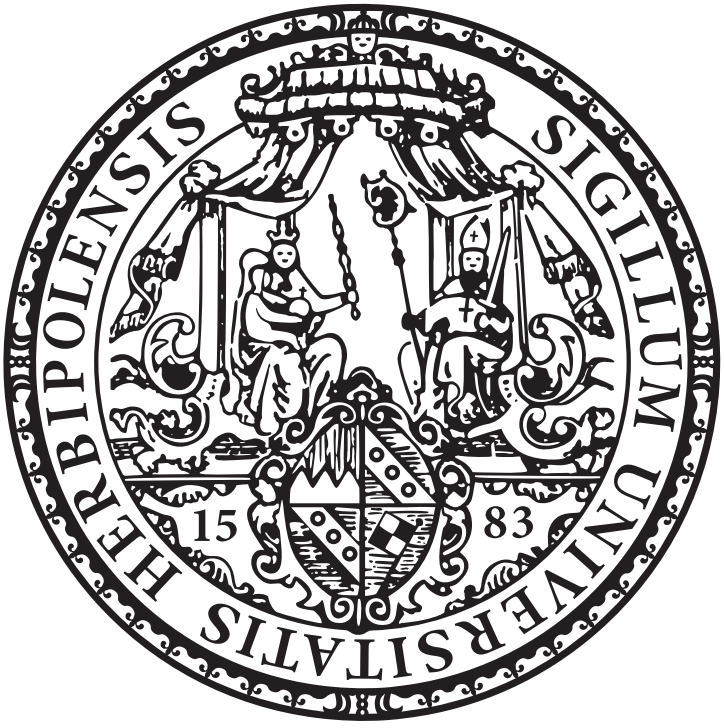
Fakultät
für Physik und Astronomie

Master thesis

Efficient solutions of memory integrals



Author: Fabian Pfeuffer
Fabian.Pfeuffer@stud-mail.uni-wuerzburg.de

Supervisor: Prof. Dr. W. Porod

Abstract

In this work, we present the implementation and effects of a history cut-off reduction scheme for the Kadanoff-Baym equations in case of a spacial homogeneous system in (1+1)-dimensions. As our model, we used a scalar theory with a φ^4-theory interaction term and considered loop corrections to the self-energy up to the second order. We found out, that a history size which is too small for the given system can alter the course of the system's time evolution strongly and might even lead to divergences in the correlation functions. However, when the size of the fixed history is increased, the deviation between the data of the time evolution for different histories rapidly declines. This indicates that it is not always necessary to keep the full history of the system. Besides the implementation of the history cut-off reduction scheme, we also developed a procedure, the DHS algorithm, to find a fitting history during the time evolution of the system. We tested different weighting factors for the weighted average that is used in this procedure and found out that weighting the determined history size for each momentum mode with their linear entropy leads to reasonable results.

Using the history cut-off reduction scheme, we observed the system's time evolution for the so-called quench and tsunami initial conditions up to late times $x^0 > 3000$ for 2-loop corrections to the self-energy. What we found is, that for both quench and tsunami, thermalization could be observed. Additionally, we look at the tsunami for a thermal background. For this setup up, we observed the particle distribution and the linear entropy up to $x^0 = 2500$. At the end of the time evolution, we fit the particle distribution with a Bose-Einstein distribution and saw an agreement between the fit and the data, which indicated the system's approach to thermal equilibrium. For the total linear entropy, we saw a fast initial increase followed by a steady state, which also indicates a thermalized state of the system.

Lastly, we investigated the effects of the lattice size and the lattice spacings on the system's time evolution. We observed that finite lattice effects are only relevant for 1-loop corrections, when the observation time x^0 becomes greater than $(0.573 \pm 0.018)V_s$, or for 2-loop corrections, when the system's volume is small, i.e. $V_s < 20$. For the lattice spacings we saw, that a rough spacing alters the course of the time evolution significantly. However, the courses of the time evolutions for different lattice spacings start to approach each other with a quadratically decreasing error, when the lattice spacings are decreased.

Zusammenfassung

In dieser Arbeit wird die Implementation, sowie die Effekte eines "history cut-off" Reduktionsverfahrens für die Kadanoff-Baym Gleichungen, für den Fall eines räumlich homogenen Systems in (1+1)-Dimensionen präsentiert. Wir verwenden eine skalare Theorie mit φ^4-Wechselwirkung als Modell und betrachten Loop-Korrekturen zur Selbstenergie bis zur zweiten Ordnung. Wir haben beobachtet, dass eine für ein gegebenes System zu kurze Vorgeschichte dazu führt, dass große Abweichungen oder sogar Divergenzen bei der Zeitentwicklung der Korrelationsfunktionen auftreten. Eine Vergrößerung der Vorgeschichte führt schnell zu einer signifikanten Verkleinerung dieser Abweichungen. Daraus wird deutlich, dass es nicht zwangsweise nötig ist die komplette Vorgeschichte des Systems bei der Lösung der Kadanoff-Baym Gleichungen zu berücksichtigen. Neben dem Reduktionsverfahren haben wir eine zusätzliche Routine entwickelt, den DHS Algorithmus, der dazu dient eine passende Größe für die Vorgeschichte des Systems zu finden. In der Routine tritt ein gewichteter Mittelwert auf, für den sich gezeigt hat, dass die lineare Entropie ein sinnvoller Gewichtungsfaktor ist.

Für das implementierte "history cut-off" Verfahren haben wir die Zeitentwicklung unseres Systems für "Quench"- und "Tsunami"-Anfangsbedingungen untersucht. Dabei betrachteten wir das System bis zu Zeiten $x^0 > 3000$ und berücksichtigen 2-Loop Korrekturen zur Selbstenergie. Wir haben beobachtet, dass das System sowohl für "Quench" als auch "Tsunami" thermalisiert. Weiterhin haben wir den "Tsunami" noch unter der zusätzlichen Annahme eines thermischen Hintergrunds untersucht, wobei wir die Teilchenverteilung und die lineare Entropie beobachtet haben. Das System wurde bis zur Zeit $x^0 = 2500$ entwickelt und die Teilchenverteilung mit einer Bose-Einstein-Verteilung gefittet. Die Übereinstimmung zwischen Teilchenverteilung und Fit war sehr gut, was darauf schließen lässt, dass das System tatsächlich ein thermodynamisches Gleichgewicht erreicht hat. Die total lineare Entropie ist zum Beginn der Zeitentwicklung rasch gestiegen, hat dann aber einen konstanten Wert angenommen. Auch dies ist ein Indiz für das Annähern an einen Gleichgewichtszustand.

Abschließend werden die Effekte der Gitterparameter untersucht. Es hat sich gezeigt, dass Störungen aufgrund der endlichen Größe des Gitters nur im Fall von 1-Loop, oder bei 2-Loop Korrekturen mit Gittergrößen $V_s < 20$, relevant sind. Bei der Untersuchung der Gitterabstände hat sich gezeigt, dass ein grobes Gitter einen großen Einfluss auf die Zeitentwicklung des Systems hat. Wird der Gitterabstand verkleinert, so verringert sich jedoch der Fehler zwischen den Daten für die unterschiedlichen Gitterabstände mit einer quadratischen Abhängigkeit.

Contents

Abstract **I**

1 Introduction **1**

2 Nonequilibrium quantum field theory **3**
2.1 The nonequilibrium problem . . . 3
2.2 Nonequilibrium Green's function . . . 6
2.3 The generating functional . . . 7
2.4 2PI-effective action . . . 10
2.5 Decoherence and entropy in nonequilibrium processes . . . 12
2.5.1 Relevant and irrelevant observables . . . 12
2.5.2 Linear entropy of a Gaussian state . . . 14

3 Numerical Implementation **18**
3.1 Spacial setup . . . 18
3.2 Time evolution . . . 20

4 Runtime analysis **23**
4.1 Memory usage . . . 23
4.2 Computation time . . . 24

5 Model **28**
5.1 Quench initial conditions . . . 30
5.1.1 1-loop approximation . . . 31
5.1.2 2-loop setting-sun approximation . . . 32
5.2 Tsunami initial conditions . . . 33
5.2.1 1-loop approximation . . . 34
5.2.2 2-loop setting-sun approximation . . . 34

6 Numerical effects **42**
6.1 Finite lattice effects . . . 42
6.1.1 1-loop approximation . . . 42
6.1.2 2-loop setting-sun approximation . . . 43
6.2 Time and spacial spacing . . . 45
6.2.1 Time spacing dependency . . . 45
6.2.2 Spacial spacing dependency . . . 47

7 Efficient solutions for memory integrals **50**
7.1 The history cut-off . . . 51
7.2 Cyclic memory . . . 54
7.3 Effects of the history cut-off . . . 56
7.4 Finding a suitable history size . . . 61
7.4.1 Calculating a fitting measure . . . 61
7.4.2 Determine history size algorithm . . . 63

8 Conclusion **72**
8.1 Summary . . . 72
8.2 Outlook . . . 73

Appendix **75**

A Settings **75**

B Additional Graphs **78**
B.1 Model . . . 78
B.2 Numerical effects . . . 79
B.3 Effects of a history cut-off . . . 80

C Code segments **87**
C.1 Double cyclic memory . . . 87
C.2 DHS algorithm . . . 88
C.3 Resizing the systems history . . . 92

D Graphical user interface **95**

References **99**

1. Introduction

As of today, one of the great unsolved mysteries of our observable universe is the asymmetry of matter and antimatter. As far as we know, everything around us is almost entirely made out of what we define as matter. The closest systems for this observation is the earth itself and our solar system, which both could not exist if they where made of equal parts of matter and antimatter. Antiprotons can be observed in cosmic rays, but they are outnumbered by protons 1 to 10^4, which suggest that they are created through cosmic ray collisions in the interstellar medium (see [1]). One could argue that some galaxies are partially or entirely made out of antimatter. However, the absence of γ rays from matter-antimatter annihilation between such galaxies and the intergalactic gas suggests that, within a distance of 10 Mpc, matter significantly dominates antimatter [1]. The matter-antimatter asymmetry is typically expressed by the baryon-to-photon ratio β, which, through recent measurements, indicates a ratio of

$$\beta = \frac{n_B - n_{\bar{B}}}{n_\gamma} \approx 6 \cdot 10^{-10}.$$

This is significantly higher than the ratios $n_B/n_\gamma = n_{\bar{B}}/n_\gamma \sim 10^{-18}$, which would be expected from a homogeneous baryon-symmetric universe [1]. However, this leaves the question where all the antimatter has gone.

A possible answer to this question might be given by nonequilibrium many-body physics. The generation of a baryon-antibaryon asymmetric universe from an initially baryon-antibaryon symmetric one can be described by the theory of baryogenesis. Amongst other things, baryogenesis requires processes out of thermal equilibrium [2]. Typically, one uses Boltzmann equations to describe these processes. However, Boltzmann equations assume the classical particle picture with instantaneous interactions and well localized particle excitations. Therefore, they are only a classical approximation to the quantum mechanical equations. Shortly after the Big Bang, where baryogenesis is expected to occur, the reliability of these approximations is questionable. For this reason, more complex nonequilibrium quantum field theoretical approaches might be necessary. These nonequilibrium quantum field theoretical approaches lead to the so-called Kadanoff-Baym equations, which are able to describe nonequilibrium quantum processes from first principle [3].

Nonequilibrium quantum field theory, and in particular the Kadanoff-Baym equations, are not limited to high energy physics theories like baryogenesis. They also have various applications in other fields of modern physics. The Kadanoff-Baym equations are used in

low energy condensed matter physic topics, for example, the evolution of photon excited correlated electron systems [4]. Furthermore, they might even give answers to fundamental questions about the concepts of quantum decoherence, entropy and the thermalization of quantum systems [5] [6] [7]. This is a testament to the universality of this theory. However, they have one major downside. That is, that they are on the one hand not analytically solvable and on the other hand require considerable amounts of computation power and memory for their numerical solution.

Fortunately, the high computational requirements for solving the Kadanoff-Baym equations can be reduced significantly. Using the work of Thomas Garratt, a PhD student at the department of particle physics (TPII) of the University of Würzburg as a basis, we implemented a reduction scheme for the simplified case of a spacial homogeneous system into the framework. In this work, we'll explain and discuss this reduction scheme and its effects on the numerical solution of the Kadanoff-Baym equations for a spacial homogeneous system.

2. Nonequilibrium quantum field theory

Before we start our introduction to the theory of nonequilibrium mechanics, we want to give a brief remark on natural units. As it is common in quantum field theory, we choose the natural unit system given by $\hbar = c = k_B = 1$.

Because

$$[\text{energy}] = [\text{mass}] = [\text{temperature}] = [\text{length}]^{-1} = [\text{time}]^{-1},$$

everything will be given in units of the system's initial mass M_0, which we'll come across later on in this work, when choosing the natural unit system.

2.1. The nonequilibrium problem

We want to begin with the description of the general nonequilibrium problem. Consider a many-particle system, which can be represented by the n particle state $|n\rangle$ in Fock space. This system can be described by the time independent Hamiltonian H which acts on the Fock space. The general nonequilibrium problem begins in thermal equilibrium, where the system is time translation invariant, hence the time independence of H. In this state, one can define a temperature T and describe the system by a equilibrium density operator

$$\rho_{eq} = \frac{e^{-\beta H}}{\mathrm{Tr}(e^{-\beta H})}, \tag{1}$$

where β is the inverse temperature ([8] p. 79). To bring the system out of thermal equilibrium, at a time t_0, an additional term $H'(t)$, which represents a time-dependent disturbance, is introduced to H in such a way, that the system is given by H for $t \leq t_0$ and

$$\mathcal{H}(t) = H + H'(t). \tag{2}$$

for times $t > t_0$ ([8] p. 81). This time dependent term breaks the time translation invariance and thus the system is no longer in thermal equilibrium.

In nonequilibrium quantum statistical mechanics, our main interest doesn't lie within the specific dynamics of the microscopic variables, but instead within the calculation of macroscopic observables, which are given by the expectation value of operators $\mathcal{O}(t)$, to determine the state of our system. These expectation values can be computed via

$$\langle \mathcal{O}(t) \rangle = \mathrm{Tr}\left(\rho(t)\mathcal{O}_S\right) = \mathrm{Tr}\left(\rho(t_0)\mathcal{O}_{\mathcal{H}}(t)\right), \tag{3}$$

where $\mathcal{O}_S$ represents the operator in the Schrödinger and $\mathcal{O}_\mathcal{H}(t)$ the operator in the Heisenberg picture for the Hamiltonian (2). The time evolution of the operator is governed by the Heisenberg equation

$$-i\frac{d}{dt}\mathcal{O}_\mathcal{H}(t) = [\mathcal{H}(t), \mathcal{O}_\mathcal{H}(t)] - i\left[\frac{\partial}{\partial t}\mathcal{O}_S(t)\right]_\mathcal{H}, \tag{4}$$

while the evolution of the density operator in the Schrödinger picture is given by the von-Neumann-Liouville equation

$$i\partial_t\rho(t) = [\mathcal{H}(t), \rho(t)]\,. \tag{5}$$

Both (4) and (5) can be formally solved with the ansatz

$$\mathcal{O}_\mathcal{H}(t) = U^\dagger(t,t_0)\mathcal{O}_S U(t,t_0), \qquad \rho(t) = U(t,t_0)\rho(t_0)U^\dagger(t,t_0), \tag{6}$$

where $U(t,t_0)$ is the time evolution operator ([9] p. 4-5), given by ([8] p. 84)

$$U(t,t') = Te^{-i\int_{t'}^{t} d\bar{t}\mathcal{H}(\bar{t})}. \tag{7}$$

T is the time order operator, which shifts terms with smaller time arguments to the left and vise versa.

By changing into the interaction picture, we can seperate the time independent part H of the Hamiltonian from the time dependent perturbation. First, we rewrite (3) using (6) and get

$$\mathrm{Tr}\left(\rho(t_0)\mathcal{O}_\mathcal{H}(t)\right) = \mathrm{Tr}\left(\rho(t_0)U_\mathcal{H}^\dagger(t,t_0)\mathcal{O}_S U_\mathcal{H}(t,t_0)\right). \tag{8}$$

Afterwards, we express the Schrödinger picture operator in the Heisenberg picture of the time independent Hamiltonian H, i.e.

$$\mathcal{O}_S = U_H(t,t_0)\mathcal{O}_H(t)U_H^\dagger(t,t_0), \tag{9}$$

with $U_H^\dagger(t,t_0) = e^{-iH(t-t_0)}$. After inserting this into (8), the terms $U_H^\dagger(t,t_0)U_\mathcal{H}(t,t_0)$ and its adjoint, can be combined to

$$V(t,t_0) = U_H^\dagger(t,t_0)U_\mathcal{H}(t,t_0) = Te^{-i\int_{t_0}^{t} d\bar{t}H'(\bar{t})}, \tag{10}$$

and we have isolated the evolution operator $V(t,t_0)$ for the interaction Hamiltonian

([8] p. 84-86). With this, the operator $\mathcal{O}_{\mathcal{H}}(t)$ can be written as a time evolution

$$\mathcal{O}_{\mathcal{H}}(t) = V^{\dagger}(t,t_0)\mathcal{O}_H(t)V(t,t_0) = \widetilde{T}e^{i\int_{t_0}^{t} d\bar{t}H'(\bar{t})}\mathcal{O}_H(t)Te^{-i\int_{t_0}^{t} d\bar{t}H'(\bar{t})}, \tag{11}$$

where $\widetilde{T}$ is the anti-time order operator, shifting terms with smaller time arguments to the right and vise versa. The time evolution in (11) evolves the system from t_0 to t and back to t_0. This is the so called in-in formalism, which is suitable for the calculation of expectation values.

The in-in time evolution can be conveniently described via a closed time path $\mathcal{C}$, also known as the Keldish contour, with a forward $\mathcal{C}^+$ and a backward $\mathcal{C}^-$ part, as shown in Fig. 1.

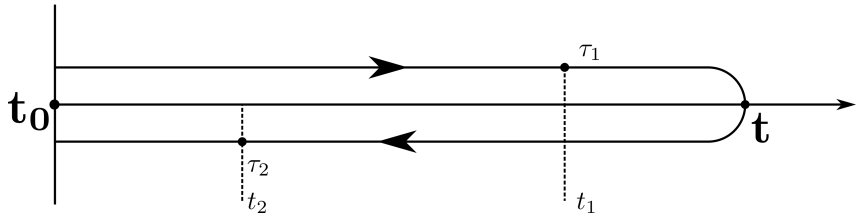


Figure 1: Closed time contour $\mathcal{C}$, also known as the Keldish contour. The time path starts at a point t_0, goes to t and back to t_0. The shift of the time path away from the time axis is only for illustrative purposes and has no physical meaning.

On this time path, $\mathcal{O}_{\mathcal{H}}$ can be conveniently written as ([8] p. 85)

$$\mathcal{O}_{\mathcal{H}}(t) = T_{\mathcal{C}}e^{i\int_{\mathcal{C}} d\tau H'(\tau)}\mathcal{O}_H(t), \tag{12}$$

where $T_{\mathcal{C}}$ is the contour ordering operator that is defined by

$$\begin{aligned} T_{\mathcal{C}}A(t_1)B(t_2) &= \begin{cases} A(t_1)B(t_2)\Theta(t_1-t_2) + B(t_2)A(t_1)\Theta(t_2-t_1) & \text{for } t_1,\ t_2 \text{ on } \mathcal{C}^+ \\ A(t_1)B(t_2)\Theta(t_2-t_1) + B(t_2)A(t_1)\Theta(t_1-t_2) & \text{for } t_1,\ t_2 \text{ on } \mathcal{C}^- \\ A(t_1)B(t_2) \text{ for } t_1 \text{ on } \mathcal{C}^+,\ t_2 \text{ on } \mathcal{C}^- \\ B(t_2)A(t_1) \text{ for } t_1 \text{ on } \mathcal{C}^-,\ t_2 \text{ on } \mathcal{C}^+, \end{cases} \\ &\equiv A(t_1)B(t_2)\Theta_{\mathcal{C}}(t_1-t_2) + B(t_2)A(t_1)\Theta_{\mathcal{C}}(t_2-t_1) \end{aligned} \tag{13}$$

where $A(x)$ and $B(x)$ are two arbitrary operators and $\Theta(x)$ is the Heaviside step function. In the last line of (13) we introduced the contour step function $\Theta_{\mathcal{C}}(t_1-t_2)$, which is defined through the contour ordering in the lines above it ([10] p. 22). Note that in case of the time contour, one has to differentiate between the real times t and the contour time τ. A

contour time $\tau_2 > \tau_1$ can correspond to either $t_2 < t_1$ or $t_2 > t_1$, depending on where the τ_i lie on the contour $\mathcal{C}$.

2.2. Nonequilibrium Green's function

In the previous chapter, we discussed the general nonequilibrium problem and introduced the Keldish contour as a convenient tool for the calculation of the average values of operators $\mathcal{O}$. The one ingredient that we somewhat neglected was the density operator ρ. For the setup of a nonequilibrium system, one has to specify an initial nonequilibrium density operator $\rho(t_0)$. However, instead of $\rho(t_0)$, one can also specify all initial correlation functions of the system ([10] p. 19).

As we'll see later on, the framework we're using is built on determining these correlation functions. This is why we want to give a brief introduction to nonequilibrium Green's functions (NEGF), also known as correlation functions, in the following.

Consider the Schrödinger equation for a wave function $\langle x|\Phi(t)\rangle = \Phi(x,t)$

$$i\hbar\frac{\partial}{\partial t}\Phi(x,t) = H(t,x)\Phi(x,t). \tag{14}$$

Using the Green's function G, which is defined by

$$\left(i\hbar\frac{\partial}{\partial t} - H(t,x)\right)G(x,t;x',t') = i\hbar\delta(x-x')\delta(t-t'), \tag{15}$$

the solution of (14) can be constructed by the convolution

$$\Phi(x,t) = \int dx' G(x,t;x',t')\Phi(x',t'), \tag{16}$$

between the wave and the Green's function. Now, consider the position space wave function again, however this time we use the time evolution operator $U(t,t')$, which transitions a state from time t' to t to rewrite the equation in the following way:

$$\begin{aligned}\Phi(x,t) = \langle x|\Phi(t)\rangle = \langle x|U(t,t')|\Phi(t')\rangle &= \int dx'\langle x|U(t,t')|x'\rangle\langle x'|\Phi(t')\rangle \\ &= \int dx'\langle x,t|x',t'\rangle\Phi(x',t').\end{aligned} \tag{17}$$

In the second to last step, we inserted a complete state $\int dx'|x'\rangle\langle x'| = 1$. If we compare (17) with (16), we see that the Green's function is equivalent to the transition element

i.e.

$$G(x,t;x',t') \equiv \langle x,t|x',t'\rangle. \tag{18}$$

Because the transition element, and therefore the Green's function, describes the propagation of a particle from point x' at time t' to some other point x at time t, the Green's function is also known as the propagator or correlation function, correlating the two spacetime points to one another (see [8] p. 4).

Now, we can draw the connection between our goal, the calculation of average values of operators $\mathcal{O}(t)$ via (3) and the appearance of NEGFs. In quantum field theory, all operators can be expressed by field operators $\Phi(x,t)$ and $\Phi^\dagger(x,t)$ ([11] p. 1). The particle density operator for example is given by

$$n(x,t) = \Phi^\dagger(x,t)\Phi(x,t). \tag{19}$$

If we consider a pure state $\rho = |\Phi\rangle\langle\Phi|$, the average particle density can be expressed by

$$\begin{aligned}\langle n(x,t)\rangle &= \mathrm{Tr}\left(\rho\Phi^\dagger(x,t)\Phi(x,t)\right) = \sum_\Psi \langle\Psi|\Phi\rangle\langle\Phi|\Phi^\dagger(x,t)\Phi(x,t)|\Psi\rangle \\ &= \langle\Phi|\Phi^\dagger(x,t)\Phi(x,t)|\Phi\rangle = \langle\Phi'(x,t)|\Phi'(x,t)\rangle = G(x,t,x,t),\end{aligned} \tag{20}$$

where we used $\mathrm{Tr}\,(...) = \sum_\Psi \langle\Psi|...|\Psi\rangle$, as well as the orthogonality condition $\langle\Psi|\Phi\rangle = \delta_{\Psi\Phi}$. As we can see in (20), the average particle density equals the equal time Green's function. Because all the other possible operators can be expressed by a combination of the field operators Φ and $\Phi^\dagger$, an arbitrary N-point-NEGF is therefore in general given by

$$G(1,...,N) = \langle\Phi_1^{i_1}(1),...,\Phi_N^{i_N}(N)\rangle, \tag{21}$$

with i_j indicating either the field operator or its adjoint. The arguments N represent a set of variables, for example position x_N, time t_N and spin σ_N. As we can see from the example (20), knowing all N-point-correlation functions is equivalent to knowing all operator expectation values and therefore knowing ρ.

2.3. The generating functional

In the previous chapter, we introduced the method of nonequilibrium Green's functions as a tool to perform nonequilibrium calculations and an alternative to the knowledge of the density operator ρ. Now, we want to present an efficient way on how to calculate these Green's functions.

As we have seen in (21), nonequilibrium Green's functions can be expressed by the expectation values of field operators. All the necessary ingredients we need to describe our system are therefore the initial conditions, given by either the initial density operator ρ_0, or all initial correlation functions, and the field operators $\Phi(x_i, t_i)$. In the following, we'll chose the field operators to be scalar fields. Both initial conditions and field operators are part of the generating functional of correlation functions

$$Z[J, R, \rho_0] = \mathrm{Tr}\left(\rho_0 T_{\mathcal{C}} e^{i\left(\int_{\mathcal{X}} J(x)\Phi(x) + \frac{1}{2}\int_{\mathcal{X}\mathcal{Y}} \Phi(x)R(x,y)\Phi(y)\right)}\right), \tag{22}$$

which is a generalization of the partition function for nonequilibrium systems and defined on the time contour $\mathcal{C}$ (see Fig. 1) ([10] p. 21). The coordinate x and y, respectively, is a spacetime variable, containing the contour time x_0 and the spacial component $\mathbf{x}$. For the integral, we used the shorthand notation $\int_{\mathcal{X}} \equiv \int_{\mathcal{C}} dx_0 \int d\boldsymbol{x}$. $J(x)$ and $R(x, y)$ are two source terms which are used to generate the correlation functions.

One can calculate nonequilibrium correlation functions from (22) by using the functional derivative with respect to the source $J(x)$, which fulfills

$$\frac{\delta J(x)}{\delta J(y)} = \delta_{\mathcal{C}}(x - y) = \delta_{\mathcal{C}}(x^0 - y^0)\delta(\boldsymbol{x} - \boldsymbol{y}), \tag{23}$$

and then setting the source terms J and R to zero. For example the one-point function, also known as the macroscopic field, can be calculated by

$$\left.\frac{\delta Z[J, R, \rho_0]}{i\delta J(x)}\right|_{J,R=0} = \langle\Phi(x)\rangle. \tag{24}$$

Higher order correlation functions can be obtained by differentiating (22) multiple times with respect to J ([10] p. 22).

Because of the explicit appearance of the trace and the initial density operator ρ_0, (22) is still in a somewhat inconvenient form. This can be avoided by changing into the functional integral representation and fixing the initial conditions afterwards. In the following, we'll present a brief outline how this change is done. More information about the derivation can be found in [10].

To evaluate the trace, we use the eigenstates

$$\Phi^{\pm}(t_0, \boldsymbol{x})|\varphi^{\pm}\rangle = \varphi_0^{\pm}(\boldsymbol{x})|\varphi^{\pm}\rangle \tag{25}$$

of the fields Φ for the forward $\mathcal{C}^+$ and the backward $\mathcal{C}^-$ contour at time t_0. The $\varphi(x)$ are

classical fields and fulfill the boundary condition $\varphi^{\pm}(x^0 = t_0, \boldsymbol{x}) = \varphi_0^{\pm}(\boldsymbol{x})$ for x^0 on $\mathcal{C}^{\pm}$. The functional integral representation of (22) is then given by

$$Z[J,R,\rho_0] = \int \prod_x d\varphi_0^+(x) \prod_y d\varphi_0^-(y) \langle \varphi^+|\rho_0|\varphi^-\rangle \times \int_{\varphi_0^+}^{\varphi_0^-} \mathcal{D}'\varphi e^{i\left(S[\varphi]+\int_{\mathcal{X}} J(x)\varphi(x)+\frac{1}{2}\int_{\mathcal{X}\mathcal{Y}}\varphi(x)R(x,y)\varphi(y)\right)}, \tag{26}$$

where $S[\varphi]$ is a classical action for the real scalar field φ. In this representation, the generating functional is split into two central parts. The first integral contains the information about the initial conditions of the system, while the second integral describes the system's quantum fluctuations ([10] p. 25).

We now want to give an expression for the initial conditions. The most general density matrix to describe the initial conditions of the system can be parametrized as

$$\langle \varphi^+|\rho_0|\varphi^-\rangle = \mathcal{N}\cdot\exp\left(i\left(a_0 + \int_{\mathcal{X}} a_1(x)\varphi(x) + \frac{1}{2}\int_{\mathcal{X}\mathcal{Y}} a_2(x,y)\varphi(x)\varphi(y) + \frac{1}{3!}\int_{\mathcal{X}\mathcal{Y}\mathcal{Z}} a_3(x,y,z)\varphi(x)\varphi(y)\varphi(z) + ...\right)\right). \tag{27}$$

$\mathcal{N}$ is a normalization constant and the a_i are the coefficients of a power series in the fields φ. Because the initial conditions only contribute at $t = t_0$, the exponent has to vanish for $t > t_0$, which in turn constrains the coefficients a_i to vanish identically for $t > t_0$. Plugging (27) back into (26), neglecting the irrelevant constants $\mathcal{N}$ and $\exp(ia_0)$ and redefining the source terms $J(x) + a_1(x) \to J(x)$ and $R(x,y) + a_2(x,y) \to R(x,y)$, we end up with ([10] p. 31-32)

$$Z[J,R,\rho_0] = \int \mathcal{D}'\varphi e^{i\left(S[\varphi]+\int_{\mathcal{X}} J(x)\varphi(x)+\frac{1}{2}\int_{\mathcal{X}\mathcal{Y}} R(x,y)\varphi(x)\varphi(y)+\frac{1}{3!}\int_{\mathcal{X}\mathcal{Y}\mathcal{Z}} a_3(x,y,z)...\right)}. \tag{28}$$

In this work, we'll restrict ourselves to the observation of Gaussian initial conditions, which suffice for many practical purposes [10], where the highest order of φ that appears in (27) is of the second order. Therefore all a_i with $i \geq 3$ vanish and the generating functional takes the simple form

$$Z[J,R] = e^{iW[J,R]} = \int \mathcal{D}'\varphi e^{i\left(S[\varphi]+\int_{\mathcal{X}} J(x)\varphi(x)+\frac{1}{2}\int_{\mathcal{X}\mathcal{Y}} R(x,y)\varphi(x)\varphi(y)\right)}. \tag{29}$$

It should be noted, that there are no approximations for the dynamics of the system. We merely restricted our choice of initial conditions to be Gaussian. During its time evolution,

higher irreducible correlations, which correspond to non-Gaussian density matrices, can still build up inside the system (cf. [10] p. 34).

2.4. 2PI-effective action

Our main interest lies in the dynamics of the correlation functions. For their description, we need the equations of motion of the nonequilibrium quantum system. The equations of motion can be derived from the two-particle-irreducible effective action (2PI-effective action) $\Gamma[\Phi, G]$, which in turn can be obtained from the generating functional (29). We can get $\Gamma[\Phi, G]$ by Legendre transforming the exponent of the generating function $W[J, R]$ after the source terms J and R, resulting in

$$\begin{aligned}\Gamma[\Phi, G] &= W[J, R] - \int_{\mathcal{X}} \frac{\delta W[J, R]}{i\delta J(x)} J(x) - \int_{\mathcal{X}\mathcal{Y}} \frac{\delta W[J, R]}{i\delta R(x, y)} R(x, y) \\ &= W[J, R] - \int_{\mathcal{X}} \Phi(x) J(x) - \int_{\mathcal{X}\mathcal{Y}} \frac{1}{2} \left(\Phi(x)\Phi(y) + G(x, y)\right) R(x, y),\end{aligned} \tag{30}$$

where $G(x, y)$ is the connected two-point function

$$G(x, y) = 2\langle T_{\mathcal{C}} \Phi(x)\Phi(y)\rangle - \langle\Phi(x)\rangle\langle\Phi(y)\rangle. \tag{31}$$

The effective action in (30) is the quantum mechanical equivalent of a classical action. Therefore the equations of motion can be extracted via its stationarity conditions given by

$$\frac{\delta\Gamma[\Phi, G]}{\delta\Phi(x)} = -J(x) - \int_{\mathcal{Y}} R(x, y)\Phi(y) \tag{32}$$

$$\frac{\delta\Gamma[\Phi, G]}{\delta G(x, y)} = -\frac{1}{2} R(x, y). \tag{33}$$

These two equations are the quantum equations of motion for Φ and G (see [10] p. 36).

We can write the effective action from (30) in a loop expansion, so that we can easily perform approximations if necessary. In this case the effective action takes the form

$$\Gamma[\phi, G] = S[\phi] + \frac{i}{2}\mathrm{Tr}_{\mathcal{C}} \ln G^{-1} + \frac{i}{2}\mathrm{Tr}_{\mathcal{C}}\, G_0^{-1}(\phi) G + \Gamma_2[\phi, G] + \mathrm{Tr}_{\mathcal{C}}\, G^{-1} G, \tag{34}$$

where $iG_0^{-1} = \delta^2 S[\Phi]/\delta\Phi(x)\delta\Phi(y)$ is the classical inverse propagator. The first three terms are the one-loop contribution and the term $\Gamma_2[\phi, G]$ contains all higher order contributions. The last term $\mathrm{Tr}_{\mathcal{C}} G^{-1} G$ leads to a normalization constant and is irrelevant for our further

discussion. If we vary (34) with respect to G, i.e. calculate the stationarity condition (33) again, and compare them with one another, we get

$$G^{-1}(x,y) = G_0^{-1}(x,y) - \Sigma(x,y) - iR(x,y). \tag{35}$$

The term $\Sigma(x,y)$ is the proper self-energy defined by (see [10] p. 35-37)

$$\Sigma(x,y;\phi,G) \equiv 2i\frac{\delta\Gamma_2[\phi,G]}{\delta G(x,y)}. \tag{36}$$

This equation can be inverted by building the convolution with $G(x,y)$ on both sides, which leads us to the partial differential equation

$$\int_{\mathcal{Z}} G_0^{-1}(x,z)G(z,y) - \int_{\mathcal{Z}} \left[\Sigma(x,z) + iR(x,z)\right] G(z,y) = \delta_{\mathcal{C}}(x-y). \tag{37}$$

In the absence of a macroscopic field $\langle\Phi\rangle = 0$, the inverse operator G_0^{-1} is given by

$$G_0^{-1} = i\left(\Box_x + m^2\right)\delta_{\mathcal{C}}(x-y). \tag{38}$$

Inserting (38) into (37) leads to the evolution equation of G (see [10] p. 41)

$$\left(\Box_x + m^2\right) G(x,y) + i\int_{\mathcal{Z}} \left[\Sigma(x,z) + iR(x,z)\right] G(z,y) = -i\delta_{\mathcal{C}}(x-y). \tag{39}$$

The source term R only contributes at the initial time t_0. Therefore the contribution of the integral part with R vanishes for all times $x^0 > t_0$. Because of this, the R term is more of interest for the specifications of the initial conditions and less for the explicit time evolution of the correlation functions, which is why we only consider the $R = 0$ case in our further discussion (see [10] p. 41).

From a standpoint of physical intuition, the object G is somewhat hard to grasp. To make its physical contend clearer, we can decompose the two-point function. This can be done by introducing the two functions

$$\rho(x,y) = i\langle[\Phi,\Phi]_-\rangle, \tag{40}$$

$$F(x,y) = \frac{1}{2}\langle[\Phi,\Phi]_+\rangle, \tag{41}$$

where $[\ \ ]_-$ denotes the commutator and $[\ \ ]_+$ the anti-commutator, respectively. The first function ρ is called the spectral function or spectral component and encodes the spectrum of the theory, while the second function F is called the statistical propagator

or statistical component and gives information about occupation numbers (see [10]). The full two-point propagator G is connected to ρ and F via

$$G(x,y) = F(x,y) - \frac{i}{2}\rho(x,y)\cdot \mathrm{sgn}_{\mathcal{C}}(x^0 - y^0), \tag{42}$$

where $\mathrm{sgn}_{\mathcal{C}}(x^0-y^0) = \Theta_{\mathcal{C}}(x^0-y^0) - \Theta_{\mathcal{C}}(y^0-x^0)$. Similar to (40) and (41), the self-energy can also be decomposed into a statistical $\Sigma_F(x,y)$ and a spectral part $\Sigma_\rho(x,y)$ (see [10]). Using these decompositions, as well as the shorthand notation $\int_{y^0}^{x^0} dz = \int_{y^0}^{x^0} dz^0 \int_{-\infty}^{\infty} d^d z$, (39) can be split into the coupled evolution equations for the statistical and the spectral function

$$\left[\Box_x + M^2(x)\right]\rho(x,y) = -\int_{y^0}^{x^0} dz \Sigma_\rho(x,z)\rho(z,y), \tag{43}$$

$$\left[\Box_x + M^2(x)\right]F(x,y) = -\int_{0}^{x^0} dz \Sigma_\rho(x,z)F(z,y) + \int_{0}^{y^0} dz \Sigma_F(x,z)\rho(z,y). \tag{44}$$

The two equations (43) and (44) are known as the Schwinger-Dyson or Kadanoff-Baym equations (KBE) (see [10] p. 45). Due to the so-called memory integrals on the right hand side of (43) and (44), which integrate over the systems history and grow with the time evolution, solving the Kadanoff-Baym equations is a difficult task. It is the main topic of this work to find a efficient solution to numerically solve them.

2.5. Decoherence and entropy in nonequilibrium processes

The process of quantum decoherence and the production of entropy are related to nonequilibrium quantum dynamics, especially in terms of thermalization ([5], [12]). Entropy production in nonequilibrium systems is an interesting topic and will be made use of later on in this work. Because of this, we'll in the following, give a brief introduction to observables, quantum purity and entropy for nonequilibrium systems. Our introduction to this topic will be mainly based on [13] and [9].

2.5.1. Relevant and irrelevant observables

The entropy of a system is not uniquely defined. There are various types of entropies one could define for some system at hand, for example in case of a system in thermal equilibrium, one can define the thermostatic entropy S_{th} (thermostatic refers to a thermodynamic system in equilibrium). This entropy is a function of a set of measurable

macroscopic thermostatic variables like for example the particle number and the energy, which define the system on a macroscopic scale.

On the other hand, one could consider the other extreme, an entropy that is fully defined on the microscopic scale of the system, as it is the case for the von-Neumann entropy S_{vN} given by

$$S_{vN} = -kD \ln D. \tag{45}$$

Here, in terms of its probabilistic features, the system's state is fully defined on a microscopic scale by the total density operator D. In thermal equilibrium, the von-Neumann entropy can be identified with the thermostatic entropy S_{th}. However, when one considers a dynamical system, as it is the case in nonequilibrium physics, this identification is no longer justified.

As already stated above, the knowledge of the total density operator D completely determines the state of the system on a statistical level, so that the expectation value of any observable A can be calculated by

$$\langle A \rangle = \mathrm{Tr}\left(DA\right). \tag{46}$$

In practice, the complete knowledge of the microstate of a system, and therefore D, is not obtainable. One has to differentiate between macroscopically observable quantities $\boldsymbol{\mathcal{A}}_i$, that correspond to the expectation value of an observable $\langle \mathcal{A}_i \rangle$ and the remaining unobservable quantities. In accordance with [13], we term the observable quantities as relevant set $R \equiv \{\mathcal{A}_i\}$ and the unobservable quantities as irrelevant set. The relevant observables fulfill the equation

$$\mathrm{Tr}\left(D\mathcal{A}_i\right) = \langle \mathcal{A}_i \rangle = \boldsymbol{\mathcal{A}}_i, \tag{47}$$

which defines their expectation values. Note however, that the density operator D that fulfills (47) is not uniquely defined. For different sets of unobservable quantities, D can still fulfill (47).

To avoid such a redundancy, one can split the total density operator into two parts D_0 and D_1, where D_0 is the relevant density operator, which fulfills (47), and is given purely by the observables $\boldsymbol{\mathcal{A}}_i$. D_1 is the irrelevant density operator given by the remaining unobservable variables.

In this way, the total system can now be split into two parts. The part that is observable through the relevant set R and described by D_0, which one conveniently terms as the

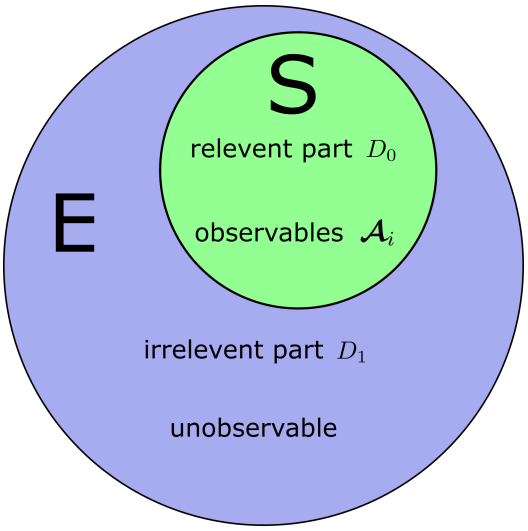


Figure 2: Schematic depiction of an isolated quantum system, which can be separated into an observable part, the system S described by a density operator D_0 and an unobservable part, the environment E described by the density operator D_1.

system S and a second part, the environment E, which is unobservable and described by D_1. Together, the system and the environment build the open system shown in Fig. 2 (see [9]).

We're interested in the least biased state D_0, i.e. a D_0 which contains the minimal amount of information and is still compatible with (47) [13]. That is because it contains the least amount of necessary information which still enables us to make predictions about the system through measuring the set of $\mathcal{A}_i$'s. Such a D_0 can be achieved by choosing it in such a way, that it maximizes the von-Neumann entropy (45). This is because the von-Neumann entropy is in fact a measure for the "*lack of information which arises from the incompleteness of [the] statistical description by means of the density operator D*" [13].

2.5.2. Linear entropy of a Gaussian state

In section 2.3, we stated that we'll only consider Gaussian initial conditions. For this reason, we limited the following discussion to a Gaussian density operator.

Consider the most general Gaussian density operator given by

$$\rho(t) = \frac{1}{Z} \exp\left(-\frac{1}{2}\left(\alpha(t)Q^2 + \beta(t)\{Q,P\} + \gamma(t)P^2 + \delta(t)Q + \eta(t)P\right)\right) \tag{48}$$

where Q and P are the generalized coordinate and the conjugated momentum, α, β, γ, δ and η are real valued functions and the partition function Z is a normalization constant

so that $\mathrm{Tr}\,(\rho) = 1$. By performing a shift $Q = q + \langle Q \rangle$ and $P = p + \langle P \rangle$ where $\langle \cdot \rangle$ denotes the expectation value of the operator, (48) can be simplified to (see [9])

$$\rho(t) = \frac{1}{Z} \exp\left(-\frac{1}{2} \left(\alpha(t) q^2 + \beta(t) \{q, p\} + \gamma(t) p^2 \right) \right). \tag{49}$$

Our goal in this discussion is the calculation of $\mathrm{Tr}\,(\rho^2)$ which we'll further use to derive the linear entropy of the system. To perform calculations, it is convenient to diagonalize (49). This can be achieved by introducing a pair of operators

$$b(t) = \sqrt{\frac{\sigma}{2\alpha\hbar}} \left(i\frac{\alpha}{\sigma} q + \left(1 + \frac{i\beta}{\sigma} \right) p \right), \qquad b^\dagger(t) = \sqrt{\frac{\sigma}{2\alpha\hbar}} \left(-i\frac{\alpha}{\sigma} q + \left(1 - \frac{i\beta}{\sigma} \right) p \right), \tag{50}$$

where $\sigma(t) = \sqrt{\alpha\gamma - \beta^2}$ and $\hbar = 1$ for the natural unit system we chose. With these two operators (49) can be written as

$$\rho(t) = \frac{1}{Z'} \exp\left(-\sigma(t) N(t) \right), \qquad N(t) = b^\dagger(t) b(t), \qquad Z' = e^{\frac{\sigma}{2}} Z, \tag{51}$$

where $N(t)$ is the number operator of a Fock basis $|n\rangle$ with the creation and annihilation operators $b^\dagger(t)$ and $b(t)$. In this basis, $N(t)$ fulfills the eigenvalue equation $N(t)|n\rangle = n|n\rangle$ and the density operator takes the form

$$\rho(t) = \frac{1}{Z'} \sum_{n=0}^{\infty} e^{-\sigma(t) n} |n\rangle\langle n|. \tag{52}$$

Now, the density operator is in a form in which we can easily perform the calculation of $\mathrm{Tr}\,(\rho(t)^2)$ ([9] p. 12). But before we perform this calculation, we still need to fix Z, which can be done by using the normalization condition $\mathrm{Tr}\,(\rho(t)) = 1$. This leads us to

$$\mathrm{Tr}\,(\rho(t)) = \frac{1}{Z'} \sum_{i,n=0}^{\infty} \langle i | e^{-\sigma(t) n} | n \rangle \langle n | i \rangle = \frac{1}{Z'} \sum_{n=0}^{\infty} \left(e^{-\sigma(t)} \right)^n = \frac{1}{Z'} \cdot \frac{1}{1 - e^{-\sigma(t)}} = 1, \tag{53}$$

where we used $\langle n|i\rangle = \delta_{ni}$ in the second and the geometrical series in the third step. The normalization constant then takes the form

$$Z' = \frac{1}{1 - e^{-\sigma(t)}} = 1 + \frac{1}{e^{\sigma(t)} - 1}. \tag{54}$$

Taking a closer look at the second term in (54), one can see that its shape is that of a Bose-Einstein distribution with $\dfrac{E}{T} = \sigma$. In fact, when calculating the expectation value

of N we get

$$\begin{aligned}\langle N(t)\rangle = \mathrm{Tr}\left(\rho(t)N(t)\right) = \frac{1}{Z'}\sum_{i,n=0}^{\infty}\langle i|e^{-\sigma(t)n}|n\rangle\langle n|N(t)|i\rangle &= \frac{1}{Z'}\sum_{n=0}^{\infty} n\left(e^{-\sigma(t)}\right)^n \\ &= \left(1-e^{-\sigma(t)}\right)\cdot\frac{e^{-\sigma(t)}}{\left(1-e^{-\sigma(t)}\right)^2} \\ &= \frac{1}{e^{\sigma(t)}-1},\end{aligned} \tag{55}$$

where we used a commonly known expression

$$\sum_{n=0}^{\infty} a_0\cdot nq^n = a_0\frac{q}{(1-q)^2}, \tag{56}$$

that is related to the geometrical series, in the second to last step. If one compares this to (54), one can see that (55) follows the same Bose-Einstein distribution as the second term in (54). We can therefore identify

$$\frac{1}{e^{\sigma(t)}-1} = \langle N(t)\rangle \equiv \bar{n}(t) \tag{57}$$

as the average statistical particle number, which we denote with $\bar{n}(t)$ ([9] p.12). With (52) and (54) at hand, we are now able to calculate

$$\mathrm{Tr}\left(\rho(t)^2\right) = \frac{1}{Z'^2}\sum_{i,n,m=0}^{\infty}\langle i|e^{-\sigma(t)n}|n\rangle\langle n|e^{-\sigma(t)m}|m\rangle\langle m|i\rangle = \frac{1}{Z'^2}\sum_{n=0}^{\infty}\left(e^{-2\sigma(t)}\right)^n. \tag{58}$$

After inserting Z' and using the geometrical series again, we get

$$\begin{aligned}\mathrm{Tr}\left(\rho(t)^2\right) &= \frac{1}{\left(1-e^{-\sigma}\right)^{-2}}\cdot\frac{1}{1-e^{-2\sigma}} = \frac{1}{\left(1-e^{-\sigma}\right)^{-2}}\cdot\frac{e^{\sigma}-1+1}{e^{\sigma}-1+1-e^{-\sigma}} \\ &= \frac{1}{(1+\bar{n})^2}\cdot\frac{\frac{1}{\bar{n}}+1}{\frac{1}{\bar{n}}+\frac{1}{1+\bar{n}}} = \frac{1}{2\bar{n}(t)+1}.\end{aligned} \tag{59}$$

$\mathrm{Tr}\left(\rho^2\right)$ is a measure for the purity of the state described by ρ, which can be seen from the property

$$\mathrm{Tr}\left(\rho^2\right)\begin{cases}=1, & \text{pure state,}\\ <1, & \text{mixed state.}\end{cases} \tag{60}$$

It is therefore reasonable to address $\mathrm{Tr}\left(\rho^2\right)$ as the quantum purity μ of a state ([14] p. 10). Note that the statistical particle number $\bar{n}$ is related to the purity of a state. It can be interpreted as an indicator for the number of decohered regions in a Gaussian state

([9] p. 14). Finally, we want to introduce another measure connected to μ, the linear entropy S_L, which is defined as

$$S_L(\rho) = (1 - \mu) = \left(1 - \mathrm{Tr}\left(\rho^2\right)\right), \tag{61}$$

in the case of continuous variables ([14] p.10 & p.38). The linear entropy is the first-order approximation of the von Neumann entropy. It is therefore connected to the lack of information that arises from the description of our system by the density operator ρ ([14] p.10). Using (61) should enables us to quantify this lack of information and draw conclusions about the flow of information between the system S and its environment E.

3. Numerical Implementation

In this chapter, we want to present the setup to numerically solve the KBE in $(d+1)$ dimensions under the assumption of a spacial homogeneous system with d dimensions. A numerical solution for the homogeneous KBE is nothing new regarding the treatment of the KBE. Several authors have already done calculations of this kind (see. [15], [3], [5], and more.). We chose this simplified framework, because the main focus of this work is not another solution for the KBE for a spatial homogeneous system in $d+1$ dimensions, but rather to find an efficient way of solving this problem, especially in regards of providing a basis to efficient solution for the significantly more advanced problem of inhomogeneous systems.

3.1. Spacial setup

For our implementation, we mainly followed the work of J. Berges [15]. We restrict ourselves to $d = 1$ and consider a scalar field theory for a spatial homogeneous system, meaning that only the distance $b = |\mathbf{x} - \mathbf{y}|$ between two space-time coordinates x and y is of importance. The restriction to homogeneous systems allows us to shift the problem completely into momentum space by Fourier transforming

$$\widetilde{F}(x^0, y^0, \mathbf{p}) = \frac{1}{(2\pi)^d} \int d\boldsymbol{z}\ F(x^0, y^0, \mathbf{z}) \cdot e^{i\mathbf{p}\mathbf{z}} \tag{62}$$

$$\widetilde{\rho}(x^0, y^0, \mathbf{p}) = \frac{1}{(2\pi)^d} \int d\boldsymbol{z}\ \rho(x^0, y^0, \mathbf{z}) \cdot e^{i\mathbf{p}\mathbf{z}}, \tag{63}$$

where $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y}$. The change from real to momentum space causes the differential operator of the KBE to become algebraic in its spacial component, i.e.

$$\left[\Box_x + \mathrm{M}(x)^2\right] \to \left[\left(\partial_{x^0}^2 + \mathbf{p}^2\right) + \mathrm{M}(x)^2\right]. \tag{64}$$

The Fourier transformation effectively reduces the $2(d+1)$ dimensional phase space to a $(d+2)$ dimensional one. Instead of starting from real space and performing a Fourier transformation, we redefine $\widetilde{F} \to F$ and $\widetilde{\rho} \to \rho$ and initialize the system in momentum space. When needed, the components of the 2-point-function can then be transformed back into real space via a reversed Fourier transformation. For our numerical solution of the KBE, we use the FFTW package (see. [16]) for all Fourier transformations, that need to be done.

To perform numerical calculations, we are bound to perform a discretization. This is

done by the standard method of lattice discretization. The spacial part of this lattice is a d-dimensional hypercube with periodic boundary conditions, a side length of N_{sp} lattice points and a spacing denoted by a_s between them. The time part of the lattice is quadratic with a side length of N_t lattice points and a spacing denoted by a_t. Due to the lattice discretization, the momentum modes can only take discrete values given by

$$p_{n_i} = \frac{2}{a_s}\sin\left(\frac{\pi n_i}{N_{sp}}\right), \tag{65}$$

where $i = 1, ..., d$ indicates the dimension and $n_i = 0, ..., N_{sp} - 1$ a specific lattice point for the i-th dimension (see. [3] p. 469). The squared momentum can then simply be calculated using the scalar product which yields

$$\mathbf{p}^2 = \sum_{i=1}^{d} \frac{4}{a_s^2}\sin^2\left(\frac{a_s p_i}{2}\right), \qquad p_i = \frac{\pi n_i}{N_{sp} a_s}. \tag{66}$$

The lattice discretization introduces a momentum cutoff $-\frac{\pi}{a_s} \leq p \leq \frac{\pi}{a_s}$. From the perspective of the lattice, momentum modes with absolute values higher than $|p_{max}| = \frac{\pi}{a_s}$ look just like lower frequency modes on the lattice. This effect is shown for a 1-dimensional lattice with $N_{sp} = 11$ spacial points, that is depicted in Fig. 3.

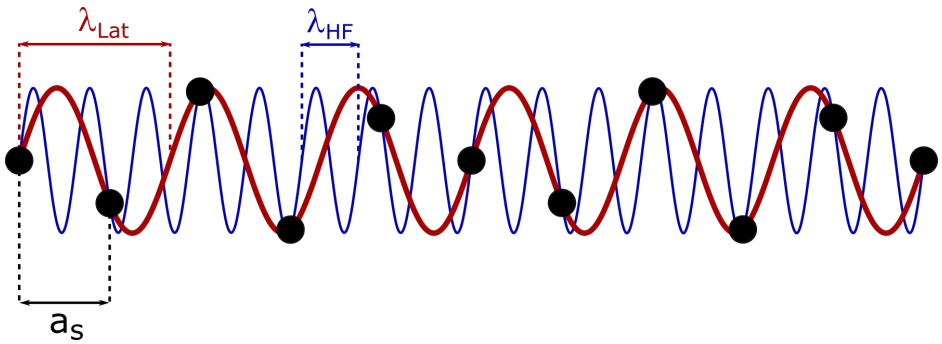


Figure 3: Depiction of the momentum cutoff of high frequency modes due to a lattice discretization. The black dots represent the lattice points with a spacing of a_s in 1-dimension. A high frequency mode with $p = 1.6 \cdot p_{max}$ is given by the blue curve. The response of the lattice is depicted by the red curve with $p = 0.6 \cdot p_{max}$. One can see that the lattice is too rough to resolve the high frequency mode, causing the mode to appear as though it was the lower frequency mode (red curve).

The high momentum mode with wave length λ_{HF} can't be resolved by the lattice and is projected onto a corresponding low frequency mode with wave length λ_{Lat} within the cutoff interval. As we'll see in sec. 6.2.2, cutting off high momentum modes can significantly influence the dynamics of the system.

3.2. Time evolution

In case of a lattice discretization, the continuous time coordinates x^0 and y^0 are turned into their discrete version $x^0_{dis} = n\,a_t$ and $y^0_{dis} = m\,a_t$, where n and m are two integer numbers. We will use $x^0 \equiv x^0_{dis}$ and $y^0 \equiv y^0_{dis}$ for the rest of this work, if not explicitly stated otherwise. Because of the discretization, we can write the statistical and the spectral component $F(n,m)$ and $\rho(n,m)$ as functions of the two integers n, m and the momentum $\mathbf{p}$, which we suppressed in the notation for the upcoming discussion.

We now have to deal with the derivative and the integral in (43) and (44). For our discretization scheme, we again chose the same approach as J. Berges [15], which is the finite difference method for the derivative and the trapezoidal rule for the integral. In case of a second derivative, appearing in the KBE, the finite difference method consists of two steps. First we perform a difference quotient in the forward direction, meaning $(F(n+1,m) - F(n,m))/a_t$ and equivalently for ρ, and then another one in the backwards direction $(F(n,m) - F(n-1,m))/a_t$. The second derivative of x^0 then turns into

$$\partial^2_{x^0} F(x^0, y^0) \to \frac{1}{a_t^2} \left(F(n+1,m) + F(n-1,m) - 2F(n,m) \right). \tag{67}$$

As already stated above, in order to calculate the memory integrals in the KBE, we made use of the trapezoidal rule, where the integral is approximated by a set of trapezes with respective area $A = a_t \left(F(n,m) + F(n+1,m) \right)/2$. In its discretized version, the first memory integral of the statistical function F is then given by

$$\int_0^{x^0} dz\; \Sigma_\rho(x,z) F(z,y) \to a_t \left(\frac{\Sigma_\rho(n,0)F(0,m)}{2} + \sum_{l=1}^{n-1} \Sigma_\rho(n,l)F(l,m) + \frac{\Sigma_\rho(n,n)F(n,m)}{2} \right). \tag{68}$$

The other memory integrals transform in the same way. Because of the way all the trapezes are added up to calculate the integral, we get boundary terms that contain a factor of $\frac{1}{2}$. Note however, that because of the anti-commutation relation (40), all boundary terms for ρ and Σ_ρ with equal time components $n = m$ vanish. Taking this fact into account, and using previously discussed approximation schemes, we can rewrite the KBE into a

numerically solvable form given by [15]

$$
\begin{aligned}
F(n+1,m,\mathbf{p}) = {} & 2F(n,m,\mathbf{p}) - F(n-1,m,\mathbf{p}) \\
& - a_t^2 \left(\mathbf{p}^2 + M(n)^2\right) F(n,m,\mathbf{p}) \\
& - a_t^3 \left\{ \frac{\Sigma_\rho(n,0,\mathbf{p})F(0,m,\mathbf{p})}{2} - \frac{\Sigma_F(n,0,\mathbf{p})\rho(0,m,\mathbf{p})}{2} \right. \\
& + \sum_{l=1}^{m-1} \left(\Sigma_\rho(n,l,\mathbf{p})F(l,m,\mathbf{p}) - \Sigma_F(n,l,\mathbf{p})\rho(l,m,\mathbf{p})\right) \\
& \left. + \sum_{l=m}^{n-1} \left(\Sigma_\rho(n,l,\mathbf{p})F(l,m,\mathbf{p})\right) \right\}.
\end{aligned}
\tag{69}
$$

After fixing the initial conditions $F(0,0,\mathbf{p})$, $F(1,0,\mathbf{p})$ and $F(1,1,\mathbf{p})$, the initial values of ρ are fixed to $\rho(0,0,\mathbf{p}) = 0$, $\rho(1,1,\mathbf{p}) = 0$ and $\rho(1,0,\mathbf{p}) = a_t$ by its anti-commutation relation, (69) allows us to stepwise determine all the correlation functions of the system. The process on how the evolution takes place is illustrated in Fig. 4.

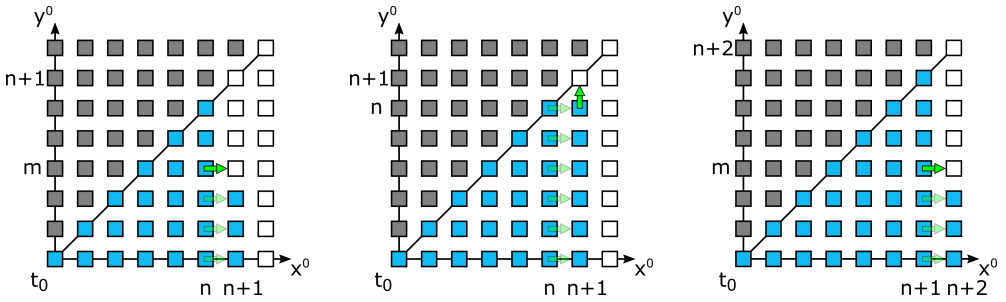


Figure 4: Depicted is the process of the discrete time evolution for statistical function $F(n,m,p)$ from a), b) the n-th to n+1-th time step and further to c) the n+2-th step. Each block represents N_{sp} momentum values of $F(n,m,p)$. Blue blocks have been, white blocks have to be and grey blocks do not need to be calculated. The opaque arrow indicates the value that is currently calculated, the transparent one indicate values that have already been calculated in the current x^0 step.

At the n-th step, the values of the correlation function $F(n,m,p)$ for the n+1-th step are calculated stepwise, starting from $m = 0$ and continuing until $m = n$ (Fig. 4 a)). The opaque arrow in Fig. 4 indicates the value that is calculated for the current m, while the transparent arrows indicate values that have been calculated previously in the same n step. Blue blocks represent values that have already been and white blocks values that will be calculated during the time evolution. The grey blocks indicate possible values of F which do not need to be calculated due to the commutation relations. Each block contains

N_{sp} momentum points. After calculating the values up until $m = n$, the last remaining component is the equal time one for the n+1-th step. By making use of the commutation relation for F (41), one can then calculate the $F(n+1, n+1)$ from $F(n+1, n)$ (Fig. 4 b)). Afterwards, the process starts anew for the n+2-th step (Fig. 4 c)). The time evolution for ρ is done in a similar way. Additionally, due to the anti-commutation relations, the values $\rho(n, n, p)$ and $\rho(n, n-1, p)$ are already fixed and no further calculations are needed for them.

4. Runtime analysis

Solving the Kadanoff-Baym equations is a difficult task. This is especially true for the amount of memory and computation time that is needed for the numerics. The rapidly increasing usage of both memory and computation time the further time evolution goes, makes an analysis of the scaling behavior of them important. In the following, we'll give a brief discussion on this scaling behavior.

4.1. Memory usage

We want to start our discussion with the use of storage. In general, every value of F and ρ that has to be stored is a complex number with double precision. A double precision number takes 8 bytes [17]. A complex double number consists of a real and an imaginary part, both of which are of double precision, and therefore needs 16 bytes. The total amount of data points we have to store for the full calculation is given by

$$N_{data}^{hom}(N_t, N_{sp}, d) = 2 \cdot N_t^2 N_{sp}^d \tag{70}$$

in case of a homogeneous system for N_t time steps and N_{sp} spacial points in d dimensions, where we only consider the 2-point Greens function. For the more general case of n-point functions in an inhomogeneous system, we would need

$$N_{data}^{inhom}(N_t, N_{sp}, d, n) = 2 \cdot \left(N_t N_{sp}^d\right)^n, \tag{71}$$

where n is the number of points for the Greens function. The factor of 2 in both (70) and (71) comes from the need that we have to calculate both F and ρ. The number of data points we have to store, can be reduced by a factor of 2 by making use of the commutation relations. However, this makes the calculations slightly more complicated and increases the computation time. Most of the time, the memory is not the limiting factor, so we kept this factor of 2 for simplicity reasons.

From the amount of data points, we can estimate the total memory that is needed to run the calculation. Suppose we have a system of 150 spacial points and 800 time steps in 1-dimension. For most of the calculations done in this work, such a number of time steps would only be enough to observe the early to intermediate time behavior of the system. To observe the late time behavior, depending on the chosen run parameters, several thousand steps might be necessary. The amount of storage we need for our example model would

already be

$$S = 16 \cdot \text{Bytes} \cdot N_{data}^{hom}(800, 150, 1) = 3.072 \text{ GB}. \tag{72}$$

Considering that the data is stored in the RAM, the usage of the hard drive would be very slow, this is already quite a lot of storage that is needed to perform the computation. If we further take into consideration that the number of data points and therefore the memory usage scales with $\mathcal{O}(N_t^2)$, then it becomes obvious that observing several thousand steps is a very difficult task. It should also be noted, that for our toy model, we only considered one spacial dimension. The case of higher dimensions or for inhomogeneous systems is on a completely different scale and without reduction techniques a hopeless endeavor.

4.2. Computation time

Besides the storage, the computational work load poses another problem to the solution of the KBE. We want to start with the dependency of the computation time on the number of spacial points. The number of spacial points remains constant for each time step throughout the system's time evolution. Each additional spacial point adds the same amount of work load, so the computation time should depend linearly on the total number of spacial points $N_{sp_{total}} = N_{sp}^d$. This dependency can be seen in Fig. 5.

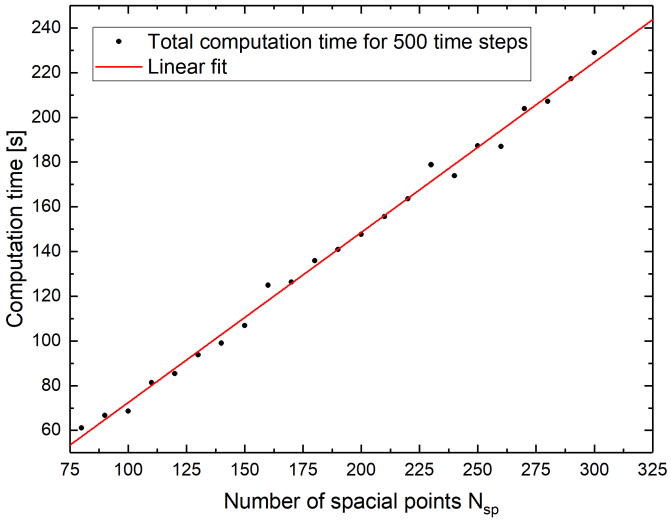


Figure 5: Depicted is the total computation time T_{total} needed to perform 500 time steps, starting from $x^0 = 0$, as a function of the total number of spacial points $N_{sp_{total}} = N_{sp}$. The data is fitted by a linear function, indicated by the red curve.

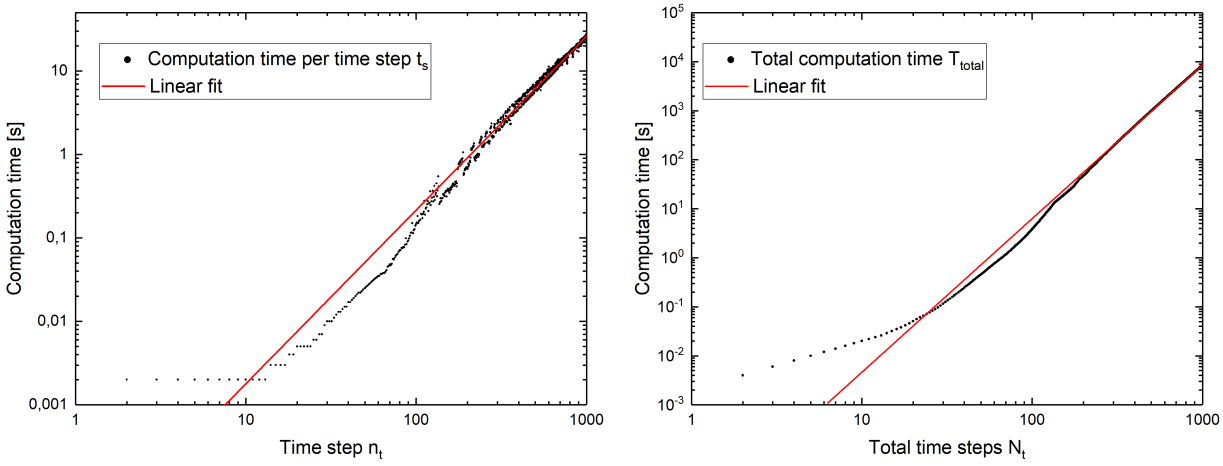
For the benchmark in Fig. 5, we chose to calculate 500 time steps, starting from $x^0 = 0$, and measure the total computation time $T_{total}(N_{sp_{total}})$ that is necessary to perform the

computation. We varied the total number of spacial points $N_{sp_{total}} = N_{sp}^1 = N_{sp}$ from 80 to 300 and ran our calculations on the Julia High Performance Cluster (HPC) using the 8 CPUs of the Intel® Xeon Gold 6134 Processor for one standard node [18]. From Fig. 5, one can see that, as expected, T_{total} closely follows a linear course. To find the concrete dependency of T_{total} on $N_{sp_{total}}$ for the chosen setup, we fit the data with a linear function. The fit is indicated by the red line in Fig. 5 and has a slope of $m_T = (0.761 \pm 0.011)\, s$.

Besides the spacial dependency, we also have to consider the dependency of the computation time per step t_s on the current time step n_t. With each time step, the amount of data points that need to be calculated in the next one is increased by one (see Fig. 4). When we take into account that the memory integrals run over the complete history of the system, we would expect a scaling behavior of

$$t_s(n_t, N_{sp_{total}}) = N_{sp_{total}} \cdot \left(a_0 + a_1 \cdot n_t + a_2 \cdot n_t^2\right) = \left(c_0 + c_1 \cdot n_t + c_2 \cdot n_t^2\right), \qquad (73)$$

where the a_i are constants specific to our solver and the machine the program is running on. Again, we ran a test computation for $N_{sp} = 200$ spacial points up to $n_t = 1000$ on the Julia HPC for the same setup as above. The result can be seen in Fig. 6.



(a) (b)

Figure 6: a) Double logarithmic plot of the computation time per step t_s as a function of the current time step n_t with $N_{sp} = 200$ spacial points for a test computation on the Julia high performance cluster. For $n_t \geq 200$, the data on the double logarithmic scale is fitted linearly (red curve). b) Double logarithmic plot of the total computation time T_{total} needed to perform a total amount of N_t time steps. Again, the data is fitted linearly (red curve) on the double logarithmic scale, in this case for $N_t \geq 200$.

Fig. 6 a), shows the course of t_s as a function of n_t on a double logarithmic scale. We chose this scale, because it is convenient to determine the scaling behavior of t_s later on. One can see that for a $n_t \geq 200$, the graph tends to a linear course. On the other hand, for $n_t < 200$, the linear course gets notably distorted. This behavior is expected when we consider (73). For large n_t, the quadratic term outweighs the others and should almost completely determine the scaling behavior of t_s, which results in a linear course on the double logarithmic scale. In case of smaller n_t, the contribution of the constant and linear term become comparable to the quadratic one and the course of t_s is no longer linear on this scale.

We want to find the scaling behavior of t_s. To do this, we perform a linear fit of $\log(t_s(\log(n_t)))$ for large n_t, i.e. $n_t \geq 200$. This fit is indicated by the red line in Fig. 6 a) and has a slope of $m_t = (2.081 \pm 0.009)$, which suggests a scaling behavior of $\mathcal{O}(n_t^{2.081})$ for t_s. This is slightly larger than the $\mathcal{O}(n_t^2)$ scaling we would expect from (73) for large n_t, even within the error range. The small difference might be caused by parts of our solver, which are not directly related to the solution of the KBE, like the storage of the data on the hard drive.

Besides the computation time necessary for a single step, we also calculated the total time $T_{total}(N_t, N_{sp_{total}})$ needed to perform all N_t time steps. The total computation time can be gained by simply integrating, or in this case summing over t_s for all n_t, i.e.

$$T_{total}(N_t, N_{sp_{total}}) = \int_0^{N_t} dn_t \; t_s(n_t, N_{sp_{total}}) \rightarrow \sum_{n_t=0}^{N_t} t_s(n_t). \tag{74}$$

The course of the total computation time can be seen in Fig. 6 b), where we again chose a double logarithmic scale. We can see that, similar to t_s, T_{total} describes a linear course for $N_t \geq 200$ that gets distorted for $N_t < 200$. Due to the summation, we would expect that the scaling order of T_{total} is one order higher than that of t_s and should therefore be $\mathcal{O}(N_t^3)$. Again, we performed a linear fit, this time of $\log(T_{total}(\log(N_t)))$ for $N_t \geq 200$. The fit is indicated by the red line in Fig. 6 b) and has a slope of $m_t = (3.133 \pm 0.001)$. We can see that the scaling behavior of T_{total} has indeed increased by roughly one order compared to t_s. The deviation from the expected $\mathcal{O}(N_t^3)$ are again probably caused by a part of the solver that is not directly related to the solution of the KBE.

From Fig. 6, the problem of solving the KBE becomes obvious. For the time spacing $a_t = 0.1$ (a time stepping we frequently use in this work), that we chose to perform this test computation with, we can only observe the system for a relatively small time frame of $x^0_{max} = N_t \cdot a_t = 100$. Being able to observe this small time frame for the full computation

took us roughly 2.3 h. Later on in this work, we'll consider $x^0 > 1000$ using a reduction scheme to observe thermalization. From the fit of T_{total}, that we performed previously, we can extrapolate the time it would take us for a full computation of the KBE up to only $x^0 = 1000$. In fact, it would would take us roughly 100 days to perform such a computation for the setup we used above. Needless to say, that is not an acceptable time consumption. Because of this huge time consumption, we need a reduction scheme for the computation of the KBE.

We want to close this section with a note about parallelization. One could argue that the long computation time could be shortened by a sufficient parallelization and enough CPUs. An elaborate parallelization was done in [3], where the time history of the system is split into parallelized strips. The computations for these strips are parallelized and each of these parallelized units exchanges values of F and ρ with one another, that are needed for further calculations (for more details see [3]). However, a completely independent parallelization of these strips is not possible due to the appearance of the memory integrals in the calculations. Even when the time evolution is split into separate computation blocks, each block still has to wait until the necessary data for its computation is produced. A parallelization for the time evolution with respect to the time arguments is therefore only partially possible, but for the momentum modes, the case is different. After performing the computation of the effective mass and the self energies at each x^0 step, the remaining calculations for each momentum mode are independent from the others. This enables one to, with the exception of the calculation of the effective mass and the self energies, fully parallelize the time evolution for each momentum mode. This is what we did in our solver.

5. Model

In this section, we want to introduce the model we use for testing our approach to reduce the necessary computational workload to solve the KBE. For our test model, we consider a quantum theory for a real scalar field φ with one component and a φ^4-coupling in one spacial dimension. The classical action has a O(1)-symmetry and is given by

$$S[\varphi] = \int_{\mathcal{X}} dx \; \frac{1}{2}\partial^\mu \varphi(x)\partial_\mu \varphi(x) - \frac{m^2}{2}\varphi(x)^2 - \frac{\lambda}{4!}\varphi(x)^4, \tag{75}$$

where $\int_{\mathcal{X}}$ is a shorthand notation for $\int_{\mathcal{C}} dx^0 \int d\boldsymbol{x}$, like in sec. 2.3 (see also [19]). The coupling is denoted as λ and the bare mass parameter is given by m.

In sec. 2.3 and 2.4, we discussed the derivation of the 2PI-effective action. We performed a loop expansion, resulting in

$$\begin{aligned}\Gamma[\varphi, G] &= S[\varphi] + \frac{i}{2}\mathrm{Tr}_{\mathcal{C}}\ln G^{-1} + \frac{i}{2}\mathrm{Tr}_{\mathcal{C}} G_0^{-1}(\varphi)G + \Gamma_2[\varphi, G] + const \\ &= \Gamma[\varphi, G]^{1\mathrm{Loop}} + \Gamma_2[\varphi, G] + const.\end{aligned}$$

This led to the Kadanoff-Baym equations (43) and (44), which are exact for the complete knowledge of the self-energies Σ_F and Σ_ρ. However, their exact expression is given by the infinite sum of diagrams

$$\Gamma_2[\varphi, G] = \Gamma_2^{2\mathrm{Loop}}[\varphi, G] + \Gamma_2^{3\mathrm{Loop}}[\varphi, G] + ... \tag{76}$$

contained in $\Gamma_2[\varphi, G]$, which can only be computed approximately. For the purpose of this work, we only compute (76) up to the 3-loop order, which corresponds to the two diagrams shown in Fig. 7 (see [10]).

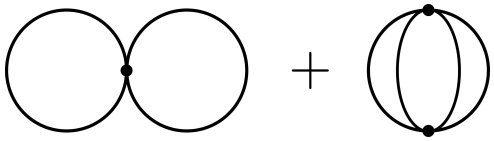


Figure 7: 2- and 3-loop vacuum diagrams, which contribute to the 2PI-effective action.

The 3-loop diagram is already enough for the system to take scattering processes from the memory integrals into account [15].

From the loop diagrams Fig. 7, we can derive the self-energy using (36). The derivative of (76) with respect to G reduces the loop order of the diagrams by one, resulting in the new diagrams shown in Fig. 8.

Figure 8: 1- and 2-loop diagrams that contribute to the self-energy.

This approximation is called the setting-sun approximation [3]. The 2-loop diagram (setting-sun diagram) in Fig. 8 leads to the equations

$$\Sigma_F(x,y) = -\frac{\lambda^2}{6}\left(F(x^0,y^0,|\mathbf{x}-\mathbf{y}|)^3 - \frac{3}{4}\rho(x^0,y^0,|\mathbf{x}-\mathbf{y}|)^2 F(x^0,y^0,|\mathbf{x}-\mathbf{y}|)\right), \tag{77}$$

$$\Sigma_\rho(x,y) = -\frac{\lambda^2}{6}\left(3F(x^0,y^0,|\mathbf{x}-\mathbf{y}|)\rho(x^0,y^0,|\mathbf{x}-\mathbf{y}|) - \frac{1}{4}\rho(x^0,y^0,|\mathbf{x}-\mathbf{y}|)^3\right), \tag{78}$$

for the decomposed self-energy. Note that these self-energies have to be calculated in real space. Because we consider the system in momentum space, we have to Fourier transform the correlation function F and ρ before computing (77) and (78). This can be done with a fast Fourier transform (FFT) routine like for example the FFTW [16], which we used.

The 1-loop diagram in Fig. 8 causes a shift in the system's mass, leading to the effective mass [15]

$$M(x^0,F)^2 = m^2 + \int \frac{d\mathbf{p}}{2\pi} F(x^0,x^0,\mathbf{p}). \tag{79}$$

The mass term m^2 can be calculated at the initial time $x^0 = 0$ from

$$M(0,F)^2 = M_{Init}^2 = m^2 + \int \frac{d\mathbf{p}}{2\pi} F(0,0,\mathbf{p}), \tag{80}$$

where M_{Init} is a free initial mass parameter. Putting everything together, the KBE in the

setting-sun approximation for a homogeneous system read

$$\begin{aligned}
&\left[\partial_{x^0}^2 + \boldsymbol{p}^2 + m^2 + \int \frac{d\mathbf{p}}{2\pi} F(x^0, x^0, \mathbf{p})\right] \rho(x^0, y^0, \boldsymbol{p}) \\
=&\frac{\lambda^2}{6}\frac{1}{2\pi^d}\int_{y^0}^{x^0} dz^0 \int d\boldsymbol{z}\, e^{i\boldsymbol{p}\boldsymbol{z}} \left(3F(x^0, z^0, \boldsymbol{z})\rho(x^0, z^0, \boldsymbol{z}) - \frac{1}{4}\rho(x^0, z^0, \boldsymbol{z})^3\right) \rho(z^0, y^0, \boldsymbol{p}),
\end{aligned}$$

$$\begin{aligned}
&\left[\partial_{x^0}^2 + \boldsymbol{p}^2 + m^2 + \int \frac{d\mathbf{p}}{2\pi} F(x^0, x^0, \mathbf{p})\right] F(x^0, y^0, \boldsymbol{p}) \\
=&\frac{\lambda^2}{6}\frac{1}{2\pi^d}\left(\int_0^{x^0} dz^0 \int d\boldsymbol{z}\, e^{i\boldsymbol{p}\boldsymbol{z}} \left(3F(x^0, z^0, \boldsymbol{z})\rho(x^0, z^0, \boldsymbol{z}) - \frac{1}{4}\rho(x^0, z^0, \boldsymbol{z})^3\right) F(z^0, y^0, \boldsymbol{p})\right. \\
&\left. - \int_0^{y^0} dz^0 \int d\boldsymbol{z}\, e^{i\boldsymbol{p}\boldsymbol{z}} \left(F(x^0, z^0, \boldsymbol{z})^3 - \frac{3}{4}\rho(x^0, z^0, \boldsymbol{z})^2 F(x^0, z^0, \boldsymbol{z})\right) \rho(z^0, y^0, \boldsymbol{p})\right),
\end{aligned}$$

where $\boldsymbol{z} = |\boldsymbol{x} - \boldsymbol{y}|$. In the following, we'll look at the numerical results from these equations for different initial conditions.

We have seen that the KBE beyond 1-loop corrections in principle demand that one has to take the system's complete history and with it all unequal time correlation functions that build up inside the system at $x^0 > 0$ into account. However, in sec. 4, we have also seen that for late time observations $x^0 > 1000$, using the system's whole history is a hopeless endeavor. This chapter should introduce the model and provide the reader with a sense of what to expect for the different initial conditions that we'll use throughout this work. In this section, as well as sec. 6, we therefore already make use of the reduction scheme, which will be introduced later on in sec. 7, to enable the reader to get an understanding of the system's late time evolution.

5.1. Quench initial conditions

The first set of initial conditions we want to discuss are the quench initial conditions. The idea behind the quench is to initialize the system in an equilibrium state, where the statistical particle density is given by the Bose-Einstein-distribution

$$n_p(0) = \frac{1}{e^{\frac{\omega}{T_0}} - 1}, \qquad \omega = \sqrt{p^2 + M_0^2}, \qquad M_0^2 = 2 \cdot M_{Init}^2, \tag{81}$$

with an initial temperature T_0 and an initial effective mass M_0. Through an instantaneous drop of the effective mass square M_0^2 from $2 \cdot M_{Init}^2$ to M_{Init}^2, the system is brought out of its equilibrium state and one can then observe its relaxation process (see [15]).

Before we discuss the results, we still need to add one piece of information. That is, how to initialize the statistical correlation function $F(0,0,\boldsymbol{p})$, $F(1,0,\boldsymbol{p})$ and $F(1,1,\boldsymbol{p})$ mentioned in sec. 3. The initial conditions can be derived from the equations [15]

$$F(0,0,p) = \frac{n_p(0)+1/2}{\omega}, \quad \partial_t F(t,0,p)|_{t=0} = 0, \tag{82}$$

$$F(0,0,p)\partial_t\partial_{t'}F(t,t',p)|_{t=t'=0} = \left(n_p(0)+1/2\right)^2. \tag{83}$$

By discretizing the partial derivatives, one can then calculate the initial values for F, which are given by

$$F(0,0,p) = \frac{n_p(0)+1/2}{\omega}, \tag{84}$$

$$F(1,0,p) = \frac{n_p(0)+1/2}{\omega}\cdot\cos\left(\omega a_t\right), \tag{85}$$

$$F(1,1,p) = \frac{n_p(0)+1/2}{\omega}. \tag{86}$$

Remember that because of the anti-commutation relations it follows that $\rho(0,0,p) = 0$, $\rho(1,0,p) = a_t$ and $\rho(1,1,p) = 0$. Equipped with these initial conditions, we can now discuss the quench case.

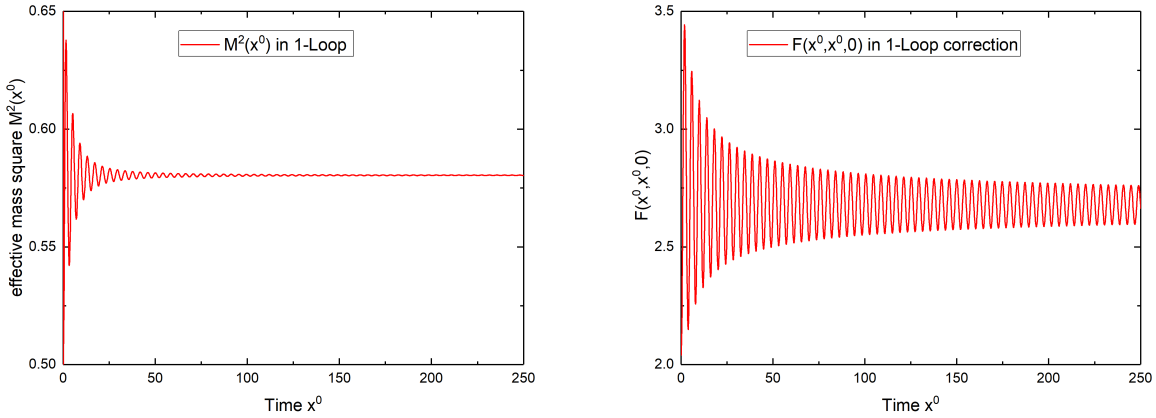
5.1.1. 1-loop approximation

We want to start our discussion by only considering the 1-loop case for the quench. In this case, there are no memory integrals and thus scattering does not occur. Nonetheless, it'll be useful later on for our discussion of numerical effects.

When the system is brought out of equilibrium, the effective mass begins to oscillate rapidly, which can be seen in Fig. 9 a). The oscillation is damped fast and the system approaches an asymptotically constant new mass. Naively one might think, that the system has approached a new equilibrium state, however, this is not the case. If the system were to approach an equilibrium state, it would become invariant under time translation. This in turn would mean that $F(x^0,y^0) \to F(|x^0-y^0|)$, which, in case of the 1-loop approximation would result in a constant mass $M^2(x^0,F) = M^2_{Init}$ (see [15] p. 25-26). This can be seen from (79) and (80), which, when taking the time translation invariance into account, leads to

$$M^2_{eq}(x^0,F) = m^2 + \int\frac{d\boldsymbol{p}}{2\pi}F(\left|x^0-x^0\right|,\boldsymbol{p}) = m^2 + \int\frac{d\boldsymbol{p}}{2\pi}F(0,\boldsymbol{p}) = M^2_{Init}. \tag{87}$$

However, the system's new mass $M^2(x^0 = 250) = 0.58$ differs from the mass $M^2_{eq} = 0.5$, we would expect in thermal equilibrium for our chosen initial mass of $M^2_{Init} = 0.5$. The system does not thermalize.



(a) (b)

Figure 9: Time evolution of a) the effective mass $M(x^0)$ as well as b) the equal time correlation function $F(x^0, x^0, 0)$ for the quench initial conditions in the 1-loop approximation up to $x^0 = 250$. For this simulation, we used the settings $\mathcal{S}_1$, which can be found in appendix A.

That the system does not approach thermal equilibrium in the 1-loop approximation can be seen nicely in Fig. 9 b). After the quench, the system oscillates with a high amplitude. The amplitude of the equal time correlation function $F(x^0, x^0, 0)$ is rapidly damped, though at later times, when the effective mass has approximately become a constant, $F(x^0, x^0, 0)$ is still oscillating and the damping tends to zero. The system doesn't reach equilibrium.

5.1.2. 2-loop setting-sun approximation

In the setting-sun approximation, the calculation becomes significantly more costly. In the case of 2-loop corrections, the memory integrals come into play. At times $x^0 > 0$, unequal time correlations build up inside the system and scattering starts to take place. The scattering effects enable the system to thermalize, which can be seen in Fig. 10.

After the quench, $F(x^0, x^0, 0)$ shoots up, similar to the 1-loop case (see insert Fig. 10). However, the oscillation is rapidly damped down and this initial oscillatory phase ends at about $x^0 = 20$. Afterwards, the system goes into a much longer, smooth drifting phase, where the zeroth mode significantly increases. The drifting phase ends at roughly $x^0 = 500$ and the system slowly transitions into thermal equilibrium, where no more

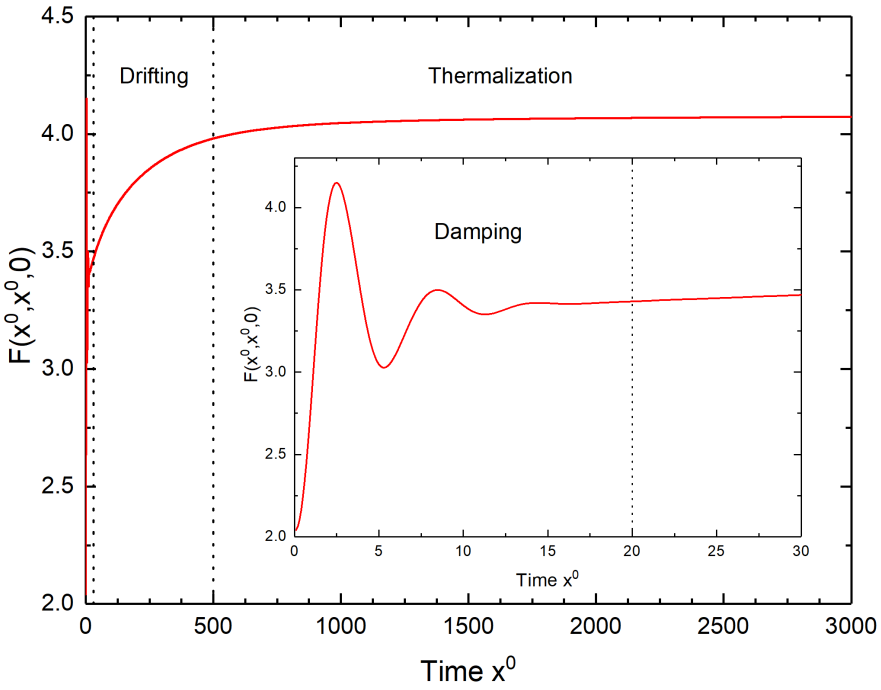


Figure 10: Course of the equal time zeroth mode $F(x^0, x^0, 0)$ for a quench in the setting-sun approximation. The inserted graph shows the early time behavior from $x^0 = 0$ to 30. After the early oscillatory phase, the system transitions into a drifting phase. At roughly $x^0 = 500$ the drifting phase ends and thermal equilibrium is approached. The settings $\mathcal{S}_1$ were used to observe this course.

changes occur. To be sure of the stability, we explicitly investigated the time evolution to up to $x^0 = 6000$, as well as for different momentum modes. Up to this time, the system didn't seem to leave its equilibrium state again.

5.2. Tsunami initial conditions

Now, we want to discuss a different set of initial conditions, the so-called tsunami. For the tsunami, the initial particle distribution has the form of a Gaussian peak [15]

$$n_0(\mathbf{p}) = A \exp\left(-\frac{(|\mathbf{p}| - p_{ts})^2}{2\sigma}\right), \tag{88}$$

with an Amplitude A, a peak width σ and a center at $p = p_{ts}$. In Fig. 11, the initial particle distribution is shown as a function of the 1-dimensional momentum p.

The distribution is symmetric under $p \to -p$ and one can think of such initial conditions as two colliding wave packages in one dimension. The effective mass at the initial time $x^0 = 0$ is given by $M(0, F) = M_{Init}$ and the renormalized mass can be determined through

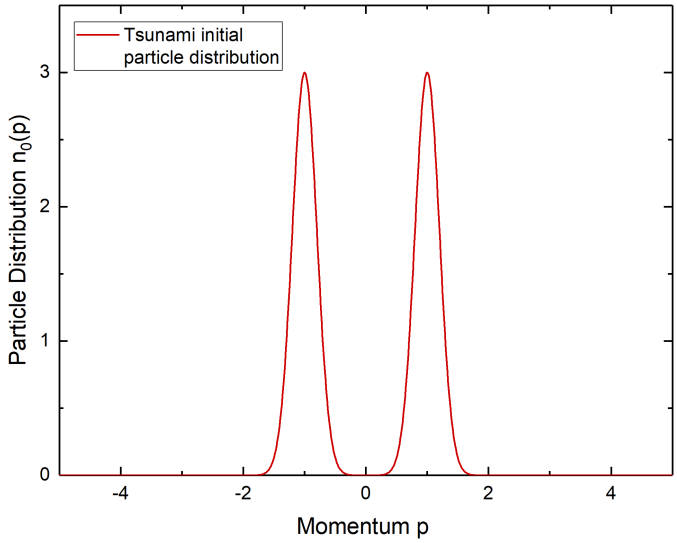


Figure 11: Initial particle distribution $n_0(p)$ for the tsunami initial conditions as a function of the momentum. The distribution is peaked around $p = p_{ts} = 1$ with an amplitude of $A = 3$ and a width of $\sigma = 0.2$.

(80) ([15] p. 35). The initial values for F and ρ are the same as for the quench.

5.2.1. 1-loop approximation

The 1-loop case for the tsunami initial conditions is extremely plane. That is, because there is neither scattering, nor is there a forced change in the effective mass. The two wave packages do not interact and all quantities of the system remain constant. From our previous argument for the 1-loop quench, one can conclude that the constancy of the effective mass indicates a time translation invariant state of the system. Due to the missing interaction, the system has therefore reached its thermalized state from the very beginning.

5.2.2. 2-loop setting-sun approximation

Other than the 1-loop case, the 2-loop one is very interesting. In case of the tsunami, there is another momentum mode, which is especially interesting besides the zeroth momentum one. That is the mode p_{peak} where the peak of the Gaussian particle distribution lies. Fig. 12 and 13 show the course of modes $p = \{0, p_{peak}, p_{max}\}$ for the particle distribution given in Fig. 11.

In Fig. 12 we show the early time behavior (damping phase) for the tsunami. The damping phase ranges from $x^0 = 0$ to roughly 25 and makes up only a very small part of

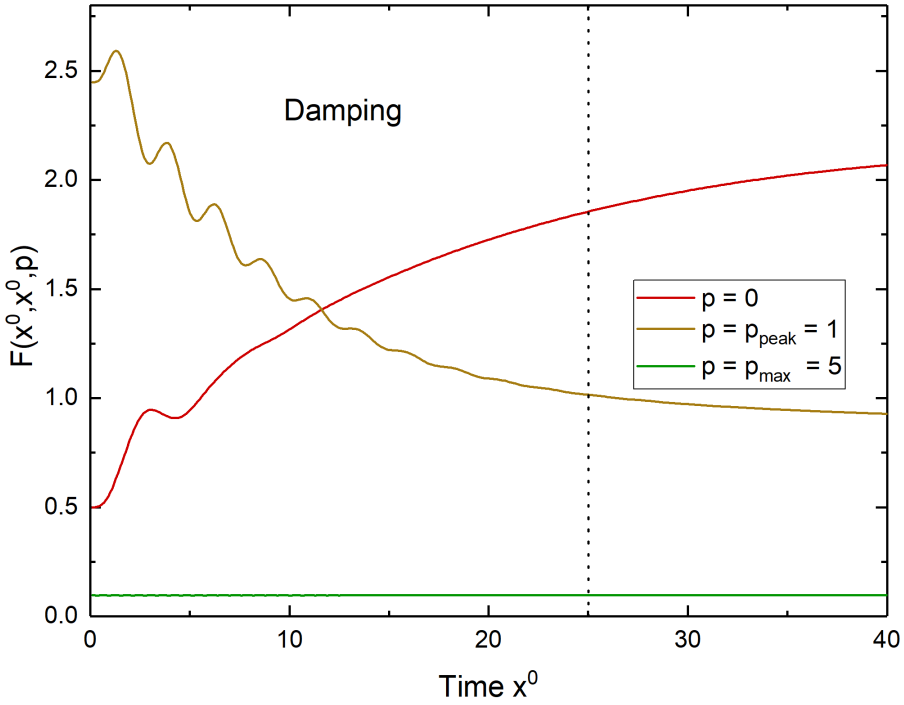


Figure 12: Early time behavior of the equal time correlation function $F(x^0, x^0, p)$ for the tsunami initial condition and the momentum modes $p = 0$ (red), $p = p_{peak} = 1$ (brown) and $p = p_{max} = 5$ (green). The dotted line indicates the estimated end of the early time damping phase. The settings we used for this plot are given in $\mathcal{S}_1$ (appendix A).

the total time evolution. After one small oscillation, the zero mode immediately shows a drifting behavior. One might estimate the time frame of the damping phase even smaller, however, we chose $x^0 = 25$ because of the peak momentum mode. While the zero mode significantly increases at early times, the peak mode on the other hand decreases rapidly. The change of the maximal momentum mode $p_{max} = 5$ is minuscule. However, even for the high momentum modes , one can observe a growth, that can be seen in Fig. B.1 (see appendix B.1).

After the damping phase, similar to the quench, the system transitions into the drifting phase (Fig. 13). This phase is significantly longer than the previous one, ranging to about $x^0 = 1000$. One can see, that the change in the peak momentum mode during the drifting phase become small rather fast. On the other hand the zero momentum mode shows an interesting behavior. The initial rapid increase of the zero mode causes it to overshoot. At $x^0 = 60$, it seems to reach a peak value. From this point, the mode slowly decreases drifts to a lower value, until the system only shows minor changes. At this point the system transitions to the thermalization phase.

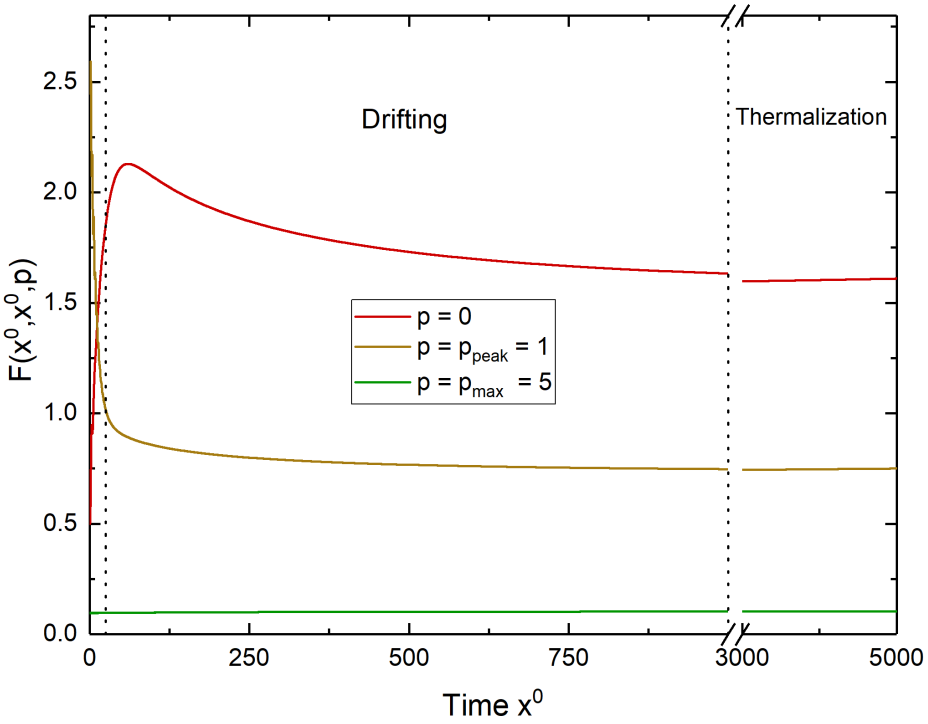


Figure 13: Full time evolution up to $x^0 = 5000$ of the equal time correlation function $F(x^0, x^0, p)$ for the tsunami initial condition and the momentum modes $p = 0$ (red), $p = p_{peak} = 1$ (brown) and $p = p_{max} = 5$ (green). The evolution can be split into three phases. The damping (see Fig.12), drifting and the thermalization phase. The settings we used for this plot are given in $\mathcal{S}_1$ (appendix A).

The transition between the drifting and the thermalization phase can be seen best from the highest momentum mode p_{max}, whose course is shown in Fig. B.1. After a long oscillation phase, the mode also shows a drifting behavior, with a minor, but steep increase. At $x^0 = 1000$, the steep increase flattens and even the highest momentum mode only shows slow, slight changes, indicating its gradual approach to thermal equilibrium.

Particle distribution

As we saw in the introduction to nonequilibrium systems, one can derive macroscopic variables from the correlation functions of the system. In case of the tsunami, the course of these macroscopic variables is more interesting then it is for the quench, which is why we didn't look at them previously.

We slightly change the tsunami initial condition by adding a thermal background given by a Bose-Einstein-distribution

$$n_0(p) = \frac{1}{e^{\epsilon_0(p)/T_0} - 1}, \qquad \epsilon_0 = \sqrt{p^2 + M_0^2}. \tag{89}$$

The tsunami peak is simply added on top of the thermal background, leading to the new initial particle distribution

$$n_0(p) = A \exp\left(-\frac{(|p| - p_{ts})^2}{2\sigma}\right) + 1/\left(\exp\left(\epsilon_0(p)/T_0\right) - 1\right). \tag{90}$$

The tsunami on a fixed thermal background enables us to do a comparison between the system's background temperature at the initial time and the temperature which the system reaches after thermalization. First, we need an expression for the time dependent particle distribution $n(x^0, p)$ of the system. The time dependent particle distribution of the system can be calculated via

$$n(x^0, p) = \left(F(x^0, y^0, p)\partial_{x^0}\partial_{y^0}F(x^0, y^0, p) - \left(\partial_{x^0}F(x^0, y^0, p)\right)^2\right)^{\frac{1}{2}}\Big|_{x^0=y^0} - \frac{1}{2}. \tag{91}$$

For the 1-loop approximation, this is a conserved quantity. However, in the setting-sun approximation, the particle distribution does not remain constant due to scattering. Besides $n(x^0, p)$, we also need the time dependent energy density $\epsilon(x^0, p)$, given by

$$\epsilon(x^0, p) = \left(\frac{\partial_{x^0}\partial_{y^0}F(x^0, y^0, p)}{F(x^0, y^0, p)}\right)^{\frac{1}{2}}\Big|_{x^0=y^0} \tag{92}$$

for our discussion. We evaluated the derivatives $\partial_{x^0}\partial_{y^0}F(x^0, y^0, p)$ as well as $\partial_{x^0}F(x^0, y^0, p)$, that are used in (91) and (92), using finite differences

$$\begin{aligned}
\partial_{x^0}F(x^0, y^0) &\to \frac{F(x^0 + a_t, y^0) - F(x^0 - a_t, y^0)}{2a_t} \\
\partial_{x^0}\partial_{y^0}F(x^0, y^0) &\to \frac{F(x^0 + a_t, y^0 + a_t) - F(x^0 + a_t, y^0 - a_t) - F(x^0 - a_t, y^0 + a_t) + F(x^0 - a_t, y^0 - a_t)}{4(a_t)^2}.
\end{aligned}$$

For the tsunami part of our initial conditions, we chose $A = 4,\ p_{ts} = 2.5$ and $\sigma = 0.1$. The full set of settings, which we used for the rest of this section, can be found in appendix A, set $\mathcal{S}_2$.

When plotting snapshots of $\ln\left(1 + 1/n(x^0, p)\right)$ for various times against the energy density $\epsilon(x^0, p)$, we get the graph shown in Fig. 14.

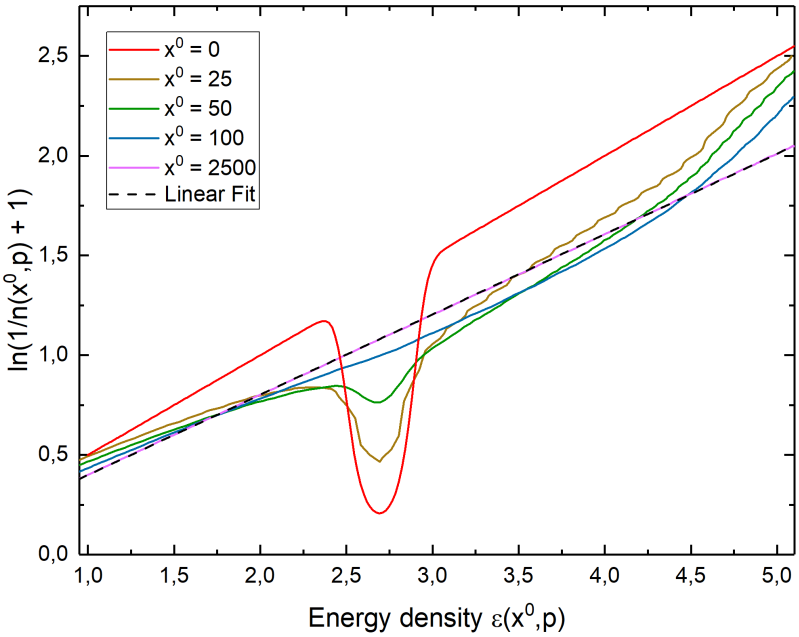


Figure 14: The graph shows snapshots of the course of $\ln(1 + 1/n(x^0, p))$ as a function of the energy density $\epsilon(x^0, p)$ for different times during the evolution of the tsunami. The initially deep distortion of the tsunami peak is quickly flattened. After a much longer time, the system has approached a new equilibrium state, indicated by the pink line. For this equilibrium state, a linear fit (black dashed line) was performed and a new equilibriums temperature was determined.

The red line shows the the course of $\ln(1 + 1/n_0(p))$ for the initial particle distribution $n_0(p)$ with a thermal background of temperature $T_0 = 2$. For $p \ll p_{peak}$ and $p \gg p_{peak}$, the graphs course is linear and thus following a Bose-Einstein-distribution, as expected from the thermal background. Around $\epsilon(x^0, p_{peak})$ the graph dips down, indicating the tsunami peak. The lines brown to pink show snapshots of the particle distribution for different times. One can clearly see that the tsunami peak flattens rapidly and deforms the part of the system's thermal background that has previously been undisturbed. As time passes $\ln(1 + 1/n(x^0, p))$ approaches a linear course, which can be seen from the pink line that shows $\ln(1 + 1/n_0(p))$ at a time $x^0 = 2500$. At this point, the system has completely approached a new equilibrium state and follows a Bose-Einstein distribution. We performed a linear fit (black dashed line) of the final state, resulting in a slope of $m_T = (0.40256 \pm 0.00005)$, which fits the pink line very well. By taking the inverse of slop, we can calculate a new equilibrium temperature $T_{eq} = (2.4841 \pm 0.0003)$. The energy of the tsunami was transfered to the thermal background, resulting in a rise in temperature of $\Delta T = (0.4841 \pm 0.0003)$.

Linear entropy

Another quantity which is directly connected to the particle distribution for a nonequilibrium Gaussian system is the quantum purity, which we calculated in sec. 2.5.2. From the quantum purity one can then derive the linear entropy (61). The quantum purity and thus the entropy of a nonequilibrium evolution is a interesting quantity that, similar to the particle density (cf. [5]), enables us to characterize the systems degree of thermalization. For this reason, we'll take a closer look at the entropy during the time evolution of a tsunami with a thermal background, but before that, we first want to briefly show the analogies between our model and the Gaussian state we discussed in sec. 2.5.2.

For the real scalar theory we chose, we define an observer that can only measure equal time two-point functions (cf. [5], [12])

$$F_p(x^0) = \langle \Phi_p(x^0)\Phi_p(x^0)\rangle = F(x^0,y^0,p)\big|_{x^0=y^0}\,, \tag{93}$$

$$R_p(x^0) = \frac{1}{2}\langle \big[\Phi_p(x^0)\Pi_p(x^0)\big]_+\rangle = \partial_{x^0}\partial_{y^0}F(x^0,y^0,p)\big|_{x^0=y^0}\,, \tag{94}$$

$$K_p(x^0) = \langle \Pi_p(x^0)\Pi_p(x^0)\rangle = \partial_{y^0}F(x^0,y^0,p)\big|_{x^0=y^0}\,, \tag{95}$$

which are the relevant observables of the least biased density D_0 (see sec. 2.5.2) for the system at hand. One can now split this least biased state into subsystems a, where each subsystem has a set of local observables $\mathcal{A}_{ia}$ and is described by an effective Gaussian density matrix [13]. In our case, these subsystems correspond to the different momentum modes p and we can decompose the least biased state to $D_0(x^0) = \prod_p D_p(x^0)$ with

$$D_p(x^0) = \frac{1}{Z}\exp\left(-\kappa_p(x^0)\left(K_p(x^0)q^2 - R_p(x^0)\,[q,p]_+ + F_p(x^0)p^2\right)\right). \tag{96}$$

This corresponds to the shifted general Gaussian density matrix (49) for (cf. [5])

$$\alpha(x^0) = -2\kappa_p(x^0)K_p(x^0), \quad \beta(x^0) = 2\kappa_p(x^0)R_p(x^0), \quad \gamma(x^0) = -2\kappa_p(x^0)F_p(x^0). \tag{97}$$

For a Gaussian state, as we saw in sec. 2.5.2, the quantum purity of the system is then simply given by

$$\mathrm{Tr}\left(D_p^2(x^0)\right) = \frac{1}{2n_p(x^0)+1}, \tag{98}$$

where $n_p(x^0)$ is the quasi-particle density, and the linear entropy by

$$S_L(x^0,p) = 1 - \mathrm{Tr}\left(D_p^2(x^0)\right). \tag{99}$$

The quasi-particle density is given by [5]

$$n_p(x^0) + \frac{1}{2} = \sqrt{F_p(x^0)K_p(x^0) - R_p^2(x^0)}. \tag{100}$$

Inserting (93)-(95) into (100) then leads to the same expression of the particle distribution that we gave in (91). With the quantum purity at hand, we can now calculate the linear entropy $S_L(x^0, p)$ of each mode. In Fig. 15 we, present the time evolution of the entropy for the three different momentum modes $p = \{0, p_{\text{peak}}, p_{max}\}$, as well as the average linear entropy per mode

$$\bar{S}_L(x^0) = \frac{1}{N_{sp}} \sum_{i=0}^{N_{sp}} S_L(x^0, i), \tag{101}$$

for the thermal background tsunami up to $x^0 = 2500$.

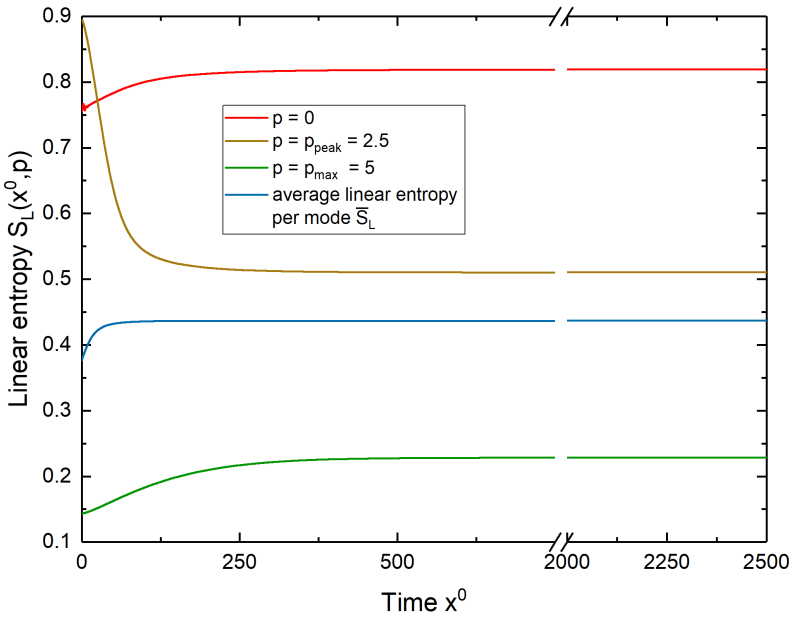


Figure 15: Course of the linear entropy during the time evolution of a tsunami with thermal background for different momentum modes (red, brown, green), as well as for the average entropy per mode $\bar{S}_L(x^0)$ (blue) up to $x^0 = 2500$. After a time interval of relatively fast change, the graphs approach a constant course, where no more changes occur.

Initially, the peak momentum mode is highly occupied, which also leads to a high entropy. As the peak mode begins to lose occupation due to scattering, the peak modes entropy rapidly decreases. This might seem somewhat problematic given the fact that the entropy should increase monotonously. However, while the entropy of the peak mode decreases, low and high momentum modes record a significant increase in linear entropy. As a matter of fact, when looking at the average entropy per mode, illustrated by the blue line

in Fig. 15, one can see that the system's total entropy increases. This is in agreement with the second law of thermodynamics and is to be expected from a thermodynamic system when approaching thermal equilibrium. After the relatively quick changing period, at roughly $x^0 = 500$, the entropy for all the momentum modes, as well as the average entropy of the system, have become constant, indicating the system's approach to thermal equilibrium. Even up to $x^0 = 2500$, no more changes in the linear entropy can be observed.

6. Numerical effects

The knowledge of the numerical effects that can influence the evolution of the system, as well as the numerical stability of the algorithm, play an important role in choosing meaningful parameters and the interpretation of the results at hand.

6.1. Finite lattice effects

The numerical effects in the algorithm are a result from the lattice discretization one has to perform. For this reason, the parameters we're most interested in are the ones that determine the lattice discretization, i.e. the lattice spacings a_t (time spacing), a_s (spacial spacing) and the spacial volume V_s of the lattice.

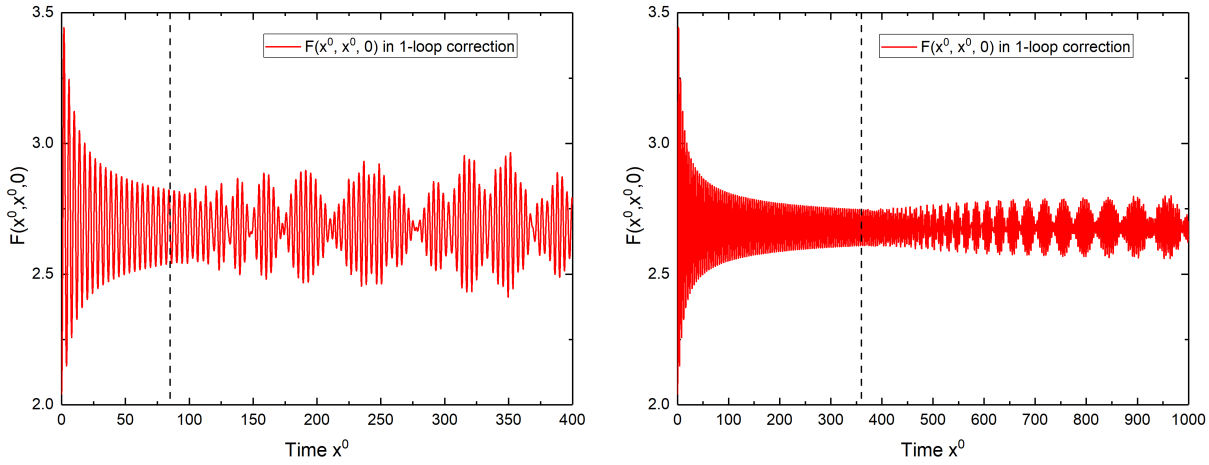
We'll begin our discussion with the observation of finite lattice effects. To produce the data for this section, we used the settings $\mathcal{S}_3$ (appendix A). The finite lattice effects result from the finite length, or volume V_s of the lattice and can cause disturbances in the system. The significance of these effects differs for 1- and 2-loop corrections, which is why we'll discuss them separately in the following.

6.1.1. 1-loop approximation

At first, we consider the case of 1-loop corrections and restrict ourselves to the quench initial condition, because, as we saw in sec. 5.2, the tsunami initial condition for 1-loop corrections is rather uninteresting.

In Fig. 16 we show the equal time correlation function $F(x^0, x^0, 0)$ of a quench for two different lattice sizes $V_{s_a} = 160$ and $V_{s_b} = 640$.

In both cases, the system begins to show a beat-like behavior, which occurs at about $x^0 = (90 \pm 5)$ for a) and $x^0 = (370 \pm 10)$ for b), respectively. As one can see, the disturbances scale with the size of the lattice. To find the dependency at which point the disturbance occurs, we varied the lattice size from 160 to 640 in steps of 80 and observed at which point in time the system gets disturbed. We found that the occurrence of the disturbance follows a linear dependency with $x^0_{\text{disturbance}} = (0.573 \pm 0.018) \cdot V_s$. These disturbances are the result of the superposition for the different momentum modes during the calculation of the effective mass. When calculating the effective mass, the system is damped due to the overlapping of the $F(x^0, x^0, p)$ which oscillate with different phases.



(a) (b)

Figure 16: Equal time correlation function $F(x^0, x^0, 0)$ of the 1-loop quench for two different total spacial volumes a) $V_{s_a} = 160$ b) $V_{s_b} = 640$. In the beginning, $F(x^0, x^0, 0)$ behaves like we've seen in sec. 5.1.1, but begins to show a unexpected beat-like behavior at a) $x^0_{\text{disturbance}_a} = (90 \pm 5)$ and b) $x^0_{\text{disturbance}_b} = (370 \pm 10)$, indicated by the vertical dashed black lines.

After some time, the systems phase information is recovered, causing a reversion of the initial damping (see [15] p.23-24).

6.1.2. 2-loop setting-sun approximation

For the 1-loop case, we found that the system gets disturbed by finite lattice effects after $x^0 = (0.573 \pm 0.018) \cdot V_s$. In the case of the 2-loop corrections, which introduces scattering effects, such a behavior can't be observed. Due to the scattering effects, disturbances from the finite lattice are suppressed with increasing lattice size (see [15] p. 24). This fact can be seen in Fig. 17, where we show the course of $F(x^0, x^0, 0)$ for quench initial conditions (the case of tsunami initial conditions can be found in Fig. B.2 in the appendix B.2) in the 2-loop approximation, for different lattice sizes.

We varied the lattice size from $V_s = 10$ to $V_s = 150$. From Fig. 17, one can clearly see, that for small lattice sizes $V_s \leq 20$, the system shows noticeable deviations in its course. However, the bigger the lattice size gets, the smaller the deviations become. For a volume of $V_s = 40$ there are already no more visible differences compared to the course of the graph with the much higher volume $V_s = 150$.

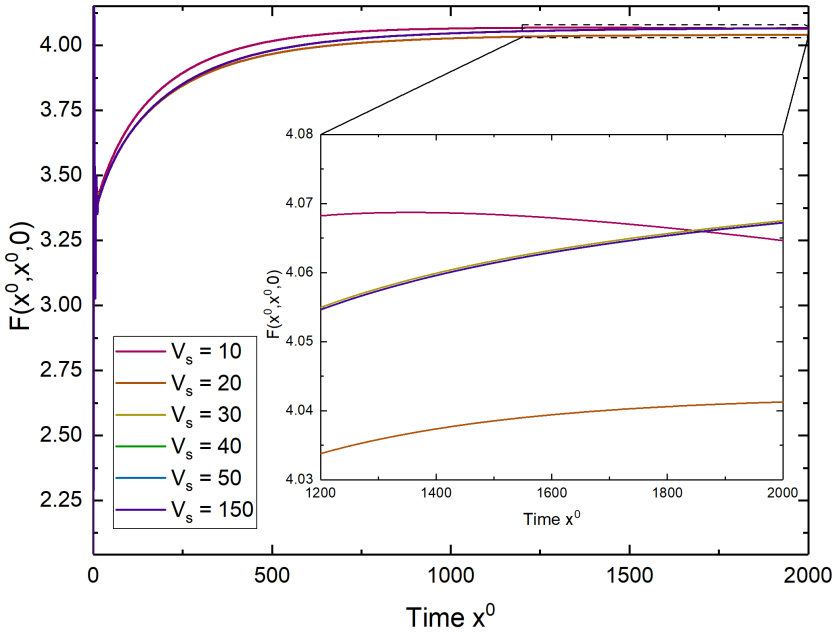


Figure 17: Depicted is the course of the equal time correlation function $F(x^0, x^0, 0)$ for quench initial conditions and different lattice sizes V_s, ranging from $V_s = 10$ to 150. The insert shows a magnification of the graph from $x^0 = 1200$ to 2000.

To get a better grasp on the deviations between the different runs, we want to quantify them. We'll do this by calculating the mean squared error

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i \right)^2 , \tag{102}$$

where Y_i are the values predicted by a function for a sample of n data points and the $\hat{Y}_i$ are the observed data. We don't have access to an analytic expression we could use as a function for the Y_i. For this reason, when we calculate the MSE in this work, we use the data from the most precise run (in this case $V_s = 150$) as a reference. In Tab. 1, we calculated the MSE from the equal time $F(x^0, x^0, 0)$ for different spacial volumes using the biggest volume $V_s = 150$ as the reference function. For all measurements, the observation time ranged from 0 to 2000.

As we already saw in Fig. 17 and Fig. B.2, we get noticeable, or, in case of the tsunami, even big deviations for $V_s = 10$. However, the deviation decreases very fast and is pretty much already negligibly small for volumes bigger than $V_s = 30$. From Tab. 1 we can see that for the quench (tsunami), the deviation calculated via the MSE is already three (seven) orders of magnitude smaller when we chose a volume of $V_s = 40$, than it is for $V_s = 10$. We can conclude that in the case of 2-loop corrections, finite lattice effects get drastically reduced by increasing the systems volume and become unnoticeable for

Volume	$MSE[10^{-3}]$	
	Quench	Tsunami
10	0.49	36.3
20	0.43	0.026
30	$1.3 \cdot 10^{-4}$	$1.5 \cdot 10^{-6}$
40	$4.0 \cdot 10^{-6}$	$1.1 \cdot 10^{-7}$
50	$1.9 \cdot 10^{-6}$	$3.0 \cdot 10^{-8}$
60	$3.0 \cdot 10^{-7}$	$3.0 \cdot 10^{-9}$
150	0	0

Table 1: Listed are the deviations from the course of the equal time correlation function $F(x^0, x^0, 0)$ for different lattice sizes V_s against a reference course with $V_s = 150$.

$V_s \geq 40$. To be safe for our computation, but still avoid long run times, we chose a volume of $V_s = 60$ or greater for our measurements throughout this work.

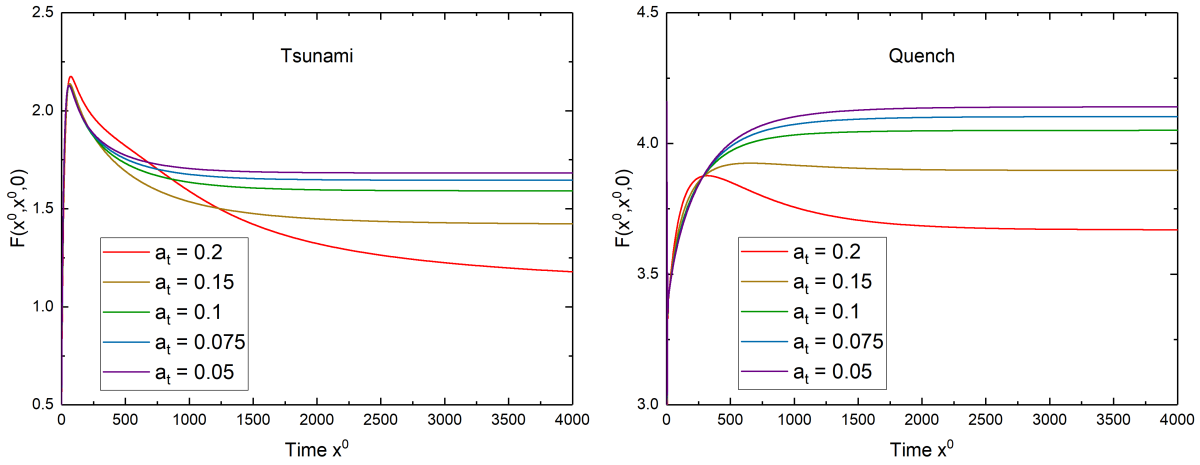
6.2. Time and spacial spacing

Besides the dependency on the lattice size, the other two important factors for numerical effects and stability are the time and the spacial spacing a_t and a_s of the lattice. First, we want to take a closer look at the time spacing.

6.2.1. Time spacing dependency

We observed the system up until $x^0 = 4000$ with a fixed lattice spacing of $a_s = 0.4$ for different time spacings a_t. For these settings, as we saw in sec. 5, we can assume that the system should be very close to a thermalized state at the end of the time evolution. Our results are shown in Fig. 18 for both tsunami and quench. To produce this data, we used the settings $\mathcal{S}_4$ (appendix A).

From Fig. 18, we can see that, for both the tsunami and the quench, the course of $F(x^0, x^0, 0)$ significantly depends on the time spacing. For the rough spacing of $a_t = 0.2$, can see significant deviations, however, the system still thermalizes. When we decrease the time spacing, the deviation between the graphs seem to decrease stronger than linear. An additional observation that can be made is that smaller spacings tend to cause a higher value of the zero momentum mode when the system approaches an equilibrium state. To get a better grasp on how big the deviation truly is, we again calculate the MSE. The mean square error that we calculated is shown in Tab. 2, where we used the $a_t = 0.05$ course as a reference.



(a) (b)

Figure 18: The graphs shows the course of the equal time correlation function $F(x^0, x^0, 0)$ for different time spacings in the case of a) tsunami and b) quench initial conditions. A rough time spacing of $a_t = 0.2$ leads to significant deviations in the course of the time evolution, but does not hinder thermalization. Decreasing a_t causes the different courses to approach each other faster than linear.

a_t	$MSE[10^{-3}]$	
	Quench	Tsunami
0.05	0	0
0.075	1.04	1.06
0.1	6.03	6.21
0.15	43.9	45.2
0.2	161	121

Table 2: Mean square error MSE of $F(x^0, x^0, 0)$ for different time spacings a_t and for both quench and tsunami initial conditions. The course of $F(x^0, x^0, 0)$ for $a_t = 0.05$ was chosen to be the reference function to calculate the MSE.

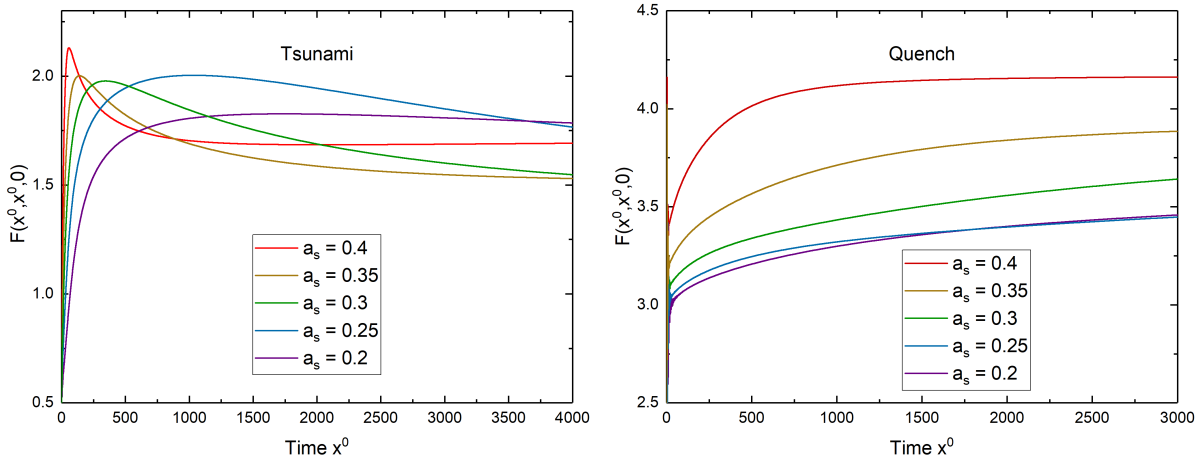
From Tab. 2, we can see that by decreasing the time spacing, the mean square error between the different courses decreases in fact quadratically. This is to be expected from our solving method. In sec. 3, we presented the discretization method (67) we chose for the second partial derivative in time. For this finite difference method, the error is expected to scale with a_t^2 (see [20] p. 183-184). Thus, we would also expect the MSE to scale quadratically with the changing time spacing.

Note that, like previously mentioned in sec. 5, these long time runs are only possible using the reduction technique, which we'll discuss later on. With or without this reduction, decreasing the time spacing comes at a price of a significant increase in computation time to reach the same final time x^0_{final}. It is therefore important to choose a_t in such a way

that the results are still meaningful, while the computation time stays in an acceptable range. The spacing $a_t = 0.1$ leads to results close to $a_t = 0.075$ and $a_t = 0.05$, while still allowing a fast computation up to late times, which is why we most of the time chose $a_t = 0.1$ in this work.

6.2.2. Spacial spacing dependency

We want to conclude the observation of numerical effects with the analysis of the system's dependency on the spacial spacing. Again, we observe the equal time correlation function $F(x^0, x^0, 0)$ both quench and tsunami with a fixed time spacing $a_t = 0.05$ and vary the spacial spacing a_s. The result of our investigation is presented in Fig. 19. We used set $\mathcal{S}_5$ (appendix A) to compute the data.



(a) (b)

Figure 19: Depicted are the courses of $F(x^0, x^0, 0)$ for different spacial spacing a_s with a fixed time spacing $a_t = 0.05$. For tsunami initial conditions a), one can see that decreasing the lattice spacing damps down the overshoot of the zeroth momentum mode. Additionally, the time until thermal equilibrium is reached seems to increase significantly. In the case of quench initial conditions b), decreasing the lattice spacing on one hand also decreases the occupation of the zero momentum mode and on the other hand increases the thermalization time.

As it can be seen from Fig. 19, the system is even more sensitive to the change of a_s and the from it resulting change in the momentum cutoff. However, both quench and tsunami seem to at least show a behavior, which indicates the a approach to thermal equilibrium for all observed a_s. Opposite to the case of the time spacing, for the spacial spacing, a smaller a_s decreases the occupation of the zeroth momentum mode. This can

be seen nicely for the quench (Fig. 19 b)), for the tsunami (Fig. 19 a)) this behavior can only be partially seen in the figure. Nonetheless, for the tsunami, the decrease of the zeroth mode can be seen at the damping of the overshoot during the drifting phase when decreasing a_s and at the decreasing tendency of $F(x^0, x^0, 0)$ for $a_s < 0.4$. The courses for spacing $a_s = 0.2$ indicates a similar behavior, however, in the limited observation time, this behavior is only indicated through a slight decline. Besides this decrease of $F(x^0, x^0, 0)$, smaller spacial spacing also significantly increases the thermalization time. While the zeroth mode for $a_s = 0.4$ seems to have reached thermal equilibrium after roughly $x^0 = 1500$, only the $a_s = 0.35$ course seems to be close to thermalization in the observed time frame up to $x^0_{max} = 4000$. For all the other a_s, the thermalization time seems to be significantly longer. In case of the quench, both the tendency of decrease of the zeroth mode for smaller a_s as well as the increased thermalization time can be seen directly from Fig. 19 b).

Similar to the time spacing, we also calculated the MSE between the different courses of $F(x^0, x^0, 0)$, taking $a_s = 0.2$ as our reference course. The calculated MSE is shown in Tab. 3.

a_s	$MSE[10^{-3}]$	
	Quench	Tsunami
0.2	0	0
0.25	0.41	4.7
0.3	2.2	6.5
0.35	16.6	9.5
0.4	57.5	11.4

Table 3: Listed is the MSE from the course of the equal time correlation function $F(x^0, x^0, 0)$ for both quench and tsunami initial conditions and different lattice spacings a_s. The course of $a_s = 0.2$ was used as a reference to calculate the MSE.

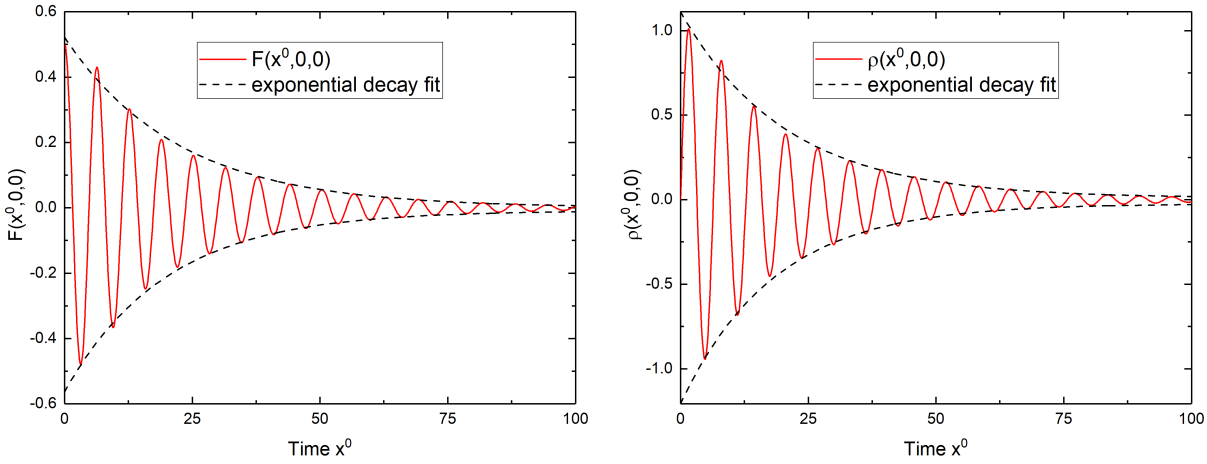
From Tab. 3, we can see that the deviations for the quench and the tsunami differ quite significantly. For the quench, the MSE again indicates a quadratical decrease with decreasing spacial spacing a_s. In the differential operator (64) of the KBE, the spacial derivative is, similar to the time derivative, a second order one. For this reason, we might also expect a quadratic dependency of the MSE on the spacial spacing, which is shown by the quench.

In case of the tsunami, this quadratic dependency of the MSE is not observed. There are two reasons for this. The first is the way in which the zeroth mode of the tsunami behaves under changing a_s. Different from the quench, the tsunami shows a significant crossing

between the courses of $F(x^0, x^0, 0)$ for the different a_s. This crossing strongly influence the result gained by the MSE. The second reason is the limited time frame we observed. If the system thermalizes (i.e. becomes constant in time) and we observe it for a long enough time, the change of the MSE due to the crossing of the graphs should become negligible. In this case, we expect would the MSE to show a quadratic dependency on a_s for the tsunami, too.

7. Efficient solutions for memory integrals

In sec. 3, we presented the basics of the numerical implementation for solving the KBE. However, as we saw in sec. 4, the computational workload and therefore the total time needed to perform the computation as well as the necessary storage memory, increases with $\mathcal{O}(N_t^{3.133})$ and $\mathcal{O}(n_t^{2.081})$, respectively, for the time evolution. This rapid increase makes it impossible to evolve the system to late times, where thermalization occurs. To keep the computational effort within an accessible range, a reduction scheme is necessary. Fortunately, there is a possibility for such a reduction, namely a history cut-off. We'll get to the history cut-off in a bit, but to understand the reason behind it, we first need to take a closer look at the unequal time correlations of F and ρ. We want to note, that we used the settings from set $\mathcal{S}_6$ (appendix A) throughout this chapter to enable easier comparisons between the plots shown in the following.



(a) (b)

Figure 20: Depicted are the unequal time correlations (red curve) a) $F(x^0, 0, 0)$ and b) $\rho(x^0, 0, 0)$ for tsunami initial conditions. One can clearly see the decay of the amplitude for both functions. To show that this decay is exponential, we fitted the amplitude of the oscillation from above and below with an exponential decaying function, indicated by the dashed black curves.

Fig. 20 shows the decay over time of the initial time correlations $F(x^0, 0, 0)$ and $\rho(x^0, 0, 0)$ for a tsunami with a relatively low coupling of $\lambda = 3$. One can see, that both F and ρ decay rapidly in a similar fashion. We performed a fit of the amplitudes' peak values over time, represented by the dashed, black curve that envelops the oscillation. The fit

function is a decaying exponential function $y(x) = y_0 + A\exp(-x/t)$ with

$$
\begin{aligned}
&F \text{ above: } y_0 = 0.001 \pm 0.007 \quad A = 0.521 \pm 0.011 \quad t = 22.3 \pm 1.2 \\
&F \text{ below: } y_0 = -0.009 \pm 0.003 \quad A = -0.5545 \pm 0.0075 \quad t = 19.79 \pm 0.56 \\
&\rho \text{ above: } y_0 = 0.011 \pm 0.011 \quad A = 1.101 \pm 0.021 \quad t = 20.62 \pm 0.93 \\
&\rho \text{ below: } y_0 = -0.023 \pm 0.006 \quad A = -1.184 \pm 0.019 \quad t = 19.37 \pm 0.55
\end{aligned}
$$

The decay rate of F and ρ are in agreement with the error range. This exponential decay is in fact quite general to the evolution of the KBE and appears for all correlation functions $F(x^0, y^0, p)$ and $\rho(x^0, y^0, p)$ the further they are away from the time diagonal $t_d = |x^0 - y^0| = 0$. It should be noted that the decline of the unequal time correlations depends of the system's coupling constant λ. We'll come across this dependency later on in sec. 7.4.

The loss of the past correlations is the very reason why the observation of the late time evolution of the KBE is a feasible endeavor. From a physical point of view, one might think of this behavior in the following way. The equal time correlation functions describe the measurable current state of the system through macroscopic observables, while the off-diagonal correlations $y^0 \neq x^0$ at time x^0 describe the correlation between the current state to some previous configuration. The exponential damping of these correlations can then be thought of as a memory loss of the system. The system "forgets" about its previous states as time passes. Naturally, we build the reduction scheme in such a way, that it makes use of this memory loss.

7.1. The history cut-off

The loss of the system's past enables us to chose a specific point, at which we can simply cut off all previous correlations (the history) of the system, without changing its future course too significantly (see [3]). Because we cut off the system's history in the process, this reduction scheme is termed a history cut-off. The important question, at which point one should perform this cut-off so that one gets a good ratio between the reduction of computational workload and deviation from the unreduced history is not a trivial matter and will be discussed later in this work. First, we want to show how to implement such a history cut-off into a solver. For this implementation, we made use of [3].

To implement the history cut-off, we introduce a fixed history matrix (or fixed history) $H(x^0, y^0, p)$, which is a cutout of the system's lattice. The area of the fixed history is $A_H = (a_t \cdot N_H)^2 = t_H^2$, where t_H is the size or length of the history matrix in time and

N_H (history steps) are the number of data points that make up the history length. This fixed history is then evolved along the time diagonal, like it is shown in Fig. 21.

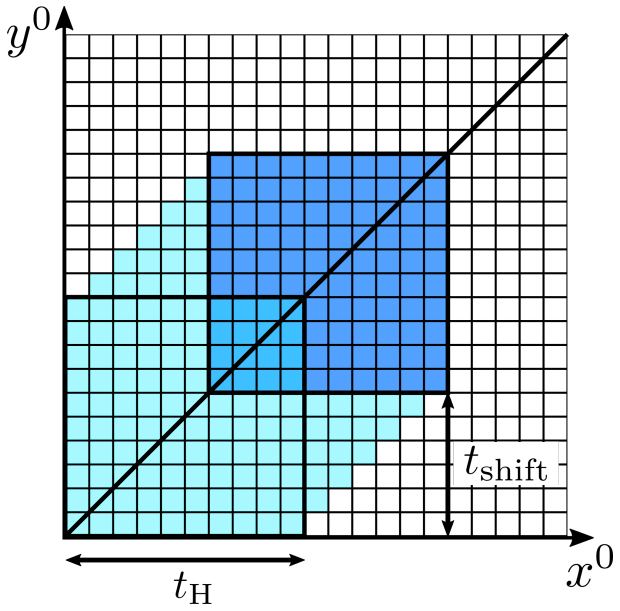


Figure 21: Depicted is the evolution of a fixed history matrix (thick bordered square) with size t_H along the time diagonal. Each square of the grid represents the correlation functions for different time values, that could be calculated during the evolution (Note that each square contains an additional N_{sp} number of spacial points). White squares represent correlation functions that have not been, or will not be calculated during the time evolution. The light blue squares are values that have been previously calculated and have already been discarded. The dark blue ones represent the data, that is stored for the current state of the system. When evolving the fixed history along the time diagonal it gets shifted by t_{shift} in the x^0 and y^0 direction.

After the history matrix is completely filled once, it is shifted along the time diagonal, thereby discarding all values with $y^0 < (x^0 - a_t N_H) = a_t N_{\text{shift}} = t_{\text{shift}}$. It is important to account for this shifting, by keeping the lower bound Mem_{LB} of all memory integrals in the Kadanoff-Baym equations within a range $t_{\text{shift}} \leq \text{Mem}_{LB} \leq x^0$. All values for $F(x^0, y^0)$ and $\rho(x^0, y^0)$ where $y^0 < t_{\text{shift}}$ holds obviously do not need to be calculated anymore and can be neglected.

With the history cut-off in place, the discrete equations of motion for the statistical component (103), for $x^0 > t_H$, take the form

$$\begin{aligned} F(n+1,m,\boldsymbol{p}) &= 2F(n,m,\boldsymbol{p}) - F(n-1,m,\boldsymbol{p}) \\ &\quad - a_t^2 \left(\boldsymbol{p}^2 + M(n)^2\right) F(n,m,\boldsymbol{p}) \\ &\quad - a_t^3 \Bigg\{ \frac{\Sigma_\rho(n,N_{\text{shift}},\boldsymbol{p})F(N_{\text{shift}},m,\boldsymbol{p})}{2} - \frac{\Sigma_F(n,N_{\text{shift}},\boldsymbol{p})\rho(N_{\text{shift}},m,\boldsymbol{p})}{2} \\ &\quad + \sum_{l=N_{\text{shift}}+1}^{m-1} \left(\Sigma_\rho(n,l,\boldsymbol{p})F(l,m,\boldsymbol{p}) - \Sigma_F(n,l,\boldsymbol{p})\rho(l,m,\boldsymbol{p})\right) \\ &\quad + \sum_{l=m}^{n-1} \left(\Sigma_\rho(n,l,\boldsymbol{p})F(l,m,\boldsymbol{p})\right) \Bigg\}, \end{aligned} \tag{103}$$

with m ranging from N_{shift} to n. For the calculation of the spectral function $\rho(n,m,\boldsymbol{p})$ nothing changes. Its memory integrals already only range from y^0 to x^0, so no additional shift is necessary in this case (see (43)). However, values of ρ with $m < N_{\text{shift}}$ do not need to be calculated, due to the shift of the history matrix.

Reducing the history of the system in such a way is accompanied by a tremendous saving in computational effort and memory. In case of a fixed history, the necessary storage memory can still be calculated using (70). However, the total number of time steps N_t for the whole lattice is now replaced by the number of history steps N_H. For a given history, the storage requirements are therefore fixed and no longer dependent on the total observation time. This is the first contribution of the history cut-off to the possibility of observing the late time behavior of systems, which might require the calculation of several thousand steps. The second and in most cases also most important effect of introducing a fixed history size is the tremendous reduction of computational effort. For a fixed history size the number of points that need to be calculated for each time step remains constant (see Fig. 21 and 22). Furthermore, because of the required shifting of Mem_{LB}, the range of the memory integrals also remains constant. After the history is filled, this leads to a constant amount of time necessary to perform each additional time step, enabling us in principle to calculate an arbitrary number of steps. However, reducing the system's history comes at the cost of altering the result of the numerical computation. Depending on the fixed history size, this change might be minuscule, quite significant or even devastating to the time evolution. We'll discuss this dependency in a bit, but for now, we'll show how to implement the fixed history in an efficient way.

7.2. Cyclic memory

A reasonable storage architecture to implement a fixed history into a program is the use of a double circular buffer. A circular buffer is a type of memory architecture, that operates after the first in, first out principle. This means that the newest data that is stored overwrites the oldest one in the storage (see [21]). In our case, the storage is not really a buffer, which is why we'll use the term circular memory or cyclic memory in the following.

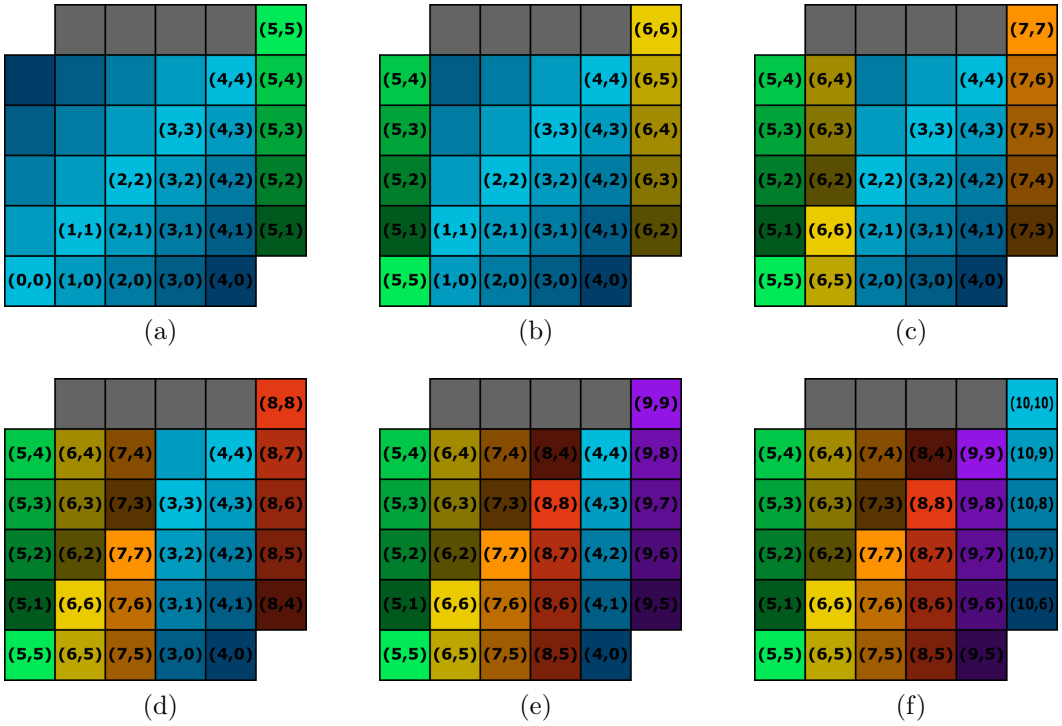


Figure 22: (a)-(f) shows a schematic depiction for the storage process of the cyclic memory in cause of the statistical component F starting from the time $x^0 = t_H$, where the fixed history matrix has been filled once. For the spectral component, the general process is equivalent. The additional column and row, that is added in each graph symbolizes the new F-values for the n+1-th step, that are calculated and stored in the history storage. The grey colored part is already determined by the commutation relations and is therefore neither calculated nor stored. It is kept in this figure to visualize the analogy to Fig. 21.

We'll illustrate the filling process for a fixed history with a size of $N_H = 5$ steps for the statistical function F. Note that the way the history is filled is in principle equivalent for both F and ρ. However, because of its anti-commutation relation the calculation of $\rho(n, n)$ and $\rho(n, n-1)$ is not necessary and can be omitted.

Our starting point is the time $x^0 = t_H = 5 \cdot a_t$, shown in Fig. 22 a), where the history has just been filled once for the first time. The blue boxes represent the values of F that

are already stored in the history matrix. The additional, shifted, different colored (here green) boxes indicate the N_H new values of F for the (N_H+1)-th time steps. The gray boxes illustrate the values of $F(n, m)$ with $n < m$, which do not need to be calculated due to the commutation relations. Lastly, the brightness of the boxes indicates their distance from the time diagonal.

The green values get permuted once in x^0 and y^0 direction. The x^0 permutation puts them in the first column, while the y^0 permutation places the equal time value $F(5, 5)$ on the diagonal, and thus, preserving the overall structure of the storage. This process is repeated with an increasing number of permutations for further time steps, indicated by the different colored values (Fig. 22 b)-f)). After repeating this cycling N_H times (Fig. 22 f)), at $x^0 = 10 \cdot a_t = 2 \cdot t_H$, the system is in a similar configuration as it was initially at time t_H. The values above the diagonal became irrelevant, due to their second time component being lower than N_{shift} and one might think of them as the empty blue squares from Fig. 22 a). Only the values on the diagonal and below contribute to the calculations, as it has been the case for our starting point. The process then repeats itself again for $x^0 = 3 \cdot t_H$ and so forth until the desired final time is reached.

In a matrix like representation, this cycle process can be implemented by performing two modulo operations, one in each time direction, i.e. $H\left(a_t(n \ mod \ N_H), a_t(m \ mod \ N_H)\right)$ [3]. In our work, we chose to store our data inside a line vector. We use the column-major ordering to map data from the matrix representation onto this line vector. A convenient way to perform the cycling and manage the transition from matrix to line vector is the use of a mapping function

$$f_H : \mathbb{N}^2 \to \mathbb{N},$$

which cycles the time indices via $(n \ mod \ N_H, m \ mod \ N_H)$ in the matrix representation and then maps the result onto the line vector. A more detailed description, as well as the specific implementation of this function is shown in appendix C.1.

Before we change to the discussion of our numerical results for the history cut-off, we want to give the reader a way to visually imagine the double cyclic memory. First, consider a single cyclic memory. One can think of this as nothing more than a line with periodic boundary conditions. These boundary conditions in turn can be interpreted as the action of "gluing" together both ends of this line, resulting in a circle.

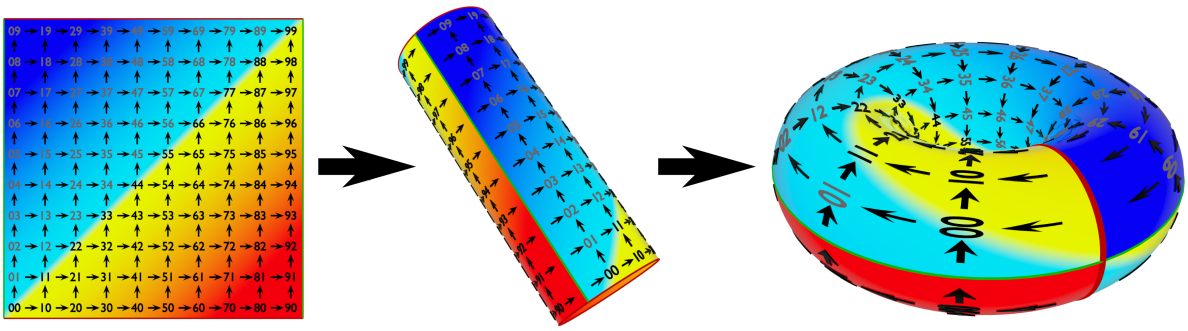


Figure 23: Geometrical visualization of the cycled memory for a square storage. The periodic boundary conditions of the cycling function are depicted by the colored lines on the top and bottom (red), as well as on the left and right (green) of the plane. Enforcing one boundary conditions effectively wraps the plane a) into a cylinder b). When enforcing the second boundary condition, the top and the bottom of the cylinder have to be connected, resulting in a torus shape c).

In case of the fixed history matrix, the line would be replaced by a plane like it is shown in Fig. 23 a). When we impose periodic boundary conditions in x^0 direction, we have to "glue" the left and the right side of the plane together which results in a cylinder (Fig. 23 b)). The second boundary condition, the one in y^0 direction, can then be achieved by "gluing" the top and the bottom of the cylinder together. What we end up with is a torus shape (Fig. 23 c)) as an illustration of our double cyclic memory.

7.3. Effects of the history cut-off

In the following, we want to analyze the deviation that the history cut-off causes, as well as the effects it has on the system's time evolution. We observed both tsunami and quench with $a_s = 0.4$, $a_t = 0.1$ and coupling constants $\lambda_{\text{Tsunami}} = 7$ and $\lambda_{\text{Quench}} = 2.5$ for various history sizes ranging from $t_H = 1$ to $t_H = 50$. For our analysis, we again use the zeroth momentum mode $F(x^0, x^0, 0)$. The tsunami case is shown in Fig. 24, where we present the course of $F(x^0, x^0, 0)$ for a sample of the history sizes we tested. The corresponding plot for the quench is given in Fig. B.3 and had to be placed into appendix B.3.

One can see that the course of $F(x^0, x^0, 0)$ for small history sizes $t_H = 8$ and $t_H = 10$ deviates quite strongly from the course of a large history size $t_H = 50$. However, as the history size increases, the graphs approach each other fast and the deviation becomes small quickly. This behavior can be observed for both tsunami and quench and is to be expected considering the exponential decaying early time correlations illustrated in Fig. 20. From Fig. 24 and B.3 one can also see that the deviation decreases more rapidly for tsunami than it does for the quench. This observation can be explained by the differently fast

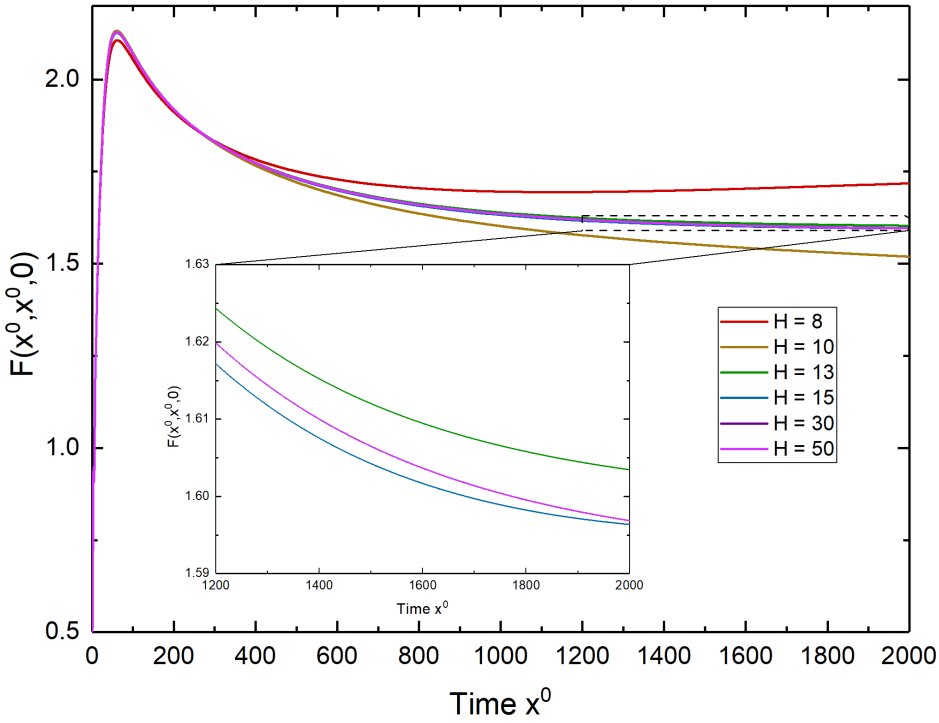


Figure 24: The graph shows the course of the equal time correlation function $F(x^0, x^0, 0)$ with tsunami initial conditions for different history sizes t_H ranging from 8 to 50. While small histories of $t_H \leq 10$ show quite significant deviations, larger histories are barely distinguishable. For a better visualization, a zoomed version of the course for larger histories from $x^0 = 1200$ to 2000 is shown in the insert. For $t_H = 13$ and 15 one can still see a small but with the magnification observable difference between the courses. However, between $t_H = 30$ and $t_H = 50$, this is no longer the case.

decline of the initial correlations between quench and tsunami for the chosen parameters. In Fig. 25, we show the decline initial time correlation $F(x^0, 0, 0)$ for both quench and tsunami up to $x^0 = 50$.

One can see that the correlations for the tsunami (red) decrease faster than they do for the quench (blue). This faster decline of the correlations makes them negligible earlier, which results in a smaller deviation for the same history size in case of the tsunami than it does compared to the quench.

We again calculated the MSE between the measured courses of $F(x^0, x^0, 0)$ for different history sizes, with $t_H = 50$ as our reference course. The results are presented in Tab. 4. There are a couple of observations that can be made from Tab. 4. First, we didn't present deviations for $t_H < 6$, that is because for such small histories, the system becomes unstable and shows a divergent behavior. Second, the system does not show a monotonous decrease in deviation for history sizes from small history sizes $t_H \leq 13$. This is a result

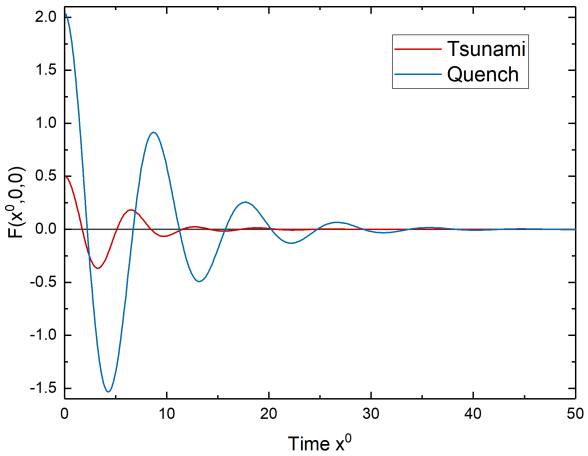


Figure 25: Initial time correlation $F(x^0, 0, 0)$ of a system with a coupling $\lambda = 7$ for tsunami (red) and $\lambda = 2.5$ for quench (blue) initial conditions.

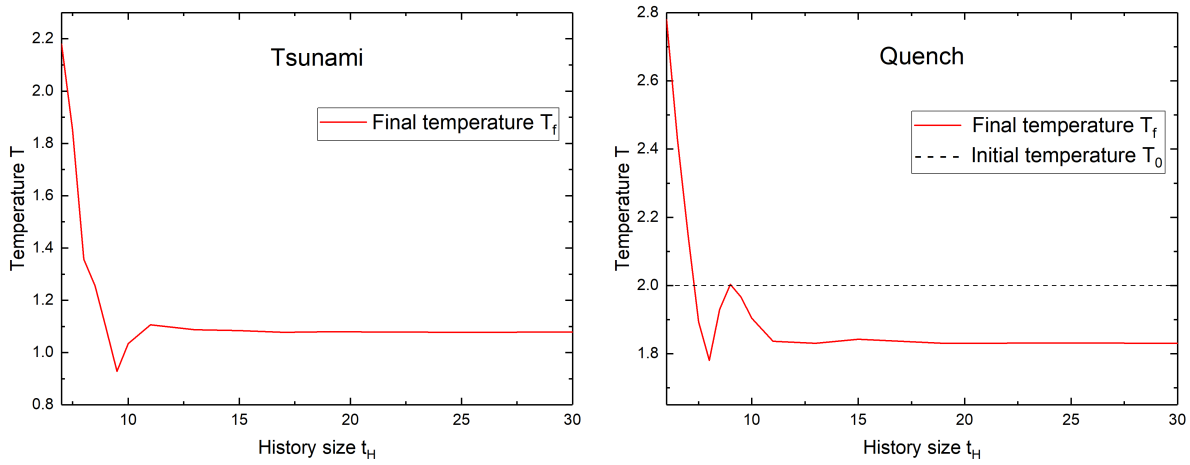
t_H	$MSE[10^{-3}]$		t_H	$MSE[10^{-3}]$	
	Quench	Tsunami		Quench	Tsunami
50	0	0	11	0.74	0.042
30	$1.7 \cdot 10^{-6}$	$2.3 \cdot 10^{-7}$	10	2.5	1.8
25	$4.1 \cdot 10^{-4}$	$5.3 \cdot 10^{-6}$	9	7.8	0.062
19	$5.4 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	8	51.4	4.9
15	0.20	$3.7 \cdot 10^{-3}$	7	176	17.5
13	0.82	0.016	6	900	1.7

Table 4: Listed are the deviations from the course of the equal time correlation function $F(x^0, x^0, 0)$ for both quench and tsunami initial conditions and different history sizes t_H. The course for the large history $H = 50$ was used as a reference to calculate the MSE.

from the high sensitivity that the system shows in its course when considering changes in small history sizes. The high sensitivity can be attributed to the significant portions of the unequal time correlation functions for small histories, that are either within or outside the history, depending on its size. For larger history sizes $t_H \geq 15$, the unequal time correlations have already declined far enough, so that small changes in t_H do not influence the system's time evolution notably.

Because of its strong activity, the correlation function $F(x^0, x^0, 0)$ for the zeroth mode is a good quantity to observe for the evolution of the system. However, we don't want to completely neglect the other momentum modes. For this reason, we want to also look at quantities which contain the information about all momentum modes. We can get these quantities from the the particle density $n(x^0, p)$ and the linear entropy $S_L(x^0, p)$. By fitting a Bose-Einstein distribution to the graph of the particle density $n(\epsilon)$ as a function of the energy density $\epsilon(x^0, p)$ (see sec. 5.2.2), and therefore including all momentum modes, we

can calculate the final temperature T_f of the system at the end of the time evolution $x^0 = 2000$ for the different histories we chose. The results for both tsunami and quench are shown in Fig. 26.



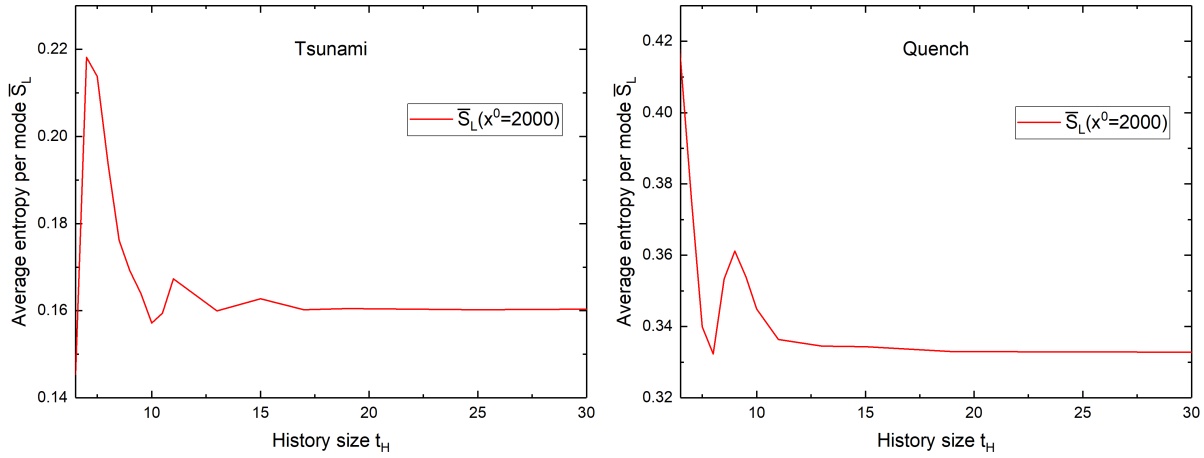
(a) (b)

Figure 26: Depicted is the final temperature T_f of the system after a time $x^0 = 2000$ for both a) tsunami and b) quench initial conditions and different history sizes. The temperature was computed by fitting a Bose-Einstein distribution to the particle density $n(\epsilon)$.

From Fig. 26 we can see, that for both tsunami and quench, the temperature changes quite drastically for history sizes $t_H < 12.5$. In case of the quench, the temperature for small history sizes doesn't even seem to make sense, due to the fact that the late time temperature is bigger than the initial temperature T_0. For the quench, where we drop the system's initial mass by half at the beginning of the time evolution and therefore take energy out of the system, we would rather expect the temperature to decrease (as it is the case for larger history sizes). These history sizes are just too small for the system to yield reasonable results. However, we can see that for increasing history sizes, the temperature rapidly converges against a constant value, indicating that the deviation of the time evolution of the system is only minuscule.

Besides the temperature, we can also observe the average linear entropy per mode $\bar{S}_L(x^0)$, given in Fig. 27 for $x^0 = 2000$, which due to being an average value, takes also all of the system's momentum modes into account.

The behavior of $\bar{S}_L$ is similar to the one we observed for the temperature. Here, histories $t_H < 15$ result in strong fluctuations of the final average linear entropy. However, like it was the case for the temperature, when we increase the system's history size far enough,



(a) (b)

Figure 27: Depicted is the average linear entropy per mode $\bar{S}_L(x^0)$ of the system after a time $x^0 = 2000$ for both a) tsunami and b) quench initial conditions and different history sizes.

$\bar{S}_L$ approaches a constant value. This again indicates, that the deviation to the time evolution is minuscule for larger history sizes.

All of our observations lead to the conclusion, that a consideration of the system's full history for the KBE is not necessary to achieve results which only show small deviations. But there remains the question on how big we have to choose our history size for a given system. In the next section, we'll try to give an answer to this question.

7.4. Finding a suitable history size

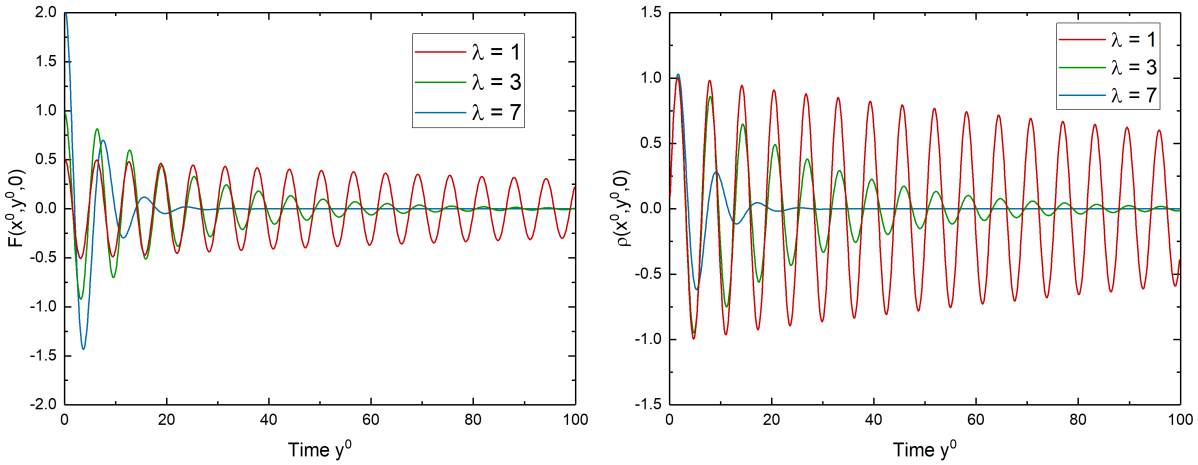
As we saw in the previous sections, reducing the system's history to a fixed size leads to a tremendous decrease in computational effort and memory usage, making it possible to even observe KBE late time evolution. However, we also saw, that choosing a history size for the system that is too small can lead to big deviations, while calculations with big history sizes take a long time and use a lot of memory. Because of this, it is important to find a way to determine a suitable history size, which is big enough to avoid huge deviations and still small enough to avoid unnecessary long computation times. We now want to present our method on how to determine such a history size.

7.4.1. Calculating a fitting measure

The first step to determine a good history size is to find a fitting measure on how much correlation is lost when introducing the history cut-off. We'll achieve this by making use of the decay of unequal time correlation functions.

From Fig. 20 and Fig. 25, we saw the exponential decline of the initial correlations $F(x^0, 0, 0)$ and $\rho(x^0, 0, 0)$ as time passes. We stated, that this decline is quite general and can be observed for all correlation functions $F(x^0, y^0, p)$ and $\rho(x^0, y^0, p)$ for an increasing distance $t_d = |x^0 - y^0|$ away from the time diagonal $t_d = 0$ and depends on the coupling constant λ of the system. Because we'll make use of $F(x^0, y^0, p)$ and $\rho(x^0, y^0, p)$ in the following, we want to show their decline again for different couplings (see Fig. 28).

In Fig. 28 we present the courses of $F(x^0, y^0, 0)$ and $\rho(x^0, y^0, 0)$ as a function of y^0 for a fixed value of $x^0 = 100$. We used the tsunami initial conditions and three different couplings $\lambda = \{1, 3, 7\}$. In the figure, we can see the discussed exponential decline, which is most prominent for the high coupling $\lambda = 7$ and only small for $\lambda = 1$. Using these correlation functions in our approach of finding the most suitable history size for the observed system is similar to the short memory approximation from [13], where the Green's function is used to determine the cut-off point.



(a) (b)

Figure 28: Depicted are the unequal time correlation functions a) $F(x^0, y^0, 0)$ and b) $\rho(x^0, y^0, 0)$ of a system with tsunami initial conditions for different coupling constants λ. The time component x^0 is fixed to $x^0 = 100$. Similar to the initial time correlations, both $F(x^0, y^0, 0)$ and $\rho(x^0, y^0, 0)$ decay exponentially for increasing y^0. For the small coupling $\lambda = 1$ this decay is only small and the amplitude of the unequal time correlations changes only slightly within the time interval. For the high coupling $\lambda = 7$ the decay is fast and the amplitudes of F and ρ effectively vanishes for $y^0 > 35$.

We start our search for suitable history size by defining the measure

$$\Lambda_F(x^0, \tau, p) = \int_\tau^{x^0} dz^0 \left| F(x^0, z^0, p) \right| \tag{104}$$

$$\Lambda_\rho(x^0, \tau, p) = \int_\tau^{x^0} dz^0 \left| \rho(x^0, z^0, p) \right| . \tag{105}$$

The variable τ, will be our cut-off parameter, which we vary to determine our new history size. One might wonder why we consider the course of $F(x^0, y^0, p)$ and $\rho(x^0, y^0, p)$ along the y^0- like it was shown in Fig. 28 and not, like we showed in Fig. 20, along the x^0-direction. As we saw from these figures, they all showed the exponential decay of unequal time correlation functions the further we get away from the time diagonal. Therefore as a matter of fact, we could also use the decline of unequal time correlations along the x^0-direction. We simply chose the y^0 dependency because it kept our equations for finding a suitable history size in a nicer form.

In the next chapter, we'll present the general procedure on how we determine a fixed history size using the measures (104) and (105).

7.4.2. Determine history size algorithm

Before we present the determine history size (DHS) algorithm (see also appendix C.2), we first want to explain how to use it to avoid confusion. In general, when using the program to do late time calculations, one has specify a maximal history size $t_{H_{max}}$ for the system. In case of a completely unknown model, this maximal history should be as big as possible, preferably as big as the system, where the program is running on, allows it to be. If the model is known to some degree, one might do an educational guess and choose a smaller but still sufficiently big history size. Now, the program solves the KBE until $x^0 = t_{H_{max}}$. At that point, if enabled, the DHS algorithm starts to calculate a new history size t'_H. After the calculation is finished, the maximal history $t_{H_{max}}$ is resized to t'_H, using the *resizeHistory*-routine (appendix C.3). The time evolution of the system then continues with the reduced history t'_H until the final time x^0_f is reached (see also Fig. 29).

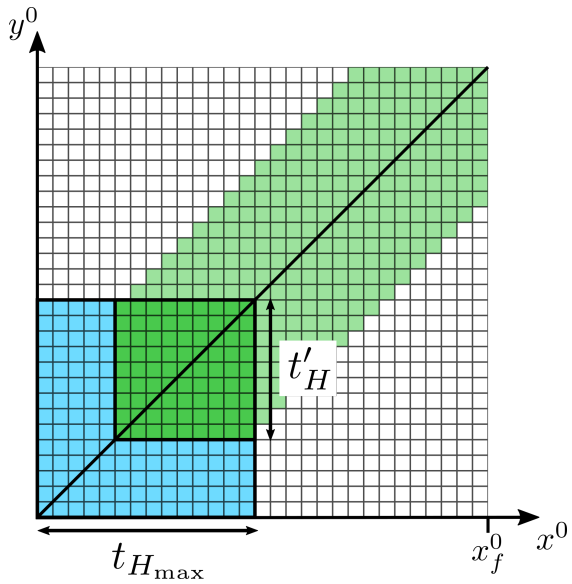


Figure 29: Illustration of the resizing of an initial maximum history $t_{H_{max}}$ (blue + green square) to a new history t'_H (green square) determined by the DHS algorithm and its subsequent time evolution. Colored squares represent values of the correlation functions that are, or will be (light green squares) calculated during the evolution of the system. White squares are correlation function values that no longer appear in the computations.

Note, that in principle we could also determine a new history at a later time $x^0 > t_{H_{\max}}$ and perform smaller history changes more than once during the time evolution. During our studies, we experimented with these approaches, but further future investigation is needed to get concrete results. For now, let's take a closer look at the DHS method.

To determine a suitable history size, we proceed as follows. At some time step x_c^0 (following the example from above, we $x_c^0 = t_{H_{max}}$), we first calculate the maximum value of Λ_F and Λ_ρ, i.e.

$$\Lambda_F^{\max}(x_c^0, t_{\text{shift}}, p) = \int_{t_{\text{shift}}}^{x_c^0} dz^0 \left| F(x_c^0, z^0, p) \right| \tag{106}$$

$$\Lambda_\rho^{\max}(x_c^0, t_{\text{shift}}, p) = \int_{t_{\text{shift}}}^{x_c^0} dz^0 \left| \rho(x_c^0, z^0, p) \right|. \tag{107}$$

The lower bound of the integral is given by the shifting parameter t_{shift}, that has been previously shown in Fig. 21. When the DHS algorithm is run at $x_c^0 \leq t_{H_{max}}$, the shifting parameter would be zero. However, one could also run the algorithm at some later time $x_c^0 > t_{H_{max}}$. In this case, as we saw earlier in this chapter, the shifting parameter would be given by $t_{\text{shift}} = x_c^0 - t_H$.

After we are done calculating the maximum values of Λ_F and Λ_ρ, we compute (104) and (105) using the relations

$$\Lambda_F(x_c^0, \tau, p) = \Lambda_F^{\max}(x_c^0, t_{\text{shift}}, p) - \int_{t_{\text{shift}}}^{\tau} dz^0 \left| F(x_c^0, z^0, p) \right| \tag{108}$$

$$\Lambda_\rho(x_c^0, \tau, p) = \Lambda_\rho^{\max}(x_c^0, t_{\text{shift}}, p) - \int_{t_{\text{shift}}}^{\tau} dz^0 \left| \rho(x_c^0, z^0, p) \right|. \tag{109}$$

We now stepwise increase the parameter τ from t_{shift} to x_c^0 until either

$$\frac{\Lambda_F(x_c^0, \tau, p)}{\Lambda_F^{\max}(x_c^0, t_{\text{shift}}, p)} \leq \eta, \qquad \text{or} \qquad \frac{\Lambda_\rho(x_c^0, \tau, p)}{\Lambda_\rho^{\max}(x_c^0, t_{\text{shift}}, p)} \leq \eta. \tag{110}$$

So in principle, we compare the total absolute area under the graph of $F(x_c^0, z^0, p)$ and $\rho(x_c^0, z^0, p)$, with z^0 ranging from t_{shift} to x_c^0, with an ever increasingly truncated area, where z^0 only ranges from τ to x_c^0. The parameter η, which has to be specified by the user, sets a lower bound for the relative remaining area under the curve of the unequal time correlation functions $F(x_c^0, z^0, p)$ and $\rho(x_c^0, z^0, p)$ after the history of the system is changed. Because of this we'll refer to η as the relative remaining correlations. We define the value of τ at which one of the conditions (110) is fulfilled as $\tau_{cut}(p)$. This $\tau_{cut}(p)$ determines the cut-off time, for which correlations with $y^0 \leq \tau_{cut}(p)$ will be discarded, which is equivalent to shrinking the system's history to $t'_H(p) = x_c^0 - \tau_{cut}(p)$.

However, doing the calculation in such a way would lead us to perhaps up to p different new history sizes, one for each momentum mode p. We do not want to split our history matrix into p differently sized smaller ones. We unify the p different history sizes $t'_H(p)$

into one new history size by calculating their weighted average $\bar{t}'_H$, i.e.

$$\bar{t}'_H = \frac{\sum_{p=0}^{N_{sp}} w_p \cdot t'_H(p)}{\sum_{p=0}^{N_{sp}} w_p}. \tag{111}$$

The weighting factor w_p for each mode is necessary, because on one hand, as we have seen in sec. 5.2.2, not every momentum mode thermalizes equally fast and on the other hand, different modes contribute differently strong to the general evolution of the system as it was for example the case of the tsunami peak momentum mode. In the following, we'll neglect the bar of the average and, similar than before, use $\bar{t}'_H \equiv t'_H$.

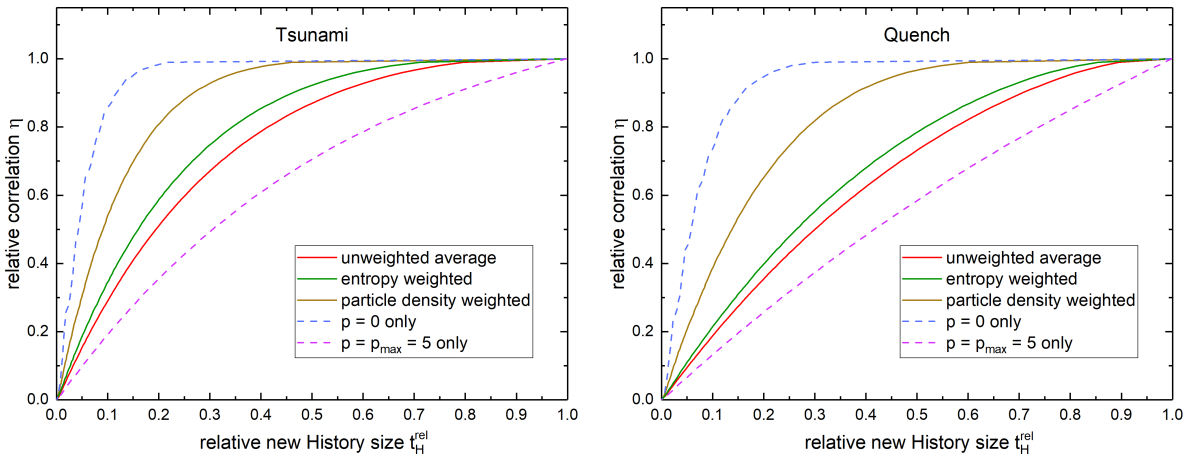
The weighting factor

The question that needs to be answered is, what to choose for the weighting factor w_p. To answer this question, we tested different weighting factors for the DHS algorithm. We considered the unweighted average i.e. $w_p = 1$, a weighting of each mode with its linear entropy $S_L(p, x_c^0)$ and its particle distribution $n(p, x^0)$. We chose these weighting factors because they, on one hand, reflect the thermalization state of the system, and on the other hand contain the occupation information about the different modes. As we stated previously, it seems reasonable to give momentum modes that have a high occupation and contribute strongly to the system's dynamics (like the peak mode of the tsunami or the zeroth mode) more weighting then momentum modes with a minuscule occupation and a tiny contribution. The difference between the particle density and the linear entropy, which from (61) we know that it is a function of the particle density, is, that the linear entropy is limited to values between 0 and 1 for $n(p, x^0) \to \infty$ and 0, respectively. Therefore the linear entropy is a more "tame" version of the particle distribution, which is not limited from above.

To test the DHS algorithm for the different weighting factors, we used both tsunami and quench initial conditions. The other settings are equivalent to the ones used in Fig. 24 and B.3 (see set $\mathcal{S}_6$ in appendix A).

We started the computation with a fixed maximum history of $t_{H_{\max}} = 100$. After filling this maximum history, the DHS algorithm was called to determine a new history. We varied the parameter η and measured the new history size t'_H that the algorithm determines for different weightings.

The results for $\lambda_{\text{Tsunami}} = 7$ and $\lambda_{Quench} = 2.5$ are shown in Fig. 30, where we show the dependency between η and the relative new history size $t'_H/t_{H_{\max}}$, which we'll from now on denote as t_H^{rel}.



(a) (b)

Figure 30: Depicted is the dependency between the relative remaining correlations η and the relative new history size $t'_H/t_{H_{\max}}$, determined by the DHS algorithm. The DHS algorithm was used for an initial history of $t_{H_{\max}} = 100$ and different weighting factors for both a) tsunami and b) quench initial conditions. Additionally, the results from the DHS algorithm when we only consider the zeroth or the maximum momentum mode, and thus do not compute the average (111), are indicated by the dashed curves.

The parameter η is a free input parameter, so its choice lies with the user of the algorithm. As of now, we do not know the concrete dependence between η and the deviations in the course of the system's time evolution. Further investigation regarding this area is needed in the future. However, considering that we observe an exponential decay of the unequal time correlations, taking the point where $1/e$ of the correlations are discarded, i.e. $\eta = 1 - 1/e \approx 0.63$, seems to be a reasonable choice.

In Tab. 5, we show the determined history sizes from Fig. 30 for $\eta = 0.63$. We can compare these new history sizes computed by the DHS algorithm with our results from Fig. 24 and B.3, that we showed previously. What we learned from these figures is that the courses of $F(x^0, x^0, 0)$ for $t_H = 30$ and $t_H = 50$ is already indistinguishable. Smaller histories like $t_H = 15$ still lead to good results. However, they show a visible alteration for the course of the correlation functions. Additionally, from the history dependency of the temperature and the average linear entropy (see Fig. 26 and Fig. 27), we saw that both final temperature and average linear entropy seem to converge against a constant value for $t_H \geq 20$ in the case of both quench and tsunami initial conditions. This leads

Determined history t'_H		
	Quench	Tsunami
$p = p_{\max} = 5$ only	55	42.6
unweighted average	40.8	27.3
entropy weighted	35.7	22.5
particle density weighted	19.2	12.4
$p = 0$ only	7.74	5.5

Table 5: Listed are the new history sizes t'_H for different weighting factors, determined by the DHS algorithm for $\eta = 1 - 1/e \approx 0.63$. We considered quench and tsunami initial conditions with couplings $\lambda_{\text{Quench}} = 2.5$ and $\lambda_{\text{Tsunami}} = 7$, respectively, and a starting maximum history size of $t_{H_{max}} = 100$.

us to the conclusion that the most suitable history size of the system should be around $20 \leq t_H^{\text{optimal}} \leq 30$. When we look at Tab. 5, we can see that the zeroth mode $p = 0$ extremely underestimates the history size. Conversely, when we only consider the maximum momentum mode, the determined history seems far too big. All the averages on the other hand lead to acceptable history sizes, but only the entropy weighted average seems to truly fit the criteria we set for the system's optimal new history matrix t_H^{optimal}. For this reason, the linear entropy is the weighting factor of our choice and will be used for the next discussion.

Avoiding unnecessary history cuts

There is one problem of the DHS algorithm that has not been addressed until now. That is the fact that without further checks, the DHS algorithm will always cut the history. In the worst case scenario this cut will have a linear dependency between η and t_H^{rel}. To make it more clear where this problem originates from, we'll consider the example of a vanishing coupling constant $\lambda = 0$. In this case, the amplitude of the $x^0 \neq y^0$ correlations does not decay exponentially with increasing distance from the time diagonal, but remains constant instead. This can for example be seen in Fig. 31, where we show the absolute value of the unequal time correlation function $F(x^0 = t_H, y^0, 0)$, with $t_H = 100$, as a function of y^0, like it is used in the DHS algorithm.

The DHS algorithm now computes the total area under this curve and stores it as $\Lambda_F^{\max}(x_c^0, t_{\text{shift}}, 0)$. Starting at $t_{\text{shift}} = 0$, the algorithm then begins to subtract the with the cutoff parameter τ increasing area under the curve from the total area and thus calculates $\Lambda_F(x_c^0 = t_H, \tau, 0)$, which is then further used to determine the relative remaining correlation η from (110) (see sec. 7.4.2). The problem is now, that because of the vanishing coupling, the amplitude of the oscillation of $F(x^0, y^0, 0)$ did not decline from $y^0 = x^0 = t_H$

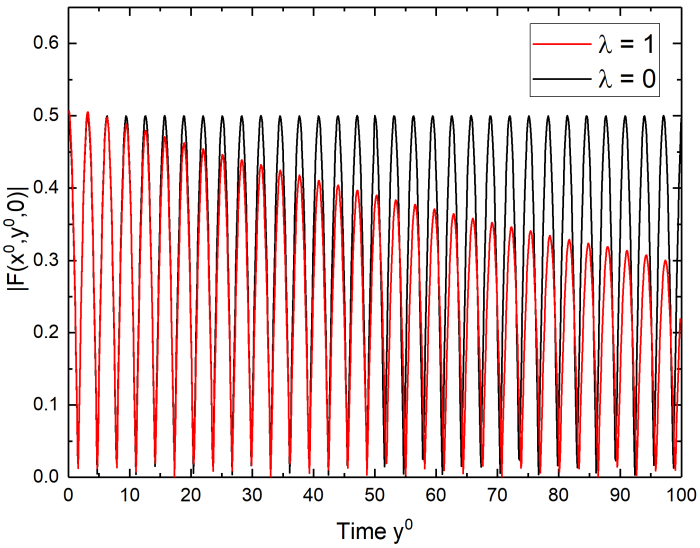


Figure 31: Depicted is the absolute value of the correlation function $F(x^0, y^0, 0)$ at $x^0 = 100$ for the two couplings $\lambda = \{0, 1\}$. The maximum starting history was $t_{H_{\max}} = 100$ and tsunami initial conditions were used. For $\lambda = 0$ one can see that the correlations do not decline at all, while they decline to some degree for $\lambda = 1$. Integrating the graph along y^0 lead to a close to linear increasing area for $\lambda = 0$. For $\lambda = 1$ this increase should be slightly diminished for large y^0, but still somewhat close to linear.

to $y^0 = 0$. This means, that the area $\Lambda_F(x_c^0 = t_H, \tau, 0)$ declines roughly linear with increasing τ, which in turn results in an almost linear dependency between η and t'_H. The algorithm would then run until the chosen η (for example $\eta = 1 - 1/e$) is reached. In this case the relative new history size would be $t_H^{rel} \approx \eta = 1 - 1/e$ despite the fact that the unequal time correlations have not declined in the slightest.

This problem remains when the coupling is too small for the predetermined maximal history size $t_{H_{\max}}$ of the system, like it is the case for the coupling $\lambda = 1$, which we also show in Fig. 31. In this case, the correlation functions do not have the time to decline sufficiently within the range of this fixed history and the course of $t'_H(\eta)$ will be somewhat close to a linear one. If the DHS algorithm then starts to cut the history even further, the resulting smaller history might lead to significant deviations. It goes without saying, that the DHS algorithm should in this case refrain from a further reduction of the already too small history and request a bigger one from the user.

In the following, we'll address this issue and present our method on how to deal with it. We already stated that the entropy weighting is our measure of choice. For this reason, we'll focus our discussion on this weighting method. Note however, that the solution to our problem is not limited to the entropy weighting and could also be used for others.

Above, we've discussed that the DHS algorithm shows a close to linear dependency between η and t_H^{rel} if the maximal history $t_{H_{max}}$ is too small for the chosen coupling constant λ. This problem is reflected in Fig. 32, where we show the dependency between η and t_H^{rel}, determined by the DHS algorithm, for various coupling constants and tsunami initial conditions. We chose the same maximal history $t_{H_{\max}} = 100$, as we did for Fig. 30.

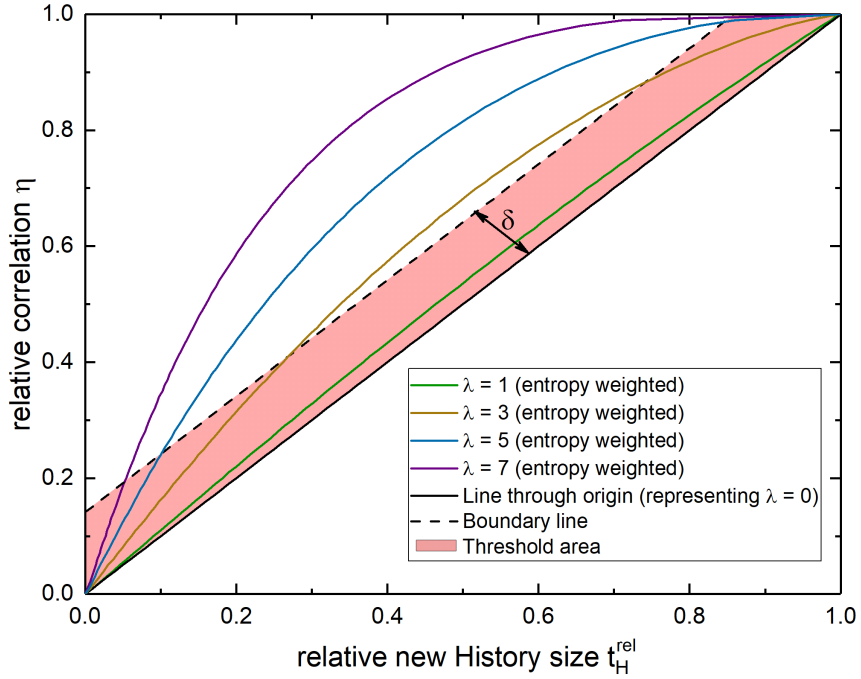


Figure 32: Courses between the relative new history size $t'_H/t_{H_{\max}} = t_H^{rel}$ and the relative remaining correlations η, determined with the DHS algorithm for different couplings and an entropy weighting. The considered system has tsunami initial conditions and a maximal starting history $t_{H_{\max}} = 100$. The red threshold area is parameterized by δ, which describes the distance between the black line through the origin and the dashed boundary line. All graphs that are completely within the threshold area will be deemed too linear and the DHS algorithm avoids cutting and demands a larger maximal history $t_{H_{max}}$.

In Fig. 32 we can see that, as we already argued above, the dependency between η and t_H^{rel} for $\lambda = 1$ is comparatively close to the linear one that is indicated by the black line. This is expected, because, as we can see especially good from Fig. 31, a maximal history of $t_{H_{\max}} = 100$ does not seem to be big enough for the unequal time correlation functions to decline sufficiently enough. In this case, we should refrain from cutting the history even further and request a bigger maximal history size. On the other hand when we consider higher couplings $\lambda \geq 3$, like we did in Fig. 28, we can see that the dependency between η and t_H^{rel} deviates quite significantly from the linear course and cutting the history further should be acceptable.

As of now, we implemented a rather basic solution to the problem of unnecessary history cuts. That is, we defined a threshold area, indicated by the red area in Fig. 32, which corresponds to the area between the black line through the origin and another parallel, dashed black line which is shifted by a distance δ away from the first one (see Fig. 32). This distance δ can in principle range from 0 to $1/\sqrt{2}$. When the DHS algorithm does its calculation, it sweeps through η to get the dependency $\eta(t_H^{rel})$, like they are shown in Fig. 32. If the graph of $\eta(t_H^{rel})$, does not leave the threshold area at least at one point, its course is deemed too linear. In this case, the DHS algorithm avoids further cutting and suggests the user to restart the computation with a bigger initial maximum history size (see also appendix C.2). On the other hand, if the graph leaves the area, like it is the case for $\lambda = \{3, 5, 7\}$ in Fig. 32, the DHS algorithm deems the chosen maximal history $t_{H_{\max}}$ as cuttable. In this case, the DHS algorithm takes the users input value of η (for example $\eta = 1 - 1/e$), determines the corresponding new history size t'_H and resizes the old history to t'_H via the *resizeHistory*-routine (appendix C.3). For further information about the DHS algorithm see appendix C.2.

The last remaining question that needs to be answered is how to choose the threshold area or rather the parameter δ from Fig. 32, so that the DHS algorithm accepts reasonable courses between η and t_H^{rel} and declines the ones where the course is too linear. As of now, we use δ as a free parameter in the DHS algorithm and can only give an estimate on which δ should be chosen. We get this estimate in the following way. At first, we use Fig. 32 and our previous criteria $\eta = 1 - 1/e$, to determine new suitable history sizes for the different couplings. The results are shown in Tab. 6.

$\eta = 0.63$	
λ	t'_H
1	59.6
3	44.8
5	32.5
7	22.5

Table 6: Listed are the new history sizes t'_H determined by the DHS algorithm with an entropy weighting for different coupling constants λ and tsunami initial conditions. The initial maximal history size is $t_{H_{\max}} = 100$ and the relative remaining correlation is $\eta = 1 - 1/e = 0.63$.

We considered the equal time correlation function $F(x^0, x^0, p)$ for different momentum modes, couplings and history sizes from appendix B.3 (Fig. B.7-B.9) and sec. 7 (Fig. 24). We want to choose our estimate in such a way, that the suggested history sizes from Tab. 6, for the corresponding couplings, should at worst only lead to small deviations in the course of the equal time correlation function. When we do the comparison, only

the case of $\lambda = 1$ (see Fig. B.7) still shows significant deviations around the determined history $t'_{H_{\lambda=1}} = 59.6$. The $F(x^0, x^0, p)$ for the higher couplings $\lambda = 3$ to $\lambda = 7$, show small deviations around their determined history sizes. Because of this, δ should be chosen in such a way, that the threshold area completely includes the graph of $\eta(t_H^{rel})$ in Fig. 32 for $\lambda = 1$, while it should still exclude parts of the graph for $\lambda = 3$. From Fig. 32 we can determine that δ should therefore lie between $0.026 \ll \delta_{\text{estimate}} < 0.130$ to avoid cutting a system's history, which is already too small. We want to emphasize, that an estimate in favor of larger δ should suffice for the choice of δ. If δ is too big, the DHS algorithm only suggests the user to choose a larger history size, but does not alter the current one or interrupt the time evolution. The choice on how to proceed from this point lies within the users own discretion.

8. Conclusion

8.1. Summary

In this work, we implemented the history cut-off reduction scheme for the case of a spacial homogeneous system in (1+1)-dimensions into our solver of the Kadanoff-Baym equations. With the reduction scheme in place, we observed and discussed the time evolution of the system for our test model, a scalar field theory with a φ^4-interaction, to get an idea of its general behavior for different initial conditions. For both quench and tsunami we considered the case of 1- and 2-Loop corrections. We observed that 1-Loop corrections could not describe the thermalization of the system. However, for 2-Loop corrections, at late times, the system approaches an equilibrium state, where no more changes occurred. For the tsunami initial conditions, we additionally observed the particle distribution, as well as the linear entropy under the further condition of a thermal background. The particle distribution and the linear entropy both indicated thermalization and we could observe a temperature increase between the initial temperature of the background and the system's temperature after thermalization.

To further improve our understanding of the numerical solution for the Kadanoff-Baym equations, we discussed numerical effects. Finite lattice effects occurred for both 1- and 2-loop approximations, however for the latter, these effects were highly suppressed and changed the course of the evolution only for very small volumes. Besides finite lattice effects, we also analyzed the system's dependency on the time and spacial lattice spacing. For decreasing time spacing we saw that the numerical error decreases quadratically. In case of the spacial spacing, the quench also indicated a quadratic decrease in the numerical error for the zeroth mode. However, the results for the tsunami were inconclusive within the observation time.

Lastly, we introduced and discussed the history cut-off reduction scheme. We gave a justification of the history cut-off through the exponential decline of the correlation functions with their distance from the time diagonal and showed the steps necessary to implement the history cut-off. Furthermore, we used a simplified example to illustrate an efficient way of implementing the history cut-off into a numerical solver, using a double cyclic memory. After presenting the implementation of the history cut-off, we discussed its effects on the course of the system's time evolution depending on the size of the history. We showed that, depending on the initial conditions and the coupling, small history sizes lead to significant deviations. However, the courses quickly approach each other when the history size is increased. We then proceed to presented our method, the DHS algorithm,

to find a suitable history size for the system. This method was discussed for different couplings and weighting factors for the weighted average that is used in the calculations. We decided that the linear entropy is the most reasonable weighting factor to determine a fitting history sizes. Afterwards we discussed an issue, that arises from the way the DHS algorithm works and presented a way to avoid it.

Besides the work that is directly presented here, we also implemented a graphical user interface into our framework. A basic introduction to it can be found in appendix D.

8.2. Outlook

The history cut-off for the case of a spacial homogeneous system was only a small stepping stone to test the viability and the best approach to implement it for the more complex case of a spacial inhomogeneous system. We already started working on a history cut-off implementation for the inhomogeneous system. However, more work is needed before a functioning version can be reported. This work lays the foundation for the implementation of the history cut-off in the inhomogeneous case and should also helps to provide a better understanding of the numerical solution of the Kadanoff-Baym equations, so that further improvements can be made in the future. One of these future improvements, whose implementation is already planed, is a perhaps improved version of the DHS algorithm for the solver of the inhomogeneous Kadanoff-Baym equations.

Acknowledgment

First of all, I want to thank my supervisors, Prof. Dr. Werner Porod and Thomas Garratt, for their help and support during my work for this thesis. I am grateful that there was always someone listening when I had a theoretical question, or when I wanted to discuss how to approach the work done in this thesis.

I want express my deep thanks to my family, who always supported me throughout my studies and the creation process of this work. Without their aid, this work would have never been possible and I'm very grateful to have such amazing parents and such a nice sister.

I also want to thank Simon Geisler. He is a very good friend of mine and our discussions, be they physics related or not, are always fun and productive. You are a central pillar in my endeavors to always improve. I'm glad that we met each other during our studies and had the chance to complete our master's degree together.

Lastly, I want to thank all my other friends, who kept me entertained and helped me to relax during the creation of this thesis.

Appendix

A. Settings

The following tables show a detailed list of the settings, that were used in our solver to produce the data that is shown in this work. The settings tables are denoted as sets $\mathcal{S}_i$, with $i = 1, ..., 6$ and grouped together depending on the section in which they have been used. Parameters that equal "varied" are the ones, which vary or are varied for the figures that are listed at the bottom of the table.

Model

For the model section (sec. 5), we choose two different sets. The first is used in the beginning of the section, when the quench and the tsunami are shown for the first time. The second set is used for the observation of the particle density and the linear entropy. To get a good resolution for the plot of $n(x^0, p)$, we used a significantly greater number of spacial points $N_{sp} = V_s/a_s$ in the second set.

Set $\mathcal{S}_1$		
	Tsunami	Quench
Model parameters	$\lambda = 7$ $M_{Init} = 1$	$\lambda = 2.5$ $M_{Init} = 0.5$ $T_0 = 2$
Initial conditions	$A = 3$ $p_{ts} = 1$ $\sigma = 0.2$	
Lattice parameters	$a_t = 0.1$ $a_s = 0.4$ $V_s = 60$	$a_t = 0.1$ $a_s = 0.4$ $V_s = 60$
History size	$t_H = 15$	$t_H = 15$
Fig. 9-13,B.1		

Set $\mathcal{S}_2$	
	Tsunami + thermal background
Model parameters	$\lambda = 7$ $M_{Init} = 1$ $T_0 = 2$
Initial conditions	$A = 4$ $p_{ts} = 2.5$ $\sigma = 0.1$
Lattice parameters	$a_t = 0.025$ $a_s = 0.4$ $V_s = 400$
History size	$t_H = 15$
Fig. 14,15	

Numerical effects

We choose to split our parameter choice for section 6 into three tables $\mathcal{S}_{3-5}$. The first and second table show the settings for finite volume effects and time spacing effects, respectively. For the last table, we additionally added the number of spacial points N_{sp}. We did this, because we varied both a_s and V_s to always get a constant integer number ($N_{sp} = 350$) for the spacial steps.

Set $\mathcal{S}_3$		
	Tsunami	Quench
Model parameters	$\lambda = 7$ $M_{Init} = 1$	$\lambda = 2.5$ $M_{Init} = 0.5$ $T_0 = 2$
Initial conditions	$A = 3$ $p_{ts} = 1$ $\sigma = 0.2$	
Lattice parameters	$a_t = 0.1$ $a_s = 0.4$ $V_s =$ varied	$a_t = 0.1$ $a_s = 0.4$ $V_s =$ varied
History size	$t_H = 15$	$t_H = 15$
Fig. 16,17,B.2		

Set $\mathcal{S}_4$		
	Tsunami	Quench
Model parameters	$\lambda = 7$ $M_{Init} = 1$	$\lambda = 2.5$ $M_{Init} = 0.5$ $T_0 = 2$
Initial conditions	$A = 3$ $p_{ts} = 1$ $\sigma = 0.2$	
Lattice parameters	$a_t =$ varied $a_s = 0.4$ $V_s = 60$	$a_t =$ varied $a_s = 0.4$ $V_s = 60$
History size	$t_H = 15$	$t_H = 15$
Fig. 18		

Set $\mathcal{S}_5$		
	Tsunami	Quench
Model parameters	$\lambda = 7$ $M_{Init} = 1$	$\lambda = 2.5$ $M_{Init} = 0.5$ $T_0 = 2$
Initial conditions	$A = 3$ $p_{ts} = 1$ $\sigma = 0.2$	
Lattice parameters	$a_t = 0.05$ $a_s =$ varied $V_s =$ varied $N_{sp} = 350$	$a_t = 0.05$ $a_s =$ varied $V_s =$ varied $N_{sp} = 350$
History size	$t_H = 15$	$t_H = 15$
Fig. 19		

Efficient solution of memory integrals

Throughout sec. 7 we used the same basics settings, so that we can easily compare our results. The only quantities that were varied are the history size t_H and the coupling constant λ.

Set $\mathcal{S}_6$		
	Tsunami	Quench
Model parameters	λ = varied $M_{Init} = 1$	λ = varied $M_{Init} = 0.5$ $T_0 = 2$
Initial conditions	$A = 3$ $p_{ts} = 1$ $\sigma = 0.2$	
Lattice parameters	$a_t = 0.1$ $a_s = 0.4$ $V_s = 60$	$a_t = 0.1$ $a_s = 0.4$ $V_s = 60$
History size	t_H = varied	t_H = varied
Fig. 20, 24-32,B.3-B.9		

B. Additional Graphs

The following section contains additional plots, that we could not fit into the main part of this thesis.

B.1. Model

Tsunami initial conditions

Fig. B.1 shows the time evolution of the isolated maximum momentum mode for the tsunami initial conditions from sec 5.2. This momentum mode is also shown in Fig. 12 and 13, but its amplitude is too small to notice anything than a constant line. For this reason an additional plot was made.

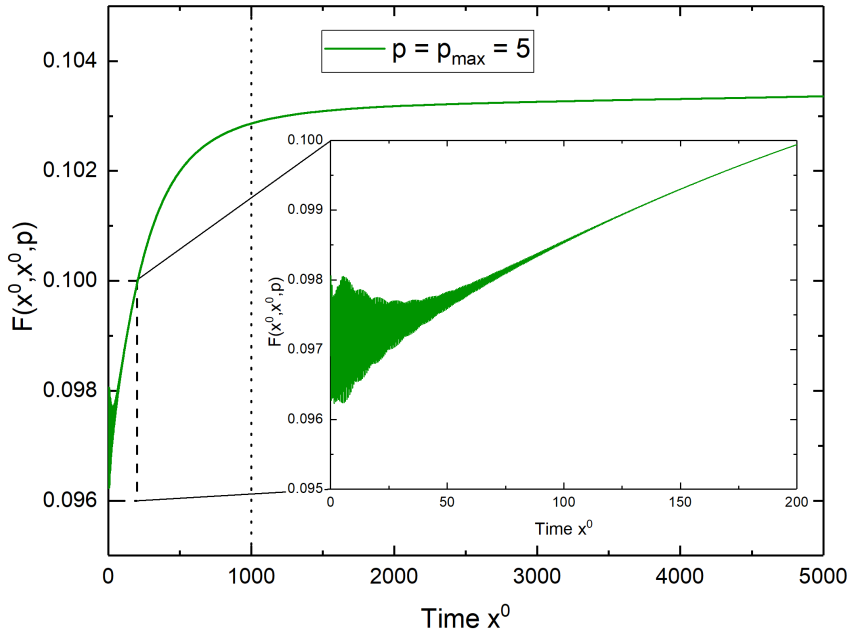


Figure B.1: Time evolution of the maximum momentum $p_{max} = 5$ mode for the equal time correlation function $F(x^0, x^0, p_{max})$ with tsunami initial conditions. The insert shows a magnification of the early time behavior of the mode. The settings we used for this plot are given in $\mathcal{S}_1$ (appendix A).

B.2. Numerical effects

Finite lattice effects

The finite lattice effects could already be seen in Fig. 16. For completeness, we also observed the tsunami. The result of this observation can be seen in Fig. B.2, where we show the time evolution of the tsunami for different finite lattice sizes V_s.

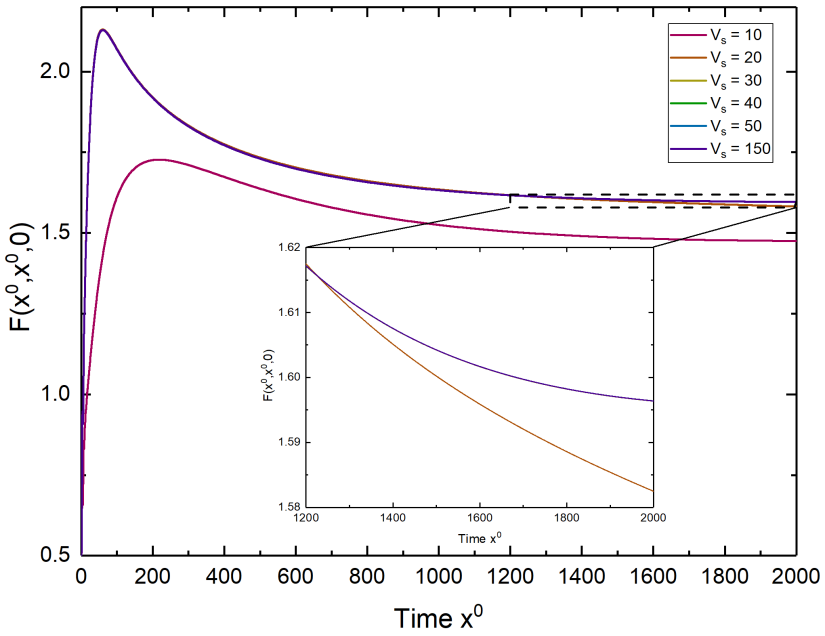


Figure B.2: Depicted is the course of the equal time correlation function $F(x^0, x^0, 0)$ for tsunami initial conditions and different lattice sizes V_s, ranging from $V_s = 10$ to 150. The insert shows a magnification of the graph from $x^0 = 1200$ to 2000. The settings we used for this plot are given in $\mathcal{S}_3$ (appendix A).

B.3. Effects of a history cut-off

This section shows additional plots for the effects of a history cut-off. The settings $\mathcal{S}_6$ were used to produce the data for all the plots, which are shown in this section.

Quench $\lambda = 2.5$

Here, we present the quench for $\lambda = 2.5$, that we discussed in sec. 7 together with the tsunami. We choose to place this Fig. B.3 here, because we wanted to keep a nice formatting for this work.

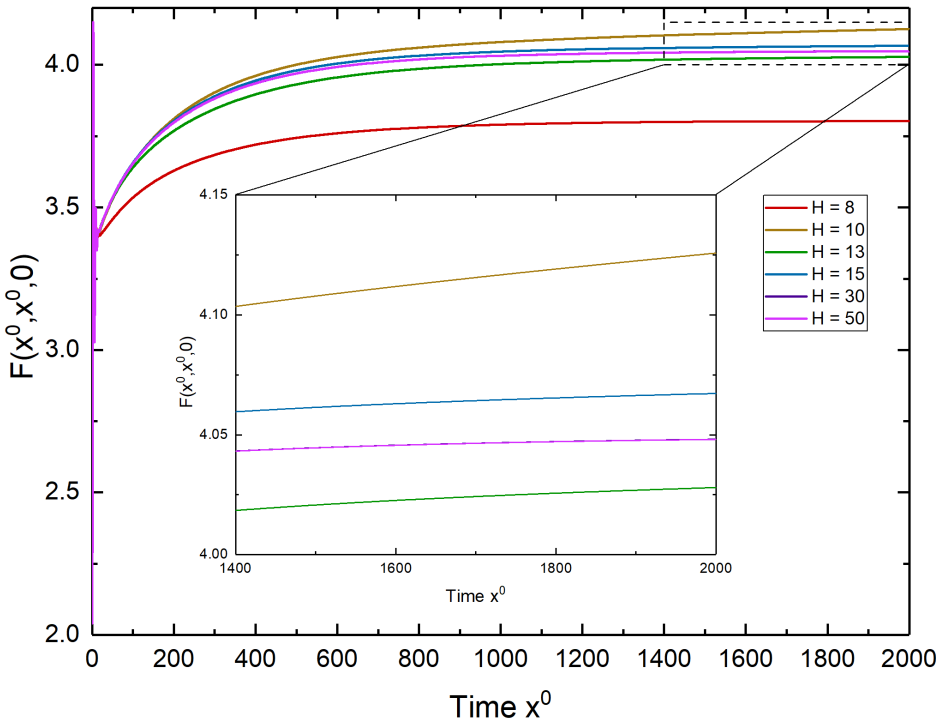
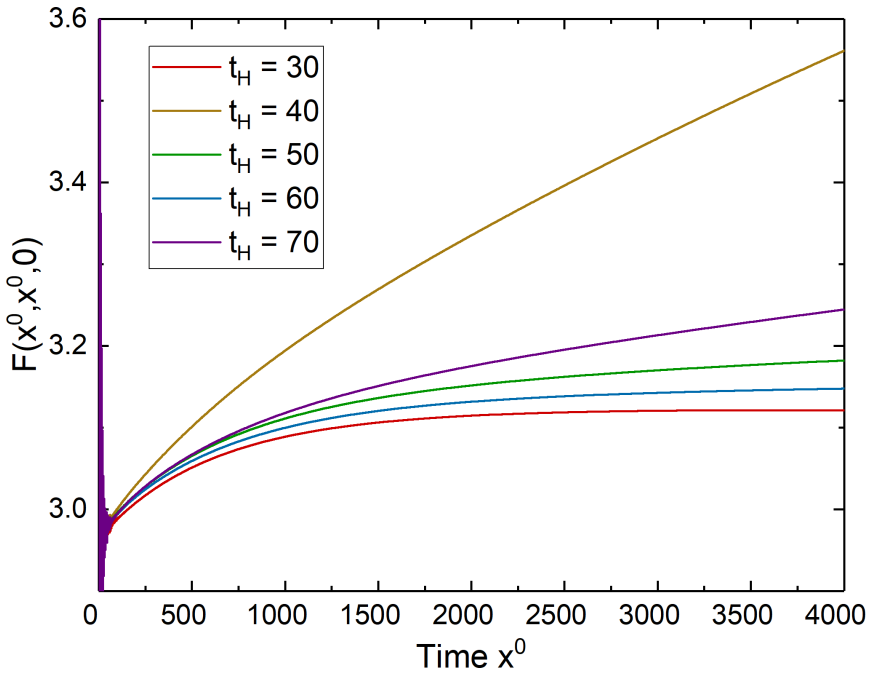


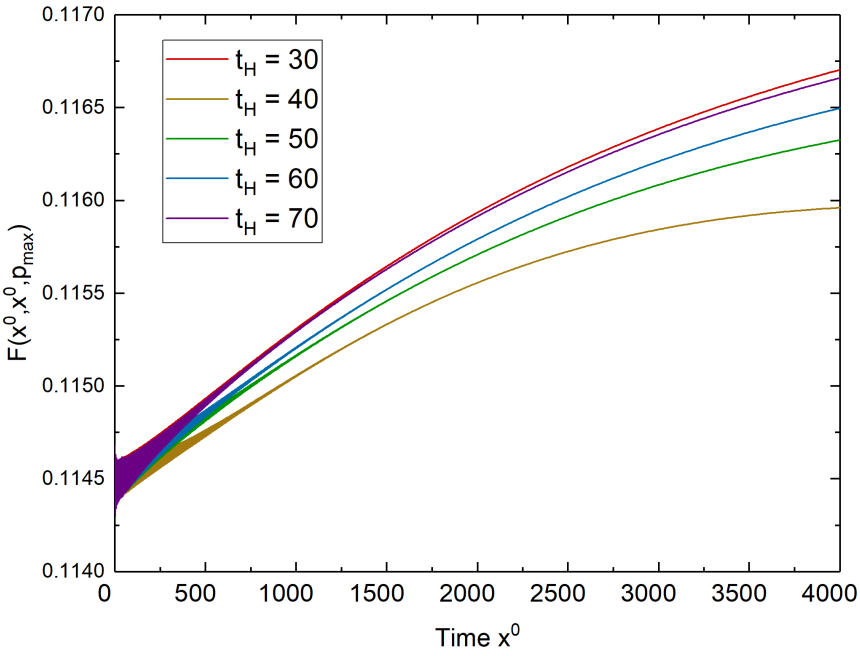
Figure B.3: Depicted is the equal time correlation function $F(x^0, x^0, 0)$ with quench initial conditions for different history sizes t_H ranging from 8 to 50. Similar to Fig. 24, the course of $F(x^0, x^0, 0)$ shows large deviations for $t_H \leq 10$. However, for the quench, even larger histories show comparatively large deviations. Nonetheless when comparing $t_H = 30$ and 50 the deviation again becomes minuscule.

Besides the case of $\lambda = 7$ for the tsunami and $\lambda = 2.5$ for the quench, we observed the system for various other coupling constants and history sizes. The results are shown in the figures B.4-B.9 below.

Quench $\lambda = 1$



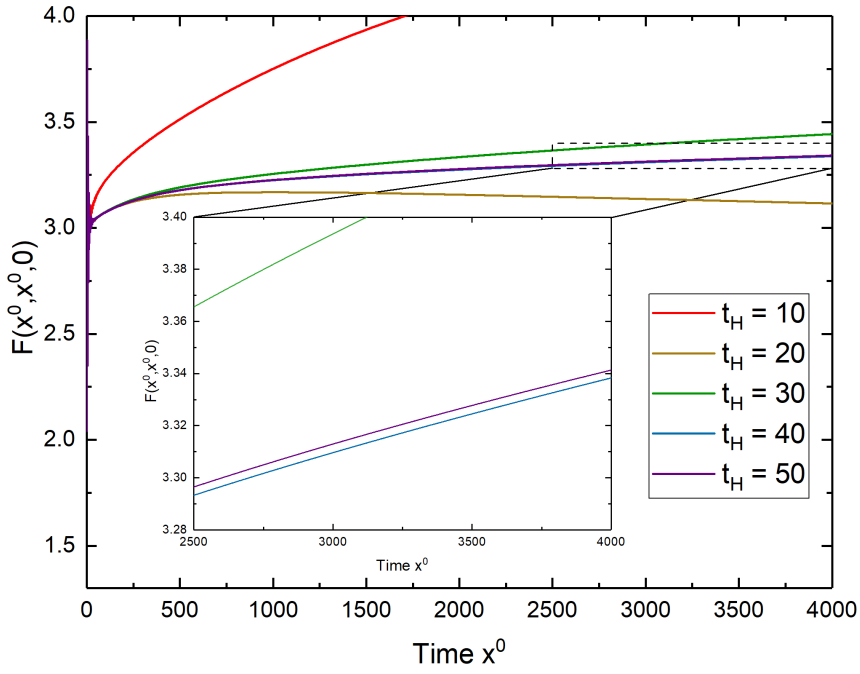
(a)



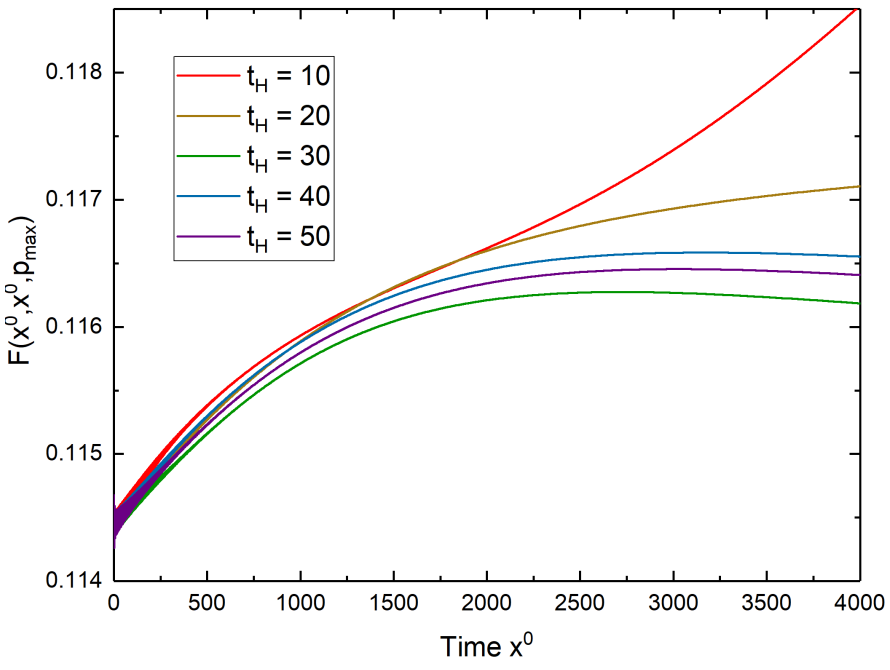
(b)

Figure B.4: Time evolution of the equal time correlation function $F(x^0, x^0, p)$ for the momentum modes a) $p = 0$ and b) $p = p_{\max} = 5$ and different history sizes. The system has quench initial conditions and a coupling of $\lambda = 1$.

Quench $\lambda = 1.5$



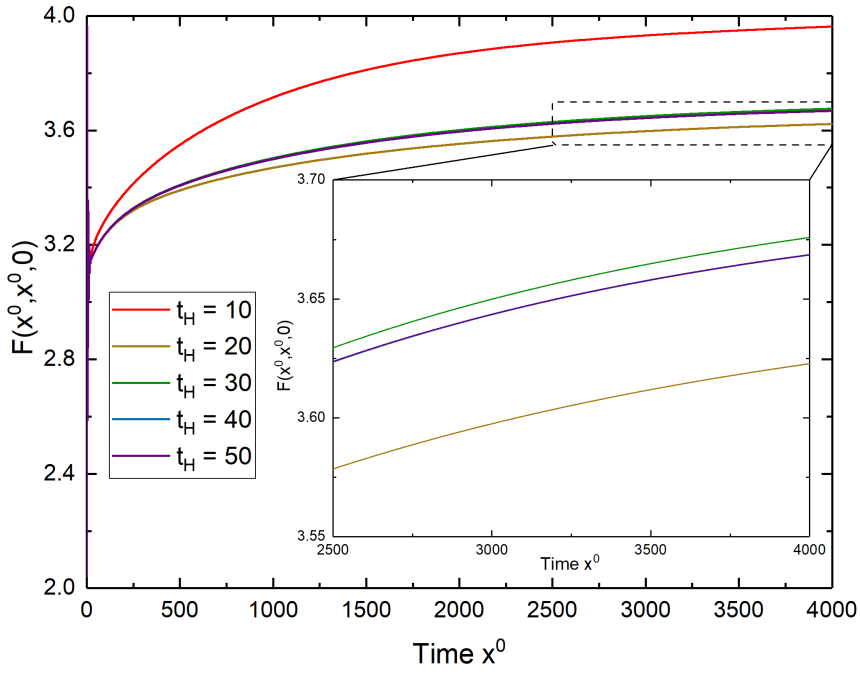
(a)



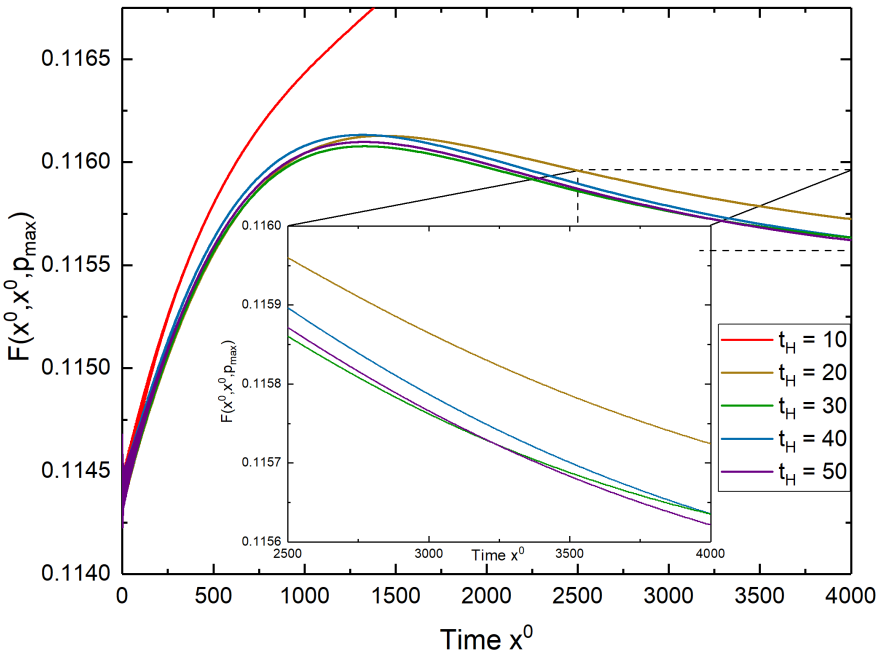
(b)

Figure B.5: Time evolution of the equal time correlation function $F(x^0, x^0, p)$ for the momentum modes a) $p = 0$ and b) $p = p_{\max} = 5$ and different history sizes. The system has quench initial conditions and a coupling of $\lambda = 1.5$.

Quench $\lambda = 2$



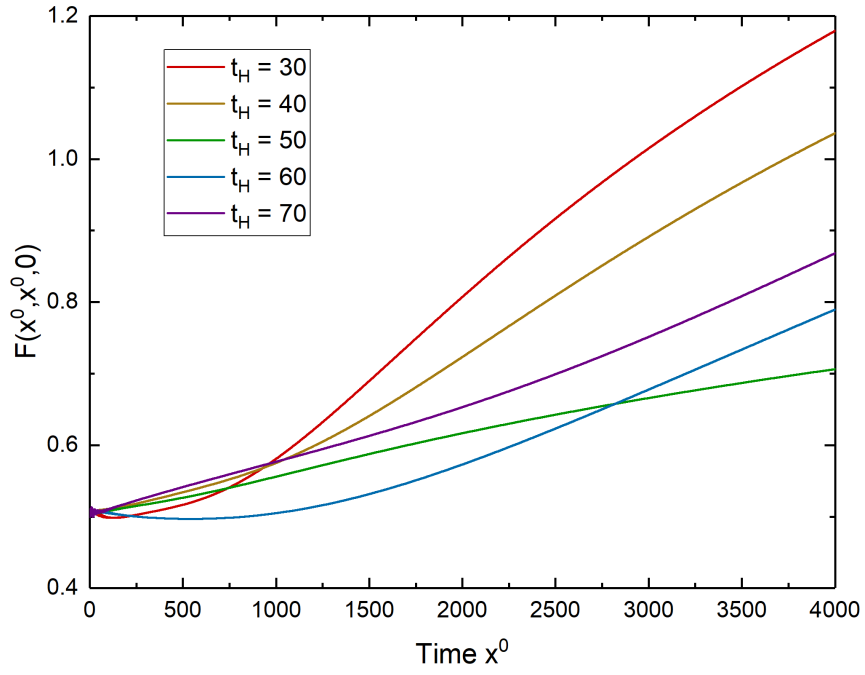
(a)



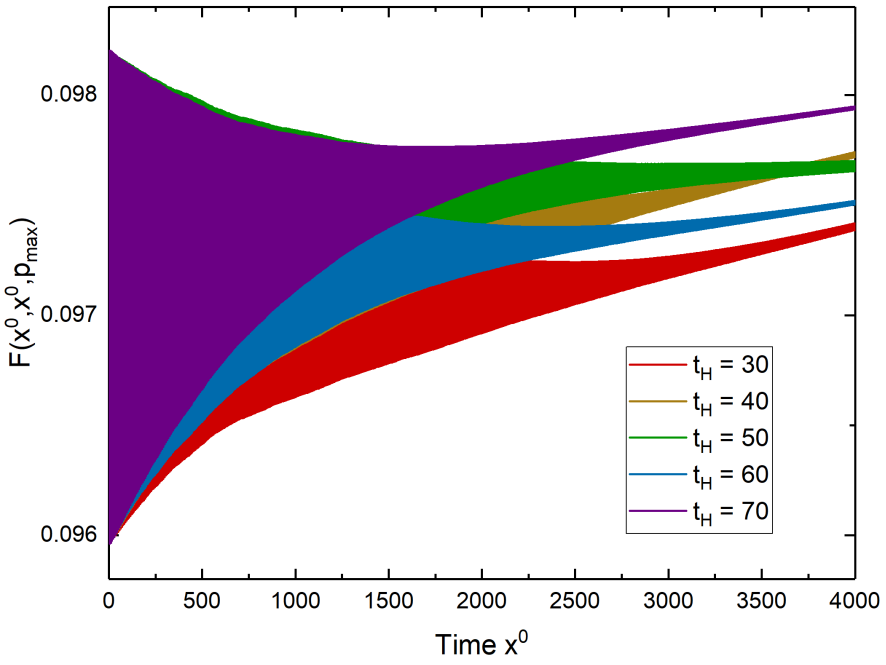
(b)

Figure B.6: Time evolution of the equal time correlation function $F(x^0, x^0, p)$ for the momentum modes a) $p = 0$ and b) $p = p_{\max} = 5$ and different history sizes. The system has quench initial conditions and a coupling of $\lambda = 2$.

Tsunami $\lambda = 1$



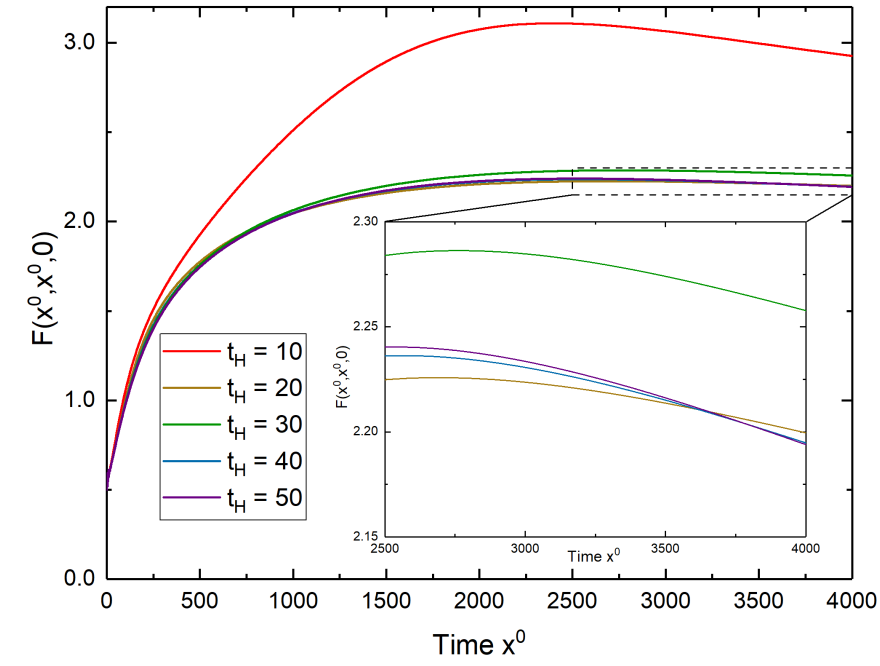
(a)



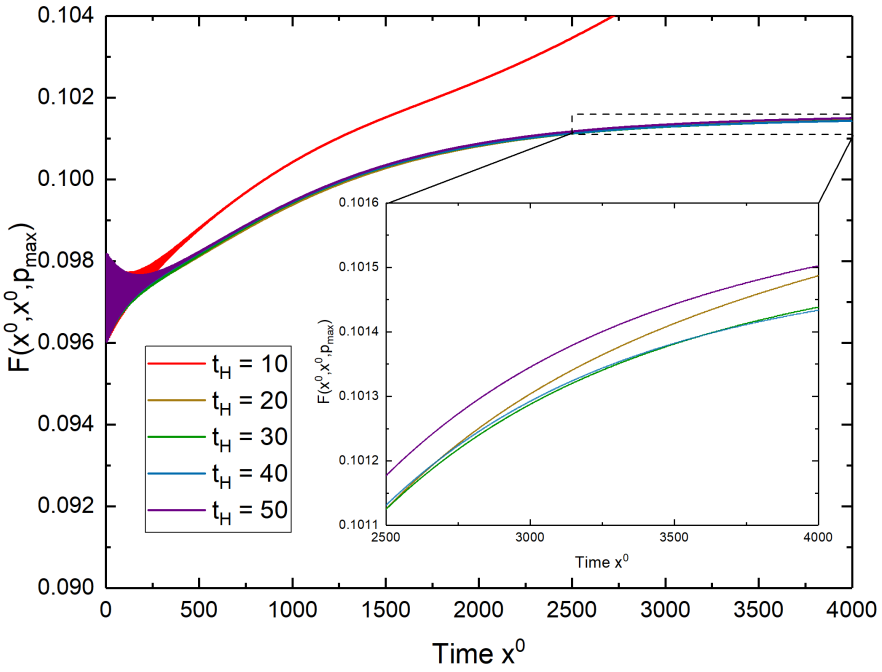
(b)

Figure B.7: Time evolution of the equal time correlation function $F(x^0, x^0, p)$ for the momentum modes a) $p = 0$ and b) $p = p_{\max} = 5$ and different history sizes. The system has tsunami initial conditions and a coupling of $\lambda = 1$.

Tsunami $\lambda = 3$



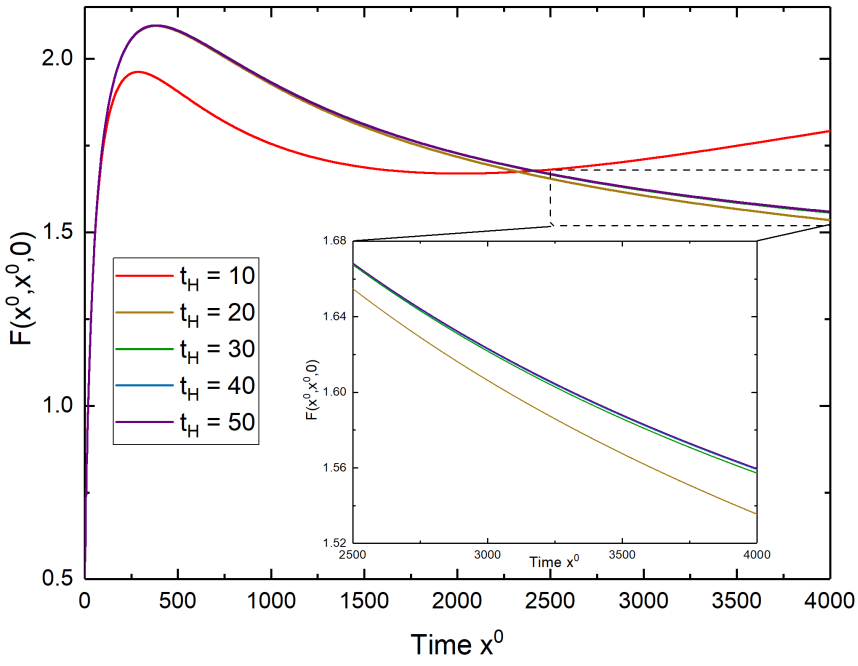
(a)



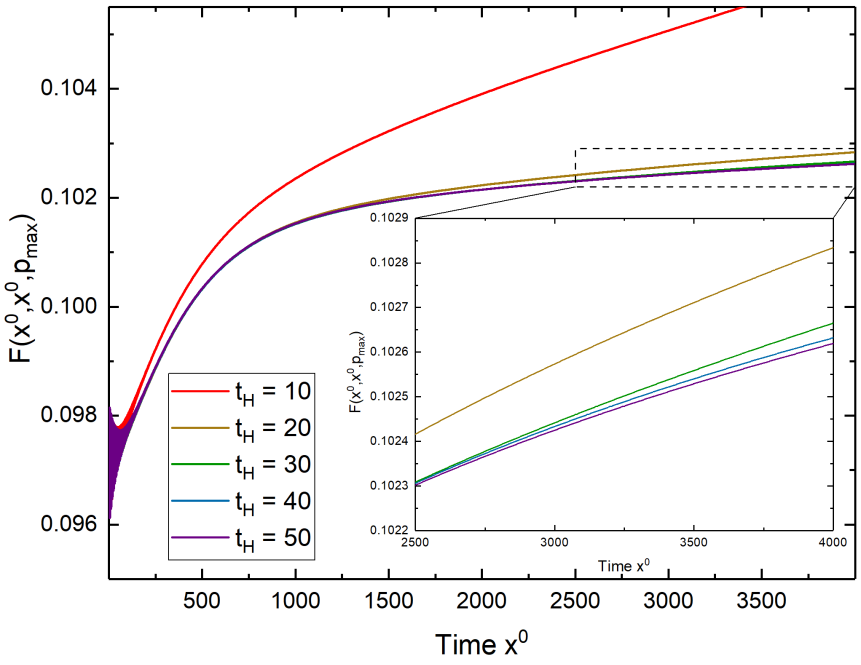
(b)

Figure B.8: Time evolution of the equal time correlation function $F(x^0, x^0, p)$ for the momentum modes a) $p = 0$ and b) $p = p_{\max} = 5$ and different history sizes. The system has tsunami initial conditions and a coupling of $\lambda = 3$.

Tsunami $\lambda = 5$



(a)



(b)

Figure B.9: Time evolution of the equal time correlation function $F(x^0, x^0, p)$ for the momentum modes a) $p = 0$ and b) $p = p_{\max} = 5$ and different history sizes. The system has tsunami initial conditions and a coupling of $\lambda = 5$.

C. Code segments

C.1. Double cyclic memory

This section shows two code extractions, which perform the cycling of the time indicies, as well as the transformation from the matrix to the line vector representation. These functions determine where data has to be written into and read from the memory and more or less represent the double cycled memory itself. Whenever data is written to or has to be read from the memory, the respective cycling function has to be called to determine the line vector index correctly. Otherwise the memory will return the wrong data or write it at a false position.

```
1 unsigned int Npointfunc::calcCycleHomIndex(unsigned int x0, unsigned int
      y0, unsigned int p)
2 {
3   return (x0HistoryStep*(x0%mHistorySize) + y0Step*(y0%mHistorySize) + p
      );
4 }
```

Listing 1: Code for the cycling routines, that generate the effect of the double cyclic memory for the spacial homogeneous case.

The first code segment manages the cycling for the spacial homogeneous case for two-point functions. The real time indicies x^0, y^0 and p are taken as inputs by the function. Using the member variable *mHistorySize* (N_H), the two time indicies are cycled via a modulo (%) operation. The remaining cycled real time index is then transformed to a line vector index by multiplying it with the x^0/y^0-stepping factor of the history matrix. The stepping factors are given by x0HistoryStep $= N_H \cdot$ y0Step $= N_H \cdot N_{sp}$ for the x^0-stepping and y0Step $= N_{sp}$ for the y^0-stepping. The spacial index does not need cycling and is in the case of the line vector representation simply added at the end.

```
1 unsigned int Npointfunc::calcCycleIndex(std::vector<unsigned int>
    paramVec){
2   unsigned int indexOut = 0;
3
4   for(int i = 0; i < paramVec.size(); i+=2){
5     indexOut += mStepSizeVec[i]*(paramVec[i]%mHistorySize); //Adding the
          cycled time indizes
6   }
7   for(int i = 1; i < paramVec.size(); i+=2){
8     indexOut += mStepSizeVec[i]*(paramVec[i]);  //Adding the spacial
        indizes
```

```
9     }
10
11    return indexOut;
12  }
```

Listing 2: Code for the cycling routines, that generate the effect of the double cyclic memory for the general, inhomogeneous case.

The second code segment shows the more general case for an inhomogeneous system with an arbitrary number of n-point-functions. In this case the number of input parameters is not fixed like it has been the case in the previous code segment. For this reason, the input parameters are handed over via a vector. Every component of this vector with an even index contains a time index, which is cycled, while every component with an uneven index contains a spacial component, where no cycling is necessary. For the computation of the line vector index, each cycled (or not cycled in the case of the spacial part) component is multiplied by their respective stepping factor which is contained in a vector *mStepSizeVec*, that is stored as a member variable in the *Npointfunc*-class.

C.2. DHS algorithm

In the following, we present the implementation of the full DHS algorithm, which avoids unnecessary history shrinking when the system's old history matrix is already too small. This version of the code only contains the case of the entropy weighting.

The algorithm is separated in two parts. The first is the *findNewHistory*-routine, which manages the determination of the system's new history size t'_H for a given η like it has been discussed in sec. 7.4. The second part is the *determineHistorySize*-routine (the full DHS algorithm), which contains the *findNewHistory*-routine to find a new history, but also checks if the system's history should be shrunk at all.

We begin our discussion with the *findNewHistory*-routine.

```
1  unsigned int Model::findNewHistory(double eta,unsigned int t, Npointfunc
       * func)
2  {
3
4     unsigned int HistoryShift = func->getShiftParameter();
5     unsigned int SpacialPoints = lattice->SpacialPoints();
6
7     double AverageNewHistSizeEntropyWeight=0; //Average new history size
8     double Total_Entropy=0;    //The total entropy is needed for the
          denominator of the weighted average
```

```
9
10    //-----For loop to find the new history size for each mode----
11    for(unsigned int p = 0; p < SpacialPoints; p++){
12      double Stat_TotalIntegral = 0, Spec_TotalIntegral = 0;
13      std::vector<double> Stat_IntegralParts, Spec_IntegralParts;
14
15      double intVal = 0;
16
17      //---For loop that performs the numerical integration of the
            absolute value of the statistical and spectral component-------
18      for(unsigned int i = HistoryShift; i < t-1; i++){
19
20        //Integral for the statistical part
21        intVal = 0.5*parameters.dt*(std::abs(func->val(func->
              calcCycleHomIndex(t,i,p))) + std::abs(func->val(func->
              calcCycleHomIndex(t,i+1,p))));
22        Stat_IntegralParts.push_back(intVal);
23        Stat_TotalIntegral += intVal;
24
25        //Integral for the spectral part
26        intVal = 0.5*parameters.dt*(std::abs(func->specval(func->
              calcCycleHomIndex(t,i,p))) + std::abs(func->specval(func->
              calcCycleHomIndex(t,i+1,p))));
27        Spec_IntegralParts.push_back(intVal);
28        Spec_TotalIntegral += intVal;
29      }
30      //-----------------------------------------------
31
32      double Stat_CurrentIntegral = Stat_TotalIntegral;
33      double Spec_CurrentIntegral = Spec_TotalIntegral;
34
35      unsigned int tau_cut = 0;
36
37      //----While loop that runs until the relative remaining correlation
            eta drops under the input threshhold-----
38      while(tau_cut < Stat_IntegralParts.size()){
39
40        Stat_CurrentIntegral -= Stat_IntegralParts[c];
41        Spec_CurrentIntegral -= Spec_IntegralParts[c];
42
43        tau_cut++;
44        if(Stat_CurrentIntegral/Stat_TotalIntegral < eta) break;
45        if(Spec_CurrentIntegral/Spec_TotalIntegral < eta) break;
46      }
47      //---------------------------------
```

```
48      unsigned int Mode_NewHistSize = t-HistoryShift-tau_cut; //
            Calculating the new history size for each mode
49
50
51      double Mode_Entropy = calcLinearEntropy(t,p,func);   //Calculating
            the entropy of the mode
52
53      AverageNewHistSizeEntropyWeight += Mode_Entropy*Mode_NewHistSize; //
            Calculating the numerator of the weighted average
54      Total_Entropy += Mode_Entropy;
55    }
56    //-----------------------------------------------
57
58    AverageNewHistSizeEntropyWeight/=Total_Entropy; //Calculating the
          weighted average
59
60    return (unsigned int)AverageNewHistSizeEntropyWeight;
61    }
```

Listing 3: Code of the *findNewHistory*-routine that, is used to find a fitting history size for a given value of the relative remaining correlation η

The *findNewHistory*-routine is the heart of the DHS algorithm and runs over all momentum modes p. For each mode, the total integral of the absolute value of both correlation functions (106) and (107) is calculated via the trapeze integral, that was already used for calculating the memory integrals. While calculating $\Lambda^{\max}_{F/\rho}$, each segment of the numerical integration is stored in a vector *Stat/Spec_IntegralParts*. After the calculation of $\Lambda^{\max}_{F/\rho}$ is done, the algorithm uses the previously stored segments of the integral to compute (108) and (109) and checks if the relative correlation integrals (110) drop below the threshold η. From the variable τ_{cut}, the new history sizes *Mode_NewHistSize* for the given η and for each mode are determined and together with the linear entropy *Mode_Entropy* of each mode, their weighted average *AverageNewHistSizeEntropyWeight* is calculated and returned.

Now, we'll focus our discussion on the complete DHS algorithm, which is shown in the *determineHistorySize*-routine

```
1  unsigned int Model::determineHistorySize(double eta, double delta,int
       sweepCount, unsigned int t, Npointfunc* func){
2
3    unsigned int newHistory;
4    unsigned int oldHistory = func->getHistorySize();
5    double relativeHistory;
```

```
6
7     if(delta >= 1/sqrt(2)){
8       std::cout << "DHS: The delta you choose to avoid unnecessary
            shrinking is too big!" << std::endl
9       << "Please choose a delta smaller than 1/sqrt(2)" << std::endl;
10      return oldHistory;
11    }
12    for(int j = 0; j < sweepCount; j++){
13      double etaSweep = 1/sweepCount*j;
14
15      newHistory = findNewHistory(etaSweep,t,func);
16      relativeHistory = (double)newHistory/oldHistory;
17
18      if(std::abs(etaSweep - relativeHistory)/sqrt(2) > delta){
19        newHistory = findNewHistory(eta,t,func);
20        return newHistory;
21      }
22
23    }
24    std::cout << "DHS: The chosen initial maximum history might be too
          small for the system!" << std::endl
25    << "If possible, please choose a bigger starting history size" << std
          ::endl;
26    return oldHistory;
27  }
```

Listing 4: Code of the *determineHistorySize*-routine to determine a fitting history size and avoid cutting a too small one

At first, the routine checks, if the input δ is bigger or equal than $1/\sqrt{2}$. From sec.7.4.2 and Fig. 32, we know that δ has to be smaller than $1/\sqrt{2}$, otherwise the algorithm wouldn't accept any possible course of $\eta(t_H^{rel})$. If δ is too big, a warning is given to the user and the algorithm returns the previous history size. Thus the DHS algorithm does not search for a new history size in this case. After the check of δ, the routine starts to sweep through η from 0 to 1 with a rate of $1/sweepCount$ to find the course of $\eta(t_H^{rel})$. The routine checks if at least one of the point for the course of $\eta(t_H^{rel})$ is further way from the diagonal than the distance δ. This means that at least one point must lie outside the threshold area, as we discussed in sec.7.4.2. If this is the case, the algorithm uses the η that was specified by the user, determines the corresponding new history size *newHistory* and returns it. Otherwise, the algorithm gives a warning that the old history might be too small and returns it. Thus, no cut is performed in this case.

C.3. Resizing the systems history

When determining a new history size during the run time of the numerical solution, a resizing of the previous history and therefore storage is necessary. In our program, this is done via the *resizeHistory*-routine shown below.

```
1  void Npointfunc::resizeHistory(unsigned int New_HistorySize,unsigned int
       t)
2  {
3    std::vector<std::complex<double> > NewVal;  //Storage for the data
       that gets transfered from the old history storage
4
5    int N = lattice->getN();
6    int dimension = lattice->getdim();
7    unsigned long long New_HistoryArraySize = pow(New_HistorySize,2)*pow(N
       ,dimension);   //Calculate size ofthe new history
8    unsigned int New_x0HistoryStep = New_HistorySize*pow(N,dimension);
           //Calculate size of the new stepping factors
9    unsigned int New_y0Step = pow(N,dimension);     //Calculate size of the
       new stepping factors
10
11   NewVal.resize(2*New_HistoryArraySize);
12
13   int ShiftParam = (int)(t-New_HistorySize)>0?t-New_HistorySize:
       mShiftParameter;  //Determine the shifting parameter for the new
       history size
14
15   //--------Transfering the remaining data from the old history to the
       temporary storage (NewVal)------
16   for(unsigned int x0 = ShiftParam;x0 <= t;x0++){
17     for(unsigned  int y0 = ShiftParam;y0<=x0;y0++){
18       for(unsigned int p = 0; p < y0Step; p++){
19         unsigned int oldIndex = calcCycleHomIndex(x0,y0,p);
20         unsigned int newIndex = calcArbitraryHistoryIndex(x0,y0,p,
             New_HistorySize,New_x0HistoryStep,New_y0Step);
21
22         NewVal.at(newIndex) = this->val(oldIndex);
23         NewVal.at(newIndex+New_HistoryArraySize) = this->specval(
             oldIndex);
24       }
25     }
26   }
27   //-------------------------------------------------------------------------
28
```

```
29    //---Adjusting the parameters of the old history to fit the new one
          ----
30    mHistorySize = New_HistorySize;
31    mHistoryArraySize = New_HistoryArraySize;
32    x0HistoryStep = New_x0HistoryStep;
33    size=2*New_HistoryArraySize;
34    dynsize=2*x0HistoryStep;
35    //-------------------------------------------------------------------------


36
37    values=NewVal;  //replacing the old storage with the new one
38
39    setShiftParameter(ShiftParam);  //Saving the shifting parameter t-
          shift to account for the systems new history
40    return;
41  }
```

Listing 5: Code of the *resizeHistory*-routine, which changes the current history size of a system to a new one and transfere the data of the former to the latter

At first a new storage *NewVal* for the data of the old history, that has to be kept for the new one, is created. This storage is resized accordingly to fit the new history. Afterwards, the shift parameter t'_{shift} of the new history is determined and the necessary values of the old history are transfered to the *NewVal*. This is done using a spacial kind of cycling function *calcArbitraryHistoryIndex*, which is shown below. This cycling function allows the determination of the correct line vector index for an externally input history. Note that this function is in principle equivalent to *calcCycleHomIndex*. However, because of its generality, its performance is comparatively slow, which is why it is only used in this case.

```
1  unsigned int Npointfunc::calcArbitraryHistoryIndex(unsigned int x0,
       unsigned int y0, unsigned int p, unsigned int HistorySize,unsigned
       int x0HistoryStep, unsigned int y0Step)
2  {
3    return x0HistoryStep*(x0%HistorySize ) + y0Step*(y0%HistorySize) + p;
4  }
```

Listing 6: Code of cycling function for an arbitrary history, that is parsed as an input, for a 2-point function and a homogeneous system

After the transfer of the data is finished, the settings of the old history, which are stored inside the *Npointfunc*-class for faster performance, are overwritten by the new settings. This settings overwrite has to be done after the transfer of the data, otherwise the

calcCycleHomIndex-function returns the wrong values during the data transfer. When all the settings are adjusted, the old history storage is replaced by the new one and the new time shift t'_{shift} is applied. The system now has a new history *New_HistorySize* and the computation can continue.

D. Graphical user interface

Besides our main work on the implementation of the history cut-off reduction scheme, we also worked on a graphical user interface (GUI), which uses the Qt framework for c++ [22]. The use of a GUI has various advantages for the usability of our solver. In the following, we'll give an overview of the graphical user interface and discuss its pros and cons. We want to emphasize, that, at present time, the GUI is only a basic one. A lot of functions need to be added and the design could also be improved further in the future. Nonetheless, the basic features that the GUI provides already simplify the use of our solver by quite a lot.

The central part of the GUI is the *MainWindow*, which initially is made up of the three tabs *General Parameters*, *Model Parameters* and *Plots*. In the *General Parameters* (see Fig. D.10) the user can choose all the currently possible parameters, which are not related to the specific model that is observed. Amongst other things, these parameters include the total observation time, the size of the lattice, the size of the fixed history matrix (for a full calculation one can either choose $t_H \geq t_{\text{total}}$ or $t_H = 0$), or if for example memory integrals should be taken into account.

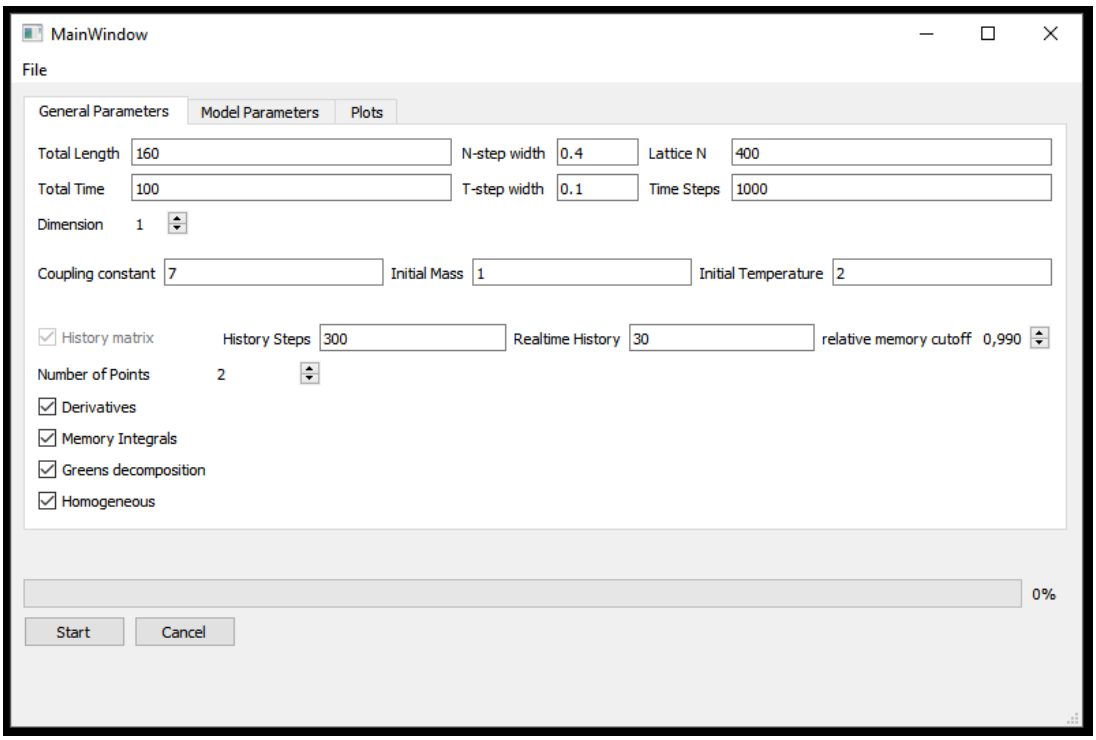


Figure D.10: Design of the *General Parameters* tab of our GUI

In the second tab, *Model Parameters* (see Fig. D.11), the user can choose settings, which are specific to the implemented model, or in the future, change the model itself. For the

homogeneous case, there is only one model available at present time. For this reason there is currently no option to switch between different models. However, using the drop down menu, one can choose between the different initial conditions, which currently include the tsunami and the quench. Additionally, for the tsunami initial condition, the user can choose position, width and amplitude of the tsunami peak and is, in the 1-dimensional case, directly provided with a plot of the initial particle distribution $n_0(p)$ as a function of the momentum p.

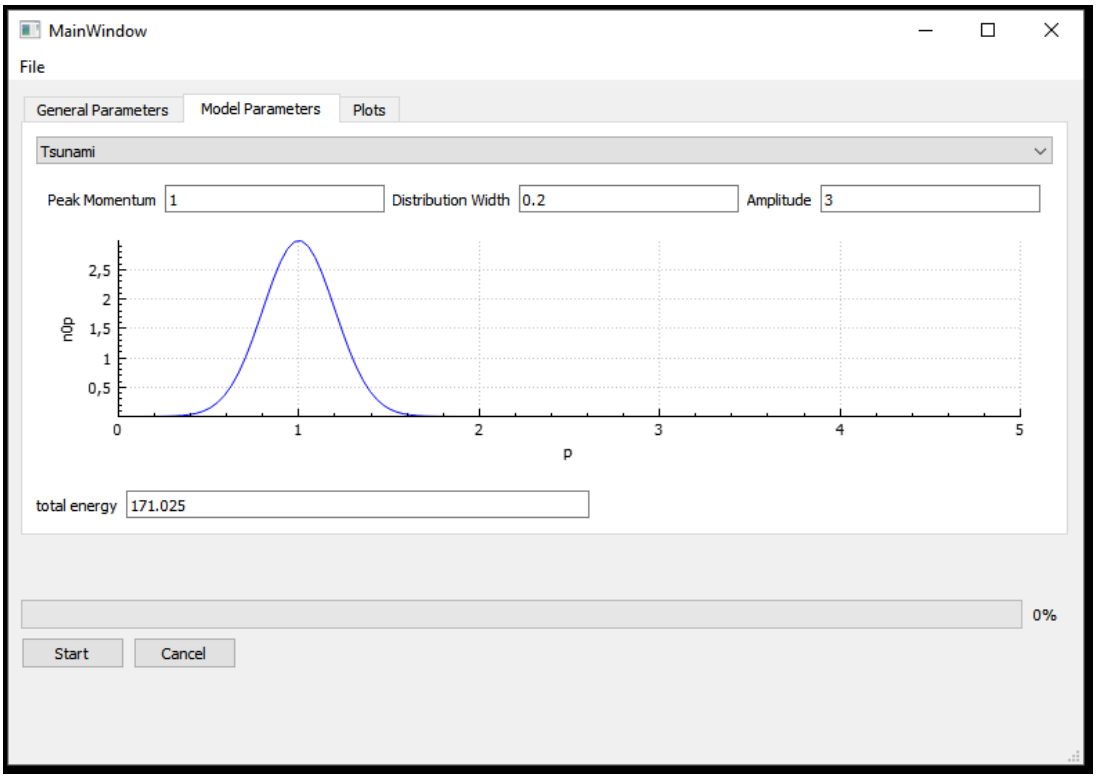


Figure D.11: Design of the *Model Parameters* tab of our GUI

In the *Plots* tab (see Fig. D.12), one can pick from a list of quantities, which should be plotted live during the evolution of the system. The plots are selected by marking the corresponding check box. Evolutions for quantities which are not on the list first have to be implemented inside the source code.

At the top of the screen, under the *file* menu, the user can choose to save his current settings inside a text file, or load the settings from an already existing file. At the bottom of the screen, the user has the option to start the run, or cancel the current one. Additionally, the user is further provided with a progress bar, which indicates the percentage of the total time steps that have already been calculated.

After all settings have been specified, the user can begin the run by pressing the start button. The GUI then starts to allocate the memory, which is necessary for the computation

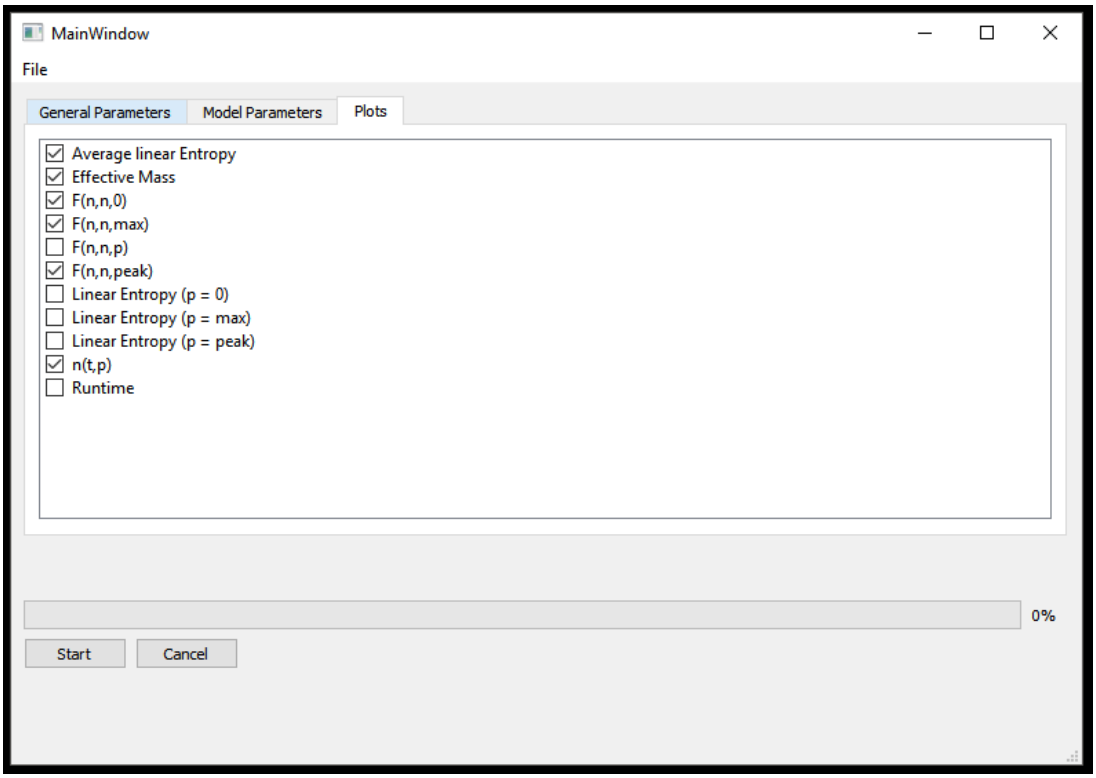


Figure D.12: Design of the *Plots* tab of our GUI

and spawns a solver object, that performs the computation. At the same time, additional tabs appear, each of them containing one of the plots, which have been chosen previously in the *Plots* tab (see Fig. D.13). While the solver performs the numerical solution of the Kadanoff-Baym equations, these plots a filled with live data of the calculations. By right clicking them, each plot provides the possibility to save its data, either as a simple picture file (jpeg or png), or as a text file (txt), where all the data of the plot is stored. All plots in the GUI are created using the *QCustomPlot* package [23].

The usage of the GUI has several advantages. First, the user does not have to directly access the code for changes, as it has been the case for previous builds. This allows even users without the knowledge of the actual code to perform computations using our solver. Additionally and perhaps most importantly, the direct feedback of the produced data through the live plots enables the user to directly observe the time evolution of the system. This in turn can for example help to find errors, or cancel runs with an unsatisfactory choice of parameters before completely finishing them and therefore avoiding the waste of a lot of time.

However, the GUI also has disadvantages. First of all, the GUI tends to generally slow down the computation. From our experience, depending on the data that is plotted during the run, the computation speed can be slowed down by roughly 25% compared to

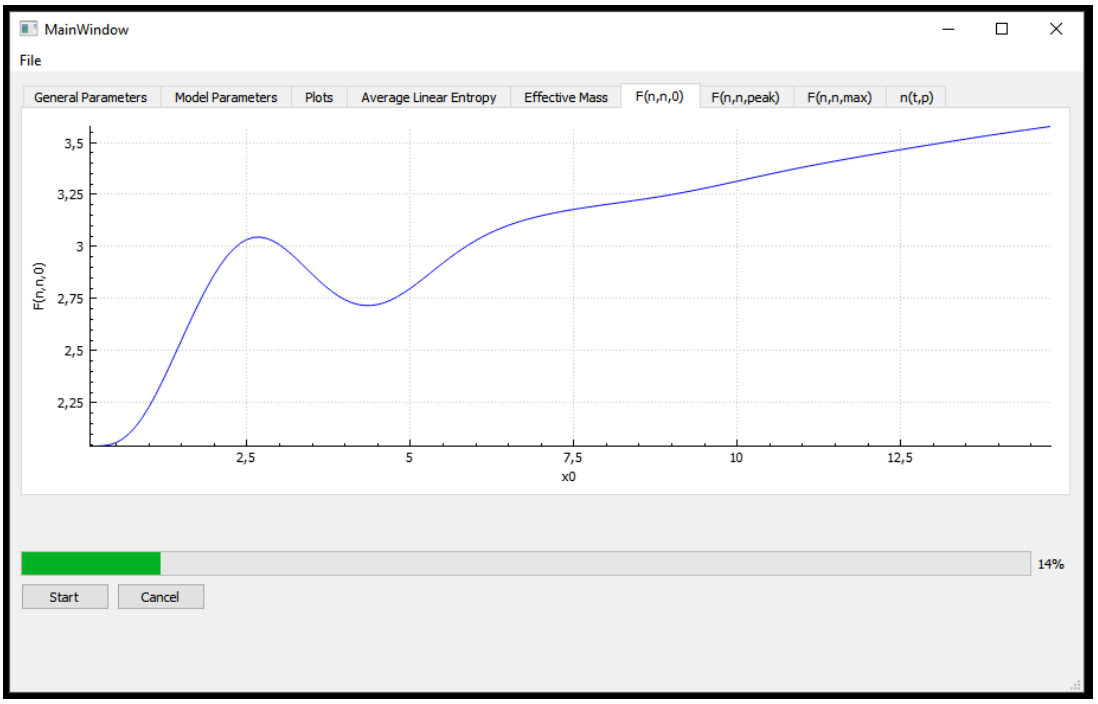


Figure D.13: Plot tabs (the active one is the plot that shows the equal time correlation function $F(x^0, x^0, 0)$) that appear at the start of the computation when they are chosen in the *Plots* tab. The plots show the live evolution of the data that is produced by the solver.

a computation that is now using the GUI. The second disadvantage is less a problem of the GUI itself, then it is one of the system where it is ran on. What we mean by this is that one typically wants to perform complex computations like solving the Kadanoff-Baym equations on a high performance cluster like the Julia HPC, which we use. The Julia HPC does not support graphical user interfaces, so the GUI-version of the program can't be run on the high performance cluster and the GUI has to be deactivated.

There are still a lot of possible ways in which the GUI can be improved in the future. For example, we could implement the option to pause and resume a calculation that is currently running. Nonetheless, in our opinion, even in its current state, the advantages of the GUI outweigh the disadvantage of the increased computation time. The significantly greater user friendliness, as well as the direct feedback from the live plots, make a great addition to our solver and enable its use to a broader audience.

References

[1] C. Bambi and A. D. Dolgov, "Antimatter in the Milky Way," *Nucl. Phys.*, vol. B784, pp. 132–150, 2007. doi:10.1016/j.nuclphysb.2007.06.010 [astro-ph/0702350].

[2] J. M. Cline, "Baryogenesis," in *Les Houches Summer School - Session 86: Particle Physics and Cosmology: The Fabric of Spacetime Les Houches, France, July 31-August 25, 2006*, MacGill University, Montreal, 2006. [arXiv:hep-ph/060914 [hep-ph]].

[3] M. Garny and M. M. Müller, "Quantum boltzmann equations in the early universe," in *High Performance Computing in Science and Engineering, Garching/Munich 2009* (S. Wagner, M. Steinmetz, A. Bode, and M. M. Müller, eds.), (Berlin, Heidelberg), pp. 463–474, Springer Berlin Heidelberg, 2010.

[4] M. Schüler, E. Martin, and W. Philipp, "Truncating the memory time in nonequilibrium DMFT calculations," *Phys. Rev. B 97, 245129*, 2018. doi:10.1103/PhysRevB.97.245129.

[5] A. Giraud and J. Serreau, "Decoherence and thermalization of a pure quantum state in quantum field theory," *Phys. Rev. Lett.*, vol. 104, p. 230405, 2010. doi:10.1103/PhysRevLett.104.230405 [arXiv:0910.2570 [hep-ph]].

[6] D. Campo and R. Parentani, "Decoherence and entropy of primordial fluctuations. i. formalism and interpretation," *Phys. Rev. D*, vol. 78, p. 065044, Sep 2008. doi:10.1103/PhysRevD.78.065044.

[7] D. Campo and R. Parentani, "Decoherence and entropy of primordial fluctuations. ii. the entropy budget," *Phys. Rev. D*, vol. 78, p. 065045, Sep 2008. doi:10.1103/PhysRevD.78.065045.

[8] J. Rammer, *Quantum field theory of non-equilibrium states*. Cambridge: Cambridge Univ. Press, first ed., 2007.

[9] D. Glavan and T. Prokopec, "Lecture notes: A pedestrian introduction to non-equilibrium qft," October,9 2017. URL: `https://www.staff.science.uu.nl/~proko101/LecturenotesNSTP530M2017oct06.pdf`. Last visited on 01.12.18.

[10] J. Berges, *Nonequilibrium Quantum Fields: From Cold Atoms to Cosmology*. University of Heidelberg, March, 10 2015. [arXiv:1503.02907 [hep-ph]].

[11] L. P. Kadanoff and G. A. Baym, *Quantum Statistical Mechanics*. Cambridge, Mass: Perseus Books, 2009.

[12] J. F. Koksma, T. Prokopec, and M. G. Schmidt, "Decoherence and dynamical entropy generation in quantum field theory," *Phys. Lett.*, vol. B707, pp. 315–318, 2012. doi:10.1016/j.physletb.2011.12.049 [arXiv:1101.5323 [quant-ph]], eprint =.

[13] R. Balian, "Incomplete descriptions and relevant entropies," *American Journal of Physics*, vol. 67, pp. 1078–1090, Dec. 1999. doi:10.1119/1.19086.

[14] G. Adesso, *Entanglement of Gaussian states*. PhD thesis, Salerno U., 2007. [arXiv:quant-ph/0702069].

[15] J. Berges, "Controlled nonperturbative dynamics of quantum fields out-of-equilibrium," *Nucl. Phys.*, vol. A699, pp. 847–886, 2002. doi:10.1016/S0375-9474(01)01295-7 [hep-ph/0105311].

[16] J. G. Steven and F. Matteo, *Fastest Fourier Transform in the West (FFTW), Version 3.3.5*, 2016. `http://www.fftw.org/`.

[17] W. Prof. Kahan, "Lecture notes: Ieee standard 754 for binary floating-point arithmetic," October,1 1997. URL: `https://people.eecs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF`. Last visited on 17.12.18.

[18] H. T. U. of Würzburg, *Julia HPC documentation*, Accessed: December, 8 2018. `http://doku.hpc.uni-wuerzburg.de/`.

[19] J. Berges, "Introduction to nonequilibrium quantum field theory," *AIP Conf. Proc.*, vol. 739, pp. 3–62, 2005. doi: 10.1063/1.1843591 [hep-ph/0409233].

[20] O. J. Peter, *Introduction to Partial Differential Equations*. Springer, Cham, 2014. doi:10.1007/978-3-319-02099-0.

[21] E. Azar and M. Alebicto, *Swift Data Structure and Algorithms*. UK, Birmingham: Packt Publishing, first ed., November, 18 2016. [ISBN:9781785884504].

[22] Q. D. Frameworks, *Qt, version 5.11.0*, May, 22 2018. `https://www.qt.io/`.

[23] E. Eichhammer, *QCustomPlot, version 2.0.0*, September, 4 2017. `https://www.qcustomplot.com/index.php/introduction`.

Erklärung

Hiermit versichere ich, dass ich meine Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum: .....

(Unterschrift)