Investigation of the Time Dependent Schrödinger-Newton Equation

By

PETER JAY SALZMAN
B.S. (San Francisco State University) 1996
M.S. (University of California at Davis) 2000

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Physics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Steve Carlip, chair — Date

Richard Scalettar — Date

Andreas Albrecht — Date

Committee in charge

2005

Investigation of the Time Dependent Schrödinger-Newton Equation

Peter Jay Salzman
September 2005
Physics

Investigation of the Time Dependent Schrödinger-Newton Equation

Abstract

The Schrödinger-Newton equation is a highly non-linear integro-partial differential equation that arises from both the theory of semiclassical gravity of Møller and Rosenfeld and as a possible mechanism to explain why macroscopic objects are not observed to be in states of superposition.

In this dissertation, the evolution of the discretized time dependent Schrödinger-Newton equation with a Gaussian initial condition was obtained for various masses, and its behavior was studied at length.

The scaling properties of the equation were used to generalize the solutions, and using these generalized solutions, a matter-wave experiment is proposed that may be used to test the validity of the theory of semiclassical gravity.

Peter Jay Salzman
September 2005
Physics

Investigation of the Time Dependent Schrödinger-Newton Equation

Abstract

The Schrödinger-Newton equation is a highly non-linear integro-partial differential equation that arises from both the theory of semiclassical gravity of Møller and Rosenfeld and as a possible mechanism to explain why macroscopic objects are not observed to be in states of superposition.

In this dissertation, the evolution of the discretized time dependent Schrödinger-Newton equation with a Gaussian initial condition was obtained for various masses, and its behavior was studied at length.

The scaling properties of the equation were used to generalize the solutions, and using these generalized solutions, a matter-wave experiment is proposed that may be used to test the validity of the theory of semiclassical gravity.

Acknowledgments

- The Department of Energy for grant DE-FG02-91ER40674 which helped support this research.

- Rhonda Frances Salzman, Steve Carlip, Richard Scalettar, Andy Albrecht and John Burke for reading, commenting, and editing this dissertation.

- Rhonda Frances Salzman for programming help when my code didn't work, keeping a cool head when my research wasn't working out, and for giving me strength as I carried the One Ring up to the precipice of Mount Doom.

- John Clauser and Markus Arndt who graciously answered emailed questions. You obviously spent much time on your response, and it was very appreciated.

- Steve Carlip, Richard Scalettar, Jim VanMeter, Rhonda Frances Salzman and John Burke for letting me bounce ideas off of them.

- The UCD Physics Department Graduate Program Coordinators of past, Lynn Rabena, and present, Robyn Tornay who kept on top of my "paperwork" and worked miracles when it became past due. Thanks for taking care of me!

- I've heard horror stories about the staff at Graduate Studies at other schools, but the staff at UCD have been nothing but kind, efficient and pleasant. I feel lucky to have had them during my graduate school tenure.

- Susan Lea, John Burke, Steve Carlip, and Richard Scalettar for a phenomenal education. My sole intention in going to school was to learn about subjects I find personally interesting and stimulating; the degrees were incidental. You have all made a huge impact on my life.

- Kip S. Thorne, Miguel Alcubierre, S. V. Krasnikov, and Gene Roddenberry for inspiring me to study gravitation.

- My wife Rhonda Frances Salzman and John Burke for being my best friends.

- My loving family, who can mercifully now stop asking me if it's done yet.

- My cat Geordi for staying up and keeping me company. He only wanted a head scratch and a little food, and was willing to stay up till insane hours while I worked.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Schrödinger-Newton equation, first proposed by Diosi [1] in 1984, is a single particle Schrödinger equation that can be interpreted as describing a self-gravitating particle. A number of studies of the time-independent equation have been carried out, both numerical [2, 3], and analytical [4]. A fair amount is known about its asymptotic behavior and the lowest eigenfunctions and eigenvalues. However, since the equation is nonlinear, new solutions cannot be obtained by superposition, and its time evolution remains largely unknown.

In this thesis, I report my investigation and numerical solutions of the time dependent Schrödinger-Newton equation, and in doing so, propose a matter-wave diffraction experiment which can be performed to test the validity of the semiclassical theory of gravity.

There are essentially two main avenues of research that lead to the Schrödinger-

Newton equation:

1. one of many attempts to develop a quantum theory of gravity.

2. one of many attempts at obviating the need for a wave packet reduction postulate in quantum mechanics.

These two paths to the equation are unrelated, and the fact that they lead to the same equation is somewhat of an accident. As a background for the reader, I will discuss each one briefly.

1.1 The Problem of Quantizing Gravity

Despite the fact that some of the most capable minds physics has to offer have been working on the problem of quantizing gravity, we are not much closer to a solution than when the programme first started, back around 1935. In light of the difficulty of quantizing general relativity, it is prudent to ask whether the problem can be avoided altogether. As a theory of the geometry of spacetime, general relativity is conceptually quite different from other field theories. For example, while quantum field theories generally live in a fixed background (like the Minkowski spacetime), general relativity is a theory of the dynamics of the background. Whereas we think quantum gravity would necessarily require commutation relations on the metric tensor, causality becomes poorly defined once we abandon the notion of a classical metric. In view of these difficulties, one might be tempted to say that perhaps gravity is inherently classical, and should not be quantized at all. Yet this can not be the whole story. Spacetime is directly coupled to matter sources via the Einstein field equations themselves, but quantum mechanics applies to these very sources. Perhaps the correct approach is not to quantize gravity, but to keep it classical and couple it to quantized matter sources.

The simplest model of classical general relativity coupled to quantum matter, "semiclassical gravity", was proposed in the early 1960's by Møller [5] and Rosenfeld [6]. In this theory, the Einstein field equations become:

$$G_{\mu\nu} \;=\; \frac{8\pi G}{c^4}\,\langle\psi|T_{\mu\nu}|\psi\rangle \tag{1.1}$$

where the operator-valued stress-energy tensor of matter is replaced by an expectation value. As a Hartree-like approximation to quantized gravity, such a system certainly makes sense. But as Kibble and Randjbar-Daemi noted [7], viewed as a fundamental theory, such a model leads to significant nonlinearities in quantum mechanics: the Schrödinger equation for the wave function $|\psi\rangle$ depends on the spacetime metric $g_{\mu\nu}$, which in turn depends on $|\psi\rangle$.

Although Møller [5] and Rosenfeld [6] first proposed the semiclassical programme as a method of coupling gravity to quantum fields in a rather vague fashion, there is a formal basis for eq. (1.1). The linear Schrödinger equation can, of course, be derived from an action principle. The normalization constraint can be built into the action via a Lagrange multiplier:

$$S_q[\psi,\lambda] \;=\; \int dt \Big\{\, \mathrm{Im}\,\langle\dot\psi(t)|\psi(t)\rangle \;-\; E(\psi(t)) + \lambda(t)\;\left[\langle\psi(t)|\psi(t)\rangle \;-\; 1\right]\Big\}$$

where $E = \langle\psi(t)|\hat{H}|\psi(t)\rangle$ is the energy expectation. Variation of the parameter yields the equation of constraint on normalization:

$$\frac{\delta S_q}{\delta\lambda} = \langle\psi(t)|\psi(t)\rangle - 1 = 0$$

and variation of the wavefunction gives the Schrödinger equation in the form of:

$$\frac{\delta S_q}{\delta\psi} = i\,|\dot{\psi}(t)\,\rangle - \hat{H}\,|\psi(t)\,\rangle + \lambda(t)\,|\psi(t)\,\rangle = 0$$

The fact that λ itself is indeterminate is related to the invariance of S_q under choice of phase. The vacuum Einstein field equations are also derivable from an action principle with an action of:

$$S_g[g] = \frac{c^4}{16\pi G}\int d^4x\sqrt{-g}\,R$$

The simplest possible way to combine quantum mechanics and general relativity is to combine their two actions:

$$S_{qg}[\psi,\lambda,g] = S_q[\psi,\lambda] + S_g[g] \tag{1.2}$$

The independent variables of the resultant action are the metric, state vector, and Lagrange multiplier.

Variation of eq. (1.2) with respect to λ and $|\psi(t)\,\rangle$ gives the constraint and evolution equations again, but this time with a metric-dependent Hamiltonian. Variation with respect to $g_{\mu\nu}$ yields eq. (1.1). The details are difficult to see, but the interested reader is referred to Kibble and Randjbar-Daemi [7] or Kibble [8].

There have been a number of arguments against the semiclassical programme, eq. (1.1). Some of them, while appearing quite convincing at first blush, are actually not as compelling as they first seem:

- In 1981, Duff [9] showed that any theory containing both classical and quantum fields has the unfortunate property that classically equivalent theories which are related only by transformation of variables lead to inequivalent semiclassical theories. However, as Kibble points out [8], semiclassical gravity picks out gravity as a distinguished field, so it may not be unreasonable to forbid transformations of the metric based on other fields.

- Gisin [10] showed, under broad circumstances, that the addition of deterministic non-linearities into Schrödinger's equation leads to causality violations. He showed that a spatially separated entangled quantum system, like an EPR pair, evolving under such an equation can be used to send superluminal messages. Measurements on one half of the system can be immediately detected by measurements on the other half of the system. This type of superluminal communication has been dubbed by Polchinski [11] as an 'EPR phone'. However, a number of people including Czachor [12–14], Goldin [15], Jordan [16], and Polchinski [11] have shown that there are several loopholes in Gisin's proof. In particular, they showed that even though a propagator may be nonlinear and nonlocal, to observers, spatially separated entangled systems appear to evolve independently from each other. There are no observable superluminal signals, and therefore, no definitive refutation of eq. (1.1) based on these grounds.

However, other arguments against semiclassical gravity seem to be more serious:

- The Bianchi identity,

$$G^{\mu\nu}{}_{;\nu} \equiv 0$$

implies that the Einstein tensor is conserved at all times. However, if we decompose the state vector as a sum of amplitudes in the Heisenberg picture,

$$|\psi\,\rangle = \sum_i c_i(x^\alpha)\,\psi_i$$

we see that the left hand side of eq. (1.1) is *not* conserved during wavefunction collapse, whereas the right hand side is:

$$\frac{8\pi G}{c^4}\,\langle\psi|T^{\mu\nu}|\psi\rangle_{\,;\nu} = \frac{8\pi G}{c^4}\sum_{i,j}(c_i^* c_j)_{;\nu}\,\langle\psi_i|T^{\mu\nu}|\psi_j\rangle \neq 0 \equiv G^{\mu\nu}{}_{;\nu}$$

This indicates that if we were to accept eq. (1.1) as a correct theory that describes the interface between quantum mechanics and general relativity, we must either give up the standard Copenhagen interpretation or the notion of wavefunction collapse [17].

- If a classical gravitational field can interact with a quantum particle in the manner of eq. (1.1), it is possible to scatter gravitational waves off quantum particles. If this scattering event induces wave packet reduction, Eppley and Hannah show [18] that we must either give up the Heisenberg uncertainty relations or conservation of momentum. They go on to show that if the gravitational wave does not induce wave packet reduction, then we can only save the uncertainty principle if we accept superluminal signalling. This objection has a long history; Bohr and Rosenfeld [19] showed back in 1933 that any theory containing both classical and quantized fields will necessarily violate certain basic properties of any quantum theory, like an uncertainty principle.

- The previous two arguments against eq. (1.1) involved thinking about consequences of wave packet reduction in such a theory. But this is nothing new; the 'R-process' has always been a source of consternation, but as Kibble points out [8] it is even more troubling in a non-linear theory which does not obey the superposition principle, like semiclassical gravity. We can try to sidestep the issue of wave packet reduction by appealing to Everett's 'many worlds' interpretation of quantum mechanics, where there exists a universal wavefunction which never collapses. Unfortunately, this does not help matters much. Lab experiments performed by Page and Geilker [17] have shown that eq. (1.1) is inconsistent with nature if we adopt the many-worlds interpretation of quantum mechanics.

While these objections to semiclassical gravity are fairly serious, none of them is decisive. A key question is whether the quantum nonlinearities in semiclassical gravity are significant. There are three possible answers to this question:

1. The effect is unobservably small.

2. The effect is in obvious conflict with experiment.

3. Gravitational interactions cause wave function collapse, in which case we might want to modify quantum mechanics, not general relativity, as a first step in unification of the two theories.

The last possibility (to make sense) would require that gravitational wave function collapse be a fast process for macroscopic objects but a cosmologically slow process for atomic particles, thus obviating the need for a postulate of wave function reduction. While perhaps unlikely, this mechanism would provide an implementation of Penrose's proposal [20] that gravitational effects are responsible for wave function collapse in quantum theory. Colloquially, this would allow atoms to remain quantum mechanical and cats to remain classical.

As previously mentioned, there are two different but related areas of research that motivate the Schrödinger-Newton equation. The first approach was just discussed: combination of quantum mechanics and general relativity. Penrose's proposal of gravitational wavefunction collapse provides the segue into the second approach: the problem of the measurement process.

1.2 The Problem of the Measurement Process

Penrose [21] (as perhaps every other quotable physicist who ever lived) wrote, a bit dramatically, that quantum mechanics has two bodies of fact in its favor. First, it has a remarkable success with experiment. It agrees with the most delicate and intricate experiments humanity has ever performed. Second, it is a theory of astonishing and profound mathematical beauty. However, quantum mechanics also has one thing against it: it makes absolutely no sense.

Penrose was mainly talking about the axiom of wavefunction collapse. Penrose's problem with wave packet reduction and John S. Bell's search for the "interface of the quantum and classical worlds", both boil down to one thing: the problem of the measurement process. The central problem seems to be that, as it stands, there is no way to formulate quantum mechanics without making reference to "measurements". Yet, as Clifton and Monton smartly comment in [22], surely the word "measurement" is too ambiguous of a concept to be taken as a primitive of a fundamental physical theory.

Popular literature focuses on the non-deterministic nature of quantum mechanics, and Einstein's quip about God and dice is often quoted. However, most of the time, quantum mechanics is a deterministic and unitary theory with no dice or probability at all in the sense that given an initial condition, $|\psi(0)\rangle$, the axioms of quantum mechanics tell us precisely how to perform a time evolution, $\hat{U}(t)\,|\psi(0)\rangle = |\psi(t)\rangle$, in a completely deterministic and unitary manner. Penrose dubs this the U process.

The conceptual problems begin at the measurement process, which cannot be described as a unitary evolution of the wavefunction, so an *ad hoc* axiom is added to the postulates of quantum mechanics known as *reduction of the wave packet*. A "measurement" involves the collapse of a wavefunction into an eigenstate of the operator being measured. Penrose dubs this the R process.

Many papers and books have been devoted to the R process, but to summarize briefly some of the more serious objections to it:

1. It is clearly not relativistically invariant.

2. It paints a strange picture of a system evolving in one of two very incompatible ways: the continuous U process or the discontinuous R process.

3. What should we make of superposed states that blink out of existence?

4. The axioms make no mention about precisely what triggers R or any details about the process. Pragmatically, it makes no difference, as long as the event happens by the time the measurement is made. But we certainly should, if possible, want to understand something as fundamental as the measurement process.

5. The application of quantum mechanics to the macroscopic world leads to patently paradoxical situations, often involving cats and friends of Wigner.

Most people would be glad if the whole notion of wave packet reduction went away, and there is a cottage industry of physicists who try to do away with it.

1.2.1 Quantum Mechanical Spontaneous Localizations and Collapse Dynamics

One class of attempts to explain why quantum superpositions of a macroscopic object are not observed is collectively known as "collapse models" (originally referred to as Quantum Mechanics with Spontaneous Localizations, or QMSL). QMSL's main concern is with "the most embarrassing kind of superposition": a superposition of states describing a single particle, separated by a large region of space. Most explicit and viable collapse models are variations of an early QMSL model, named "GRW Theory" after its principle investigators, Ghirardi, Rimini and Weber. [23, 24]

Consider the state vector of N distinguishable particles in the coordinate representation, $\psi(\vec{\mathbf{q}}_1, \ldots, \vec{\mathbf{q}}_N)$. A localization (known as a "hit") on the i'th particle is defined to mean that ψ is spontaneously multiplied by the thin Gaussian function:

$$G(\vec{\mathbf{q}}_i, \vec{\mathbf{x}}) \;=\; C\, e^{-|\vec{\mathbf{q}}_i - \vec{\mathbf{x}}|^2/2d^2}$$

where d is a fundamental constant of the theory and represents the localization accuracy. The (not yet normalized) wavefunction becomes:

$$\Psi_i(\vec{\mathbf{q}}_1, \ldots, \vec{\mathbf{q}}_N; \vec{\mathbf{x}}) \;=\; C\, \psi(\vec{\mathbf{q}}_1, \ldots, \vec{\mathbf{q}}_N)\, e^{-|\vec{\mathbf{q}}_i - \vec{\mathbf{x}}|^2/2d^2}$$

The localization suppresses single-particle superpositions in which the particle exists at physical locations separated by a distance of at least d. As an example, suppose a single particle wavefunction has a bi-modal distribution ("like an Asian camel") with peaks located near $x = a$ and $x = b$. A hit occuring within within a distance of d of

$x = a$ will result in a wavefunction where most of the probability density is clustered around point a (the Gaussian is very small at point b, so after the localization the wavefunction will be negligible there). On the other hand, if the localization occurs at a point far away from a or b, the resulting wavefunction will be nearly unchanged after normalization.

A localization occurs at a randomly distributed time according to a Poisson distribution with frequency f, which is another fundamental constant of the theory. The *location* of a hit is determined by the probability density $|\psi_i|^2$, so localizations tend to occur wherever the particle is most likely to be found. Thus, if a particle is described by states which are nearly zero everywhere except for two spatially different locations, spontaneous localizations have a tendency to suppress the states which are large at either one location or the other. The net result is that the particle's wavefunction clusters around a single position.

When the fundamental constants are given by $f = 10^{-16}$ s^{-1} and $d = 10^{-5}$ m^{-1}, it is found that microscopic systems undergo localization, on average, every hundred million years while macroscopic system undergo localization every $10^{-7}s$. While this seems very good, there are misgivings with the whole notion of collapse dynamics:

1. We now have yet two more fundamental constants of nature. Physicists are trying to reduce the number of fundamental constants, not increase them.

2. Until recently, attempts at a relativistic generalization of collapse models have met with failure due to untractable divergences. In 2004, Tumulka reports [25]

a covariant formulation of GRW theory, however, this theory is non-local and violates Bell's inequality.

3. Collapse models, at their heart, are phenomenological attempts at solving what really should be a foundations issue. [26] They may or may not solve the measurement problem without hidden variables, so they certainly can not be any *worse* than standard quantum mechanics, however, they make no attempt to explain the rationale behind why the hits occur in the first place.

4. GRW doesn't quite do away with wavepacket reduction. Ideally, G would be a Dirac delta function, and the localization would put the particle in an eigenstate of position. This would do away with the need for wavepacket reduction. However, this also means a spontaneous infinite uncertainty in momentum. Therefore, G cannot be a delta function and wavepacket reduction must still be present in QMSL theory.

5. As Shimony [27] points out, QMSL is tailored for the measurement of a particular object variable, however, this injects a certain anthropocentric element into what should be a fundamental and universal theory.

6. Lewis points out [28] that QMSL requires relaxation of the notion that mutually exclusive events be represented by mutually orthogonal vectors, which requires that we abandon the enumeration principle[1] for macroscopic objects. Ghirardi

[1] If marble 1 is in the box, and if marble 2 is in the box ... and if marble N is in the box, then all N marbles are in the box. More generally, if statement A_1 is true, and A_2 is true, $\cdots$, and A_N is true, then all N statements together are true.

and Bassi respond [29] that Lewis's argument fails, in practice, because the wavefunction he uses to exhibit the violation cannot persist for more than 'a split second'. Clifton and Monton argue [22] that Lewis's attack on QMSL is valid, but the enumeration principle cannot *empirically* fail. Ghirardi and Bassi respond [29] that the analyses of Lewis and Clifton and Monton are flawed because they do not take into account the correct interpretation of a QMSL type theory.

7. It's possible for a localization to occur where the particle's wavefunction density is initially small. In this situation, a particle's wavefunction can spontaneously localize very far from the bulk of the probability density. This is called the "tail problem", and was first stated by Shimony [27] as:

   > [We should] not tolerate "tails" [in a wavefunction] which are so broad that [the] different parts can be discriminated by the senses, even if very low probability amplitude is assigned to the tail.

   Proponents of QMSL say that the low probability of tails forming makes QMSL, for all practical purposes, "good enough". Still, much energy has gone into resolving the tail problem [30]. Ghirardi writes [26] that the tail problem will only be decisively resolved by modifying the standard probabilistic interpretation of quantum mechanics. Clearly, "the jury is still out" on this issue.

1.2.2 Relative States and Many-Worlds Interpretation

Another class of attempts at solving the measurement problem was born out of the doctoral thesis of Hugh Everett, published in 1956. Everett proposed the relative states interpretation of quantum mechanics, which simply dropped the wavepacket reduction postulate from standard quantum mechanics altogether and took the resulting theory to be universally correct. To Everett, the measurement problem amounted to the fact that the observer is distinct from what is being measured in standard quantum theory.

Suppose an apparatus is used to measure L_z for a spin-1/2 particle which is in a superposition of states. The initial state of the observer-particle system is:

$$|0\rangle_{\text{obs}}\Big(\alpha\,|\uparrow\rangle \;+\; \beta\,|\downarrow\rangle\Big)$$

Standard quantum theory says that once the apparatus makes a measurement, one of the particle states collapses, and the resulting system is in one of two states: $|0\rangle_{\text{obs}}|\uparrow\rangle$ with probability $|\alpha|^2$ or $|0\rangle_{\text{obs}}|\downarrow\rangle$ with probability $|\beta|^2$. However, in the relative state formulation, there is only *one* possible state for the system after measurement, the entangled state:

$$\alpha\,|0\rangle_{\text{obs}}|\uparrow\rangle + \beta\,|0\rangle_{\text{obs}}|\downarrow\rangle \tag{1.3}$$

At the end of the measurement, the apparatus can only be specified as an entangled state relative to the state of the particle. Everett does not explain how an observer gets a determinate measurement from a state described by eq. (1.3), and it is not at all clear why such a state would reproduce the predictions of standard quantum theory.

However, there have been serious attempts to recover Born probabilities from the relative states type theories. After all, these theories are attractive in that other than dropping the wave packet reduction postulate, they propose very little change to standard quantum mechanics. For example, Deutsch [31] published a proof that Born-compatible probabilities arise naturally from the non-probabilistic axioms of quantum mechanics along with the non-probabilistic axioms of classical decision theory. In his paper, he envisions a decision maker faced with decisions about the outcome of quantum measurements which have not been made yet. Deutsch shows that a *rational* decision maker makes decisions as if the probabilistic axiom of quantum mechanics is true. In other words, the probabilistic aspect of quantum mechanics is implied by the non-probabilistic axioms. To Deutsch, a rational decision maker is one which abides by the reduced rules of decision theory.[2]

Deutsch's proof has been criticized [32] for containing crucial hidden assumptions which make the argument substansively less convincing (and in fact the authors provide an alternative derivation of quantum probability). For further references on obtaining standard Born probabilities from relative state type theories (either explicitly or implicitly) see [31] as well as [33] which also contains references to papers which are critical of many such attempts.

[2]To avoid circular reasoning, Deutsch uses a *reduced* version of decision theory in which anything that directly or indirectly refers to probability is deleted from the theory.

Everett's theory cannot be considered to be a serious solution to the measurement problem, but it left the opportunity open for many people to interpret his work, and provided a springboard for other theories like many-worlds, many-minds, many-histories, relative-facts, and the Bare theory. The host of no-collapse interpretations which came from Everett's work has a long list of adherents, including Hawking, Feynman, DeWitt, Wheeler, Gell-Mann, and Weinberg.

Many-Worlds Interpretation

The most popular no-collapse theory is the many-worlds interpretation, pioneered by DeWitt in 1971, although there are many variants of MWI itself. DeWitt[3] postulates that in an appropriate basis there exists a world, which is regarded as a distinct macrostate, for each term in a state like eq. (1.3):

$$|\Psi_{\text{universe}}\rangle = \sum_i a_i \, |\phi_{\text{world } i}\rangle \tag{1.4}$$

Each distinct world contains a reality in which the corresponding determinant measurement is made. Thus, for state 1.3, there are two distinct worlds; one in which the observer measures $|0\rangle_{\text{obs}} |\uparrow\rangle$ and another one which measures $|0\rangle_{\text{obs}} |\downarrow\rangle$.

The universe is represented by both a state vector and the set of all dynamical variables. As the dynamical variables evolve in time, the state vector decomposes into orthogonal vectors. Each vector represents a world which receives an Everett term, and that is the world in which the corresponding measurement is made.

[3]DeWitt's variant of MWI is now sometimes called the "many-splitting" interpretation.

Thus, the worlds, which are non-interacting, are considered to be orthogonal parts of a universal wavefunction. This universal wavefunction never collapses, but evolves under the deterministic and unitary U process. It is not regarded as the quantum 'state' of the universe, but rather, as the universe itself.

There are two consequences to this. First, since superpositions of states no longer collapse to a single state, the R process is not needed to explain why a particular eigenvalue was obtained. Second, the observer no longer plays a special role in measurement since we no longer need to worry about what causes R to begin with.

The many-worlds interpretation addresses many of the objections to collapse dynamics theories. In particular, it:

1. Needs no new fundamental constants.
2. Trivially extends to relativistic theories.
3. As opposed to collapse dynamics, it explains, not just models, what happens during the measurement process.

While the many-worlds interpretation contains all the laws of quantum mechanics, modulo wave packet reduction during a quantum measurement, there are still objections to it:

1. Postulating the existence of so many worlds seems to be a violation of Ockham's Razor. One would think that we would need only our world to explain our laboratory observations.

2. Quantum mechanics allows us to compose a universal wavefunction in terms of any basis we choose but many-worlds interpretation theories seem to pick out the preferred basis of world branches, eq. (1.4). However, a lot of work has gone into solving this apparent inconsistency. One avenue of attempts has been to show that the preferred "Everett basis" for the universal wavefunction is environmentally selected through interactions with the environment, or, decoherence. See [33] for an overview and further references on attempts to solve the preferred basis problem of the many-worlds interpretation.

3. There have been numerous proposals for testing the validity of MWI over standard quantum mechanics, On a pragmatic level, we have no idea how to experimentally distinguish a many-worlds interpretation theory from a collapse theory. The most obvious test would be to look for interference effects between worlds containing different macroscopic states.

4. Many-worlds interpretation theories still do not explain the apparent localization of macroscopic objects.

1.2.3 The Role of Gravity in Wave Packet Reduction

Another avenue of approach to solving the measurement problem is to postulate that gravity plays a role in wave packet reduction, which has been investigated by Penrose [20, 21], Diosi [1], Károlyházy [34], and a great many others. The motivation often comes from one aspect of the measurement problem: uncertainty in a position of a macroscopic object's center of mass.

From Newtonian mechanics, a free macroscopic object's center of mass is either stationary or moves uniformly on a straight line. Furthermore, its center of mass always seems to have a well defined position. On the other hand, free particles evolving under Schrödinger's equation are described by wave packets. The free Schrödinger equation is essentially a diffusion equation, and like most diffusion equation solutions, quantum wave packets are never stationary: they spread with time. This spreading leads to increasingly larger uncertainties in the position of the center of mass. That is the paradox of applying quantum mechanics to macroscopic objects.

Considering its remarkable success in the laboratory, it has been asked if there is anything that could be done about this paradox without appreciably changing quantum mechanics. Conceptually, if we were to form a deep and narrow potential for macroscopic objects, their wavefunction would be more or less stationary, with a well defined center of mass. Likewise, for a shallow potential well, the quantum mechanics for microscopic objects would remain essentially unchanged.

Since we want this behavior to occur in absence of external forces, this idea of deep and shallow potential wells must be due to a property of the object itself. Large objects must inherently produce deep wells, and small objects must inherently produce shallow wells. The size of the well is tied to the gravitational mass of the object, so gravitational potential energy is a natural agent for the process. Such a hypothesis is certainly logical: if true, it would explain why quantum theory has been successful when applied to atoms but nonsensical when applied to baseballs.

There are many starting points one could take in applying gravity's role in quantum mechanics. One such starting point would the semiclassical theory of Møller and Rosenfeld:

$$G_{\mu\nu} = \frac{8\pi G}{c^4} \langle\psi|T_{\mu\nu}|\psi\rangle \tag{1.1}$$

In the next section, eq. (1.1) will be the starting point in the derivation of the Schrödinger-Newton equation.

1.3 The Schrödinger-Newton Equation

Whether semiclassical gravity is used to bridge quantum mechanics and general relativity or to explain why macroscopic objects do not appear as superpositions, the full system eq. (1.1) is too complicated to tackle directly. However, much can be learned by looking at the linearized weak-field limit of semiclassical gravity. Simplification of eq. (1.1) proceeds in the standard way with the following assumptions:

1. For weak gravitational fields, we write down a metric for a nearly flat manifold,

   $$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}$$

   where the $h_{\mu\nu} << 1$ are small perturbations.

2. The source of the weak gravitational field is mostly due to mass density, not stress energy:

   $$\frac{|T^{ij}|}{T^{00}} = \frac{|T^{ij}|}{\rho} \ll 1$$

3. Stress energy sources move with $v \ll c$; time derivatives can be dropped.

With these assumptions, eq. (1.1) becomes a linearized theory, the details of which can be found in any text on General Relativity.[4] The result is the familiar Poisson equation:

$$\nabla^2 \phi(\vec{\mathbf{x}}, t) \;=\; 4\pi G \,\langle \psi | m | \psi \rangle \;=\; 4\pi G m |\psi(\vec{\mathbf{x}}, t)|^2$$

With the equally familiar solution:

$$\phi(\vec{\mathbf{x}}, t) \;=\; -Gm \int \frac{|\psi(\vec{\mathbf{x}}', t)|^2}{|\vec{\mathbf{x}} - \vec{\mathbf{x}}'|}\, d^3x' \tag{1.5}$$

The multi-particle Schrödinger equation for N interacting masses is given by:

$$-\sum_{i=1}^{N} \frac{\hbar^2}{2m_i} \nabla_i^2 \psi(\vec{\mathbf{x}}, t) + \sum_{i,j=0}^{N} V_{ij}(|\vec{\mathbf{x}}_i - \vec{\mathbf{x}}_j|)\, \psi(\vec{\mathbf{x}}, t) + \sum_{i=1}^{N} m_i\, \phi(\vec{\mathbf{x}}_i, t)\, \psi(\vec{\mathbf{x}}, t) \;=\; i\hbar \frac{\partial \psi}{\partial t} \tag{1.6}$$

where V_{ij} denotes the interaction potential energy between masses i and j, and ϕ_i gives the potential of mass i, which was obtained via linearizing the semiclassical gravity equation. For this research, the simplest possible system was investigated; the time evolution of a single particle of mass m:

$$-\frac{\hbar^2}{2m} \nabla^2 \psi(\vec{\mathbf{x}}, t) - m\, \phi(\vec{\mathbf{x}}, t)\, \psi(\vec{\mathbf{x}}, t) \;=\; i\hbar \frac{\partial \psi}{\partial t}$$

Using the gravitational potential from eq. (1.5), this equation becomes the Schrödinger-Newton equation:

$$-\frac{\hbar^2}{2m} \nabla^2 \psi(\vec{\mathbf{x}}, t) - Gm^2 \int \frac{|\psi(\vec{\mathbf{x}}', t)|^2}{|\vec{\mathbf{x}} - \vec{\mathbf{x}}'|}\, d^3x'\; \psi(\vec{\mathbf{x}}, t) \;=\; i\hbar \frac{\partial \psi}{\partial t} \tag{1.7}$$

[4] For example, [35], section 17.4

A few words should be said about this equation. It is a non-linear single-particle Schrödinger equation, often called the Schrödinger-Newton equation, and describes the evolution of a non-local and self-interacting particle of mass m. The particle itself can be visualized as being "smeared", superposed over a probability density, $\rho(\vec{\mathbf{x}},t) = m|\psi(\vec{\mathbf{x}},t)|^2$. Although the idea is unorthodox, it certainly makes sense; the mass is distributed in space according to a weighted average of where the particle is most likely to be found.

As a final note, the time-independent Schrödinger-Newton equation has very interesting similarities to the Hartree equation[5] of electronic structure theory. A large amount of effort has gone into studying the Hartree equation, and the machinery used to solve it (stochastic methods, Green's functions, Monte Carlo, etc.) may be relevant to future study of the many-body Schrödinger-Newton equation, which is a logical avenue for future research.

1.4 Analytic Properties

Before describing the numerical work, we first consider a few analytic properties of the Schrödinger-Newton equation.

The Schrödinger-Newton equation modifies the quantum evolution by postulating that particle wavefunctions couple to gravitational potential, which is a reasonable thing to expect. This coupling leads to a non-linear Schrödinger equation, but these

[5]The Hartree equation approximates a many-body Schrödinger equation with a single particle equation in which the single particle sees the average potential of all the other particles.

non-linearities do not disturb the interpretation of $|\psi(\vec{\mathbf{x}},t)|^2$ as a probability density. For instance, an ordinary probability current can be written down:

$$\frac{\partial|\psi|^2}{\partial t} \;=\; \nabla\cdot\left[\frac{i\hbar}{2m}\left(\psi^*\nabla\psi-\psi\nabla\psi^*\right)\right]$$

And preservation of normalization is built directly into the semiclassical action, eq. (1.2):

$$\frac{d}{dt}\int|\psi(\vec{\mathbf{x}},t)|^2\,d^3x \;=\; 0$$

As a consequence, the expectation of the momentum operator is conserved:

$$\frac{d}{dt}\,\langle\psi|\hat{p}|\psi\rangle \;=\; \frac{d}{dt}\int\psi^*(\vec{\mathbf{x}},t)\,(-i\hbar\nabla)\,\psi(\vec{\mathbf{x}},t)\,d^3x \;=\; 0$$

as is the expectation of the energy operator:

$$\frac{d}{dt}\,\langle\psi|\hat{E}|\psi\rangle \;=\; \frac{d}{dt}\int\psi^*(\vec{\mathbf{x}},t)\left(-\frac{\hbar^2}{2m}\nabla^2-Gm^2\int\frac{|\psi(\vec{\mathbf{x}},t)|^2}{|\vec{\mathbf{x}}-\vec{\mathbf{x}}'|}\,d^3x\right)\psi(\vec{\mathbf{x}},t)\,d^3x \;=\; 0$$

Next consider the effect of rescaling coordinates. As discussed in Appendix A, the Schrödinger-Newton equation is invariant under the transformation:

$$m\to\mu m,\qquad \vec{\mathbf{x}}\to\mu^{-3}\vec{\mathbf{x}},\qquad t\to\mu^{-5}t,\qquad \psi\to\mu^{9/2}\psi \tag{1.8}$$

where μ is an arbitrary constant. In particular, consider an initial Gaussian wave function

$$\psi(r,0)=\left(\frac{\alpha}{\pi}\right)^{3/4}e^{-\alpha r^2/2} \tag{1.9}$$

of width $\alpha^{-1/2}$. One might expect a two-parameter family of solutions, determined by α and m. However, the scaling eq. (1.8) implies that if $\psi(\alpha,m;\vec{\mathbf{x}},t)$ is a solution, so is:

$$\mu^{9/2}\,\psi(\mu^6\alpha,\,\mu;\,\mu^{-3}\vec{\mathbf{x}},\,\mu^{-5}t)$$

Thus it is sufficient to consider a one-parameter family of solutions. We can choose our parameter as follows. Up to factors of order one, the initial gravitational energy of a Gaussian wave packet eq. (1.9) is

$$\bar{V} = Gm^2\alpha^{1/2}$$

while the initial kinetic energy is

$$\bar{E} = \frac{\hbar^2\alpha}{m} \tag{1.10}$$

The ratio $\bar{V}/\bar{E}$ is independent of the scaling parameter μ. Suppose, for instance, that wave functions really do "collapse" for certain ranges of parameters. Let $T_{\rm coll}$ be the characteristic collapse time. If we choose an arbitrary fixed reference mass m_0 and define the arbitrary scale factor μ such that $m_0 = \mu\, m$, then it follows from eq. (1.8) that

$$T_{\rm coll}(m,\bar{V}/\bar{E}) = \left(\frac{m}{m_0}\right)^5 T_{\rm coll}(m_0,\bar{V}/\bar{E}) \tag{1.11}$$

and hence that $(m_0/m)^5\,T_{\rm coll}(m,\bar{V}/\bar{E})$ depends only on the single "collapse" parameter $\bar{V}/\bar{E}$.

1.4.1 Strength of Self-Interaction

A different type of intuition needs to be used when dealing with non-linear equations such as eq. (1.7). For instance, increasing the mass of any standard linear

Schrödinger equation changes the time scale, not the physics, of the problem. The non-linearities here mean that changing the mass can have a profound effect on the nature of the equation's solutions. In other words, small changes in parameter do not necessarily lead to small changes in the solution.

Scaling eq. (1.7) into dimensionless quantities can be used to gain insight into the possible domains where the non-linearities become important. Consider eq. (1.7) with a spherically symmetric initial condition so that all angular derivatives vanish and the wavefunction is a function of r alone. The wavefunction can be written as a sum of Fourier components:

$$\psi \;=\; \int_0^\infty A(k,t)\, j_0(kr)\, dk$$

where $j_0(kr)$ is a spherical Bessel function of order zero. The potential can be expanded using the standard expansion[6] for $1/R$:

$$\int \frac{|\psi(\vec{\mathbf{r}}',t)|^2|}{|\vec{\mathbf{r}}-\vec{\mathbf{r}}'|}\, d^3r' \;=\; \int |\psi(\vec{\mathbf{r}}',t)|^2 \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} \frac{4\pi}{2\ell+1}\frac{r_<^\ell}{r_>^{\ell+1}}\, Y_\ell^m(\theta,\phi)\, Y_\ell^{m*}(\theta',\phi')\, d^3r'$$

where $r_> = \max(r,\, r')$ and $r_< = \min(r,\, r')$. Using the orthogonality of spherical Bessel functions and spherical harmonics, and the following scalings:

$$r = \frac{Gm}{c^2}\rho \qquad k = \frac{c^2}{Gm}\alpha \qquad t = \frac{2G^2m^3}{\hbar c^4}\tau \qquad A(k,t) = \sqrt{\frac{c^2}{Gm}}a(\alpha,\tau)$$

[6]See [36], eq. (3.70), pg. 102.

we arrive[7] at the following dimensionless version of eq. (1.7):

$$\left(i\frac{\partial}{\partial\tau} - \alpha'^2\right) a(\alpha',\tau) \; = \\ -\,\epsilon^2\alpha'^2 \int_\alpha a(\alpha,\tau) \int_\rho j_0(\alpha\rho) j_0(\alpha'\rho) \int_{\alpha''} a^*(\alpha'',\tau) \int_{\alpha'''} a(\alpha''',\tau) \\ \times \int_{\rho'} j_0(\alpha''\rho') j_0(\alpha'''\rho') \frac{\rho'^2}{\rho_>}\, d\rho'\, d\alpha'''\, d\alpha''\, d\rho\, d\alpha \quad (1.12)$$

The left side of eq. (1.12) is the scaled free particle Schrödinger equation. The right hand side represents the gravitational potential. The parameter

$$\epsilon^2 = \left(\frac{4Gm^2}{\hbar c}\right)^2$$

is a measure of the potential strength.

For $\epsilon^2 >> 1$, the particle is strongly self-interacting, and for $\epsilon^2 << 1$ the particle is very nearly free of gravitational non-linearities. For an electron, $\epsilon^2 \sim 10^{-90}$. This might be an indication that semiclassical gravity predicts that electrons remain quantum mechanical entities, free of self-localization. On the other hand, for a .1 kg baseball, $\epsilon^2 \sim 10^{31}$. This may be an indication that a baseball's wavefunction quickly collapses in its own gravitational potential well. Setting $\epsilon \equiv 1$ to see what order of mass is the "border" between having negligible and non-negligible self-gravitational effects, it is found that this border mass is about 10^{-8} kg, which is the Planck mass.

While the fact that the Planck mass should be some kind of border between quantum mechanical and classical behavior is hardly surprising, it should be noted

[7] The details of this calculation are longer than anybody, including me, should care to see. But I will provide them upon request.

that the width of the particle also plays a role in any possible collapse dynamics. However, it is not immediately clear how to translate a wavefunction's "width" (for example, the parameter α in eq. (1.9)) in terms of a macroscopic property. Should it be the width of a baseball? The baseball's Compton's wavelength? The uncertainty in the position of the surface electrons that make up the baseball?

1.5 Matter-Wave Experiments

The proposed test of semiclassical gravity depends on results obtained via matter-wave diffraction or interference, so some time will be spent describing these types of experiments and their history. The interested reader is urged to read the cited references for further information. After discussing matter-wave experiments in general and their history, I will discuss what appears to be the most popular way to conduct the experiment: using a Talbot-Lau interferometer.

In 1924, Louis de Broglie predicted that matter exhibits wavelike properties at a wavelength inversely proportional to the mass's momentum. de Broglie's matter-wave duality hypothesis was experimentally demonstrated by the electron diffraction experiments of Clinton Davisson and Lester Germer [37] in 1927 and by the neutron diffraction experiments of Shull and Wollan [38] in 1948.

But de Broglie's prediction wasn't just for microscopic masses; the prediction was for *all* masses. Despite this, the naked-eye world is unequivocally classical. Clearly, the boundary between the microscopic and macroscopic worlds occurs somewhere within the 31 orders of magnitude between the masses of an electron and a rock. A natural question to ask is: where is this boundary between the classical and quantum mechanical worlds?

Since quantum mechanics works exceedingly well for microscopic objects, a pragmatic approach to finding this boundary is by performing experiments which demonstrate diffraction or interference of ever increasing masses. In the simplest experiment of this type, one would direct a beam of masses towards a Young's two slit barrier with one slit covered. The resulting image would be, more or less, a dot. To demonstrate the wavelike property of the mass under investigation, one would simply need to uncover the other slit to reveal interference.

However, this programme is easier said than done. Such an experiment cannot be done for a rock or even a virus, for the de Broglie wavelength will be far too small; interference fringes would be much too close together to be measurable by any type of technology available in the foreseeable future. Additionally, the slit spacing appropriate for such small wavelengths would be difficult if not impossible to build. Lastly, decoherence becomes a serious limitation for such an experiment as the mass becomes larger. The larger an object is, the more likely its state gets entangled with the environment, causing a loss of coherence.

Much experimental and theoretical effort by extremely ingenious researchers has gone into devising experiments that circumvent the problems of decoherence and detection of waves with such small wavelength. The history of matter-wave experiments is documented [39], but in the 1990's saw, for the first time, such experiments performed on entire atoms and molecules.

- In 1991, Oliver Carnal and Jürgen Mlynek [40] used a Young's double slit barrier to demonstrate diffraction of entire Helium atoms.
- In that same year, Pritchard et. al. [41] used a Mach-Zehnder interferometer to demonstrate interference fringes from Sodium atoms.
- In 1994 Clauser and Li [42] used a similar interferometer, called a Talbot-Lau interferometer, to demonstrate matter-wave diffraction of Potassium atoms.
- In that same year, Schöllkopf and Toennies [43] demonstrated diffraction of Helium dimers and diatomic molecules like H_2 and D_2. Although this experiment didn't break the mass record, it did demonstrate diffraction for more complicated structures than previously demonstrated.
- In 1999, Nairz, Arndt, and Zeilinger demonstrated diffraction for the fullerenes C_{60} and C_{70} [39] using a nano-fabricated diffraction grating.
- In 2002, Zeilinger et al. used a Talbot-Lau interferometer to redo the diffraction experiments of C_{60} and C_{70} [44].
- In 2003, Zeilinger et al. used a Talbot-Lau interferometer to demonstrate the wave properties of fluorofullerene $C_{60}F_{48}$ and tetraphenylporphyrin (TTP) $C_{44}H_{30}N_4$. At 1632 amu and 108 covalently bonded atoms, $C_{60}F_{48}$ is, so far, the most massive and complicated molecule to exhibit wavelike properties. At 614 amu, tetraphenylporphyrin is not the most massive molecule to display

wavelike properties, however, it is a "biomolecule" that can be found in biosystems containing, for example, chlorophyll and hemoglobin. Additionally, some physicists had argued that only highly symmetric or even spherically symmetric molecules would exhibit interference. TPP, being roughly planar and twice as broad as C_{60} proved such claims were wrong.

The future of matter-wave diffraction appears exciting. Researchers agree that the Talbot-Lau interferometer is extremely well suited for these types of experiments. There is talk of using this apparatus for matter-wave experiments involving proteins, live viruses, and nano-crystals. [45, 46] To date, the experiments have shown that there may be no fixed boundary between the classical and quantum mechanical worlds. Making the transition between the two different types of behavior may only be a function of the exchange of path information between the system and its environment. Arndt and Zeilinger write [45],

> Extrapolating our results to bigger masses and higher temperatures, we believe that neither collisions nor thermal decoherence will be a problem in these cases. No fundamental limit of quantum interference is yet in sight...

which is good news for this thesis. In chapter 4, I will show that such an experiment can be used to test the semiclassical theory of gravitation.

1.6 Talbot-Lau Interferometry

There are many excellent references on Talbot-Lau interferometry [44, 46, 47] by experts in the field, so no attempt will be made to provide an exhaustive treatment of this device. Instead, a very simplistic overview will be given to acquaint the unfamiliar reader for the purposes of this dissertation.

When one holds a diffraction grating up to monochromatic light and places a magnifying glass directly over the grating, a clear and sharp image of the grating can be seen through the magnifier. If the magnifier is moved away from the grating, the image of the grating goes out of focus. However, as the distance increases further, it will be found that the image of the grating will come into sharp focus at some distance z_T, called the Talbot distance. Rayleigh showed that this distance is $z_T = d^2/\lambda$ where d is the grating constant (space between grating slits) and λ is the wavelength of the light impinging the diffraction grating. This aliased image of the diffraction grating can be illustrated with a straightforward calculation using the Kirchhoff-Fresnel theory of light, applied to de Broglie matter-waves.

A Talbot-Lau interferometer consists of a wave source, three diffraction gratings (named g_1, g_2, and g_3), and a detector, as shown in Figure 1.6.

The overall gist of the interferometer is as follows:

- The wave source produces the type of waves desired, here, de Broglie matter-

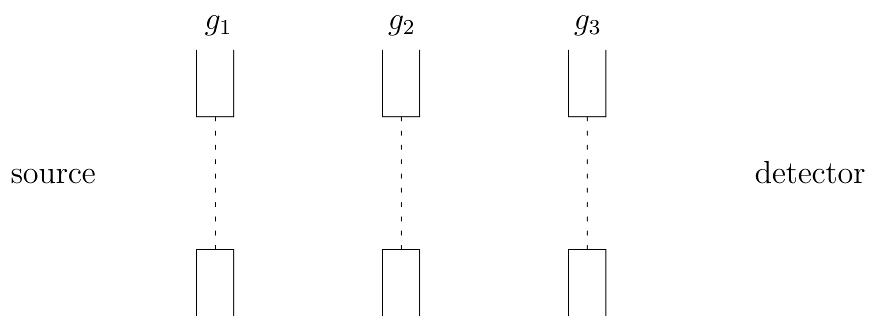


Figure 1.1: Talbot-Lau interferometer setup. As particles leave the source (to the left of g_1), they pass through three diffraction gratings, labelled g_1, g_2, and g_3. The resulting interference is detected by an apparatus situated to the right of g_3.

waves. Unlike many other types of interferometers, the beam from the source does not need to be precisely collimated: it can contain waves with wave vectors pointing in different directions. This is an advantage over other types of optical devices that require spatially coherent waves.

- The beams impinge upon g_1. From Huygen's Principle, each slit in grating g_1 acts as point wave source. The purpose of g_1 is to prepare the required coherence for the next grating. In this way, g_1 can be considered to be the "source" of the interferometer.

- Each slit of g_1 acts as an independent wave source for g_2. By the Talbot-Lau effect, an image of g_2 forms at the Talbot distance.

- To demonstrate interference, g_3 (which has the same period as the expected interference) is placed at the Talbot distance so that the diffraction image of g_2

forms on its surface. g_3 can be moved laterally to either block the interference fringes or allow them to pass through.

- By shifting the lateral position of g_3, a detector at the end of the interferometer should be able to detect an alternating increase and decrease of the number of transmitted molecules.

- The entire apparatus is placed in an evacuated chamber to limit decoherence effects.

There are numerous advantages of the Talbot-Lau interferometer over other interferometers and methods to detect atomic and molecular de Broglie waves which are discussed in depth in the literature. [42, 44–47] Some of these advantages are given by:

- The Talbot-Lau interferometer can accept a large solid angle beam from the wave source. Not only does this mean a relaxed restriction on collimating apparatus, it also leads to stronger interference signals from a weaker wave source.

- The de Broglie wavelength is inversely proportional to mass, so as we test larger masses, the wavelength becomes smaller. It's an engineering challenge to manufacture diffraction gratings with a grating constant that can accommodate these waves. In addition, other issues become important with decreased size, such as van der Waal attraction between the particle and the grating. However, Talbot-Lau interferometers require a grating period that scales as the square root of the wavelength whereas other optical devices require gratings that scale linearly with the wavelength. This means diffraction gratings for Talbot-Lau interferometers are larger.

- Talbot-Lau interferometers rely on near field effects, so are generally small and rugged compared with far-field interferometers.

In fact, the advantages of near-field interferometry are so compelling, that is has even been suggested to perform matter-wave experiments with live viruses. [46, 48]

Chapter 2

Solving the Equation Numerically

This chapter will discuss how the Schrödinger-Newton equation was discretized and numerically solved, and the complications that arose along the way.

2.1 Statement of the Numerical Problem

The Schrödinger-Newton equation, eq. (1.7), is considered for the case of a self-gravitating mass m centered at the origin. The initial condition is a Gaussian of parameter α:

$$\psi(r,0) = \left(\frac{\alpha}{\pi}\right)^{3/4} e^{-\alpha r^2/2} \tag{2.1}$$

Since this equation is rotationally invariant, we expect any solution to retain the spherical symmetry of the initial condition. Thus, in spherical coordinates, ψ will depend only on time and the radial coordinate. The angular derivatives in eq. (1.7) vanish, and we are left with a 1+1 dimensional integro-partial differential equation:

$$-\frac{\hbar^2}{2m}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\psi}{\partial r}\right) \;-\; G\,m^2\,\psi(r,t)\int\frac{|\psi(\vec{\mathbf{r}}',t)|^2}{|\vec{\mathbf{r}}-\vec{\mathbf{r}}'|}\,d^3r' \;=\; i\hbar\frac{\partial\psi}{\partial t}$$

and the potential can be simplified (see appendix B for details on simplifying the potential integral):

$$\begin{aligned} &-\frac{\hbar^2}{2m}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\psi}{\partial r}\right) \\ &-\;4\pi G\,m^2\left(\frac{1}{r}\int_0^r|\psi(\vec{\mathbf{r}}',t)|^2\,r'^2\,dr' + \int_r^\infty|\psi(\vec{\mathbf{r}}',t)|^2 r'\,dr'\right)\psi(r,t) \;=\; i\hbar\frac{\partial\psi}{\partial t} \end{aligned} \tag{2.2}$$

Because the Schrödinger-Newton Hamiltonian is spherically symmetric, it commutes with the Parity operator. Since there is no explicit time dependence, parity is a constant of the motion. The Gaussian initial condition has even parity, so the solution must also have even parity:

$$\psi(\vec{\mathbf{r}},t) \;=\; \psi(-\vec{\mathbf{r}},t)$$

This, along with continuity of the derivative gives the first boundary condition:

$$\operatorname*{limit}_{\vec{\mathbf{r}}\to 0}\frac{d\psi}{dr} \;=\; 0 \tag{2.3}$$

The second boundary condition is a restatement of the fact that the wavefunction must be a square integrable function:

$$\operatorname*{limit}_{\vec{\mathbf{r}}\to\infty}\psi(\vec{\mathbf{r}},t) \;=\; 0 \tag{2.4}$$

Thus, the statement of the numerical problem is:

> Solve eq. (2.2) given the initial condition eq. (2.1) and boundary conditions eq. (2.3) and eq. (2.4) on the domain $r \in [0,\infty)$ for some finite time interval.

2.2 Derivatives at the Polar Origin

Although we are dealing with only 1 spatial coordinate, the "sphericalness" of the geometry cannot be ignored. The r component of the Laplacian expressed in spherical coordinates is:

$$\nabla^2{}_r = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) = \frac{\partial^2}{\partial r^2} + \frac{2}{r}\frac{\partial}{\partial r}$$

This is troublesome for anyone who desires a numerical calculation of things like diffusive fluids, electrostatic potentials or wavefunctions at the origin of a spherical coordinate system, since computers are usually reticent about taking limits involving $1/r$ as $r \to 0$. As the reader knows, the computer is not at fault: the $1/r$ term is not a true singularity, but a coordinate singularity. The Laplacian exists at the origin, but its polar representation is no longer valid there.

Often, when solving a numerical problem such as this, one either converts to Cartesian coordinates or excludes the origin by setting the boundary 'close' and leaving the matter at that.

Turning a 1-D problem into a 3-D problem is not wise, so using to Cartesian coordinates is not an option. In addition, the most interesting things that could come out of this project, like gravitational wavefunction collapse, occur right at the origin, so excluding even the smallest region about the origin is also not an option. As a result, one of the things that needs to be dealt with is how to numerically handle a Laplacian at the polar origin. I developed a method to do this and came up with two nice arguments.[1]

2.2.1 First Argument

Consider the Laplacian operator in the Cartesian basis at the origin:

$$\nabla^2\Lambda = \frac{\partial^2\Lambda}{\partial x^2} + \frac{\partial^2\Lambda}{\partial y^2} + \frac{\partial^2\Lambda}{\partial z^2} \qquad (r = 0)$$

This expression can be approximated by using the difference equation for the 2nd derivative, centered on a sphere of radius Δr, in each dimension:

$$\begin{aligned} \nabla^2\Lambda &= \frac{\Lambda_{dx} - 2\Lambda_0 + \Lambda_{-dx}}{(\,\Delta x\,)^2} + \frac{\Lambda_{dy} - 2\Lambda_0 + \Lambda_{-dy}}{(\,\Delta y\,)^2} + \frac{\Lambda_{dz} - 2\Lambda_0 + \Lambda_{-dz}}{(\,\Delta z\,)^2} \\ &= \frac{\Lambda_{dx} + \Lambda_{-dx} + \Lambda_{+dy} + \Lambda_{-dy} + \Lambda_{dz} + \Lambda_{-dz} - 6\Lambda_0}{(\,\Delta r)^2} \end{aligned} \tag{2.5}$$

where Δx, Δy, and Δz are all radii of the sphere or radius Δr. Rotating the axes gives the same equation. In fact, if we rotate the axes and add the results N times, we get an area average in the limit as N approaches infinity. The area average of eq. (2.5) is:

$$\nabla^2\Lambda = \frac{6\left(\bar{\Lambda} - \Lambda_0\right)}{(\,\Delta r)^2} \tag{2.6}$$

where $\bar{\Lambda}$ is the mean value of Λ on the surface of a sphere. Eq. (2.6) is valid for the Laplacian at a polar origin for a general function Λ. For the special case of spherical symmetry, we have the addition condition that $\bar{\Lambda} = \Lambda_{dx} = \ldots = \Lambda_{-dz}$, which enables us to write:

$$\nabla^2\Lambda = \frac{3\left[\Lambda_{dr} - 2\Lambda_0 + \Lambda_{-dr}\right]}{(\,\Delta r)^2}$$

[1]Acknowledgement goes to Jim VanMeter who helped me finish the first argument.

which is the difference equation for a 2nd order derivative centered at the origin:

$$\nabla^2\Lambda \;=\; 3\frac{\partial^2\Lambda}{\partial r^2} \qquad (r=0) \tag{2.7}$$

2.2.2 Second Argument

A more heuristic method of determining the expression for the Laplacian at the polar origin is to expand the $\partial_r\Lambda(r)$ term of the Laplacian in a Taylor series and consider the limit as r approaches zero.

$$\operatorname*{limit}_{r\to 0}\ \frac{2}{r}\frac{\partial\Lambda}{\partial r} \;=\; \operatorname*{limit}_{r\to 0}\ \frac{2}{r}\left(\left.\frac{\partial\Lambda}{\partial r}\right|_{r=0} + \; r\left.\frac{\partial^2\Lambda}{\partial r^2}\right|_{r=0} + \frac{r^2}{2!}\left.\frac{\partial^3\Lambda}{\partial r^3}\right|_{r=0} + \; \cdots \right) \;=\; 2\frac{\partial^2\Lambda}{\partial r^2}$$

where we made use of the vanishing first derivative at the polar origin due to spherical symmetry. Thus:

$$\nabla^2\Lambda \;=\; \frac{\partial^2\Lambda}{\partial r^2} + \frac{2}{r}\frac{\partial\Lambda}{\partial r} \;=\; 3\frac{\partial^2\Lambda}{\partial r^2} \qquad (r=0)$$

in complete agreement with eq. (2.7).

2.3 Finite Element Convention

The solution to a 1+1 dimensional PDE can be thought of as a surface, since the solution is a function of one space and one time coordinate: $u = u(r,t)$. In discretizing the solution space, we form a 2-dimensional array where each element, `u[j][n]`, represents the solution at a distance $j\,\Delta x$ from the spatial origin and at a time of $n\,\Delta t$ beyond the temporal origin.

It is common to use subscript and superscript indices to indicate the location of a solution element within the solution space. In literature, both u_j^n and u_n^j conventions are used. Throughout this paper, the upper index will be the time index, and the lower index will be the spatial index. Thus, a solution at position $r = j\,\Delta r$ and time $t = n\,\Delta t$ will be denoted as u_j^n, except in code listings, in which case it will be represented as `u[j][n]`.

2.4 Discretizing Schrödinger's Equation: First Steps

This section describes and explains the choice of discretization for eq. (2.2) for the reader who may not be familiar with obtaining PDE solutions numerically.

Basic calculus describes how to approximate first and second order derivatives, and in fact, using a Taylor series we can generate derivative approximations to any order we choose. One can simply 'plug' these approximations into eq. (2.2), being mindful of the expression for the Laplacian at the polar origin, eq. (2.7). With this method, an integro-PDE solver can be written in a very short time. The benefit of such a scheme is that it's an *explicit*[2] algorithm, and therefore, very easy to program and solve.

Unfortunately, most of these schemes can be shown to be unconditionally unstable[3]

[2]An explicit algorithm is one where an unknown quantity is expressed completely in terms of known quantities. On the other hand, implicit schemes are algorithms where unknown quantities are expressed in terms of other unknown quantities. Such algorithms usually involve inversion operators or solving large systems of linear equations.

[3]In numerical parlance, an unstable algorithm is one which has the property that small errors, like truncation and precision roundoff, grow geometrically or worse with each timestep. Left unchecked,

using Von Neumann's stability analysis.[4] Even if the algorithm is stable, this method of differencing is non-unitary; the wavefunction does not keep its normalization. A unitary algorithm is desirable, otherwise the wavefunction needs to be renormalized at each timestep, adding another opportunity for precision loss at every timestep. Therefore, differencing eq. (2.2) using direct Taylor approximations of derivatives is ruled out.

A more tenable approach for differencing Schrödinger's equation can be motivated by writing down its time evolution:

$$\hat{H}\psi \;=\; i\hbar\frac{\partial\psi}{\partial t} \tag{2.8}$$

where $\hat{H}$ is the energy operator in polar coordinates, again assuming spherical symmetry:

$$\hat{H} \;=\; -\frac{\hbar^2}{2m}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\psi}{\partial r}\right) + V\psi$$

The formal solution of eq. (2.8) is given by:

$$\psi(r,\,\Delta t) \;=\; e^{-i\hat{H}\,\Delta t/\hbar}\psi(r,0)$$

and, numerically, we would approximate the exponential by its Taylor series in what

the errors in an unstable algorithm can easily exceed the solution itself or even the largest number the computer can represent internally.

[4]There are many ways to difference a differential equation. Von Neumann's stability analysis is a method to determine the stability of a given numerical algorithm. It is a local method that looks at the behavior of individual Fourier components of the solution under the algorithm in question. If the components grow unconditionally, the algorithm is unstable. For linear equations, Von Neumann's analysis is a sufficient condition for numerical stability. For non-linear equations, it remains a necessary condition for stability. See [49], pgs. 836–837.

is known as the explicit FTCS (forward time, centered space) discretization:[5]

$$\psi_j^{n+1} \;=\; \left(\hat{\mathbb{1}} - \frac{i\hat{H}\,\Delta t}{\hbar}\right)\psi_j^n \tag{2.9}$$

However, Von Neumann's analysis[6] shows that this scheme, like many explicit differencing schemes, is unconditionally unstable.

We could also try an implicit differencing scheme, since implicit schemes tend to be unconditionally stable.[7] The implicit scheme is motivated by translating the future wavefunction back in time:

$$e^{+i\hat{H}\,\Delta t/\hbar}\psi(r,\,\Delta t) \;=\; \psi(r,0)$$

which can be numerically approximated by:

$$\psi_j^{n+1} \;=\; \left(\hat{\mathbb{1}} + \frac{i\hat{H}\,\Delta t}{\hbar}\right)^{-1}\psi_j^n \tag{2.10}$$

This might have been a good choice, however, eq. (2.10) is only 1st order accurate in time because of the truncation of the exponential's Taylor series. We could keep adding higher order terms in Δt to gain whatever accuracy we desire, but the more terms left in the expansion, the more difficult it becomes to generate the inverse operator. In any event, this method still produces a non-unitary wavefunction.

[5] See [49], figure 19.1.1 on pg. 836, and pg. 847.
[6] See [49], pgs. 836 and 847.
[7] See [49] pgs. 849 and 852.

2.5 Cayley's Form

Explicit differencing the energy operator was discussed first and discounted outright as being numerically hopeless for accurate solutions. Then differencing the time translation operator was discussed, and that seemed to be a better method, but it is still non-unitary.

One of the best ways to difference eq. (2.8) is with *Cayley's form*,[8] which can be loosely thought of as an average of the FTCS scheme given by eq. (2.9), and the implicit scheme given by eq. (2.10). One way to motivate Cayley's form is to write down the tautology that the wavefunction at time $n+1$, translated back by $-\Delta t/2$ is equal to the wavefunction at time n translated forward by $\Delta t/2$. In essence, we have ψ^{n+1} and ψ^{n} "meet" at the middle of the timestep:

$$e^{i\hat{H}\,\Delta t/2\hbar}\psi_j^{n+1} \;=\; e^{-i\hat{H}\,\Delta t/2\hbar}\psi_j^{n}$$

Approximating the time translation operators, as we did in eq. (2.9) and eq. (2.10),

$$\left[\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right]\psi_j^{n+1} \;=\; \left[\hat{\mathbb{1}} - \frac{i\,\Delta t}{2\hbar}\hat{H}\right]\psi_j^{n} \tag{2.11}$$

There is an important difference between Cayley's scheme eq. (2.11) and the explicit eq. (2.9) and implicit eq. (2.10) methods. Since the expansion appears on both sides of the equation, eq. (2.11) turns out to not only be *second* order in time, but unitary as well!

The Cayley scheme can be written down in a way that does not require an explicit

[8] See [49], pg. 853.

computation of an inverse matrix. Solving eq. (2.11) for the unknown wavefunction:

$$\begin{aligned}
\psi_j^{n+1} &= \left(\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right)^{-1}\left(\hat{\mathbb{1}} - \frac{i\,\Delta t}{2\hbar}\hat{H}\right)\psi_j^n \\
&= \left(\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right)^{-1}\left(\hat{\mathbb{2}} - \left[\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right]\right)\psi_j^n \\
&= \left(2\left[\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right]^{-1} - \hat{\mathbb{1}}\right)\psi_j^n
\end{aligned}$$

Define the $\hat{Q}$ operator and its inverse as:

$$\hat{Q} = \frac{1}{2}\left(\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right) \qquad \Longleftrightarrow \qquad \hat{Q}^{-1} = 2\left(\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right)^{-1} \tag{2.12}$$

so that we can write:

$$\begin{aligned}
\psi_j^{n+1} &= \left(\hat{Q}^{-1} - \hat{\mathbb{1}}\right)\psi_j^n \\
&= \hat{Q}^{-1}\psi_j^n - \psi_j^n \\
&= \chi_j^n - \psi_j^n
\end{aligned}$$

Thus, given the known wavefunction ψ^n at timestep n, the prescription for obtaining the unknown wavefunction ψ^{n+1} at timestep $n+1$ is:

1. Solve the linear system of equations for the unknown χ^n from the known ψ^n:

$$\hat{Q}\chi^n = \psi^n$$

2. Obtain the new wavefunction by subtracting the old wavefunction from χ^n:

$$\psi^{n+1} = \chi^n - \psi^n$$

The only thing left to do is find an explicit form for the $\hat{Q}$ operator, which, by virtue of containing a Laplacian, is a functional that depends on the choice of coordinate system.

Those who are familiar with numerical physics may recognize that the work done here alludes to the Crank-Nicholson algorithm.[9] Indeed, the Cayley form and the Crank-Nicholson algorithm are related: when one differences both sides of Cayley's form, a Crank-Nicholson type discretization results. What is done here amounts to moving both sides of Cayley's form to one side before discretizing. This form is still, indirectly, a Crank-Nicholson form.

2.5.1 The Difference Equations

We will now find an explicit form for $\hat{Q}$ and develop the difference equations used to solve eq. (2.2). Starting with the general expression for $\hat{Q}$ (eq. (2.12)), making use of spherical symmetry, and the expression for the Laplacian at the polar origin (eq. (2.7)), $\hat{Q}$ is given by:

$$\begin{aligned} \hat{Q} &= \frac{1}{2}\left[\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\hat{H}\right] \\ &= \frac{1}{2}\left[\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}\left(-\frac{\hbar^2}{2m}\nabla^2 + V\right)\right] \\ &= \begin{cases} \frac{1}{2}\left[\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}V_j^n \; - \; \frac{i\hbar\,\Delta t}{4m}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right)\right] & r \neq 0 \\ \frac{1}{2}\left[\hat{\mathbb{1}} + \frac{i\,\Delta t}{2\hbar}V_j^n \; - \; \frac{3i\hbar\,\Delta t}{4m}\frac{\partial^2}{\partial r^2}\right] & r = 0 \end{cases} \end{aligned} \tag{2.13}$$

[9] See [49], pg. 849.

In light of the prescription to obtain ψ^{n+1} from ψ^n, we need to find the explicit form of $\hat{Q}$ acting on the column vector χ^n. Because the polar origin must be handled separately, and special consideration needs to be made for the endpoint of the spatial grid (which will be explained shortly), three cases of $\hat{Q}\chi^n$ must be considered.

Difference Equations, Case 1: $j \neq 0, N-1$

Starting with eq. (2.13), $\hat{Q}\chi^n$ away from the origin and the endpoint of the spatial grid is:

$$\begin{aligned}\hat{Q}\chi_j^n &= \frac{1}{2}\left[1 + \frac{i\,\Delta t}{2\hbar}V_j^n \;-\; \frac{i\hbar\,\Delta t}{4m}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right)\right]\chi_j^n \\ &= \frac{1}{2}\left[1 + \frac{i\,\Delta t}{2\hbar}V_j^n\right]\chi_j^n \;-\; \frac{i\hbar\,\Delta t}{4m}\left[\frac{1}{r}\frac{\partial\chi_j^n}{\partial r} \;+\; \frac{1}{2}\frac{\partial^2\chi_j^n}{\partial r^2}\right]\end{aligned}$$

Approximating the first and second order derivatives,

$$\begin{aligned}&\hat{Q}\chi_j^n \\ &= \frac{1}{2}\left[1 + \frac{i\,\Delta t}{2\hbar}V_j^n\right]\chi_j^n \;-\; \frac{i\hbar\,\Delta t}{4m}\left[\frac{1}{j\,\Delta r}\left(\frac{\chi_{j+1}^n \;-\; \chi_{j-1}^n}{2\,\Delta r}\right) \;+\; \frac{1}{2}\left(\frac{\chi_{j+1}^n - 2\chi_j^n + \chi_{j-1}^n}{(\,\Delta r)^2}\right)\right] \\ &= -\frac{i\hbar\,\Delta t}{8m(\Delta r)^2}\left(\frac{j-1}{j}\right)\chi_{j-1}^n \;+\; \frac{1}{2}\left[1 + \frac{i\,\Delta t}{2\hbar}V_j^n + \frac{i\hbar\,\Delta t}{4m(\Delta r)^2}\right]\chi_j^n \;-\; \frac{i\hbar\,\Delta t}{8m(\Delta r)^2}\left(\frac{j+1}{j}\right)\chi_{j+1}^n\end{aligned}$$

Defining the constants:

$$R = \frac{\Delta t}{(\,\Delta r)^2} \qquad K = \frac{i\hbar}{8m} \qquad P = \frac{i\,\Delta t}{2\hbar} \tag{2.14}$$

allows us to write the difference equations away from the origin ($j \neq 0$) and the point directly before infinity ($j \neq N-1$) in a compact form:

$$\boxed{\hat{Q}\chi_j^n \;=\; -KR\left(\frac{j-1}{j}\right)\chi_{j-1}^n \;+\; \frac{1}{2}\left[1 + PV_j^n + 2KR\right]\chi_j^n - \;KR\left(\frac{j+1}{j}\right)\chi_{j+1}^n \qquad (j \neq 0, N-1)} \tag{2.15}$$

The Difference Equations, Case 2: ($j = N-1$)

A special case of eq. (2.15) must be considered. A necessary condition of any square integrable function, like a wavefunction, is that it goes to 0 at infinity, which provides a common boundary condition. Since infinity is not internally representable by a computer,[10] a numerical physicist must *choose* a point, far from any expected dynamics, to represent infinity. If the spatial grid starts at 0 and is of size N, the gridpoint $j = N$ is the point at infinity. That is, $j = N-1$ is the point directly *before* spatial infinity, and therefore, the last gridpoint that has a non-zero value of the wavefunction. With this in mind, eq. (2.15) can be used to obtain the difference equation for the point directly before spatial infinity:

$$\boxed{\hat{Q}\chi_{N-1}^n \;=\; -KR\left(\frac{N-2}{N-1}\right)\chi_{N-2}^n \;+\; \frac{1}{2}\left[1 \;+\; PV_{N-1}^n + 2KR\right]\chi_{N-1}^n \qquad (j = N-1)} \tag{2.16}$$

[10]Actually, the IEEE-754 floating point standard *does* make a provision to represent both infinity (`inf`) such as `1/0` and not-a-number (`nan`) such as `sqrt(-1)`, however, any calculation involving `inf` can only result in either `inf` or `nan`, so for numerical research purposes, when `inf` enters our calculations, the calculation becomes completely worthless.

The Difference Equations, Case 3: $j = 0$

Starting with the general expression for $\hat{Q}$ (eq. (2.13)), and approximating the 2nd order derivative, $\hat{Q}\chi^n$ becomes:

$$\begin{aligned}
\hat{Q}\chi_j^n &= \frac{1}{2}\left[1 + \frac{i\,\Delta t}{2\hbar}V_j^n - \frac{3i\hbar\,\Delta t}{4m}\frac{\partial^2}{\partial r^2}\right]\chi_j^n \\
&= \frac{1}{2}\left[1 + \frac{i\,\Delta t}{2\hbar}V_j^n\right]\chi_j^n - \frac{3i\hbar\,\Delta t}{8m}\left(\frac{\chi_{j+1}^n - 2\chi_j^n + \chi_{j-1}^n}{(\Delta r)^2}\right) \\
&= -\frac{3i\hbar\,\Delta t}{8m(\Delta r)^2}\chi_{j-1}^n + \frac{1}{2}\left[1 + \frac{i\,\Delta t}{2\hbar}V_j^n + \frac{3i\hbar\,\Delta t}{2m(\Delta r)^2}\right]\chi_j^n - \frac{3i\hbar\,\Delta t}{8m(\Delta r)^2}\chi_{j+1}^n
\end{aligned}$$

Using the constants defined in eq. (2.14), and using the fact that $j = 0$:

$$\hat{Q}\chi_0^n = -3KR\chi_{-1}^n + \frac{1}{2}\left[1 + PV_j^n + 12KR\right]\chi_0^n - 3KR\chi_1^n \tag{2.17}$$

Spherical symmetry demands that $\chi_{-1} = \chi_1$ at all times, so:

$$\boxed{\hat{Q}\chi_0^n = \frac{1}{2}\left[1 + PV_0^n + 12KR\right]\chi_0^n - 6KR\chi_1^n \qquad (j = 0)} \tag{2.18}$$

2.5.2 The Difference Equations: Comments

Because of the different form of the Laplacian at the origin and the boundary condition at numerical infinity, the explicit form for $\hat{Q}\chi_j^0$ gives three types difference equations:

1. The origin: At j=0, eq. (2.18) is used.

2. The point directly before infinity: At $j = N-1$, eq. (2.16) is used.

3. Elsewhere: At $j \in [1, N-2]$, eq. (2.15) is used.

which generate N linear equations for the N unknown components of the χ_j vector. These equations form a sparse matrix in tridiagonal form:

$$\begin{pmatrix} b_0 & c_0 & & & & & \\ a_1 & b_1 & c_1 & & & & \\ & a_2 & b_2 & c_2 & & & \\ & & & \ddots & & & \\ & & & & a_{N-2} & b_{N-2} & c_{N-2} \\ & & & & & a_{N-1} & b_{N-1} \end{pmatrix} \begin{pmatrix} \chi_0^n \\ \chi_1^n \\ \chi_2^n \\ \vdots \\ \chi_{N-2}^n \\ \chi_{N-1}^n \end{pmatrix} = \begin{pmatrix} \psi_0^n \\ \psi_1^n \\ \psi_2^n \\ \vdots \\ \psi_{N-2}^n \\ \psi_{N-1}^n \end{pmatrix} \tag{2.19}$$

where the left hand side is the expression of $\hat{Q}\chi_j^n$. The zeroth coefficients, b_0 and c_0, which represent the difference equations at the origin, are generated by eq. (2.18). The coefficients a_{N-1} and b_{N-1}, are generated by eq. (2.16). All other coefficients are generated by eq. (2.15). It is noteworthy that only the b_i are time dependent, since they contain a sum over potential energy, and the potential energy changes from timestep to timestep.

I initially developed an algorithm to solve tridiagonal systems, but have since learned that it is already known as the Thomas Algorithm, and is discussed in appendix C.

2.6 Summary

An informal flowchart of the process to numerically solve the problem as stated in section 2.1 is given below:

1. Load the initial condition, eq. (2.1), into the $N \times 1$ vector, ψ_j^0.

2. Calculate the potential V_j^0 from ψ_j^0, as outlined in appendix B.

3. Form the tridiagonal matrix via the difference equations eq. (2.18), eq. (2.15), and eq. (2.16).

4. Solve the tridiagonal system using the Thomas Algorithm as outlined in appendix C.

5. Obtain the new wavefunction by performing the subtraction $\psi_j^1 = \chi_j^0 - \psi_j^0$.

6. Go back to step 2 and repeat until we reach the final timestep.

2.7 Program Details

This section discusses some numerical issues that warrant mention in a nearly language independent manner.

2.7.1 Initial Condition

The choice of initial condition was based upon what a particle wavefunction ought to look like. Although the exact profile of a particle wavefunction is unknown, a Gaussian is a very good guess: it's plausible, simple, and spherically symmetric, which takes advantage of the Schrödinger-Newton symmetries. Therefore, it was decided to use a normalized Gaussian of parameter α to be the initial condition used for this project.

$$\psi(r,0) = \left(\frac{\alpha}{\pi}\right)^{3/4} e^{-\alpha r^2/2} \tag{2.20}$$

It's useful to be able to talk about the width w of the Gaussian being used as the initial condition for the PDE solver. However, width is a nebulous concept for something with infinite extent. As Figure 2.1 demonstrates, the width is somehow inversely proportional to α. Dimensional analysis suggests $w \approx 1/\sqrt{\alpha}$, which is as accurate a relation as can be proposed since we're not even sure that the Gaussian is the exact particle wavefunction to use in the first place.

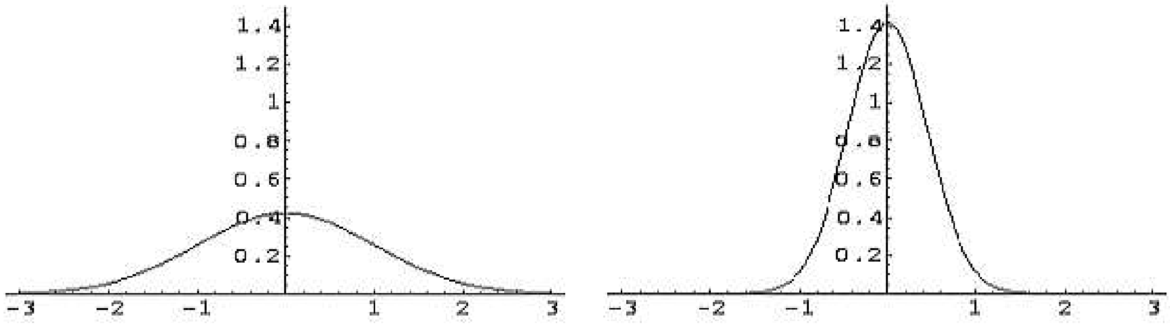


Figure 2.1: The Gaussian parameter α is a measure of the wave packet's width w. From dimensional analysis, it's ostensibly proportional to w^{-2}. The profiles shown use $\alpha = 1$ on the left, and $\alpha = 5$ on the right.

There's an ambiguity in how to relate the width of the Gaussian wavefunction with the *object* it represents. For example, what should α be for a 1 kg baseball? Should $1/\sqrt{\alpha}$ be the diameter of a baseball (.08 m)? The lattice size of the solid (10^{-10} m)? Its Compton wavelength (10^{-42} m)?

It turns out there *is* a definitive answer to this question: the de Broglie wavelength. This will be thoroughly discussed in chapter 4.

2.7.2 Choice of the Point at Infinity

In any numerical situation where an infinity is encountered, the numerical physicist must deal with it in a manner which is computationally reasonable while ensuring that the results are accurate. Although the expected dynamics of a self gravitating particle at the origin is expected to occur near the origin, points far away from the origin must also be considered for two reasons:

1. However unlikely, there might be interesting behavior far from the origin. After all, the Schrödinger-Newton equation is highly non-linear.

2. The point at infinity provides a crucial boundary condition. In order for ψ to be a square integrable function, the wavefunction must go to zero as $r \to \infty$.

There are many clever ways to map an infinite solution domain onto a finite grid. The method used was chosen for its simplicity and reasonability. The initial wavefunction is real and has a maximum at $r = 0$. It is desired to calculate the location, called `rend`, where $\psi(r, 0)$ takes on a specified fraction, called `rendratio`, of the maximal value. Since $\psi(r) = (\alpha/\pi)^{3/4} e^{-\alpha r^2/2}$,

$$\texttt{rendratio} = \frac{\psi(\texttt{rend})}{\psi(0)} = e^{-\alpha \texttt{rend}^2/2} \qquad \Longrightarrow \qquad \texttt{rend} = \sqrt{-\frac{2\ln(\texttt{rendratio})}{\alpha}}$$

`rend` is the point *just before* the point at numerical infinity. Thus, `rend+dr` is considered numerical infinity, and the boundary condition at infinity, eq. (2.4), is numerically enforced by ensuring that $\psi(\texttt{rend} + \texttt{dr}, \texttt{t}) = 0$ for all `t`.

Although `rendratio` is not a parameter of a particular physical system, it is an important program parameter.

- If `rendratio` is too large (`rend` is too small) it may be impossible to accurately enforce the boundary condition at numerical infinity, or an unacceptable amount of probability will 'leak out of the universe'. Furthermore, numerical artifacts may occur. For example, if `rend` is too close to the origin, ψ may reflect off infinity and interfere with the dynamics at the origin.
- If `rendratio` is too small (`rend` is too large), then the number of spatial grid elements (`N`) or the size of individual grid elements (`dr`) will be too large to be accurate for the difference equations.

While we're enforcing ψ to be zero at a point where it *really* shouldn't be, and probability will necessarily leak out with time, choosing a good value of `rendratio` will minimize the error associated with these defects to well below that of numerical uncertainty. If, when analyzing the results, dynamics are observed anywhere near numerical infinity, the program is rerun with `rendratio` decreased to bring the dynamics closer to $r = 0$.

2.7.3 The Spatial and Temporal Grid

Like `rendratio`, the grid size `N` is another program input. Once `rend` (the point before numerical infinity) is found, space is divided up into `N` numerical elements of size `dr`:

```
long double dr = rend / (N - 1);
```

When using Von Neumann's analysis on numerical algorithms that solve differential equations, conditions for numerical stability are usually expressed as a condition on the size of $R = \Delta t/(\Delta r)^2$. Large R is associated with numerical instability and small R is associated with numerical stability. This translates into keeping timesteps 'small' compared to the size of the spatial grid. The algorithm used for this project happens to be unconditionally stable[11] for all values of R.[12] Although the Crank-Nicholson algorithm is unconditionally stable, `R` is used as a program parameter to calculate `dt` from `dr`.

```
long double dt = R * dr * dr;
```

A simulation for a time dependent equation would ideally run forever. However, as with the spatial grid, a temporal cutoff must be declared while hoping that we don't miss important dynamics in enforcing the cutoff.

The ending time of the simulation is stored in a variable named `endt`. Then, the

[11] See [49], pg. 849.

[12] Numerical stability does not imply accuracy. We still do not want `dt` to be too large.

number of timesteps required to finish the simulation, `MaxTStep`, is calculated by:[13]

```
unsigned long long int MaxTstep = ceill(endt / dt);
```

The zeroth timestep is defined to be the initial condition.

[13] The `ceill`(x) function returns the smallest integer not less than x.

Chapter 3

Numerical Considerations

This chapter discusses numerical issues and techniques used to generate, analyze and organize the output of the numerical PDE solver and the associated tools used for this project.

3.1 Choice of Platform

The PDE solver and associated tools to analyze and organize the data were written in C and Perl on the GNU/Linux platform. While there are powerful commercial software packages that can do many of the tasks described in this chapter, I chose not to use these packages for a number of reasons. First was the issue of cost. Every tool used with this project was Open Source software run on the free GNU/Linux platform. Second was speed. A prototype PDE solver was written in Mathematica and was found to be well over an order of magnitude slower than C code. Third was extensibility. As the project progressed, I wanted new functionality and flexibility in certain numerical tasks. It was easier to get precisely what I wanted by writing my own code. Fourth was familiarity. I don't like the black box approach to numerical research. By writing all my own code, I'm intimately familiar with every aspect of how data gets generated and analyzed. The last issue was portability. By writing the project in C on Linux, I was able to run my code everywhere. Even computers that don't have Linux installed on them were used by booting a Knoppix Linux disk, letting the program run, and using ssh to retrieve the data.

3.2 Stateful Numerics

In principle, most problems can be solved on a computer. In practice, some numerical problems cannot be solved in a reasonable amount of time, that is, it would take far too many timesteps to solve the problem. For numerical time evolution projects like this one, the concern is that interesting dynamics may occur after either the program ends or the researcher gives up and quits the program. Often the researcher knows that the dynamics can't be reached within a reasonable "wall clock time." But once the program ends, for any reason, the run is over. A phenomenal amount of computational effort goes wasted whenever a system goes down for maintenance, during a power outage, etc.

I've developed a technique that allows a program to be stopped and started at will. The program could even be stopped and then started on a different computer.[1] With a little effort, it can even be used to extend a job that has reached the final time step of the simulation. It can even be used to pick up where the program left off in the case of operating system crash or power outage. I've never seen this technique used, even though it could be of enormous value to all numerical researchers. I call it *stateful numerics* and will demonstrate the principle so that others can use this technique as well.[2]

To use Stateful Numerics, every program variable and parameter must either be a preprocessor `#define` or wrapped in some kind of advanced data type (ADT), like a C struct or C++ class. In this simplified example, the ADT `state_type` is a declaration of a struct that will be used to wrap all the program's run parameter variables:

```
#ifndef _COMMON_H_
#define _COMMON_H_
#include <tgmath.h>


typedef struct state_type
{
   long double time;
```

[1]The program needs to be restarted on a computer of the same type of architecture.

[2]I have since learned that there are software packages, like "Condor", that perform a similar function, which is sometimes called "checkpointing". Although using Condor is much easier than implementing stateful numerics, installing a large software package on every computer used for computation may not be an option for a researcher, particularly since we often run code on other people's machines. Another reason why using stateful numerics would be desirable is compatibility. Condor was in use at UC Davis but had to be removed since it no longer functioned after a department-wide upgrade of Redhat Linux.

```
   long double V[N];         // Potential array
   long double complex K, P; // Widely used complex constants
   long double complex a[N], b[N], c[N], X[N], wf[N];
   unsigned long long int MaxTStep, tstep;
} state_type;

extern state_type *s;

#endif
```

Note that only variables that exist throughout the entire length of the run need to wrapped by the ADT. For example, the potential array `V[N]` needs to exist during the entire run, however, a local variable `i` which is used to loop over an array within the tridiagonal matrix solving subroutine does not need to be wrapped by the ADT. Generally, any variable that remains in scope during the program's "main loop" will need to be wrapped.

Next, a `state_type` needs to be instantiated and initialized. Since every persistent program variable will be wrapped by `s`, it is most convenient to make `s` global, although this is not necessary. Another variable (not wrapped by `s`) must be declared global:

```
volatile int stop;
```

The `stop` variable is essentially boolean and will be used to determine when the program should stop. For example, if the program receives a control-c at the keyboard (if the user wants to halt the program) or if the computer is being shut down and the program receives the HUP or KILL signal, or if a cron job sends a signal to the program to tell it to "go to sleep", the `stop` variable[3] is set to 1 by a user defined signal callback. Since `stop` is asynchronously set by callback methods, its value can change at any time, even when its value is not explicitly set by executing code. To prevent the compiler from making any assumptions about the value of `stop` due to code optimizations, this variable is declared to be **volatile**.

Next, signal callback functions must be used to arrange for `stop` to be set to 1 when the appropriate signal is raised. For example, this function sets `stop` to 1 whenever control-c is pressed:

```
/*  We received a control C.  The user wants to stop the program.  This means
 *  setting the global stop variable to "1" so we can finish up the main loop
 *  and exit gracefully.
 */
void int_callback(int sig_number)
{
   if (sig_number != SIGINT) {
      fprintf(stderr, "%s:%d %s error: "
         "recieved wrong signal number %d not %d\n",
         __FILE__, __LINE__, __FUNCTION__, sig_number, SIGFPE);
```

[3]Since `stop` is a global int, it is initialized to zero.

```
        exit(2);
    }

    stop = 1;
}
```

The main loop of the program, the one that does all the calculations, runs until either the final timestep is reached or if `stop` is set to a value different from zero:

```
    // Main Loop
    //
    while ( ! stop && ( s->time < endtime ) )
    {
        // Do numerical work here.

        if ( stop )
            Graceful_Shutdown();
    }
```

Thus, when the program is told to stop for any reason, it stops gracefully, meaning, the current timestep calculation or iteration is first allowed to complete and the program shuts down before beginning the next timestep calculation or iteration.

This explains how a program knows when and how to gracefully shut down, but how is the state actually saved? Using command line options, you instruct the program whether the job is being started from scratch or restarted from a previously interrupted job.

For new jobs, a memory map of the state variable `s` is created, as illustrated by this simplified example code which creates a map file named `char *name`:

```
// Create a map file
fd = open(name, O_CREAT|O_RDWR, 0644);

// Make sure the file is big enough to hold the contents of s.
lseek(fd, sizeof(*s) - 1, SEEK_SET);

// Create a memory map for s.
s = mmap(0, size, PROT_READ | PROT_WRITE, MAP_SHARED,
   name, 0)) == MAP_FAILED);

// Close the map file.
close(fd);
```

A memory map maps a file into memory, so each access to memory (reading from a variable or writing to a variable) is actually done on a file residing on a hard drive. This sounds slow, but isn't because of caching, and provides two main benefits:

1. The state of the entire program is stored on disk, which we can then use at a later time to restart the application where it left off.

2. A disk file this small would be cached in memory anyhow. Using `mmap` is much faster than simply writing the variable directly to disk because for disk writing, the kernel has to transfer a lot of numerical data between user space buffers and kernel space buffers, and then perform input/output on its buffers. With mmap, the data is directly written to and read from the mapped memory buffers.

On the other hand, if the program is restarting a previously suspended job, we simply need to open the memory map file and perform a `mmap` on it:

```
if ((fd = open(name, O_RDWR)) == -1) {
   die("Failed to open file '%s'", name);
} else if ((result = mmap(0, size, PROT_READ | PROT_WRITE,
      MAP_SHARED, fd, 0)) == MAP_FAILED) {
   die("Failed to map file '%s'", name);
}


if ((fd != -1) && close(file))
   die("Failed to close file '%s'", name);
```

It's not necessary to `munmap()` the memory map since memory maps are automatically unmapped when the program terminates.

Stateful numerics can even save a job during a catastrophic shutdown, like a power outage: the only "point of failure" is when the memory map buffer is flushed to disk. Since stateful numerics is rather complicated to implement, it would only be used for a complex numerical task. This implies that the program spends an overwhelming portion of its time doing numerical calculations rather than flushing buffers to disk. Therefore, the probability that the catastrophic failure occurs during the buffer flush is quite low.

3.3 Visualizing Time Evolution

When solving a time dependent equation like the Schrödinger-Newton equation, it's instructive to be able to visualize the time evolution of the initial condition. This means making a "movie" of the data. In this section, I describe how that was accomplished.

The output of the PDE solver, for a given timestep j, is the wavefunction at $t = j\,\Delta t$. The parameter `PLOTS` was used to control when data was written to a log file by using the modulo operator. This is necessary because a particular run can have 10^6 or more timesteps. If the current timestep is `tstep` and the last timestep is `MaxTStep`:

```
while ( time < endtime )
{
   // Solve for the wavefunction at current timestep

   if (tstep % (MaxTStep / PLOTS) == 0)
      Plot_Data();
```

Data files were written `PLOTS` times. Many types of data were saved, but the most important type of data file was the probability distribution: r versus $r^2|\psi(r,t)|^2$. The probability distributions were saved into files named after the time step, and zero padded:

```
void Plot_WFSQ( const long double complex wf[N],
   const unsigned long long int tstep )
{
   FILE *fp;
   long double r;
   char filename[STRLEN];

   snprintf(filename, STRLEN-1, "%s%011llu", "data/WFSQ/WFSQ.", tstep);
   if ((fp = fopen(filename, "w")) == NULL)
      die("Plot_WFSQ: Couldn't open file for writing.");

   for(register int j=0; j<N; ++j) {
      r = j*dr;
      fprintf(fp, "%Le, %Le\n", r, r * r * cabsl(wf[j])*cabsl(wf[j]));
   }

   fclose(fp);
}
```

The zero padded file names are important because it makes directory listings of the probability distribution files list in time order, for example:

```
$ ls -1 data/WFSQ
WFSQ.00000000000
WFSQ.00000000333
WFSQ.00000000666
WFSQ.00000000999
WFSQ.00000001332
WFSQ.00000001665
...
```

Without zero padding, the file names would list in ASCII alphabetical order, as in:

```
$ ls -1 data/WFSQ
WF.0
WF.1332
WF.1665
WF.333
WF.666
WF.999
...
```

The first file, `WFSQ.00000000000`, contains $(r,\ r^2\,|\psi(r, 0\,\Delta t)|^2)$ values for each spatial gridpoint. The second data file, `WFSQ.00000000333`, contains $(r,\ r^2\,|\psi(r, 333\,\Delta t)|^2)$, and so on. For example:

```
0.000000e+00, 0.000000e+00
3.925832e-12, 2.122586e+07
7.851663e-12, 8.462125e+07
1.177749e-11, 1.893443e+08
1.570333e-11, 3.340076e+08
1.962916e-11, 5.167007e+08
     ...
```

A Perl program was written to generate a gnuplot script. Since directory listings of the data files are time ordered, plots of the data files are displayed in time order. The Perl program grew to be quite sophisticated (over 1000 lines of Perl code), and allowed me to scale axes, focus on individual data points, set zoom level, set graph labels, set pauses, run the evolution backwards, and much more.

The last few lines of the Perl plotter were responsible for generating the gnuplot script that displays the probability distribution time evolution files in rapid succession, simulating a movie of the PDE's evolution:

```
# Erase previous plots and open a new plot file.
unlink("$var{dir}/makeplot");
open(FP, ">$var{dir}/makeplot")
   or die("Can't open $var{dir}/makeplot for writing");
```

```
# Obtain a list of timesteps from data/WF.
my @files = glob "$var{dir}/WFSQ/WFSQ*";

Print_Interesting_Stuff();

LOOP: foreach my $file ( @files )
{
   my $tstep = join '',  $file =~ /WFSQ.([0-9]+)/;
   next LOOP if ( $tstep * $run{dt} < $var{min_time} );
   Create_A_Plot($tstep, \%run, \%var);
}

system("gnuplot", "$var{dir}/makeplot");
```

The last line calls gnuplot to generate the movie. A sample of the Perl program's output shows the type of input that gnuplot expects:

```
$ head -12 makeplot
set xrange [0:2.5e-8]
set yrange [0:3e8]
set label 1 'rend=9.10456277e-08 dt=1.00000000e-07, dr=2.27670987e-11'
   at graph .42, .81
```

```
set label 2 'tstep=00000000000, t=0' at graph .42, .84
set pointsize 1
plot 'data/WFSQ/WFSQ.00000000000', 'data/FPSQ/FPSQ.00000000000'
set xrange [0:2.5e-8]
set yrange [0:3e8]
set label 1 'rend=9.10456277e-08 dt=1.00000000e-07, dr=2.27670987e-11'
   at graph .42, .81
set label 2 'tstep=00000000333, t=3.33e-05' at graph .42, .84
set pointsize 1
plot 'data/WFSQ/WFSQ.00000000333', 'data/FPSQ/FPSQ.00000000333'
```

The Perl plotter was used to generate most of the graphs used in this dissertation. This technique of visualizing time evolution data can be used easily and freely on any operating system that supports Perl and gnuplot, which includes any modern major operating system.

3.4 Numerical Accuracy

A very wide assortment of techniques was employed to ensure the accuracy of these results. Accuracy of the solutions started with the choice of algorithms and went on to include understanding details of the C library and architectures of the computers used to run the PDE solver.

3.4.1 Precision

The **long double** datatype was used for every real floating point variable and **long double complex** was used for all complex floating point variables. On the majority of the computer architectures used, this gives 18 digits of precision and can store powers of 10 between 10^{4932} and 10^{-4931}. This precision and range of exponents is much more than enough for this project, which by virtue of various combinations of $\hbar$ and G can be numerically demanding.

3.4.2 Accuracy

It is not enough to use stable algorithms and double precision variables and expect the solutions to be correct. Highly precise results are meaningless if they're not accurate. Consider the following code which attempts to print a number that's representable by a **long double**:

```
int main(void)
{
    long double trouble = 2.0 * pow(10, 4931);
    printf("Trouble: %Le\n", trouble);
    return 0;
}
```

The output of this test program is:

```
$ gcc -W -Wall foo.c -lm; ./a.out
Trouble: inf
```

What happened to the **long double** representation of 10^{4931}? For all its wonderful traits, C is not a forgiving language and is certainly not as "user friendly" as, say, Fortran-90. The function `pow()` is defined to accept doubles and returns doubles. Since 2×10^{4931} is not representable as a **double**, the result was infinite despite the fact that the result was stored in a **long double**. To correct this, the programmer should either use the new so-called 'type generic' mathematics or call **long double** versions of the relevant math functions, like `powl()`. Every line of code was scanned for such sources of error.

The Use of Floating Point Signals

Another safeguard against numerical errors was the use of IEEE signals. A computer's CPU and operating system can catch many common floating point errors like `FE_OVERFLOW` (floating point overflow). In numerical work, it's desirable to catch these signals since once a floating point exception is generated, the rest of the program's output can no longer be trusted. On GNU/Linux, floating point signals are turned off by default, so they have to be explicitly enabled via code like:

```
#include <fenv.h>

// By default all FPE's are masked off... "fix that".
```

```
//
void fpe_trap_enable(void)
{
  /* Enable FPE's.  By default all FPE's will not raise a signal when
   * they happen.  See fenv.h for magic constants.
   *    FE_INEXACT     inexact result - don't do this!
   *    FE_DIVBYZERO   division by zero
   *    FE_UNDERFLOW   result not representable due to underflow
   *    FE_OVERFLOW    result not representable due to overflow
   *    FE_INVALID     invalid operation
   */
  feenableexcept(FE_DIVBYZERO | FE_UNDERFLOW | FE_OVERFLOW | FE_INVALID);
}
```

Rather than let `inf`, `nan`, underflowed or overflowed floating point numbers propagate through the calculations, this code arranges for the program to halt which is desirable since the worst case scenario would be for the exceptions to generate incorrect results which are not 'obviously' incorrect.

The use of signals and callbacks can be used, minimally, to end the program and print the cause of the exception. A sophisticated system can be employed to print the exact file, function, line number, and variable that caused the exception to occur. Without signals, the researcher often has no way of knowing that the program is generating faulty output until the program runs its course, which can take a very long time. In the worst case scenario, the incorrect numerical output is indistinguishable from a reasonable solution. This code illustrates how to arrange for a user written function named `fpe_callback()` to be executed whenever a FPE is generated:

```
/* Setup a signal handler for SIGFPE */
   struct sigaction action;

   memset(&action, 0, sizeof(action));
   action.sa_sigaction = fpe_callback;  // which callback function to call
   sigemptyset(&action.sa_mask);        // other signals to block
   action.sa_flags = SA_SIGINFO;        // give details to callback

   if (sigaction(SIGFPE, &action, 0))
      die("Failed to register signal handler.");
```

And this is a listing for `fpe_callback()`: an example of what can be done once a FPE is generated. Here, the cause of the FPE is printed so I can study what the problem was and avoid it in the future:

```
//  We generated a SIGFPE.  No sense in continuing with the program
//  since our numbers are now garbage.  Just print a message and die.
//
void fpe_callback(int sig_number, siginfo_t *info, void *data)
{
```

```
   data = data;      /* used for SIGIO (see F_SETSIG in fcntl) */

   if (sig_number != SIGFPE) {
      fprintf(stderr, "%s:%d %s: recieved wrong signal %d not %d\n",
         __FILE__, __LINE__, __FUNCTION__, sig_number, SIGFPE);
      exit(2);
   }
   fpe_print_cause(stderr, info);
   exit(1);
}


void fpe_print_cause(FILE *file, siginfo_t *info)
{
   fprintf(file,
      "FPE reason %d = "\%s", from address 0x%X\n",
      info->si_code,
      info->si_code == FPE_INTDIV ? "integer divide by zero"     :
      info->si_code == FPE_INTOVF ? "integer overflow"           :
      info->si_code == FPE_FLTDIV ? "FP divide by zero"          :
      info->si_code == FPE_FLTOVF ? "FP overflow"                :
      info->si_code == FPE_FLTUND ? "FP underflow"               :
```

```
        info->si_code == FPE_FLTRES ? "FP inexact result"          :
        info->si_code == FPE_FLTINV ? "FP invalid operation"       :
        info->si_code == FPE_FLTSUB ? "subscript out of range"     :
        "unknown",
        (unsigned int) info->si_addr
    }
```

The one SIGFPE signal that was *not* useful was `FPE_INEXACT`. Because of how floating point numbers are internally represented, many common floating point operations will raise the `FPE_INEXACT` flag. Even a calculation like `6.0L / 3.0L`, which yields an exact integer result, yields an inexact floating point result.

Coding Techniques

Effort was put into learning good numerical coding techniques. For example, what engineers call "feedback" is an often used optimization while programming but happens to be detrimental when dealing with numerical calculations. A simplified example of feedback is the common paradigm of incrementing a floating point variable by: `t = t + dt` or equivalently `t += dt`. Rounding errors are compounded with each timestep, and the error in the floating point variable is proportional to the number of timesteps. For this example, a better solution is to use an **int** variable representing the timestep; there is no error in integer arithmetic.

As another example of using defensive coding techniques to minimize the error in the solution, consider the following program which attempts to print the cosine of $\pi/2$ divided by some small number:

```
int main(void)
{
    // long double
    long double varl = cosl(M_PIl / 2.0L);
    printf("%Le\n", varl);
    varl /= 1.0E-20L;
    printf("%.12Le\n", varl);
    return 0;
}
```

The answer, of course, is zero, but the output is surprising:

```
$ ./a.out
-2.710505e-20
-2.710505431214e+00
```

Although $\cos(\pi/2)$ is numerically small, a division or multiplication operation can turn a numerically insignificant number into a number as significant as one pleases. Therefore, it's not enough to simply translate an algorithm into code; one needs to consider the terms in the algorithm and try to arrange the arithmetic to minimize this effect.

The Compiler

One normally does not consider the compiler itself to be protection against numerical error, but it was found that in addition to having excellent optimization capabilities, gcc was also an effective way to protect the code from human error. C is an unforgiving language, and compiler warnings were extremely useful. Here is an excerpt from the Makefile which was used to turn on compiler warnings:

```
WARN = -Wall -Wstrict-prototypes -Wmissing-prototypes -Waggregate-return \
      -W -Wpointer-arith -Wcast-qual -Wcast-align -Wmissing-declarations \
      -Wnested-externs -Wredundant-decls -Wwrite-strings -Winline -Werror
CFLAGS = $(WARN) -std=c99 -g3 $(OPTIMIZATION_FLAGS)
```

Discrepancy Plots

A necessary requirement for the PDE solver to calculate correct solutions to the Schrödinger-Newton equation is being able to calculate correct solutions to the free particle. This was important when testing the algorithms and difference equations, but is also important when running jobs since one cannot expect accurate solutions for all values of the input parameters. For example, there are bounds on variables like `dr` and `dt`, and outside these bounds, the computed solution is no longer accurate.

To help verify accuracy for a given run, the PDE solver was re-run with the same input parameters but with the potential turned off. The resulting numerically computed free particle was compared to the analytic free particle to see that the difference equations were generating accurate solutions. Although this doesn't *ensure* a correct solution, it helps build confidence in its validity.

One method of comparing the numeric and analytic free particle wavefunctions was to compute the discrepancy in the probability distributions at time $t_j = j\,\Delta t$, defined as:

$$\delta\text{prob}_j \;=\; 4\pi \sum_{i=0}^{N} r_i^2 \left(\left|\psi(r_i,t_j)\right|^2 - \left|\,\psi_n(r_i,t_j)\right|^2 \right) \Delta r$$

where $\psi(r_i, t_j)$ and $\psi_n(r_i, t_j)$ are respectively the analytic and numeric free particles. Unlike direct comparison of the wavefunctions, there's no question of an arbitrary constant phase that might appear in the numerical solution, and a probability discrepancy that's small compared to 1 means an accurate homogeneous solution. Figure 3.1 shows an example probability discrepancy plot for $m = 9.1 \times 10^{-31}$ kg and $\alpha = 1.0 \times 10^{0}$ m^{-2}. Although the discrepancy monotonically increases, by the millionth timestep, the overall discrepancy is still quite small.

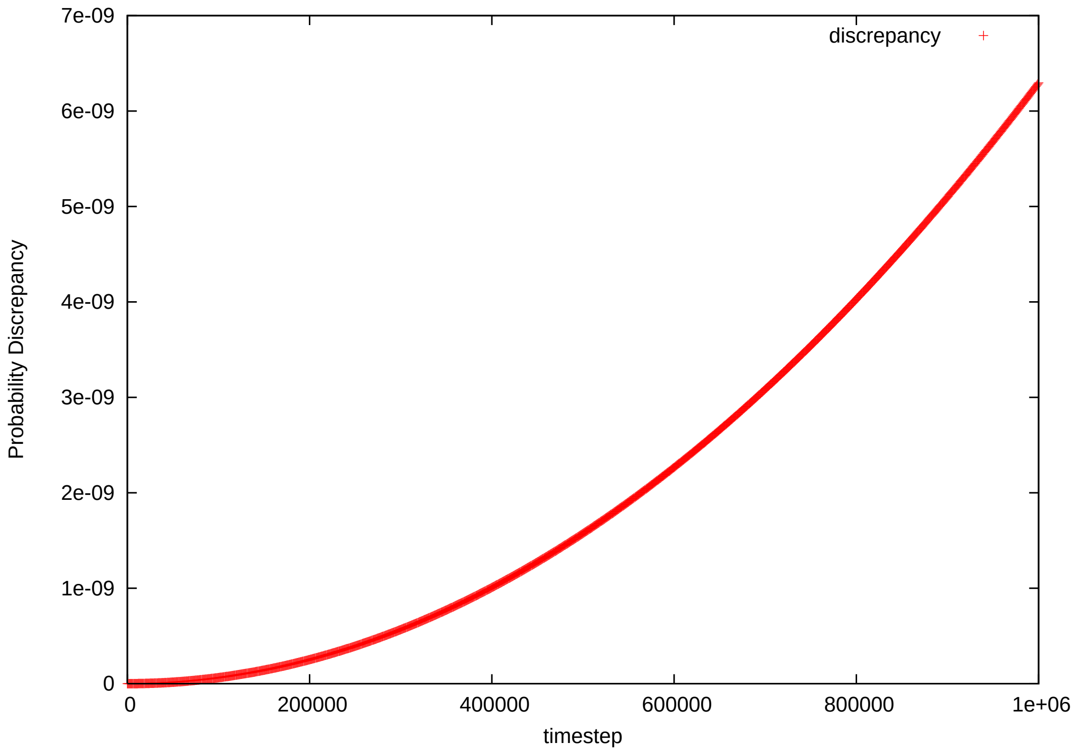


Figure 3.1: Probability discrepancy versus timestep for $m = 9.1 \times 10^{-31}$ kg and $\alpha = 1.0 \times 10^{0}$ m^{-2}. Although the error grows monotonically, after a million timesteps, it's still very small.

3.4.3 Normalization

Normalization was a non-issue. As discussed in chapter 2, the Cayley form has unitarity built into it. As can be seen in Figure 3.2, the numerically computed gravitational wavefunction retains its normalization quite well.

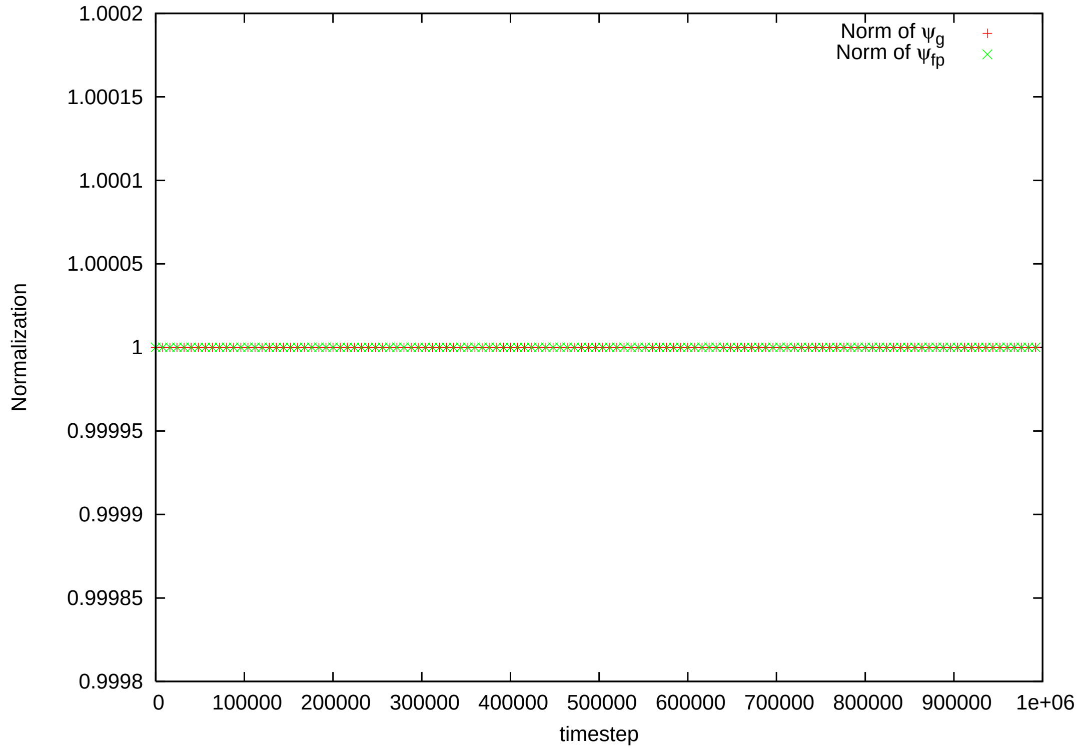
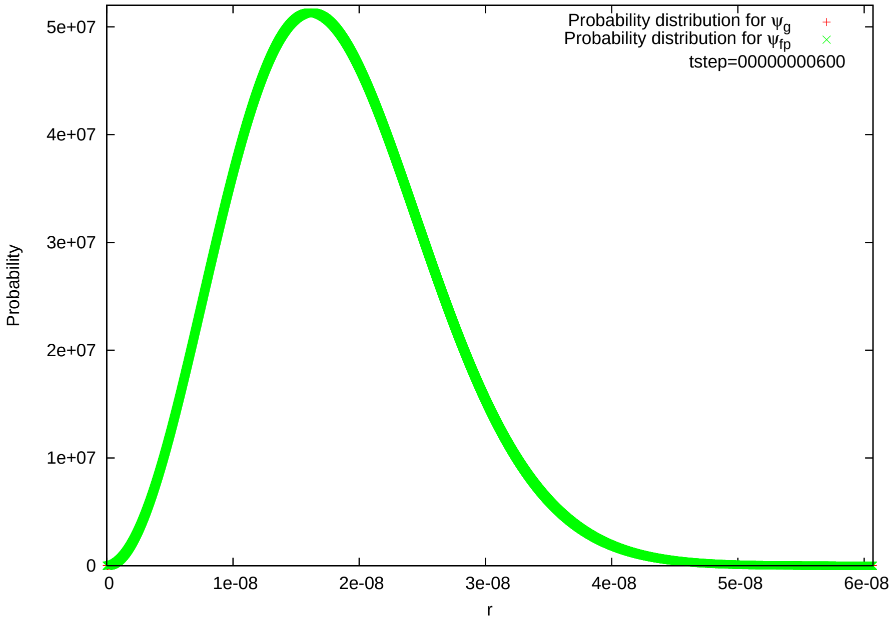
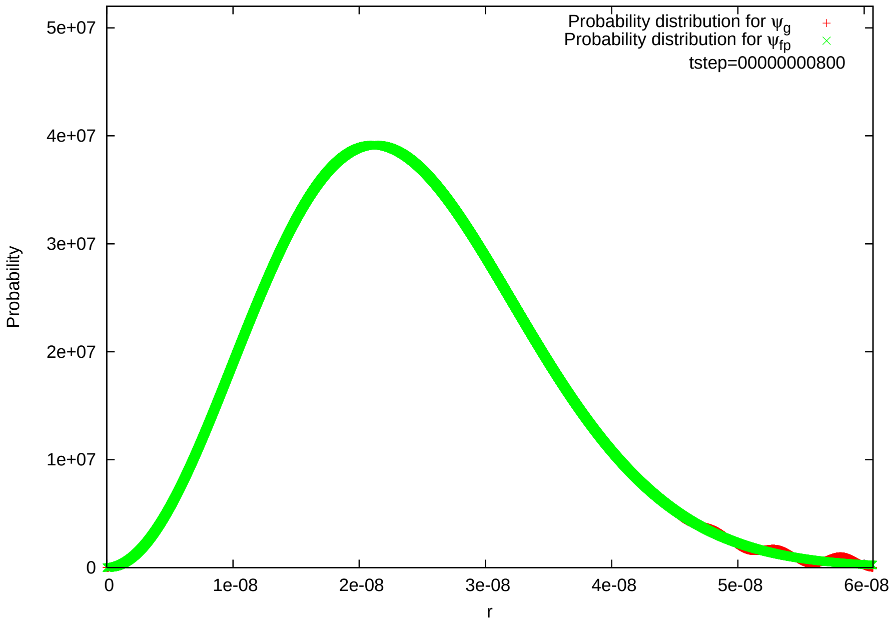


Figure 3.2: Numerical wavefunction norm versus timestep for $m = 9.1 \times 10^{-31}$ kg and $\alpha = 1.0 \times 10^{0}$ m^{-2}. Even after a million timesteps, the numerical wavefunction retains its normalization quite well. Unitarity is one of the benefits of using the Cayley discretization method to numerically solve a Schrödinger equation.

3.4.4 Reflection Off Numerical Infinity

Even when numerical infinity is placed too close to the origin, the numerically computed wavefunction retains its normalization, even while the analytic wavefunction "leaks" out of the simulation. Figure 3.3 shows the time evolution of two wavefunctions. The green curve (`x`) represents the analytic free particle and the red curve (`+`) represents the free particle computed by the difference equations. The two wavefunctions are identical until the numerical wavefunction hits numerical infinity.


Probability distribution for ψ_g
Probability distribution for ψ_{fp}
tstep=00000000600
Probability
r
5e+07
4e+07
3e+07
2e+07
1e+07
0
0
1e-08
2e-08
3e-08
4e-08
5e-08
6e-08


Probability distribution for ψ_g
Probability distribution for ψ_{fp}
tstep=00000000800
Probability
r
5e+07
4e+07
3e+07
2e+07
1e+07
0
0
1e-08
2e-08
3e-08
4e-08
5e-08
6e-08

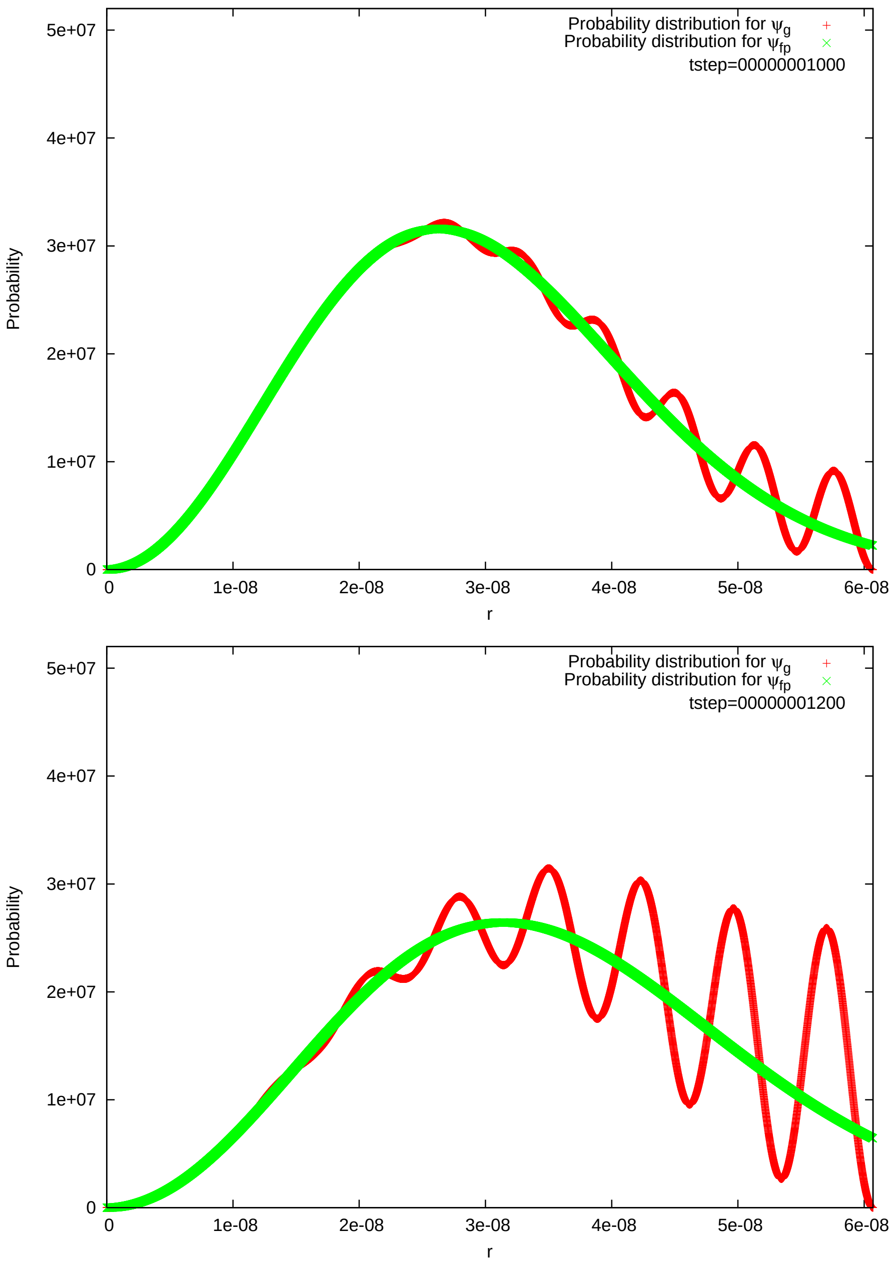


Figure 3.3: A wavefunction reflecting off numerical infinity. Even wavefunctions that are highly localized near the origin have infinite extent, so this behavior always occurs. However, if numerical infinity is “far” enough, this effect does not have a significant role in the main body dynamics.

Notice that the probability for the analytic free particle wavefunction "leaks" out of the universe. This is because it was derived (see appendix D) on the semi-infinite interval $[0, \infty)$, that is, infinity is the "true" spatial infinity.

However, the wavefunction generated by the difference equations reflects off the far boundary of the simulation. This is because the Neumann boundary condition at numerical infinity causes the wavefunction to reflect back towards the origin. A real wavefunction that managed to reach true spatial infinity would also have to behave this way as well.

3.5 Speed

The PDE solver took a very long time to run, so optimizing for speed was important. The most effective speed optimization was aggressively using pre-processor `#define`'s in the place of automatic and static/global variables. Not only did extensive use of pre-processor constants reduce the required memory thumbprint, but it reduced the "wall-clock time" that the program took to complete by well over an order of magnitude. It also had the effects of *safely* making commonly used variables global in scope and reducing the work required to implement stateful numerics since literal constants do not need to be wrapped by the state ADT.

The downside is that the PDE solver had to be recompiled for each minor change of input parameter (as opposed to taking input via command line options), but the benefits in using pre-processor constants whenever possible was more than worth this minor inconvenience.

The gcc compiler itself provides a wealth of highly effective speed optimizations. However, the vast array of compiler options can get out of hand for a numerical researcher. Unfortunately, one needs to be somewhat of an expert to understand many of the more powerful optimization options, especially since they come with warnings like “may produce incorrect results”. Much effort was put into determining which set of compiler optimization flags were optimal *and safe* to use.

Benchmarking, profiling, and tuning is an important part of any long-term numerical project. On Linux, benchmarking is done with bash’s[4] `time` command, which is invoked with the name of an executable. The output of `time` is a set of three numbers:

```
$ time ./avatar
real    2m47.476s
user    2m47.143s
sys     0m0.094s
```

The important time here is “user” which is the amount of time that the computer spends on the application to be benchmarked. The “real” time indicates the wall clock time it took for the application to finish. The “sys” time indicates how long the computer spent within system calls while executing the application. The real time changes with system load, and the sys time is simply not relevant here.

Compiler optimization is a heuristic pursuit: turning all the options “on” is no

[4]bash is the default shell that comes with GNU/Linux.

guarantee that the code will run faster than turning only a few options on. In fact, in some instances, a particular optimization can make the code go *slower*. A common example is loop unrolling.[5] There are situations where loop unrolling slows down the program, and only experimentation will reveal whether that optimization is right for a given piece of code. As another example, the gcc option `-O1` is the lowest optimization level while `-O3` is the most aggressive. However, as can be seen below, the program actually ran slower with `-O3` than with `-O1` except when combined with `-ffast-math`. The following table lists some benchmark results with various gcc optimization options for a short run.[6] Although the absolute times are unimportant,

[5]Unrolling loops when the number of iterations can be determined at compile time.
[6]The run was performed several times to make sure the timing is accurate.

the relative timings for each group of optimizations are very interesting.

Optimization Flags	User Time
none	2 min 48.674 sec
-O1	2 min 12.570 sec
-O2	2 min 20.114 sec
-O3	2 min 20.309 sec
-O3 -funroll-loops	2 min 18.888 sec
-O3 -funroll-loops -mcpu=athlon-xp	2 min 18.957 sec
-O3 -funroll-loops -march=athlon-xp	2 min 16.599 sec
-O1 -funroll-loops -march=athlon-xp -ffast-math	0 min 30.918 sec
-O2 -funroll-loops -march=athlon-xp -ffast-math	0 min 28.702 sec
-O3 -funroll-loops -march=athlon-xp -ffast-math	0 min 28.638 sec

A 79% decrease in run time was achieved through the use of `-ffast-math`, however, this option carries an ominous sounding warning in the gcc man pages:

> This option should never be turned on by any -O option since it can result in incorrect output for programs which depend on an exact implementation of IEEE or ISO rules/specifications for math functions.

However, this option is almost always safe to use. As a general rule of thumb, if you do not know the details of the IEEE floating point number format, and you are not writing code that relies on determining convergence by calculating precise error intervals (in other words code that just happens to use floating point numbers) then this option is safe to use. It may interfere with graceful handling of `inf` and `nan`, but all numerical code really ought to catch these conditions with signals (as outlined in section 3.4.2) and halt the program when they occur.

The `-mcpu` and `-march` options are processor specific optimization flags. The `-mcpu` flag optimizes the executable for a specific architecture but does not use instruction sets that are unique to that architecture. Thus, code compiled with `-mcpu` will run on entire families of architectures (e.g. IA32) but is tuned for a particular CPU. The `-march` flag produces stronger optimizations in the sense that it makes use of architecture dependent instruction sets: code compiled with `-march` will only work for the platform it was compiled for.

3.5.1 Crestplots

An alternative way of analyzing the time evolution behavior of the gravitational wavefunction is via "crestplots". After a run was finished, a Perl program was executed which found the most probable location of the particle[7] which amounts to finding the peak of $|r\psi(r)|^2$. The program then created a plot so the time evolution could be studied. The crestplot, along with a knowledge of Δt can be used to determine the speed and acceleration of the collapsing or expanding wavefunction. For example, Figure 4.3 shows the crestplot for a collapsing gravitational wavefunction (the red '`+`') and an expanding free particle wavefunction (the green '`x`') as a func-

[7]Or particles, if the free particle was also being plotted to provide something to compare the numerical wavefunction to.

tion of timestep. The horizontal axis is timestep and the vertical axis is distance from the origin in meters. In this figure, the free particle appears stationary, while the gravitational particle becomes more and more likely to be found precisely at the origin.

Chapter 4

Results

4.1 Summary of Numerical Results

The Schrödinger-Newton equation was repeatedly solved using a fixed Gaussian parameter $\alpha_0 = 5 \times 10^{16}$ m^{-2} with various particle masses. Each program run computed the wavefunctions for both the self-gravitating particle ψ_g and the analytically computed free particle ψ_{fp} with the same mass and initial condition. The free particle provided a known and familiar benchmark with which to compare ψ_g. The numerically computed self-gravitating wavefunction was observed to exhibit four main types of behaviors depending on the particle mass: stationary, seemingly chaotic, collapsing, and expanding.

The raw numerical results are presented in Table 4.1. In the next section I will go into the wavefunction behavior in more detail.

Mass	Behavior	T (ns)
Above 8.0×10^{-13} kg Above 4.8×10^{14} u	Stationary	N/A
5.0×10^{-13} kg to 8.0×10^{-13} kg 3.0×10^{14} u to 4.8×10^{14} u	Seemingly chaotic	N/A
$m_{0\,\mathrm{high}} = 4.9 \times 10^{-13}$ kg $m_{0\,\mathrm{high}} = 2.9 \times 10^{14}$ u	Largest mass to collapse	100.0
$m_{0\,\mathrm{low}} = 1.3 \times 10^{-23}$ kg $m_{0\,\mathrm{low}} = 7.8 \times 10^{3}$ u	Smallest mass to collapse	1.0
3.2×10^{-24} kg to 1.2×10^{-23} kg 1.9×10^{3} u to 7.2×10^{3} u	Seemingly chaotic	N/A
7.5×10^{-26} kg to 3.1×10^{-24} kg 4.5×10^{1} u to 1.9×10^{3} u	Slow expansion	N/A
Below 7.5×10^{-26} kg Below 4.5×10^{1} u	Indistinguishable from free particle	N/A

Table 4.1: Bounding masses that induce wavefunction collapse along with collapse time T.

4.2 The General Characteristics of Wavefunction Behavior

Below a certain mass, the numerical solutions to the Schrödinger-Newton equation generally resembled free particle solutions except within a mass range, where the solutions differed remarkably. Within a range of masses, ψ_g was observed to spontaneously localize at the origin, that is, wavefunction collapse due to self-gravitational interaction. At the fringes of this range of masses, the wavefunction would "dance" (for lack of a better term) as if "trying to decide" whether to follow the free particle solution or collapse to the origin.

4.2.1 Stationary Behavior

For relatively large mass ($m > m_{0\,\mathrm{high}}$), ψ_g appeared stationary for any reasonable number of timesteps, or moved at such a slow rate that the program would have to run for prohibitively long times to determine what ψ_g would eventually do. A stationary wavefunction may eventually collapse, expand, or even remain stationary—the final behavior is unknown.

Figure 4.1 shows a crestplot for a particle of mass 1.0×10^{-11} kg (6.0×10^{15} u) and Gaussian parameter $\alpha = \alpha_0$ as a function of timestep for a run in which ψ_g was stationary. The green curve (marked by "`x`") shows the free particle result while the red curve (marked by "`+`") shows the self-gravitating particle.

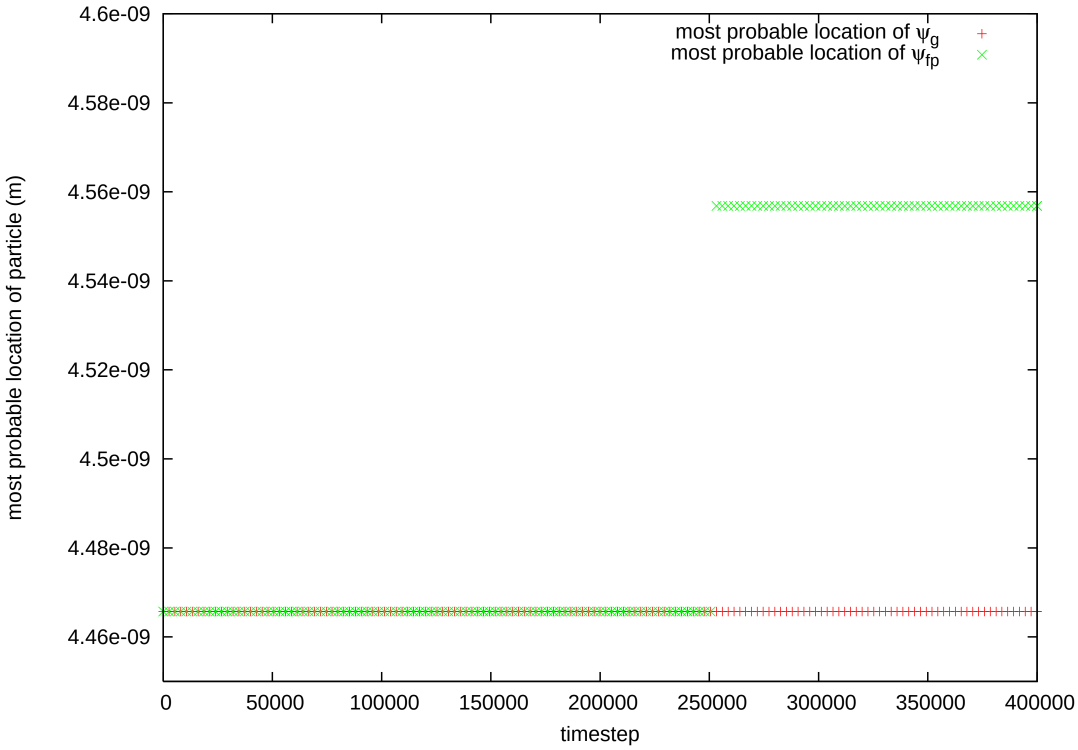


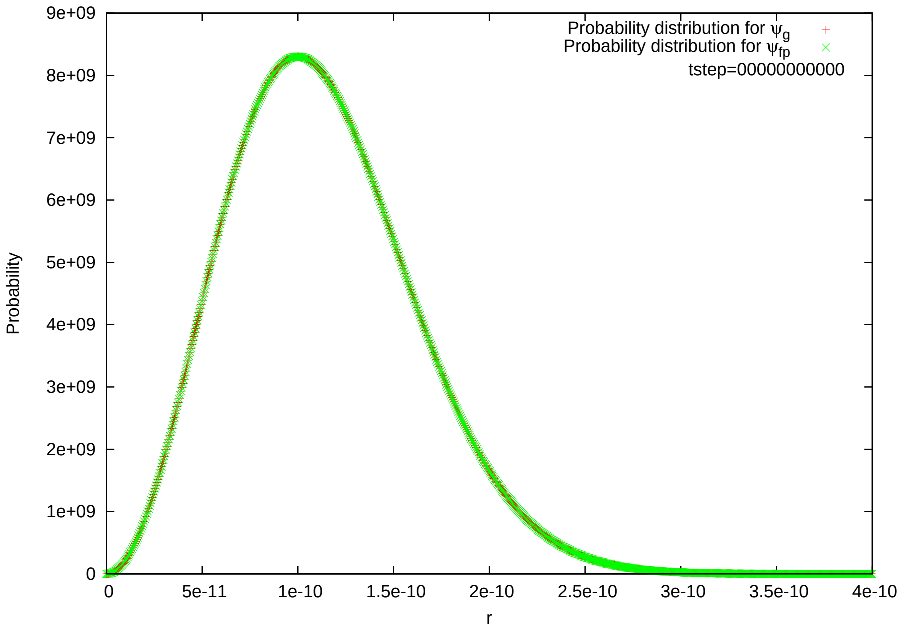
Figure 4.1: A crestplot showing the most probable particle location versus timestep for $m = 1.0 \times 10^{-11}$ kg (6.0×10^{15} u). The free particle expands while the self-gravitating particle remains stationary. Even for wavefunctions evolving under the Schrödinger-Newton equation that do not collapse, their expansion relative to a free particle is always retarded.

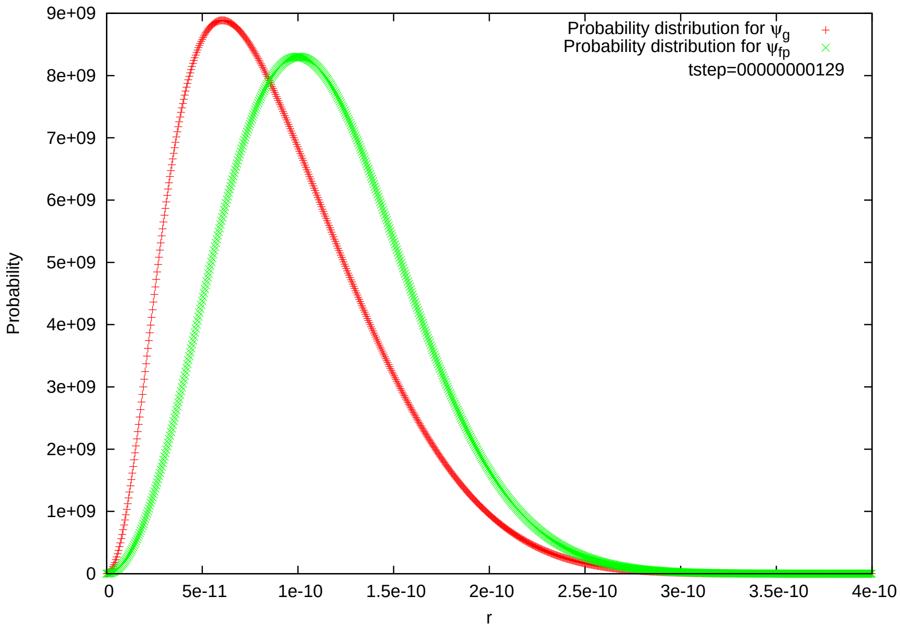
Part way though the simulation, ψ_{fp} spread by a miniscule amount whereas ψ_g remained stationary, and it's unclear how long one must wait for ψ_g to do anything. For results like this one, it will be said that the behavior of the Schrödinger-Newton equation is simply unknown for the pair (α_0, m).

4.2.2 Wavefunction Collapse

For intermediate masses ($m_{0\,\mathrm{low}} \leq m \leq m_{0\,\mathrm{high}}$), ψ_g was observed to collapse to the origin.

An important consideration while analyzing the results of the PDE Solver was the question of "what constitutes a collapse." When a collapsed wavefunction was suspected, the run was repeated multiple times with the number of spatial gridpoints increased. Wavefunction collapse for a given set of run parameters was declared if the most probable location of the particle reached the second spatial gridpoint ($r = \Delta r$) independent of the number of spatial gridpoints. Once the probability peak reached Δr, the numerical simulation could no longer be trusted, since $\hat{H}\psi(r_i, t)$ is a function of $\psi(r_{i-1}, t)$ and $\psi(r_{i+1}, t)$. The only way to get more information about the dynamics at the origin is to increase the number of spatial gridpoints by decreasing Δr (which can be achieved by increasing `N` or decreasing `rendratio`) and Δt. A fairly typical example of observed collapse is illustrated in Figure 4.2 which shows a collapse for $\alpha = 1.0 \times 10^{20}$ m^{-2}, $m = 1.0 \times 10^{-15}$ kg (6.0×10^{11} u).


Probability distribution for ψg
Probability distribution for ψfp
tstep=00000000000
Probability
r
9e+09
8e+09
7e+09
6e+09
5e+09
4e+09
3e+09
2e+09
1e+09
0
5e-11
1e-10
1.5e-10
2e-10
2.5e-10
3e-10
3.5e-10
4e-10


Probability distribution for ψg
Probability distribution for ψfp
tstep=00000000129
Probability
r
9e+09
8e+09
7e+09
6e+09
5e+09
4e+09
3e+09
2e+09
1e+09
0
5e-11
1e-10
1.5e-10
2e-10
2.5e-10
3e-10
3.5e-10
4e-10

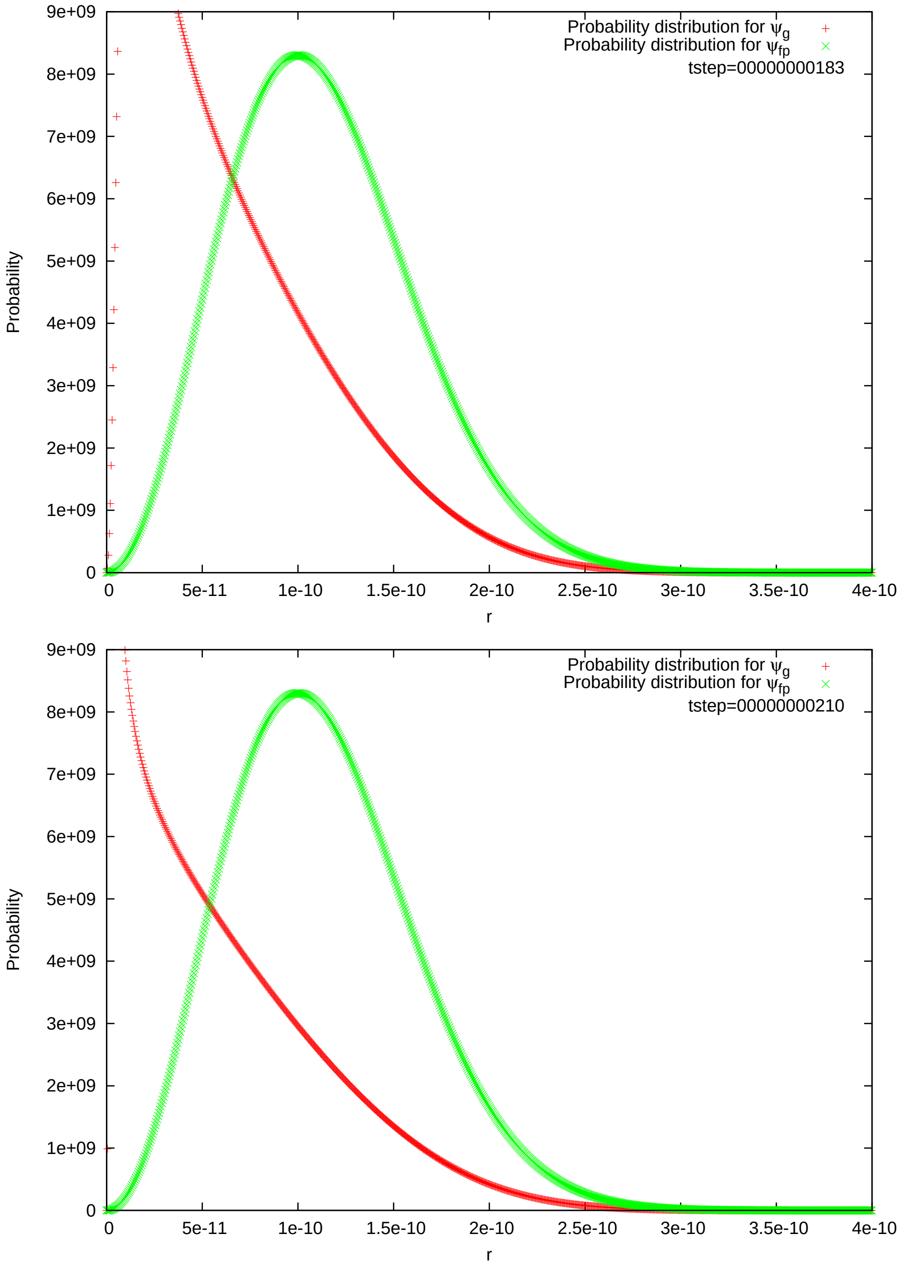


Figure 4.2: Profile of a typical collapsing wavefunction: the most probable location of the particle limits to $r = 0$ and the bulk of probability density shifts towards the origin.

Figure 4.3 shows the most probable location of ψ_g and $\psi_{\rm fp}$ as a function of timestep for the same collapsing system shown in Figure 4.2. The change in concavity for ψ_g is interesting in that it suggests that as the probability distribution gets closer to the origin, some effect becomes larger that tries to prevent further collapse. Whether a self-gravitating particle actually collapses or not depends on how large this effect becomes.

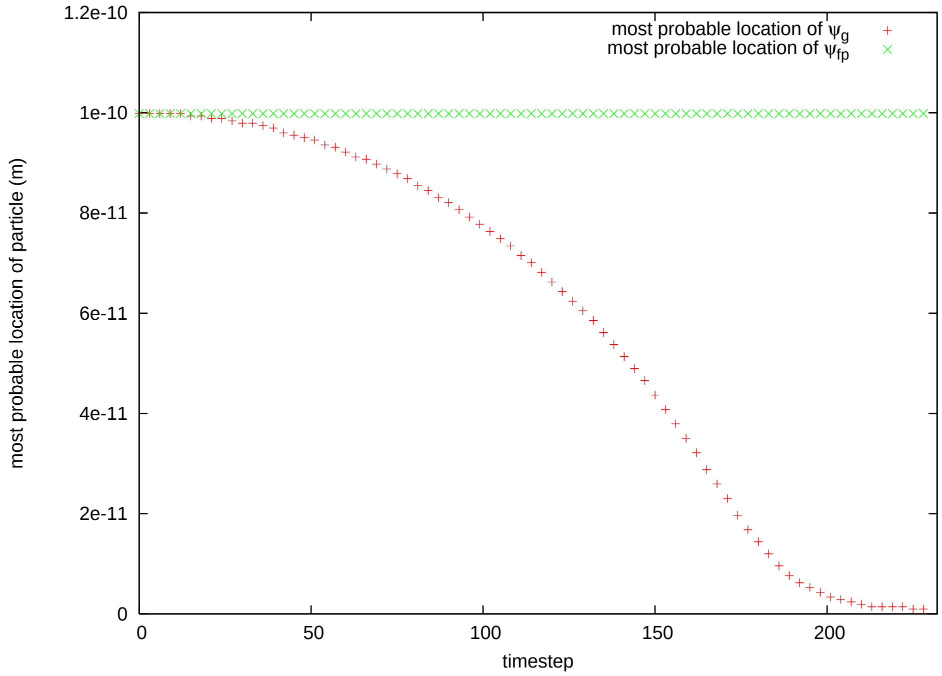
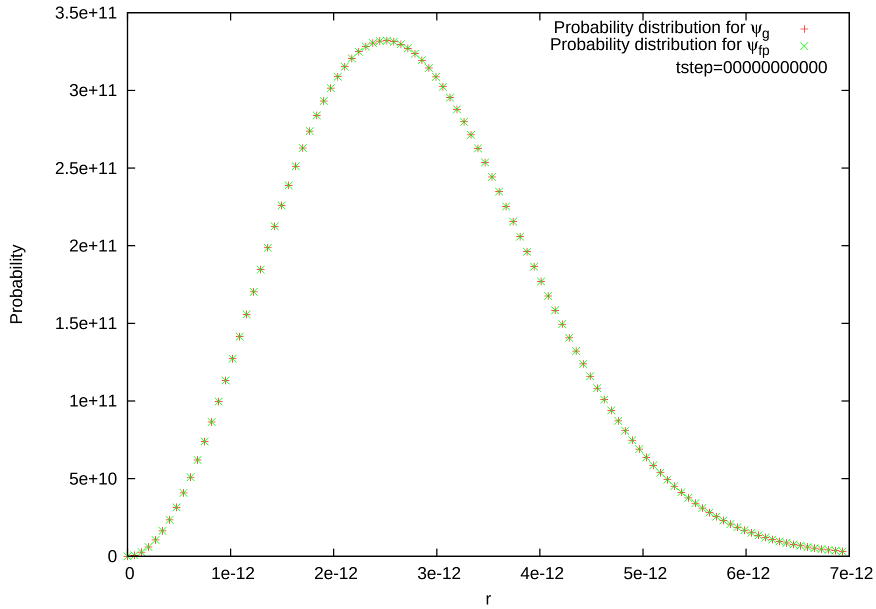
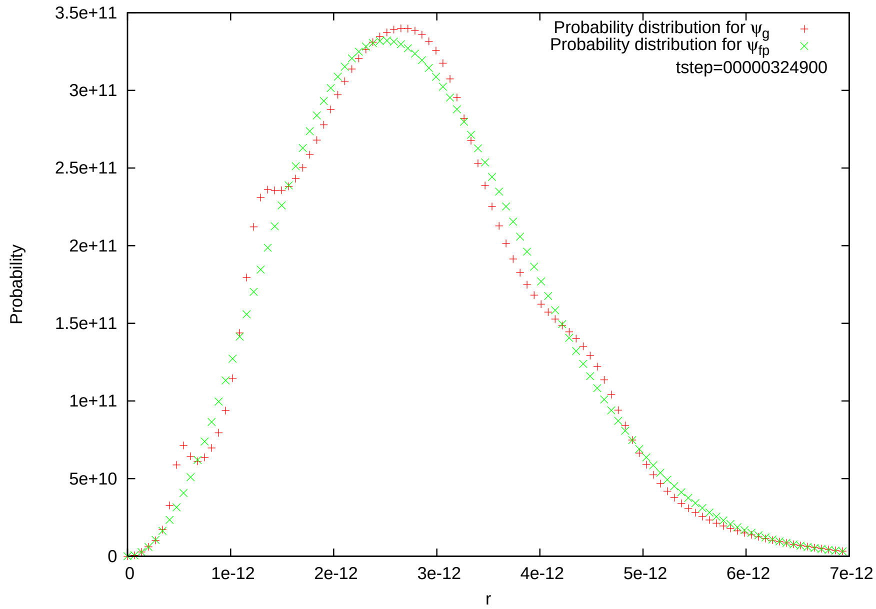


Figure 4.3: Crestplot for the collapsing system shown in Figure 4.2. The change in concavity of the graph suggests that as the wavefunction collapses, some effect that retards collapse tends to grow larger.

4.2.3 Seemingly Chaotic Behavior

At constant α_0, as the particle mass is decreased to approach $m_{0\,\mathrm{high}}$ from above, or is increased to approach $m_{0\,\mathrm{low}}$ from below, ψ_g is observed to exhibit fluctuating behavior. This behavior can be characterized in any number of ways: "dancing" about a single point, oscillation of height, or travelling waves riding on top of the wavefunction's profile. In most cases, it looks like the wavefunction is "trying to decide" whether to collapse or follow the free particle solution. One example of this behavior can be seen in Figure 4.4. In this case, the most probable particle location oscillates, at first slowly but with rapidly increasing amplitude, about the point $r_0 \approx 2.5 \times 10^{-12}$ m. Additionally, small waves of probability form at the peak and travel in both directions from r_0, riding the probability distribution's profile like small signals on top of a carrier wave. These probability waves increase in number and height until ψ_g takes on the appearance of a fractal.


Probability distribution for ψ_g
Probability distribution for ψ_{fp}
tstep=00000000000
Probability
r
0
5e+10
1e+11
1.5e+11
2e+11
2.5e+11
3e+11
3.5e+11
1e-12
2e-12
3e-12
4e-12
5e-12
6e-12
7e-12


Probability distribution for ψ_g
Probability distribution for ψ_{fp}
tstep=00000324900
Probability
r
0
5e+10
1e+11
1.5e+11
2e+11
2.5e+11
3e+11
3.5e+11
1e-12
2e-12
3e-12
4e-12
5e-12
6e-12
7e-12

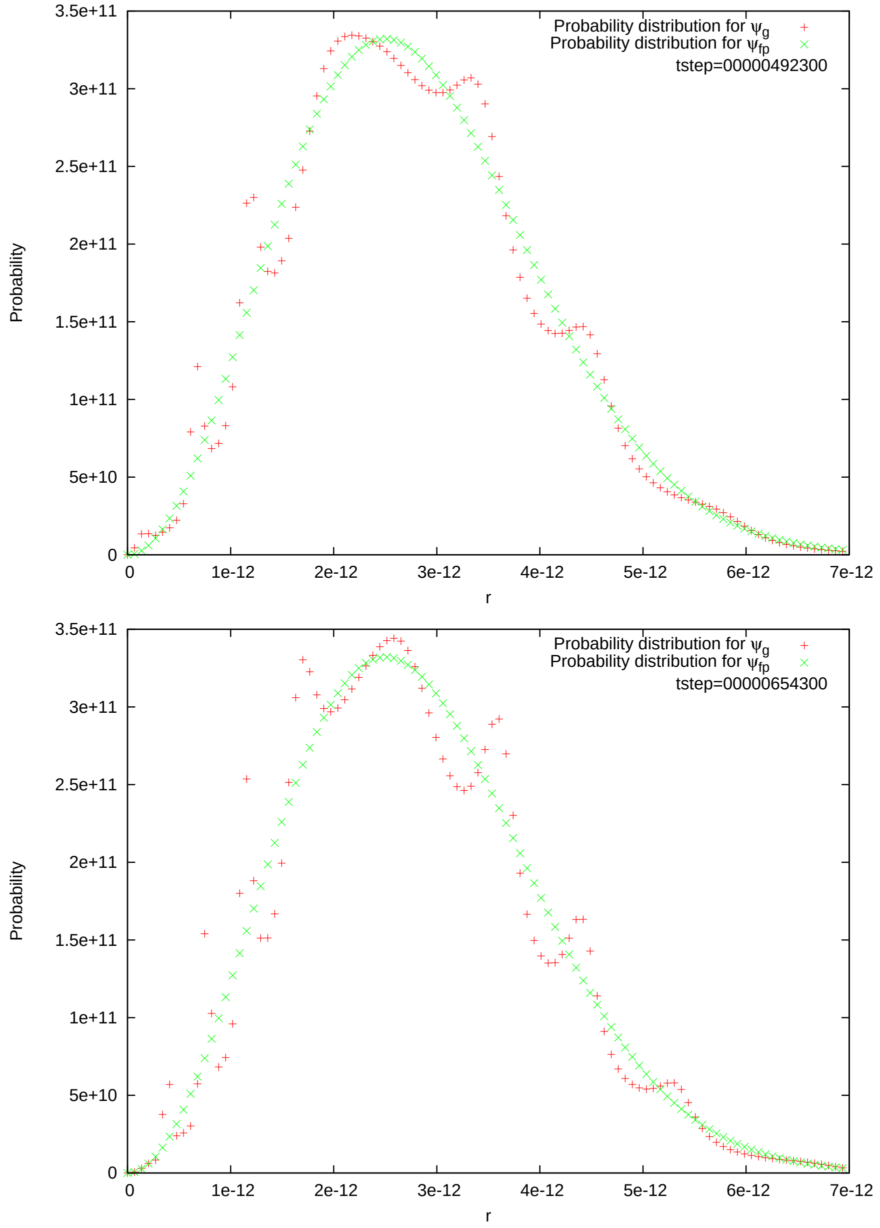


Figure 4.4: Gravitational wavefunction becoming more unpredictable with time, superimposed on a free particle wavefunction of the same mass and initial condition. The number and intensity of peaks increase with time until the wavefunction takes on the appearance of a fractal (everywhere continuous, nowhere differentiable).

A crestplot of this particular example of seemingly chaotic behavior, displayed in Figure 4.5, doesn't show the small signal waves of probability, but does show oscillation with increasing amplitude of the main probability distribution.

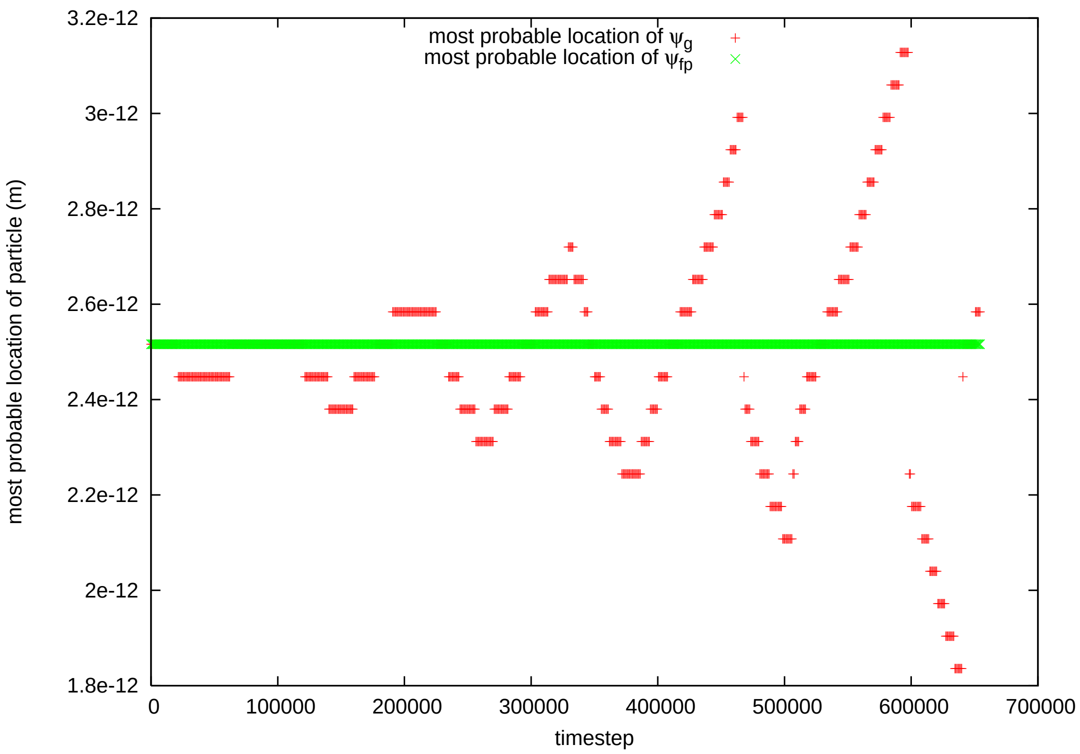
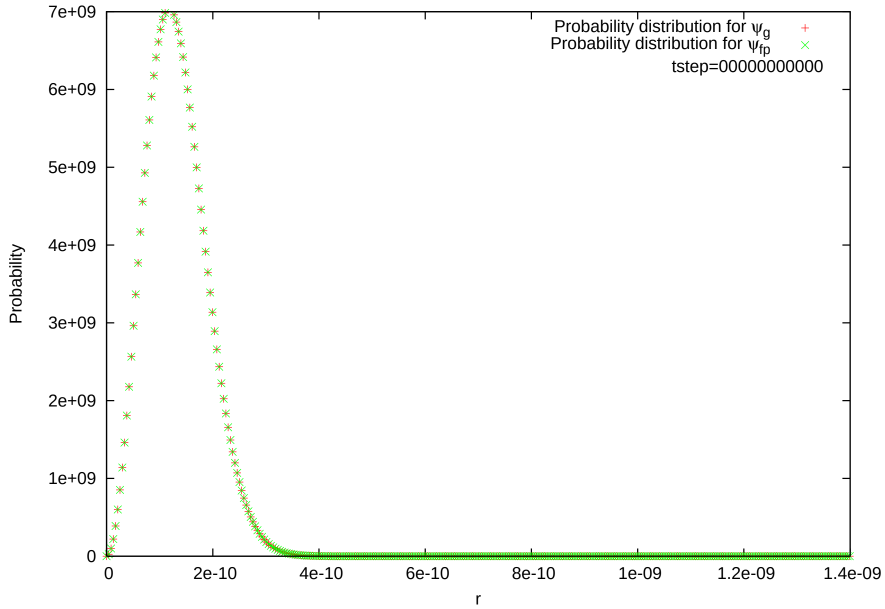


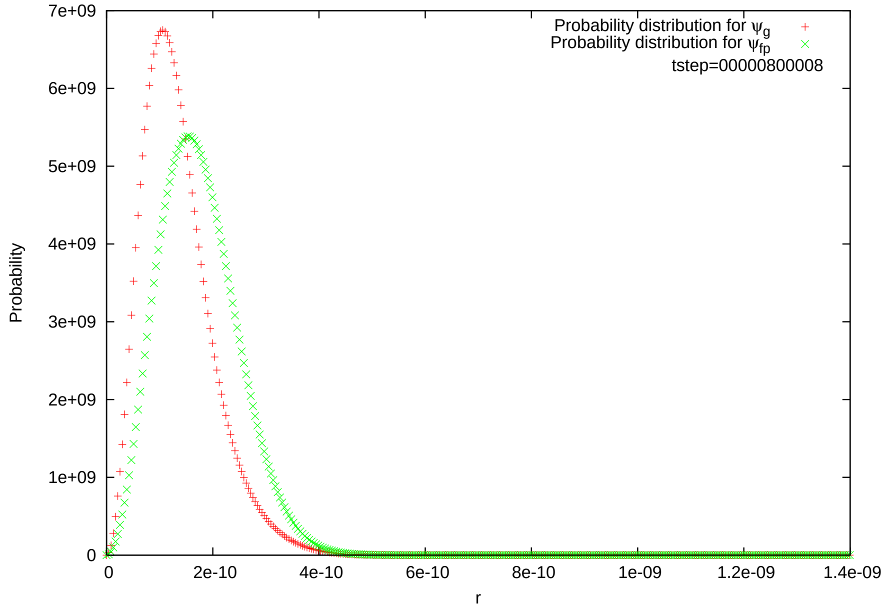
Figure 4.5: Crestplot for the seemingly chaotic dynamics illustrated in Figure 4.4. The most probable particle location oscillates about the free particle's peak. The amplitude of these oscillations increases with timestep until the simulation's profile is no longer recognizable as a wavefunction.

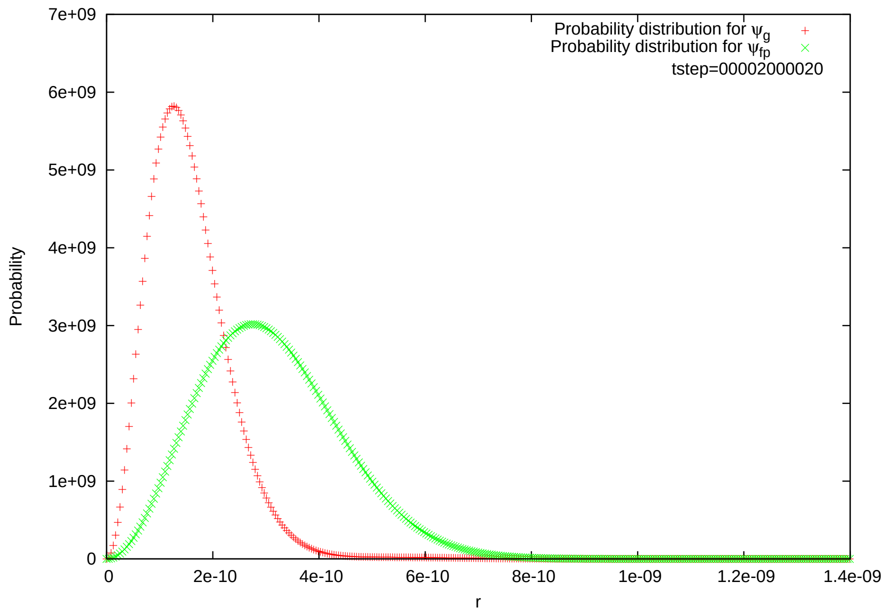
I will not comment deeply on the physical relevance of the observed fluctuating behavior. If semiclassical gravity is indeed a true theory of gravitation, there may be new and profoundly interesting physics here, but it may also be the case that this behavior is a numerical artifact. In any event, it appears that the (α, m) pairs that induce seemingly chaotic behavior may be exhibiting some kind of instability or resonance in the solution space of the Schrödinger-Newton equation. It would be interesting to investigate whether this behavior could be reproduced with different types of initial conditions.

4.2.4 Expansion

For $\alpha = \alpha_0$, below $m = m_{0\,\mathrm{low}}$, ψ_g expands. For masses close to $m_{0\,\mathrm{low}}$, this expansion is slower than the expansion for a free particle wavefunction of the same mass and initial condition, as demonstrated in Figure 4.6. In this figure, ψ_g and ψ_{fp} are plotted together to compare their behavior. At the beginning of the simulation, they have the same initial condition, but ψ_g expands less rapidly. The potential well of ψ_g retards the expansion, but for this (α, m) pair, the walls of the potential well are not steep enough to induce collapse.


Probability distribution for ψ_g
Probability distribution for ψ_{fp}
tstep=00000000000
Probability
r
0
1e+09
2e+09
3e+09
4e+09
5e+09
6e+09
7e+09
0
2e-10
4e-10
6e-10
8e-10
1e-09
1.2e-09
1.4e-09


Probability distribution for ψ_g
Probability distribution for ψ_{fp}
tstep=00000800008
Probability
r
0
1e+09
2e+09
3e+09
4e+09
5e+09
6e+09
7e+09
0
2e-10
4e-10
6e-10
8e-10
1e-09
1.2e-09
1.4e-09



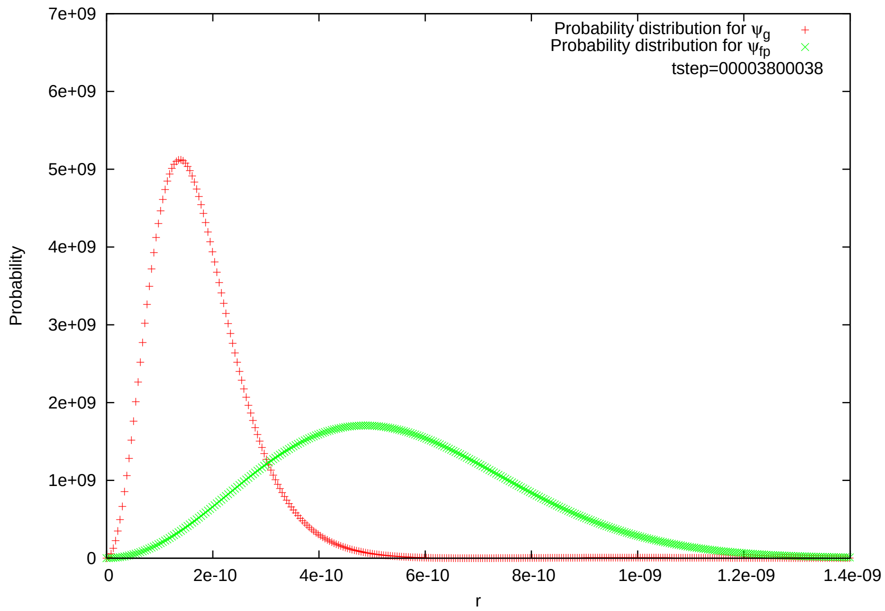


Figure 4.6: Self-gravitating particle lagging behind a free particle wavefunction. As the mass decreases, the gravitational wavefunction becomes indistinguishable from a free particle. However, as the particle mass increases, the expansion lag between the gravitational and free particle increases until the gravitational wavefunction begins to collapse towards the origin.

It is interesting to look at Figure 4.7, the crestplot for this run, which shows the most probable location of each particle as a function of timestep.

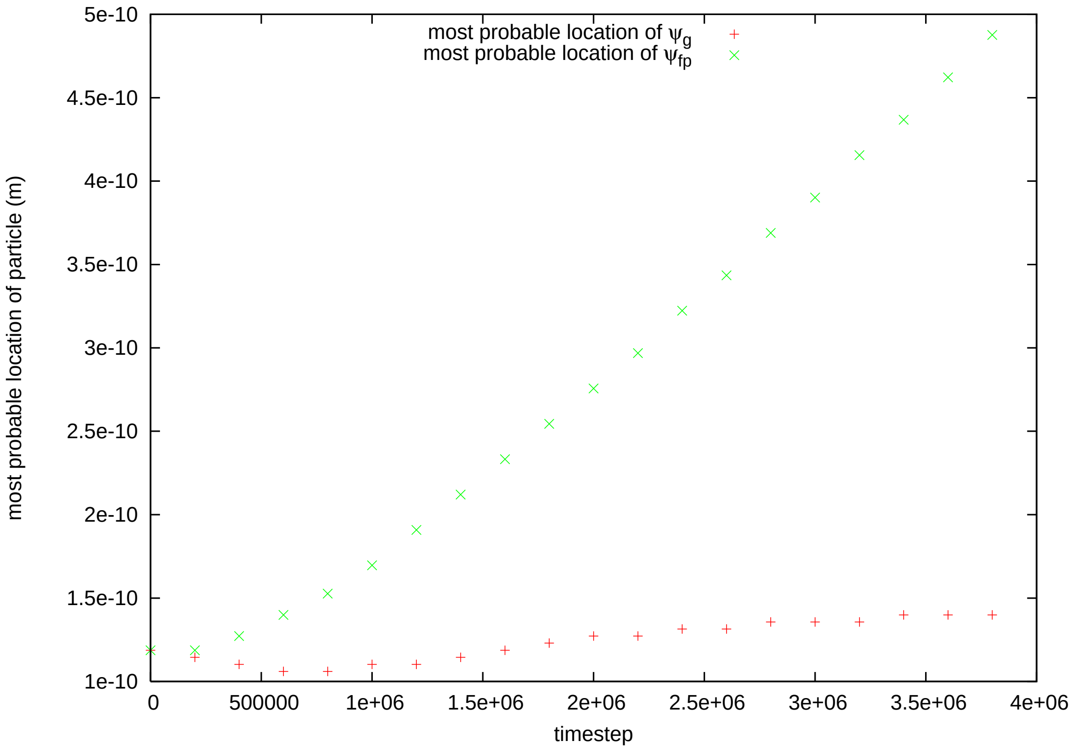


Figure 4.7: Crestplot for self-gravitating particle lagging behind a free particle wavefunction. Note the initial short-lived collapse before the lagging expansion occurs. This is a fairly typical property of all Schrödinger-Newton wavefunctions that expand.

As expected, both wavefunctions expand, with ψ_g "lagging" behind ψ_{fp}, however, there is one remarkable feature which isn't readily apparent from time evolution plot. The self-gravitating particle actually begins to collapse towards the origin, but at around timestep 600,000, reverses itself and begins to expand. This is actually a general feature of masses near $m_{0\,\text{low}}$. For smaller particle masses, this "bounce" at the origin ceases (to within the precision of the simulation). At masses much lower than $m_{0\,\text{low}}$, ψ_g becomes numerically indistinguishable from the free particle wavefunction. This makes sense, since the smaller the particle's mass is, the more shallow the potential well will be.

4.3 Generalizing the Results

As discussed in section 4.1, a Gaussian initial condition under the Schrödinger-Newton equation collapsed to the origin for a range of masses between $m_{0\,\text{high}}$ and $m_{0\,\text{low}}$. For masses *just* outside this range, the evolution of ψ_g was observed to be seemingly chaotic. For masses above this range, ψ_g was stationary (meaning unknown eventual behavior). Lastly, below this range, ψ_g expanded. This expansion was slower than the free particle, but in the limit as $m \to 0$, $\psi_g \to \psi_{\text{fp}}$ since the m^2 term in the potential ensures that the Schrödinger-Newton equation reduces to the free particle equation in the low mass limit.

These results were obtained for a single Gaussian parameter α_0. To get a full understanding of the solution space of the Schrödinger-Newton equation, we would need to run the program for all masses for every α. Unfortunately, the non-linear nature of this equation means that we lose a lot of the mathematical muscle that has been developed for the study of differential equations; in particular, we cannot form new solutions from superpositions of known solutions. It also means we cannot gain insight into solutions for different initial conditions, since initial conditions that are "close" are not guaranteed to yield solutions that are also "close".

However, the scaling properties of the equation turn out to be of immense help here. From the scaling properties of the Schrödinger-Newton equation, discussed in appendix A, we can use eq. (A.5) to view $m_{0\,\text{high}}$ and $m_{0\,\text{low}}$ as functions of α. This gives two curves, $m_{\text{high}}(\alpha)$ and $m_{\text{low}}(\alpha)$:

$$m_{\text{high}}(\alpha) = m_{0\,\text{high}} \left(\frac{\alpha}{\alpha_0}\right)^{1/6} \qquad m_{\text{low}}(\alpha) = m_{0\,\text{low}} \left(\frac{\alpha}{\alpha_0}\right)^{1/6} \tag{4.1}$$

We can use these equations and the α_0 results to obtain mass ranges for stationary, fluctuating, collapsing, and expanding ψ_g for arbitrary α. This enables us draw Figure 4.8, a qualitative[1] graph of α versus m, displaying the regions of the various behaviors.

The vertical dashed line represents $\mu = 1$: the actual numerical data taken at $\alpha = \alpha_0$. Changing α (moving horizontally in the figure) means changing the scale parameter μ according to $\alpha = \mu^6 \alpha_0$. The two curves represent the scaled values of $m_{0\,\text{high}}$ and $m_{0\,\text{low}}$ for various values of μ (or α). The gravitational wavefunction collapses for all (α, m) pairs that fall between these two curves.

[1]The curves are not drawn to scale.

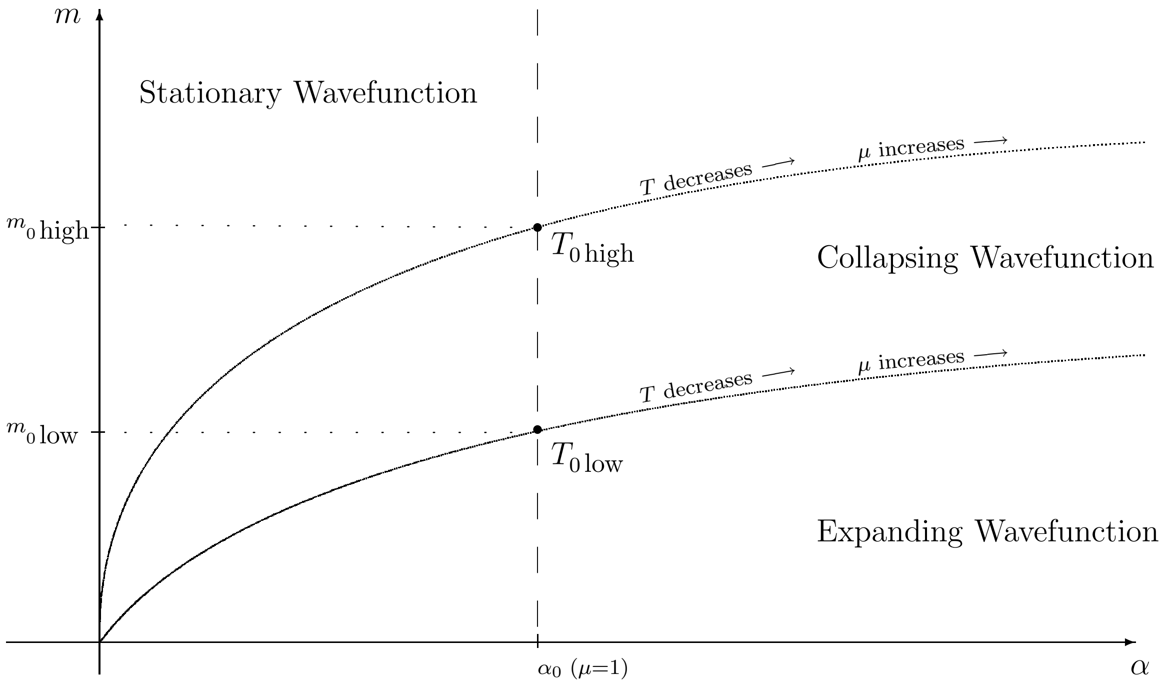


Figure 4.8: The (α, m) solution space for the Schrödinger-Newton equation exhibiting the regions of behavior. The dashed vertical line represents the data taken at constant $\alpha = \alpha_0$ for various masses. Any (α, m) pair that falls between the two curves yields a solution that collapses to the origin.

4.4 Visualizing Constant Mass

A researcher wanting to perform an experiment that tests the validity of the predictions of semiclassical gravity is not interested in constant α for various masses. In practice, a researcher would have a particular object in mind, like a C_{70} molecule, and would want to know what kind of behavior semiclassical gravity predicts as a function of α. Rather than being interested in vertical lines (constant α, varying m) of Figure 4.8, he would be interested in horizontal lines (constant m, varying α).

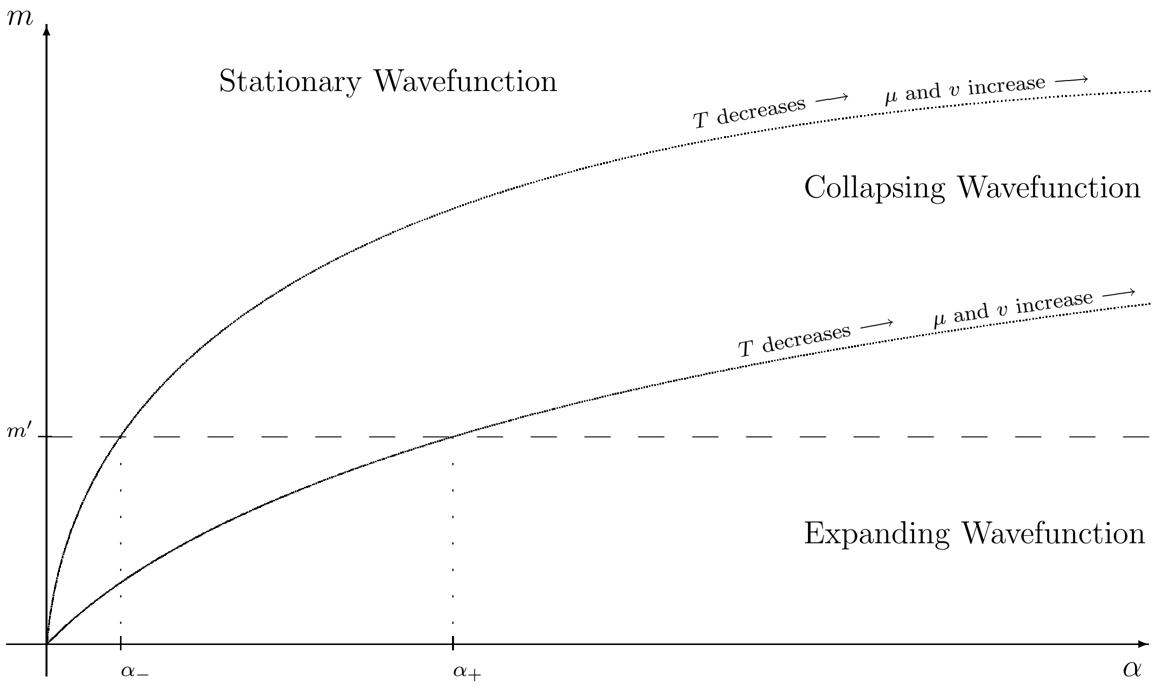


Figure 4.9: The (α, m) solution space for the Schrödinger-Newton equation exhibiting the regions of behavior. This diagram emphasizes the solution space for a line of constant mass, which is what a researcher doing a matter-diffraction experiment would be interested in.

From Figure 4.9 we can see the predictions that semiclassical gravity makes for a particle wavefunction for a mass m':

- For $\alpha < \alpha_-$, my research makes no prediction.
- For $\alpha_- \leq \alpha \leq \alpha_+$, the particle wavefunction should collapse.
- For $\alpha > \alpha_+$, the particle wavefunction should expand.

The numerical values of $\alpha_\pm$ can be obtained by inverting eq. (4.1):

$$\alpha_-(m') = \left(\frac{m'}{m_{0\,\mathrm{high}}}\right)^6 \alpha_0 \qquad \alpha_+(m') = \left(\frac{m'}{m_{0\,\mathrm{low}}}\right)^6 \alpha_0$$

which, stated numerically in SI units,

$$\alpha_-(m') = \left(3.6 \times 10^{90}\ \mathrm{kg}^{-6}\mathrm{m}^{-2}\right)\ m'^6 \qquad \alpha_+(m') = \left(1.0 \times 10^{54}\ \mathrm{kg}^{-6}\mathrm{m}^{-2}\right)\ m'^6$$

and in unified atomic mass units:

$$\alpha_-(m') = \left(8.4 \times 10^{-71}\ \mathrm{u}^{-6}\mathrm{m}^{-2}\right)\ m'^6 \qquad \alpha_+(m') = \left(2.2 \times 10^{-7}\ \mathrm{u}^{-6}\mathrm{m}^{-2}\right)\ m'^6$$

This leads to the pragmatic question: "Physically, what is α?" As discussed in section 2.7.1, there are a few good guesses one can make, however, there appears to be a definitive answer to this question.

As discussed in section 1.5, the most recent experiments in matter-wave diffraction have used a Talbot-Lau interferometer in which a wave (here, a matter-wave) impinging on a diffraction grating displays the Talbot effect: an aliased image of the diffraction grating is created at a precise distance from the diffraction grating given by:

$$L_T = \frac{d^2}{\lambda}$$

where d is the grating constant (spacing between the slits) and λ is the wavelength of the impinging wave. Getting this distance correct is critical to the success of the recent matter-wave experiments, since if it's wrong, the aliased image of the diffraction grating will not occur. In all cases, the correct Talbot distance was obtained using the de Broglie wavelength. Therefore, the width of a particle represented by a Gaussian wavefunction of parameter α is precisely the de Broglie wavelength:

$$w \approx \frac{1}{\sqrt{\alpha}} = \lambda_{\text{de Broglie}} = \frac{h}{mv}$$

which is interesting because it tells us that the width of the initial probability distribution is completely dependent on the particle's mass and speed and has no dependence on the physical extent of the particle. The connection between the width of the wavefunction and the de Broglie wavelength indicates that although we've viewed the wavefunction behavior as a function of (α, m), we could also view the wavefunction behavior as a function of (v, m), where v is the speed of the particle as it enters the interferometer. Figure 4.10 shows this relation.

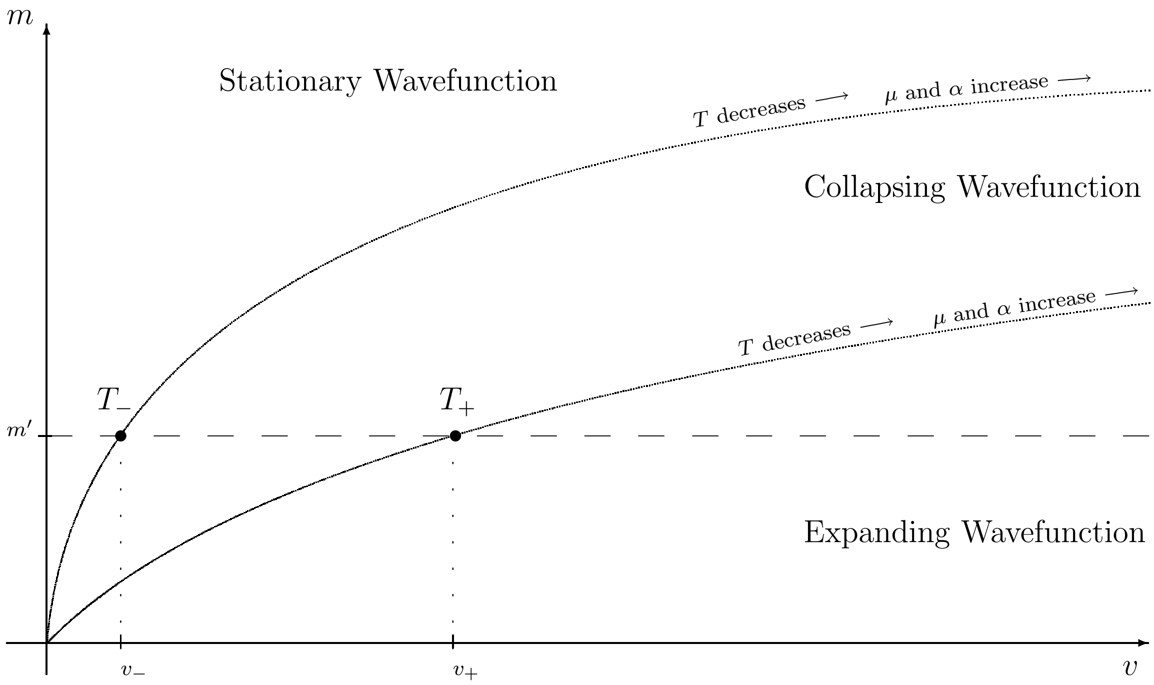


Figure 4.10: The (v, m) solution space for the Schrödinger-Newton equation exhibiting the regions of behavior. This diagram emphasizes the equivalence between the Gaussian parameter α and particle speed v for a line of constant mass, which is what a researcher doing a matter-diffraction experiment would be interested in. This type of diagram would be used to obtain the collapse times for the particle travelling at the boundary collapse speeds.

By the de Broglie equation, $v_\pm$ are given by:

$$v_-(m') = \frac{h}{m'}\left(\frac{m'}{m_{0\,\mathrm{high}}}\right)^3\sqrt{\alpha_0} \qquad v_+(m') = \frac{h}{m'}\left(\frac{m'}{m_{0\,\mathrm{low}}}\right)^3\sqrt{\alpha_0}$$

It may be more informative to restate this key result numerically in terms of SI units,

$$v_-(m') = \left(1.3\times10^{12}\ \mathrm{m/kg^2\cdot s}\right)m'^2 \qquad v_+(m') = \left(6.7\times10^{43}\ \mathrm{m/kg^2\cdot s}\right)m'^2$$

or, for researchers who think in terms of atomic masses, the unified atomic mass unit:

$$v_-(m') = \left(3.5\times10^{-42}\ \mathrm{m/u^2\cdot s}\right)m'^2 \qquad v_+(m') = \left(1.9\times10^{-10}\ \mathrm{m/u^2\cdot s}\right)m'^2$$

One last issue to address would be the time it takes for the collapse, if a collapse happens. We can easily find the collapse times T_- and T_+ at the collapse behavior boundary speeds v_- and v_+ as visualized on Figure 4.10. Researchers interested in the collapse times for speeds between v_- and v_+ would need to investigate that area of the parameter space more thoroughly. However, it turns out that this may not be necessary, as will be demonstrated shortly. The boundary times for collapse can be obtained from the time scaling discussed in appendix A:

$$T = \mu^{-5}\,T_0$$

We can re-express the scale factor to write down expressions for the collapse time at the collapse boundaries in terms of α,

$$T_+ = \left(\frac{\alpha_0}{\alpha_+}\right)^{(5/6)} T_{0\,\mathrm{low}} \qquad T_- = \left(\frac{\alpha_0}{\alpha_-}\right)^{(5/6)} T_{0\,\mathrm{high}}$$

or in terms of mass,

$$T_+ \;=\; \left(\frac{m_{0\,\mathrm{low}}}{m'}\right)^5 T_{0\,\mathrm{low}} \qquad\qquad T_- \;=\; \left(\frac{m_{0\,\mathrm{high}}}{m'}\right)^5 T_{0\,\mathrm{high}}$$

or using the de Broglie relation, speed:

$$T_+ \;=\; \alpha_0^{5/6} \left(\frac{h}{m'\, v_+}\right)^{5/3} T_{0\,\mathrm{low}} \qquad\qquad T_- \;=\; \alpha_0^{5/6} \left(\frac{h}{m'\, v_-}\right)^{5/3} T_{0\,\mathrm{high}}$$

As an example, in a recent matter-wave experiment performed by Zeilinger et. al [50], Fluorofullerenes ($C_{60}F_{48}$) with atomic mass of 1632 u were sent through a Talbot-Lau interferometer at a mean speed of 105 m/s. This translates into a de Broglie wavelength of 2.33 pm.

The results of this research indicate that the wavefunction of $C_{60}F_{48}$ should collapse between a particle speed of $v_- = 4.4 \times 10^{-11}$ m/s and $v_+ = 1.6 \times 10^0$ m/s, two orders of magnitude slower than the particle's mean speed in the laboratory. The collapse times for these speeds is very small. At v_-, the predicted collapse time is 4×10^{-13} s and at v_+, the predicted collapse time is 4.0×10^{-15} s.

The fact that the $C_{60}F_{48}$ molecules did *not* localize is not a case for or against semiclassical gravity since their mean speed was greater than v_+. Semiclassical gravity predicts that these molecules should exhibit quantum mechanical behavior at the mean speed used in the interferometry experiment, 105 m/s. But perhaps a proposal can be made to settle the matter on the theory of Møller and Rosenfeld once and for all.

4.5 Proposal for the Experimental Confirmation of Semiclassical Gravity

In light of the successes of the Talbot-Lau interferometer in matter-wave experiments, there have been proposals to perform such experiments with even larger masses. In 1997, Clauser concluded [46] that Talbot-Lau interferometry may be applied to particles of mass 10^8 nucleons (10^{-19} kg) and even small viruses in the near future. In 2002, Zeilinger et. al. wrote [48] that matter-wave experiments with objects the size of a virus are at the extreme edge of currently available technology. In 2005, Zeilinger and Arndt wrote that matter-wave interferometry for objects such as proteins, small viruses and nanocrystals with atomic mass of up to 10^6 units (10^{-21} kg) should be feasible. While these experiments have not been performed yet, it appears that they certainly will be in the near future. [45, 46, 48, 51].

Recently, in a joint effort by groups at Cornell University and Tel Aviv University, the mass of DNA strand composed of 1578 base pairs was detected to be 999×10^3 u (1.6×10^{-21} kg). [52] If a DNA strand of this mass could be used in a matter diffraction experiment, according to this research, its wavefunction would collapse for speeds between:

If a researcher manages to perform such a diffraction experiment with the speed of the DNA strand approaching anything similar to the mean speed of the various fullerene experiments, semiclassical gravity predicts that the DNA wavefunction

Collapse speed		Collapse time	
v_+	1.84×10^{2} m/s	T_+	3.0×10^{-20} s
	3.0×10^{-0} m/s		3.3×10^{-17} s
v_-	3.43×10^{-30} m/s	T_-	2.3×10^{35} s

Table 4.2: Collapse speed and time range for a DNA strand of mass 999×10^{3} u.

should collapse in its own potential well. In other words, no quantum interference should be observable between these speeds.

Such an experiment is said to be feasible with current technology. On the other hand, the proposed diffraction experiment involving a virus appears to have even better prospects because of a better range of collapse speeds. The mass of a virus is nominally listed as 6.0×10^{6} u (1.0×10^{-20} kg). According to this research, its wavefunction should collapse for speeds between:

Collapse speed		Collapse time	
v_+	6.7×10^{3} m/s	T_+	3.7×10^{-24} s
	1.2×10^{2} m/s		3.4×10^{-21} s
v_-	1.3×10^{-28} m/s	T_-	2.8×10^{31} s

Table 4.3: Collapse speed and time range for a virus of mass 6.0×10^{6} u.

The wavefunction for a virus sent through a Talbot-Lau interferometer at these collapse speeds should collapse in the specified amount of time. In other words, semiclassical gravity predicts that at these speeds, within the corresponding time, the virus should behave as a classical, not quantum mechanical object. If quantum mechanical interference *were* to be observed at a point beyond the corresponding collapse time, then the Schrödinger-Newton equation would no longer be predicting the general behavior of the viral wavefunction, and therefore, semiclassical gravity could be decisively ruled out as a theory of gravity.

4.6 Conclusion

As an explanation for why quantum interference is suppressed in the macroscopic world, the role of gravity can not be discounted, but it does seem less likely.

If the solutions to the Schrödinger-Newton equation collapsed unconditionally for all masses greater than some critical mass at a given width, then the role of gravity in explaining why quantum interference is not observed in the macroscopic world would be very convincing. The equation is non-linear, and it *may* be the case that for different types of initial conditions the solutions *do* unconditionally collapse in the "large mass" range. However, the Gaussian wavefunction is a very reasonable single-particle wavefunction, and the fact that collapse only occurs for a finite range of masses makes gravitation an unlikely candidate to explain the boundary between the classical and quantum world. On the other hand, the model used here—that of a single self-gravitating but otherwise completely non-interacting particle—is highly unrealistic. Therefore, nothing definitive can be said. It would be interesting to investigate a model for two or more interacting *and* self-gravitating particles, as with eq. (1.6). To date, this has not been investigated.

Experiment is the final judge of a physical theory, and based on the recent matter diffraction experiments, there is no direct evidence against semiclassical gravity as a quantum theory of gravitation. To date, the mass and speeds used in the experiments fall within the non-classical region of the Schrödinger-Newton solution space. However, technology and experimental techniques have been steadily improving. The expected upper mass with such experiments has gone from "10^8 nucleons in the near future" to "10^6 u being feasible". As it stands, an experiment involving objects with the mass of a DNA strand or a virus is, while a monumentally difficult experiment to perform, feasible. And such an experiment may be used to refute the semiclassical theory of Møller and Rosenfeld.

Appendix A

Scaling Properties of the Schrödinger-Newton Equation

The Schrödinger-Newton equation is given by:

$$-\frac{\hbar^2}{2m}\nabla^2\psi(\vec{\mathbf{r}},t) \;-\; G\,m^2\,\psi(\vec{\mathbf{r}},t)\int_V \frac{|\psi(\vec{\mathbf{r}},t)|^2}{|\vec{\mathbf{r}}-\vec{\mathbf{r}}'|}\,d^3r' \;=\; i\,\partial_t\,\psi(\vec{\mathbf{r}},t) \tag{A.1}$$

If we make a coordinate scaling using the dimensionless parameters ϵ, ζ, and η,

$$x \rightarrow \epsilon\,x \qquad m \rightarrow \zeta\,m \qquad t \rightarrow \eta\,t$$

then eq. (A.1) equation becomes:

$$-\frac{1}{\zeta\,\epsilon^2}\frac{\hbar^2}{2m}\nabla^2\psi(\vec{\mathbf{r}},t) \;-\; \frac{\zeta^2}{\epsilon}G\,m^2\,\psi(\vec{\mathbf{r}},t)\int_V \frac{|\psi(\vec{\mathbf{r}},t)|^2}{|\vec{\mathbf{r}}-\vec{\mathbf{r}}'|}\,d^3r' \;=\; \frac{1}{\eta}i\,\partial_t\,\psi(\vec{\mathbf{r}},t)$$

This equation, and its solutions, are left invariant under this coordinate scaling if we demand:

$$\frac{1}{\zeta\,\epsilon^2} \;=\; \frac{\zeta^2}{\epsilon} \;=\; \frac{1}{\eta} \tag{A.2}$$

which is a set of two independent relations among the three scale variables. This leaves one degree of freedom when choosing values for (ϵ, ζ, η). Stated differently, only *one* dimensionless parameter, which we'll call μ, is needed to specify the values of ϵ, ζ, and η, and these values are determined by solving eq. (A.2). It can be checked that eq. (A.2) is satisfied by choosing a unitless parameter μ such that:

$$\epsilon = \mu^{-3} \qquad \zeta = \mu \qquad \eta = \mu^{-5}$$

Therefore, the following coordinate scaling leaves eq. (A.1) invariant:

$$x_1 = \mu^{-3} x_0 \qquad m_1 = \mu\, m_0 \qquad t_1 = \mu^{-5} t_0 \tag{A.3}$$

We will not be concerned with specific values of position versus time in any specific solution. Rather, we will focus on how the general nature of solutions and their characteristic evolution times depend on particle mass and the width ($\sim 1/\sqrt{\alpha}$) of the initial wavefunction. Because of the invariance, each pair of values of m and α define an equivalence class of solutions parameterized by μ. We expect the nature of any solution not to depend separately on values of m and α, but on the equivalence class to which the solution belongs. Thus, it is most useful to express the scaling that leaves the Schrödinger-Newton equation invaiant as:

$$m_1 = \mu\, m_0 \qquad \alpha_1 = \mu^6 \alpha_0 \qquad t_1 = \mu^{-5} t_0 \tag{A.4}$$

Program runs which display wavefunction collapse can be characterized by 3 numbers:

m: Mass of the self-gravitating particle. Explicit in the equation.

α**:** Gaussian parameter of the initial condition: a measure of the particle's width, and therefore, speed via de Broglie's relation $\sqrt{\alpha} = mv/h$.

T: The time it takes for the wavefunction to collapse.

From the scaling eq. (A.4), we can obtain a relationship between these variables:

$$\mu^6 = \left(\frac{m_1}{m_0}\right)^6 = \frac{\alpha_1}{\alpha_0} = \left(\frac{T_0}{T_1}\right)^{6/5}$$

to obtain explicit relationships between each characteristic parameter of the numerical run:

$$\text{between } m \text{ and } \alpha: \quad m(\alpha) = m_0 \left(\frac{\alpha}{\alpha_0}\right)^{1/6} \qquad \alpha(m) = \alpha_0 \left(\frac{m}{m_0}\right)^6 \tag{A.5}$$

$$\text{between } m \text{ and } T: \quad m(T) = m_0 \left(\frac{T_0}{T}\right)^{1/5} \qquad T(m) = T_0 \left(\frac{m_0}{m}\right)^5 \tag{A.6}$$

$$\text{between } T \text{ and } \alpha: \quad \alpha(T) = \alpha_0 \left(\frac{T_0}{T}\right)^{6/5} \qquad T(\alpha) = T_0 \left(\frac{\alpha_0}{\alpha}\right)^{5/6} \tag{A.7}$$

This allows us to obtain many solutions from a single numerical solution. If a solution represented by (m_0, α_0, T_0) is obtained, then it is also the *same* solution for $(m_1, \alpha_1, T_1) = (\mu\, m_0, \mu^6\, \alpha_0, \mu^{-5}\, T_0)$. Stated another way, (m_0, α_0) and (m_1, α_1) have equivalent behavior at the predictable timescales $T_1 = \mu^{-5} T_0$.

Appendix B

Calculation of the Self-Potential

B.1 Analytic Considerations

The generator of the Schrödinger Newton dynamics is given by an integral describing a gravitational self-potential which arises from the coupling between the sources of stress energy and quantum fields.

$$I \;=\; \underset{\text{all space}}{\int\!\!\int\!\!\int} \frac{|\psi(\vec{\mathbf{r}}', t)|^2}{|\vec{\mathbf{r}} - \vec{\mathbf{r}}'|}\, d^3r'$$

Expanding the denominator in spherical harmonics[1]

$$I \;=\; 4\pi \int_0^\infty \int_0^\pi \int_0^{2\pi} \sum_{\ell=0}^{\infty} \sum_{m=0}^{\ell} \frac{|\psi(\vec{\mathbf{r}}', t)|^2}{2\ell+1}\, \frac{r_<^\ell}{r_>^{\ell+1}}\, Y_\ell^m(\theta,\phi)\, Y_\ell^{m*}(\theta',\phi')\, r'^2 \sin(\theta')\, dr'\, d\theta'\, d\phi'$$

[1] See [36] eq (3.70), pg. 102.

where $r_> = \max(r, r')$ and $r_< = \min(r, r')$.

Assuming spherical symmetry for $\psi(\vec{r}, t) = \psi(r, t)$, the only ϕ' dependence is a factor of $e^{im(\phi-\phi')}$ in the products of the spherical harmonics. So the integral over ϕ' vanishes for all m except $m = 0$. Performing the trivial ϕ' integration, and making the substitution $\mu' = \cos(\theta')$,

$$I \;=\; 2\pi \sum_{\ell=0}^{\infty} \int_0^{\infty} \int_{-1}^{1} |\psi(\vec{\mathbf{r}}', t)|^2 \, \frac{r_<^{\ell}}{r_>^{\ell+1}} \, P_\ell(\mu) \, P_\ell(\mu') \, r'^2 \, d\mu' \, dr' \tag{B.1}$$

Relating the Legendre polynomial to derivatives of Legendre polynomials,[2] and considering the terms different from $\ell = 0$ gives:

$$I_{\ell\neq 0} \;=\; 2\pi \sum_{\ell=1}^{\infty} P_\ell(\mu) \int_0^{\infty} \int_{-1}^{1} |\psi(\vec{\mathbf{r}}', t)|^2 \, \frac{r_<^{\ell}}{r_>^{\ell+1}} \, \frac{r'^2}{2\ell+1} \left(\frac{dP_{\ell+1}}{d\mu'} - \frac{dP_{\ell-1}}{d\mu'} \right) d\mu' \, dr'$$

Integrating over μ', and using[3] $P_n(\pm 1) = (\pm 1)^n$, we find that the integrand is identically zero for all $\ell \neq 0$. Going back to eq. (B.1) and recalling that $P_0(\mu) = 1$, we find:

$$I_{\ell=0} \;=\; 4\pi \int_0^{\infty} |\psi(\vec{\mathbf{r}}', t)|^2 \, \frac{r'^2}{r_>} \, dr' \;=\; \frac{4\pi}{r} \int_0^{r} |\psi(\vec{\mathbf{r}}', t)|^2 \, r'^2 \, dr' \,+\, 4\pi \int_r^{\infty} |\psi(\vec{\mathbf{r}}', t)|^2 r' \, dr'$$

[2] See [36] eq. (3.28), pg. 89.

[3] See [36], pg. 87.

So the gravitational self potential energy is given by:

$$\begin{aligned} V_g(r,t) &= -Gm^2 \underset{\text{all space}}{\int\!\!\int\!\!\int} \frac{|\psi(\vec{\mathbf{r}}',t)|^2}{|\vec{\mathbf{r}}-\vec{\mathbf{r}}'|}\, d^3r' \\ &= -4\pi Gm^2 \left(\frac{1}{r}\int_0^r |\psi(\vec{\mathbf{r}}',t)|^2\, r'^2\, dr' \;+\; \int_r^\infty |\psi(\vec{\mathbf{r}}',t)|^2\, r'\, dr' \right) \end{aligned} \tag{B.2}$$

One might worry about the 2nd term of eqn(B.2); after all, the electrostatic field of a point charge at r is dependent only on charge at $r' \leq r$, but the 2nd integral is clearly an integral over $r' \geq r$. However, there is no problem here. Changing notation to the electrostatic equivalent,

$$V_e(r,t) = \frac{4\pi}{r}\int_0^r \rho(r')\, r'^2\, dr' + 4\pi \int_r^\infty \rho(r')\, r'\, dr'$$

a quick calculation shows that the resulting electric field is not dependent on charge outside of the field point:

$$\begin{aligned} \vec{\mathbf{E}}(r) &= -\frac{\partial V_e}{\partial r} \\ &= \frac{4\pi}{r^2}\int_0^r \rho(r')\, r'^2\, dr' \;-\; \frac{4\pi}{r}\rho(r)\, r^2 \;+\; 4\pi\rho(r)\, r \\ &= \frac{4\pi}{r^2}\int_0^r \rho(r')\, r'^2\, dr' \end{aligned}$$

which clearly shows that although the potential depends on charge located at $r' > r$, the electric field does not. As expected, $\vec{\mathbf{E}}(r)$ is independent of charge density outside r, and by analogy, $\vec{\mathbf{g}}(r)$ is independent of mass density outside r.

B.2 Numerical Considerations

In psuedo-numerical code notation, eq. (B.2) can be expressed as:

$$
\begin{aligned}
V_j^n &= -4\pi G m^2 \left(\frac{1}{j\,\Delta r} \sum_{i=0}^{j-1} |\psi_i^n|^2 \,(i\,\Delta r)^2 \;\Delta r + \sum_{i=j}^{N-1} |\psi_i^n|^2 \,(i\,\Delta r)\;\Delta r \right) && \text{(B.3)} \\
&= -4\pi G \,(m\,\Delta r)^2 \left(\frac{1}{j} \sum_{i=0}^{j-1} |\psi_i^n|^2 i^2 \;+\; \sum_{i=j}^{N-1} |\psi_i^n|^2 i \right) && \text{(B.4)}
\end{aligned}
$$

That is, to get the potential at a point j on the spatial grid, we sum over all ψ_i. To obtain the potential at all points on the grid, we vary j from 0 to $N-1$. However, eq. (B.4) is computationally very expensive. The program is in the potential calculating routines for the vast majority of the time, so performing this integral as efficiently as possible is very important. After much head scratching, I came up with a convoluted scheme to compute this sum with the least number of computational steps which is included at the end of this appendix.

One might worry about the $1/r$ term in the potential energy integral, eq. (B.2) (or equivalently, the $1/j$ term in eq. (B.4)), but with a little thought, the potential integral is regular and well behaved everywhere. The problem is completely analagous to the $1/r$ divergence in the expression for the Laplacian in polar coordinates. At the origin, polar coordinates become ill-defined. Numerically, the r dependence in the limits save the integral from diverging.

```
void SetGravPotential(long double V[], long double complex wf[],
                      long double dr, long double m)
{
   register long double sum1 = 0.0L, sum2 = 0.0L;

   // I wish C were static, but it can't because dr isn't really "constant".
   long double C = -4.0L * M_PIl * m * m * dr * dr;

   for (register int j=1; j<N; ++j)
   {
      // This loop affects V[N-1] but doesn't affect V[0]
      sum2 += (long double)j * (long double)j * conjl(wf[j]) * wf[j];
      V[j] = sum2 / (long double)j;
   }

   for (V[0]=0.0L, register int j=N-1; j>=1; --j)
   {
      // This loop affects V[0] but doesn't affect V[N-1]
      sum1 += (long double)j * conjl(wf[j]) * wf[j];
      V[j-1] += sum1;
      V[j-1] *= C;    // Multiply by overall constant
   }
```

```
   // Multiply last element by C, since it's not done in the previous loop.
   V[N-1] *= C;
}



void SetFreePotential(long double V[])
{
   for (register int i=0; i<N; ++i)
      V[i] = 0.0L;
}
```

Appendix C

Tridiagonal Matrices

At each timestep t the program requires solving a set of N linear equations with N unknowns. The solution to this system of equations is essentially the wavefunction at timestep $t + \Delta t$. Being a novice numerical physicist at the outset of this project, I did a search for numerical libraries to solve linear equations. This did not work out for a number of reasons:

1. I did not want a black box. Schrödinger's equation is infamously unstable with regard to numerical solutions. I wanted to know all the details of my code to understand what kind of precision to expect in the solutions.

2. Many of the high-quality routines are in Fortran anyhow.

3. The ones in C mainly used **floats**. In an attempt to gain high accuracy, my code uses **long double** exclusively. Using these matrix solvers would result in

a loss of precision.

4. My code makes use of the **complex** variable type, introduced into C in the C99 Standard and implemented in GNU's gcc compiler sometime around 2001. At the time of this writing, I know of no canned C matrix solvers that work with matrices of **complex long double** data types.

5. The system of linear equations generated by my difference equations are tridiagonal in nature. Solving the equations with a typical canned routine would be computationally extremely wasteful.

6. Many solvers written in C were written in archaic C and generated compiler warnings. Worse, they were poorly written since they trampled on the input matrices to avoid using `malloc()`. I needed a routine that did not trample on its arguments.

For these reasons, I decided to write my own matrix solver. I developed an algorithm that solves tridiagonal matrices in an efficient manner, and implemented the algorithm. I later found that this algorithm is already fairly well known as the "Thomas Algorithm".

C.1 Developing the Thomas Algorithm

For each time step t, the difference equations generate a system of N linear equations in N unknowns of the form:

$$\begin{array}{llllllll}
b_0 x_0 & + & c_0 x_1 & & & & & = d_0 \\
a_1 x_0 & + & b_1 x_1 & + & c_1 x_2 & & & = d_1 \\
 & & a_2 x_1 & + & b_2 x_2 & + & c_2 x_3 & = d_2 \\
 & & & & & \ddots & & \\
 & & & & a_{N-2} x_{N-3} & + \; b_{N-2} x_{N-2} & + \; c_{N-2} x_{N-1} & = d_{N-2} \\
 & & & & & a_{N-1} x_{N-2} & + \; b_{N-1} x_{N-1} & = d_{N-1}
\end{array}$$

This form is called "tridiagonal". The coefficients a_i (called the super-diagonal), b_i (called the diagonal), and c_i (called the sub-diagonal) are generated by the difference equations themselves. It is noteworthy that the b_i coefficients are the only ones which are time dependent, by virtue of containing sums of the time dependent potential. The coefficients d_i represent the known wavefunction at time t. The unknowns, x_i (the χ_j^n vector) essentially represent the unknown wavefunction at time $t + \Delta t$. The index i represents the location within the grid. C arrays of size N are indexed from 0 to $N - 1$, so that is the convention we will use for this discussion. The first thing we will do is eliminate the a_i.

Multiply the 0th equation by a_1/b_0 and subtract from the 1st equation. Let $B_0 = b_0$ and $D_0 = d_0$. Define $B_1 = b_1 - \frac{a_1}{B_0} c_0$, and $D_1 = d_1 - \frac{a_1}{B_0} D_0$.

$$
\begin{array}{llllllllll}
B_0x_0 & + & c_0x_1 & & & & & & & = D_0 \\
& & B_1x_1 & + & c_1x_2 & & & & & = D_1 \\
& & a_2x_1 & + & b_2x_2 & + & c_2x_3 & & & = d_2 \\
& & & & & & \ddots & & & \\
& & & & a_{N-2}x_{N-3} & + & b_{N-2}x_{N-2} & + & c_{N-2}x_{N-1} & = d_{N-2} \\
& & & & & & a_{N-1}x_{N-2} & + & b_{N-1}x_{N-1} & = d_{N-1}
\end{array}
$$

Multiply the 1st equation by a_2/B_1 and subtract from the 2nd equation. Define $B_2 = b_2 - \frac{a_2}{B_1}c_1$, and $D_2 = d_2 - \frac{a_2}{B_1}D_1$.

$$
\begin{array}{llllllllll}
B_0x_0 & + & c_0x_1 & & & & & & & = D_0 \\
& & B_1x_1 & + & c_1x_2 & & & & & = D_1 \\
& & & & B_2x_2 & + & c_2x_3 & & & = D_2 \\
& & & & & & \ddots & & & \\
& & & & a_{N-2}x_{N-3} & + & b_{N-2}x_{N-2} & + & c_{N-2}x_{N-1} & = d_{N-2} \\
& & & & & & a_{N-1}x_{N-2} & + & b_{N-1}x_{N-1} & = d_{N-1}
\end{array}
$$

Keep proceeding in this manner until all the a_i are eliminated. Once we multiply equation $N-2$ by a_{N-1}/B_{N-2} and subtract that from equation $N-1$, we get:

$$
\begin{array}{ccccccccccc}
B_0x_0 & + & c_0x_1 & & & & & & & & = D_0 \\
 & & B_1x_1 & + & c_1x_2 & & & & & & = D_1 \\
 & & & + & B_2x_2 & + & c_2x_3 & & & & = D_2 \\
 & & & & & & & \ddots & & & \\
 & & & & & & & & B_{N-2}x_{N-2} & + \quad c_{N-2}x_{N-1} & = D_{N-2} \\
 & & & & & & & & & B_{N-1}x_{N-1} & = D_{N-1}
\end{array}
$$

It is now clear how to find the x_i. Equation $N-1$ gives x_{N-1}, and using this result with equation $N-2$ gives x_{N-2}, and so on.

C.2 Summary

Taking $a_0 = c_{N-1} = 0$ as a convention, we obtain the x_i recursively:

$$x_i = \frac{D_i - c_i x_{i+1}}{B_i} \qquad \text{where} \quad i = N-1, \ldots, 0$$

where

$$B_i = b_i - \frac{a_i}{B_{i-1}} c_{i-1} \quad \text{and} \quad D_i = d_i - \frac{a_i}{B_{i-1}} D_{i-1} \qquad \text{where} \quad i = 0, \ldots, N-1$$

Instead of storing and performing computations on N^2 numbers (where N is extremely large and most of the elements are zero), this algorithm works with only $3N$ numbers.

```
/*  trisolve.c
 *
 *  Solves Q(t) X(t) = Psi(t).  Update the solution, but be careful
 *  about trampling on the data!   We solve a tridiagonal system of
 *  long double complex variables.
 */
#include "defines.h"   // Provides gridsize N
#include "trisolve.h"

void trisolve(long double complex a[N], long double complex b[N],
              long double complex c[N], long double complex d[N],
              long double complex X[N])
{
   long double complex dbar[N], bbar[N];
   register int i;

   /* Transform the matrix */
   bbar[0] = b[0];
   dbar[0] = d[0];
   for (i=1; i<N; ++i)
   {
```

```
      bbar[i] = b[i] - a[i] * c[i-1] / bbar[i-1];
      dbar[i] = d[i] - a[i] * dbar[i-1] / bbar[i-1];
   }


   /* Form the solution */
   X[N-1] = dbar[N-1] / bbar[N-1];
   for (i=N-2; i>=0; --i)
      X[i] = (dbar[i] - c[i]*X[i+1]) / bbar[i];
}
```

Appendix D

The Analytic Free Particle

D.1 Verifying the program

One of the most important questions a researcher can ask is, "Are my results accurate." Numerically, this project can be broken down into two components: solution of the free particle and solution of the particle in a potential. By parametrically "turning down the potential" and verifying that

$$\operatorname*{limit}_{V \to 0} \psi_{\mathrm{grav}}(r,t) \;=\; \psi_{\mathrm{free}}(r,t)$$

one can test that the components related to solving the free particle: the Crank-Nicholson algorithm, derivatives at the polar origin, the difference equations, basic coding, and program logic, are all correct. Once it is shown that the program computes the time evolution for the free particle correctly, the only question left is whether it computes the potential correctly, since the potential gets placed into the difference equations. Therefore, a considerable amount of effort should be put into verifying the free particle. Once this is done, a great deal of confidence can be put into the PDE solver.

The solution of the free particle is non-trivial. Although Schrödinger's free particle equation is linear, it is a time dependent partial differential equation. For the benefit of the interested reader, the solution is derived here.

D.2 Obtaining the analytic free particle solution

The solution to Schrödinger's free particle will be expanded in spherical eigenfunctions of the Laplacian operator: spherical Bessel functions and spherical harmonics. Since the particle is unbound, there are no conditions on the wave number, and the sum is an integral over all momenta.

$$\phi_{\ell,m} = \int_0^\infty C_{\ell,m}(k) j_\ell(kr) Y_\ell^m(\theta,\phi)\, dk \tag{D.1}$$

As usual for a free particle, the k eigenvalue is $\sqrt{2mE}/\hbar$ and continuous. The problem at hand is to solve Schrödinger's equation

$$-\frac{\hbar^2}{2m}\nabla^2\psi = i\hbar\frac{\partial\psi}{\partial t}$$

given the normalized initial condition

$$\psi(r,0) \;=\; \left(\frac{\alpha}{\pi}\right)^{3/4} e^{-\alpha r^2/2} \tag{D.2}$$

The free particle Hamiltonian is a special case of a spherically symmetric Hamiltonian. Since our initial condition is spherically symmetric, our wavefunction will evolve under spherical symmetry as well. Therefore, $\phi_{\ell,m} \equiv 0$ except for $\ell = m = 0$. We can thus write the general expansion eq. (D.1) of our initial condition eq. (D.2) as:

$$\left(\frac{\alpha}{\pi}\right)^{3/4} e^{-\alpha r^2/2} \;=\; \int_0^\infty C(k)\, j_0(kr)\, dk$$

Using[1] $j_0(kr) = \sqrt{\pi/2kr}J_{1/2}(kr) = \sin(kr)/kr$ to rewrite this expression in a form more convenient for using orthogonality,

$$\left(\frac{\alpha}{\pi}\right)^{3/4} e^{-\alpha r^2/2} \;=\; \sqrt{\frac{\pi}{2r}} \int_0^\infty \frac{C(k)}{\sqrt{k}} J_{1/2}(kr)\, dk$$

Multiplying through by $k'r^2 j_0(k'r) = k'r^2 \sin(k'r)/k'r = k'r^2\sqrt{\pi/2k'r}J_{1/2}(k'r)$ and integrating over r:

$$\left(\frac{\alpha}{\pi}\right)^{3/4} \int_0^\infty r\sin(k'r)\, e^{-\alpha r^2/2}\, dr \;=\; \frac{\pi}{2}\int_0^\infty C(k)\sqrt{\frac{k'}{k}} \int_0^\infty J_{1/2}(kr) J_{1/2}(k'r) r\, dr\, dk \tag{D.3}$$

Using the orthogonality relation for Bessel functions[2] $\int_0^\infty J_m(kx)J_m(k'x)xdx = \frac{1}{k}\delta(k-$

[1]See [36] eq. (16.9), pg. 740.
[2]See [36] eq. (3.112), pg. 110.

k'), we can simplify the RHS of eq. (D.3):

$$\frac{\pi}{2}\int_0^\infty C(k)\sqrt{\frac{k'}{k}}\int_0^\infty J_{1/2}(kr)J_{1/2}(k'r)r\,dr\,dk \;=\; \frac{\pi}{2}\int_0^\infty C(k)\sqrt{\frac{k'}{k}}\frac{\delta(k-k')}{k}\,dk \tag{D.4}$$

$$= \frac{\pi}{2}\frac{C(k')}{k'} \tag{D.5}$$

And the LHS of eq. (D.3) can be computed with Gradshtyn and Ryzhik, 5th ed. pg 529, eq(3.952.1) (surpress all primes now that we are done with them):

$$\left(\frac{\alpha}{\pi}\right)^{3/4}\int_0^\infty r\sin(kr)e^{-\alpha r^2/2}\,dr \;=\; \frac{k}{\sqrt{2}\sqrt[4]{\pi\alpha^3}}e^{-k^2/2\alpha} \tag{D.6}$$

Equating eq. (D.5) and eq. (D.6) and solving for the constant we get:

$$C(k) \;=\; \sqrt{2}\left(\frac{k}{\pi}\right)^2\left(\frac{\pi}{\alpha}\right)^{3/4}e^{-k^2/2\alpha}$$

So the initial condition, expressed in the basis of eigenfunctions of the free particle Hamiltonian is:

$$\psi(r,0) \;=\; \left(\frac{\alpha}{\pi}\right)^{3/4}e^{-\alpha r^2/2} \;=\; \frac{\sqrt{2}}{\pi^2}\left(\frac{\pi}{\alpha}\right)^{3/4}\int_0^\infty k^2e^{-k^2/2\alpha}j_0(kr)\,dk$$

$$= \frac{\sqrt{2}}{\pi^2 r}\left(\frac{\pi}{\alpha}\right)^{3/4}\int_0^\infty k\sin(kr)e^{-k^2/2\alpha}\,dk \tag{D.7}$$

Applying formal time evolution to eq. (D.7),

$$\psi(r,t) \;=\; \frac{\sqrt{2}}{\pi^2 r}\left(\frac{\pi}{\alpha}\right)^{3/4}\int_0^\infty k\sin(kr)e^{-k^2/2\alpha}\,e^{-iE_Kt/\hbar}\,dk$$

and since the energy eigenvalues are given by $E_k = (\hbar k)^2/2m$,

$$\psi(r,t) \;=\; \frac{\sqrt{2}}{\pi^2 r}\left(\frac{\pi}{\alpha}\right)^{3/4}\int_0^\infty k\sin(kr)e^{-\frac{k^2}{2}\left[\frac{1}{\alpha}+\frac{i\hbar t}{m}\right]}\,dk$$

Expanding the $\sin(kr)$ term, defining $\beta = (m + i\alpha\hbar t)/2m\alpha$, and changing variable from k to $-k$,

$$\psi(r,t) \;=\; \frac{\sqrt{2}}{\pi^2 r}\left(\frac{\pi}{\alpha}\right)^{3/4}\int_{-\infty}^\infty \frac{k}{2i}e^{-\beta k^2+ikr}\,dk$$

Completing the square in the exponential,

$$\psi(r,t) \;=\; \frac{\sqrt{2}}{\pi^2 r}\left(\frac{\pi}{\alpha}\right)^{3/4}\frac{e^{-r^2/4\beta}}{2i}\int_{-\infty}^\infty ke^{-\beta(k-\frac{ir}{2\beta})^2}\,dk$$

And finally, changing variable from k to $u = k - ir/2\beta$,

$$\psi(r,t) \;=\; \frac{\sqrt{2}}{\pi^2 r}\left(\frac{\pi}{\alpha}\right)^{3/4}\frac{e^{-r^2/4\beta}}{2i}\int_{-\infty}^\infty \left(u+\frac{ir}{2\beta}\right)e^{-\beta u^2}\,du$$

The first term of the integrand is odd over an interval symmetric about the origin, so it's zero. Performing the trivial second integral and simplifying the result,

$$\psi(r,t) \;=\; \left(\frac{\alpha}{\pi}\right)^{3/4}\left(\frac{m}{m+i\alpha\hbar t}\right)^{3/2} e^{-\frac{\alpha}{2}r^2\left(\frac{m}{m+i\alpha\hbar t}\right)} \tag{D.8}$$

This is the exact solution for the time dependent Schrödinger equation for a free particle given the initial condition eq. (D.2). Upon inspection, eq. (D.8) reduces to the initial condition. A direct demonstration that eq. (D.8) satisfies the time dependent Schrödinger equation and has a normalization of unity is much longer than its value here justifies,

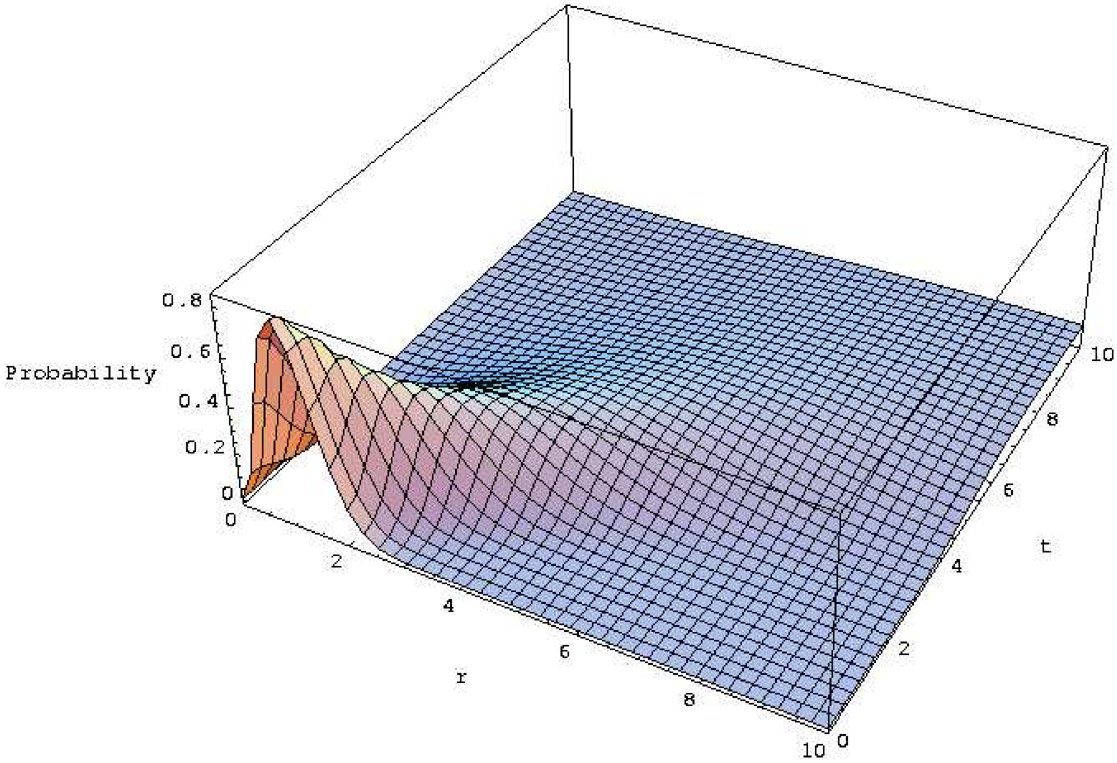


Figure D.1: Plot of free particle probability versus time for $\hbar = m = \alpha = 1$.

Bibliography

[1] L. Diósi. Gravitation and Quantum-Mechanical Localization of Macro-Objects. *Physics Letters*, 105A(4, 5):199–202, 1984.

[2] H. David Bernstein, Eldar Giladi, and Kingsley R. W Jones. Eigenstates of the Gravitational Schrödinger Equation. *Modern Physics Letters A*, 13(29):2327–2336, 1998.

[3] Irene M Moroz, Roger Penrose, and Paul Tod. Spherically-Symmetric Solutions of the Schrödinger-Newton Equations. *Class. Quantum Grav.*, 15:2733–2742, 1998.

[4] Deepak Kumar and Vikram Soni. Single Particle Schrödinger Equation With Gravitational Self-Interaction. *Physics Letters A*, 271(3):157–166, 2000.

[5] C. Møller. The Energy-Momentum Complex in General Relativity and Related Problems. In M. A. Professor Lichnerowicz and M. A. Professor Tonnelat, editors, *Les Thories Relativistes de la Gravitation*, Colloques Internationaux du Centre

National de la Recherche Scientifique, pages 2–25. Editions du Centre National de la Recherche Scientifique, 1962.

[6] L. Rosenfeld. On Quantization of Fields. *Nuclear Physics*, 40:353–356, 1963.

[7] T. W. B. Kibble and S. Randjbar-Daemi. Non-Linear Coupling of Quantum Theory and Classical Gravity. *Journal of Physics A (Mathematical and General)*, 13:141–148, 1980.

[8] T. W. Kibble. Is a Semi-Classical Theory of Gravity Viable? In C. J. Isham, R. Penrose, and D. W. Sciama, editors, *Quantum Gravity 2: A Second Oxford Symposium*, pages 63–80. Clarendon Press, Oxford, 1981.

[9] M. J. Duff. Inconsistency of Quantum Field Theory in Curved Space-Time. In C. J. Isham, R. Penrose, and D. W. Sciama, editors, *Quantum Gravity 2: A Second Oxford Symposium*, pages 81–105. Clarendon Press, Oxford, 1981.

[10] Nicolas Gisin. Stochastic Quantum Dynamics and Relativity. *Helvetica Physica Acta*, 62:363–371, 1989.

[11] Joseph Polchinski. Weinberg's Nonlinear Quantum Mechanics and the Einstein-Podolsky-Rosen Paradox. *Physical Review Letters*, 66(4):397–400, 1991.

[12] Marek Czachor and Heinz-Dietrich Doebner. Correlation Experiments in Nonlinear Quantum Mechanics. *Physical Review A*, 301:139–152, 2002. arXiv: quant-ph/0110008.

[13] Marek Czachor. Nambu-type Generalization of the Dirac Equation. *Physical Review A*, 225:1–12, 1997. arXiv: quant-ph/9501008.

[14] Marek Czachor. Nonlinear Schrödinger Equation and Two-Level Atoms. *Physical Review A*, 53:1310–1315, 1996. arXiv: quant-ph/9501007.

[15] Gerald A. Goldin. Diffeomorphism Group Representations and Quantum Nonlinearity: Gauge Transformations and Measurement. In H. D. Doebner, V. K. Dobrev, and P. Nattermann, editors, *Nonlinear, Deformed and Irreversible Quantum Systems*, pages 125–139. World Scientific Publishing, 1995.

[16] Thomas F. Jordan. Reconstructing a Nonlinear Dynamical Framework for Testing Quantum Mechanics. *Annals Of Physics*, 225:83–113, 1993.

[17] Don N. Page and C. D. Geilker. Indirect Evidence for Quantum Gravity. *Physical Review Letters*, 47(14):979–982, 1981.

[18] Kenneth Eppley and Eric Hannah. The Necessity of Quantizing the Gravitational Field. *Foundations Of Physics*, 7(1/2):51–68, 1977.

[19] Niels Bohr and L. Rosenfeld. Zur Frage der Messbarkeit der elektromagnetischen Feldgrössen. *Matematisk-Fysiske Meddelelser (Kongelige Danske Videnskabernes Selskab)*, 12(8):479–534, 1933.

[20] Roger Penrose. Gravity and Quantum Mechanics. In R. J. Gleiser, C. N. Kozameh, and O. M. Moreschi, editors, *Genral Relativity and Gravitation 1992*, pages 179–189. Institute of Physics Publishing, 1993.

[21] Roger Penrose. Gravity and State Vector Reduction. In Roger Penrose and C. J. Isham, editors, *Quantum Concepts in Space and Time*, pages 129–145. Oxford Scientific Publications, 1986.

[22] Robert Clifton and Bradley Monton. Losing Your Marbles in Wavefunction Collapse Theories. *arXiv: quant-ph/9905065*, 1999.

[23] G. C Ghirardi, A. Rimini, and T. Weber. A Model for a Unified Quantum Description of Macroscopic and Microscropic Systems. In L. Accardi and W. V. Waldenfels, editors, *Quantum Probability and Applications II*. Springer-Verlag, 1985.

[24] G. C Ghirardi, Rimini A., and T. Weber. Unified Dynamics for Microscopic and Macroscopic Systems. *Physical Review D*, 34(2):470–491, 1986.

[25] Roderich Tumulka. A Relativistic Version of the Ghirardi-Rimini-Weber Model. *arXiv: quant-ph/0406094*, 2004.

[26] Giancarlo Ghirardi. Collapse Theories. On the web at the URL: `http://plato.stanford.edu/archives/spr2002/entries/qm-collapse/`, Spring 2002.

[27] Abner Shimony. Desiderata for a Modified Quantum Dynamics. In Arthur Fine, Micky Forbes, and Linda Wessels, editors, *Proceedings of the 1990 Biennial Meeting of the Philosophy of Science Association, volume two*. University of Chicago Press, 1991.

[28] Peter J. Lewis. Quantum Mechanics, Orthogonality, and Counting. *Brit. J. Sci*, 48:313–328, 1997.

[29] GianCarlo Ghirardi and Angelo Bassi. Do Dynamical Reduction Models Imply that Arithmetic Does Not Apply to Ordinary Quantum Mechanics? *British Journal for the Philosophy of Science*, 50:49–64, 1999.

[30] L. F. Santos and C. O. Escobar. A Proposed Solution to the Tail Problem of Dynamical Reduction Models. *Physics Letters A*, 278:315–318, 2001.

[31] David Deutsch. Quantum Theory of Probability and Decisions. *Proceedings of the Royal Society of London, Series A*, 455:3129–3137, 1999.

[32] H. Barnum, C. M. Caves, J. Finkelstein, C. A. Fuchs, and R. Schack. Quantum Probability from Decision Theory? *Proceedings of the Royal Society of London, Series A*, 456:1175–1182, 2000.

[33] Maximilian Schlosshauer. Decoherence, the Measurement Problem, and Interpretations of Quantum Mechanics. *Reviews of Modern Physics*, 76(4):1267–1305, October 2004.

[34] F. Károlyházy, A. Frenkel, and B. Lukács. On the Possible Role of Gravity in the Reduction of the Wave Function. In Roger Penrose and C. J. Isham, editors, *Quantum Concepts in Space and Time*, pages 109–128. Oxford Scientific Publications, 1986.

[35] Charles W. Misner, Kip S. Thorne, and John Archibald Wheeler. *Gravitation*. Freeman and Company, 1973.

[36] John David Jackson. *Classical Electrodynamics*. Wiley & Sons, 2nd edition, 1975.

[37] C. Davisson and L. H. Germer. Diffraction of Electrons by a Crystal of Nickel. *Physical Review*, 30(6):705–741, 1927.

[38] E. O. Wollan and C. G. Shull. The Diffraction of Neutrons by Crystalline Powders. *Physical Review*, 73(8):830–841, 1948.

[39] Olaf Nairz, Markus Arndt, and Anton Zeilinger. Experimental Challenges in Fullerene Interferometry. *Journal Of Modern Optics*, 47(14/15):2811–2821, 2000.

[40] O. Carnal and J. Mlynek. Young's Double-Slit Experiment with Atoms: A Simple Atom Interferometer. *Physical Review Letters*, 66(21):2698–2692, 1991.

[41] David W. Keith, Christopher R. Ekstrom, Quentin A. Turchette, and David E. Pritchard. An Interferometer for Atoms. *Physical Review Letters*, 66(21):2693–2696, 1991.

[42] John F. Clauser and Shifang Li. Talbot-vonLau Atom Interferometry with Cold Slow Potassium. *Physical Review A*, 49(4):2213–2216, 1994.

[43] W. Schöllkopf and J. P. Toennies. Nondestructive Mass Selection of Small van der Waals Clusters. *Science*, 266:1345–1349, 1994.

[44] Björn Brezger, Lucia Hackermüller, Stefan Uttenthaler, Julia Petschinka, Markus Arndt, and Anton Zeilinger. Matter-Wave Interferometer for Large Molecules. *Physical Review Letters*, 88(10):100404, 2002.

[45] Markus Arndt and Anton Zeilinger. Probing the Limits of the Quantum World. *Physics World*, March 2005.

[46] John F. Clauser. De Broglie Wave Interference of Small Rocks and Live Viruses. In Robert S. Cohen, Michael Horne, and John Stachel, editors, *Experimental Metaphysics*, pages 1–11. Kluwer Academic Publishers, 1997.

[47] Björn Brezger, Markus Arndt, and Anton Zeilinger. Concepts for Near-Field Interferometers with Large Molecules. *Journal of Optics B: Quantum and Semiclassical Optics*, 5:S82–S89, April 2003.

[48] Markus Arndt, Olaf Nairz, and Anton Zeilinger. Interferometry with Macromolecules: Quantum Paradigms Tested in the Macroscopic World. In Reinhold A. Bertlemann and Anton Zeilinger, editors, *Quantum [Un]speakables: From Bell to Quantum Information*. Springer, 2002.

[49] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1995.

[50] Björn Brezger, Markus Arndt, and Anton Zeilinger. Wave Nature of Biomolecules and Fluorofullerenes. *Physical Review Letters*, 91(9):090408, August 2003.

[51] Markus Arndt, Olaf Nairz, Julian Vos-Andreae, Claudia Keller, Gerbrand van der Zouw, and Anton Zeilinger. Wave-particle Duality of C_{60} Molecules. *Nature*, 401:680–682, October 1999.

[52] B. Ilic, Y. Yang, K. Aubin, R. Reichenbach, S. Krylov, and H. G. Craighead. Enumeration of DNA Molecules Bound to a Nanomechanical Oscillator. *Nano Letters*, 5(5):925–929, 2005.