



UNIVERSITY OF PUNE

M.Sc PROJECT

*Development and Implementation of
Poisson Solver for Finite sized systems and
Spin Density Functional method for
investigation of 3D Confined Electron
Systems*

Author:

SANDIP DE

Supervisor:

PROF. D.G KANHERE

May 18, 2009

Contents

Thank you !!!	4
List of Figures	5
List of Tables	6
1 INTRODUCTION	7
1.1 THE MANY BODY SCHRÖDINGER EQUATION	7
1.2 ADIABATIC APPROXIMATION(BORN-OPPENHEIMER)	8
1.3 SELF CONSISTENT FIELD CALCULATION	8
1.4 REQUIREMENTS TO BUILD THE CODE	9
2 SCHRÖDINGER EQUATION	10
2.1 INTRODUCTION	10
2.2 SOLVING SCHRÖDINGER EQUATION IN ONE DIMENSION	10
2.3 SOLVING SCHRÖDINGER EQUATION IN 3-D	11
2.4 HIGHER ORDER FINITE DIFFERENCE METHOD	12
2.5 MATRIX STORAGE FORMAT	13
2.5.1 ROW-INDEXED COMPACT STORAGE	13
2.6 DIAGONALIZATION	14
2.7 MATRIX VECTOR MULTIPLICATION	14
2.8 TESTING THE PROGRAM	16
2.8.1 3D-HARMONIC WELL	16
2.8.2 3D INFINITE BOX	19
2.9 FUN WITH WELLS	21
2.9.1 A QUASI 2D CASE	21
2.9.2 GAUSSIAN POTENTIAL	23
2.9.3 DOUBLE GAUSSIAN WELL	23
2.10 CONCLUSION	25

3 HARTREE POTENTIAL	26
3.1 INTRODUCTION	26
3.2 SOLVING POISSON'S EQUATION	26
3.3 SOLVING POISSON'S EQUATION FOR FINITE SIZED SYSTEM	28
3.3.1 DEVELOPMENT OF THE METHOD	28
3.3.2 USING THE METHOD FOR COULOMB POTENTIAL	31
3.3.3 SINGULARITY PROBLEM	32
3.3.4 CHOOSING THE MESH SIZE	33
3.4 ALGORITHMIC DEVELOPMENT	34
3.5 TESTING THE PROGRAM	35
3.5.1 GAUSSIAN CHARGE DENSITY	36
3.5.2 UNIFORMLY CHARGED SPHERE	36
3.5.3 GAUSSIAN TYPE POTENTIAL	38
4 EXCHANGE AND CORRELATION	40
5 THE DENSITY FUNCTIONAL THEORY CODE	41
5.1 THE ALGORITHM	41
6 THE 3D CONFINED ELECTRON SYSTEM	43
6.1 INTRODUCTION	43
6.2 RESULTS	43
6.3 FUTURE PLANS	45
7 BUILDING A GRAPHICAL INTERFACE	46
References	48

Thank You !!!

It gives me a great pleasure to express my deep gratitude towards Prof. D.G. Kanhere for his valuable guidance and encouragement throughout this work . When I look back now, I find that last five months were most exciting months of my life. Each and everyday I learned something new. I am very much grateful to prof D.G kanhere, who showed me the exciting world of computational physics. At the end of my college life, I find myself very lucky to have him as my teacher .

I would not be able to do even half of the works ,I have done in this project ,without the help of Sajeev Chacko and Bhalchandra Pujari. They were always ready to help me. They are responsible for almost all the computational knowledge,that I have. Chacko sir is not here now, but I still get help from him through Internet whenever required. I am really very grateful to both of them.

I am thankful to my lab mates Bhalchandra , Vaibhav ,Dhani,Amol,Ghazi and Ideh for generating academic and friendly atmosphere during the course of work. Special Thanks goes to Dhani !! She was another source of encouragement. Her appreciations and criticisms helped me improving my works in a great extent.

Thanks to VMD and GNUPLOT. All the pictures are generated using these softwares.

Above all,I am immensely grateful to my family for their constant support and encouragement starting from my childhood.

List of Figures

2.1	3D-Harmonic potential of $\omega = 6$ plotted at z=0 plane	16
2.2	First 20 wave functions of 3D-harmonic oscillator	18
2.3	First 11 wave functions of 3d-box	20
2.4	Changes in degeneracy pattern as we squeeze our system along one dimension	21
2.5	1st 20 wave functions for a quasi 2D system	22
2.6	Results for Gaussian potential	23
2.7	Double Gaussian wells	24
2.8	Seven bound states for a slightly merged double Gaussian well	24
2.9	Eight bound states for a half merged double Gaussian wells	25
3.1	Decomposition of $1/r$ potential in short range and long range parts	31
3.2	Periodicity of charge density	34
3.3	Potentials due to the periodic charge densities	34
3.4	Results for 3D-Gaussian Charge Density	37
3.5	The potential due to a charged Sphere	37
3.6	Results for Gaussian type potential	39
4.1	Exchange and Correlation potential	40
6.1	Charge density of a three electron quantum dot	44
7.1	The graphical interface of the program	47

List of Tables

2.1	Higher order finite-difference coefficients	12
2.2	Row-indexed storage example	14
2.3	Eigen values for the 3D-harmonic oscillator of $\omega = 6$	17
2.4	Eigen values in case of 3D infinite box of length 3.3 Bohr	19
3.1	The results for different κ in case of gaussian charge density	36
3.2	Results for general Gaussian type potentials	38
6.1	Energy of confined system for different number of electrons present	45

1

INTRODUCTION

The ultimate objective of this project is to build a three dimensional Density functional theory code for confined electron system , a model for three dimensional quantum Dot. The project is fully technical in nature and my main focus was towards the the development of the components required to build this code. So I will briefly discuss the key aspects of Density functional theory ,without going into much detail .

§1.1 THE MANY BODY SCHRÖDINGER EQUATION

We can describe any material system by a number of nuclei and electrons interacting through coulombic forces.We can write the Hamiltonian of such a system in the following general form:[1]

$$\hat{H} = - \sum_{I=1}^P \frac{\hbar^2}{2M_I} \nabla_I^2 - \sum_{i=1}^N \frac{\hbar^2}{2m} \nabla_i^2 + \frac{e^2}{2} \sum_{I=1}^P \sum_{J \neq I}^P \frac{Z_I Z_J}{|\vec{R}_I - \vec{R}_J|} + \frac{e^2}{2} \sum_{i=1}^N \sum_{j \neq i}^N \frac{1}{|\vec{r}_i - \vec{r}_j|} - e^2 \sum_{I=1}^P \sum_{i=1}^N \frac{Z_I}{|\vec{R}_I - \vec{r}_i|} \quad (1.1)$$

Where $\vec{R} = \{R_I\}, I = 1, 2, \dots, P$ is a set of P nuclear coordinates, and $\vec{r} = \{\vec{r}_i\}, i = 1, 2, \dots, N$ is a set of N electronic coordinates. Z_I and M_I are the P nuclear charges and masses ,respectively.All the ingredients are perfectly known and ,in principle , all properties can be derived by solving the many body Schrödinger equation:

$$\hat{H}\Psi_i(\vec{r}, \vec{R}) = E_i\Psi_i(\vec{r}, \vec{R}) \quad (1.2)$$

In practice , the problem is almost impossible to treat in a full quantum mechanical framework.The full Schrödinger equation cannot be easily decoupled into a set of independent equations. Hence in general we have to deal with $(3P + 3N)$ coupled degrees of freedom.The usual practice is to use some sensible approximations to solve this problem.

§1.2 ADIABATIC APPROXIMATION(BORN-OPPENHEIMER)

Mass of electrons are much smaller than that of nucleus . A proton has a mass larger than a electron by a factor of 1836.So we can always assume that nucleus stays in stationary state when describing the electronic motion at any instant of time. This is known as Born-Oppenheimer approximation. By applying this approximation we decouple the full wave function in to two parts

$$\Psi(\vec{R}, \vec{r}, t) = \Theta_m(\vec{R}, t)\psi_m(\vec{R}, \vec{r}) \quad (1.3)$$

Where $\Theta_m(\vec{R}, t)$ is the nuclear wave functions and $\psi_m(\vec{R}, \vec{r})$ is the electronic wave function of m th stationary state.

§1.3 SELF CONSISTENT FIELD CALCULATION

To solve the many body electronic wave function we assume that *many-electron wave function can be written as a simple product of one electron wave functions*. Each of these satisfies the one electron Schrödinger equation in an effective potential that takes into account the interaction with the other electrons in a mean field way. [1]

$$\begin{aligned} \psi(\vec{R}, \vec{r}) &= \prod_i \psi_i(\vec{r}_i) \\ \left(\frac{\hbar^2}{2m} \nabla_i^2 + V_{eff}^{(i)}(\vec{R}, \vec{r}) \right) \psi_i(\vec{r}) &= E_i \psi_i(\vec{r}) \end{aligned} \quad (1.4)$$

where effective potential V_{eff} is expressed as :

$$V_{eff} = V_{external} + V_{Hartree} + V_{Exchange} + V_{Correlation} \quad (1.5)$$

Except for the external potential all other terms in the equation are dependent on charge density functional $\rho(\vec{r})$, given by the equation:

$$\rho(\vec{r}) = -e \sum_i^{\text{occupied states}} |\psi_i|^2 \quad (1.6)$$

So the condition is the following:

- We need V_{eff} to solve the Schrödinger equation and get ψ .
- We need ρ to find V_{eff} .
- We need ψ to form ρ

So we need to guess a initial charge density and then run our calculation self consistently until the convergence is achieved.

§1.4 REQUIREMENTS TO BUILD THE CODE

There are mainly four components in any self consistency code ,

1. A 3D one electron Schrödinger equation solver
2. A program that can take ρ as input to generate the Hartree potential and Hartree energy.
3. An exchange potential and energy calculator
4. A correlation energy and potential generator.

All of these components are technically challenging to develop especially for three dimensional case. My project is mainly based on building the first two components in the list.

2

SCHRÖDINGER EQUATION

§2.1 INTRODUCTION

We want to solve the three dimensional one electron Schrödinger equation in real space . To do this we will use a method based on utilizing the *higher order finite-difference method*.In this method the unknown variables are the wave functions on a discrete grid.Within this approach,the *discrete* real space grid is the basis.Derivatives are approximated by a function which sums over the weighted values of the wave function at the neighboring points.To describe this method I will first take the one dimensional case and use only two neighbouring points to approximate the derivative.

§2.2 SOLVING SCHRÖDINGER EQUATION IN ONE DIMENSION

In one dimensional case Schrödinger equation is given as

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} + V\psi = E\psi \quad (2.1)$$

For computational purpose we will use a system of units in which \hbar, m, e all are 1 . The approximation of the second derivative can be found out using the Taylor series expansion of ψ

$$\frac{d^2\psi_i}{dx^2} = \frac{1}{h_x^2}(\psi_{i+1} - 2\psi_i + \psi_{i-1}) \quad (2.2)$$

Here $\psi_i \equiv \psi(x_i)$, value of the wave function at x_i . h_x is the distance between x_i and x_{i+1} . So our Schrödinger equation at the point $x = x_i$ is

$$\begin{aligned} & -\frac{1}{2} \frac{d^2\psi_i}{dx^2} + (V_i - E_i)\psi_i = 0 \\ & -\frac{\psi_{i+1}}{2h_x^2} + \left(\frac{1}{h_x^2} + V_i - E_i \right) \psi_i - \frac{\psi_{i-1}}{2h_x^2} = 0 \end{aligned} \quad (2.3)$$

If we have n_x number of points then we have n_x number of linear algebraic equation of n_x unknowns. These equations can be solved by generating a Hamiltonian matrix and diagonalizing it. Boundary conditions on wave function can be given from physical considerations.

$$\psi_1 = 0 \text{ and } \psi_{n_x} = 0$$

Above conditions also suggest, how the mesh should be chosen. From Eq.(2.3) the structure of the matrix can be easily understood:

$\hat{H}_{i,i} = \frac{1}{h_x^2} + V_i$ $\hat{H}_{i,i\pm 1} = -\frac{1}{2h_x^2}$ All other $\hat{H}_{i,j} = 0$		(2.4)
--	--	-------

Now the Schrödinger equation becomes

$$\hat{H}\psi_i = E_i\psi_i \quad (2.5)$$

So the eigen values (E_i) can be found by diagonalizing \hat{H} . The \hat{H} matrix looks like the following:

$$\hat{H} = \begin{pmatrix} \left(\frac{1}{h_x^2} + V_1\right) & -\frac{1}{2h_x^2} & 0 & 0 & 0 & \dots & 0 \\ -\frac{1}{2h_x^2} & \left(\frac{1}{h_x^2} + V_2\right) & \frac{1}{2h_x^2} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{2h_x^2} & \left(\frac{1}{h_x^2} + V_3\right) & \frac{1}{2h_x^2} & 0 & \dots & \vdots \\ \vdots & 0 & \frac{1}{2h_x^2} & \left(\frac{1}{h_x^2} + V_4\right) & \frac{1}{2h_x^2} & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2h_x^2} & \left(\frac{1}{h_x^2} + V_{n_x}\right) \end{pmatrix} \quad (2.6)$$

The dimension of the matrix is $n_x \times n_x$. From this analogy it is very easy to go for three dimensional case.

§2.3 SOLVING SCHRÖDINGER EQUATION IN 3-D

In three dimension the single electron Schrödinger equation is given as

$$-\frac{\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \psi + V\psi = E\psi \quad (2.7)$$

Using Eq.(2.2) we can write ,

$$\begin{aligned} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \psi(i, j, k) &= \frac{1}{h_x^2} (\psi(i-1, j, k) + \psi(i+1, j, k)) \\ &\quad + \frac{1}{h_y^2} (\psi(i, j-1, k) + \psi(i, j+1, k)) \\ &\quad + \frac{1}{h_z^2} (\psi(i, j, k-1) + \psi(i, j, k+1)) \\ &\quad - 2 \left(\frac{1}{h_x^2} + \frac{1}{h_y^2} + \frac{1}{h_z^2} \right) \psi(i, j, k) \end{aligned} \quad (2.8)$$

Here h_x, h_y, h_z are the mesh widths in x,y and z directions. If we have n_x, n_y, n_z points in x,y,z directions, then we have $n_x n_y n_z$ number of points in the mesh hence those many linear equations. Obviously the dimension of the Hamiltonian will also be $n_x n_y n_z \times n_x n_y n_z$. For building the Hamiltonian matrix we need to map the 3-D wave functions into one dimensional array. This can be done by following mapping:

$$r = ((i-1)n_y + j-1) n_z + k \quad (2.9)$$

Here $i = 1, 2, 3, \dots, n_x$; $j = 1, 2, 3, \dots, n_y$ and $k = 1, 2, 3, \dots, n_z$. After doing this mapping and considering Eq.(2.8) ,we can build the Hamiltonian in the same way as it was done in Eq.(2.6).

§2.4 HIGHER ORDER FINITE DIFFERENCE METHOD

Until now I was taking three points formula for second order derivative (see Eq.2.2).Now I will employ the higher order finite difference expression for second derivative. We approximate $\partial^2\psi/\partial x^2$ at (i, j, k) by

$$\frac{\partial^2}{\partial x^2} = \sum_{n=-N}^N C_n \psi(i+n h_x, j, k) + O(h_x^{2N+2}), \quad (2.10)$$

where N is a positive integer. This approximation is accurate to $O(h_x^{2N+2})$ upon the assumption that ψ can be approximated accurately by a power series in h_x . Until now we were working with N=1. Now we will go to N=6 ,thus taking $(6+1+1) = 13$ points formula for $\partial^2\psi/\partial x^2$, which is extremely accurate.

Table 2.1: Expansion coefficients for higher order finite-difference expressions of second derivative

N	C_i	$C_{i\pm 1}$	$C_{i\pm 2}$	$C_{i\pm 3}$	$C_{i\pm 4}$	$C_{i\pm 5}$	$C_{i\pm 6}$
1	-2	1					
6	$-\frac{5369}{1800}$	$\frac{12}{7}$	$-\frac{15}{56}$	$\frac{10}{189}$	$-\frac{1}{112}$	$\frac{2}{1925}$	$-\frac{1}{16632}$

Using the coefficients given in the table we are now ready to generate the 3D-Hamiltonian of the system.

§2.5 MATRIX STORAGE FORMAT

The Hamiltonian matrix is of dimension $n_x n_y n_z \times n_x n_y n_z$. For our final calculation we need at least a mesh of $100 \times 100 \times 100$ points. So the matrix has 10^{12} elements. To store such huge amount of data we need at least 8000 GB of memory !!! .Here I assumed each real number will take 8 byte space. Such huge memory is not available for us. So we use the fact that the matrix is symmetric and very much sparse therefore don't need to store the whole matrix but only the non zero elements in upper or lower half of the matrix.This decreases the memory requirement dramatically and finally only about 400 MB (in this case) is needed using the matrix storage format described below.

■■■2.5.1 ROW-INDEXED COMPACT STORAGE

To represent a matrix \mathfrak{A} of dimension $N \times N$,the row-indexed scheme [2] sets up two one dimensional arrays , call them sa and ija .The first of these stores the matrix element values and the second store integer indices .The storage rules are:

- The first N locations of sa store \mathfrak{A} 's diagonal matrix elements.(Note that diagonal elements are stored even if they are zero; this is at most a slight storage inefficiency, since diagonal elements are nonzero in most realistic applications.)
- Each of the first N locations of ija stores the index of the array sa that contains the first off-diagonal element of the corresponding row of the matrix. (If there are no off-diagonal elements for that row, it is one greater than the index in sa of the most recently stored element of a previous row.)
- Location 1 of ija is always equal to $N + 2$ (It can be read to determine N).
- Location $N + 1$ of ija is one greater than the index in sa of the last off-diagonal element of the last row. (It can be read to determine the number of nonzero elements in the matrix, or the logical length of the arrays sa and ija .) Location $N + 1$ of sa is not used and can be set arbitrarily.
- Entries in sa at locations $N + 2$ contain \mathfrak{A} 's off-diagonal values, ordered by rows and, within each row, ordered by columns.
- Entries in ija at locations $N + 2$ contain the column number of the corresponding element in sa .

The advantage of this scheme is that it requires storage of only about two times the number of nonzero matrix elements.Other methods can require as much as three or five times.As an

example of this scheme , consider the matrix

$$\mathfrak{A} = \begin{bmatrix} 3 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 7 & 5 & 9 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 6 & 5 \end{bmatrix}$$

The matrix can be stored in two one dimensional array each of length 11 as shown in table (2.5.1). Here x is an arbitrary value. Notice that, according to the storage rules, the value

Table 2.2: Row-indexed storage example

index k	1	2	3	4	5	6	7	8	9	10	11
$ija(k)$	7	8	8	10	11	12	3	2	4	5	4
$sa(k)$	3	4	5	0	5	x	1	7	9	2	6

of N (namely 5) is $ija(1) - 2$, and the length of each array is $ija(ija(1) - 1) - 1$, namely 11. The diagonal element in row i is $sa(i)$, and the off-diagonal elements in that row are in $sa(k)$ where k loops from $ija(i)$ to $ija(i + 1) - 1$, if the upper limit is greater or equal to the lower one (as in FORTRAN do loops).

Our matrix is symmetric so we only need to store the elements in the upper or lower half of the matrix. Thus while storing in this format we will assume that all lower half elements are zero and thus simply skip them.

§2.6 DIAGONALIZATION

We need to diagonalize the matrix stored in compressed format. This is the most difficult and expensive part of the whole method. Diagonalizing such a huge matrix of dimension of the order $10^6 \times 10^6$ is not possible by using normal text book methods. For this purpose we used a program based on Davidson's [3] algorithm for finding few selected eigen values of a large,sparse ,real symmetric matrix. The program also provides the corresponding eigen functions. There are large number of parameters involved in the program and we need to tune those properly for our purpose. This program needs a matrix vector multiplier program for our compressed matrix . Therefore I developed a program for this purpose which is given in the following section.

§2.7 MATRIX VECTOR MULTIPLICATION

For a general symmetric matrix \mathfrak{A} , which is multiplied by a vector \vec{x} to its right resulting in vector C , the program looks like the following.

```
do i=1,n
C(i)=A(i,i)*x(i) ! loop over diagonals
```

```

enddo
do i=1,n-1
  do j=i+1,n      ! loop over off diagonal terms
    C(i)=C(i)+A(i,j)*x(j)
    C(j)=C(j)+A(i,j)*x(i)
  enddo
enddo

```

For our case the matrix \mathfrak{A} is stored in row indexed format in two one dimensional array , hamilton_elements and hamilton_index. The modified code for this case is given below.

```

!-----PROGRAM BY SANDIP DE-----01.12.08 -----
! This is the matrix vector multiplication routine written
! in Fortran 90 , required by davidson diagonalization subroutine
! This subroutine multiplies the hamiltonian matrix of original dim N
! (which is stored in compressed row-index storage format )
! with a vector x of dim(N*M1,1) to its right to produce
! C(N*M1,1) vector
!
subroutine mat_vec_multi(N,M1,x,C)
use kind_param
use global_var
implicit none
integer(I4B)::M1,N ! M1=number of vectors multiplied at a time
real(R8B),dimension(N*M1),intent(in)::x
real(R8B),dimension(N*M1),intent(out)::C
integer(I4B):: i,k,l,index_m1
C=0.0
do l=1,N*M1
  i=l-N*((l-1)/(N))
  C(l)=hamilton_elements(i)*x(l)+C(l)    ! Start with diagonal term .
  index_m1=l-i
  !Loop over Off diagonal terms .
  do k=hamilton_index(i),hamilton_index(i+1)-1
    C(l)=C(l)+hamilton_elements(k)*x(index_m1+hamilton_index(k))
    C(index_m1+hamilton_index(k))=C(index_m1+hamilton_index(k))
    &           +hamilton_elements(k)*x(l)
  enddo
enddo
return
end subroutine mat_vec_multi
! -----

```

§2.8 TESTING THE PROGRAM

The program is tested by different means.I have studied different type of systems and verified the results with analytical results available. Here I will present only two simple cases , 3d infinite well and 3d-harmonic oscillator.

■■■2.8.1 3D-HARMONIC WELL

Consider a 3D harmonic potential given by the equation

$$V(x, y, z) = \frac{1}{2} (\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2) \quad (2.11)$$

This is the most general form of the harmonic potential.The program can take different values for ω_x, ω_y and ω_z .For testing I will use

$$\omega_x = \omega_y = \omega_z = 6$$

The potential at $z=0$ is plotted in Fig 2.1. The eigen energies are given as

$$E_{n_x, n_y, n_z} = \left(n_x + \frac{\hbar}{2} \right) \omega_x + \left(n_y + \frac{\hbar}{2} \right) \omega_y + \left(n_z + \frac{\hbar}{2} \right) \omega_z \quad (2.12)$$

Where n_x, n_y, n_z can take values 1, 2, 3, ... etc.

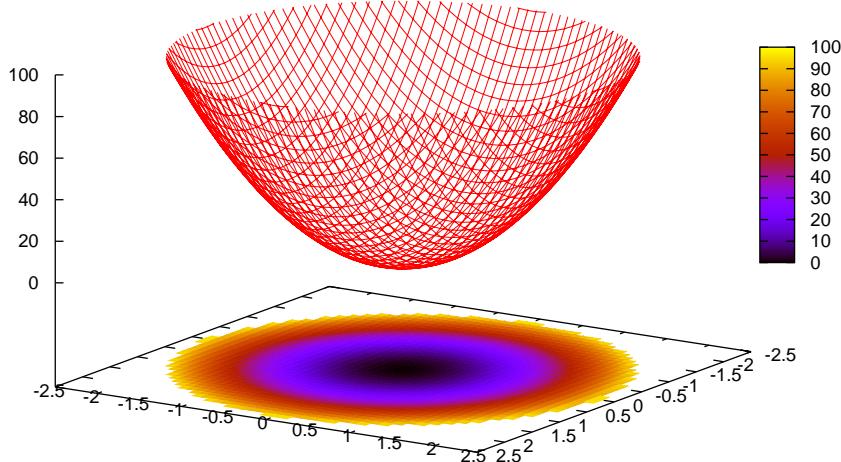


Figure 2.1: 3D-Harmonic potential of $\omega = 6$ plotted at $z=0$ plane

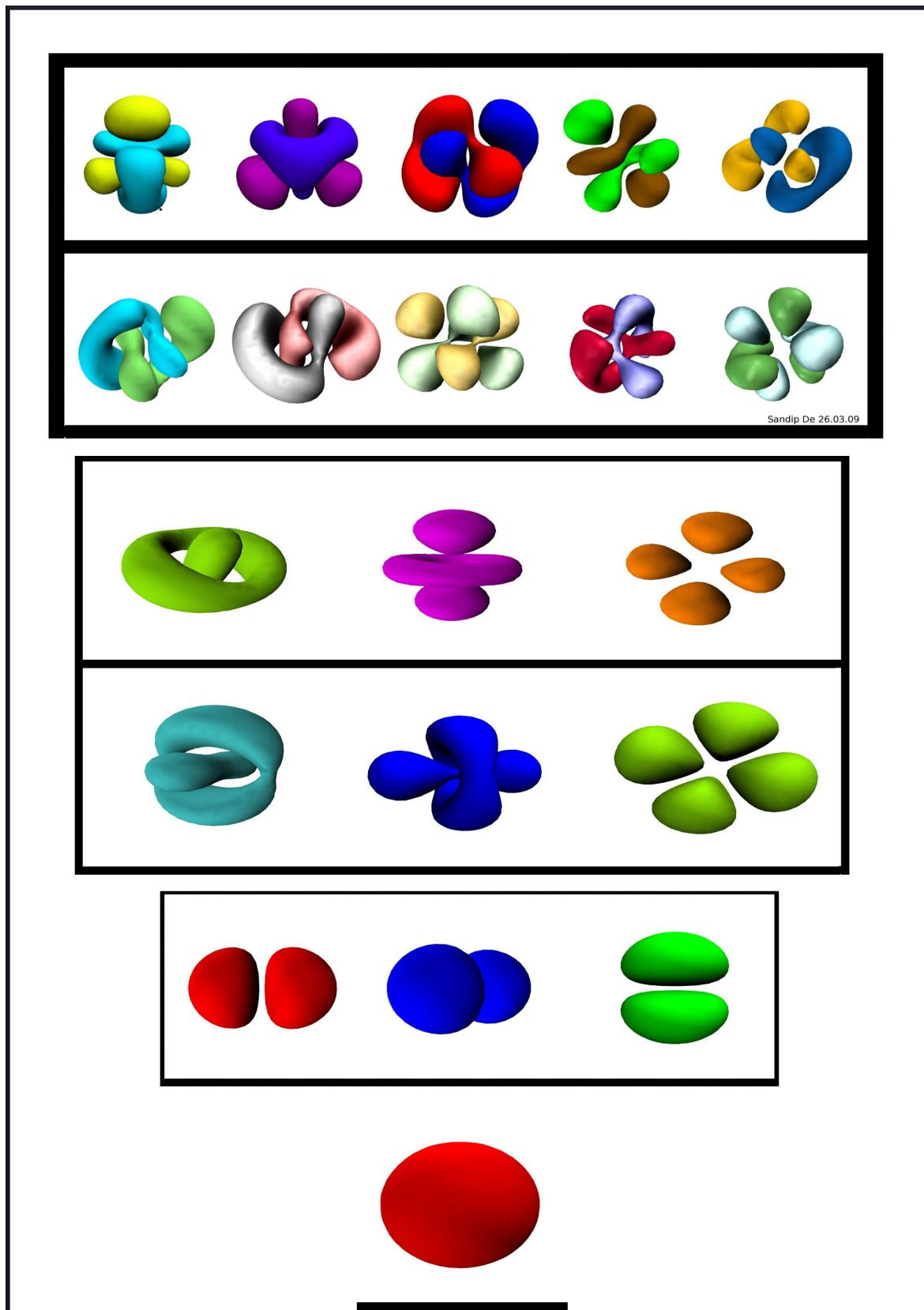
Table 2.3 shows the exact and obtained eigen values for the harmonic potential.Fig. 2.2 Shows the first 20 wave functions.

Table 2.3: Eigen values for the 3D-harmonic oscillator of $\omega = 6$

No	Exact Value	Value obtained	n_x, n_y, n_z	Degeneracy
1	9.000000	9.000092	0 0 0	1
2	15.000000	15.00004	0 0 1	
3	15.000000	15.00006	0 1 0	3
4	15.000000	15.00007	1 0 0	
5	21.000000	21.00005	0 0 2	
6	21.000000	21.00005	0 1 1	
7	21.000000	21.00009	0 2 0	6
8	21.000000	21.00010	1 0 1	
9	21.000000	21.00011	1 1 0	
10	21.000000	21.00014	2 0 0	
11	27.000000	27.00010	0 0 3	
12	27.000000	27.00016	0 1 2	
13	27.000000	27.00020	0 2 1	
14	27.000000	27.00023	0 3 0	
15	27.000000	27.00025	1 0 2	10
16	27.000000	27.00026	1 1 1	
17	27.000000	27.00031	1 2 0	
18	27.000000	27.00048	2 0 1	
19	27.000000	27.00061	2 1 0	
20	27.000000	27.00062	3 0 0	

So the program is able to produce the eigen values with high accuracy. This verifies the correctness of my program.

Figure 2.2: First 20 wave functions of 3D-harmonic oscillator. The wave functions are grouped according to their degeneracy. Different colors in each picture corresponding to opposite iso values



■■■2.8.2 3D INFINITE BOX

Next I will use my program to obtain the eigen values and eigen functions of a 3D infinite box. The potential energy for this system is defined as

$$V(x, y, z) = \begin{cases} 0 & \text{if } |x| < L_x, |y| < L_y \text{ and } |z| < L_z \\ \text{inf} & \text{if } |x| \geq L_x, |y| \geq L_y \text{ and } |z| \geq L_z \end{cases} \quad (2.13)$$

Where L_x, L_y, L_z are length , breadth and height of the box. I will test the result for a square box of length 3.3 Bohr. The analytical result for eigen energies is given by

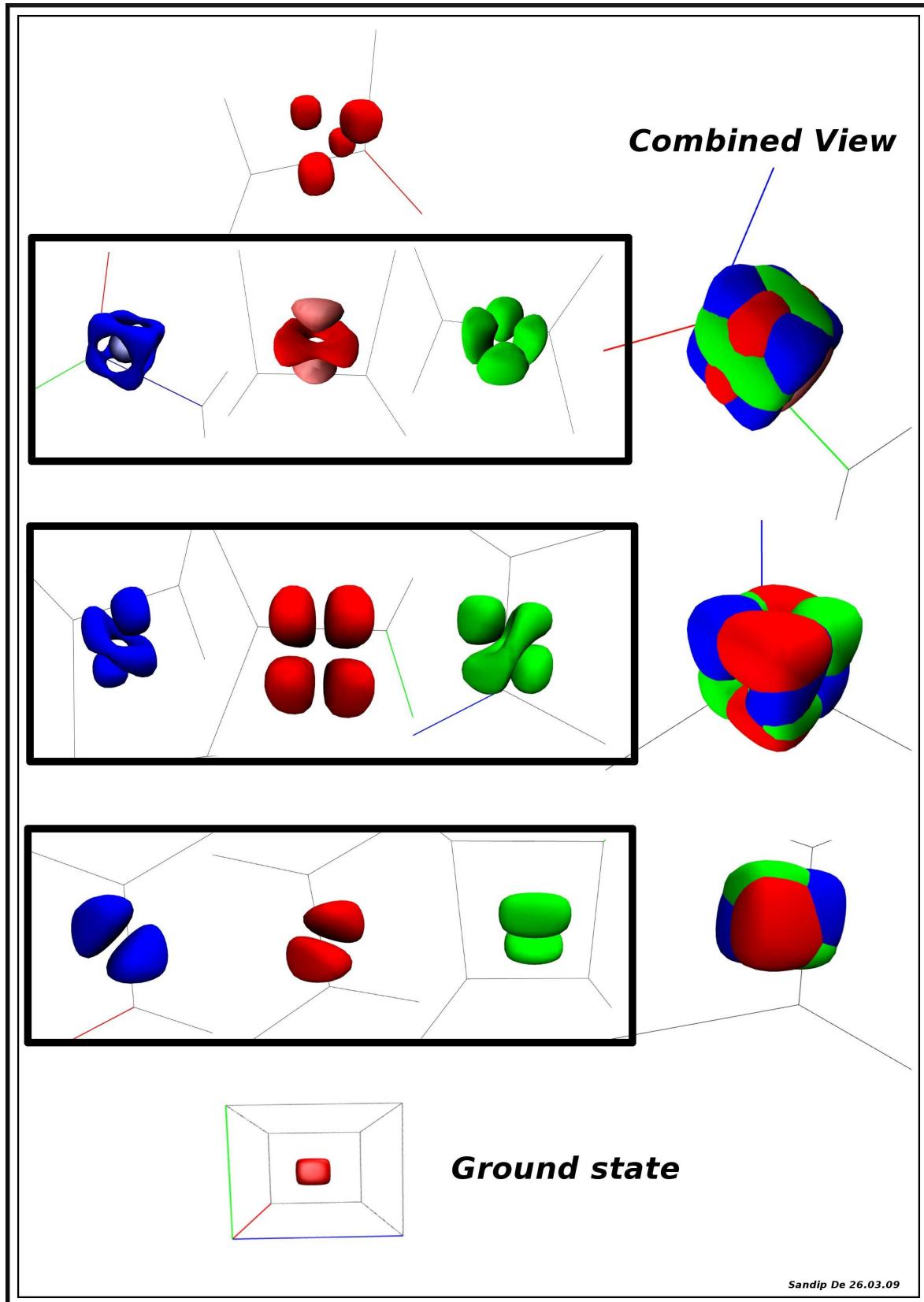
$$E_{n_x, n_y, n_z} = \frac{\pi^2 \hbar^2}{2m} \left(\left(\frac{n_x}{L_x} \right)^2 + \left(\frac{n_y}{L_y} \right)^2 + \left(\frac{n_z}{L_z} \right)^2 \right) \quad (2.14)$$

Where n_x, n_y, n_z can take values 0, 1, 2, 3, ... etc. Table 2.8.2 shows the analytical values and values obtained. Note that when we are saying that potential is *inf* out side the box , computationally we are putting a high value (in this case 100) out side the box. The results are matching with the analytical results upto three decimal places.Fig 2.8.2 shows first 11 eigen functions.

Table 2.4: Eigen values in case of 3D infinite box of length 3.3 Bohr

No	Analytical Value	Value obtained	n_x, n_y, n_z	Degeneracy
1	1.359450	1.359641	1 1 1	1
2	2.718899	2.719276	1 1 2	
3	2.718899	2.719276	1 2 1	3
4	2.718899	2.719276	2 1 1	
5	4.078349	4.068911	1 2 2	
6	4.078349	4.068911	2 1 2	3
7	4.078349	4.068925	2 2 1	
8	4.984649	5.018644	1 1 3	
9	4.984649	5.018644	1 3 1	3
10	4.984649	5.018644	3 1 1	
11	5.437799	5.438547	2 2 2	1
12	6.344098	6.343279	1 2 3	
13	6.344098	6.343280	1 3 2	
14	6.344098	6.343280	2 1 3	6
15	6.344098	6.343280	2 3 1	
16	6.344098	6.343280	3 1 2	
17	6.344098	6.343280	3 2 1	

Figure 2.3: First 11 wave functions of 3D-box. The wave functions are grouped according to their degeneracy. Each wave functions are plotted for both positive and negative iso values.

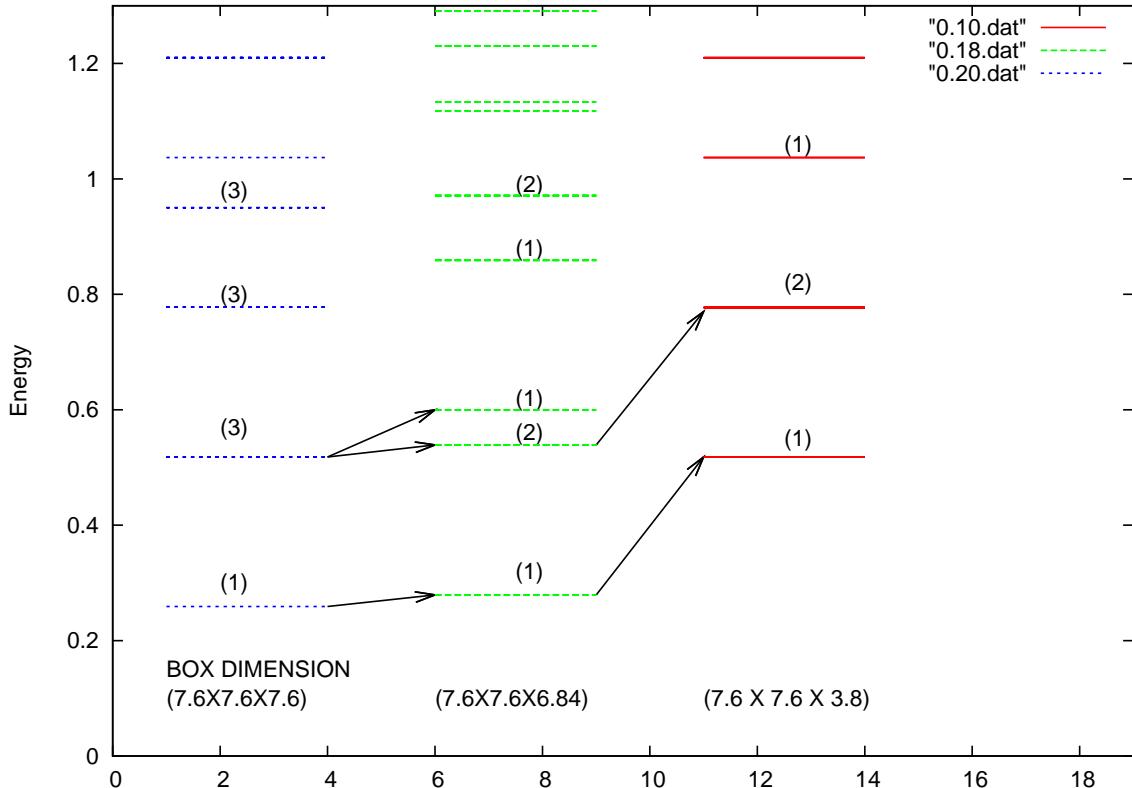


§2.9 FUN WITH WELLS

■ 2.9.1 A QUASI 2D CASE

As I mentioned, I tried to keep the program as general as possible. Now if we start squeezing the cubic box along any direction , some degeneracy will be lifted and the energy levels will start splitting. As we go on squeezing more and more ,the degeneracy pattern of 2D system will become more and more prominent. We should be able to find the wave function of this quasi-2D system.The wave functions should be analogous to the perfect 2D system.Fig 2.4 shows how the energy levels start splitting as we go on squeezing along one dimension.

Figure 2.4: Changes in degeneracy pattern as we squeeze our system along one dimension

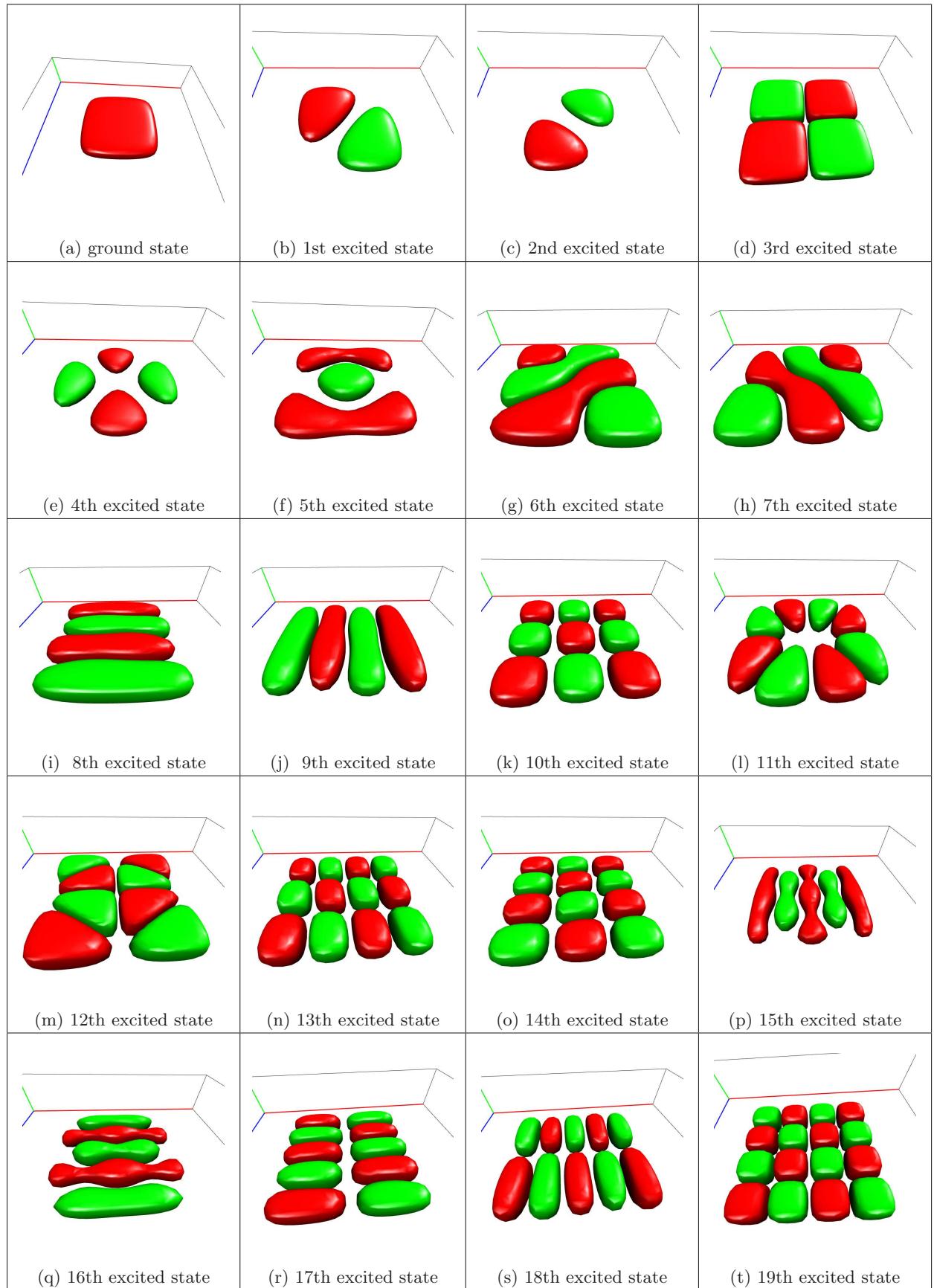


The degeneracy pattern of the perfect 2D well is given as

$$1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, \dots$$

Fig 2.5 shows same degeneracy patterns of the wave functions in the case of this quasi 2D infinite well.

Figure 2.5: First 20 eigen states of a quasi 2D infinite well are shown in the figure. The degeneracy patterns are matched with perfect 2D system. Red is corresponding to positive iso value and Green corresponds to negative iso value



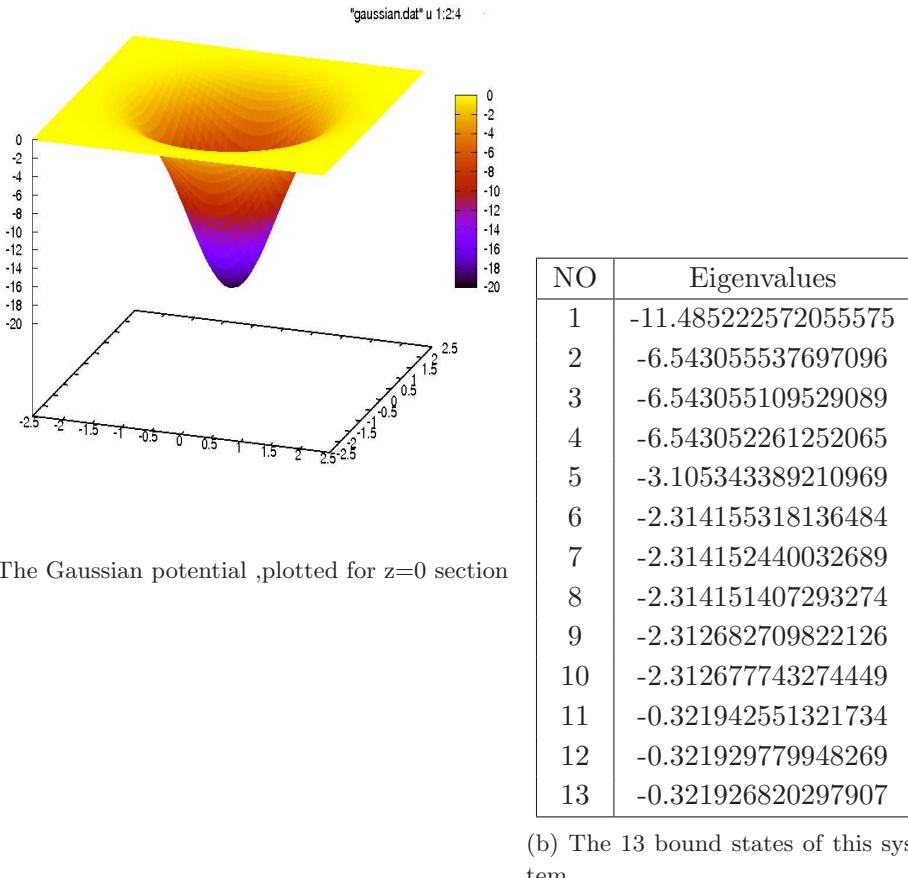
➡ 2.9.2 GAUSSIAN POTENTIAL

Let's turn our attention to some other type of potentials. Take a Gaussian

$$V(x, y, z) = -V_0 \exp(-\kappa^2(x^2 + y^2 + z^2))$$

Fig 2.9.2b shows the 13 bound states found for $\kappa = 2$ and $V_0 = 20$. The wave functions are

Figure 2.6: Results for Gaussian potential

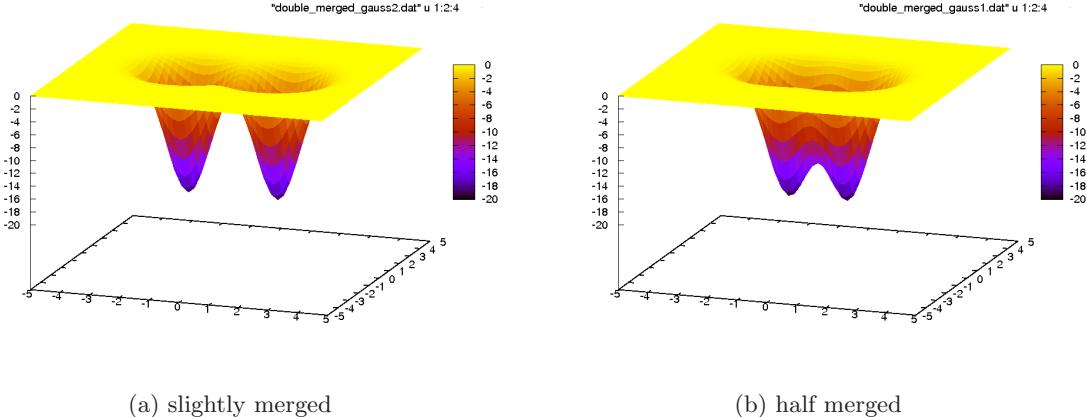


not given here as they look similar to that of harmonic oscillator.

➡ 2.9.3 DOUBLE GAUSSIAN WELL

Now for fun , consider two cases of double Gaussian wells , one slightly merged and one half merged as shown in Fig 2.7.

Figure 2.7: Double Gaussian wells in 3D, z=0 section is shown here.



The wave functions corresponding to potential in Fig 2.7.a are shown in Fig. 2.8. Note that the ground state in this case is degenerate !! This happened because in the ground state the two wells behave as two non-interacting , infinitely separated systems . As the energy increases electron starts feeling the effects of merging of the two wells. Fig 2.9 shows the wave functions corresponding to the half merged wells in Fig. 2.7.b.In this case we can see the effects on the ground state clearly.The results agree with the expected results.

Figure 2.8: Seven bound states for a slightly 3D-merged double Gaussian well.The graph shows z=0 section

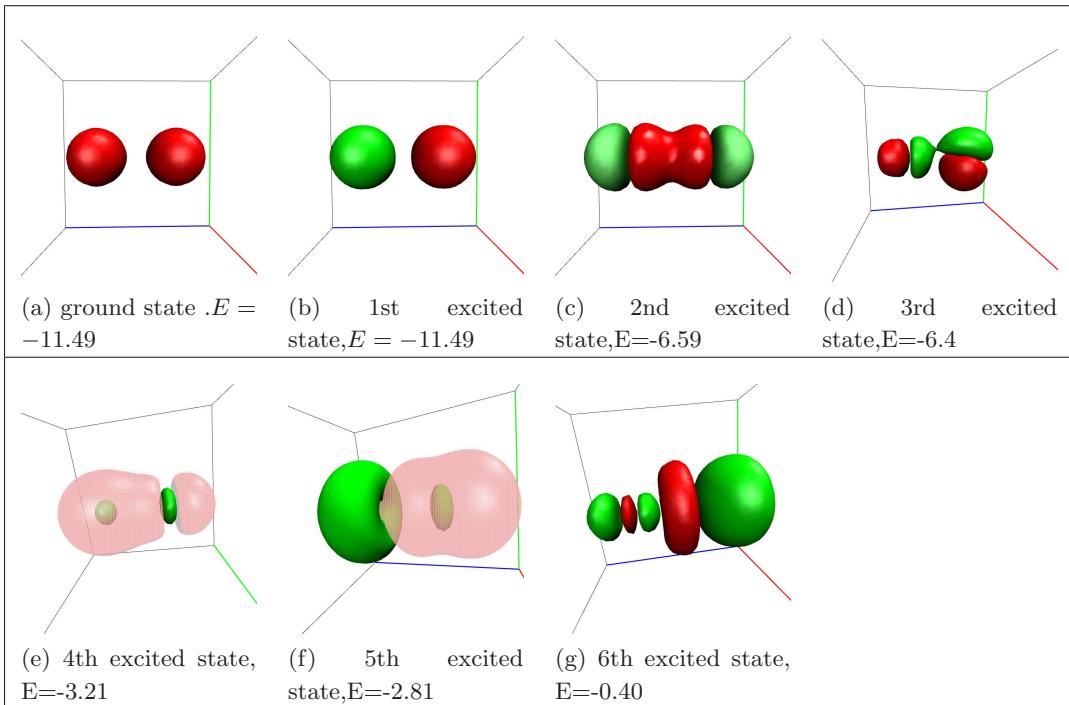
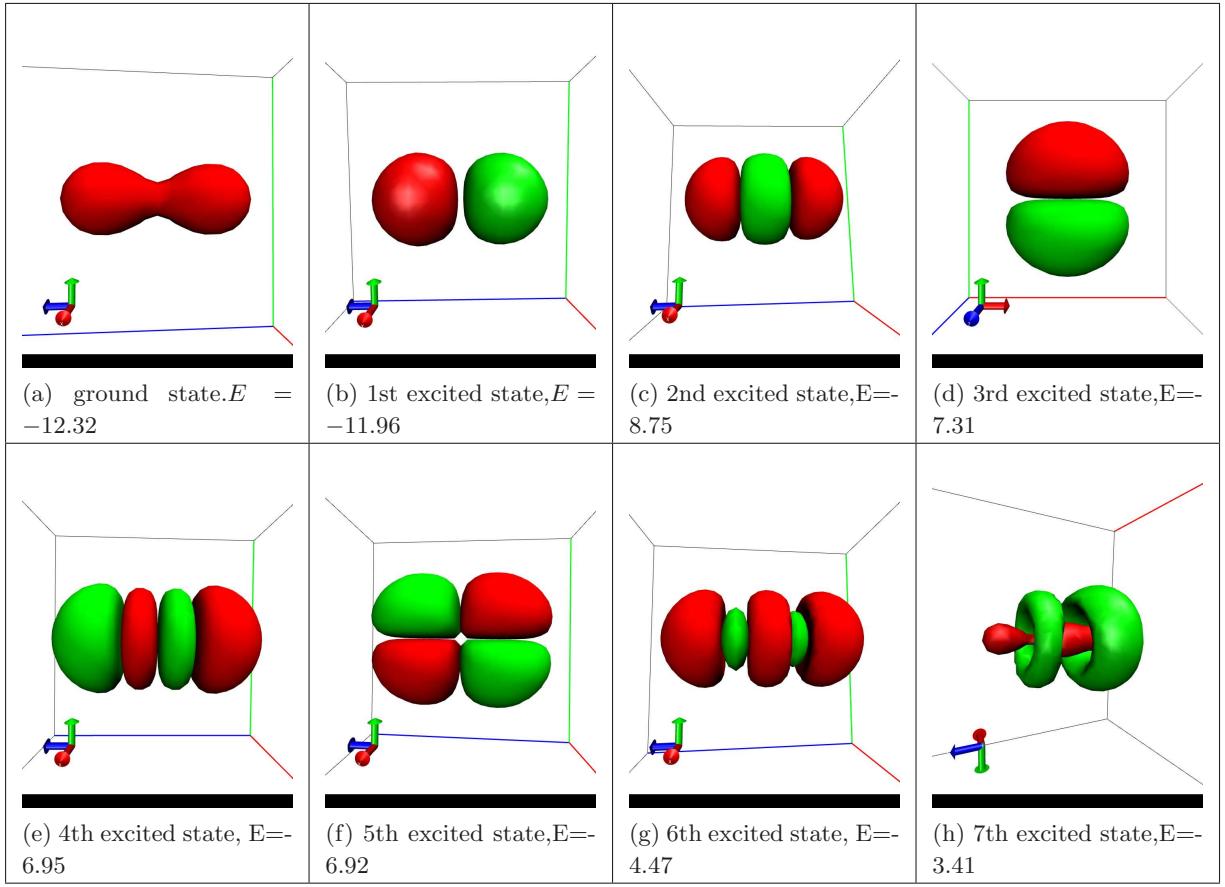


Figure 2.9: Eight bound states for a half merged 3D-double Gaussian wells. The graph shows $z=0$ section



§2.10 CONCLUSION

Now we have a program that solves 3D-Schrödinger equation. The program is not dependent on any type of symmetry. Given any type of meaningful potential it can solve the equation and give eigen values and eigen functions as outputs , So one of our main requirements for building the 3D-DFT code is now ready !!

3

HARTREE POTENTIAL

§3.1 INTRODUCTION

Hartree potential is defined as the potential energy due to the electrons present in the system. If we treat the electrons as a smooth distribution of negative charge with charge density $\rho(\vec{r})$, the potential energy of the system is given as

$$V_H(\vec{r}) = -e \iiint d^3 r' \rho(\vec{r}') \frac{1}{|\vec{r} - \vec{r}'|} \quad (3.1)$$

where with charge density $\rho(\vec{r})$ is given as

$$\rho(\vec{r}) = -e \sum_i^{\text{occupied state}} f_i |\psi_i(\vec{r})|^2 \quad (3.2)$$

f_i is the occupancy of i th state. Hartree energy is given as

$$\begin{aligned} E_H(\rho) &= \frac{-e}{2} \int d^3 r \rho(\vec{r}) V_H(\vec{r}) \\ &= \frac{e^2}{2} \int d^3 r \int d^3 r' \frac{\rho(\vec{r}) \rho(\vec{r}')}{|\vec{r} - \vec{r}'|} \end{aligned} \quad (3.3)$$

Equations 3.1 and 3.3 are nothing but the coulomb potential and energy of a particle of charge (-e), in system having charge density defined by Eq.3.2. To calculate the Hartree potential and energy we need a program that solves Poisson's equation. The program will take charge density as input and return potential and energy as output. Solving Poisson's equation is not very easy for finite sized system. Methods for solving Poisson's equation will be discussed in the following sections.

§3.2 SOLVING POISSON'S EQUATION

We know that relation between charge density ρ and potential ϕ is given by the Poisson's equation

$$\nabla^2 \phi(\vec{r}) = -4\pi \rho(\vec{r}) \quad (3.4)$$

This is a second order inhomogeneous differential equation and can be solved only if Dirichlet or Neumann Boundary conditions [4] are known. The conditions are the followings

- **Dirichlet Condition:** The potential is specified on the surface of the boundary.
- **Neumann Condition:** The electric field (normal derivative of potential) is specified on the surface.

The uniqueness theorem guarantees the uniqueness of the solution subject to any one of these two boundary conditions . Here we have the Dirichlet boundary condition.

$$\phi(\vec{r}) \rightarrow 0 \quad \text{as} \quad |\vec{r}| \rightarrow \infty$$

There are mainly three general ways to solve Poisson's equation.

1. The most obvious way is to use the integral form of the equation.

$$\phi(\vec{r}) = \int \frac{\rho(\vec{r} - \vec{r}')}{|\vec{r} - \vec{r}'|} d^3 r' \quad (3.5)$$

This calculation is computationally very expensive as it is a three dimensional integration. For a considerable accuracy we need at least a 3D grid of $50 \times 50 \times 50$ points. Even if we simply sum the values without going into sophisticated methods of integration , the number of calculations are at least $(50 \times 50 \times 50)^3 \approx 2 \times 10^{15}$ Hence this method is okay for small systems but not for large systems.

2. Another method is to solve this (eq.3.4) inhomogeneous second order differential equation using numerical methods . In our case the boundary condition is $\phi(\vec{r}) \rightarrow 0$ as $|\vec{r}| \rightarrow \infty$. If we want to use this method we have two choices:

- Use a very large grid so that on the boundary the potential can be taken to be zero.
- Use normal grid and somehow find out the boundary condition for the grid ie the potential on the boundaries.

Both of the choices are not very promising as these are also computationally very expensive.

3. We can use a reciprocal space based method to solve this problem. Take a Fourier transform both side of the Poisson's equation (3.4),we get

$$g^2 \phi(\vec{g}) = 4\pi \rho(\vec{g}) \quad (3.6)$$

where \vec{g} is reciprocal space vector and is given as

$$\vec{g} = \frac{2\pi n_x}{L_x} \hat{i} + \frac{2\pi n_y}{L_y} \hat{j} + \frac{2\pi n_z}{L_z} \hat{k} \quad \text{where } n_x, n_y, n_z = 0, \pm 1, \pm 2, \pm 3 \dots \quad (3.7)$$

L_x, L_y, L_z are the dimension of the unit cell or the periodicities of the system along the x ,y & z dimension. So we see that in reciprocal space the Poisson's equation becomes a linear algebraic equation , which is very easy to solve. Therefore we take a Fourier transform of $\rho(\vec{r})$ to get $\rho(\vec{g})$, calculate $\phi(\vec{g})$ and then come back to real space by taking a back Fourier transform of the potential. Computers can handle Fourier transforms very efficiently by using "Fast Fourier Transform" algorithm [5].Various libraries are available now ,that can be used to do this FFT.

From above we can use the third option.

Note that this option demands to have infinitely replicated periodic system and we are interested in finite sized systems. Therefore we needed to modify the method before using it for our purpose.

§3.3 SOLVING POISSON'S EQUATION FOR FINITE SIZED SYSTEM

As mentioned in the last section , solving Poisson's equation is very easy in reciprocal space using the equation 3.6 when we have *infinitely replicated periodic system* .But to employ that method in finite sized system,we need to develop a relation between the the results of *infinitely replicated periodic system* and that of *finite sized system*.Here we use the formalism as described in the paper [6] by G.J Martyna & M.E Tuckerman.First I will discuss the method in detail for general cases and then develop the algorithm for our purpose.

►3.3.1 DEVELOPMENT OF THE METHOD

A function $\rho(\vec{r})$,that satisfies periodic boundary conditions on the surface of a parallelepiped can be expanded in the complete orthonormal set of plane-waves[4].

$$\rho(\vec{r}) = \frac{1}{V} \sum_{\vec{g}} \bar{\rho}(\vec{g}) \exp(i\vec{g} \cdot \vec{r}) \quad (3.8)$$

Here \vec{g} is same as defined in equation 3.7.V is the volume of the parallelepiped.At this point I want you to be very clear about the two notations that I will use here :

$$\bar{\rho}(\vec{g}) = \int_V \rho(\vec{r}) \exp(-i\vec{g} \cdot \vec{r}) d^3 r \quad (3.9)$$

Note that the integration is on the finite volume. Therefore $\bar{\rho}(\vec{g})$ is the finite transform or Fourier series or plane wave representation and the other notation is ,

$$\tilde{\rho}(\vec{g}) = \int_{all\ space} \rho(\vec{r}) \exp(-i\vec{g} \cdot \vec{r}) d^3 r \quad (3.10)$$

which is the actual Fourier transform of $\rho(\vec{r})$ and the integration is on all space.

A function that vanishes at the boundaries can be regarded as satisfying periodic boundary

conditions . If we choose our unit cell such that the charge density goes to zero at the boundaries then we can impose periodic boundary conditions on ρ . The average potential energy taken over a density function $\rho(\vec{r})$ is given as

$$\begin{aligned}\langle \phi \rangle &= \frac{1}{2} \int_V d^3 r \int_V d^3 r' \phi(\vec{r} - \vec{r}') \rho(\vec{r}) \rho(\vec{r}') \\ &= \frac{1}{2V^2} \sum_{\hat{g}, \hat{g}'} \bar{\rho}(\vec{g}) \bar{\rho}(\vec{g}') \int_V d^3 r \int_V d^3 r' \\ &\quad \times \exp(i\vec{g} \cdot \vec{r}) \exp(i\vec{g}' \cdot \vec{r}') \phi(\vec{r} - \vec{r}')\end{aligned}\tag{3.11}$$

Here we have used equation 3.8. Now we can say that the potential $\phi(\vec{r})$ will also have the periodicity of the domain (the first/nearest image form), which means that the potential can be represented in the plane wave basis

$$\begin{aligned}\langle \phi \rangle^{(1)} &= \frac{1}{2V^3} \sum_{\hat{g}, \hat{g}', \hat{g}''} \bar{\rho}(\vec{g}) \bar{\rho}(\vec{g}') \bar{\rho}(\vec{g}'') \phi(\vec{g}'') \int_V d^3 r \int_V d^3 r' \\ &\quad \times \exp(i\vec{g} \cdot \vec{r}) \exp(i\vec{g}' \cdot \vec{r}') \exp(i\vec{g}'' \cdot (\vec{r} - \vec{r}')) \\ &= \frac{1}{2V} \sum_{\hat{g}} |\rho(\vec{g})|^2 \bar{\phi}(-\vec{g})\end{aligned}\tag{3.12}$$

The superscript (1) explicitly indicates the first image form. The expression for an infinitely replicated system can be obtained by replacing $\bar{\phi}(\vec{g})$ by $\tilde{\phi}(\vec{g})$ (see equation 3.10), and care has to be taken at $\vec{g} = 0$.

Clusters can be defined as systems confined to a finite volume large enough that the product of the density and observables vanishes at large spatial distances from the system's center of mass. Therefore, the size of the boundary can be permitted to take on the role of a convergence parameter ($\epsilon^{(1)}$) controlling the approach of the first image form to the infinite isolated system limit

$$\langle \phi \rangle^{(cluster)} = \langle \phi \rangle^{(1)} + \epsilon^{(1)}\tag{3.13}$$

We will choose our volume big enough that $\epsilon^{(1)} = 0$ condition is achieved. We discuss later how this can be done (see Sec. 3.3.4). For the moment we will assume that the cluster limit is achieved and further analysis will be performed on the first image form (see Sec. 3.3.4)

$$\langle \phi \rangle \equiv \langle \phi \rangle^{(cluster)} = \langle \phi \rangle^{(1)}\tag{3.14}$$

What we have to do is to take a parallelepiped of suitable size and find the potential using equation 3.12.

However ,it is difficult to obtain $\bar{\phi}(\vec{g})$ numerically or analytically. Therefore ,consider the two functions $\phi^{long}(\vec{r})$ and $\phi^{short}(\vec{r})$ such that

$$\begin{aligned}\phi(\vec{r}) &= \phi^{long}(\vec{r}) + \phi^{short}(\vec{r}) \\ \bar{\phi}(\vec{g}) &= \bar{\phi}^{long}(\vec{g}) + \bar{\phi}^{short}(\vec{g})\end{aligned}\tag{3.15}$$

We require that $\phi^{short}(\vec{r})$ vanish exponentially quickly at large distances from the center of the parallelepiped and that $\phi^{long}(\vec{r})$ contain the long range dependence of the full potential, $\phi(\vec{r})$. With these two requirements, it is possible to write

$$\begin{aligned}\bar{\phi}^{short}(\vec{g}) &= \int_V \exp(-i\vec{g} \cdot \vec{r}) \phi^{short}(\vec{r}) d^3r \\ &= \int_{all\ space} \exp(-i\vec{g} \cdot \vec{r}) \phi^{short}(\vec{r}) d^3r + \epsilon(\vec{g}) \\ &= \tilde{\phi}^{short}(\vec{g}) + \epsilon(\vec{g})\end{aligned}\quad (3.16)$$

with an exponentially small error, $\epsilon(\vec{g})$, provided the range of $\phi^{short}(\vec{r})$ is chosen to be small compared to the size of the parallelepiped. $\phi^{short}(\vec{r})$ will be made a function of a convergence parameter, α_{conv} , which can be used to adjust the range of $\phi^{short}(\vec{r})$ such that $\epsilon(\vec{g}) \approx 0$ and hence the error $\epsilon(\vec{g})$ will be neglected in the following sections. We now have

$$\bar{\phi}^{short}(\vec{g}) = \tilde{\phi}^{short}(\vec{g}) \quad (3.17)$$

Therefore we can write from equation 3.15 as ,

$$\begin{aligned}\bar{\phi}(\vec{g}) &= \bar{\phi}^{long}(\vec{g}) + \tilde{\phi}^{short}(\vec{g}) \\ &= [\bar{\phi}^{long}(\vec{g}) - \tilde{\phi}^{long}(\vec{g})] + \tilde{\phi}^{long}(\vec{g}) + \tilde{\phi}^{short}(\vec{g}) \\ &= \tilde{\phi}^{screen}(\vec{g}) + \tilde{\phi}(\vec{g})\end{aligned}\quad (3.18)$$

where $\tilde{\phi}(\vec{g}) = \tilde{\phi}^{long}(\vec{g}) + \tilde{\phi}^{short}(\vec{g})$ is the Fourier transform of the full potential , evaluated at the quantized \vec{g} and

$$\tilde{\phi}^{screen}(\vec{g}) = \bar{\phi}^{long}(\vec{g}) - \tilde{\phi}^{long}(\vec{g}) \quad (3.19)$$

Using these equations in equation 3.12 , we get

$$\langle \phi \rangle = \frac{1}{2V} \sum_{\hat{g}} |\rho(\vec{g})|^2 [\tilde{\phi}(-\vec{g}) + \tilde{\phi}^{screen}(-\vec{g})] \quad (3.20)$$

The new function appearing in the average cluster potential energy is the difference between the Fourier series and Fourier transform of the long range part of the potential energy evaluated at the quantized \vec{g} [see eq. 3.19] and is referred as screening function because it is constructed to "screen" the interaction of the system with an infinite array of periodic images.

The form of average potential energy in reciprocal space given by Eq.(3.20) is useful because it provides a direct relationship between the formulas for finite and infinitely replicated periodic system(for which $\tilde{\phi}^{screen}(-\vec{g}) = 0$ for all \vec{g}).There are two components in the expression for the screening function (Eq.(3.19)) : $\tilde{\phi}^{long}(\vec{g})$,which is assumed to be known analytically, and $\bar{\phi}^{long}(\vec{g})$,which may need to be evaluated numerically by taking a 3D FFT.

3.3.2 USING THE METHOD FOR COULOMB POTENTIAL

In last section it was assumed that the total potential can be decomposed into two terms having long and short range dependence(see Eq. 3.15). The functions will be made up of a convergence parameter α which will be adjusted to a value so that the Eq. (3.17) is satisfied. Consider the Coulomb potential, $\phi^{coul}(\vec{r}) = 1/|\vec{r}|$. This can be written as

$$\begin{aligned} \frac{1}{r} &= \frac{\text{erf}(\alpha r)}{r} + \frac{\text{erfc}(\alpha r)}{r} \\ &= \phi^{long,Coul}(\vec{r}) + \phi^{short,Coul}(\vec{r}) \end{aligned} \quad (3.21)$$

Here erf stands for error function and given as

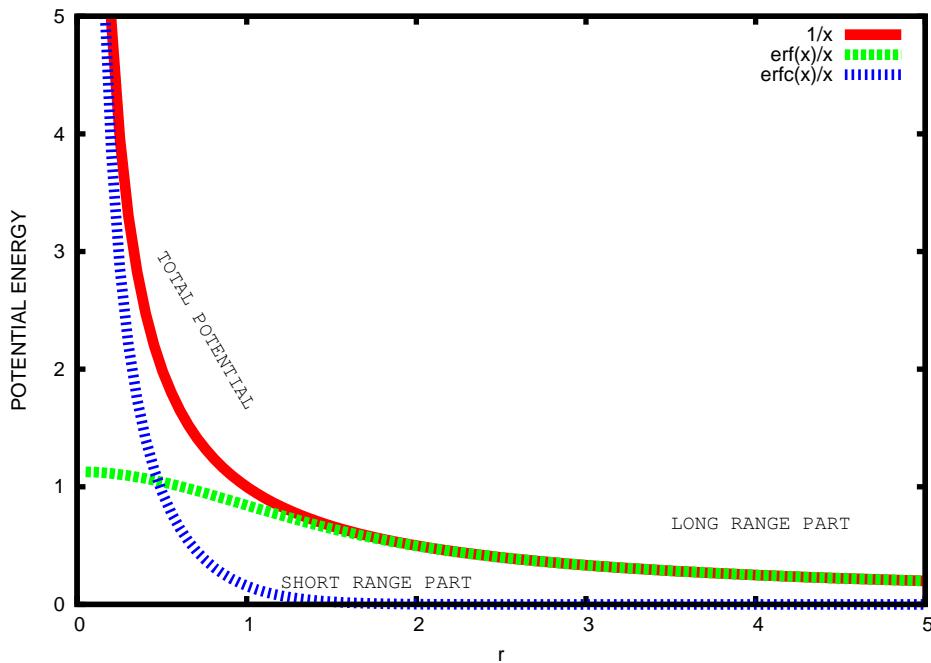


Figure 3.1: Decomposition of $1/r$ potential in short range and long range parts using $\alpha = 1$. We can reduce the range of the short range part by increasing the value of α and vice versa.

$$\begin{aligned} \text{erf}(\alpha r) &= \frac{2}{\sqrt{\pi}} \int_0^{\alpha r} \exp(-x^2) dx \\ \text{erfc}(\alpha r) &= 1 - \text{erf}(\alpha r) \\ &= \frac{2}{\sqrt{\pi}} \int_{\alpha r}^{\infty} \exp(-x^2) dx \end{aligned} \quad (3.22)$$

The convergence parameter α has been introduced to control the range (relative to boundary) on which $\phi^{short}(\vec{r})$ is nonzero. It is found [6] that taking $\alpha L \geq 7$, where L is the smallest side

of the boundary parallelepiped,gives complete convergence. The Coulomb potential satisfies the periodic boundary conditions required by the Fourier Series expansion [4].However the first derivative of coulomb potential is not continuous and this will lead to slow convergence of Fourier Series at small cell size.

Although we can choose large α to make the short range part go to zero very quickly, but for large α , $erf(\alpha r)/r \rightarrow 1/r$ and cartesian integration becomes intractable.It is found that $\alpha L \approx 7$ is sufficiently large to force the truncation error to be zero and Eq. (3.17) is satisfied and also sufficiently small to allow efficient integration on Cartesian grid.

Now we have following analytical results available:

$$\begin{aligned}\tilde{\phi}^{Coul}(\vec{g}) &= \frac{4\pi}{g^2} \\ \tilde{\phi}^{Coul,long}(\vec{g}) &= \frac{4\pi}{g^2} \exp\left(-\frac{g^2}{4\alpha^2}\right) \\ \tilde{\phi}^{Coul,short}(\vec{g}) &= \tilde{\phi}^{Coul}(\vec{g}) - \tilde{\phi}^{Coul,long}(\vec{g}) \\ &= \frac{4\pi}{g^2} \left(1 - \exp\left(-\frac{g^2}{4\alpha^2}\right)\right)\end{aligned}\quad (3.23)$$

We will not use the Eq.(3.20) as that equation is useful when we are modifying an existing code of a periodic system to use that code for our finite sized system.Here we will use the most general form of the equation Eq.(3.12).We want to calculate $\bar{\phi}^{coul}(\vec{g})$.

$$\begin{aligned}\bar{\phi}^{coul}(\vec{g}) &= \bar{\phi}^{long,Coul}(\vec{g}) + \bar{\phi}^{short,Coul}(\vec{g}) \\ &= \bar{\phi}^{long,Coul}(\vec{g}) + \tilde{\phi}^{short,Coul}(\vec{g})\end{aligned}\quad (3.24)$$

The last term of Eq (3.24) is analytically known (see Eq.(3.23))and the first term can be evaluated by taking a 3D-FFT of the long range part.

►3.3.3 SINGULARITY PROBLEM

In developing the method I did not mention how to handle the singularity of $1/r$ at $r=0$. We did it without mentioning, by decomposing the $1/r$ into two parts.The short range part now contains the singularity and long range part is smooth continuous function (see fig. 3.1) . We used the analytical result for $\tilde{\phi}^{short,Coul}(\vec{g})$ which can not be evaluated by FFT as it would give poor accuracy due to singularity at origin. This is also the reason for not calculating the $\tilde{\phi}^{Coul}(\vec{g})$ directly by FFT. But we can calculate $\bar{\phi}^{long,Coul}(\vec{g})$ by FFT as the long range part

has no singularity. Next we examine the value of $\tilde{\phi}^{short,Coul}(\vec{g})$ at $\vec{g} = 0$.

$$\begin{aligned}
 \tilde{\phi}^{Coul,short}(\vec{g}) &= \tilde{\phi}^{Coul}(\vec{g}) - \tilde{\phi}^{Coul,long}(\vec{g}) \\
 &= \frac{4\pi}{g^2} \left(1 - \exp \left(-\frac{g^2}{4\alpha^2} \right) \right) \\
 &= \frac{4\pi}{g^2} \left(1 - \left\{ 1 - \frac{g^2}{4\alpha^2} + \frac{1}{2!} \left(\frac{g^2}{4\alpha^2} \right)^2 - \dots \right\} \right) \\
 &= -\frac{\pi}{\alpha^2} + \frac{4\pi}{g^2} \left(\frac{1}{2!} \left(\frac{g^2}{4\alpha^2} \right)^2 - \frac{1}{3!} \left(\frac{g^2}{4\alpha^2} \right)^3 + \dots \right) \\
 \tilde{\phi}^{Coul,short}(\vec{0}) &= -\frac{\pi}{\alpha^2}
 \end{aligned} \tag{3.25}$$

So we can see that the $\tilde{\phi}^{Coul,short}(\vec{0})$ is nonsingular. Here the whole formalism is completed and we are now ready to develop the algorithm based on this theory.

■■■3.3.4 CHOOSING THE MESH SIZE

In section (3.3) I assumed that we will choose our mesh size big enough to make the error $\epsilon^{(1)} = 0$ in the Eq.(3.13). Now I return to the question , about the optimized size of the mesh or the parallelepiped. For this consider a Gaussian charge density

$$\rho(\vec{r}) = \left(\frac{\kappa}{\pi} \right)^{3/2} \exp(-\kappa^2 r^2)$$

First we need to make the mesh big enough ,so that the charge density goes to zero on the boundaries and thus we can employ the periodic boundary condition on $\rho(\vec{r})$. For $\kappa = 1$ we can see in the Fig (3.2) , $L > 5$ is sufficient ,where L is the minimum length of the side of our parallelepiped .

Although we impose periodic boundary condition on our charge density, we actually want to calculate the potential due to only our charge density and not due to all periodic images Hence we want to minimize the contribution of the infinite periodic images to our potential. This can be done by taking a larger box such that the effect of the image charge densities are less.By doing this we want to minimize the effect of the other periodic images to the short range part of our required potential ,so that the only long range part carries the effect which can be eliminated by developing the screening function (Eq. 3.19) .

The potential due to the Gaussian charge density is given as $\phi(\vec{r}) = \text{erf}(\kappa r)/r$

Fig (3.4a) shows that the effect due to the image charges are significant if we take L=5 . So we need to take a bigger mesh. Experience shows that for for Gaussian charge density $\kappa L > 15$ is sufficient (see Fig (3.4b)).

Normally it can be said that if we take L=a , so that the charge density goes to zero before boundaries ,then we want L=2a to carry out our calculation.This can be take as a thumb rule and we will do a double grid transformation for this calculation.Note that I am not saying that

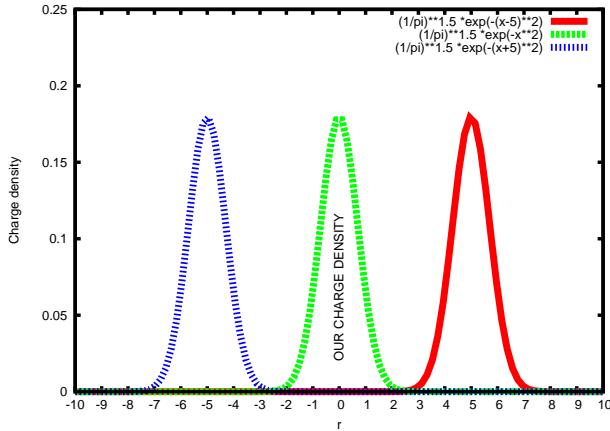
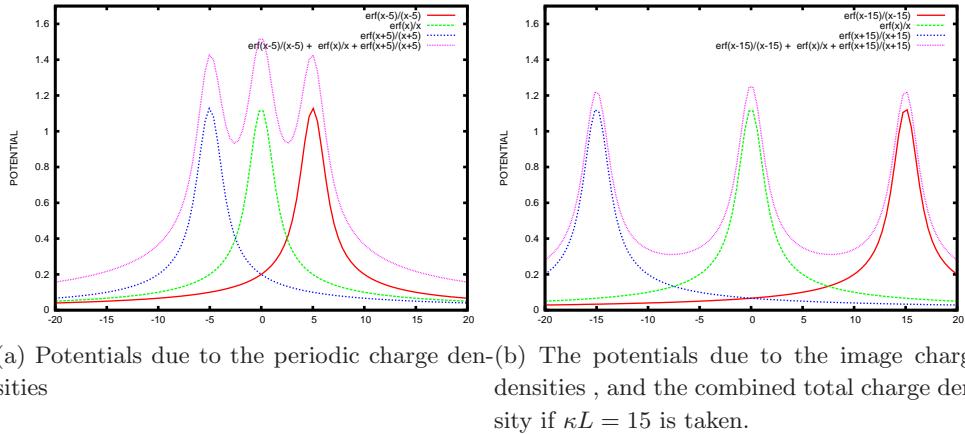


Figure 3.2: The middle one is our original charge density. We can see if we take our minimum length of the side of our parallelepiped to be 5 , we can make charge density periodic.

Figure 3.3: Potentials due to the periodic charge densities



this will completely eliminate the effect of the infinite periodic images; because the potential has a $1/r$ dependence thus not zero at any finite r . The effect of the periodic images will be finally "screened" by introducing the screening function (see Eq.(3.19)).

§3.4 ALGORITHMIC DEVELOPMENT

In last few sections I have developed the theory required to develop the algorithm for my program.I assume that the input charge density is defined over a small mesh which is large enough , so that the charge density goes to zero before boundary. But as we saw in Sec 3.3.4 , we need a double sized mesh to carry out our calculation.So we will do a double grid transformation, carry out our calculation on that bigger mesh and finally come back to our original small mesh to produce the potential in the original mesh.So the algorithm is as

follows:

1. Input $\rho(x, y, z)$ in a mesh of dimension (L_x, L_y, L_z) . The number of points in x,y and z directions are (n_x, n_y, n_z) . The grid width in x,y and z directions are (dx, dy, dz) .
2. Do a double grid transformation. To do this insert the original mesh at the center of a mesh having dimension $(2L_x, 2L_y, 2L_z)$. The new mesh has $(2n_x, 2n_y, 2n_z)$ number of points and same grid widths (dx, dy, dz) . So the mesh has Volume=8 original mesh Volume. In this mesh the value of the charge density outside our original mesh is assigned to zero.
3. Take 3D-FFT of $\rho(x, y, z)$ to get $\rho(\vec{g})$. \vec{g} is defined in Eq.3.7. The 3D FFT has been done using the library package "FFTW 3.2" [7].
4. Assign the analytical value of $\tilde{\phi}^{short}(\vec{g})$ in the mesh from Eq.(3.23). At $\vec{g} = 0$ assign the value $\tilde{\phi}^{short}(0) = \pi/\alpha^2$ (See Eq. (3.25)).
5. Take the long range part of $1/r$ to be $\phi^{long}(\vec{r}) = erf(\alpha r)/r$ and take 3D-FFT to get $\bar{\phi}^{long}(\vec{g})$. At $r=0$ assign the analytical value of $\phi^{long}(0) = 2\alpha/\sqrt{\pi}$.
6. Get $\bar{\phi}(\vec{g})$ by adding $\tilde{\phi}^{short}(\vec{g})$ and $\bar{\phi}^{long}(\vec{g})$.
7. Calculate the average potential of the system using Eq.(3.12)

$$\langle \bar{\phi}(\vec{g}) \rangle = |\rho(\vec{g})|^2 \times \bar{\phi}(\vec{g})$$

Where I have used $\bar{\phi}(-\vec{g}) = \bar{\phi}(\vec{g})$.

8. Take a inverse FFT of $\langle \bar{\phi}(\vec{g}) \rangle$ to get $\langle \phi(\vec{r}) \rangle$.
9. Come back to the original small mesh by discarding the values of potential for the points which are not inside of our original small mesh.
10. Calculate energy using equation

$$E_H(\rho) = \frac{-e}{2} \int d^3r \rho(\vec{r}) V_H(\vec{r})$$

These are the main steps in the program. Other than these there are some technical steps like producing the input data in a format required by the FFTW package , converting the output of the FFTW program in a format of our interest ,etc .

§3.5 TESTING THE PROGRAM

The results of the program is verified by different means. The potential is checked with analytical result point by point and maximum and minimum error is calculated. Here I will present some cases.

➡ 3.5.1 GAUSSIAN CHARGE DENSITY

The input Gaussian charge density and corresponding potential is given as

$$\rho(\vec{r}) = \left(\frac{\kappa^2}{\pi} \right)^{3/2} \exp(-\kappa^2 r^2)$$

$$\phi(\vec{r}) = \frac{\text{erf}(\kappa r)}{r}$$

The mesh parameters are given as

$(n_x, n_y, n_z) = (100, 100, 100)$
$(dx, dy, dz) = (0.2, 0.2, 0.2)$
$(L_x, L_y, L_z) = (20, 20, 20)$ Bohr

Table 3.1: The results for different κ in case of gaussian charge density

κ	Max Error	Min Error
0.10	0.279E-01	0.218E-01
0.20	0.145E-02	0.763E-03
0.30	0.941E-05	0.355E-05
0.40	0.883E-08	0.244E-08
0.50	0.118E-11	0.243E-12
0.60	0.444E-15	0.000E+00
0.70	0.555E-15	0.000E+00
0.80	0.666E-15	0.000E+00
0.90	0.777E-15	0.000E+00
1.00	0.999E-15	0.000E+00
1.10	0.999E-15	0.000E+00

Table 3.1 shows that maximum accuracy is achieved for $\kappa = 0.6$, For which $\kappa L = 12$. Although it is expected that if we take large κ we will get better result , in practice it depends on the mesh width. As κ increases we need to make the mesh finer i.e. smaller mesh widths . For this reason we get a decrease in accuracy after increasing κ more than 0.6 . Any way the result is correct up to 5th decimal places in the worst case. The input charge density and output potential for a general case are shown in the Fig (3.4).

➡ 3.5.2 UNIFORMLY CHARGED SPHERE

Here I will verify the output of the program in the case of a uniformly charged sphere of radius a . In this problem the charge density is given as

$$\rho(\vec{r}) = \begin{cases} \text{Const} & \text{if } r \leq a \\ 0 & \text{if } r > a \end{cases}$$

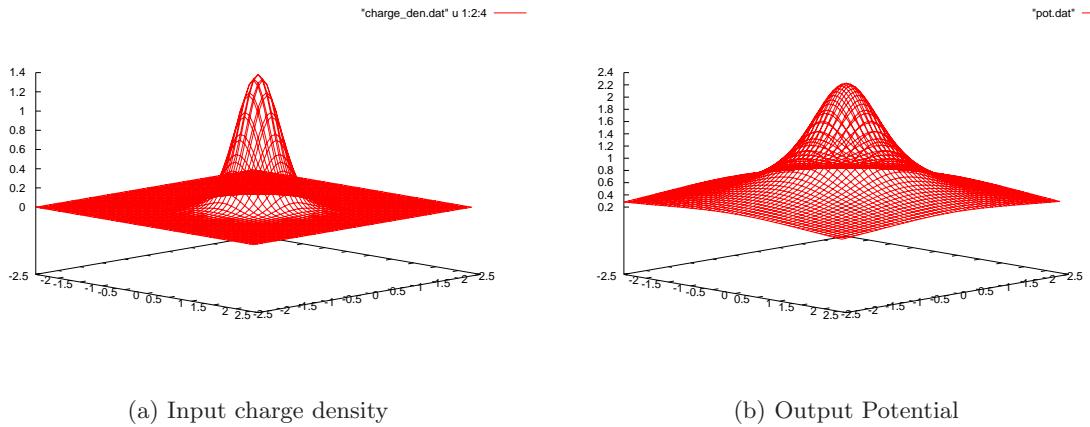


Figure 3.4: Input and Output of the program in case of 3D-Gaussian Charge Density for $L = 5 \kappa = 2$. The graph shows z=0 section

The analytical result for potential is given as

$$\phi(\vec{r}) = \begin{cases} \frac{Q}{2a} \left(3 - \frac{r^2}{a^2} \right) & \text{When } r \leq a \\ \frac{Q}{r} & \text{When } r > a \end{cases}$$

Where Q is the total charge of the sphere , $Q = \frac{4}{3}\pi a^3 \rho$. Output of the program is checked with these analytical results and found that the result is matching up to 5th decimal place. The potential is shown in the figure (3.5).

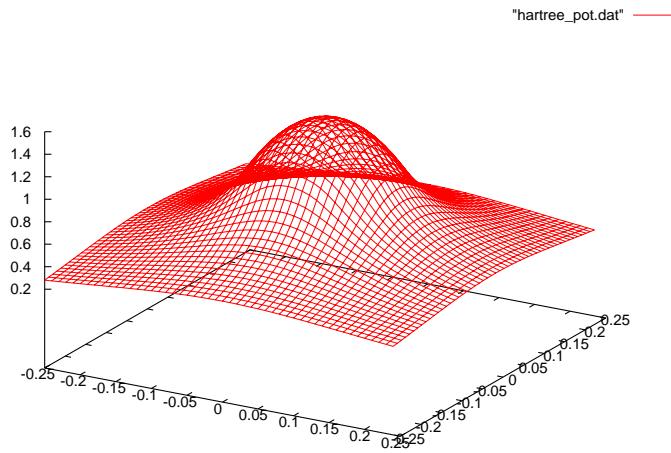


Figure 3.5: The potential due to a charged Sphere

The accuracy in case of Gaussian charge density was much better than this.The reason

behind this is, we can not build a perfect sphere in a Cartesian grid. It will always be a polygon. As we decrease our mesh width the polygon will be better approximation to a circle ,thus we will say that we have built a sphere. Here I have used a mesh width of 0.1 . If I decrease this width I will surely get a better match with the analytical result. Note that we are not interested in constant charge densities and using it to test the program.

3.5.3 GAUSSIAN TYPE POTENTIAL

Consider a general potential of the form

$$\phi(\vec{r}) = \exp(-\kappa r^n)$$

where $n = 2,3,4,\dots$ If we operate with ∇^2 on the potential we get.

$$\begin{aligned}\nabla^2 \phi(\vec{r}) &= \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} (\exp(-\kappa r^n)) \right) \\ &= -\kappa n r^{(n-2)} (n + 1 - \kappa n r^n) \exp(-\kappa r^n)\end{aligned}$$

So from Poisson's equation (Eq. 3.4) we get corresponding

$$\rho(\vec{r}) = -\frac{\kappa n}{4\pi} r^{(n-2)} (n + 1 - \kappa n r^n) \exp(-\kappa r^n)$$

I will give this ρ as input to check the result. Table 3.2 show the results for different n and κ . Fig (3.6) shows the graphs of input charge density and output potential. From the table

Table 3.2: Results for general Gaussian type potentials

n	kappa	max diff	min diff	n	kappa	max diff	min diff
2	3.00	0.439E-07	0.114E-07	2	5.00	0.239E-12	0.428E-13
3	3.00	0.231E-03	0.110E-05	3	5.00	0.384E-03	0.184E-05
4	3.00	0.777E-15	0.000E+00	4	5.00	0.888E-15	0.000E+00
5	3.00	0.272E-06	0.106E-08	5	5.00	0.454E-06	0.177E-08
6	3.00	0.206E-10	0.000E+00	6	5.00	0.180E-09	0.000E+00
7	3.00	0.148E-08	0.651E-18	7	5.00	0.827E-08	0.225E-13
8	3.00	0.326E-07	0.202E-14	8	5.00	0.117E-06	0.168E-14
9	3.00	0.347E-06	0.247E-13	9	5.00	0.813E-06	0.789E-11
10	3.00	0.198E-05	0.511E-11	10	5.00	0.374E-05	0.354E-10

(a) For $\kappa = 3$

(b) For $\kappa = 5$

we can conclude that my program yields results with sufficient accuracy for suitable choice of mesh parameters.

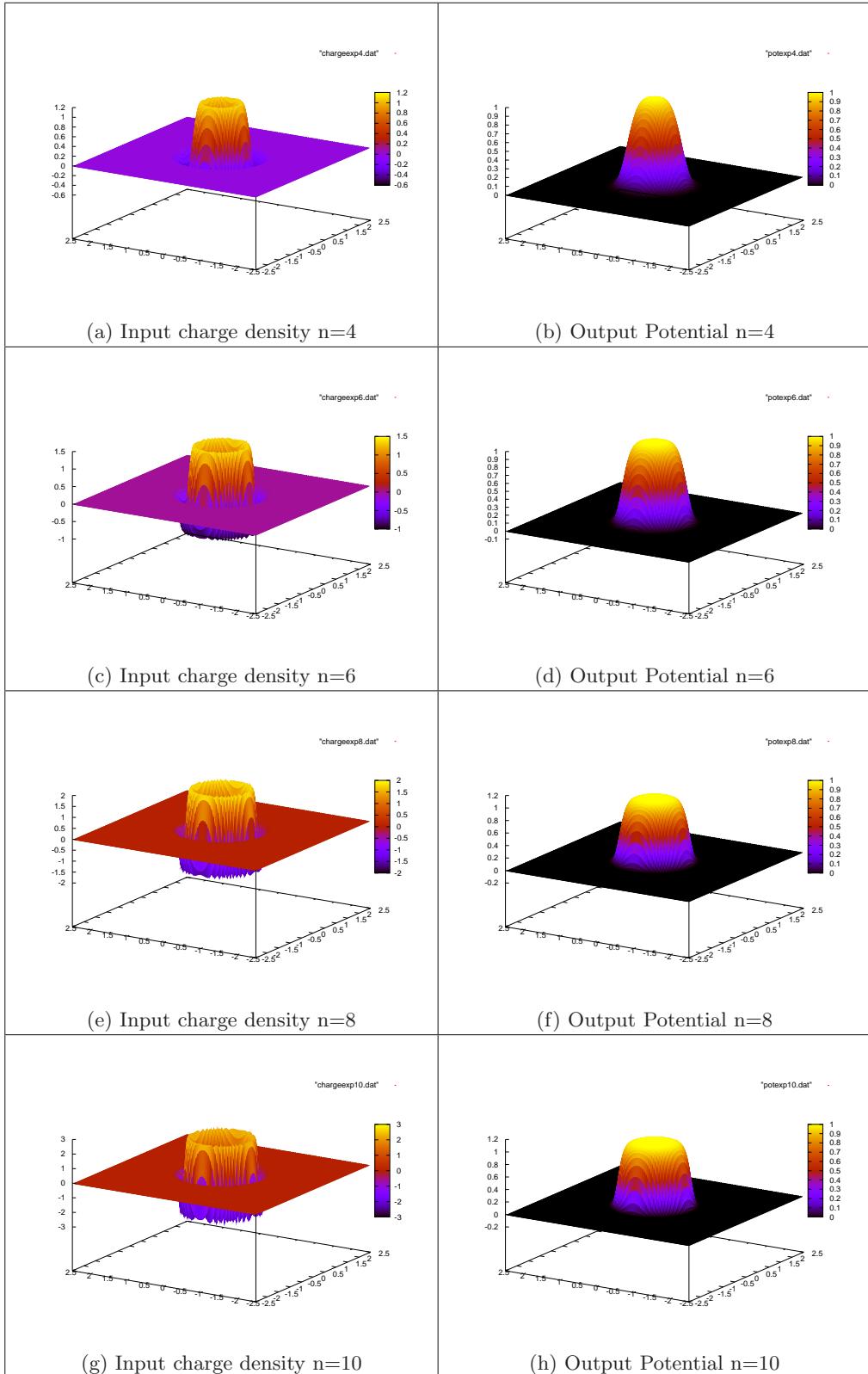


Figure 3.6: Input and Output of the program in case of 3D-potential of the form $\exp(-\kappa r^n)$ for $n = 4, 6, 8, 10$. All the graphs show z=0 section

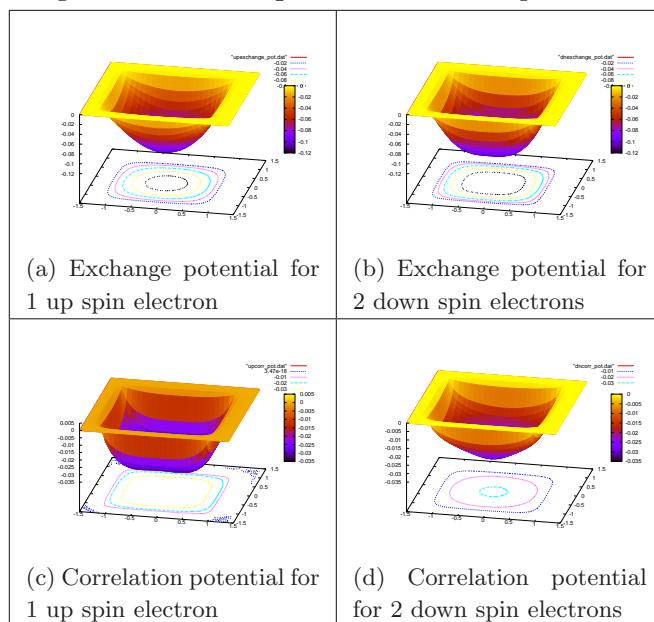
4

EXCHANGE AND CORRELATION

We need a program that can do a spin polarized exchange-correlation calculation. We will use local density approximation in our calculation. I will not go into details about the methods of calculation and use a standard library package "LIBXC" [8] containing all types of methods for calculating exchange and correlation, presently available. Libxc-1.0 alpha just released few days ago ,on 31.03.09 as a stand alone library package. Many recognized softwares like *Octopus*, *Abinit* etc , use this library.

Fig.(4.a and b) shows the exchange potential for 1 up spin and 2 down spin electrons in an infinite box. Fig .(4c and d) shows the correlation potentials.

Figure 4.1: Exchange and Correlation potentials



5

THE DENSITY FUNCTIONAL THEORY CODE

In last chapters I have described the developments of the components required to build the DFT code ,in detail .Now all the components required to build the code ,as mentioned in in Sec.1.4 , are now ready . As I mentioned in introduction, that this project is purely technical and I am not in a position to describe the backbone of the Density functional theory. Here I will not develop the algorithm for the program but simply use it.

§5.1 THE ALGORITHM

The algorithm is pretty simple and straight forward, when we have all the components are available. we just have to plug in the parts in the right order. The algorithm is like this:

1. Form the initial guess for ρ
2. Give this ρ as input to the Poisson equation solver to get Hartree potential V_H and Hartree energy E_H .
3. Use the ρ to get the exchange and correlation terms for potential and energy.
4. Add the potential contributions from the above two steps, to form the effective potential V_{eff} .
5. Solve the one electron Schrödinger equation by using this V_{eff} and get ψ .
6. Generate new charge density ρ_{new} using the formula

$$\rho_{new} = \beta \sum_i^{\text{occupied}} f_i |\psi|^2 + (1 - \beta) \rho_{old}$$

where β is the mixing parameter and usually taken to be small (< 0.2). f_i is the occupancy of the i th state . $f_i = 1$ for non degenerate states and for perfectly degenerate states $f_i = 1/\text{degeneracy}$.

7. Check if

$$|\rho_{new} - \rho_{old}| \leq \varepsilon$$

where ε is the predefined convergence limit. It is generally taken as 10^{-4} .

- If the above condition is true then the convergence achieved. calculations is stopped
- .
- If not, go to step 2.

The guarantee to convergence is given by the Density functional theory , which states:

There exists a unique charge density for the given effective potential and vice versa.

6

THE 3D CONFINED ELECTRON SYSTEM

§6.1 INTRODUCTION

Confined electron system is a model to study quantum dots. The quantum dot is defined as a system of interacting electrons in an external confinement taken as a cubic-well potential in the present case. Although quantum dots are generally modeled by considering the external confinement to be parabolic in shape, the current technology enables us to engineer other types of confinements also. It is also to be noted that generally we consider 2dimensional models for quantum dots ,that is the electron is strictly confined in a z plane. Our group already have a such model [9]. In this project I have built a 3 dimensional model of quantum dot and using that model we will study the similarity or differences with the results obtained with the 2D model.

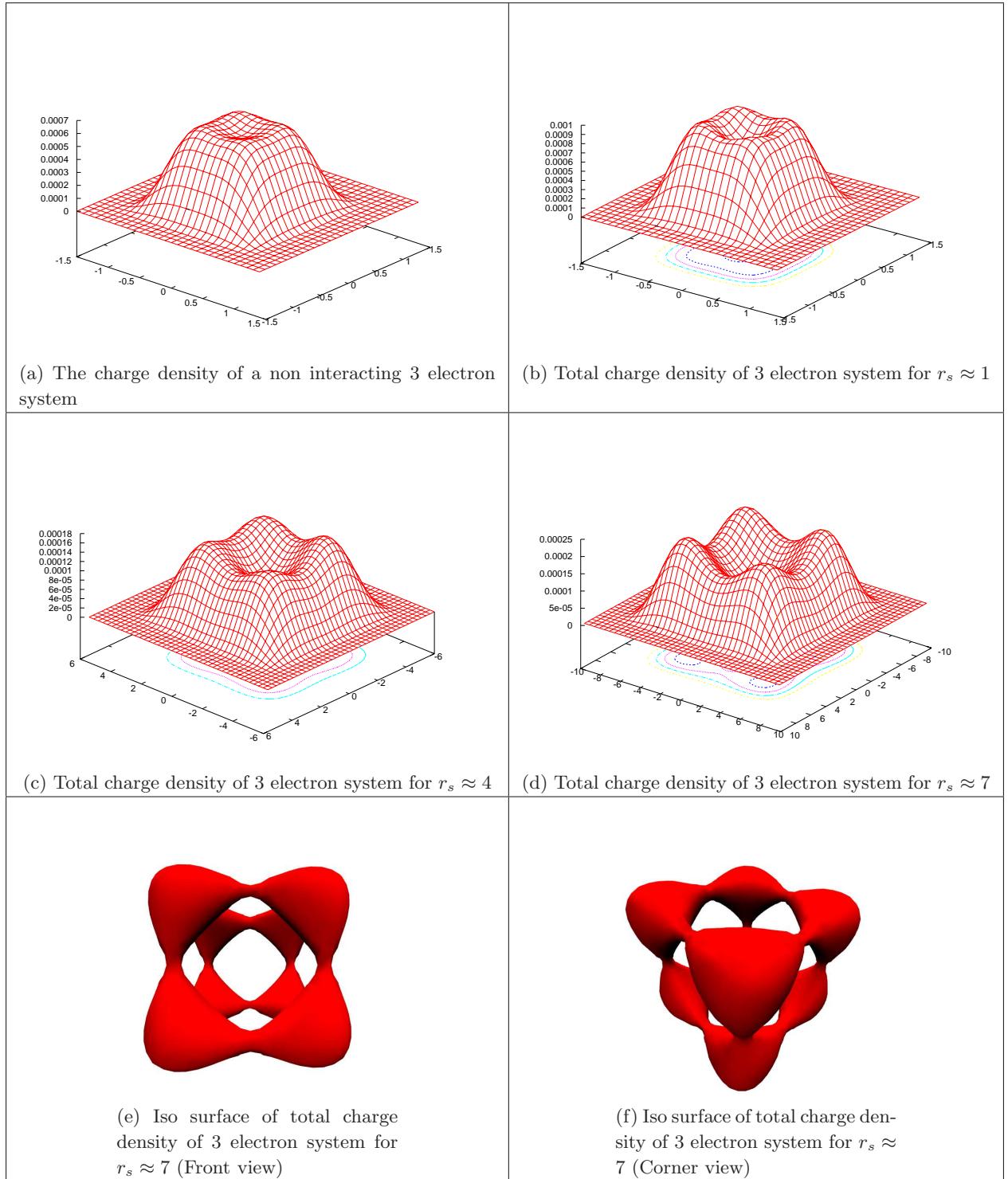
§6.2 RESULTS

We define our density parameter r_s as

$$r_s = \left(\frac{3V}{4\pi N} \right)^{1/3} \quad (6.1)$$

where V is the volume of the box and N is the total number of electrons present in the system. We have carried out experiments over various sizes of quantum dots. I vary r_s by varying one of the quantity namely the number of electrons or the size of the confinement. It is instructive bring out the difference between the non interacting charge density shown in Fig 6.2a and the SDFT charge density shown in Fig 6.1. The results are analogous to the results obtained by our group in the case of 2D confinement [9] except the fact that in 2D the electrons try to localized themselves at the four corners of the square, and here the electrons try to be localized at the eight corners of the cube. It is evident from Fig. 6.1 (a) and (b) that for low r_s that is higher density cases, the SDFT picture is in fair agreement with the non

Figure 6.1: Charge density of a 3 electron quantum dot for (a)non interacting case (b,c,d) SDFT picture for different r_s . All the graphs are plotted at $z=0$ section.(e,f)Charge density in the case of $r_s \approx 7$.The two figure shows to views taken from different side



interacting one(Fig.6.2a). However as r_s increases the SDFT charge density begins to differ considerably with the non interacting case. The figure shows that as r_s increases the charge density captures the localized behaviour .In confined Systems at low densities ,the confinement strength weakens and the coulomb interaction dominates,leading to such localization. Fig.6.2f shows the 3dimensional charge density for $r_s \approx 7$.The localization of the charge density at the corners are very much clear from the figure.

Table 6.1 shows the energy obtained for different number of electrons present in a given cubic box. In this case the r_s is varied by varying the total number of electrons.

Table 6.1: Energy of confined system for different number of electrons present

Total number of electrons	up-spin electrons	down -spin electrons	r_s	Energy
2	1	1	1.058	11.48
	2	0		14.12
3	2	1	0.924	23.52
	3	0		25.99
4	3	1	0.8402	37.39
	2	2		37.40
	4	0		39.76

From table 6.1 we can see that ground state of a 2 electron confined system is singlet ,as expected. For 3 electron system 2up 1 down is the ground state . For 4 electron system we got two nearly degenerate levels namely 3 up-1down and 2up-2down. It is hard to say which one is ground state in this case. We need to do a calculation with higher accuracy to find out the ground state. In general for a constant number of electrons the energy decreases when we increase the size of the confinement as kinetic energy and hartree energy both decreases in this case.

§6.3 FUTURE PLANS

The developmental part is over now. We have the model ready to study different cases. The main future plans are:

- We will go up to at least 20 electrons to study the effect on the charge density.
- We will increase r_s more to get highly localized behaviour of charge density.
- We will study the effect of introducing one or more attractive impurity which can capture predefined number of electrons.

7

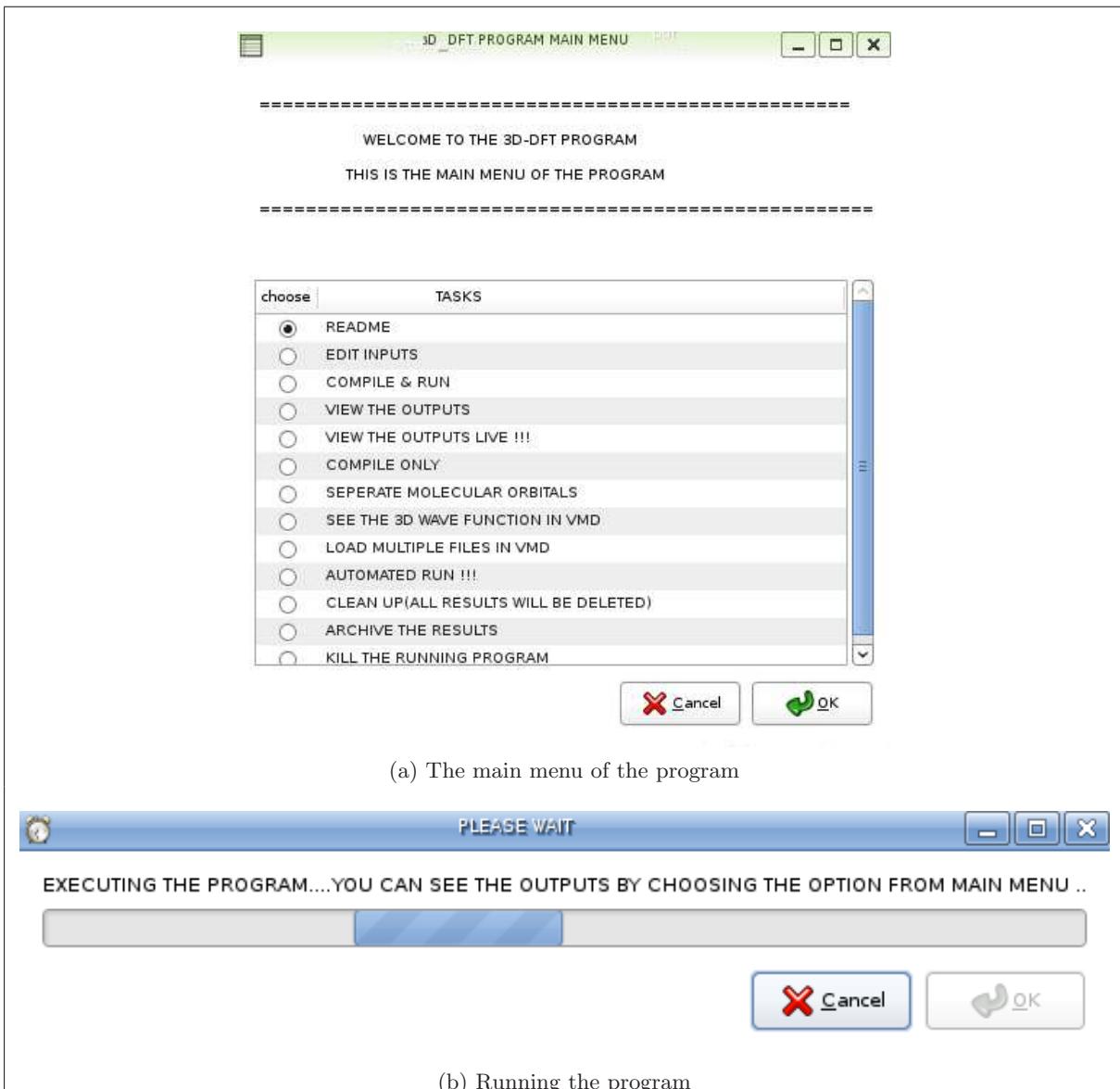
BUILDING A GRAPHICAL INTERFACE

I have also developed a graphical user interface for this program. The interface is developed for Linux by mainly using two scripting language **Zenity** and **Dialog**. The key features of this interface are :

1. All the inputs and outputs are easily accessible through the interface.
2. The program can be easily compiled and run through the interface.
3. Outputs of the program can be seen live , when the program is running.
4. Preparing the raw outputs of the program due to which visualizing analyzing is easy.
5. One can directly load the outputs to a visualizer program to see the wave functions.
6. It contains many scripts which can automatically run the program and archive the results from a predefined inputs.

Fig (7.1) shows two snapshots of the interface.

Figure 7.1: The graphical interface of the program



References

- [1] N Gidopoulos J.Kohanoff. *Density Functional Theory :Basics ,new trends and applications.* 2002.
- [2] *NUMERICAL RECIPES IN FORTRAN 77:THE ART OF SCIENTIFIC COMPUTING.* Cambridge university press.
- [3] Andreas Stathopoulos and C.F Fischer. A davidson program for finding a few selected extreme eigen-pairs of a large,sparse ,real ,symmetric matrix. *Computer Physics Communications*, 79:268–290, 1994.
- [4] J.D Jackson. *Classical Electrodynamics.*
- [5] J.W & J.W Tukey Cooley. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19:297–301, 1965.
- [6] Glenn J. Martyna and Mark E. Tuckerman. A reciprocal space based method for treating long range interactions in ab initio and force-field-based calculations in clusters. *The Journal of Chemical Physics*, 110(6):2810–2821, 1999.
- [7] Matteo Frigo and Steven G. Johnson. Fastest fourier transform in west (fftw 3.2).
- [8] Miguel A. L. Marques. Libxc-1.0 alpha, 31-03-2009.
- [9] Bhalchandra Pujari *et all.* Electronic structure of many-electron square-well quantum dots with and without an attractive impurity: Spin-density-functional theory, 2007.