Abstract

A Bose Einstein condensate (BEC) is a gaseous superfluid form of matter created by cooling a collection of atoms to a temperature a few billionths of a degree above absolute zero. BEC have interesting properties in that they have the potential to bring the properties of quantum mechanics in to the macroscopic world.
My research focuses on deriving the numerical solution of the Gross-Pitaevskii equation, which governs BEC behavior. This project will model BEC properties such as the ground state and associated energy, as well as the dynamics of wave equations. The model will be programmed in FORTRAN coding and computed by the UNR Physics Departments NEON processor. This work can be used as a foundation for future research in atomic physics as a means to predict behavior of BECs in various environments without the cost, time, or labor of physical experimentation.

Contents

1 Introduction **4**

2 Literature Review **5**

3 Methodology **6**
3.1 The One Dimensional Schrodinger Equation 6
3.2 Scaled Coordinates 7
3.3 Finite Differencing- Discrete Representation 7
3.4 The Crank-Nicholson Method 8
3.5 Numerical Considerations to Matrix Algebra 9
3.6 The One Dimensional Gross-Pitaevskii Equation 9
3.7 The Radial Gross-Pitaevskii Equation 10
3.8 Propagation in Imaginary Time and the Heat Equation 11
3.9 Initial Condition and the Thomas-Fermi Approximation 12
3.10 Normalization 13
3.11 Expected Wave-functions 13
3.12 Expected Energies 15
3.13 Inter-atomic Interaction Coefficient 16

4 Results **16**
4.1 Propagation in Imaginary Time to Analytically Derived Ground State 16
4.2 First-order Approximation of the Energy in the Ground State of the GP Equation 17

5 Conclusion **18**

6 Appendix **25**

List of Figures

1 Propagation in imaginary time to the ground state of the harmonic oscillator in cartesian coordinates. 19
2 Ground state of a one dimensional BEC in cartesian coordinates with varying inter-atomic interactions. The confining potential is that of the harmonic oscillator. 20
3 Ground state of a two dimensional, axially symmetric BEC in scaled cylindrical coordinates with varying inter-atomic interactions. The confining potential is that of the harmonic oscillator. 21
4 Energy of BEC as a function of small values of the inter-atomic interaction coefficient (U). 22
5 Numerically derived ground state energies of the GP equation (blue) with varying inter-atomic interactions as well as the approximate energies (purple) given by first order perturbation theory. 23

Modeling of Bose-Einstein Condensates

Justin Luria

May 5, 2006

1 Introduction

[1]A Bose-Einstein Condensate is a gaseous superfluid formed by atoms cooled to temperatures very close to zero degrees kelvin. In other words, it is a collapse of several atoms (bosons) into a single quantum state. The resulting condensate is not a compact amalgamation of bosons but rather a group of atoms indistinguishable from each other. In this manner, Bose-Einstein Condensates can be thought of as a single macroscopic atom.

BEC states are allowed due to the special properties of bosons (particles with integral spin). Because indistinguishable bosons (bosons in close proximity) must be symmetric, this tends to push indistinguishable bosons together. This exchange force is not a force but a movement towards symmetry that causes Bosons to clump together. Symmetry also allows wave packets (particles) to attain the same quantum state without violating a normalization condition. Therefore, as theorized by Satyendra Nath Bose and Albert Einstein, if particles are slowed to a near stop inside of a vacuum they are statistically likely to achieve the same quantum state (and thus becoming a Bose-Einstein Condensate).

The first gaseous BEC was created by Eric Cornell and Carl Wieman in 1995 at the University of Colorado, Boulder. By cooling a dilute gas of Rubidium atoms to a mere 170 nanokelvins they were able to collapse the gas in to a gaseous superfluid. Since then Bose-Einstein Condensation has become an important field of research in quantum mechanics, optics, and frontier computing.

The thesis work presented in this paper is the result of FORTRAN programming that models the behavior of Bose-Einstein Condensates in the

[1]I would like to thank NSF-EPSCoR for funding this research. I would also like to thank my mentor, Dr. Andrei Derevianko, for his help and motivation in this work.

presence of a magnetic trap. This work can be used as a foundation for future research in atomic physics as a method to predict behavior of BECs in various environments without the cost or labor of physical experimentation.

2 Literature Review

The first attempt at creating a Bose-Einstein Condensate was out of hydrogen. Hydrogen atoms were cooled in a dilution refrigerator, confined by a magnetic trap, and then further cooled by evaporation. But this process was flawed in that the hydrogen atoms bonded to each other to form molecules, and thus avoiding the same quantum state.

Alkali metals are ideal for making BECs because they have a favorable internal energy-level structure for cooling to very low temperatures. By combining techniques of evaporative cooling and laser cooling, researchers were finally able to create a BEC out of Rubidium. [DGPS99]

The starting point in the theory of Bose-Einstein Condensation takes the form of Gross-Pitaevskii theory. Using a mean-field approach, Gross-Pitaevskii theory is able to model most two-body interactions in a BEC using relatively simple equations.

This reference [DGPS99] gives insight in to the workings of Bose Einstein Condensates and Gross-Pitaevskii theory . Using the information in this reference I can derive numerical formulas to evaluate the Hamiltonian and ground state energy of each wave function my program produces. Knowing the energy of a wave function is useful because it provides a means to check the validity of my answer and the convergence of wave functions. This reference, however, does not give any information regarding the means of numerically finding the solution to the Gross-Pitaevskii equation. It is a paper dealing with the basic concepts and experimentation fundamentals.
A study on inter-atomic interactions in Bose-Einstein Condensates [HJCC97] gives a brief summary of how to solve the Gross-Pitaevskii Equation in cylindrical coordinates. This study provides a method for solving some of the equations I shall be working with.

The reference [HJCC97] discusses some unit considerations such as normalization conditions, mean field as a function of a scattering length, scaled coordinates, and dynamic differencing. The mean field in the BEC is modeled by the nonlinear term, $NU|\psi(r,t)|^2\psi(r,t)$ where $U = 4\pi\hbar^2 a/m$ is a function of scattering length a, a quantity that can be determined in the lab. The coordinates that are used in the computational process are scaled by a factor $L = \sqrt{\hbar/(2m\omega)}$. This reference also suggests using a helper function $\phi = \sqrt{L}\rho\psi(\rho,\xi,t)$. Using this helper function, ϕ, the Hamiltonian becomes

more manageable to work with in cylindrical coordinates.

The main problem with working in cylindrical coordinates is the singularity at $\rho = 0$. Systems are generally unstable around $\rho = 0$ because the differential equations governing the behavior of the wave function blow up to large numbers. The authors of [HJCC97] proposed a dynamic differencing such that forward differencing is used near $\rho = 0$, shifting towards central differencing as ρ increases. By working out a series expansion in the hamiltonian, the rate which the shift occurs can be obtained. With this shifting factor, the numerical methods used to propagate the wave function will be more stable and allow for larger grid spacing (which in turn should allows for quicker computation).
Appendix C of the Ph.D. thesis [Wil99] focuses on the numerical solution of the Gross-Pitaevskii equation. The fundamentals of my numerical work followed this dissertation.

The discretization of the GP equation is necessary to do numerical work. It would be impractical to do this work analytically (working with functions), so the problem must be broken down in to matrices and arrays (of numbers). While the discretization is given in this paper, the details on the scaling of variables is vague. The author simply sets $\hbar = \omega = m = 1$ which is sufficient enough to discretize the GP equation, but not sufficient to give proper units to the wave function once it has been derived. This was acceptable for the type of work the author was doing, but not for this thesis work.

This reference [Wil99] was helpful in understanding the basics of my research. Through it I understood the mechanics which I later applied to different problems (such as the GP equation in cylindrical coordinates). However it was written in such a way that made it difficult to transfer the understanding of the material to different problems (different coordinate systems, potentials, etc). This thesis will focus on the problem in such a way as to give it more applicability.

3 Methodology

3.1 The One Dimensional Schrodinger Equation

In order to create complex programs, it is wise to start with an initial construct. I will begin by programming the one-dimensional Schrodinger equation in a harmonic potential. Not only does this provide insight in to the numerical method and programming, it provides a structure on which I can build on to create more complex programs. The Schrodinger equation governs wave equations with no inter-atomic interactions. It is a linear second-order

parabolic partial differential equation that can be written as [Gri05],

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t)\ . \tag{1}$$

3.2 Scaled Coordinates

As the equation is written, it would be difficult to process numerically. The constants in the equation are several powers of ten below zero, as the represent objects which are extremely small. It is not wise to process something with so many negative orders of magnitude because it requires a great deal of memory and precision to calculate such numbers. So instead of processing the equation as it stands we introduce a unitless scaled function, $\phi(\xi,\lambda)$ to replace $\psi(x,t)$, where $\lambda = t\omega$, $\xi = \dfrac{x}{\sqrt{\frac{\hbar}{m\omega}}}$, and ω is the harmonic oscillator frequency . With this replacement the new form of the Schrodinger equation that can be processed using numerical computation is:

$$i\frac{\partial\psi(\xi,\lambda)}{\partial\lambda} = -\frac{1}{2}\frac{\partial^2\phi(\xi,\lambda)}{\partial\xi^2} + V(\xi)\phi(\xi,\lambda). \tag{2}$$

Now the differential equation can be processed in an order of magnitude that the processor is capable of handling.

3.3 Finite Differencing- Discrete Representation

In order to process more information in a shorter period of time, I did my programming work in a basic computing language (FORTRAN). While it is possible for me to do programming in a higher language like MathCad (which will solve differential equations for me), it will process information slower. While FORTRAN can not solve any sort of equation analytically, it can perform simple algebraic operations. So it is necessary to put the scaled form of the Schrodinger equation in to an algebraic form. We do this by giving an index to $\phi(\xi,\lambda)$ such that $\phi_i^n = \phi(ih, n\tau)$ where i,n are indexes on ξ,λ with grid spacing h,τ respectively. Using Euler's differencing method, the discretized form of the Schrodinger equation is

$$i\frac{\phi_i^{n+1} - \phi_i^n}{\tau} = \left[-\frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{2h^2} + V(ih)\phi_i^n\right]\ . \tag{3}$$

3.4 The Crank-Nicholson Method

The Crank-Nicholson method is a technique used to solve parabolic partial differential equations by averaging implicit and explicit forms of the discrete representation of the differential equation [Ise96]. The first step is to solve the discrete representation of the Schrodinger equation for ϕ_i^{n+1}. In this form it is clear how ϕ_i^{n+1} relates to ϕ_i^n. The coefficient in front of ϕ_i^n is called the propagator and in this form can be written as the matrix equation

$$\overrightarrow{\phi^{n+1}} = (I - i\tau H)\,\overrightarrow{\phi^n}, \tag{4}$$

$$H = \left(-\frac{(\delta_{r+1,s} - 2\delta_{r,s} + \delta_{r-1,s})}{2h^2} + V(rh)\delta_{r,s} \right) . \tag{5}$$

This is called the explicit form of the discrete representation. For parabolic equations it is acceptable to approximate the propagator as the first two terms of the Taylor series representing $e^{-i\tau H}$. By bringing the propagator to the opposite side of Eq. (4) and converting it back in to the first two terms of the Taylor series, we arrive at the implicit form of the discrete representation.

$$(I + i\tau H)\,\overrightarrow{\phi^{n+1}} = \overrightarrow{\phi^n} . \tag{6}$$

There are problems with both the implicit and explicit forms in that they do not preserve the norm (scaling) of ϕ and are sensitive to the grid spacing τ. But these problems can be overcome by "averaging" the two forms. By finding the Padé approximate polynomial to e^{-iH}, we arrive at a solution that takes the properties of both implicity and explicit forms. This entire process is called the Crank-Nicholson method. Using the Crank-Nicholson method, the new form of the propagator becomes

$$\overrightarrow{\phi^{n+1}} = \left(I + \frac{i\tau H}{2} \right)^{-1} \left(I - \frac{i\tau H}{2} \right) \overrightarrow{\phi^n} . \tag{7}$$

It should be noted that as long as the Hamiltonian is hermitian, the propagator is unitary. That means that the norm of the wave function is preserved after each iteration in real time.

The Crank-Nicholson method as described in [Wil99] is in terms of matrices, which is appropriate for describing functions and not just values of a function for a given coordinate. My methodology differed slightly from this reference in that I actually programmed the inverse of a tridiagonal matrix instead of solving an equivalent matrix equation. The technique I used may be slightly less computationally efficient, but it is much simpler to work with.

3.5 Numerical Considerations to Matrix Algebra

The matrix algebra used to simplify the equations used in the Crank-Nicholson method can be very helpful. But it is important to consider the numerical consequences of using matrix algebra. The first aspect to note when applying formulas to matrix algebra is the indexing.

The matrix representing the propagator in Eq. (4) is a square matrix with rank equal to the number of points in the ϕ array. The Kronecker delta in H is a function of the left and right index in the matrix representing the diagonal. For example, $\delta_{r,s}$ would describe the identity matrix. One dimensional functions should follow the indexing on the column, so that $f(i) = f(r) \neq f(s)$. So then (for example) a discretized harmonic oscillator potential when put in the Crank-Nicholson propagator would become, $V(\xi) = V(ih) = \frac{1}{2}(ih)^2 = \frac{1}{2}(rh)^2$, where the matrix representing the propagator has indexes r, s.

The inverse of the implicit part of the propagator in Eq. (7) is also important to consider. H is only a tridiagonal matrix, meaning a band matrix with a width of three. Therefore when taking the inverse of the matrix, it is unnecessary to take as many steps required for a full matrix inverse. The number of operations per inverse can be reduced by a power of N (being the number of points in ϕ). This reduction can be achieved by solving N equations using a substitution algorithm or doing a Gauss-Jordan inversion algorithm without doing algebra on the entire row [ea92]. My program utilizes that latter of these two methods.

The grid spacing is another aspect to be aware of. h can be considered to be the fineness of the spacing on ξ and τ the fineness on λ (time). The total space spanned is equal to the number of points in ϕ multiplied by the grid-spacing h. It is important to choose a grid fine enough to accurately represent ϕ, but not so fine as to strain the computer when N must increased to accommodate the same region. For the one-dimensional Schrodinger equation, it is sufficient to cover up to four harmonic oscillator wavelengths with a step size of $h = .04$ (i.e. N=100).

3.6 The One Dimensional Gross-Pitaevskii Equation

Once the programming for the one dimensional Schrodinger equation has been checked for accuracy (using simple quantum mechanics equations), we can simply modify the equation in order to arrive at the solution for the Gross-Pitaevskii equation [DGPS99]. The Gross-Pitaevskii Equation in

scaled coordinates (with harmonic oscillator potential) can be written as

$$i\frac{\partial\psi(\xi,\lambda)}{\partial\lambda} = -\frac{1}{2}\frac{\partial^2\phi(\xi,\lambda)}{\partial\xi^2}\phi(\xi,\lambda) + \frac{1}{2}\xi^2\phi(\xi,\lambda) + U|\phi(\xi,\lambda)|^2\phi(\xi,\lambda)\ , \quad (8)$$

where U that depends on the diagonal scattering length. This new interatomic interaction term describes the interaction between the atoms inside the BEC. When discretizing and solving for ϕ^{n+1}, the new form of Eq. (7) is

$$\overrightarrow{\phi^{n+1}} = \left(I + \frac{i\tau H^n}{2}\right)^{-1}\left(I - \frac{i\tau H^n}{2}\right)\overrightarrow{\phi^n}, \quad (9)$$

$$H^n_{r,s} = \left(-\frac{(\delta_{r+1,s} - 2\delta_{r,s} + \delta_{r-1,s})}{2h^2} + \frac{1}{2}(rh)^2\delta_{r,s} + U|\phi^n_r|^2\delta_{r,s}\right)\ . \quad (10)$$

Now that the nonlinear term has been introduced, the Hamiltonian now takes on an index in time. That means that while the Schrodinger equation needs the Hamiltonian computed only once, the GP equation requires that the Hamiltonian be evaluated after every propagation and that the new wave function is normalized every iteration.

3.7 The Radial Gross-Pitaevskii Equation

It is useful to solve the radial GP equation at the same time as solving the cartesian GP equation. Many of the useful potentials used in experimentation can be more easily written in cylindrical coordinates rather than cartesian coordinates. Also, by comparing the results for the cartesian and radial GP equations it serves as an extra error check on programming. By using the scaling $L = \sqrt{\frac{\hbar}{2m\omega}}$ and the Laplacian in cylindrical coordinates, the radial GP equation can be written as [HJCC97]

$$i\frac{\partial\psi(\rho,\lambda)}{\partial\lambda} = -\left(\frac{\partial^2}{\partial\rho^2} + \frac{1}{\rho^2} - \frac{\partial}{\rho\partial\rho}\right)\phi(\rho,\lambda) + \frac{1}{4}\rho^2\phi(\rho,\lambda) + \frac{1}{\rho^2}U|\phi(\rho,\lambda)|^2\phi(\rho,\lambda)\ . \quad (11)$$

The same numerics used on the cartesian GP equation can also be used on the radial GP equation. When discretizing and solving for ϕ^{n+1}, the new form of Eq. (7) is

$$\overrightarrow{\phi^{n+1}} = \left(I + \frac{i\tau H^n}{2}\right)^{-1}\left(I - \frac{i\tau H^n}{2}\right)\overrightarrow{\phi^n}, \quad (12)$$

$$H^n = \left(-(\frac{(\delta_{r+1,s} - 2\delta_{r,s} + \delta_{r-1,s})}{h^2} + \frac{1}{(rh)^2}\delta_{r,s} - \frac{\delta_{r+1,s} - \delta_{r-1,s}}{(rh)h})\right) + \left(\frac{1}{4}(rh)^2\delta_{r,s} + \frac{1}{(rh)^2}U|\phi^n|^2\delta_{r,s}\right)\ . \quad (13)$$

Now that the nonlinear term has been introduced, the Hamiltonian now takes on an index in time. That means that while the Schrodinger equation needs the Hamiltonian computed only once, the GP equation requires that the Hamiltonian be evaluated after every propagation and that the new wave function is normalized every iteration.

It should be noted that this scheme has been revealed, using von Neumann analysis, to be unstable. This can be resolved however, by choosing an appropriate time step. Since the highest order of the grid spacing h in the Hamiltonian is h^{-2}, if τ is chosen such that $\frac{\tau}{h^2} \ll 1$ then propagation should still be accurate.

3.8 Propagation in Imaginary Time and the Heat Equation

Wave functions can be described by a linear combination of stationary states. That means that every wave function will look similar to

$$c_1\psi_1 e^{\frac{-iE_1t}{\hbar}} + c_2\psi_2 e^{\frac{-iE_2t}{\hbar}} + c_3\psi_3 e^{\frac{-iE_3t}{\hbar}} \dots , \tag{14}$$

with E_k corresponding to the energy of the ψ_k eigenstate. Now suppose the time variable t is replaced with $-it$. Then the new decomposition of the wave function is

$$c_1\psi_1 e^{\frac{-E_1t}{\hbar}} + c_2\psi_2 e^{\frac{-E_2t}{\hbar}} + c_3\psi_3 e^{\frac{-E_3t}{\hbar}} \dots . \tag{15}$$

Now all of the eigenstates decay as exponentials. If left alone, the wave function would decay to zero with the higher energy eigenfunctions decaying the fastest. If the wave function is frequently normalized however, only one solution remains... the eigenfunction with the lowest energy, the ground state! This is referred to as the method of steepest decent [Wil99].

This method can be accomplished numerically by setting τ equal to $-i\tau$. Note that the method of steepest decents requires frequent renormalization of the wave function, otherwise the ground state is lost in round off error.

Results of the propagation in imaginary time of the cartesian GP equation is shown in Fig. 2, results of the propagation in imaginary time of the radial GP equation is shown in Fig. 3. Note that the value $U = 0$ corresponds to results for the Schrodinger equation

Another physical comparison of propagation in imaginary time is the

heat equation in real time. By taking Eq. 8 and making the imaginary time substitution from λ to $-i\lambda$, the new equation becomes the Sturm-Liouville problem that describes the temperature in a metal bar under different heating conditions,

$$\frac{\partial\psi(\xi,\lambda)}{\partial\lambda} = \frac{1}{2}\frac{\partial^2\phi(\xi,\lambda)}{\partial\xi^2} - V(\xi)\phi(\xi,\lambda). \tag{16}$$

The heat equation when propagated in real time will converge to a final steady state solution. This fact further illustrates the point that when propagating in imaginary time, the wave equation will reach a steady state function, and as discussed, that steady state value is the ground state.

3.9 Initial Condition and the Thomas-Fermi Approximation

While any wave function when propagated in imaginary time will converge to the ground state solution, some initial wave functions converge faster than others. Starting with an initial wave function that approximates the ground state will propagate to the ground state faster and not be as computationally expensive as other initial wave functions.

One method for modeling the ground state of the Gross-Pitaevskii equation is the Thomas-Fermi approximation [PS02]. For sufficiently large clouds the ratio of the kinetic energy to the interaction (potential) energy is very small. Thus an appropriate approximation to the ground state (in terms of energy) is the differential equation describing the GP equation without the kinetic energy term. Then considering that the ground state is stationary,

$$i\hbar\frac{\partial\Psi(r,t)}{\partial t} = H\psi = E\psi = V(r)\psi + U|\psi(r)|^2\psi, \tag{17}$$

$$|\psi(r)|^2 = \frac{[E - V(r)]}{U}\ . \tag{18}$$

In the case of the one dimensional harmonic oscillator $V = \frac{1}{2}m\omega^2r^2$ and $E = \frac{1}{2}\hbar\omega$, which in scaled coordinates means $V = \frac{1}{2}x^2$ and $E = \frac{1}{2}$. This wave function is only valid for $E > V$, otherwise the wave function is set to zero. The physical representation of the wave function is a parabolic "bubble" with boundary at the points where the potential is equal to the total energy, $V(r) = E$.

3.10 Normalization

There can be an infinite number of solutions to a particular differential equation. What normally selects a specific function for a given differential equation is an initial value. Due to the statistical interpretation of quantum mechanical problems however, there is a normalization condition required of the wave function that solves our differential equation. The area under the wave function squared must equal a certain value. We use this selection rule because the normalization can be related to the number of particles in the BEC, making the normalization value much easier to define than the wave function at a specific spatial coordinate.

For the one dimensional Schrodinger and Gross-Pitaevskii equations, the normalization condition put on the wave function and the non-dimentionalized version $\phi(\xi) = \psi(\xi L)$ is :

$$\int_{-\infty}^{\infty} |\psi(r,t)|^2 dr = \langle\psi|\psi\rangle = N, \tag{19}$$

$$L\int_{-\infty}^{\infty} \phi^*\phi d\xi = N\ . \tag{20}$$

where N is the total number of particles in the system. When working in cylindrical coordinates, the helper function $\phi = \sqrt{L}\rho\psi$ requires a different normalization condition, which can be derived by substituting $\frac{\phi}{\rho}$ for ψ in the normalization for ψ in cylindrical coordinates

$$\int_0^{2\pi}\int_0^{\infty} |\psi(r)|^2 r dr d\theta = N, \tag{21}$$

$$2\pi\int_0^{\infty} |\frac{\phi(\rho)}{\rho\sqrt{L}}|^2 \rho L d(\rho L) = N, \tag{22}$$

$$2\pi L\int_0^{\infty} \frac{|\phi(\rho)|^2}{\rho} d\rho = \langle\phi|\phi\rangle_{cyn} = N\ . \tag{23}$$

3.11 Expected Wave-functions

Numerical solutions must be validated against the physics of the problem. By using the scaled coordinate systems used for the programming, it is also possible to derive analytical results. In the case of cartesian coordinates with the scaling $\sqrt{\frac{\hbar}{m\omega}}$, the ground state of the harmonic oscillator potential [Gri05] and numerical equivalent is subsequently (with the final calculation making sure that $\langle\psi|\psi\rangle = N$)

$$\psi_0(x) = \sqrt{N}\left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-\frac{m\omega}{2\hbar}x^2}, \tag{24}$$

$$\phi_0(\xi) = \psi_0(\xi L) = \sqrt{N}\left(\frac{1}{\pi L^2}\right)^{\frac{1}{4}} e^{-\frac{1}{2}\xi^2}, \tag{25}$$

$$\phi_0^*\phi_0 = \frac{N}{L\sqrt{\pi}}e^{-\xi^2}\ , \langle\psi_0|\psi_0\rangle = \int_{-\infty}^{\infty}\phi_0^*\phi_0 d(\xi L) = \int_{-\infty}^{\infty}\frac{N}{\sqrt{\pi}}e^{-\xi^2}d\xi = N \tag{26}$$

That means for the case of a single particle (N=1) the normalization for our scaled wave function is one, and that the expected ground state squared is $\phi_0^*\phi_0 = \frac{1}{\sqrt{\pi}}e^{-\xi^2}$ with coordinates in units of L.

In the case of cylindrical coordinates the scaling is different with $L = \sqrt{\frac{\hbar}{2m\omega}}$ and . Also in this scheme, $\phi(\rho) = \sqrt{L}\rho\psi(\rho L)$. This is for simplification of the propagator and (an added bonus for simplicity) force the boundary conditions of $\phi(\xi)$ to zero. Then the expected value of $|\phi(\xi)|^2$ becomes

$$\psi_0(r) = r\left(\frac{Nm\omega}{\pi\hbar}\right)^{\frac{1}{2}} e^{-\frac{m\omega}{2\hbar}r^2}, \tag{27}$$

$$\phi_0(\rho) = \sqrt{L}\rho\psi_0(\rho L) = \rho\left(\frac{N}{2\pi L}\right)^{\frac{1}{2}} e^{-\frac{1}{4}\rho^2}, \tag{28}$$

$$\begin{aligned}\phi_0^*\phi_0 &= \left(\frac{N\rho^2}{2\pi L}\right) e^{-\frac{1}{2}\rho^2}\ , \langle\psi_0|\psi_0\rangle = 2\pi L\int_0^{\infty}\frac{\phi_0^*\phi_0}{(\rho)}d\rho \\ &= 2\pi\int_0^{\infty}\left(\frac{N\rho}{2\pi}\right) e^{-\frac{1}{2}\rho^2}d\rho = N\ .\end{aligned} \tag{29}$$

That means for the case of a single particle ($N = 1$) the normalization for our scaled wave function is one, and that the expected ground state squared is $\phi_0^*\phi_0 = \frac{\rho^2}{2\pi}e^{-\frac{1}{2}\rho^2}$ with coordinates in units of L.

3.12 Expected Energies

In order to check the solution of the Gross-Pitaevskii equations in cartesian and cylindrical coordinates, the scaled energies must also be known. By applying techniques similar to those used in the previous section, the energies can be derived. By transforming the equation $\langle\psi|H|\psi\rangle = E$ to an equation in ϕ we derive the unitless energy. For cartesian coordinates the one-dimensional Hamiltonian, H, is equal to $-\frac{1}{2}\frac{d^2}{dx^2} + \frac{1}{2}x^2$. It is also useful to find the scaled energies for the kinetic and potential energies separately.

$$E = \langle\psi_0|H|\psi_0\rangle = \int_\infty^\infty \phi^* H\phi d(\xi L) = N\int_\infty^\infty \frac{1}{2\sqrt{\pi}}e^{-\xi^2}d\xi = \frac{N}{2}, \tag{30}$$

$$KE = \langle\psi_0| - \frac{1}{2}\frac{d^2}{dx^2}|\psi_0\rangle = N\int_\infty^\infty \frac{1}{2\sqrt{\pi}}(1-\xi^2)e^{-\xi^2}d\xi = \frac{N}{4}, \tag{31}$$

$$PE = \langle\psi_0|\frac{1}{2}x^2|\psi_0\rangle = N\int_\infty^\infty \frac{1}{2\sqrt{\pi}}(\xi^2)e^{-\xi^2}d\xi = \frac{N}{4}\ . \tag{32}$$

This result matches with what is expected for the harmonic oscillator: the kinetic energy is equal to the potential energy. While the energy of the scaled harmonic oscillator ground state for $N = 1$ is $\frac{1}{2}$, the energy of the unscaled harmonic oscillator is $\frac{1}{2}\hbar\omega$. From this it can be reasoned that to transform a scaled energy to an unscaled energy, simply multiply by $\hbar\omega$.

Similar calculations can be performed in cylindrical coordinates. However the equations again have a different scaling and substitution scheme all together, making the calculations slightly more dense. For cylindrical coordinates the transformed Hamiltonian, H, is equal to $-(\frac{\partial^2}{\partial\rho^2} + \frac{1}{\rho^2} - \frac{\partial}{\rho\partial\rho}) + \frac{1}{4}\rho^2$. Then the equations for the energies in cylindrical coordinates becomes

$$E = \langle\psi_0|H|\psi_0\rangle = 2\pi\int_0^\infty \frac{\phi^* H\phi}{\rho}d(\rho L) = 2\pi N\int_0^\infty \frac{1}{2\pi}\rho e^{-\frac{1}{2}\rho^2}d\rho = N, \tag{33}$$

$$KE = \langle\psi_0|H - \frac{1}{4}\rho^2|\psi_0\rangle = 2\pi N\int_0^\infty \frac{1}{8\pi}\rho(4-\rho^2)e^{-\frac{1}{2}\rho^2}d\rho = \frac{N}{2}, \tag{34}$$

$$PE = \langle\psi_0|\frac{1}{4}\rho^2|\psi_0\rangle = 2\pi N\int_0^\infty \frac{1}{8\pi}\rho^3 e^{-\frac{1}{2}\rho^2}d\rho = \frac{N}{2}\ . \tag{35}$$

Again the result is that the kinetic energy is equal to the potential energy

and that to derive the unscaled energy the scaled energy must be multiplied by a factor of $\hbar\omega$.

3.13 Inter-atomic Interaction Coefficient

So far the symbol U has been used to represent the coefficient in front of the nonlinear term. But like everything else in this problem, there exists certain physical properties behind this coefficient. For cartesian coordinates this value is proportional to the scattering length α, given by $U = \frac{N\alpha\hbar^2}{m}$, while for cylindrical coordinates the factor is $\frac{4\pi N\alpha\hbar^2}{m}$. When scaled the inter-atomic interaction coefficient in cartesian coordinates is simply the scattering length $N\alpha$. The scaled coefficient in cylindrical coordinates becomes $U = 4\pi N\alpha$, where N is the number of particles in the trap [HJCC97]. With these coefficients in place the methodology behind this thesis work is complete.

4 Results

4.1 Propagation in Imaginary Time to Analytically Derived Ground State

The next step in error checking the program was to propagate an initial wave function in imaginary time down to the ground state of the harmonic oscillator. The analytically derived wave function for the ground state of the harmonic oscillator in one dimension(in our scaled coordinate system) is $\phi(\xi) = e^{-\xi^2}$ for cartesian coordinates and in two dimensions with $\psi(\rho) = \frac{1}{2\pi}\rho^2 e^{-\frac{1}{2}\rho^2}$. Comparing analytical and numerical wave functions provides an excellent means of testing the validity of the code and the numerical method.

Results of propagating the one dimensional Schrodinger equation in imaginary time are shown in Fig. 1. The wave functions are converging to the analytical solution (shown in green) as predicted.

Propagation of the Gross-Pitaevskii equation in imaginary time are dependent on the inter-atomic interaction coefficient (U). Ground state solutions for the one dimensional GP equation are shown in Fig. 2 while the ground state solutions for the two dimensional, axially symmetric GP equation are shown in Fig. 3. Note for increasing values of U, wave functions are more spread out from the center. This agrees with the intuitive picture of a BEC with repulsive inter-atomic interactions.

4.2 First-order Approximation of the Energy in the Ground State of the GP Equation

To check the results for the Gross-Pitaevskii equation, we must be more careful. The eigenstates of the Gross-Pitaevskii equation are not be known (it is a nonlinear equation), so it is impossible to check that a linear combination of eigenstates is preserved. Instead the energy of the ground state must be recovered and verified using perturbation theory. The first order approximation of the energy of the ground state,

$$E^{(1)} = \langle \phi_0 | U | \phi_0 |^2 | \phi_0 \rangle, \tag{36}$$

where ϕ_0 is the ground state of the harmonic oscillator and U is the strength of the interatomic interactions. That means for Figs. 2,3 the value $U = 0$ corresponds to the results for the Schrodinger equation. For small inter-atomic interactions , $\langle \phi | H | \phi \rangle \approx U \langle \phi_0 | | \phi_0 |^2 | \phi_0 \rangle - E_0$, the second part of the equation, a linear function in U, being capable of analytically solved for. So if $\langle \phi | H | \phi \rangle$ is plotted versus small values of U then we should get a linear graph with slope

$$\langle \phi_0 | | \phi_0 |^2 | \phi_0 \rangle = 2\pi \int_0^\infty \frac{(\phi_0^* \phi_0)^2}{\rho} d\rho = .07958 \ . \tag{37}$$

As shown in Fig. 4, the slope of the graph of energy versus inter-atomic coefficient closely matches expected results.

For larger values of U, the 2nd-order correction becomes a non-negligible part of the correction to the energy of the BEC. The formula for the second-order correction to the energy of the ground state is

$$E_0^{(2)} = U^2 \sum_{m \neq 1} \frac{| \langle \phi_m^{(0)} | | \phi_0 |^2 | \phi_0^{(0)} \rangle |^2}{E_0^{(0)} - E_m^{(0)}}. \tag{38}$$

Note that when $U \ll 1$ then $U > U^2$. But as U increases, the second order terms contribute more to the energy of the BEC (assuming the approximations are well-behaved, i.e. the first order energy correction is greater than the second-order energy correction and so on). Since the numerator is positive and the denominator is negative, ($E_0 < E_m$) for all m, then the second order correction is always negative. So for small values of U, the energy plotted versus U should be linear with a slope of .07958 and as U increases the plot should deviate negatively from the linear approximation. This is confirmed by the results shown in Fig. 4.

5 Conclusion

To reiterate, through a variety of numerical techniques I have derived the ground state solutions to the Schrodinger and Gross-Pitaevskii equations. The numerically derived wave functions are meant to model novel gaseous BECs in a confining potential. The results obtained have matched favorably to analytical results, confirming the validity of the numerical method. When propagating in imaginary time, the wave equations relaxed to analytically derived wave equations accurate to three significant figures. Energies derived from the ground state of the Gross-Pitaevskii equation also match favorably with expected results. For small inter-atomic interactions, energies appear to be linear with slope approximately equal to analytical results. The energies given by first order perturbation theory also closely approximate numerically derived energies.

By all of these diagnostics I conclude that the numerical approximation for the one dimensional GP equation in cartesian coordinates and the two dimensional, axially symmetric equation in cylindrical coordinates is reliably accurate. If the accuracy of the numerical approximation isn't suitable for a specific problem, simply decrease grid spacing to obtain a more desirable result.

It is hoped that in the future this work can be used with various potentials and Hamiltonians to accurately describe the physical phenomena surrounding Bose-Einstein condensates. For example, this work can be used to model dipolar interactions within, and ballistic expansion of Bose-Einstein condensates.

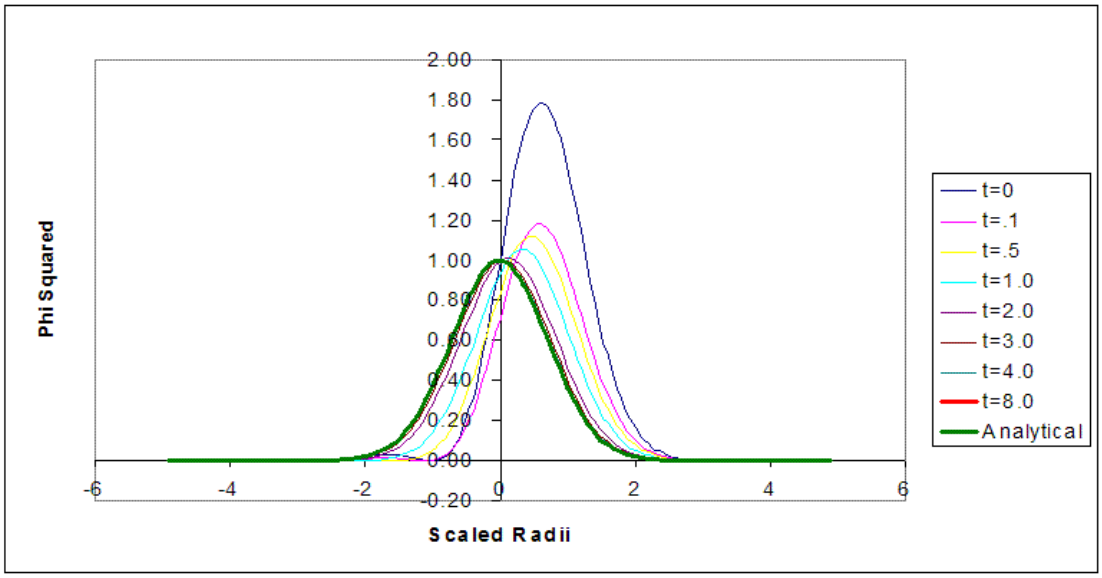

Figure 1: Propagation in imaginary time to the ground state of the harmonic oscillator in cartesian coordinates.

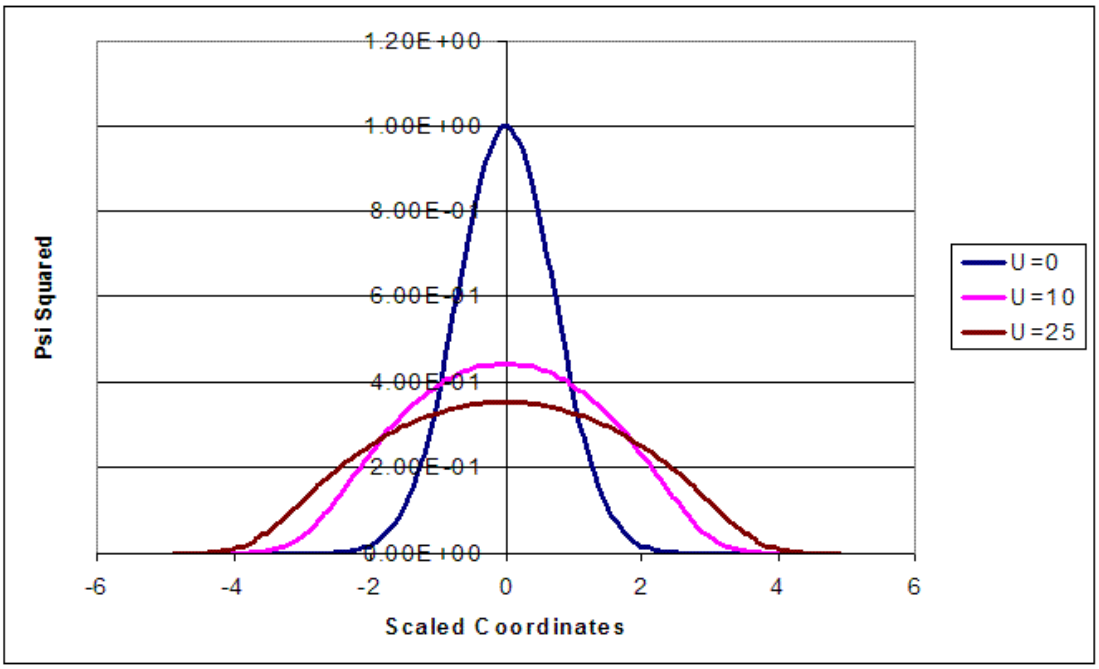

Figure 2: Ground state of a one dimensional BEC in cartesian coordinates with varying inter-atomic interactions. The confining potential is that of the harmonic oscillator.

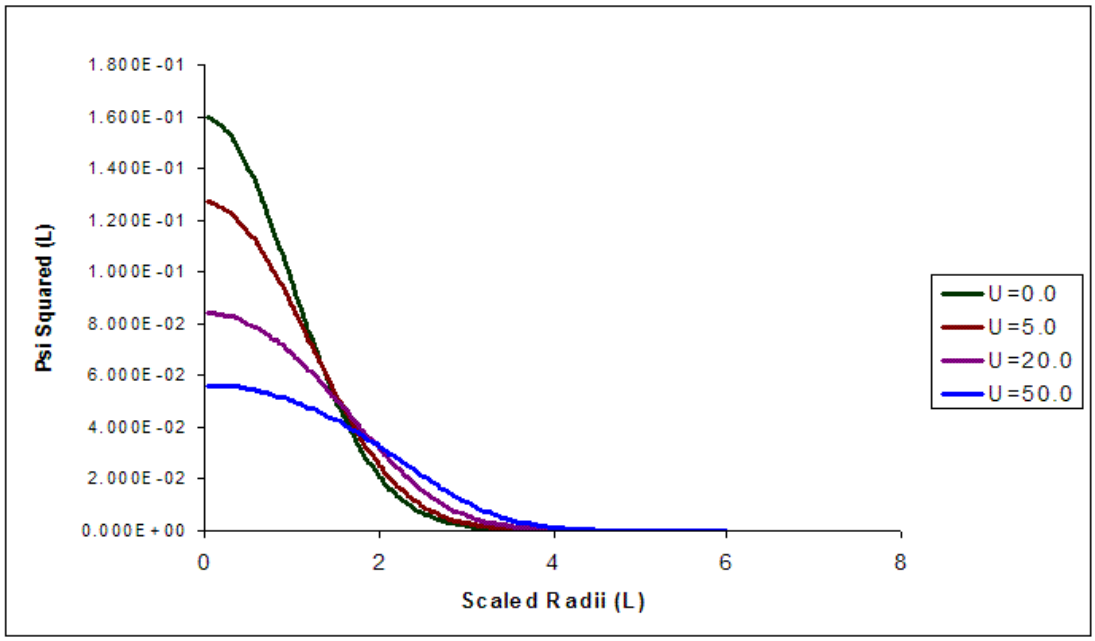

Figure 3: Ground state of a two dimensional, axially symmetric BEC in scaled cylindrical coordinates with varying inter-atomic interactions. The confining potential is that of the harmonic oscillator.

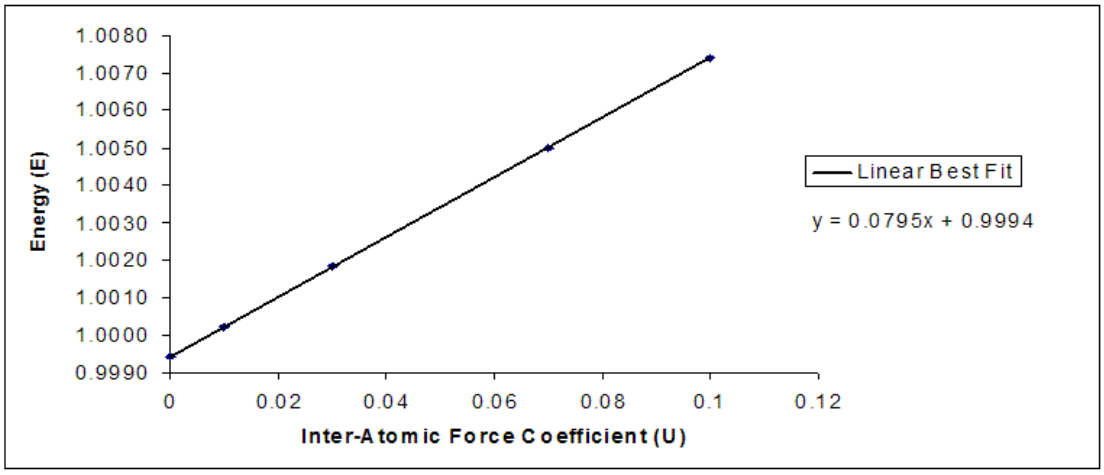

Figure 4: Energy of BEC as a function of small values of the inter-atomic interaction coefficient (U).

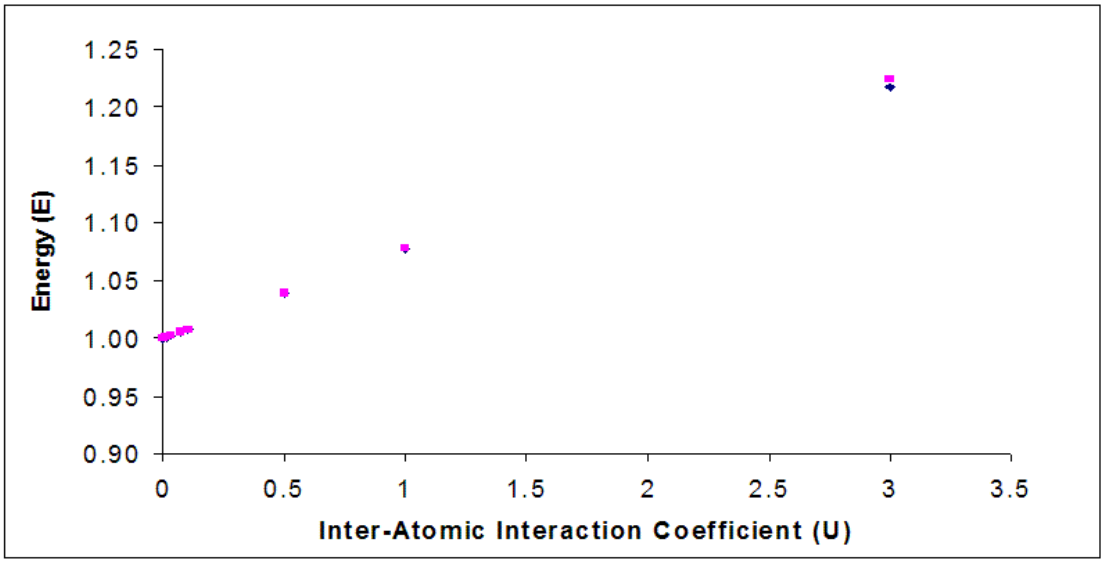

Figure 5: Numerically derived ground state energies of the GP equation (blue) with varying inter-atomic interactions as well as the approximate energies (purple) given by first order perturbation theory.

References

[DGPS99] F. Dalfovo, S. Giorgini, L. P. Pitaevskii, and S. Stringari. Theory of Bose-Einstein condensation in trapped gases. *Rev. Mod. Phys.*, 71(3):463–512, 1999.

[ea92] E. Anderson et. al. *CLAPACK Routines*. http://gams.nist.gov/serve.cgi/Package/CLAPACK/, 1992.

[Gri05] D J Griffiths. *Introduction to Quantum Mechanics*. Pearson Prentice Hall, 2nd edition, 2005.

[HJCC97] M. J. Holland, D. S. Jin, M. L. Chiofalo, and J. Cooper. Emergence of interaction effects in Bose-Einstein condensation. *Phys. Rev. Lett.*, 78(20):3801, May 1997.

[Ise96] Arieh Iserles. *Numerical Analysis of Differential Equations*. Cambridge University Press, 1996.

[PS02] C. J. Pethick and H. Smith. *Bose-Einstein Condensation in Dilute Gasses*. Cambridge University Press, 2002.

[Wil99] J E Williams. *The Preparation of Topological Modes in a Strongly-Coupled Two-Component Bose-Einstein Condensate*. PhD thesis, University of Colorado, Boulder, 1999.

6 Appendix

Coding for 1D Gross-Pitaevskii Equation

```
!The following is the programming for the Gross-Pitaevskii equation
!in Cartesian coordinates.

PROGRAM OneDimGPit
  IMPLICIT NONE

  REAL, EXTERNAL :: kron, sqinter, mu, muprime
  REAL, PARAMETER :: M=99, TERM=700
  INTEGER, PARAMETER :: Mc=99, TERMc=700, num_pts=2*Mc-1
  INTEGER :: i,j,n,q
  REAL :: h, tau, lambda, xi,norm,normt
  COMPLEX :: carrier
  COMPLEX, DIMENSION (2*Mc-1,TERMc) :: PHI
  REAL, DIMENSION (2*Mc-1) :: PHISQ1, PHISQ2
  COMPLEX, DIMENSION (2*Mc-1,2*Mc-1,TERMc) :: G
  COMPLEX, DIMENSION (2*Mc-1,2*Mc-1) :: INV, MIN1, MIN2, prop, IM

  h=.05
  tau=.01
  lambda=20.0
  IM=0




 !Pre-Workhorse stuff. Initial PSI, IM input, etc.
  DO i=1,num_pts
    xi = (i-Mc) * h

  !Initial Phi function
    PHI(i,1)=  EXP(-.5*xi**2)*EXP((-.5)*(0.0,1.0)*tau)

  !Identity Matrix Input
    IM(i,i)=1.0
```

```
  END DO




  norm= sqinter(PHI,num_pts,TERMc,h,1)



  WRITE(*,*) "Propagating..."

G=0 !The Workhorse
  DO n=1,TERMc-1

   !Populate H matrix, which I have called G(i,j,n)
    DO i=1,num_pts
      DO j=1,num_pts
        xi = (i-Mc) * h



         G(i,j,n)= (-1/(2.0*h**2)*(kron(i,j+1) - 2.0*kron(i,j) &
                   + kron(i,j-1)) + &
                   (1/2.0*xi**2 + lambda*PHI(j,n)*CONJG(PHI(j,n))) &
                   *kron(i,j))*(0.0,-1.0)



        MIN1(i,j)=IM(i,j)+(0.00,1.00)*tau*G(i,j,n)/2
        MIN2(i,j)=IM(i,j)-(0.00,1.00)*tau*G(i,j,n)/2


      END DO
    END DO
```

```
   !Invert the first term
    CALL TRIINV (MIN1,num_pts,INV)



   !The propogator
    prop=MATMUL(INV,MIN2)



     !Putting it all together and propagating, PHI(n+1)=prop*PHI(n)
      DO i=1,num_pts
         carrier=0
         DO j=1,num_pts
           carrier=carrier+prop(i,j)*PHI(j,n)
         END DO
         PHI(i,n+1)=carrier
       END DO


       normt=sqinter(PHI,num_pts,TERMc,h,n+1)
       DO i=1,num_pts
         !PHI(i,n+1)=((3.1415926**.5/normt))*PHI(i,n+1)
         PHI(i,n+1)=((norm/normt)**(.5))*PHI(i,n+1)

       END DO



! End Workhorse
  END DO




!Final Hamiltonian Term
  DO i=1, num_pts
    DO j=1,num_pts
       xi = (i-Mc) * h
```

```
        G(i,j,TERMc)= (-.5/(h**2)*(kron(i,j+1) - 2.0*kron(i,j) &
                  + kron(i,j-1)) + &
                  (.5*xi**2 + lambda*PHI(j,TERMc)*CONJG(PHI(j,TERMc))) &
                  *kron(i,j))*(0.0,-1.0)


  END DO
END DO




DO i=1,num_pts
  !WRITE(*,*) PHI(i,TERMc)
  PHISQ1(i)=PHI(i,TERMc)*CONJG(PHI(i,TERMc))
  PHISQ2(i)=PHI(i,TERMc-1)*CONJG(PHI(i,TERMc-1))
END DO

!WRITE(*,*) "And Phi Squared, middle time..."

!DO i=1, num_pts
!  PRINT *, PHISQ2(i)
!END DO


!WRITE(*,*) "And Phi Squared, final time..."

DO i=1,num_pts
  PRINT *, PHISQ1(i)
END DO


!WRITE(*,*) "Norms for various times: "
!DO i=10,TERMc,10
!  PRINT *,  sqinter(PHI,num_pts,TERMc,h,i)
!  PRINT *,  sqinter(PHI,num_pts,TERMc,h,TERMc-1)
!END DO
```

```
  !WRITE(*,*) "Mu: "
  !DO i=10,TERMc,10
  !  PRINT *, mu(PHI,G,num_pts,TERMc,h,i)
  !END DO

  !WRITE(*,*) "Mu Prime (analytical):"
  !DO i=10,TERMc,10
  ! PRINT *,  muprime(PHI,num_pts,TERMc,h,TERMc,lambda)+.5
  !END DO




END PROGRAM



!Kronecker Delta Function REAL FUNCTION kron(i,j)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: i,j

  IF (i==j) THEN
    kron = 1.0
  ELSE
    kron = 0.0
  END IF
END FUNCTION kron



REAL FUNCTION mu(M,G,N,T,h,time)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: N,T,time
  REAL, EXTERNAL :: kron, sqinter
  INTEGER :: i,j
  COMPLEX :: carrier
  COMPLEX, INTENT(IN), DIMENSION(N,T) :: M
```

```
  COMPLEX, INTENT(IN), DIMENSION(N,N,T) :: G
  COMPLEX, DIMENSION(N,1) :: dimmer1, dimmer2, secterm
  REAL, INTENT(IN) :: h
  COMPLEX, DIMENSION (N,N)  :: E

  DO i=1,N
    dimmer1(i,1)=CONJG(M(i,time))
    dimmer2(i,1)=M(i,time)
    DO j=1,N
      E(i,j)=G(i,j,time)*(0.0,1.0)
    END DO
  END DO



  secterm=MATMUL(E,dimmer2)

  carrier=0
  DO i=1,N
    rho=i*h
    carrier=carrier+dimmer1(i,1)*secterm(i,1)
  END DO

  mu=carrier*h/sqinter(M,N,T,h,time)
  !mu=carrier*h

END FUNCTION mu



REAL FUNCTION muprime(M,N,T,h,time,lambda)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: N,T,time
  REAL, EXTERNAL :: kron, sqinter
  INTEGER :: i,j
  COMPLEX :: carrier
  COMPLEX, INTENT(IN), DIMENSION(N,T) :: M
  COMPLEX, DIMENSION(N,1) :: dimmer1, dimmer2, secterm
  REAL, INTENT(IN) :: h,lambda
  COMPLEX, DIMENSION (N,N)  :: E
```

```
  DO i=1,N
    dimmer1(i,1)=CONJG(M(i,1))
    dimmer2(i,1)=M(i,1)
    DO j=1,N
      E(i,j)= lambda*M(j,time)*CONJG(M(j,time))*kron(i,j)
    END DO
  END DO



  secterm=MATMUL(E,dimmer2)

  carrier=0
  DO i=1,N
    carrier=carrier+dimmer1(i,1)*secterm(i,1)
  END DO

  muprime=carrier*h/sqinter(M,N,T,h,1)
  !muprime=carrier*h

END FUNCTION muprime









!Square-Integrator using Simpson's Rule

REAL FUNCTION sqinter(M,N,T,h,time)
  IMPLICIT NONE


  INTEGER, INTENT(IN) :: N,T,time
  INTEGER :: i
  COMPLEX, INTENT(IN), DIMENSION (N,T) :: M
  REAL, INTENT(IN) :: h
```

```
  REAL :: S0,S1,S2,S



!
  S  = 0.0
  S0 = 0.0
  S1 = 0.0
  S2 = 0.0
  DO I = 2, N-1, 2
    S1 = S1+M(I-1,time)*CONJG(M(I-1,time))
    S0 = S0+M(I,time)*CONJG(M(I,time))
    S2 = S2+M(I+1,time)*CONJG(M(I+1,time))
  END DO
  S = h*(S1+4.0*S0+S2)/3.0

! ! If N is even, add the last slice separately

! NOTE: In this program N is programed to be odd

! So this isn't necessary.
  IF (MOD(N,2).EQ.0) S = S &
     +H*(5.0*M(N,time)*CONJG(M(N,time)) &
     +8.0*M(N-1,time)*CONJG(M(N-1,time)) &
     -M(N-2,time)*CONJG(M(N-2,time)))/12.0

 sqinter=S


END FUNCTION sqinter




!Inversion of a tridiagonal matrix using Gauss-Jordan elimination
!to the identity matrix and recording the elimination matrices.
SUBROUTINE TRIINV (A,N,X)
  IMPLICIT NONE

  REAL, EXTERNAL :: kron
  INTEGER, INTENT (IN) :: N
```

```
COMPLEX, INTENT (IN), DIMENSION (N,N) :: A
COMPLEX, DIMENSION(N,N) :: COP
COMPLEX, INTENT (OUT), DIMENSION (N,N):: X
COMPLEX, DIMENSION (N,N) :: B
INTEGER :: i,j
COMPLEX :: temp

!Initialize B to the identity matrix
!Copy A in to COP
DO i=1,N
  DO j=1,N
    B(i,j)=kron(i,j)
   COP(i,j)=A(i,j)
  END DO
END DO




DO i=1,N
  IF(i/=N) THEN

    temp=COP(i,i)
    DO j=1,i
      B(i,j)=B(i,j)/temp
    END DO
    DO j=i,i+1
      COP(i,j)=COP(i,j)/temp
    END DO

    temp=COP(i+1,i)
    DO j=1,i
      B(i+1,j)=B(i+1,j)-B(i,j)*temp
    END DO
    DO j=i,i+1
      COP(i+1,j)=COP(i+1,j)-COP(i,j)*temp
    END DO

  ELSE
    temp=COP(N,N)
    DO j=1,N
```

```
        B(N,j)=B(N,j)/temp
      END DO
      COP(N,N)=1
    END IF


  END DO



  DO i=N,1,-1
    IF(i/=1) THEN

      temp=COP(i-1,i)
      DO j=1,N
        B(i-1,j)=B(i-1,j)-B(i,j)*temp
        COP(i-1,j)=COP(i-1,j)-COP(i,j)*temp
      END DO
    END IF

  END DO



  DO i=1,N
    DO j=1,N
      X(i,j)=B(i,j)
    END DO
  END DO

END SUBROUTINE TRIINV
```

Appendix B: Coding for 2D Gross-Pitaevskii Equation

```
 PROGRAM OneDimGPit
```

```
  IMPLICIT NONE

  REAL, EXTERNAL :: kron, sqinter, mu,kinen
  REAL, PARAMETER :: M=100, TERM=400
  INTEGER, PARAMETER :: Mc=100, TERMc=400, num_pts=Mc
  INTEGER :: i,j,r,s,n
  REAL :: h, lambda, rho,norm,normt
  COMPLEX :: carrier,tau
  COMPLEX, DIMENSION (num_pts) :: PHI1,PHI2
  REAL, DIMENSION (num_pts) :: PHISQ1, PHISQ2
  COMPLEX, DIMENSION (num_pts,num_pts) :: INV1, MIN1, &
                              MIN2, prop ,prop2,IM,Hp,Gp

  h=.05
  tau=.01
  lambda=1
  IM=0

  !If imaginary time then the following line
  tau=tau*(0.0,-1.0)

 !Pre-Workhorse stuff. Initial PSI, IM input, etc.
  DO i=1,num_pts
    rho = i * h

  !Initial Phi function
    PHI1(i)=  EXP(-.5*rho**2)*EXP((-.5)*(0.0,1.0)*tau) &
              +rho*EXP(-.5*rho**2)*EXP(-1.5*(0.0,1.0)*tau)&
              + 10*SIN(rho)




  !Identity Matrix Input
    IM(i,i)=1.0



  END DO
```

```
norm= sqinter(PHI1,num_pts,h)

WRITE(*,*) "Wait for it! ... Wait for it!"




 !Populate H matrix, which I have called Hp(r,s)
  DO r=1,num_pts
    DO s=1,num_pts
      rho = r * h




      !1/R*Derivative(R*Derivative)=1/R * (1st derivative + R*2nd-derivative)


         Hp(r,s)=  - kron(r,s)/(rho**2) + .25*rho*kron(r,s)
         Gp(r,s)= 1.0/(2*rho*h)*(kron(r,s-1)-kron(r,s+1)) - &
             (kron(r,s+1)-2.0*kron(r,s)+kron(r,s-1))/(h**2) &
                  +lambda*PHI(r,n)*CONJG(PHI(r,n)))*kron(r,s)




      MIN1(r,s)=IM(r,s)+(0.00,1.00)*tau*Gp(r,s)/2.0
      MIN2(r,s)=IM(r,s)-(0.00,1.00)*tau*Gp(r,s)/2.0




    END DO
  END DO
```

```
   !Invert the first term
    CALL TRIINV (MIN1,num_pts,INV1)



   !The propogator
    prop=MATMUL(INV1,MIN2)

    DO i=1,num_pts
       DO j=1,num_pts
       prop2(i,j) = EXP((0.0,-1.0)* tau *Hp(i,j))


       END DO
    END DO


     prop=MATMUL(prop,prop2)

!The Workhorse
  DO n=1,TERM-1

    !Putting it all together and propagating, PHI(n+1)=prop*PHI(n)
    PHI2=MATMUL(prop,PHI1)




     !renormalize every  iteration of time

     normt=0.0

     DO i=1,num_pts

       normt= normt+ PHI2(i) * CONJG(PHI2(i)) * h
     END DO
```

```
      PRINT *, "normt: ", normt

      DO i=1,num_pts
        PHI2(i)=((normt)**(-.5))*PHI2(i)
        !PHI(i)=((norm/normt)**(.5))*PHI(i)

      END DO


  PHI1=PHI2



  PRINT *, "Mu", mu(PHI2,Gp+Hp,num_pts,h)
  PRINT *, "kinen", kinen(PHI2,num_pts,h)

! End Workhorse
  END DO




  DO i=1,num_pts
    PHISQ1(i)=PHI2(i)*CONJG(PHI2(i))
  END DO




  WRITE(*,*) "And Phi Squared, final time..."

  DO i=1,num_pts
    PRINT *, PHISQ1(i)
  END DO



END PROGRAM
```

```
!Kronecker Delta Function REAL FUNCTION kron(i,j)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: i,j

  IF (i==j) THEN
    kron = 1.0
  ELSE
    kron = 0.0
  END IF
END FUNCTION kron



REAL FUNCTION mu(M,G,N,h)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: N
  REAL, EXTERNAL :: kron, sqinter
  INTEGER :: i,j
  COMPLEX :: carrier
  COMPLEX, INTENT(IN), DIMENSION(N) :: M
  COMPLEX, DIMENSION(N,1) :: dimmer1, dimmer2, secterm
  REAL, INTENT(IN) :: h
  COMPLEX, INTENT(IN), DIMENSION (N,N)  :: G
  COMPLEX, DIMENSION(N,N) :: E

  DO i=1,N
    dimmer1(i,1)=CONJG(M(i))
    dimmer2(i,1)=M(i)
    DO j=1,N
      E(i,j)=G(i,j)
    END DO
  END DO



  secterm=MATMUL(E,dimmer2)

  carrier=0
```

```
  DO i=1,N
    carrier=carrier+dimmer1(i,1)*secterm(i,1)
  END DO

  mu=carrier*h/sqinter(M,N,h)
  !mu=carrier*h

END FUNCTION mu

REAL FUNCTION kinen(M,N,h)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: N
  REAL, EXTERNAL :: kron, sqinter
  INTEGER :: i,j
  COMPLEX :: carrier
  COMPLEX, INTENT(IN), DIMENSION(N) :: M
  COMPLEX, DIMENSION(N,1) :: dimmer1, dimmer2
  REAL, INTENT(IN) :: h
  COMPLEX, DIMENSION(N,N) :: E,D2

  DO i=1,N
    dimmer1(i,1)=CONJG(M(i))
    dimmer2(i,1)=M(i)
  END DO

  DO i=1,N
    DO j=1,N
      D2(i,j)= (kron(i,j-1)-2.0*kron(i,j)+kron(i,j+1))/(h**2)
    END DO
   END DO


  dimmer1=MATMUL(D2,dimmer1)


  carrier=0.0
  DO i=1,N
    carrier=carrier+dimmer1(i,1)*dimmer2(i,1)
  END DO
```

```
  kinen=carrier*h


END FUNCTION kinen




!Square-Integrator using Simpson's Rule
REAL FUNCTION sqinter(M,N,h)
  IMPLICIT NONE


  INTEGER, INTENT(IN) :: N
  INTEGER :: i
  COMPLEX, INTENT(IN), DIMENSION (N) :: M
  REAL, INTENT(IN) :: h
  REAL :: S0,S1,S2,S
!
  S  = 0.0
  S0 = 0.0
  S1 = 0.0
  S2 = 0.0
  DO I = 2, N-1, 2
    S1 = S1+M(I-1)*CONJG(M(I-1))
    S0 = S0+M(I)*CONJG(M(I))
    S2 = S2+M(I+1)*CONJG(M(I+1))
  END DO
  S = H*(S1+4.0*S0+S2)/3.0

! ! If N is even, add the last slice separately

! NOTE: In this program N is programed to be odd

! So this isn't necessary. !
  IF (MOD(N,2).EQ.0) S = S &
     +H*(5.0*M(N)*CONJG(M(N)) &
     +8.0*M(N-1)*CONJG(M(N-1)) &
     -M(N-2)*CONJG(M(N-2)))/12.0
```

```
 sqinter=S


END FUNCTION sqinter



SUBROUTINE TRIINV (A,N,X)
  IMPLICIT NONE

  REAL, EXTERNAL :: kron
  COMPLEX, INTENT (IN), DIMENSION (N,N) :: A
  COMPLEX, DIMENSION(N,N) :: COP
  INTEGER, INTENT (IN) :: N
  COMPLEX, INTENT (OUT), DIMENSION (N,N):: X
  COMPLEX, DIMENSION (N,N) :: B
  INTEGER :: i,j
  COMPLEX :: temp

  !Initialize B to the identity matrix
  !Copy A in to COP
  DO i=1,N
    DO j=1,N
      B(i,j)=kron(i,j)
      COP(i,j)=A(i,j)
    END DO
  END DO




  DO i=1,N
    IF(i/=N) THEN

      temp=COP(i,i)
      DO j=1,i
        B(i,j)=B(i,j)/temp
      END DO
      DO j=i,i+1
```

```
      COP(i,j)=COP(i,j)/temp
    END DO

    temp=COP(i+1,i)
    DO j=1,i
      B(i+1,j)=B(i+1,j)-B(i,j)*temp
    END DO
    DO j=i,i+1
      COP(i+1,j)=COP(i+1,j)-COP(i,j)*temp
    END DO

  ELSE
    temp=COP(N,N)
    DO j=1,N
      B(N,j)=B(N,j)/temp
    END DO
    COP(N,N)=1
  END IF

END DO



DO i=N,1,-1
  IF(i/=1) THEN


    temp=COP(i-1,i)
    DO j=1,N
      B(i-1,j)=B(i-1,j)-B(i,j)*temp
    END DO
    DO j=1,N
      COP(i-1,j)=COP(i-1,j)-COP(i,j)*temp
    END DO
  END IF

END DO



DO i=1,N
```

```
    DO j=1,N
      X(i,j)=B(i,j)
    END DO
  END DO


END SUBROUTINE TRIINV
```