A numerical study of the Schrödinger-Newton equations



Richard Harrison
St.Peter's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2001

Acknowledgements

I would like to thank my supervisors Paul Tod and Irene Moroz, Roger Penrose for the idea behind my thesis, Nick Trefethen for his help in use of spectral methods and his excellent lectures, Ian Sobey for his helpful comments and Lionel Mason for getting me started with this project. I would also like to thank the EPSRC for their financial support. Last but not least I would like thank my mother, father and sister for their support.

Abstract

The Schrödinger-Newton (S-N) equations were proposed by Penrose [18] as a model for gravitational collapse of the wave-function. The potential in the Schrödinger equation is the gravity due to the density of $|\psi|^2$, where ψ is the wave-function. As with normal Quantum Mechanics the probability, momentum and angular momentum are conserved. We first consider the spherically symmetric case, here the stationary solutions have been found numerically by Moroz et al [15] and Jones et al [3]. The ground state which has the lowest energy has no zeros. The higher states are such that the $(n+1)$th state has n zeros. We consider the linear stability problem for the stationary states, which we numerically solve using spectral methods. The ground state is linearly stable since it has only imaginary eigenvalues. The higher states are linearly unstable having imaginary eigenvalues except for n quadruples of complex eigenvalues for the $(n+1)$th state, where a quadruple consists of $\{\lambda, \bar{\lambda}, -\lambda, -\bar{\lambda}\}$. Next we consider the nonlinear evolution, using a method involving an iteration to calculate the potential at the next time step and Crank-Nicolson to evolve the Schrödinger equation. To absorb scatter we use a sponge factor which reduces the reflection back from the outer boundary condition and we show that the numerical evolution converges for different mesh sizes and time steps. Evolution of the ground state shows it is stable and added perturbations oscillate at frequencies determined by the linear perturbation theory. The higher states are shown to be unstable, emitting scatter and leaving a rescaled ground state. The rate at which they decay is controlled by the complex eigenvalues of the linear perturbation. Next we consider adding another dimension in two different ways: by considering the axisymmetric case and the 2-D equations. The stationary solutions are found. We modify the evolution method and find that the higher states are unstable. In 2-D case we consider rigidly rotationing solutions and show they exist and are unstable.

Contents

1 Introduction: the Schrödinger-Newton equations **1**

1.1 The equations motivated by quantum state-reduction 1

1.2 Some analytical properties of the S-N equations 4

1.2.1 Existence and uniqueness 4

1.2.2 Rescaling 4

1.2.3 Lagrangian form 4

1.2.4 Conserved quantities 5

1.2.5 Lie point symmetries 7

1.2.6 Dispersion 7

1.3 Analytic results about the time-independent case 8

1.3.1 Existence and uniqueness 8

1.3.2 Variational formulation 8

1.3.3 Negativity of the energy-eigenvalue 9

1.4 Plan of thesis 10

1.5 Conclusion 10

2 Spherically-symmetric stationary solutions **12**

2.1 The equations 12

2.2 Computing the spherically-symmetric stationary states 13

2.2.1 Runge-Kutta integration 13

2.2.2 Alternative method 14

2.3 Approximations to the energy of the bound states 15

3 Linear stability of the spherically-symmetric solutions **18**

3.1 Linearising the S-N equations 18

3.2 Separating the $O(\epsilon)$ equations 20

3.3 Boundary Conditions 22

3.4 Restriction on the possible eigenvalues 24

3.5 An inequality on $\mathrm{Re}(\lambda)$ 25

4 Numerical solution of the perturbation equations **26**
4.1 The method . . . 26
4.2 The perturbation about the ground state . . . 28
4.3 Perturbation about the second state . . . 33
4.4 Perturbation about the higher order states . . . 36
4.5 Bound on real part of the eigenvalues . . . 39
4.6 Testing the numerical method by using Runge-Kutta integration . . . 39
4.7 Conclusion . . . 41

5 Numerical methods for the evolution **46**
5.1 The problem . . . 46
5.2 Numerical methods for the Schrödinger equation with arbitrary time independent potential . . . 47
5.3 Conditions on the time and space steps . . . 48
5.4 Boundary conditions and sponges . . . 49
5.5 Solution of the Schrödinger equation with zero potential . . . 50
5.6 Schrödinger equation with a trial fixed potential . . . 51
5.7 Numerical evolution of the S-N equations . . . 52
5.8 Checks on the evolution of S-N equations . . . 53
5.9 Mesh dependence of the methods . . . 54
5.10 Large time behaviour of solutions . . . 54

6 Results from the numerical evolution **56**
6.1 Testing the sponges . . . 56
6.2 Evolution of the ground state . . . 57
6.3 Evolution of the higher states . . . 62
6.3.1 Evolution of the second state . . . 62
6.3.2 Evolution of the third state . . . 67
6.4 Evolution of an arbitrary spherically symmetric Gaussian shell . . . 69
6.4.1 Evolution of the shell while changing v . . . 72
6.4.2 Evolution of the shell while changing a . . . 74
6.4.3 Evolution of the shell while changing σ . . . 77
6.5 Conclusion . . . 79

7 The axisymmetric S-N equations **81**
7.1 The problem . . . 81
7.2 The axisymmetric equations . . . 81
7.3 Finding axisymmetric stationary solutions . . . 82
7.4 Axisymmetric stationary solutions . . . 83
7.5 Time-dependent solutions . . . 84

8 The Two-Dimensional S-N equations **97**
8.1 The problem . . . 97
8.2 The equations . . . 97
8.3 Sponge factors on a square grid . . . 98
8.4 Evolution of dipole-like state . . . 99
8.5 Spinning Solution of the two-dimensional equations . . . 102
8.6 Conclusion . . . 103

A Fortran programs **108**
A.1 Program to calculate the bound states for the S-N equations . . . 108

B Matlab programs **116**
B.1 Chapter 2 Programs . . . 116
B.1.1 Program for asymptotically extending the data of the bound state calculated by Runge-Kutta integration . . . 116
B.1.2 Programs to calculate stationary state by Jones et al method . . . 116
B.2 Chapter 4 Programs . . . 119
B.2.1 Program for solving the Schrödinger equation . . . 119
B.2.2 Program for solving the $O(\epsilon)$ perturbation problem . . . 121
B.2.3 Program for the calculation of (4.10) . . . 122
B.2.4 Program for performing the Runge-Kutta integration on the $O(\epsilon)$ perturbation problem . . . 123
B.3 Chapter 6 Programs . . . 123
B.3.1 Evolution program . . . 123
B.3.2 Fourier transformation program . . . 126
B.4 Chapter 7 Programs . . . 126
B.4.1 Programs for solving the axisymmetric stationary state problem . . . 126
B.4.2 Evolution programs for the axisymmetric case . . . 133
B.5 Chapter 8 Programs . . . 138
B.5.1 Evolution programs for the two dimensional S-N equations . . . 138

Bibliography **143**

List of Figures

2.1 First four eigenfunctions 15
2.2 The least squares fit of the energy eigenvalues 16
2.3 The number of nodes with the Moroz et al [15] prediction of the number of nodes 17
2.4 Log-log plot of the energy values with $\log(n)$ 17

4.1 The smallest eigenvalues of the perturbation about the ground state 29
4.2 All the computed eigenvalues of the perturbation about the ground state . . 29
4.3 The first eigenvector of the perturbation about the ground state. Note the scales 30
4.4 The second eigenvector of the perturbation about the ground state 31
4.5 The third eigenvector of the perturbation about the ground state 31
4.6 The change in the sample eigenvalue with increasing values of N ($L = 150$) 32
4.7 The change in the sample eigenvalue with increasing values of L ($N = 60$) . 32
4.8 The lowest eigenvalues of the perturbation about the second bound state . . 34
4.9 All the computed eigenvalues of the perturbation about the second bound state 34
4.10 The first eigenvector of the perturbation about the second bound state, Note the scales 35
4.11 The second eigenvector of the perturbation about the second bound state . 35
4.12 The third eigenvector of the perturbation about the second bound state . . 36
4.13 The eigenvalues of the perturbation about the third bound state 37
4.14 The second eigenvector of the perturbation about the third bound state . . 37
4.15 The third eigenvector of the perturbation about the third bound state . . . 38
4.16 The third eigenvector of the perturbation about the third bound state . . . 38
4.17 The eigenvalues of the perturbation about the fourth bound state 40
4.18 The first eigenvector of the perturbation about the ground state, using Runge-Kutta integration; compare with figure 4.3 41
4.19 The second eigenvector of the perturbation about the ground state, using Runge-Kutta integration; compare with figure 4.4 42

4.20 The first eigenvector of the perturbation about the second bound state, using Runge-Kutta integration; compare with figure 4.10 42
4.21 The second eigenvector of the perturbation about the second bound state, using Runge-Kutta integration; compare with figure 4.11 43
4.22 The second eigenvector of the perturbation about the third bound state, using Runge-Kutta integration; compare with figure 4.14 43
4.23 The real part of third eigenvector of the perturbation about the third bound state, using Runge-Kutta integration; compare figure 4.15 44
4.24 The imaginary part of third eigenvector of the perturbation about the third bound state, using Runge-Kutta integration; compare with figure 4.16 . . . 44

6.1 Testing the sponge factor with wave moving off the grid 56
6.2 Testing sponge factor with wave moving off the the grid 57
6.3 The graph of the phase angle of the ground state as it evolves 58
6.4 The oscillation about the ground state 59
6.5 Evolution of the ground state 59
6.6 The eigenvalue associated with the linear perturbation about the ground state 60
6.7 The oscillation about the ground state at a given point 61
6.8 The Fourier transform of the oscillation about the ground state 61
6.9 The graph of the phase angle of the second state as it evolves 63
6.10 The long time evolution of the second state in Chebyshev methods 63
6.11 Evolution of second state tolerance E-9 approximately 65
6.12 $f(t) - A$ 65
6.13 Fourier transformation of $f(t) - A$ 66
6.14 Decay of the second state 66
6.15 The short time evolution of the second bound state 67
6.16 Oscillation about second state at a fixed radius 68
6.17 Fourier transform of the oscillation about the second state 68
6.18 Evolution of third state 70
6.19 $f(t) - A$ with respect to the third state 70
6.20 Fourier transform of growing mode about third state 71
6.21 Probability remaining in the evolution of the third state 71
6.22 Progress of evolution with different velocities and times 72
6.23 Difference of the other time steps compared with $dt = 1$ 73
6.24 Richardson fraction done with different h_i's 73
6.25 Difference in N value 74

6.26 Evolution of lump at different a 75
6.27 Comparing differences with different time steps 75
6.28 Richardson fraction with different h_i's 76
6.29 Difference in N value 76
6.30 Evolution of lump at different σ 77
6.31 Comparing differences with different time steps 78
6.32 Richardson fractions 78
6.33 Difference in N value 79

7.1 Ground state, axp1 85
7.2 Contour plot of the ground state, axp1 85
7.3 "Dipole" state next state after ground state , axp2 86
7.4 Contour Plot of the dipole, axp2 86
7.5 Second spherically symmetric state, axp3 87
7.6 Contour Plot of the second spherically symmetric state, axp3 87
7.7 Not quite the 2nd spherically symmetric state ,axp4 88
7.8 Contour plot of the not quite 2nd spherically symmetric state, axp4 88
7.9 Not quite the 3rd spherically symmetric state $E = -0.0162$, axp5 89
7.10 Contour plot of Not quite the 3rd spherically symmetric state $E = -0.0162$, axp5 89
7.11 3rd spherically symmetric sate ,axp6 90
7.12 Contour plot of 3rd spherically symmetric state ,axp6 90
7.13 Axially symmetric state with $E = -0.0263$, axp7 91
7.14 Contour plot of axp7 - double dipole 91
7.15 State with $E = -0.0208$ and $J = 3.1178$,axp8 92
7.16 Contour plot of the state with $E = -0.0208$ and $J = 3.1178$,axp8 92
7.17 Evolution of the dipole 94
7.18 Average Richardson fraction with $dt = 0.25, 0.5, 1$ 95
7.19 Average Richardson fraction with $dt = 0.125, 0.25, 0.5$ 95
7.20 Evolution of the dipole with different mesh size 96

8.1 Sponge factor 99
8.2 Lump moving off the grid with $v = 2$, $N = 30$ and $dt = 2$ 100
8.3 Lump moving off the grid with $v = 2$, $N = 30$ and $dt = 2$ 100
8.4 Lump moving off the grid with $v = 2$, $N = 30$ and $dt = 2$ 101
8.5 Stationary state for the 2-dim S-N equations 101
8.6 Evolution of a stationary state 102

8.7 Real part of spinning solution 104
8.8 Imaginary part of spinning solution 104
8.9 Evolution of spinning solution 105
8.10 Evolution of spinning solution 105

Chapter 1

Introduction: the Schrödinger-Newton equations

1.1 The equations motivated by quantum state-reduction

The Schrödinger-Newton equations (abbreviated as S-N equations) are :

$$\mathrm{i}\hbar\frac{\partial\Psi}{\partial\tilde{t}} = \frac{-\hbar^2}{2m}\nabla^2\Psi + m\Phi\Psi, \tag{1.1a}$$

$$\nabla^2\Phi = 4\pi G m|\Psi|^2, \tag{1.1b}$$

where $\hbar$ is Planck's constant, m is the mass of the particle, Ψ is the wave function, Φ is the potential, G is the gravitational constant and t is time.

To get the time-independent form of the S-N equations (1.1) we consider the substitution

$$\Psi(x,y,z,t) = \Psi(x,y,z)\mathrm{e}^{-\mathrm{i}\frac{\tilde{E}}{\hbar}\tilde{t}}, \tag{1.2a}$$

$$\Phi(x,y,z,t) = \Phi(x,y,z), \tag{1.2b}$$

which gives the time-independent S-N equations,

$$-\frac{\hbar^2}{2m}\nabla^2\Psi + m\Phi\Psi = \tilde{E}\Psi, \tag{1.3a}$$

$$\nabla^2\Phi = 4\pi G m|\Psi|^2. \tag{1.3b}$$

The boundary conditions for the equations are the same as those for the usual Schrödinger equation and the Poisson equation, i.e we require the wave function to be smooth for all $x \in \mathbb{R}^3$ and normalisable, we also require the potential function to be smooth and zero at large distances.

One idea for a theory of quantum state reduction was put forward by Penrose [17], [18]. The idea was that a superposition of two or more quantum states, which have a significant amount of mass displacement between the states, ought to be unstable and reduce to one of the states within a finite time. This argument is motivated by a conflict between the basic

principles of quantum mechanics and those of general relativity. This idea requires that there be a special set of quantum states which collapse no further. This would be called a preferred basis. This preferred basis would be the stationary quantum states. Penrose [18] states that "the phenomenon of quantum state reduction is a gravitational phenomenon, and that essential changes are needed in the framework of quantum mechanics in order that its principles can be adequately married with the principles of Einstein's general relativity." According to Penrose, superpositions of these states ought to decay within a certain characteristic average time of T_G, where $T_G = \frac{\hbar}{E_G}$ and E_G is the gravitational self-energy of the difference bewteen the mass distributions of the two superposed states. That is, the self-energy is $\frac{1}{2}\int_{\mathbb{R}^3} |\nabla\psi|^2 dV$, where $\psi = \psi_1 - \psi_2$ the difference of the two superposed states mass distributions. [The S-N equations are a first approximation we can consider the case where gravity is taken to be Newtonian and space-time is non-relativistic.]

These equations were first considered by Ruffini and Bonazzola [21] in connection with the theory of self-gravitating boson-stars. There the S-N equations are the non-relativistic limit of the governing Klein-Gordon equations. Boson-stars consist of a large collection of bosons under the gravitational force of their own combined mass. Ruffini and Bonazzola [21] considered the problem of finding stationary boson-stars in the non-relativistic spherically symmetric case which corresponds to finding stationary spherically symmetric solutions of the S-N equations. These equations have also been considered by Moroz, Penrose and Tod [15] where they computed stationary solutions in the case of spherical symmetry. Moroz and Tod [14] prove some analytic properties of the S-N equations. They have also been considered by Bernstein, Giladi and Jones [3] who developed a better way than a shooting method for calculating the stationary solutions in the case of spherical symmetry. Bernstein and Jones [4] have started considering a method for the dynamical evolution.

Our object in this thesis is to study the S-N equations in the time dependent and time independent cases.

We can consider the non-dimensionalized S-N equations, via a transformation $(\Psi, \Phi, \tilde{t}, R) \rightarrow (\psi, \phi, t, r)$ where:

$$\begin{aligned} \psi &= \alpha\Psi, \\ \phi &= \beta\Phi, \\ t &= \gamma\tilde{t}, \\ r &= \delta R, \end{aligned} \tag{1.4}$$

such that the S-N equation become :

$$\mathrm{i}\frac{\partial\psi}{\partial t} = -\nabla^2\psi + \phi\psi, \tag{1.5a}$$
$$\nabla^2\phi = |\psi|^2, \tag{1.5b}$$

Normalisation is preserved, i.e $\int |\Psi|^2 d^3x = 1$ and $\int |\psi|^2 d^3X = 1$ if $\alpha^2 = \delta^{-3}$. The gravitation equation (1.5b) then becomes,

$$\frac{\beta}{\delta^2}\nabla^2\Phi = \alpha^2|\Psi|^2, \tag{1.6}$$

from this we deduce that $\dfrac{\alpha^2\delta^2}{\beta} = 4\pi Gm$, thus $\beta = \dfrac{1}{4\pi Gm\delta}$. The Schrödinger equation becomes,

$$\frac{\mathrm{i}\alpha}{\gamma}\Psi_{\tilde{t}} = -\frac{\alpha}{\delta^2}\nabla^2\Psi + \alpha\beta\Psi\Phi, \tag{1.7}$$

which becomes,

$$\frac{\mathrm{i}m}{\gamma\beta}\Psi_{\tilde{t}} = -\frac{m}{\delta^2\beta}\nabla^2\Psi + m\Psi\Phi, \tag{1.8}$$

so we then deduce that $\dfrac{m}{\gamma\beta} = \hbar$ and $\dfrac{m}{\delta^2\beta} = \dfrac{\hbar^2}{2m}$ so,

$$\frac{4\pi Gm^2}{\delta} = \frac{\hbar^2}{2m}, \tag{1.9}$$

and

$$\delta = \frac{8\pi Gm^3}{\hbar^2}. \tag{1.10}$$

For γ we have

$$\gamma = \frac{32\pi^2G^2m^5}{\hbar^3}. \tag{1.11}$$

Now from (1.5) the non-dimensionalized time independent S-N equations are

$$E\psi = -\nabla^2\psi + \phi\psi, \tag{1.12a}$$
$$\nabla^2\phi = |\psi|^2, \tag{1.12b}$$

where $\tilde{E} = E\dfrac{2G^2m^5}{\hbar^2}$. We can eliminate the E term from (1.12) by letting

$$\psi = S, \tag{1.13a}$$
$$E - \phi = V, \tag{1.13b}$$

which reduces (1.12) to

$$\nabla^2 S = -SV, \tag{1.14a}$$
$$\nabla^2 V = -|S|^2, \tag{1.14b}$$

which was the form of the equations consider in [14], [15].

1.2 Some analytical properties of the S-N equations

1.2.1 Existence and uniqueness

Existence and uniqueness for solutions is established by the following simplified version of the theorem of Illner et al [13].

Given $\chi(x) \in H^2((R)^3)$ with L_2-norm equal to 1, the system (1.5) has a unique, strong solution $\psi(x,t)$ global in time with $\psi(x,0) = \chi(x)$ and $\int |\psi|^2 = 1$. Illner et al [13] give regularity properties for ψ and ϕ.

1.2.2 Rescaling

Note that if (ψ, ϕ, x, t) is a solution of the S-N equations then
$(\lambda^2\psi, \lambda^2\phi, \lambda^{-1}x, \lambda^{-2}t)$ is also a solution where λ is any constant. Suppose that

$$\int_V |\psi|^2 dV = \lambda^{-1}, \tag{1.15}$$

where λ is a real constant, and consider

$$\hat{\psi} = \lambda^2\psi(\lambda x, \lambda^2 t), \tag{1.16a}$$

$$\hat{\phi} = \lambda^2\phi(\lambda x, \lambda^2 t). \tag{1.16b}$$

Then $\hat{\psi}$ and $\hat{\phi}$ satisfy the S-N equations with

$$\int_V |\hat{\psi}|^2 d\hat{V} = 1. \tag{1.17}$$

We also note that the rescaling of the solutions will also cause the rescaling of the energy eigenvalues (as well as the action). that is:

$$E_{new} = \lambda E. \tag{1.18}$$

1.2.3 Lagrangian form

Note that (1.5) can be obtained from the Lagrangian:

$$\int\int [\nabla\psi \cdot \nabla\bar{\psi} + \frac{1}{2}\phi|\psi|^2 + \frac{\mathrm{i}}{2}(\psi\bar{\psi}_t - \bar{\psi}\psi_t)]d^3x dt, \tag{1.19}$$

where it is understood that ϕ is the solution of $\nabla^2\phi = |\psi|^2$. Alternatively, one may solve the Poisson equation with the appropriate Green's function and consider the Lagrangian.

$$\int\int [\nabla\psi \cdot \nabla\bar{\psi} - \frac{1}{8\pi}\int \frac{|\psi(x)|^2|\psi(y)|^2}{|x-y|}d^3y + \frac{\mathrm{i}}{2}(\psi\bar{\psi}_t - \bar{\psi}\psi_t)]d^3x dt. \tag{1.20}$$

See Christian [5] and Diosi [7] for details.

1.2.4 Conserved quantities

The system (1.5) admits several conserved quantities, all of which are to be expected from linear quantum mechanics. Define

$$\rho = |\psi|^2, \tag{1.21a}$$

$$J_i = -\mathrm{i}(\bar{\psi}\psi_i - \psi\bar{\psi}_i), \tag{1.21b}$$

$$S_{ij} = -\psi\bar{\psi}_{ij} - \bar{\psi}\psi_{ij} + \bar{\psi}_i\psi_j + \psi_i\bar{\psi}_j + 2\phi_i\phi_j - \delta_{ij}\phi_k\phi_k, \tag{1.21c}$$

where $\psi_i = \dfrac{\partial\psi}{\partial x_i}$ and $\phi_i = \dfrac{\partial\phi}{\partial x_i}$ then

$$\dot{\rho} = -J_{i,i}, \tag{1.22a}$$

$$\dot{J}_i = -S_{ij,j}, \tag{1.22b}$$

where the dot denotes differentiation with respect to time. Therefore

$$P = \int \rho d^3x = \text{constant}, \tag{1.23}$$

that is, the total probability is conserved. Next define the total momentum P_i by

$$P_i = \int J_i d^3x. \tag{1.24}$$

Then

$$\dot{P}_i = 0. \tag{1.25}$$

Define the centre of mass $< x_i >$ by

$$< x_i >= \int \rho x_i d^3x, \tag{1.26}$$

then

$$< \dot{x}_i > = P_i, \tag{1.27}$$

so that, as expected, the centre-of-mass follows a straight line. The total angular momentum is

$$L_i = \int \epsilon_{ijk} x_j J_k d^3x, \tag{1.28}$$

and then, by the symmetry of S_{ij}, it follows that

$$\dot{L}_i = 0. \tag{1.29}$$

We define the kinetic energy T and potential energy V in the obvious way by

$$T = \int |\nabla\psi|^2 d^3x, \tag{1.30a}$$

$$V = \int \phi|\psi|^2 d^3x. \tag{1.30b}$$

Note from (1.5b) that

$$\int \phi\dot{\rho} d^3x = \int \dot{\phi}\rho d^3x. \tag{1.31}$$

It follows from this that the quantity

$$\mathcal{E} = T + \frac{1}{2}V, \tag{1.32}$$

is conserved. We shall sometimes call this the conserved energy or the action. It is closely related to the action for the time-independent equations. The total energy $E = T + V$ is consequently not conserved. The identity

$$\rho\phi_i = (\phi_i\phi_j - \frac{1}{2}\delta_{ij}\phi_k\phi_k)_i \tag{1.33}$$

is a consequence of (1.5b) and it follows from this that the averaged 'self-force' is zero, i.e that

$$< F_i >= \int \phi_i \rho d^3x = 0, \tag{1.34}$$

which may be thought of as Newton's Third Law.
We may define a sequence of moments $q^{(n)}_{i\ldots j}$ by

$$q^{(n)}_{i\ldots j} = \int \rho \underbrace{x_i \ldots x_j}_{n} d^3x, \tag{1.35}$$

and other tensors $p^{(n)}_{i\ldots jk}$ and $s^{(n)}_{i\ldots jkm}$ by

$$p^{(n)}_{i\ldots jk} = \int \underbrace{x_{(i}\ldots x_j}_{n} J_{k)} d^3x, \tag{1.36a}$$

$$s^{(n)}_{i\ldots jkm} = \int \underbrace{x_{(i}\ldots x_j}_{n} S_{km)} d^3x. \tag{1.36b}$$

Then it follows from (1.22b) that

$$\dot{q}^{(n)}_{i\ldots j} = np^{(n-1)}_{i\ldots jk}, \tag{1.37a}$$

$$\dot{p}^{(n)}_{i\ldots jk} = ns^{(n-1)}_{i\ldots jkm}. \tag{1.37b}$$

In particular this means that $p^{(n)}_{i\ldots jk}$ and $s^{(n)}_{i\ldots jkm}$ are always zero for steady solutions.

1.2.5 Lie point symmetries

Following the general method of Stephani [24] we can find the Lie point symmetries of (1.5). These consist of rotations and translations together, with a generalised Galilean transformation which can be expressed as follows:

$$\underline{x} \to \hat{\underline{x}} = \underline{x} + \underline{P}(t), \tag{1.38a}$$

$$\psi \to \hat{\psi} = \psi(\underline{x} + \underline{P}, t) \exp[\frac{-\mathrm{i}}{2}\dot{\underline{P}}.\underline{x} + \frac{\mathrm{i}}{4}\int |\dot{\underline{P}}|^2 dt], \tag{1.38b}$$

$$\phi \to \hat{\phi} = \phi(\underline{x} + \underline{P}, t) + \frac{1}{2}\underline{x}.\ddot{\underline{P}}, \tag{1.38c}$$

which is a Galilean transformation if and only if $\ddot{\underline{P}} = \underline{0}$. By a Galilean transformation, we can reduce the total momentum to zero, and then by a translation we can place the centre-of-mass at the origin.

These clearly are all Lie point symmetries. The Galilean invariance of (1.5) was noted by Christian [5]. The analysis leading to the claim that there are no more Lie pont symmetries is straightforward but has not been published.

1.2.6 Dispersion

One of the moments is particularly significant, namely the dispersion:

$$< |\underline{x}|^2 >= q_{ii}^{(2)} = \int |\underline{x}|^2 \rho d^3x. \tag{1.39}$$

Following Arriola and Soler [2], we find

$$\ddot{q}_{ii}^{(2)} = 2(\int x_i \dot{J}_i d^3x) = 2\int S_{ii} d^3x = \int (8|\nabla\psi|^2 + 2\phi|\psi|^2) d^3x, \tag{1.40}$$

so that

$$\frac{d^2 < |\underline{x}|^2 >}{dt^2} = 4\mathcal{E} + 4T = 8\mathcal{E} - 2V. \tag{1.41}$$

Now recall that, as a consequence of the maximum principle (see e.g. [8]) ϕ is everywhere negative. Thus

$$\frac{d^2 < |\underline{x}|^2 >}{dt^2} > 8\mathcal{E}. \tag{1.42}$$

The dispersion grows at least quadratically with time if $\mathcal{E}$ is positive, but we cannot conclude this if $\mathcal{E}$ is negative.

1.3 Analytic results about the time-independent case

1.3.1 Existence and uniqueness

In Moroz and Tod [14] it was shown that the system (1.12) has infinitely many spherically-symmetric solutions, all with negative energy-eigenvalue and ψ real. (It is easy to see that ψ may always be assumed real in stationary solutions.) These authors did not show, but it is believed that, for each integer n there is a unique (up to sign) real spherically-symmetric solution with n zeroes, and the energy-eigenvalues increase monotonically in n to zero.

1.3.2 Variational formulation

The system (1.12) can be obtained from a variational problem, by seeking stationary points of the action

$$I = \frac{1}{2}(\mathcal{E} - E) = \int [\frac{1}{2}|\nabla\psi|^2 + \frac{1}{4}\phi|\psi|^2 - \frac{1}{2}E|\psi|^2]d^3x, \tag{1.43}$$

subject to

$$\nabla^2\phi = |\psi|^2, \tag{1.44}$$

or by solving (1.12b) with the relevant Green's function and considering,

$$I = \int [\frac{1}{2}|\nabla\psi|^2 - \frac{1}{16\pi}\int \frac{|\psi(x)|^2|\psi(y)|^2}{|x-y|}d^3y - \frac{1}{2}E|\psi|^2]d^3x. \tag{1.45}$$

If we vary $\psi \to \psi + \delta\psi$, $\phi \to \phi + \delta\phi$ then the first variation of (1.43) is

$$\delta I = \int \frac{1}{2}\delta\bar{\psi}(-\nabla^2\psi + \phi\psi - E\psi) + c.c]d^3x, \tag{1.46}$$

from which we obtain (1.12a) as the expected Euler-Lagrange equations, while the second variation, subject to (1.12b), is

$$\delta^2 I = \int [|\nabla\delta\psi|^2 + \phi|\delta\psi|^2 - \frac{1}{2}|\nabla\delta\psi|^2 - E|\delta\psi|^2]d^3x. \tag{1.47}$$

By exploiting various standard inequalities (Tod [25]) would show that the action is bounded below:

$$\mathcal{E} \geq -\frac{1}{54\pi^4}. \tag{1.48}$$

One expects that the direct method in the Calculus of Variations should now prove that the infimum of I is attained, that the minimising function is analytic (since the system (1.12) is elliptic) and that the minimising function will be the ground state found numerically by Moroz et al [15] and proved to exist by Moroz and Tod [14]. Note that, at the ground state, the second variation (1.47) cannot be negative.

1.3.3 Negativity of the energy-eigenvalue

We write E_0 for the energy-eigenvalue of the ground state. We now prove that for any stationary state the E energy-eigenvalue is negative. Following Tod [25], we define the tensor

$$T_{ij} = \psi_i\bar{\psi_j} + \bar{\psi_i}\psi_j + \phi_i\phi_j - \delta_{ij}(\psi_k\bar{\psi_k} + \frac{1}{2}\phi_k\phi_k + \phi|\psi|^2 - E|\psi|^2). \tag{1.49}$$

Then as a consequence of (1.12) it follows that

$$T_{ij,j} = 0, \tag{1.50}$$

whence

$$0 = \int (x_i T_{ij})_{,j} d^3x = \int T_{ii} d^3x, \tag{1.51}$$

so that

$$0 = \int [-|\nabla\psi|^2 - \frac{5}{2}\phi|\psi|^2 + 3E|\psi|^2] d^3x. \tag{1.52}$$

and

$$3E = T + \frac{5}{2}V. \tag{1.53}$$

With $E = T + V$ this shows that for a steady solution

$$T = -\frac{1}{3}E, \tag{1.54a}$$

$$V = \frac{4}{3}E, \tag{1.54b}$$

$$\mathcal{E} = \frac{1}{3}E, \tag{1.54c}$$

as $T > 0$ by definition it follows that $E < 0$.

Tod [25] shows that

$$0 \leq (-V) \leq \frac{4}{3\sqrt{3}\pi^2} T^{\frac{1}{2}}. \tag{1.55}$$

Arriola and Soler [2] have a stronger result, with $4\sqrt{\frac{-E_0}{3}} T^{\frac{1}{2}}$ on the right hand side. Since $\mathcal{E} = T + \frac{1}{2}V$ we may solve for T to find the bounds

$$T^{\frac{1}{2}} \leq \frac{1}{3\sqrt{3}\pi^2} + [\mathcal{E} + \frac{1}{27\pi^4}]^{\frac{1}{2}}, \tag{1.56a}$$

$$\geq \frac{1}{3\sqrt{3}\pi^2} - [\mathcal{E} + \frac{1}{27\pi^4}]^{\frac{1}{2}}. \tag{1.56b}$$

These results still hold at each instant for the time-dependent case, so that the kinetic and potential energies are separately bounded in terms of the action or conserved energy at all times.

1.4 Plan of thesis

In this thesis we start with a review of the methods which can be used to compute the stationary solutions in the case of spherical symmetry (chapter 2). Then in chapter 3 we consider the case of the linear perturbation about the spherically symmetric stationary solutions, obtaining an eigenvalue problem (3.24) to solve. Also in chapter 3 we obtain restrictions on the eigenvalues analytically such that the eigenvalues are purely real or imaginary or an integral condition on the eigenvectors vanishes (see section 3.4). In chapter 4 we solve the eigenvalue problem using spectral methods for the first few stationary solutions and check the results using Runge-Kutta integration as well as confirming the conditions on the eigenvalues. In chapter 5 we consider the problem of finding a numerical method to evolve the time-dependent S-N equations and we consider the boundary conditions to put on the numerical problem. Also in chapter 5 we consider adding a small heat term, called a sponge factor to the Schrödinger equation to absorb scatter waves. In chapter 6 we evolve numerically the problem with different initial conditions and check the convergence of the evolution method with different mesh and time step sizes. We consider the evolution of the stationary states to see if they are stable and we also look at the stationary states with added perturbation and compare the frequencies of oscillation with the linear perturbation theory. We consider the axisymmetric S-N equations in chapter 7 and look at the evolution as well as the time-independent equations. In chapter 8 we consider the 2-dimensional S-N equations (8.2) or equivalently the translationally symmetric case. The evolution is considered as well as the concept of finding two spinning lumps orbiting each other.

1.5 Conclusion

The results obtained from considering the spherically symmetric case are:

- The ground state is linearly stable.(See section 4.2)
- The linear perturbation about the n-th excited state, which is to say the $(n+1)$-th state, has n quadruples of complex eigenvalues as well as pure imaginary pairs. (See sections 4.3, 4.4)
- The ground state is stable under the full (nonlinear) evolution. (See section 6.5)
- Perturbations about the ground state oscillate with the frequencies obtained by the linear perturbation theory. (See section 6.5)
- The higher states are unstable and will decay into a "multiple" of the ground state, while emitting some scatter off the grid. (See section 6.3)

- The decay time for higher states is controlled by the growing linear mode obtained in the linear perturbation theory. (See section 6.3)
- Perturbations about higher states will oscillate for a while (until they decay) according to the linear oscillation obtained by the linear perturbation theory. (See section 6.3)
- The evolution of different exponential lumps indicates that any initial condition appears to decay, that is they scatter and leave a "multiple" of the ground state. (See section 6.4)

The results obtained from considering the axially symmetric case are:

- Stationary solutions that are axisymmetric exist and the first one is like the dipole of the Hydrogen atom. (See section 7.4)
- Evolution of the dipole-like solution shows that it is unstable in the same way as the spherically symmetric stationary solutions are, that is it emits scatter off the grid leaving a "multiple" of the ground state and that lumps of probability density attract each other. (See section 7.5)

The results obtained from considering the 2-dimensional case are:

- Evolution of the higher states are unstable, emitting scatter and leaving a "multiple" of the ground state. (See section 8.4)
- There exist rotating solutions, but these are unstable. (See section 8.5)
- Lumps of probability density attract each other and come together emitting scatter and leave a "multiple" of the ground state. (See section 8.4)

Chapter 2

Spherically-symmetric stationary solutions

2.1 The equations

In the case of spherical symmetry and time-independence we can assume without loss of generality ψ is real. So (1.14) becomes

$$(rS)'' = -rSV, \tag{2.1a}$$

$$(rV)'' = -rS^2. \tag{2.1b}$$

We have the boundary conditions $S \to 0$ as $r \to \infty$ and $S' = 0 = V'$ at $r = 0$. We also note that if (S, V, r) is a solution then so is $(\lambda^2 S, \lambda^2 V, \lambda^{-1} r)$. At large r bounded solutions to (2.1) decay like

$$V = A - \frac{B}{r}, \tag{2.2a}$$

$$S = \frac{C}{r} e^{-kr}, \tag{2.2b}$$

where

$$A = V_0 - \int_0^\infty x S^2 dx, \tag{2.3a}$$

$$B = \int_0^\infty x^2 S^2 dx, \tag{2.3b}$$

where V_0 is the initial value of the potential V, (i.e. $V_0 = V(0)$).

2.2 Computing the spherically-symmetric stationary states

2.2.1 Runge-Kutta integration

Now (2.1) can be re-written as a system of four first order ODEs

$$Y_1' = Y_2, \tag{2.4a}$$
$$Y_2' = \frac{-2Y_2}{r} - Y_2 Y_4, \tag{2.4b}$$
$$Y_3' = Y_4, \tag{2.4c}$$
$$Y_4' = \frac{-2Y_4}{r} - Y_4^2, \tag{2.4d}$$

where $Y_1 = S$ and $Y_3 = V$.

The numerical technique used by Moroz et al [15] to obtain the stationary states uses fourth-order Runge-Kutta integration on (2.4), starting at $r = 0$ and integrating outwards towards infinity. The initial values are picked so that the boundary conditions at $r = 0$ are satisfied and then other values are guessed. The normalisation invariance allows for either $V(0), S(0)$ be set equal to a chosen constant. The states are obtained by refining the initial guess so that a solution which tends to zero when r is large is obtained.

There are various methods for obtaining the correct initial values which correspond to stationary states. The method used by Moroz et al [15] was to integrate up to a fixed value in the domain, waiting for the routine to fail or blow up, then plotting the solution so far, and refine the guess based on which way the function blows up.

Another is a shooting method, which involves integration until some value for r and looking at the value of S at that point, then modifying the initial values such that the S at that point is zero. The problem with this routine is that as the function blows up the step size decreases so that the tolerance remains low. This increases the computation time needed. Also the routine will fail if the routine takes to long, so the value for r cannot be too large.

To calculate the stationary states, I have used a modified shooting method, to avoid the problem which occurs with Runge-Kutta integrations, which is the exponential blow up of solutions. I have modified the integration in such way that the program will integrate over small steps using a 4th order Runge-Kutta NAG routine. After each step it will terminate if the solution is too large in absolute magnitude. In the case where $V_0 = 1$ we say that 1.5 is large since the first state occurs around 1.088. It will also terminate if the solution for S blows up exponentially. From information on which side of the axis the solution becomes unbounded we can refine the initial conditions to obtain the eigenfunction. Using this method we are able to obtain the first 50 values for S_0 or equivalently the first 50 energy levels.

For the stationary spherically symmetric case it was shown in Moroz et al [15] that the eigenvalue is

$$\tilde{E} = \frac{2G^2m^5}{\hbar^2}\frac{A}{B^2}, \tag{2.5}$$

where A and B are given by (2.3), and V_0 is the initial value of the potential function V.

Using the above method we calculated A, B and $\frac{2A}{B}$ which is the energy up to a factor of $\frac{G^2m^5}{\hbar^2}$ and which compare with the first 16 eigenvalues of Jones et al [3]. The first 20 eigenvalues calculated with the above routine are given in Table 2.1. We also plot in figure 2.1 the first four spherically symmetric states normalised such that $\int |\psi|^2 d^3x = 4\pi$. Note that the n-th state has $(n-1)$ zeroes or "nodes".

Number of zeros	Energy Eigenvalue	Jones et al [3] Eigenvalues
0	-0.16276929132192	-0.163
1	-0.03079656067054	-0.0308
2	-0.01252610801692	-0.0125
3	-0.00674732963038	-0.00675
4	-0.00420903256689	-0.00421
5	-0.00287386420271	-0.00288
6	-0.00208619042678	-0.00209
7	-0.00158297244845	-0.00158
8	-0.00124207860434	-0.00124
9	-0.00100051995162	-0.00100
10	-0.00082314193054	-0.000823
11	-0.00068906850493	-0.000689
12	-0.00058527053127	-0.000585
13	-0.00050327487416	-0.000503
14	-0.00043737620824	-0.000437
15	-0.00038362194847	-0.000384
16	-0.00033920111442	
17	-0.00030207158301	
18	-0.00027072080257	
19	-0.00024400868816	
20	-0.00022106369652	

Table 2.1: The first 20 eigenvalues

2.2.2 Alternative method

An alternative method, which we use later on in chapters 4 and 6 to compute the eigenfunction at the Chebyshev points is given below. Jones et al [3] used an iterative numerical scheme for computing the n-node stationary states, instead of using a shooting method. An outline of their method is as follows:

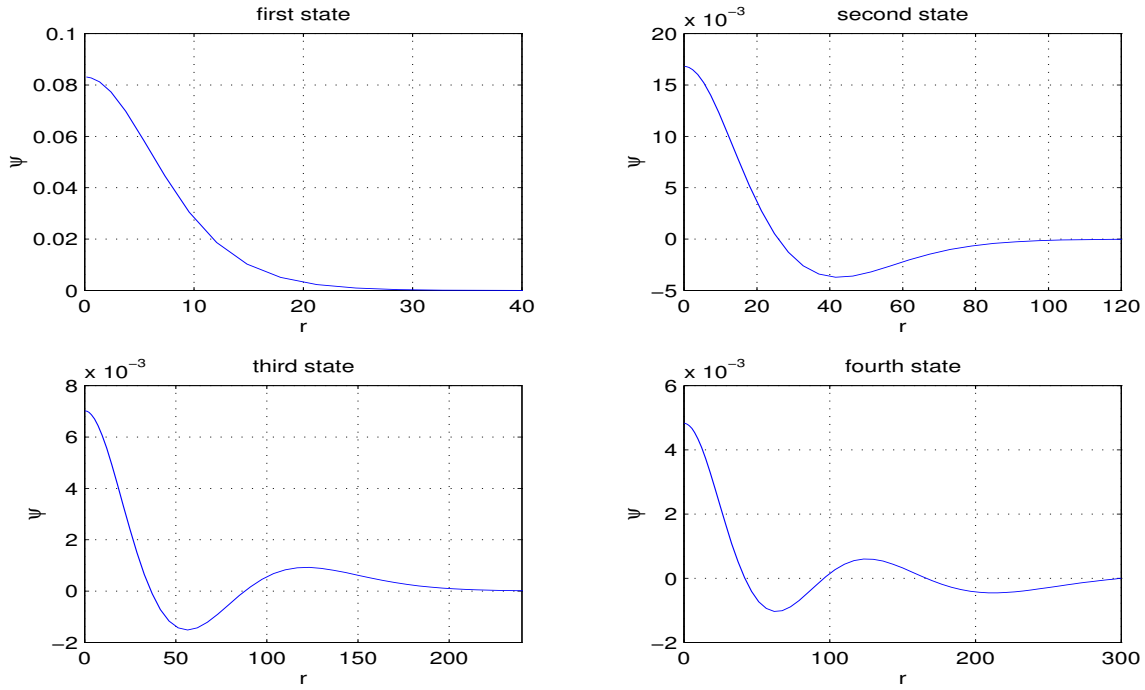


Figure 2.1: First four eigenfunctions

1. Set an outer radius R.

2. Supply an initial guess for $u_n = r\psi_n$.

3. Solve for Φ in
$$\frac{\partial^2 \Phi}{\partial r^2} + \frac{2}{r}\frac{\partial \Phi}{\partial r} = r^{-2}u_n^2. \tag{2.6}$$

4. Solve for the n-node eigenvalue and eigenfunction of
$$\frac{\partial^2 u_n}{\partial r^2} = 2u_n(\Phi - \epsilon). \tag{2.7}$$

5. Iterate the previous two steps until the eigenvalue converges sufficiently, a typical criterion being that the change in ϵ_n from one iteration to the next is less than 10^{-9}.

6. Iterate the previous five steps, increasing R until ϵ_n stops changing with the change in R.

2.3 Approximations to the energy of the bound states

Jones et al [3] claim, that the energy eigenvalues $\frac{2A}{B^2}$ closely follow a least-squares fit of the formula:
$$E_n = -\frac{\alpha}{(n+\beta)^\gamma}, \tag{2.8}$$

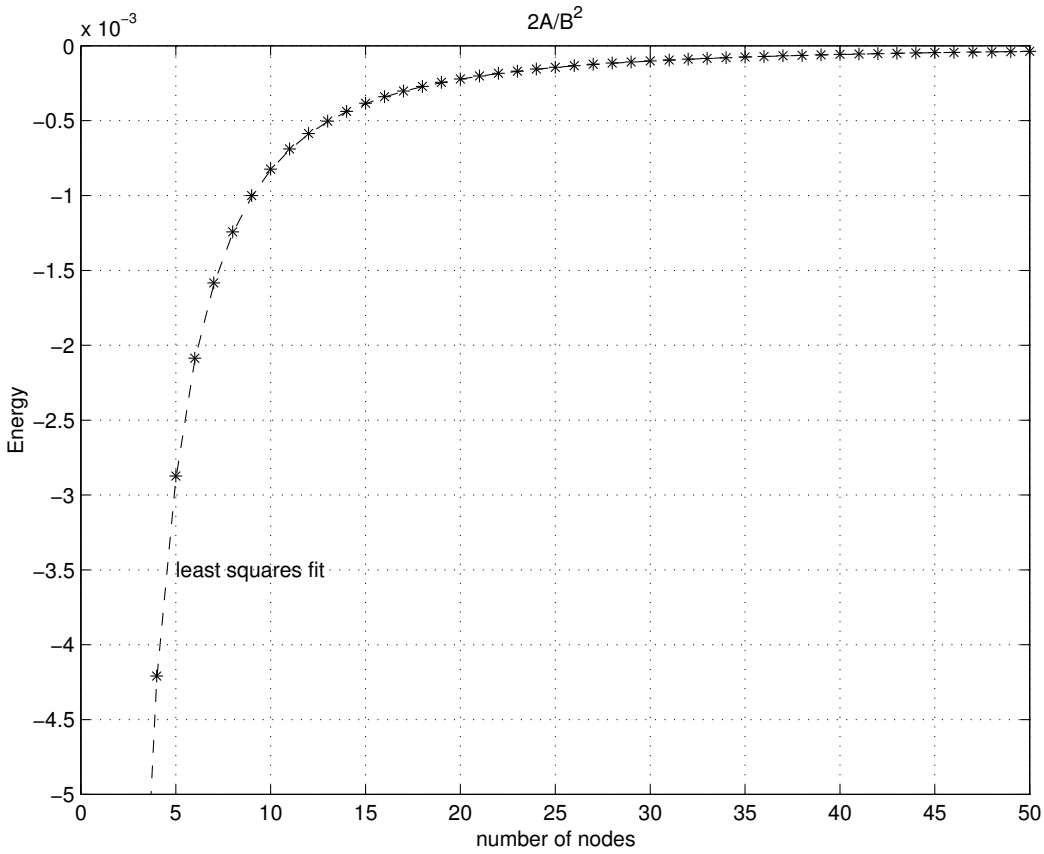


Figure 2.2: The least squares fit of the energy eigenvalues

where $\alpha = 0.096$, $\beta = 0.76$ and $\gamma = 2.00$. This appears to be a very good fit of the first 50 eigenvalues. The eigenvalues and the fit are plotted in figure 2.2. Moroz et al [15] claim that the the number of zeros is of the order of $\exp(\frac{V_0^2}{S_0^2})$. Figure 2.3 shows that $\exp(\frac{V_0^2}{S_0^2})$ is an over estimate for the number of nodes, which appears to converge as n gets large.

The log-log plot figure 2.4 shows that we have a gradient of -2 as n gets large. This is what we expect since $E_n = \frac{-\alpha}{(n+\beta)^\gamma}$ is such a good fit when $\gamma = 2$.

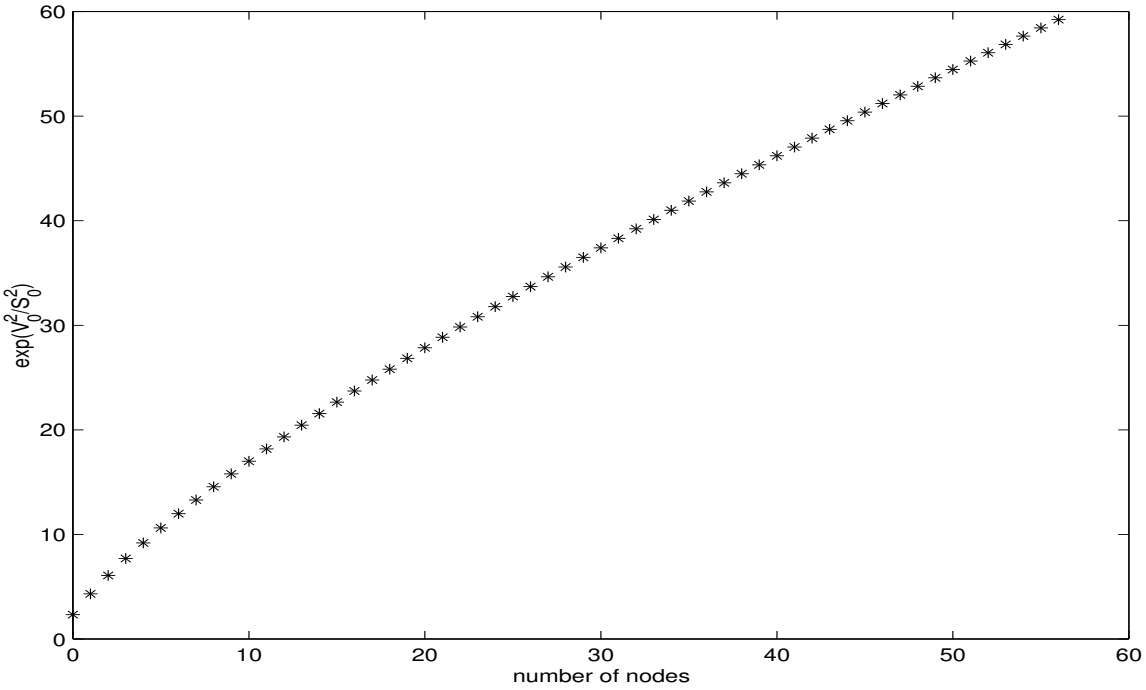


Figure 2.3: The number of nodes with the Moroz et al [15] prediction of the number of nodes

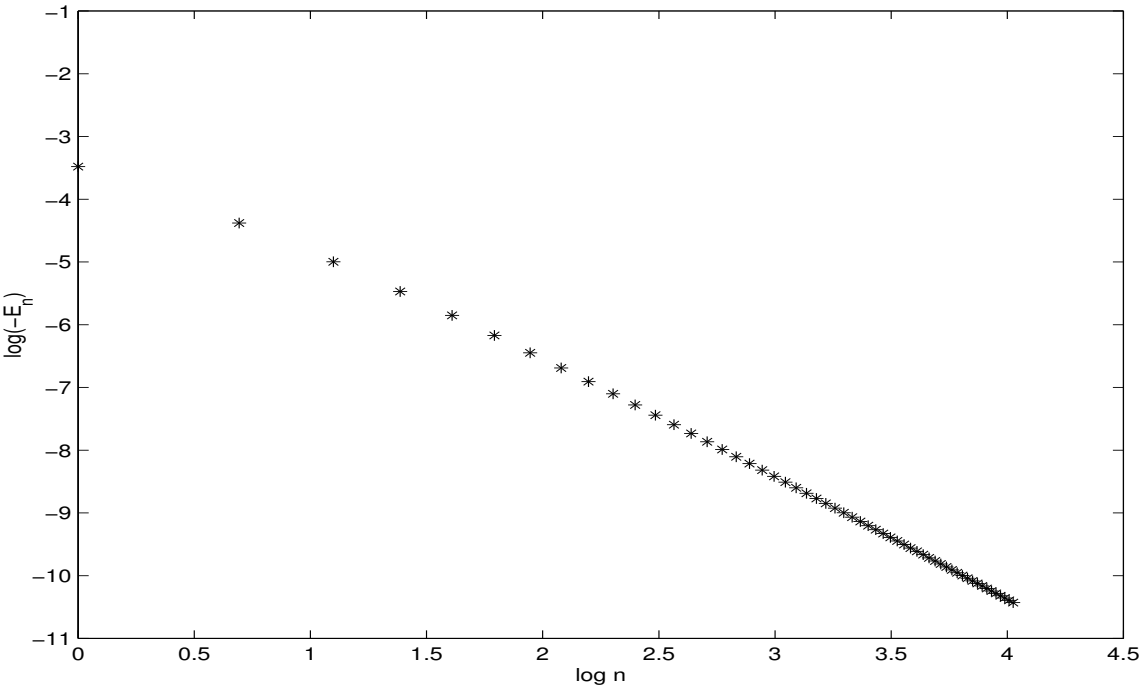


Figure 2.4: Log-log plot of the energy values with $\log(n)$

Chapter 3

Linear stability of the spherically-symmetric solutions

3.1 Linearising the S-N equations

In this chapter 3 we shall set up the linear stability problem, ready for numerical solution in chapter 4.

We look for a solution to (1.5) of the form :

$$\psi = \psi_0(r,t) + \epsilon\psi_1(r,t) + \epsilon^2\psi_2(r,t) + \ldots, \tag{3.1a}$$

$$\phi = \phi_0(r,t) + \epsilon\phi_1(r,t) + \epsilon^2\phi_2(r,t) + \ldots, \tag{3.1b}$$

where $\epsilon \ll 1$. Substitution of (3.1) into the S-N equations (1.5) gives:

$$\begin{aligned} \mathrm{i}\psi_{0t} + \nabla^2\psi_0 + \epsilon[i\hbar\psi_{1t} + \nabla^2\psi_1] + \epsilon^2[\mathrm{i}\hbar\psi_{2t} + \nabla^2\psi_2] = \\ \psi_0\phi_0 + \epsilon(\psi_0\phi_1 + \psi_1\phi_0) + \epsilon^2(\psi_0\phi_2 + \psi_1\phi_1 + \psi_2\phi_0), \end{aligned} \tag{3.2a}$$

$$\begin{aligned} \nabla^2\phi_0 + \epsilon\nabla^2\phi_1 + \epsilon^2\nabla^2\phi_2 = \\ \psi_0^2 + \epsilon(\psi_0\bar{\psi}_1 + \bar{\psi}_0\psi_1) + \epsilon^2(\psi_0\bar{\psi}_2 + \psi_1\bar{\psi}_1 + \bar{\psi}_0\psi_2). \end{aligned} \tag{3.2b}$$

Finally equating the powers of ϵ we obtain
ϵ^0:

$$\mathrm{i}\psi_{0t} + \nabla^2\psi_0 = \psi_0\phi_0, \tag{3.3a}$$

$$\nabla^2\phi_0 = |\psi_0|^2, \tag{3.3b}$$

ϵ^1:

$$\mathrm{i}\psi_{1t} + \nabla^2\psi_1 = \psi_0\phi_1 + \psi_1\phi_0, \tag{3.4a}$$

$$\nabla^2\phi_1 = \psi_0\bar{\psi}_1 + \bar{\psi}_0\psi_1, \tag{3.4b}$$

ϵ^2:

$$\begin{aligned} \mathrm{i}\psi_{2t} + \nabla^2\psi_2 &= \psi_0\phi_2 + \psi_1\phi_1 + \psi_2\phi_0, && (3.5a)\\ \nabla^2\phi_2 &= \psi_0\bar{\psi}_2 + \psi_1\bar{\psi}_1 + \bar{\psi}_0\psi_2. && (3.5b) \end{aligned}$$

We consider the case of spherical symmetry so that
$\nabla^2 f = \dfrac{1}{r^2}(r^2 f_r)_r = \dfrac{1}{r}(rf)_{rr}$, then (3.3) becomes

$$\begin{aligned} \mathrm{i}(r\psi_0)_t + (r\psi_0)_{rr} &= r\psi_0\phi_0, && (3.6a)\\ (r\phi_0)_{rr} &= r\psi_0\bar{\psi}_0. && (3.6b) \end{aligned}$$

Since we are interested in the stability of the stationary problem we take

$$\begin{aligned} \psi_0 &= R_0(r)\mathrm{e}^{-\mathrm{i}Et}, && (3.7a)\\ \phi_0 &= E - V_0(r), && (3.7b) \end{aligned}$$

where R_0 is real, so that

$$\begin{aligned} (rR_0)_{rr} &= -rR_0V_0, && (3.8a)\\ (rV_0)_{rr} &= -rR_0^2. && (3.8b) \end{aligned}$$

Substituting into (3.4), the $O(\epsilon)$ problem we have

$$\begin{aligned} \mathrm{i}(r\psi_1)_t + (r\psi_1)_{rr} &= r(E - V_0)\psi_1 + r\phi_1 R_0(r)\mathrm{e}^{-\mathrm{i}Et}, && (3.9a)\\ (r\phi_1)_{rr} &= rR_0(r)\mathrm{e}^{-\mathrm{i}Et}\bar{\psi}_1 + rR_0(r)\mathrm{e}^{\mathrm{i}Et}\psi_1. && (3.9b) \end{aligned}$$

To eliminate $\mathrm{e}^{-\mathrm{i}Et}$, we seek solutions of the form

$$\begin{aligned} \psi_1 &= R_1(r,t)\mathrm{e}^{-\mathrm{i}Et}, && (3.10a)\\ \phi_1 &= \phi_1, && (3.10b) \end{aligned}$$

where R_1 is complex and ϕ_1 is real so that (3.9) simplifies to give

$$\begin{aligned} \mathrm{i}(rR_1)_t + (rR_1)_{rr} &= R_0(r\phi_1) - V_0(rR_1), && (3.11a)\\ (r\phi_1)_{rr} &= R_0(r\bar{R}_1) + \bar{R}_0(rR_1). && (3.11b) \end{aligned}$$

For convenience we introduce $P = r\phi_1$, $R = rR_1$. Note that P and R must vanish at the origin. The $O(\epsilon)$ problem then becomes,

$$\begin{aligned} \mathrm{i}R_t + R_{rr} &= R_0P - V_0R, && (3.12a)\\ \P_{rr} &= R_0\bar{R} + \bar{R}_0R. && (3.12b) \end{aligned}$$

3.2 Separating the $O(\epsilon)$ equations

We look for a solution of the $O(\epsilon)$ problem in the form

$$R = (A+B)\mathrm{e}^{\lambda t} + (\bar{A}-\bar{B})\mathrm{e}^{\bar{\lambda} t}, \tag{3.13a}$$

$$P = W_1\mathrm{e}^{\lambda t} + W_2\mathrm{e}^{\bar{\lambda} t}, \tag{3.13b}$$

where we assume for now that λ is not real and A, B, W_1 and W_2 are functions which are time independent. As we are considering the spherical symmetric case they depend upon r only. Now since P is real we note that $W_1 = \bar{W}_2$, so we can let $W = W_1 = \bar{W}_2$. Substituting into (3.12a) gives,

$$\begin{aligned} &\mathrm{i}(\lambda(A+B)\mathrm{e}^{\lambda t} + \bar{\lambda}(\bar{A}-\bar{B})\mathrm{e}^{\bar{\lambda} t}) + (A_{rr}+B_{rr})\mathrm{e}^{\lambda t} + (\bar{A}_{rr}-\bar{B}_{rr})\mathrm{e}^{\bar{\lambda} t} = \\ &R_0(W\mathrm{e}^{\lambda t} + \bar{W}\mathrm{e}^{\bar{\lambda} t}) - V_0((A+B)\mathrm{e}^{\lambda t} + (\bar{A}-\bar{B})\mathrm{e}^{\bar{\lambda} t}). \end{aligned} \tag{3.14}$$

Equating the coefficients of $\mathrm{e}^{\lambda t}$, $\mathrm{e}^{\bar{\lambda} t}$ and noting that we can do this only if $\bar{\lambda} \neq \lambda$ so that λ is not real, the coefficient of $\mathrm{e}^{\lambda t}$ gives

$$\mathrm{i}\lambda(A+B) + (A_{rr}+B_{rr}) = R_0 W - V_0(A+B), \tag{3.15}$$

while the coefficient of $\mathrm{e}^{\bar{\lambda} t}$ gives

$$\mathrm{i}\bar{\lambda}(\bar{A}-\bar{B}) + (\bar{A}_{rr}-\bar{B}_{rr}) = R_0\bar{W} - V_0(\bar{A}-\bar{B}). \tag{3.16}$$

Substituting into (3.12b) we obtain

$$\begin{aligned} W_{rr}\mathrm{e}^{\lambda t} + \bar{W}_{rr}\mathrm{e}^{\bar{\lambda} t} &= R_0((\bar{A}+\bar{B})\mathrm{e}^{\bar{\lambda} t} + (A-B)\mathrm{e}^{\lambda t}) + \\ &\quad \bar{R}_0((A+B)\mathrm{e}^{\lambda t} + (\bar{A}-\bar{B})\mathrm{e}^{\bar{\lambda} t}). \end{aligned} \tag{3.17}$$

Again equating coefficients of $\mathrm{e}^{\lambda t}$, $\mathrm{e}^{\bar{\lambda} t}$ gives,

$$W_{rr} = R_0(A-B) + \bar{R}_0(A+B), \tag{3.18}$$

and

$$\bar{W}_{rr} = R_0(\bar{A}+\bar{B}) + \bar{R}_0(\bar{A}-\bar{B}). \tag{3.19}$$

Since $R_0 = \bar{R}_0$ we have

$$W_{rr} = 2R_0 A, \tag{3.20a}$$

$$\bar{W}_{rr} = 2R_0\bar{A}, \tag{3.20b}$$

which is just one equation, so we have,

$$R_0 W = \mathrm{i}\lambda(A+B) + (A_{rr} + B_{rr}) + V_0(A+B), \tag{3.21a}$$

$$\overline{(R_0 \bar{W})} = R_0 W = -\mathrm{i}\lambda(A-B) + (A_{rr} - B_{rr}) + V_0(A-B). \tag{3.21b}$$

Therefore

$$B_{rr} + V_0 B = -\mathrm{i}\lambda A, \tag{3.22}$$

and

$$R_0 W = \mathrm{i}\lambda B + A_{rr} + V_0 A. \tag{3.23}$$

To summarise, the $O(\epsilon)$ problem leads to three coupled linear O.D.E's for the perturbation:

$$W_{rr} = 2R_0 A, \tag{3.24a}$$

$$B_{rr} + V_0 B = -\mathrm{i}\lambda A, \tag{3.24b}$$

$$A_{rr} + V_0 A = R_0 W - \mathrm{i}\lambda B. \tag{3.24c}$$

If λ is real, we note that (3.13b) reduces to,

$$R = A\mathrm{e}^{\lambda t}, \tag{3.25a}$$

$$P = W\mathrm{e}^{\lambda t}. \tag{3.25b}$$

Substituting into the $O(\epsilon)$ equations (3.12) gives,

$$\mathrm{i}\lambda A\mathrm{e}^{\lambda t} + A_{rr}\mathrm{e}^{\lambda t} = R_0 W\mathrm{e}^{\lambda t} - V_0 A\mathrm{e}^{\lambda t}, \tag{3.26a}$$

$$W_{rr}\mathrm{e}^{\lambda t} = R_0 \bar{A}\mathrm{e}^{\lambda t} + R_0 A\mathrm{e}^{\lambda t}. \tag{3.26b}$$

Equating coefficients of $\mathrm{e}^{\lambda t}$ we obtain

$$\mathrm{i}\lambda A + A_{rr} = R_0 W - V_0 A, \tag{3.27a}$$

$$W_{rr} = R_0(A + \bar{A}). \tag{3.27b}$$

If we let $A = a + \mathrm{i}b$ where a and b are real functions of r, and substitute into (3.27) we obtain

$$\mathrm{i}\lambda(a + \mathrm{i}b) + (a_r r + \mathrm{i}b_{rr}) = R_0 W - V_0(a + \mathrm{i}b), \tag{3.28a}$$

$$W_{rr} = 2R_0 a, \tag{3.28b}$$

so that equating real and imaginary parts gives

$$W_{rr} = 2R_0 a, \tag{3.29a}$$

$$b_{rr} + V_0 b = -\lambda a, \tag{3.29b}$$

$$a_{rr} + V_0 a = R_0 W - \lambda b. \tag{3.29c}$$

We consider the real eigenvalues of

$$\begin{aligned} W_{rr} &= 2R_0 A, && (3.30a)\\ B_{rr} + V_0 B &= -\mathrm{i}\lambda A, && (3.30b)\\ A_{rr} + V_0 A &= R_0 W - \mathrm{i}\lambda B. && (3.30c) \end{aligned}$$

Suppose that the eigenvalues are such that A is real. This implies that B is imaginary, while W is real. We can therefore set $A = a$ and $B = \mathrm{i}b$ where a,b are real functions of r. Then (3.30) become

$$\begin{aligned} W_{rr} &= 2R_0 a, && (3.31a)\\ b_{rr} + V_0 b &= -\lambda a, && (3.31b)\\ a_{rr} + V_0 a &= R_0 W - \lambda b, && (3.31c) \end{aligned}$$

which are the same as (3.28), so (3.24) covers both.

We note that $\lambda = 0$ will be an eigenvalue of (3.28), with $a = 0$, $b = \mathrm{i}R_0$ and $W = 0$, but this corresponds to just a rotation in the phase factor.

We note that (3.24) transformed by $(A, B, W) \to (A, -B, W)$ becomes,

$$\begin{aligned} W_{rr} &= 2R_0 A, && (3.32a)\\ B_{rr} + V_0 B &= \mathrm{i}\lambda A, && (3.32b)\\ A_{rr} + V_0 A &= R_0 W + \mathrm{i}\lambda B, && (3.32c) \end{aligned}$$

so if λ is eigenvalue then $-\lambda$ is a eigenvalue. Also if of $(A, B, W) \to (\bar{A}, \bar{B}, \bar{W})$ then the equations become

$$\begin{aligned} W_{rr} &= 2R_0 A, && (3.33a)\\ B_{rr} + V_0 B &= -\mathrm{i}\bar{\lambda} A, && (3.33b)\\ A_{rr} + V_0 A &= R_0 W - \mathrm{i}\bar{\lambda} B, && (3.33c) \end{aligned}$$

and if λ is an eigenvalue then $-\bar{\lambda}$ is an eigenvalue. So if λ is an eigenvalue then so are $\bar{\lambda}$,$-\lambda$ and $-\bar{\lambda}$. That is in the case of λ complex, eigenvalues exist in groups of four, otherwise they exist in pairs or the singleton $\lambda = 0$.

3.3 Boundary Conditions

We now consider the boundary conditions for the $O(\epsilon)$ problem (3.12). Since $\phi_1 = \frac{1}{r}P$ and $R_1 = \frac{1}{r}R$ and we require $\phi_0 + \epsilon\phi_1$ to be a physically representable potential function and

$R_0 + \epsilon R_1$ to be a physically representable wave-function, ϕ_1 and R_1 must be well-behaved functions of r at r = 0. This implies $\phi_1, R_1 \rightarrow$ finite values as $r \rightarrow 0$, so that

$$P(0) = 0, R(0) = 0. \tag{3.34}$$

The condition that $R_0 + \epsilon R_1$ be a physically representable wave-function is that it be normalisable, i.e the following integral exists:

$$\int_0^\infty (R_0 + \epsilon R_1)\overline{(R_0 + \epsilon R_1)} r^2 dr. \tag{3.35}$$

This becomes

$$\int_0^\infty (R_0\bar{R}_0 + \epsilon(\bar{R}_1 R_0 + \bar{R}_0 R_1) + \epsilon^2 R_1 \bar{R}_1) r^2 dr, \tag{3.36}$$

and upon equating coefficients of ϵ we require the following integrals to exist:
coefficient of ϵ^0

$$\int_0^\infty (R_0\bar{R}_0) r^2 dr, \tag{3.37}$$

coefficient of ϵ^1

$$\int_0^\infty (\bar{R}_1 R_0 + \bar{R}_0 R_1) r^2 dr, \tag{3.38}$$

coefficient of ϵ^2

$$\int_0^\infty (R_1\bar{R}_1) r^2 dr. \tag{3.39}$$

The $O(1)$ condition (3.37) requires the wave-function of the stationary spherically symmetric problem to be normalisable, and the remaining two integrals become, in terms of R,

$$\int_0^\infty (\bar{R} + R) R_0 r dr, \tag{3.40a}$$

$$\int_0^\infty (R\bar{R}) dr. \tag{3.40b}$$

The integral (3.40a) exists provided $\bar{R}+R$ does not grow exponentially with r, as R_0 behaves like e^{Er} as $r \rightarrow \infty$. The integral (3.40b) implies that $|R|^2 \rightarrow 0$ as $r \rightarrow \infty$. Hence $R \rightarrow 0$ as $r \rightarrow \infty$ is a necessary but not a sufficient condition for the wave-function to be normalisable. We expect that the solution for which this is not the case will blow up exponentially, so that the boundary condition that $R \rightarrow 0$ as $r \rightarrow \infty$ should be sufficient. For the potential $\phi_0 + \epsilon\phi_1$ we require function to be well-behaved at $r = 0$. We also have an arbitrary scaling to the potential function, corresponding to the freedom we have of where to set the zero on the energy scale. We can therefore choose $\phi_0 \rightarrow 0$ and $\phi_1 \rightarrow 0$ as $r \rightarrow \infty$.

When $R = (A+B)\mathrm{e}^{\lambda t} + (\bar{A}-\bar{B})\mathrm{e}^{\bar{\lambda}t}$ and $P = W\mathrm{e}^{\lambda t} + \bar{W}\mathrm{e}^{\bar{\lambda}t}$, the boundary conditions on R and ϕ imply that,

$$\begin{aligned} R(0) = 0 &\Rightarrow A(0) = 0, B(0) = 0, \\ P(0) = 0 &\Rightarrow W(0) = 0, \\ R(\infty) = 0 &\Rightarrow A(\infty) = 0, B(\infty) = 0, \\ (rP)(\infty) = 0 &\Rightarrow W(\infty) = 0. \end{aligned}$$

So to summarise the boundary condition on A, B and W are

$$A(0) = 0, B(0) = 0, W(0) = 0, A(\infty) = 0, B(\infty) = 0, W(\infty) = 0. \tag{3.41}$$

3.4 Restriction on the possible eigenvalues

We claim that λ^2 is real for the perturbation equations unless $\displaystyle\int_0^\infty \bar{A}B dr = 0$. The following proof is due to Tod [26]. Consider (3.24) where (R_0, V_0) satisfy

$$(rR_0)'' = -rR_0V_0, \tag{3.42a}$$

$$(rV_0)'' = -rR_0^2. \tag{3.42b}$$

We now consider

$$\begin{aligned} -\mathrm{i}\lambda \int_0^\infty A\bar{B}dr &= \int_0^\infty B_{rr}\bar{B} + V_0 B\bar{B}dr, \\ &= \int_0^\infty V_0|B|^2 - |B_r|^2 dr. \end{aligned} \tag{3.43}$$

We note that the R.H.S of (3.43) is real since V_0 is real. We also consider:

$$\begin{aligned} -\mathrm{i}\lambda \int_0^\infty \bar{A}Bdr &= \int_0^\infty A_{rr}\bar{A}V_0 A\bar{A} - R_0 W\bar{A}dr, \\ &= \int_0^\infty V_0|A|^2 - |A_r|^2 - \frac{1}{2}|W_r|^2 dr. \end{aligned} \tag{3.44}$$

We note that the R.H.S of (3.44) is real since V_0 is real. Combining the two results we have that:

$$\frac{-\mathrm{i}\lambda \int_0^\infty A\bar{B}dr}{\mathrm{i}\bar{\lambda}\int_0^\infty A\bar{B}dr}, \tag{3.45}$$

is real if $\mathrm{i}\bar{\lambda}\displaystyle\int_0^\infty A\bar{B}dr \neq 0$ which implies that $\dfrac{\lambda}{\bar{\lambda}}$ is real i.e that λ^2 is real . Hence either λ^2 is real or $\displaystyle\int_0^\infty A\bar{B}dr = 0$.

3.5 An inequality on Re(λ)

From the linearised perturbation system (3.24) it is possible to prove an inequality for the real part of λ . First we obtain

$$\mathrm{i}\int[\lambda A\bar{A}+\bar{\lambda}B\bar{B}]d^3x=-\int R_0B\bar{W}d^3x, \tag{3.46}$$

and then by the Holder and Sobolev inequalities, as in Tod [25], we find

$$|\int R_0B\bar{W}|\leq C_1(\int|A|^2)^{\frac{1}{2}}(\int|B|^2)^{\frac{1}{2}}(\int|R_0|^3)^{\frac{2}{3}}, \tag{3.47}$$

with $C_1=\dfrac{2^{\frac{4}{3}}}{3\pi^{\frac{4}{3}}}$. We choose a normalisation of the perturbation so that

$$\int|A|^2=\cos^2\theta,\int|B|^2=\sin^2\theta, \tag{3.48}$$

and set $\lambda=\lambda_R+\mathrm{i}\lambda_I$ to find from (3.46)

$$|\lambda_R+\mathrm{i}\lambda_I\cos 2\theta|\leq 2C_1\sin\theta\cos\theta(\int|R_0|^3)^{\frac{2}{3}}. \tag{3.49}$$

Now use Sobolev and Section 1.2 to find

$$\int|R_0|^3\leq C_1^{\frac{3}{4}}(-\frac{E}{3})^{\frac{3}{4}}, \tag{3.50}$$

so finally

$$|\lambda_R|\leq\frac{4}{9\pi^2}\sqrt{-E}. \tag{3.51}$$

Note that if the normalisation is different to one we need to rescale the E value.

Chapter 4

Numerical solution of the perturbation equations

4.1 The method

The $O(1)$ perturbation equations (3.24) are linear whereas the S-N equations are not. We can therefore solve these equations using spectral methods, by approximating the problem to that of solving a matrix eigenvalue problem. See [27] for more details about spectral methods. We use Chebyshev polynomial interpolation to get a differentiation matrix, where the Chebyshev polynomials are such that

$$p_n(x) = \cos(n\theta), \tag{4.1}$$

with $\theta = \cos^{-1}(x)$. They also satisfy the differential equation

$$\frac{d}{dx}((1-x^2)^{\frac{1}{2}}\frac{dp_n(x)}{dx}) = n^2 p_n(x). \tag{4.2}$$

When using Chebyshev polynomials we sample our data at Chebyshev points that is at $x_i = \cos(\frac{i\pi}{N})$ where $i = 0, 1 \ldots N, N+1$ and $N+1$ is the number of Chebyshev points. We note that these points correspond to zeros of the Chebyshev polynomials.

We let $p(x)$ be the unique polynomial of degree N or less such that $p(x_i) = v_i$ for $i = 0, 1 \ldots N, N+1$, where the v_i's are values of a function at the Chebyshev points. Define w_i's by $w_i = p'(x_i)$ for $i = 0, 1 \ldots N, N+1$. The differentiation matrix for polynomials of degree N, denoted D_N, is defined to be such that

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \\ w_N + 1 \end{pmatrix} = D_N \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \\ v_N + 1 \end{pmatrix}, \tag{4.3}$$

which is :

$$
\begin{aligned}
D_{0,0} &= \frac{2N^2+1}{6}, \\
D_{N,N} &= -\frac{2N^2+1}{6}, \\
D_{i,i} &= -\frac{x_i}{2(1-x_i^2)} \quad 1 \le i \le N-1, \\
D_{i,j} &= \frac{c_i}{c_j}\frac{(-1)^{i+j}}{(x_i - x_j)} \qquad (i \neq j),
\end{aligned} \tag{4.4a}
$$

where $c_i = 2$ for $i = 0$ or $i = N$ and $c_i = 1$ otherwise.

The second derivative matrix is just D_N^2 since the $p'(x)$ is the unique polynomial of degree N or less through w_i's.

We note that since the interval we are interested in is $[0, L]$ instead of $[-1, 1]$ we rescale points by $X_i = \frac{L}{2}(1 + x_i)$. We also need to rescale the differentiation matrix so $D_{[0,L]} = \frac{1}{L} D_{[-1,1]}$, all differentiation matrix below are on the interval $[0, L]$. The requirement that A,B and W be zero at the boundary is applied by deleting the first and last rows and columns of the differentiation matrix in question, since $A(0) = 0 = A(L)$, $B(0) = 0 = B(L)$ and $W(0) = 0 = W(L)$. In this case it is the D_N^2 matrix. This yields a $(N-1) \times (N-1)$ matrix denoted by $\tilde{D}_N^2$.

For the perturbation equations we therefore obtain the matrix eigenvalue equation

$$
\begin{pmatrix} -2R_0 & 0 & \tilde{D}_N^2 \\ 0 & \tilde{D}_N^2 + V_0 & 0 \\ -\tilde{D}_N^2 - V_0 & 0 & R_0 \end{pmatrix}
\begin{pmatrix} A \\ B \\ W \end{pmatrix}
= \mathrm{i}\lambda
\begin{pmatrix} 0 & 0 & 0 \\ -I & 0 & 0 \\ 0 & I & 0 \end{pmatrix}
\begin{pmatrix} A \\ B \\ W \end{pmatrix}, \tag{4.5}
$$

where

$$
\begin{pmatrix} A \\ B \\ W \end{pmatrix} =
\begin{pmatrix}
A(X_1) \\ \vdots \\ A(X_{n-1}) \\ B(X_1) \\ \vdots \\ B(X_{n-1}) \\ W(X_1) \\ \vdots \\ W(X_{n-1})
\end{pmatrix}, \tag{4.6}
$$

And

$$
R_0 = \begin{pmatrix}
R_0(X_1) & 0 & \dots & \dots & 0 \\
0 & R_0(X_2) & 0 & \dots & 0 \\
\vdots & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & 0 \\
0 & \dots & \dots & 0 & R(X_{n-1})
\end{pmatrix}, \tag{4.7}
$$

$$V_0 = \begin{pmatrix} V_0(X_1) & 0 & \dots & \dots & 0 \\ 0 & V_0(X_2) & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & V(X_{n-1}) \end{pmatrix}, \tag{4.8}$$

We then solve this generalised eigenvalue problem to obtain the eigenvalues and eigenvectors. We note that since this generalised matrix eigenvalue problem (4.5) is singular, we suspect that the eigenvalues might be inaccurate due to this. We therefore rewrite (4.5) as:

$$\begin{pmatrix} 0 & \tilde{D}_N^2 + (V_0 + V) \\ \tilde{D}_N^2 + (V_0 + V) - 2R_0(\tilde{D}_N^2)^{-1}R_0 & 0 \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \mathrm{i}\lambda \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix}, \tag{4.9}$$

that is we are inverting or solving the equation for W in terms of A.

The difference between the calculated values for the eigenvalues turns out to be small so we can use (4.5) to obtain (A, B, W) straight away.

4.2 The perturbation about the ground state

We consider now the results obtained by the solving (4.5) about the ground state of the spherically symmetric S-N equations (1.1), which has no zeros. Recall the eigenvalue for the ground state in the non-dimensionalized unit is 0.082.

In figure 4.1 we plot eigenvalues obtained by solving (4.5), excluding the near zero eigenvalue which does not correspond to a perturbation. To compute these results, we used $N = 60$ and length of the interval $L = 150$. We also plot all the eigenvalues obtained from solving (4.8) in figure 4.2 (note that the scale is different) to see that up to these limits there are no eigenvalues other than imaginary ones. The eigenvalues obtained are presented in table 4.1.

± 0.00000011612065
$\pm 0 - 0.03412557804571\mathrm{i}$
$\pm 0 - 0.06030198252911\mathrm{i}$
$\pm 0 - 0.06882503249714\mathrm{i}$
$\pm 0 - 0.07310346395426\mathrm{i}$
$\pm 0 - 0.07654635649084\mathrm{i}$
$\pm 0 - 0.08100785899036\mathrm{i}$
$\pm 0 - 0.08665500294956\mathrm{i}$

Table 4.1: Eigenvalues of the perturbation about the ground state

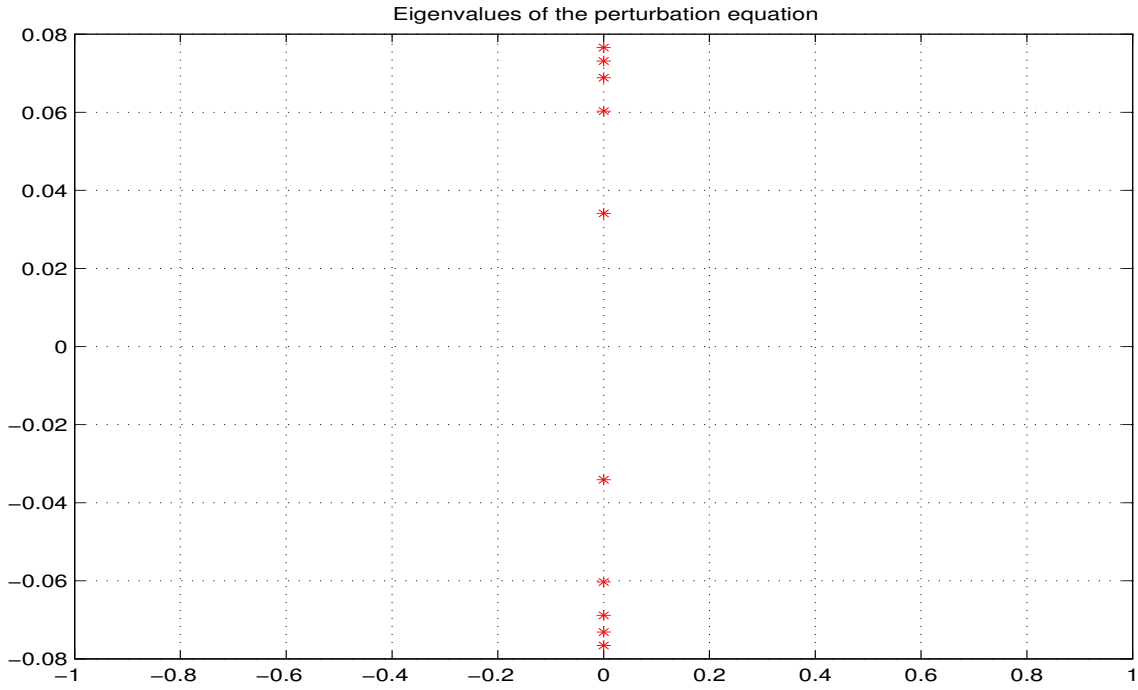


Figure 4.1: The smallest eigenvalues of the perturbation about the ground state

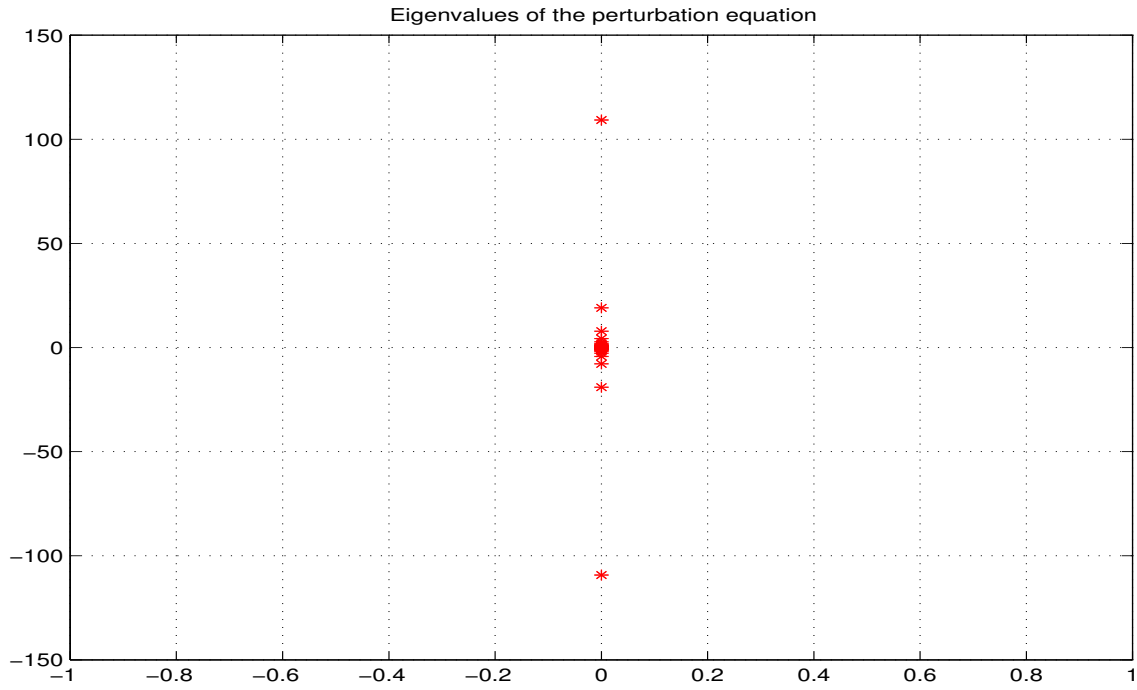


Figure 4.2: All the computed eigenvalues of the perturbation about the ground state

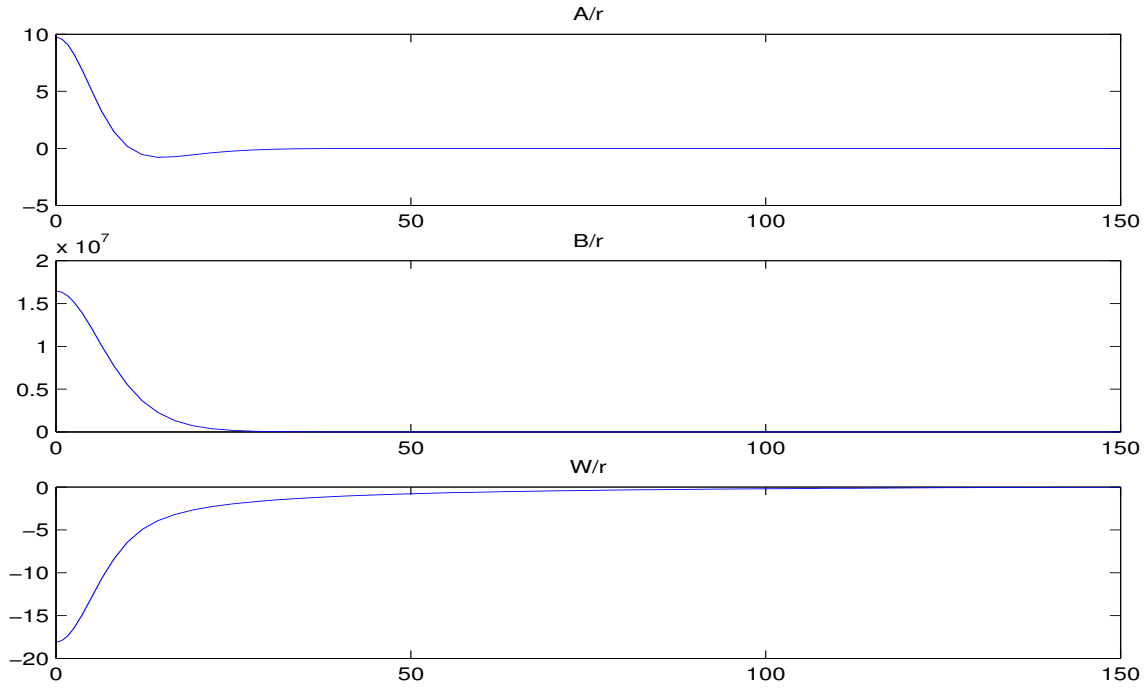


Figure 4.3: The first eigenvector of the perturbation about the ground state. Note the scales

We note that the near-zero eigenvalue obtained corresponds to the trivial zero-mode, as the eigenfunction shows. In figure 4.3 since A,W are very small compared to B, we can deduce that this solution corresponds to a phase rotation of the background solution, up to numerical error.

We note that all the eigenvalues are imaginary and that this agrees with the result obtained in section 3.4. We conclude that the ground state is linearly stable, that is the perturbations are only oscillatory.

We note that the eigenvalues are symmetric about the real axis, which was expected (section 3.2).

We plot the first three eigenvectors for the ground state as $\frac{A}{r}$,$\frac{B}{r}$ and $\frac{W}{r}$ in figures 4.3, 4.4, 4.5.

To test convergence of the eigenvalues we plot graphs against increase in N and increase in L. As an example we plot the graph of the eigenvalue 0.0765463562i as the value of N increases. We note that this shows the eigenvalue is converging with N see figure 4.6.

We can also plot the graph of the eigenvalue 0.0765463562i as a sample eigenvalue with increasing L instead of N, and again this shows the eigenvalue converging with increasing L see figure 4.7.

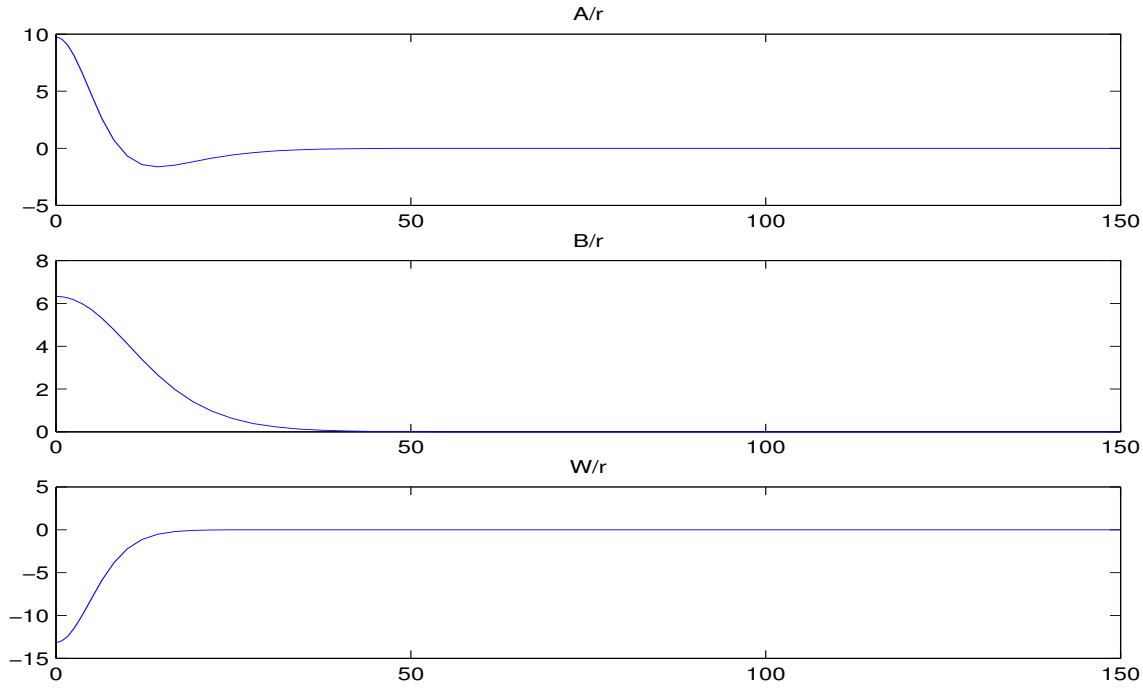


Figure 4.4: The second eigenvector of the perturbation about the ground state

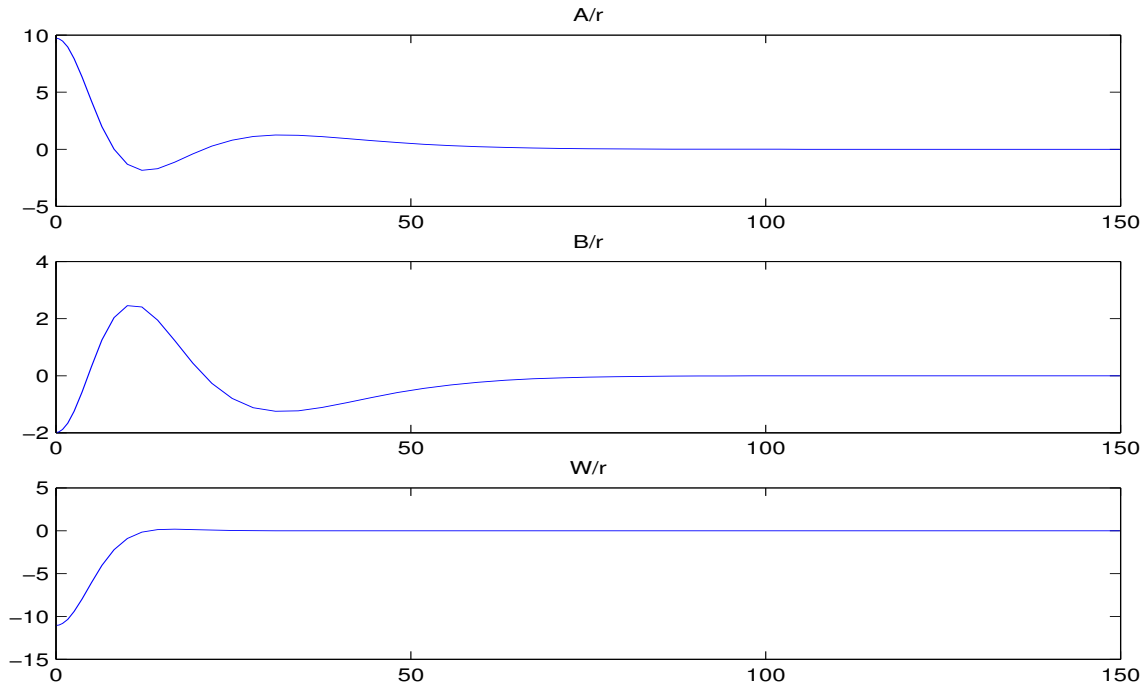


Figure 4.5: The third eigenvector of the perturbation about the ground state

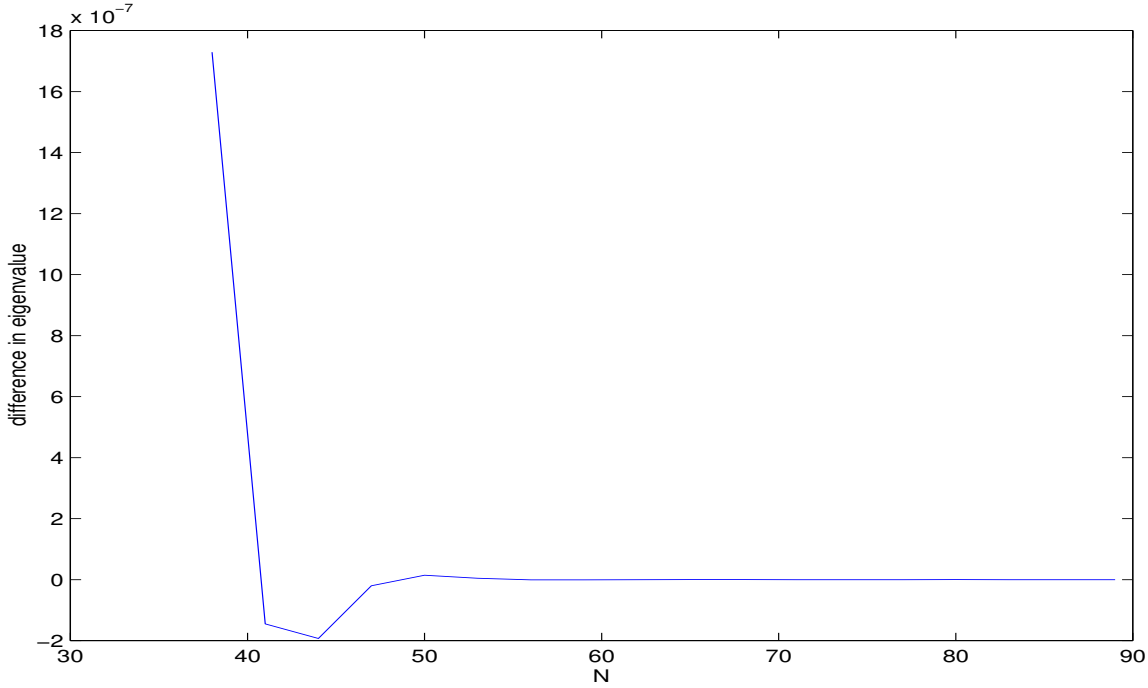


Figure 4.6: The change in the sample eigenvalue with increasing values of N ($L = 150$)

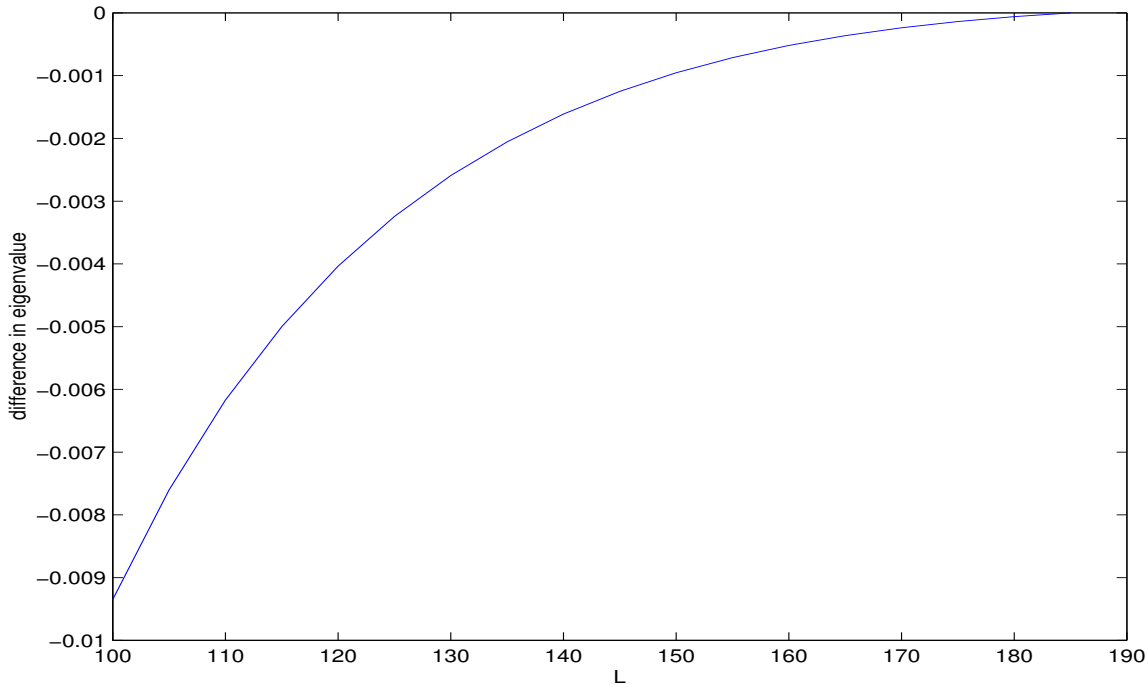


Figure 4.7: The change in the sample eigenvalue with increasing values of L ($N = 60$)

4.3 Perturbation about the second state

We now consider perturbation about the second state, when the unperturbed wave-function has one zero and the energy eigenvalue is 0.0308.

Using the same method we compute the numerical solutions of the perturbation about the second bound state. This time we obtain some eigenvalues with nonzero real parts.

We note that the eigenvalues have the symmetries expected from section 4.2, which we expected from the equations. In figure 4.8 we plot the lower eigenvalues for the case where $N = 60$ and $L = 150$. We also plot all the eigenvalues obtained in figure 4.9 on a different scale, to see that up to these limits there are no other complex ones. The lowest eigenvalues about the second state are presented in table 4.2.

± 0	$-0.000000036489 02\mathrm{i}$
± 0	$-0.003001491743 00\mathrm{i}$
± 0	$-0.008598596817 40\mathrm{i}$
$\pm 0.001393269309 81$	$-0.010040231793 00\mathrm{i}$
$\pm 0.001393269309 81$	$+0.010040231793 00\mathrm{i}$
± 0	$-0.015336750050 44\mathrm{i}$
± 0	$-0.021056908673 67\mathrm{i}$
± 0	$-0.027618455294 94\mathrm{i}$

Table 4.2: Eigenvalues of the perturbation about the second state

We have some complex eigenvalues so if our results are to satisfy the result in section 3.4 we require that $\int_0^\infty \bar{A}B dr = 0$. To see whether this is the case up to numerical error we compute

$$Q = \frac{|\int_0^L \bar{A}B dr|}{(\int_0^L |A|^2 dr)^{\frac{1}{2}} (\int_0^L |B|^2 dr)^{\frac{1}{2}}}, \tag{4.10}$$

If this much less than one then we know that $\int_0^\infty \bar{A}B dr = 0$. In the case where $L = 145$ and $N = 60$ we present in table 4.3 the calculated values of Q with the eigenvalues.

λ	Q
$0 - 0.000000036489 02\mathrm{i}$	0.221948256841 22
$0 - 0.003001491743 00\mathrm{i}$	0.234766460580 70
$0 - 0.008598596817 40\mathrm{i}$	0.368015057350 57
$-0.001393269309 81 - 0.010040231793 00\mathrm{i}$	$3.533821320897923e-14$
$0.001393269309 81 + 0.010040231793 00\mathrm{i}$	$3.919537745928816e-14$
$0 - 0.015336750050 44\mathrm{i}$	0.893726779146 25

Table 4.3: Q for different eigenvalues of the perturbation about the second state

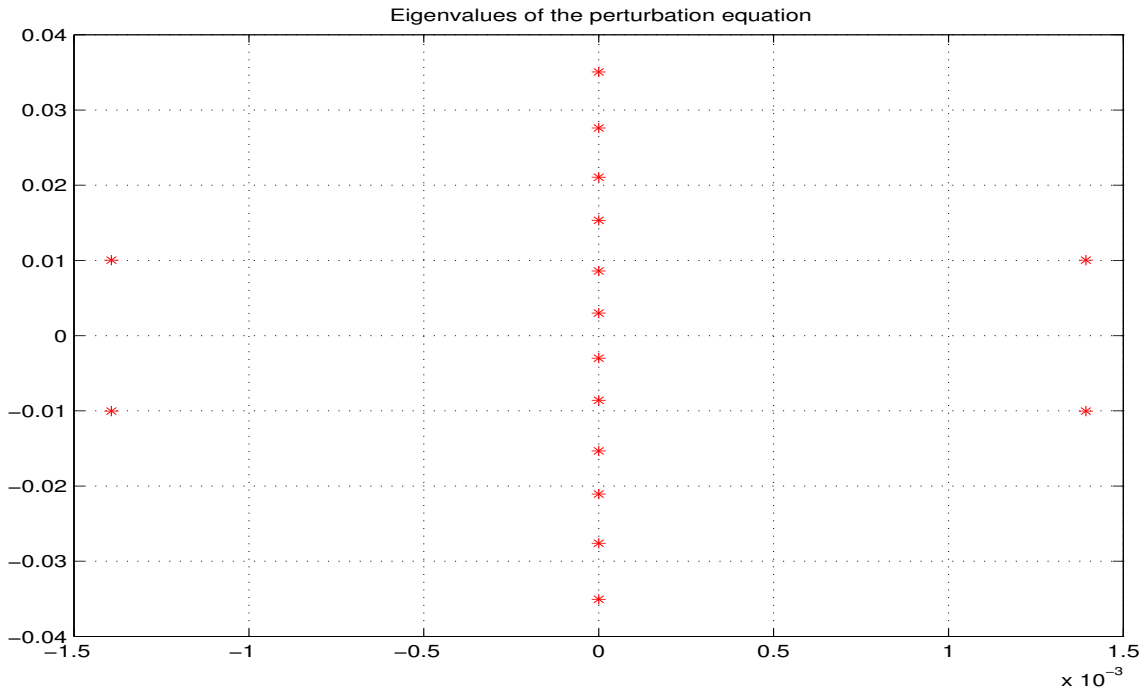


Figure 4.8: The lowest eigenvalues of the perturbation about the second bound state

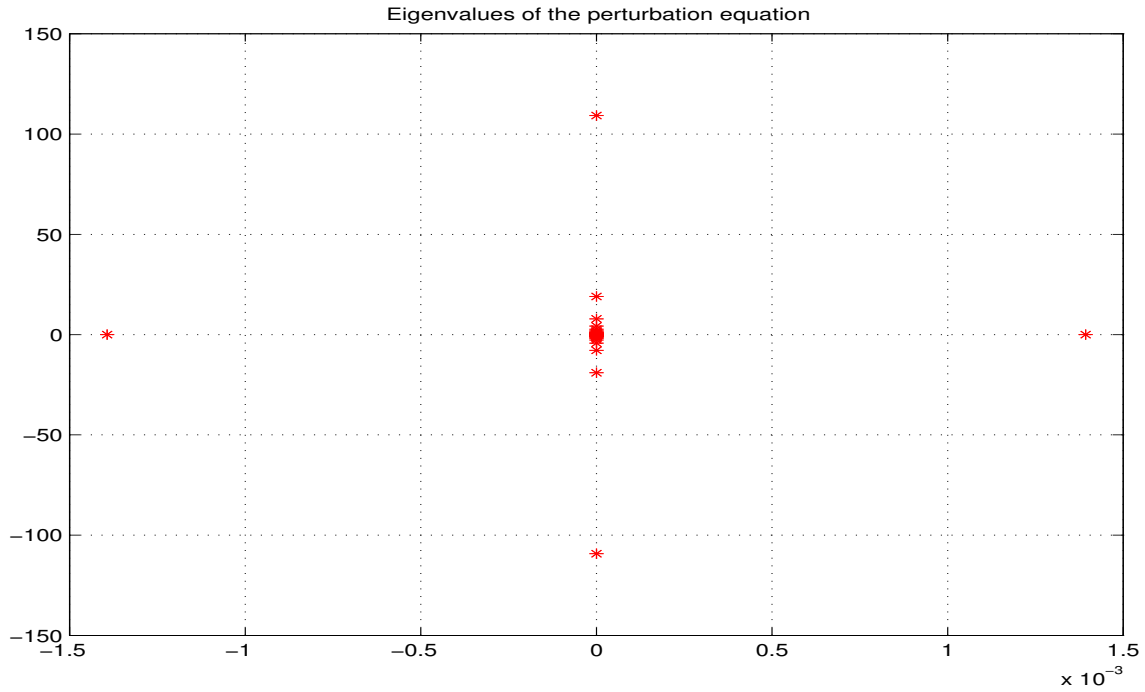


Figure 4.9: All the computed eigenvalues of the perturbation about the second bound state

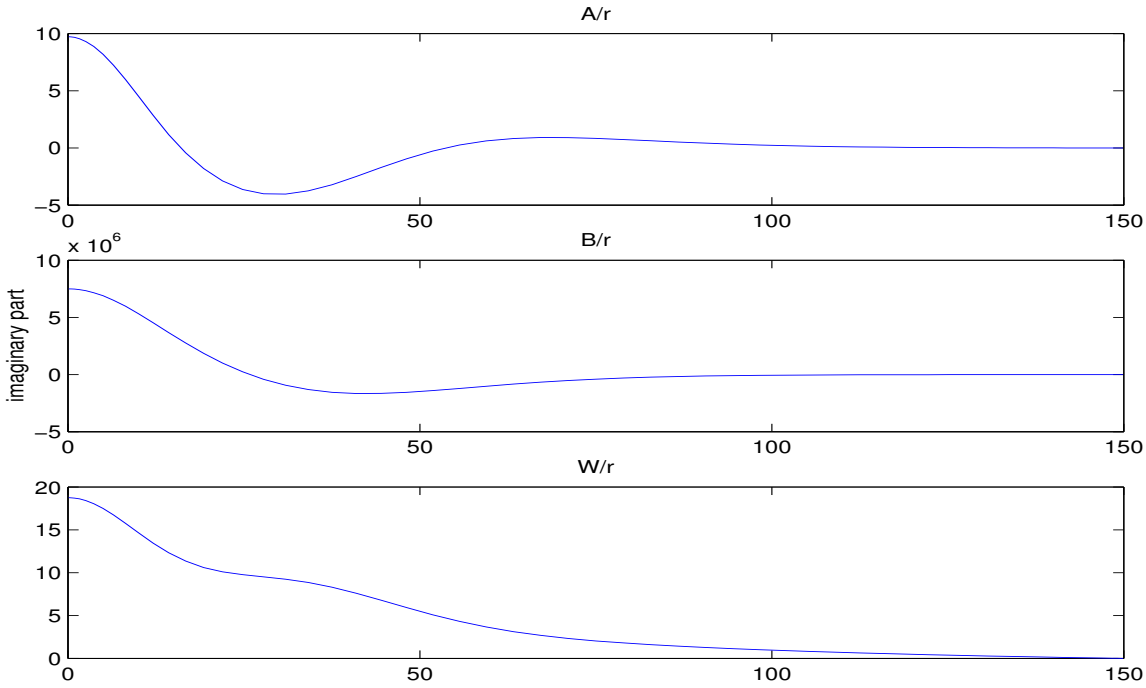


Figure 4.10: The first eigenvector of the perturbation about the second bound state, Note the scales

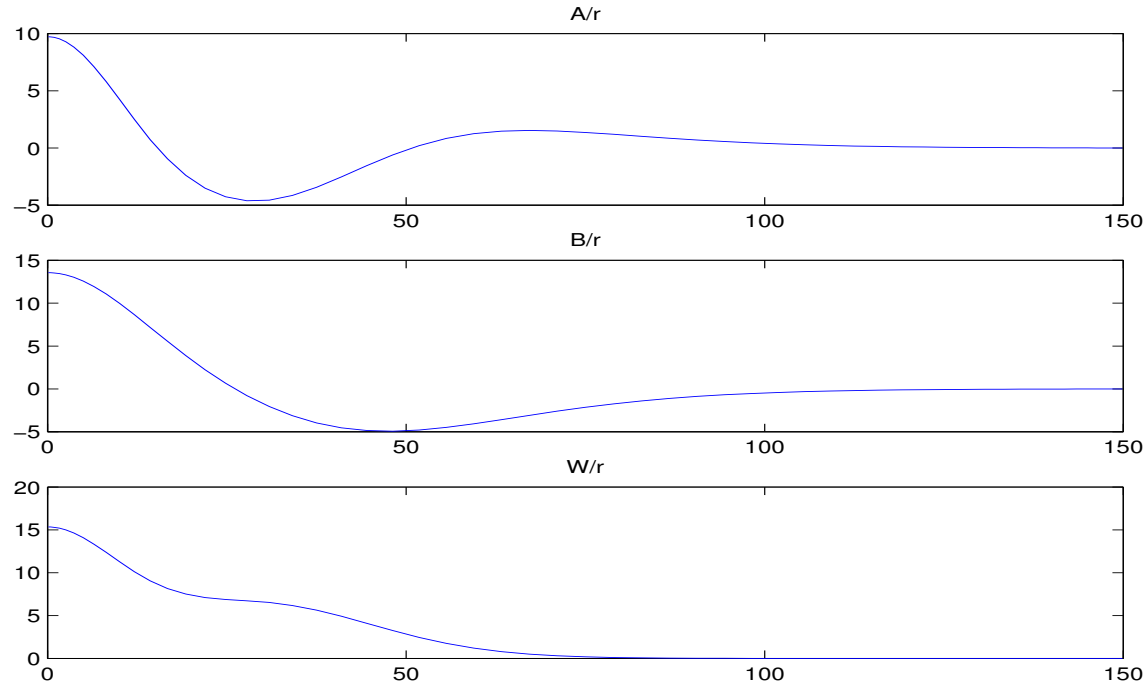


Figure 4.11: The second eigenvector of the perturbation about the second bound state

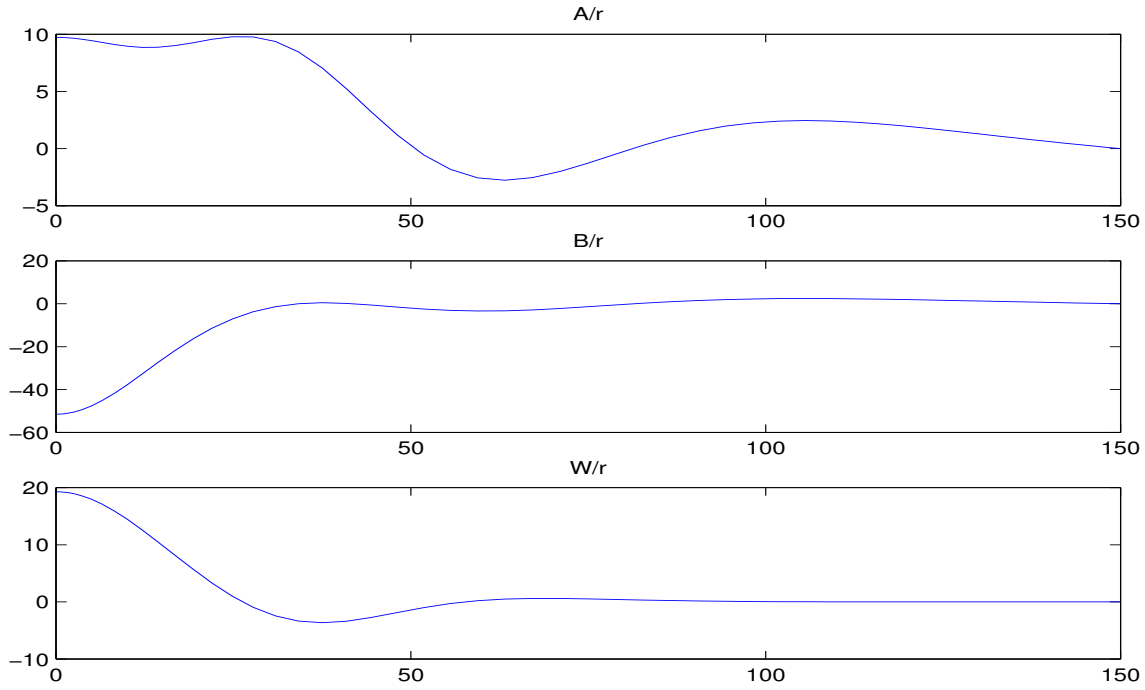


Figure 4.12: The third eigenvector of the perturbation about the second bound state

We note that for the eigenvalues with nonzero real part Q is almost zero, so the results of section 3.4 are confirmed. In figure 4.10 we plot the eigenvector corresponding to the near zero eigenvalue. Note that this is approximately $B = R_0$ hence this eigenvector corresponds to a phase rotation, i.e a trivial zero-mode.

In figures 4.11, 4.12 we plot the next two eigenfunctions corresponding to the next two eigenvalues.

4.4 Perturbation about the higher order states

We now consider the numerical method about the third state, with $N = 60$ and $L = 450$. In figure 4.13 we have the eigenvalues of the perturbation about the third state, the state with just two zeros in the unperturbed wave-function. The eigenvalues are presented in table 4.4.

Note again that the first near zero eigenvalue just corresponds to a phase rotation. In figure 4.14 we plot the eigenfunction corresponding to the next eigenvalue after the near zero one. We note that the result of section 3.4 can be confirmed.

In figure 4.15, 4.16 we plot the eigenfunction of a complex eigenvalue. In figure 4.15 the real part is plotted and in figure 4.16 the imaginary part.

Now for the fourth bound state with $L = 700$ and $N = 100$ we have eigenvalues which are presented in table 4.5.

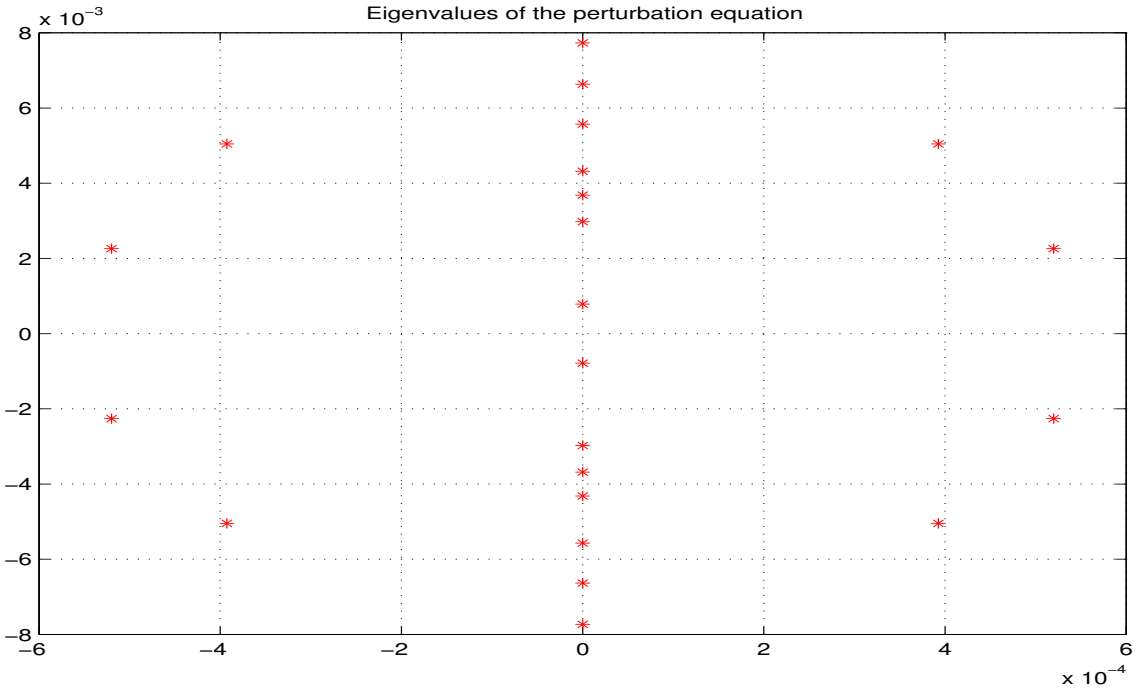


Figure 4.13: The eigenvalues of the perturbation about the third bound state

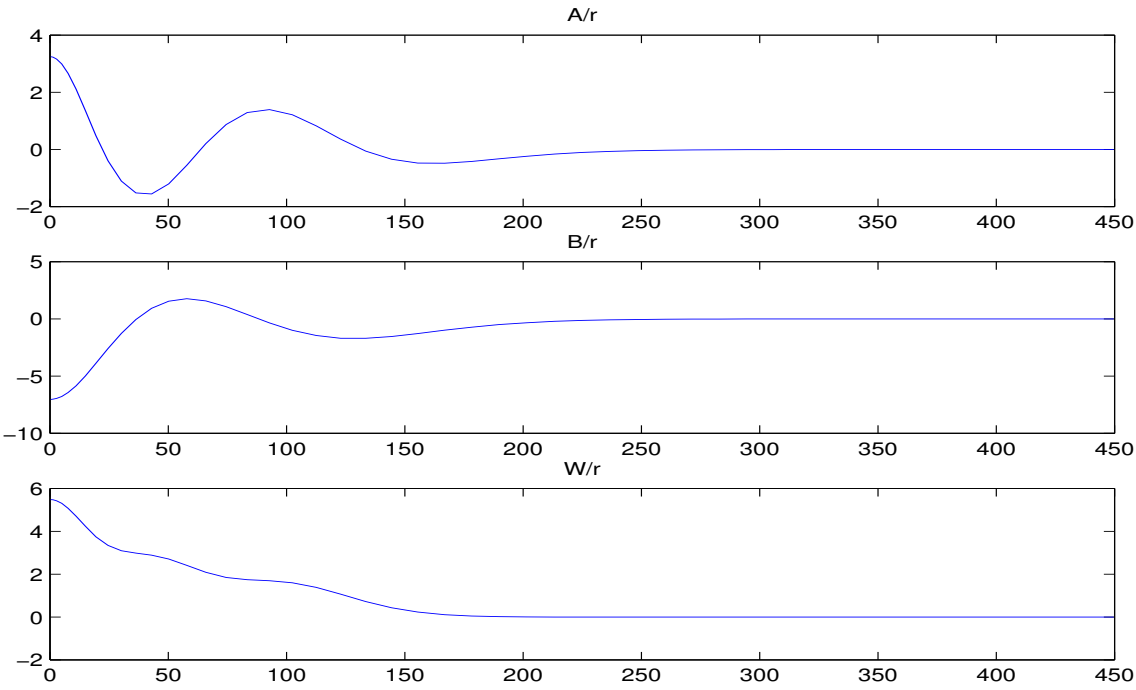


Figure 4.14: The second eigenvector of the perturbation about the third bound state

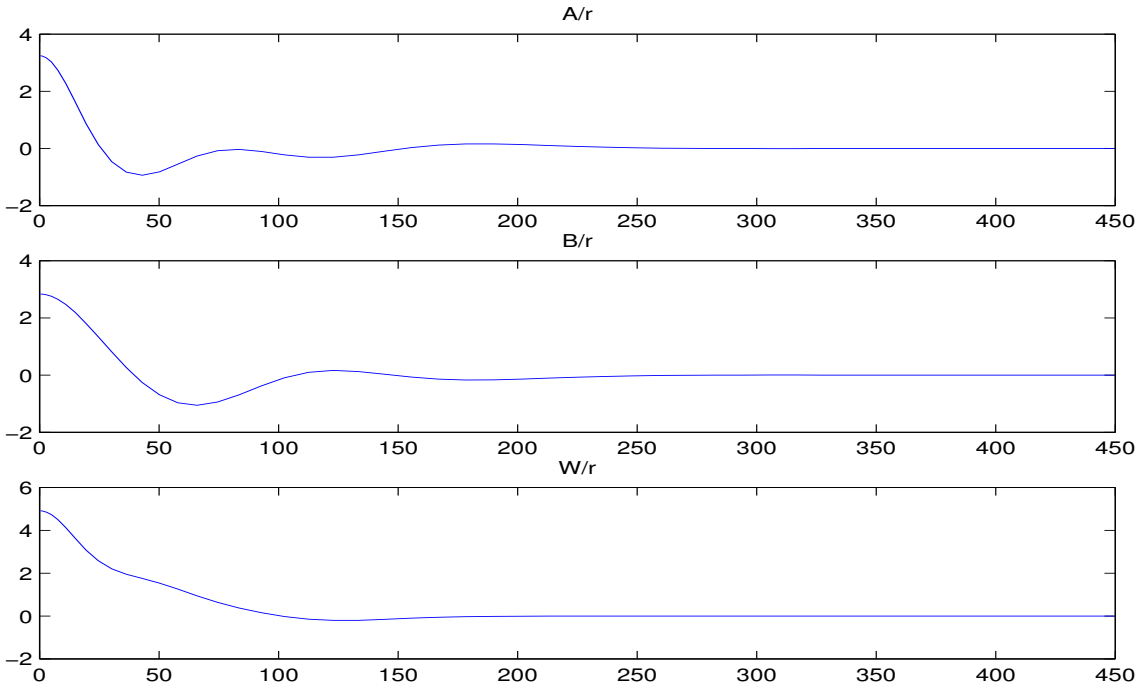


Figure 4.15: The third eigenvector of the perturbation about the third bound state

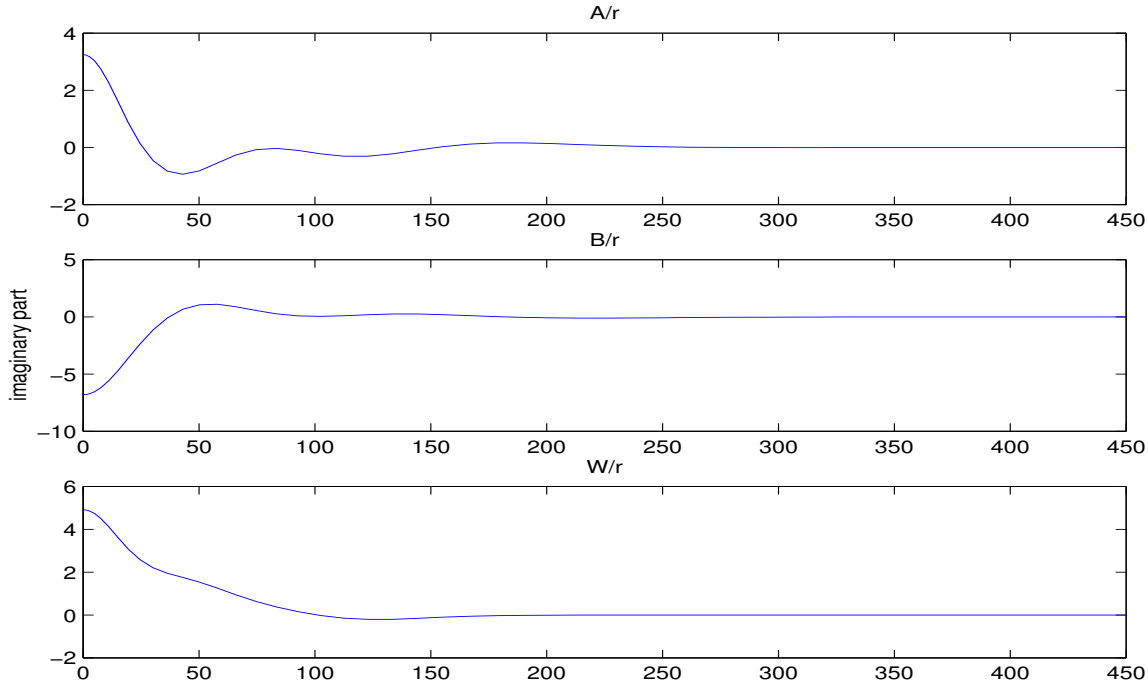


Figure 4.16: The third eigenvector of the perturbation about the third bound state

±0.00000001236655	
±0	−0.00078451579204i
±0.00051994798783	−0.00225911281180i
±0.00051994798783	+0.00225911281180i
±0	−0.00297504982576i
±0	−0.00368240791656i
±0	−0.00431823282039i
±0.00039265148039	−0.00504446459212i
±0.00039265148039	+0.00504446459212i
±0	−0.00557035273905i

Table 4.4: Eigenvalues of the perturbation about the third state

±0.00000000975044	
±0	−0.00031089278959i
±0.00022482647750	−0.00088225206795i
±0.00022482647750	+0.00088225206795i
±0	−0.00134992819716i
±0.00017460656919	−0.00168871696256i
±0.00017460656919	+0.00168871696256i
±0	−0.00185003885932i
±0	−0.00220985235805i
±0	−0.00260936445117i
±0.00015985257765	−0.00308656938411i
±0.00015985257765	+0.00308656938411i
±0	−0.00340215999936i

Table 4.5: Eigenvalues of perturbation about the fourth state

There we notice that there are three quadruples.

4.5 Bound on real part of the eigenvalues

From section 3.5 we have a bound on the real part of the eigenvalues of the perturbation equations. In table 4.6 we compare this bound with the values obtained, and see that it is satisfactorily confirmed.

4.6 Testing the numerical method by using Runge-Kutta integration

The result obtained via the Chebyshev numerical methods can be verified by using Runge-Kutta integration as in section 2.2.1, that is we convert (3.24) into a set of first order

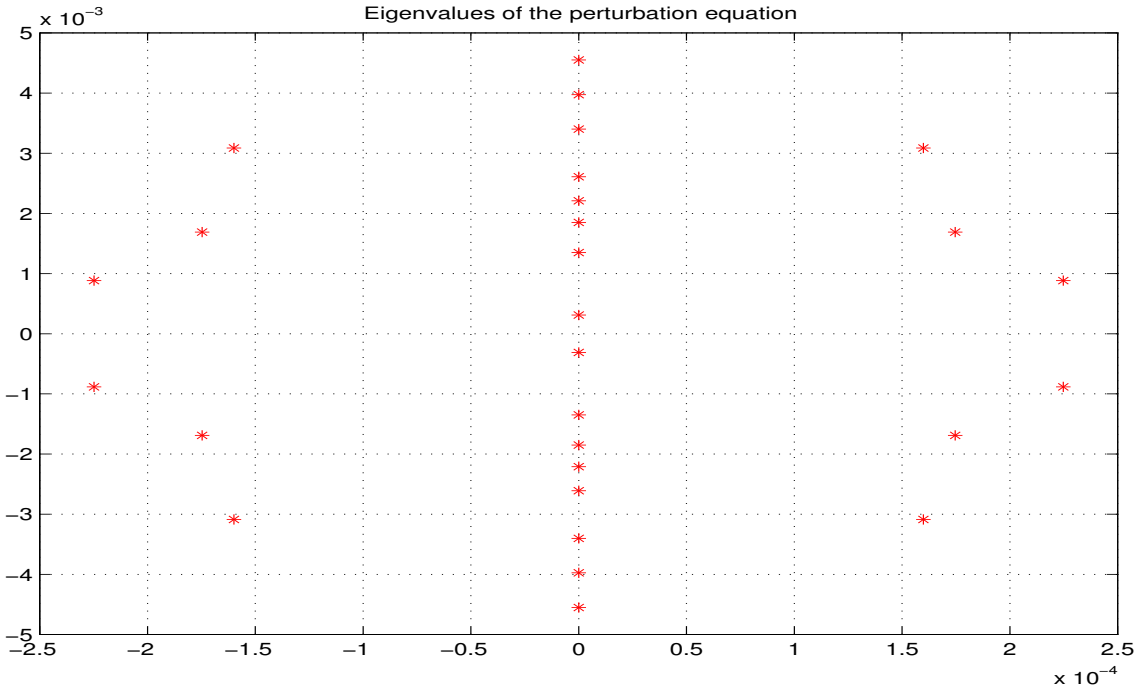


Figure 4.17: The eigenvalues of the perturbation about the fourth bound state

State	Numerical maximum real part	Bound
1	0	0.00362396540325
2	0.0013932693098	0.00157632209179
3	0.0005199479878	0.00100532361160
4	0.0002248264775	0.00073784267376
5	0.0001143754114	0.00058275894209
6	0.0000652878367	0.00048153805083

Table 4.6: Bound of the real part of the eigenvalues

O.D.E's,

$$Y_1' = Y_2, \tag{4.11a}$$
$$Y_2' = -V_0 Y_1 - i\lambda Y_1 + R_0 Y_5, \tag{4.11b}$$
$$Y_3' = Y_4, \tag{4.11c}$$
$$Y_4' = -V_0 Y_3 - i\lambda Y_3, \tag{4.11d}$$
$$Y_5' = Y_6, \tag{4.11e}$$
$$Y_6' = 2R_0 Y_3, \tag{4.11f}$$

where $Y_1 = A$, $Y_3 = B$, $Y_5 = W$ and λ is an eigenvalue. The boundary conditions at the origin are $Y_1(0) = 0$, $Y_3(0) = 0$ and $Y_5(0) = 0$. For the initial conditions on Y_6, Y_4 and

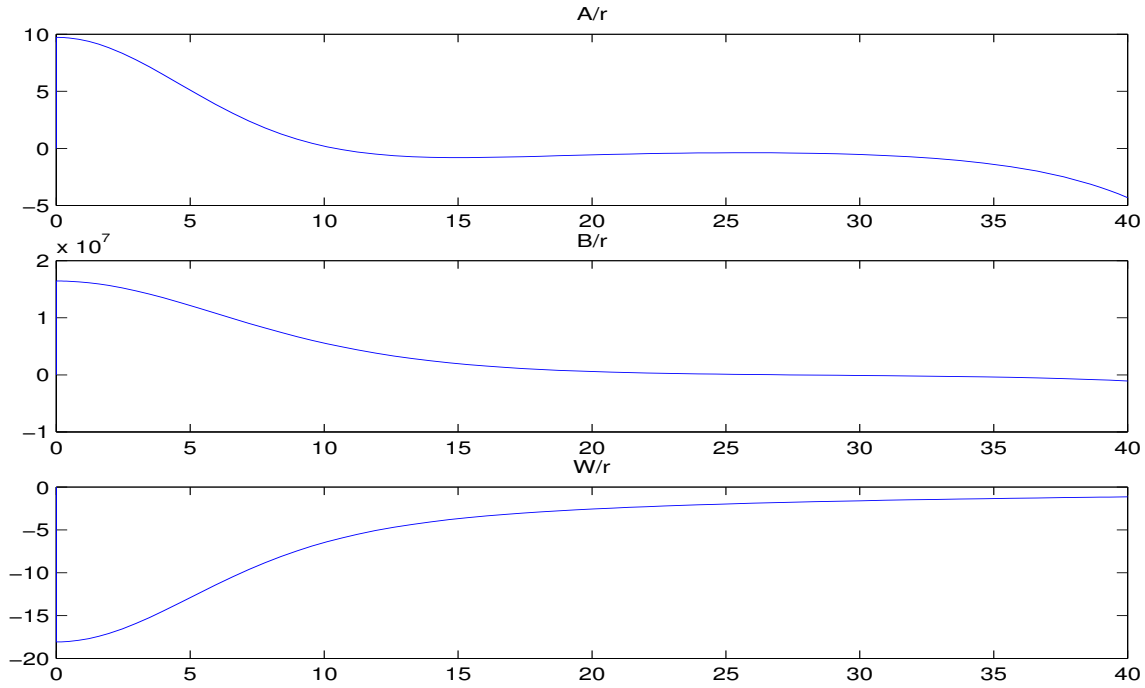


Figure 4.18: The first eigenvector of the perturbation about the ground state, using Runge-Kutta integration; compare with figure 4.3

Y_2 we can use the values obtained from the eigenvectors obtain when we solved the matrix eigenvalue problem.

In figures 4.18, 4.19 we plot the result doing a Runge-Kutta integration on (4.11) for the first two eigenvalues of (3.24) with R_0 and V_0 corresponding to the first bound state. The first two eigenvalues are those worked out in section 4.2. We note that except for the blowing up near the ends which we expect since the Runge-Kutta method is sensitive to inaccuracies in the initial data, the solutions obtain by Runge-Kutta integration correspond to the eigenvectors obtain by solving (4.5).

Proceeding to do Runge-Kutta integration on the results obtained with the perturbation about the second bound state in section 4.3, we plot the results obtained in figures 4.20, 4.21.

To test the perturbation about the third bound state, we proceed in the same way. The result plotted figure 4.22 show the first eigenvector of the perturbation about the third bound state. Figures 4.23, 4.24 show the second eigenvector with real and imaginary parts.

4.7 Conclusion

In this section, we have analysed the linear stability of the stationary spherically-symmetric states using a spectral method, and have checked the results with a Runge-Kutta method.

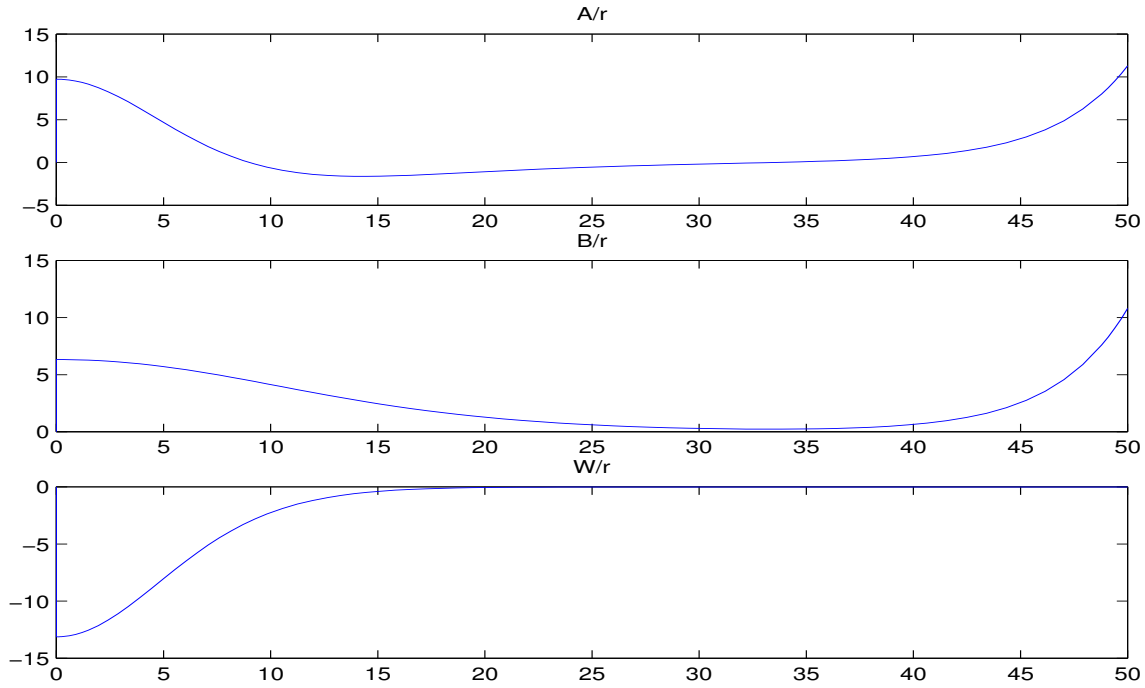


Figure 4.19: The second eigenvector of the perturbation about the ground state, using Runge-Kutta integration; compare with figure 4.4

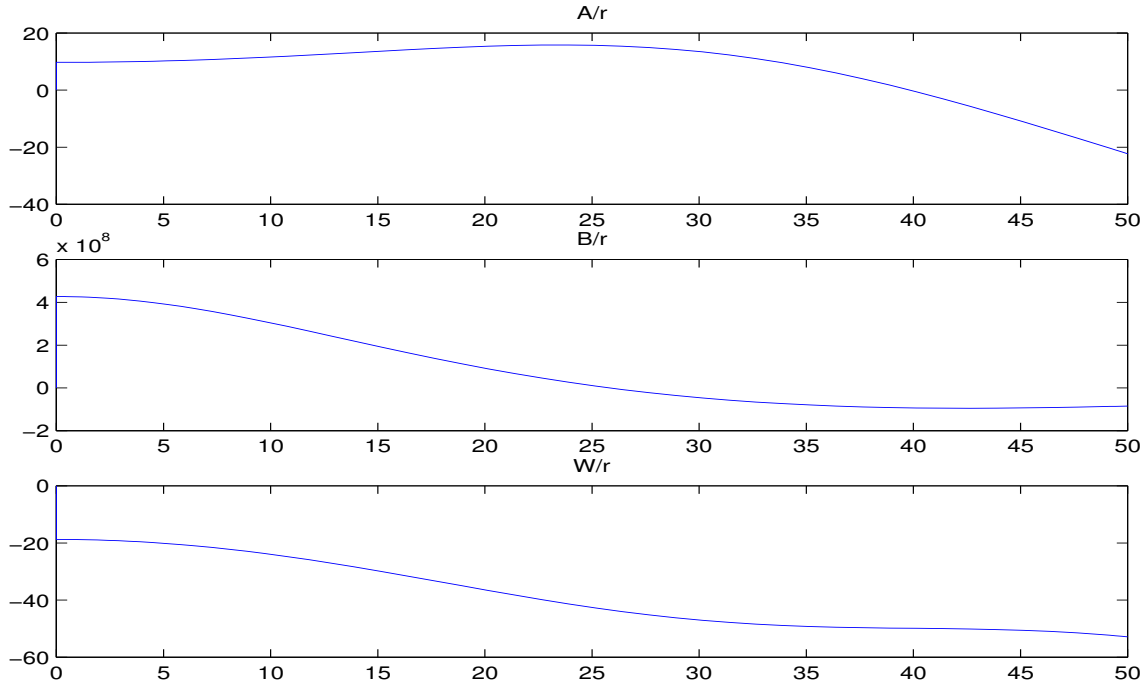


Figure 4.20: The first eigenvector of the perturbation about the second bound state, using Runge-Kutta integration; compare with figure 4.10

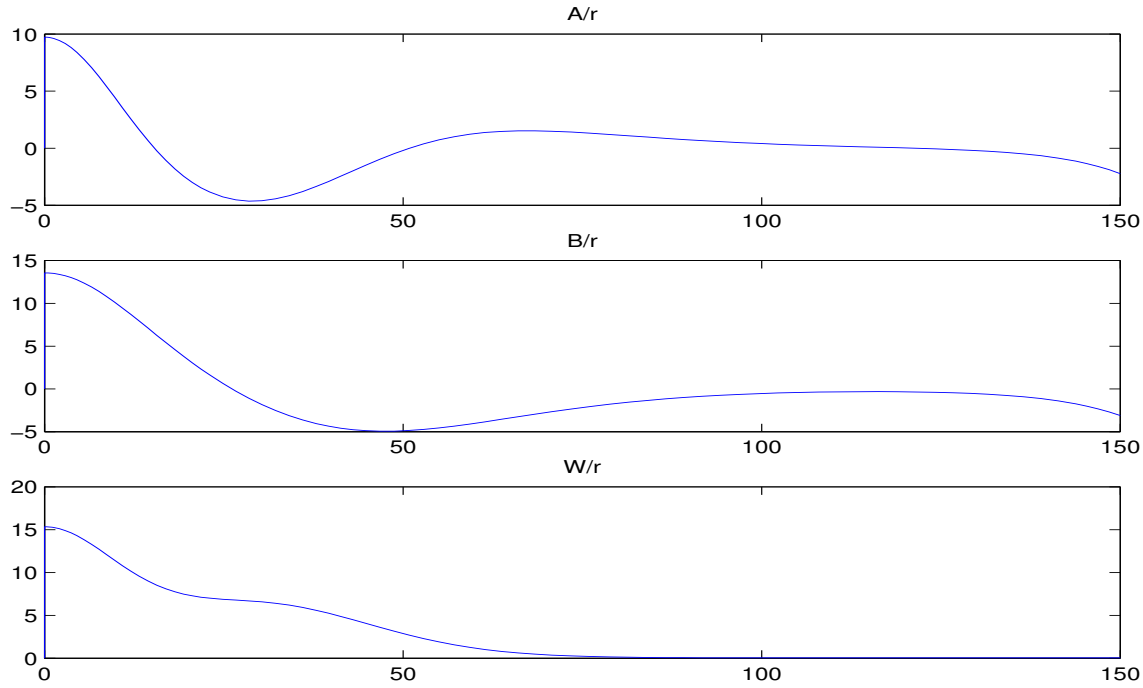


Figure 4.21: The second eigenvector of the perturbation about the second bound state, using Runge-Kutta integration; compare with figure 4.11

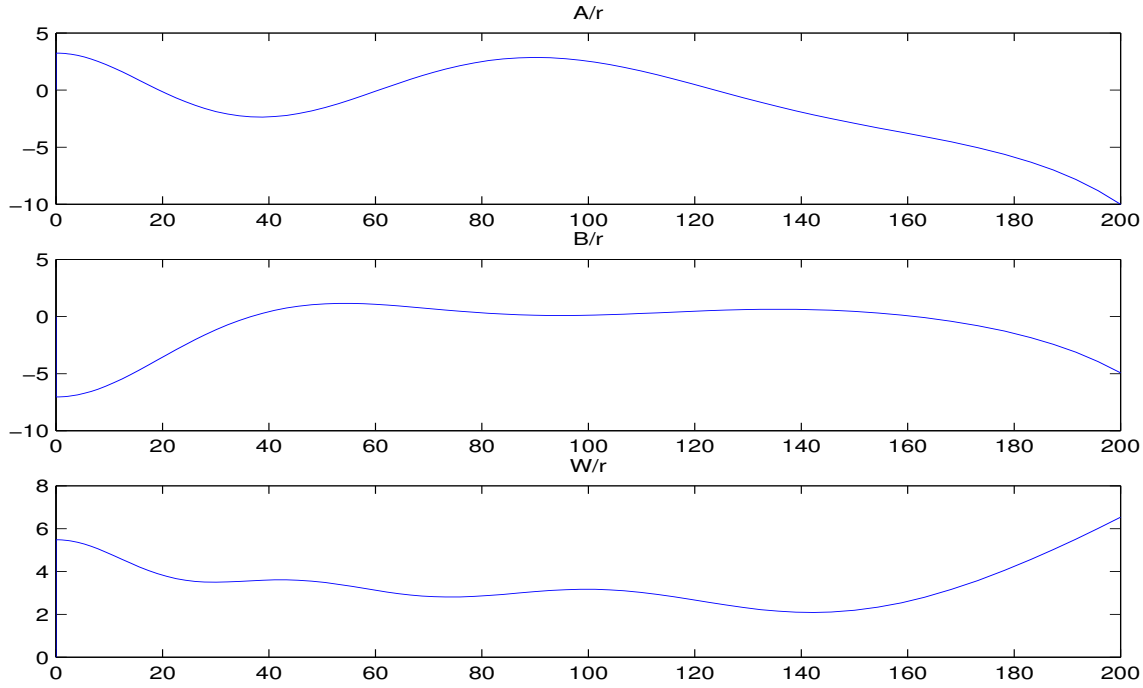


Figure 4.22: The second eigenvector of the perturbation about the third bound state, using Runge-Kutta integration; compare with figure 4.14

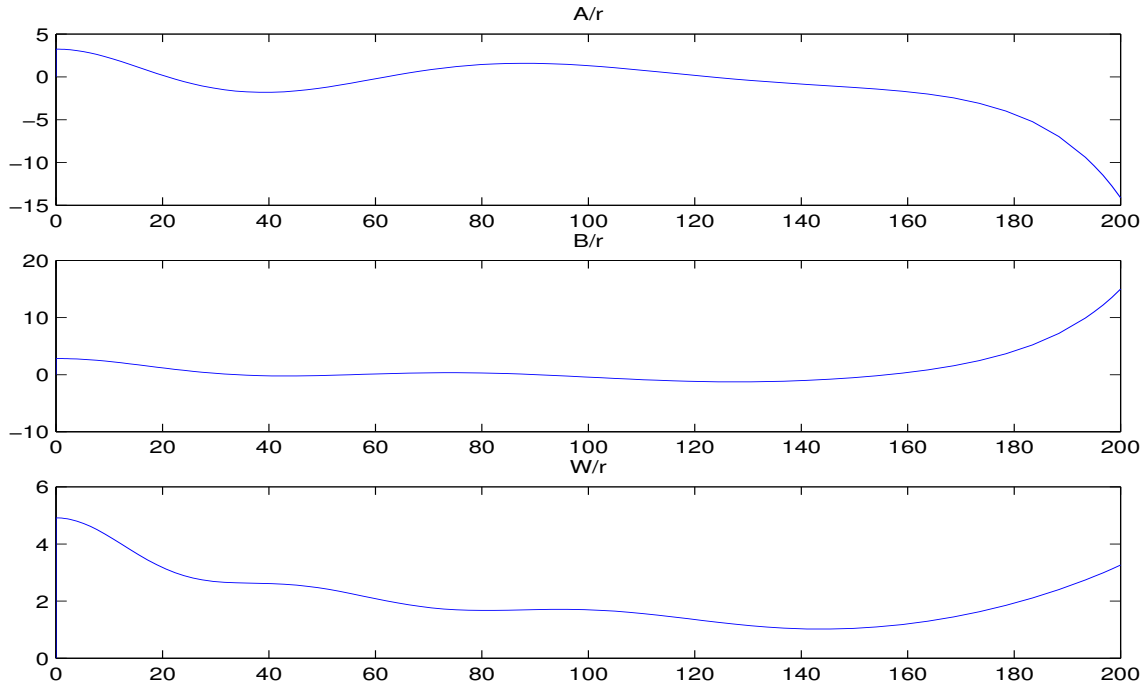


Figure 4.23: The real part of third eigenvector of the perturbation about the third bound state, using Runge-Kutta integration; compare figure 4.15

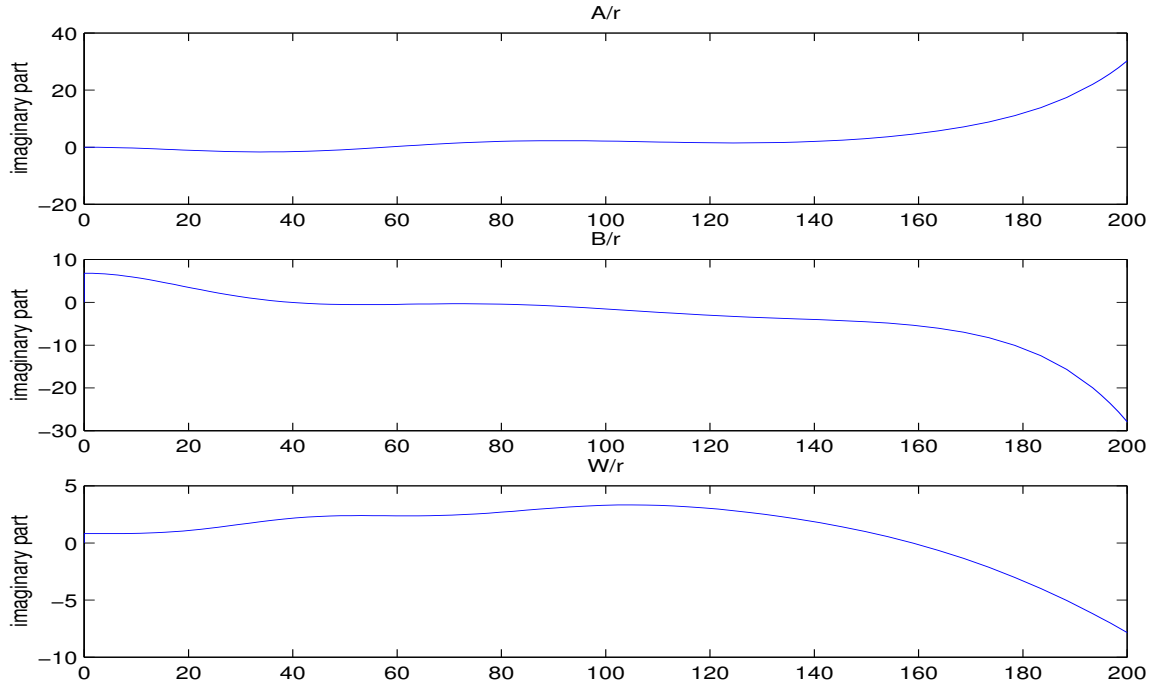


Figure 4.24: The imaginary part of third eigenvector of the perturbation about the third bound state, using Runge-Kutta integration; compare with figure 4.16

We have checked convergence of the eigenvalues obtained by the spectral method under increase of the number N of Chebyshev points and the radius L of the spherical grid. The calculation of section 4.3 showed that complex eigenvalues necessarily occur in quadruples (λ,$-\lambda$,$\bar{\lambda}$ and $-\bar{\lambda}$) while real or complex ones occur in pairs. In the numerical calculation, we find that:

- the ground state is linearly stable, in that the eigenvalues are purely imaginary;
- the n-th excited state, which is to say the $(n+1)$-th state, has n quadruples of complex eigenvalues as well as pure-imaginary pairs;
- no purely real eigenvalues appear, and in particular no zero-modes occur.

Thus only the ground state, which we saw in section 1.2 is the absolute minimiser of the conserved energy, is linearly stable. All other spherically-symmetric stationary states are linearly unstable.

In section 3.4, we found that complex eigenvalues could only arise if a certain integral of the perturbed wave-function was zero, and this is satisfactorily verified by numerical solutions found here.

We must now turn to the nonlinear stability of the ground state, which requires a numerical evolution of the nonlinear equations.

Chapter 5

Numerical methods for the evolution

5.1 The problem

In the next two chapters, the aim is to find and test a numerical technique for solving (1.5) with the restriction of spherical symmetry. We want to evolve an initial ψ long enough to see the dispersive effect of the Schrödinger equation and the concentrating effect of the gravity. In particular, we want to see how far the linearised analysis of chapter 3 and chapter 4 is an accurate picture.

In this chapter, we consider what the problems are in this programme, and methods for dealing with them. Chapter 6 contains the numerical results.

Since the problem is nonlinear, we first consider the simpler problem of solving the time-dependent Schrödinger equation in a fixed potential. We need a numerical method for evolution, and a technique for dealing with the boundary that will allow the wave function to escape from the grid, keeping reflection from the boundary to a minimum. This is the content of section 5.2 to section 5.4 below. In section 5.5, we consider how to test the method against an explicit solution of the zero-potential Schrödinger equation, and what we can check with a non-zero but fixed potential. In section 5.7, we face the problem of evolving the full S-N equations, which is to say evolving the potential as well as the wave-function. We describe an iteration for this and list some checks which we can make on the reliability and convergence of the method. Finally, in section 5.10 we introduce the notion of residual probability which we may describe as follows. On the basis of the linearised calculations of chapter 3 and chapter 4, we expect the ground state to be the only stable state. Thus we have a preliminary picture of the evolution: any initial condition will disperse to large distances (i.e. off the grid) leaving a remnant at the origin consisting of a ground state rescaled as in subsection 1.2.2 and with total probability less than one. If the

initial condition is a state with negative energy and this preliminary picture is sound then we can estimate the residual probability remaining in this rescaled ground state.

5.2 Numerical methods for the Schrödinger equation with arbitrary time independent potential

We consider numerical methods that solve the 1-dimensional Schrödinger equation with given initial data, that is solving,

$$\mathrm{i}\frac{\partial\psi}{\partial t} = -\frac{\partial^2\psi}{\partial x^2} + \psi\phi. \tag{5.1}$$

We note that Numerical Recipes on Fortran [20] and Goldberg et al [9] use a numerical method to solve the Schrödinger equation which they say needs to preserve the Hamiltonian structure of the equation or equivalently the numerical method is time reversible. To see why this is consider the case of an explicit method given by:

$$\mathrm{i}\frac{\psi^{n+1}-\psi^n}{\delta t} = D_2(\psi^n) + \phi\psi^n, \tag{5.2}$$

where ψ^n is the wave-function at the nth time step and D_2 is the approximation for the differentiation operator. Examples of what D_2 can be are finite difference or a Chebyshev differentation matrix. Now we consider what happens to the mode $\mathrm{e}^{\mathrm{i}kx}$, when

$$D_2(\mathrm{e}^{\mathrm{i}kx}) = -k^2\mathrm{e}^{\mathrm{i}kx}. \tag{5.3}$$

Letting $\psi^{n+1} = \lambda\mathrm{e}^{\mathrm{i}kx}$ and substituting into (5.2) we get

$$\lambda = 1 - \mathrm{i}\delta t(k^2+\phi). \tag{5.4}$$

So here we have an amplification factor of $|\lambda|$ which provided $k^2+\phi \neq 0$ has $|\lambda| > 1$ and this method leads to a growth in the normalisation of the mode. If we used an implicit method instead, that is:

$$\mathrm{i}\frac{\psi^{n+1}-\psi^n}{\delta t} = D_2(\psi^{n+1}) + \phi\psi^{n+1}. \tag{5.5}$$

We get an amplifying factor of

$$\lambda = \frac{1}{1+\mathrm{i}\delta t(k^2+\phi)}. \tag{5.6}$$

So in this case we note that $|\lambda| < 1$ provided $k^2+\phi \neq 0$, that is, this method leads to decaying modes.

We could consider renormalising the wave-function as the numerical method progresses, but this would have several draw backs. At each time step the modes get amplified by

different amounts so the ratio in different modes will change. Also for the S-N equation renormalisation would require rescaling the space axis which would need a interpolation at each step.

To get a numerical method that will preserve the constants of the evolution of the equation (namely the normalisation) we need to use a Crank-Nicholson method. With this method the discretization of the Schrödinger equation is:

$$\mathrm{i}\frac{\psi^{n+1}-\psi^n}{\delta t} = -\frac{1}{2}(D_2(\psi^{n+1}) + D_2(\psi^n)) + \frac{\phi}{2}(\psi^{n+1}+\psi^n), \tag{5.7}$$

which has an amplifying factor of

$$\lambda = \frac{1 - \frac{1}{2}\mathrm{i}\delta t(k^2+\phi)}{1 + \frac{1}{2}\mathrm{i}\delta t(k^2+\phi)}, \tag{5.8}$$

which is unitary ($|\lambda| = 1$) since it is in the form of a number over its complex conjugate.

Since the boundary conditions are straightforward a spectral method, that is Chebyshev differentation matrix, is seen to give better results for less computation time compared to finite difference method.

5.3 Conditions on the time and space steps

We note that the Crank-Nicholson method might be unitary, but that does not mean that the phase is correct. Suppose we consider the case of the linear Schrödinger equation in one dimension with $\phi = 0$ which has solutions of the form $\psi_k(x,t) = \mathrm{e}^{-\mathrm{i}k^2 t}\mathrm{e}^{\mathrm{i}kx}$. Now the wave evolves with a change in phase factor (λ) of:

$$\lambda = \frac{1 - \frac{1}{2}\mathrm{i}\delta t(k^2)}{1 + \frac{1}{2}\mathrm{i}\delta t(k^2)}, \tag{5.9}$$

which is just that of the factor obtained in (5.8). Expanding λ in powers of δt we have:

$$\begin{aligned} \lambda &= (1 - \frac{\mathrm{i}}{2}\delta t k^2 - \frac{1}{2}\delta t^2 k^4 + \frac{\mathrm{i}}{4}\delta t^3 k^6 + \ldots)(1 - \frac{\mathrm{i}}{2}\delta t k^2), \\ &= 1 - \mathrm{i}\delta k^2 t - \frac{1}{2}\delta t^2 k^4 - \frac{\mathrm{i}\delta t^3 k^6}{2} + \ldots. \end{aligned} \tag{5.10}$$

while the actual phase factor should be :

$$\mathrm{e}^{-\mathrm{i}k^2\delta t} = 1 - \mathrm{i}\delta t k^2 - \frac{\delta t^2 k^4}{2} + \frac{\mathrm{i}k^6\delta t^3}{6} + h.o.t \tag{5.11}$$

So to make the other terms small we can require $\frac{k^6\delta t^3}{12}$ to be small, as in [9]. The phase factor is correct but only to this order.

Now suppose that ψ is a stationary state solution of the Schrödinger equation with potential. We expect the solution to evolve like $\psi^n = \mathrm{e}^{(-\mathrm{i}Et)}\psi^0$ where E is the energy of the stationary solution. The discretized method in the Schrödinger equation is :

$$\mathrm{i}\frac{\psi^{n+1}+\psi^n}{\delta t} = \frac{1}{2}[-D_2(\psi^{n+1}) - D_2(\psi^n) + \phi(\psi^{n+1}+\psi^n)]. \tag{5.12}$$

Now since ψ^n and ψ^{n+1} are stationary states we know that,

$$-D_2(\psi^N) + \phi\psi^N = E\psi^N, \tag{5.13}$$

for any N. Substitution into the method gives,

$$\mathrm{i}(\psi^{n+1}-\psi^n) = \frac{\delta tE}{2}\psi^{n+1} + \frac{\delta tE}{2}\psi^n, \tag{5.14}$$

and so we obtain,

$$\psi_i^{n+1} = (1+\frac{\mathrm{i}\delta tE}{2})^{-1}(1-\frac{\mathrm{i}\delta tE}{2})\psi^n. \tag{5.15}$$

Expanding out with $|\frac{\delta tE}{2}| < 1$ we have,

$$\psi^{n+1} = [1-\delta tE + \frac{(-\mathrm{i}\delta tE)^2}{2!} + \frac{(-\mathrm{i}\delta tE)^3}{4} + O(\delta t^4)]\psi^n. \tag{5.16}$$

In order that the phase is calculated correctly we require that the error due to the term $\frac{\delta t^3 E^3}{12}$ be small.

5.4 Boundary conditions and sponges

The boundary conditions which we want to impose on the Schrödinger equation are that the wave-function must remain finite as r tends to zero and the wave-function must tend to zero as r tends to infinity, since otherwise the wave-function would not be normalisable. To try to impose these boundary conditions numerically we can solve the Schrödinger equation in $r\psi = \chi$ in which case it becomes:

$$\mathrm{i}\chi_t = -\chi_{rr} + \phi\chi. \tag{5.17}$$

The boundary condition at $r = 0$ is just a matter of setting $\chi = 0$ for the first point in the domain. The other boundary condition can be approximated in one of the following ways:

- We can at a given $r = R$ say set that $\chi = \frac{\psi}{r}$ where ψ is the actual analytic solution. The problem with this is that only in a certain few cases will we know the actual solution.

- We set the condition that $\chi = 0$ at $r = R$ for R large so that the other boundary condition is approximated. The problem with this boundary condition is that it will reflect all the outward going probability and send it back towards the origin.

- The other thing we can do to reduce the effect of probability bouncing back is to introduce sponge factors at $r = R$ were R is large. That is, we consider instead of the Schrödinger equation the equation,

$$(\mathrm{i} + e(r))\chi_t = -\chi_{rr} + \phi\chi, \tag{5.18}$$

  where the function $e(r)$ is a strictly negative (one sign) so that the equation acts like a heat equation to reduce the probability which is assumed to be heading off the grid. There will also be a reduction in the normalisation integral. For example we take

$$e(r) = \mathrm{e}^{(0.1(r-R))}, \tag{5.19}$$

  so as give a smooth function which only has an effect at the boundary since for small r the sponge factor effectively vanishes. There will still be some reflection off the sponge but this is a smaller effect.

5.5 Solution of the Schrödinger equation with zero potential

To check that our numerical method is functioning correctly we aim to check it in the case of spherical symmetry with the same boundary condition where a solution is known both analytically and numerically. This also gives a check on the boundary conditions.

In the spherically symmetric case with zero potential a solution to the Schrödinger equation is:

$$\begin{aligned} r\psi = \chi = \frac{C\sqrt{\sigma}}{(\sigma^2+2)^{\frac{1}{2}}}[\exp(-\frac{(r-vt-a)^2}{2(\sigma^2+2\mathrm{i}t)} + \frac{\mathrm{i}vr}{2} - \frac{\mathrm{i}v^2t}{4}) \\ - \exp(-\frac{(r+vt+a)^2}{2(\sigma^2+2\mathrm{i}t)} - \frac{\mathrm{i}vr}{2} - \frac{\mathrm{i}v^2t}{4})], \end{aligned} \tag{5.20}$$

we shall call this a moving Gaussian shell, when $t = 0$. Here we have the boundary condition that $\chi = 0$ at $r = 0$ and $\chi = 0$ at $r = \infty$. This is a wave bump "moving" at a velocity v starting at a distance a from the origin. C is chosen such that the wave-function is normalised. That is:

$$1 = C^2\sqrt{\pi}[1 - \exp(\frac{-a^2}{\sigma^2} - \frac{v^2\sigma^2}{4})]. \tag{5.21}$$

We note that C will not exist when $\exp(-\frac{a^2}{\sigma^2} - \frac{v^2\sigma^2}{4}) = 1$ which is when $v = 0$ and $a = 0$, but in this case we will have the limiting solution:

$$r\psi = \chi = \frac{Ar}{(\sigma^2 + 2\mathrm{i}t)^{\frac{3}{2}}} \exp(-\frac{r^2}{2(\sigma^2 + 2\mathrm{i}t)}), \tag{5.22}$$

where A is such that

$$A^2 = \frac{2\sigma^3}{\sqrt{\pi}}. \tag{5.23}$$

Note that the solution (5.20) will in the long term tend to zero everywhere, since there is no gravity or other forces to stop the natural dispersion of the wave equation.

We note that we cannot numerically model the boundary condition at $r = \infty$ so we can do the following things to check the numerical calculation of the solution, and also to approximate the effect of the boundary condition at infinity:

- Check that normalisation is preserved, which should be the case since the numerical method preserves this, when there are no sponge factors or when there is no probability moving off the grid.
- We can set the boundary value at $r = R$ to be the value of the analytical solution at that point. We note that this will only work for the case where the analytical solution is known; that is, it can only be done for testing.
- We can set the boundary condition to be $\chi = 0$ at $r = R$ where R is chosen so that it is large compared to the initial data. This method will cause the wave to reflect back off the boundary.
- We can use sponge factors, that is, change the equation so that is becomes a type of heat equation, where the heat is absorbed at the boundary. We note that this will reduce the wave which is reflected from the boundary, but that normalisation will not be preserved.

5.6 Schrödinger equation with a trial fixed potential

We can consider the Schrödinger equation with a potential which is constant with respect to time. In this case we know that the energy E is preserved where E is given by

$$E = T + V, \tag{5.24}$$

where

$$T = \int |\nabla\psi|^2 d^3x, \tag{5.25}$$

and

$$V = \int \phi |\psi|^2 d^3x, \tag{5.26}$$

with the choice of potential $\phi = \frac{1}{(r+1)}$ we can work out the bound states, that is we can solve the eigenvalue problem

$$\nabla^2 \psi + \phi\psi = E\psi \tag{5.27}$$

This gives us eigenvalues E_n say with eigenfunctions ψ_n where the ψ_n are independent of time. We know by general theory that

$$\int \psi_n \psi dr = c_n \mathrm{e}^{\mathrm{i}E_n t}. \tag{5.28}$$

where ψ is a general wave-function and c_n is a constant.

In summary we can check in this case,

- Normalisation.
- Energy of the wave-function.
- The inner products with respect to the bound state to see that they are of constant modulus and the phase change correctly

5.7 Numerical evolution of the S-N equations

The method that Bernstein et al [4] use for numerical evolution of the wave-function of the S-N equations consists of a Crank-Nicholson method for the time-dependent Schrödinger equation and an iterative procedure to obtain the potential at the next time step, the first guess being that of the current potential. They transform the S-N equations in the spherically symmetric case into a simpler form which is:

$$\mathrm{i}\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial r^2} + \phi u, \tag{5.29}$$

$$\frac{d^2(r\phi)}{dr^2} = \frac{|u|^2}{r}, \tag{5.30}$$

where $u = r\psi$.

Then the following procedure is used to calculate the wave-function and potential at the next time step:

1. Choose $\phi^{n+1} = \phi^n$ to be an initial guess for ϕ at the $(n+1)$th time step.

2. Calculate u^{n+1} from the discretized version of the Schrödinger equation, which is

$$2\mathrm{i}\frac{u^{n+1}-u^n}{\delta t} = -D_2(u^{n+1}) - D_2(u^n) + [\phi^{n+1}u^{n+1} + \phi^n u^n], \tag{5.31}$$

where this equation is solved on the region $r \in [0, R]$ and R is large, where there are no sponge factors. These could be added to stop outgoing waves being reflected from the imposed boundary condition at $r = R$.

3. Calculate the Φ which is the solution of the potential equation for the u^{n+1}. That is Φ that solves:

$$\frac{d^2(r\Phi)}{dr^2} = \frac{|u^{n+1}|^2}{r}. \tag{5.32}$$

4. Calculate U by the discretized version of the Schrödinger equation with Φ being the potential at the $(n+1)$ step, that is:

$$2\mathrm{i}\frac{U-u^n}{\delta t} = -D_2(U) - D_2(u^n) + [\Phi U + \phi^n u^n], \tag{5.33}$$

where the boundary conditions are the same as in step 2.

5. Consider $|U - u^{n+1}|$ if less than a certain tolerance stop else take $\phi^{n+1} = \Phi$ and continue with step 2.

This outline of a general method is independent of the Poisson solver and numerical method used to evolve the Schrödinger equation. That is, we could use either a finite difference method or spectral methods. We note that Crank-Nicholson is second order accurate with respect to time and is such that it preserves the normalisation. This is the reason why we need to work out the potential at the $n+1$ time step instead of just using an implicit method in the potential.

5.8 Checks on the evolution of S-N equations

We recall that as we saw in section 1.2 for the S-N equations the energy is no longer conserved, but the action $I = T + \frac{1}{2}V$ is. We also note that if ψ corresponds to a bound state with energy eigenvalue E_n then

$$I = \frac{1}{3}E_n. \tag{5.34}$$

Thus it is useful to call $3I$ the conserved energy E_I. In the S-N equation in general we can check:

- Preservation of the normalisation, at least when it is not affected by the boundary condition due to the sponge factors.
- Preservation of the action $I = T + \frac{1}{2}V$, at least when not affected by the sponge.
- That the change in the potential function satisfies $\phi_t = \mathrm{i}\int_0^r (\bar{\psi}\psi_r - \psi\bar{\psi}_r)dr$,at least in the case of spherical symmetry.

In the case of evolving about a stationary state we can check

- That the phase factor should evolve at a constant rate at the corresponding energy of the bound state at least for a while.

5.9 Mesh dependence of the methods

To be sure that the evolution method produces a valid solution up to a degree of numerical error, we need to show that as the mesh and time steps decrease that the method converges to a solution. In the case of the finite difference method we expect errors to be second order in time and space steps. We expect this since the truncation error in the case of just solving the linear Schrödinger equation is second order. We also know that the method is unconditionally stable, that is there is no requirement on the ratio $\dfrac{\Delta t}{\Delta x^2}$, so we can refine the mesh in both direction independently and expect convergence. To test for this convergence we can use "Richardson Extrapolation". In a case where we expect the error to be like

$$O_h \sim I + Ah^p, \tag{5.35}$$

where O_h is the calculated value with space or time step size h, I actual value and p the order of the error. To get just a function of p we can calculate:

$$\frac{O_{h_1} - O_{h_3}}{O_{h_2} - O_{h_3}}, \tag{5.36}$$

where h_1,h_2 and h_3 are three different values of the step size. We note in the case of the spectral methods we do not expect the method to be second order in space, but we do still expect it to be second order in time.

5.10 Large time behaviour of solutions

For the Schrödinger equation with a fixed potential, the general solution will consist of a linear combination of bound states and a scattering state. For large times, the scattering state will disperse leaving the bound states. For the S-N equations, we have seen in chapter 4

that the bound states apart from the ground state are all linearly unstable, and we shall see the same thing for nonlinear instability in chapter 6. Consequently, we might conjecture that the solution at large values of time for the S-N equations with arbitrary initial data would be a combination of a multiple of the ground state, rescaled as in subsection 1.2.2 to have total probability less than one, together with a scattering state containing the rest of the probability but which disperses. We shall see support for this conjecture in chapter 6.

Assuming the truth of this picture, if the conserved energy is negative initially we can obtain a bound for the probability remaining in the multiple of the ground state.

What we mean by a multiple of the ground state is the following: if $\psi_0(r)$ is the ground state then consider $\psi_{0\alpha}(r) = \alpha^2\psi_0(\frac{r}{\alpha})$. It follows from the transformation rule for the S-N equations (subsection 1.2.2), that this is a stationary state but with:

$$\int_0^\infty |\psi_{0\alpha}|^2 = \alpha, \tag{5.37}$$

We also note that

$$\mathcal{E}_{0\alpha} = \alpha^3\mathcal{E}_0. \tag{5.38}$$

So if we start with an initial value $\mathcal{E}_I$ of the action which is negative then we claim

$$\mathcal{E}_I = \mathcal{E}_S + \alpha^3\mathcal{E}_0, \tag{5.39}$$

for some α, where $\mathcal{E}_S$ is the scattered action and $\alpha^3\mathcal{E}_0$ is the energy due to the remaining probability near the origin. So then we have that

$$\alpha^3 = \frac{(\mathcal{E}_S + |\mathcal{E}_I|)}{|\mathcal{E}_0|}, \tag{5.40}$$

since the scattered energy is positive, which leads to the inequality

$$\alpha^3 > \frac{\mathcal{E}_I}{\mathcal{E}_0}. \tag{5.41}$$

Chapter 6

Results from the numerical evolution

6.1 Testing the sponges

To test the ability of the sponge factor to absorb an outward going wave, we use the initial condition of an outward going wave from (5.5). We can use the exponential function of (5.20) at $t = 0$,

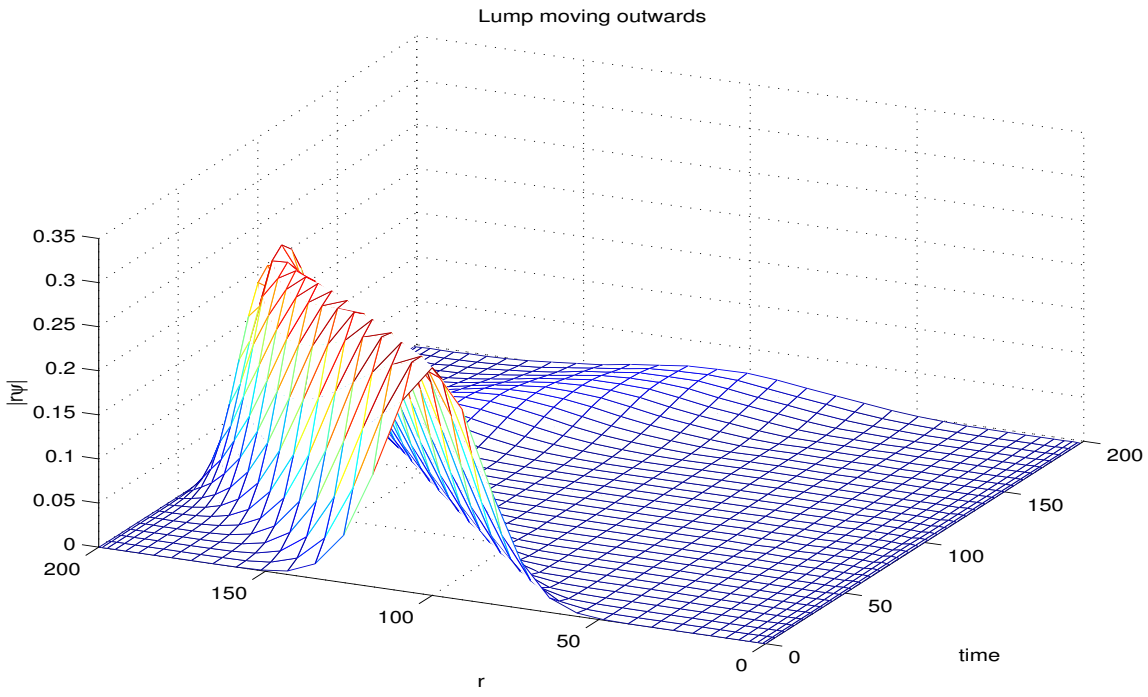


Figure 6.1: Testing the sponge factor with wave moving off the grid

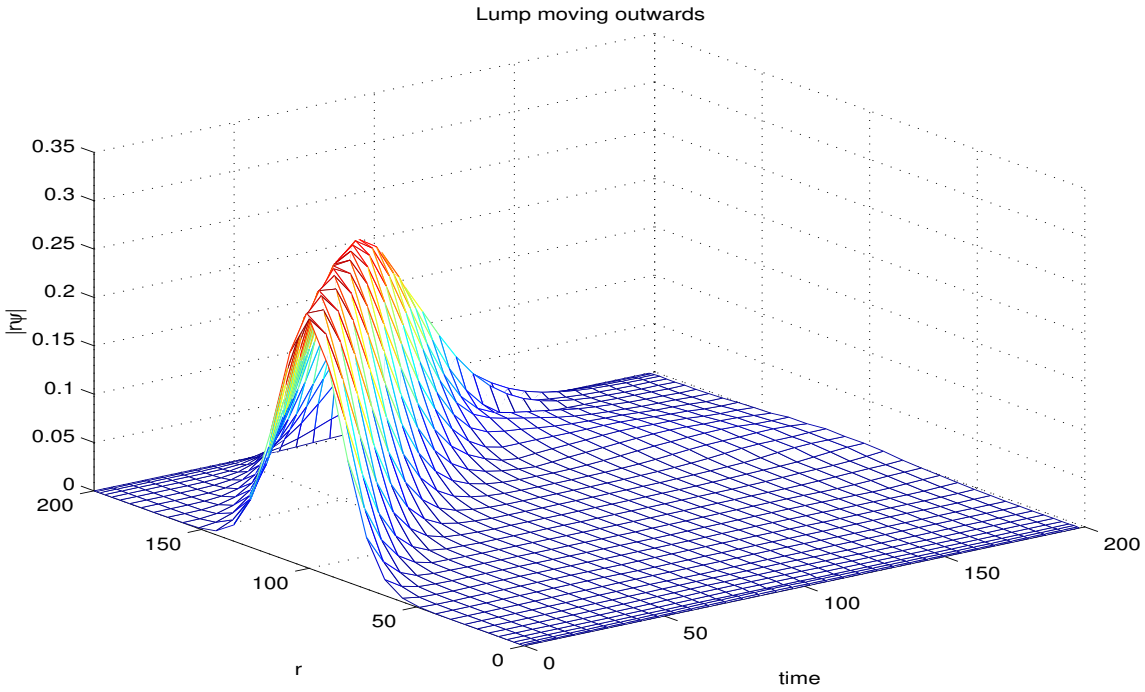


Figure 6.2: Testing sponge factor with wave moving off the the grid

$$r\psi = \frac{C\sqrt{\sigma}}{(\sigma^2+2it)^{\frac{1}{2}}}[\exp(-\frac{(r-vt-a)^2}{2(\sigma^2+2it)}+\frac{ivr}{2}-\frac{iv^2t}{4}) - \exp(-\frac{(r+vt+a)^2}{2(\sigma^2+2it)}-\frac{ivr}{2}-\frac{iv^2t}{4})] \tag{6.1}$$

Here v is chosen so that this corresponds to an outward going wave and is sufficiently large so that the wave-function will scatter, C is the normalisation factor and a and σ determine the position and shape of the distribution.

In figure 6.1 we plot a Gaussian shell moving off the grid, here the sponge factor is $-10\exp(0.5(r-R_{end}))$. This sponge factor gives a gradual increase toward the boundary, but we notice that there is some reflection off the boundary.

In figure 6.2 we plot an Gaussian shell moving off the grid, here the sponge factor is $-2\exp(0.2*(r-R_{end}))$. We note that this sponge factor work betters than one used for figure 6.1 since it reflects less back from the boundary.

6.2 Evolution of the ground state

The ground state is a stationary state and is linearly stable by chapter 4 and we expect it to be nonlinearly stable for small perturbations, since it has the lowest energy. We note that due to inaccuracy in the calculation of the ground state and the numerical error that will be

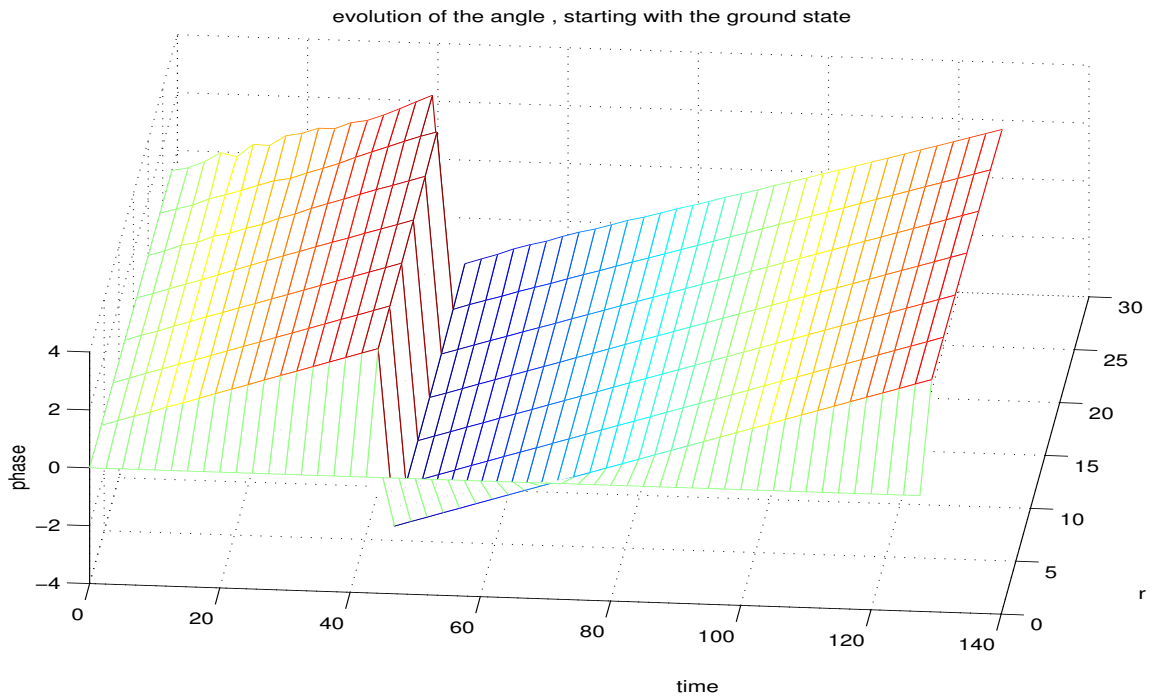


Figure 6.3: The graph of the phase angle of the ground state as it evolves

made in the evolution, the numerical evolution will actually be the evolution of the ground state with a small perturbation, but this shouldn't make any difference to the overall end result here.

Since the ground state is a solution to the time-independent S-N equations we expect that there should be a constant change in phase corresponding to the energy of the state and the absolute value of the wave-function should be constant.

Here we do get a constant phase (see figure 6.3) and we can check that it is of the correct value by calculating the energy of the wave-function.

We can also check the conservation of energy and the conservation of normalisation.

The next thing to try is the long time evolution of the ground state. Now we evolve the ground state for a long time, about 1E5 in the time scale which is about 5E5 time steps. We find a growing oscillation which reaches a bound and stabilises at a fixed oscillation, see figures 6.4, 6.5.

By adjusting the time step, we can reduce the amplitude of the limiting oscillation. We also can adjust the sponge factor to absorb more at the boundary to reduce the amplitude of the boundary. So therefore we know that this oscillation about the ground state is introduced by numerical error. We also deduce that the errors occur due to the long tail of the ground state at which the wave-function is approximately zero and small error here will have a great effect on the normalisation and the energy so causing more error close to

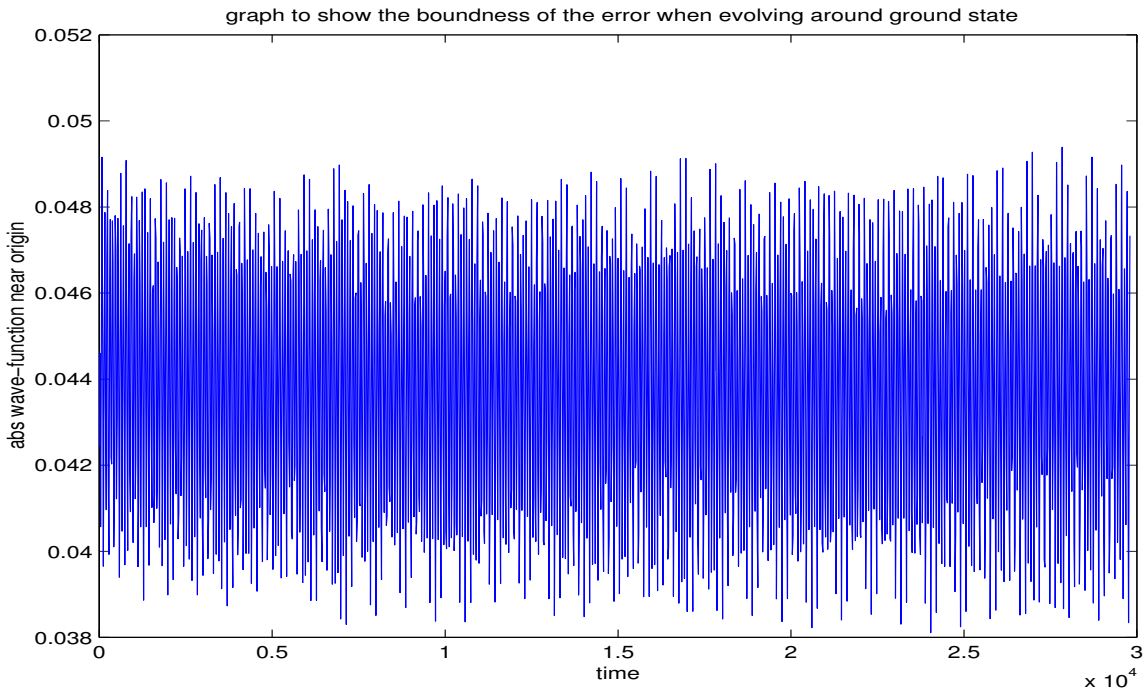


Figure 6.4: The oscillation about the ground state

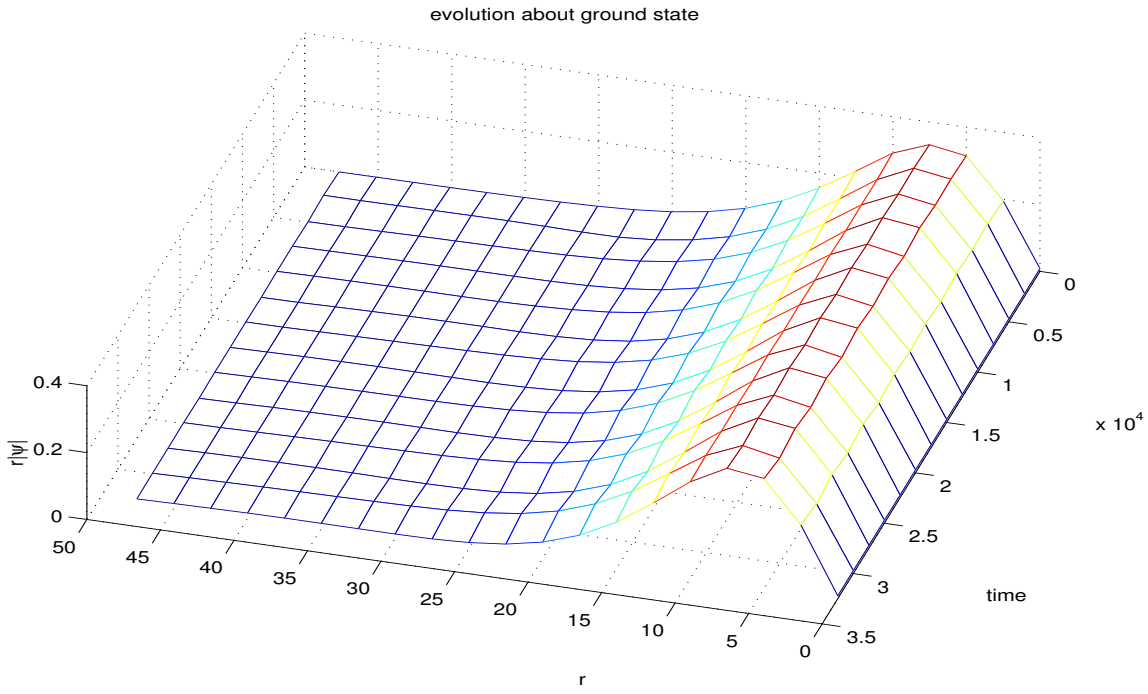


Figure 6.5: Evolution of the ground state

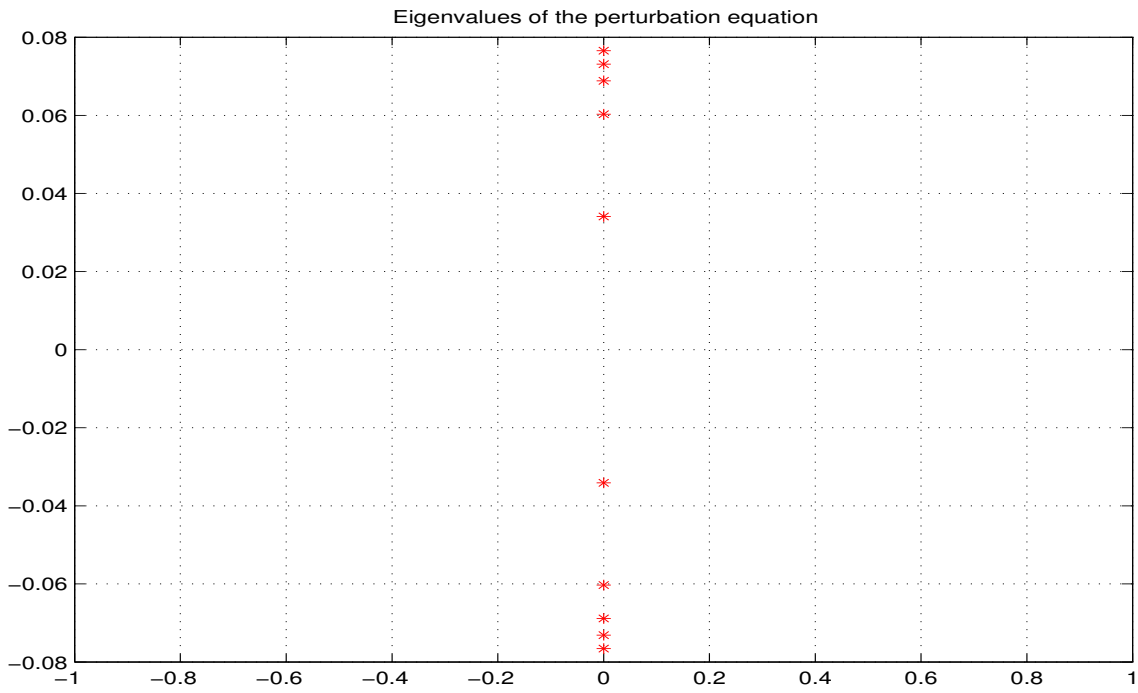


Figure 6.6: The eigenvalue associated with the linear perturbation about the ground state

the origin.

Now we want to consider an evolution of the ground state with deliberately added perturbations. The aim here being to see if we can induce finite perturbation around the ground state. We expect that these would be at the frequencies determined by the first order linear perturbation about the ground state and the corresponding higher order linear perturbations, which are all imaginary.

We take a ground state, add on to it an exponential distribution and evolve. Then we can Fourier transform the absolute value of the wave-function near the origin to obtain the different frequencies that make up the oscillation. We can compare these observed frequencies to those which were obtained in chapter 4 doing the first order linear perturbation.

We use in our evolution method, $N = 50$ Chebyshev points and $dt = 1$ as the time step to obtain $n = 170,000$ sampling points. With this amount of sampling points and time step we should be able to obtain the frequencies to an accuracy of $\frac{\pi}{ndt} = 3.6E-5$. In figure 6.7 we plot a section of the oscillation about the ground state at a given point. In figure 6.8 we plot the Fourier transform of the oscillation about the ground state at a given point. In table 6.1 we present the frequencies observed (see figure 6.8) with those obtained in chapter 4, which are plotted in figure 6.6.

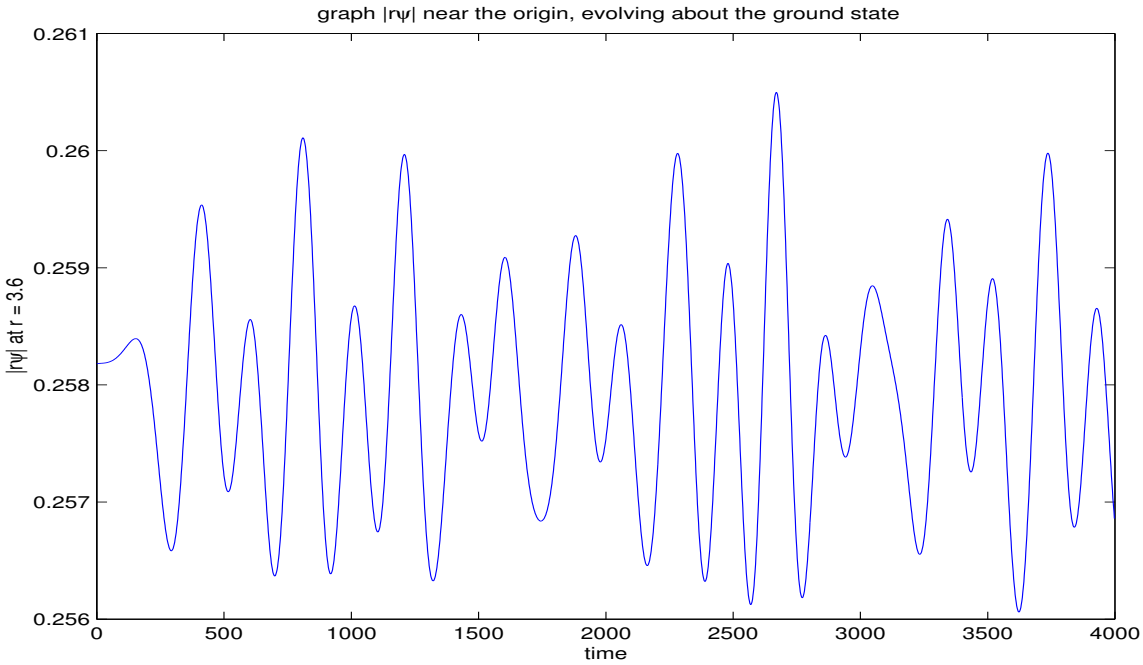


Figure 6.7: The oscillation about the ground state at a given point

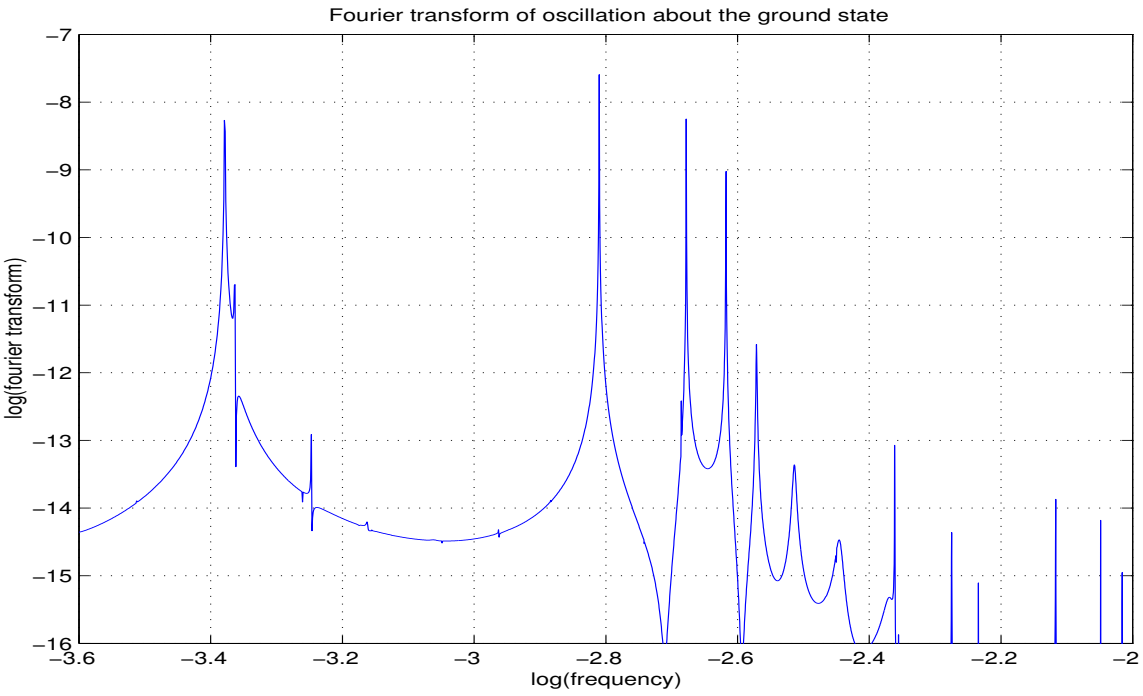


Figure 6.8: The Fourier transform of the oscillation about the ground state

linearised frequencies	frequencies	log of size of peak .
± 0.0341	0.0341	-8.300
± 0.0603	0.0602	-7.632
± 0.0688	0.0687	-8.265
± 0.0731	0.0730	-9.038
± 0.0765	0.0764	-11.603
± 0.0810	0.0810	-13.431
± 0.0867	0.0866	-14.466
± 0.0934	0.0943	-13.102
± 0.1012	0.1028	-14.377
± 0.1099	0.1071	-15.115
± 0.1196	0.1204	-16.322

Table 6.1: Linearised eigenvalues of the first state and observed frequencies about it, $n = 170,000$

6.3 Evolution of the higher states

6.3.1 Evolution of the second state

We note that whether or not the higher stationary states are non-linearly stable they should evolve for some time like stationary states before numerical errors grow sufficiently to make them decay.

For the evolution of the second state we find that we have again the correct phase factor (see figure 6.9).

There are a few issues to look at with the evolution of the higher stationary states when the state are evolved for long enough.

1. Do they decay when perturbed into some scatter and some multiple of the ground state? Numerical calculation of a stationary state is only correct up to numerical errors, so if the state is non-linearly unstable it is likely that it will decay from just evolving.

2. How long does the state take to decay?

3. What is the amount of probability left in the ground state?

4. Is this above the estimate of section 5.10 based on the assumption that the energy that is scattered is positive?

As we can see in figure 6.10 the second state is decaying, emitting scatter and leaving something that looks like a multiple of the ground state.

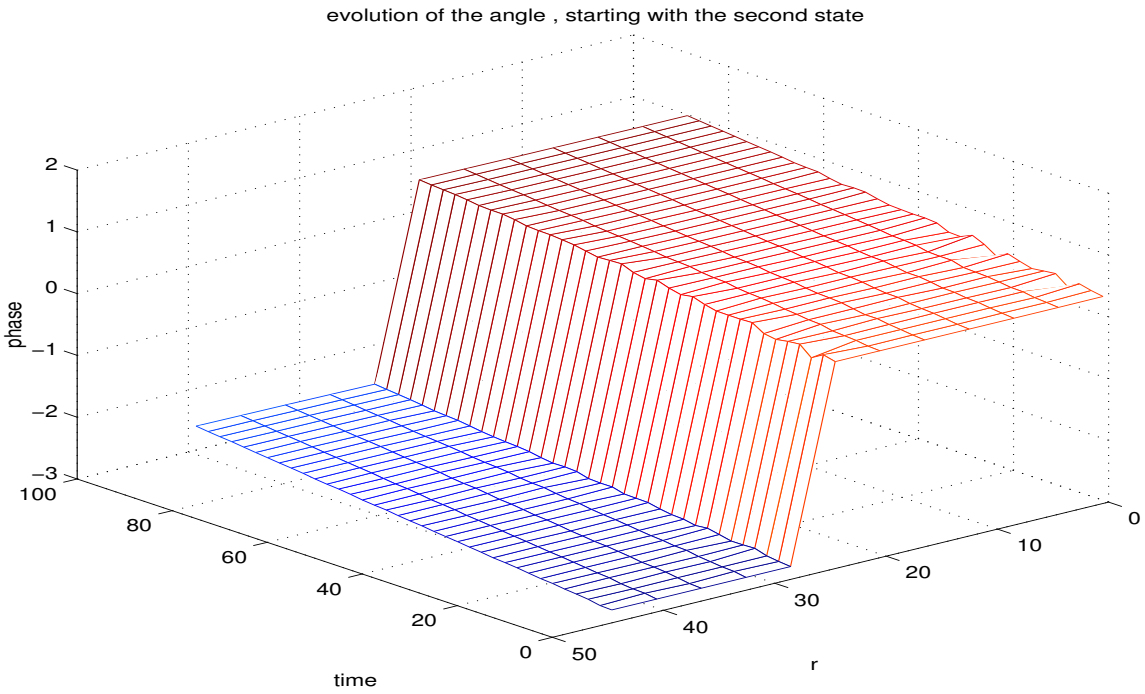


Figure 6.9: The graph of the phase angle of the second state as it evolves

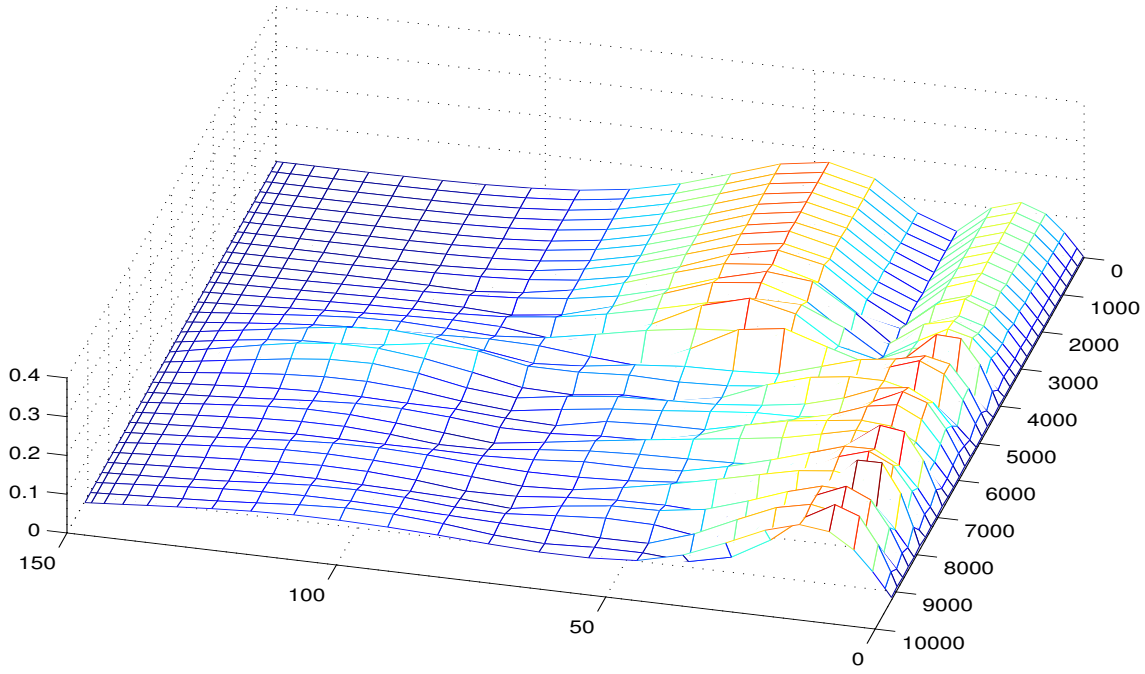


Figure 6.10: The long time evolution of the second state in Chebyshev methods

We can see in figure 6.11 the evolution of the second state. This is decaying into something that looks like a multiple of the ground state, but here we have used in a lower tolerance in the computation and the result is that the second state takes longer to decay which would correspond to the error growing at a smaller rate.

As in the case of the ground state we can consider $|r\psi|$ at a particular point to obtain a time series. The resulting time series divides up into three sections. The first section is the evolution of the second stationary state, that is the wave-function is just a perturbation about the second state. The next section is the part where it decays into the ground state emitting scatter off the grid. The final section corresponds to a "multiple" of the ground state with a perturbation about it.

The first section has an exponentially growing oscillation which we expect to be the unstable growing mode obtained in the linear perturbation theory (chapter 4). To confirm the expectation that this is the growing mode we consider modifying the function so that we would just have the oscillation with no growing exponential. To remove the growing exponential we consider:

$$f(t) = \frac{|r\psi(r_1, t)| - |r\psi(r_1, 0)|}{\exp(\text{real}(\lambda)t)} + |r\psi(r_1, 0)| \tag{6.2}$$

where λ is the eigenvalue with nonzero real part obtained in solving (3.24) about the second state and r_1 is the point at which the time series has been obtained, which is near the origin. We define A to be such that $A = |r\psi(r_1, 0)|$. We note that $f(t) - A$ is almost just an oscillation and we plot this in figure 6.12.

We then Fourier transform $f(t) - A$ (see figure 6.13) and we obtain approximately one frequency which is at 0.0095, which is approximately that of the growing mode in the linear perturbation theory. Here we used $n = 6000$ sampling points and $dt = 1$ was the time step.

The growing mode here provides information about the time taken for the state to decay, since the perturbation has to be significantly large before the nonlinear effects will cause the decay of the wave-function.

The second section of the time series we shall ignore, since it is neither perturbation about the second state nor perturbation about the ground state.

We expect the third section to be an oscillation about a "multiple" of the ground state and so we should get frequencies of oscillation but the linear perturbation frequencies should be transformed to make up for the difference in normalisation. What we find is the normalisation is still decreasing, so we are unable to obtain any consistent frequencies.

We can plot the resulting probability in the region in figure 6.14, with bound on the action from section 5.10. In figure 6.14 the normalisation and the action bound seem to be converging toward the same value.

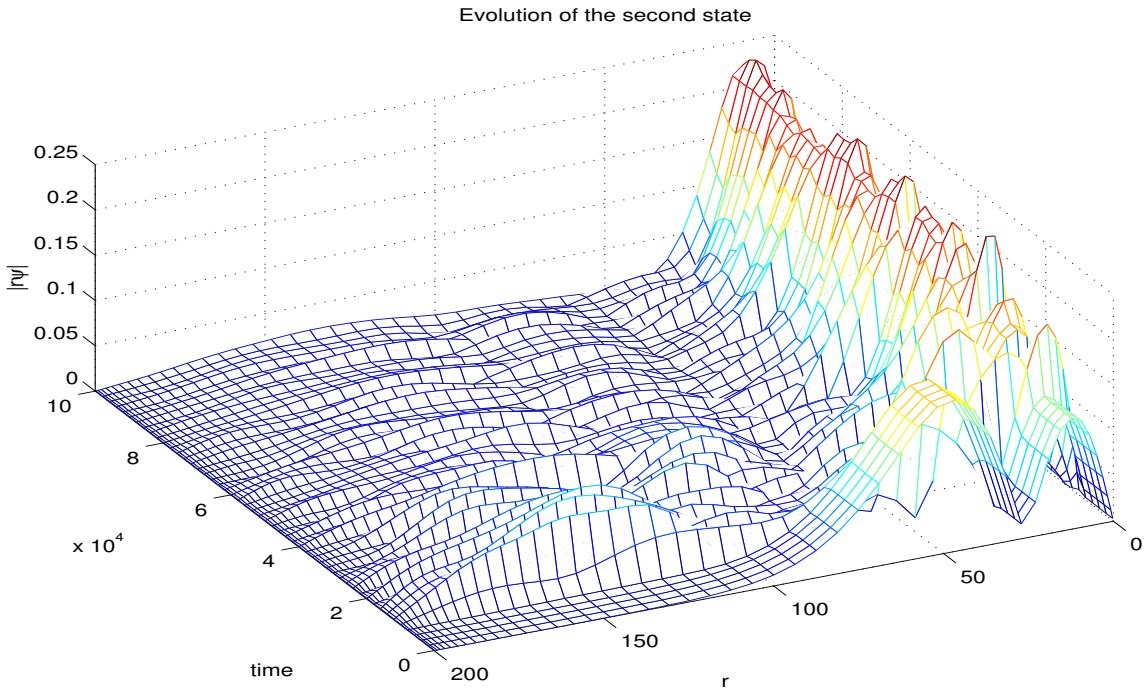


Figure 6.11: Evolution of second state tolerance E-9 approximately

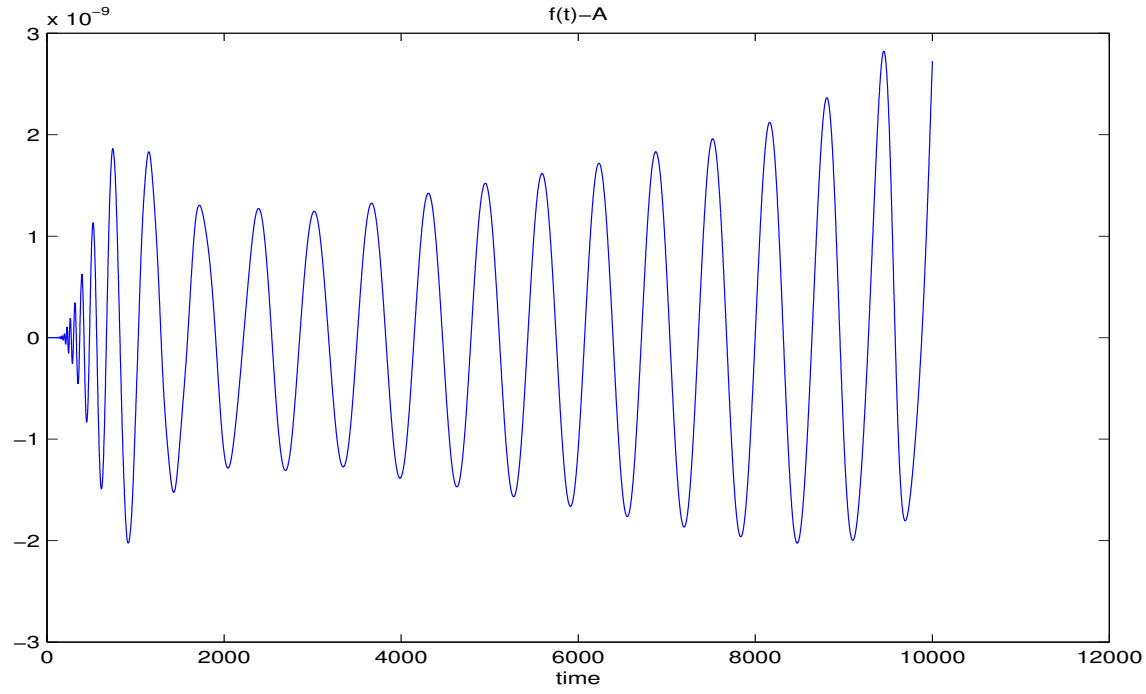


Figure 6.12: $f(t) - A$

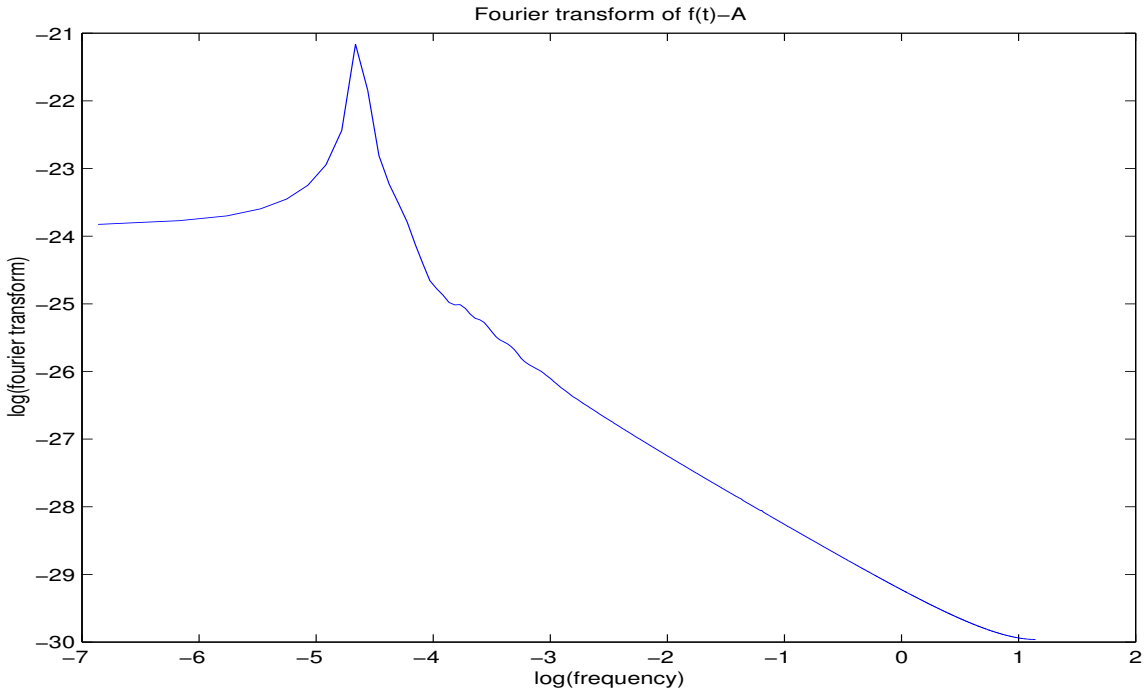


Figure 6.13: Fourier transformation of $f(t) - A$

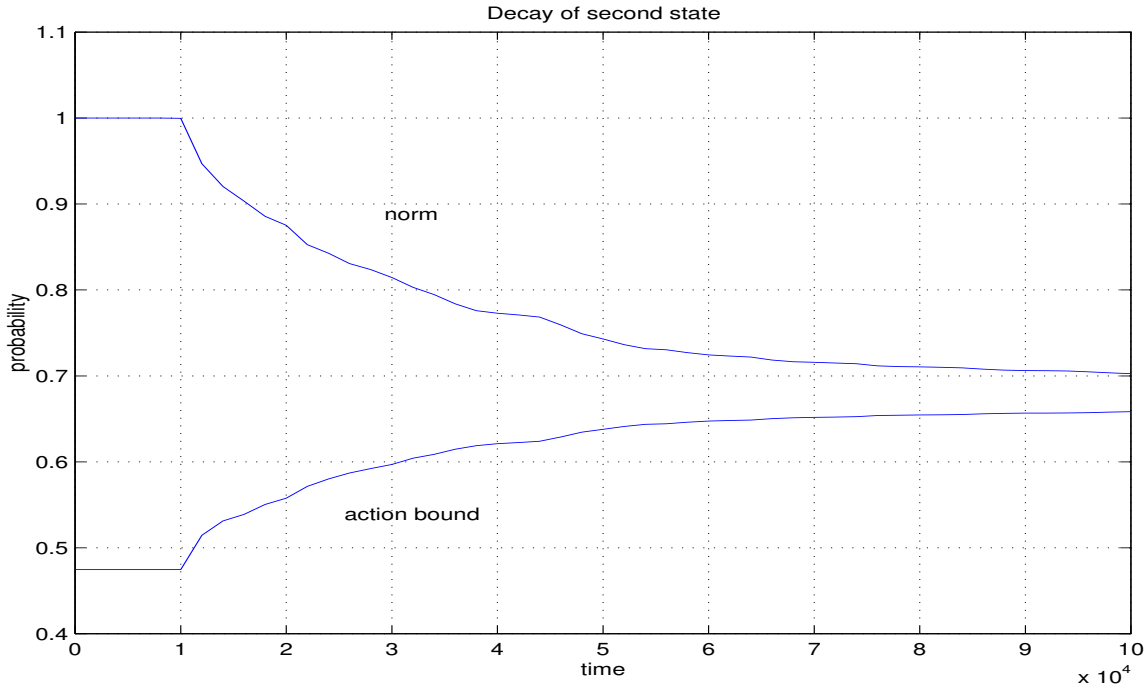


Figure 6.14: Decay of the second state

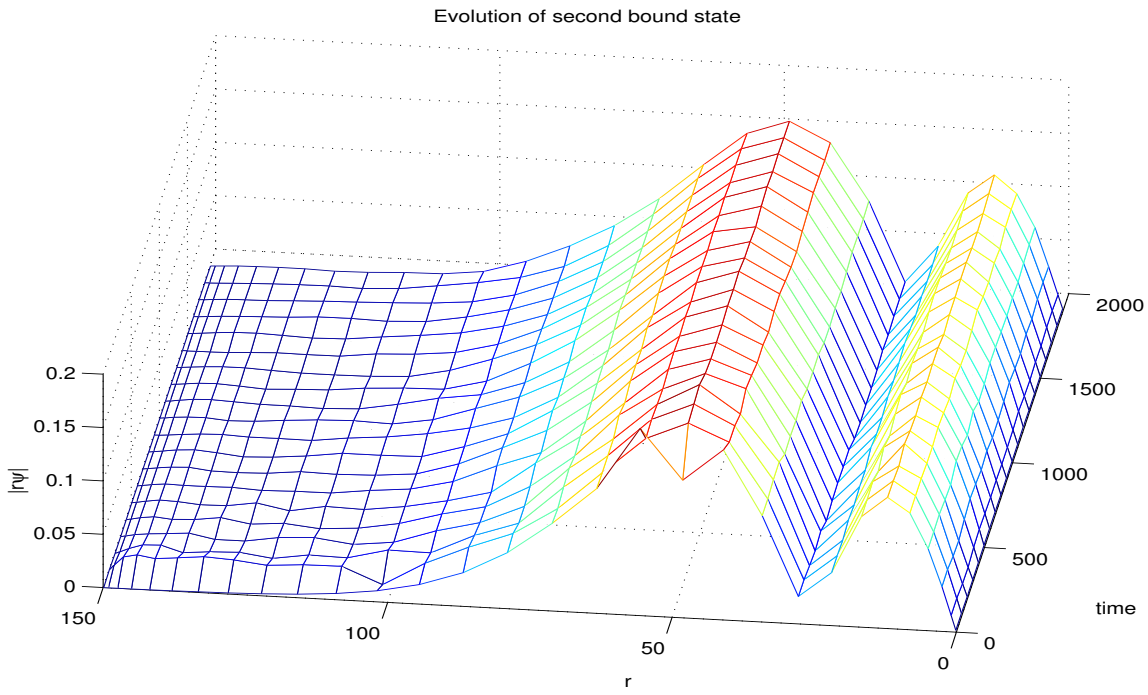


Figure 6.15: The short time evolution of the second bound state

Although we have found the second state to be unstable as expected and the second state grows initially with an exponential mode, we can obtain more of the linear perturbation modes. We consider the evolution of the second state for a short time with an added perturbation to see if it does give any frequencies, and then check whether these agree with the linear perturbation theory. We note that since we can only evolve for a short time before the state decay the accuracy will be low. In figure 6.15 we plot the evolution of the second state with an added perturbation, and in figure 6.16 we plot the oscillation about the second state at a fixed radius.

In table 6.2 we present the observed frequencies about the second state compared with the eigenvalues of the linear perturbation. We note that the accuracy of the observed frequencies here is 0.002.

In figure 6.17 we plot the Fourier transform of the second state with added perturbation, note that the graph is not smooth due to the low accuracy.

6.3.2 Evolution of the third state

Now we consider the evolution of the third stationary state. In figure 6.18 we plot the evolution of the third state, as expected we see that it is unstable and decays, emitting scatter and leaving a "multiple" of the ground state.

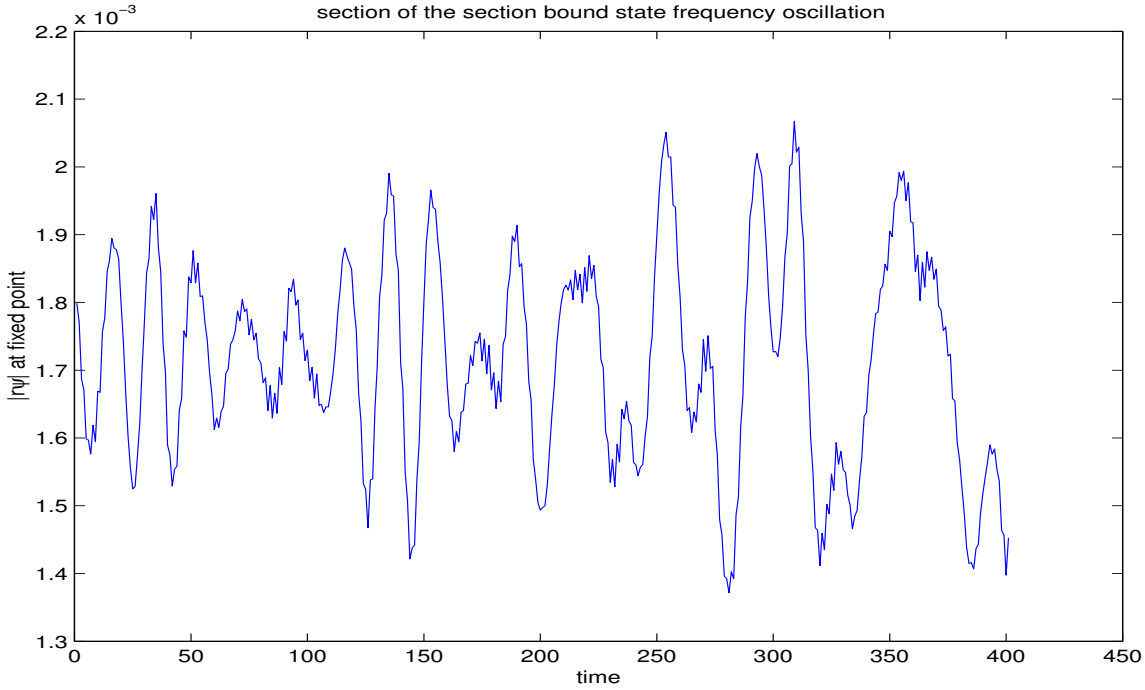


Figure 6.16: Oscillation about second state at a fixed radius

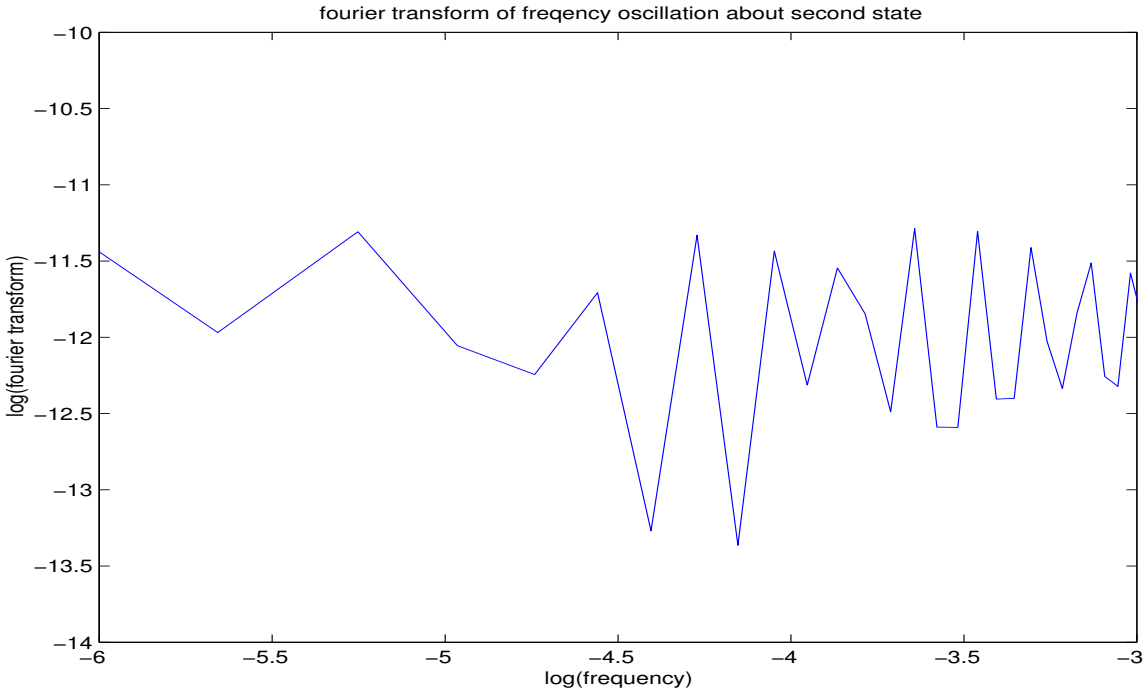


Figure 6.17: Fourier transform of the oscillation about the second state

linearised eigenvalues	frequencies
0.0000	
$0 - 0.0030i$	
$0 - 0.0086i$	
$-0.0014 - 0.0100i$	
$0.0014 + 0.0100i$	0.0114
$0 - 0.0153i$	
$0 - 0.0211i$	0.0190
$0 - 0.0276i$	0.0266
$0 - 0.0351i$	0.0342
$0 - 0.0434i$	0.0416
$0 - 0.0526i$	0.0530
$0 - 0.0627i$	0.0643
$0 - 0.0737i$	0.0726
$0 - 0.0855i$	0.0853
$0 - 0.0982i$	0.0987
$0 - 0.1118i$	0.1123
$0 - 0.1263i$	0.1247
$0 - 0.1417i$	0.1419
$0 - 0.1579i$	0.1577

Table 6.2: Linearised eigenvalue of the second state and frequencies about it.

We can again consider the perturbation about the third state initially as with the second state before it decays, we find that it is exponentially growing. We now transform as in (6.2) but with λ now being the eigenvalue of the linear perturbation about the third state with the maximum real part. In figure 6.19 we plot $f(t) - A$, now we Fourier transform $f(t) - A$ (see figure 6.20) and obtain two frequencies at 0.0022 and 0.0050 which correspond to the imaginary parts of the two complex eigenvalues of the linear perturbation.

We plot the resulting probability in the region in figure 6.21, with bound on the action from section 5.10. In figure 6.21 the normalisation and the action bound seem to be converging toward the same value.

6.4 Evolution of an arbitrary spherically symmetric Gaussian shell

In this section we consider the evolution of Gaussian shells or lumps (5.20) varying the three associated parameters. The mean velocity v the rate the shell is expanding outwards, we note the centre of mass is not moving. The mean position a, the radial position of the peak of the shell. The width σ of the shell. Since varying these factors will produce varied initial conditions, we can perform a mesh analysis and a time step analysis on the evolution

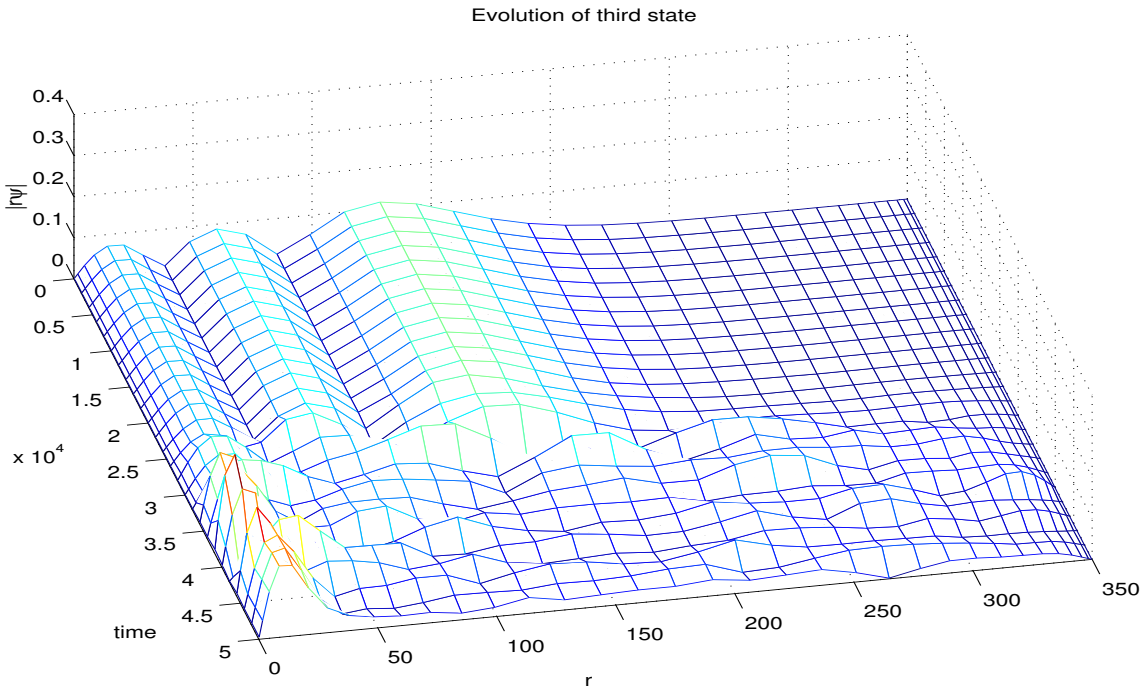


Figure 6.18: Evolution of third state

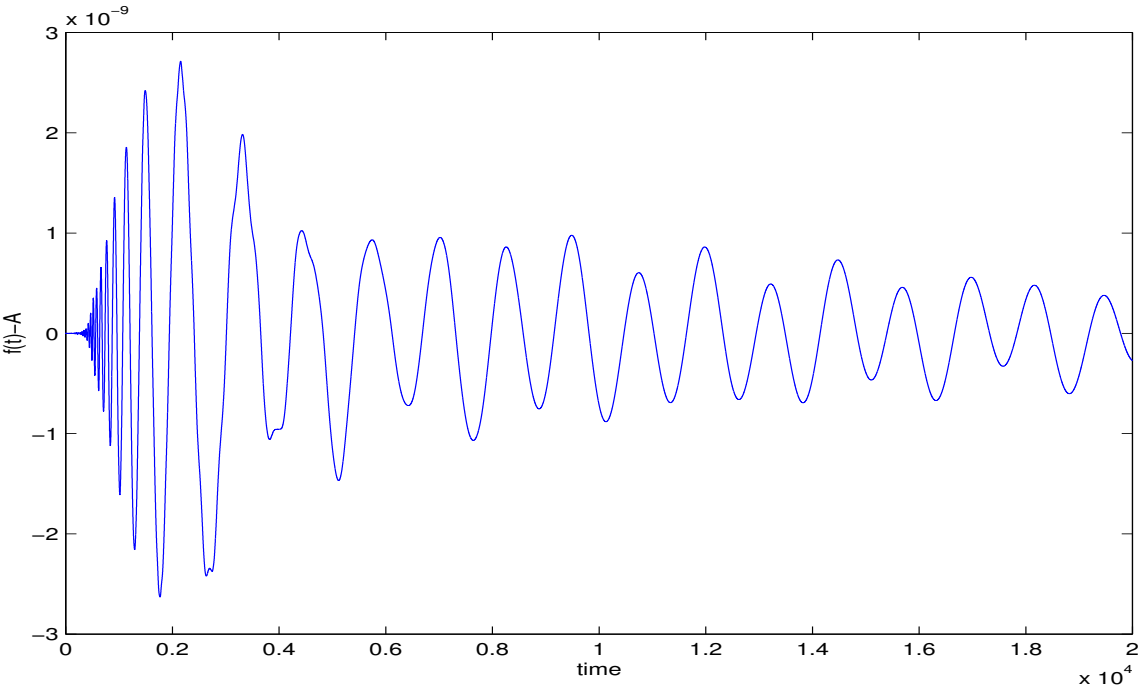


Figure 6.19: $f(t) - A$ with respect to the third state

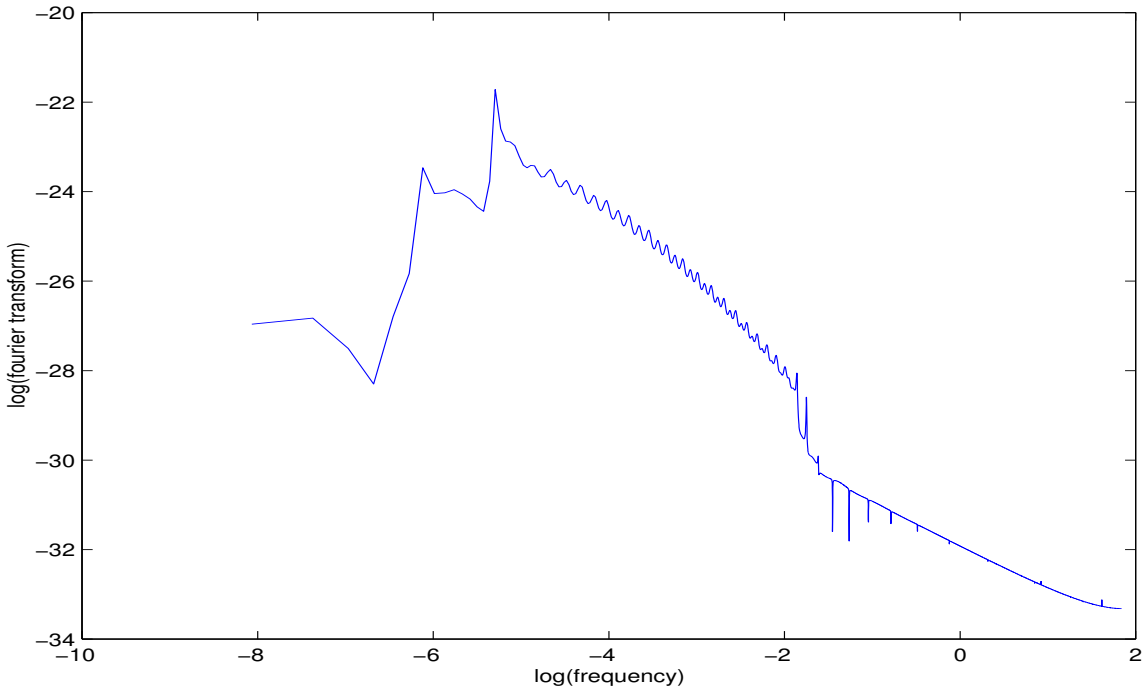


Figure 6.20: Fourier transform of growing mode about third state

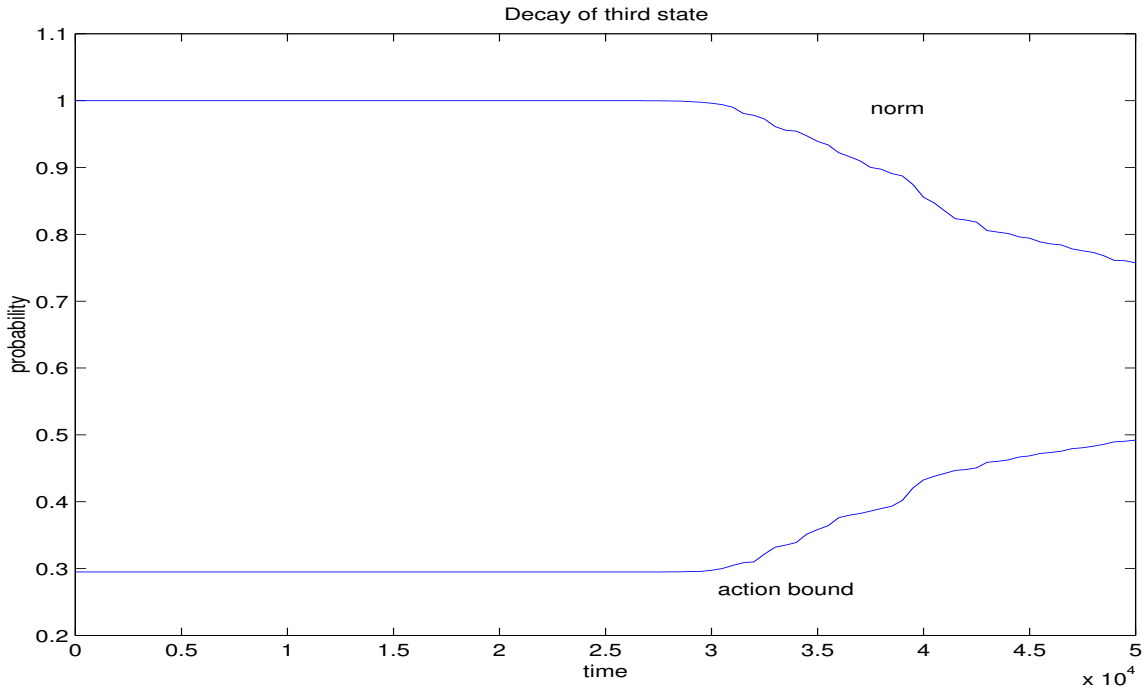


Figure 6.21: Probability remaining in the evolution of the third state

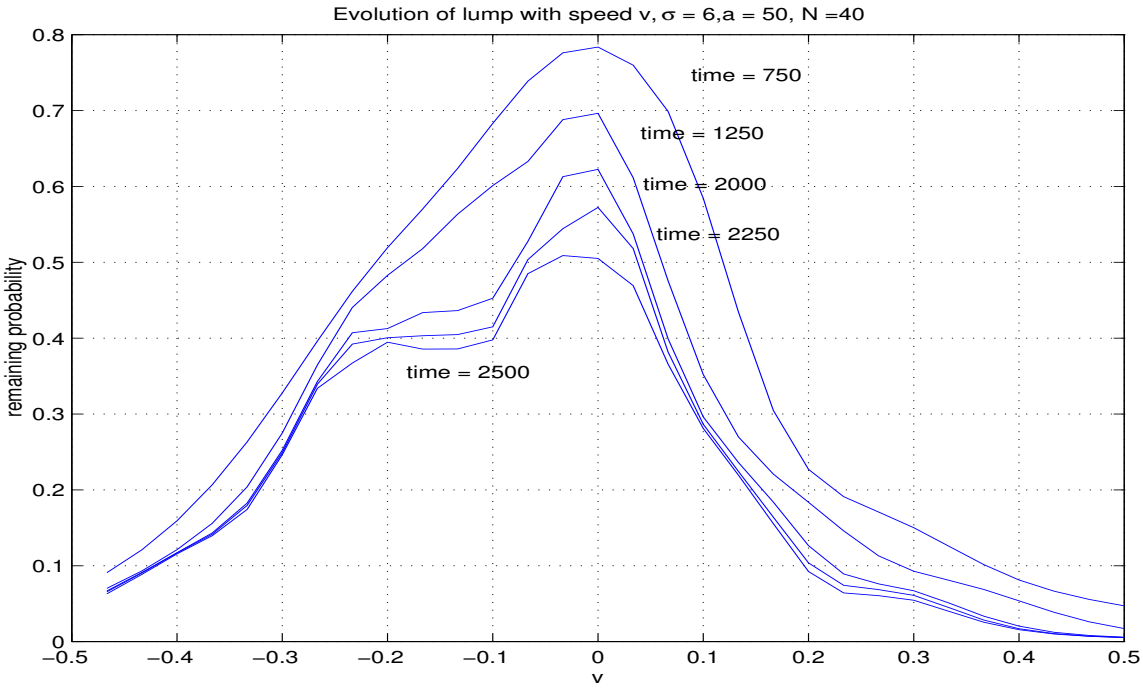


Figure 6.22: Progress of evolution with different velocities and times

as outlined in section 5.9. We also look at how much of the wave-function is left at the origin which we expect to form a "multiple" of the ground state in the sense of section 5.10. We expect that the more energy in the system to begin with the less energy and therefore probability eventually will remain near the origin.

6.4.1 Evolution of the shell while changing v

In figure 6.22 we plot the remaining probability at different times and at different initial mean velocities v, while a and σ are constant initially. We notice that the negative values of v correspond to an outward going wave-function moving off the grid and a positive v is an inward going wave-function. We also notice that the graph is not symmetric about $v = 0$ even though the wave-function energy is symmetric about $v = 0$.
We compare the difference in different time steps to the result in figure 6.23 and note that the difference is small. We calculate the "Richardson fraction" for different values of h_i's we plot the different Richardson fractions in figure 6.24.

We also show in table 6.3 the mean calculated value against the values expected for quadratic convergence in time. Since the values are close to the expected values we are justified in supposing that the convergence is quadratic in the time step.

We can also plot the variation with in the number of Chebyshev points to show convergence with increasing N (this time it does not make sense to do Richardson Extrapolation).

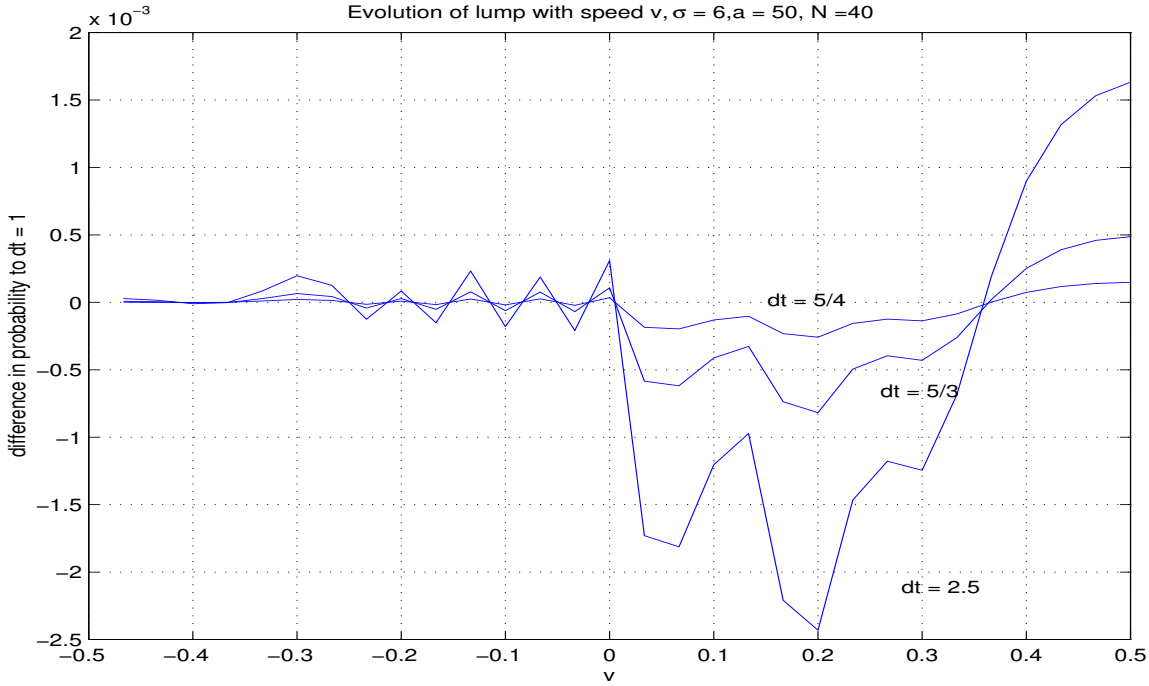


Figure 6.23: Difference of the other time steps compared with $dt = 1$

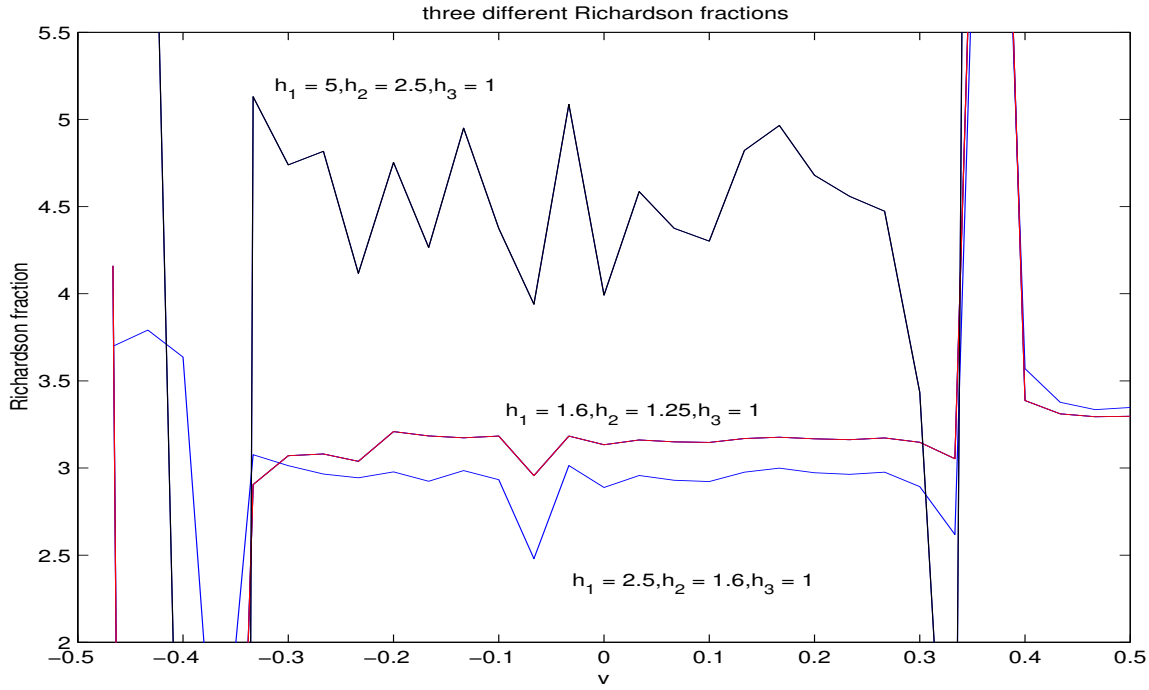


Figure 6.24: Richardson fraction done with different h_i's

h_1	h_2	h_3	expected value	median value calculated
5	2.5	1	4.5714	4.7094
2.5	1.667	1	2.9531	2.9761
1.667	1.25	1	3.1605	3.1652

Table 6.3: Richardson fractions

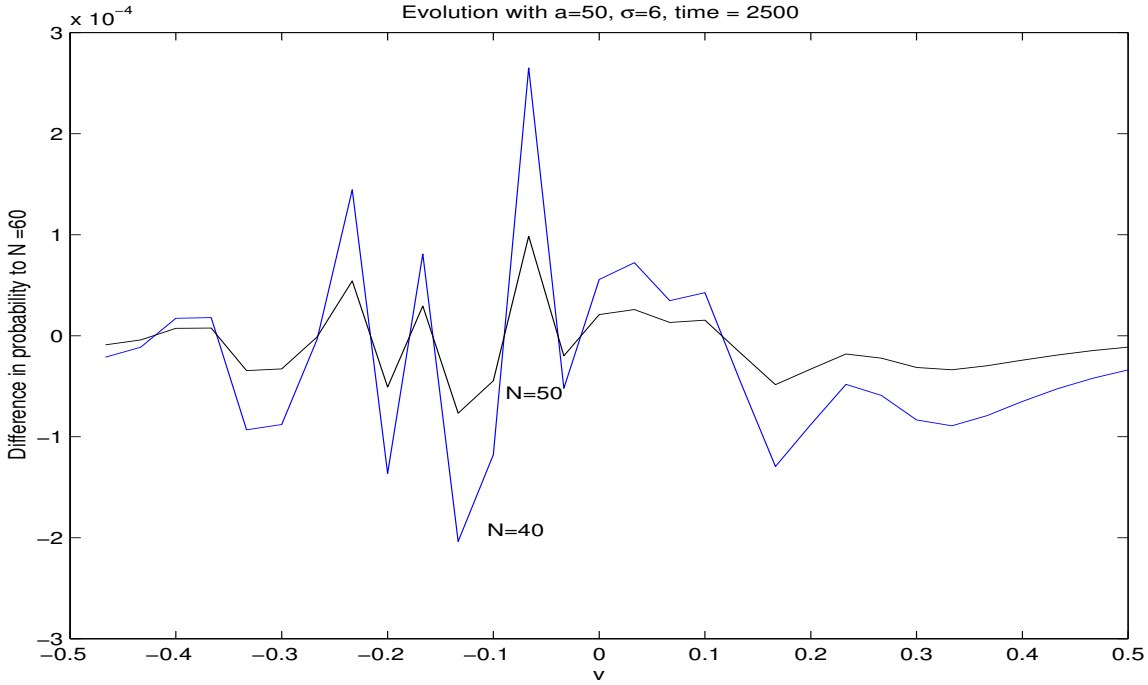


Figure 6.25: Difference in N value

The plot is in figure 6.25.

6.4.2 Evolution of the shell while changing a

We plot in figure 6.26 the remaining probability at different times with different mean positions a initially, while $v = 0$ and σ is constant initially. We also plot in figure 6.27 the values for the remaining probability with different time steps.

We can again calculate the "Richardson fraction" as in section 5.9 in the time steps, in figure 6.28 we plot the "Richardson fraction" at different values of a. In table 6.4 we show the mean calculated value against the expected values and again we see that we have quadratic convergence in the time step.

We can also plot the variation with in the number of Chebyshev points to show convergence with increasing N (this time it does not make sense to do Richardson Extrapolation). The plot is in figure 6.29.

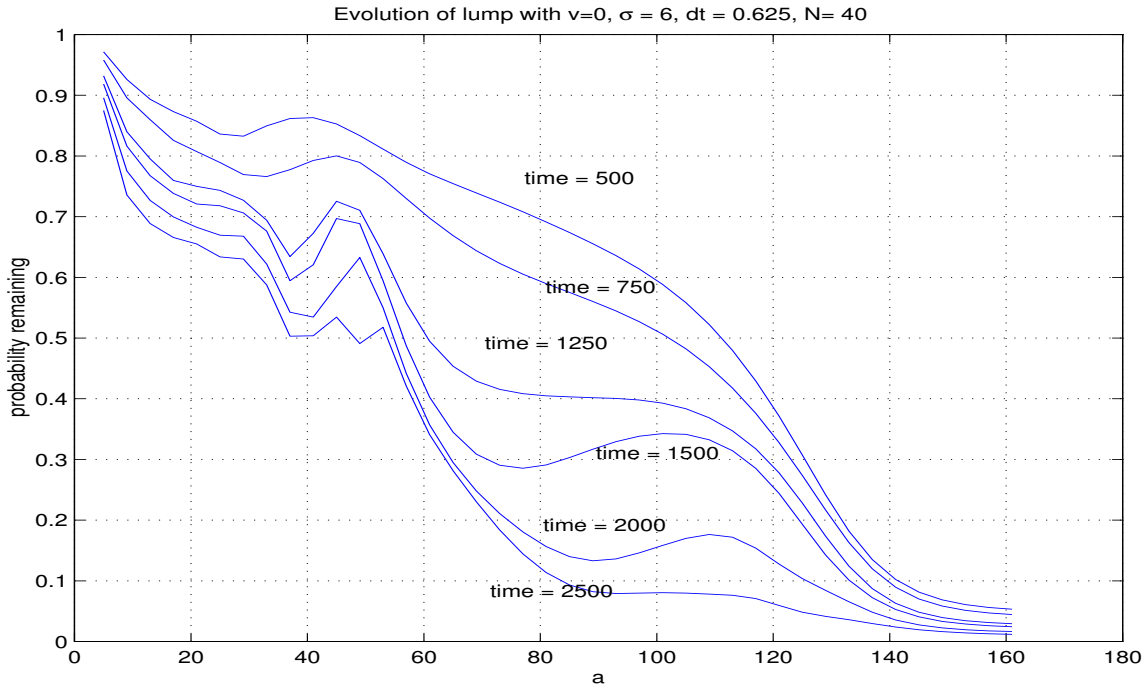


Figure 6.26: Evolution of lump at different a

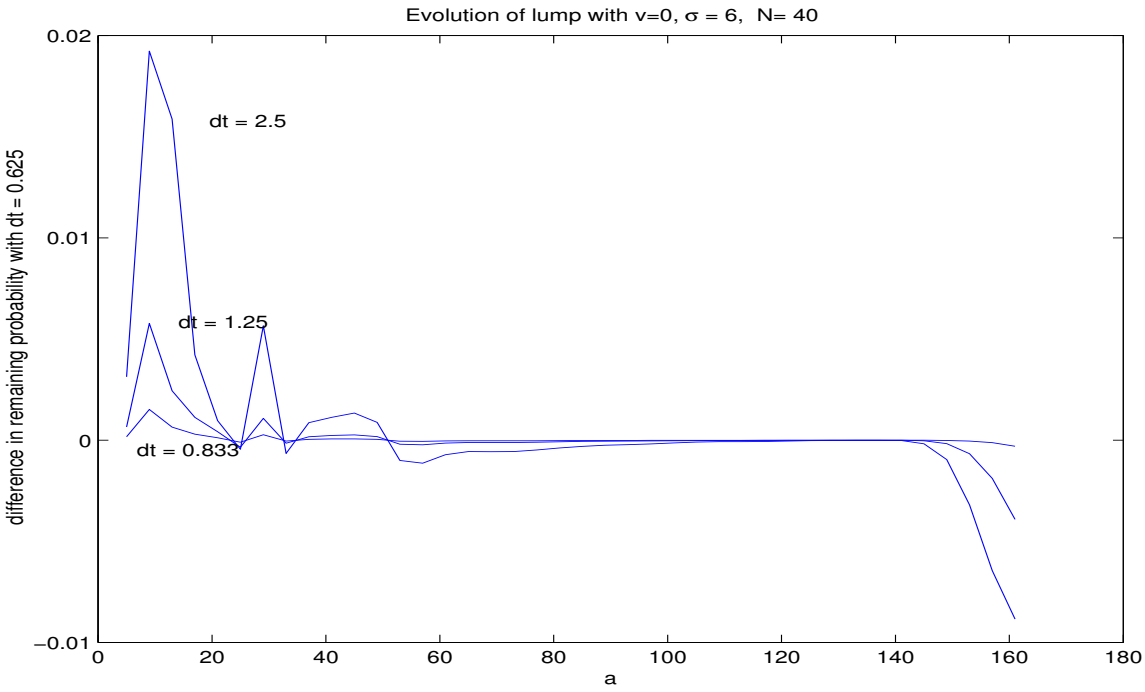


Figure 6.27: Comparing differences with different time steps

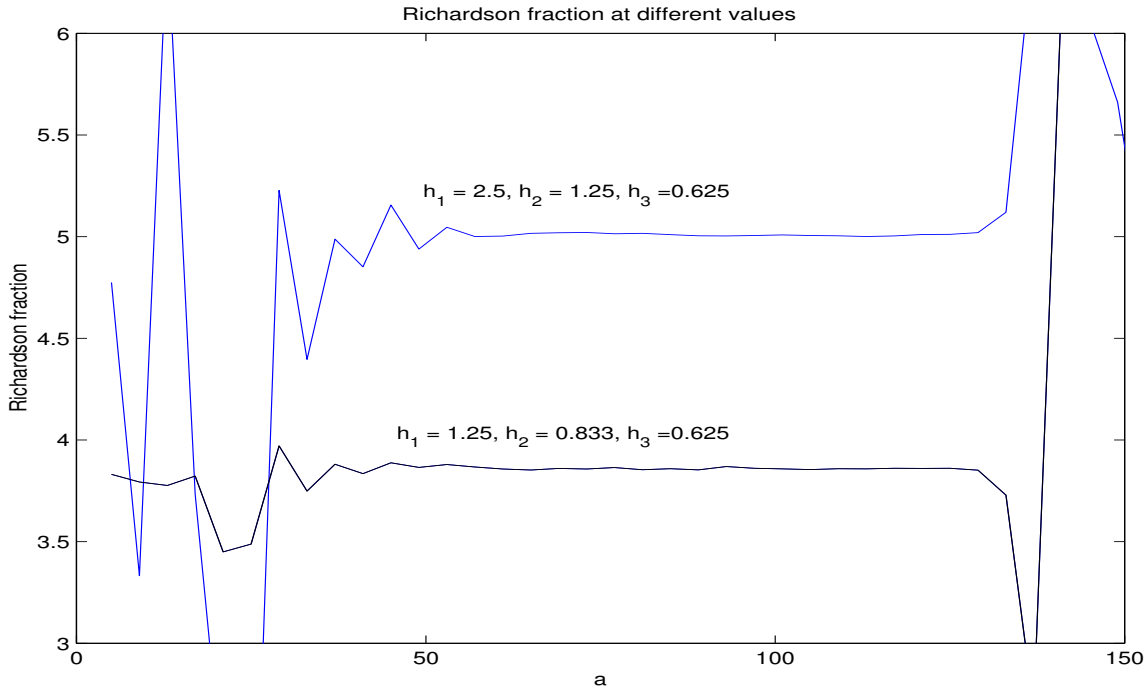


Figure 6.28: Richardson fraction with different h_i's

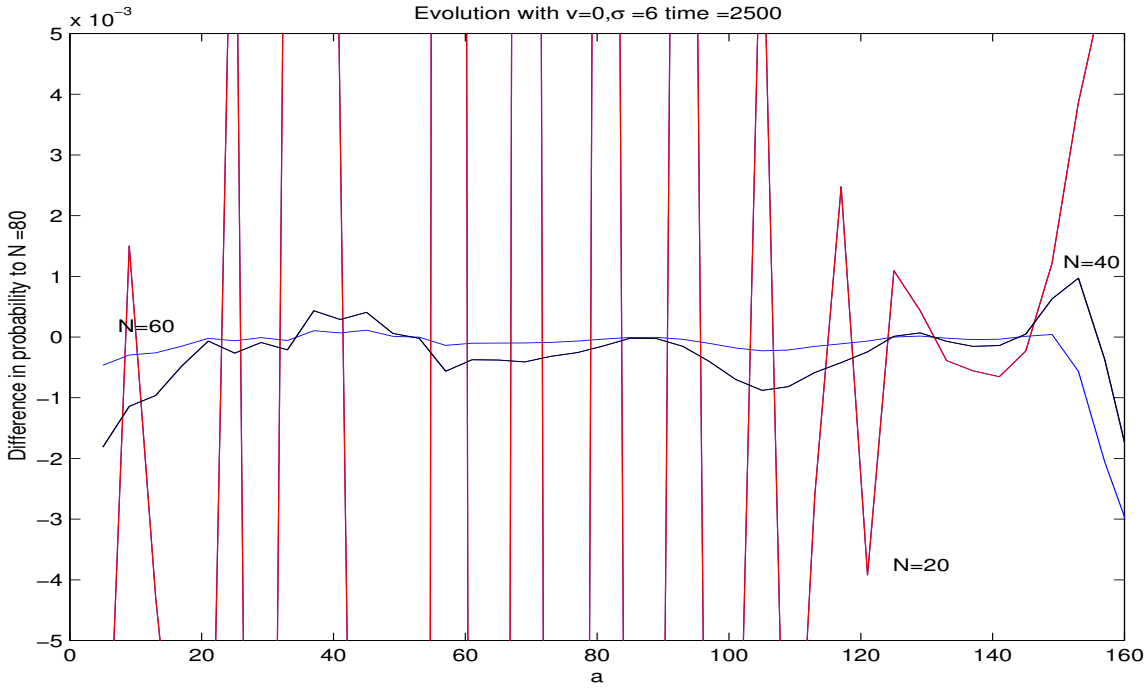


Figure 6.29: Difference in N value

h_1	h_2	h_3	expected value	median value calculated
2.5	1.25	0.625	5	5.0055
1.25	0.833	0.625	3.8571	3.8588

Table 6.4: Richardson fractions

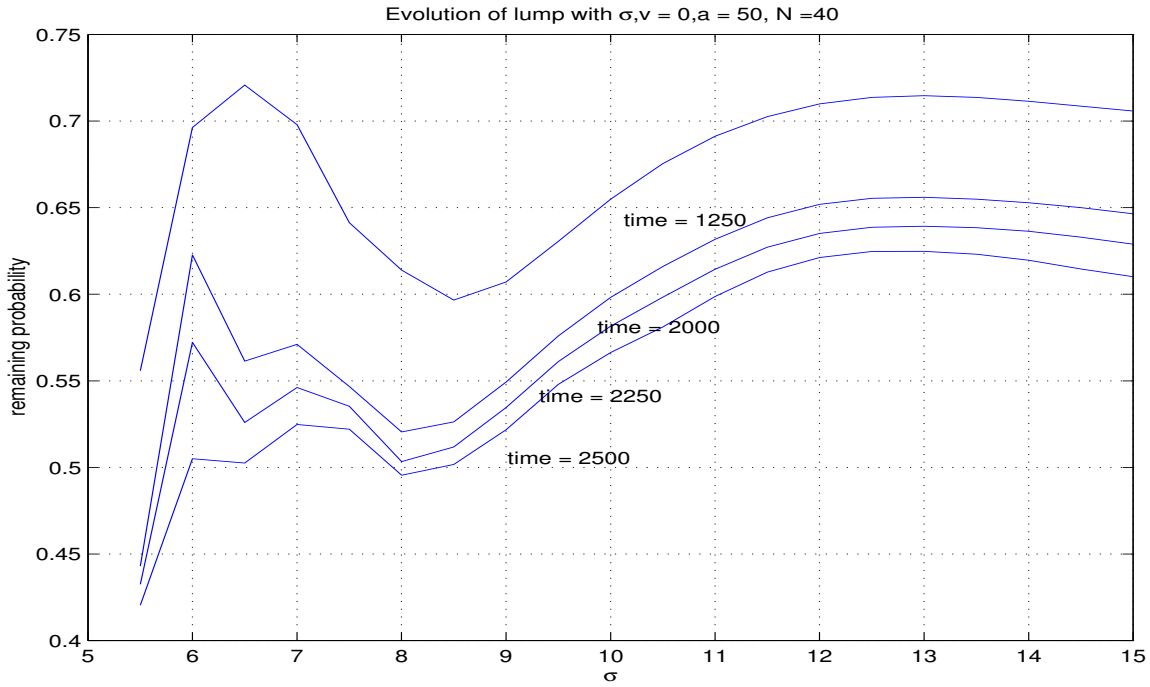


Figure 6.30: Evolution of lump at different σ

6.4.3 Evolution of the shell while changing σ

We plot in figure 6.30 the remaining probability at different times with different widths σ initially, while $v = 0$ and $a = 50$. We also plot in figure 6.31 the values for the remaining probability at different time steps.

We can again calculate the "Richardson fraction", using data obtain at different time steps. In figure 6.32 we plot the "Richardson fraction" at different values of σ. In table 6.5 we show the mean calculated value against the expected values again we have quadratic convergence.

h_1	h_2	h_3	expected value	median value calculated
0.72	0.625	0.55	2.455	2.458
0.83	0.625	0.55	1.908	1.914

Table 6.5: Richardson fractions

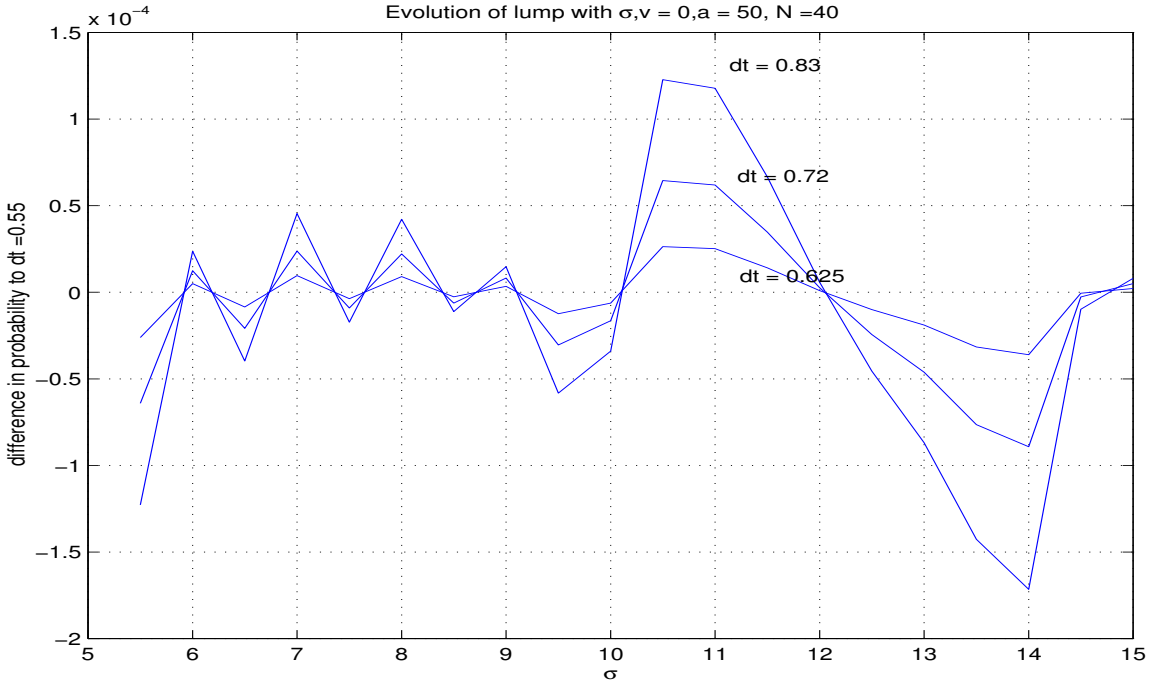


Figure 6.31: Comparing differences with different time steps

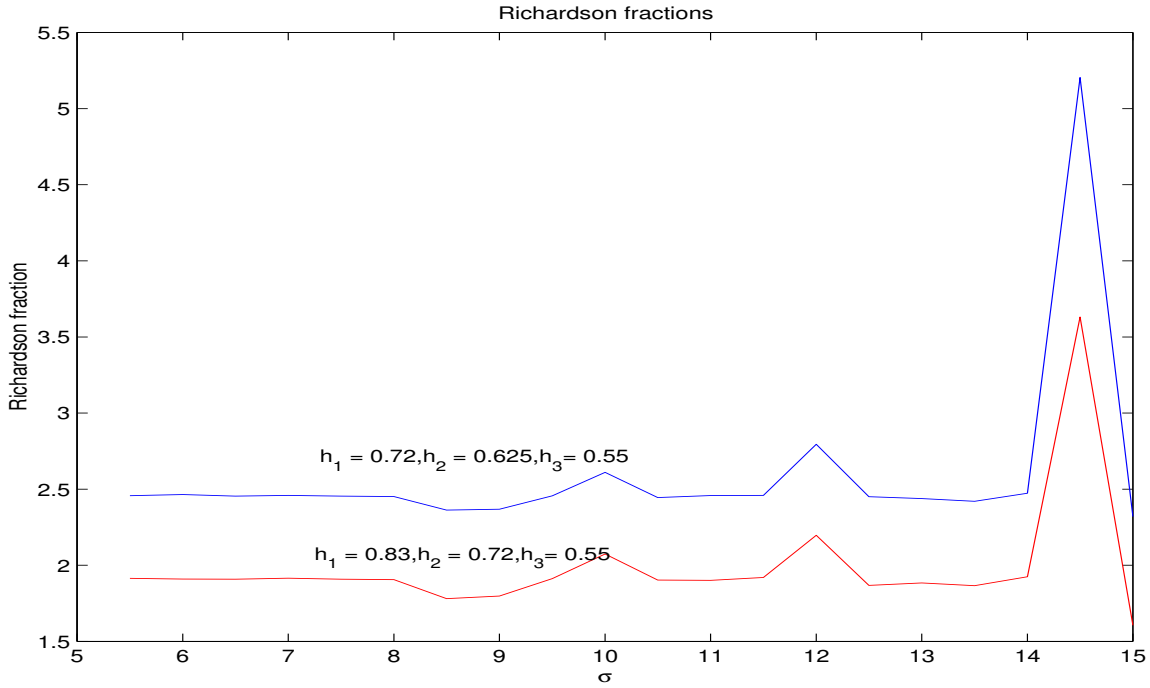


Figure 6.32: Richardson fractions

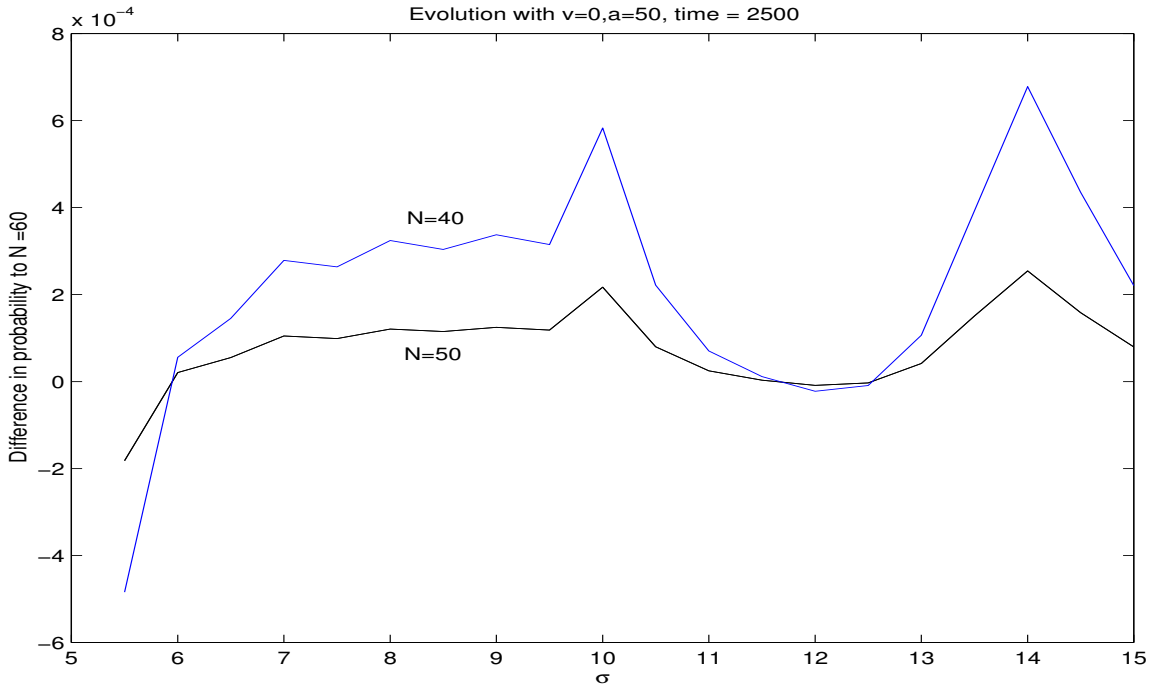


Figure 6.33: Difference in N value

We can also plot the variation with in the number of Chebyshev points to show convergence with increasing N (this time it does not make sense to do Richardson Extrapolation). The plot is in figure 6.33.

6.5 Conclusion

In this chapter, we have evolved the S-N equations using different initial condition, using the methods of Chapter 5. We have checked the ability of the sponge factors to absorb outward going wave-functions moving off the grid. We have checked that the evolution method converges to the solution by evolving the S-N equations with different initial conditions and changing the time steps and the number of Chebyshev points in the grid. To check convergence with respect to the time step we have used Richardson extrapolation and found it to be converging like the second order in the time steps.

The numerical calculations show that:

- The ground state is stable under full nonlinear evolution.
- Perturbations about the ground state oscillate with the frequencies obtained by the linear perturbation theory.

- Higher states are unstable and will decay into a "multiple" of the ground state, while emitting some scatter off the grid.
- The decay time for the higher states is controlled by the growing linear mode obtained in linear perturbation theory.
- Perturbations about higher states will oscillate for a little while (until they decay) according to the linear oscillation obtained by the linear perturbation theory.
- The evolution of different Gaussian shells indicates that any initial condition appears to decay as predicted in section 5.10, that is they scatter to infinity and leave a "multiple" of the ground state.

Chapter 7

The axisymmetric S-N equations

7.1 The problem

There are two ways of adding another spatial dimension to the problems we are solving. In this chapter, we consider solving the S-N equations for an axisymmetric system in 3 dimensions. Then the wave-function is a function of the polar coordinates r and θ but independent of the polar angle. In chapter 8, we consider the S-N equations in Cartesian coordinates with a wave-function independent of z.

For the axisymmetric problem, we shall first find stationary solutions. This will give some stationary solutions with nonzero total angular momentum including a dipole-like solution that appears to minimise the energy among wave functions that are odd in z. We then consider the time-dependent problem. In particular we evolve the dipole-like state, which turns out to be nonlinearly unstable - the two regions of probability density attract each other and fall together leaving as in chapter 6 a multiple of the ground state.

7.2 The axisymmetric equations

We consider the S-N equations which in the general non-dimensionalised case are:

$$i\psi_t = -\nabla^2\psi + \phi\psi, \tag{7.1a}$$

$$\nabla^2\phi = |\psi|^2. \tag{7.1b}$$

Now consider spherical polar coordinates:

$$x = r\cos\alpha\sin\theta, \tag{7.2a}$$

$$y = r\sin\alpha\sin\theta, \tag{7.2b}$$

$$z = r\cos\theta, \tag{7.2c}$$

where $r \in [0, \infty)$ and $\alpha \in [0, 2\pi)$, $\theta \in [0, \pi]$.
The Laplacian becomes:

$$\nabla^2\psi = \frac{1}{r}\frac{\partial^2(r\psi)}{\partial r^2} + \frac{1}{r^2 sin\theta}\frac{\partial(sin\theta\frac{\partial\psi}{\partial\theta})}{\partial\theta} + \frac{1}{r^2 sin^2\theta}\frac{\partial^2\psi}{\partial\alpha^2}, \tag{7.3}$$

so in the case where ψ is only dependent on θ and r the S-N equations become:

$$i\psi_t = -\frac{1}{r}\frac{\partial^2(r\psi)}{\partial r^2} + \frac{1}{r^2 sin\theta}\frac{\partial(sin\theta\frac{\partial\psi}{\partial\theta})}{\partial\theta} + \phi\psi, \tag{7.4a}$$

$$|\psi|^2 = \frac{1}{r}\frac{\partial^2(r\phi)}{\partial r^2} + \frac{1}{r^2 sin\theta}\frac{\partial(sin\theta\frac{\partial V}{\partial\theta})}{\partial\theta}. \tag{7.4b}$$

Now setting $u = r\psi$ the equations become:

$$iu_t = -u_{rr} + \frac{1}{r^2 sin\theta}(sin\theta u_\theta)_\theta + \phi u, \tag{7.5a}$$

$$\frac{|u|^2}{r} = \phi_{rr} + \frac{1}{r^2 sin\theta}(sin\theta\phi_\theta)_\theta. \tag{7.5b}$$

For the boundary conditions we still want ψ to be finite at the origin so that the wave-function is well-behaved, and to satisfy the normalisation condition we require that $\psi \to 0$ as $r \to \infty$. So we have that $u(0, \theta) = 0$ for all $\theta \in [0, \pi]$ and that $u(r, \theta) \to 0$ for all θ as $r \to \infty$. We note that these boundary conditions are spherically symmetric so that the solutions in the spherically symmetric case are still solutions. We know that $\psi_\theta = 0$ at $\theta = 0$ and $\theta = \pi$ since otherwise we would have a singularity in the Laplacian.

The stationary S-N equations are in the general non-dimensionalised case:

$$E_n\psi = -\nabla^2\psi + \phi\psi, \tag{7.6a}$$

$$\nabla^2\phi = |\psi|^2, \tag{7.6b}$$

so the axially symmetric stationary S-N equation are:

$$E_n u = -u_{rr} + \frac{1}{r^2 sin\theta}(sin\theta u_\theta)_\theta + \phi u, \tag{7.7a}$$

$$\frac{|u|^2}{r} = \phi_{rr} + \frac{1}{r^2 sin\theta}(sin\theta\phi_\theta)_\theta. \tag{7.7b}$$

7.3 Finding axisymmetric stationary solutions

To find axisymmetric solutions of the stationary S-N equations we proceed to adapt Jones et al [3] as follows,

1. Take as an initial guess for the potential $\phi = \dfrac{-1}{(1+r)}$.
2. Using the potential ϕ we now solve the time-independent Schrödinger equation that is the eigenvalue problem $\nabla^2\psi - \phi\psi = -E_n\psi$ with $\psi = 0$ on the boundary.

3. Select an eigenfunction in such a way that the procedure will converge to a stationary state. (This needs care, see below.)

4. Once the eigenfunction has been selected calculate the potential due to that eigenfunction.

5. Let the new potential be the one obtained from step 4 above but symmetrise potential since we require that ϕ should be symmetric around the $\alpha = \frac{\pi}{2}$ so that numerical errors do not cause the wave-function to move along the axis (the solution we want should have its centre of gravity at the centre of our grid.) Small errors in symmetry will cause the solution to move along the z-axis and off the grid.

6. Now provided that the new potential does not differ from the previous potential by a fixed tolerance in the norm of the difference, stop else continue from step 2.

In step 2 the method we use to solve the eigenvalue problem involves Chebyshev differentiation in two directions that is in r and θ.

In the case of the Hydrogen atom we know explicitly the wave-function, that is a solution of Schrödinger equation with a $\frac{1}{r}$ potential. In this case the number of zeros in the radial direction and the axial directions follow a characteristic pattern, because the solutions are separable. We can use this in our selection procedure in step 3 to get started, but as the iteration process continues the pattern of zeros wont be fixed and we may need another method. We also note in the case of the Schrödinger equation with a $\frac{1}{r}$ potential that each stationary state in the axially symmetric case has different values of constants E and J^2. These values do not change much under iteration of the stationary state problem and we can use this to obtain the next solution with values of E, J^2 which differ only by a small amount. Here E is :

$$E = \int_{\mathbb{R}^3} -\bar{\psi}\nabla^2\psi + \phi|\psi|^2 d^3x, \tag{7.8}$$

and J^2 is :

$$J^2 = \int_{\mathbb{R}^3} \bar{\psi}[\frac{1}{sin\theta}\frac{\partial(sin\theta\frac{\partial\psi}{\partial\theta})}{\partial\theta} + \frac{1}{sin^2\theta}\frac{\partial^2\psi}{\partial\alpha^2}]d^3x. \tag{7.9}$$

We use a combination of the above properties to select solutions at step 3.

7.4 Axisymmetric stationary solutions

We present in table 7.2 the first few stationary states of the axially symmetric solution of the S-N equations called axp1 to axp8 together with energy and angular momentum. For

comparison we give the energy of the first few spherically symmetric stationary states in table 7.1.

Axp1 in figure 7.1 is the ground state which is from table 7.2, has vanishing J^2. Axp2 in figure 7.3 is a dipole state. It is odd as a function of z, and minimises the energy among odd functions. It was previously found in [22] and has nonzero J^2. Axp3 and axp6 are the second and third spherical states. The others are higher multi-poles, but we are not able to find a simple classification based on, say, energy and angular momentum as there would be for the states of the Hydrogen atom.

We plot the graphs the stationary states that appear in table 7.2, in figures 7.1, 7.3, 7.5, 7.7, 7.9, 7.11, 7.13, 7.15 and the contour plots of these stationary states in figures 7.2, 7.4, 7.6, 7.8, 7.10, 7.12, 7.14, 7.16.

Number of state	Energy Eigenvalue
0	-0.16276929132192
1	-0.03079656067054
2	-0.01252610801692
3	-0.00674732963038
4	-0.00420903256689

Table 7.1: The first few eigenvalues of the spherically symmetric S-N equations

Energy	J	name
-0.1592	zero	axp1
-0.0599	5.1853	axp2
-0.0358	0.002	axp3
-0.0292	2.3548	axp4
-0.0263	17.155	axp7
-0.0208	3.1178	axp8
-0.0162	5.2053	axp5
-0.0115	-1.9E-6	axp6

Table 7.2: Table for E and J^2 of the axially symmetric S-N equations

7.5 Time-dependent solutions

Now we consider the evolution of the axisymmetric time-dependent S-N equations (7.4). We use the same method as in section 5.7, but instead of solving the Schrödinger equation and the Poisson equation in one space dimension we now need to solve them in two space dimensions.

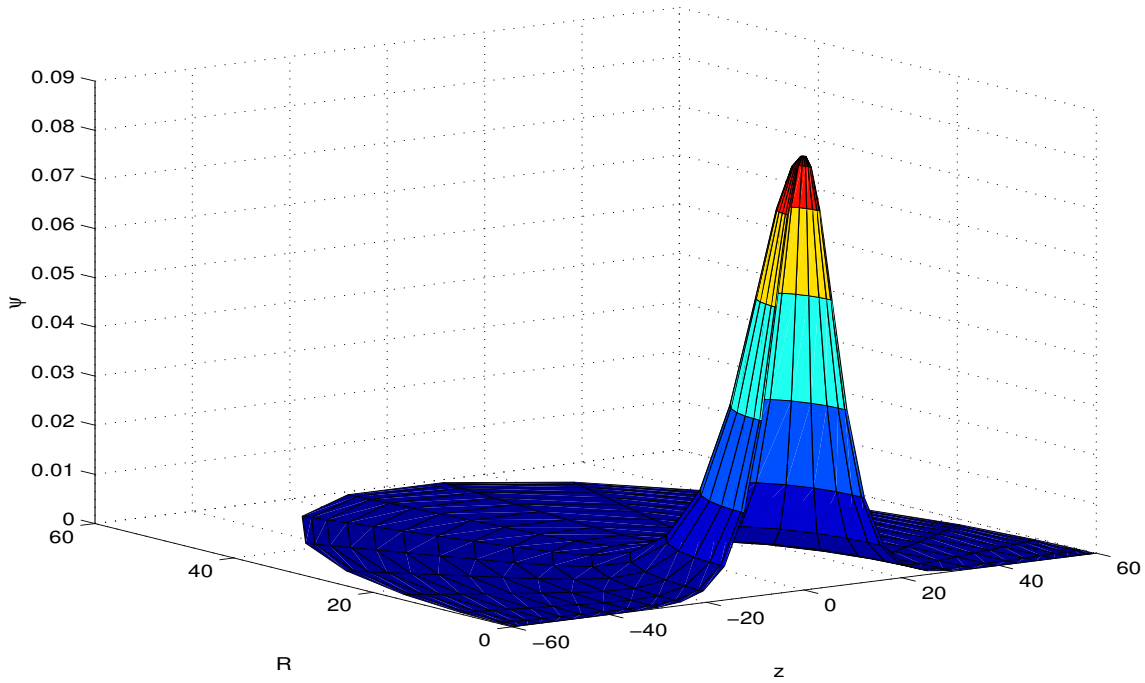


Figure 7.1: Ground state, axp1

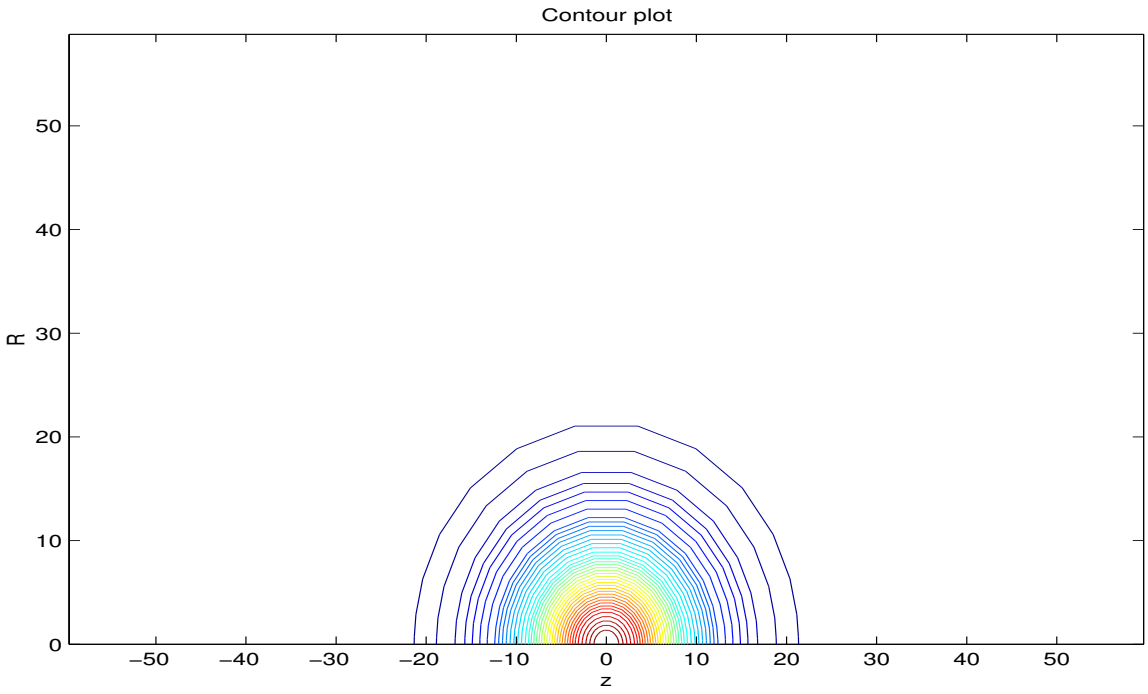


Figure 7.2: Contour plot of the ground state, axp1

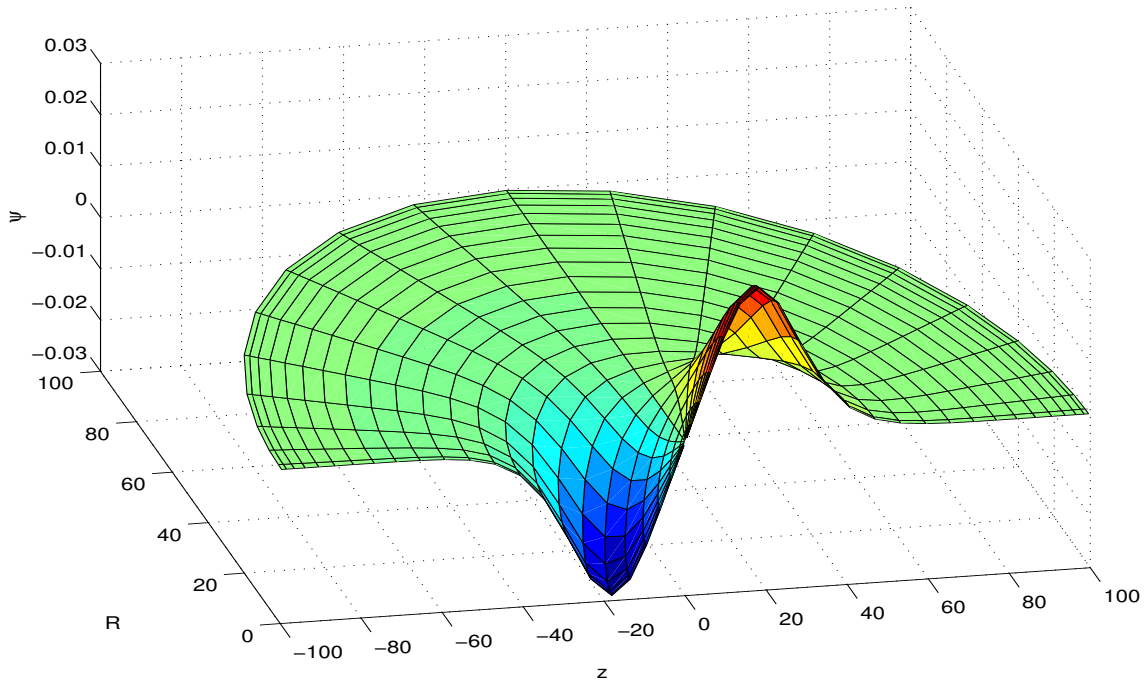


Figure 7.3: "Dipole" state next state after ground state , axp2

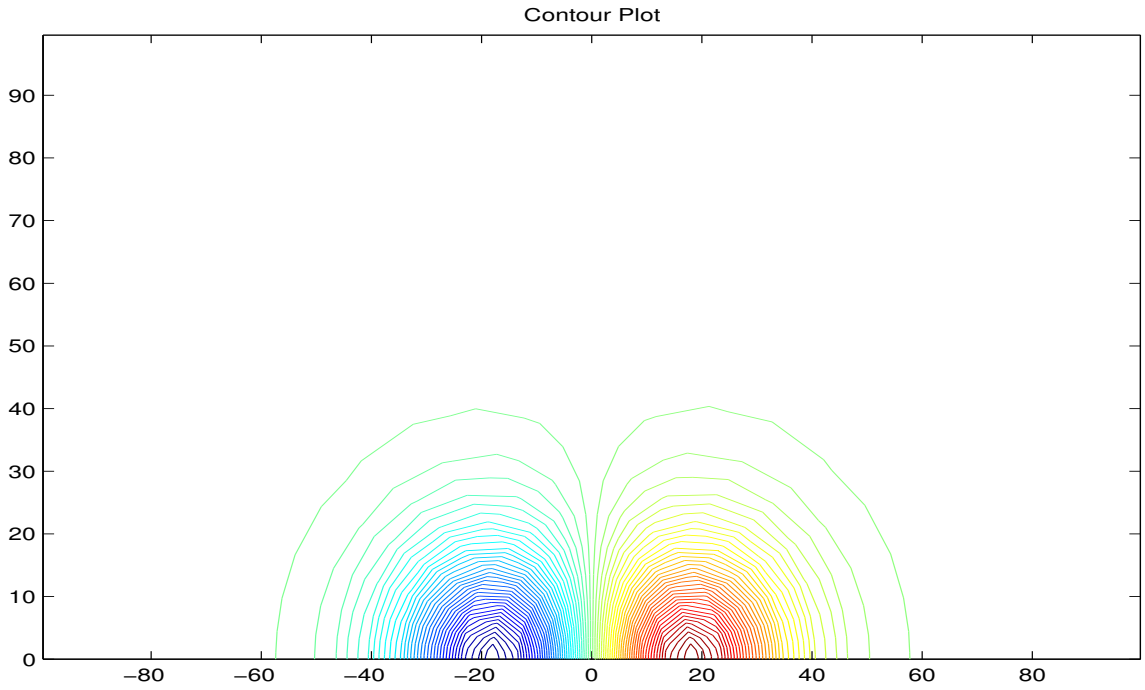


Figure 7.4: Contour Plot of the dipole, axp2

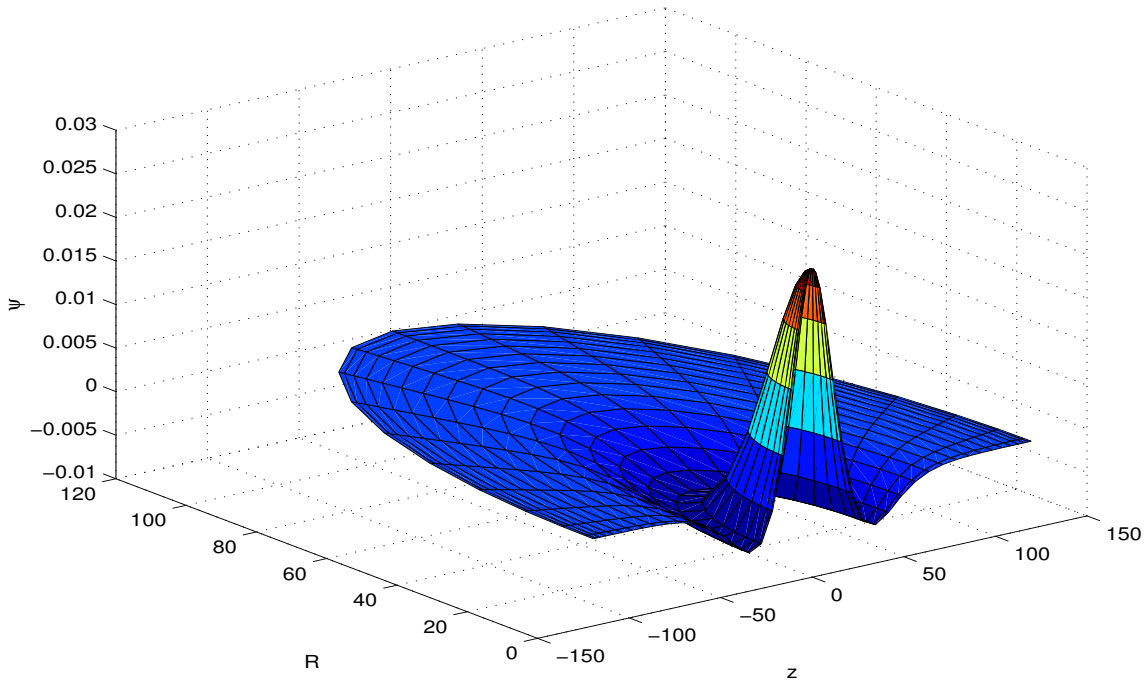


Figure 7.5: Second spherically symmetric state, axp3

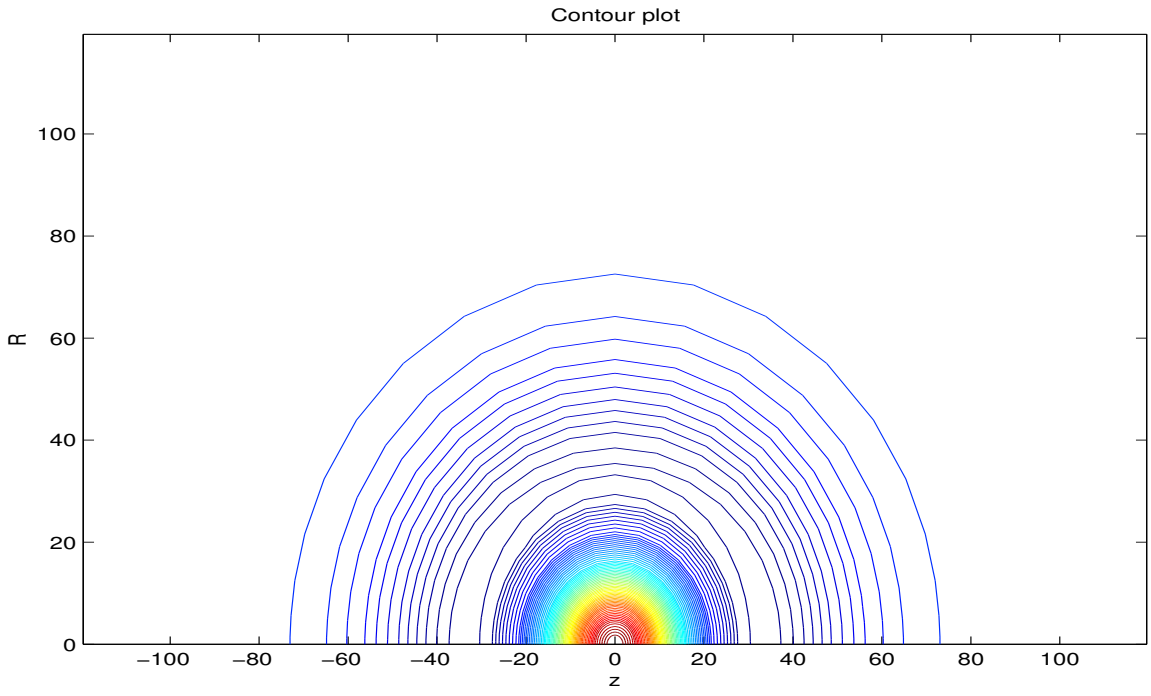


Figure 7.6: Contour Plot of the second spherically symmetric state, axp3

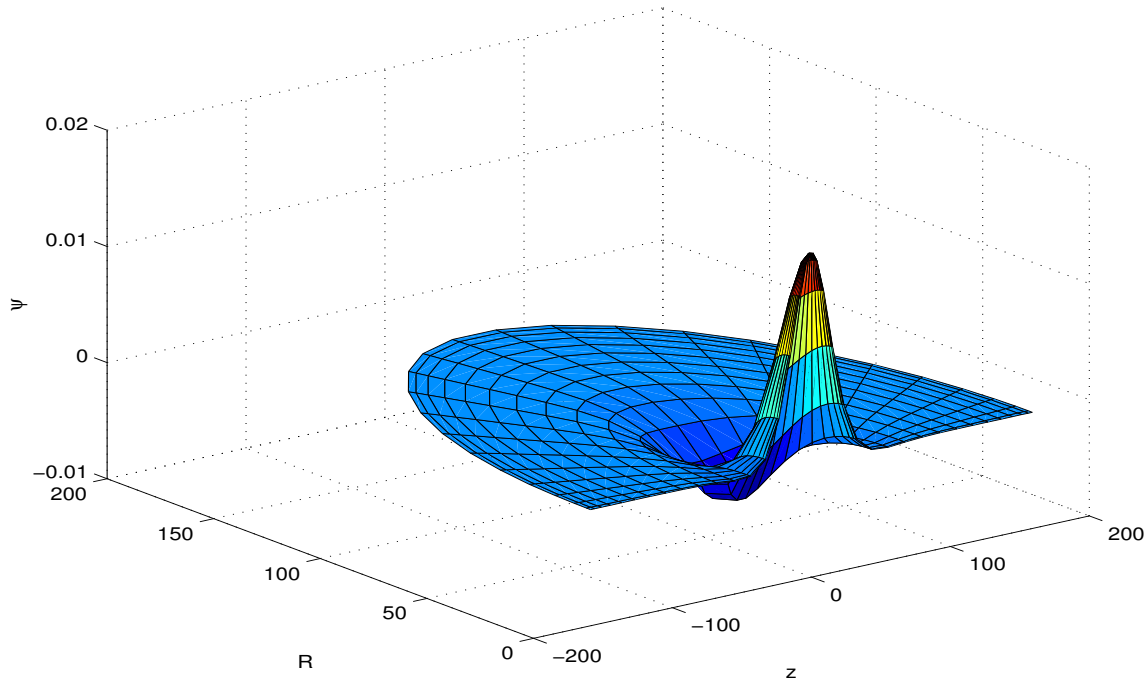


Figure 7.7: Not quite the 2nd spherically symmetric state ,axp4

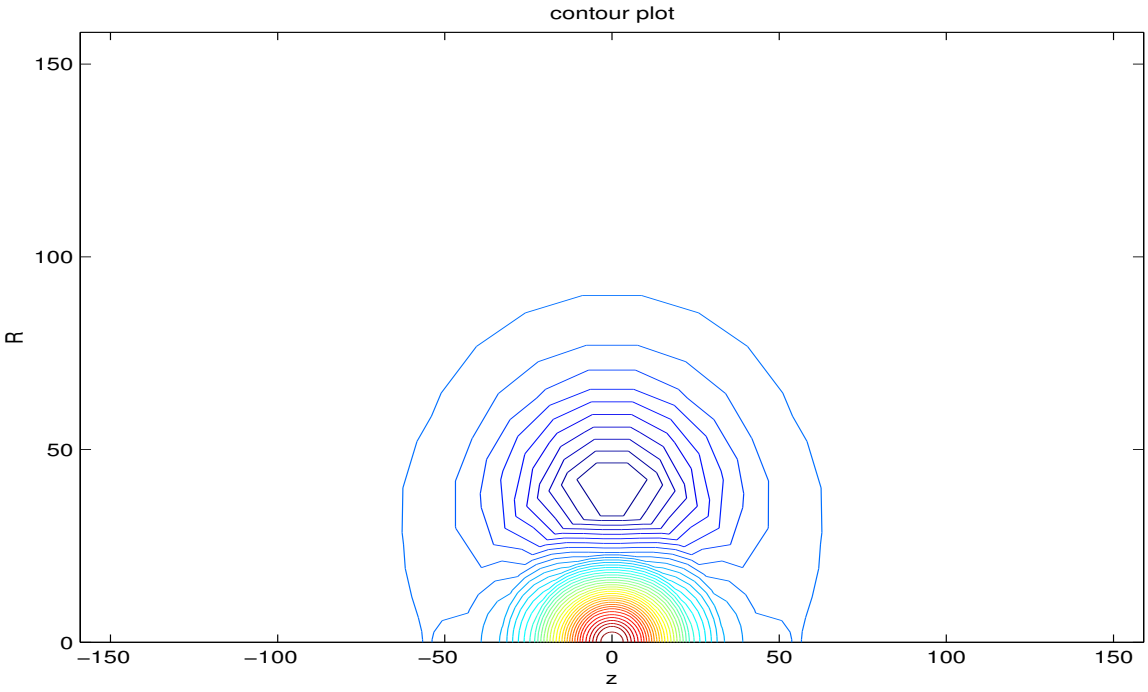


Figure 7.8: Contour plot of the not quite 2nd spherically symmetric state, axp4

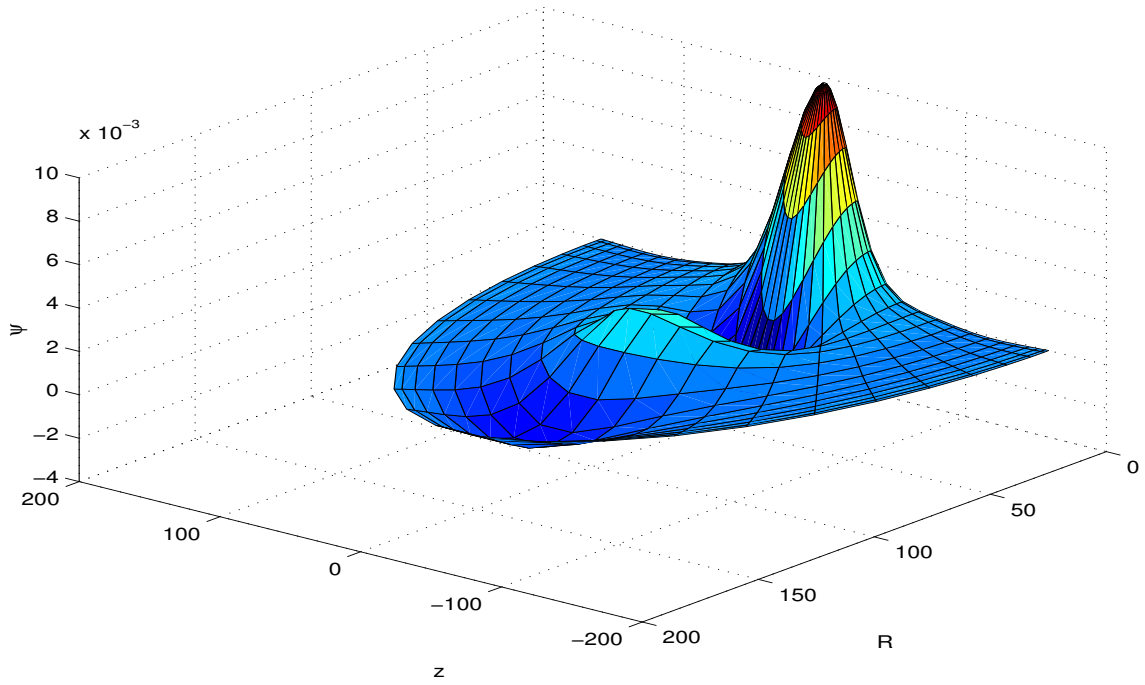


Figure 7.9: Not quite the 3rd spherically symmetric state $E = -0.0162$, axp5

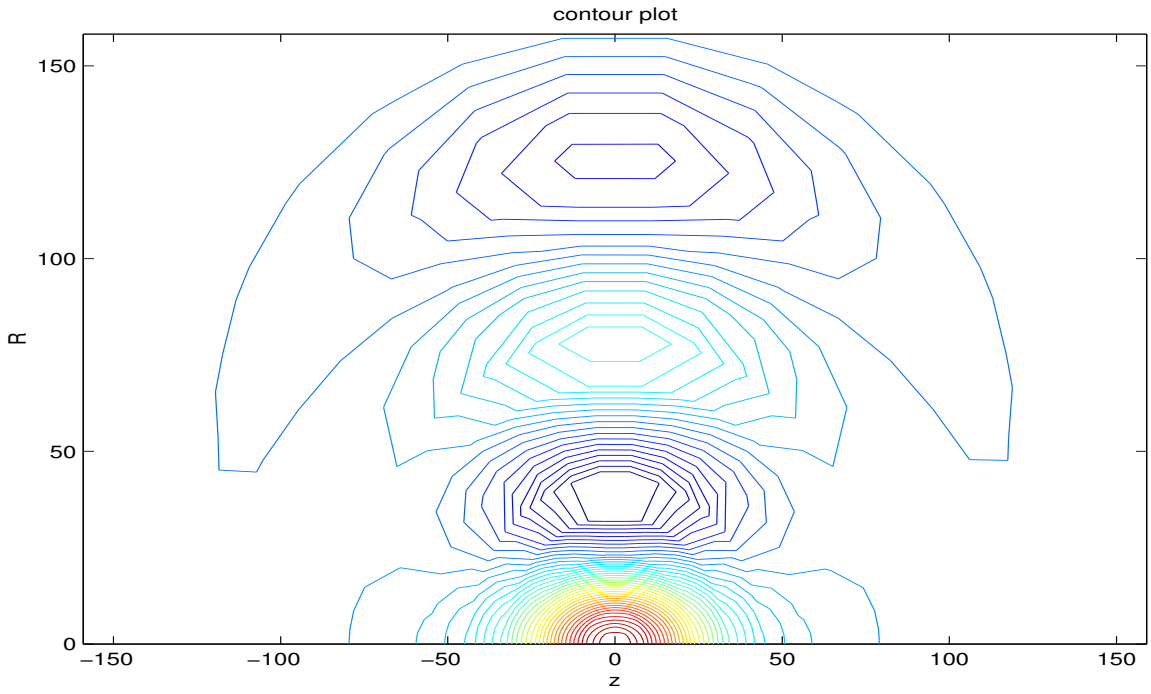


Figure 7.10: Contour plot of Not quite the 3rd spherically symmetric state $E = -0.0162$, axp5

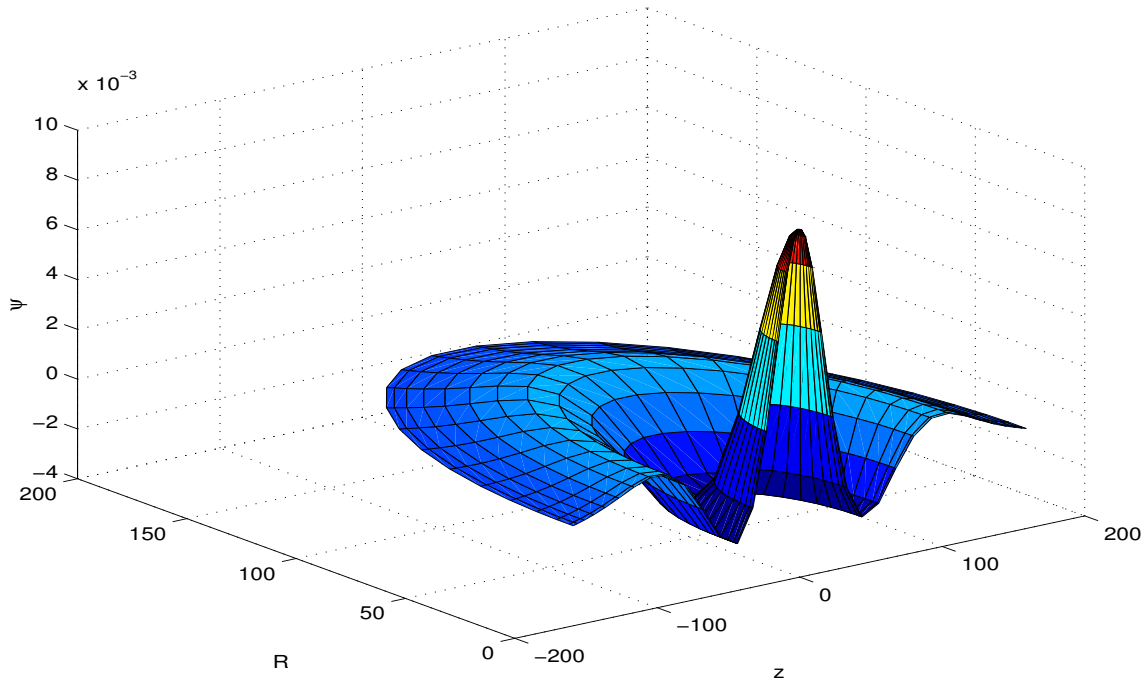


Figure 7.11: 3rd spherically symmetric sate ,axp6

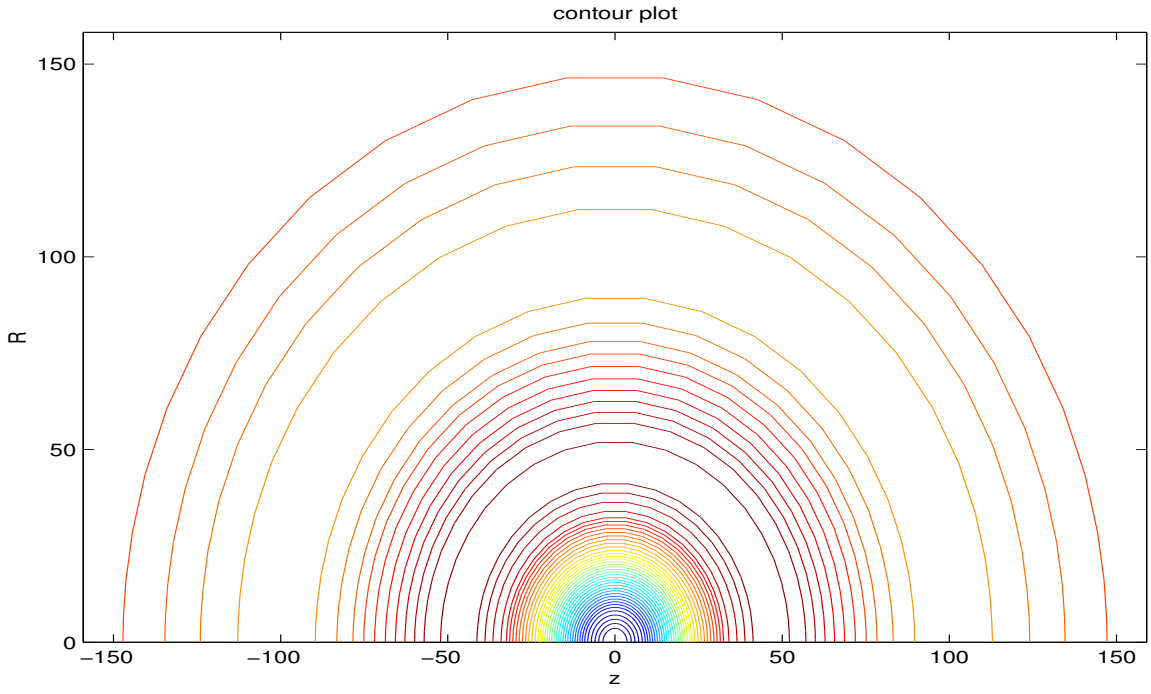


Figure 7.12: Contour plot of 3rd spherically symmetric state ,axp6

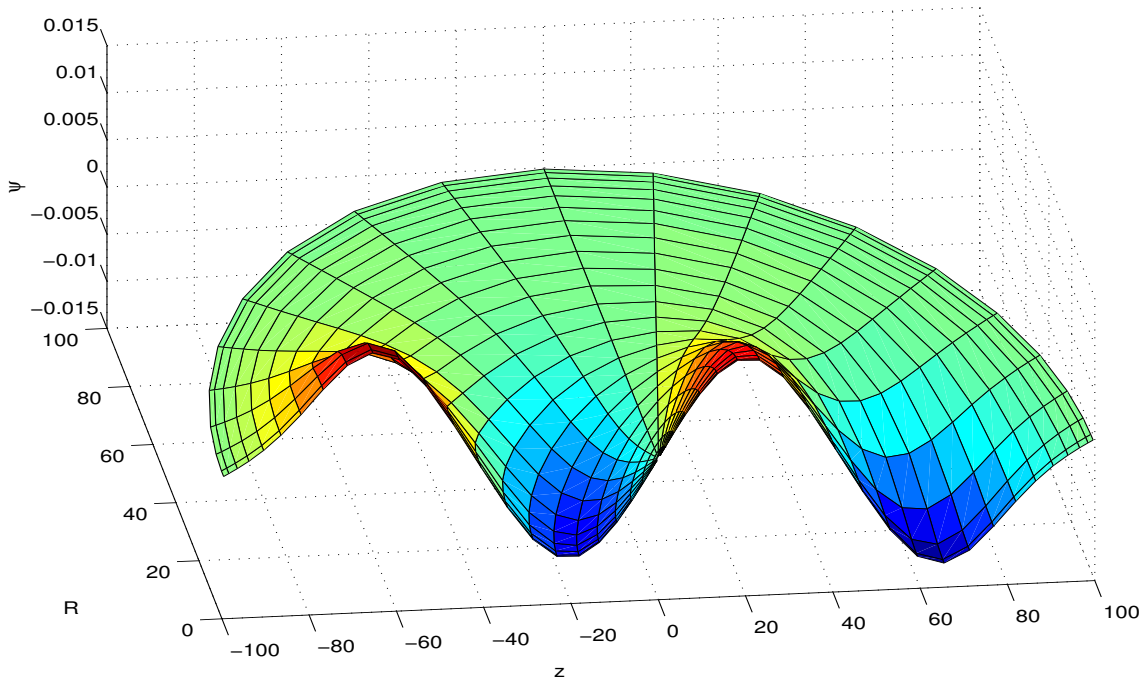


Figure 7.13: Axially symmetric state with $E = -0.0263$, axp7

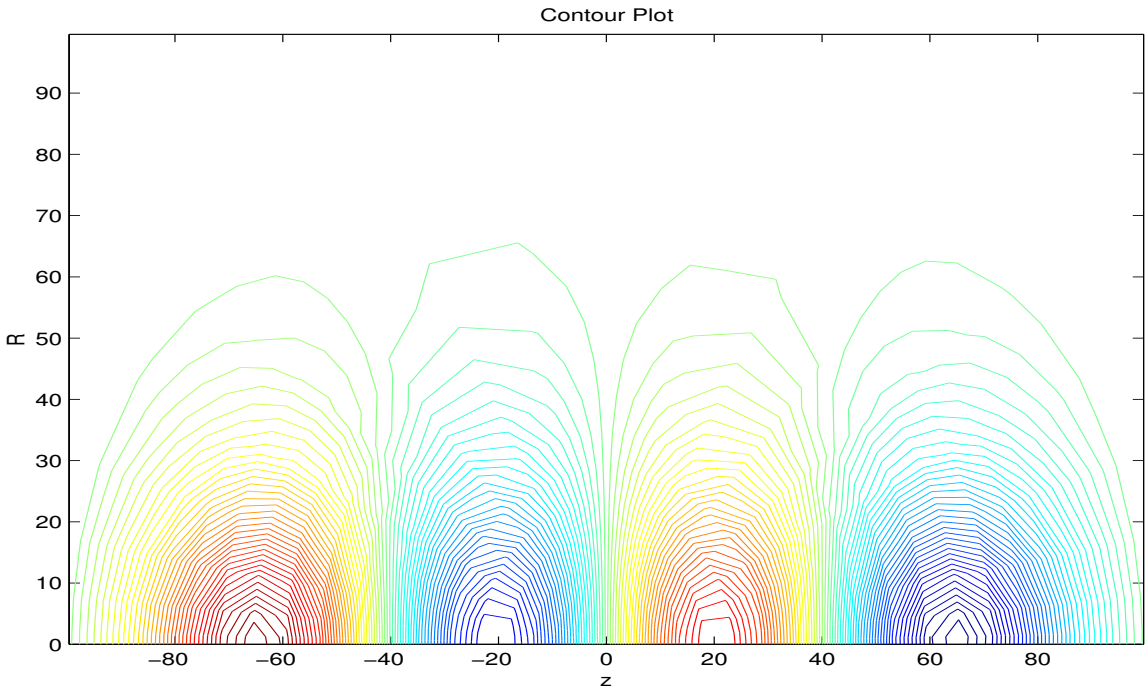


Figure 7.14: Contour plot of axp7 - double dipole

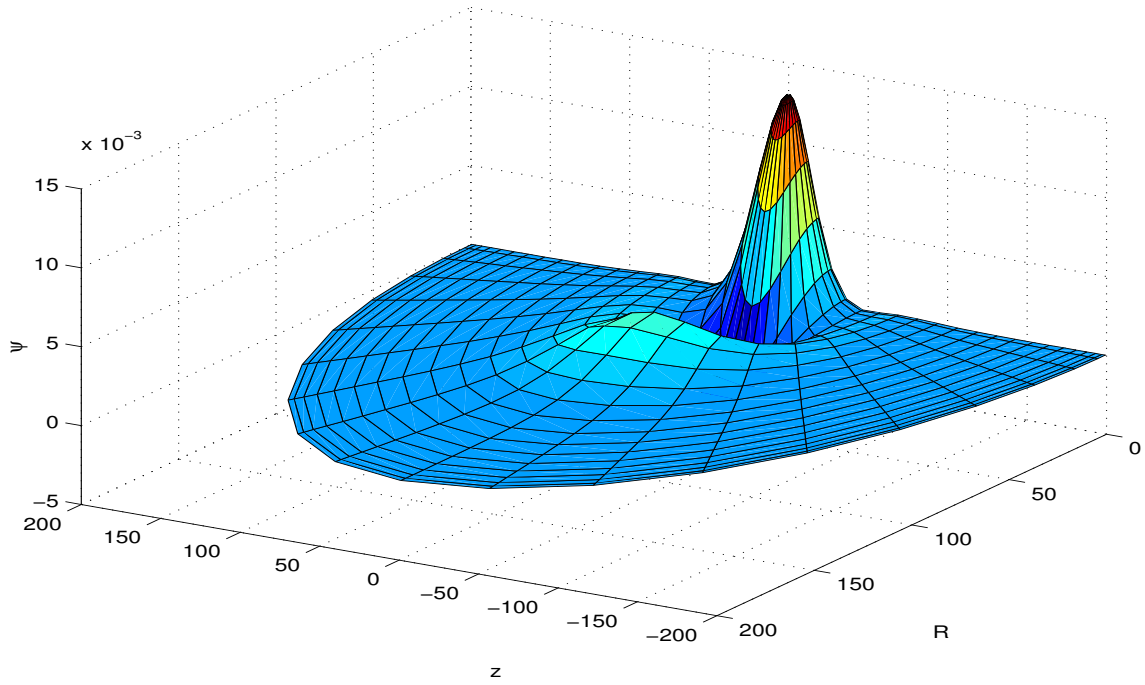


Figure 7.15: State with $E = -0.0208$ and $J = 3.1178$,axp8

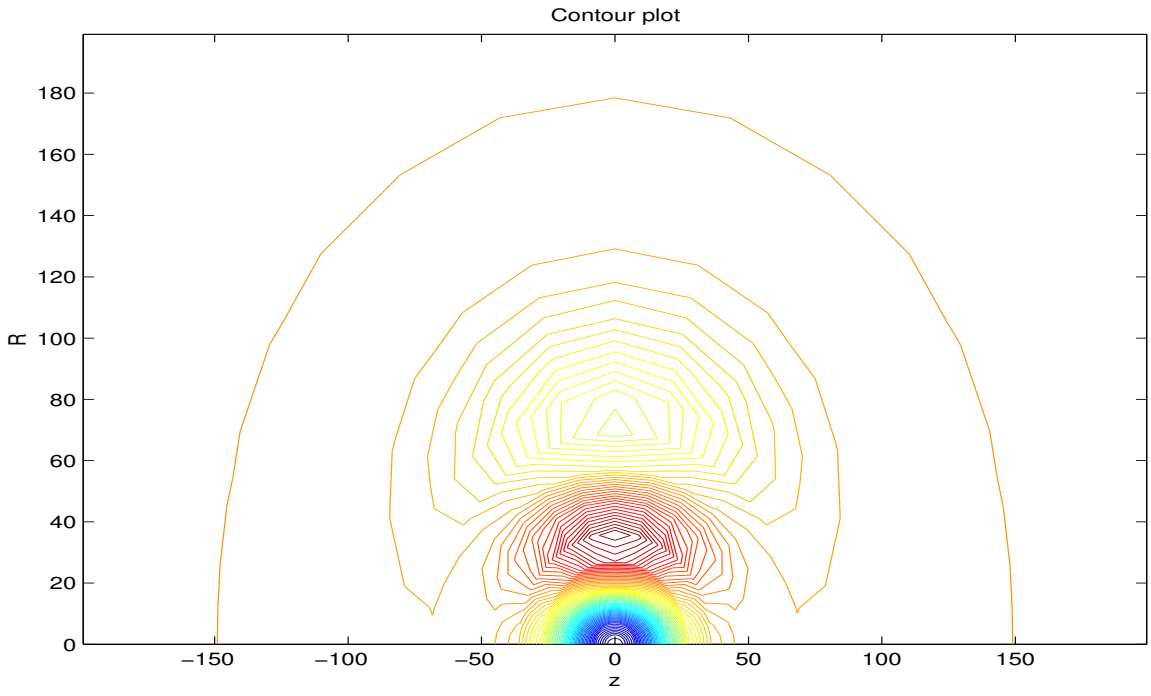


Figure 7.16: Contour plot of the state with $E = -0.0208$ and $J = 3.1178$,axp8

We want to use ADI (alternating direction implicit) method to solve the Schrödinger equation, so we need to split the Laplacian into two parts dependent on each coordinate. We take the operator for the r direction to be:

$$L_1 = \frac{1}{r}\frac{\partial^2}{\partial r^2}, \tag{7.10}$$

which is the r derivative part of the equation which acting on $u = r\psi$. This leaves the remaining operator to be:

$$L_2 = \frac{1}{r}[\frac{\partial^2}{\partial \theta^2} + cot(\theta)\frac{\partial}{\partial \theta}], \tag{7.11}$$

which is the θ derivative part of the equation, and which acts on u. We note that the last operator is not just a function of one variable, but the important part is that it is differential operators in one variable, which is enough to use a ADI(LOD) method to solve the Schrödinger equation as used by Guenther [10]. The ADI scheme is:

$$(1 - iL_1)S = (1 + iL_2)U^n, \tag{7.12a}$$

$$(1 - iL_2)T = (1 + iL_1)S, \tag{7.12b}$$

$$(1 + i\phi^{n+1})U^{n+1} = (1 - i\phi^n)T, \tag{7.12c}$$

Now we also need a similar way to solve the Poisson equation for the potential. Let $w = rV$ then the discretized equation for the potential is:

$$-L_1 w_{i,j} - L_2 w_{i,j} = f_{i,j}, \tag{7.13}$$

where $f_{i,j} = \dfrac{|u_{i,j}|^2}{r_i^2}$ is the density distribution. Rather than solving this equation as it stands (that is $N^2 \times N^2$ matrix problem where N is the number of grid points), we use a Peaceman-Rachford ADI iteration [1] on the equation. Let $\phi^0_{i,j} = 0$ be an initial guess for the solution of the equation and define the iteration to be:

$$-L_1\phi^{n+1}_{i,j} + \rho\phi^{n+1}_{i,j} = \rho\phi^n_{i,j} + L_2\phi^n_{i,j} + f_{i,j}, \tag{7.14a}$$

$$-L_2\phi^{n+2}_{i,j} + \rho\phi^{n+2}_{i,j} = \rho\phi^{n+1}_{i,j} + L_1\phi^{n+1}_{i,j} + f_{i,j}, \tag{7.14b}$$

where ρ is small factor chosen such that the iteration converges. We continue with the iteration until it converges.

We can now evolve the S-N equations using the above method starting with the dipole (first axially symmetric state) and evolving with the same sponge factors as that used in Chapter 6. We find that again the state behaves like that of a stationary state that is evoling which constant phase, before it decays. It decays in the same way as in the

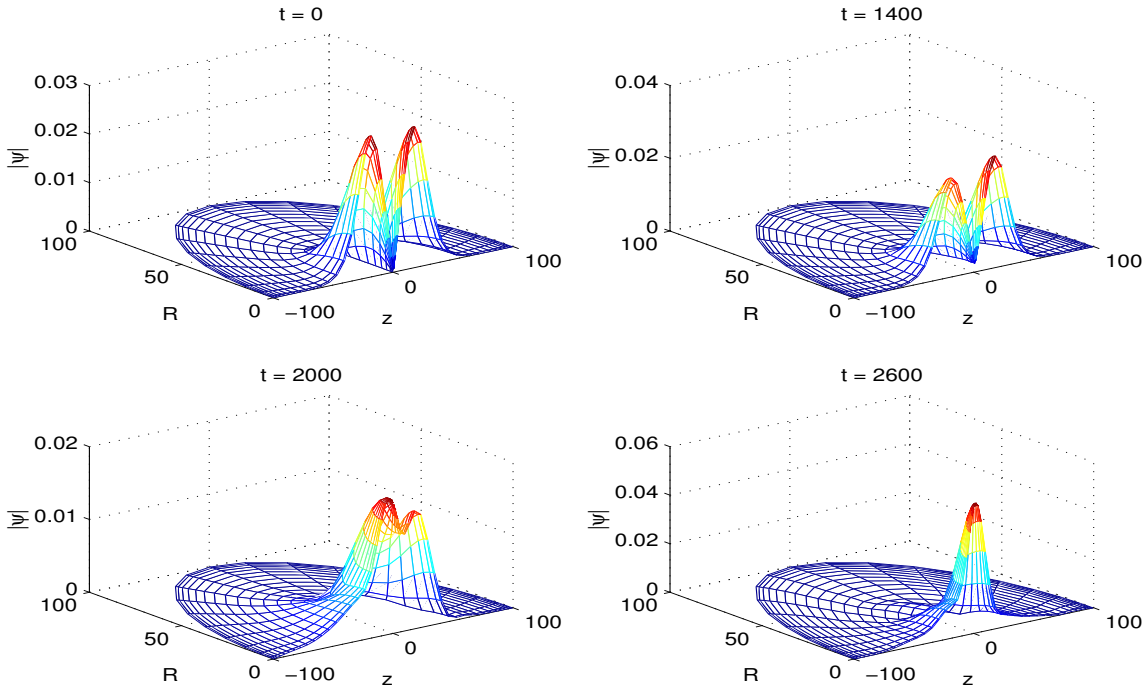


Figure 7.17: Evolution of the dipole

spherically symmetric states emitting scatter off the grid and leaving a "multiple" of the ground state behind. One difference here is that the angular momentum is lost with the wave scattering off the grid, to leave a "multiple" of the ground state. In figure 7.17 we plot the result of the evolution.

In this we can also check the method for convergence, using Richardson fraction in time step. We consider three different time steps for evolution of the dipole, where $dt = 0.25, 0.5, 1$. We plot in figure 7.18 the average Richardson fraction, which if the convergence is quadratic should be 0.2 and if it is linear it should be 0.33. Here we note that the convergence is linear in time. We also plot the Richardson fraction in the case of $dt = 0.125, 0.25, 0.5$ in figure 7.19 and note that again this shows that the convergence is linear in time.

We can also check how the method converges with decreasing the mesh size. In figure 7.20 we plot the evolution of the dipole with different mesh size $N = 30, N2 = 25$ instead of $N = 25, N2 = 20$. Where N is the number of Chebyshev poins in the r direction and $N2$ is the number of point in the θ direction. We note that the state still decays into a "multiple" of the ground state.

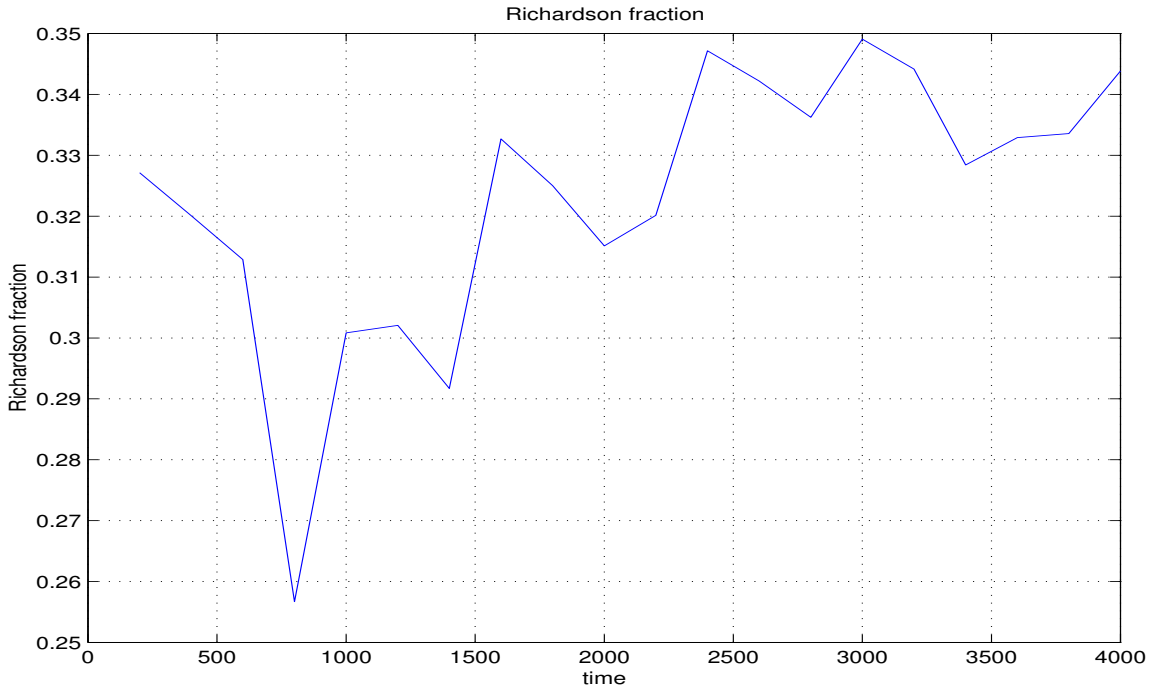


Figure 7.18: Average Richardson fraction with $dt = 0.25, 0.5, 1$

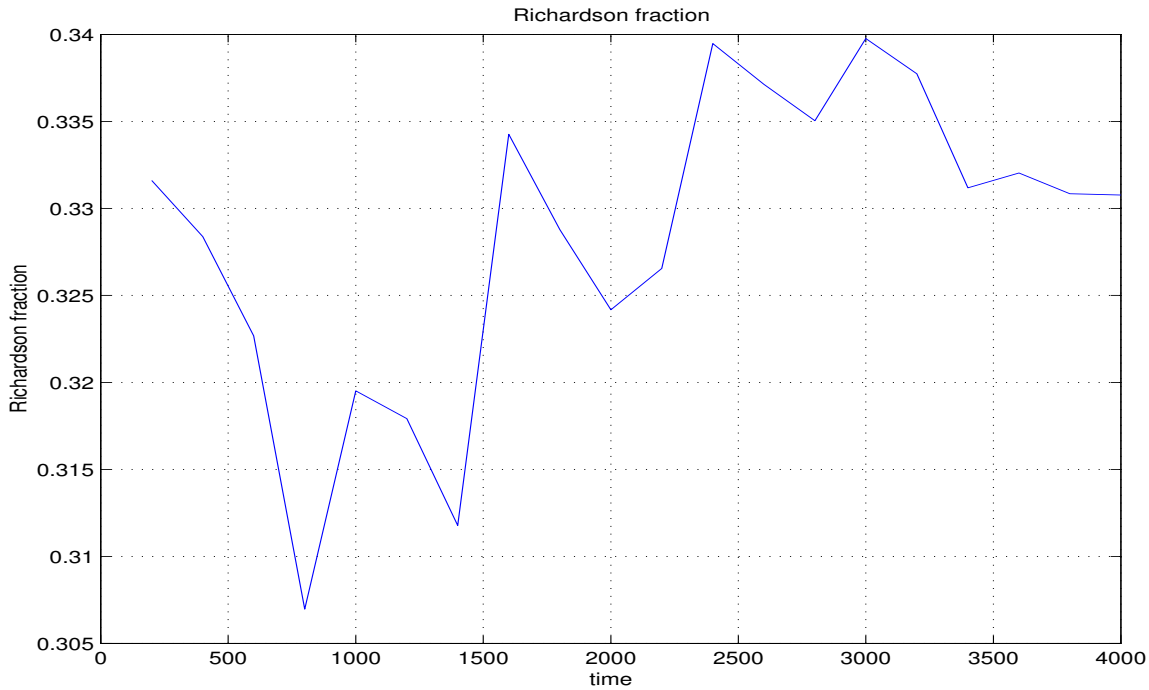


Figure 7.19: Average Richardson fraction with $dt = 0.125, 0.25, 0.5$

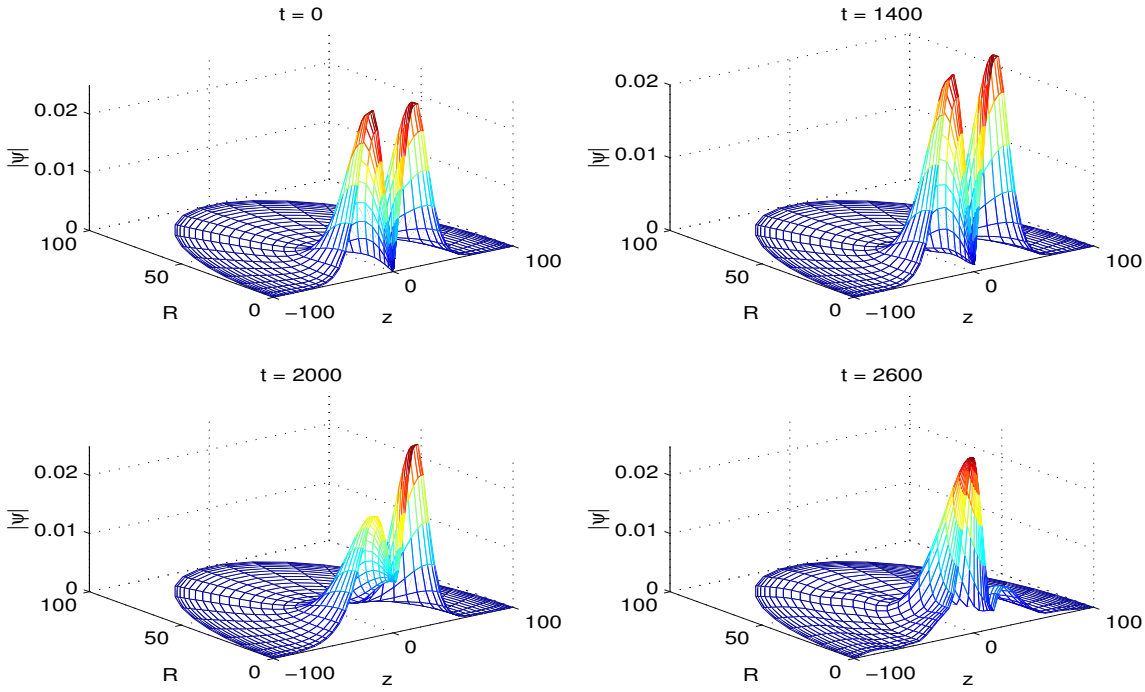


Figure 7.20: Evolution of the dipole with different mesh size

Chapter 8

The Two-Dimensional S-N equations

8.1 The problem

In this chapter, we consider the S-N equations in a plane, that is in Cartesian coordinates x, y. We shall find a dipole-like stationary solution, and some solutions which are like rigidly rotating dipoles. These rigidly rotating solutions are unstable, however, and will merge, radiating angular momentum.

8.2 The equations

Consider the 2-dimensional case, that is, the case where the Laplacian is just

$$\nabla^2\psi = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})\psi, \tag{8.1}$$

with the boundary conditions of zero at the edge of a square, instead of on the sphere. So the non-dimensionalized Schrödinger-Newton equations then in 2D are :

$$i\psi_t = -\psi_{xx} - \psi_{yy} + \phi\psi \tag{8.2a}$$

$$\phi_{xx} + \phi_{yy} = |\psi|^2 \tag{8.2b}$$

To evolve the full non-linear case we require to use the same method as section 5.7, that is, we need a method to solve the Schrödinger equation and Poisson equation. That is, we require a method to solve the Schrödinger equation in 2D as well as the Poisson equation.

To model the Schrödinger equation on this region, we still, as in the case of the one dimension need a numerical scheme that will preserve the normalisation as it evolves.

We can use a Crank-Nicholson method here, but we need to modify this since the matrices become large and no longer tridiagonal in the case of finite differences. Therefore we use an ADI (alternating direction implicit) method to solve the Schrödinger equation.

This works when $V = 0$. The ADI method approximates Crank-Nicolson to the second order.

We can introduce to deal with a non-zero potential an extra term in the scheme. We the same scheme as used in Guenther [10] to intoduce the potential so we have the scheme:

$$(1 - iD2_x)S = (1 + iD2_y)U^n \tag{8.3a}$$

$$(1 - iD2_y)T = (1 + iD2_x)S \tag{8.3b}$$

$$(1 + iV^{n+1})U^{n+1} = (1 - iV^n)T \tag{8.3c}$$

The last equation (8.3c) deals with the potential, and $D2$ is the denotes a Chebyshev differentiation matrix or a finite-difference method for approximating the derivative in the direction indicated, by the subscript. We also need a quicker way to solve the Possion equation in 2D. The discretised version of the Possion equation we want to solve is:

$$-D2_x\phi_{i,j} - D2_y\phi_{i,j} = f_{i,j} \tag{8.4}$$

where $f_{i,j} = |\psi|^2$.

Rather than solving this equation as it stands (that is, $N^2 \times N^2$ matrix problem where N is the number of grid points), we use a Peaceman-Rachford ADI iteration [1] on the equation as described below. Let $\phi^0_{i,j} = 0$ be an initial guess for the solution of the equation and then we define the iteration to be,

$$-D2_x\phi^{n+1}_{i,j} + \rho\phi^{n+1}_{i,j} = \rho\phi^n_{i,j} + D2_y\phi^n_{i,j} + f_{i,j}, \tag{8.5a}$$

$$-D2_y\phi^{n+2}_{i,j} + \rho\phi^{n+2}_{i,j} = \rho\phi^{n+1}_{i,j} + D2_x\phi^{n+1}_{i,j} + f_{i,j}, \tag{8.5b}$$

where ρ is a small number chosen to get convergence. We proceed with the iteration until it converges, that is, $|\phi^{n+2} - \phi^n|$ is less than some tolerance.

8.3 Sponge factors on a square grid

Here we consider sponge factors on the square grid in two dimensions, we choose sponge factor to be

$$e = \min[1, \mathrm{e}^{0.5(\sqrt{(x^2+y^2)}-20)}], \tag{8.6}$$

since we do not want any corners in the sponge factors. We plot a graph of this sponge factor in figure 8.1.

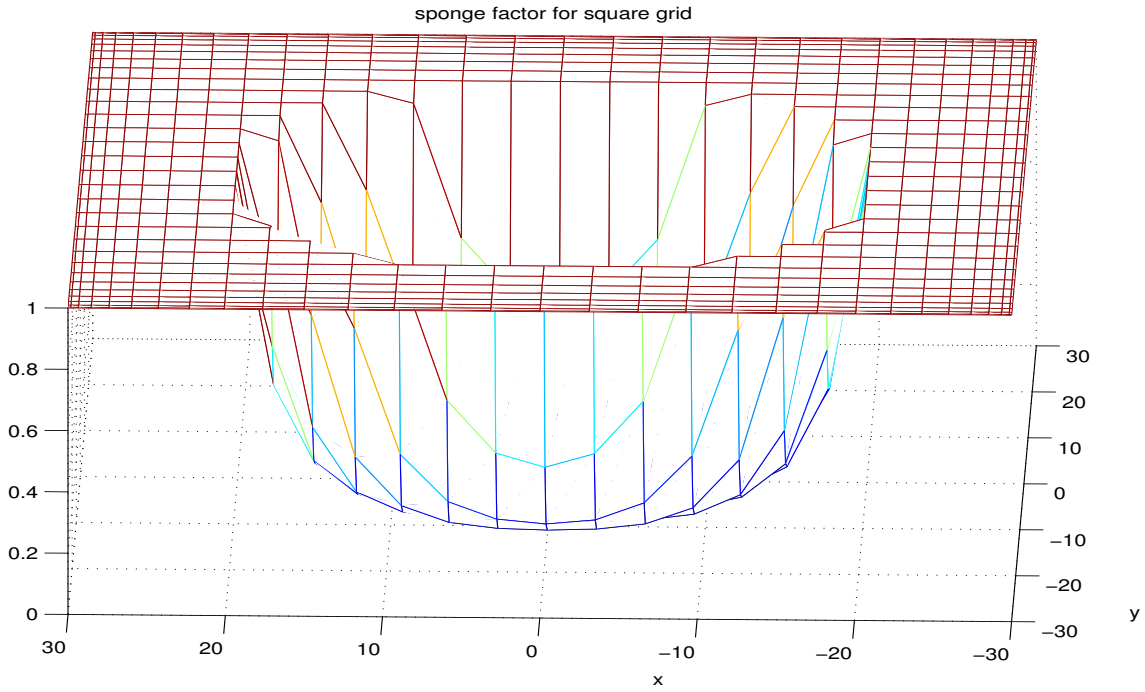


Figure 8.1: Sponge factor

Again to test the sponge factors we send an exponential lump off the grid, where the initial function is of the form:

$$\psi = \mathrm{e}^{-0.01((x-a)^2+(y-b)^2))}\mathrm{e}^{\mathrm{i}vy}, \tag{8.7}$$

a wave-function centred initially at $x = a$ and $y = b$ and moving in the y direction with velocity v. We plot in figures 8.2, 8.3 and 8.4 the lump moving off the grid as a test of the sponge factor. We note that most of the lump is absorbed.

8.4 Evolution of dipole-like state

As in chapter 6 and chapter 7 we can consider now the evolution of a higher state, that is, a state with higher energy than the ground state. We can find such stationary states by using the same method as for axially symmetric state section 7.3, but modified to the 2D case by changing the differential operators. In this case we find that the first stationary state after the ground state has two lumps like that of the dipole (see figure 8.5).

Now we can use the method of section 8.2 to evolve this stationary state with sponge factors. We see that the dipole-like state evolves for a while like a stationary state, that is rotating at constant phase, and then it decays and becomes just one lump, approximately the ground state (see figure 8.6).

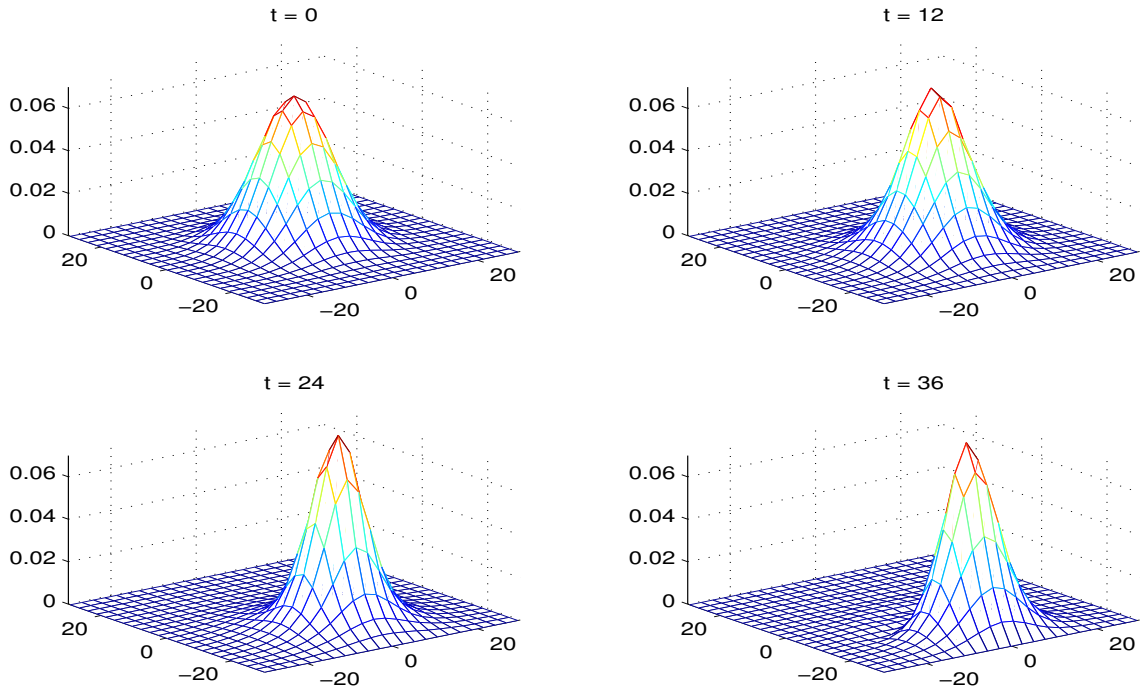


Figure 8.2: Lump moving off the grid with $v = 2$, $N = 30$ and $dt = 2$

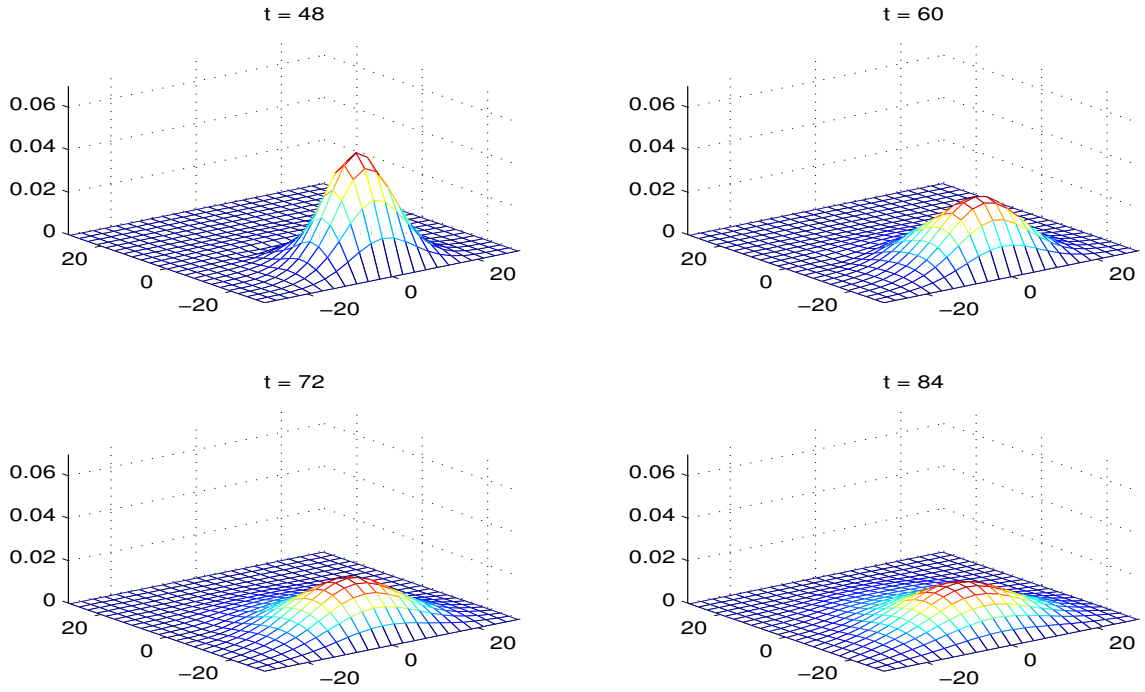


Figure 8.3: Lump moving off the grid with $v = 2$, $N = 30$ and $dt = 2$

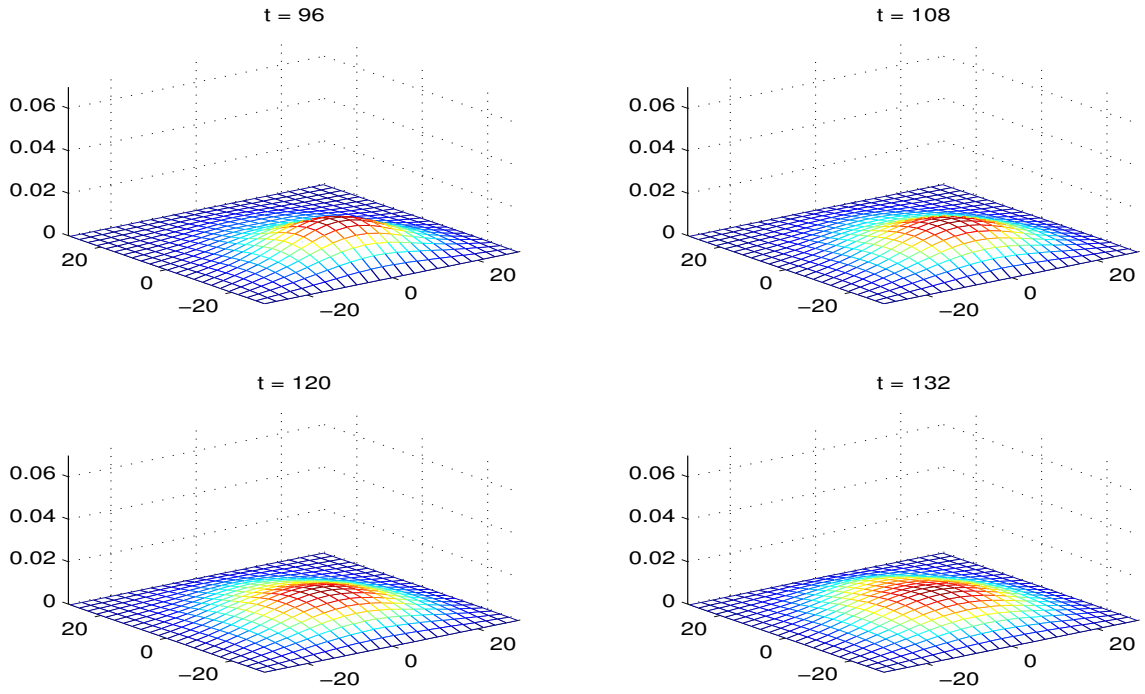


Figure 8.4: Lump moving off the grid with $v = 2$, $N = 30$ and $dt = 2$

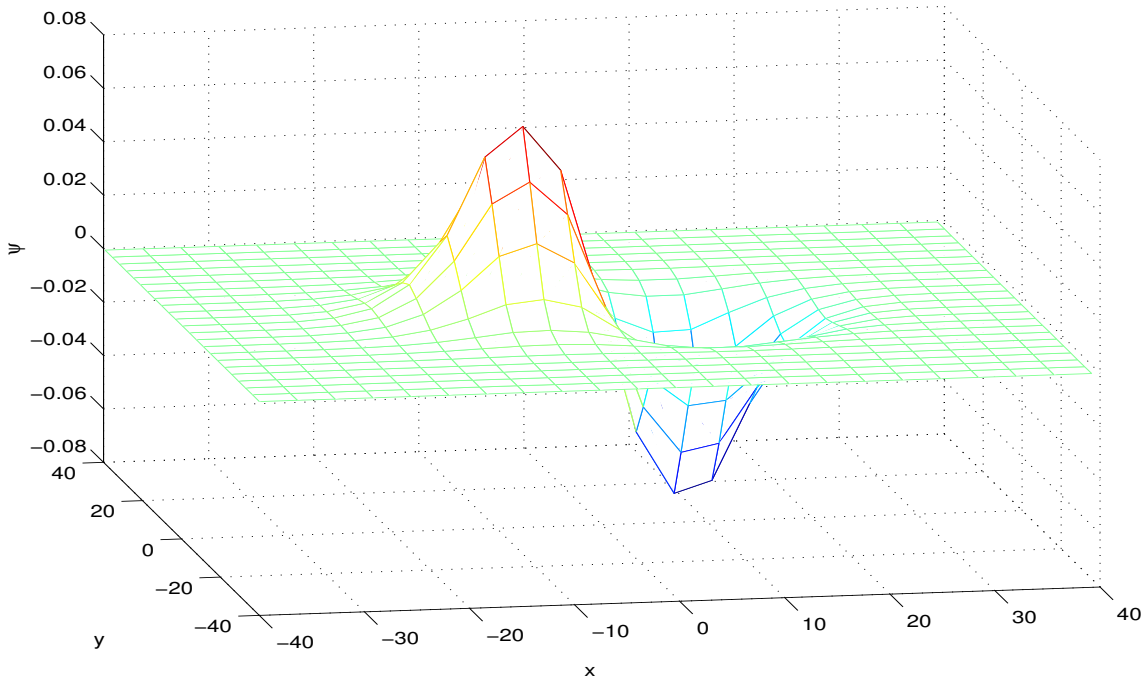


Figure 8.5: Stationary state for the 2-dim S-N equations

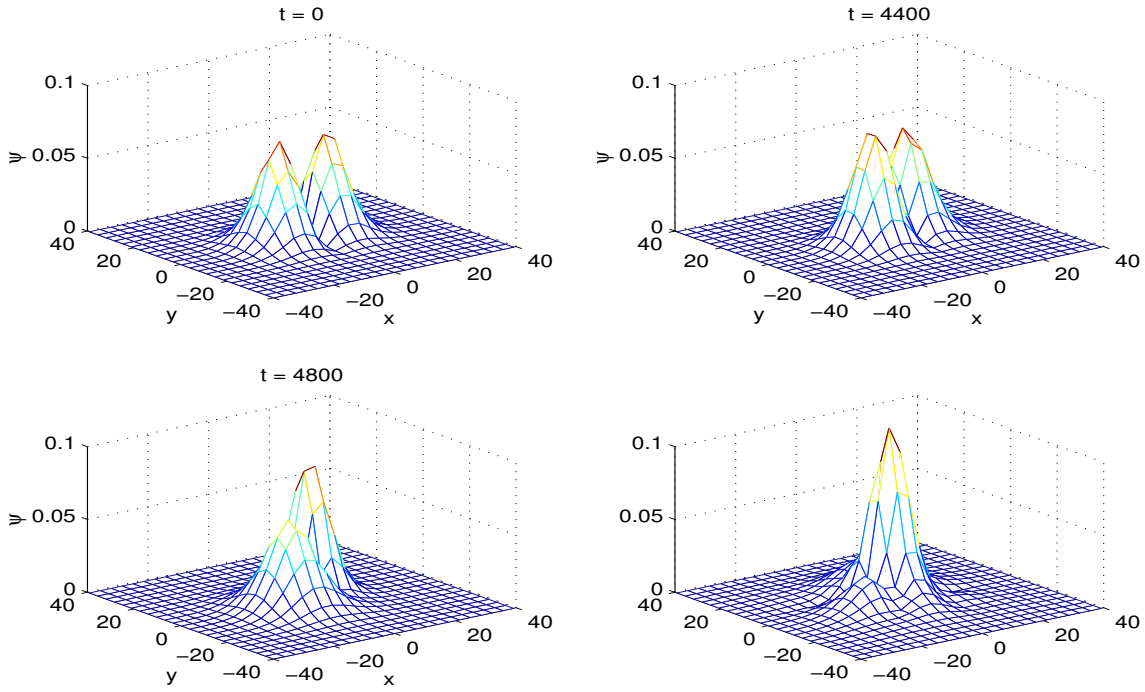


Figure 8.6: Evolution of a stationary state

8.5 Spinning Solution of the two-dimensional equations

To get a spinning solution we could try setting up two ground states or lumps distance apart and set them moving around each other by choosing the initial velocities of the lumps. We could instead try to seek a solution, which satisfies:

$$\psi(r,\theta,t) = \mathrm{e}^{-\mathrm{i}Et}\psi(r,\theta+\omega t,0), \tag{8.8}$$

where ω is a real scalar and r and θ are the polar coordinates. This is redefining the stationary state so that rotating solution are considered. In Cartesian coordinates (8.8) becomes:

$$\psi(x,y,t) = \mathrm{e}^{-\mathrm{i}Et}\psi(x\cos(\omega t)+y\sin(\omega t), -x\sin(\omega t)+y\cos(\omega t),0), \tag{8.9}$$

Now we let,

$$X = x\cos(\omega t)+y\sin(\omega t), \tag{8.10a}$$

$$Y = -x\sin(\omega t)+y\cos(\omega t), \tag{8.10b}$$

which is a rotation of ωt. We note that $\frac{dX}{dt} = Y$ and $\frac{dY}{dt} = -X$.
Then $\mathrm{i}\frac{\partial \psi}{\partial t}$ becomes:

$$\mathrm{i}\frac{\partial \psi}{\partial t} = \mathrm{e}^{-\mathrm{i}Et}[E\psi(X,Y,0) + \mathrm{i}\omega Y\psi_X(X,Y,0) - \mathrm{i}\omega X\psi_Y(X,Y,0)], \tag{8.11}$$

where ψ_X denotes partial differentiation of ψ in the first variable and ψ_Y in the second variable.

So substituting (8.9) into 2-dimensional S-N equations we have:

$$E\psi + \mathrm{i}\omega Y\psi_X - \mathrm{i}\omega X\psi_Y = -\nabla^2\psi + \phi\psi, \tag{8.12a}$$

$$\nabla^2\phi = |\psi|^2, \tag{8.12b}$$

we note that these equations are dependent on X and Y only. Note that $\bar{\psi}$ is a solution with $-\omega$ instead of ω, which corresponds to a solution rotating in the other direction. If ψ is a function of r then $[Y\psi_x - X\psi_y]$ vanishes and we get the ordinary stationary states in that case regardless of what value ω is.

We can modify the methods used to calculate the stationary state solution in the axially symmetric case (section 7.3), to calculate the numerical solution of (8.12) at a fixed ω. We note that $\psi(X,Y,0)$ is no longer going to be of just one phase.

Do non-trivial solutions of these equations exist that is a solution such that $[Y\psi_x - X\psi_y]$ does not vanish, hence a solution that will spin? We can find such a solution with $\omega = 0.005$ which we plot the real part in figure 8.7 and the imaginary part in figure 8.8.

We can use the time-dependent evolution method as in section 8.2, with the initial data being a spinning solution. In figure 8.9 we plot the evolution of the state before it decays showing that the state does spin about $x = 0$ and $y = 0$. In figure 8.10 we plot the section of the evolution showing that the state does decay, emitting scatter off the grid and leaving a "multiple" of the ground state.

8.6 Conclusion

For the 2-dimensional S-N equations we can conclude that:

- They behave as the spherically symmetric and axisymmetric equations that is the ground state is stable and the higher states are unstable.
- Spinning solutions do exist and they decay like the other solutions to a "multiple" of the ground state.

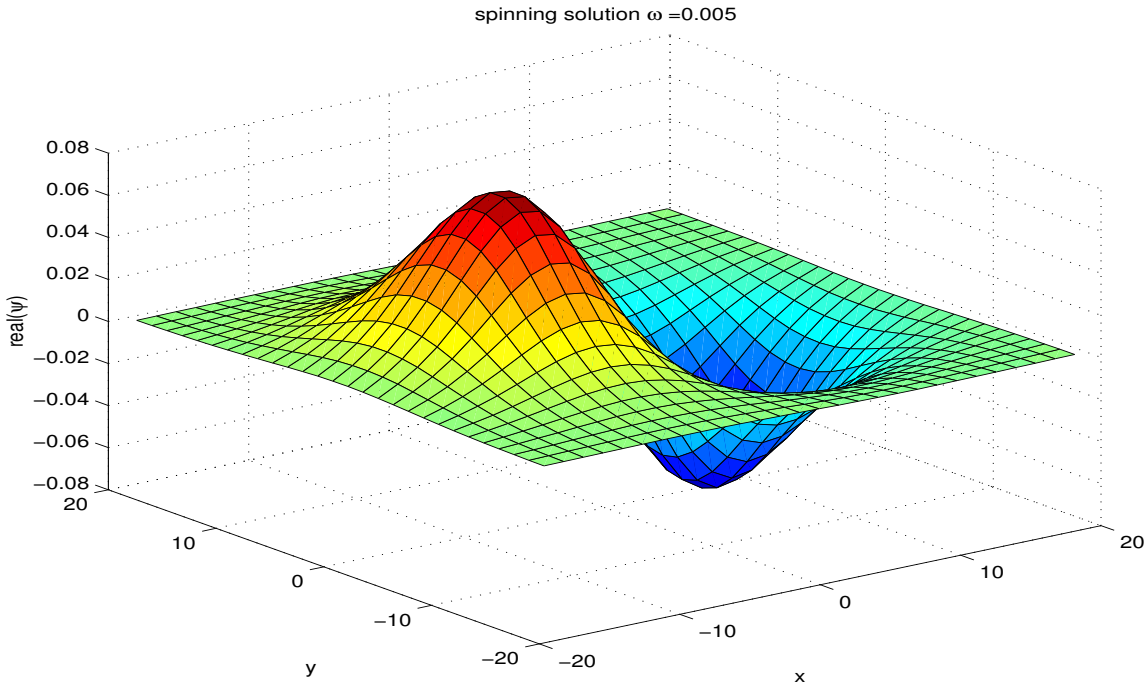


Figure 8.7: Real part of spinning solution

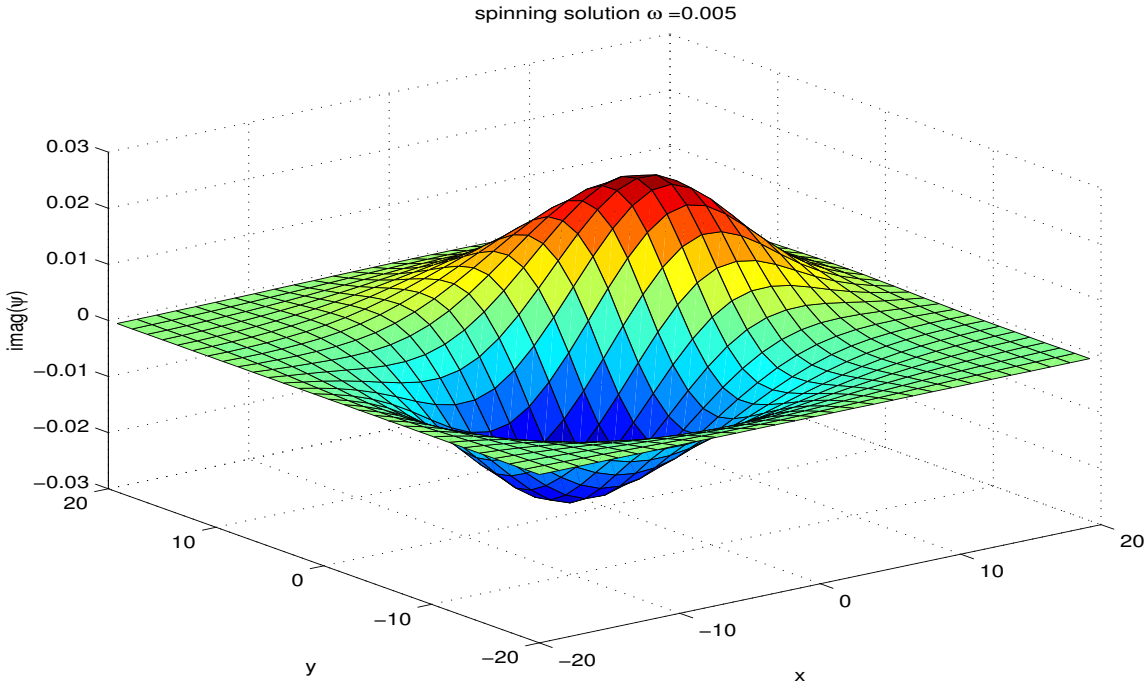


Figure 8.8: Imaginary part of spinning solution

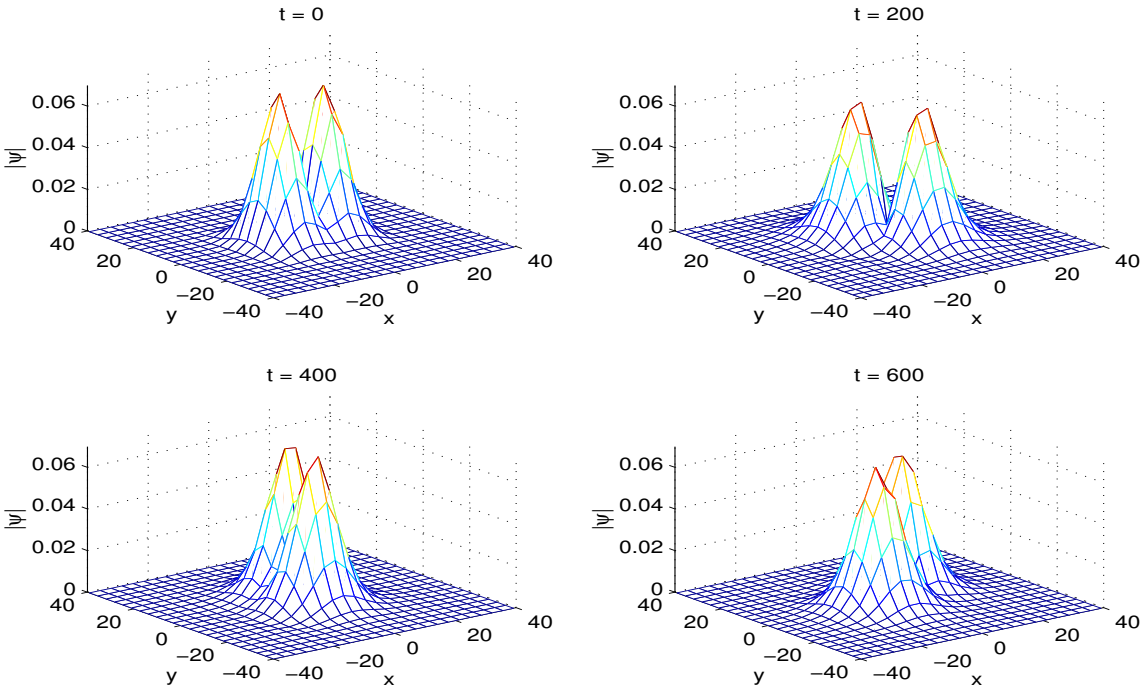


Figure 8.9: Evolution of spinning solution

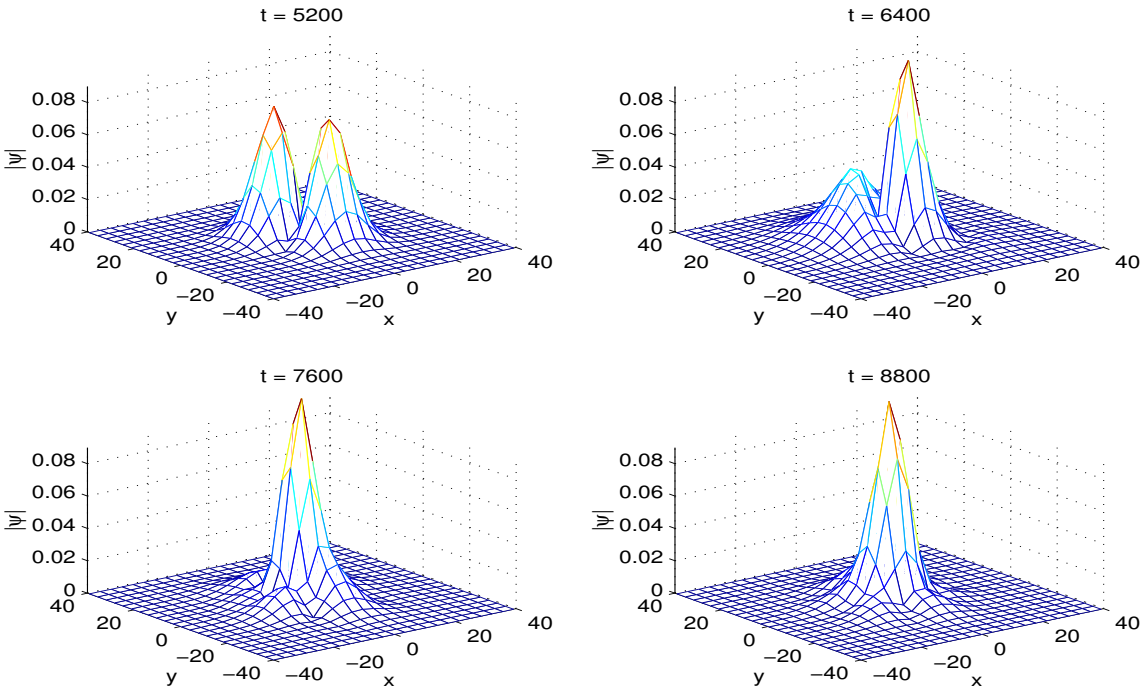


Figure 8.10: Evolution of spinning solution

Bibliography

[1] W.F.Ames, Numerical Methods for Partial Differential Equations 2nd ed (1977)

[2] E.R.Arriola, J.Soler, Asymptotic behaviour for the 3-D Schrödinger-Poisson system in the attractive case with positive energy App.Math.Lett.**12** (1999) 1-6

[3] D.H.Bernstein, E.Giladi, K.R.W.Jones, Eigenstate of the Gravitational Schrödinger Equation. Modern Physics Letters **A13** (1998) 2327-2336

[4] D.H.Bernstein, K.R.W.Jones, Dynamics of the Self-Gravitating Bose-Einstein Condensate. (Draft)

[5] J.Christian, Exactly soluble sector of quantum gravity Phys.Rev. D56 (1997) 4844.

[6] L.Diosi, Gravitation and the quantum-mechanical localization of macro-objects. Phys.Lett **105A** (1984) 199-202

[7] L.Diosi, Permanent State Reduction: Motivations, Results and By-Products. Quantum Communications and Measurement. Plenum (1995) 245-250

[8] L.C.Evans, "Partial differential equations" AMS Graduate Studies in Mathematics **19** AMS (1998)

[9] A.Goldbery, H.Schey, J.Schwartz, Computer-generated motion pictures of one-dimensional quantum mechanical transmission and reflection phenomena Am.J.Phys **35** 3 (1967) 177-186

[10] R.Guenther, "A numerical study of the time dependent Schrödinger equation coupled with Newtonian gravity" Doctor of Philosophy thesis for University of Texas at Austin (1995).

[11] H.Lange, B.Toomire, P.F.Zweifel, An overview of Schrödinger-Poisson Problems. Reports on Mathematical Physics **36** (1995) 331-345

[12] H.Lange, B.Toomire, P.F.Zweifel, Time-dependent dissipation in nonlinear Schrödinger systems **36** (1995) 1274-1283 Journal of Mathematical Physics.

[13] R.Illner, P.F.Zwiefel, H.Lange, Global existence, uniqueness and asymptotic behaviour of solutions of the Wigner-Poisson and Schödinger-Poisson systems. Math.Meth in App.Sci.**17** (1994) 349-376

[14] I.M.Moroz, K.P.Tod, An Analytical Approach to the Schrödinger-Newton equations. Nonlinearity **12** (1999) 201-16

[15] I.M.Moroz, R.Penrose, K.P.Tod, Spherically-symmetric solutions of the Schrödinger-Newton equations. Class Quantum Grav. **15** (1998) 2733-2742

[16] K.W.Morton, D.F.Mayers, Numerical Solution of Partial Differential Equations (1994)

[17] R.Penrose, On gravity's role in quantum state reduction. Gen.Rel.Grav. **28** (1996) 581-600

[18] R.Penrose, Quantum computation, entanglement and state reduction Phil.Trans.R.Soc. (Lond) **A 356** (1998) 1927

[19] D.L.Powers, Boundary Value Problems, Academic Press 1972.

[20] W.Press, S.Teukolsky, W.Vetterting, B.Flannery Numerical Recipes in Fortran (second edition)

[21] R.Ruffini, S.Bonazzola, Systems of Self-Gravitating Particles in General Relativity and the concept of an Equation of State. Phys.Rev.**187** (1969) 1767

[22] B.Schupp, J.J. van der Bij, An axially-symmetric Newtonian boson star. Physics Letters B 366 (1996) 85-88

[23] H.Schmidt, Is there a gravitational collapse of the wave-packet ? (preprint)

[24] H.Stephani, "Differential equations: their solution using symmetries" Cambridge: CUP (1989)

[25] K.P.Tod, The ground state energy of the Schrödinger-Newton equations Phys.Lett.**A 280** (2001) 173-176

[26] Private correspondence with K.P.Tod.

[27] L.N.Trefethen, Spectral Methods in MATLAB, SIAM 2000.

[28] L.N.Trefethen, Lectures on Spectral Methods.(Oxford Lecture Course)

Appendix A

Fortran programs

A.1 Program to calculate the bound states for the S-N equations

```
! integrates the spherically symmetric Schrodinger-Newton equations
! Integrates the system of 4 coupled equations
! dX/dt = FN(X,Y)
! dY/dt = FN(X,Y)
!
! using the NAG library routine D02PVF D02PCF
! using the NAG library routines X02AJF X02AMF
!
! Program searchs for bound states and the values of
!  r for which they blow up
!
      Program SN
      Implicit None
      External D02PVF
      External D02PCF
      External OUTPUT
      External X02AJF
      External X02AMF
      External FCN
      External COUNT
      Double precision, dimension(4) :: Y,YP,YMAX,THRES,YS
      Double precision, dimension(4) :: PY,PY2
      Double Precision, dimension(32*4) :: W
      Double Precision :: RATIO,INSIZE,Y1P,INT1,INT2
      Double Precision  ::  X,XEND,STEP,TOL,XE,HSTART
      Double Precision :: X02AJF,X02AMF
      Double Precision :: Y1,STEP2,XBLOW,XBLOWP
      INTEGER :: DR,DR2,DOUT,GOUT,ENCO,ENCO2
      INTEGER :: N, I,IFAIL,METHOD , LENWRK
      CHARACTER*1 :: TASK
      Logical :: ERRAS
```

```
! compute a few things !
        N = 4
        INSIZE = 0.01
        Y1P = 1.2
        X = 0.0000001
        XEND = 2000
        XE = X
        TOL = 100000000000*X02AJF()
        THRES(1:4) = Sqrt(X02AMF())
        METHOD = 2
        TASK ='U'
        ERRAS = .FALSE.
        HSTART = 0.0
        LENWRK = 32*4
        IFAIL = 0
        GOUT = 0
! Ask for user input .
! to find root below given value
!
        Y(3) = 1.0
        Y(2) = 0.0
        Y(4) = 0.0
! Ask for upper value
        Print*, " TOL ", TOL
        Print*, " Enter upper value "
        Y1 = 1.09
        Y(1) = Y1
        PY(1:4) = Y(1:4)
        PY2(1:4) = Y(1:4)
! Step size
        STEP = 0.0005
        OPEN(Unit='10',file='fort.10',Status='new')
        OPEN(Unit='11',file='fort.11',Status='new')
DO WHILE ( GOUT == 0 )
        DR = 0
!
! D02PVF is initially called to give initial starting values to
! subroutine ,
! One has too determine the initial direction in which our
! first value blows up
        Step2 = INSIZE ! Initial trial step in Y(1)
        DR2 = 1 ! direction of advance in stepping
! DR is direction of blow up
CALL D02PVF(N,X,Y,XEND,TOL,THRES,METHOD,TASK,ERRAS,HSTART,W,LENWRK,IFAIL)
   DO WHILE (DR == 0)
        XE = XE + STEP
        PY2(1:4) = PY(1:4)
        PY(1:4) = Y(1:4)
```

```
        IF (XE < XEND) THEN
        CALL D02PCF(FCN,XE,X,Y,YP,YMAX,W,IFAIL)
        END IF
        IF (XE > 3*STEP) THEN
        RATIO = (XE-2*STEP)*PY2(1)/((XE-STEP)*PY(1))
        RATIO = RATIO - (XE-STEP)*PY(1)/(XE*Y(1))
        IF ((RATIO*RATIO) < 1E-15) THEN
        RATIO = (XE-STEP)*PY(1)/(XE*Y(1))
        IF (RATIO < 1) THEN
           IF (RATIO > -1) THEN
           IF (Y(1) > 0) THEN
           DR = 1
           END IF
           IF (Y(1) < 0) THEN
           DR = - 1
           END IF
        END IF
        END IF
        END IF
        END IF
        IF (Y(1) > 2) THEN
           DR = 1
        ENDIF
        IF (Y(1) < -2) THEN
           DR = -1
        ENDIF
        IF (XE > XEND) THEN
           DR = 10
        ENDIF
   ENDDO
! Do while is ended ...
        XBLOWP = 0
        XBLOW = 0
   DO WHILE (DR /= 10)
        Y1 = Y1-STEP2*REAL(DR2)
        !Print*, "Y(1) ", Y1
        Y(3) = 1.0
        Y(2) = 0.0
        Y(1) = Y1
        Y(4) = 0.0
        XBLOWP = XBLOW
        XBLOW = 0
        ENCO = 0
        ENCO2 = 0
        DOUT = 0
        X = 0.0000001
        HSTART = 0.0
        XE = X
```

```
CALL D02PVF(N,X,Y,XEND,TOL,THRES,METHOD,TASK,ERRAS,HSTART,W,LENWRK,IFAIL)
DO WHILE (DOUT == 0)
     XE = XE + STEP
     YS = Y
     PY2(1:4) = PY(1:4)
     PY(1:4) = Y(1:4)
     IF (XE < XEND) THEN
     CALL D02PCF(FCN,XE,X,Y,YP,YMAX,W,IFAIL)
     END IF
     CALL COUNT(YS,Y,ENCO,ENCO2)
     IF (Y(1) < 0.0001) THEN
        IF (Y(1) > -0.0001) THEN
           XBLOW = XE
        ENDIF
     ENDIF
     IF (XE > 3*STEP) THEN
     RATIO = (XE-2*STEP)*PY2(1)/((XE-STEP)*PY(1))
     RATIO = RATIO - (XE-STEP)*PY(1)/(XE*Y(1))
     IF ((RATIO*RATIO) < 1E-15) THEN
     RATIO = (XE-STEP)*PY(1)/(XE*Y(1))
     IF (RATIO < 1) THEN
        IF (RATIO > -1) THEN
        IF (Y(1) > 0) THEN
        DOUT = 1
        !Print*, "eup", XBLOW
        IF (DR /= 1) THEN
           STEP2 = STEP2/10
           DR = 1
           DR2 = DR2*(-1)
        END IF
        END IF
        IF (Y(1) < 0) THEN
        DOUT = 1
        !Print*, "edown", XBLOW
        !Print*, "zeros", ENCO
        IF (DR /= -1) THEN
           STEP2 = STEP2/10
           DR = -1
           DR2 = DR2*(-1)
           END IF
        END IF
     END IF
     END IF
     END IF
     END IF
     IF (Y(1) > 1.5) THEN
         DOUT = 1
         !Print*, "up", XBLOW
```

```
            IF (DR /= 1) THEN

                STEP2 = STEP2/10
                DR = 1
                DR2 = DR2*(-1)
            ENDIF
        ENDIF
        IF (Y(1) < -1.5) THEN
             DOUT = 1
             !Print*, "down " ,XBLOW
             !Print*, "zeros ", ENCO
             IF (DR /= -1) THEN
                STEP2 = STEP2/10
                DR = -1
                DR2 = DR2*(-1)
             ENDIF
        ENDIF
        IF (XE > XEND) THEN
           DR = 10
           DOUT = 1
        ENDIF
        IF (STEP2+Y1 == Y1 ) THEN
           DR = 10
           DOUT = 1
        ENDIF
   ENDDO
   ENDDO
! obtained value for bound state now write it in file
! And look for another
   Print*, " bound state ",Y1
   PRINT*, " Number of zeros is ", ENCO
   Print*, " Blow up value is ", XBLOW
    Write(10,*),Y1 ,XBLOW,ENCO
! Now consider the intergation to get A / B^2 !!
        Y(3) = 1.0
        Y(2) = 0.0
        Y(1) = Y1
        Y(4) = 0.0
        XBLOWP = XBLOW
        XBLOW = 0
        ENCO = 0
        ENCO2 = 0
        DOUT = 0
        X = 0.0000001
        HSTART = 0.0
        XE = X
        INT1 = 1
        INT2 = 0
```

```
  CALL D02PVF(N,X,Y,XEND,TOL,THRES,METHOD,TASK,ERRAS,HSTART,W,LENWRK,IFAIL)
   DO WHILE (XE < XBLOWP)
        XE = XE + STEP
        PY2(1:4) = PY(1:4)
        PY(1:4) = Y(1:4)
        IF (XE < XEND) THEN
        CALL D02PCF(FCN,XE,X,Y,YP,YMAX,W,IFAIL)
        END IF
        INT1 = INT1 - STEP*XE*Y(1)*Y(1)
        INT2 = INT2 + STEP*XE*XE*Y(1)*Y(1)
        IF (XE > 3*STEP) THEN
        RATIO = (XE-2*STEP)*PY2(1)/((XE-STEP)*PY(1))
        RATIO = RATIO - (XE-STEP)*PY(1)/(XE*Y(1))
        IF ((RATIO*RATIO) < 1E-15) THEN
        RATIO = (XE-STEP)*PY(1)/(XE*Y(1))
        IF (RATIO < 1) THEN
           IF (RATIO > - 1) THEN
           Print*, RATIO
           IF (Y(1) > 0) THEN
           DR = 1
           END IF
           IF (Y(1) < 0) THEN
           DR = - 1
           END IF
        END IF
        END IF
        END IF
        END IF
        END IF
        IF (Y(1) > 2) THEN
           DR = 1
        ENDIF
        IF (Y(1) < -2) THEN
           DR = -1
        ENDIF
        IF (XE > XEND) THEN
           DR = 10
        ENDIF
   ENDDO
Print*, "A is : ", INT1
Print*, "B is : ", INT2
Write(11,*) INT1,INT2
    X = 0.0000001
    XE = X
    INSIZE = (Y1P-Y1)/10
    Y1P = Y1
    Y1 = Y1 - 0.0000001
    Y(1) = Y1
    Y(2) = 0.0
```

```
    Y(3) = 1.0
    Y(4) = 0.0
    ENDDO

END PROGRAM SN
!
! FUNCTION EVALUATOR
!
      SUBROUTINE FCN(S,Y,YP)
      Implicit None
      Double Precision :: S,Y(4),YP(4)
      YP(1) = Y(2)
      YP(2) =-2.D0*Y(2)/S-Y(3)*Y(1)
      YP(3) = Y(4)
      YP(4) =-2.D0*Y(4)/S-Y(1)*Y(1)
      RETURN
      END

!
! OUTPUT - subroutine

SUBROUTINE OUTPUT(Xsol,y)
Implicit None

Double Precision :: Xsol
Double Precision, DIMENSION(4) :: Y

PRINT*, Xsol,y(1),y(3)
WRITE(2,*), Xsol,y(1),y(3)

END

!
! Subroutine Count to count the bumps in our curve

SUBROUTINE COUNT(Yold,Ynew,Number,way)
Implicit None

Double Precision, DIMENSION(4) :: Yold, Ynew
Integer :: Number, way

If (Yold(1) > 0) THEN
      If  (Ynew(1) < 0) THEN
         ! we have just reached a turing point
         way = -1
         Number = Number + 1
      ENDIF
ENDIF
```

```
If (Yold(1) < 0) THEN
      If (Ynew(1) > 0) THEN
         way = 1
         Number = Number + 1
      ENDIF
ENDIF
END
```

Appendix B

Matlab programs

B.1 Chapter 2 Programs

B.1.1 Program for asymptotically extending the data of the bound state calculated by Runge-Kutta integration

```
% To work out the asymptotic solution %
% r the point to match to !!
% the nb2 is the solution matching to at the moment !!
% matching with exp(-r)/r
% 1/r
%
b = 5000;
%step = nb2(2,1) - nb2(1,1);
yr = interp1(nb2(:,1),nb2(:,2),r);
yr2 = interp1(nb2(:,1),nb2(:,2),r+1);
d = real(log(yr*r/(yr2*(r+1))));
s = exp(-d*r)/r;
c = yr/s;
vr = interp1(nb2(:,1),nb2(:,3),r);
vr2 = interp1(nb2(:,1),nb2(:,3),r+1);
f = (vr2*(r+1)-vr*r)
e = vr*r- r*f;
% then work out values of the function
for i = 1:b
tail(i,1) = nb2(1,1) + (i-1)*step ;
tail(i,2) = c*exp(-d*tail(i,1))/tail(i,1);
tail(i,3) = e/tail(i,1) + f;
end
```

B.1.2 Programs to calculate stationary state by Jones et al method

```
function [f,evalue] = cpffsn(L,N,n);
% function [f,evalue] = cpffsn(L,N,n);
%
[D,x] = cheb(N);
```

```
x2 = (1+x)*(L/2);
phi = 1./(1+x2);
[f,evalue] = cpfsn(L,N,n,x2,phi);

function [f,evalue] = cpfsn(L,N,n,x,phi)
% [f,evalue] = CPFSN(L,N,n,x,phi)
%
% solve the bounds states of the Schrodinger-Newton Equation
% L length of interval
% N number of cheb points
% n the nth bound state
% x the grid points wrt function is required
% phi initial guess for potential
%
[D,xp] = cheb(N);
dist = L/2;
cx = (xp+1)*dist;
D2 = D^2/(dist^2);
D2 = D2(2:N,2:N);
newphi = interp1(x,phi,cx);
oldphi = newphi;
res = 1;
while (res > 1E-13)
[ft,evalue] = cpesn(L,N,n,newphi,cx);
nom = inprod(ft,ft,cx,N);
ft = ft/sqrt(nom);
g = -abs(ft(2:end-1)).^2./cx(2:end-1);
pu = D2\g;
pu = [0;pu;0];
newphi = pu(1:N)./cx(1:N);
newphi(N+1) = newphi(N);
res = max(abs(newphi-oldphi))
oldphi = newphi;
end
fx = interp1(cx,ft,x);
f =-fx;
%sq = inprod2(fx,fx,x,N);
%f = interp1(sq*x,fx/sq,x);
%sq2 = inprod2(f,f,x,N);
%sq = sq*sq2;
%f = interp1(sq*x,fx/sq,x);

function [f,evalue]  = cpesn(L,N,n,phi,x)
% [f,evalue] = CPESN(L,N,n,x)
%
% works out nth eigenvector of schrodinger equation with self-potential
% initial guess of 1/(r+1) with the length of the interval L ,
% N is number of cheb points
```

```
%  interpolates to x
%
[V,eigs] = cpsn(L,N,phi,x);
cx(1) = 0;
cx(N+1) = L;
dist = (cx(N+1)-cx(1))/2;
for i = 1:(N+1)
    cx(i) = (-cos((i-1)*pi/(N)))*dist+dist+cx(1);
end
E = diag(eigs);
E0 = 0*E;
for a = 1:n
[e,i] = max(E-E0);
E0(i) = NaN;
end
ev = [0; V(:,i); 0];
f = interp1(cx,ev,x);
evalue = e;

function [V,eigs] = cpsn(L,N,phi,x)
% [V,eigs] = cpsn(L,N,phi,x)
%
% potential of phi wrt x with usual boundary conditions.
% using cheb differential operators
%
% N = number of cheb points
% L = length of interval
%
cx(1) = 0;
cx(N+1) = L;
dist = (cx(N+1)-cx(1))/2;
for i = 1:(N+1)
    cx(i) = (-cos((i-1)*pi/(N)))*dist+dist+cx(1);
end
%
% working out the cheb difff matrix
%
D = cheb(N);
D2W = D^2/dist^2;
D2 = D2W(2:N,2:N);
ZE = zeros(N,N);
I = diag(ones(N-1,1));
%
%Matrices C1 and C2
%
% Matrices C1 and C2 to solve
% D2 + 1/rR = ER
%
```

```
cphi = interp1(x,phi,cx(2:N));
C1 = D2 + diag(cphi);
C2 = I;
%Compute eigenvalues
[V,eigs] = eig(C1,C2);

%cheb.m - (N+1)*(N+1) chebyshev spectral differentiaion
% Matrix via explicit formulas

function [D,x] = cheb(N)
if N==0 D=0; x=1; break, end;
ii = (0:N)';
x = cos(pi*ii/N);
ci = [2;ones(N-1,1);2];
D = zeros(N+1,N+1);
for j = 0:N;    %compute one column at a time
   cj = 1; if j==0 | j==N cj = 2; end
   denom = cj*(x(ii+1)-x(j+1)); denom(j+1) = 1;
   D(ii+1,j+1) = ci.*(-1).^(ii+j)./denom;
   if j > 0 & j< N
D(j+1,j+1) = -.5*x(j+1)/(1-x(j+1)^2);
   end
end
D(1,1) = (2*N^2+1)/6;
D(N+1,N+1) = -(2*N^2+1)/6;
```

B.2 Chapter 4 Programs

B.2.1 Program for solving the Schrödinger equation

```
% schrodinger equation with potential from the bound state
% To create function of nb1
% with cheb point spaces
% cos(i*pi/2k)
% N = number of points
clear bx1 bv1 br1
dist = (nb1(l,1)-nb1(1,1))/2;
for i = 1:(N+1)
    bx1(i) = (-cos((i-1)*pi/(N)))*dist+dist+nb1(1,1);
    br1(i) = interp1(nb1(:,1),nb1(:,2),bx1(i));
    bv1(i) = interp1(nb1(:,1),nb1(:,3),bx1(i));
    br2(i) = interp1(nb2(:,1),nb2(:,2),bx1(i));
    br3(i) = interp1(nb3(:,1),nb3(:,2),bx1(i));
end
% working out the cheb difff matrix
%
D = cheb(N);
D2 = D^2/dist^2;
```

```
D2 = D2(2:N,2:N);
D = D(1:N,1:N);
V0 = diag(bv1(2:N)+Estate);
I = diag(ones(N-1,1));
%Matrices C1 and C2
%for the schrodinger equation
%D2 + V0 = e
%
C1 = D2 + V0;
C2 = -I;
%
%Compute eigenvalues
[V,eigs] = eig(C1,C2);
E = diag(eigs);
hold off
subplot(1,1,1);
plot(diag(eigs),'r*');
grid;
pause;
[y,i] = min(E)
subplot(3,2,1);
plot(bx1(2:N),V(1:N-1,i)./bx1(2:N)')
f = V(1:N-1,i).^2;
f(2:N) = f(1:N-1);
f(1) = 0;
int = dist*(D\f);
int(1)
title('first eigenvector ');
grid on;
subplot(3,2,2);
ratio = (V(1,i)./bx1(2))/br1(2);
ratio2 = ratio.^2/int(1);
ratio2 = ratio2^(1/3);
error2 = interp1(bx1(2:N),br1(2:N)'*ratio,ratio2*bx1(2:N));
error = error2' - V(1:N-1,i)./bx1(2:N)';
plot(bx1(2:N),error)
title('error to nb1')
E(i) = max(E)+1;
[y2,i2] = min(E)
subplot(3,2,3);
plot(bx1(2:N),V(1:N-1,i2)./bx1(2:N)')
f = V(1:N-1,i2).^2;
f(2:N) = f(1:N-1);
f(1) = 0;
int = dist*(D\f);
int(1)
title('second eigenvector ');
grid on;
```

```
subplot(3,2,4);
ratio = (V(1,i2)./bx1(2))/br2(2);
ratio2 = ratio.^2/int(1);
ratio2 = ratio2^(1/3);
error2 = interp1(nb2(:,1),nb2(:,2)*ratio,ratio2*bx1(2:N));
error = error2' - V(1:N-1,i2)./bx1(2:N)';
plot(bx1(2:N),error)
title('error to nb2')
E(i2) = max(E)+1;
[y2,i3] = min(E)
subplot(3,2,5);
plot(bx1(2:N),V(1:N-1,i3)./bx1(2:N)')
f = V(1:N-1,i3).^2;
f(2:N) = f(1:N-1);
f(1) = 0;
int = dist*(D\f);
int(1)
title('third eigenvector is:');
grid on;
subplot(3,2,6);
ratio = (V(1,i3)./bx1(2))/br3(2);
ratio2= ratio.^2/int(1);
ratio2 = ratio2^(1/3);
error2 = interp1(nb3(:,1),nb3(:,2)*ratio,ratio2*bx1(2:N));
error = error2' - V(1:N-1,i3)./bx1(2:N)';
plot(bx1(2:N),error)
title('error to nb3')
```

B.2.2 Program for solving the $O(\epsilon)$ perturbation problem

```
% first order perturbation
% To create function of nb1
% with cheb point spaces
% cos(i*pi/2k)
% N = number of points
clear bx1 bv1 br1
dist = (nb1(l,1)-nb1(1,1))/2;
for i = 1:(N+1)
    bx1(i) = (-cos((i-1)*pi/(N)))*dist+dist+nb1(1,1);
    br1(i) = interp1(nb1(:,1),nb1(:,2),bx1(i));
    bv1(i) = interp1(nb1(:,1),nb1(:,3),bx1(i));
end
%
% working out the cheb difff matrix
%
D = cheb(N);
D2W = D^2/dist^2;
D2 = D2W(2:N,2:N);
```

```
V0 = diag(bv1(2:N));
R0 = diag(br1(2:N));
ZE = zeros(N-1,N-1);
I = diag(ones(N-1,1));
%Matrices C1 and C2
%for the S_N equations
% subject to the equations
%   -2R0A + D2W = 0
%  D2B + V0B = -eA
%  -D2A - V0A + R0W = eB
%  with boundary conditions
% A = 0 , B =0 W = 0 at endpoints
% DA = 0 at infinity
%       DB = 0 at infinity
C1 = [-2*R0,ZE,D2;ZE,D2+V0,ZE;-D2-V0,ZE,R0];
C2 = [ZE,ZE,ZE;-I,ZE,ZE;ZE,I,ZE];

%Compute eigenvalues
[V,eigs] = eig(C1,C2);
j = sqrt(-1);
hold off
E = j*diag(eigs);
Enew = sort(E);
plot(Enew(1:N),'r*');
grid
title('Eigenvalues of the perturbed S-N equation ')
```

B.2.3 Program for the calculation of (4.10)

```
% abint.m - program to work out integral of  A*B
%
D = cheb(N);
D = D(1:N,1:N);
f = abs(V(1:N-1,i)).^2;
f(2:N) = f(1:N-1);
f(1) = 0;
A = dist*(D\f);
g = abs(V(N:2*N-2,i)).^2;
g(2:N) = g(1:N-1);
g(1) = 0;
B = dist*(D\g);
f = V(1:N-1,i);
g = V(N:2*N-2,i);
f = conj(f);
fg = f.*g;
fg(2:N) = fg(1:N-1);
fg(1) =0;
int = dist*(D\fg);
```

```
ratio = sqrt(A(1))*sqrt(B(1))\abs(int(1))
```

B.2.4 Program for performing the Runge-Kutta integration on the $O(\epsilon)$ perturbation problem

```
% see the solution of the first order perturbation equations
% intial data
%
global nb1;
e = eigenvalue;
global e;
y0 = [0 1 0 1 0 1]';
y0(2) = initial value for y0(2);
y0(4) = initial value for y0(4);
y0(6) = initial value for y0(6);
X0 = 0.000001;
Xfinal = L - length of the interval;
F = 'yfo';
Xspan = [X0,Xfinal];
[xp,yp] = ode45(F,Xspan,y0);
subplot(3,1,1)
plot(xp,imag(yp(:,3)./xp))
title('A/r')
subplot(3,1,2)
plot(xp,imag(yp(:,1)./xp))
title('B/r')
subplot(3,1,3)
plot(xp,imag(yp(:,5)./xp))
title('W/r')
```

```
function Yprime = yfo(X,Y)
% this is function for the first order pert equations
% where V , R are the vaule of the potential and wave function
% related to the zero th order perturbation.
% e is the eigenvalue ( e = -i\lambda )
global e
Yprime(1) = Y(2);
Yprime(2) = -V(X)*Y(1)+e*Y(3);
Yprime(3) = Y(4);
Yprime(4) = -V(X)*Y(3)+e*Y(1)+R(X)*Y(5);
Yprime(5) = Y(6);
Yprime(6) = 2*R(X)*Y(3);
Yprime = Yprime';
```

B.3 Chapter 6 Programs

B.3.1 Evolution program

```
%snevc2.m
```

```
%Iteration scheme for the schrodinger-newton equations
%using crank -nicolson + cheb ,
% Modified 13/3/2001 -- for perturbation about ground state.
clear i; clear norm; clear Eu;
clear fgp; clear g; clear psi; clear psi1; %clears used data formats
hold off
N = 70;
N2 = N;
R = 350; count = 1;
dt = 0.5; one = 1;
[D,x] = cheb(N);
D2 = D^2;
D2 = D2(2:N,2:N);
x = 0.5*((1-x)*R);
u = cpfsn(R,N2,3,x,1./(x+1)); %calculate stationary solution
x = x(2:N); u = u(2:N);
dist = 0.5*R;
D2 = D2/dist^2;
t = 0; nu = dt;
a = 50; v = 0;%-0.1;
av = a+v*t;
vs = (v^2*t)/4;
sigma = 15;
rsig = sqrt(sigma);
brack = 1/(sigma^2);
rbrack = sqrt(brack);
wave = rbrack*rsig*exp(-0.5*brack*(x-av).^2+(0.5*i*v*x)-i*vs);
wave2= rbrack*rsig*exp(-0.5*brack*(x+av).^2-(0.5*i*v*x)-i*vs);
pert = wave-wave2;
% alternative initial data !
uold = u;  %+0.01*pert;
u = uold;
fwe = u;
rho = abs(u(1:end)).^2./x(1:end);
pu = D2\rho;
phi = pu./x;
% solve the potential equation
% sponge factors
e = -2*exp(0.2*(x-x(end)));
%e = zeros(N-1,1);
o = zeros(N-1,1);
Eu(count) = abs(u(5));
fgp(count) = abs(u(1));
fgp2(count) = abs(u(15));
% start of the iteration scheme
for zi2 = 1:100
 for zi = 1:1000
  count = count +1;
```

```
  phi1 = phi;  % initial guess !
  Res = 9E9;
  while abs(Res) > 3E-8
   u = uold;
   for z = 1:N-1
    g(z) = (i+e(z))*nu/2;
    psi1(z) = one*(i*dt/2)*(phi1(z)+o(z));
    psi(z) = one*(i*dt/2)*(phi(z) +o(z));
   end
   maindaig = ones(N-1,1)+psi1';
   d = (ones(N-1,1)-psi').*u - g.'.*(D2*u);
   b = diag(maindaig) + diag(g)*D2;
   u = b\d;
   unew = u;
   % this was the solving of the P.D.E !
   % work out potential for u^n+1 %%
   rho = abs(u(1:end)).^2./x(1:end);
   pu = D2\rho;
   phi1 = pu./x;
%then recalution of the result ! to check
% Now to check the L infinty norm of the residual
   u = uold;
   for z = 1:N-1
    g(z) = (i+e(z))*nu/2;
    psi1(z) = one*(i*dt/2)*(phi1(z)+o(z));
    psi(z) = one*(i*dt/2)*(phi(z) +o(z));
   end
   maindaig = ones(N-1,1)+psi1';
   d = (ones(N-1,1)-psi').*u -  g.'.*(D2*u);
   b = diag(maindaig) + diag(g)*D2;
   u = b\d;
   Res = max(abs(u-unew));
   %Res = max(abs(phi1-phi))
% end of checking the L inifinity norm .%
  end
  Eu(count) = abs(u(5));
  fgp(count) = abs(u(1));
  fgp2(count) = abs(u(30));
  norm(zi2) = inprod(u,u,x,N-1);
  pe(zi2) = pesn(u,x,phi,N);
  t = t + dt;
%fwe(:,end+1) = u;
  uold = unew;
  phi = phi1;
  %save fp.mat fgp
  %save Ep.mat Eu
  %save nn.mat norm
 end
```

```
 fwe(:,end+1) = u;
 save fp.mat fgp fgp2 %Eu
 save nn.mat fwe norm
 %tone = t*ones(N-1,1);
 %plot3(tone,x,abs(u),'b');
 %drawnow
 %hold on
 %hold on
 %plot(x,phi1,'g');
 %plot(nb1(:,1),abs(nb1(:,1).*nb1(:,2)),'r')
 %plot(x,angle(u),'k');
end
```

B.3.2 Fourier transformation program

```
N = max(size(f))/2;
dt = 0.5;
freq = pi/(N*dt);
n = 2*N;
Fn = fft(f);
Fn = [conj(Fn(N+1)) Fn(N+2:end) Fn(1:N+1)]; % rearrange frequency points
Fn = Fn/n; % scale results by fft length
idx = -N:N;
Fn(N+1) = 0;
idxs = idx*freq;
plot(log(idxs(N+1:2*N+1)),log(abs(Fn(N+1:2*N+1))))
xlabel('log(frequency)');
ylabel('log(fourier transform)');
```

B.4 Chapter 7 Programs

B.4.1 Programs for solving the axisymmetric stationary state problem

```
function [f,aphi,evalue,lres,count] = cp2fsn(dis,N,N2,n,l)
% function [f,aphi,evalue] = cp2fsn(dis,N,N2,n,l)
%
%
warning off
[D,x] = cheb(N);
[E,y] = cheb(N2);
dis2 = pi/2;
x2= dis*(x+1); % x is the radius points.
y2 = dis2*(y+1);
x2= x2(2:N);
D = D/dis;
E = E/dis2;
D2 = D^2;
E2 = E^2;
```

```
D = D(2:N,2:N);
D2 = D2(2:N,2:N);
I = eye(N2+1);
fac1 = diag(1./x2.^2);
fac2 = diag((cot(y2-eps)+cot(y2+eps))/2);
% initial guess for potential
for i = 1:N2+1
for j = 1:N-1
phi((N-1)*(i-1)+j) = 1./(1000*x2(j)+1);
end
end
aphi = phi;

L = kron(I,D2) + kron(E2,fac1)+ kron(fac2*E,fac1); %Laplacian kinda of !
% actually Le Laplacian for axially symetric case !
newphi = phi;
oldphi = newphi;
res = 1;
mang = 0; mener = 0;
count = 0;
% start of a iteration
 while(res > 1E-4)
  count = count +1;
  [ft,evalue,mener,mang] = cp2esnea(dis,N,N2,mener,mang,newphi,n,l);
  % get nth evector correponding to potential
  % next aim to work out the potential, corresponding to evector !
  data = ft;
  norms
  ft = ft/nom;
  clear g;
  for i = 1:N2+1
  for j = 1:N-1
  g((N-1)*(i-1)+j) = -abs(ft((N-1)*(i-1)+j))^2/x2(j);
  end
  end
  g = g.';
  pu = L\g;
  for i = 1:N2+1
  for j = 1:N-1
  nwphi((N-1)*(i-1)+j) = pu((N-1)*(i-1)+j)/x2(j);
  end
  end
  for i = 1:N2+1
  for j = 1:N-1
  newphi((N-1)*(i-1)+j) = 0.5*(nwphi((N-1)*(i-1)+j)+nwphi((N-1)*(N2+1-i)+j));
  end
  end
  res2 = abs(max(newphi-nwphi));
```

```
  res = abs(max(newphi-oldphi))/abs(max(newphi))
  oldphi = newphi;
  data = newphi;
  %axpl
  %drawnow;
  lres = res;
  if count > 50
   lres = res;
   res = 0;
  end
  end
  f = ft;
  aphi = newphi;

function [f,evalue,ener,ang] = cp2esnea(dis,N,N2,pener,pang,phi,n,l);
%
%
[V,eigs] = cp2sn2(dis,N,N2,phi);
En = diag(eigs);
[e,ii] = sort(real(-En));
W = 0;
b = 0;
vclose = 10E8;
found = 0;
limit = floor(max(ii)/4);
for as = 1:limit
     data = V(:,ii(as));
     norms2
     if (nom > 0.04)
      data = data/nom;
      enr;
      [n2,l2] = fnl(data,N,N2);
      close = (pener-ener)^2 + 1E-4*(pang-ang)^2+ 0.1*((n2-n)^2+ 10*(l2-l)^2);
      if (close < vclose)
       vclose = close;
      end
     end
end
while(W == 0);
 b = b+1;
 data = V(:,ii(b));
 norms2
 if (nom > 0.04)
   data = data/nom;
   enr;
   W = 1;
   [n2,l2] = fnl(data,N,N2);
   close = (pener-ener)^2 + 1E-4*(pang-ang)^2+ 0.1*((n2-n)^2+ 10*(l2-l)^2);
```

```
 end
 if (W == 1)
  W = 0;
 if (close < vclose*1.001)
  %axpl2
  %drawnow;
  W = 1;
 end
 end
 %if (W == 1)
 %W = input('correct state');
 %end
end
evalue = En(ii(b));
f = V(:,ii(b));

% find n and l number of the given grid ..
function [n,l] = fnl(data,N,N2);
% where  the u is r\phi
%
for i = 1:N2+1
for j = 1:N-1
u(j,i) = data((N-1)*(i-1)+j);
end
end
clear no noo;
j = 0;
for i = 1:N-1
 j = j+1;
 no(i) = cz2sn(u(i,1:N2+1));
 if (no(i) == -1)
  j = j -1;
 end
 if (j > 0)
  noo(j) = no(i);
 end
end
l = mean(no);
%if (j > 0)
% l = mean(noo);
%end
clear no2 noo2;
i = 0;
for j = 1:N2+1
 i = i +1;
 no2(j) = cz2sn(u(1:N-1,j));
 if (no2(j) == -1)
  i = i -1;
```

```
 end
 if (i > 0)
  noo2(i) = no2(j);
 end
end
n = mean(no2);
%if (i > 0)
%n = mean(noo2);
%end
%l = ceil(l);
%n = ceil(n);
%l = round(l);
%n = round(n);

% to calculate the norm of a
% axially sysmetric data set!
%
% Jab = r^2sin \alpha
%
clear ener
clear ang
ener = 0;
ang = 0;
[E,y] = cheb(N2);
dis2 = pi/2;
y2 = pi/2*(y+1);
[D,x] = cheb(N);
x2= dis*(x+1);
x2= x2(2:N);
D = D/dis;
E = E/dis2;
D2 = D^2;
E2 = E^2;
D = D(2:N,2:N);
D2 = D2(2:N,2:N);
I = eye(N2+1);
I2 = eye(N-1);
fac1 = diag(1./x2.^2);
fac2 = diag((cot(y2+eps)+cot(y2-eps))/2);

L = +kron(I,D2)+kron(E2,fac1)+kron(fac2*E,fac1); %Laplacian kinda of !

J = kron(E2,I2) + kron(fac2*E,I2);
%data2 = data;
%for i = 1:N2+1
%for j = 1:N-1
%data2((N-1)*(i-1)+j) = data((N-1)*(i-1)+j)/x2(j);
%end
```

```
%end
nab2 = L*data;
aab2 = J*data;
for i = 3:N2+1
for j = 2:N-1
darea = (x2(j)-x2(j-1))*(y2(i)-y2(i-1));
jab = sin(y2(i));
jab2 = sin(y2(i));
ener = ener - jab*abs(data((N-1)*(i-1)+j))^2*darea*phi((N-1)*(i-1)+j);
ener = ener - jab*conj(data((N-1)*(i-1)+j))*nab2((N-1)*(i-1)+j)*darea;
ang = ang - jab*conj(data((N-1)*(i-1)+j))*aab2((N-1)*(i-1)+j)*darea;
end
end
norms2

% to calculate the norm of a
% axially sysmetric data set!
%
% Jab = r^2sin \alpha
%
clear nom nom2
% Setup the grid
nom = 0; nom2 = 0;
[E,y] = cheb(N2);
y2 = pi/2*(y+1);
[D,x] = cheb(N);
x2= dis*(x+1); % x is the radius points.
x2= x2(2:N);
for i = 2:N2+1
for j = 2:N-1
darea = (x2(j)-x2(j-1))*(y2(i)-y2(i-1));
jab = sin(y2(i));
nom = nom + jab*abs(data((N-1)*(i-1)+j))^2*darea;
end
end
nom = nom/2;
nom = sqrt(nom);

% cz2sn.m
% function to work out number of crossing given matrix array...
function number = cz2sn(W);
%
a = max(size(W));
count = 0;
posne = 2;
eps2 = 1E4*eps;
if (W(1) > eps2)
posne = 1;
```

```
end
if (W(1) < -eps2)
posne = 0;
end
for i = 2:a
 if (posne == 1)
  if W(i) < -eps2
  posne = 0;
  count = count+1;
  end
 end
 if (posne == 0)
  if W(i) > eps2
  posne = 1;
  count = count+1;
  end
 end
 if (posne == 2)
  if W(i) > eps2
  posne = 1;
  end
  if W(i) < -eps2
  posne = 0;
  end
 end
end
if posne == 2
count = -1;
end
number = count;

function [V,eigs] = cp2sn2(dis,N,N2,phi);
% [V,eigs] = cp2sn2(dis,N,N2,phi);
%
% program to solve the stationary schrodinger in axially symmetric case.
% r,\alpha are the variable in which we solve for at the
% present both has N points !
% boundary condition are different on the \alpha !
% i.e we expect phi to be N-1*N-1 vector !
%
[E,y] = cheb(N2);
y2 = pi/2*(y+1);
dis2 = pi/2;
[D,x] = cheb(N);
x2= dis*(x+1); % x is the radius points.
x2= x2(2:N);
D = D/dis;
E = E/dis2;
```

```
D2 = D^2;
E2 = E^2;
D = D(2:N,2:N);
D2 = D2(2:N,2:N);
I = eye(N2+1);
fac1 = diag(1./x2.^2);
fac2 = diag((cot(y2-eps)+cot(y2+eps))/2);

L = +kron(I,D2)+kron(E2,fac1) + kron(fac2*E,fac1); %Laplacian kinda of !

L = L + diag(phi);
%
[V,eigs] = eig(L);
```

B.4.2 Evolution programs for the axisymmetric case

```
% nsax2.m Schrodinger-Newton equation
% on an axially symmetric grid -
% full non-linear method using chebyshev, ADI(LOD) wave equation
% that the wave equation done in chebyshev differention
% matrix in 2d
clear xx2 yy2
load dpole
[DD,x3] =cheb(N);
x3 = dis*(x3+1);
x3 = x3(2:N);
[EE,y3] = cheb(N2);
y3 = pi/2*(y3+1);
clear DD EE
for cc = 1:N-1
for cc2 = 1:N2+1
f2(cc2,cc) = f((cc2-1)*(N-1)+cc);
fred(cc2,cc) = aphi((cc2-1)*(N-1)+cc);
xx2(cc2,cc) = x3(cc)*cos(y3(cc2));
yy2(cc2,cc) = x3(cc)*sin(y3(cc2));
end
end
dis = 50;
grav = 1;
clear xx yy vv
% grid setup %
N = 25;
N2 = 20;
[E,y] = cheb(N2);
dis2 = pi/2;
y2 = pi/2*(y+1);
[D,x] = cheb(N);
x2 = dis*(x+1);
```

```
x2 = x2(2:N);
D = D/dis;
E = E/dis2;
% create xx,yy
for i = 1:N2+1
for j = 1:N-1
xx(i,j) = x2(j)*cos(y2(i));
yy(i,j) = x2(j)*sin(y2(i));
end
end
% initial data %
%vv = interp2(xx2,yy2,f2,xx,yy);
vv = f2;
%vv = 0.1*sqrt(xx.^2+yy.^2).*exp(-0.01*((xx).^2+(yy).^2));
%vva = 0.9*sqrt(xx.^2+yy.^2).*exp(-0.01*((xx+30).^2+(yy).^2));
%vv = vv + vva;
% time step !
dt = 0.5;
% plotting the function on the screen at intervals !
plotgap = round((2000/10)/dt); dt = (2000/10)/plotgap;
vvold = vv;
vvnew = 0*vv;
phi = 0*vv;
pot = phi;
% sponge factors
%e = 0*vv;
e = 1*exp(0.3*(sqrt(xx.^2+yy.^2) - 2*dis));  %i.e no sponge factors
%% calculate of differentiation matrices
D2 = D^2;
D2 = D2(2:N,2:N);
DD = D(2:N,2:N);
E2 = E^2;
I = eye(N2+1);
I2 = eye(N-1);
fac1 = diag(x2);
fac2 = diag((cot(y2+eps)+cot(y2-eps))/2);
fac3 = diag(1./x2);
fac2(1,1) = 1E10;
fac2(end,end) = 1E10;
L1 = E2 + fac2*E;
L2 = D2;
% calculate the starting potential %
u = grav*vv; % wave-function
v = 0*pot; % initial guess of new pot for potadi
phi = -padi(dis,u,v,N,N2); % calculation of the potential
%phi = 0*pot;
time =0;
pic = vv;
```

```
% evolution loop begin here ....
for n = 0:30*plotgap
 t = n*dt;
 if rem(n,plotgap) == 0,    % - Plot results at multiples of plotgap
    %subplot(2,2,n/plotgap+1),
    %[xxx,yyy] = meshgrid(-dis:dis*.02:dis,-dis:dis*.02:dis);
    %vvv = interp2(xx,yy,vv,xxx,yyy,'cubic');
    %mesh(xxx,yyy,abs(vvv)), % colormap([0 0 0])
    %mesh(xx,yy,phi)
    time(end+1) =t;
    pic(:,:,end+1) =vv;
    %view(that);
    %hold off
    %mesh(xx,yy,abs(vv)./sqrt(xx.^2+yy.^2));
    %surf(xx,yy,imag(vv))
    %hold on
    %surf(xx,-yy,abs(vv)) %./sqrt(xx.^2+yy.^2));
    %shading interp
    %axis([-1 1 -1 1 -1 1]),
    %surf(xx,yy,abs(uaa))
    %axis([-2*dis 2*dis 0 2*dis -0.2 0.2])
    %title(['t = ' num2str(t)]), drawnow
    % calculate of the norm..
    %cn2d
    %cnorm
 end
Res = 1;
pot = phi;
npot = phi;
while (abs(Res) > 1E-4) % set up for the iteration ..
% calculate of the partials
urr = zeros(N2+1,N-1);
% for a = 1:N2+1
% urr(a,1:N-1) = (L2*vv(a,1:N-1).').';
% end
uaa = zeros(N2+1,N-1);
for a = 1:N-1
uaa(1:N2+1,a) = (1/x2(a)^2)*L1*vv(1:N2+1,a);
end
% --- ADI(LOD) -- %
vv2 = 0*vv;
% (1+dr)Un2 = (1-da)Un1
for a = 1:N2+1
sv2 = vv(a,1:N-1);
ds = sqrt(-1)*uaa(a,1:N-1) + e(a,1:N-1).*uaa(a,1:N-1);
d = (sv2 + 0.5*dt*ds).';
```

```
bs2 = sqrt(-1)*L2 + diag(e(a,1:N-1))*L2;
b = eye(N-1) - 0.5*dt*bs2;
vv2(a,1:N-1) = (b\d).';
end
% end part1 %
% caculate urr,uaa
% for a = 1:N-1
% uaa(1:N2+1,a) = (1/x2(a)^2)*L1*vv2(1:N2+1,a);
% end
for a = 1:N2+1
urr(a,1:N-1) = (L2*vv2(a,1:N-1).').';
end
% (1+da)Un3 = (1-dr)Un2
vv3 = 0*vv2;
for a = 1:N-1
sv = vv2(1:N2+1,a);
ds = sqrt(-1)*urr(1:N2+1,a) + e(1:N2+1,a).*urr(1:N2+1,a);
d = sv + 0.5*dt*ds;
bs2 = (1/x2(a)^2)*sqrt(-1)*L1 + (1/x2(a)^2)*diag(e(1:N2+1,a))*L1;
b = eye(N2+1) - 0.5*dt*bs2;
vv3(1:N2+1,a) = b\d;
end
% end part2 %
% (1-V)Un4 = (1+V)Un3
vvnew = 0*vv3;
for a = 1:N-1
d = vv3(1:N2+1,a) + 0.5*sqrt(-1)*dt*vv3(1:N2+1,a).*pot(1:N2+1,a);
bpm2 = sqrt(-1)*diag(npot(1:N2+1,a));
b = eye(N2+1) - 0.5*dt*bpm2;
vvnew(1:N2+1,a) = b\d;
end
% end part3 %
% now the calculate of the potential of the new potential
% due to the new wave-funtion
u = grav*vvnew; % wave-function
v = 0*pot; % initial guess of new pot for potadi
n2pot = -padi(dis,u,v,N,N2); % calculation of the potential
%n2pot = 0*pot;
Res = norm(n2pot-npot);
npot = n2pot; % +0.5*pot;
%pot = npot;
end
phi = n2pot;
vv = vvnew;
%inaxprod(abs(vv),abs(vv),dis,N,N2,1)
save axtest pic time N N2 xx yy dis
end
```

```
function phi = padi(dis,u,v,N,N2)
% padi.m
% Aim: To use Peaceman-Rachford ADI iteration
% to find the solution to the poisson equation in 2D,with axially
% sysmetric case.
%
% u is the wave-function !
% v is the potential !

% grid setup !
[E,y] = cheb(N2);
y2 = pi/2*(y+1);
dis2 = pi/2;
[D,x] = cheb(N);
x2 = dis*(x+1); % x is the radius points.
x2 = x2(2:N);
D = D/dis;
E = E/dis2;

% calculation of the differnetiation matrics
D2 = D^2;
E2 = E^2;
D2 = D2(2:N,2:N);
D = D(2:N,2:N);
I = eye(N2+1);
I2 = eye(N-1);
rho = 0.017;
rho2 = 0.017;
fac1= diag(1./x2);
fac2= diag((cot(y2+eps)+cot(y2-eps))/2);
fac2(1,1) = 1E5;
fac2(end,end) = 1E5;
L1 = E2 + fac2*E;
L2 = D2;

% readjust density function !
uu = abs(u).^2*fac1;
umod = uu;
% uu =u;
% start of the iteration
Res = 1E12;
while (abs(Res) > 1E-3)
  % calculate wrr,waa
  wrr = zeros(N2+1,N-1);
  %for a = 1:N2+1
  %  wrr(a,1:N-1) = (L2*v(a,1:N-1).').';
  %end
  waa = zeros(N2+1,N-1);
```

```
  for a= 1:N-1
    waa(1:N2+1,a) = (1/x2(a)^2)*L1*v(1:N2+1,a);
  end
% -- ADI -- %
% (-L2+rho)S = (L1+rho-umod)Wn
A = -L2+rho*I2;
vv = 0*v;
for a = 1:N2+1
sv2 = rho*v(a,1:N-1) + waa(a,1:N-1) - umod(a,1:N-1); %abs(uu(a,1:N-1)).^2;
vv(a,1:N-1) = (A\(sv2.')).';
end
% end part1 of ADI
% calculate waa,wrr
%for a = 1:N-1
%waa(1:N2+1,a) = L1*vv(1:N2+1,a);
%end
for a = 1:N2+1
wrr(a,1:N-1) = (L2*vv(a,1:N-1).').';
end
newv = 0*v;
% part2 ADI
%A = -L1+rho2*I;
% (-L1+rho2)Wn+1 = (L2+rho2-umod)S
for a = 1:N-1
sv = rho2*vv(1:N2+1,a) + wrr(1:N2+1,a) - umod(1:N2+1,a); %abs(uu(1:N2+1,a)).^2;
A = -L1*(1/x2(a)^2) + rho2*I;
newv(1:N2+1,a) = A\sv;
end
% end of ADI
test = abs(newv-v);
Res = norm(test(1:N2+1,1:N-1));
padires = Res;
v = newv;
end
fac = diag(1./x2);
phi = v*fac;
```

B.5 Chapter 8 Programs

B.5.1 Evolution programs for the two dimensional S-N equations

```
% ns2d.m Schrodinger-Newton equation
% on a 2d grid -
% full non-linear method using chebyshev, ADI(LOD) wave equation
% that the wave equation done in chebyshev differention
% matrix in 2d
dis = 40;
grav = 1;
```

```
speed =0;
% grid setup %
N = 36;
[D,x] = cheb(N);
x = dis*x;
y = x';
D = D*1/dis;
[xx,yy] = meshgrid(x,y);
load twin
[DD,x2] = cheb(N2);
x2 = dis2*x2(2:N2);
y2 = x2';
for cc = 1:N2-1
for cc2 = 1:N2-1
f2(cc,cc2) = f((cc-1)*(N2-1)+cc2);
end
end
%f3 = 0.5*(f2+f2(N2-1:-1:1,:));
[xx2,yy2] = meshgrid(x2,y2);
vv = interp2(xx2,yy2,f2,xx,yy);
for cc = 1:N+1
for cc2 = 1:N+1
if isnan(vv(cc,cc2)) == 1
 vv(cc,cc2) = 0;
end
end
end
dt = 1;
% plotting the function on the screen at intervals !
plotgap = round(400/(2*dt)); dt = 400/(2*plotgap);
% initial data %
%vv = 0.07*exp(-0.01*((xx-15).^2+(yy+15).^2))*exp(sqrt(-1)*0.5);
%vva = 0.05*exp(-0.01*((xx+35).^2+(yy-35).^2));
%vv = vv + vva;
% to add initial velocity
%vv = 0.07*exp(-0.02*((xx).^2+yy.^2)).*exp(-sqrt(-1)*speed*(yy+xx)/sqrt(2));
%vva = 0.07*exp(-0.01*((xx+10).^2+yy.^2)).*exp(sqrt(-1)*speed*yy);
%vv = vv+vva;
%initial velocity perhaps %.*exp(-sqrt(-1)*0.1*xx);
vvold = vv;
vvnew = 0*vv;
phi = 0*vv;
pot = phi;
% sponge factors
e = 0*vv;
e = min(1,1*exp(0.5*(sqrt(xx.^2+yy.^2)-(dis-10))));
%% calculate of differentiation matrices
D2 = D^2;
```

```
D2 = D2(2:N,2:N);
% calculate the starting potential %
u = grav*vv; % wave-function
v = 0*pot; % initial guess of new pot for potadi
phi = potadi(dis,u,v,N); % calculation of the potential
time = 0;
pic = vv;

% evolution loop begins here
for n = 0:50*plotgap
 t = n*dt;
 if rem(n,plotgap) == 0,    % - Plot results at multiples of plotgap
    %subplot(2,2,n/plotgap+1),
    time(end+1) = t;
    pic(:,:,end+1) = vv;
    %[xxx,yyy] = meshgrid(-dis:dis*.02:dis,-dis:dis*.02:dis);
    %vvv = interp2(xx,yy,vv,xxx,yyy,'cubic');
    %mesh(xxx,yyy,abs(vvv)), % colormap([0 0 0])
    %mesh(xx,yy,phi)
    %view(that);
    %mesh(xx,yy,imag(vv));
    %axis([-1 1 -1 1 -1 1]),
    %axis([-dis dis -dis dis -0.03 0.03])
    %title(['t = ' num2str(t)]), drawnow
    % calculate of the norm..
    %cn2d
    %cnorm
 end
Res = 1;
pot = phi;
npot = phi;
while (abs(Res) > 1E-3) % set up for the iteration ..
% calculate  uxx
uyy = zeros(N+1,N+1);
for a = 2:N
uyy(a,2:N) = (D2*vv(a,2:N).').';
end
uxx = zeros(N+1,N+1);
for a = 2:N
uxx(2:N,a) = D2*vv(2:N,a);
end
% --- ADI(LOD) -- %
vv2 = 0*vv;
% (1+dy)Un2 = (1-dx)Un1
for a = 2:N
sv2 = vv(a,2:N);
ds = sqrt(-1)*uxx(a,2:N) + e(a,2:N).*uxx(a,2:N);
d = (sv2 + 0.5*dt*ds).';
```

```
bs2 = sqrt(-1)*D2 + diag(e(a,2:N))*D2;
b = eye(N-1) - 0.5*dt*bs2;
vv2(a,2:N) = (b\d).';
end
% end part1 %
% caculate uxx,uyy
for a = 2:N
uxx(2:N,a) = D2*vv2(2:N,a);
end
for a = 2:N
uyy(a,2:N) = (D2*vv2(a,2:N).').';
end
% (1+dx)Un3 = (1-dy)Un2
vv3 = 0*vv2;
for a = 2:N
sv = vv2(2:N,a);
ds = sqrt(-1)*uyy(2:N,a) + e(2:N,a).*uyy(2:N,a);
d = sv + 0.5*dt*ds;
bs2 = sqrt(-1)*D2 + diag(e(2:N,a))*D2;
b = eye(N-1) -0.5*dt*bs2;
vv3(2:N,a) = b\d;
end
% end part2 %
% (1-V)Un4 = (1+V)Un3
vvnew = 0*vv3;
for a = 2:N
d = vv3(2:N,a) - 0.5*sqrt(-1)*dt*vv3(2:N,a).*pot(2:N,a);
bpm2 = sqrt(-1)*diag(npot(2:N,a));
b = eye(N-1) + 0.5*dt*bpm2;
vvnew(2:N,a) = b\d;
end
% end part3 %
% now the calculate of the potential of the new potential
% due to the new wave-funtion
u = grav*vvnew; % wave-function
v = 0*pot; % initial guess of new pot for potadi
n2pot = potadi(dis,u,v,N); % calculation of the potential
Res = norm(n2pot-npot);
npot = n2pot; % +0.5*pot;
%pot = npot;
end
phi = n2pot;
%vv = 0.5*(vvnew - vvnew(:,N+1:-1:1));
vv = vvnew;
end
save test3 pic time N xx yy dis

function phi = potadi(dis,u,v,N)
```

```
% potadi.m
% Aim: To use Peaceman-Rachford ADI iteration
% to find the solution to the poisson equation in 2D
%
% u is the wave-function !
% v is the potential !

% grid setup !
[D,x] = cheb(N);
x = dis*x;
y = x';
[xx,yy] = meshgrid(x,y);
D = D*1/dis;

% calculation of the differnetiation matrics
D2 = D^2;
D2 = D2(2:N,2:N);
I = eye(N-1);
rho = +0.01;

% start of the iteration
Res = 1;
while (abs(Res) > 1E-4)
  % calculate uxx,uyy
  uyy = zeros(N+1,N+1);
  for a = 2:N
    uyy(a,2:N) = (D2*v(a,2:N).').';
  end
  uxx = zeros(N+1,N+1);
  for a= 2:N
    uxx(2:N,a) = D2*v(2:N,a);
  end
% -- ADI -- %
A = -D2+rho*I;
vv = 0*v;
for a = 2:N
sv2 = rho*v(a,2:N) + uxx(a,2:N) - abs(u(a,2:N)).^2;
vv(a,2:N) = (A\(sv2.')).';
end
% end part1 of ADI
% calculate uxx,uyy
for a = 2:N
uxx(2:N,a) = D2*vv(2:N,a);
end
for a = 2:N
uyy(a,2:N) = (D2*vv(a,2:N).').';
end
newv = 0*v;
```

```
% part2 ADI
for a = 2:N
sv = rho*vv(2:N,a) +uyy(2:N,a) - abs(u(2:N,a)).^2;
newv(2:N,a) = A\sv;
end
% end of ADI
test = newv-v;
Res = norm(test(2:N,2:N));
v = newv;
end
phi = v;
```