

Program RAtom. Wersja 1.3

Zbigniew Romanowski*

Interdisciplinary Centre for Materials Modelling ul. Pawinskiego 5a 02-106 Warsaw, Poland

(Dated: 18 stycznia 2009)

Program **RAtom** dedykowany jest do rozwiązywania równania Kohn-Shama dla atomu w formalizmie funkcjonału gęstości (DFT). W programie **RAtom** została zaimplementowana metoda rozwiązywania równania Kohn-Shama dla atomu neutralnego lub jonu w przybliżeniu potencjału sferycznie symetrycznego. Zaimplementowany algorytm oparty jest na metodzie elementu skończonego z funkcjami bazowymi Lobatto.

Keywords: Równanie Kohn-Shama; DFT; Równanie Poissona; Wielomiany Lobatto; Metoda elementu skończonego.

Contents

I. Wstęp	1
II. Definicja problemu	2
A. Równanie Kohna-Shama	2
B. Gęstość elektronowa	2
C. Zmiana zmiennej	3
D. Współczynnik obsadzenia stanu	3
E. Potencjał Hartree	3
F. Składowe energii całkowitej atomu	4
1. Całkowanie numeryczne	4
G. Procedura SCF	5
1. Początkowa gęstość elektronowa	5
2. Pętla SCF	5
H. Warunek stopu w SCF	6
I. Wyznaczanie gęstości elektronowej w SCF	6
J. Aproksymowanie gęstości elektronowej	6
K. Działanie ∇^2 na funkcję własną	6
III. Korzystanie z programu	7
A. Uruchomienie programu	7
B. Plik wejściowy .inp	7
C. Opis parametrów wejściowych	9
1. Definiowanie atomu	9
2. Przybliżenie energii korelacji i wymiany	9
3. Początkowa gęstość elektronowa	9
4. Solver	9
5. Aproksymacja gęstości elektronowej	10
6. Procedura SCF	10
7. Zapisywanie wyników	10
8. Przykładowe wartości parametrów	10
D. Plik wynikowy .out	11
E. Pliki wynikowe .nto, .L=?	11
F. Pliki wynikowe .rho, .app	11
IV. Przykłady zastosowań	11
Literatura	12

I. WSTĘP

Program **RAtom** służy do numerycznego rozwiązywania równania Kohna-Shama dla atomu lub jonu w

ramach formalizmu funkcjonału gęstości elektronowej, DFT^{2,11,12,14–16,18–21,25,26,28,37} (ang. *Density Functional Theory*). Zakłada się, że atom (jon) jest układem sferycznie symetrycznym. Dla układu z symetrią sferyczną trójwymiarowe równanie Kohna-Shama redukuje się do równania w jednym wymiarze, który rozwiązywany jest przez **RAtom**.

Program **RAtom** rozwiązuje nieliniowy układ równań różniczkowo-całkowych numerycznie wykorzystując metodę elementu skończonego^{1,3,5,13,17,27,36}. Wynikiem działania programu są wartości własne odpowiadające stanom elektronowym Kohna-Shama oraz energia całkowita atomu (jonu) i jej składowe.

Zadanie rozwiązywania nieliniowego równania Kohna-Shama dla atomu było już analizowane przez autora metodą różnic skończonych³¹ oraz metodą elementu skończonego z wykorzystaniem funkcji B-spline^{6–9,29,33} jako bazy³². W tej wersji programu równania różniczkowe i zagadnienie własne rozwiązywane są metodą elementu skończonego z funkcjami bazowymi Lobatto^{34,35}. Funkcje bazowe Lobatto nazywane są także wycalkowanymi wielomianami Legendrea (ang. *integrated Legendre polynomials*) mogą mieć dowolny stopień. Ponadto mają własności, które są bardzo pożądane z punktu widzenia numerycznego. To znaczy:

- mają skończony nośnik,
- łatwo tworzy się bazę wysokiego stopnia,
- funkcje bazowe stopnia większego niż 2 są równe zero na krańcach przedziału,
- macierz przekrycia jest rzadka i dobrze uwarunkowana.

Zastosowanie metody elementu skończonego sprowadza równanie różniczkowe do algebraicznego układu równań. Jednakże, z powodu powyższych własności, zastosowanie funkcji bazowych Lobatto prowadzi do macierzy o mniejszych rozmiarach w porównaniu do metody wykorzystującej B-spline.

Dodatkowo, w tej wersji programu został zaimplementowany algorytm adaptacyjny. Algorytm rozwiązywania

zagadnienia własnego, równania różniczkowego i aproksymacji oprate są na algorytmach adaptacyjnych. Zastosowanie algorytmów adaptacyjnych pozwala uzyskać bardzo dokładne wyniki małym kosztem obliczeniowym i co ważne bez ingerencji użytkownika. Wszystkie potrzebne parametry do adaptacyjności podawane są na początku działania programu.

Integralną składową programu **RAtom** jest biblioteka **RFem1D**, gdzie został zaimplementowany adaptacyjny algorytm rozwiązywania (metodą elementu skończonego z wykorzystaniem wielomianów Lobatto) równania różniczkowego i zagadnienia własnego w jednym wymiarze. Szczegółowy opis algorytmów zaimplementowanych w bibliotece **RFem1D** znajduje się w oddzielnym dokumencie.

Układ dokumentu jest następujący. W sekcji II przedstawiony jest rozwiązywany problem. Podane zostały wszystkie zaimplementowane wzory oraz zastosowane algorytmy numeryczne. W szczególności, opisana została metoda rozwiązywania nieliniowego zagadnienia własnego oraz algorytm aproksymacji gęstości elektronowej. W sekcji III podany został opis uruchamiania i wykorzystania programu **RAtom**. W sekcji IV podane zostały przykłady otrzymania bardzo dokładnych rozwiązań.

II. DEFINICJA PROBLEMU

W tej sekcji zdefiniowany jest rozwiązywany problem. Podane są wszystkie konieczne wyrażenia wykorzystywane podczas implementacji programu **RAtom**.

A. Równanie Kohna-Shama

W tej sekcji zdefiniowany jest problem, które można rozwiązać za pomocą programu **RAtom**. Sekcja ta nie zawiera wyprowadzeń tych wzorów, które można znaleźć w innych pracach autora^{31,32}.

Formalnie równanie Kohna-Shama ma postać równania Schrödingera z jedno-elektronową funkcją falową

$$\hat{H}\Psi_\mu(\mathbf{r}) = E_\mu \Psi_\mu(\mathbf{r}), \quad (1)$$

gdzie indeks μ oznacza liczby kwantowe jednoznacznie identyfikujące stan elektronowy Ψ_μ . Zastosowanie sferycznego układu współrzędnych $\mathbf{r} = (r, \theta, \varphi)$ umożliwia przedstawienie jedno-elektronowej funkcji falowej $\Psi_{n,\ell,m}(\mathbf{r})$ w postaci iloczynu

$$\Psi_{n,\ell,m}(\mathbf{r}) = \frac{1}{r} R_{n,\ell}(r) Y_{\ell,m}(\theta, \varphi) \quad (2)$$

W powyższym wzorze $Y_{\ell,m}$ jest harmoniką sferyczną a $R_{n,\ell}$ jest funkcją radialną. Z warunków fizycznych można pokazać, że funkcje radialne $R_{n,\ell}$ spełniają warunki brzegowe

$$\lim_{r \rightarrow 0} R_{n,\ell}(r) = \lim_{r \rightarrow \infty} R_{n,\ell}(r) = 0 \quad (3)$$

Ponadto, zastosowanie sferycznego układu współrzędnych sprowadza ogólną postać równania Kohna-Shama (1) do jednowymiarowego zagadnienia własnego

$$\left[-\frac{1}{2} \frac{d^2}{dr^2} + \frac{\ell(\ell+1)}{2r^2} + V(r) \right] R_{n,\ell}(r) = \epsilon_{n,\ell} R_{n,\ell}(r) \quad (4)$$

W powyższym równaniu ℓ jest poboczną liczbą kwantową a n jest indeksem numerycznym kolejne wartości własne dla ustalonej pobocznej liczby kwantowej ℓ . Równanie (4) rozwiązywane jest na przedziale $[0, r_c]$ z zerowymi warunkami brzegowymi

$$R_{n,\ell}(0) = R_{n,\ell}(r_c) = 0. \quad (5)$$

W równaniu (4) funkcja $V(r)$ jest potencjałem oddziaływania, który w DFT przedstawiony jest w postaci sumy potencjałów

$$V(r) = -\frac{Z}{r} + V_h(r) + V_{xc}(r). \quad (6)$$

W powyższym równaniu Z jest ładunkiem jądra atomu, V_h jest potencjałem elektrostatycznym pochodzącym od chmury elektronowej, V_{xc} jest potencjałem energii korelacji-wymiany obliczanym według wybranego przybliżenia.

B. Gęstość elektronowa

Do obliczenia powyższego potencjału potrzebna jest gęstość elektronowa $\tilde{\rho}$, która oblicza się ze wzoru

$$\tilde{\rho}(r) = \frac{1}{4\pi r^2} \sum_{\ell=0}^{\ell_{\max}} (2\ell+1) \sum_{n=0}^{N_\ell} f_{n,\ell} R_{n,\ell}^2(r), \quad (7)$$

W powyższym wzorze $f_{n,\ell}$ oznacza współczynnik obsadzenia stanu $\Psi_{n,\ell}$. Suma tych współczynników jest równa całkowitej liczbie elektronów w układzie

$$N = \sum_{n,\ell} f_{n,\ell}. \quad (8)$$

Jeżeli temperatura układu jest równa 0K, to $f_{n,\ell} = 1$ gdy spin elektronu jest uwzględniony, a gdy spiny elektronów nie są uwzględniane mamy $f_{n,\ell} = 2$. W tej wersji programu **RAtom** spiny elektronów nie są uwzględniane.

We wzorze (7) symbol ℓ_{\max} oznacza maksymalną poboczną liczbę kwantową, dla której rozwiązywane jest równanie (4). Wartość ℓ_{\max} wyznaczyć można na podstawie liczby atomowej atomu dla którego rozwiązywane jest równanie Kohna-Shama.

W programie **RAtom** jest możliwość zwiększenia maksymalnej liczby pobocznej wynikającej z liczby atomowej atomu. Do tego celu służy parametr **Atom_ExtraAngular**, określający liczbę dodatkowych pobocznych liczb kwantowych, wtedy

$$\ell_{\max} = \ell_{\max} + \text{Atom_ExtraAngular}. \quad (9)$$

We wzorze (7) symbol N_ℓ oznacza liczbę wartości własnych, które należy wyznaczyć dla pobocznej liczby kwantowej ℓ . Liczbę N_ℓ wyznacza się na podstawie liczby atomowej (parametr `Atom_Proton`). Wartości przypisywane liczbie N_ℓ podane są w tabeli I.

Tablica I: Standardowa liczba obliczanych wartości własnych, N_ℓ , przez program `RAtom` dla każdego atomu z układu okresowego pierwiastków. Wartości zostały podane dla kilku najmniejszych pobocznych liczb kwantowych $\ell = 0, 1, 2, 3$. Kreska, $-$, oznacza, że stan w atomie neutralnym nie jest obsadzony.

N_ℓ	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$
0	—	H ... Be	H ... Ca	H ... Zn
1	H, He	B ... Mg	Sc ... Sr	Ga ... ∞
2	Li ... Ne	Al ... Zn	Y ... ∞	—
3	Na ... Ar	Ga ... Cd	—	—
4	K ... Kr	In ... ∞	—	—
5	Rb ... ∞	—	—	—

Tabelę tę należy rozumieć w następujący sposób. W komórce tabeli I znajdują się wszystkie atomy, którym dla pobocznej liczby kwantowej ℓ (podanej w kolumnie) obliczana jest standardowa liczba wartości własnych N_ℓ (podanych w wierszu). Na przykład, dla atomów od Li do Ne dla $\ell = 0$ obliczne są $N_\ell = 2$ wartości własne. Gdy w komórce znajduje się kreska, $-$, to w atomie neutralnym stan ten nie jest obsadzony. Natomiast, gdy w komórce znajduje się znak $\alpha \dots \infty$, to atomy o liczbie atomowej większej lub równej atomu α . Dodatkowo, dla atomów znajdujących się w wierszu $N_\ell = 0$ nie są obliczne żadne wartości własne.

W programie `RAtom` dla zadanego atomu jest możliwość zwiększenia liczby N_ℓ podanej w tabeli I. Do tego celu służy parametr `Atom_ExtraShell`, określający liczbę dodatkowych pobocznych liczb kwantowych, wtedy

$$N_\ell = N_\ell + \text{Atom_ExtraShell}. \quad (10)$$

Obliczenia zajmują najmniej czasu gdy oba parametry `Atom_ExtraAngular`, `Atom_ExtraShell` są równe 0.

C. Zmiana zmiennej

Ze względów obliczeniowych wprowadza się wielkość

$$\varrho(r) = 4\pi r^2 \tilde{\varrho}(r), \quad (11)$$

gdzie $\tilde{\varrho}$ zdefiniowane jest wzorem (7). Wielkość ta jest gęstością elektronową $\tilde{\varrho}$ wycalkowaną po kącie bryłowym 4π .

Ponieważ rozwiązywane jest zagadnienie w sferycznej wnęce o promieniu r_c , to mamy

$$\tilde{\varrho}(r_c) = \varrho(r_c) = 0. \quad (12)$$

Ponadto wiemy, że $\tilde{\varrho}(0)$ jest skończone. Więc, z powodu czynnika r^2 we wzorze (11), mamy $\varrho(0) = 0$.

Opisanie algorytmu rozwiązywania nieliniowego równania Kohna-Shama z wykorzystaniem wielkości $\varrho(r)$ jest bardziej zwięzłe niż z wykorzystaniem wielkości $\tilde{\varrho}(r)$. Dlatego wszystkie wzory podane poniżej odwołują się do tej wielkości. Konsekwentnie stosując $\varrho(r)$ implementacja numeryczna jest prostsza i bardziej efektywna.

Dla porządku poniżej podane jest jawne wyrażenie dla wprowadzonej wielkości

$$\varrho(r) = \sum_{\ell=0}^{\ell_{\max}} (2\ell + 1) \sum_{n=0}^{N_\ell} f_{n,\ell} R_{n,\ell}^2(r), \quad (13)$$

gdzie symbole występujące w tym wzorze zostały objaśnione w sekcji II B.

D. Współczynnik obsadzenia stanu

Każdemu stanowi elektronowemu przypisany jest współczynnik obsadzenia $f_{n,\ell}$. Współczynnik ten określa liczbę elektronów znajdujących się na stanie określonym przez parę liczb kwantowych (n, ℓ) bez uwzględnienia degeneracji ze względu na magnetyczną liczbę kwantową. Gdy spis elektronu jest uwzględniony, to temperaturze 0K współczynnik $f_{n,\ell}$ może przyjmować tylko dwie wartości: $f_{n,\ell} = 0$ gdy stan jest nieobsadzony lub $f_{n,\ell} = 1$ gdy stan jest obsadzony. W temperaturach większych od 0K należy zastosować rozkład Fermiego-Diraca do obliczania współczynników obsadzeń $f_{n,\ell}$.

Pozostaje kwestia ustalenia, które stany są obsadzone, a które nie są obsadzone. Do ustalenia stanów obsadzonych została wykorzystana zasada mówiąca, że obsadzane są stany o najniższej energii, czyli stany o najmniejszej wartości własnej. Oznacza to, że aby obliczyć współczynniki obsadzenia $f_{n,\ell}$ należy:

1. Obliczyć wartości własne $\epsilon_{n,\ell}$ stanów $\Psi_{n,\ell}$ odpowiadające wszystkim parom (n, ℓ) takim, że $\ell = 0, \dots, \ell_{\max}$ oraz $n = 0, \dots, N_\ell$.
2. Posortować malejąco ciąg trójek $E = \{(n, \ell, \epsilon_{n,\ell})\}$ po wartości własnej $\epsilon_{n,\ell}$.
3. W posortowanym ciągu E przypisywać każdemu stanowi (n, ℓ) liczbę $f_{n,\ell} = 1$ tak długo aż liczba obsadzonych stanów z uwzględnieniem degeneracji ze względu na magnetyczną liczbę kwantową będzie większa niż liczba elektronów w układzie.

Powyżej opisany algorytm umożliwia obliczyć w prosty i niezawodny sposób współczynniki obsadzeń $f_{n,\ell}$ potrzebne do obliczenia gęstości elektronowej.

E. Potencjał Hartree

Aby obliczyć potencjał Hartree, V_h , rozwiązywane jest równanie Poissona, które w sferycznym układzie współ-

rzędnych przyjmuje postać

$$\frac{1}{r} \frac{d^2}{dr^2} [rV_h(r)] = -4\pi\tilde{\varrho}(r). \quad (14)$$

Równanie to ma rozwiązanie, które można wyprowadzić na podstawie funkcji Green'a dla operatora Laplace'a¹⁰

$$V_h(r) = 4\pi \left[\frac{1}{r} \int_0^r t^2 \tilde{\varrho}(t) dt + \int_r^\infty t \tilde{\varrho}(t) dt \right]. \quad (15)$$

Bezpośrednie zastosowanie równanie (15) do obliczania potencjału V_h prowadzi do bardzo nieefektywnego algorytmu, gdyż dla każdego argumentu r dwa numeryczne całkowania muszą być wykonane. Równanie (15) zostało wykorzystane tylko do obliczenia wartości brzegowej dla r_c . Wprowadzając podstawienie

$$U_h(r) = rV_h(r) \quad (16)$$

równanie (14) upraszcza się do postaci

$$-\frac{d^2 U_h(r)}{dr^2} = 4\pi r \tilde{\varrho}(r) = \frac{\varrho(r)}{r}. \quad (17)$$

Powyższe równanie rozwiązywanie jest na przedziale $[0, r_c]$ z dwoma warunkami brzegowymi Dirichleta

$$U_h(0) = 0 \quad \text{oraz} \quad U_h(r_c) = r_c V_h(r_c). \quad (18)$$

Wartość $V_h(r_c)$ jest obliczane ze wzoru (15). Z założenia wiemy, że funkcje $R_{n,\ell}$ są niezerowe tylko na przedziale $[0, r_c]$. Oznacza to, że gęstość elektronowa zdefiniowana równaniem (13) może być niezerowa tylko na przedziale $[0, r_c]$. Tym samym obliczanie potencjału V_h dla $r = r_c$ ze wzoru (15) sprowadza się do obliczenia jednej całki

$$V_h(r_c) = \frac{4\pi}{r_c} \int_0^{r_c} t^2 \tilde{\varrho}(t) dt. \quad (19)$$

Jednakże, całka

$$N = 4\pi \int_0^{r_c} t^2 \tilde{\varrho}(t) dt = \int_0^{r_c} \varrho(t) dt. \quad (20)$$

równa jest liczbie elektronów w atomie N . Wykorzystując tę równość mamy

$$V_h(r_c) = N/r_c. \quad (21)$$

Równanie (17) zdefiniowane na przedziale $[0, r_c]$ wraz z warunkami brzegowymi Dirichleta

$$U_h(0) = 0 \quad \text{oraz} \quad U_h(r_c) = N \quad (22)$$

rozwiązywane jest za pomocą algorytmów zawartych w bibliotece RFem1D.

F. Składowe energii całkowitej atomu

Gdy zadana jest gęstość elektronowa układu ϱ , to na podstawie formalizmu DFT można obliczyć energię całkowitą atomu ze wzoru

$$E_t = \sum_{n,\ell} f_{n,\ell} \epsilon_{n,\ell} + \int_0^{r_c} W(r) \varrho(r) dr. \quad (23)$$

W powyższym wzorze potencjał pomocniczy W

$$W(r) = [\epsilon_{xc}(r) - V_{xc}(r)] - \frac{1}{2} V_h(r) \quad (24)$$

wyrażony jest przez różnicę gęstości energii korelacji-wymiany ϵ_{xc} i potencjału energii korelacji-wymiany V_{xc} . Energie całkowitą E_t można także obliczyć jako sumę jej składowych

$$E_t = E_k + E_n + E_h + E_{xc}. \quad (25)$$

W powyższym wzorze E_k jest energią kinetyczną

$$E_k = \sum_{n,\ell} f_{n,\ell} \epsilon_{n,\ell} + \int_0^{r_c} V(r) \varrho(r) dr,$$

E_n jest energią oddziaływania elektrostatycznego chmury elektronowej z jądrem atomowym

$$E_n = Z \int_0^{r_c} \frac{\varrho(r)}{r} dr, \quad (26)$$

E_h jest energią oddziaływania elektrostatycznego chmury elektronowej z samą sobą

$$E_h = \frac{1}{2} \int_0^{r_c} V_h(r) \varrho(r) dr, \quad (27)$$

E_{xc} jest energia korelacji-wymiany

$$E_{xc} = \int_0^{r_c} \epsilon_{xc}(r) \varrho(r) dr. \quad (28)$$

Wszystkie powyższe całki obliczane są numerycznie.

1. Całkowanie numeryczne

Energię całkowitą atomu i jej składowe oblicza się obliczając całki z poprzedniej sekcji numerycznie. Wszystkie te całki mają postać

$$I = \int_0^{r_c} w(r) \varrho(r) dr, \quad (29)$$

gdzie funkcja $w(r)$ jest zadana. Dla gęstości elektronowej ϱ znana jest jej aproksymacja a tym samym ciąg węzłów $0 = r_0 < r_1 < \dots < r_M = r_c$. Związku z tym powyższą całkę przedstawiamy w postaci sumy

$$I = \sum_{i=0}^M \int_{r_i}^{r_{i+1}} w(r) \varrho(r) dr, \quad (30)$$

Pomiędzy węzłami funkcja ϱ zadana jest wielomianem stopnia `Rho_Deg`. Dlatego też całkę na przedziale $[r_i, r_{i+1}]$ obliczamy za pomocą kwadratur Gaussa. Przyjęte zostało, że stopień zastosowanej kwadratury Gaussa wynosi $(\text{Rho_Deg} + 3)$, gdzie 3 zostało dodane ‘na wszelki wypadek’.

G. Procedura SCF

Równanie (4) jest nieliniowe, gdyż potencjał V zadany wzorem (6) zależy od gęstości elektronowej ϱ , która jest wyznaczana ze wzoru (13). Ten układ równań różniczkowo-całkowych rozwiązywany jest iteracyjnie. Proces iteracyjny rozpoczyna się od wyboru początkowej gęstości elektronowej.

1. Początkowa gęstość elektronowa

Wybór początkowej gęstości elektronowej ϱ_0 ma bardzo istotny wpływ na liczbę koniecznych iteracji w procedurze SCF do otrzymania żądanej zbieżności. W szczególności, gdyby ϱ_0 było równe szukanej gęstości elektronowej rozpatrywanego atomu, to zbieżność w procedurze SCF otrzymano by w jednym kroku. Dlatego też ważne jest, aby ϱ_0 było bliskie szukanemu rozwiązaniu.

Z drugiej strony, w zasadzie, ϱ_0 może być dowolną funkcją spełniającą następujące warunki:

1. Musi być funkcją dodatnią, $\tilde{\varrho}_0(r) \geq 0$ dla $r \geq 0$.
2. Musi całkować się do liczby elektronów

$$N = 4\pi \int_0^\infty r^2 \tilde{\varrho}_0(r) dr = \int_0^\infty \varrho_0(r) dr. \quad (31)$$

3. Musi zanikać eksponencjalnie dla dużych r .
4. Musi należeć do obszaru zbieżności procedury SCF.

Łatwo jest wybrać funkcję ϱ spełniającą pierwsze trzy warunki, trudniej jest spełnić warunek czwarty. Łatwo zauważyć, że gęstość elektronowa postaci

$$\varrho_0(r) = 4\pi r^2 \tilde{\varrho}_0(r) = \frac{N^4}{16} r^2 e^{-Nr/2}, \quad (32)$$

gdzie N jest liczbą elektronów w atomie, spełnia wszystkie trzy wyżej wymienione warunki. Ponadto, doświadczenia numeryczne pokazują, że dla $\varrho_0(r)$ zadanego wzorem (32) potrzeba od 20 do 30 iteracji SCF dla $N \leq 50$.

`RAtom` był często wykorzystywany do analizowania atomów N, Al, Ga, In. Na podstawie doświadczenia zdobytego podczas analizowania tych atomów została zaproponowana gęstość elektronowa postaci

$$\varrho_0(r) = 4\pi c r^2 e^{-\alpha r}, \quad (33)$$

gdzie stałe c, α zostały znalezione eksperymentalnie. Powyższy wzór ma postać identyczną ze wzorem (32), jednakże aby go stosować należy podać wartości stałych c, α .

Tablica II: Obliczone stałe początkowej gęstości elektronowej $\varrho_0(r)$ dla atomów N, Al, Ga, In.

Atom	c	α
N	15.283	3.8
Al	86.0581	5.5
Ga	423.074	7.0
In	3368.9916	12.0

Wartości tych stałych podane są w Tabeli II. Stosując te wartości zbieżność w procedurze SCF otrzymuje się szybciej, szczególnie dla atomu In.

W programie `RAtom` jest możliwość podania własnych współczynników c, α . Jeżeli paramter `Rho_Default` jest równy ‘Yes’, to stosowany jest wzór (32). Jeżeli paramter `Rho0_Default` jest równy ‘No’, to stosowany jest wzór (33), a wartości współczynników c, α podane są w parametrach `Rho0_c` oraz `Rho0_Alpha`. Użytkownik zobowiązany jest podać takie wartości c, α , aby spełniona była równość (31).

Początkowa gęstość elektronowa ϱ_0 jest aproksymowana apaptacyjnie kawałkami wielomianami Lobatto. W programie `RAtom` stopień wykorzystanych wielomianów określony jest parametrem `Rho_Deg`, a dokładność adaptacyjnej aproksymacji parametrem `Rho_Delta`. Otrzymana podczas aproksymacji siatka $0 = r_0 < r_1 < \dots < r_M = r_c$ jest wykorzystana do obliczenia numerycznie całki (31). Całka ta ma być równa (z dużą dokładnością) liczbie elektronów. Całkę tę oblicza się jako sumę całek po każdym przedziale $[r_i, r_{i+1}]$

$$\int_0^\infty \varrho_0(r) dr = \sum_{i=0}^M \int_{r_i}^{r_{i+1}} \varrho_0(r) dr, \quad (34)$$

gdzie całkę $\int_{r_i}^{r_{i+1}} \varrho_0(r) dr$ oblicza się kwadraturami Gaussa rzędu `Rho_Deg`. Obliczona wartość całki podawana jest na początku uruchomienia programu `RAtom`.

2. Pętla SCF

Na mocy twierdzenia Banacha o punkcie stałym nieliniowe równanie Kohna-Shama (4) rozwiązywany jest według poniżej podanego schematu.

1. Niech $\varrho = \varrho_0$.
2. Oblicz V_h rozwiązując równanie Poissona (16).
3. Oblicz potencjał V ze wzoru (6).
4. Dla zagadnienia (4) znajdź kilka najmniejszych wartości własnych i odpowiadających im funkcji własnych.
5. Oblicz współczynniki obsadzeń $f_{n,\ell}$.
6. Oblicz nowy rozkład ładunku ϱ^{ks} ze wzoru (13).

7. Jeżeli warunek stopu spełniony, to BREAK.
8. Wyznacz nową gęstość elektronową ϱ^{new} jako kombinację liniową ϱ^{ks} oraz ϱ .
9. Oblicz ϱ jako aproksymację ϱ^{new} .
10. GOTO punkt 2.

Po zakończeniu pętli SCF można obliczyć dodatkowe wielkości takie jak składowe energii całkowitej zdefiniowane w Sekcji II F.

H. Warunek stopu w SCF

Nie jest oczywiste jaki powinien być warunek stopu procedury SCF opisanej w sekcji II G. Warunek stopu jest szczególnie problematyczny, gdy zbieżność jest wolna, a tym samym wielkości fizyczne w dwóch kolejnych krokach iteracji SCF niewiele się od siebie różnią. Ponadto, z punktu widzenia wydajności, obliczanie warunku stopu powinno zawierać niewiele czasu w porównaniu do czasu potrzebnego na wykonanie jednej pełnej iteracji SCF.

Zwykle w procedurze SCF jako warunek stopu bierze się pod uwagę różnicę energii całkowitej w dwóch kolejnych krokach iteracji SCF. Jednakże, obliczenie energii całkowitej zadanej wzorem (23) wymaga obliczenia całki numerycznie co jest dosyć czasochłonne. Ponadto, energia całkowita jest zwykle wielką liczbą i trudno ją wyznaczyć szybko z dużą dokładnością.

Z doświadczeń numerycznych wynika, że energia całkowita jest wolno zbieżna. Szczególnie jest to widoczne, gdy chce się otrzymać wyniki z dużą dokładnością.

Na podstawie tych doświadczeń numerycznych zaproponowałem, aby warunkiem stopu była różnica w kolejnych krokach sumy wartości własnych. Niech

$$E_e^i = \sum_{\ell=0}^{\ell_{\max}} \sum_{n=0}^{N_{\ell}} \epsilon_{n,\ell}^i \quad (35)$$

będzie sumą wszystkich obliczonych wartości własnych w i -tej iteracji SCF. Wtedy, procedura SCF jest zakończona, gdy spełniony jest warunek

$$|E_e^i - E_e^{i+1}| < \text{Scf_Diff}, \quad (36)$$

gdzie wielkość `Scf_Diff` jest parametrem programu `RAtom`. Zastosowanie powyższego warunku stopu jest bardzo wydajne, gdyż obliczenie E_e^i wymaga tylko wykonania sumy (35) zawierającej niewiele elementów.

Drugim warunkiem stopu jest maksymalna liczba iteracji w procedurze SCF. Warunek ten został wprowadzony z powodów technicznych i jest często wykorzystywany podczas analizowania znaczenia parametrów wejściowych. W programie `RAtom` paramter ten nazywa się `Scf_MaxIter`.

I. Wyznaczanie gęstości elektronowej w SCF

Po rozwiązaniu liniowego równania Kohn-Shama, sposób w jaki wyznaczana jest gęstość elektronowa ma istotny wpływ na szybkość zbieżności procedury SCF.

Niech ϱ^{ks} oznacza gęstość elektronową wyznaczoną ze wzoru (13). Najprostszym sposobem to przyjąć, że aktualna gęstość elektronowa równa jest tej wielkości. Jednakże takie przypisanie prowadzi do bardzo wolnej a czasami braku zbieżności.

Aby zwiększyć stabilność procedury SCF i zwiększyć jej szybkość zbieżności do wyznaczenia aktualnej gęstości elektronowej wykorzystuje się gęstość elektronową przed rozwiązaniem liniowego równania Kohn-Shama ϱ oraz po rozwiązaniu liniowego równania Kohn-Shama ϱ^{ks} . Na podstawie kombinacji liniowej tych dwóch wielkości

$$\varrho^{\text{new}}(r) = \alpha \varrho(r) + (1 - \alpha) \varrho^{\text{ks}}(r) \quad (37)$$

tworzona jest nowa gęstość elektronowa ϱ^{new} wykorzystywana w kolejnym kroku iteracji SCF. Współczynnik $0 \leq \alpha \leq 1$ jest nazywany współczynnikiem mieszania (ang. *mixing coefficient*).

W programie `RAtom` wartość współczynnika α określone jest parametrem `Scf_Mix`. Wartość tego parametru zależy od analizowanego układu. $\alpha = 0.5$ jest zwykle wystarczające, aby procedura SCF była zbieżna.

J. Aproksymowanie gęstości elektronowej

Zastosowanie kombinacji liniowej (37) wymaga przechowywania przez pewien czas działania programu dwóch gęstości elektronowych jednocześnie, co prowadzi do większego zużycia pamięci operacyjnej. Nadmiarowości zużycia pamięci operacyjnej nie można uniknąć, gdy się stosuje ten algorytm, ale nie jest to problem dla obecnych komputerów

Nowa gęstość elektronowa ϱ^{new} jest zdefiniowana wzorem (37) dla funkcji. Reprezentowanie takiego wzoru numerycznie nie jest banalne. Aby uporać się z tą trudnością gęstość ϱ^{new} otrzymuje się jako aproksymację funkcji występującej po prawej stronie równania (37).

Funkcja aproksymowana jest wielomianami sklejającymi trzeciego stopnia (ang. *cubic spline*). W implementacji zostały wykorzystane funkcje `spline`, `splint` z książki *Numerical Recipes*³⁰. Aproksymacja jest wykonana obliczając wartość funkcji w równo oddalonych węzłach. W programie `RAtom` liczbę tych węzłów określa paramter `Rho_Node`. Tysiąc węzłów jest zwykle wystarczające aby uzyskać sensowne wartości.

K. Działanie ∇^2 na funkcję własną

Podczas obliczania całki kinetycznej w metodzie LCAO dla molekuly, gdzie funkcjami bazowymi są funkcje własne swobodnego atomu, potrzebne jest obliczenie

działania operatora ∇^2 na funkcję własną. Działanie to ma być obliczone w sferycznym układzie współrzędnych.

Operator ∇^2 w sferycznym układzie współrzędnych ma postać^{4,24}

$$\nabla^2 = \frac{1}{r} \frac{\partial^2}{\partial r^2} r - \frac{1}{r^2} \hat{L}^2, \quad (38)$$

gdzie \hat{L}^2 jest operatorem kwadratu momentu pędu. Funkcjami własnymi operatora \hat{L}^2 są zespolone harmoniki sferyczne

$$\hat{L}^2 Y_{\ell,m} = \ell(\ell+1) Y_{\ell,m}. \quad (39)$$

Wykorzystując powyższą zależność można policzyć działanie operatora ∇^2 na funkcję bazową zdefiniowaną wzorem (2)

$$\begin{aligned} \nabla^2 \Phi &= \nabla^2 \left[\frac{R(r)}{r} Y_{\ell,m} \right] \\ &= \left[\frac{1}{r} \frac{\partial^2}{\partial r^2} r - \frac{1}{r^2} \hat{L}^2 \right] \left[\frac{R(r)}{r} Y_{\ell,m} \right] \\ &= Y_{\ell,m} \frac{1}{r} \frac{d^2 R(r)}{dr^2} - \frac{R(r)}{r^3} \hat{L}^2 Y_{\ell,m} \\ &= \left[\frac{1}{r} \frac{d^2 R(r)}{dr^2} - \ell(\ell+1) \frac{R(r)}{r^3} \right] Y_{\ell,m} \end{aligned} \quad (40)$$

Wprowadzając funkcje radialną

$$\tilde{R}(r) = \frac{1}{r} \frac{d^2 R(r)}{dr^2} - \ell(\ell+1) \frac{R(r)}{r^3} \quad (41)$$

otrzymujemy

$$\begin{aligned} \nabla^2 \Psi_{n,\ell,m}(r, \theta, \varphi) &= \nabla^2 \left[\frac{R_{n,\ell}(r)}{r} Y_{\ell,m}(\theta, \varphi) \right] \\ &= \tilde{R}_{n,\ell}(r) Y_{\ell,m}(\theta, \varphi). \end{aligned} \quad (42)$$

Oznacza to, że działając operatorem ∇^2 na funkcję bazową $\Psi_{n,\ell,m}(\mathbf{r})$ z częścią katową określoną funkcją $Y_{\ell,m}$, otrzymujemy funkcję bazową o tej samej zależności katowej $Y_{\ell,m}$, a część radialną zadana jest funkcją $\tilde{R}_{n,\ell}(r)$, która zdefiniowana jest wzorem (41).

Program **RAtom** oblicza wartość funkcji $\tilde{R}_{n,\ell}(r)$ dla argumentu r należącego do nośnika funkcji $R_{n,\ell}(r)$. Wartości te zapisywane są do pliku wraz z funkcją $R_{n,\ell}(r)$.

III. KORZYSTANIE Z PROGRAMU

A. Uruchomienie programu

Program **RAtom** jest programem uruchamianym z linii komend. Jedynym argumentem uruchomienia jest nazwa pliku zawierającego parametry definiujące problem i algorytmy numeryczne. Efektem działania programu są:

- wartości i funkcje własne atomu,
- energia całkowita atomu i jej składowe,

- gęstość elektronowa.

Przed uruchomieniem programu należy przygotować plik z parametrami wejściowymi i nadać mu rozszerzenie **.inp**. Jeżeli przygotowany plik ma nazwę **C.inp**, to aby uruchomić program należy w linii komend wpisać:

```
> ratom C
```

Obliczone wielkości zapisywane są do plików wynikowych. Nazwy plików wynikowych różnią się tylko rozszerzeniami. **RAtom** generuje następujące pliki wynikowe, które zawierają:

- **.out** \Rightarrow wartości własne i energię atomu,
- **.nto** \Rightarrow funkcje własne atomu,
- **.L=0** \Rightarrow węzły i współczynniki wielomianów Lobatto funkcji własnych dla pobocznej liczby kwantowej ℓ . (W tym przypadku $\ell = 0$.)
- **.rho** \Rightarrow rozkład gęstości elektronowej,
- **.app** \Rightarrow węzły i współczynniki wielomianów Lobatto dla aproksymacji gęstości elektronowej,
- **.log** \Rightarrow informacje diagnostyczne.

W kolejnych sekcjach podany jest opis formatów plików wejściowych i wynikowych wraz ze znaczeniem parametrów w nich zawartych. Ogólnie, zarówno pliki wejściowe jak i wynikowe są plikami tekstowymi. Jeżeli linia w pliku zaczyna się od znaku #, to traktowana jest jak komentarz. Wszystkie wielkości podane są w jednostkach atomowych, to znaczy długość podana jest w bohrach a energia w hartree. W jednostkach atoomowych mamy:

- masa i ładunek elektronu $m = 1, e = 1$,
- stała Plancka $\hbar = 1$,
- 1 bohr = 0.529117 Å,
- 1 hartree = 27.21165 eV.

B. Plik wejściowy .inp

Plik wejściowy do programu **RAtom** ma rozszerzenie **.inp** i jest plikiem tekstowym. W jednej linii pliku znajduje się jedno polecenie. Linie zaczynające się od znaku # są pomijane. Kolejność podawania komend w pliku nie jest istotna. Każda komenda zaczyna się od przedrostka określającego grupę parametrów. Parametry zostały podzielone na następujące grupy parametrów.

- **Atom** - definicje rozpatrywany atom (jon).
- **XC** - definiuje zastosowane przybliżenia na potencjał korelacji i wymiany.
- **Rho0** - definiuje początkową gęstość elektronową.

- **Solver** - definiuje parametry dla algorytmu rozwiązywania zagadnienia własnego i równania Poissona.
- **Rho** - definiuje sposób aproksymacji gęstości elektronowej.
- **Scf** - definiuje parametry dla procedury SCF.
- **Out** - definiuje parametry zapisu wyników.

Poniżej podane są nazwy i znaczenie wszystkich komend. Szczegółowy opis parametrów wraz z komentarzem znajduje się w sekcji III C. Każdy parametr musi być ustalonego typu. Typ parametru oznaczany jest symbolem. Wyjaśnienie znaczenia symboli podane jest poniżej.

- \mathbb{R} , liczba rzeczywista.
- \mathbb{R}^+ , liczba rzeczywista dodatnia.
- \mathbb{Z} , liczba całkowita.
- \mathbb{Z}^+ , liczba całkowita dodatnia.
- \mathbb{S} , jeden łańcuch znakowy.
- $\mathbb{S} = \{s_1, s_2, \dots\}$, zbiór łańcuchów znakowych.

W pliku wejściowym wszystkie wielkości podane są w jednostkach atomowych, to znaczy długość podana jest w bohrach a energia w hartree. Poniżej podane są wszystkie nazwy parametrów z pliku wejściowego `.inp`. Dla każdego parametru podany jest jego typ oraz zakres dozwolonych wartości.

- **Atom_Proton**, \mathbb{Z}^+ . Liczba protonów w atomie.
- **Atom_Electron**, \mathbb{Z}^+ . Liczba elektronów w atomie.
- **Atom_ExtraShell**, \mathbb{Z}^+ . Liczba obliczanych nieobsadzonych (dodatkowych) funkcji własnych (stanów) dla ustalonej pobocznej liczby kwantowej ℓ .
- **Atom_ExtraAngular**, \mathbb{Z}^+ . Liczba dodatkowo uwzględnionych pobocznych liczb kwantowych ℓ .
- **XC_Exch**, $\mathbb{S} = \{\text{null}, \text{slater}\}$. Nazwa przybliżenia dla energii wymiany.
- **XC_Corr**, $\mathbb{S} = \{\text{null}, \text{vwn}\}$. Nazwa przybliżenia dla energii korelacji.
- **Rho0_Default**, $\mathbb{S} = \{\text{Yes}, \text{No}\}$. Jeżeli ‘Yes’, to zastosowany zostanie wzór (32) na obliczanie początkowej gęstości elektronowej. Jeżeli ‘No’, to zostaną przeczytane parametry **Rho0_c**, **Rho0_Alpha** i zastosowany zostanie wzór (33).
- **Rho0_c**, **Rho0_Alpha**, \mathbb{R}^+ . Parametry c, α ze wzoru (33) definiujące początkową gęstość elektronową.
- **Solver_EigAdapt**, $\mathbb{S} = \{\text{Yes}, \text{No}\}$. Jeżeli ‘Yes’, to zastosowany zostanie algorytm adaptacyjny do rozwiązywania zagadnienia własnego.

- **Solver_EigNode**, \mathbb{Z}^+ . Liczba węzłów w siatce radialnej do rozwiązywania zagadnienia własnego. Jeżeli stosowany jest algorytm adaptacyjny to jest to początkowa liczba węzłów w siatce.
- **Solver_EigLobatto**, $\mathbb{Z}^+ \geq 2$. Rząd zastawianych wielomianów Lobatto do rozwiązywania zagadnienia własnego.
- **Solver_EigAbsMaxCoef**, \mathbb{R}^+ . Maksymalna dozwolona wartość współczynnika rozwinięcia w procedurze adaptacyjnej zagadnienia własnego.
- **Solver_EigAbsTol**, \mathbb{R}^+ . Bezwzględny błąd obliczeń wartości własnych podczas rozwiązywania uogólnionego zagadnienia własnego $\mathbf{Hc} = E\mathbf{Sc}$. Przekazywany do procedury `dsbevx` z biblioteki LAPACK.
- **Solver_PsnAdapt**, $\mathbb{S} = \{\text{Yes}, \text{No}\}$. Jeżeli ‘Yes’, to zastosowany zostanie algorytm adaptacyjny do rozwiązywania równania Poissona.
- **Solver_PsnNode**, \mathbb{Z}^+ . Liczba węzłów w siatce radialnej do rozwiązywania równania Poissona. Jeżeli stosowany jest algorytm adaptacyjny to jest to początkowa liczba węzłów w siatce.
- **Solver_PsnDeg**, $\mathbb{Z}^+ \geq 2$. Rząd zastawianych wielomianów Lobatto do rozwiązywania równania Poissona.
- **Solver_PsnAbsMaxCoef**, \mathbb{R}^+ . Maksymalna dozwolona wartość współczynnika rozwinięcia w procedurze adaptacyjnej dla równania Poissona.
- **Solver_Rc**, \mathbb{R}^+ , w bohrach. Promień ustalenia zerowego warunku brzegowego Dirichleta dla funkcji własnych równania Kohna-Shama.
- **Rho_Deg**, \mathbb{Z}^+ . Rząd zastawianych wielomianów Lobatto podczas aproksymacji gęstości elektronowej.
- **Rho_Delta**, \mathbb{R}^+ . Gęstość elektronowa jest aproksymowana kawałkami wielomianami. Parametr ten definiuje dokładność aproksymacji na każdym przedziale.
- **Scf_MaxIter**, \mathbb{Z}^+ . Maksymalna liczba iteracji w procedurze SCF.
- **Scf_Diff**, \mathbb{R}^+ , w hartree. Warunek stopu w procedurze SCF, patrz sekcja II H.
- **Scf_Mix**, $[0, 1]$. Współczynnik mieszania gęstości elektronowych w procedurze SCF, patrz sekcja III.
- **Out_RhoNode**, \mathbb{Z}^+ . Liczba punktów pomiędzy węzłami gęstości elektronowej. W tych węzłach obliczana jest gęstość elektronowa i zapisywana do pliku. Do pliku zapisywana jest gęstość elektronowa z równania (13).

- **Out_EigNode**, \mathbb{Z}^+ . Liczba punktów pomiędzy węzłami funkcji własnej. W tych węzłach obliczana jest funkcja własna i zapisywana do pliku. Całkowita liczba punktów w pliku może być różna dla każdej pobocznej liczby kwantowej ℓ , gdyż dla każdego ℓ jest inny solver adaptacyjny. Do pliku zapisywane są funkcje $R_{n,\ell}$ z równania (2).

C. Opis parametrów wejściowych

W tej sekcji podany jest opis parametrów wejściowych z uwzględnieniem ich znaczenia fizycznego i numerycznego.

1. Definiowanie atomu

Program **RAtom** potrafi rozwiązywać równanie Kohna-Shama dla atomu neutralnego i naładowanego (jonu dodatniego i ujemnego). Układ definiowany jest przez liczbę elektronów (parametr **Atom_Electron**) i liczbę protonów (parametr **Atom_Proton**). Atom jest neutralny, gdy oba parametry są sobie równe. Aby zdefiniować jon dodatni wartość parametru **Atom_Electron** musi być mniejsza od parametru **Atom_Proton**. Aby zdefiniować jon ujemny wartość parametru **Atom_Electron** musi być większa od parametru **Atom_Proton**.

Domyślnie tylko stany obsadzone są obliczane. Aby obliczyć stany nieobsadzone należy nadać wartości większe od zera parametrom **Atom_ExtraShell** lub **Atom_ExtraAngular**. Parametr **Atom_ExtraShell** definiuje liczbę obliczanych nieobsadzonych stanów dla każdej pobocznej liczby kwantowej ℓ . Na przykład dla atomu węgla standardowo obliczane są stany: $1s_{\ell=0,n=0}$, $2s_{\ell=0,n=1}$, $2p_{\ell=1,n=0}$. Gdy mamy **Atom_ExtraShell** = 1, to dla atomu węgla obliczane są obsadzone stany $1s$, $2s$, $2p$ oraz dodatkowo dwa nieobsadzone: $3s_{\ell=0,n=2}$, $3p_{\ell=1,n=1}$.

Aby policzyć stany dla wyższych pobocznych liczb kwantowych ℓ należy ustalić wartość parametru **Atom_ExtraAngular** na większą od zera. Na przykład gdy **Atom_ExtraAngular** = 1, to dla atomu węgla obliczane są trzy obsadzone stany $1s$, $2s$, $2p$ oraz dodatkowo $3d_{\ell=2,n=0}$ jeden nieobsadzony stan.

Przypisując wartości większe od zera parametrowi **Atom_ExtraShell** oraz **Atom_ExtraAngular** możemy elastycznie określać liczbę obliczanych stanów nieobsadzonych. Ta funkcjonalność programu wykorzystywana jest podczas generowania funkcji bazowych dla aplikacji rozwiązyjących równanie Kohna-Shama dla molekuly lub klastra atomów metodą kombinacji liniowej orbitali atomowych (DFT-LCAO).

Trzeba pamiętać, że obliczenia dla nieobsadzonych (wzbudzonych) stanów własnych mogą dawać dodatnie wartości własne. Jest to konsekwencja wykorzystania formalizmu DFT, który jest poprawny tylko dla stanu

podstawowego. Aby poprawnie opisywać stany wzbudzone należy zastosować formalizm TDFT (ang. *Time-Dependent Density Functional Theory*) opisujący zachowanie się układu w czasie (zachowanie stanów niestacjonarnych).

2. Przybliżenie energii korelacji i wymiany

Typ zastosowanego przybliżenia na energię korelacji określany jest parametrem **XC_Corr**. Ta wersja programu obsługuje tylko przybliżenie VWN i aby je użyć należy nadać wartość 'vwn' parametrowi **XC_Corr**. Parametr **XC_Exch** określa typ przybliżenia energii wymiany. Ta wersja programu obsługuje tylko przybliżenie podane przez Slatera i aby je użyć należy nadać wartość 'slater' parametrowi **XC_Exch**. Aby pominąć wkład energii korelacji lub wymiany odpowiedniemu parametrowi należy nadać wartość 'null'.

3. Początkowa gęstość elektronowa

Aby rozpocząć procedurę SCF należy wybrać początkową gęstość elektronową, tak jak to jest dyskutowane w sekcji II G 1. Gdy parametr **Rho0_Default** jest równy 'Yes', to początkowa gęstość elektronowa ustalana jest ze wzoru (32) i żadne inne dane nie muszą być podawane w pliku wejściowym.

Gdy parametr **Rho0_Default** jest równy 'No', to początkowa gęstość elektronowa ustalana jest ze wzoru (33) i wtedy parametry **Rho0_c**, **Rho0_Alpha** muszą być zdefiniowane. W takiej sytuacji, dodatkowo, parametry **Rho0_c**, **Rho0_Alpha** muszą być tak dobrane, aby po wycałkowaniu gęstość elektronowa równała się liczbie elektronów.

4. Solver

Liniowe zagadnienie własne oraz równanie Poissona jest rozwiązywane za pomocą metody elementu skończonego z zastosowaniem funkcji bazowych Lobatto. Algorytmy te zostały szczegółowo opisane w dokumentacji do biblioteki **RFem1D**, która jest wykorzystywana w programie **RAtom**. Tutaj podane jest tylko znaczenie wykorzystywanych parametrów biblioteki **RFem1D**.

Są dwie dostępne metody rozwiązywania zagadnienia własnego oraz równania Poissona: adaptacyjna i nieadaptacyjna. Aby zastosować metodę adaptacyjną dla zagadnienia własnego należy parametrowi **Solver_EigAdapt** przypisać wartość 'Yes'. Aby zastosować metodę adaptacyjną dla równania Poissona należy parametrowi **Solver_PsnAdapt** przypisać wartość 'Yes'. Aby zastosować metodę nieadaptacyjną należy tym parametrom przypisać wartość 'No'.

Dla algorytmu nieadaptacyjnego, liczbę węzłów w siatce wykorzystywanej do rozwiązywania zagadnienie

własnego definiuje parametr `Solver_EigNode`. Jeżeli stosowany jest algorytm adaptacyjny do rozwiązywania zagadnienia własnego, to parametr `Solver_EigNode` określa początkową liczbę węzłów w siatce.

Dla algorytmu nieadaptacyjnego, liczbę węzłów w siatce wykorzystywanej do rozwiązywania równania Poissona definiuje parametr `Solver_PsnNode`. Jeżeli stosowany jest algorytm adaptacyjny do rozwiązywania równania Poissona, to parametr `Solver_PsnNode` określa początkową liczbę węzłów w siatce.

Parametry `Solver_EigDeg`, `Solver_PsnDeg` definiują stopień zastosowanych wielomianów Lobatto podczas rozwiązywania zagadnienia własnego i równania Poissona, odpowiednio. W tej wersji programu `RAtom` zostały zaimplementowane wielomiany Lobatto do stopnia 10 włącznie. Wartość tych parametrów musi być większa niż 2. Sugeruje się aby dla obu problemów korzystać z wielomianów Lobatto stopnia 6.

Algorytm adaptacyjny dla zagadnienia własnego i dla równania Poissona dzieli elementy na połowy tak długo, aż największy z najmniejszych współczynników na elemencie będzie mniejszy niż `Solver_EigAbsMaxCoef` i `Solver_PsnAbsMaxCoef`, odpowiednio. Zwykle, gdy te parametry są równe 10^{-2} , to otrzymuje się wartości własne z dokładnością do 2 lub 3 miejsc po przecinku.

Nażłenie zerowych warunków brzegowych na funkcje własne równania Kohna-Shama odpowiada umieszczeniu atomu w sferycznie symetrycznej studni potencjału o nieskończonej głębokości. Promień tej sferycznie symetrycznej studni określa parametr `Solver_Rc` oznaczany w tym artykule symbolem r_c . Formalnie, aby otrzymać atom swobodny należy przyjąć $r_c = \infty$. Jednakże, dla większości atomów wartość $r_c < 100$ jest wystarczająca, aby otrzymać wartości własne odpowiadające swobodnemu atomowi. Zmniejszając wartość tego parametru można otrzymać funkcje własne, które mogą tworzyć zlokalizowaną bazę do zastosowań w metodzie LCAO-DFT dla molekuł.

5. Aproksymacja gęstości elektronowej

Gęstość elektronowa jest aproksymowana adaptacyjnie wielomianami Lobatto. Stopień tych wielomianów określany jest parametrem `Rho_Deg`. Dokładność aproksymacji definiuje parametr `Rho_Delta`.

Trzeba pamiętać, że `Rho_Delta` określa kwadrat błędu, dlatego ta wartość powinna być dosyć mała, aby otrzymać dokładne wyniki. Ponieważ aproksymacja jest zaimplementowana efektywnie, to zwykle przyjmuje się wartość 10^{-14} dla `Rho_Delta`.

6. Procedura SCF

Radialne, nieliniowe równanie Kohna-Shama rozwiązywane jest iteracyjnie metodą samouzgodnienia (SCF)

nazywane też w matematyce metoda punktu stałego. Parametr `Scf_MaxIter` określa maksymalną liczbę iteracji SCF. Gdy liczba iteracji SCF przekroczy wartość tego parametru, to program kończy działanie zapisując (tak jak zwykle) obliczone wielkości do plików wyjściowych. Wartości i funkcje własne otrzymane bez osiągnięcia zbieżności mogą być obciążone dużym błędem. Wyciąganie wniosków fizyko-chemicznych na podstawie takich danych obciążone jest wysokim ryzykiem i nie jest rekomendowane.

Procedura SCF uaktualnia iteracyjnie gęstość elektronową atomu ρ . Jednakże, z powodu małego promienia zbieżności, gęstość elektronowa wykorzystana w kolejnym kroku procedury SCF nie jest gęstością elektronową obliczoną w obecnym kroku SCF. Zamiast tego, w kolejnym kroku SCF wykorzystuje się kombinację liniową obecnie obliczonej gęstości elektronowej, ρ_{cur} , oraz gęstości elektronowej z poprzedniego kroku SCF, ρ_{old} . To znaczy mamy

$$\rho(r) = \alpha \rho_{\text{cur}}(r) + (1 - \alpha) \rho_{\text{old}}(r) \quad (43)$$

Gdy $\alpha = 1$ to gęstość elektronowa ρ_{old} nie jest uwzględniana. Gdy $\alpha = 0$ to gęstość elektronowa ρ_{cur} nie jest uwzględniana. W typowych przypadkach $\alpha = 0.5$ jest wystarczające do otrzymania zbieżności. Współczynnik miksowania gęstości elektronowych, $0 \leq \alpha \leq 1$, określony jest parametrem `Scf_Mix`.

Procedura SCF kończy się sukcesem, spełniony jest warunek stopu. Warunek stopu jest spełniony różnicą sum wartości własnych w dwóch kolejnych krokach SCF jest mniejsza niż `Scf_Diff`. Wartość tego parametru podawana jest w hartree.

7. Zapisywanie wyników

Otrzymana gęstość elektronowa zapisana jest do pliku. W pliku zapisane są wierszami pary $(r_i, \rho(r_i))$, gdzie sposób i liczba generowanych r_i opisana jest poniżej.

Liczba par $(r_i, \rho(r_i))$ zdefiniowana jest przez dwa parametry. Pierwszym parametrem jest liczba węzłów wygenerowana przez adaptacyjny algorytm aproksymujący. Liczbę węzłów wygenerowaną przez ten algorytm pośrednio determinowana jest przez parametr `Rho_Delta`. Drugim parametrem wpływającym na liczbę par jest `Out_RhoNode`, który określa ile jest utworzonych węzłów w równej odległości pomiędzy węzłami z procedury aproksymacyjnej. Analogiczne znaczenie ma parametr `Out_EigNode` tylko dotyczy funkcji własnych.

8. Przykładowe wartości parametrów

W tabeli III podane są przykładowe wartości parametrów wejściowych. Przykłady dotyczą atomu azotu i atomu indu. Podane wartości umożliwiają otrzymanie wartości własnych i energii całkowitej z dokładnością

do 6 miejsc po przecinku, co można porównać z danymi referencyjnymi^{22,23}.

Tablica III: Parametry pliku wejściowego dla atomu N i In.

Parametr	N	In
Atom_Proton	7	49
Atom_Electron	7	49
Atom_ExtraShell	0	0
Atom_ExtraAngular	0	0
Atom_Rc	20	30
XC_Exch	slater	slater
XC_Corr	vwn	vwn
Rho0_Default	No	No
Rho0_c	15.283	3368.9916
Rho0_Alpha	3.8	12.0
Solver_EigAdapt	Yes	Yes
Solver_EigNode	4	4
Solver_EigDeg	6	6
Solver_EigAbsMaxCoef	1E-4	1E-4
Solver_EigAbsTol	1E-9	1E-9
Solver_PsnAdapt	Yes	Yes
Solver_PsnNode	4	4
Solver_PsnDeg	6	6
Solver_PsnAbsMaxCoef	1E-4	1E-4
Rho_Deg	7	7
Rho_Delta	1E-14	1E-14
Scf_MaxIter	40	40
Scf_Mix	0.5	0.5
Scf_Diff	1e-7	1e-7
Out_RhoNode	10	10
Out_EigNode	10	10

D. Plik wynikowy .out

Pierwszą informacją w pliku .out są zinterpretowane wartości parametrów przeczytane z pliku .inp. Jest to bardzo przydatne, gdyż pozwala uniknąć wielu błędów spowodowanych nieprawidłowym podaniem wartości parametrów w pliku wejściowym.

Drugą informacją zawartą w pliku .out są obliczone wartości własne dla każdego analizowanego stanu elektronowego (n, ℓ) . Aby zwiększyć czytelność dla każdej pary (n, ℓ) podana jest nazwa stanu oraz jej obsadzenie elektronami. W kolejnej części pliku podana jest energia całkowita i jej składowe. Wszystkie podane wielkości obliczone są po wykonaniu ostatniej iteracji w procedurze SCF.

Format tego pliku nie jest ustalony i w przyszłości może ulec zmianie.

E. Pliki wynikowe .nto, .L=?

Obliczone funkcje własne $R_{n,\ell}$ zapisane są w plikach z rozszerzeniem .nto. Każda funkcja własna $R_{n,\ell}$ zapisywana jest w oddzielnym pliku. Nazwa tego pliku po-

chodzi od nazwy stanu elektronowego reprezentowanego przez funkcję $R_{n,\ell}$.

Każdy plik .nto jest plikiem tekstowym. W każdej linii pliku znajduje się para liczb rzeczywistych $(r_i, R_{n,\ell}(r_i))$. Współrzędne $\{r_i\}$ ustalane są przez procedurę adaptacyjną i przez parametr Out_EigNode. Dokładniejszy opis znaczenia tych parametrów znajduje się w sekcji III C 7.

Plik .nto idealnie nadaje się do tworzenia wykresów funkcji $R_{n,\ell} : [0, r_c] \mapsto \mathbb{R}$, gdyż węzły $\{r_i\}$ rozmieszczone są tak, aby odzwierciedlały zachowanie się funkcji $R_{n,\ell}$. Dodatkowo, na podstawie wartości $(r_i, R_{n,\ell}(r_i))$ i interpolacji funkcjami sklejanymi³⁰ (ang. *cubic spline*), można odtworzyć funkcję $R_{n,\ell}(r)$ i wykorzystać ją w innym programie.

Gdy potrzebna jest bardzo duża dokładność wyznaczonych funkcji własnych $R_{n,\ell}$, to do ich odtworzenia można wykorzystać współczynniki wielomianów Lobatto zapisanych w pliku .L=?. Zwykle utworzone są trzy pliki .L=0, .L=1, .L=2 dla $\ell = 0, 1, 2$, odpowiednio. Każdy z tych plików zawiera współrzędne przedziału otrzymanego w procedurze adaptacyjnej i wartości współczynników wielomianów Lobatto.

F. Pliki wynikowe .rho, .app

Radialna gęstość elektronowa ϱ i gęstość wycalkowana po sferze $\tilde{\varrho}$ zapisana jest do pliku o rozszerzeniu .rho. Jest to plik tekstowy. W każdej linii pliku znajduje się trójka liczb rzeczywistych $(r_i, \varrho(r_i), \tilde{\varrho}(r_i))$. Współrzędne $\{r_i\}$ ustalane są przez procedurę adaptacyjną i przez parametr Out_RhoNode. Dokładniejszy opis znaczenia tych parametrów znajduje się w sekcji III C 7.

Plik .rho idealnie nadaje się do tworzenia wykresów funkcji $\varrho, \tilde{\varrho} : [0, r_c] \mapsto \mathbb{R}$, gdyż węzły $\{r_i\}$ rozmieszczone są tak, aby odzwierciedlały zachowanie się funkcji $\varrho, \tilde{\varrho}$. Dodatkowo, na podstawie wartości $(r_i, \varrho(r_i))$ oraz $(r_i, \tilde{\varrho}(r_i))$ i interpolacji funkcjami sklejanymi³⁰, można odtworzyć funkcje $\varrho, \tilde{\varrho}$ i wykorzystać ją w innym programie.

Gdy potrzebna jest bardzo duża dokładność wyznaczonej gęstości elektronowej ϱ , to do jej odtworzenia można wykorzystać współczynniki wielomianów Lobatto zapisanych w pliku .app. Dla każdego przedziału otrzymanego w procedurze adaptacyjnej aproksymacji podane są odpowiadające współczynniki wielomianów Lobatto.

IV. PRZYKŁADY ZASTOSOWAŃ

W tej sekcji zostana podane wyniki otrzymane za pomocą programu RAtom. Z powodu zainteresowań autora rozpatrzone zostały atom N, Al, Ga, In. Zastosowane parametry dla atomu N oraz In są podane w tabeli III. Natomiast w tabeli IV zostały podane parametry dla Al i Ga.

Tablica IV: Parametry pliku wejściowego dla atomu Al i Ga.

Parametr	Al	Ga
Atom_Proton	13	31
Atom_Electron	13	31
Atom_ExtraShell	0	0
Atom_ExtraAngular	0	0
Atom_Rc	30	30
XC_Exch	slater	slater
XC_Corr	vwn	vwn
Rho0_Default	No	No
Rho0_c	86.0581	423.074
Rho0_Alpha	5.5	7.0
Solver_EigAdapt	Yes	Yes
Solver_EigNode	4	4
Solver_EigDeg	6	6
Solver_EigAbsMaxCoef	1E-4	1E-4
Solver_EigAbsTol	1E-9	1E-9
Solver_PsnAdapt	Yes	Yes
Solver_PsnNode	4	4
Solver_PsnDeg	6	6
Solver_PsnAbsMaxCoef	1E-4	1E-4
Rho_Deg	7	7
Rho_Delta	1E-14	1E-14
Scf_MaxIter	40	40
Scf_Mix	0.5	0.5
Scf_Diff	1e-7	1e-7
Out_RhoNode	10	10
Out_EigNode	10	10

Tablica V: Energia całkowita E_t , kinetyczna E_k , wzajemnego oddziaływania elektronów E_h , oddziaływania z jądrem atomowym E_n i korelacji-wymiany E_{xc} dla atomu N, Al, Ga, In. Parametry wejściowe podane są w tabeli III i tabeli IV.

	N	Al	Ga	In
E_t	-54.025016	-241.315573	-1921.846438	-5737.309049
E_k	53.731738	240.663490	1919.894749	5733.944110
E_h	25.799846	112.670732	833.754949	2334.432426
E_n	-127.416240	-577.205759	-4603.576746	-13654.870604
E_{xc}	-6.140359	-17.444038	-71.919391	-150.814982

Otrzymana energia całkowita i jej składowe podane są w tabeli V. Wartości te można porównać z danymi referencyjnymi^{22,23}. Z otrzymanych wartości wynika, że największy wkład do energii całkowitej E_t ma energia oddziaływania z jądrem atomowym E_n , która co do wartości jest około dwa i pół raza większa od energii a całkowitej. Ponadto, energia kinetyczna E_k ma bardzo zbliżoną wartość (z dokładnością do znaku) do energii całkowitej. Natomiast, najmniejszy wkład wnosi energia korelacji-wymiany E_{xc} .

Zbieżność otrzymano po wykonaniu około 30 iteracji SCF. Porównanie z danymi referencyjnymi pozwala wywnioskować, że parametry wejściowe podane w tabeli III i tabeli IV są wystarczające do uzyskania energii całkowitej z dokładnością do 4 lub 5 cyfr po przecinku. Dokładność taka jest zwykle wystarczająca podczas badania układów molekularnych w chemii kwantowej.

Dla każdego przypadku czas obliczeń na komputerze Pentium IV z procesorem 1.9 GHz jest mniejszy niż 1 minuta. Potrzebna pamięć RAM jest mniejsza niż 5 MB.

* Electronic address: romz@wp.pl

¹ Babuska, I. and T. Strouboulis: 2001, *The Finite Element Method and Its Reliability*. Oxford: Oxford University Press.

² Beck, T. L.: 2000, 'Real-space mesh techniques in density-functional theory'. *Rev. Mod. Phys.* **72**, 1041–1080.

³ Brenner, S. C. and L. R. Scott: 1996, *The Mathematical Theory of Finite Element Methods*. Springer.

⁴ Byron, F. W. and R. W. Fuller: 1992, *Mathematics of Classical and Quantum Physics*. New York: Dover.

⁵ Ciarlet, P. G.: 1978, *The Finite Element Method for Elliptic Problems*. Holland: North-Holland Publishing Company.

⁶ de Boor, C.: 1978, *A Practical Guide to Splines*. New York: Springer-Verlag.

⁷ de Boor, C. and B. Swartz: 1977, 'Comments on the com-

parison of global methods for linear two-point boundary value problems'. *Math. Comput.* **31**, 916–921.

⁸ de Boor, C. and B. Swartz: 1980, 'Collocation approximation to eigenvalues of an ordinary differential equations: The principle of the thing'. *Math. Comput.* **35**, 679–694.

⁹ de Boor, C. and B. Swartz: 1981, 'Collocation approximation to eigenvalues of an ordinary differential equations: Numerical illustrations'. *Math. Comput.* **36**, 1–19.

¹⁰ Delley, B.: 1990, 'An all-electron numerical method for solving the local density functional for polyatomic molecules'. *J. Chem. Phys.* **92**, 508–517.

¹¹ Dreizler, R. and E. Gross: 1995, *Density Functional Theory*. New York: Plenum Press.

¹² Eschrig, H.: 1996, *The Fundamentals of Density Functional Theory*. Stuttgart: Teubner.

¹³ Fletcher, C. A. J.: 1984, *Computational Galerkin Methods*.

- Berlin: Springer.
- ¹⁴ Goedecker, S.: 1999, 'Linear scaling electronic structure methods'. *Rev. Mod. Phys.* **71**, 1085–1123.
 - ¹⁵ Handy, N. C.: 1994, 'Density Functional Theory'. In: B. O. Roos (ed.): *Lecture Notes in Quantum Chemistry II*. Heidelberg.
 - ¹⁶ Hohenberg, P. and W. Kohn: 1964, 'Inhomogeneous Electron Gas'. *Phys. Rev.* **136**, B864–B871.
 - ¹⁷ Johnson, C.: 1987, *Numerical solution of partial differential equations by the finite element method*. Cambridge: Cambridge University Press.
 - ¹⁸ Jones, R. O. and O. Gunnarsson: 1989, 'The density functional formalism, its applications and prospects'. *Rev. Mod. Phys.* **61**, 689–746.
 - ¹⁹ Koch, W. and M. C. Holthausen: 2000, *A Chemist's Guide to Density Functional Theory*. New York: Wiley.
 - ²⁰ Kohn, W., A. D. Becke, and R. G. Parr: 1996, 'Density Functional Theory of Electronic Structure'. *J. Chem. Phys.* **100**, 12974–12980.
 - ²¹ Kohn, W. and L. J. Sham: 1965, 'Self-Consistent Equations Including Exchange and Correlation Effects'. *Phys. Rev.* **140**, A1133–A1138.
 - ²² Kotochigova, S., Z. H. Levine, E. L. Shirley, M. D. Stiles, and C. W. Clark: 1997, 'Local-density-functional calculations of the energy of atoms'. *Phys. Rev. A* **55**, 191. (Erratum: *Phys. Rev. A* **56**, 5191 (1997)).
 - ²³ Kotochigova, S., Z. H. Levine, E. L. Shirley, M. D. Stiles, and C. W. Clark: 2007, '<http://physics.nist.gov/PhysRefData/DFTdata/contents.html>'.²⁴
 - ²⁴ Liboff, R. L.: 1987, *Introductory Quantum Mechanics*. New York: Addison Wesley.
 - ²⁵ March, N. H.: 1992, *Electron Density Theory of Atoms and Molecules*. London: Academic.
 - ²⁶ Martin, R. M.: 2004, *Electronic Structure: Basic Theory and Practical Methods*. Cambridge: Cambridge University Press.
 - ²⁷ McOwen, R.: 2002, *Partial Differential Equations: Methods and Applications*. New Jersey: Prentice Hall.
 - ²⁸ Parr, R. G. and W. Yang: 1989, *Density-Functional Theory of Atoms and Molecules*. Oxford: Oxford University Press.
 - ²⁹ Prenter, P. M.: 1989, *Splines and variational methods*. New York: Wiley.
 - ³⁰ Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: 1990, *Numerical Recipes in C*. Cambridge: Cambridge University Press.
 - ³¹ Romanowski, Z.: 2007, 'Numerical solution of Kohn-Sham equation for atom'. *Acta Phys. Pol. B* **38**, 3263–3285.
 - ³² Romanowski, Z.: 2008, 'B-Spline finite element solution of Kohn-Sham equation for atom'. *Modelling Simul. Mater. Sci. Eng.* **16**, 015003.
 - ³³ Schumaker, L. L.: 1981, *Spline Functions: Basic Theory*. New York: John Wiley & Sons.
 - ³⁴ Šolín, P.: 2006, *Partial Differential Equations and the Finite Element Method*. Hoboken: Wiley Interscience.
 - ³⁵ Šolín, P., K. Segeth, and I. Doležal: 2004, *High-Order Finite Element Method*. London: CHAPMAN & HALL/CRC.
 - ³⁶ Wait, R. and A. R. Mitchell: 1986, *Finite Element Analysis and Applications*. New York: Wiley.
 - ³⁷ Wang, Y. A. and E. A. Carter: 2000, *Progress in Theoretical Chemistry and Physics*, Chapt. Orbital-free kinetic-energy density functional theory, pp. 117–184. Dordrecht: Kluwer.