Project 2

Quantum Optics Group 3
Hayden McGuinness, Anthony Clark, Anika Pflanzer

16 May 2007

1 Problem 1: Simulation of Quantum Diffusion

1.1 Theory

The Schrodinger equation for a two level atom driven by a Weiner process, which simulates homodyne detector of the X_1 quadrature, is given by

$$d|\Psi\rangle = -\frac{i}{\hbar}H|\Psi\rangle dt - \frac{\Gamma}{2}\left[\sigma^\dagger\sigma - \langle\sigma+\sigma^\dagger\rangle + \frac{1}{4}\langle\sigma+\sigma^\dagger\rangle^2\right]|\Psi\rangle dt + \sqrt{\Gamma}\left[\sigma - \frac{1}{2}\langle\sigma+\sigma^\dagger\rangle\right]|\Psi\rangle dW. \tag{1}$$

The equivalent master equation can be found by noting that, to second order, the equation for the density matrix can be written as

$$d\rho = d(|\Psi\rangle\langle\Psi|) = d(|\Psi\rangle)\langle\Psi| + |\Psi\rangle d(\langle\Psi|) + d(|\Psi\rangle)d(\langle\Psi|). \tag{2}$$

The compontents of $d\rho$, namely $d|(\Psi\rangle)$ and $d(\langle\Psi|)$ are either given by equation (1) or the conjugate of (1). The $d(|\Psi\rangle)d(\langle\Psi|)$ term of (2) simplifies greatly due to the fact that dt^2 is taken to be zero while from Ito's rule $dW^2 = dt$. Therefore

$$\begin{aligned} d(|\Psi\rangle)d(\langle\Psi|) &= \Gamma\left[\sigma - \tfrac{1}{2}\langle\sigma+\sigma^\dagger\rangle\right]|\Psi\rangle\langle\Psi|\left[\sigma^\dagger - \tfrac{1}{2}\langle\sigma+\sigma^\dagger\rangle\right]dt \\ &= \Gamma\left[\sigma\rho\sigma^\dagger - \tfrac{1}{2}\sigma\langle\sigma+\sigma^\dagger\rangle - \tfrac{1}{2}\langle\sigma+\sigma^\dagger\rangle\sigma^\dagger + -\tfrac{1}{4}\langle\sigma+\sigma^\dagger\rangle^2\right]dt \end{aligned} \tag{3}$$

Adding this term to to the $d(|\Psi\rangle)\langle\Psi|$ and $|\Psi\rangle d(\langle\Psi|)$ terms (straight forward algebra that won't be shown here) yields the result

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \Gamma D[\sigma]\rho + \sqrt{\Gamma}HH[\sigma]\rho dW \tag{4}$$

where

$$D[\sigma]\rho = \sigma\rho\sigma^\dagger - \frac{1}{2}(\sigma^\dagger\sigma\rho + \rho\sigma^\dagger\sigma) \tag{5}$$

and

$$HH[\sigma]\rho = \sigma\rho + \rho\sigma^\dagger - \langle\sigma + \sigma^\dagger\rangle\rho. \tag{6}$$

If $\sigma \to \sigma e^{-i\phi}$ then the equation for the density matrix would change accordingly, which leads to the expression

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + \Gamma D[\sigma e^{-i\phi}]\rho + \sqrt{\Gamma}HH[\sigma e^{-i\phi}]\rho dW \tag{7}$$

1.2 Data and results

The simulations were made for different time steps and a varying number of trajectories. The code can be found on the link on our webpage or under */projects/project2/problem1*. The files that have actually been changed are *sdesample.f90* and *sderk support.f90*. In *sderk support.f90*, we inserted our differential equations in component form, while we changed the input, the output , the stepnumber and the stepsize. The parameters Ω, $\Delta$and Γ are also defined in this file. The outputfiles are *ntraj1.txt, ntraj20.txt, ntraj100.txt, ntraj1000.txt* where each output is 5 collumns wide contatining time, numerical excited state population, total population, excited state population of the corresponding Bloch equation and difference between the exact and the analytic solution in this order.
To judge how accurate numerical solutions to equation (1) are compared to available analytic solutions. There is no analytic solution to this differential equation, but for an average over a large number of trajectories the solution should converge to the one for the optical Bloch equation. This behaviour will be investigated in the following.

The system is driven exactly on resonance (Δ=0), so we expect the excited state population to converge to

$$\rho_{ee} = \frac{\Omega^2}{2\Omega^2 + \Gamma^2}(1 - e^{-\frac{3}{4}t}(\cos(\Omega_\Gamma t) + \frac{3}{4\Omega_\Gamma}\sin(\Omega_\Gamma t). \tag{8}$$

for a large number of trajetcories.

For the numerical solution initially all population was in the ground state, Ω was set to 10, a time step size of .01 (where both quantities are normalized by Γ, set to 1.0 here) was used, and the total time of evolution was 3s.

First, the step size was varied to check whether we were in the right regime: a small variation of the stepsize shouldn't make a big difference. This was done for our chosen step size .0.01, the results for tstep=0.005 match those of .01 within the desired precision, so our stepsize is of the right magnitude.

For a large number of trajectories, the numerical result should converge to the analytic solution of the optical bloch equations. To study the dependence of the convergence on the number of trajectories, the program was varied such that it averaged over a given number of trajectories, defined by the user, after each time step. The number of trajectories was chosen as 1, 20, 100 and 10000.

The convergence of the numerical result is well visible in the plots: with an increasing number of trajectories, the numerical result gets closer to the analytic solution of the OBEs. The graphs correspond to our expectations: we can see damped Rabi oscillations with noise that is due to the stochastic Wiener process. The comparison to the exact analytic solution to the optical bloch equations shows the convergence of the error for increasing number of trajectories.

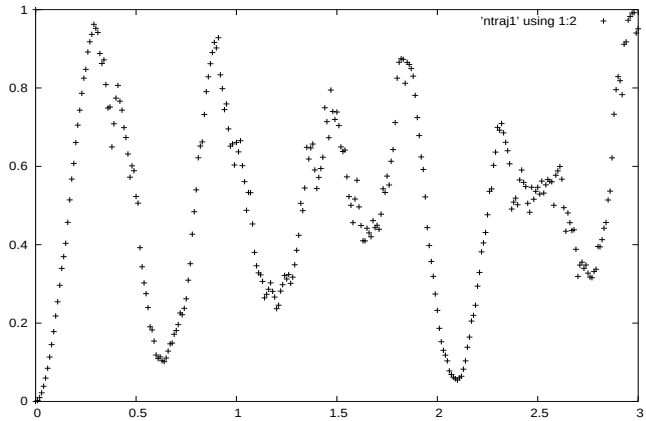

Figure 1: Excited state population for one trajectory, $\Omega = 10$, $\Delta = 0$

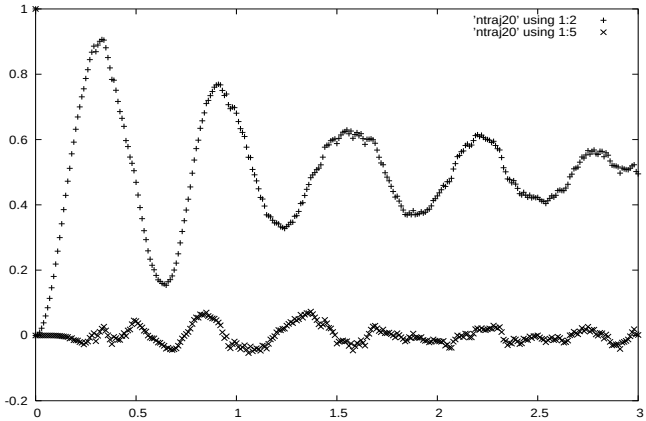

Figure 2: Excited state population for 20 trajectories, $\Omega = 10$, $\Delta = 0$, the upper line is the excited state solution, while the lower line is the difference between the OBE solution and the numerical result

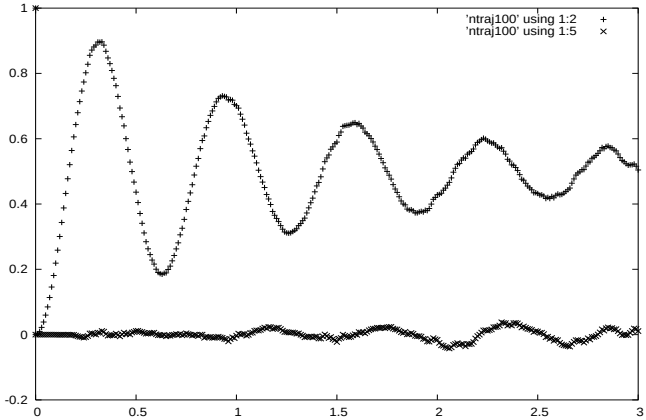

Figure 3: Excited state population for 100 trajectories, $\Omega = 10$, $\Delta = 0$, the upper line is the excited state solution, while the lower line is the difference between the OBE solution and the numerical result

1.3 Numerical Error

In the previous part of the problem, we studied the convergence of the numerical solution to the OBEs for a varying number of trajectories. Now, the number of trajectories will be kept one and we want to investigate the error dependence on time and stepsize. To find out about the error convergence, we consider the function

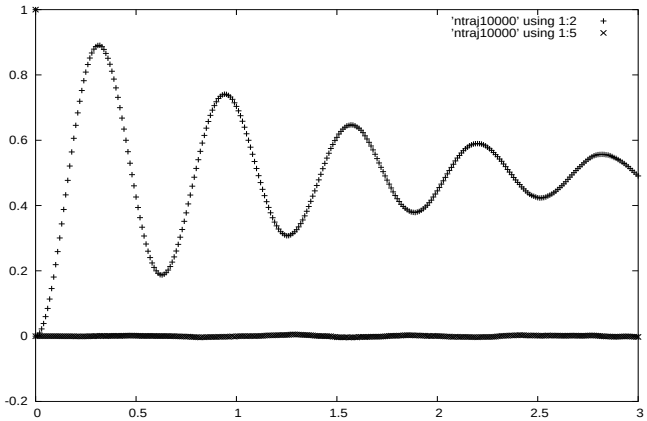

Figure 4: Excited state population for 10000 trajectories, $\Omega = 10$, $\Delta = 0$, the upper line is the excited state solution, while the lower line is the difference between the OBE solution and the numerical result

$$\tilde{f}(t) - f(t) = e(t, \Delta t) = e_\alpha(t)(\Delta t)^\alpha + O(\Delta t^{\alpha=0.5}), \tag{9}$$

where $\tilde{f}(t)$ is the numerical result to an SDE solver and f(t) is the exact solution to the corresponding differential equation. In order to see the convergence of the error in dependence on the stepsize, we need a large numer of data at different timesteps. The command runscript is used to obtain this. It changes the number of substeps for each integration from 1 up to 512 and puts the data in 'outn'. The collumns in the output files are again t, ρ_{ee} analytic, total population, ρ_{ee} exact and difference between analytic and exact result. The input was again $\Delta = 0$, Γ=1, Ω=10, while the stepsize was set to .01 and the number of steps to 300. then the number of substeps was raised by the command runscript from 1 to 512. These files are read with an octave command called testscript, that takes the difference between the SDE integrator solution and the exact solution (collumns 2 and 4) and divides it by the analytic solution. The graph shows error over stepsize, and it is visible that the error rises quickly after going through a minimum at a stepsize of 0.01. Our solution wil therefore be the closest to the exact solution for a stepsize around .01, which was also our initial choice. This result is reasonable as for a larger stepsize we expect a less exact result in general. The line is the best fit which does notconnect our data points very well as their behaviour is not linear. This is due to the lack of an analytic result to this differential equation; we cannot see the expected to the order of 1.5 behaviour of the error.

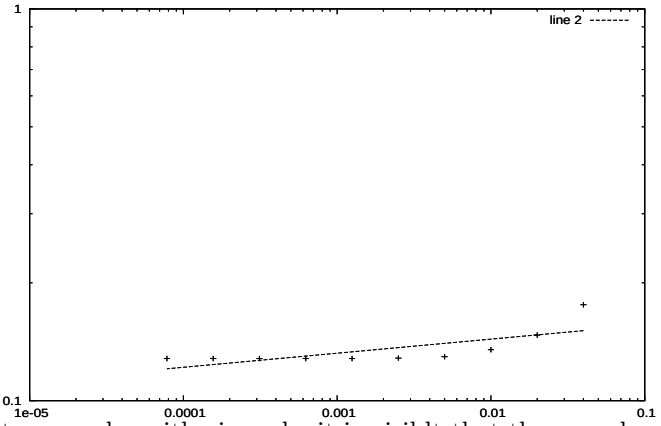

Figure 5: Error over timestep on a logarithmic scale, it is visiblt that the error has a dip slightly before .01 and inceases fast after that, the crosses are the measurement points while the line is the best fit

To see that the integrator works fine, we check the testscript command on the original code. In this plot the behaviour proportional to $\Delta t^{1.5}$ becomes visible.

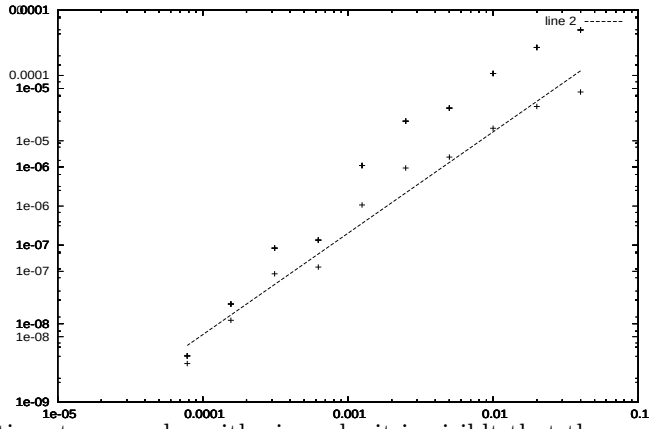

Figure 6: Error over timestep on a logarithmic scale, it is visiblt that the error scales with $\Delta t^{1.5}$

2 Problem 2: Numerical Evolution of a Quantum PDE

2.1 Introduction

In this problem we investigate the numerical solutions for the time evolution of a Gaussian wavepacket placed in a quantum harmonic oscillator, and compare these to the analytically obtainable results.

2.2 Split Operator Method

The split operator method for solving the Schrodinger equation is a means for time evolving an initial wavefunction for a known system Hamiltonian. In the case of the Harmonic oscilator with unit mass, the Hamiltonian,

$$H = \frac{p^2}{2} + \frac{1}{2}\omega^2 x^2 = T(p) + V(x), \tag{10}$$

contains no explicit time dependence, and so the time evolution operator is given by

$$U(t_1, t_0) = e^{\frac{iH\Delta t}{\hbar}} \tag{11}$$

where $\Delta t = t_1 - t_0$. Now we make use of the fact that the kinetic term of the Hamiltonian has only momentum dependence and the potential term has only position dependence to rewrite this as a product of exponential operators. Mathematically we want Q_n that solve the equation:

$$e^{T+V} = \prod_n e^{Q_n} \tag{12}$$

The Q_n that solve this are not unique, in fact it turns out that there are infinitely many possible decompositions. The first decomposition we consider is the tbree-term decomposition

$$e^{\lambda(A+B)} = e^{\lambda A} e^{\lambda B} e^{-Z}, \tag{13}$$

Expressing the exponential operators on both sides by their Taylor series expansions and expanding the operator Z in powers of λ, it is straightforward to equate terms of like order in λ from both sides to make the decomposition valid for any λ.

$$\sum_{n=1}^{\infty} \frac{\lambda^n (T+V)^n}{n!} = \sum_{m,l,q=0}^{\infty} \frac{\lambda^{m+l} A^m B^l Z^q}{m!l!q!} \tag{14}$$

$$Z = \sum_n \lambda^n Z_n \tag{15}$$

$$\begin{aligned}
&\lambda^0: && 1 = 1 - Z_0 && \rightarrow && Z_0 = 0 \\
&\lambda^1: && A + B = A + B - Z_1 && \rightarrow && Z_1 = 0 \\
&\lambda^2: && \tfrac{1}{2}(A^2 + AB + BA + B^2) = \tfrac{A^2}{2} + \tfrac{B^2}{2} + AB - Z_2 && \rightarrow && Z_2 = [A, B]
\end{aligned} \tag{16}$$

The lowest order nonzero Z_n we get is $Z_2 = [A, B]$, so we see that the lowest nontrivial order of λ in the series expansion of e^{-Z} will be λ^2. Then letting $A \rightarrow T, B \rightarrow V, and \lambda \rightarrow \Delta t$ gives the decomposition for the time evolution operator:

$$e^{\lambda(T+V)} = e^{\lambda T} e^{\lambda V} + \mathcal{O}((\Delta t)^2), \tag{17}$$

Neglecting the e^{-Z} operator introduces error of order $\mathcal{O}(\Delta t)^3$.

To reduce errors to $\mathcal{O}(\Delta t)^3$, we can use a decomposition of form:

$$e^{\lambda(A+B)} = e^{\lambda \frac{B}{2}} e^{\lambda A} e^{\lambda \frac{B}{2}} e^{-Z}, \tag{18}$$

Writing the exponentials on both sides in terms of their Taylor series expansions, expanding Z as above and equating powers of λ gives,

$$\begin{aligned}
&\lambda^0: && 1 = 1 - Z_0 && \rightarrow && Z_0 = 0 \\
&\lambda^1: && A + B = A + B - Z_1 && \rightarrow && Z_1 = 0 \\
&\lambda^2: && \tfrac{1}{2}(A^2 + AB + BA + B^2) = \tfrac{1}{2}(A^2 + AB + BA + B^2) - Z_2 && \rightarrow && Z_2 = 0 \\
&\lambda^3: && \tfrac{1}{6}(A^3 + A^2 + ABA + AB^2 + BA^2 + BAB + B^2A + B^3) \\
& && = \tfrac{A^3}{6} + \tfrac{A^2B}{8} + \tfrac{AB^2}{4} + \tfrac{B^2A}{4} + \tfrac{BA^2}{8} + \tfrac{ABA}{8} + \tfrac{B^2}{6} - Z_3 \\
& && \rightarrow Z_3 = \tfrac{-A^2B}{24} + \tfrac{AB^2}{12} + \tfrac{B^2A}{12} - \tfrac{BA^2}{24} + \tfrac{ABA}{12} + \tfrac{BAB}{6}
\end{aligned} \tag{19}$$

so we see that the error in ignoring the e^{-Z} operator is $\mathcal{O}((\Delta t)^2)$. It is apparrent that this decompositions yielding higher order error terms can be found by using further symetrized products of exponential operators. The higher order splittings will require smaller time steps for equivilent accuracy, thus lowering the computational resources required, but will take more resources to carry out the additional exponential operators required to carry out each time step. The optimal splitting/timestep combination is therefore found by balancing out these two factors.

The Schrodinger equation for a particle in a potential well was numerically evolved for different step sizes and for orders two, four and six in the Richardson expansion. The initial wavepacket in the position basis was a Gaussian with the form

$$\psi = \frac{1}{\sqrt{\sqrt{2\pi}\sigma_x}} e^{(-(x-x0)^2/(4\sigma_x^2))} \tag{20}$$

where $x0 = 0.3$ and $\sigma_x = 0.4$. There were 128 grid points on which the value of the wavepacket was calculated. The dimensionless time steps in which the global error was calculated were 0.00125, 0.0025, 0.005 and 0.01 for order methods two, four and six.

Since the exact solution is know to be periodic with period $T = 2\pi/\omega$ (in these cases $\omega = 2\pi$ so $T = 1$) error was calculated by comparing the numerically obtained solution after 10 cycles to the initial conditions. Global error was calculated for several different parameters, expectation value of position, mean error of the wavepacket and for the single largest position value of the wavepacket (i.e. the error in the value of the "top" of the wavepacket, at $x = 0.3$).

The only change in the given code was a change in the precision of the output from eight digits to 13 digits. This was necessary due to the order six method being accurate beyond eight significant figures. Since this change is very minor the code will not be presented here. A simple MatLab script was designed to calculate errors and plot the results.

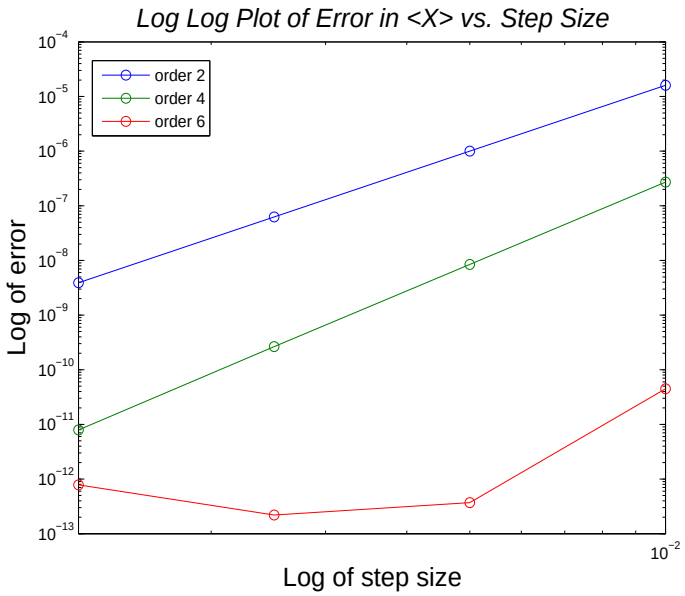

Figure 7: Log Log plot of error in the expectation value of position calculated as a function of step size for orders two (blue), four (green) and six (red).

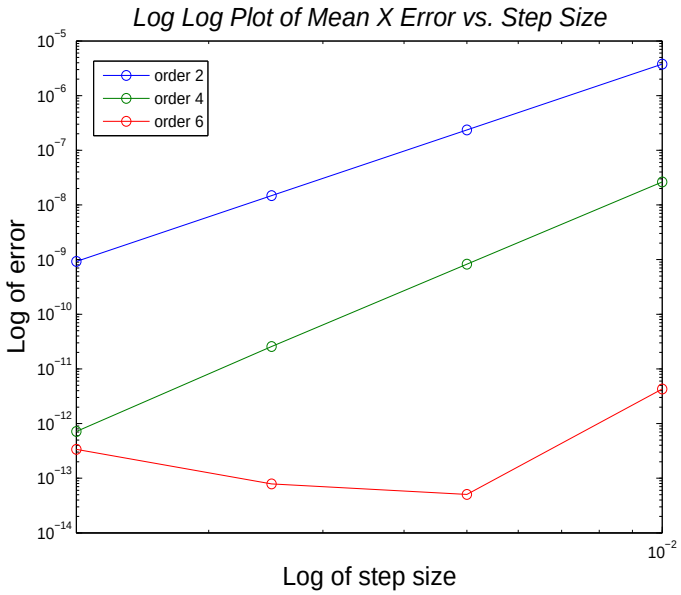

Figure 8: Log Log plot of error in the mean wavepacket value calculated as a function of step size for orders two (blue), four (green) and six (red).

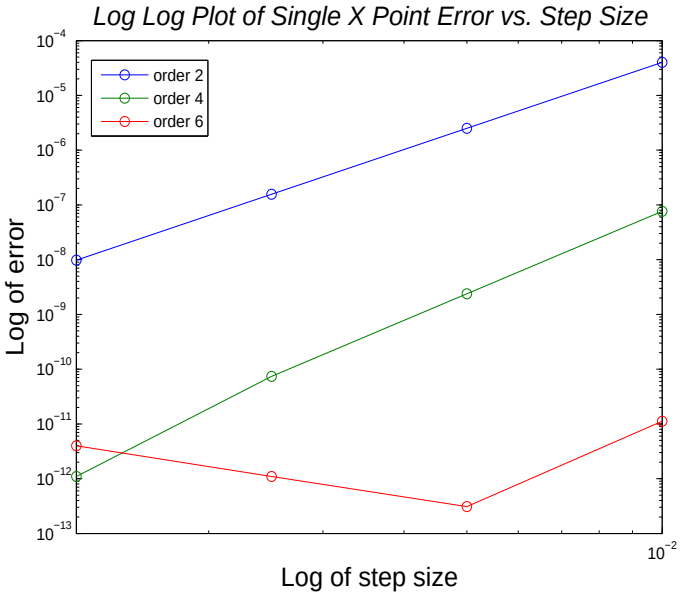

Figure 9: Log Log plot of error of a single position point calculated as a function of step size for orders two (blue), four (green) and six (red).

The error results are shown in loglog plot style for expectation value, mean wavepacket and single point in figures 7,8 and 9 respectively. As would be expected for relatively large times steps, orders two and four show a very linearly relationship between log(error) and log(step size). This suggests very strongly that error as a function of step size is given by a power law relation. For order six the error does not exhibit this relationship, but can be understood from the premise that there is an optimal step size where the error is

least. This is due to for "too small" of step sizes round off error becomes a significant factor while for "too large" of step sizes there is the more familiar error of the numerical approximation becoming less accurate. So for the appropriate range of step sizes this behavior is also expected.

What is not expected is the power of the power law behavior for orders two and four in the parameter errors. For order two the powers are 4.000398, 4.000315 and 4.000258 for global errors in expectation value, mean wavepacket and single point respectively. For order four the powers are 5.018858, 5.050588 and 5.324281[1] for global errors in expectation value, mean wavepacket and single point respectively. From theory, the expected powers should be two and four for orders two and four respectively, at least for one of the parameter errors.That is why three different parameter errors were calculated. It might be the case that one expects, for example, expectation value error to scale differently from that of mean wavepacket value or of single position value, but this seems not to be the case. Simple problems, such as not having enough data or not running the simulation for long enough (more than 10 cycles) seems inadequate due to the clear trend in the data. From our knowledge these results are not explainable.[2]

3 Problem3: Quantum feedback control of the motion of a quantum-mechanical particle

In this part of the project we model the feedback control of a particle in a one-dimensional harmonic oscillator where the position of the particle is continuously measured. The stochastic master equation for such a system is given by

$$d\rho = -\frac{i}{\hbar}[H,\rho]dt + 2k\mathcal{D}[x]\rho dt + \sqrt{2k}\mathcal{H}[x]\rho dW \tag{21}$$

with the Hamiltonian from problem 2 in the form

$$H = \frac{p^2}{2} + \frac{1}{2}\omega^2 x^2. \tag{22}$$

We use the easier stochastic Schrodinger equation projected into the position basis, so the evolution of the wave function ψ is described by

$$d|\psi> = -\frac{i}{\hbar}m|\psi > dt - k(x - \langle x\rangle)^2|\psi > dt + \sqrt{2k}(x - \langle x\rangle)|\psi > dW \tag{23}$$

in our program.

To implement the stochastic process into the solution of the partial differential equation, the second order operator splitting method implemented above was modified that the spatial parts of the time evolution were generated by the sderk routine. First, we used the SDE integartor to let the state evolve from t to t+Δt, using the ordinary differential equation obtained by discarding the momentum term in the Hamiltonian. Then the fft routine was used to take the fourier transform of the function, the split operator solver routine was used to apply the drift operator. Then the inverse fourier transform was taken and the time step was

[1] This value is significantly different from the others most likely due to the error of the smallest step size entering the region of "optimal" step size, where error due to numerical approximation becomes comparable to roundoff error instead of dominating the error.

[2] Unfortunately, we are running out of time on this problem set and must move on the more difficult third problem.

completed using the SDE integrator as before. This of course is then repeated until the input final time is reached.

The measurement process has the effect of heating the particle up. In order to counteract this heating, we want use the state information obtained from measurements, in particular the centroid position $\langle x \rangle$ at each time step, to change the potential the particle is in in such a way that the particle is slowed down. The parameter that we have control over in an such an experiment is the strength ω of the oscilator potential well. Noting that the process of measuring position tends to localize the particle in space, it is reasonable to rely primarily on the centroid position to build the feedback signal, as if the particle were classical, disregarding the energy due to the spatial spread of the particle's wavefunction. In this case to cool the atom we need to adjust ω such that it is small when $\langle x \rangle$ is large and large when $\langle x \rangle$ is small resulting in effective cooling of the particle. This method of switching ω high and small suddenly is called the "bang- bang" method. Its goal is to apply kicks to the particle such that its environmental interaction is disrupted. It maximizes the rate of energy extraction. In order to cool the particle we have to extract energy from the system at a higher rate than the system feeds it in.

To do this in a real homodyne experiment, we must first determine $\langle x \rangle$. The quantity which is actually measured is an electrical signal generated by a photodetector. This can be modeled as

$$dy(t) = \langle x \rangle \, dt + \frac{dW}{\sqrt{8k}} \tag{24}$$

where k is the measurement strength associated with the detector. This signal thus contains both information about the particle's centroid position $\langle x \rangle$ and quantum noise. To get the quantity we need therefore to filter out the noise. The quantum noise takes place on a much smaller timescale than the actual signal we want to detect, so we pass the photodetector signal through a lowpass filter. The voltage output from the low pass filter is described by the differential equation

$$\frac{\partial V}{\partial t} = -\gamma V + \gamma V_p h \tag{25}$$

where V is the output voltage, $V_p h$ is the input signal from the photodetector, and $\gamma = \frac{1}{RC}$. Solving this, it is straightforward to determine that the steady-state solution for each frequency component of the input signal is attenuated by a factor of $\sqrt{1 + (\frac{\omega}{\gamma})^2}$, and experiences a phase shift of $\arctan(\frac{\omega}{\gamma})$. Though the use of the low-pass filter is motivated by the attenuation of high frequencies, the phase shift can also be used advantageously, as the trap frequency to use for continuous feedback according to the description above could be taken to be the square of the oscilation of the position expectation value, phase shifted by $\frac{\pi}{2}$, and with some constant offset frequency added. Thus if we choose an appropriate time constant for the filter, we can accomplish both reduction of the noise and the needed phase shift.

To simulate the action of the low-pass filter with our computer model, we must numerically integrate equation (25). The easiest way to do this is to simply replace the derivative with a ratio of finite differences. Then the output signal at the $j + 1$ timestep is given in terms of the input and output signals at the j timestep by:

$$V_{j+1} = (1 - \gamma \Delta t) V_j + \gamma V_{phj} \tag{26}$$

This signal is then squared and fed back to reset the value of ω to

$$\omega = \omega_{off} + \omega_A V^2 \tag{27}$$

where ω_{off} is an offset frequency, and ω_A a scaling amplitude, which serve as tuning parameters to optimize the feedback.

3.1 Data and results

Unfortunately, we have been unable to simulate realistic cooling, although we have been able to show that cooling works if we use the knowlege of the position expectation value of the particle we have access to in our artificial simulation.

First we ran the program without any cooling, which means that the measurement strength k=0 and a frequency of $/omega = 2/pi$. We start with a Gaussian wave paket as an initial condition for the particle's wave function. The code for this can be found under **/users/p686g3/projects/project2/problem3/bang**. The result is as we expect–no cooling and measurements result in heating of the particle.

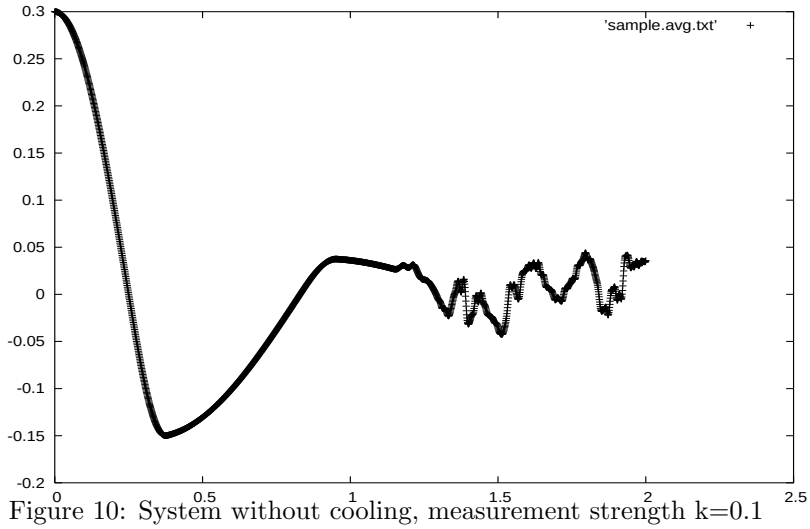

Figure 10: System without cooling, measurement strength k=0.1

Using a simple bang bang method where the initial frequency of $/omega = 2/pi$ is divided by 1.5 if $\langle x \rangle$ is high and gets multiplied by 1.5 if $\langle x \rangle$ is low lets us observe cooling. The measurement strength was set to k=0.02 and k=0.5 . This code can also be found under **/users/p686g3/projects/project2/problem3/bang**. Plots for $\langle x \rangle$ for two values of the measurement strength follow.

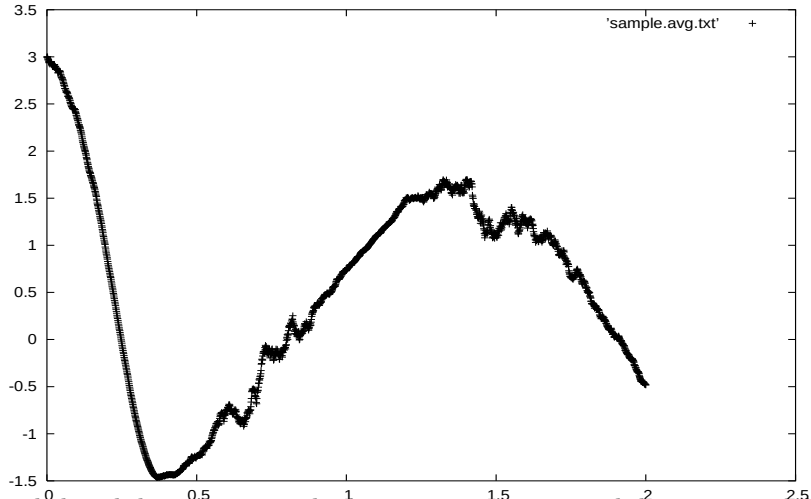

Figure 11: System cooled with bang bang method, measurement strength k=0.5, initial frequency $\omega = 2\pi$

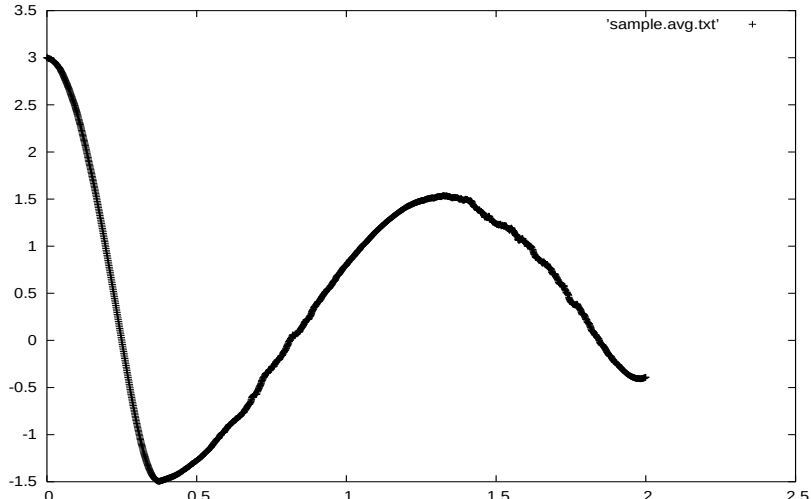

Figure 12: System cooled with bang bang method, measurement strength k=0.01, initial frequency $\omega = 2\pi$

A more direct way to accomplish cooling with our omniscient knowledge of the state is by directly integrating the position expectation value and feeding this back to the trap frequency. The plot below shows $\langle p^2 \rangle$ over time with such a cooling method used. The cooling method effectively prevents the peaks in the kinetic energy from increasing due to measurement heating as they would if no cooling were used. The code for this method is contained in the **/users/p686g3/projects/project2/problem3/snowflake** directory on atomoptics, although this mechanism is commented out in favor of an attempt at realistic cooling in the main integration loop in the file sch1d.f90.

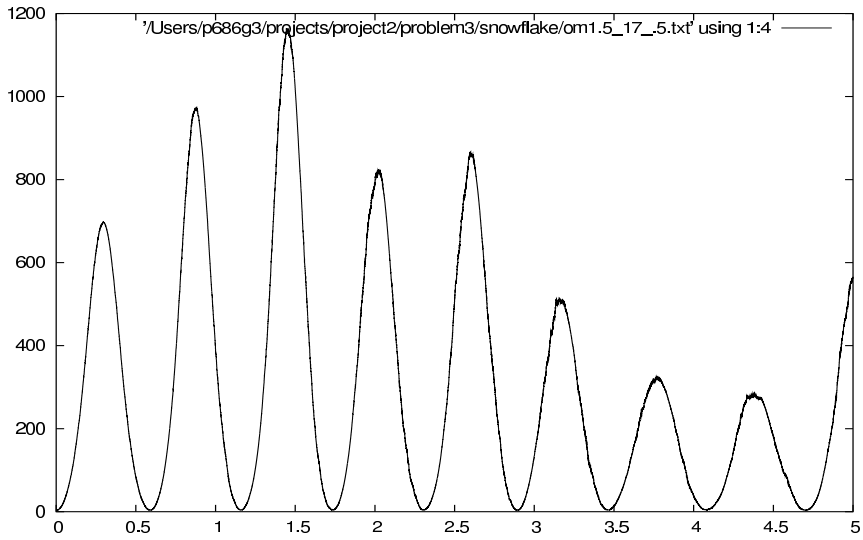

Figure 13: System cooled by feedback from integration of $\langle x \rangle$, measurement strength k=0.5, initial frequency $\omega = 2\pi$

We made attempts at employing the lowpass filter method to get rid of the noise, but were unable to get convincing results. These attempts can be found in **/users/p686g3/projects/project2/problem3/scheisse/bang** and as the currently uncommented feedback method in **/users/p686g3/projects/project2/problem3/snowflake**.