

Metody ab initio w fizyce ciała stałego

Laboratorium 1b

5 listopad 2025

1 Cel

- Zapoznanie się z obsługą WIEN2k za pomocą terminala.
- Wyznaczenia optymalnej stałej sieci Al w strukturze fcc.
- Wyznaczenia optymalnej stałej sieci Al w strukturze bcc i potwierdzenie, że nie jest ona preferowana w porównaniu z fcc.

1.1 Stworzenie nowego katalogu

WIEN2k wykonuje obliczenia w katalogu, w którym wywołamy komendy. Obowiązuje konwencja nazw plików, w której pierwszy człon jest taki sam jak nazwa katalogu, a po kropce znajduje się drugi człon charakteryzujący plik.

Najpierw przejdźmy do katalogu `~/WIEN2k/Al` i utwórzmy nowe katalogi na dzisiejsze zajęcia, na przykład fcc i bcc.

1.2 Zdefiniowanie struktury

Skopiujmy wcześniej pobrany plik `Al.cif` do utworzonego przed chwilą katalogu fcc i przejdźmy do niego. Program `cif2struct`

```
cif2struct Al.cif
```

wstępnie przygotuje plik `fcc.struct`. Jeżeli plik `Al.cif` zawierał informacje o symetriach struktury, to istnieje szansa, że powstały `fcc.struct` już będzie gotowy do inicjalizacji. Niemniej zawsze warto skorzystać z programu `sgroup`, który stworzy nowy plik `.struct` ze znalezionymi operacjami symetrii. Na przykład stwórzmy plik `Al_sg.struct` wywołując

```
sgroup fcc.struct -wi -wo fcc_sg.struct
```

i porównajmy jego zawartość z poprzednim `fcc.struct`. Skopiujmy plik `fcc_sg.struct` w miejsce poprzedniego pliku `fcc.struct` bez operacji symetrii. Zobaczmy jeszcze jak wygląda uzyskana struktura korzystając z `XCrysDena`

```
xcrysden --wien_struct fcc.struct
```

Screenshot struktury z widocznymi nazwami atomów do sprawozdania.

Alternatywną opcją tworzenia pliku `.struct` jest wykorzystanie programu

```
makestruct_lapw
```

który będzie po kolei prosił o podawanie informacji o strukturze.

1.3 Inicjalizacja obliczeń

Do inicjalizacji służy program

```
init_lapw
```

który będzie po kolei wywoływać programy, wyświetlać outputy do podglądu i od czasu do czasu pytać o podanie parametrów. Przyjmijmy potencjał wymiennie korelacyjny GGA i siatkę punktów k 10^3 .

1.4 Cykl samouzgodniony

Uruchamiamy go za pomocą programu run_lapw

```
nohup run_lapw &
```

Dzięki funkcji nohup program nie przerwie się nawet jak wyjdziemy z terminala czy nas coś rozłączy a ampersant & puści program w tle i zwolni terminal do dalszej pracy. Output z wywołanego run_lapw jest zapisywany w nohup.out, ale można też przekierować go do wybranego przez siebie pliku, na przykład do aout

```
nohup run_lapw > aout &
```

Cykl zakończy się, jeżeli wielkości ETEST (energia całkowita) i CTEST (gęstość ładunku) w kolejnych iteracjach będą się różnić o mniej niż pewne ustalone wartości. Próg zbieżności energii można zmienić korzystając z parametru -ec, a próg zbieżności gęstości przez -cc.

Po skończonym cyklu SCF zapiszmy obliczenia za pomocą programu save_lapw, na przykład w katalogu o nazwie r1

```
save_lapw -d r1
```

1.5 Optymalizacja struktury

Do optymalizacji użyjemy programu

```
x optimize
```

w którym zadamy zmianę objętości (opcja 1) w procentach (na przykład -8 -4 0 4 8, liczby podawać pojedynczo i enter po każdej). Dzięki temu powstaje skrypt optimize.job. Można w nim zmodyfikować wiersz z run_lapw. Następnie uruchamiamy ten skrypt za pomocą

```
nohup ./optimize.job &
```

i w pliku nohup.out możemy podglądać jak idą obliczenia.

Następnie musimy wyekstraktować zależność energii od objętości. W tym celu używamy:

```
grep line :ENE '*.scf' 1 > Al.analysis  
grep line :VOL '*.scf' 1 >> Al.analysis
```

(uwaga: apostrofy powinny być proste). W pliku Al.analysis mamy teraz serię energii i serię objętości.

Następnie analizujemy te wyniki wpisując w programie

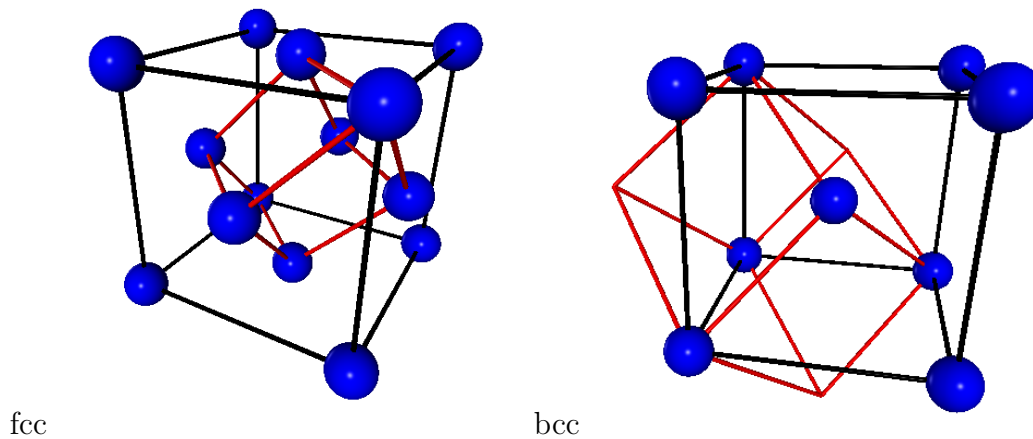
```
eplot_lapw
```

To odczytuje plik Al.analysis, dopasowuje funkcję Birch-Murnaghana i wyrzuca parametry tej funkcji i tabelkę taką jak poprzednio oraz robi stosowny rysunek.

Aby zrobić samemu taki rysunek, trzeba przekopiować tabelkę z outputu do nowego pliku (na przykład e_vs_vol.dat) i wyrysować funkcję $E(a)$, gdzie E to energia, a - stała sieci ($a = \sqrt[3]{4V}$). Ponadto w tamtym pliku nad tabelą jest dopasowana funkcja Birch-Murnaghana $E = E_0 + \dots$ i wartości parametrów dopasowania (V_0, B, B'_0, E_0). Ma ono postać

$$E(V) = E_0 + \frac{9}{16} \left(\frac{BV_0}{14703.6} \right) \left\{ \left[\left(\frac{V_0}{V} \right)^{2/3} - 1 \right]^3 B'_0 + \left[\left(\frac{V_0}{V} \right)^{2/3} - 1 \right]^2 \left[6 - 4 \left(\frac{V_0}{V} \right)^{2/3} \right] \right\}, \quad (1)$$

gdzie $a_0 = (4V_0)^{1/3}$ dla fcc i $a_0 = (2V_0)^{1/3}$ dla bcc. Na wyrysowane wyżej punkty $E(a)$ nanieśmy również tę funkcję $E_{fit}(a)$.



Rysunek 1: Komórki elementarne i prymitywne fcc oraz bcc.

2 bcc Al

Przejdźmy do katalogu `~/WIEN2K/Al/bcc`. Kopiujemy plik `*.struct` z poprzednich obliczeń, w którym jest zdefiniowana struktura *fcc* Al.

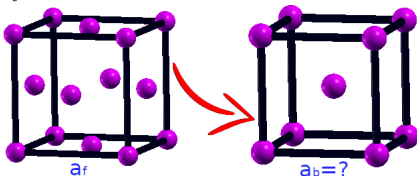
Tym razem chcemy użyć fikcyjnej struktury *bcc*, więc musimy zmodyfikować plik `Al.struct`, tak, by była w nim taka struktura *bcc*, w której odległości międzyatomowe będą takie same jak w strukturze *fcc*. Przyjrzyjmy się więc tym strukturom i znajdziemy wzór, który pozwoli nas obliczyć taką stałą sieci *bcc*, by ten warunek był spełniony.

$$d = \frac{\sqrt{2}}{2} a_{fcc}$$

$$d = \sqrt{\left(\frac{a_{bcc}}{2}\right)^2 + \left(\frac{\sqrt{2}a_{bcc}}{2}\right)^2} = \frac{\sqrt{3}}{2} a_{bcc}$$

$$a_{bcc} = \sqrt{\frac{2}{3}} a_{fcc}$$

$$a_{fcc} = 7.632472 a_B, \text{ więc } a_{bcc} = 6.231887 a_B$$



Następnie skopiujemy plik `Al.struct` do `Al.bcc.struct` i zmieniamy typ i stałe sieci na nowe ([pod tym linkiem](#) jest opis wszystkich parametrów w pliku `*.struct`).

Następnie otwieramy plik `Al.struct` w `xcrysden` i skorzystamy z jego możliwości obliczania odległości międzyatomowych. Klikamy "Distance" na dole okna → wybieramy dwa atomy będące sąsiadami → "Done" → czytamy wynik w oknie. Następnie z plikiem `Al.bcc.struct` robimy to samo i porównujemy, czy sprawdzone dystanse są takie same.

Następnie musimy znaleźć symetrie nowej struktury. W tym celu wpisujemy `sgroup Al-bcc.struct -wi -wo Al-bcc_sg.struct`. Wynikiem jest plik `Al-bcc_sg.struct`, porównajmy go z `Al_bcc.struct`.

Nowy plik jest gotowy do użycia, kopiujemy go tak by człon nazwy pliku `struct` był taki sama jak nazwa katalogu.

Analogicznie jak dla struktury `fcc` w poprzednim punkcie, wykonajmy inicjalizację, cykl `scf` i optymalizację objętości struktury `bcc`.

3 Sprawozdanie

Sprawozdanie w formacie pdf proszę przesyłać na teamsach.

W sprawozdaniu należy umieścić:

1. Krótki wstęp, w którym będzie cel wykonanych obliczeń oraz rysunek obydwu struktur z XCrysDena.
2. Sekcję informację o metodach, w której będzie zawarte:
 - użyte oprogramowanie do obliczeń DFT
 - opis metody obliczeniowej, która jest zaimplementowana w wykorzystanym kodzie DFT (full potential LAPW, rozwijanie funkcji falowej w dwóch bazach, orbitali i fal płaskich
 - użyte parametry do obliczeń (potencjał wymiennie-korelacyjny, promień RMT, rozmiar siatki punktów \mathbf{k}).
3. Sekcję wyniki, w której będzie zawarte:
 - postać równania stanu Bircha-Murnaghana,
 - rysunek z obliczonymi wartościami $E(V)$ i dopasowaną funkcją, na tym rysunku porównać wyniki `fcc` i `bcc`
 - podać wyznaczone stałe sieci w obu przypadkach i uzasadnić, która struktura jest preferowana
4. Krótkie podsumowanie, w którym napisać co udało się zrobić i jaki jest wynik pracy.

Proszę opisywać osie wykresów i używać na nich czytelnych czcionek.