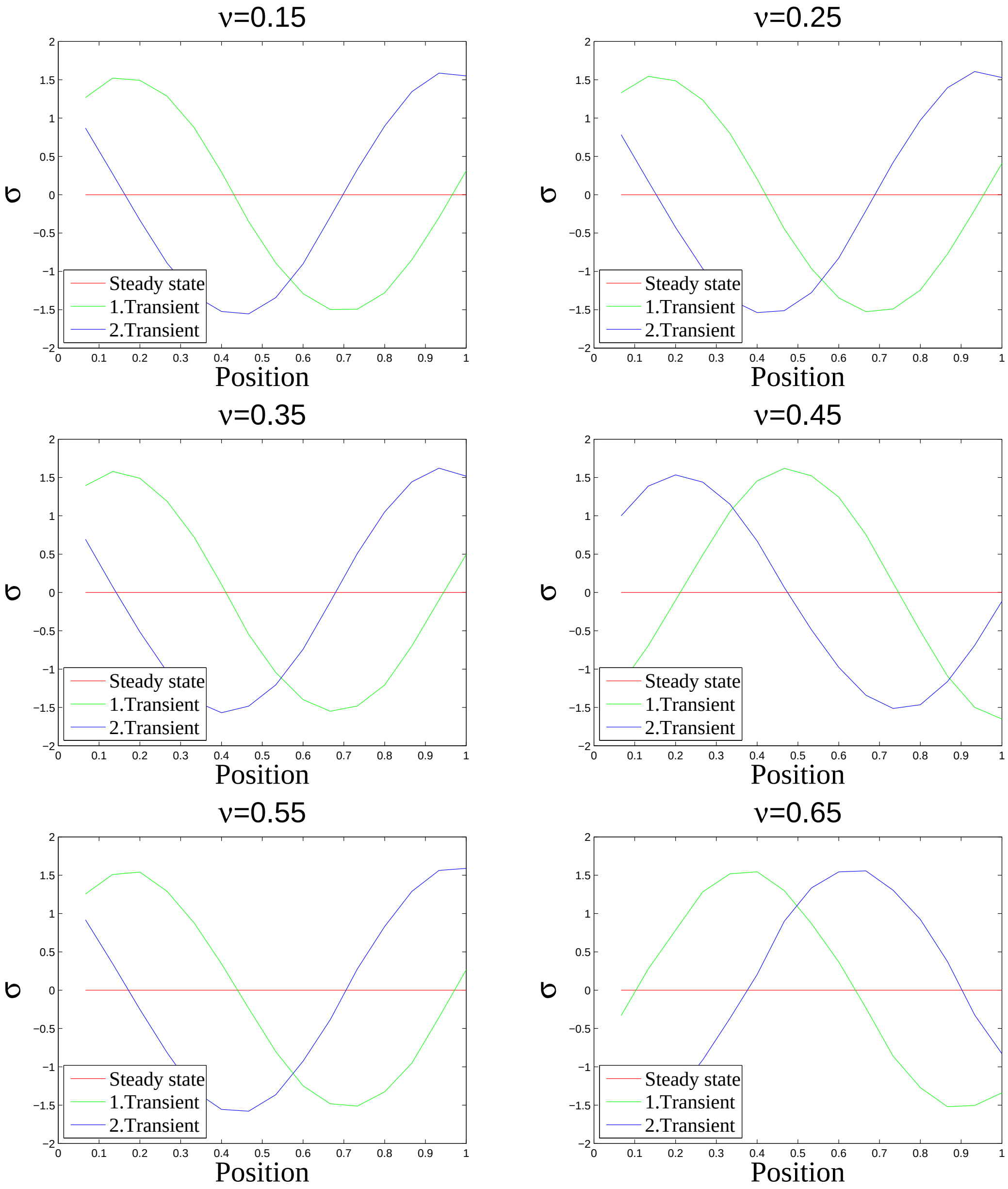


Figure 8.17.: POD modes for a direct simulation of Eq.(8.7). 10^6 time steps were carried out for each of 10 different initialisations on a 16 site grid. The initial conditions were chosen uniformly between 0 and 1. The nonlinear constant ν was varied. The other parameters were $d = 0.05$, $\Delta t = 0.01$ and $\sqrt{< \xi^2 >} = 0.01$.

9. Proper Orthogonal Decomposition DMRG

In the following I apply the proper orthogonal decomposition DMRG, introduced in Section 7.2, to three one-dimensional model equations, namely the diffusion equation, the Burgers equation [24] and an nonlinear diffusion equation [17]. For all applications a finite differencing scheme of second order accuracy, homogeneous Neumann conditions at the boundaries and the explicit Euler method are chosen for the calculations. The details on this methods are given in Chapter 3 before. The boundary conditions as well as the time integration method can be chosen - more or less - arbitrarily. However, higher order finite elements in the spatial discretisation lead to additional interactions between single dofs, i.e. a form of non-locality, and do thereby complicate the problem. For the reduced system size always four dofs were retained. This is mainly for convenience and easy comparison. The success of the method does not depend strongly on this choice.

As explained above, the quality of a reduction is measured by the L^2-error, see Eq.(5.15). It has the same units as the fields Φ which are not further specified. The time units are also arbitrary.

The error calculations in the following are performed in a separate program which gets the optimised bases from the various methods as input. Thus the simulation time do not have to coincide with the length of the POD simulations. Further, the random seed for statistical initial conditions was modified for calculation of the POD and for calculation of the error unless otherwise stated.

9.1. The Linear Diffusion Equation

The diffusion equation describes diffusive transport of a scalar field, e.g. heat transport, in a medium. For homogeneous media it is given by

$$\frac{\partial}{\partial t}\Phi(x,t) = d\Delta\Phi(x,t) \quad x \in [0,1] \tag{9.1}$$

with the diffusion constant d. Homogeneous Neumann conditions are assumed for $x = 0$ and $x = 1$, the spatial discretisation step size is $\Delta x = \frac{1}{N}$. The explicit Euler method gives for the discrete time evolution with time step size h_t the following discrete equation

$$\tilde{\Phi}(\tilde{x}, t_{n+1}) = \tilde{\Phi}(\tilde{x}, t_n) + dh_t\Delta_N\tilde{\Phi}(\tilde{x, t_n}) \tag{9.2}$$

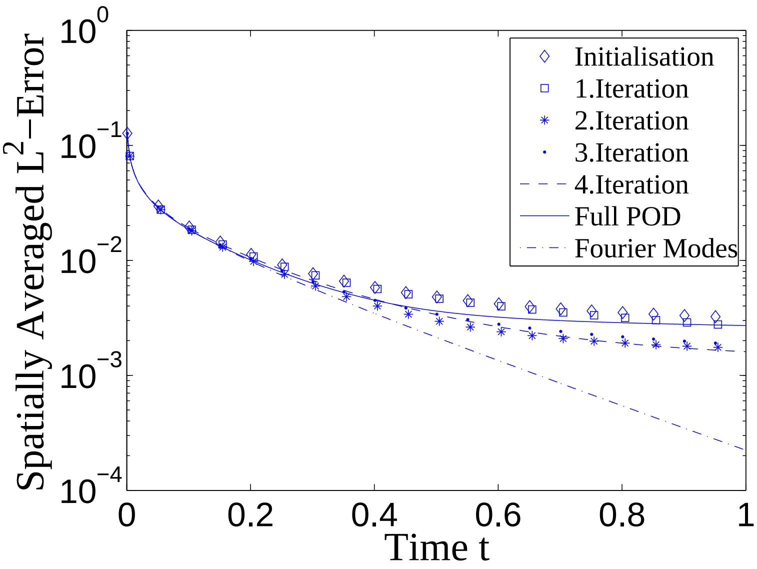


Figure 9.1.: Reduced diffusion equation L^2-error $E(t)$ for the analytical reduction (Fourier Modes) the full POD and DMRG POD after initialisation and several iterations, statistical initial condition, N=40, $h_t = 0.001$ and $d = 0.05$. The error is expressed in units of Φ, for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

where $\tilde{\Phi}$ and $\tilde{x}$ are N-dimensional vectors, indicated by $\tilde{\cdot}$. Thus the linear part L in Eq.(5.2) is given by

$$L = dh_t\Delta_N. \tag{9.3}$$

Thus the only nonzero contribution according to Eq.(5.2) is $L \equiv \Delta_N$. The eigenstates of L are the sine/cosine or Fourier modes whose contributions decay over time with characteristic life-time inversely proportional to the frequency/energy. Standard DMRG can be viewed as an approximate diagonalisation method for an linear operator. Therefore it is very effective to find the optimal reduction determined by the eigenstates, see Appendix B, in the linear case. In contrast to the diagonalisation, POD as well as our method depends on the initial conditions for the sample trajectories over which the averaging is carried out. Both POD approaches cannot exploit the linearity of the evolution equation. This affects the quality of the results for linear problems compared to diagonalisation-based methods. Nevertheless, restriction to a few sample trajectories can also be an advantage, since sometimes the interest lies on a certain region in phase space. However, for the diffusion equation normally distributed initial conditions are chosen, i.e. the field $\Phi_0(x_i)$ is normally distributed. This is then also true for the Fourier modes. By this choice effectively the whole phase space will be sampled for a high enough number of realizations. This is also due to the invariance of Eq.(9.1) under multiplication with a constant factor. For the POD it is important

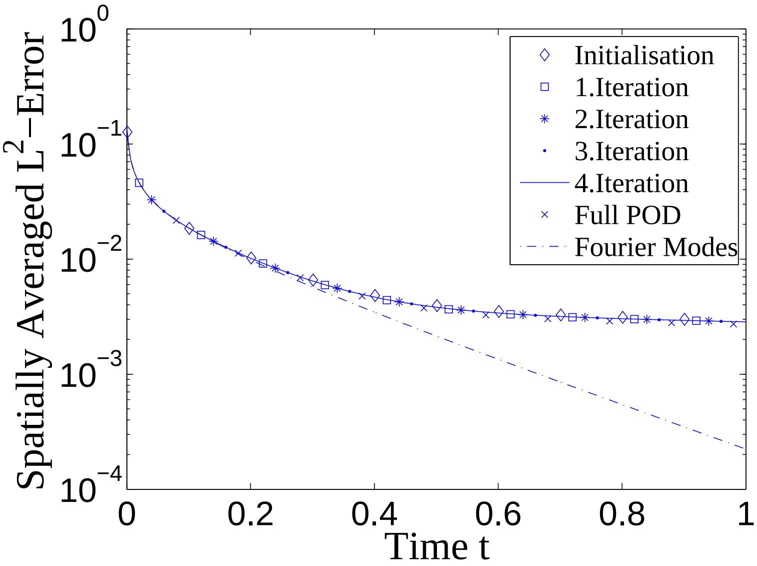


Figure 9.2.: Reduced diffusion equation L^2-error $E(t)$, identical statistical initialisation. The error is expressed in units of Φ, for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

to integrate over sufficiently long times. For short times the state moves in the direction of the highest frequency modes which are decaying most rapidly. Thus POD would give the wrong relevant modes.

The POD is in fact not a very appropriate tool to reduce the whole phase space of the diffusion equation. In Fig. 9.1 the error of the reduced fields $\hat{\Phi}$ is plotted in dependence of time. There the time step was $dt = 10^{-3}$ and the diffusion constant $d = 0.05$. The spatial resolution was 40 lattice sites within the interval $[0, 1]$. In each POD step as well as for the error calculation the ensemble average has been evaluated considering 50 realizations of the initial conditions. From this result one can state several things. First, all POD-based methods show a remaining error in the long time limit. Second, the initialisation steps of DMRG POD gives already reasonable results. An improvement due to the iteration is present, too. Third, the new algorithm is able to compute the optimal reduction with even higher accuracy than the full POD. The last point is only paradox on the first glance. The inaccuracy of the full POD is in this case influenced by the statistical initial conditions, in order to sample the full phase space. Within the algorithm, much more initial conditions are taken into account as the superblock POD is performed repeatedly. This leads to a better statistics. In Fig. 9.2 the same results are shown but using always the same initialisation for calculating all PODs (but of course not for the error calculation). It is clear that in this case, the new method has no advantage over the full POD anymore. On the other hand, the results from the proposed algorithm are not worse than that from the full system POD, which is

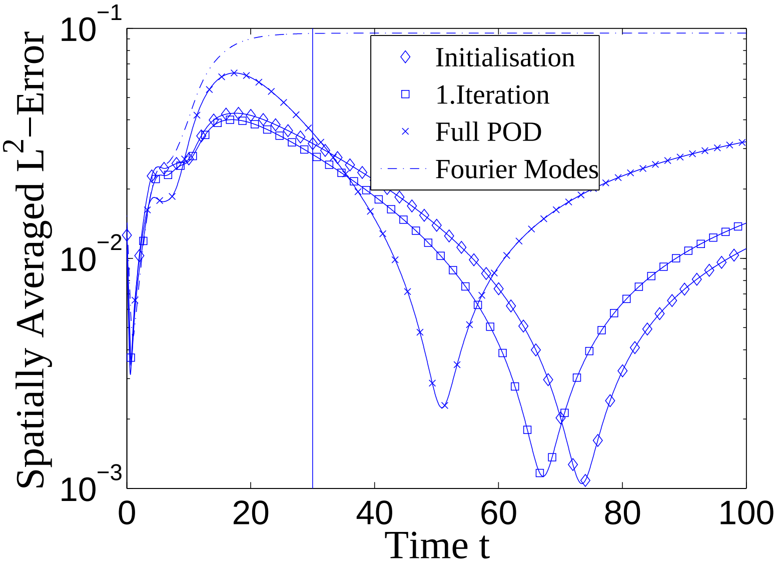


Figure 9.3.: Reduced Burgers equation L^2-error $E(t)$, deterministic initial condition, N=40, $h_t = 0.02$, $d = 0.01$ and $\nu = 0.1$. The error is expressed in units of Φ, for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

not clear a priori.

9.2. The Burgers Equation

As one nonlinear example the Burgers equation [24] is considered. It describes a diffusive as well as a convective transport of a scalar field Φ and is given by

$$\frac{\partial}{\partial t}\Phi = d\Delta\Phi + \nu(\Phi\nabla)\Phi. \tag{9.4}$$

This equation is similar to the linear diffusion equation Eq.(9.1) but with an additional term $\nu(\Phi\nabla)\Phi$, describing the convection. This term is quadratic in the field Φ and can be discretised in the form of Q in Eq.(5.2). For one space dimension, the ∇ operator is simply the spatial derivative. This has been discretised by the centred differencing scheme from Eq.(3.32). The term $(\Phi\nabla)$ is also known as convective derivative. In 1D the discretisation is given by multiplying the rows of $D_{x,N}$ with the components of $\tilde{\Phi}$:

$$(\Phi\nabla)_{N,i,j} = \tilde{\Phi}_i D_{x,N,i,j}, \tag{9.5}$$

here i, j indicate the component of the matrix/vector. Choosing

$$Q_{i,j,k} := \nu D_{x,N,j,k}\delta_{ij} \tag{9.6}$$

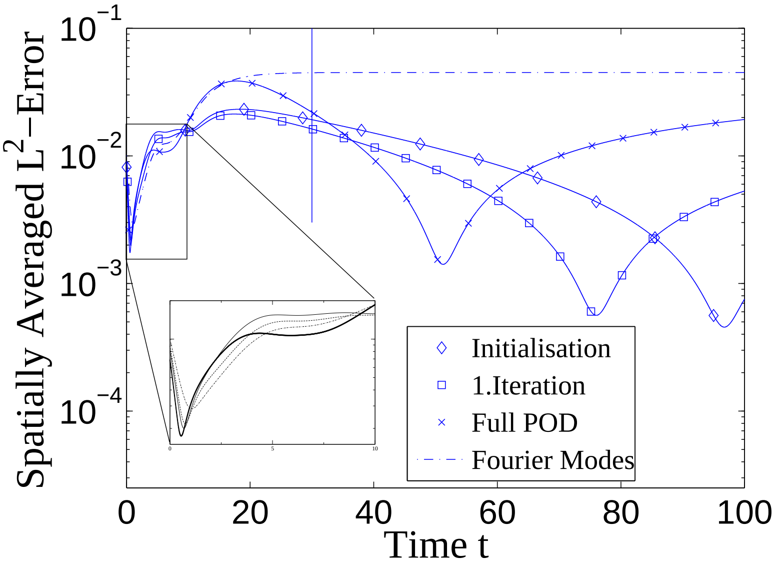


Figure 9.4.: Reduced Burgers equation L^2-error $E(t)$, deterministic initial condition, N=100, $h_t = 0.005$, $d = 0.01$ and $\nu = 0.1$. The inset shows the begin of the error evolution enlarged. The error is expressed in units of Φ, for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

gives a discretisation of the convection term, as defined in Eq.(9.5)

$$\begin{aligned}\sum_{j,k} Q_{i,j,k}\tilde{\Phi}_j\tilde{\Phi}_k &= \nu \sum_{j,k} D_{x,N,j,k}\delta_{ij}\tilde{\Phi}_j\tilde{\Phi}_k \\ &= \nu \sum_{k} \tilde{\Phi}_i D_{x,N,i,k}\tilde{\Phi}_k = \nu\left(\Phi\nabla\right)_N \tilde{\Phi}. \qquad (9.7)\end{aligned}$$

To begin with, deterministic initial condition for the calculation of all PODs are chosen. In particular these are of the form

$$\Phi(t=0,x_i) = e^{-50(x_i-1)^2} \quad x_i = 0\ldots 1. \qquad (9.8)$$

Fig. 9.3 and 9.4 show the results for the L^2-error of the evolution. Here two spatial resolutions, i.e. $N = 40$ and $N = 100$ nodes were used. The results are very similar. In contrast to the previous calculations the simulation runs for the error calculation are longer than the POD runs. The vertical line indicates the time interval of the POD runs. Here one has to state that the Fourier mode reduction is not optimal, which is not surprising as a nonlinear system and a very particular region of phase space were considered. Further, one sees that the error curves show a very pronounced minimum after which the approximation seemingly breaks down. The corresponding time point lies well after the POD time-span. These minima correspond to the fact that after the passing of the wavefront the

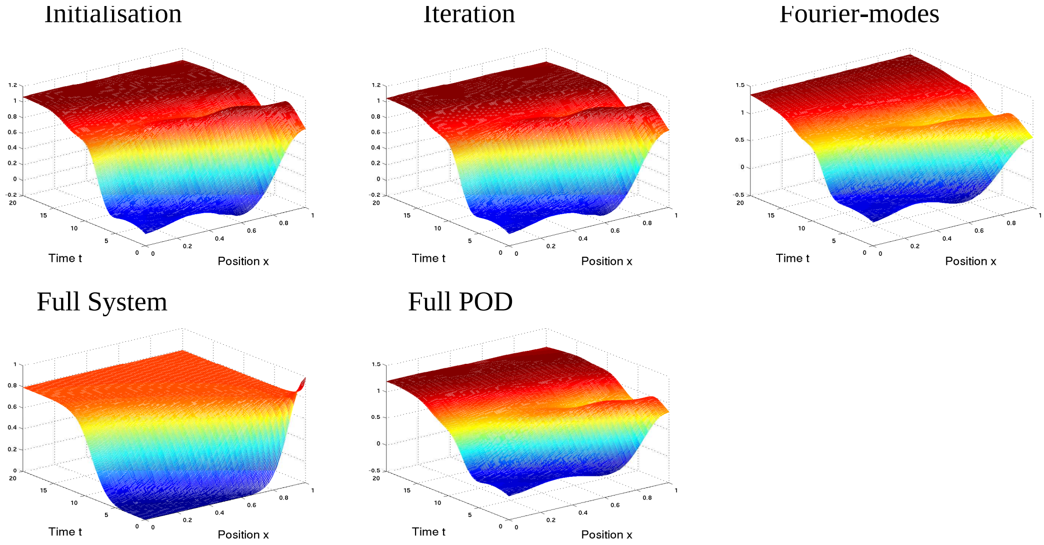


Figure 9.5.: Examples for the field evolution, Burgers equation, deterministic initial condition, $N = 40$, $h_t = 0.02$, $d = 0.01$ and $\nu = 0.1$.

profile becomes flat. The approximations do not reproduce the average value accurately, but show a spurious drift. The passing of the reduced (flat) states by the original (flat) state creates the minima in Fig. 9.3. To get some qualitative insight also the time evolution of the field for deterministic initial conditions, $N = 40$, $h_t = 0.02$, $d = 0.01$ and $\nu = 0.1$ is exemplarily shown in Fig. 9.5 for the complete system and the reduced dynamics determined by a full POD, Fourier modes, the POD-DMRG initialisation step and one iteration step. All reduced systems show artifacts, although they are less pronounced for the POD-DMRG results.

It is remarkable that the proposed method yields better results than the POD within the POD time, even for the initialisation. Here it should be recalled that the POD is optimal only for reconstructing the states used in the calculation. As stated above, the reconstruction of the dynamics that created these states, is a different thing as can be directly seen from the results.

The analysis of the Burgers equation is continued by considering statistical initial conditions. In contrast to the calculations for the diffusion equation there are only three randomly sampled parameters in the initial condition. It is given by a peak of various height H, width W and position X. In particular it is defined by the following equation

$$\Phi(t = 0, x_i) = He^{-50W(x_i - X)^2}. \tag{9.9}$$

Here, H and W are normally distributed whereas X is uniformly distributed.

The results are shown in Fig. 9.6. For all methods the error reaches a plateau very quickly. The performance of the full system POD is slightly better than that of the DMRG POD. However, the errors from the new approach are of the same

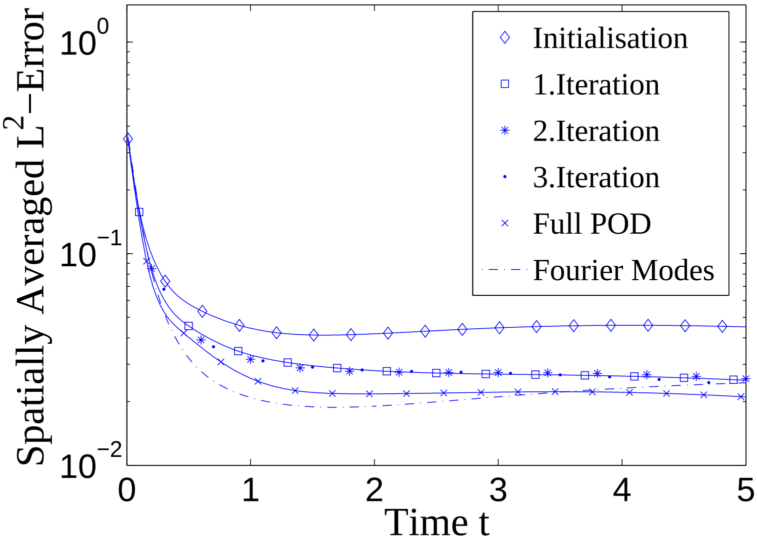


Figure 9.6.: Reduced Burgers equation L^2-error $E(t)$, statistical initial condition, N=20, $h_t = 0.01$, $d = 0.05$ and $\nu = 0.1$. The error is expressed in units of Φ, for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

order as from the full POD and one magnitude better than that of the Fourier-mode based reduction. Also the iteration brings an improvement which reaches saturation already after the first step.

For deterministic initial conditions the evolution of the error is not monotonic in contrast to the case of statistical initial conditions. This is due to the fact that deterministic initial conditions can be considered more effectively by the POD. The statistical initial conditions were drawn from a three-dimensional, see Eq.(9.9) or two-dimensional, see Eq.(9.16), subspace which is reproduced poorly by a reduction to a four-dimensional space, which has to consider the time evolution also.

9.3. Nonlinear Diffusion

Here a diffusion equation with a nonlinearity that resembles the action-potential part of the one-dimensional FitzHugh-Nagumo(FN) [46, 127] equation is considered. In particular the dynamics is defined by

$$\frac{\partial}{\partial t}\Phi = \Delta\Phi - \Phi(1-\Phi)(a-\Phi) \tag{9.10}$$

where a is a constant. Eq.(9.10) has stable equilibria at $\Phi \equiv 0$ and $\Phi \equiv 1$ and an instable equilibrium at $\Phi \equiv a$.

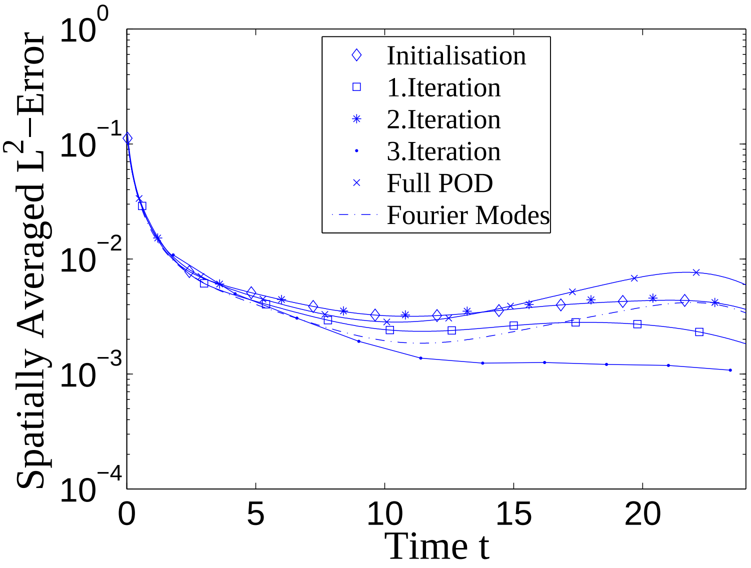


Figure 9.7.: Reduced nonlinear diffusion equation L^2-error $E(t)$, statistical initial condition, N=30, $h_t = 0.03$, $d = 0.01$ and $a = 0.5$. The error is expressed in units of Φ, for the time axis arbitrary units are employed. Note that for clarity not all data points are shown as symbols.

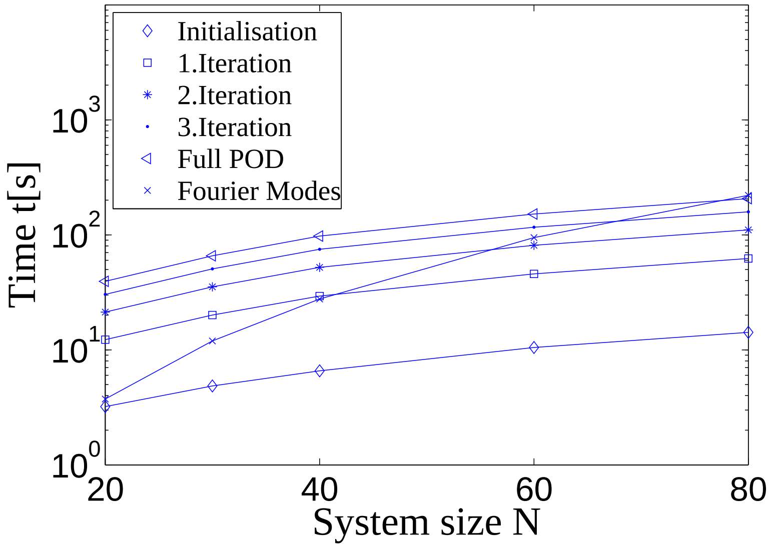


Figure 9.8.: Computing time for various system sizes and approaches, obtained by the reduced Burgers equation, statistical initial condition, N=40, $h_t = 0.005$, $d = 0.01$ and $\nu = 0.1$.

The nonlinear term is cubic in the field. It can be rewritten as

$$-\Phi(1-\Phi)(a-\Phi) = -\Phi^3 + (1+a)\Phi^2 - a\Phi. \tag{9.11}$$

Here the powers of Φ are defined component wise. The cubic part $-\Phi^3$, e.g. is discretised by

$$K_{i,j,k,l} = -\delta_{ij}\delta_{ik}\delta_{il}, \tag{9.12}$$

since

$$\sum_{j,k,l} \left(-\delta_{ij}\delta_{ik}\delta_{il}\right) \tilde{\Phi}_j \tilde{\Phi}_k \tilde{\Phi}_l = -\tilde{\Phi}_i^3. \tag{9.13}$$

Similarly, the quadratic part becomes

$$Q_{i,j,k} = (1+a)\,\delta_{ij}\delta_{ik}, \tag{9.14}$$

and the linear part together with the contribution from the diffusive term is

$$L_{i,j} = dh_t \Delta_{N,i,j} - a\delta_{ij}. \tag{9.15}$$

As initial conditions a front with uniformly distributed position X and normally distributed height H was chosen:

$$\Phi(t=0,x_i) = \frac{H}{2}\tanh((x_i - X)10). \tag{9.16}$$

Under this conditions all methods were able to reproduce well the dynamics, see Fig. 9.7. Surprisingly the full POD method gave poorer results than even the Fourier-mode based reduction. This is to a lower extent also true for the initialisation run of the DMRG POD. The iteration lead to an improvement although the second iteration gave similar results as the initialisation. Further iterations again increase the accuracy, so no general statement can be made. By applying the iteration procedure repeatedly a decay in the quality of the result was observed after a fast saturation. This can be likely attributed to the accumulation of numerical errors.

9.4. Computational Load

For all calculation steps, e.g. diagonalisation, Gram-Schmidt orthonormalisation etc., standard algorithms were applied [50, 96]. The focus was more on a concise assessment of the new algorithm instead of an optimal solution of the toy problems. For the diagonalisation of the covariance matrix, e.g. first a Householder-tridiagonalisation was performed [50], which is an $\mathcal{O}(N^3)$ algorithm. The calculation of the POD, either for the complete system or for the superblock system was performed with the same routine. This comprised the simulation as well as the diagonalisation.

full system POD	DMRG POD
$O(N^3) + S(N)$	$NN_iO(M) + S(M)$

Table 9.1.: Naive estimation of the computational load, full system size N, superblock size M, number of iterations N_i.

For a POD the simulation of the system in the time-span of interest is additionally necessary. Within the proposed approach the simulation and diagonalisation is performed only on the superblock system. A comparison of the results from Fig. 9.3 and 9.4 suggests, that the necessary number of iterations (sweeps) does not depend on the full system size N. If one denotes the superblock size with M and the number of iterations with N_i a naive estimation of the computational load is given in Table 9.4.

For a more quantitative analysis the time necessary to perform a full POD comprised of simulation and diagonalisation was measured. Then the same was done for the initialisation of the DMRG POD algorithm including all simulation and diagonalisation steps until the superblock system described the full system of dimensionality N, compare Fig. 6.6, and a first reduced basis had been calculated. Also the computing time for one further iteration step was measured in the same way as for the initialisation. The computing time is constant for all iteration steps so further data was extrapolated. The underlying equation was the deterministic initialised Burgers equation although the choice for an equation affects the computational load only marginally. As parameters were chosen $h_t = 0.005$, $d = 0.01$ and $\nu = 0.1$. Fig. 9.8 shows a logarithmic plot of the results. The DMRG POD approach shows a lower amount of computing time for the initialisation step. For higher system size this holds also for the iterations. Generally the scaling with N is favourable. Note, that here only the DMRG method should be assessed. For this purpose public assessable standard algorithms are sufficient, although much more effective methods could be possible. All calculations were performed on an Intel Dual Core machine, using a single CPU.

10. General Variational Method for Proper Orthogonal Decomposition

Big whirls have little whirls,
which feed on their velocity.
Little whirls have lesser whirls,
and so on to viscosity.

When little whirls meet little whirls,
they show a strong affection;
elope, or form a bigger whirl,
and so on by advection.

L.R. Richardson (1925), R. R. Trieling (1999)[1]

In the previous chapter the results for the POD-DMRG approach were presented. Although the assessable models are too restricted for practical use the results were good enough to encourage further progress. An algorithm to avoid restriction to (quasi) one-dimensional systems has been presented already in Section 7.3. I will now apply this algorithm to a problem which is nontrivial but well-investigated, namely the flow of a two-dimensional incompressible fluid.

10.1. Flow Problems

A fluid or gas at standard conditions (see Section 4) consists of a huge number of molecules that obey some equations of motions. Due to the number of molecules of order 10^{23} it is practically impossible to solve these equations. Further this knowledge is mostly even of no interest. Nevertheless, a systematic derivation of effective equations is far from trivial. A possible approach is to start at a microscopic model and use statistical methods to obtain a macroscopic description. A prominent example is the Boltzmann equation which determines the molecular velocity distribution functions [29, 57, 19]. Therefore it describes the fluid at a level which is more detailed than a pure macroscopic level. On the other

[1] The rhyme of L.R. Richardson is based on a version of De Morgan (1872): Great fleas have little fleas upon their backs to bite 'em,// And little fleas have lesser fleas, and so ad infinitum.// And the great fleas themselves, in turn, have greater fleas to go on;// While these again have greater still, and greater still, and so on (from *A budget of Paradoxes*, London:Longmans, Green, p.377). This itself is paraphrased from J.Swift: So, naturalists observe, a flea// Has smaller fleas that on him prey;// And these have smaller still to bite 'em;// And so proceed ad infinitum.

hand it does not describe individual particles and relies on the applicability of statistical models for the microscopic details which can also include thermal fluctuations [42]. This ansatz is thus also termed mesoscopic. From the molecular velocity distribution functions all macroscopic variables can be calculated. Due to the complexity of the Boltzmann equation numerical methods are usually necessary to find a solution. In this field much progress has been made by lattice Boltzmann methods [109, 31, 110]. It is also possible to derive effective equations for macroscopic variables. Via the Chapman-Enskog expansion it is e.g. possible to derive the Euler equation [102] or the Navier Stokes equations [57]. The Boltzmann equation is more general and gives also information on transport coefficients, as e.g. the diffusion coefficient, that occur as empiric parameter in the Navier Stokes equation.

10.1.1. Navier Stokes Equations and 2D Flows

The Navier Stokes equations constitute a general framework for the description of the macroscopic variables of a fluid and have thus great practical importance. They are also assumed to describe adequately the phenomenon of turbulence since the average vortex frequency and the Kolmogorov length, i.e. the scale at which friction dominates, are significantly higher than the molecular scales of collision frequency and mean free path length. Up to the present day turbulence is not understood completely. In order to study coherent structures that emerge in turbulent flow also the proper orthogonal decomposition was employed [107]. For numerical analyses turbulence is often included by some effective models since the resolution of the numerical description is always limited [43]. For the description of turbulence statistical approaches had been proposed [118, 112]. One problem there is the closure problem. In order to calculate correlations, higher order correlations have to be known [72]. To resolve this usually some assumptions are made at some point [60, 124, 113]. Numerical analyses were performed with the advances of computer technology. First incompressible two-dimensional flows were considered, e.g. [70, 78]. Finite differencing schemes that consider some symmetries of the equations were proposed by Arakawa [5], later spectral methods [52] became popular. It should be noted that two-dimensional turbulence differs significantly from its three-dimensional counterpart. In two-dimensional flows e.g. the effect of vortex stretching is absent. In three dimensions, if a vortex is stretched, the rotating fluid is moved to the vortex line. Conservation of angular momentum leads to an increase of angular velocity. In two dimensions the vorticity is always perpendicular to the plane of motion so that it can be described by a quasi scalar. Due to this property energy is transported from smaller scales to larger scales. This is effectively an example for self-organisation [58, 117]. Among other effects, vortex dipoles show a behaviour qualitatively similar to elementary particles [116]. Of fundamental importance for the self-organisation is the process of merging of two vortices with equal sign. This has been studied e.g. in [90]. Two-dimensional flows can be observed e.g. in stratified fluids. One important example is the atmosphere [41].[2] There also additional forces due to the rotating frame of reference as the Coriolis force have to be considered [62, 97]. Also variational methods have been applied to the two-dimensional Euler-flow for steady state problems [7]. The unforced, incompressible viscous 2D Navier Stokes equation will be used as a testing ground for the method.

To recapitulate, the Navier Stokes equations read

$$\rho\frac{\partial \mathbf{v}}{\partial t} = \mu\nabla^2\mathbf{v} - \rho(\mathbf{v}\nabla)\mathbf{v} - \nabla p, \tag{10.1}$$

$$\frac{\partial \rho}{\partial t} + \nabla\left(\rho\mathbf{v}\right) = 0. \tag{10.2}$$

Here ρ is the density and μ the dynamic fluid viscosity. The first term in Eq.(10.1) describing the viscosity and can be interpreted as a 'diffusion of momentum' while the second term in Eq.(10.1) is the convective derivative which conserves energy. For incompressible flow ρ is constant and Eq.(10.2) reduces to

$$\nabla\mathbf{v} = 0. \tag{10.3}$$

The Navier Stokes equations can be rescaled to compare flows on different length L_0 and velocity scales v_0. Using the rescaled variables $\hat{\mathbf{v}} = \frac{\mathbf{v}}{v_0}$, $\hat{p} = \frac{p}{v_0^2\rho}$ this results in

$$\begin{aligned} & v_0\tfrac{v_0}{L_0}\tfrac{\partial\hat{\mathbf{v}}}{\partial\hat{t}} = \tfrac{v_0\mu}{L_0^2\rho}\hat{\nabla}^2\hat{\mathbf{v}} - \tfrac{v_0^2}{L_0}(\hat{\mathbf{v}}\hat{\nabla})\hat{\mathbf{v}} - \tfrac{v_0^2\rho}{L_0\rho}\hat{\nabla}\hat{p} \\ \Leftrightarrow \quad & \tfrac{\partial\hat{\mathbf{v}}}{\partial\hat{t}} = \tfrac{\mu}{v_0L_0\rho}\hat{\nabla}^2\hat{\mathbf{v}} - (\hat{\mathbf{v}}\hat{\nabla})\hat{\mathbf{v}} - \hat{\nabla}\hat{p}. \end{aligned} \tag{10.4}$$

Note that each spatial derivative produces a factor $\frac{1}{L_0}$ and the time derivative a factor $\frac{v_0}{L_0}$. The factor $\mathrm{Re} := \frac{v_0L_0\rho}{\mu}$ is known as Reynolds number. It describes the ratio of inertial forces to viscous forces and will be a relevant parameter in the present studies. In the following the $\hat{}$ will be omitted and the notation $\nu := \frac{1}{\mathrm{Re}}$ will be used.

The fundamental theorem of vector calculus[3], states that under some general conditions[4] any vector field can be expressed as a sum of a irrotational (curl-free) and a divergence-free vector field [6, 2]. Further, any divergence-free vector field $\mathbf{v}$ can be written as the curl of an other vector field $\boldsymbol{\omega}$ as

$$\mathbf{v} = \nabla\times\boldsymbol{\omega}. \tag{10.5}$$

Here $\boldsymbol{\omega}$ is a vector potential. In case of the incompressible Navier Stokes equations the irrotational component vanishes due to the continuity equation, Eq.(10.2)/(10.3). Then Eq.(10.5) determines the vorticity $\boldsymbol{\omega}$. Dissipation is only due to the first

[2] Of the earth or more general of planets.

[3] Also known as Helmholtz's theorem.

[4] In a weak formulation, the vector field is only required to be defined on a bounded, simply-connected domain which boundary has to be Lipschitz-continuous.

term on the right hand side of Eq.(10.1). In incompressible, inviscid flows the vorticity $\omega = ||\boldsymbol{\omega}||_2$ and the kinetic energy $\mathcal{E} = ||\frac{1}{2}\mathbf{v}^2||_2$ are conserved [69]. In two-dimensional flows also the enstrophy $\mathcal{V} = \frac{1}{2}\boldsymbol{\omega}^2$ is conserved [69].

Taking the curl and noting that the vorticity $\boldsymbol{\omega}$ is orthogonal to the plane onto which $\mathbf{v}$ is restricted one gets the vorticity-stream function formulation:

$$\frac{\partial\omega}{\partial t} = \nu\nabla^2\omega - \frac{\partial\omega}{\partial x}\frac{\partial\psi}{\partial y} + \frac{\partial\omega}{\partial y}\frac{\partial\psi}{\partial x}, \tag{10.6}$$

where ω is the pseudo-scalar vorticity, i.e. the modulus of the vorticity $\boldsymbol{\omega}$ and ψ the stream function. The vorticity and the stream function are related via the Poisson equation

$$\nabla^2\psi = -\omega. \tag{10.7}$$

Periodic boundary conditions in both spatial dimensions were used in the following. The spatial discretisation is done by a spectral method [52], see also Section 3.5.4, i.e. a finite set of Fourier modes serve as ansatz-functions for the discretised solution. The time integration is performed by a third order stiffly stable operator splitting method as proposed in [67].

10.2. Model Problem

To compare the reduction methods the process of the merging of two adjacent vortices is used as example. For the non-viscous case this process has been studied e.g. in [90]. The initial conditions used in the following are a superposition of two equal signed vortices, each given by

$$\begin{aligned} \omega_\pm(x,y) &= \tfrac{1}{2}\varsigma_0\left(1-\tanh\sqrt{\left(x-\tfrac{1}{2}\right)^2+\left(y\pm d_h-\tfrac{1}{2}\right)^2}\right), \\ & \qquad x,y\in[0,1], \end{aligned} \tag{10.8}$$

where ς_0 denotes the initial maximal vortex intensity which is always set $\varsigma_0 = 1$ and $2d_h$ the initial distance of the vortex centres which is always $d_h = 0.15$. Thus the initial condition is $\omega_0 := \omega(t_0,x,y) = \omega_+(x,y) + \omega_-(x,y)$. This is also an example for deterministic initial conditions.

In the case of the incompressible flow the ordinary differential equation (ODE) system resulting from the spatial discretisation of Eq.(10.6) and Eq.(10.7) can be written as

$$\frac{\partial}{\partial t}\hat{\omega}_i = \Delta_{i,j}\hat{\omega}_j + J_{i,j,k}\hat{\omega}_j\hat{\psi}_k \tag{10.9}$$

$$\Delta_{i,j}\hat{\psi}_j = \hat{\omega}_i. \tag{10.10}$$

Here $\hat{\cdot}$ denotes discrete variables and use of Einsteins sum convention is made. If the basis B is already determined, the equations for the reduced dynamics have the same form but with the effective operators and initial conditions

$$\begin{aligned} \tilde{\Delta}_{i,j} &:= B_{\alpha i}\Delta_{\alpha\beta}B_{\beta j} \quad (10.11) \\ \tilde{J}_{i,j,k} &:= B_{\alpha i}J_{\alpha\beta\gamma}B_{\beta j}B_{\gamma k} \quad (10.12) \\ \tilde{\omega}_{0i} &:= B_{\alpha i}\omega_{0\alpha} \quad (10.13) \end{aligned}$$

10.3. Numerical Integration

Finite resolution tends to lead to an instability of the numerical solution schemes for the Navier-Stokes equations. In [90] this is mitigated by an artificial hyperviscosity term. A spectral discretisation and a third order operator-splitting scheme as proposed in [67] is used. The accuracy of all reduction methods has shown to decrease significantly for larger Reynolds numbers. Therefore these studies are restricted to comparatively low Reynolds numbers of Re ≤ 800. The integration schemes themselves are described in the previous sections.

10.4. Numerical Results

The flow described above was analysed for lattice sizes of typically 48×48 and up to 72×72. This resolution is considerably low for studies of such types of problems. Nevertheless the small system size makes the calculations fast and flexible and also mitigates the need to optimise the efficiency of the algorithms considered here. This also reduces possible error sources.

Note, that always the spectral variant of the variational POD algorithm was used, unless otherwise stated explicitely. This was done by reason of the higher stability of the spectral variant.

Snapshots of the Flow

To give a qualitative idea of the merging process a series of snapshots of the vorticity field for Reynolds number Re $= 400$ were included. For lower values of Re the influence of friction increases, leading to a faster decrease of the vorticity and a 'less interesting' dynamics. Fig. 10.1 and Fig. 10.2 show the time evolution of the vorticity in three-dimensional plots, contour plots and the corresponding velocity field. During the simulation both vortices merge after encircling each other for about $\frac{2}{3}$ rotations leaving a large vortex with some additional structure.

In the following the absolute or L^2-error of the reduced and full simulation is considered. Although this is an appropriate measurefor the quality of a reduction, the L^2-norm of the full solution is also relevant as it allows for a comparison with the L^2-error. For this reason the L^2-norm of the full solution is shown in Fig. 10.3 for Reynolds numbers from Re $= 100$ to Re $= 800$.

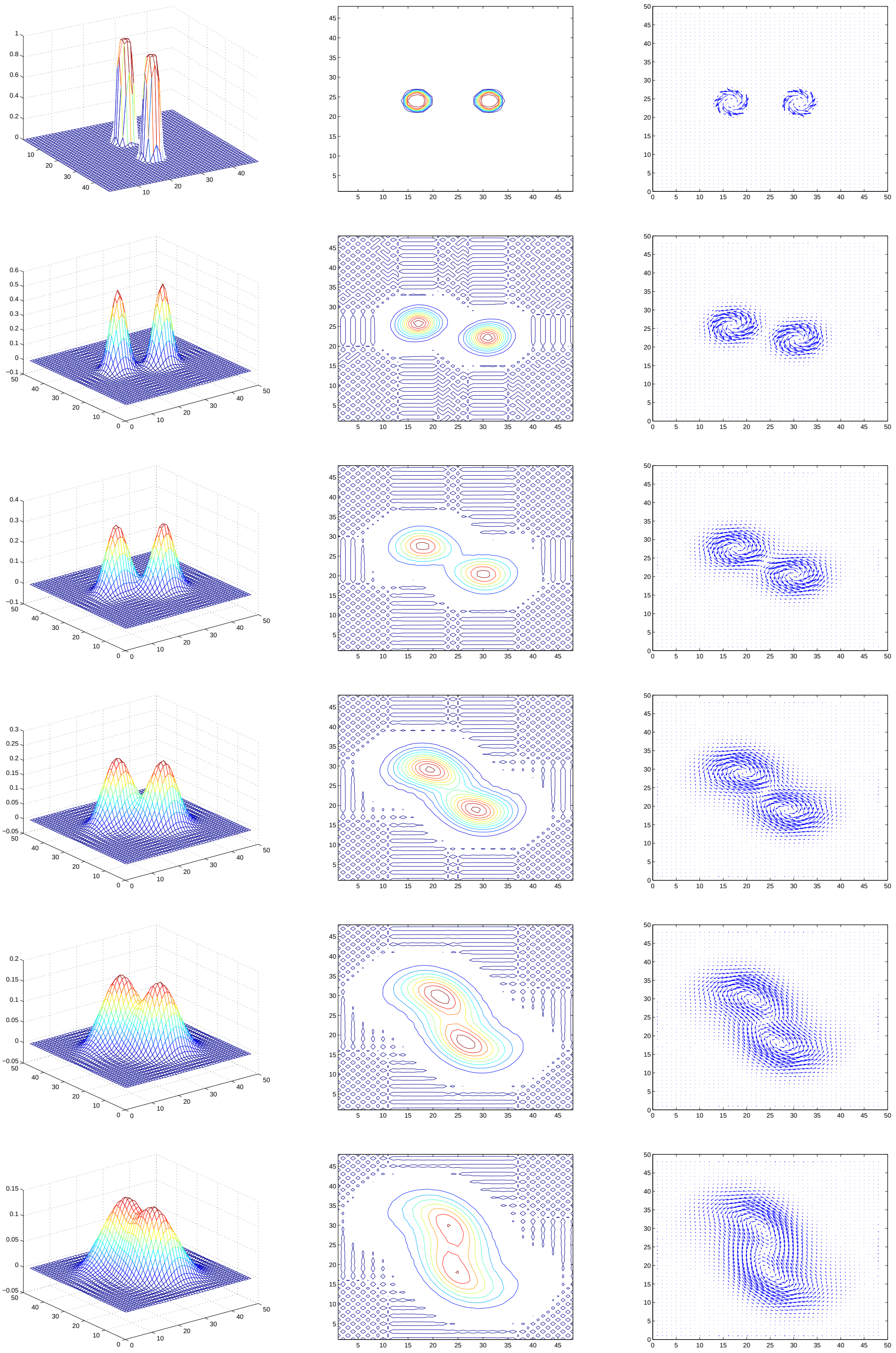


Figure 10.1.: Vorticity for the 48×48 system at Re = 400 after 1, 100, 200,300, 400 and 500 time steps as three-dimensional plot (left column), as contour plot (middle column) and the corresponding vector field (right column).

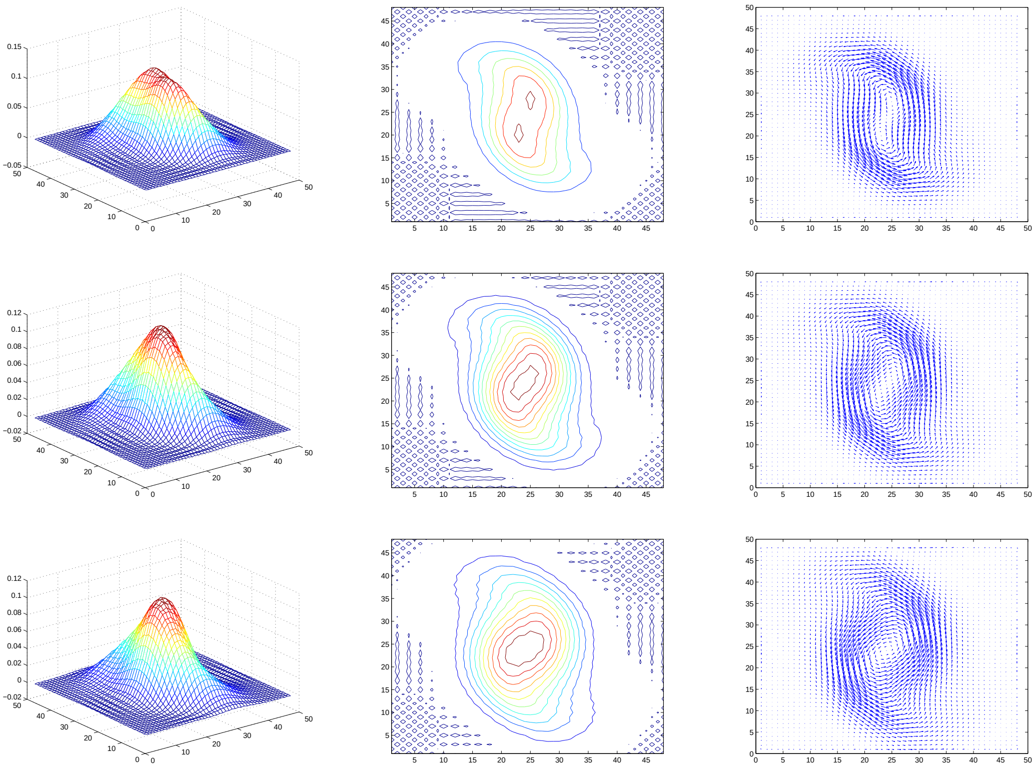


Figure 10.2.: Vorticity for the 48 × 48 system at Re = 400 after 600, 700 and 800 time steps as three-dimensional plot (left column), as contour plot (middle column) and the corresponding vector field (right column).

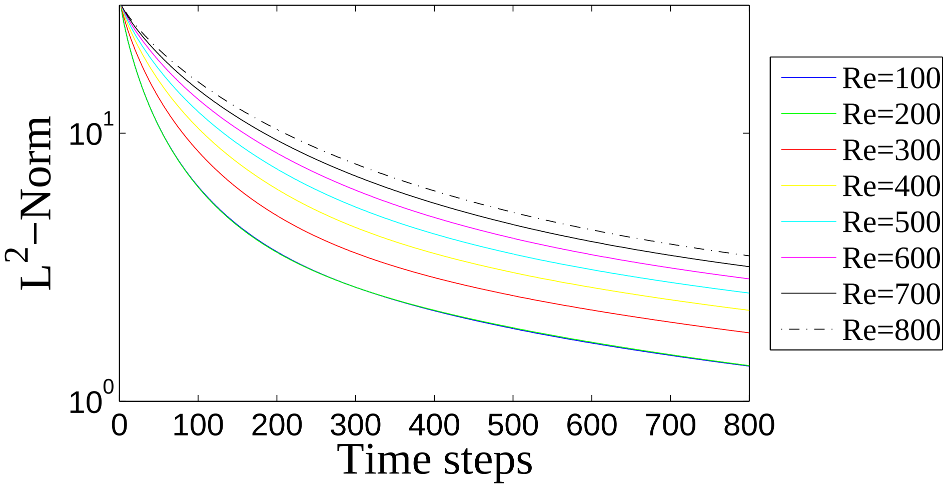


Figure 10.3.: L^2-norm of the full solution for several values of the Reynolds number.

10.4.1. Comparing the Accuracy

Effect of the Reynolds Number

To analyse the accuracy the L^2-error of the difference between the reduced field and the field using all degrees of freedom is calculated, as in the previous section. Doing the same for a basis of POD modes or Fourier modes one has a direct measure to compare these methods. Fig. 10.4 shows the results for a 48×48 lattice, a time step of $\Delta t = 0.25$ and a Reynolds number range from 100 to 800. For the variational algorithm the number of retained states M as well as the number of trial states M_{patch} was $M = 6$, $M_{\text{patch}} = 6$. The simulation time was so long that a final state with a single broad vortex was reached, compare also Fig. 10.1 and Fig. 10.2.

A decrease in performance is observed for increasing Reynolds number for all methods. For the Fourier mode reduction this is most systematical. For low Reynolds number the Fourier mode reduction is also superior to the other methods. The full POD reduction gives very similar performance for Reynolds numbers Re ≥ 400.

The variational POD shows a slight advantage for Reynolds numbers of approximately Re $= 200$ to Re $= 400$. Up to Reynolds numbers of Re $= 600$ it is comparable with the full POD.

The Fourier modes show a very large error for small times. There, most of the non-diffusive dynamics happens. The initial conditions are also very localised and therfore only poorly reproduced by a few Fourier modes. For long times the vorticity has a very broad maximum which is well reproduced by the low frequency Fourier modes. As in the previous chapter for the POD-DMRG method one finds the variational POD even superior to the full POD in a narrow Reynolds number domain. One remarkable feature of these results is the relative poor performance of both POD methods compared to the simple Fourier mode reduction. this is due to the fact that for the investigated flow the long time behaviour is dominated by a broad maximum. This state is also reached by the merging of two vortices much broader than in the correct initial conditions. The Fourier mode reduced dynamics describes qualitatively such a process and neglects fine details. On the other hand, the considerations of detail of the transient states in both POD reductions leads to a poorer performance in reconstructing the long time behaviour.

Effects of the Sweeps

The aim of the sweeps is to increase the accuracy of the reduced model. The corresponding error calculations are shown in Fig. 10.5. It can be stated that the desired result is obtained only for the Reynolds number Re $= 400$ which lies also in the domain where the variational POD performs best. In the other cases the sweeps may even decrease the accuracy. This is clearly undesirable, however the source of this behaviour is yet unknown.

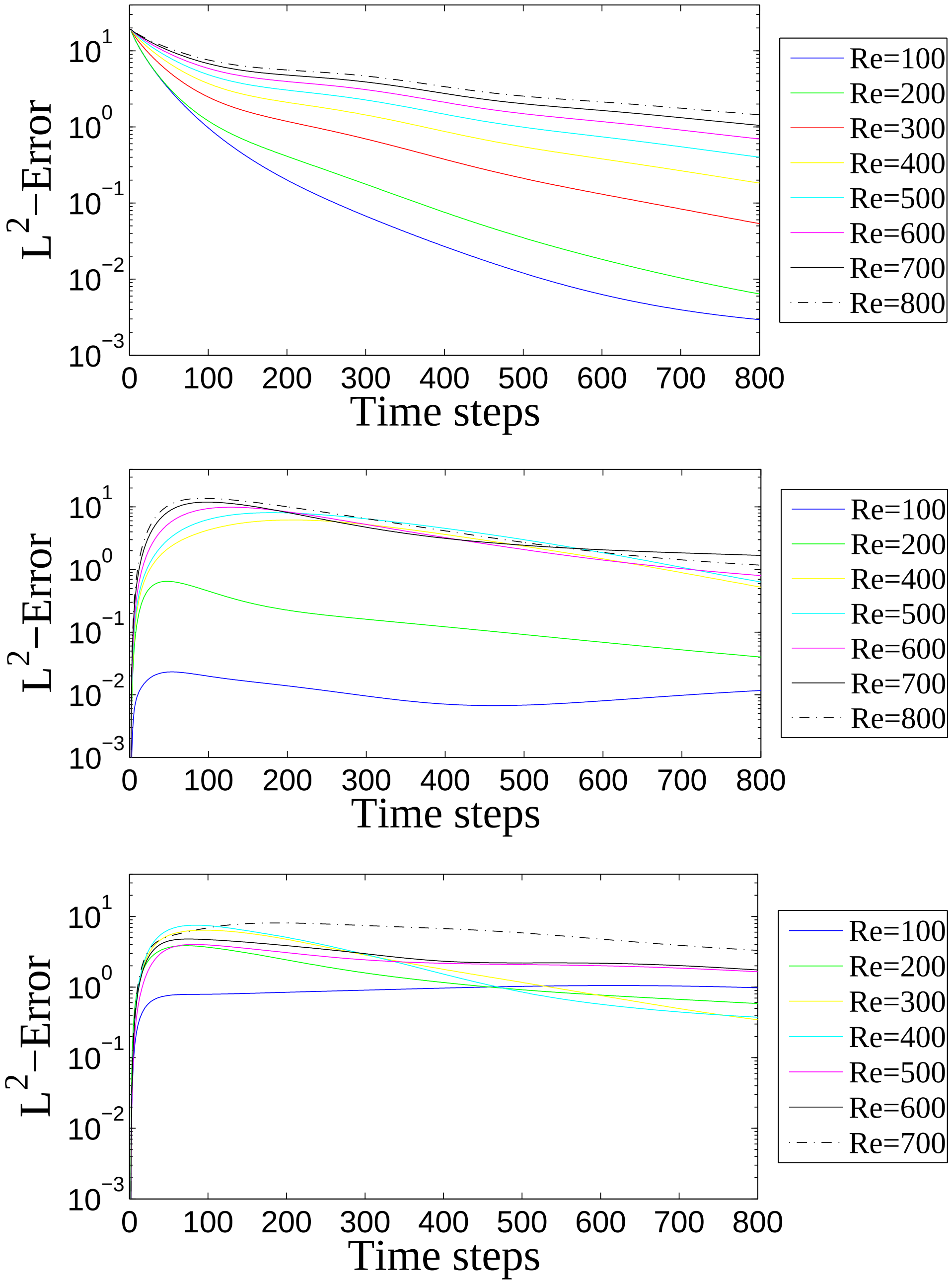


Figure 10.4.: L^2-error for Re $=100$ to Re $=800$, three iteration runs, $\Delta t = 0.25$, $\nu = 0.5$, $M = 6$, $M_{\text{patch}} = 6$ and 800 time steps. Fourier mode reduction (top), full POD reduction (middle) and variational POD reduction (bottom).

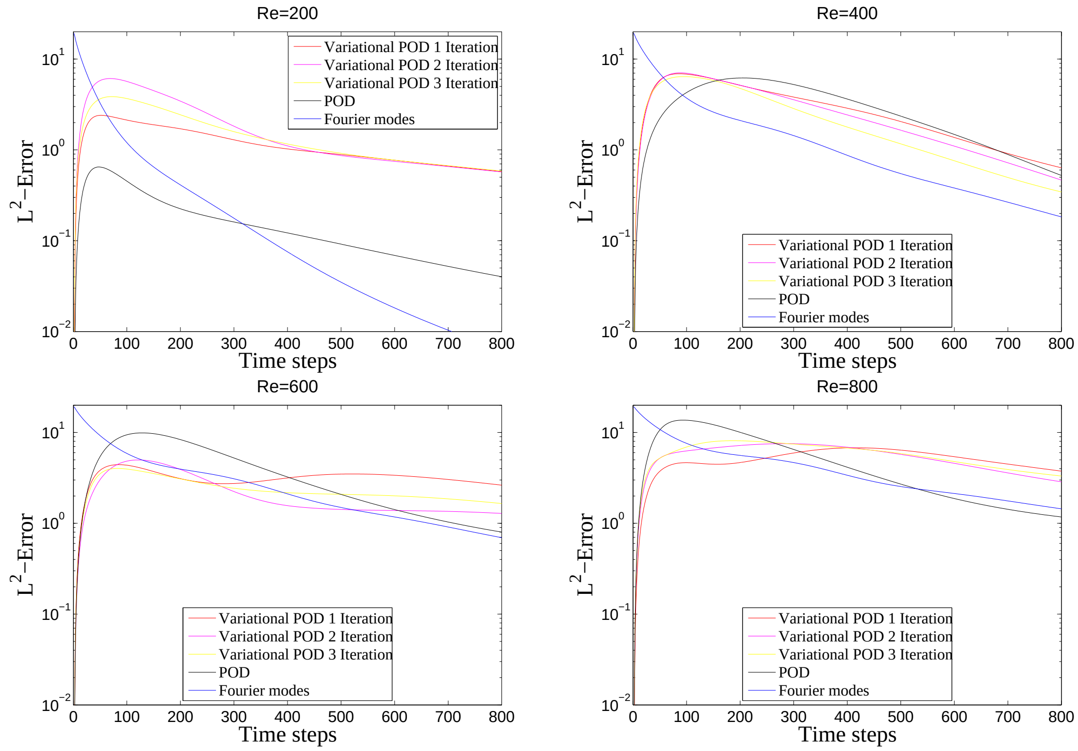


Figure 10.5.: The L^2-error for the 48×48-dimensional system with Re $= 200$ (left top), Re $= 400$ (right top), Re $= 600$ (left bottom) and Re $= 800$ (right bottom). The time step size was $h_t = 0.25$ and the number of retained and trial states $M = M_{\text{patch}} = 6$.

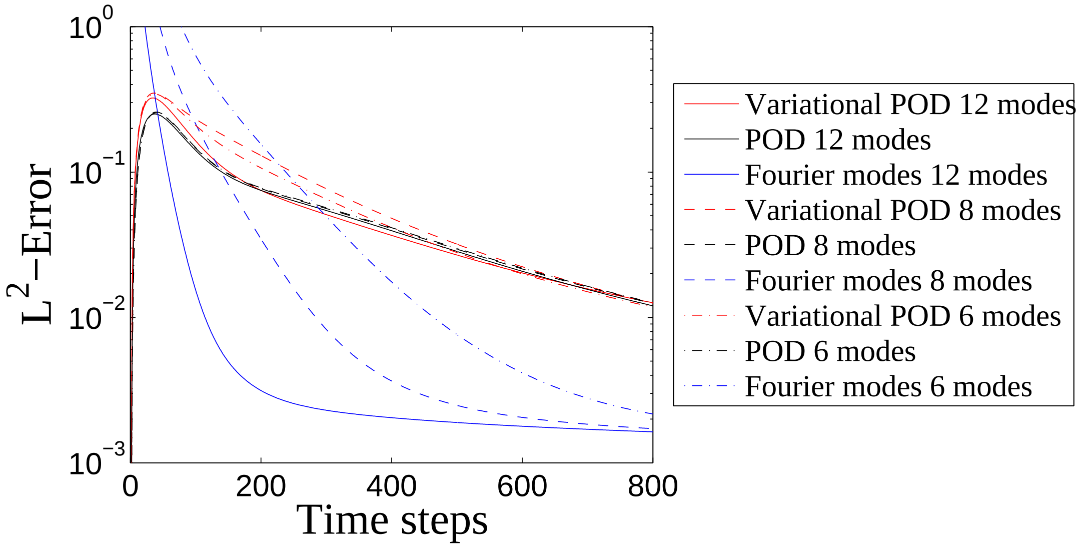


Figure 10.6.: L^2-error for Re $=$ 150, comparing $M = 6$, $M_{\text{patch}} = 6$, $M = 8$, $M_{\text{patch}} = 8$ and $M = 12$, $M_{\text{patch}} = 12$ for the numbers of retained and trial states. Three iteration runs with the spectral variant of the variational POD method were used. $h_t = 0.25$, $\nu = 0.5$ and 3200 time steps.

Effects of Different Numbers of Retained States

The number of retained states M determines the dimensionality of the reduced system and affects therefore the accuracy of the reduced model directly. The number of trial states M_{patch} was chosen equal to M. To compare the performance of the different modes the system was simulated using $M \times M = M^2$ modes. The L^2-error to the full simulation was calculated. The result for the Reynolds number Re $=$ 150 is shown in Fig. 10.6. Only a marginal reduction of the error was observed for increasing M for the full POD method. This would be expected if already a few POD modes are sufficient to describe the dynamics efficiently. However, the performance of the Fourier mode basis is for significant time spans superior to that of the POD modes. For the Fourier mode reduction itself one observes a very systematic increase of the accuracy with M. Therfore one can conclude that the Fourier modes of the lowest 12×12 wave numbers[5] are all relevant for the dynamics. Especially the initial conditions are very localised so that many Fourier modes are necessary for a good approximation. The variational POD modes show a tendency to a poorer performance than the full POD results. This occurs approximately in the time domain in which the POD results are also superior to the Fourier mode reduction. Further a decrease of accuracy was observed when increasing M from $M = 6$ to $M = 8$. This is surprising and currently no complete explanation is available. The choice of M and M_{patch} affects

[5] As the spatial domain, the corresponding Fourier space is two-dimensional.

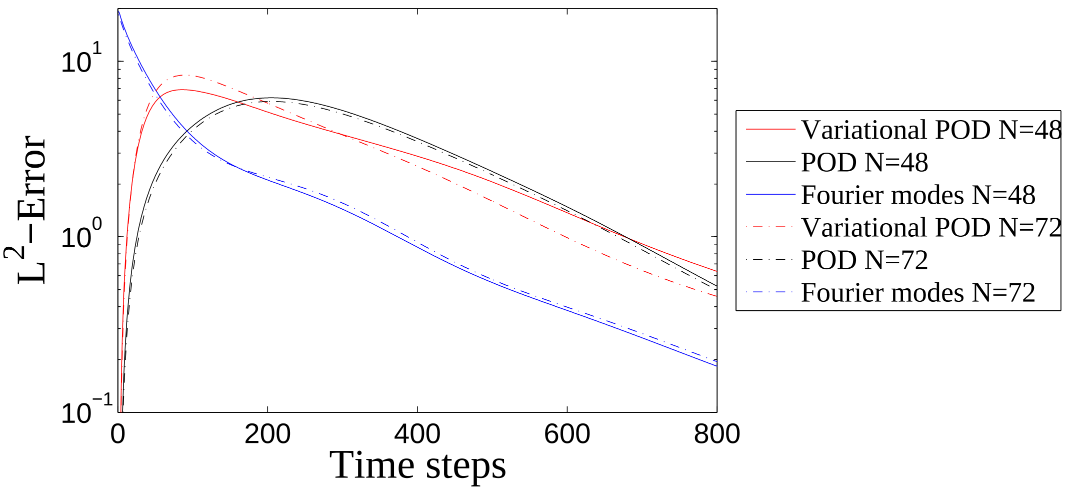


Figure 10.7.: Comparison of the L^2-error for a 48×48 and a 72×72 spatial grid. The numbers of retained and trial states are $M = 6$, $M_{\text{patch}} = 6$, the Reynolds number $\text{Re} = 400$ and the number of time steps $h_t = 0.25$. A single iteration run with 800 time steps were performed.

directly the calculation of the modes for the variational POD method in contrast to the Fourier or full POD modes. Additional modes in the variational POD method can then in principle contribute to numerical artifacts instead of increasing the quality of the approximation. However, the expected increase of accuracy is observed when increasing M further to $M = 12$.

Effect of the Spatial Resolution

The resolution of the lattice clearly determines the accuracy of the unreduced system in describing the partial differential equation of interest. To assess the impact of this parameter on the quality of the reduction calculations on the usual 48×48 grid and on a 72×72 grid were performed. Both lattice sizes are integer multiples of 6×6 and in both cases $M = 6$, $M_{\text{patch}} = 6$ was chosen. By this choice the reduced systems have in all cases the same dimensionality. The other parameters were set exemplarily to $\text{Re} = 400$, $h_t = 0.25$ and the number of time steps to 800. A single iteration run was performed. The results are shown in Fig. 10.7. The effect of increasing the lattice resolution on the Fourier mode and the POD mode reduction is very small. In case of the Fourier modes this is on the one hand due to the fact that the same Fourier modes (albeit with a higher resolution) were used. On the other hand the higher resolution does not lead to a qualitative different behaviour of the unreduced system. Thus one can assume that the resolution is high enough to give a good approximation to the continuous description. This assumption is supported by the very small increase of accuracy for the POD reduction. The results for the variational POD method are also

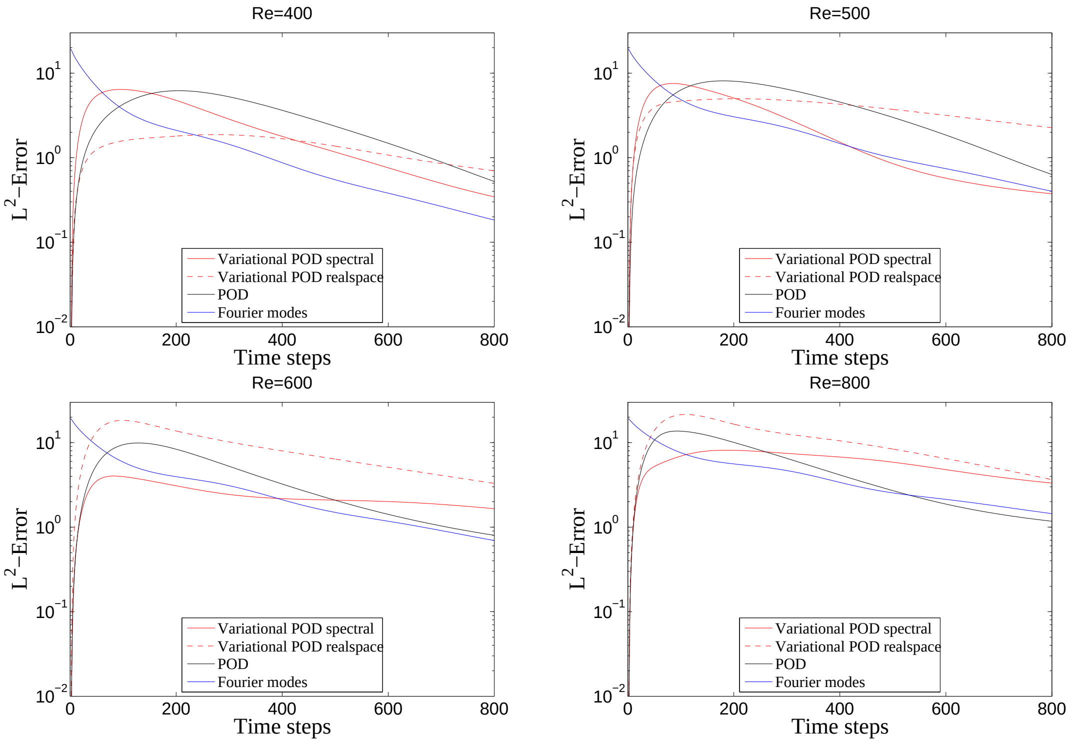


Figure 10.8.: Comparison of the realspace method with the spectral variant of the variational POD method. The L^2-error for the 48×48-dimensional system with Re = 400 (left top), Re = 500 (right top), Re = 600 (left bottom) and Re = 800 (right bottom) is presented. The time step size was $h_t = 0.25$ and the number of retained and trial states $M = M_{\text{patch}} = 6$.

similar, but the maximal error is higher for the increased resolution. For later times after time step 298 the calculations for the lower resolution yield a higher error. Qualitatively, it seems as an additional hump in the error profile for the low resolution is absent in the high resolution result. The mechanisms leading to these differences are not directly accessible. Nevertheless the reason for the larger dependence on the spatial resolution for the variational POD method compared to a Fourier or POD mode reduction is the difference in the choice for the trial states. As stated before, the patches have the same size in both calculations. Consequently, are smaller fraction of the set of all Fourier modes are sampled in each iteration step for a higher spatial resolution. Summarising, one can state that although a small dependence on the spatial resolution for the variational POD method exists the results are still comparable.

Variational POD versus the Spectral Variant

Comparing the performance of both versions of the variational POD one observes that the realspace variant is superior for Re = 400 but leads to higher errors than

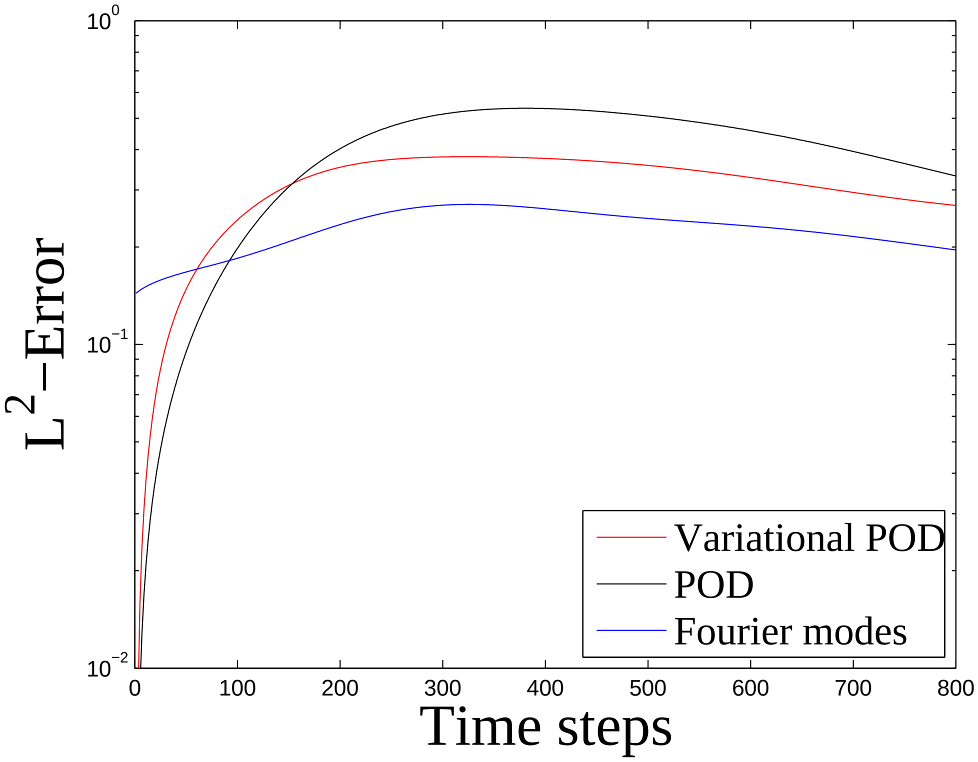


Figure 10.9.: Relative error for Re=400, $M = 6$, $M_{\text{patch}} = 6$, three iteration runs using the spectral variant of the variational POD method. $h_t = 0.25$ and $\nu = 0.5$.

the other method for higher Reynolds numbers. For Re = 500 still the maximum error is smaller as for the other approaches. The reason for this behaviour is not yet clear. For Re = 200 the realspace algorithm even did not converge. Thus the spectral version of the variational POD reduction seems to be preferable to the realspace variant. The spectral variant is also successful in a broader Reynolds number domain, see Fig. 10.4.

Evaluation of the Relative Error

While the L^2-error is certainly a relevant quality measure, also the relative error

$$E_{\text{rel}}(t) := \frac{||\,(\mathbb{1} - P)\,\Phi(t)||_2}{||\Phi(t)||_2}, \tag{10.14}$$

can be of interest. For this reason $E_{\text{rel}}(t)$ was calculated exemplarily for the Reynolds number Re = 400. The other parameters were $M = 6$, $M_{\text{patch}} = 6$, $h_t = 0.25$ and $\nu = 0.5$. The results are presented in Fig. 10.9. Apart from a short time at the beginning of the evolution, the relative error for the Fourier mode reduction is lower than for the other approaches. Also the maximal relative error is smaller for the Fourier mode reduction than for the POD approaches and the Fourier mode reduction shows the smallest variation of the relative error. For later times after time step 153 the variational POD reduction yields a lower error than the full POD reduction. In this time domain the alignment of the errors for the different methods is most counter-intuitive. However, until now the reason for this behaviour is unknown. Also the maximal relative error is lower for the variational POD reduction compared to the full POD reduction. At the beginning of the time evolution the results for the variational POD reduction show the strongest increase. The peak in the error for the Fourier mode reduction at $t = 0$ is absent considering the relative error.

Visualisation of the POD and V-POD Modes

The POD modes themselves can visualise some qualitative aspects of the flow. Therefore the most relevant modes for the full POD and the variational POD are shown in Fig. 10.10, Fig. 10.11, Fig. 10.12 and Fig. 10.13. For the first example with Re = 200 one sees a qualitative agreement with the POD method although the variational POD modes seem to be degraded in some sense. The second example show the results of the realspace method for Re = 400 which was clearly superior to the full POD. Subjectively these modes seem to be more inaccurate than the POD modes. From this one can state that the quality of the reduced basis is not intuitively accessible from the modes themselves.

Visualisation of the Error Evolution

The error for a reduced model is time dependent. This time evolution differs for the different reduction methods. To give some insight in the qualitative behaviour of the L^2-error the time evolution of the error is presented exemplarily in snapshots for the Reynolds number Re = 400 in Fig. 10.14. The error for the variational POD method is less smooth and less symmetric as for the full POD method. Beside this feature both POD methods yield similar results. In particular for the initial conditions the error is very small. In contrast to this the initial error for the Fourier mode reduction is extremely high. Due to the time evolution the error decreases significantly for the Fourier mode reduction below a level of the POD methods. Further the error is less localised for the Fourier mode reduction than for the POD methods.

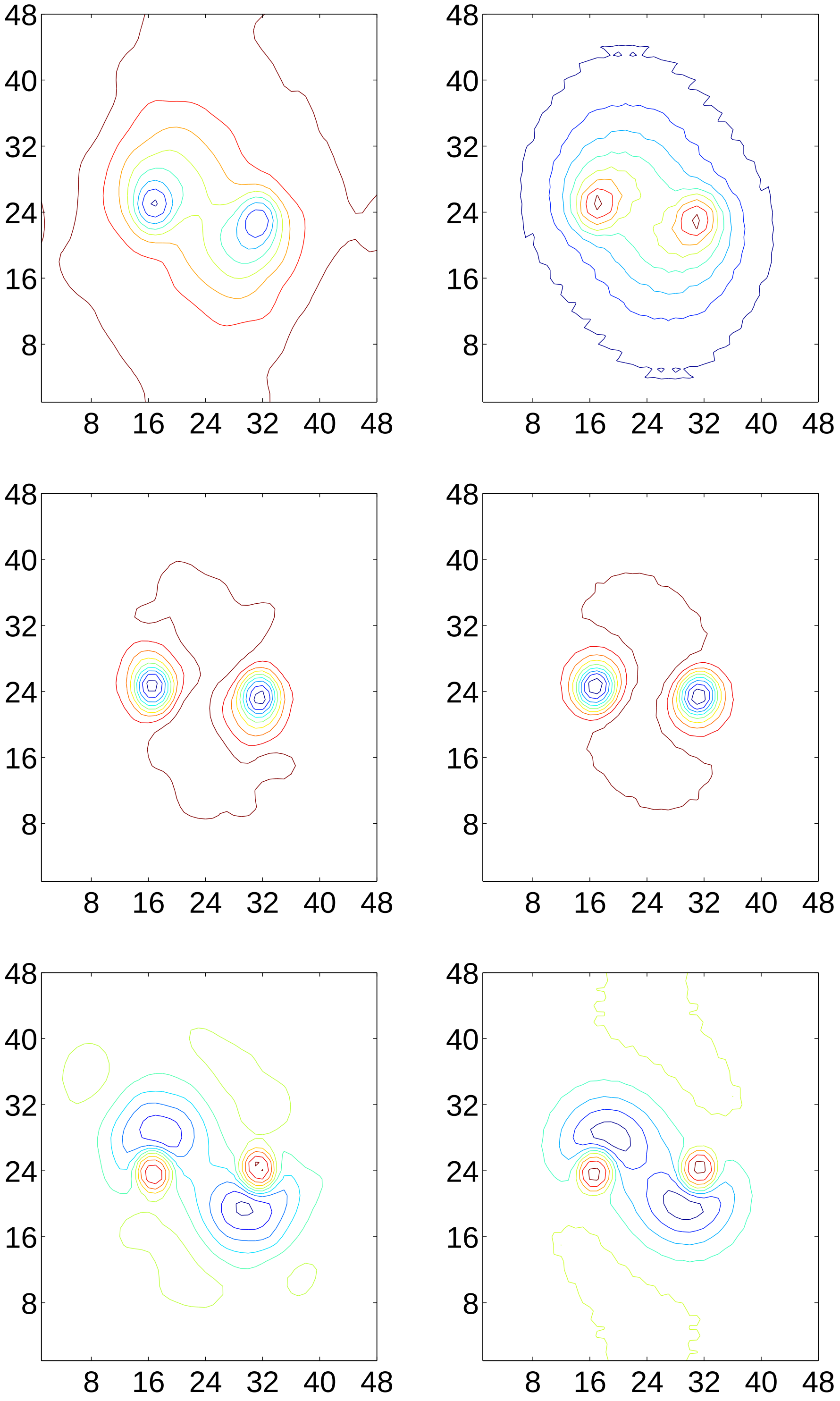


Figure 10.10.: First five POD modes for Re $= 200$, $M = 8$, $M_{\text{patch}} = 8$, three iteration runs using the spectral variant of the variational POD method with $h_t = 0.25$, $\nu = 0.5$ and 3200 time steps. Variational POD modes (left column) versus full POD modes (right column).

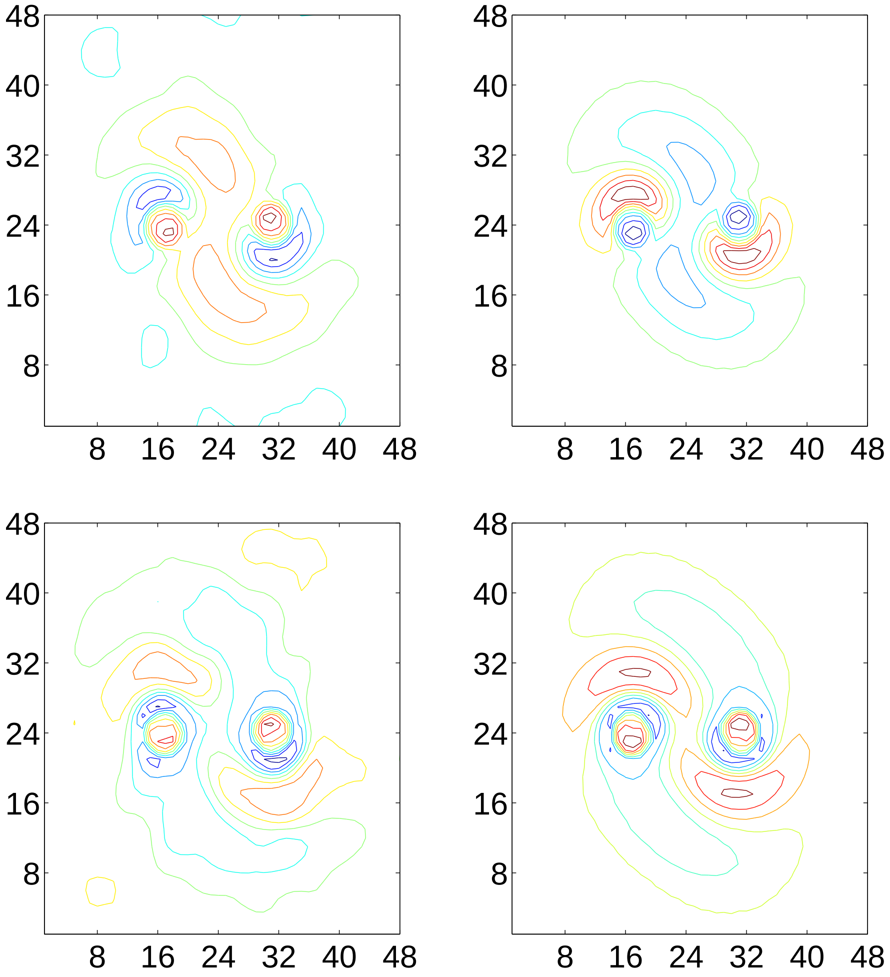


Figure 10.11.: POD modes 4 and 5 for Re = 200, $M = 8$, $M_{\text{patch}} = 8$, three iteration runs using the spectral variant of the variational POD method with $h_t = 0.25$, $\nu = 0.5$ and 3200 time steps. Variational POD modes (left column) versus full POD modes (right column).

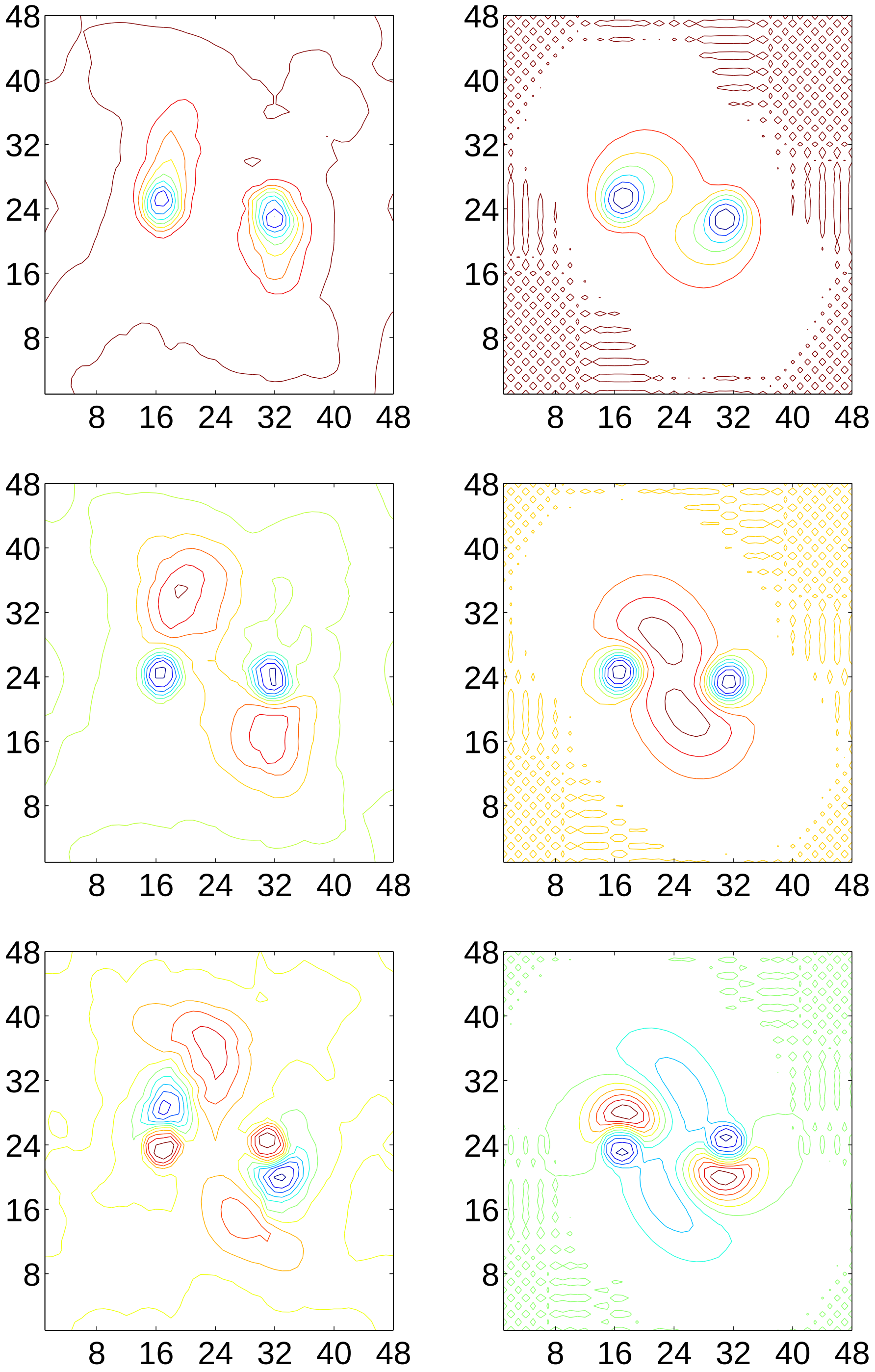


Figure 10.12.: First five POD modes for Re = 400, $M = 6$, $M_{\text{patch}} = 6$, three iteration runs using the real space variant of the variational POD method with $h_t = 0.25$, $\nu = 0.5$ and 800 time steps. Variational POD modes (left column) versus full POD modes (right column).

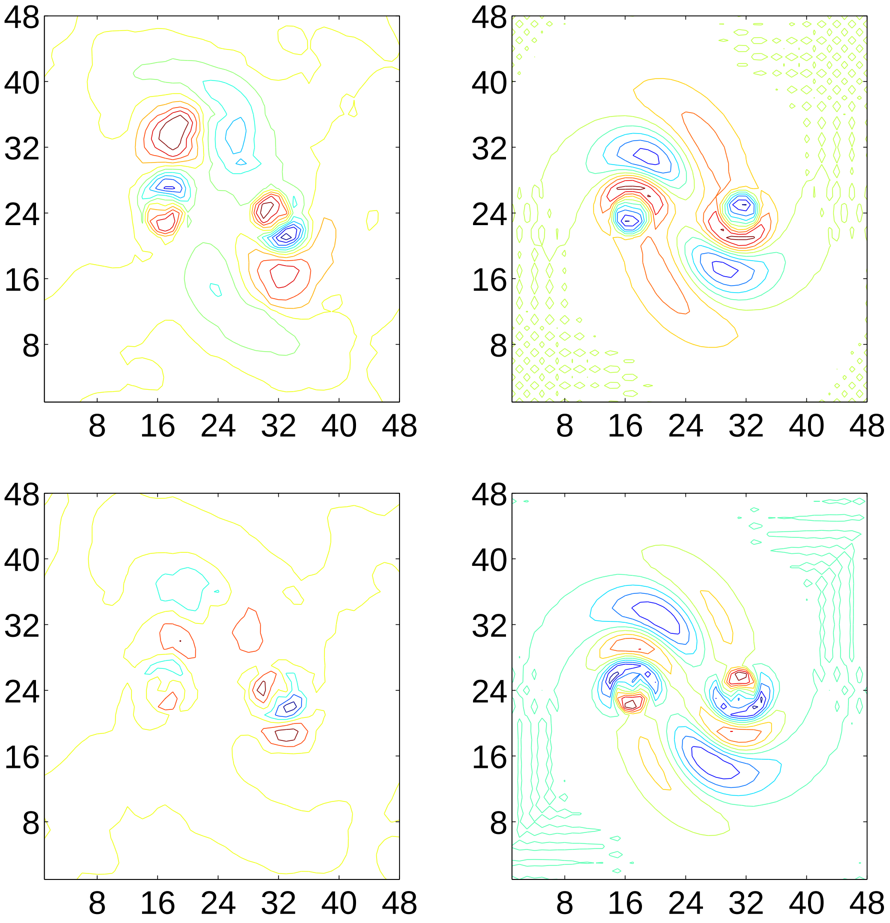


Figure 10.13.: POD modes 4 and 5 for Re $= 400$, $M = 6$, $M_{\text{patch}} = 6$, three iteration runs using the real space variant of the variational POD method with $h_t = 0.25$, $\nu = 0.5$ and 800 time steps. Variational POD modes (left column) versus full POD modes (right column).

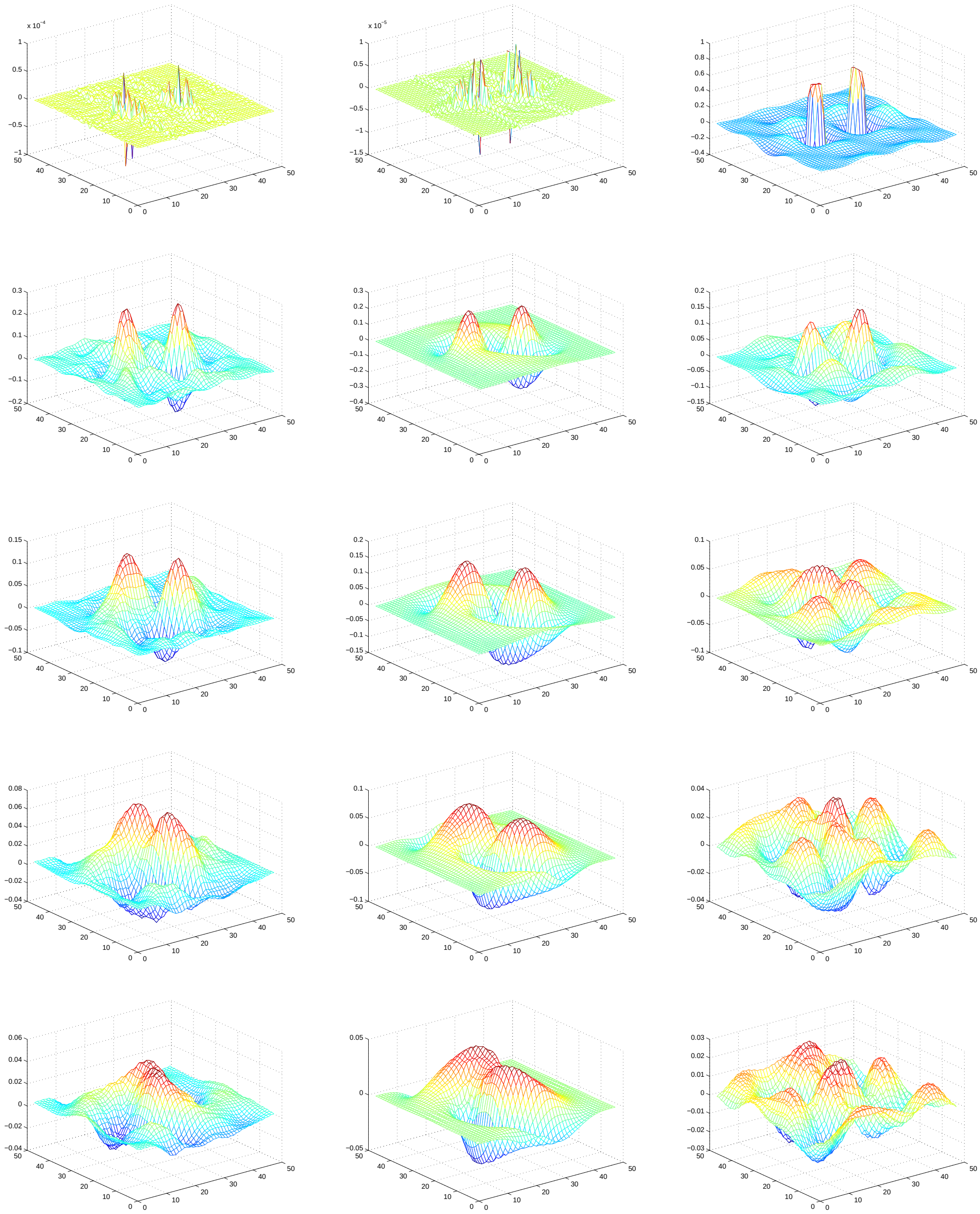


Figure 10.14.: Error for the 48 × 48 system at Re = 400 after 1, 200, 400, 600 and 800 time steps for the spectral variant of the variational POD method (left column), the full POD reduction (middle column) and the Fourier mode reduction (right row). Three iterations with $\Delta t = 0.25$, $M = 6$, $M_{\text{patch}} = 6$.

11. Conclusions

In this work I have pursued several approaches to apply the concepts of density matrix renormalisation to nonlinear dynamical systems. The aim in all applications was to find a 'small', reduced model system with low-dimensional phase space, to describe the 'relevant' dynamics of the complete system in an efficient way. The reduced models considered here were exclusively obtained via a linear projection of the original system. The relevance of a subspace was determined by an error norm based on the L^2-norm.

The examples for the nonlinear dynamical systems range from stochastic models based on microscopic processes in Chapter 8 to nonlinear partial differential equations in Chapter 9 and 3. While the first problem type requires a DMRG approach related to many body DMRG in the formulation at hand, the latter two are treated by schemes based on single particle DMRG. In the following the results of each method are assessed separately.

11.1. Schur DMRG

The stochastic models in the master equation formulation, presented in Chapter 4 were the starting point for this thesis. This approach to dynamical systems, defined essentially by cellular automat rules, is accepted and results in this field exist [77, 106]. DMRG is one way to deal with the high-dimensionality of the resulting master operator. The systems that are assessable with this method are one-dimensional (or quasi one-dimensional). Therefore this ansatz is more or less restricted to fundamental research. The change to Schur vectors instead of eigenvectors, solves some problems e.g. the spurious emergence of nonzero imaginary parts also for real eigenvalues and inaccuracies due to non-orthonormality. Once the number of retained states is chosen sufficiently high, the calculation is stable under the sweeping steps. A higher number of target states also requires a higher number of retained states. Since the use of finite precision arithmetics is unavoidable also the Schur DMRG suffers from numerical inaccuracies. However, it is still possible to obtain meaningful results also for the transient states. It is certainly possible to increase the efficiency and accuracy of the existing algorithm. The real Schur decomposition was performed using the Gnu scientific library, where it was recently added as a new feature. Improvement of the implementations in the future are likely. Also the QR-decomposition in the Schur ordering was performed without pivoting, an aspect that can decrease the accuracy. As a new approach has been proposed also the implementation might not be optimal until now.

Pros

- The method allows to asses the steady state and also the long time behaviour of CA-based 'nonlinear' dynamical systems.
- It completely avoids the problem of complex arithmetics and provides an orthonormal basis for the relevant subspace which is well defined.

Cons

- The systems which can be considered are very simplified, nevertheless they can contain nontrivial physics and their study might give answers to physical questions. Whether DMRG is the right ansatz to study such systems is not clear. For the solution of practical physical problems that arise in technical applications it seems not to be suitable.

Future Prospects

The Schur DMRG method provides a working approach to non-hermitian systems. Due to the restriction to one-dimensional systems the future of this ansatz depends on whether physical relevant questions on systems of this class arise. A description of PDEs within this framework, proposed by J.Rodriguez-Laguna[1], seems to be problematic.

11.2. POD DMRG

The POD DMRG algorithm is a crossover of two comparatively remote fields of numerical analysis. One advantage is that it is very general. The new approach also makes practically no assumption on the equations that define the dynamics. In this work a demonstration of the applicability has been given for this new algorithm to systems described by (1+1)-dimensional nonlinear partial differential equations, namely the Burgers equation and a nonlinear diffusion equation of Fischer type. It is possible to calculate an approximate POD without ever simulating the full system. The method has been tested for linear systems where its performance was even higher than the full system POD results but considerable worse than the optimal reduction. Several nonlinear systems have been considered. For the Burgers equation the results of the full POD and the new algorithm were comparable and both significantly better than a Fourier-mode based reduction. Summarising one can state that the POD-DMRG method yields good results at a reduced computational effort compared to the full POD which is a commonly accepted approach. As for the Schur DMRG method the class of assessable systems is very restricted which limits the practical use of the method.

Pros

- The algorithm can be applied to a very general class of systems.
- It is also faster than standard POD without reducing the quality of the result.

[1] private communications

- The method is iterative, the progress can be monitored during the calculation.

Cons

- Again the restriction to one-dimensional systems limits the practical use significantly.
- Reduced operators are not sparse. This increases the memory consumption.

Future Prospects

Practical applications of the POD-DMRG approach are limited by the restriction to spatially one-dimensional systems. However, there might be some niches where the use of POD-DMRG is advantageous. Nevertheless the main intention for this ansatz was to test the feasibility of a spatial blocking.

11.3. Variational POD

Encouraged from the results of the POD-DMRG algorithm I have tried to extend the approach to higher-dimensional systems. As model system the 2D-Navier-Stokes equations have been chosen. These are numerically more demanding and describe a more realistic system. The algorithm itself is a variational form of the proper orthogonal decomposition. One important point is that it can be applied also to higher-dimensional systems without significant modifications. A more physical, three-dimensional system, e.g. the 3D-Navier-Stokes equations, has not been considered because then a significant amount of time and work would have been necessary to simulate this model and analyse the results. With our means it is nevertheless only possible to evaluate the situation for a particular problem. Success for one problem does not guarantee success for a different system. However it demonstrates the viability of the approach in principle.

The performance of the variational POD was found to be comparable to the full POD. The real space variant has showed an unexplained divergence for low Reynolds numbers Re ≤ 200. For a narrow Reynolds number range the realspace as well as the spectral variant of the variational POD exhibit a performance clearly superior to the full POD and the Fourier mode reduction. Generally, the performance of all POD methods were in many cases inferior to the much simpler Fourier mode reduction. This is a hint that the 2D-Navier-Stokes equations with the considered initial conditions are not an optimal system for a reduction approach. As expected, the spectral variant has shown a higher numerical stability as the realspace variant of the variational POD. Nevertheless, in some cases the accuracy of the realspace variant was even higher than that of the spectral variant. In contrast to Chapter 9 no data on the computational load for the variational POD method was given and compared to the full POD. This has a simple reason. The new approach requires the explicit storing and processing of dense (but comparatively small) matrices and tensors. Calculating a full POD with the same algorithms (for simulating the system e.g.) would be very inefficient and would result in an unrealistic high load for the full POD. If efficiency is an issue, the full POD and the variational POD have very different requirements. Since the variational POD is new method, comparison of the computational load would depend too much on the actual implementation, rather than on the methods themselves.

Pros

- The variational POD can also be applied to a very general class of systems.
- In addition higher-dimensional systems can be analysed. This includes also finite element descriptions.
- Also the variational POD is iterative so that the progress can be monitored during the calculation.

Cons

- Reduced operators are not sparse. This increases the memory consumption.
- The efficiency of the implementation needs to be increased.

Future Prospects

From all three new methods, the variational POD is most suited for further studies. Due to its generality and simple construction it is likely to find some applications. It could also be useful for the understanding of nonlinear PDEs. Extensions of the methods are possible, e.g. the use of finite element methods or, for flow problems, an application to Lattice-Boltzmann methods. A more flexible handling of the degrees of freedom, e.g. would be useful and could also give some insights in the underlying processes. Due to the large class of accessible systems, a systematic study, where the approach is employed most profitably has to be performed.

Appendix A.

Finite Numerical Precision

The mathematical descriptions of the models discussed in this work involve fields or vector spaces over fields. Typically the field is $\mathbb{R}$ or $\mathbb{C}$ which are infinite (even uncountable) sets. For a numerical treatment only a finite description is possible. In contemporary computers these descriptions are based on a binary representation, i.e. a finite sequence of binary digits as $(b_1, \ldots, b_N)$, $b_i = 0/1$. Depending on the purpose some formats are common.

Fixed-Point Representation

Integer numbers are usually represented by the format *int*. This is an example for a fixed-point representation. The number is given in this representation simply by an expansion as

$$(b_1, \ldots, b_N) \to K = (-1)^{b_N} \sum_{i=1}^{N-1} b_i 2^i. \tag{A.1}$$

On the machines used in this work $N = 32$ bit are used for an *int*, although this can differ for other systems. Some modifications are common as *short*, *long*, *unsigned* integer, differing in the number of bits or whether an bit is used for the sign. According to Eq.(A.1) integer numbers from $-2^{32}+1$ to $2^{32}-1$ can be represented. As long as this range is not left, arithmetics for *int* are exact. Clearly operations as e.g. division has to be redefined.

Floating-Point Representation

Real numbers are usually treated in a different way. Commonly used are floating point representations. This is given by a sign bit $\boldsymbol{s}$ and two integers (see above) M, $\boldsymbol{e}$ so that the actual real number to be stored is given by

$$\boldsymbol{s} M \mathcal{B}^{\mathbf{e}-E}. \tag{A.2}$$

Here M is called the mantissa and $\boldsymbol{e}$ the exponent. The additional numbers $\mathcal{B}$, the basis (typically $\mathcal{B} = 2$) and E the exponent bias, are machine dependent and are not stored explicitely. The representation Eq.(A.2) is not unique. Decreasing the exponent and shifting the bit pattern of M to the left does not change the

	float	*double*
$\mathcal{B}$	2	2
Number of base-$\mathcal{B}$ digits in M	24	53
$\frac{\ln \epsilon_M}{\ln \mathcal{B}}$	-23	-52
$\frac{\ln \epsilon_M^-}{\ln \mathcal{B}}$	-24	-53
Number of bits in the exponent	8	11
Smallest power of $\mathcal{B}$ consistent with requiring no leading zeros in M	-126	-1022
Smallest power of $\mathcal{B}$ that causes overflow	128	1024
ϵ_M	1.19209e-07	2.22045e-16
ϵ_M^-	5.96046e-08	1.11022e-16
Smallest usable floating value	1.17549e-38	2.22507e-308
Largest usable floating value	3.40282e+38	1.79769e+308

Table A.1.: Floating number representation on the Intel machine. The rounding was compliant with IEEE standard [3].

represented number. Usually one chooses M and $\boldsymbol{e}$ so that M is shifted to the left maximally (this is termed to be the normalised representation). Then the bit pattern of M always starts with a 1 which need not to be stored explicitely giving an extra significant bit.

Unlike integer numbers, real numbers are not represented exactly. Further the arithmetics is not exact, even if the processed numbers have been represented exactly. The machine precision ϵ_M is the smallest number which can be added to 1.0 still yielding a result $\neq$ 1.0. An alternative definition, denoted here by ϵ_M^-, is the smallest number which can be subtracted to 1.0 still yielding a result $\leq$ 1.0. We use the format *double* which was initially developed from so called single precision (usually 32 bit) to achieve greater accuracy. It is a 64 bit representation and the machine precision for *double*-varibles was $\epsilon_M = 2.22045e - 16$ [1] on our machines[2]. Due to the advances in hardware technology this standard has more or less replaced the single precision format. The value of ϵ_M depends on the number of bits available for M. The smallest and largest number representable depends on the number of bits available for $\boldsymbol{e}$. To summarise, in most cases the finite precision operations yield the correct results with some small round-off error. Some operations can lead to completely different results. An example is the subtraction of two almost equal numbers. The result is defined by the few bits differing, resulting in a low accuracy. Also adding up a large number of summands can lead to problems. Once the ratio of sum and summand has reached approximately ϵ_M the result is not affected by the summing up any more.

[1] This value was obtained by the routine *machar* from the numerical recipes [96]. A complete list is given in Table A.1.

[2] The calculations were actually performed on a Dell Intel Dual Core and a Extensa 2900Lmi notebook with Pentium M processor.

Stability

While the rounding errors cannot be avoided completely also some methods can magnify errors and lead to completely wrong results. One example is the explicit Euler method which becomes unstable for large integration steps. If the stability criterion Eq.(3.20) is violated but still valid for the lower part of the spectrum of the generator of evolution, the integration is formally still correct provided the solution does not contain contributions from the problematic eigenvectors. For the Laplace operator this is even reasonable for smooth solutions. However, in practice finite accuracy always produces such contributions. Thus the calculation is incorrect although with arbitrary accuracy the method would yield the correct result.

Appendix B.

Optimal Reduction of Linear Systems

For completeness, we assess in the following the error of the reduced evolution for the linear case. For the optimal reduction we require a minimal L^2-error for the reduced field with respect to the unreduced evolution. The full time evolution in the N dimensional phase space is generated by L as

$$\Phi(t) = e^{(t-t_0)L}\Phi(t_0). \tag{B.1}$$

The explicit Euler algorithm approximates this by

$$\Phi(t) \approx (\mathbb{1} + \Delta t L)\,\Phi(t_0). \tag{B.2}$$

We assume that all eigenvalues of L are negative or zero. A positive eigenvalue would lead to an unbounded exponential growth in Eq.(B.1) which is unphysical. Considering only linear projections the reduction is defined by the operator P which is the orthogonal projection to the relevant subspace Range(P). P can be constructed from an orthonormal basis (ONB) of this space. Equivalently, it can be defined via the ONB (namely C) of Kern(P) so that $P = \mathbb{1} - CC^\dagger$.

The reduced time evolution becomes

$$\hat{\Phi}(t) = e^{(t-t_0)PLP}P\Phi(t_0) = e^{(t-t_0)\hat{L}}\hat{\Phi}(t_0), \tag{B.3}$$

since after each (infinitesimal) time step the components within the irrelevant subspace, i.e. Kern(P) are projected out. For a general P the eigenvectors of $\hat{L}$ are not the same as for L, but known eigenvectors of $\hat{L}$ are always the column vectors of C.

B.1. Long Time Optimised Projection

If we assume that the eigenvalues of L are ≤ 0, for long times $t \gg 1$ the time evolution operators $e^{t\hat{L}}, e^{tL}$ become the projectors onto the kernels of L or $\hat{L}$, respectively. In the eigenbasis ψ_i^{eig} it is simply

$$\psi_i^{eig} e^{tL} \psi_j^{eig} = \delta_{ij} e^{t\lambda_i}. \tag{B.4}$$

The product of the reduced evolution operator e^{tPLP} and P converges for long times to the projector onto $\mathrm{Kern}(PLP) \cap \mathrm{Range}(P)$. More explicitly this is

$$\begin{aligned}\lim_{t\to\infty} e^{tL} &= \mathbb{1}|_{\mathrm{Kern}(L)}\,, &\text{(B.5)}\\ \lim_{t\to\infty} e^{tPLP} &= \mathbb{1}|_{\mathrm{Kern}(PLP)}\,. &\text{(B.6)}\end{aligned}$$

This gives for the error

$$E_\infty = \lim_{t\to\infty} E\Phi(t) = \mathbb{1}|_{\mathrm{Kern}(PLP)}\,P - \mathbb{1}|_{\mathrm{Kern}(L)}\,. \tag{B.7}$$

In the long time limit we can obtain a zero error for all initial conditions if we have

$$\mathrm{Kern}(PLP) \cap \mathrm{Range}(P) \equiv \mathrm{Kern}(L). \tag{B.8}$$

This is achieved by requiring

$$\begin{aligned}&\mathrm{Kern}(L) \subset \mathrm{Range}(P), &\text{(B.9)}\\ \text{and}\quad &\mathrm{Range}(P)\quad L\text{-invariant}, &\text{(B.10)}\end{aligned}$$

as we show now.

Consider a $\phi \in \mathrm{Range}(P)$. Then $P\phi = \phi$ and due to the L-invariance of $\mathrm{Range}(P)$ it is $L\phi \in \mathrm{Range}(P)$ resulting in $PLP\phi = PL\phi = L\phi$. This gives for P with $\mathrm{Range}(P)$ being L-invariant

$$\mathrm{Kern}(PLP) \cap \mathrm{Range}(P) = \mathrm{Kern}(L) \cap \mathrm{Range}(P). \tag{B.11}$$

Eq.(B.8) can be retrieved from Eq.(B.11) just by requiring condition (B.9). Thus, in the long time limit Eq.(B.7) becomes identically zero.

B.2. Short Time Optimised Projection

For short times we consider here the reduction from a N-dimensional to a $(N-1)$-dimensional system.For further reductions the results can be applied by iteration. The projector P becomes then $P_{ij} = \mathbb{1}_{ij} - c_i c_j$ where $\mathbf{c}$ is the removed state. In order to minimise the error for the short time evolution measured by the L^2-norm we have to minimise

$$\begin{aligned}E_s(t) &= ||e^{tL}\phi - e^{PLP}P\phi||_2 &\text{(B.12)}\\ &\approx ||\left(\mathbb{1} + tL - P - PLP\right)\phi||_2 = ||E\phi||_2.\end{aligned}$$

Here, we have already used an expansion in powers of t and truncated after the first order terms.

Since we have no information on ϕ, we minimise Eq.(B.12) by using the Frobenius norm $|\cdot|_F$ of the error operator E. The Frobenius norm is consistent with the L^2-norm [50], i.e.

$$||Ax||_2 \leq |A|_F||x||_2 \quad \forall A \in R^{n\times n}, x \in R^n. \tag{B.13}$$

By inserting $P = \mathbb{1} - C$ we get for the error operator

$$\begin{aligned} E = & \quad \mathbb{1} - P + t(L - (\mathbb{1} - C)L(\mathbb{1} - C)) \\ = & \quad C + t(L - L + LC + CL - CLC) \\ = & \qquad C + t(LC + CL - CLC). \end{aligned} \tag{B.14}$$

We assume L to be symmetric, i.e. $L_{ij} = L_{ji}$. Thus L has an orthonormal eigenbasis $\{\varphi_{i\alpha}\}_{\alpha=1\ldots N}$ where the columns are the eigenvectors of L. The eigenvalues are λ_α and the matrix elements of the error operator E are decomposed in this basis as

$$\begin{aligned} E_{ij} &= \sum_{\alpha\beta} \varphi_{\alpha i} E_{\alpha\beta} \varphi_{\beta i} \\ &= C_{ij} + t\sum_n \left(L_{in}C_{nj} + C_{in}L_{nj} - \sum_m C_{in}L_{nm}C_{mj} \right) \end{aligned} \tag{B.15}$$

with

$$C_{ij} = \sum_{\alpha\beta} \varphi_{\alpha i} c_\alpha c_\beta \varphi_{\beta j}, \tag{B.16}$$

$$\sum_{mn} C_{in}L_{nm}C_{mj} = \sum_{\alpha\beta nm} \varphi_{\alpha i} c_\alpha c_n L_{nm} c_m c_\delta \varphi_{\beta j}, \tag{B.17}$$

$$\sum_n L_{in}C_{nj} = \sum_{\alpha\beta n} \varphi_{\alpha n} L_{in} c_\alpha c_\beta \varphi_{\beta j}, \tag{B.18}$$

$$\sum_n C_{in}L_{nj} = \sum_{\alpha\beta n} \varphi_{\alpha i} c_\alpha c_\beta L_{nj} \varphi_{\beta n}. \tag{B.19}$$

We use the orthogonality of the φ_α, i.e.

$$\sum_i \varphi_{\alpha i}\varphi_{\beta i} = \delta_{\alpha\beta} = \sum_i \varphi_{i\alpha}\varphi_{i\beta}, \tag{B.20}$$

and the definition of the eigenvalues λ_α

$$\sum_j L_{ij}\varphi_{j\alpha} = \lambda_\alpha \varphi_{i\alpha}. \tag{B.21}$$

In the eigenbasis the removed degree of freedom $\mathbf{c}$ can be written as $\tilde{\mathbf{c}}$ with components

$$\tilde{c}_i = \sum_\alpha \varphi_{\alpha i} c_\alpha \quad , \quad c_\beta = \sum_{i\alpha} \varphi_{\alpha i}\varphi_{\beta i} c_\alpha = \sum_i \varphi_{\beta i}\tilde{c}_i. \tag{B.22}$$

The average of L in the removed state $\mathbf{c}$, is

$$\begin{array}{rclcl} \langle L\rangle_{\mathbf{c}} & := & \sum\limits_{nm} c_n L_{nm} c_m & = & \sum\limits_{nmij} \varphi_{ni}\tilde{c}_i L_{nm}\varphi_{mj}\tilde{c}_j \\ & = & \sum\limits_{nij} \varphi_{ni}\tilde{c}_i \lambda_j \varphi_{nj}\tilde{c}_j & = & \sum\limits_{i} \tilde{c}_i^2 \lambda_i. \end{array}$$

The matrix elements from Eq.s(B.16-B.19) become

$$\begin{aligned} C_{ij} &= \tilde{c}_i\tilde{c}_j, && \text{(B.23)} \\ \sum_{mn} C_{in}L_{nm}C_{mj} &= \tilde{c}_i \langle L\rangle_{\mathbf{c}} \tilde{c}_j, && \text{(B.24)} \\ \sum_{n} L_{in}C_{nj} &= \sum_{\alpha n} \varphi_{\alpha i} L_{\alpha n} c_n \tilde{c}_j \\ &= \sum_{n} \lambda_i \varphi_{ni} c_n \tilde{c}_j \\ &= \lambda_i \tilde{c}_i \tilde{c}_j, && \text{(B.25)} \\ \sum_{n} C_{in}L_{nj} &= \sum_{n} \tilde{c}_i c_n \lambda_j \varphi_{nj} \\ &= \tilde{c}_i\tilde{c}_j\lambda_j. && \text{(B.26)} \end{aligned}$$

Thus for the matrix elements of the error operator we obtain

$$E_{ij} = \tilde{c}_i\tilde{c}_j \left(1 + t\left(\lambda_i + \lambda_j - \langle L\rangle_{\mathbf{c}}\right)\right). \tag{B.27}$$

We minimise the Frobenius norm of E given by

$$|E|_F = \sum_{ij} |E_{ij}|^2 = \sum_{ij} \tilde{c}_i^2\tilde{c}_j^2 \left(1 + t\left(\lambda_i + \lambda_j - \langle L\rangle_{\mathbf{c}}\right)\right)^2 \tag{B.28}$$

for a normalised $\mathbf{c}$, i.e.

$$1 = ||c||_2^2 = \sum_i c_i^2 = \sum_i \tilde{c}_i^2. \tag{B.29}$$

Since E is a linear operator it follows that $||E\mathbf{x}||_2 = ||\mathbf{x}||_2||E\hat{\mathbf{x}}||_2$ with $\mathbf{x} = \hat{\mathbf{x}}||\mathbf{x}||_2$. Without any restriction to $||\mathbf{x}||_2$ the zero vector would always minimise $||E\mathbf{x}||_2$. Furthermore, each lower bound K for $||\mathbf{x}||_2$ will lead to the same $\hat{\mathbf{x}}$ with $||\hat{\mathbf{x}}||_2 = K$. This is not true for the general nonlinear case as in [37].

Incorporating this condition $|E|_F$ reduces to

$$\begin{aligned} |E|_F^2 &= 1 + 2t\langle L\rangle_{\mathbf{c}} + 2t\langle L\rangle_{\mathbf{c}} - 2t\langle L\rangle_{\mathbf{c}} + t^2\langle L\rangle_{\mathbf{c}}^2 \\ &+ 2t^2\langle L\rangle_{\mathbf{c}}^2 - 2t^2\langle L\rangle_{\mathbf{c}}^2 + t^2\langle L\rangle_{\mathbf{c}}^2 - 2t^2\langle L\rangle_{\mathbf{c}}^2 + t^2\langle L\rangle_{\mathbf{c}}^2 \\ &= 1 + 2t\langle L\rangle_{\mathbf{c}} + t^2\langle L\rangle_{\mathbf{c}}^2 = \left(1 + t\langle L\rangle_{\mathbf{c}}\right)^2 \\ \Rightarrow |E|_F &= \left|1 + t\langle L\rangle_{\mathbf{c}}\right|. && \text{(B.30)} \end{aligned}$$

Consequently, in order to minimise $|E|_F$ we have to minimise $\langle L\rangle_{\mathbf{c}}$.

The minimisation itself is performed using Lagrangian multipliers for the constraint Eq.(B.29). The necessary condition for a minimum is

$$\begin{aligned} 0 &= \frac{\partial}{\partial \tilde{c}_k} \left(\langle L \rangle_{\mathbf{c}} + \eta \left(1 - ||\mathbf{c}||_2^2\right)\right) \\ &= \frac{\partial}{\partial \tilde{c}_k} \sum_i \left(\tilde{c}_i^2 \lambda_i + \eta \left(1 - \tilde{c}_i^2\right)\right) \\ &= 2\tilde{c}_k \left(\lambda_k - \eta\right). \end{aligned} \tag{B.31}$$

This is true if either $\tilde{c}_k = 0$ or $\eta = \lambda_k$. The last equation can only be true for a single value of λ_k. We denote the nonzero component as $\tilde{c}_{k'} \neq 0$ and $\tilde{c}_k = \delta_{kk'}\tilde{c}_{k'}$. From equation B.29 it follows further that $\tilde{c}_k = \delta_{kk'}$.

Inserting this in Eq.(B.30) we obtain

$$|E|_F = \left| 1 + t \sum_k \tilde{c}_k^2 \lambda_k \right| = |1 + t\lambda_{k'}|\,. \tag{B.32}$$

For small t, i.e. $t < |\lambda_i|^{-1} \quad \forall i$, this is clearly minimal if we choose $\lambda_{k'}$ to be the smallest eigenvalue.

Further iterations, e.g. n times, of selecting the irrelevant states remove successively the eigenstates corresponding to the n lowest eigenvalues. This is due to the fact that the spaces Kern(C) $\equiv$ Range(P) and Range(C) $\equiv$ Kern(P) are by construction L-invariant. This also makes the iteration unambiguous, a feature that is in general not present for nonlinear problems.

Note also that since $\lambda_i \leq 0$ the reduced states always belong to Range(L) as long as any remaining eigenvalue, i.e. an eigenvalue of $P_{n-1}LP_{n-1}$ is nonzero. Here, P_{n-1} results from the previous reduction step. In this case the error always vanishes for long times.

Summarising, the optimal short time projection leads to results that are not only consistent with the long time accuracy requirements, but even include them.

Appendix C.

Ordering of the Schur Decomposition

In the master equation approach in chapter 4 and 8 we make use of the ordered real Schur decomposition. For implementation we have chosen a public assessable algorithm for a real Schur decomposition. This is part of the currently (August 2007) latest release of the *Gnu scientific library* (`gsl`), i.e. release 1.9 [53, 47]. This algorithm does not perform a sorting of the diagonal blocks in the resulting Schur Matrix. To perform this we have used a modified version of the algorithm of Brandts [20]. Some previous work on this problem includes [8, 30, 89, 40].

As explained above the real Schur form S has a block structure. This structure has to be respected during the sorting. As we already start from a Schur decomposition we have access to all eigenvalues of S. From this information a list of block exchanges is determined. In the original algorithm this is implemented in form of the so called bubble sort algorithm. The bubble sort algorithm is a comparison based sorting algorithm. It was used to calculate an a priori list of swaps of adjacent blocks. Although bubble sort is simple, it is not very efficient and requires in the worst case $\mathcal{O}(N^2)$ steps. Also exchanging only neighbouring blocks, instead of exchanging directly the blocks at original and final position, leads to a significant amount of extra computational effort and additional numerical inaccuracies. Therefore we use the direct exchange.

For our purposes also no complete sorting is necessary. We are interested only in m target states. Thus we only have to find the m most relevant blocks and exchange them with the leading blocks. As mentioned above the exchange is done directly. For our purposes also the so called partial Schur decomposition [20], computing only the Schur vectors of interest, i.e. the target vectors, could be used. Since by now the real Schur decomposition is included in tested packages as the `gsl`, we have made no use of the partial Schur decomposition.

In the original algorithm the sorting of the complex eigenvalues was determined by the distance to a complex target value (or its complex conjugated, whatever is lower). In contrast, the sorting criterion for our purposes is the real part of the eigenvalue.

The basic component of the sorting algorithms is the exchange of two diagonal blocks which we will describe now. Exchanging two blocks affects only the

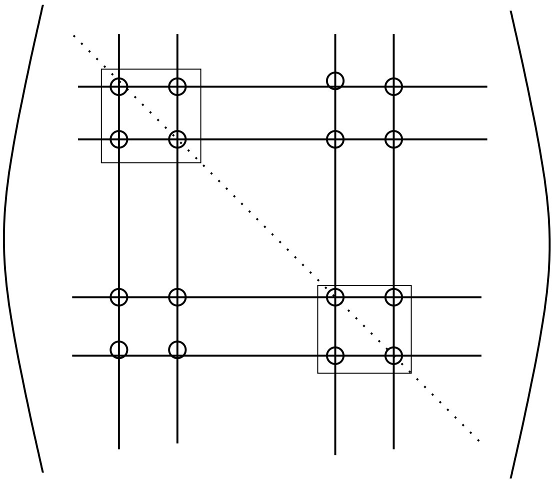


Figure C.1.: The rows and columns relevant for an exchange of two diagonal blocks, here as example for two 2 × 2 blocks. The blocks do not have to be of same size.

columns and rows of S that intersect the particular blocks as indicated in Fig. C.1. Therefore we can consider an effective matrix A composed of the blocks itself and their interactions, i.e. all entries of S that are indicated by circles in Fig. C.1. As S is of real Schur form this is also true for A, i.e.

$$A = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} . \tag{C.1}$$

The two diagonal blocks are $p \times p$ and $q \times q$ matrices, $p, q = 1, 2$, respectively. The diagonal blocks do not have to be of equal size. We assume that A_{11} and A_{22} have no common eigenvalues, which makes also sense in our context. First we have to solve the Sylvester equation for the $p \times q$ matrix X,

$$A_{11}X - XA_{22} = A_{12}. \tag{C.2}$$

Then we employ the QR-decomposition to find an orthonormal $p+q \times p+q$ matrix Q and an invertible matrix R which satisfies

$$Q^\dagger \begin{pmatrix} -X \\ \mathbb{1}_q \end{pmatrix} = \begin{pmatrix} R \\ 0 \end{pmatrix} . \tag{C.3}$$

With the orthonormal matrix Q we can now represent A in the desired form, i.e.

$$A = \begin{pmatrix} A_{22} & \tilde{A}_{12} \\ 0 & A_{11} \end{pmatrix} . \tag{C.4}$$

The QR decomposition was calculated by standard gsl-routines. For numerical stability some of the issues in [20] were considered. First, the 2 × 2 diagonal blocks were kept in standardised form, i.e.

$$\begin{pmatrix} \gamma & \mu \\ -\mu & \gamma \end{pmatrix} . \tag{C.5}$$

The solution of the Sylvester equation was also done by standard gsl-routines with partial pivoting. In [20] complete pivoting was used. The rescaling of the right hand side of Eq.(C.2) from [20] was kept.

Appendix D.

Mathematical Addenda

Linear algebra is the ideal framework to describe and manipulate finite-dimensional systems, as are exclusively considered in this work. In this Appendix some basic facts from the field of linear algebra are reproduced. This serves to determine the notation and conventions, as well as to recall some mathematical relations. Beside from providing the language, linear algebra contains a set of powerful tools that are also employed to obtain the reduction in the various approaches pursued in this thesis.

Again, no proofs of the statements below are given. If necessary these can be found in [45, 50].

D.1. Basic notation

D.1.1. Fields

Let K be a set together with two binary operations, addition and multiplication

$$+: \qquad K \times K \to K, \quad (a, b) \mapsto a + b, \tag{D.1}$$

$$\cdot: \qquad K \times K \to K, \quad (a, b) \mapsto a \cdot b. \tag{D.2}$$

For a field the followings axioms hold:

$$\begin{aligned}
&\text{Closure under } + \text{ and } \cdot : \\
&\quad a, b \in K \Rightarrow a + b \in K \wedge ab := a \cdot b \in K, && \text{(D.3)} \\
&\text{Associativity of } + : \\
&\quad a + (b + c) = (a + b) + c \quad \forall a, b, c \in K, && \text{(D.4)} \\
&\text{Associativity of } \cdot : \\
&\quad a \cdot (b \cdot c) = (a \cdot b) \cdot c \quad \forall a, b, c \in K, && \text{(D.5)} \\
&\text{Commutativity of } + : \\
&\quad a + b = b + a \quad \forall a, b \in K, && \text{(D.6)} \\
&\text{Commutativity of } \cdot : \\
&\quad a \cdot b = b \cdot a \quad \forall a, b \in K, && \text{(D.7)} \\
&\text{Distributivity of } \cdot \text{ over } + : \\
&\quad a \cdot (b + c) = (a \cdot b) + (a \cdot c) \quad \forall a, b, c \in K, && \text{(D.8)} \\
&\text{Existence of an additive neutral element :} \\
&\quad \exists\; 0 \in K, \text{ with } a + 0 = a \;\forall\; a \in K, && \text{(D.9)} \\
&\text{Existence of an additive inverse element :} \\
&\quad \forall\; a \in K \quad \exists\; -a \in K, \text{ with } a + (-a) = 0, && \text{(D.10)} \\
&\text{Existence of a multiplicative neutral element :} \\
&\quad \exists\; 1 \in K, \text{ with } a \cdot 1 = a \;\forall\; a \in K, && \text{(D.11)} \\
&\text{Existence of a multiplicative inverse element :} \\
&\quad \forall\; a \in K \backslash 0 \;\exists\; a^{-1} \in K, \text{ with } a \cdot (a^{-1}) = 0. && \text{(D.12)}
\end{aligned}$$

This is an abstract concept. For our purposes we will always consider the real or complex numbers, $\mathbb{R}$ and $\mathbb{C}$ respectively.

D.1.2. Vectors

D.2. Vector Space Axioms

A basic concept in linear algebra is that of a vector space. A vector space V over a field K, see Section D.1.1, is a set together with two binary operations, the vector addition

$$\begin{aligned}
&\quad V \times V \to V : v + w = u \in V \quad \forall v, w \in V && \text{(D.13)} \\
&\text{and the scalar multiplication} \\
&\quad K \times V \to V : \lambda v = u \in V \quad \forall \lambda \in K, v \in V. && \text{(D.14)}
\end{aligned}$$

The vector space is closed under these operations which satisfy the following axioms:

$$\begin{aligned}
&\text{Associativity :}\\
&\quad u + (v + w) = (u + v) + w \quad \forall u, v, w \in V, && \text{(D.15)}\\
&\text{Commutativity :}\\
&\quad v + w = w + v \quad \forall v, w \in V, && \text{(D.16)}\\
&\text{Additive neutral element :}\\
&\quad \exists\ 0 \in V, \text{ with } v + 0 = v\ \forall\ v \in V, && \text{(D.17)}\\
&\text{Additive inverse element :}\\
&\quad \forall\ v \in V\ \ \exists\ -v \in V, \text{ with } v + (-v) = 0, && \text{(D.18)}\\
&\text{Distributivity of scalar multiplication over vector addition :}\\
&\quad \lambda(v + w) = \lambda v + \lambda w\ \forall\ \lambda \in K,\ v \in V, && \text{(D.19)}\\
&\text{Distributivity of scalar multiplication over field addition :}\\
&\quad (\lambda + \mu)v = \lambda v + \mu v\ \forall\ \lambda, \mu \in K,\ v \in V, && \text{(D.20)}\\
&\text{Compatibility of scalar multiplication with field multiplication :}\\
&\quad \lambda(\mu v) = (\lambda\mu)v\ \forall\ \lambda, \mu \in K,\ v \in V, && \text{(D.21)}\\
&\text{Neutral element for scalar multiplication :}\\
&\quad \exists\ 1 \in K \text{ with } 1v = v. && \text{(D.22)}
\end{aligned}$$

We will treat only a special type of vector spaces, in particular only real or complex vector spaces. The vectors themselves are always finite tuples of numbers, i.e.

$$\begin{aligned}
&V = \mathbb{R}^N \text{ or } V = \mathbb{C}^N\\
&V \ni v = (v_1, \dots, v_N)\,. && \text{(D.23)}
\end{aligned}$$

The dimensionality of such a vector space is then N. In many cases[1] bold fonts are used to represent vectors, its components are typically denoted by indices.

D.2.1. Scalar product, Norm

A scalar product is a mapping of $V \times V \to K$, $\langle u, v\rangle = \alpha$. For real vector spaces it is symmetric, i.e. $\langle u, v\rangle = \langle v, u\rangle$ and linear in both variables.

The p-norm of a vector v is defined by

$$||v||_p := \frac{1}{N}\sqrt[p]{\sum_{i=1}^{N} |v_i|^p}. \tag{D.24}$$

Most relevant is the 2-norm or Euclidean norm describing the geometric length of a vector. It is also invariant under orthogonal or unitary transforms which is not true for general p. Also the 1-norm will be of interest later.

[1] But not exclusively, the particular nature of a variable will become clear within the context.

D.2.2. Basis

For a vector space of dimensionality N there exists always a set of N vectors $\{v^1, \ldots, v^N\}$ that span the vector space, i.e. every vector $u \in V$ can be written as linear combination

$$u = \sum_{i=1}^{N} \alpha_i v^i. \tag{D.25}$$

It is required that no v^i can be written as linear combination of the other v^j, $j \neq i$, i.e. the set is mutual linearly independent. The set $\{v^1, \ldots, v^N\} =: B$ is then termed a basis. The coefficients α_i are unambiguous and can be interpreted as components of a vector, the representation of u in the basis B. The choice of B is ambiguous. For practical purposes orthonormal bases are convenient, i.e. $\langle v^i, v^j \rangle = \delta_{ij}$. Each basis can be brought to this form, e.g. by the Gram-Schmidt procedure. Interpreting B columnwise as a matrix, this matrix is orthogonal, i.e. $B^\dagger B = BB^\dagger = \mathbb{1}$, in the notation described below.

The canonical basis is given by $\mathbb{1}_{ij} = \delta_{ij}$.

D.2.3. Subspaces

In a vector space V of dimension N a subspace W of dimension $M \leq N$ can be defined by selecting M linearly independent vectors w_i that span W, i.e.

$$W = \left\{ \sum_{i=1}^{M} \nu_i w_i, \;\; \nu_i \in K \right\} =: \operatorname{span}(w_1, \ldots, w_M). \tag{D.26}$$

The orthogonal complement $W^\perp$ is defined by

$$W^\perp := \{ v \in V | < v, w >= 0, \;\; \forall w \in W \}. \tag{D.27}$$

It is also a subspace and if B is an orthonormal basis for V so that the first M columns of B constitute an ONB of W, then the columns $M+1$ to N of B form an ONB of $W^\perp$. This construction is always possible.

W is a vector space of its own and embedded into V.

D.2.4. Linear Transformations, Matrixes

A linear transformation $\mathcal{L}$ between two vector spaces V, W is a mapping $V \rightarrow W$ with obeys the superposition principle

$$\mathcal{L}(\lambda v + \mu u) = \lambda \mathcal{L} v + \mu \mathcal{L} u \in W \quad \forall \lambda, \mu \in K, v, u \in V. \tag{D.28}$$

This property makes linear systems much more simple to treat than nonlinear systems. It is clear however, that nonlinearity is the generic case. For finite-dimensional vector spaces every linear transformation can be expressed by a matrix. A linear transformation and its corresponding matrix will be used synonymously. The column vectors of $\mathcal{L}$ are the image vectors of the canonical basis vectors under $\mathcal{L}$. The identity is represented by the identity or unit matrix $\mathbb{1}_{ij} = \delta_{ij}$.

Application of a linear transformation to a vector v corresponds to multiplying $\mathcal{L}$ with v. Successive linear transformation $\mathcal{L}_1 \circ \mathcal{L}_2$ are described by the matrix product $\mathcal{L}_1\mathcal{L}_2$ defined by

$$(\mathcal{L}_1\mathcal{L}_2)_{ij} = \sum_\alpha \mathcal{L}_{1,i\alpha}\mathcal{L}_{2,\alpha j} \tag{D.29}$$

Some matrices with special structures are of practical relevance. Computational or storage demands for many calculations can be reduced by exploiting these structures. This includes e.g. sparse matrices as diagonal or band diagonal matrices or triangular matrices. Matrices in which almost all entries are nonzero are termed dense.

The adjungated matrix $A^\dagger$ for a matrix A is defined by

$$A^\dagger_{ij} := (A_{ji})^\dagger, \tag{D.30}$$

where $\cdot^\dagger$ also indicates the complex conjugated of a number.

A matrix A is called orthogonal or orthonormal if the column vectors are mutual orthonormal. This property is extended to complex matrices by the concept of unitarity. Formal both properties are defined by

$$A^\dagger A = \mathbb{1}. \tag{D.31}$$

Note, that in complex arithmetics $\cdot^\dagger$ denotes the adjungated, complex conjugated matrix. This terminology extends to the linear transformation defined by A. Orthogonal transformation are generalisations of rotations.

A hermitian matrix satisfies

$$A^\dagger = A. \tag{D.32}$$

Real hermitian matrix are symmetric. A matrix is termed normal if it satisfies

$$A^\dagger A = AA^\dagger. \tag{D.33}$$

Orthogonal, unitary and hermitian matrices are always normal.

Range, Kernel and Rank

The range of a matrix or linear transformation $A : V \to W$ is defined by

$$\text{Range}(A) = \{x \in V | Ax \neq 0\}. \tag{D.34}$$

From the linearity of A it follows that Range(A) is a subspace of V. Its dimension is the rank of A. The Kernel of A is defined by

$$\text{Kern}(A) = \{x \in V | Ax = 0\}. \tag{D.35}$$

It is also a subspace of V and the orthogonal complement of Range(A). The dimension of the Kernel is termed nullity. Consequently rank-nullity theorem [45], i.e. rank(A)+nullity(A) $= N$ holds, where N is the dimension of V. A $N \times N$ matrix A with rank(A) $< N$ is termed singular.

D.2.5. Tensors

A tensor can be roughly considered as a generalisation of the concept of scalar, vector and matrix. It is an object which is defined by its transformation properties under basis changes[2]. A tensor has a order k defining the number of components or indices. For a vector space of dimension N a tensor of order k has N^k components and k indices. A zero order tensor is a scalar with is invariant under basis changes. A vector is a tensor of order 1 transforming according to Eq.(D.36) as

$$\hat{u}_i = B^\dagger_{i\alpha} u_\alpha. \tag{D.36}$$

Higher order tensor transform as

$$\hat{M}_{i,j,\ldots} = M_{\alpha\beta\ldots} B_{\alpha i} B_{\beta j} \cdots . \tag{D.37}$$

By introducing the transpose of a matrix $A^\dagger_{ij} := A_{ji}$ this is written for second order tensors

$$\hat{M} = B^\dagger M B. \tag{D.38}$$

In complex arithmetics the hermitian conjugate of C is denoted with $C^\dagger$. This is the transpose, complex conjugate of C. The complex conjugated is denoted by $C^* := \Re C - i\Im C$.

Tensors can also be considered as multi-linear transformations, e.g. $T : V_1 \times \ldots \times V_k \rightarrow W$ for a tensor T of order k. The Tensor is invariant under basis changes, but the corresponding representation transforms as indicated above. This is in analogy to the distinction between a linear transform and the corresponding matrix.

The tensor product [21] is a binary operation on two vectors v, w. This can be used e.g. for the construction of two-dimensional grids or two-dimensional discrete derivation operators as in Section 3.5.2.[3] A tensor product can also be defined for vector spaces, e.g. V, W. One way to construct a vector from the product space $V \otimes W$ is via the Kronecker product

$$\mathrm{kron}(v, w)_{i+N(\nu-1)} := v_i w_\nu \quad i = 1 : N_1 \nu = 1 : N_2, \tag{D.39}$$

where N_1 and N_2 are the dimensions of V and W, respectively. The dimesion of the product space is consequently the product of the dimensions of the original vectorspaces. The Kronecker product can be extended to higher order tensors, as matrices, see Eq.(3.40).

[2] For practical use it is often not necessary to distinguish between a tensor and the corresponding representation

[3] And more general also higher-dimensional discrete descriptions based on the one-dimensional discretisation.

Bibliography

[1] Francisco C. Alcaraz, Michel Droz, Malte Henkel, and Vladimir Rittenberg. Reaction-diffusion processes, critical dynamics and quantum chains. *Physics Reports*, 230:250–302, 1994.

[2] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault. Vector potentials in three dimensional non-smooth domains. *Mathematical Methods in the Applied Sciences*, 21:823–864, 1998.

[3] ANSI/IEEE, New York, IEEE. *IEEE Standard for Binary Floating-Point Numbers*, 1985.

[4] Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* Cambridge University Press, 2005.

[5] Akio Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. *Journal of Computational Physics*, 135(2):103–114, 1997.

[6] George B. Arfken and Hans J. Weber. *Mathematical Methods for Physicists.* Academic Press, San Diego, 6^{th} edition, 2005.

[7] R. de Roo B. W. van de Fliert, E. van Groesen and R. W. de Vries. Numerical algorithm for the calculation of nonsymmetric dipolar and rotating monopolar vortex structures. *Journal of Computational and Applied Mathematics*, 62:1–25, 1995.

[8] Z. Bai and J. W. Demmel. On swapping diagonal blocks in real schur forms. *Linear Algebra Appl.*, 186:73–95, 1993.

[9] C. Bardos and E. S. Titi. Euler equations of incompressible ideal fluids. *ArXiv Mathematics e-prints*, March 2007. math/0703406.

[10] Hans Behringer. Privat comunications.

[11] Hans Behringer, Thorsten Bogner, Alexey Polotsky, Andreas Degenhard, and Friederike Schmid. Developing and analyzing idealized models for molecular recognition. *Journal of Biotechnology*, 129:268–278, 2007.

[12] G. Berkooz, P. Holmes, and J. L. Lumley. *Turbulence, Coherent structures, dynamical systems and symmetry.* Cambridge Monographs on Mechanics, 1998.

[13] J.J. Binney. *The theory of critical phenomena.* Oxford science publications, Clarendon Press, 1993.

[14] G. D. Birkhoff. Proof of the ergodic theorem. *Proceedings of the National Academy of Sciences USA*, 17:656–660, 1931.

[15] James D. Bjorken and Sidney D. Drell. *Relativistic quantum fields.* New York, McGraw-Hill, 1968.

[16] Ph. Blanchard and E. Brüning. *Distributionen und Hilbertraumoperatoren: Mathematische Methoden der Physik.* Wien, Springer, 1^{st} edition, 1993.

[17] Thorsten Bogner. Density matrix renormalization for model reduction in nonlinear dynamics. *ArXiv Physics e-prints*, 2007. arXiv:0707.4384v1.

[18] Thorsten Bogner, Andreas Degenhard, and Friederike Schmid. Molecular recognition in a lattice model: An enumeration study. *Physical Review Letters*, 93(26):268108, 2004.

[19] Jean Pierre Boon and Sidney Yip. *Molecular Hydrodynamics.* Dover Publications, New York, 1980.

[20] J. H. Brandts. Matlab code for sorting real Schur forms. *Numerical Linear Algebra with Applications*, 9(3):249–261, 2002.

[21] Theodor Bröcker. *Lineare Algebra und Analytische Geometrie.* Basel, Birkhäuser, 2004.

[22] Stephen G. Brush. History of the Lenz-Ising model. *Reviews of Modern Physics*, 39(4):883–893, Oct 1967.

[23] T. Bui-Thanh, M. Damodaran, and K. Willcox. Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics. *AIAA Paper 4213*, 2003.

[24] J. M. Burgers. *The nonlinear diffusion equation.* Boston, Riedel, 1974.

[25] R. J. Bursill, T. Xiang, and G. A. Gehring. Thermodynamic density matrix renormalization group study of the magnetic susceptibility of half-integer quantum spin chains. *Journal of Physics*, 8(40):L583–L590, 1996.

[26] J. C. Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods.* Chichester, Wiley, 1987.

[27] Enrico Carlon, Malte Henkel, and Ulrich Schollwöck. Density matrix renormalization group and reaction-diffusion processes. *European Physical Journal B*, 12:99–114, Feb 1999.

[28] Enrico Carlon, Malte Henkel, and Ulrich Schollwöck. Critical properties of the reaction-diffusion model $2a \to 3a$, $2a \to 0$. *Physical Review E*, 63(3):036101, Feb 2001.

[29] Carlo Cercignani. *The Boltzmann Equation and Its Applications.* New York, Springer Verlag, 1987.

[30] Z. Chao and F. Zhang. Direct methods for ordering eigenvalues of a real matrix (in Chinese). *Chinese Univ. J.Comp.Math.*, 1:27–36, 1981.

[31] Shiyi Chen and Gary D. Doolen. Lattice Boltzmann method for fluid flows. *Anna. Rev. Fluid Mech.*, 30:329–364, 1998.

[32] Zhangxin Chen. *Finite Element Methods and their Applications.* Berlin, Springer, 2005.

[33] Claude Cohen-Tannoudji, Bernard Diu, and Franck Laloë. *Quantum mechanics.* New York, Wiley, 1993.

[34] C. Dasgupta, J. M. Kim, M. Dutta, and S. Das Sarma. Instability, intermittency and multiscaling in discrete growth models of kinetic roughening. *Physical Review E*, 55(3):2235–2254, July 1997.

[35] Dynamische Dichtematrix Renormierungsgruppe. DFG Projekt, 2004-2007.

[36] Lokenath Debnath. *Nonlinear partial differential equations for scientists and engineers.* Boston, Birkhäuser, 1997.

[37] Andreas Degenhard and Javier Rodríguez Laguna. Towards the evaluation of the relevant degrees of freedom in nonlinear partial differential equations. *Journal of Statistical Physics*, 106:1093–1120, 2002.

[38] Andreas Degenhard and Javier Rodríguez Laguna. Density matrix renormalization group approach to non-equilibrium phenomena. *Multiscale Modeling and Simulation (SIAM)*, 3(1):89–105, 2004.

[39] M. Doi and S. F. Edwards. *The theory of polymer dynamics.* Oxford science publications, Oxford, 1986.

[40] J. Dongarra, S. Hammarling, and J. Wilkinson. Numerical considerations in computing invariant subspaces. *SIAM J. Math. Anal. Appl.*, 13:145–161, 1992.

[41] D. G. Dritschel and B. Legras. Modeling oceanic and atmospheric vortices. *Physics Today*, 46:44–51, March 1993.

[42] Burkhard Dünweg, Ulf D. Schiller, and Anthony J. C. Ladd. Statistical mechanics of the fluctuating lattice Boltzmann equation. *ArXiv Physics e-prints*, 2007. cond-mat/0707.1581, accepted at Physical Review E.

[43] Joel H. Ferziger. *Computational methods for fluid dynamics.* Berlin, Springer, 1999.

[44] R. P. Feynman. *Statistical Mechanics.* Reading, MA., Benjamin, 1972.

[45] Gerd Fischer. *Lineare Algebra.* Wiesbaden, Vieweg, 1975-1997.

[46] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1:445–466, 1961.

[47] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual.* GNU Free Documentation License, 2^{nd} revised edition, 2006.

[48] P. R. Garabedian. *Partial differential equations.* New York, Wiley, 1967.

[49] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* New York, Academic press, 1981.

[50] G. H. Golub and C. F. VanLoan. *Matrix Computations.* Baltimore, Johns Hopkins Univ. Press, 1983-1996.

[51] A. N. Gorban, I. V. Karlin, and A. Y. Zinovyev. Invariant grids for reaction kinetics. *Physica A 333*, pages 106–154, 2004.

[52] David Gottlieb and Steven A. Orszag. *Numerical Analysis of Spectral Methods: Theory and Application.* Philadelphia, Pa., SIAM, 1977.

[53] *GNU Scientific Library (gsl).* `www.gnu.org/software/gsl`, Feb 2007. release 1.9.

[54] H. Haken. *Advaced Synergetics.* Berlin, Springer Verlag, 1983.

[55] Karen Hallberg. Density matrix renormalization: A review of the method and its applications. *ArXiv Physics e-prints*, 2003. cond-mat/0303557.

[56] T. Halpin-Healy and Y. C. Zhang. Kinetic roughening phenomena, stochastic growth, directed polymers and all that. aspects of multidisciplinary statistical mechanics. *Physics Reports*, 254:215–414, 1995.

[57] D. Hänel. *Molekulare Gasdynamik.* Berlin, Springer, 2004.

[58] A. Hasegawa. Self-organization processes in continuous media. *Advances in Physics*, 34:1–42, 1985.

[59] Harro Heuser. *Gewöhnliche Differentialgleichungen.* Stuttgart, Teubner, 1989.

[60] R. Hillerbrand. *Distribution of massless and massive particles in turbulent flows.* PhD thesis, PhD thesis, University Münster, 2007.

[61] Morris W. Hirsch and Stephen Smale. *Differential Equations, Dynamical Systems and Linear Algebra.* San Diego, Academic Press, 1974.

[62] Matthew Huber, James C. McWilliams, and Michael Ghil. A climatology for turbulent disperion in the troposphere. *Journal of the Atmospheric Sciences*, 58:2377–2394, 2001.

[63] ICAO. U.S. Standard Atmosphere. NOAA–S/T 76-1562, 1976.

[64] ICAO. Manual of the ICAO standard atmosphere (extended to 80 kilometres (262 500 feet)). 3^{rd}, 1993.

[65] M. K. Jain. *Numerical solution of differential equations.* New Delhi, Wiley Eastern, 1979.

[66] Mehran Kardar, Giorgio Parisi, and Yi-Cheng Zhang. Dynamic scaling of growing interfaces. *Physical Review Letters*, 56(9):889–892, Mar 1986.

[67] G. E. Karniadakis, M. Israeli, and S.A. Orszag. High-order splitting methods for the incompressible Navier-Stokes equation. *Journal of Computational Physics*, 97:414–443, 1990.

[68] Martin J. Klein. Principle of detailed balance. *Physical Review*, 97(6):1446–1447, Mar 1955.

[69] R. H. Kraichnan and D. Montgomery. Two-dimensional turbulence. *Reports on Progress in Physics*, 43:547–619, 1980.

[70] Robert H. Kraichnan. Inertial ranges in two-dimensional turbulence. *The Physics of Fluids*, 10:1417–1423, 1967.

[71] J. Krug and H. Spohn. Universality classes for deterministic surface growth. *Physical Review A*, 38(8):4271–4283, Oct 1988.

[72] Lev D. Landau, Evgenij M. Lifschitz, and Wolfgang Weller. *Hydrodynamik.* Berlin, Akademie-Verlag, 5^{th} revised edition, 1991.

[73] H. P. Langtangen. *Computational partial differential equations.* Berlin, Springer, 1998.

[74] Leon Lapidus and John H. Seinfeld. *Numerical solution of ordinary differential equations.* New York, Academic Press, 1971.

[75] H. Leipholz. *Theory of elasticity.* Leyden, Noordhoff, 1974.

[76] Birgit Lessmann, Tim Wilhelm Nattkemper, Preminda Kessar, Linda Pointon, Michael Khazen, Martin O. Leach, and Andreas Degenhard. Multiscale Analysis of MR Mammography Data. *Zeitschrift für Medizinische Physik (german journal of medical physics)*, 3:166–171, Aug 2007.

[77] T. M. Ligget. *Interacting particle dynamics.* Berlin, Springer Verlag, 1985.

[78] Douglas K. Lilly. Numerical simulation of two-dimensional turbulence. *The Physics of Fluids Supplement II*, pages 240–249, 1969.

[79] E. N. Lorenz. Empirical orthogonal functions and statistical weather prediction. *Scientific report 1, Statistical forecasting Project MIT*, 1956.

[80] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20:130–141, 1962.

[81] Hans Walter Lorenz. *Nonlinear dynamical economics and chaotic motion.* Berlin, Springer, 1993.

[82] H. G. Matthies and M. Meyer. Nonlinear Garlerkin methods for the model reduction of nonlinear dynamical systems. *Computers and Structures*, 81(12), 2003.

[83] M. Meyer and H. G. Matthies. Efficient model reduction in non-linear dynamics using the Karhunen-Loéve expansion and dual weighted residuals. *Computational Mechanics*, 31:179–191, 2003.

[84] Charles W. Misner, Kip S. Thorne, and John Archibald Wheeler. *Gravitation.* New York, Freeman, 1995.

[85] S. Moukouri and L. G. Caron. Thermodynamic density matrix renormalization group study of the magnetic susceptibility of half-integer quantum spin chains. *Physical Review Letters*, 77(22):4640–4643, Nov 1996.

[86] J. D. Murray. *Mathematical biology.* Berlin, Springer, 3^{rd} edition, 2002.

[87] M. A. Martín Delgado, J. Rodríguez Laguna, and G. Sierra. Single-block renormalization group: quantum mechanical problems. *Nuclear Physics B*, 601:569–590, 2001.

[88] M. A. Martín Delgado, G. Sierra, and R. M. Noack. The density matrix renormalization group applied to single-particle quantum mechanics. *Journal of Physics A: Mathematical and General*, 32:6079, 1999.

[89] K. C. Ng and B. N. Parlett. Development of an accurate algorithm for $\exp(bt)$, Part I, Programms to swap diagonal blocks, Part II. *CPAM-294*, 1988.

[90] A. H. Nielsen, X. He, J. Juul Rasmussen, and T. Bohr. Vortex merging and spectral cascade in two dimensional flows. *Physics of Fluids*, 8(9):2263–2265, 1996.

[91] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmore, and F. Thiele. A hierarchy of low dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[92] Oxford Numerical Algorithms Group. NAG Library. `http://www.nag.co.uk/`.

[93] Archimedes of Syracuse. On the measurement of the circle, c. 250?-212 BC.

[94] Koji Ohkitani and John D. Gibbon. Numerical study of singularity formation in a class of Euler and Navier-Stokes flows. *Physics of Fluids*, 12(12), 2000.

[95] I. Peschel, Xiaoqun Wang, Matthias Kaulke, and Karen Hallberg. *Density Matrix Renormalization.* Heidelberg, Springer, Lecture Notes in Physics, 1998.

[96] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes.* Cambridge Univerity Press, 1984-2006.

[97] E. R. Reiter. *Strahlströme: Ihr Einfluss auf das Wetter.* Berlin, Springer, 1970.

[98] Robert D. Richtmyer and K. W. Morton. *Difference methods for initial value problems.* New York, Interscience Publications, 1967.

[99] Javier Rodríguez-Laguna. Privat comunications.

[100] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction of compressible flows using pod and Garlerkin projection. *Physica D*, 189:115–129, 2003.

[101] Lewis H. Ryder. *Quantum Field Theory.* Cambridge University Press, 1985.

[102] L. Saint-Raymond. Convergence of Solutions to the Boltzmann Equation in the Incompressible Euler Limit. *Archive for Rational Mechanics and Analysis*, 166:47–80, 2003.

[103] J. J. Sakurai. *Modern quantum mechanics.* Reading, Mass., Addison-Wesley, 1985.

[104] Friederike Schmid. Coarse-grained models of complex fluids at equilibrium and under shear. *Computer Simulations in Condensed Matter: From Materials to Chemical Biology,*, 2:211–258, 2006.

[105] Friederike Schmid, Dominik Düchs, Olaf Lenz, and Beate West. A generic model for lipid monolayers, bilayers, and membranes. *ArXiv Physics e-prints*, 2006. physics/0608226.

[106] G. M. Schütz. *Exactly Solvable Models for Many-Body Systems far from Equilibrium.* London, Academic Press, 2001. Phase Transition and critical phenomena Vol.19, ed. C.Domb J.L.Lebowitz.

[107] L. Sirovich. Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics*, XLV:561–591, 1987.

[108] Martin Streek, Friederike Schmid, Thanh Tu Duong, Dario Anselmetti, and Alexandra Ros. Two-state migration of DNA in a structured microchannel. *Physical Review E*, 71(1):011905, 2005.

[109] Sauro Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Oxford University Press, 2001.

[110] Sauro Succi, Roberto Benzi, and Francisco Higuera. The lattice Boltzmann equation: A new tool for compuational fluid-dynamics. *Physica D*, 47, 1991.

[111] Gerard 't Hooft. The glorious days of physics - renormalization of gauge theories. *ArXiv Physics e-prints*, 1998. hep-th/9812203.

[112] G. I. Taylor. Statistical theory of turbulence, Parts I–V. *Proceedings of the Royal Society A*, 151:421–478, 1935.

[113] A. A. Townsend. *The Structure of Turbulent Shear Flow.* Cambridge University Press, 1980.

[114] Lloyd N. Trefethen. *Spectral Methods in Matlab.* Philadelphia, Pa., SIAM, 2000.

[115] Aslak Tveito and Ragnar Winther. *Introduction to partial differential equations: A computational approach.* Berlin, Springer, 2005.

[116] G. J. F. van Heijst and J. B. Flor. Dipole formation and collisions in a stratified fluid. *Nature*, 340:212–215, 1989.

[117] G.J.F. van Heijst. Self-organization of two-dimensional flows. *Nederlands Tijdschrift voor Natuurkunde*, 59:321–325, 1993.

[118] Th. von Karman. On the statistical theory of turbulence. *Proceedings of the National Academy of Sciences of the United States of America*, 23(2), 1937.

[119] J. von Neumann. Physical applications of the ergodic hypothesis. *Proceedings of the National Academy of Sciences USA*, 18:263–266, 1932.

[120] Peter Walters. *An introduction to ergodic theory.* New York, Springer, 1982.

[121] Steven Weinberg. *The Quantum Theory of Fields I-III.* Cambridge University Press, 1995.

[122] Steven R. White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69:2863, 1992.

[123] Steven R. White. Density matrix algorithms for quantum renormalization groups. *Physical Review B*, 48:10345–10356, 1993.

[124] C. D. Wilcox. *Turbulence Modeling for CFD.* La Canada, California, DCW Industries, 2^{nd} edition, 1998.

[125] Kenneth G. Wilson. The renormalization group: Critical phenomena and the Kondo problem. *Reviews of Modern Physics*, 47(4):773–840, Oct 1975.

[126] Kenneth G. Wilson. Problems in physics with many scales of length. *Scientific American*, 241:158–179, 1979.

[127] T. Yanagita, Y. Nishiura, and R. Kobayashi. Signal propagation and faliure in one-dimensional FitzHugh-Nagumo equations with periodic stimuli. *Physical Review E*, 71:036226, 2005.

[128] Anthony Zee. *Quantum Field Theory in a Nutshell.* Princeton University Press, 2003.

Index

Statutory Declaration

This thesis is the result of my own work, except where reference is made to the work of others. During the the work on this thesis the following publications were submitted or are in progress to be submitted:

- Density Matrix Renormalization for Model Reduction in Nonlinear Dynamics [17], submitted to Physical Review E.
- Evaluating Transient States of Diffusion-Reaction Systems by Non-symmetric Density Matrix Methods, in progress.
- General Variational Model Reduction applied to Incompressible Viscous Flows, in progress.

Lage, August 29, 2007

(Thorsten Bogner)

Acknowledgements

At this point I would like to thank all who supported me in this project.

I would like to express my gratitude to Friederike Schmid for the continuing support over the past years.

Furthermore I would like to thank Philippe Blanchard who officially resumed this project for support and discussion.

I would also like to thank Javier Rodríguez Laguna for support and his hospitality in Madrid together with German Sierra and M. A. Martín-Delgado.

This project was also made possible by Andreas Degenhard whom I would like to thank for his effort.

Further I would like to thank the German Science Foundation DFG for funding this work within a single project grant [35].

I am grateful to my office colleagues for their patience and the pleasant atmosphere.

This thesis would have been far less accessible without the careful proof-reading of Hans Behringer and Birgit Lessmann. I am also very gratefull for their discussions and comments.

A special thank goes to those who encouraged me during the past years, in particular to Birgit, Jens, Karolin, Ulf, Martin and Hans.

Last but not least, I thank my family for enduring support over the years.