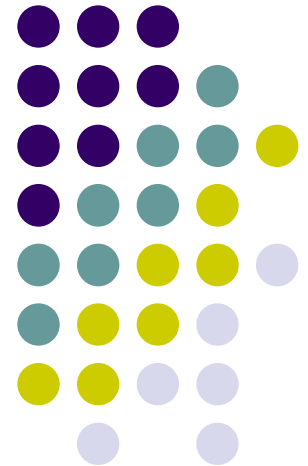


# Zastosowanie metod typu 'ab-initio' do badania właściwości materiałów



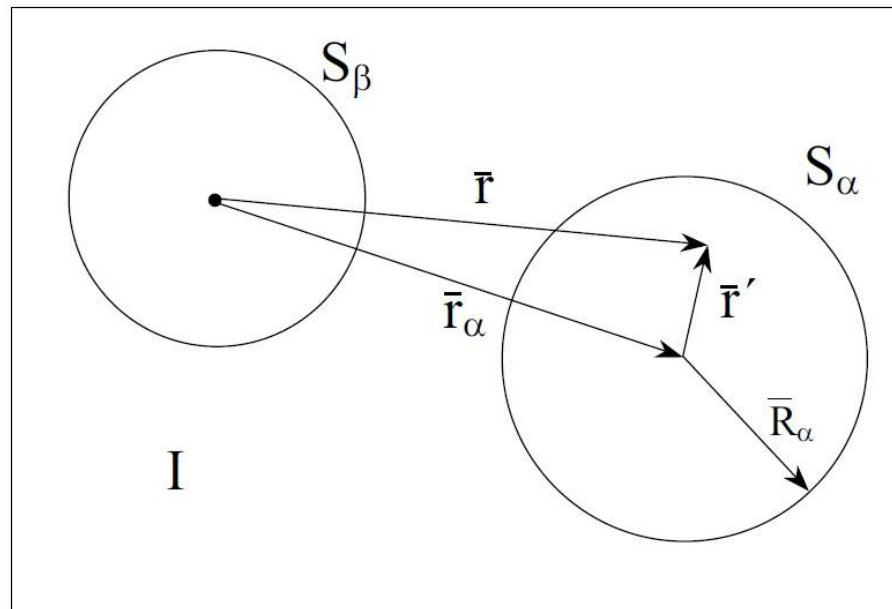
**WYKŁAD – Metody (L)APW,  
kod WIEN2k**

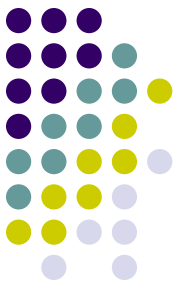




# Metody typu „atomic sphere”

- Dzielią przestrzeń na obszar (sferę) wokół atomu („muffin tin region”) i pozostałą część („interstitial region”)
- Baza funkcji falowej w obszarze sfery – funkcje typu funkcji atomu swobodnego
- Poza sferą – fale płaskie





# Dopasowane fale płaskie (APW)

- Funkcje bazy mają postać:

$$\phi_{\vec{K}}^{\vec{k}}(\vec{r}, E) = \begin{cases} \frac{1}{\sqrt{V}} e^{i(\vec{k} + \vec{K}) \cdot \vec{r}} & \vec{r} \in I \\ \sum_{\ell, m} A_{\ell m}^{\alpha, \vec{k} + \vec{K}} u_{\ell}^{\alpha}(r', E) Y_m^{\ell}(\hat{r}') & \vec{r} \in S_{\alpha} \end{cases}$$

- Główna wada – zależność od energii
- Współczynniki – z warunku ciągłości:

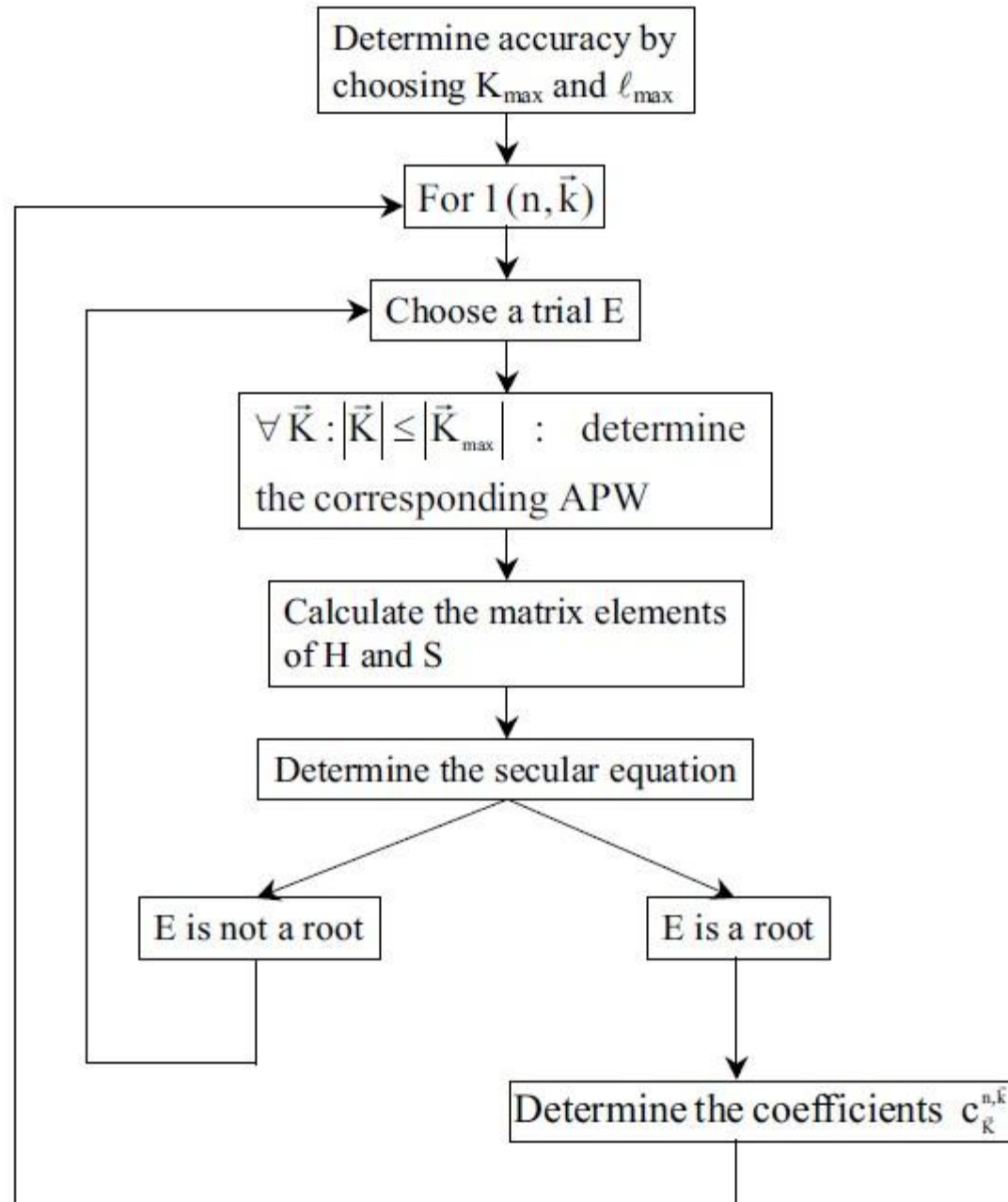
$$\frac{1}{\sqrt{V}} e^{i(\vec{k} + \vec{K}) \cdot \vec{r}} = \frac{4\pi}{\sqrt{V}} e^{i(\vec{k} + \vec{K}) \cdot \vec{r}_{\alpha}} \sum_{\ell, m} i^{\ell} j_{\ell}(|\vec{k} + \vec{K}| |\vec{r}'|) Y_m^{\ell*}(\hat{\vec{k} + \vec{K}}) Y_m^{\ell}(\hat{r}')$$

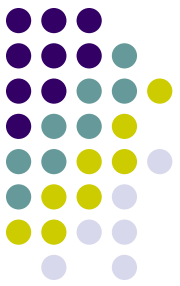
$$A_{\ell m}^{\alpha, \vec{k} + \vec{K}} = \frac{4\pi i^{\ell} e^{i(\vec{k} + \vec{K}) \cdot \vec{r}_{\alpha}}}{\sqrt{V} u_{\ell}^{\alpha}(\vec{R}_{\alpha}, E)} j_{\ell}(|\vec{k} + \vec{K}| R_{\alpha}) Y_m^{\ell*}(\hat{\vec{k} + \vec{K}})$$



# Dopasowane fale płaskie (APW)

- Najlepszy warunek obcięcia bazy:  $R_{\alpha} K_{max} = \ell_{max}$
- Zależność funkcji bazowych od energii wymaga rozpoczęcia obliczeń z pewną energią początkową („zgadniętą”)
- Algorytm działania:





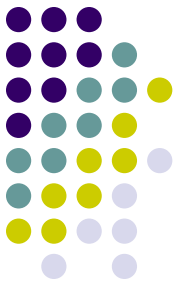
# Linearyzacja APW – LAPW

- Jak się pozbyć zależności od energii?:

$$u_{\ell}^{\alpha}(r', \epsilon_k^n) = u_{\ell}^{\alpha}(r', E_0) + (E_0 - \epsilon_k^n) \underbrace{\left. \frac{\partial u_{\ell}^{\alpha}(r', E)}{\partial E} \right|_{E=E_0}}_{\dot{u}_{\ell}^{\alpha}(r', E_0)} + O(E_0 - \epsilon_k^n)^2$$

- Funkcje bazy mają postać:

$$\phi_{\vec{K}}^{\vec{k}}(\vec{r}) = \begin{cases} \frac{1}{\sqrt{V}} e^{i(\vec{k} + \vec{K}) \cdot \vec{r}} & \vec{r} \in I \\ \sum_{\ell, m} \left( A_{\ell m}^{\alpha, \vec{k} + \vec{K}} u_{\ell}^{\alpha}(r', E_0) + B_{\ell m}^{\alpha, \vec{k} + \vec{K}} \dot{u}_{\ell}^{\alpha}(r', E_0) \right) Y_m^{\ell}(\hat{r}') & \vec{r} \in S_{\alpha} \end{cases}$$



# Linearyzacja APW – LAPW

- Czy dla wszystkich  $\ell$  tak sama zależność od energii?

$$\phi_{\vec{K}}^{\vec{k}}(\vec{r}) = \begin{cases} \frac{1}{\sqrt{V}} e^{i(\vec{k}+\vec{K})\cdot\vec{r}} & \vec{r} \in I \\ \sum_{\ell, m} \left( A_{\ell m}^{\alpha, \vec{k}+\vec{K}} u_{\ell}^{\alpha}(r', E_{1, \ell}^{\alpha}) + B_{\ell m}^{\alpha, \vec{k}+\vec{K}} \dot{u}_{\ell}^{\alpha}(r', E_{1, \ell}^{\alpha}) \right) Y_m^{\ell}(\hat{r}') & \vec{r} \in S_{\alpha} \end{cases}$$

- Czy jest sens włączać do obliczeń niskoleżące stany (np. 1s) ?
- Podział na stany „core”, „semi-core” i walencyjne – jak wybierać energie w funkcjach bazowych?
- Można to uwzględnić dodając do bazy dodatkowe funkcje – lokalne orbitale („local orbitals” - LO)



# LAPW + LO

- Dodaje się więc do bazy LAPW następujące funkcje bazowe:

$$\phi_{\alpha, LO}^{\ell m}(\vec{r}) = \begin{cases} 0 & \vec{r} \notin S_{\alpha} \\ \left( A_{\ell m}^{\alpha, LO} u_{\ell}^{\alpha}(r', E_{1, \ell}^{\alpha}) + B_{\ell m}^{\alpha, LO} \dot{u}_{\ell}^{\alpha}(r', E_{1, \ell}^{\alpha}) + C_{\ell m}^{\alpha, LO} u_{\ell}^{\alpha}(r', E_{2, \ell}^{\alpha}) \right) Y_m^{\ell}(\hat{r}') & \vec{r} \in S_{\alpha} \end{cases}$$

- Pozwala to dokładniej uwzględnić rozbieżności w energii stanów walencyjnych





## Inny sposób – APW + lo

- Można pozbyć się zależności od energii w APW inaczej: dodając do bazy APW lokalne orbitale ale zdefiniowane nieco inaczej niż w LAPW + LO (stąd „lo” w nazwie)

$$\phi_{\vec{K}}^{\vec{k}}(\vec{r}) = \begin{cases} \frac{1}{\sqrt{V}} e^{i(\vec{k} + \vec{K}) \cdot \vec{r}} & \vec{r} \in I \\ \sum_{\ell, m} A_{\ell m}^{\alpha, \vec{k} + \vec{K}} u_{\ell}^{\alpha}(r', E_{1, \ell}^{\alpha}) Y_m^{\ell}(\hat{r}') & \vec{r} \in S_{\alpha} \end{cases}$$

$$\phi_{\alpha, lo}^{\ell m}(\vec{r}) = \begin{cases} 0 & \vec{r} \notin S_{\alpha} \\ \left( A_{\ell m}^{\alpha, lo} u_{\ell}^{\alpha}(r', E_{1, \ell}^{\alpha}) + B_{\ell m}^{\alpha, lo} \dot{u}_{\ell}^{\alpha}(r', E_{1, \ell}^{\alpha}) \right) Y_m^{\ell}(\hat{r}') & \vec{r} \in S_{\alpha} \end{cases}$$



## „Mixed base” – LAPW/APW + $l_0$

- Program WIEN2k używa „mieszanej” bazy:  
LAPW/APW +  $l_0$
- Stany o wysokim  $\ell$  (np. d czy f) są rozwijane w bazie APW+ $l_0$
- Stany pozostałe – LAPW + LO
- Pozwala to łączyć zalety obu podejść



## WIEN2k – podstawowe informacje

- Realizuje metodę K-S, używa pełnego potencjału
- Baza funkcji LAPW/APW+lo
- Płatny – licencja akademicka ok. 400 Euro
- „Międzynarodowy standard”



## WIEN2k – możliwości

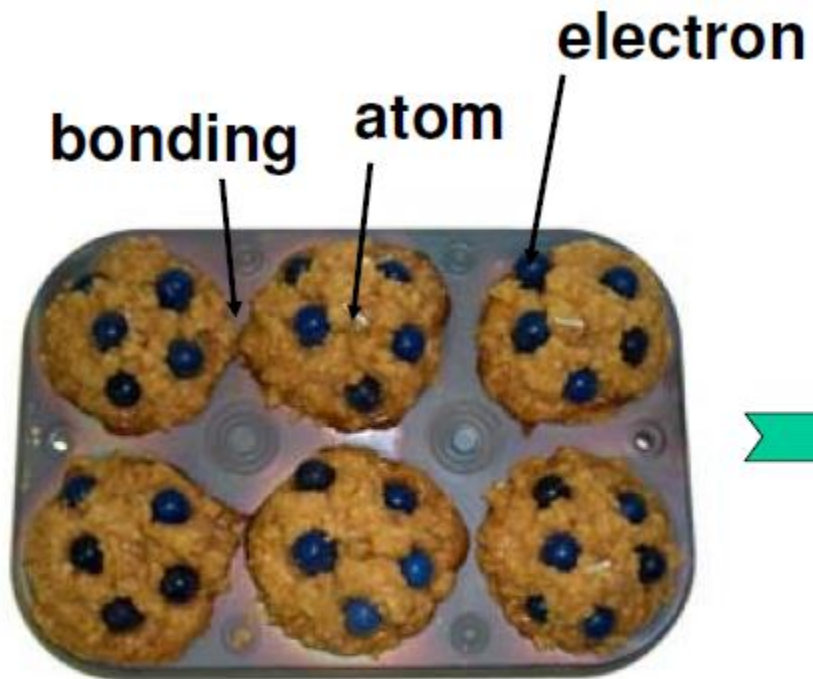
- Struktura elektronowa, gęstość stanów
- Gęstości – elektronowa i spinowa,
- Energia całkowita, siły (F-H), optymalizacja struktury
- Interface do programu liczącego widmo fononów (Phonon K. Parlińskiego)
- Układy spinowo spolaryzowane (ferro, antyferro, niekolinearny)
- Oddziaływanie spin-orbita
- Widmo emisji i absorpcji rentegowskiej
- Właściwości optyczne
- LDA, GGA, LDA+U
- Uwzględnia symetrię układu (komórki elementarnej)
- Interface webowy – generator struktur, uruchamianie zadań
- Wizualizacja – XCrysDen



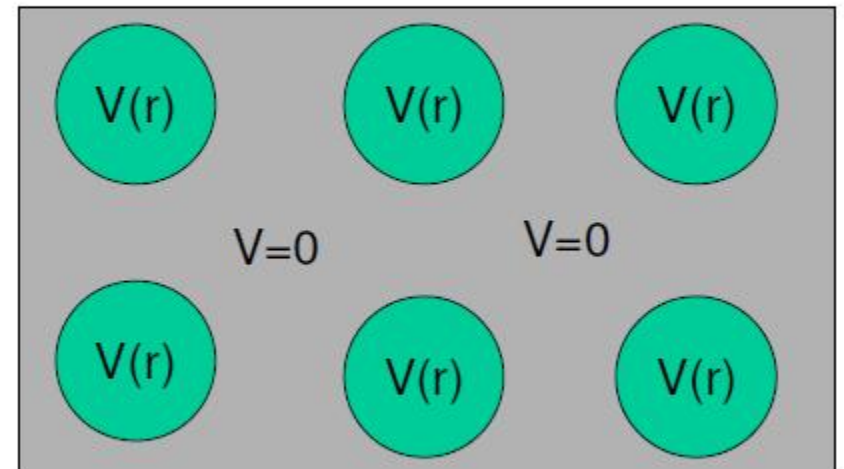
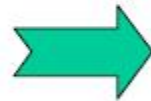
## WIEN2k – uwagi praktyczne

- Program składa się z szeregu podprogramów i skryptów
- Każdy podprogram ma oddzielny input, można nimi zarządzać przez interface webowy
- Główny parametr inputu – wielkość  $K_{max}$
- Główne skrypty – `init_lapw` i `run_lapw`
- Główne programy
  - `lapw0` – generuje potencjał
  - `lapw1` – wyznacza funkcje i wartości własne
  - `lapw2` – generuje gęstości elektronowe

# „Muffin-tin”



Every day life



Physical model