Real Space Density Function Theory

User Manual

Amy Coddington
Adam Jundt
Nicholas Voshell
Last updated 7/31/06

Table of Contents

What is RSDFT?

RSDF T, the Real Space Density Functional Theory, is a program written in MATLAB that calculates properties of the electronic structure of molecules. It is based on previous work done by Dr. Jim Chelikowsky's Material Science Group at the University of Texas, Austin in collaboration with Dr. Yousef Saad and his colleagues at the University of Minnesota. These scientists created a program entitled Pseudopotential Algorithms for Real Space Eigenvalue Calculations (PARSEC) on which RSDFT is based. The algorithms used in RSDFT are much simpler than those used in PARSEC, with the hope that RSDFT can be used for teaching and prototyping new algorithms.

This program was created for use with Mat lab 7.0 and above, although use with MATLAB 6.5 is also possible. It is also required that the user have Python version 2.4.1 or higher, as it is used as a parser. RSDFT was written for Linux machines, and while it is possible to run it on other programs, it is highly recommended that the user use a Linux machine. This program creates output that can be visualized using PVOX (Parsec Visualization Toolbox).

Overview of how RSDFT works

To calculate the properties of the molecule, RSDFT uses the Kohn-Sham formulation of the Schroedinger Equation, which is as follows:

$$Hamiltonian = \frac{-1}{2}\nabla^2 + V_{pseudopotential} + V_{XC} + V_{HF}$$

RSDFT first calculates the Laplacian and then calculates the ionic potential of the molecule. The Laplacian is calculated based on finite difference methods from Bengt Fornberg; the program represents the Laplacian as a linear term in nearby wave function values. The ionic potential is approximated by a pseudopotential proposed by Robert Shaw in his paper “Optimum Form of a Modified Heine-Abarenkov Model Potential for the Theory of Simple-Metals (1968).” The program then goes through a self-consistent loop which calculates the Ceperly-Alder exchange correlation potential, Hartree potential, and determines the eigenvalues and eigenvectors. If the results are outside of a certain tolerance, it repeats the loop until the results are within the tolerance.

How to run RSDFT

RSDFT runs from inside the MATLAB terminal. First, open up MATLAB and move to the directory where RSDFT is located. Once inside the RSDFT directory, simply type "main" at the command prompt. You will then see a series of outputs on the command line which can be interpreted as follows:

Current Plot Held	a plot window has been initialized for use later in the program
n=( )	the total number of grid points in the system
calling ilu0	if a preconditioner is being used, this will display to alert the user

Next, the self consistent loop begins and you will see a printout of each iteration. It will show you which diagonalization method is used (Lanczos or Chebyshev) and the error on that iteration. Once the algorithm has converged and the error is less than the tolerance, the program finishes and displays two plots. One of these (Figure 1) shows the location of the atoms and the other plot in Figure 2 displays the charge density. The program creates two output files: **rsdft.out** and **wfn.dat**. These will be discussed in the "Output Files" section.

User defined input files

There are two files, **include.m** and **main.m**, in the RSDFT package that allow the user to adjust the conditions of the calculations.

Include.m is the simpler of the two. The first line asks whether the user wants to use a preconditioner. Using a preconditioner slows the startup time quite considerably, but makes each iteration run faster. A 1 here means with the preconditioner and a 0 means without. The next line asks for the polynomial degree for the Chebyshev diagonalization method. The third input line concerns which diagonalization method to use. A 0 here, as it explains in the file, means the program uses Lanczos for the first step and Chebyshev thereafter. A 1 means Lanczos for every iteration. The final input line in this file concerns when, if at all, to do mixing.

Main.m is a little more complicated. The first 30 lines or so contain code pertaining to initialization and writing to the output files. On line 34 the user can designate a grid size. When reading in the parsec input file, the program chooses the grid size as whatever was in the input file. If the user wants to change the grid size from within RSDFT, this line allows users to do just that. It is recommended to choose a grid size that is some multiple of 'grid_spacing,' such as grid_spacing*.4. The next adjustable value is in line 48, where the user can specify the maximum number of iterations in the self consistent loop. Right below that, in line 51, the user can choose what tolerance to use for the self consistent loop. If the answer is within this tolerance, the self consistent loop will stop and a result will be produced. Otherwise, the program will keep looping.

Information on Specific Files

Any files in this program may be modified by the user for their benefit. This next section gives a rundown on what various files do within the program to aid with the understanding of the program.

ch_filter.m
This function, along with **chefsi1.m**, is part of the Chebyshev calculation of the wave function. It applies a Chebyshev filter to a vector x.

chefsi1.m
This function uses a Rayleigh-Ritz projection to compute the eigenvectors and eigenvalues, based on the filtered vector that is computed in **ch_filter.m**

elements.csv
This file, while not a MATLAB executable file, contains necessary information on atoms. This file is used to compute the ionic potential. The columns are as follows:
Atom Name, Atomic Number, largest l (orbital number), number of valence electrons, A (the electromagnetic potential of the ion core) at l=0, A at l=1, A at l=2, radius at l=0, radius at l=1, radius at l=2. The last six of these columns were gleaned from Table 4 in Osman Ese and J. A. Reissland's 1973 paper "Optimized model potential parameters in metals."

exc_nspn.m
This function computes the Ceperly-Alder exchange correlation potential based on the physical location of the atom and the charge density.

fd3d.m
This function computes a discretization of the Laplacian matrix using finite difference methods defined by Bengt Fornberg.

include.m
This file contains user input; for more information see above in "User Defined Input Files."

lan.m
This function computes the eigenvalues and eigenvectors based on Lanczos' methods.

main.m
This is the actual file called when running RSDFT. It initializes some variables

and then calls **rsdft.m**. It then creates plots of the molecule and of the wavefunction. More information on this file can be found in "User Defined Input Files."

parse_inp.py
This is the parser that reads in the parsec input data and transforms it into information that MATLAB can read. This file is written in Python.

pcg.m
This function computes the Hartree potential.

potential.m
This file computes the ionic potential of each atom in the molecule, based on methods described by Robert Shaw. The equation used is as follows:

$$w(r,\theta,\phi)=\frac{-Z}{r}-\sum_{0}^{l_0}[\Theta(R_l-r)*(A_l-\frac{Z}{r})*P_l(r,\theta,\phi)] \qquad \Theta(R-r)=\begin{Bmatrix}1 & if\ r<R\\ 0 & if\ r>R\end{Bmatrix}$$

rsdft.m
This is the function that controls most of the computations. It is called by **main.m** and begins with calculating the ionic potential (**potential.m**). It then enters a self consistent loop that computes the eigenvalues (using either **lan.m** or **chefsi1.m**), the charge density, the exchange correlation potential (**exc_nspn.m**) and the Hartree potential (**pcg.m**). It repeats this until the result is within a certain tolerance.

Output files

There are two output files generated by RSDFT. One, **rsdft.out**, contains information about the eigenvalues, the charge density and about the molecule. The second, **wfn.dat**, is a binary file describing the wave function. These files can be used as input for the program PVOX, a visualization tool developed by scientists in Dr. Yousef Saad's group at the University of Minnesota. To do this, simply move to the PVOX directory, conveniently located inside of the RSDFT directory. Once inside of there, simply enter "pvox" into the MATLAB command prompt and a screen will come up. There is a open file icon in the upper left hand corner of the screen, and clicking on it brings up a window that allows the user to select a ".out" file and a ".dat" file. Once the user locates the files, a click on the OK button allows the user to look at various visualizations of the data. For more information please refer to the PVOX manual located inside of the DOCUMENTATION folder.