

# Radial Kohn-Sham Equation Solver for Hydrogen-like Atoms

Wang Jianghai @Nanyang Technological University

2025-10-24

## Contents

<b>1</b>	<b>Density Functional Theory</b>	<b>2</b>
1.1	Adiabatic and Single-Electron Approximations . . . . .	2
1.2	Thomas–Fermi–Dirac Approximation . . . . .	3
1.3	Hohenberg–Kohn Theorems . . . . .	3
1.4	Radial Kohn-Sham Equations . . . . .	3
1.5	Exchange–Correlation Functionals . . . . .	5
1.5.1	Local Density Approximation (LDA) . . . . .	5
1.5.2	Generalized Gradient Approximation (GGA) . . . . .	5
1.5.3	Hybrid Functionals . . . . .	5
1.6	Software for DFT . . . . .	6
<b>2</b>	<b>Implementation</b>	<b>6</b>
2.1	Discretization . . . . .	6
2.2	Numerov Method . . . . .	7
2.3	Thomas Algorithm . . . . .	7
2.4	Self-Consistent Field . . . . .	7
2.5	Exchange–Correlation Functional . . . . .	9
2.6	Validation . . . . .	9
2.7	Code Details . . . . .	10
2.7.1	RadialDFT Class . . . . .	10
2.7.2	Boundary Conditions . . . . .	10
2.7.3	Energy Calculation . . . . .	10
<b>3</b>	<b>Usage</b>	<b>11</b>
<b>4</b>	<b>Discussion</b>	<b>11</b>
4.1	Optimization of DFT Parameters . . . . .	11
4.1.1	Optimization of Integration Parameters . . . . .	11
4.1.2	Mixing Parameter for SCF Convergence . . . . .	12
4.2	Known Numerical Issues . . . . .	13
4.3	Theoretical Extensions . . . . .	13
<b>5</b>	<b>References</b>	<b>14</b>

## Abstract

This project implements a radial Kohn–Sham (KS) self-consistent field (SCF) solver for spherically symmetric (H-like) systems, focusing on the 1s orbital. The solver uses a fixed radial mesh, constructs the Kohn–Sham potential as

$$V_{\text{KS}}(r) = V_{\text{nuc}}(r) + V_{\text{ee}}(r) + V_{\text{xc}}(r),$$

solves the radial KS differential equation with Numerov/Thomas tridiagonal propagation, computes energy components, and iterates until the KS eigenvalue ( $\varepsilon$ ) converges. Exchange-correlation is handled in a local density approximation (LDA) with an analytic parameterization. The implementation is pedagogical and suitable for learning numerical Density Functional Theory (DFT) methods and numerical ordinary differential equation (ODE) solvers (Numerov and Thomas algorithms) in spherical coordinates.

# 1 Density Functional Theory

Following the establishment of quantum mechanics and statistical physics, theoretical studies of condensed matter gradually emerged. It became clear that the macroscopic properties of solids are intrinsically linked to their electronic behavior. In principle, if one could exactly solve the electronic wave functions in a solid, all physical quantities of the system could be obtained through the corresponding quantum-mechanical operators. However, a typical solid contains on the order of  $10^{23}$  particles, each with three spatial degrees of freedom, making the many-body Hamiltonian intractably complex and analytically unsolvable. Hence, appropriate approximations are essential.

Density Functional Theory is a quantum mechanical modeling method used in physics, chemistry, and materials science to investigate the electronic structure of many-body systems. The fundamental principle of DFT is that the properties of a many-electron system can be determined using functionals of the spatially-dependent electron density rather than the many-body wavefunction, reducing the problem from a  $3N$ -dimensional space to only three dimensions. This fundamental reformulation significantly simplifies the calculation of electronic structure while retaining quantum-mechanical accuracy.

## 1.1 Adiabatic and Single-Electron Approximations

Because the mass of a nucleus is roughly 1800 times that of an electron, electrons respond much more rapidly to environmental changes than nuclei do. Thus, the physical problem can be decoupled into two parts: when describing electronic motion, the nuclei can be treated as fixed sources of potential, whereas when describing nuclear motion, the electronic distribution can be considered static. Solving the electronic problem under a fixed nuclear potential yields the lowest-energy configuration, i.e., the electronic ground state.

This separation of electronic and nuclear motion forms the basis of the Born–Oppenheimer approximation, or the adiabatic approximation. It reduces the full many-body problem to a purely electronic one, with the ground-state energy  $E(R_1, R_2, \dots, R_M)$  depending on the nuclear positions  $R_i$ , thereby defining the adiabatic potential energy surface of the system.

Even after separating nuclear and electronic motion, the electron–electron interaction remains a many-body problem involving  $N$  particles. To address this, Hartree proposed averaging the instantaneous interactions between electrons, treating each electron as moving independently in an averaged potential—the Hartree approximation, or single-electron approximation. Although this simplification transformed the many-electron problem into a set of single-electron equations, it neglected electron exchange and correlation effects. Fock later incorporated the exchange interaction, leading to the Hartree–Fock approximation. While Hartree–Fock theory accurately accounts for exchange, it still neglects correlation effects, and its accuracy deteriorates with increasing electron number.

## 1.2 Thomas–Fermi–Dirac Approximation

In 1927, Thomas and Fermi proposed the homogeneous electron gas model, which neglects electron–electron interactions and expresses the total energy as a functional of electron density. In 1930, Dirac introduced a local approximation for exchange interactions, extending the model to account for exchange energy within the density functional framework. The combined formulation is known as the Thomas–Fermi–Dirac (TFD) approximation.

## 1.3 Hohenberg–Kohn Theorems

The theoretical foundation of modern DFT lies in two theorems proved by Hohenberg and Kohn.

1. The ground-state properties of a many-electron system are uniquely determined by the electron density  $n(\mathbf{r})$ .
2. The electron density that minimizes the energy of the overall functional is the ground-state electron density.

The energy functional can be expressed as:

$$E[\psi_i] = -\frac{\hbar^2}{m} \int \psi_i^* \nabla^2 \psi_i d^3r + \int V(\mathbf{r}) n(\mathbf{r}) d^3r + \frac{e^2}{2} \iint \frac{n(\mathbf{r}) n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3r d^3r' + E_{\text{ion}} + E_{\text{XC}}[\psi_i]$$

which contains the kinetic energy of electrons, electron–nucleus Coulomb interaction, electron–electron Coulomb repulsion, and nucleus–nucleus repulsion. The remaining term,  $E_{\text{XC}}[\psi_i]$ , is the exchange–correlation energy functional, which accounts for the complex many-body effects of exchange and correlation among electrons.

## 1.4 Radial Kohn–Sham Equations

In practice, DFT calculations are performed using the Kohn–Sham approach, which replaces the original many-body problem by an auxiliary independent-particle problem.

Starting from the three-dimensional Schrödinger equation in spherical coordinates:

$$\Psi(r, \theta, \varphi) = \Psi(r) Y(\theta, \varphi)$$

where  $Y(\theta, \varphi)$  is spherical harmonics. It can be separated on two independent equations for  $r$  and  $\theta, \varphi$ . The Schrodinger equation in spherical coordinates for part depending on  $r$  only, the **radial Schrodinger equation** is

$$\left[ -\frac{1}{2r} \frac{d^2}{dr^2} r + \frac{l(l+1)}{2r^2} + V_{\text{nuc}}(r) \right] \Psi_{nl}(r) = E_{nl} \Psi_{nl}(r)$$

where  $V^{\text{NUC}}(r) = -\frac{Z}{r}$ . For the ground state (1s orbital) with  $n = 1$ ,  $l = 0$ , the centrifugal term vanishes, resulting in:

$$\left[ -\frac{1}{2} \frac{d^2}{dr^2} + V_{\text{nuc}}(r) \right] \Psi_{1s}(r) = E \Psi_{1s}(r)$$

For atoms with spherical symmetry, we can separate the wavefunction into radial and angular parts:

$$\psi_{nlm}(\mathbf{r}) = \frac{P_{nl}(r)}{r} Y_{lm}(\theta, \phi)$$

where  $Y_{lm}$  are spherical harmonics and  $P_{nl}(r)$  is the radial wavefunction.

### Radial vs. Full Wavefunction

The distinction between the full wavefunction  $\Psi_{nl}(r)$  and the radial wavefunction  $P_{nl}(r)$  is critical. The radial wavefunction is defined as  $P_{nl}(r) = r\Psi_{nl}(r)$ , where the factor of  $r$  arises from the Jacobian in spherical coordinates. The normalization condition for the radial function is  $\sqrt{4\pi \int_0^\infty P_{nl}(r)^2 dr} = 1$ , where the  $4\pi$  factor comes from integrating over the spherical harmonics. This can be verified by calculating the norm of the analytical hydrogen solution:  $4\pi \int_0^\infty \left( \frac{r}{\sqrt{\pi}} Z^{3/2} e^{-Zr} \right)^2 dr = 1$ .

Then the radial Kohn-Sham equation can be written as:

$$\left[ -\frac{1}{2} \frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} + V_{\text{KS}}(r) \right] P_{nl}(r) = \varepsilon_{nl} P_{nl}(r)$$

where  $P_{nl}(r) = r\Psi_{nl}(r)$  is the radial wave function,  $V_{\text{KS}}(\mathbf{r})$  is the Kohn-Sham potential, given by:

$$V_{\text{KS}}(\mathbf{r}) = V_{\text{nuc}}(r) + V_{ee}(r) + V_{xc}(r)$$

comprising the nuclear attraction potential

$$V_{\text{nuc}}(r) = -\frac{Z}{r}$$

the electron-electron Coulomb repulsion potential

$$V^{ee}(r) = 4\pi \int \frac{\rho(r')}{|r - r'|} r'^2 dr'$$

and the exchange-correlation potential  $V_{xc}$ .

For the ground state (1s orbital, with  $l = 0$ ), the radial Kohn-Sham equation becomes:

$$\left[ -\frac{1}{2} \frac{d^2}{dr^2} + V_{\text{KS}}(r) \right] P_{10}(r) = \varepsilon_{10} P_{10}(r)$$

This is the equation we solve numerically in this implementation.

## 1.5 Exchange–Correlation Functionals

The key unknown in DFT is the exchange–correlation functional  $E_{XC}[\rho]$ . While the Hohenberg–Kohn theorems guarantee its existence, its explicit form cannot be derived, and therefore various approximations are employed in practical calculations. The choice of an appropriate XC functional is crucial, as it directly determines both the accuracy and computational efficiency of DFT.

### 1.5.1 Local Density Approximation (LDA)

LDA assumes that the exchange–correlation energy at each point depends only on the local electron density, as in a uniform electron gas:

$$V_{XC}(\mathbf{r}) = V_{XC}^{\text{electron gas}}[n(\mathbf{r})]$$

Although this approximation neglects spatial variations in the density, it provides a tractable and often surprisingly effective method, especially for systems with slowly varying densities. However, it becomes inaccurate for systems with strong density gradients, such as covalently bonded materials.

### 1.5.2 Generalized Gradient Approximation (GGA)

GGA extends LDA by incorporating not only the local electron density but also its spatial gradient, making  $E_{XC}$  a functional of both  $n(\mathbf{r})$  and  $\nabla n(\mathbf{r})$ .

$$E_{XC}^{\text{GGA}} = E_{XC}[n(\mathbf{r}), \nabla n(\mathbf{r})]$$

This inclusion of local inhomogeneity typically improves accuracy over LDA. Common GGA functionals include **Perdew–Wang (PW91)** and **Perdew–Burke–Ernzerhof (PBE)**, both widely adopted in solid-state calculations.

### 1.5.3 Hybrid Functionals

Hybrid functionals combine the orbital-dependent Hartree–Fock exchange with density-based functionals, controlled by a mixing parameter  $\lambda$ . For instance, when  $\lambda = 0$ , the exchange is purely Hartree–Fock; when  $\lambda = 1$ , it is entirely LDA or GGA. The **Heyd–Scuseria–Ernzerhof (HSE)** hybrid functional, particularly **HSE06**, mixes 25% of short-range Hartree–Fock exchange with 75% of short-range PBE exchange, while using PBE for correlation and long-range exchange:

$$E_{XC}^{\text{HSE06}} = \frac{1}{4}E_X^{\text{SR,HF}}(\mu) + \frac{3}{4}E_X^{\text{SR,PBE}}(\mu) + E_X^{\text{LR,PBE}}(\mu) + E_C^{\text{PBE}}$$

Hybrid functionals often provide improved band-gap predictions and better descriptions of strongly correlated systems, though at significantly higher computational cost and with sensitivity to empirical parameters.

### Self-Interaction Error

A fundamental limitation inherent to approximate DFT functionals is the self-interaction error (SIE). In principle, the exact exchange–correlation potential should cancel the spurious Coulomb interaction of an electron with itself. However, approximate functionals do not fully achieve this cancellation, leading to a residual and unphysical electron self-interaction.

As a result, DFT often exhibits several systematic deficiencies. First, it tends to overstabilize delocalized electronic states, leading to an underestimation of band gaps in semiconductors and insulators; Second, it underpredicts chemical reaction barriers, as transition states with stretched bonds are artificially stabilized; Finally, the highest occupied Kohn–Sham (KS) eigenvalue no longer corresponds to the true ionization potential, violating Koopmans’ theorem.

To mitigate SIE, various self-interaction correction (SIC) schemes have been developed, aiming to remove the spurious self-Coulomb energy from each occupied orbital. Alternatively, more sophisticated many-body approaches, such as the GW approximation or Møller–Plesset perturbation theory (MPn), can be employed to provide more accurate electronic structures by explicitly accounting for electron correlation beyond standard DFT.

## 1.6 Software for DFT

Several software packages implement DFT calculations (see Wikipedia), including:

- **VASP**: A software for atomic scale materials modelling from first principles.
- **Quantum ESPRESSO**: An integrated suite of Open-Source computer codes for electronic-structure calculations and materials modeling at the nanoscale.
- **ABINIT**: A software suite to calculate the optical, mechanical, vibrational, and other observable properties of materials.
- **Gaussian**: A powerful and versatile software for modeling molecules and chemical problems.
- **ABACUS**: A software package for large-scale electronic structure calculations from first principles.

## 2 Implementation

### 2.1 Discretization

The radial equation is discretized using a uniform grid:

$$r_i = r_0 + i \cdot h, \quad i = 0, 1, \dots, N - 1$$

where  $r_0$  is the starting point (close to the origin),  $h$  is the step size, and  $N$  is the number of grid points.

## 2.2 Numerov Method

The Numerov method can be used to solve differential equations of the kind

$$\frac{\partial^2}{\partial x^2} y(x) + f(x) y(x) = F(x)$$

By using Taylor expansion for  $y(x_{i+1})$  and  $y(x_{i-1})$  around  $x_i$  and combining them, we obtain the **Numerov formula**:

$$\left(1 + \frac{h^2}{12} f_{i+1}\right) y_{i+1} - \left(2 - \frac{5h^2}{6} f_i\right) y_i + \left(1 + \frac{h^2}{12} f_{i-1}\right) y_{i-1} = \frac{h^2}{12} (F_{i+1} + 10F_i + F_{i-1}) + \mathcal{O}(h^6)$$

For the Kohn-Sham equation, we rewrite it in the form:

$$\frac{\partial^2}{\partial r^2} P(r) + \underbrace{2(\varepsilon - V^{KS}(r))}_{f(r)} P(r) = \underbrace{0}_{F(r)}.$$

## 2.3 Thomas Algorithm

The Thomas algorithm (also known as the tridiagonal matrix algorithm) is used to efficiently solve the tridiagonal system arising from the Numerov method:

1. **Forward elimination** phase creates an upper bidiagonal system

$$\alpha_{i+1} = -\frac{B_i}{A_i \alpha_i + C_i}$$

$$\beta_{i+1} = \frac{Z_i - A_i \beta_i}{A_i \alpha_i + C_i}$$

2. **Backward substitution** phase solves for the unknowns

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}$$

This approach is computationally efficient with  $\mathcal{O}(N)$  complexity.

## 2.4 Self-Consistent Field

Self-consistent field (SCF) method is used to solve KS equation numerically. We can start from some initial approximation to the solution. Let us suppose, that the set of  $\psi_i^0$  and  $\varepsilon_i^0$  is good initial guess to the solution. Using these values we can calculate KS potential and then solve KS equation with known potential. We will get new solution,  $\psi_i^1$  and  $\varepsilon_i^1$ , from which we can recalculate new KS potential,  $V^1$ . Since using just new calculated potential  $V^1$  can lead to divergence of iterations, we can set some mixing parameter  $\alpha$  and calculate the potential for next iteration as following

$$V^{\text{New}} = \alpha V^1 + (1 - \alpha) V^0$$

where  $0 < \alpha \leq 1$ . In other words, we add only some part of the new calculated potential and keep part of the previous potential in order not to change the new solution too much.

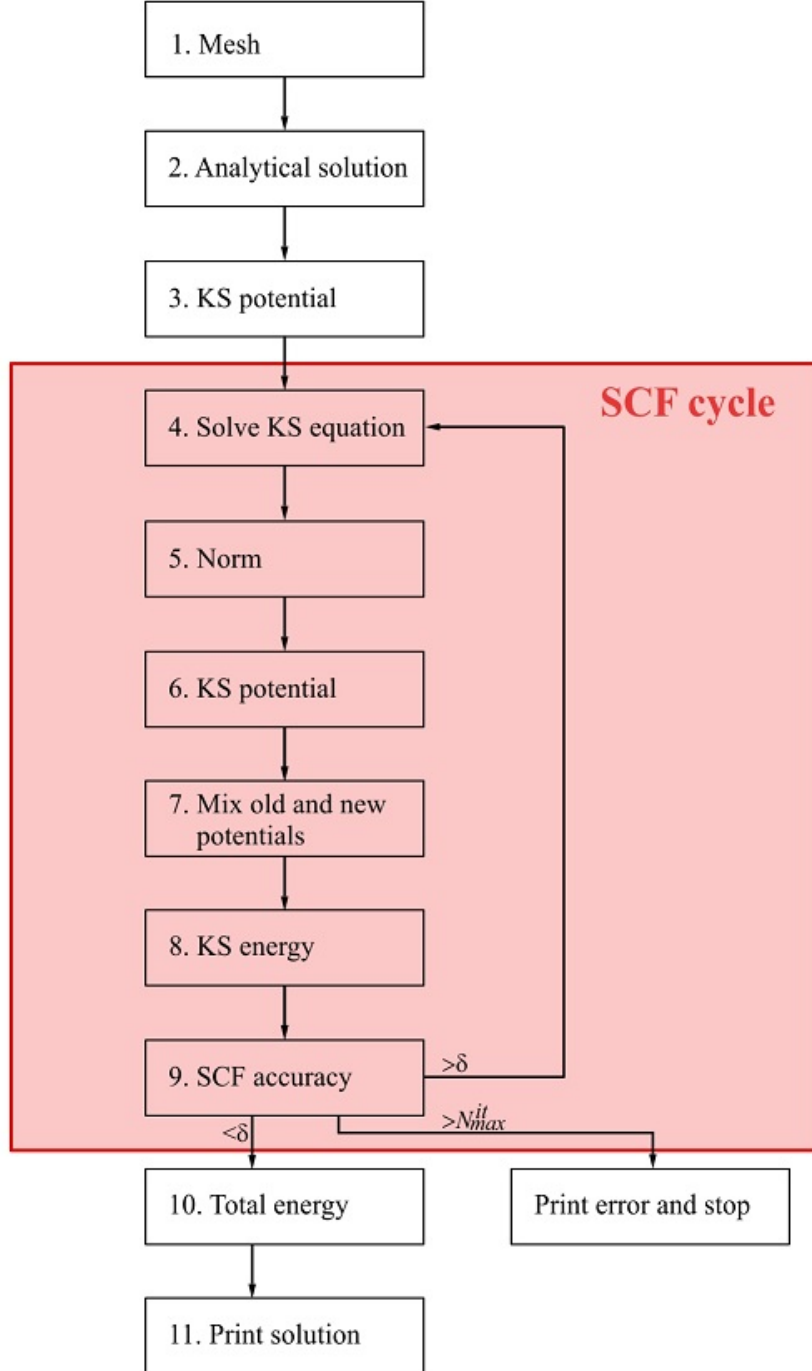


Figure 1: SCF Flowchart

The SCF procedure implements the following steps:

1. Initialize with an analytical solution for a hydrogen-like atom
2. Calculate the electron density from the wavefunction
3. Construct the Kohn-Sham potential
4. Solve the Kohn-Sham equation to obtain a new wavefunction
5. Mix the old and new potentials to improve convergence stability



6. Check for energy convergence
7. Repeat until convergence

## 2.5 Exchange-Correlation Functional

We implement the Local Density Approximation (LDA) for the exchange-correlation functional:

- The **correlation potential** uses the Perdew-Zunger parametrization of the Ceperley-Alder quantum Monte Carlo results:

$$V^C[\rho] = \begin{cases} A \ln r_s + \left(B - \frac{1}{3}A\right) + \frac{2}{3}Cr_s \ln r_s + \frac{1}{3}(2D - C)r_s, & \text{if } r_s < 1; \\ \gamma \frac{\left(1 + \frac{7}{6}\beta_1\sqrt{r_s} + \frac{4}{3}\beta_2r_s\right)}{\left(1 + \beta_1\sqrt{r_s} + \beta_2r_s\right)^2}, & \text{if } r_s \geq 1; \end{cases}$$

- For high densities ( $r_s < 1$ ):

$$V^C[\rho] = A \ln r_s + \left(B - \frac{1}{3}A\right) + \frac{2}{3}Cr_s \ln r_s + \frac{1}{3}(2D - C)r_s$$

- For low densities ( $r_s \geq 1$ ):

$$V^C[\rho] = \gamma \frac{\left(1 + \frac{7}{6}\beta_1\sqrt{r_s} + \frac{4}{3}\beta_2r_s\right)}{\left(1 + \beta_1\sqrt{r_s} + \beta_2r_s\right)^2}$$

where  $r_s = \left(\frac{3}{4\pi\rho}\right)^{\frac{1}{3}}$  is the Wigner-Seitz radius.

## 2.6 Validation

The implementation is validated by comparing the numerical solution with the analytical solution for hydrogen-like atoms. For a hydrogen-like atom with nuclear charge  $Z$ , the analytical 1s wavefunction is:

$$P_{\text{analytical}}(r) = r\Psi_{\text{analytical}}(r) = \frac{Z^{\frac{3}{2}}}{\sqrt{\pi}}re^{-Zr}$$

with energy  $E = -\frac{Z^2}{2}$  (in atomic units).

Validation checklist (minimal, reproducible tests):

- One-electron limit: with  $V_{ee} = 0$  and  $V_{xc} = 0$  the solver with  $V_{nuc}$  only must reproduce the analytic eigenvalue  $\varepsilon = -Z^2/2$  and the analytic radial function  $P_{1s}(r)$  after normalization with  $\text{Norm} = \sqrt{4\pi \int P^2 dr}$ . This test confirms correct treatment of  $P(r)$ , boundary conditions, Numerov integration, and normalization factors.
- Norm test: assert  $|4\pi \int P^2 dr - 1| < \text{tol}$  after normalization.
- Energy integral consistency: verify dimensional consistency and relative magnitudes of  $E_{ks}$ ,  $E_{ee}$ ,  $E_{xc}$  and that  $E_{xc}$  computed from the density integral is consistent (within expected differences) with  $-\int \rho V_{xc} d^3r$ .

- SCF bookkeeping: during SCF use  $\Delta\varepsilon = |\varepsilon_{new} - \varepsilon_{old}|$  as convergence metric; compute total energy  $E_{tot}$  only after the density has converged.

### Spherical Integration

When computing energy integrals in spherical coordinates, the transformation from 3D integration requires careful attention to the Jacobian factor. The integral  $\int V(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}$  becomes:

$$\int_0^{2\pi} d\phi \int_0^\pi \sin(\theta)d\theta \int_0^\infty r^2 dr V(r, \theta, \phi)\rho(r, \theta, \phi)$$

For spherically symmetric systems,  $V$  and  $\rho$  depend only on  $r$ , and integration over  $\theta$  and  $\phi$  yields  $4\pi$ . The crucial point is that the  $r^2$  factor from the Jacobian must be included in all radial energy integrals. Omitting this factor leads to incorrect values for  $E_{ee}$ ,  $E_{xc}$ , and related energy components.

## 2.7 Code Details

### 2.7.1 RadialDFT Class

The `RadialDFT` class in `DFT.py` encapsulates the numerical solution of the radial Kohn-Sham equation. Key methods include:

- `initialize()`: Sets up the initial wavefunction guess using the analytical solution
- `normalize()`: Normalizes the wavefunction using the trapezoidal rule
- `get_v_nuc()`, `get_v_ee()`, `get_v_xc()`: Calculate the potential components
- `get_v_ks()`: Combines the potential components into the Kohn-Sham potential
- `solve_ks()`: Solves the Kohn-Sham equation using the Numerov method
- `ks_energy()`, `e_xc()`, `total_energy()`: Calculate energy components
- `scf_loop()`: Implements the self-consistent field iteration

### 2.7.2 Boundary Conditions

The radial wavefunction satisfies:

- $P(r \rightarrow 0) \propto r^{l+1}$  (for 1s orbital with  $l = 0$ , so  $P(0) = 0$ )
- $P(r \rightarrow \infty) = 0$

### 2.7.3 Energy Calculation

The total energy is calculated as:

$$E_{tot} = E_{ks} + E_{ee} + E_{xc} - \int n(r)V_{xc}(r)dr$$

where:

- $E_{ks}$  is the Kohn-Sham eigenvalue
- $E_{ee}$  is the electron-electron repulsion energy
- $E_{xc}$  is the exchange-correlation energy
- The last term corrects for double-counting in the Kohn-Sham potential

### 3 Usage

The parameters for the calculation can be set in `main.py`:

```

1 Z = 6          # Nuclear charge for hydrogen-like atom
2 r0 = 1e-5      # Minimum radius (a.u.)
3 rf = 20.0      # Maximum radius (a.u.)
4 N = 10001      # Number of grid points
5
6 alpha = 0.1    # Mixing parameter for SCF
7 prec = 1e-5    # Convergence tolerance
8 max_iter = 1000 # Maximum number of SCF iterations

```

Then simply execute the calculation:

```

1 python main.py

```

The code generates several output files:

1. `wavefunction.dat`: Contains the radial wavefunction  $P(r)$  at each iteration
2. `potential.dat`: Contains the different potential components
3. `energy.dat`: Contains the energy components at each iteration
4. `wavefunction_comparison.png`: Plot comparing numerical and analytical wavefunctions
5. `potential_comparison.png`: Plot of the different potential components

## 4 Discussion

### 4.1 Optimization of DFT Parameters

The accuracy and stability of the SCF procedure in DFT also depend on the choice of numerical integration parameters and the potential mixing strategy. In this section, we systematically discuss the optimization of these parameters and supported by our numerical tests.

#### 4.1.1 Optimization of Integration Parameters

In radial DFT calculations, the total energy and eigenvalues are obtained through numerical integration over a finite radial domain  $[0, r_f]$  with  $N$  uniformly spaced grid points. In principle, increasing both  $r_f$  and  $N$  improves accuracy, as the tail of the wave function decays exponentially and should be captured sufficiently before truncation. However,

excessively large values may lead to numerical overflow, instability, or prohibitively high computational cost.

According to the analytical hydrogenic wave function, the amplitude at  $r = 10$  already decreases to approximately  $10^{-25}$ , suggesting that the region  $r_f \in [5, 20]$  is sufficient to capture the physically relevant density. Therefore, the integration range and grid density were systematically varied within this window to identify the minimal parameters yielding a converged eigenvalue  $\varepsilon$ .

A grid search was performed for optimization of integration parameters: for each  $r_f$ ,  $N$  was gradually increased until either **numerical instability** occurred or **unphysical results** appeared. This approach avoids **program crashes** associated with overly dense grids while ensuring convergence monitoring for each configuration.

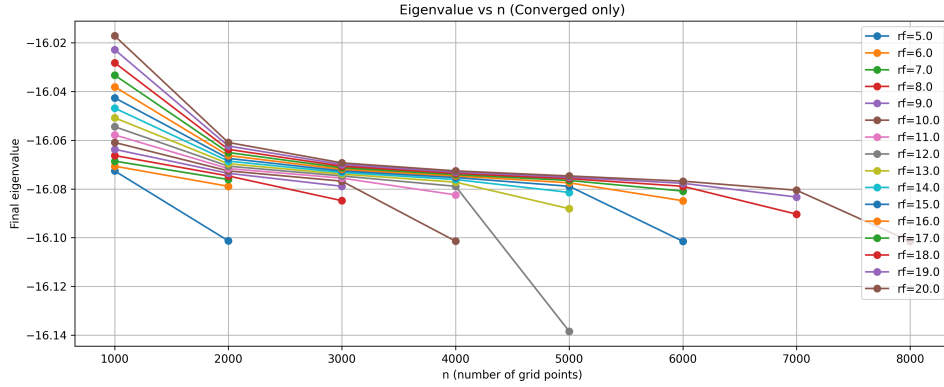


Figure 2: Integration Parameter Optimization

The results indicate that convergence is achieved around  $r_f = 17$ ,  $N = 4000$ . Beyond these values, further increase in grid resolution brings negligible improvement  $< 10^{-5}$  Ha in eigenvalues but substantially increases computation time. Hence,  $r_f = 17$  and  $N = 4000$  are recommended as optimal parameters balancing accuracy and efficiency.

#### 4.1.2 Mixing Parameter for SCF Convergence

With the optimized integration parameters, the next step was to investigate the convergence behavior of the SCF loop as a function of the potential mixing parameter  $\alpha$ . This parameter controls the update of the Kohn–Sham potential:

$$V_{\text{new}} = (1 - \alpha)V_{\text{old}} + \alpha V_{\text{out}}.$$

A small  $\alpha$  slows convergence but enhances stability, while a large  $\alpha$  accelerates convergence but risks divergence.

A systematic scan of  $\alpha$  values from 0.02 to 1.0 was performed to identify the optimal choice. For  $\alpha \leq 0.4$ , convergence was smooth and monotonic, though requiring more iterations at smaller  $\alpha$ . Remarkably, with the optimized integration parameters, the SCF cycle converged even at  $\alpha = 1$ , indicating excellent numerical stability of the potential update. However, this is not generally guaranteed for arbitrary systems, and thus  $\alpha \in [0.4, 0.6]$  is recommended as a balanced choice between convergence speed and robustness.

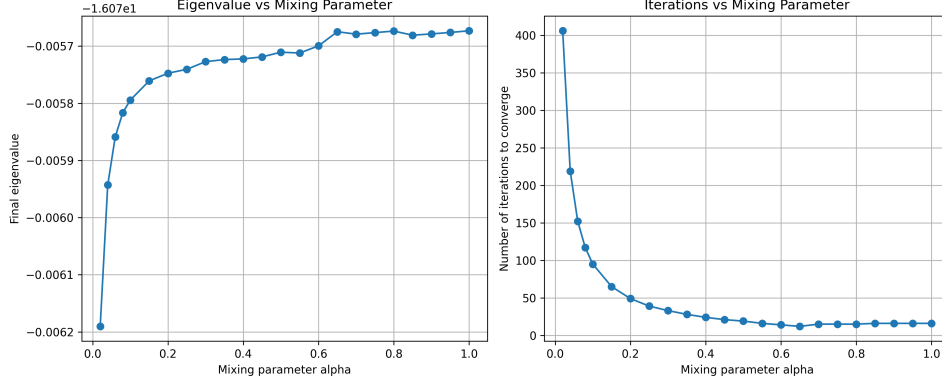


Figure 3: Mixing Parameter Optimization

## 4.2 Known Numerical Issues

Despite the overall correctness of the implemented algorithm and its agreement with analytical results under moderate grid conditions, numerical instability occurs during the tests at high grid densities.

During the SCF iterations, as  $N$  increases,  $P(r)$  at the outermost grid points decreases exponentially to zero. Consequently, the density  $\rho = P(r)/r$  approaches zero. This extreme results in overflow errors that propagate through the density evaluation, ultimately producing "NaN" values in the potential and eigenvalue. Once NaN values appear, the iterative process fails to proceed, and the calculation terminates prematurely.

Detailed inspection confirmed that all calculations are performed in double precision (float64), suggesting that the issue is not due to insufficient floating-point resolution but rather to the accumulation of numerical round-off errors in extreme-value regions. Interestingly, this instability becomes more pronounced as the grid resolution increases—contrary to expectation—while coarser grids (smaller  $N$ ) exhibit stable behavior and yield physically reasonable results.

It is noteworthy that the same algorithm implemented in Fortran does not exhibit these divergences. In the Fortran version, the eigenvalues converge smoothly without sudden jumps, even for very large  $r_f$  and  $N$ . This difference indicates that the issue may stem from Python's internal floating-point handling or intermediate variable casting during NumPy array operations.

To further investigate, a **Just-In-Time (JIT)** compiled version of the code was tested (using Numba). The JIT implementation significantly improved the execution speed but did not eliminate the NaN problem, implying that the instability is intrinsic to the numerical formulation rather than to the interpreter overhead.

In practice, the divergence can be mitigated by reducing the grid density. Another possible solution is to reformulate the density evaluation using **logarithmic grids** or **asymptotic expansions** for large ( $r$ ), which are known to improve numerical stability in atomic DFT codes.

## 4.3 Theoretical Extensions

This implementation can be extended in several ways:

1. **Multi-electron systems:** Implementing multiple Kohn-Sham orbitals and handling orbital occupations

2. **Alternative XC functionals:** Implementing GGA (Generalized Gradient Approximation) or hybrid functionals
3. **Non-spherical systems:** Extending beyond radial symmetry

## 5 References

1. Pauling, L.; Wilson, E.B. Introduction to Quantum Mechanics, McGraw-Hill, New York, 1935.
2. Hartree, D.R. The calculation of atomic structures. Chapman & Hall, Ltd., London, 1957.
3. Parr, R.G.; Yang, W. Density functional theory of atoms and molecules. Oxford University Press, New York, 1989.
4. Perdew, J.P.; Zunger, A. Self-interaction correction to density-functional approximations for many-electron systems. Phys. Rev. B 23, 5048, 1981.
5. Salvadori, M.G. Numerical methods in engineering. New York, 1952.
6. Malyshkina MV, Novikov AS. Modern Software for Computer Modeling in Quantum Chemistry and Molecular Dynamics. *Compounds*. 2021; 1(3):134-144.