

講義資料: <http://conf.msl.titech.ac.jp/D2MatE/2024Tutorial/tutorial2024-DefectCalculation.html>

チュートリアル: 欠陥の第一原理計算

神谷利夫

東京工業大学
国際先駆研究機構
元素戦略MDX研究センター

謝辞: 東北大 熊谷教授 (pydefect)
東工大 大場教授
当研究室 He 特任助教、南島君、その他多くの学生

今日の目標

- 半導体の欠陥計算の問題
 - 欠陥濃度は合成条件 (化学ポテンシャル) と E_F に強く依存する
- どのように希薄欠陥の計算を行うか
- 実際の欠陥計算における補正
を理解する

- 「ドナー準位」や「アクセプター準位」はどのように扱われるか
電荷遷移準位
- 第一原理計算の結果を半導体統計に応用
キャリア濃度の E_F 依存性、温度依存性

参考文献

格子力学 (経験的原子間ポテンシャル)

- C.R.A. Catlow, K.M. Diller, M.J. Norgett, Phys. Rev. B **18**, 2739 (1978).
- C.R.A. Catlow, Phys. Rev. B, Phys. Rev. B **39**, 1897 (1989).

第一原理計算の総説

- C. G. Van deWalle and J. Neugebauer, J. Appl. Phys. 95, 3851 (2004).
- S. Lany and A. Zunger, Phys. Rev. B **78** (2008). 235104.
- S. Lany and A. Zunger, Model. Simul. Mater. Sci. Eng. **17** (2009) 084002.
- M. Bockstedte, A. Marini, O. Pankratov, and A. Rubio, Phys. Rev. Lett. 105, 026401 (2010).
- F. Oba, M. Choi, A. Togo, and I. Tanaka, Sci. Tech. Adv. Mater. 12, 034302 (2011)

Launcher.py/VASP: 欠陥計算後処理支援

http://conf.msl.titech.ac.jp/D2MatE/D2MatE_programs.html

- 4. FFT・平滑化^{ACD}
- 5. ピークサーチ^{ACD}
- 6. 緩和時間^{ACD}
- 半導体
 - 1. 基本半導体物性計算^{ACD}
 - 2. 移動度の温度依存性^{CD}
 - 3. キャリア濃度の温度依存性^{CD}
 - 4. 重み付き移動度^{ACD}
 - 5. Jonker_plot^{CD}
 - 6. 熱電シミュレーション^{CD}
- 光電子分光
 - 1. ピーク同定^{CD}
 - 2. デコンボリューション^{CD}
 - 3. 混合相同定支援^{CD}
- 結晶構造
 - 1. CIF管理ライブラリ
[tkCIF](#)^{ACD}
 - 2. 結晶構造管理ライブラリ
[tkCrystal](#)^{ACD}
 - 3. 結合長・結合角^{ACD}
 - 4. 動径分布関数^{ACDpl}
 - 5. 格子変換^{ACDpl}
 - 6. 対称構造^{ACD}
 - 7. BVS^{ACD}
 - 8. 粉末XRDシミュレーション^{ACD}
 - 9. New! 粉末XRDによる相同定支援^{ACD}
 - 10. Madelung potential^{ACD}
- その他
 - 1. 比熱の温度依存性^{ACD}
- VASP
 - 1. 自動計算支援^{ACDplL}
入力ファイル作成支援
実行支援
 - 2. 後処理・可視化^{ACDpl}
要約
状態密度
バンド構造
物性
 - 3. 欠陥計算後処理・可視化^{ACD}
 - 4. BoltzTraP可視化^{ACD}
 - 5. ALAMODE可視化^{CD}
- LAMMPS
 - 1. 入力ファイル作成支援^{CDpl}
（後処理も）^{CDpl}

VASP – 欠陥計算 後処理

VASP計算支援



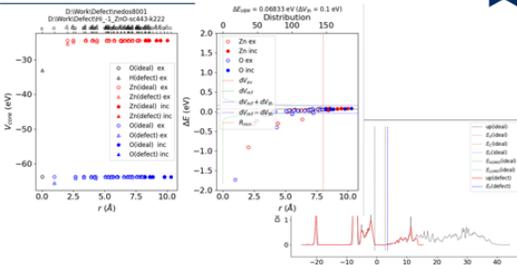
- Init for DoVASP.sh
Config and Run
・自動計算スクリプトの条件設定
・実行支援
- Post process
・計算結果の要約
・DOS、Band構造、Fat Band
・ファイル変換 (POSCAR→CIF)
(・物性計算: 動作未確認)
- Defect
・欠陥計算補正・可視化
- BoltzTraP2, ALAMODE
・可視化

Defect: VBM correction を例に

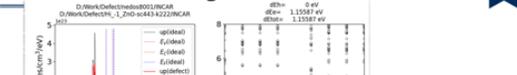


- ・ Image charge correction
- ・ VBM correction
- ・ Band filling correction
- ・ 欠陥エネルギー・濃度可視化

Defect: VBM correction



Defect: Band filling correction



pydefect: 東北大 熊谷先生

<https://kumagai-group.github.io/pydefect/>

pydefect

Search docs

Tutorial of pydefect

Tips for first-principles calculations for point defects

Tutorial for calculation of vertical transition level

Change log

» Welcome to pydefect documentation!

Welcome to pydefect documentation!

- Tutorial of pydefect
 - 1. Relaxation of the unit cell
 - 2. Calculation of band, DOS, and dielectric tensor
 - 3. Gathering unitcell information related to point-defect calculations
 - 4. Calculation of competing phases
 - 5. Construction of a supercell and defect species
 - 6. Decision of interstitial sites
 - 7. Creation of defect calculation directories
 - 8. Generation of defect_entry.json
 - 9. Parsing supercell calculation results
 - 10. Corrections of defect formation energies in finite-size supercells
 - 11. Check defect structures
 - 12. Check defect eigenvalues and band-edge states in supercell calculations
 - 13. Plot defect formation energies
- Tips for first-principles calculations for point defects
 - 1. Symmetrization of point defects
 - 2. Tips for hybrid functional calculations
- Tutorial for calculation of vertical transition level
- Change log

Indices and tables

- Index
- Module Index
- Search Page

- 必要なモデルの作成
 - 競合相情報を Materials Projectから取得
 - 入力ファイルの作成の自動化
 - 欠陥モデルサイズの自動決定
- 安定相図作成・化学ポテンシャル抽出
- 補正の自動化
 - band filling
 - image charge correction
Y. Kumagai and F. Oba,
Electrostatics-based finite-size corrections for first-principles point defect calculations
Phys. Rev. B **89** (2014) 195205.
 - energy shift

第一原理計算と 全エネルギー

第一原理計算とは

原子配列のみの入力から、
量子方程式に基づいて
精度の高い全エネルギーを出力する

自由エネルギーの近似

熱力学的エネルギー:

- 定温、定圧 : Gibbsエネルギー $G = E + PV - TS$
- 定温、定積 : Helmholtzエネルギー ($P = 0$) $F = E - TS$
- 定エネルギー、定圧 : エンタルピー ($T = 0$) $H = E + PV$
- 定エネルギー、定積 : 内部エネルギー ($T = 0, P = 0$) E

第一原理計算: 全エネルギーを精度良く計算できる

- イオンを固定した場合は比較的高速: Born-Oppenheimerの断熱近似
系の全エネルギー : 内部エネルギー E
外圧 P_{ext} と平衡な構造を計算 : エンタルピー $H = E + PV$
- イオンが動く (格子エントロピー) 場合の計算は時間がかかる
フォノン計算によりエントロピーを計算: 自由エネルギー G, F

第一原理計算による物性計算

第一原理計算

変分原理: 全エネルギーの期待値 $\langle \psi | H | \psi \rangle$ を最小化する固有状態を見つける

エネルギー準位

波動関数

全エネルギー

k 依存性

E 依存性
第1B.Z.で総和

電気双極子遷移確率
 $\langle \Psi_{fin} | x | \Psi_{ini} \rangle$

Berry位相

$|\psi|^2$

バンド構造

状態密度

光学スペクトル

電気分極

電子分布

$$m^* = \left(\frac{1}{\hbar^2} \frac{\partial^2 E(\mathbf{k})}{\partial k^2} \right)^{-1}$$

有効質量

$$n_e = \int_{E_c}^{\infty} f_e(E) D_e(E) dE$$

自由キャリア密度
イオン化不純物
状態密度有効質量

$$c_i c_j \langle \phi_j | H | \phi_i \rangle$$

化学結合の知見
結合あたりエネルギー (iCOHP)

磁気物性、半導体物性: 移動度、キャリア濃度、電子伝導度、Hall係数、熱電係数

第一原理計算による物性計算

第一原理計算

全エネルギー U

自由エネルギー

$$H = U - PV \quad F = U + TS \quad G = U - PV + TS$$

波動関数

Berry位相

電気分極

微小変位モデルから微分
(finite displacement method)

誘電率
圧電定数

フォノン分散

歪モデルから
全エネルギー微分

弾性定数

反応式中の物質の
全エネルギーの差

反応エネルギー

安定構造

欠陥モデルとの
全エネルギーの差

**欠陥生成
エネルギー**

異なる帯電状態の欠陥モデルの比較

電荷遷移準位
(ドナー／アクセプター準位)

半導体の欠陥計算 概略

欠陥計算で何を知りたいか

- ドーピングできるのか
- ドーピングできないのであれば、なぜか

ドーピングできるのであれば、以下の情報を知りたい

- 欠陥濃度
- ドナー準位、アクセプター準位
- Fermi準位
- ドーピング: 自由電子濃度、自由正孔濃度

ドーピング可能性の支配因子

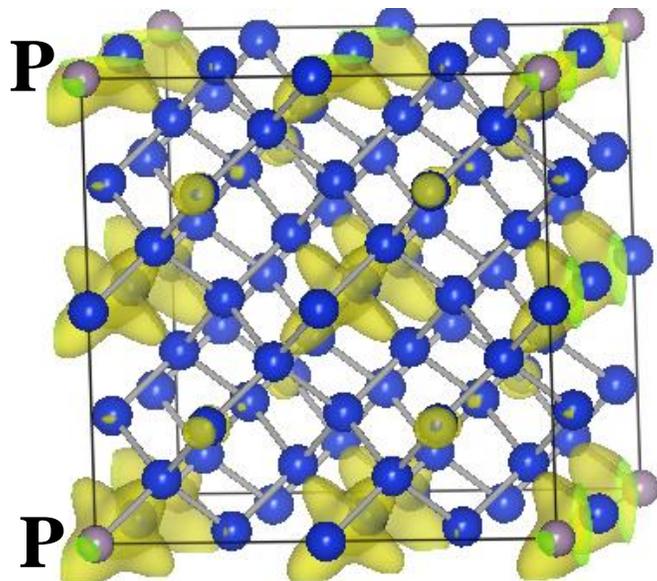
- ドーパントの固溶率
- ドナー／アクセプター準位
- ドナー／アクセプター密度
- 活性化率: イオン化ドナー／アクセプター密度

- 電荷補償欠陥

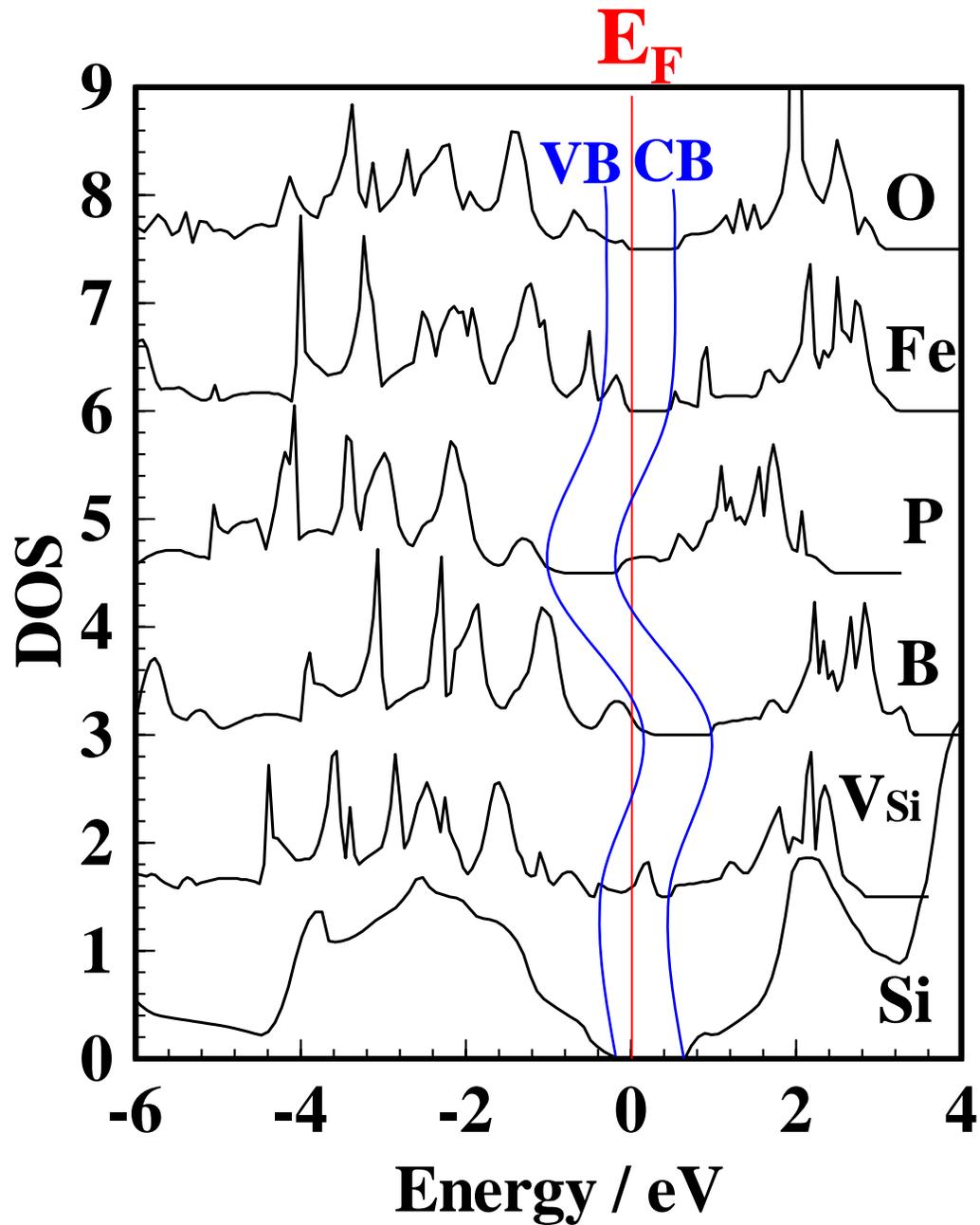
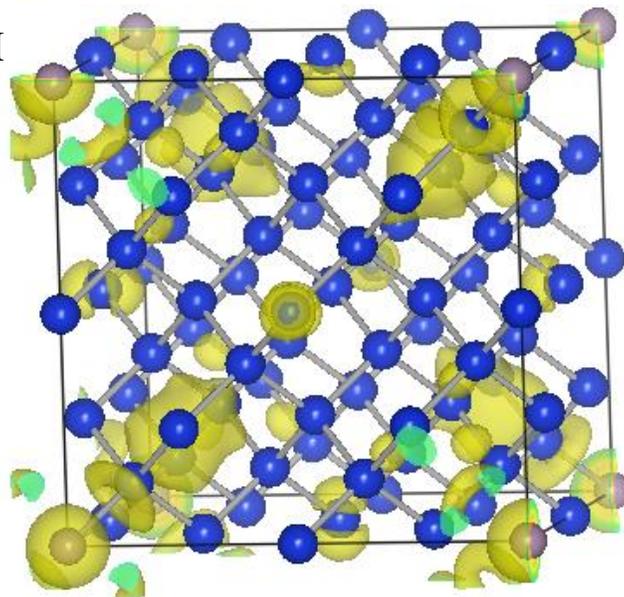
簡便法: ドーピングモデルでの計算: Si:P

VASP, PBE

$$E = E_{\text{CBM}} + 0.18 \text{ eV}$$



$$E = E_{\text{CBM}}$$



欠陥計算の問題

- DFTの計算は、せいぜい数百原子のモデル
~ 10^{20} cm^{-3} の欠陥濃度しか議論できない
⇔ 実際の半導体で問題になるのは $10^{15} \text{ cm}^{-3} \sim 10^{18} \text{ cm}^{-3}$
- 計算のモデルでは構成元素の帯電状態は一意的に決まるが、
実際にはさまざまな帯電状態が考えられる。
- 半導体のFermi準位は広い範囲で変化しうる
- 試料の作製条件 (平衡条件) でも変わる

欠陥計算の問題: 対応

- 計算自体は高濃度領域の計算を行い、希薄極限への補正を行う
- 異なる帯電状態のモデルの計算を行う
- E_F の効果は、計算対象の領域が、 E_F を持つ半導体ホスト領域と平衡状態にあるとして取り入れる
 - 欠陥生成エネルギー、欠陥・キャリア濃度などは E_F の関数
- 合成条件は 構成元素の化学ポテンシャルとして取り込む
 - 相安定領域の境界・頂点により範囲が限定される
 - ⇒ 代表的な平衡条件における化学ポテンシャルで計算

欠陥計算の特徴

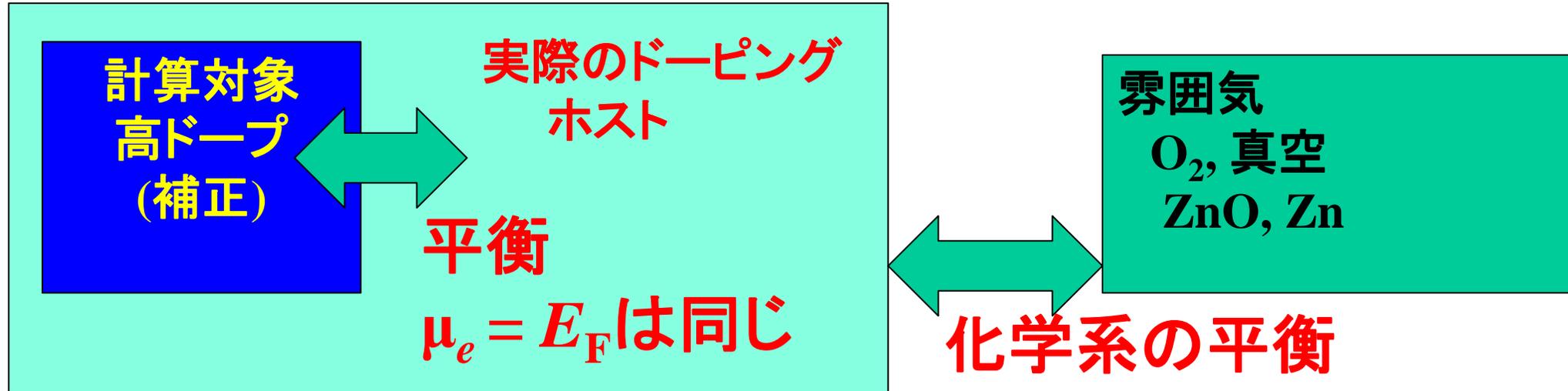
- 基本的に、理想結晶と欠陥モデルの全エネルギーが計算できればよい

古典的計算 (経験的格子力学、分子動力学) でも計算できる
どの第一原理計算プログラムを使っても計算できる

- E_F 依存性を考えるには、
電子構造、電荷密度分布などの情報が必要

欠陥モデルと現実系の平衡

電子系の平衡



目的:

$$E_{D,q}^f(E_F, \mu) \\ = E_{D,q} - E_0 - n_{Zn} \mu_{Zn} - n_O \mu_O \\ + q(E_F - E_{VBM}^0)$$

ZnOとの平衡条件: $\mu_{Zn} + \mu_O < \mu_{ZnO}$

Zn過剰条件: $\mu_{Zn} = \mu_{Zn(bulk)}$

O過剰条件: $\mu_O = \mu_{O_2}$

中間条件: $\mu_O < \mu_{O_2}, \mu_{Zn} < \mu_{Zn(bulk)}$

Gibbsエネルギーと化学ポテンシャル

定温・定圧条件の自由エネルギー:

$$\text{Gibbsエネルギー } G = E + PV - TS$$

粒子 a の化学ポテンシャルの定義:

$$\mu_a = \left(\frac{\partial G}{\partial N_a} \right)_{T,p,(N_a \text{ 以外})}$$

相 A, B 間の化学平衡 : $\mu_{a,A} = \mu_{a,B}$

電子系の平衡 : $\mu_{e,A} = \mu_{e,B}$ ($E_{F,A} = E_{F,B}$)

平衡状態: Gibbs-Duhemの法則が成立 $G(T, p, N_a) = \sum_a N_a \mu_a$

$T = 0$ Kではエンタルピーで近似できる $H_A(T, p, N_a) = \sum_A (E_A + PV_A) = \sum_a N_a \mu_a$

反応エネルギー (Gibbsエネルギー)

反応式 $nA + mB \Rightarrow A_nB_m$ の場合

定圧・低温条件での反応: Gibbsエネルギー G によって決まる

$$\Delta G(A_nB_m) = G(A_nB_m) - nG(A) - mG(B)$$

平衡状態: Gibbs-Duhemの法則

$$G = \sum \mu_i$$

$$\Delta G(A_nB_m) = G(A_nB_m) - nG(A) - mG(B) = n\Delta\mu_A + m\Delta\mu_B$$

DFTでは G を内部エネルギー E , エンタルピー $H = E + PV$ で近似

反応式 $nA + mB \Rightarrow A_n B_m$ の場合

$$\Delta G(A_n B_m) \sim E(A_n B_m) - nE(A) - mE(B) = n\Delta\mu_A + m\Delta\mu_B$$

↑化学ポテンシャルの拘束条件

化学ポテンシャルの表現: $\mu_i = \mu_{i,0} + \Delta\mu_i$

$\mu_{i,0}$: 基準物質 (安定単体) における化学ポテンシャル

$\Delta\mu_i$: $\mu_{i,0}$ からの差

自由に変えられる化学ポテンシャル = 構成元素数 - 1

相安定条件

相安定条件

SrTiN₂を例に:

1. 自由エネルギーに関する相安定条件

$$\Delta\mu_{Sr} + \Delta\mu_{Ti} + 2\Delta\mu_N = \Delta H_{SrTiN_2} < 0$$

2. 異相として単体が析出しない条件

$$\Delta\mu_{Sr} < 0, \Delta\mu_{Ti} < 0, \Delta\mu_N < 0$$

3. 他の異相が出現しない条件

$$2\Delta\mu_{Ti} + \Delta\mu_N < \Delta H_{Ti_2N}$$

$$\Delta\mu_{Ti} + \Delta\mu_N < \Delta H_{TiN}$$

$$2\Delta\mu_{Sr} + \Delta\mu_N < \Delta H_{Sr_2N}$$

$$\Delta\mu_{Sr} + \Delta\mu_N < \Delta H_{SrN}$$

$$\Delta\mu_{Sr} + 2\Delta\mu_N < \Delta H_{SrN_2}$$

$$\Delta\mu_{Sr} + 6\Delta\mu_N < \Delta H_{SrN_6}$$

化学的相安定性: SrTiN₂を例に

1. 可能性のある相: Sr, Ti, N₂, SrN, Sr₂N, SrN₂, SrN₆, TiN, Ti₂N, など
2. 熱力学条件: 自由エネルギー = 構成元素の化学ポテンシャルの和

例: $\Delta\mu_{Sr} + \Delta\mu_{Ti} + 2\Delta\mu_N = \Delta H_{SrTiN_2}$ (DFTで計算): 制約条件

$\mu_e = \mu_e^0 + \Delta\mu_e$: 元素 e の化学ポテンシャル (μ_e^0 は単体の化学ポテンシャル)
 化学ポテンシャルは合成条件におけるパラメータ: 計算結果は μ_e に関するマップになる

3. 自由エネルギー (計算されるのは一般的にエンタルピー) に関する相安定条件

$\Delta\mu_{Sr} + \Delta\mu_{Ti} + 2\Delta\mu_N = \Delta H_{SrTiN_2} = -5.87\text{eV} < 0$

2. 異相として単体が析出しない条件

$\Delta\mu_{Sr} < 0$ ①, $\Delta\mu_{Ti} < 0$ ②, $\Delta\mu_N < 0$ ③

3. 他の異相が出現しない条件:

$2\Delta\mu_{Ti} + \Delta\mu_N < \Delta H_{Ti_2N}$ ④

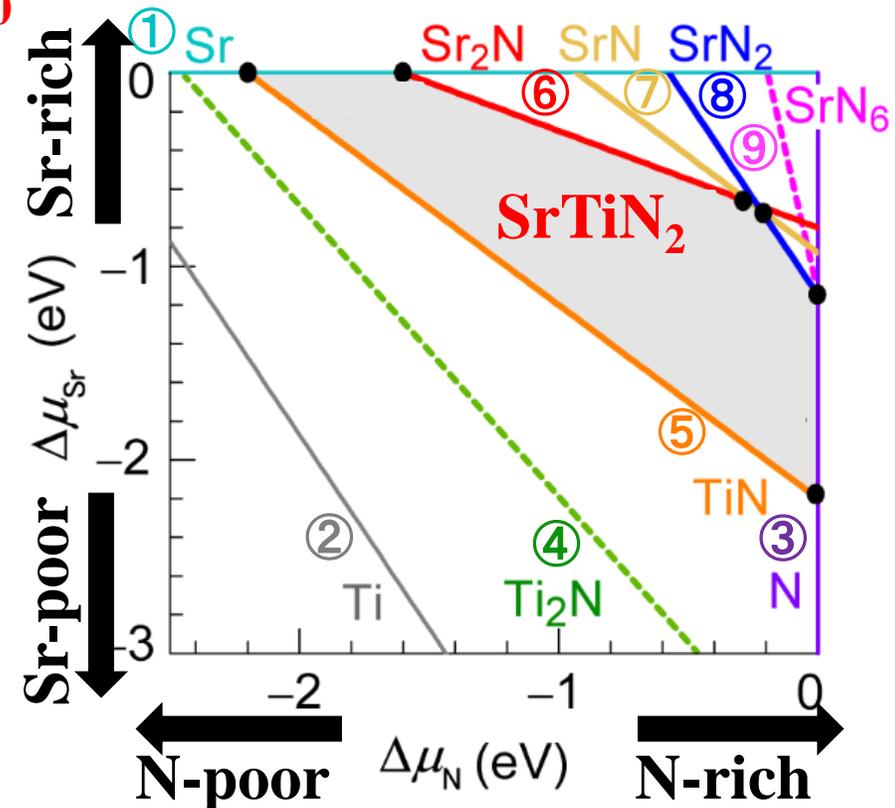
$\Delta\mu_{Ti} + \Delta\mu_N < \Delta H_{TiN}$ ⑤

$2\Delta\mu_{Sr} + \Delta\mu_N < \Delta H_{Sr_2N}$ ⑥

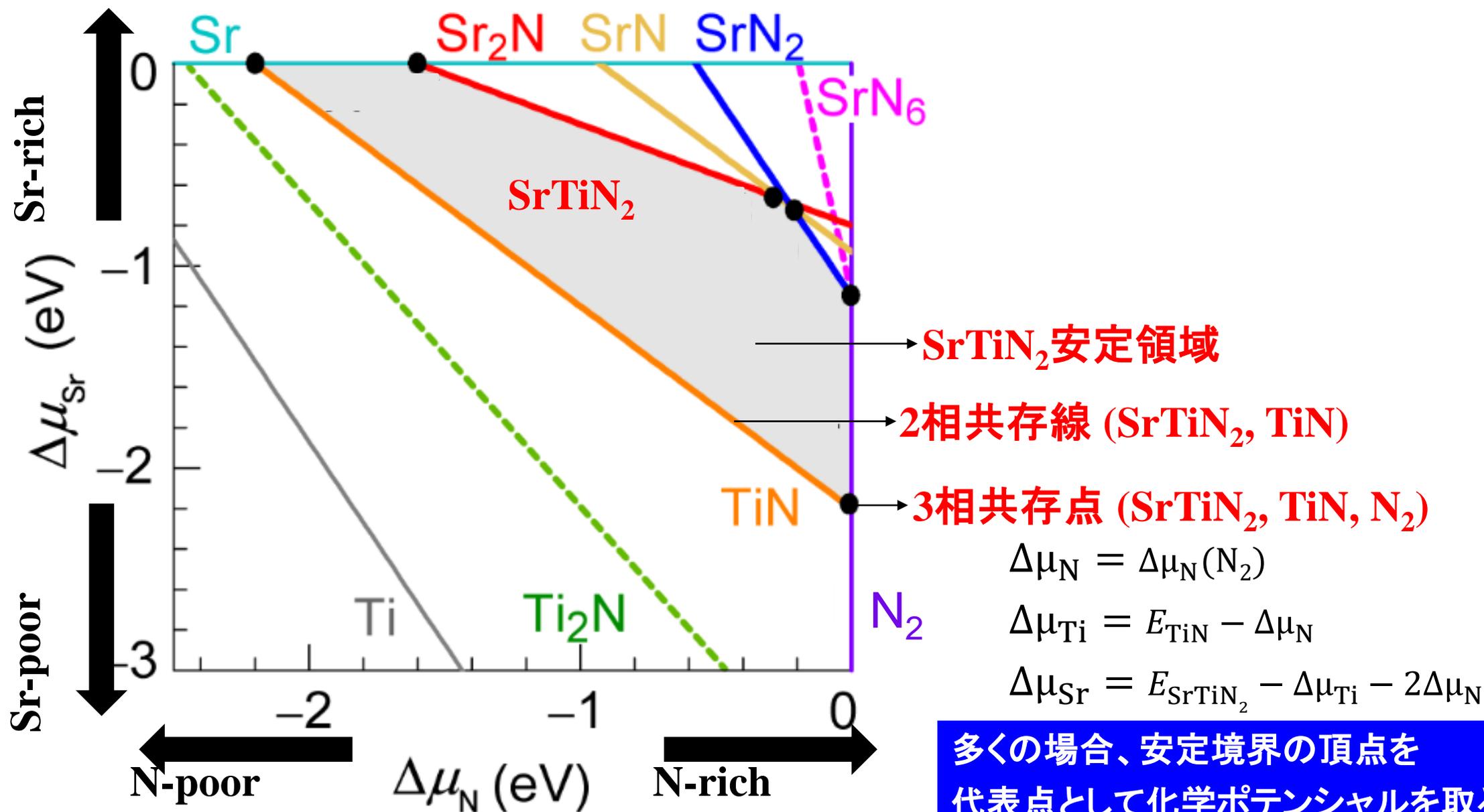
$\Delta\mu_{Sr} + \Delta\mu_N < \Delta H_{SrN}$ ⑦

$\Delta\mu_{Sr} + 2\Delta\mu_N < \Delta H_{SrN_2}$ ⑧

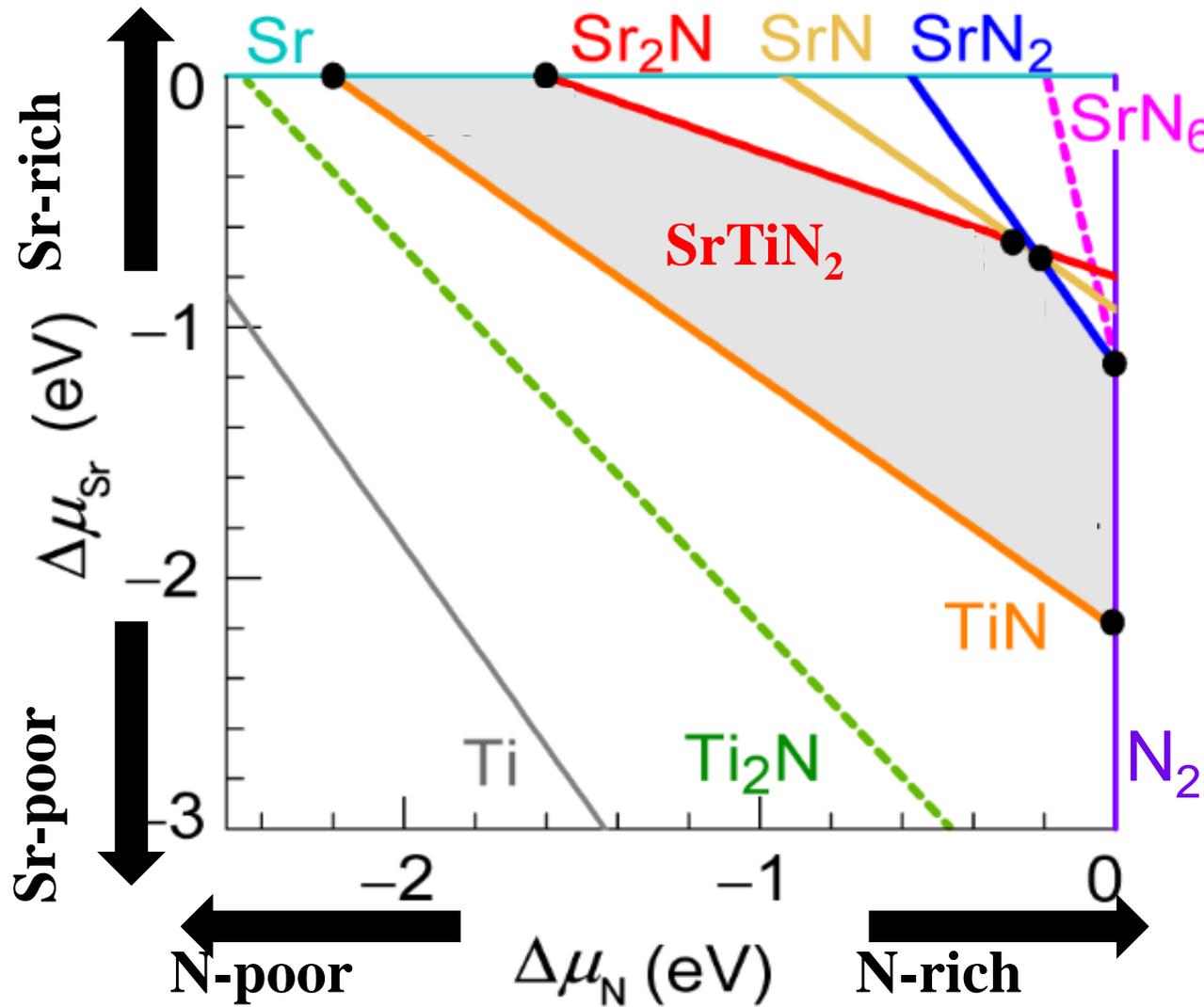
$\Delta\mu_{Sr} + 6\Delta\mu_N < \Delta H_{SrN_6}$ ⑨



安定境界と化学ポテンシャル



安定境界と化学ポテンシャル



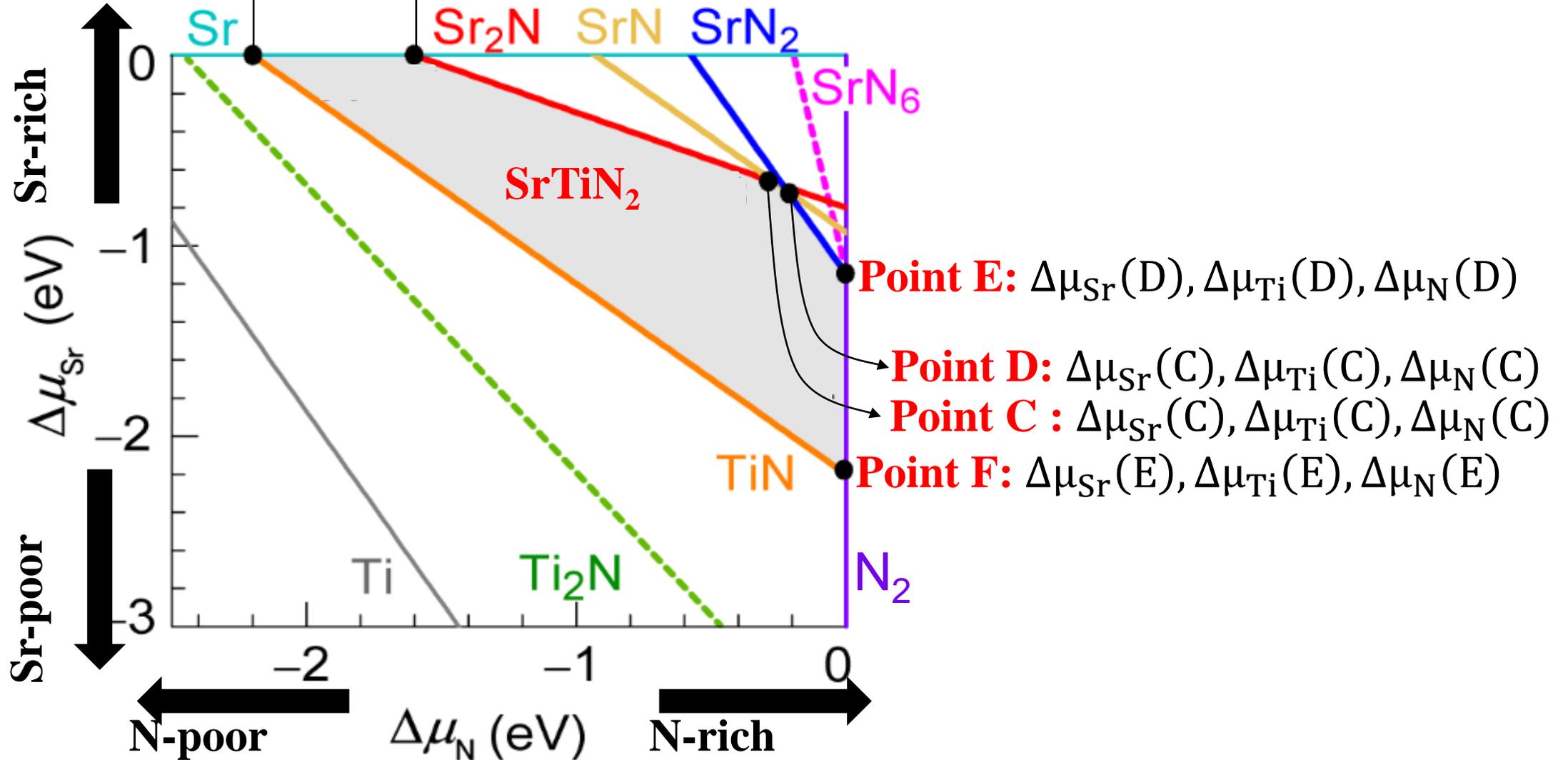
- 安定単体が析出しないとしているので
 $\Delta\mu_i$ は必ず < 0
- 平衡状態を仮定できれば、
共存相から
合成条件がどの化学ポテンシャルに
対応するかを推察できる

安定境界と化学ポテンシャル

代表点

Point A: $\Delta\mu_{\text{Sr}}(A), \Delta\mu_{\text{Ti}}(A), \Delta\mu_{\text{N}}(A)$

Point B: $\Delta\mu_{\text{Sr}}(C), \Delta\mu_{\text{Ti}}(C), \Delta\mu_{\text{N}}(C)$

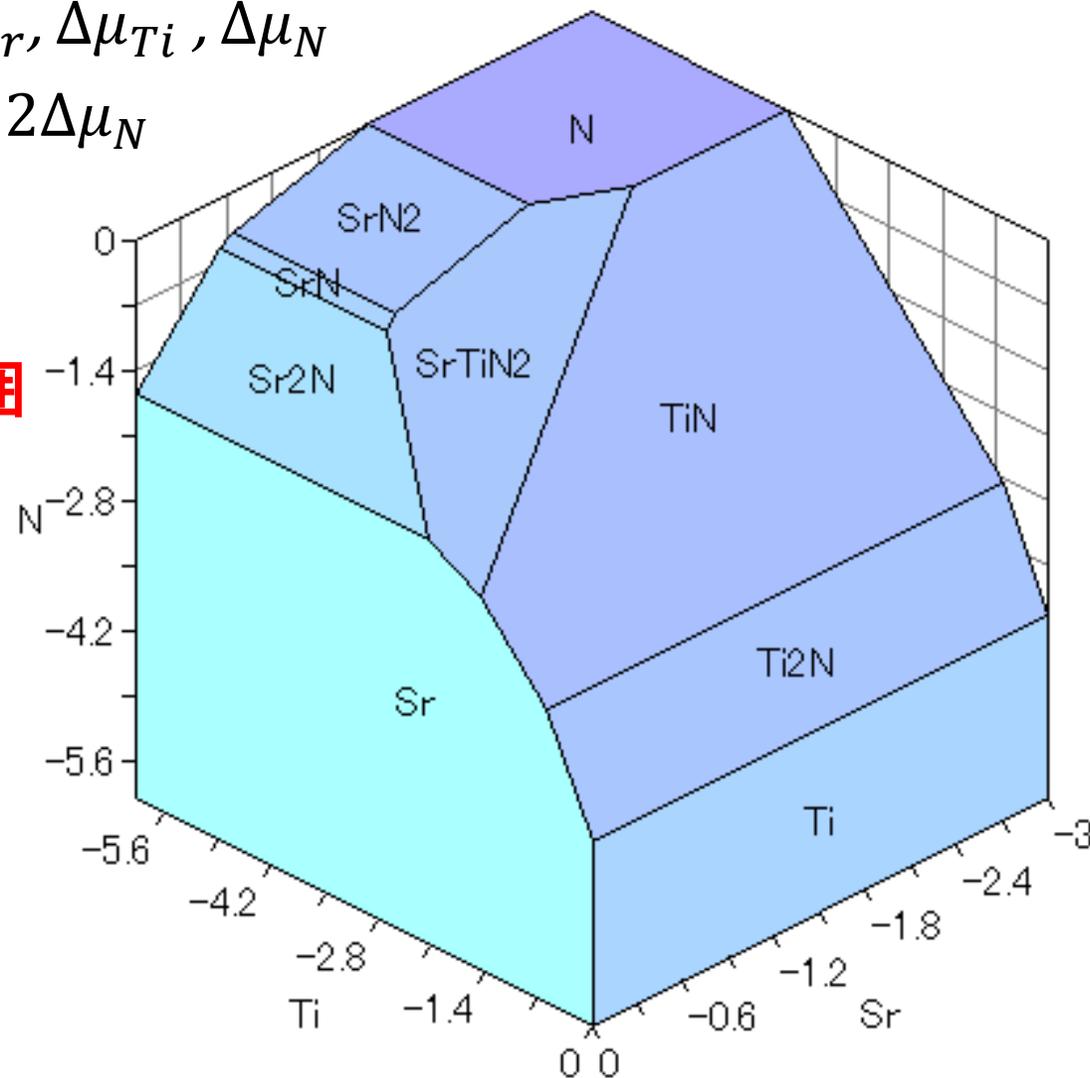


SrTiN₂の化学的相安定性

化学ポテンシャルの変数は3つ: $\Delta\mu_{Sr}$, $\Delta\mu_{Ti}$, $\Delta\mu_N$

制約条件: $E_{SrTiN_2} = \Delta\mu_{Sr} + \Delta\mu_{Ti} + 2\Delta\mu_N$

自由度は2つ: **3次元図の面内が
取りえる化学ポテンシャル範囲**

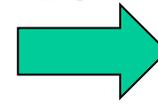


代表的な安定境界における欠陥生成エネルギー

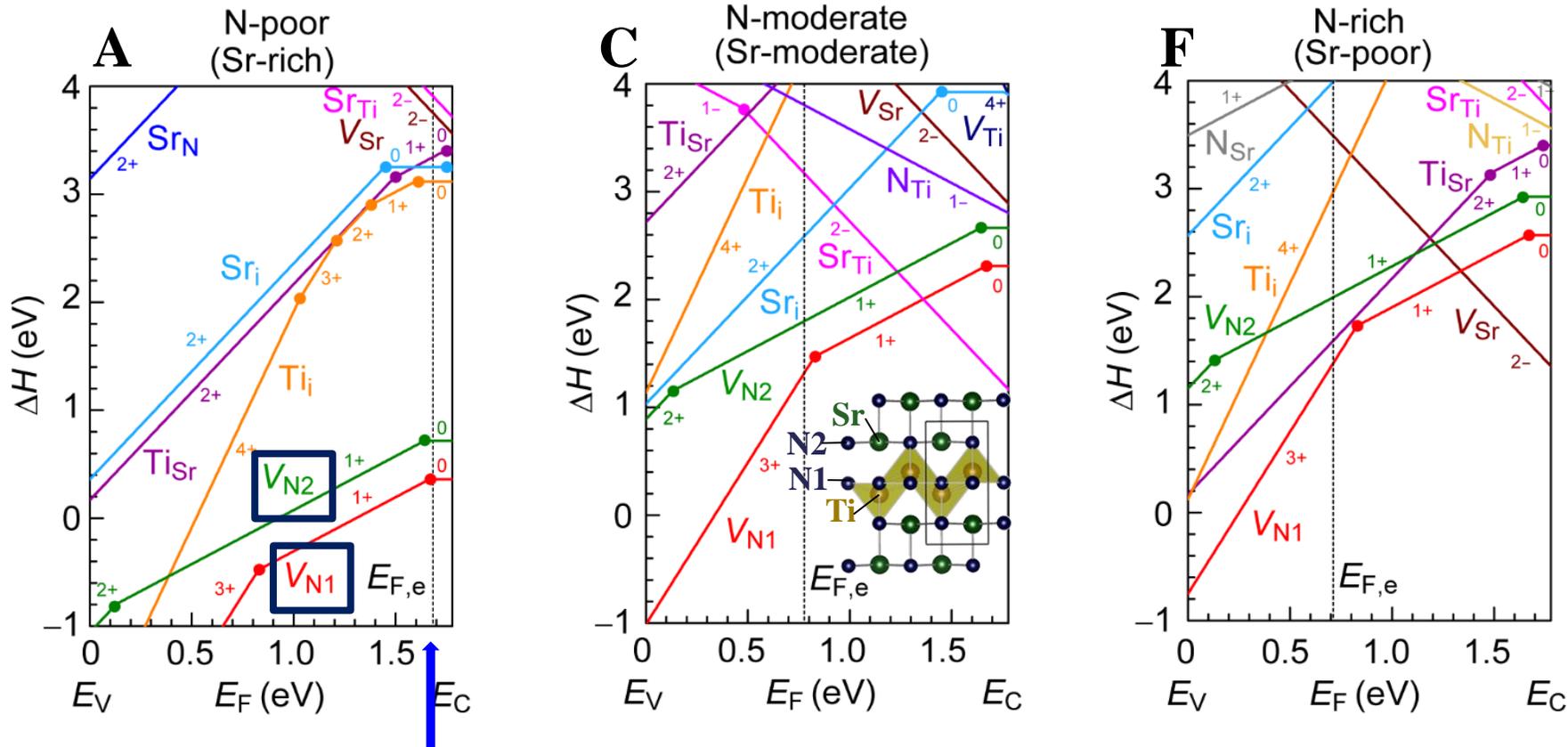
欠陥の生成エネルギーは、関連粒子 (元素、電子) の化学ポテンシャルの関数

電子の化学ポテンシャル = フェルミ準位

帯電欠陥の生成エネルギーはフェルミ準位に依存



E_F に関するグラフになる



平衡フェルミ準位 $E_{F,eq}$: 電荷中性条件から決定される

欠陥の電荷総和 + 自由正孔電荷 + 自由電子電荷 = 0

E_C に近い \Rightarrow native n-type conductor、平衡キャリア濃度: $1.1 \times 10^{18} \text{ cm}^{-3}$

気体の化学ポテンシャル

理想気体: $\mu(T, p) = \mu^0(T) + RT \ln \frac{p}{p^0}$

実在気体: $\mu(T, p) = \mu^0(T) + RT \ln \frac{f(T, p)}{p^0}$

$f(T, p)$: フガシテイ (活量)

蛇足: 活量は固体で1、気体で $p(\text{O}_2)$ でよいのか?

坪村宏、化学平衡と平衡定数、化学と教育 p.722、46巻11号 (1998)

反応 $aA + bB \Rightarrow cC + dD$ の平衡定数

$$K = \frac{a_C^c a_D^d}{a_A^a a_B^b} \quad a_i \text{は化合物 } i \text{ の活量} \quad (1)$$

成分 a の化学ポテンシャル μ_a は、標準状態の μ_a^0 を用いて

$$\mu_a = \mu_a^0 + RT \ln(a_a) \quad (3)$$

と書ける。つまり、活量 a_a は標準状態を基準とした値であり、

標準状態の活量は $a_a = 1$ である。

気体の場合は、 $a_a = p/p_0$ と、標準状態の圧力 p_0 を基準にするが、一般的に標準状態の圧力を 1 とする圧力単位を取るため、 f_a がその圧力単位での分圧になる。

平衡定数 K も本来は標準状態の濃度との比によって定義されるので、

本来は無次元であるが、実際には圧力の単位を atm、濃度の単位を mol/L として使われている。

$$K = \frac{(C_C/C_0)^c (C_D/C_0)^d}{(C_A/C_0)^a (C_B/C_0)^b}$$

実際の欠陥計算

Priority Target: 欠陥生成エネルギー

欠陥 D^q (q : 欠陥電荷) の生成エネルギー

$$E^f_{D,q}(E_F, \mu) = E_{D,q} - E_0 - n_{Zn}\mu_{Zn} - n_O\mu_O + q(E_F - E_{VBM}^0)$$

$E^f_{D,q}(E_F, \mu)$: Fermi準位 E_F の時の D^q の生成エネルギー

$E_{D,q}$: 帯電欠陥 D^q の全エネルギー

E_0 : 理想結晶の全エネルギー

μ_{Zn}, μ_O : Zn, O の化学ポテンシャル

E_{VBM}^0 : エネルギー原点。一般的に理想結晶の価電子帯端 (VBM) にとる

V_O の場合: $n_{Zn} = 0, n_O = -1$

欠陥計算の手順: pydefectのファイル構造

例: CsCl

▼	pydefect	
▼	CsCl	プロジェクトルートディレクトリ
>	cpd	2. 相安定性、化学ポテンシャル (基本格子で計算)
>	defect	3. 欠陥モデル (supercellモデルで計算)
▼	unitcell	1. 理想結晶 (基本的に 基本格子で計算)
	band	バンド構造
	dielectric	1c. 静的誘電率: Image charge correctionに使用
	dos	1b. 状態密度: E_{VBM}^0
	structure_opt	1a. 構造緩和: 格子定数

cpd: Chemical Potential Diagram

例: CsCl



2. 相安定性、化学ポテンシャル (基本格子で計算)

Cl₂ #1

Cl₂ #2

Cs

CsCl

pydefectを使うと、Materials Projectから取得

MP API Keyを取得する必要

マニュアルで追加もOK

(計算時に一括処理するためには、

ディレクトリ名には _ を入れたほうがいい)

defect: 欠陥計算

例: CsCl

pydefect

CsCl プロジェクトルートディレクトリ

cpd

defect **3. 欠陥モデル** (supercellモデルで計算)

perfect **理想結晶の計算結果 (Linuxのlink)**

Va_Cl1_0 V_{Cl}^0

Va_Cl1_1 V_{Cl}^+

Va_Cl1_-1 V_{Cl}^-

Va_Cs1_0 V_{Cs}^0

Va_Cs1_1 V_{Cs}^+

Va_Cs1_-1 V_{Cs}^-

unitcell

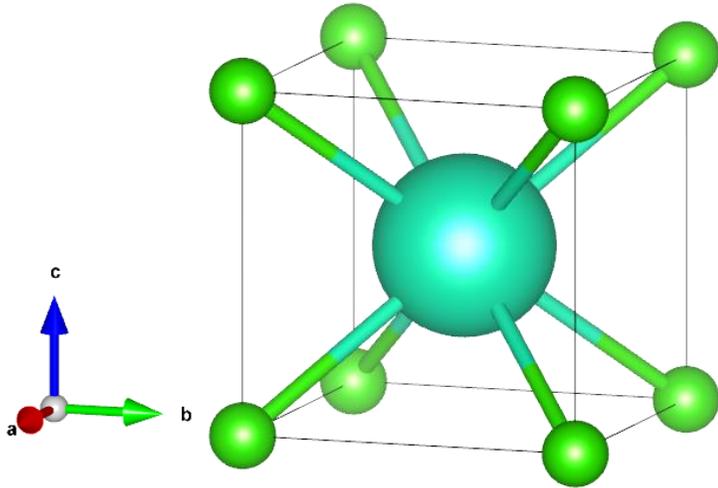
pydefectを使うと、点欠陥種と欠陥電荷を自動的に作成
マニュアルで追加、削除もOK
(ディレクトリ名には _ を入れたほうがいい)

理想結晶の計算: CsClの例

理想結晶の計算: 基本格子

構造緩和計算: 格子定数を決定

全エネルギー: E_0



構造緩和条件は厳しく:

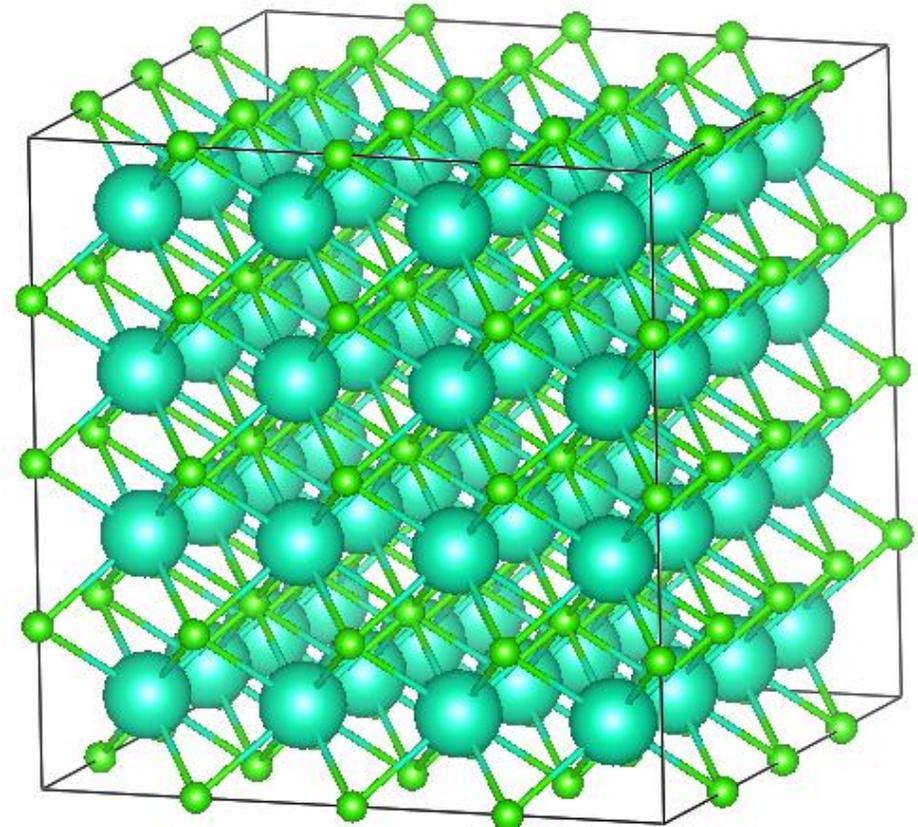
INCAR: EDIFFG = -0.005 eV/\AA

欠陥モデル: supercell

pydefectがサイズを自動決定 (4x4x4)

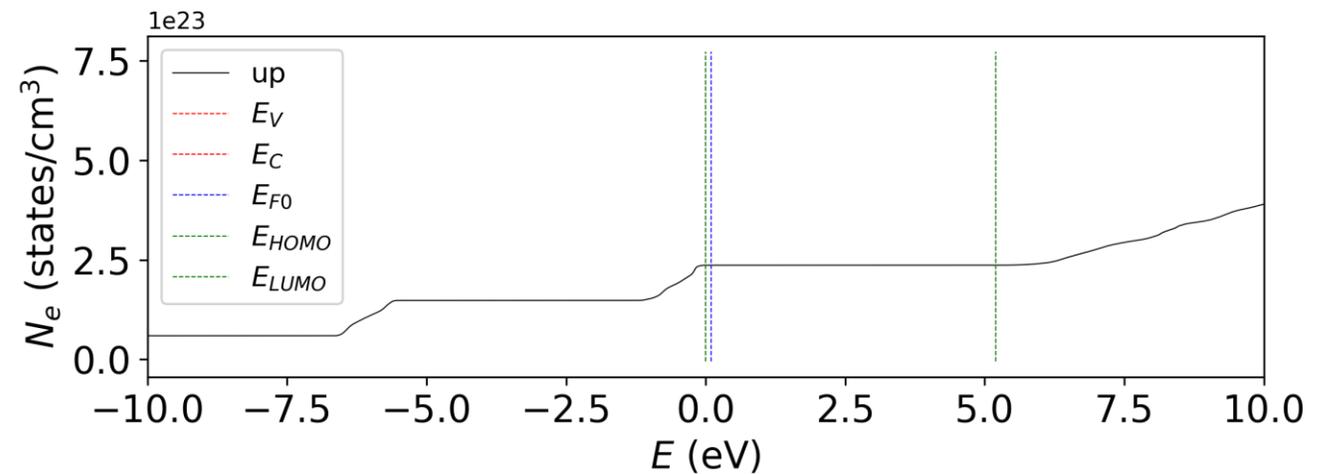
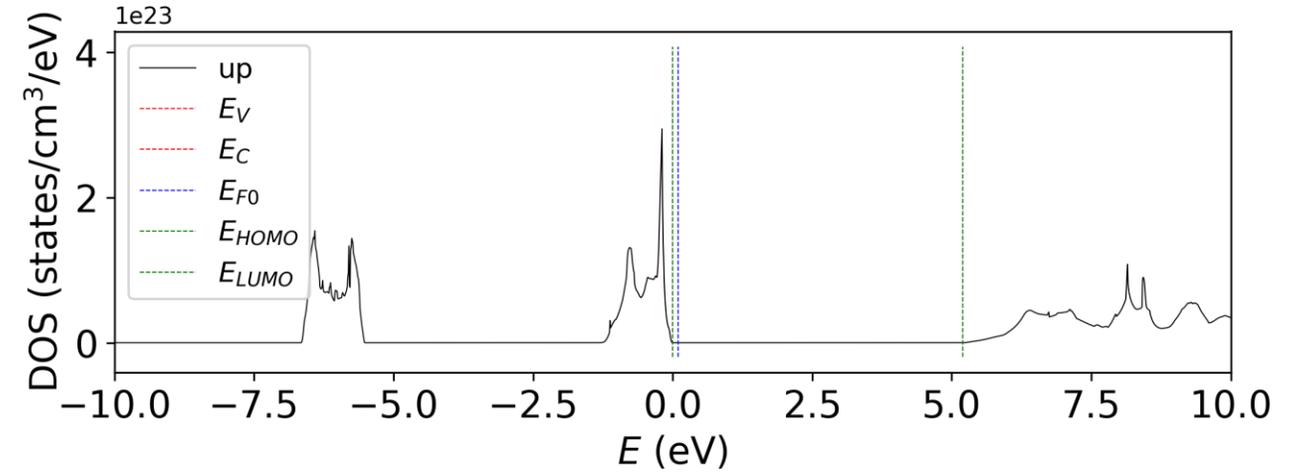
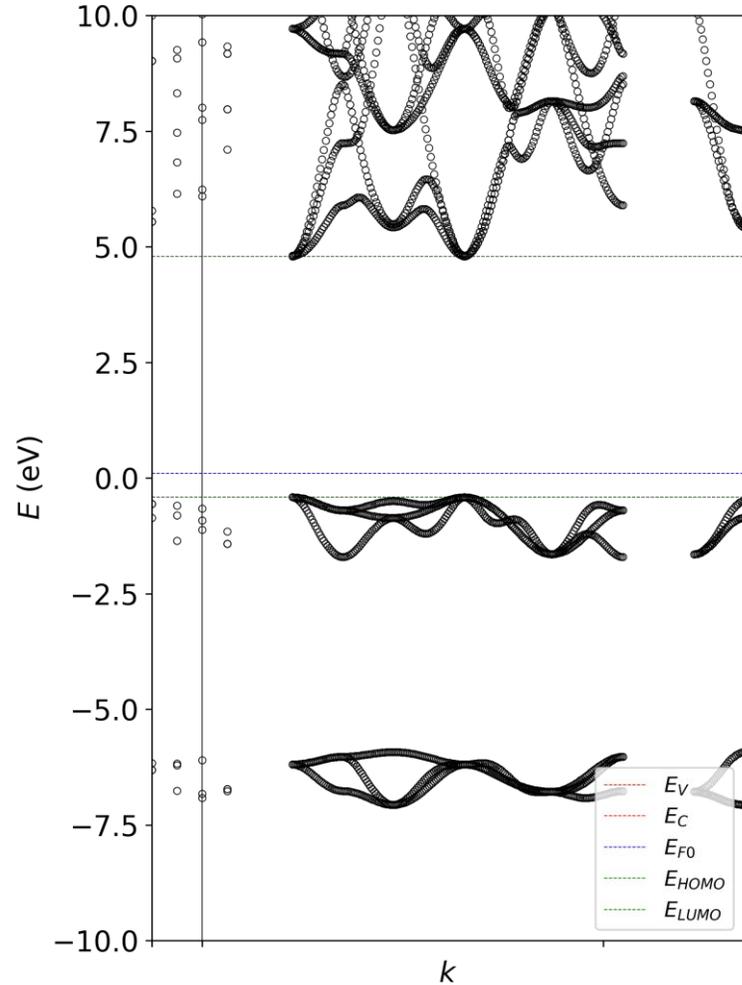
サイズ指定も可

理想結晶の格子定数を使う



理想結晶の計算: CsClの例

価電子帯端: E_{VBM}^0



理想結晶の計算: CsClの例

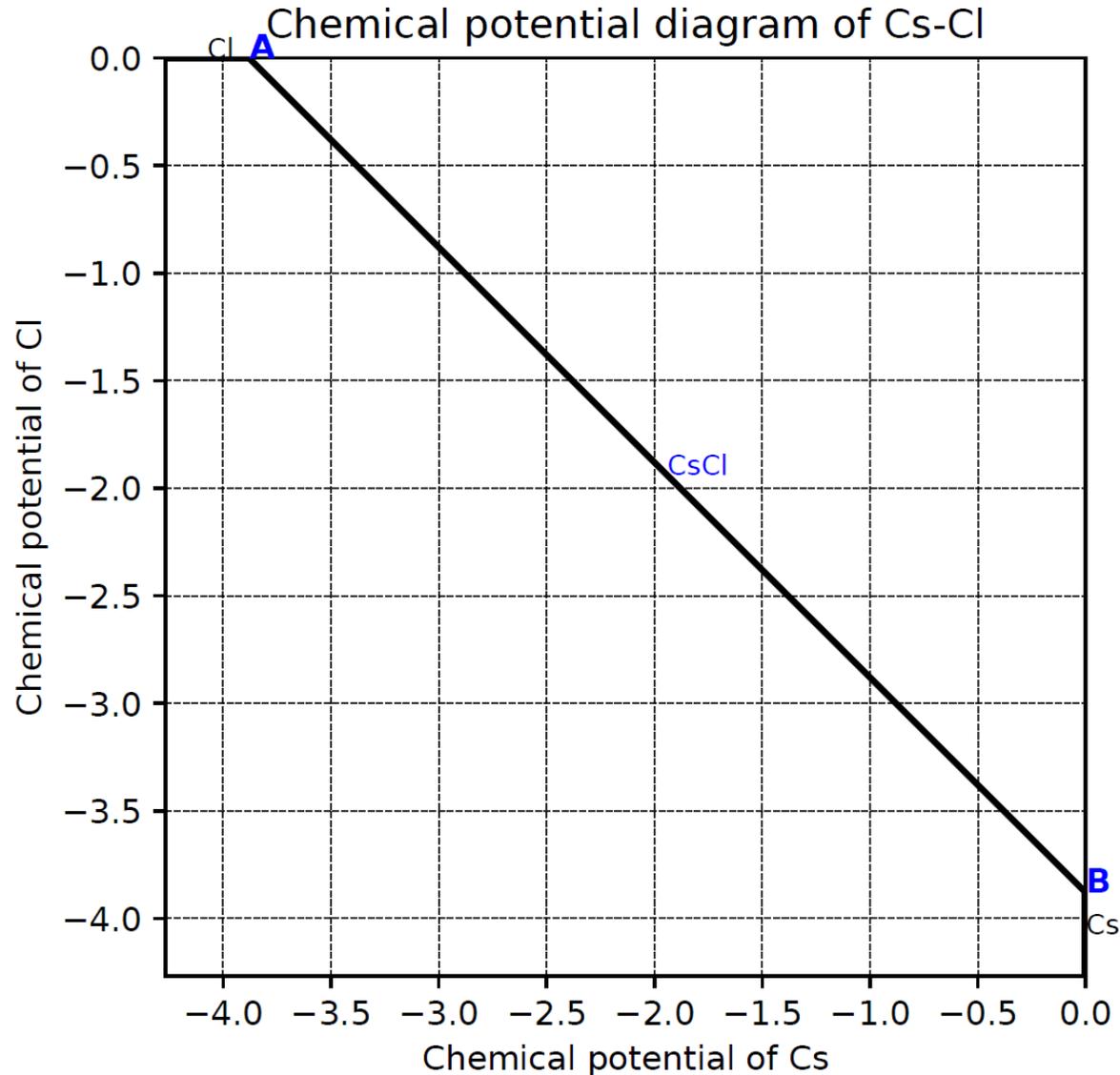
静的誘電率 INCAR: LEPSILON = True

OUTCAR:

MACROSCOPIC STATIC DIELECTRIC TENSOR IONIC CONTRIBUTION

4.706388	-0.000000	-0.000000
-0.000000	4.706388	-0.000000
-0.000000	-0.000000	4.706388

手順2: 相安定性、化学ポテンシャル (cpd)



相境界の代表点

Point A: Cl poor, Cs rich

$$\mu_{\text{Cs}} = -3.8803 \text{ eV}$$

$$\mu_{\text{Cl}} = 0$$

Point B: Cl rich, Cs poor

$$\mu_{\text{Cs}} = 0$$

$$\mu_{\text{Cl}} = -3.8803 \text{ eV}$$

手順3: 欠陥形成エネルギー

欠陥生成エネルギーの補正

C. G. Van deWalle and J. Neugebauer, J. Appl. Phys. 95, 3851 (2004).

S. Lany and A. Zunger, Phys. Rev. B **78** (2008). 235104.

S. Lany and A. Zunger, Model. Simul. Mater. Sci. Eng. **17** (2009) 084002.

M. Bockstedte, A. Marini, O. Pankratov, and A. Rubio, Phys. Rev. Lett. 105, 026401 (2010).

F. Oba, M. Choi, A. Togo, and I. Tanaka, Sci. Tech. Adv. Mater. 12, 034302 (2011)

- バンドギャップ補正
- エネルギー基準(VBM)の補正
Energy shift / E_{VBM} correction
- 高密度ドナーによる E_{F} 上昇(バンド分散)
Band filling correction
- 欠陥間の静電・分極相互作用
Image charge (charge/dipole) correction

欠陥生成エネルギーの補正 (E_{VBM} correction): V_0^0

エネルギー基準(VBM)の補正 Energy shift / E_{VBM} correction

原因: 欠陥生成、帯電により、エネルギー原点がずれる

対応: 欠陥モデルのエネルギーを ΔE_{VBM} シフトし、
理想結晶のエネルギーに合わせる

方法1: 理想結晶と欠陥モデルに共通な深いエネルギー準位を合わせる

方法2: VASPの場合: average effective core potentialを合わせる

OUTCAR: 'average (electrostatic) potential at core'

問題: 欠陥モデルのエネルギー準位には余計な分布が発生

=> 欠陥から十分離れた原子で合わせる

pydefectでは自動的に計算してくれる => 以下では、別のプログラムのグラフを見ていく

E_{VBM} correction: V_O^0 in ZnO

vasp_correction_VBM-test.bat 0.05

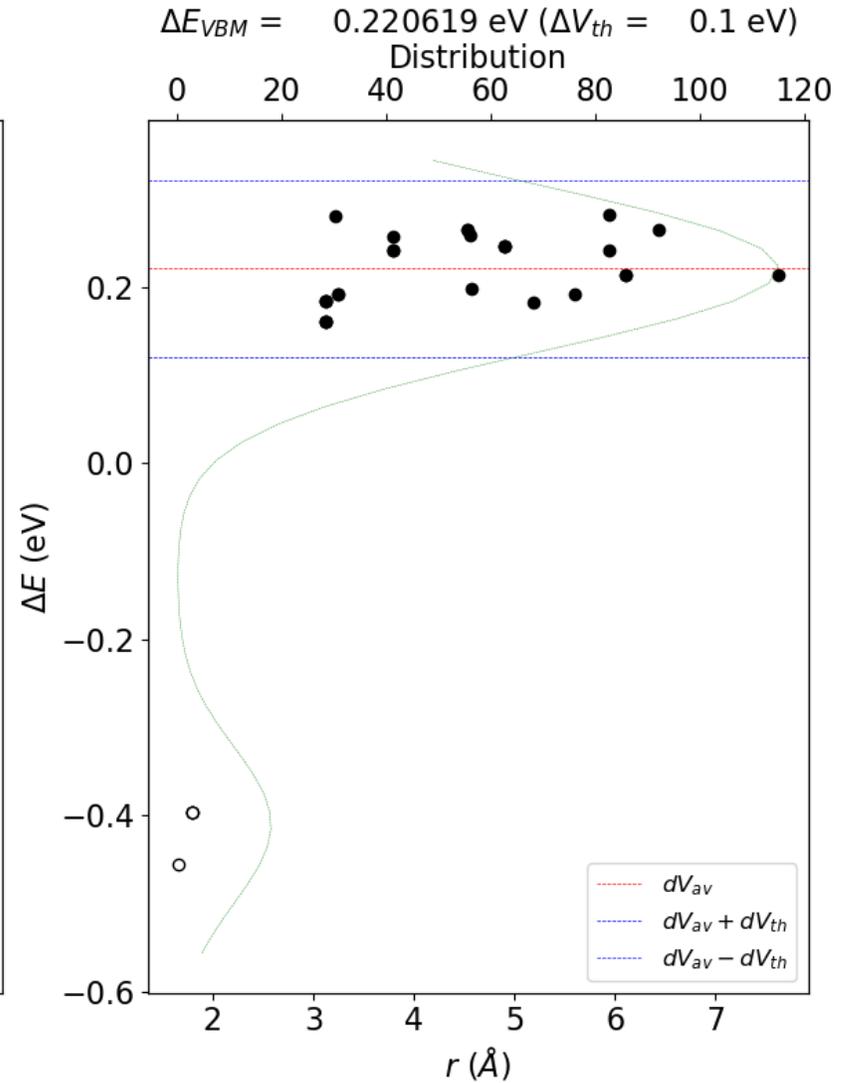
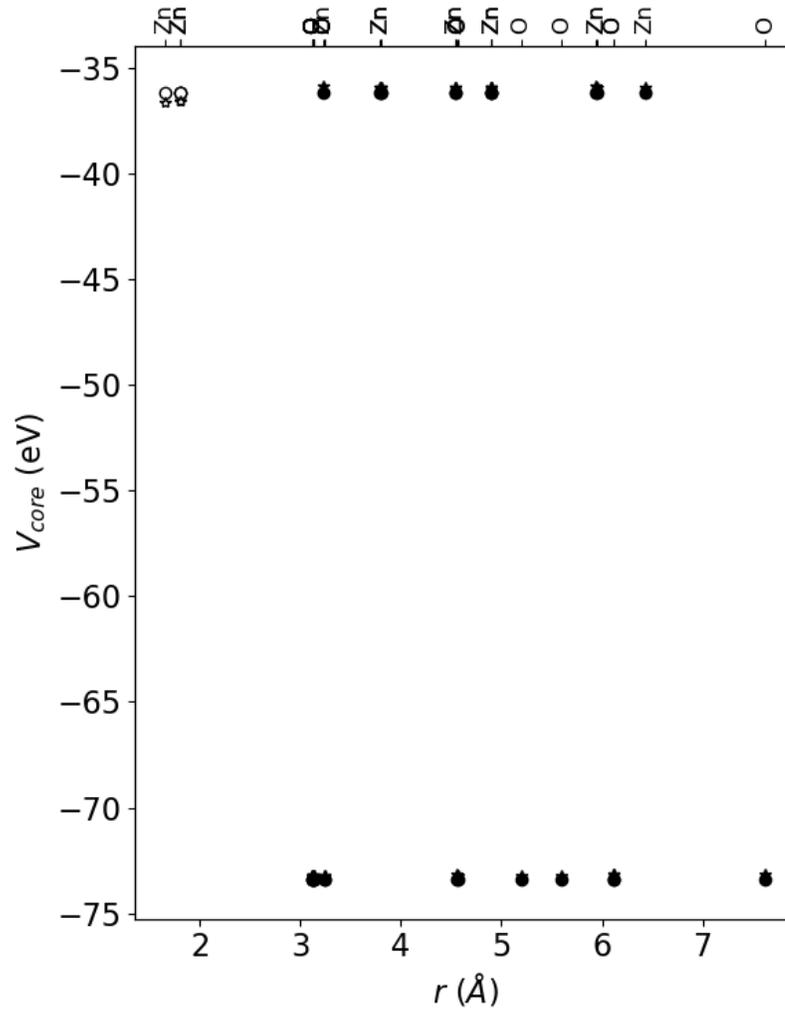
```
python vasp_correction_VBM.py exclude .\Defect\ZnO\SCF .\Defect\ZnO-VO\SCF
```

Use mode = Exclude

Output:

VBM correction:

ΔE_{VBM} : 0.220619 eV



E_{VBM} correction: V_O^0 in ZnO

vasp_correction_VBM-test.bat 0.05

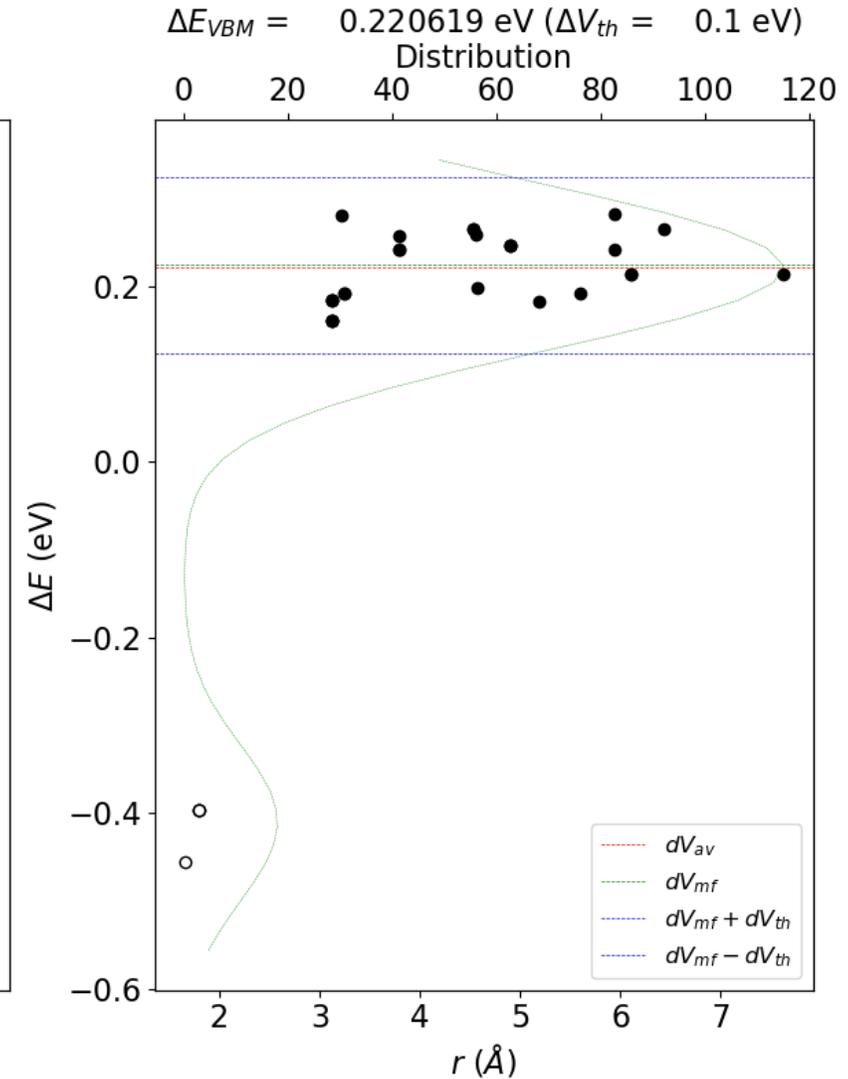
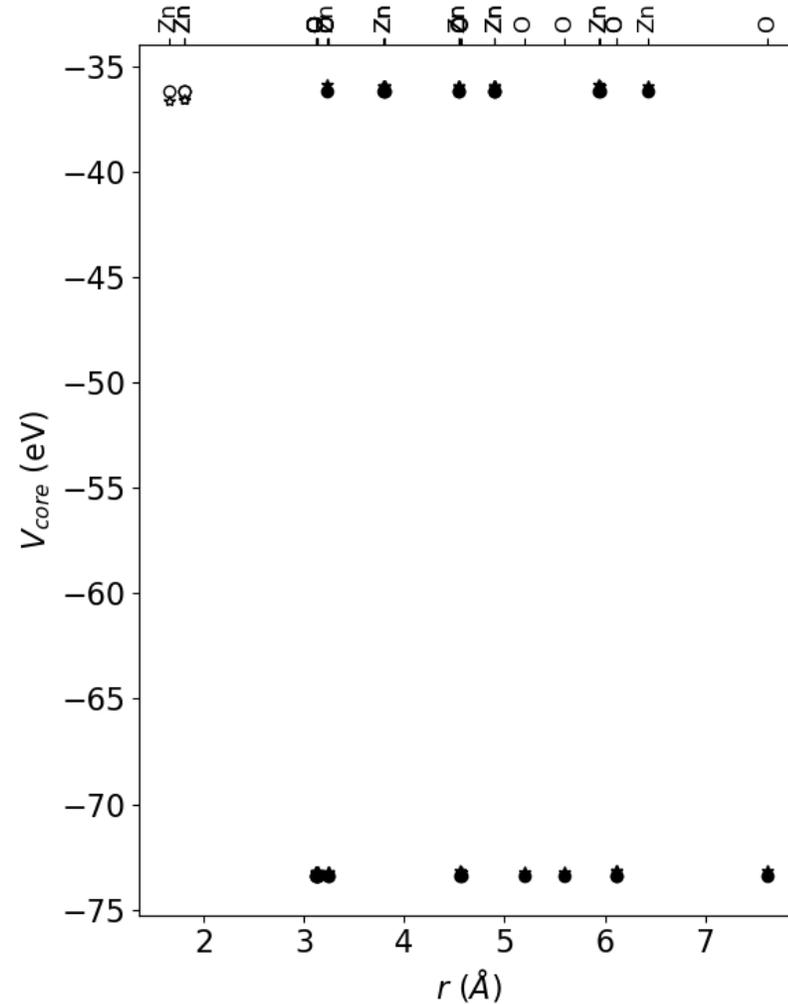
python vasp_correction_VBM.py mode .¥Defect¥ZnO¥SCF .¥Defect¥ZnO-VO¥SCF

Use mode = mode (mode value, most frequent ΔE)

Output:

VBM correction:

dEVBM: 0.220619 eV



E_{VBM} correction: V_{O^+} in ZnO

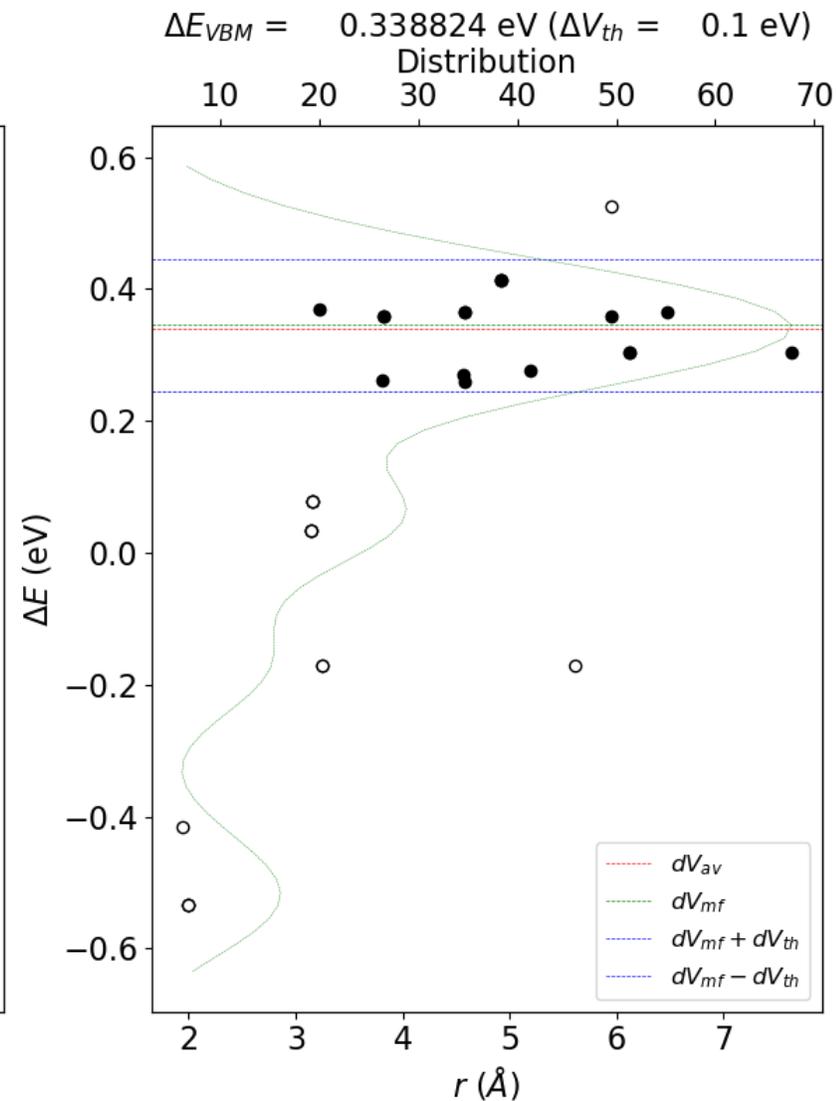
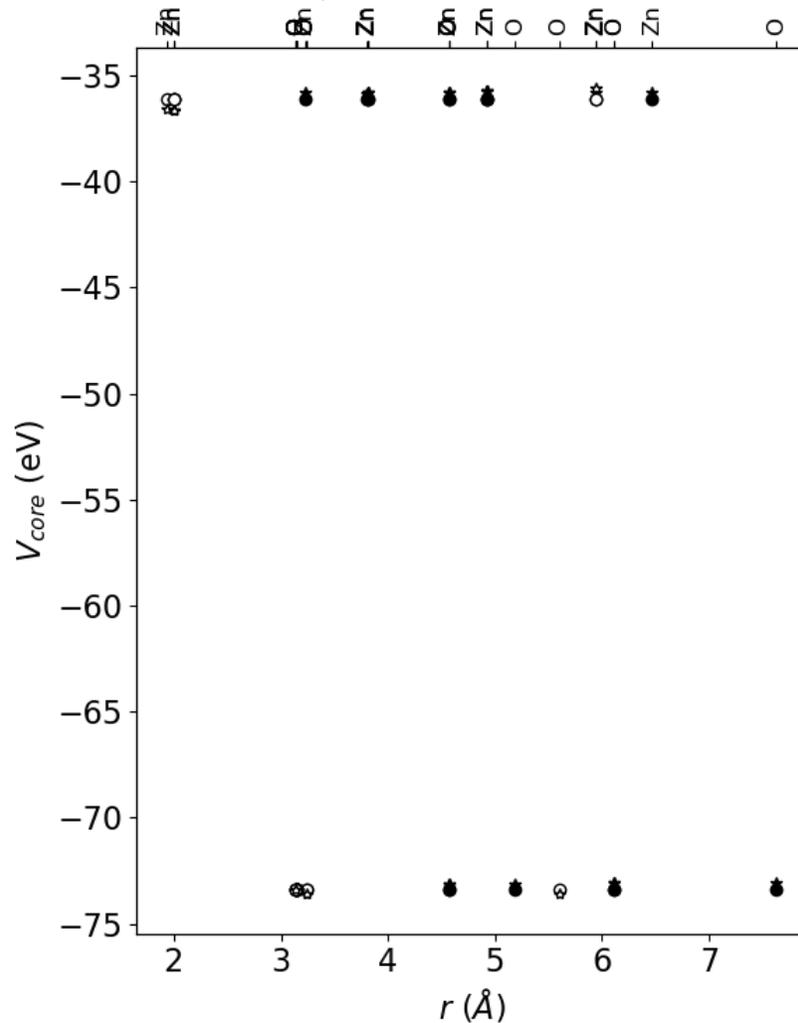
vasp_correction_VBM-test.bat

```
python vasp_correction_VBM.py mode .%Defect%ZnO%SCF .%Defect%ZnO-VOp1%SCF
```

Output:

VBM correction:

dEVBM: 0.338824 eV



高濃度ドーピング補正 (BF) correction

高密度ドナーによる E_F 上昇(バンド分散) Band filling correction

原因: 欠陥濃度が高いと、バンド分散を無視できない欠陥バンドを作る
欠陥バンドに電子／正孔が入ると、**バンド分散分、過剰なエネルギーが発生する**

対応: 欠陥濃度が**希薄極限 (dilution limit) での欠陥準位**を推定し、
ドーパされた電子／正孔の過剰エネルギーを補正

方法: 欠陥モデルの E_F の位置によって

- ・ E_F が伝導帯内: **伝導帯端を希薄極限**とし、ドーパ電子の一電子エネルギーの和を取る
- ・ E_F が価電子帯内: **価電子帯端を希薄極限**とし、ドーパ正孔の一電子エネルギーの和を取る
- ・ E_F が局在欠陥内: **局在欠陥バンド内に希薄極限**をとり、
ドーパ電子の一電子エネルギーの和を取る

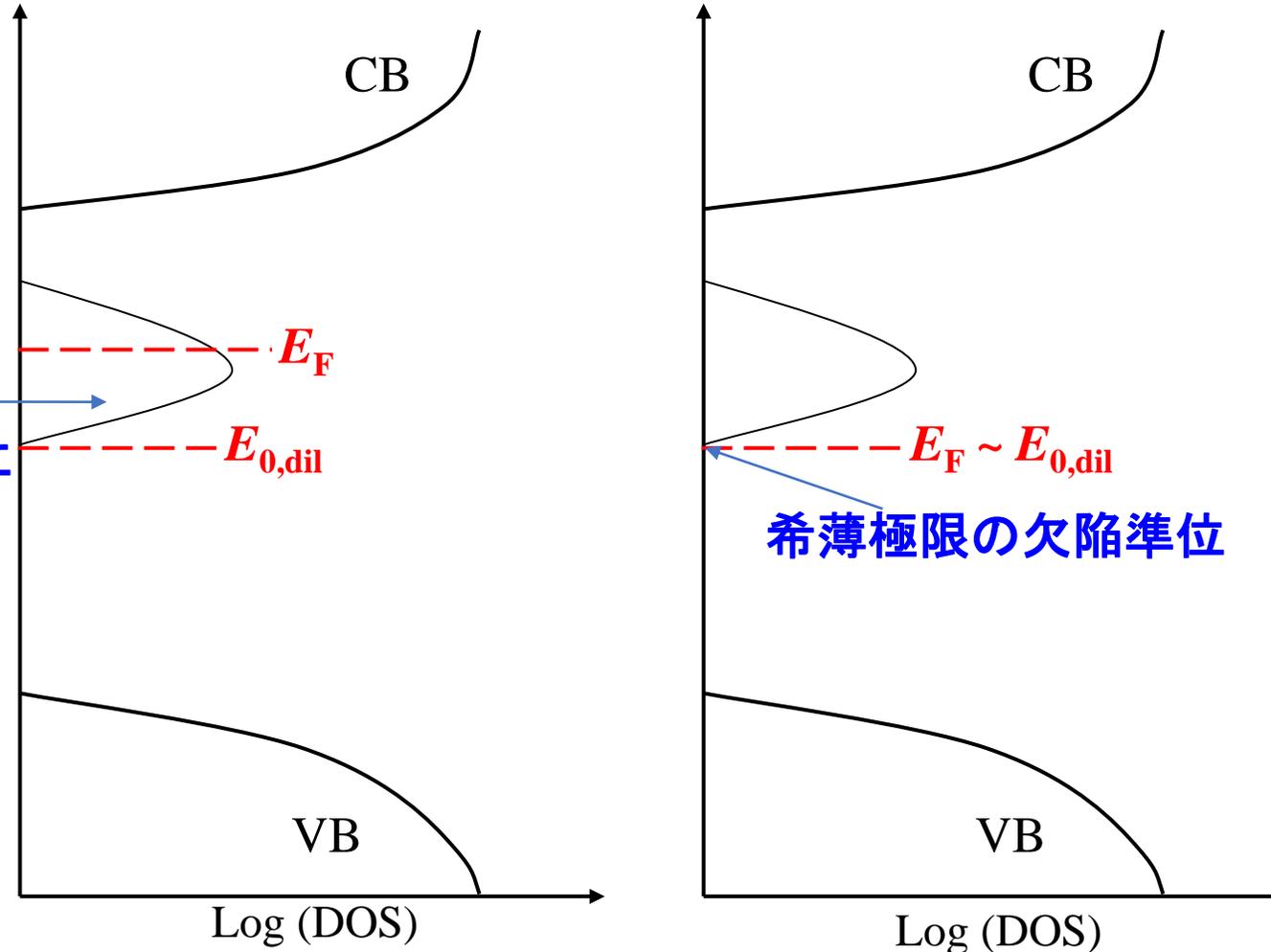
計算の都合上、局在欠陥バンド端を希薄極限にとることが多いが、
バンド幅が狭いので誤差は小さい

Band filling correction: Correction to dilution limit

Walle and Neugebauer, J. Appl. Phys. 9 (2004) 3851; Oba et al., Sci. Technol. Adv. Mater. 12 (2011) 034302

欠陥モデルが小さいので欠陥準位が分散をもつ 希薄極限ではこうあるべき

補正: $\sum w_k f_{k,l} (e_{k,l} - E_{0,dil})$
 和は $E_{0,dil}$ 以上の占有状態についてとる

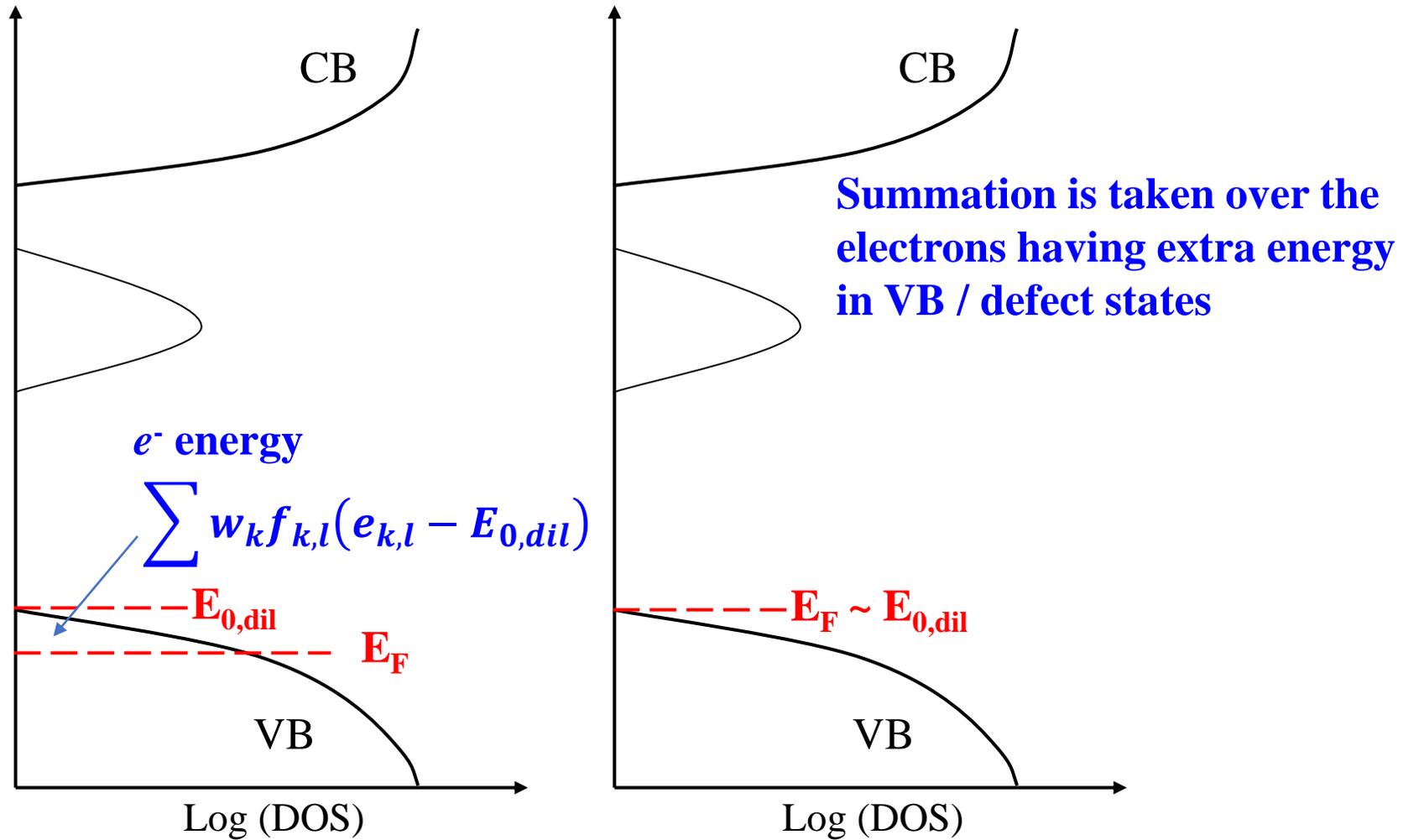


Extra energy $\Delta E_{BF} = \sum w_k f_{k,l} (e_{k,l} - E_{0,dil})$

$\Delta E_{BF} > 0$ for donor case

Band filling correction: Hole / acceptor case

Small super cell (high defect dens.) Infinite cell (dilute limit)



Extra energy $\sum w_k f_{k,l}(e_{k,l} - E_{0,dil})$
 $\Delta E_{BF} < 0$ for acceptor case

Band filling correction: V_{O}^0 in ZnO

vasp_correction_bandfilling-test.bat

python vasp_correction_bandfilling.py BF .¥Defect¥ZnO-VO

Adjust E_{FO} so as to find appropriate E_{V} and E_{C}

Default E_{FO} is 0.1 eV, which seems to be ok

Output:

Total weight: 1.0

Total Ne : 141.0

NELECT : 282.0

Count spin degeneration of 2:

Total Ne : 282.0

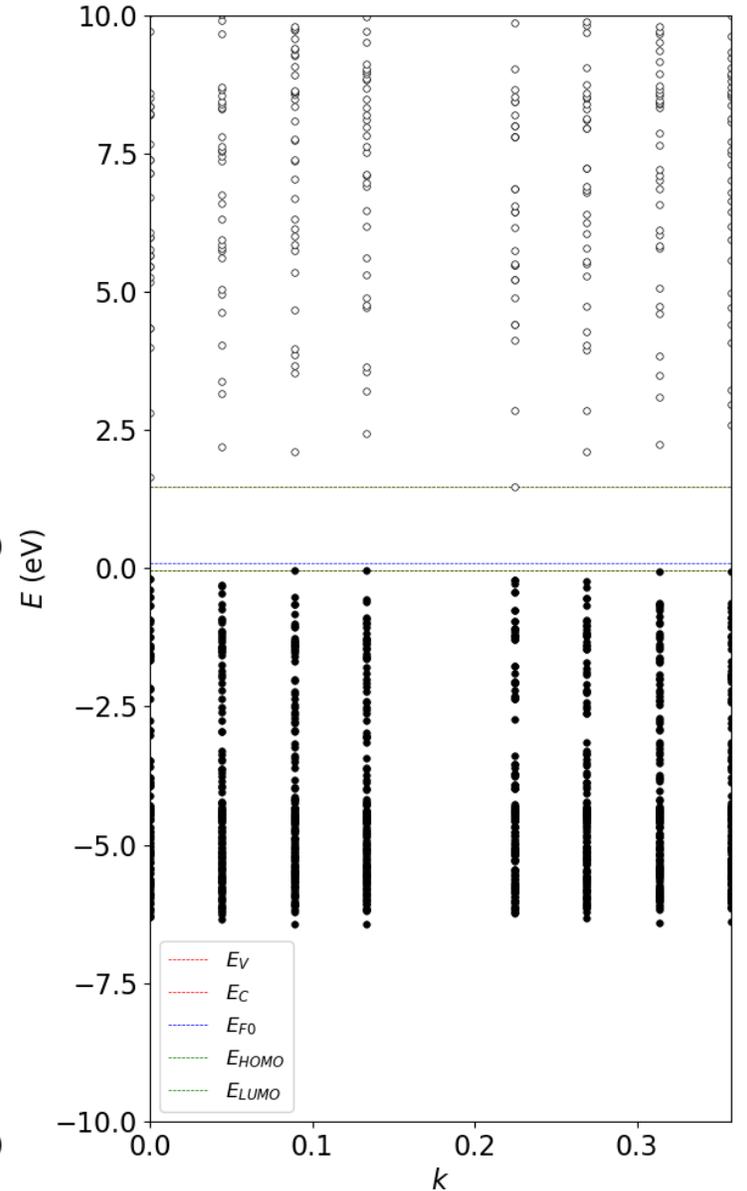
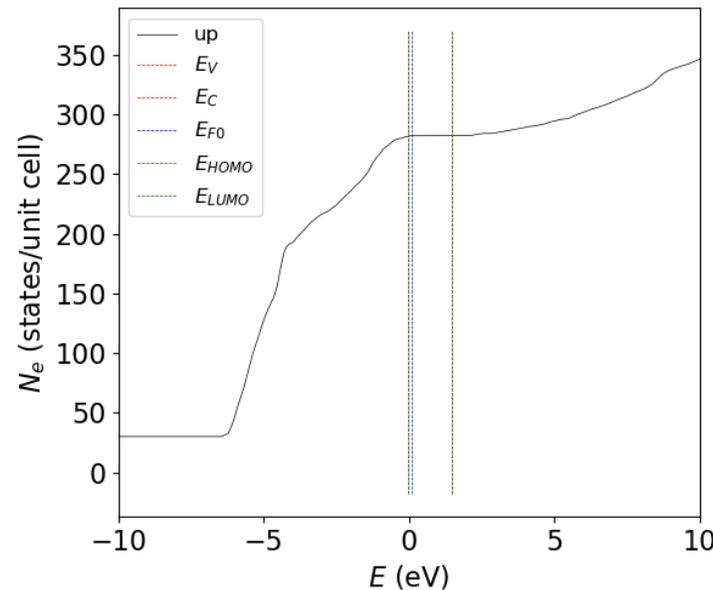
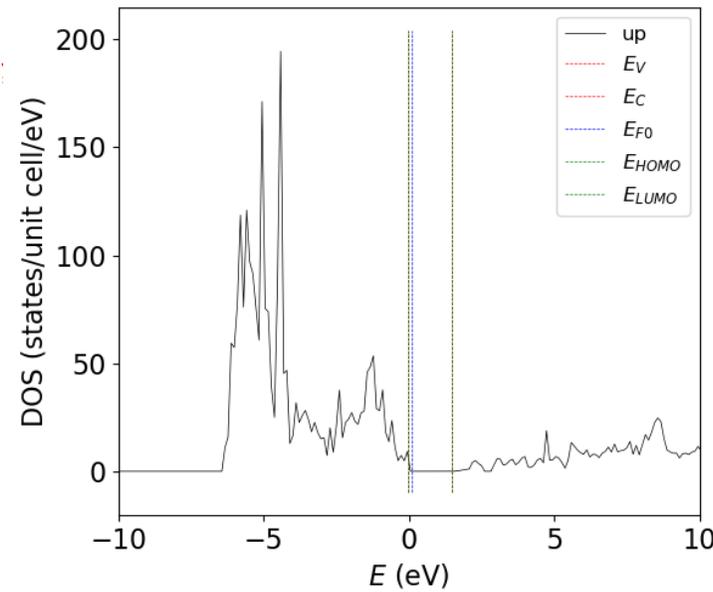
dEh = 0 eV

dEe = 0 eV

Count spin degeneration of 2:

dEh = 0 eV

dEe = 0 eV



Band filling correction: V_{O}^+ in ZnO

vasp_correction_bandfilling-test.bat

python vasp_correction_bandfilling.py BF .¥Defect¥ZnO-VOp1¥SCF

Adjust E_{F0} so as to find appropriate E_{V} and E_{C}

Default E_{F0} is 0.1 eV, which seems to be ok

Output:

Total weight: 1.0

Total Ne : 140.50000003125

NELECT : 281.0

Count spin degeneration of 2:

Total Ne : 281.0000000625

dEh = 0.00235026 eV

dEe = 0.000292896 eV

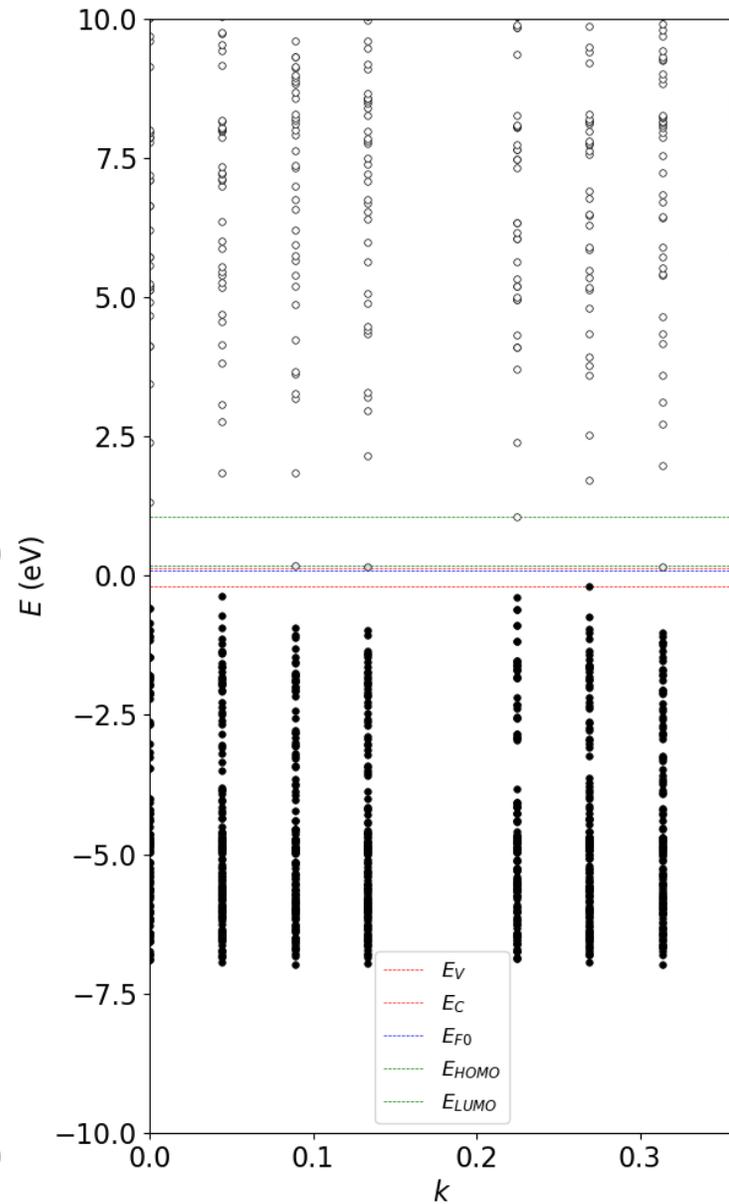
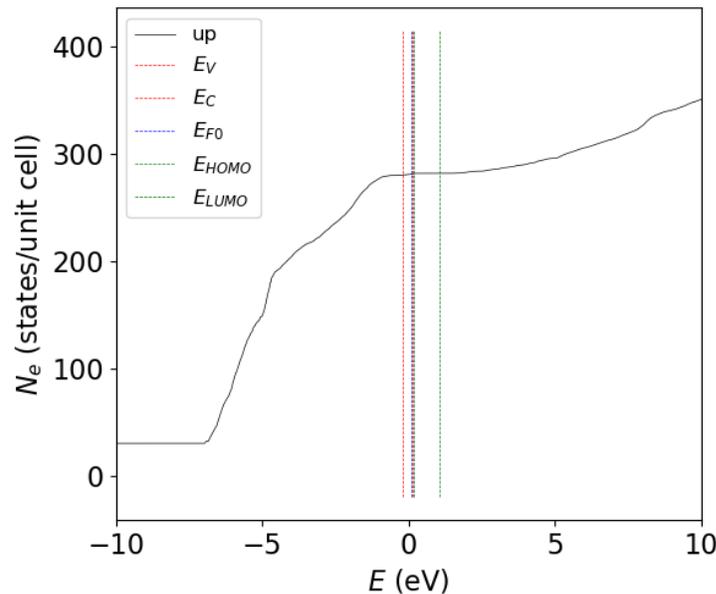
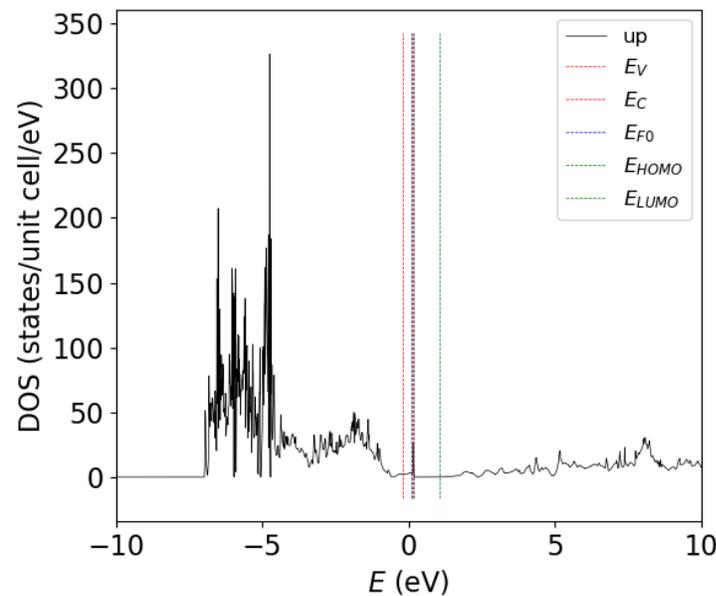
dEtot = 0.00264315 eV

Count spin degeneration of 2:

dEh = 0.00470051 eV

dEe = 0.000585792 eV

dEtot = 0.00528631 eV



Band filling correction: Ga_{Zn}^0 in ZnO

vasp_correction_bandfilling-test.bat

python vasp_correction_bandfilling.py BF .%Defect%ZnO-GaZn%

Adjust E_{F0} so as to find appropriate E_V and E_C

Default E_{F0} is 0.1 eV, but the estimated E_V seems in CB

Try $E_{F0} = -2.2$ eV

vasp_correction_bandfilling-test.bat -2.2

Output:

Total weight: 1.0

Total N_e : 139.49999999999997

NELECT : 279.0

Count spin degeneration of 2:

Total N_e : 278.99999999999994

dEh = 0 eV

dEe = 0.662459 eV

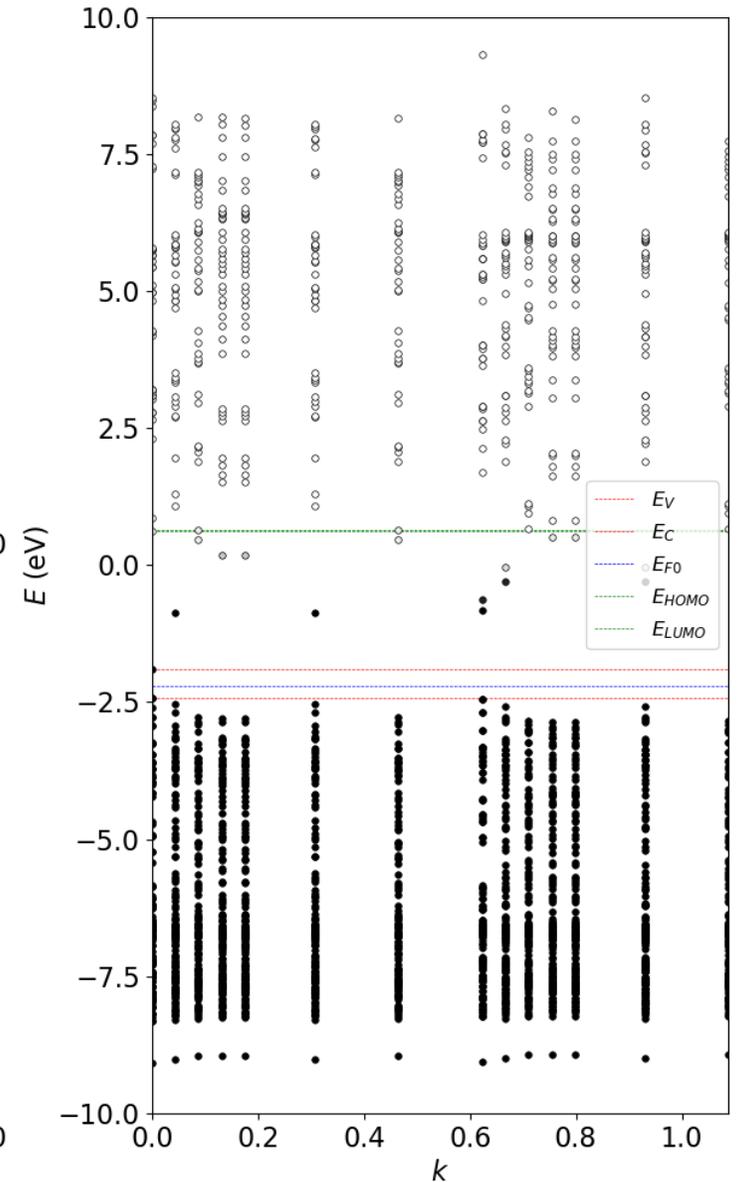
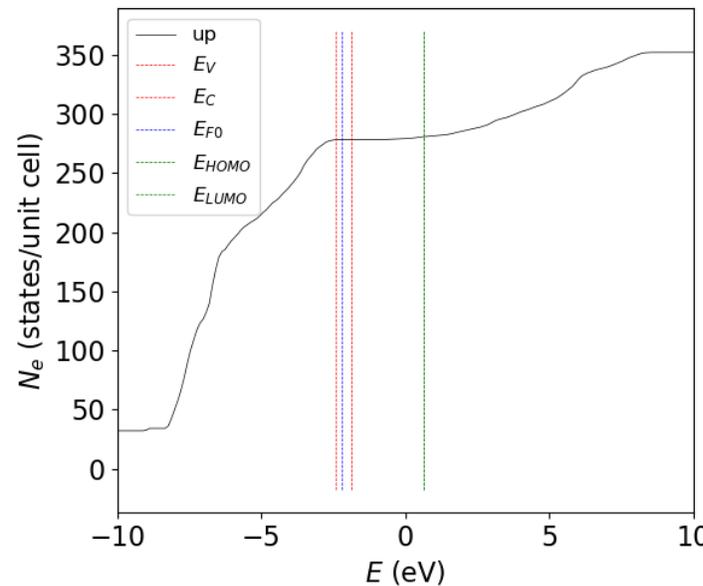
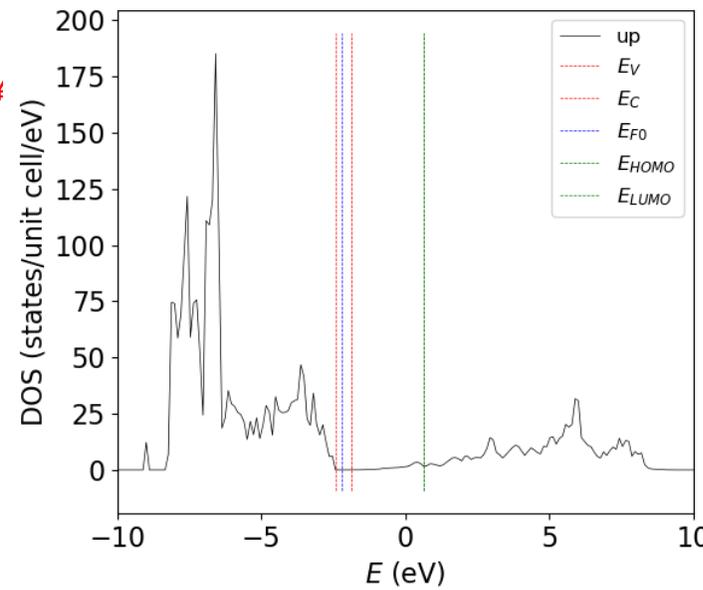
dEtot = 0.662459 eV

Count spin degeneration of 2:

dEh = 0 eV

dEe = 1.32492 eV

dEtot = 1.32492 eV



欠陥電荷/双極子相互作用 (image charge correction)

欠陥間の静電・分極相互作用 image charge (charge/dipole) correction

原因: 欠陥が周期的に配列しているため、
欠陥の電荷・電気双極子同士の静電相互作用により過剰なエネルギーが発生する

対応: 欠陥の電荷・電気双極子同士の静電相互作用を計算し、差し引く

方法:

- ・ VASPでLDIPOL=Tで計算すると、OUTCARには補正するためのパラメータが出力される
注: 立方晶近似
- ・ **pydefectを使う** (Y. Kumagai and F. Oba, Phys. Rev. B 89 (2014) 195205)
FNV correction (C. Freysoldt, J. Neugebauer, and C. G. Van de Walle, Phys. Rev. Lett. 102, 016402 (2009))

Charge/dipole correction: V_{O}^+ in ZnO

vasp_correction_charge-test.bat

```
python vasp_correction_charge.py mode CAR_dir(IDIPOL) CAR_dir(Dielectric constant)
```

Provide CAR_dir for dielectric constant calculation

```
python vasp_correction_charge.py .¥Defect¥ZnO-VOp1¥SCF .¥Defect¥ZnO¥Phonon
```

Output:

```
*** Read OUTCAR from [.¥Defect¥ZnO¥Phonon¥OUTCAR]
```

Information in OUTCAR:

```
ISPIN : 1
```

```
LEPSILON: T
```

```
e(tot): | 7.931    0    0|
        |  0  7.931    0|
        | -0    0  7.351|
```

```
trace average: 7.73768
```

(cut)

```
*** Read IDIPOL information from [.¥Defect¥ZnO-VOp1¥SCF¥OUTCAR]
```

(cut)

```
Ecorr(charge)=q^2 * aM / L      : 2.52695 eV
```

```
NELECT = 281.0
```

```
nVe    = 282.0
```

```
q      = 1.0
```

```
eps(tot)= 7.737676333333333 e0
```

```
L      = 7.23325925567137 A
```

```
aM     : 18.2781 eV*A
```

```
(1/3) * q^2 * aM / L / eps(tot,av): 0.217718 eV
```

**FNV補正の精度が高いため、
pydefectを使うことを推奨**

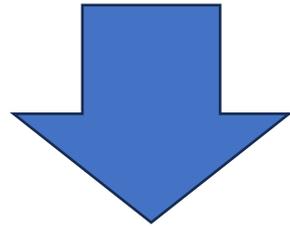
欠陥生成エネルギー

欠陥生成エネルギー (点欠陥) の一般式:

$$E^f_{D,q}(E_F, \mu) = E_{D,q} - E_0 - n_{Cs}\mu_{Cs} - n_{Cl}\mu_{Cl} + q(E_F - E_{VBM}^0)$$

↑ ここまでの補正を入れて、これが計算できた

- 安定境界を選ぶと、 μ_{Cs}, μ_{Cl} が決まる

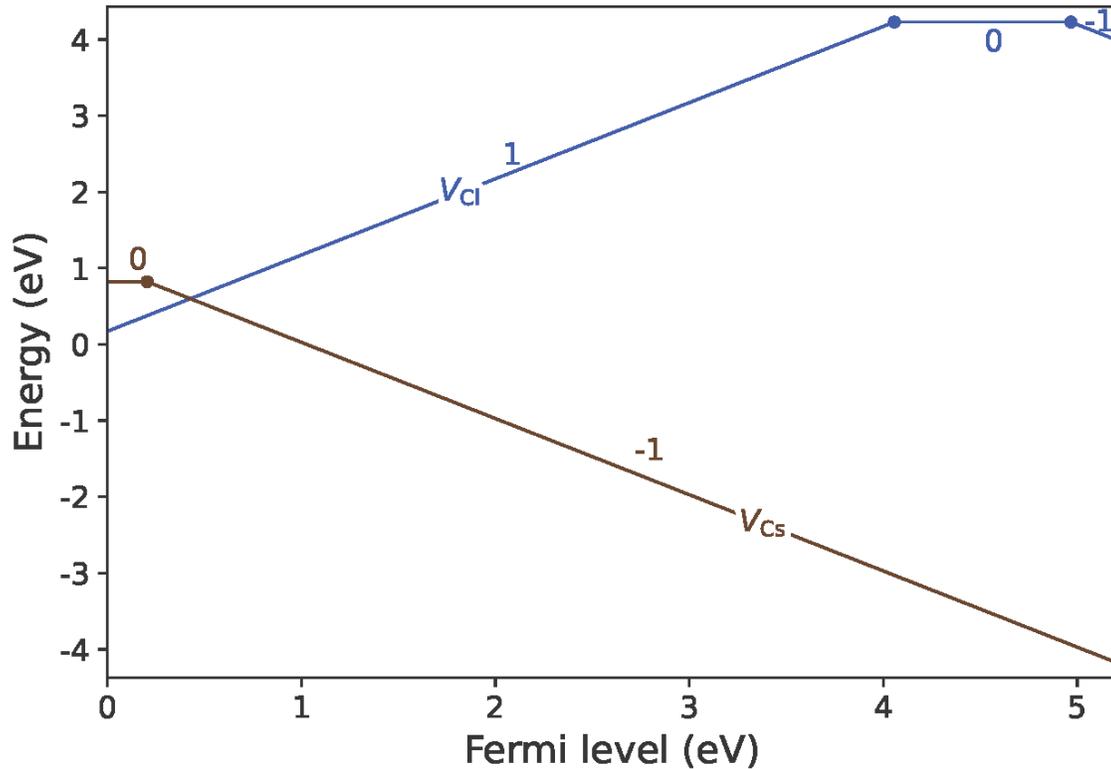


D^q のそれぞれについて、
 $E^f_{D,q}(E_F, \mu)$ を E_F の関数としてプロットする

欠陥生成エネルギー vs E_F

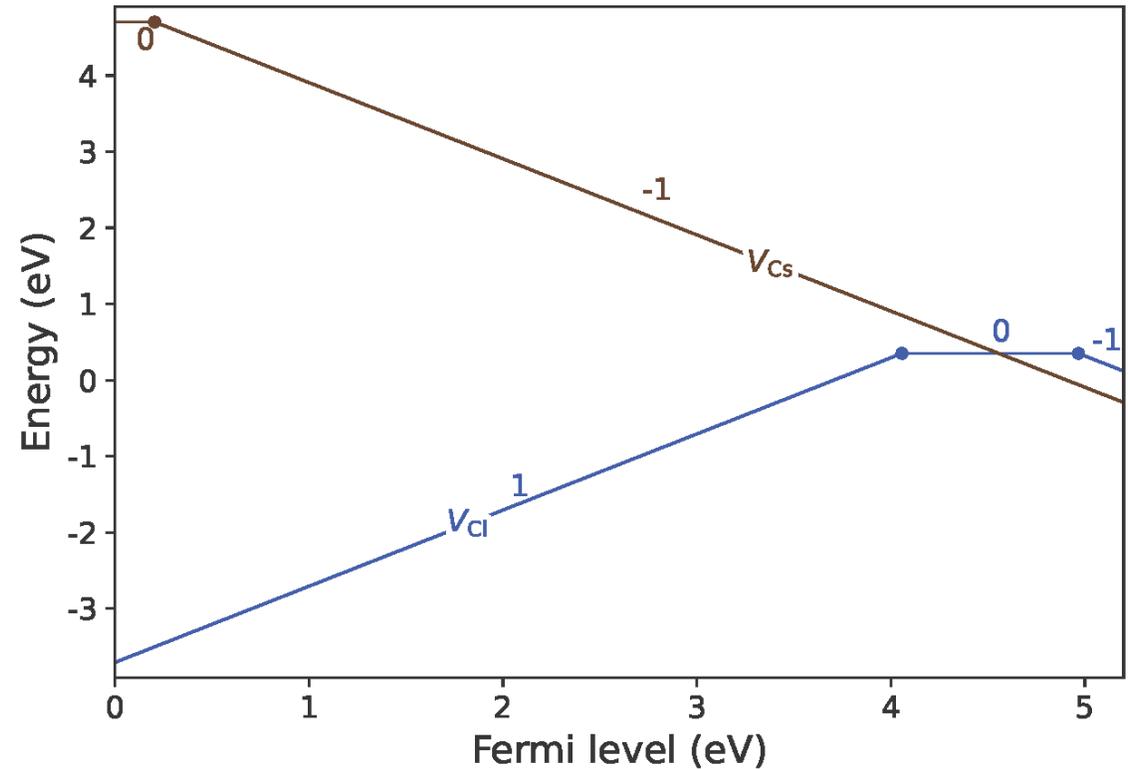
point A

CsCl



point B

CsCl



異なる安定境界では化学ポテンシャルが異なる

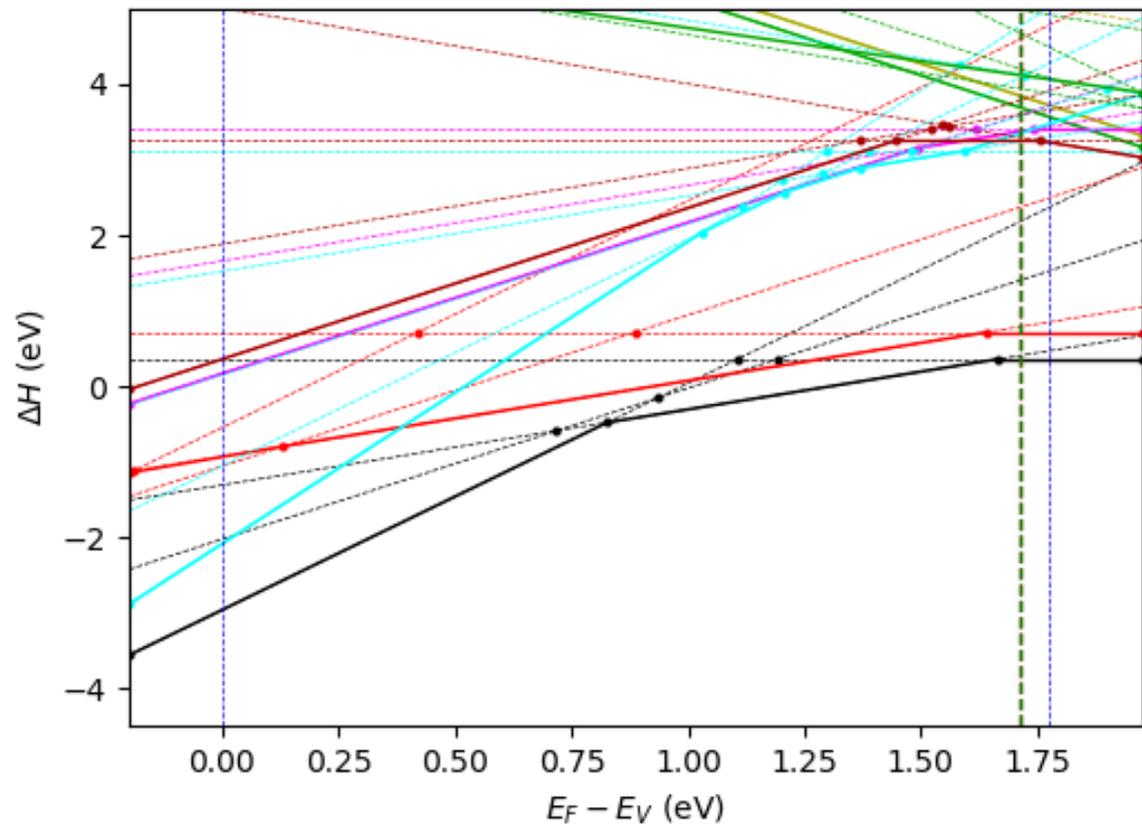
$$E_{D,q}^f(E_F, \mu) = E_{D,q} - E_0 - n_{Cs}\mu_{Cs} - n_{Cl}\mu_{Cl} + qE_F$$

化学ポテンシャルの違いは、欠陥種ごとの全直線を平行移動させる

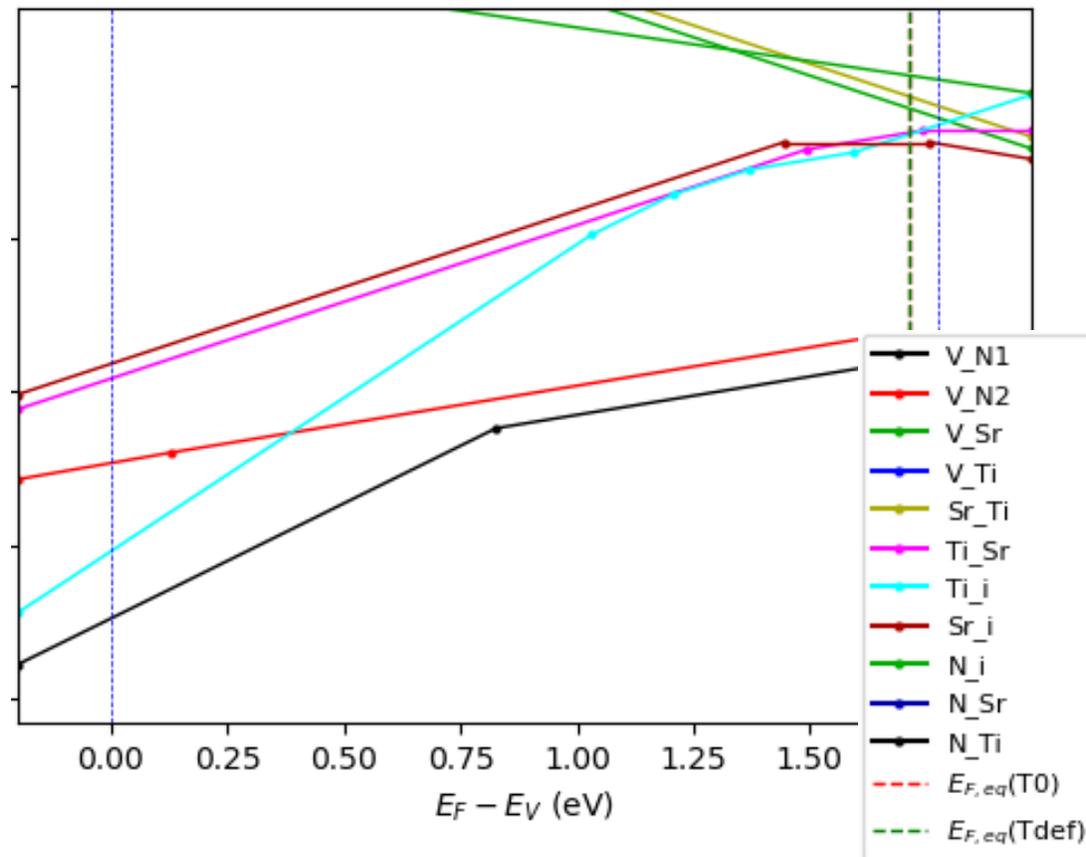
欠陥生成エネルギー図の読み方

Point Aにおける欠陥生成エネルギー: SrTiN₂の例

python vasp_defect.py EF max
全ての $\Delta H(E_F)$ をプロット



python vasp_defect.py EF min
各欠陥について
最低の $\Delta H(E_F)$ だけをプロット



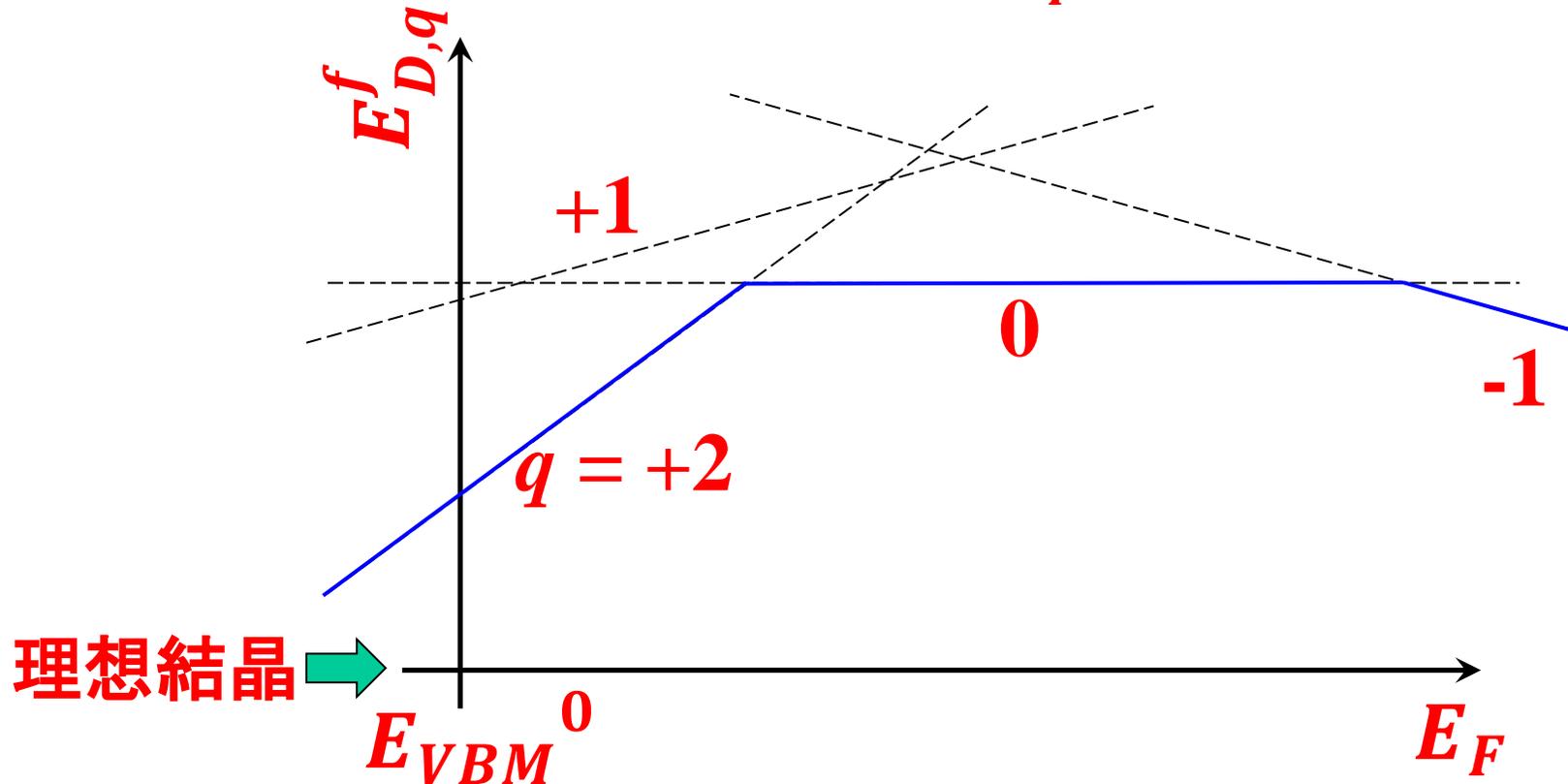
一般に、 E_F において最低の $E_F^{D,q}$ の荷電状態のみプロットする
=> エネルギーの高い荷電状態が存在しないわけではない

$E^f_{D,q}(E_F)$ 図の読み方

$$E^f_{D,q}(E_F, \mu) = E_{D,q} - E_0 - n_{Zn}\mu_{Zn} - n_O\mu_O + q(E_F - E_{VBM}^0)$$

E_{VBM}^0 (理想結晶の E_{VBM}) を原点にとる

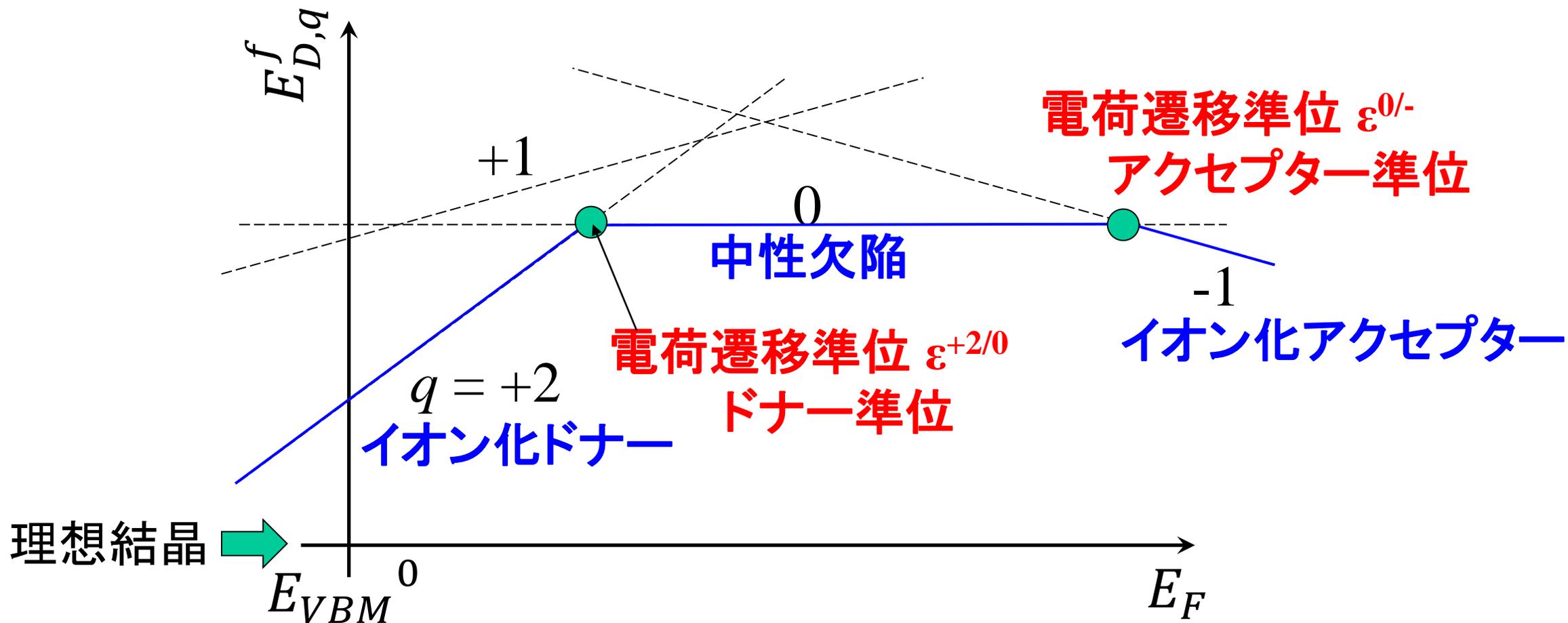
$$E^f_{D,q}(E_F, \mu) = E^f_{D,q}(0, \mu) + \underline{qE_F} \text{ 勾配 } \frac{\partial E^f_{D,q}}{\partial E_F} = q$$



電荷遷移準位

E_F が上昇すると、電子が欠陥に補足され、欠陥の電荷が変化する:

電荷遷移準位 $\epsilon^{q1/q2}$



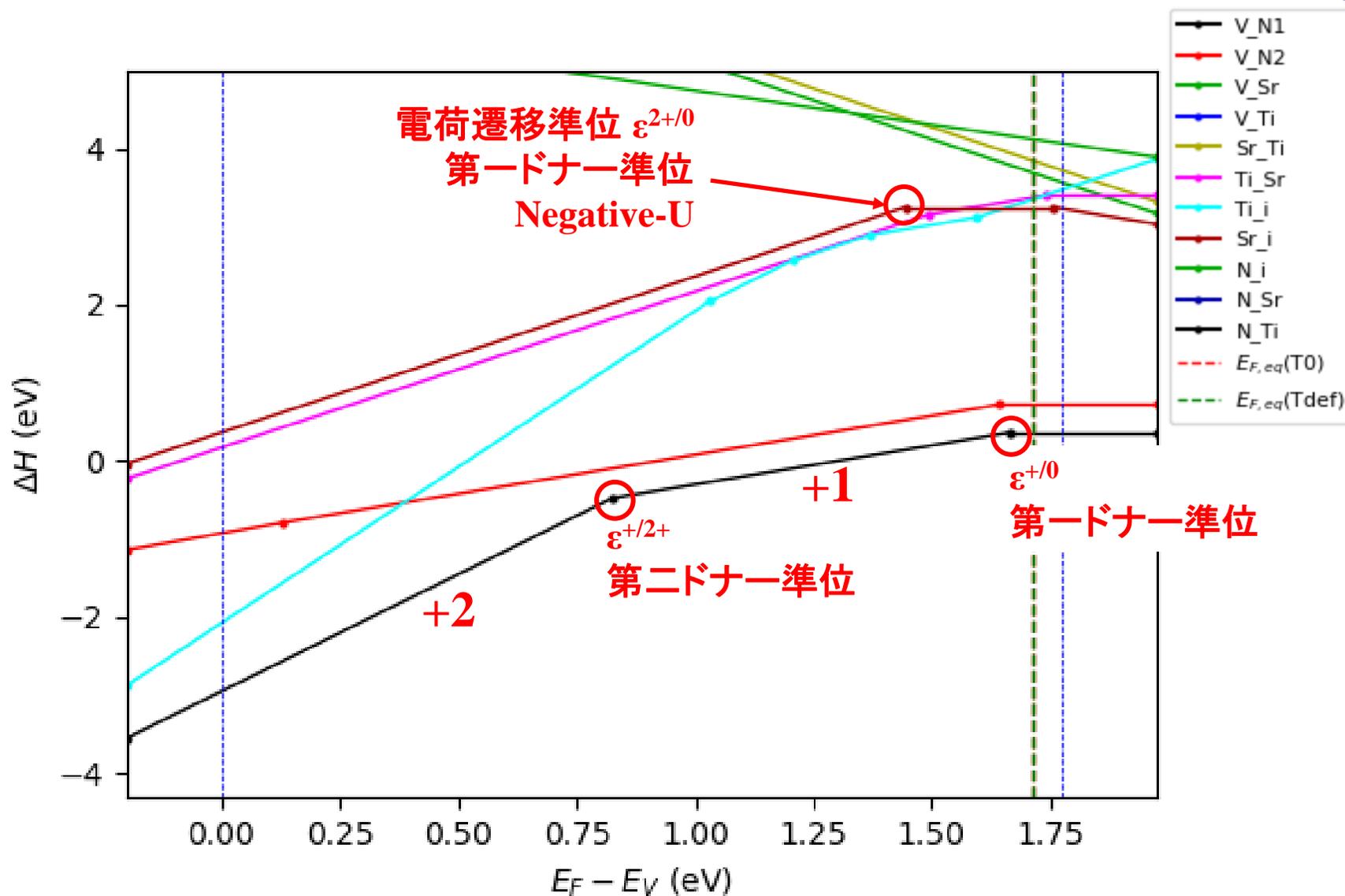
注意: ドナーとアクセプター

直観的なアクセプター、ドナーの理解と定義が異なることがあるので注意

アクセプター: 中性状態から電子を捕獲できる欠陥
ドナー: 中性状態から電子を放出できる欠陥

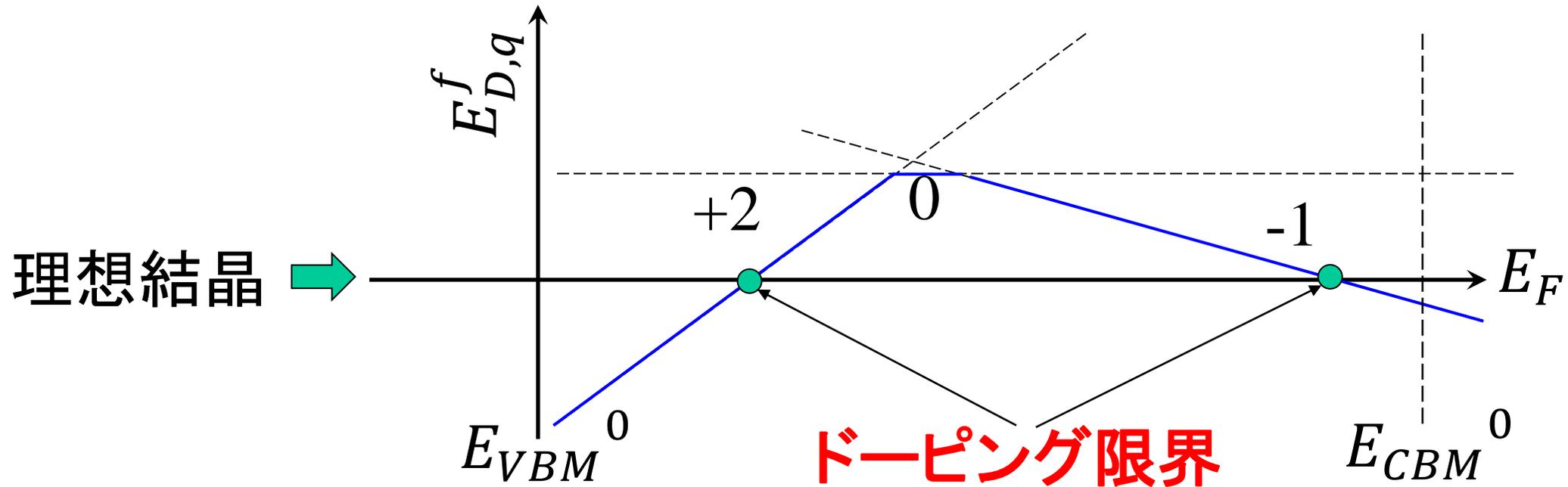
例: 電界効果型トランジスタで問題になる
電子捕獲準位は、 E_{CBM} 近傍にあるが、
アクセプター型欠陥

安定境界 Point Aにおける欠陥生成エネルギー: SrTiN₂の例



半導体の $N_e(T)$ から求めたドナー準位が、電荷遷移準位に正確に対応するかは要確認

負の $E_{D,q}^f(E_F)$: ドーピング限界

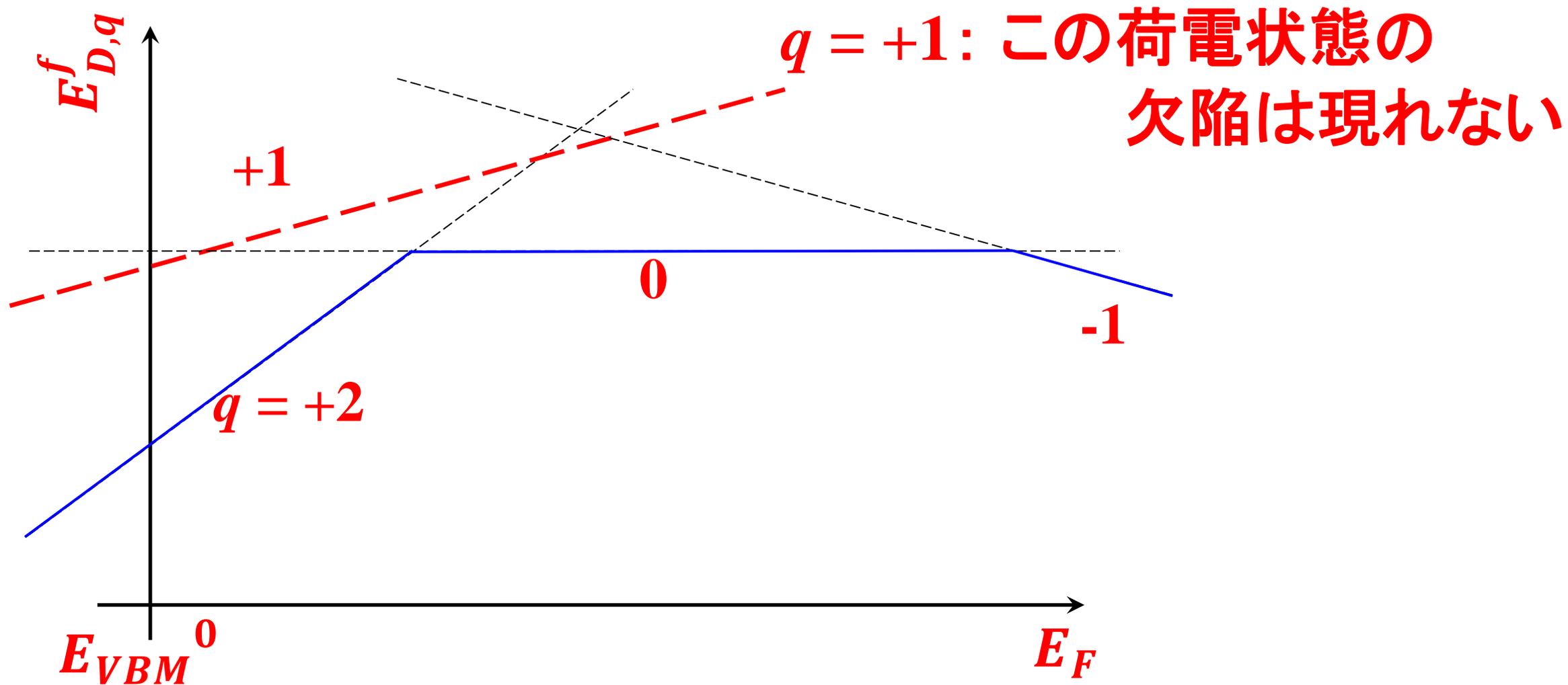


注意: 定量的な意味ではないが、実際のドーピング限界はこの範囲に入る

負の $E_{D,q}^f$ に対しては $[D^q] > [\text{ideal}]$ (理想結晶のサイト数) と大きな数値になる。
=> もはや欠陥では無くなり、別の化合物?

E_F は $E_{D,q}^f(E_F) \sim 0$ となる E_F の範囲内にピンされる

$E_{D,q}^f(E_F)$: Negative-U欠陥



欠陥の電荷

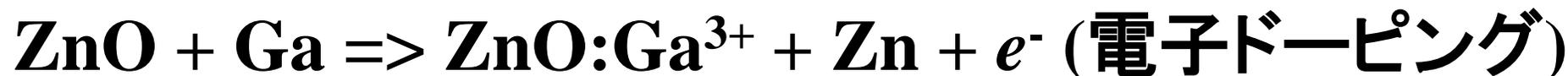
欠陥の電荷は、**ホストイオンの電荷が基準**

例: Zn^{2+} をGaで置換

- Zn^{2+} を Ga^{2+} で置換した欠陥 : Ga_{Zn}^0

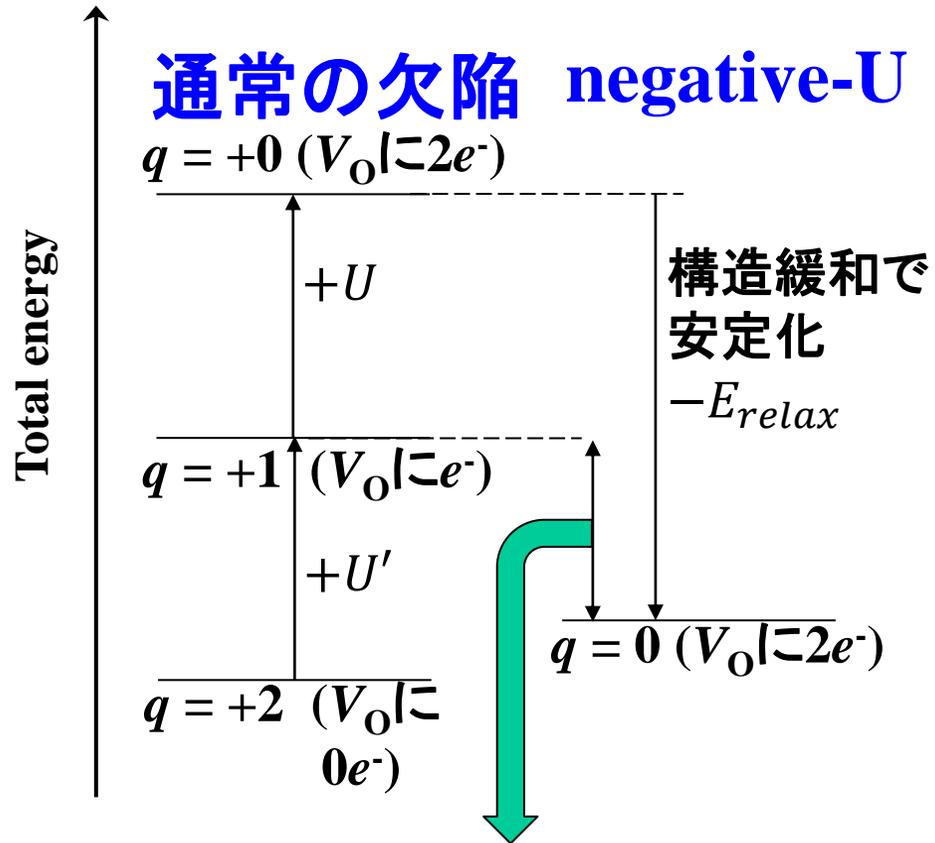


- Zn^{2+} を Ga^{3+} で置換した欠陥 : Ga_{Zn}^+



- O^{2-} を除いた酸素欠損に電子が2つ捕獲されている : V_{O}^0
- O^{2-} を除いた酸素欠損に電子が1つ捕獲されている : V_{O}^+
- O^{2-} を除いた酸素欠損に電子が捕獲されていない : V_{O}^{2+}

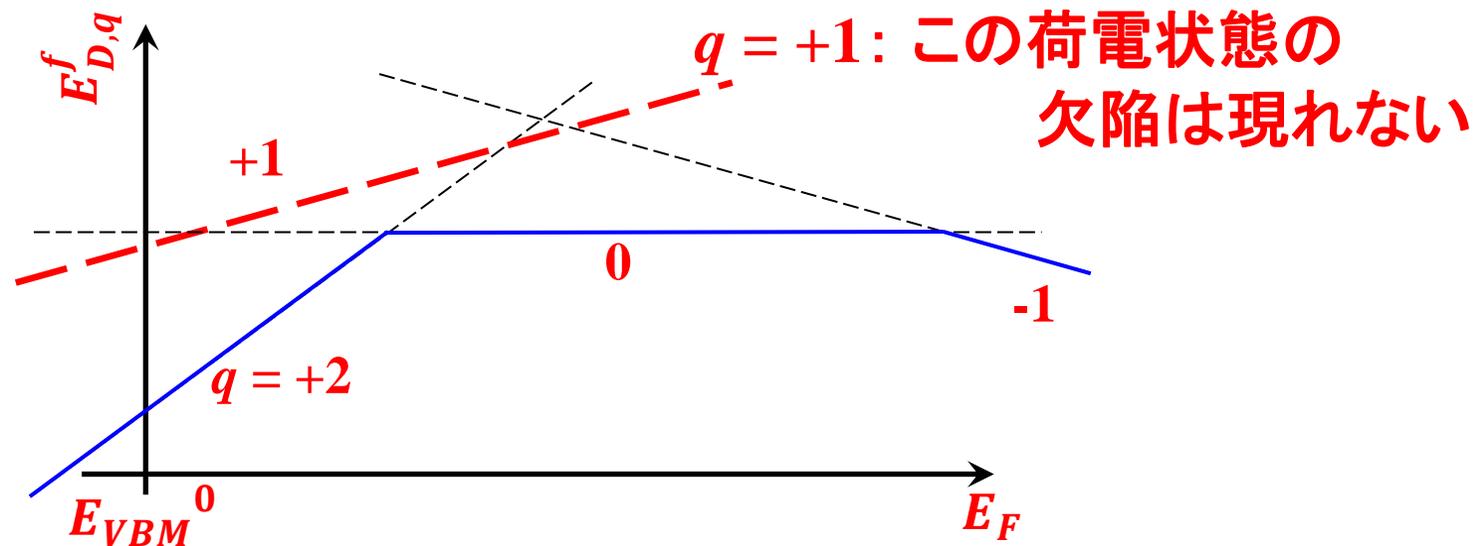
$E_{D,q}^f(E_F)$: Negative-U欠陥: ZnOの V_O を例に



$U_{eff} = U - E_{relax} < 0$
 $q = +1$ より $q = 0$ の方が安定になる

- V_O^+ に電子が捕獲されて V_O^{2-} になると、電子数が $1e^-$ から $2e^-$ に増えるため、電子間反発エネルギーは高くなる: $U > 0$
- 電子数が $1e^-$ から $2e^-$ に増えると、周囲のイオンとの静電相互作用が大きくなり、大きな構造緩和が起こる: V_O^0 は大きく構造緩和し、構造緩和エネルギー E_{relax} も大きい
- $U_{eff} = U - E_{relax} < 0$ の場合、電子の捕獲により、捕獲前の状態より全エネルギーが安定化する:
捕獲前の状態がエネルギー的に最安定ではなくなる:

Negative-U



半導体の欠陥計算

欠陥濃度

Priority Target: 欠陥生成エネルギー

欠陥 D^q (q : 欠陥電荷) の生成エネルギー

$$E^f_{D,q}(E_F, \mu) = E_{D,q} - E_0 - n_{Zn}\mu_{Zn} - n_O\mu_O + q(E_F - E_{VBM}^0)$$

$E^f_{D,q}(E_F, \mu)$: Fermi準位 E_F の時の D^q の生成エネルギー

$E_{D,q}$: 帯電欠陥 D^q の全エネルギー

E_0 : 理想結晶の全エネルギー

μ_{Zn}, μ_O : Zn, O の化学ポテンシャル

E_{VBM}^0 : エネルギー原点。一般的に理想結晶の価電子帯端 (VBM) にとる

平衡Fermi準位、平衡欠陥濃度、キャリア密度を計算するには？

統計力学

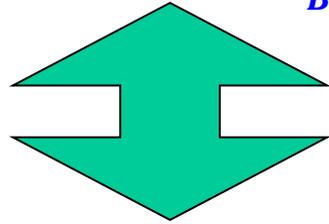
統計力学の基礎

複数の粒子を $\{N_i\}$ 個含む系の状態 i における**全エネルギーを E_i** とすると、

平衡状態における出現確率は $\exp\left(-\frac{E_i}{k_B T}\right)$ に比例する: **正準分布**

$$p(E_i) = \frac{\exp\left(-\frac{E_i}{k_B T}\right)}{Z} = \exp\left(-\frac{E_i - \mu}{k_B T}\right)$$

仮定は無い。一般的な場合に成立



Fermi-Dirac分布: $f(e_i) = \frac{1}{\exp[(e_i - \mu)/k_B T] + 1}$

Bose-Einstein分布: $f(e_i) = \frac{1}{\exp[(e_i - \mu)/k_B T] - 1}$

—電子エネルギー e_i による分布
(粒子統計を考慮しないといけない)

統計力学講義資料

<http://conf.msl.titech.ac.jp/Lecture/StatisticsC/index.html>

2023年度 Q3 統計力学(C) 講義資料 (神谷担当分)

update:

- 統計分布関数 全講義まとめ 11/25 8:19 更新 [2023QuantumStatistics-All2.pdf](#)
- 2023/11/24 第14回講義資料 2023/10/21 質問への回答 11/25 8:19 更新 [0231125QandA.pdf](#)
- 2023/10/24 質問への回答 10/27 07:01 アップロード [20231027LastQuestions.pdf](#)
- 統計分布関数 全講義まとめ 10/25 13:19 アップロード
- 2023/11/24 第14回講義資料 2023/10/21 質問への回答 11/24 12:11 アップロード
- 2023/10/24 第06回講義資料 10/25 13:19 更新 [20231025QuantumStatistics2](#)
- 2023/10/17 第05回講義資料 10/18 03:56 更新 [20231018QuantumStatistics1](#)
- 2023/10/13 第04回講義資料 10/13 05:40 更新 [20231013-CanornicalTheory.p](#)
- 2023/10/10 第03回講義資料 10/13 10:08 更新 [20231013-BoltzmannDistribut](#)
- 2023/10/06 第02回講義資料 10/07 03:58 更新 [20231007-MaxwellDistributio](#)
- 2023/10/03 第01回講義資料 10/04 09:43 更新 [20231004StatPhys01-Thermo](#)

10/27 (火) 第7回	量子統計力学の基礎 I (Fermi-Dirac分布、Bose-Einstein分布)	20201027-07-QuantumStatistics1.pdf
10/30 (金) 第8回	量子統計力学の基礎 II (正準分布) 理想Bose気体 (固体の比熱 (Einsteinの比熱式))	20201030-08-BEStatistics1-2.pdf
11/6 (金) 第9回	休講	
11/10 (火) 第10回	理想Bose気体、固体の比熱 (Debyeの比熱式)、光子と熱輻射 (Planck分布)	20201110-09-BEStatistics2-2.pdf
11/13 (金) 第11回	理想Fermi気体、金属中の電子	20201113-11-FDStatistics1.pdf
11/17 (火) 第12回	金属中の電子 (熱電子放出) 半導体中の電子、Fermi準位、真性半導体	20201118-12-FDStatistics2.pdf
11/20 (金) 第13回	半導体中の電子、Fermi準位、ドーピング 正準統計: 二準位系スピン系の磁化率	20201122-13-FDStatistics3Spin.pdf
11/24 (火) 第14回	正準統計: 多準位系スピン系の磁化率 Bose凝縮、超伝導	20201124-14-SpinBoseCondensation.pdf Eu2+-M.xlsx
12/1 (火) 第15回	試験	

欠陥濃度分布

$E_{D,q}^f(E_F, \mu)$ に対して正準分布を適用

$$\text{正準分布: } p(E_i) = \frac{\exp\left(-\frac{E_{D,q}^f(E_F, \mu)}{k_B T}\right)}{Z}$$

$$\text{分配関数: } Z = \sum \exp\left(-\frac{E_{D,q}^f(E_F, \mu)}{k_B T}\right)$$

欠陥計算ではすべての欠陥を考慮することはできないが、
それ以外の欠陥の出現確率は無視できるとして、
考慮している欠陥 D^q について和をとる

欠陥濃度分布: 点空孔

点空孔 V_M だけを考えるとき:

$$Z = \sum \exp\left(-\frac{E^f_{V_M,q}(E_F,\mu)}{k_B T}\right) = \mathbf{1} + N_{M \text{ site}} \exp\left(-\frac{E^f_{V_M,q}(E_F,\mu)}{k_B T}\right)$$

$N_{M \text{ site}}$ はモデル単位格子の M 原子のサイト数

注: Z の和には “欠陥が無い状態 $E^f_{V_M,q}(E_F,\mu) = \mathbf{0}$ ” も含まれるので、
1の項がでてくる

$$\text{欠陥濃度: } [V_M^q] = \frac{1}{V_{\text{super cell}}} \frac{\sum_{V_M,q} \exp\left(-\frac{E^f_{V_M,q}(E_F,\mu)}{k_B T}\right)}{\mathbf{1} + N_{M \text{ site}} \exp\left(-\frac{E^f_{V_M,q}(E_F,\mu)}{k_B T}\right)} \sim \frac{N_{M \text{ site}}}{V_{\text{super cell}}} \exp\left(-\frac{E^f_{V_M,q}(E_F,\mu)}{k_B T}\right)$$

注: 分子の和には 欠陥 V_M, q だけについて取るので、 $N_{M \text{ site}} \exp\left(-\frac{E^f_{V_M,q}(E_F,\mu)}{k_B T}\right)$ 。

欠陥生成エネルギーが大きければ、**分母は 1 で近似**できる

点空孔はサイトごとに独立なので、それぞれに独立に計算できる

欠陥濃度分布: 格子間欠陥

格子間欠陥 M_i だけを考えるとき:

N_i はモデル単位格子の格子間サイト数

$$\text{欠陥濃度: } [M_i^q] \sim \frac{N_i}{V_{\text{super cell}}} \exp\left(-\frac{E_{M_i,q}^f(E_F, \mu)}{k_B T}\right)$$

- 複数の異なる格子間欠陥を考えるとき
両立しない格子間欠陥がある場合は、**サイト数 N_i の数え方に注意**
- 準安定欠陥を入れる必要はないのか？
本来は全ての準安定・不安定欠陥を入れるのが統計力学の基本
サイト数の数え方を間違えなければ正しい答えが得られるが、難しい

=> 独立事象として扱える安定欠陥だけで計算してしまおう

欠陥濃度分布: 複合欠陥

複合欠陥の場合、異なる配置も考慮してサイト数 $N_{s,D}$ を数える

$$\text{欠陥濃度: } [D^q] \sim \frac{N_{s,D}}{V_{\text{super cell}}} \exp\left(-\frac{E_{D,q}^f(E_F, \mu)}{k_B T}\right)$$

特に問題

- 複合欠陥は、点欠陥とサイトを共有する
- 複合欠陥同士も、同じサイトを共有することが多い

⇒ 欠陥濃度が小さい場合、

異なる欠陥がサイトを共有する事象は無視できるとする

(真面目に計算するならMonte Carlo法などを使う?)

自作プログラムの仮定

- 異なる欠陥がサイトを共有する事象は無視できるとする
- 欠陥サイト数はマニュアルで入力
- 分配関数 Z の計算では近似は入れない

input.xlsx

Version2	Atom	Site	q	dS/kB	N0
	V	Cl1	0	5	64
	V	Cl1	-1	5	64
	V	Cl1	1	5	64
	V	Cs1	0	5	64
	V	Cs1	-1	5	64
	V	Cs1	1	5	64

自作プログラムの仮定

欠陥の凍結温度と電子の平衡温度が異なる場合にも対応

例: 1000 Kで合成したが、欠陥は 600 Kで凍結したと仮定。

300 Kにおける電子分布を知りたい

欠陥平衡: 欠陥凍結温度 T_{def}

- それぞれの欠陥 D^q について正準分布を適用し、 $[D^q]$ を計算
- 欠陥電荷 q について和を取り、全欠陥濃度 $[D]$ を計算

電子平衡: 実測温度 T_0

- 全欠陥濃度 $[D]$ は T_{def} の平衡濃度に固定
- Fermi-Dirac分布より、 $[D^q]$ を計算

電子平衡: 平衡Fermi準位

Fermi-Dirac分布: $f(e_i) = \frac{1}{\exp[(e_i - \mu)/k_B T] + 1}$

電子の化学ポテンシャル $\mu = \text{Fermi準位 } E_F$ は
全粒子数の条件から決まる

=> 半導体の場合、電荷中性条件に置き換えられる

$$n_e + N_A^- = n_h + N_D^+ \Rightarrow E_F$$

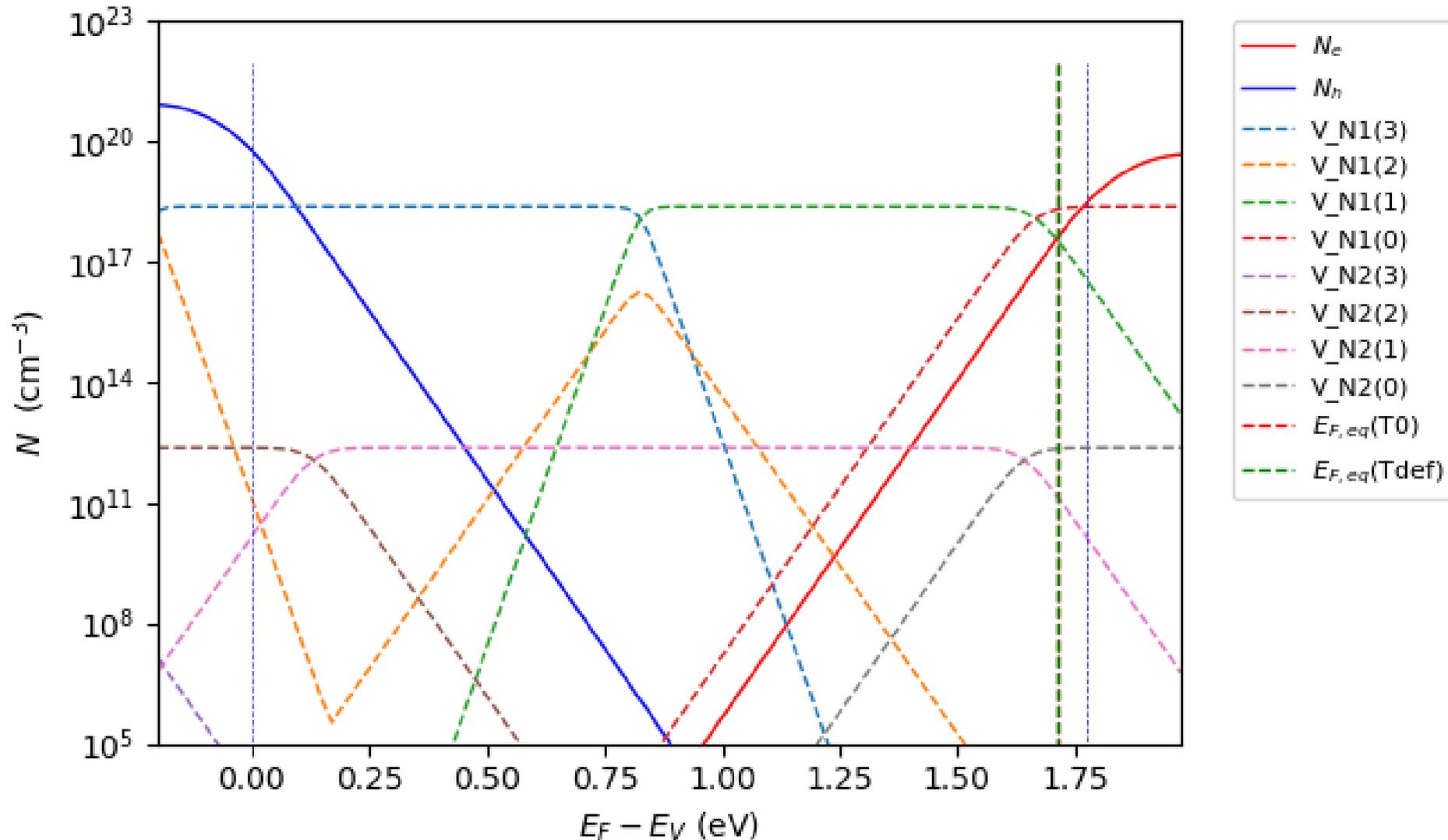
今回の場合、様々な帯電欠陥があるので、すべての和を取る

$$-n_e + n_h + \sum_{D,q} q[D^q] = 0 \Rightarrow \mu(E_F)$$

Point Aにおける欠陥濃度: SrTiN₂の例

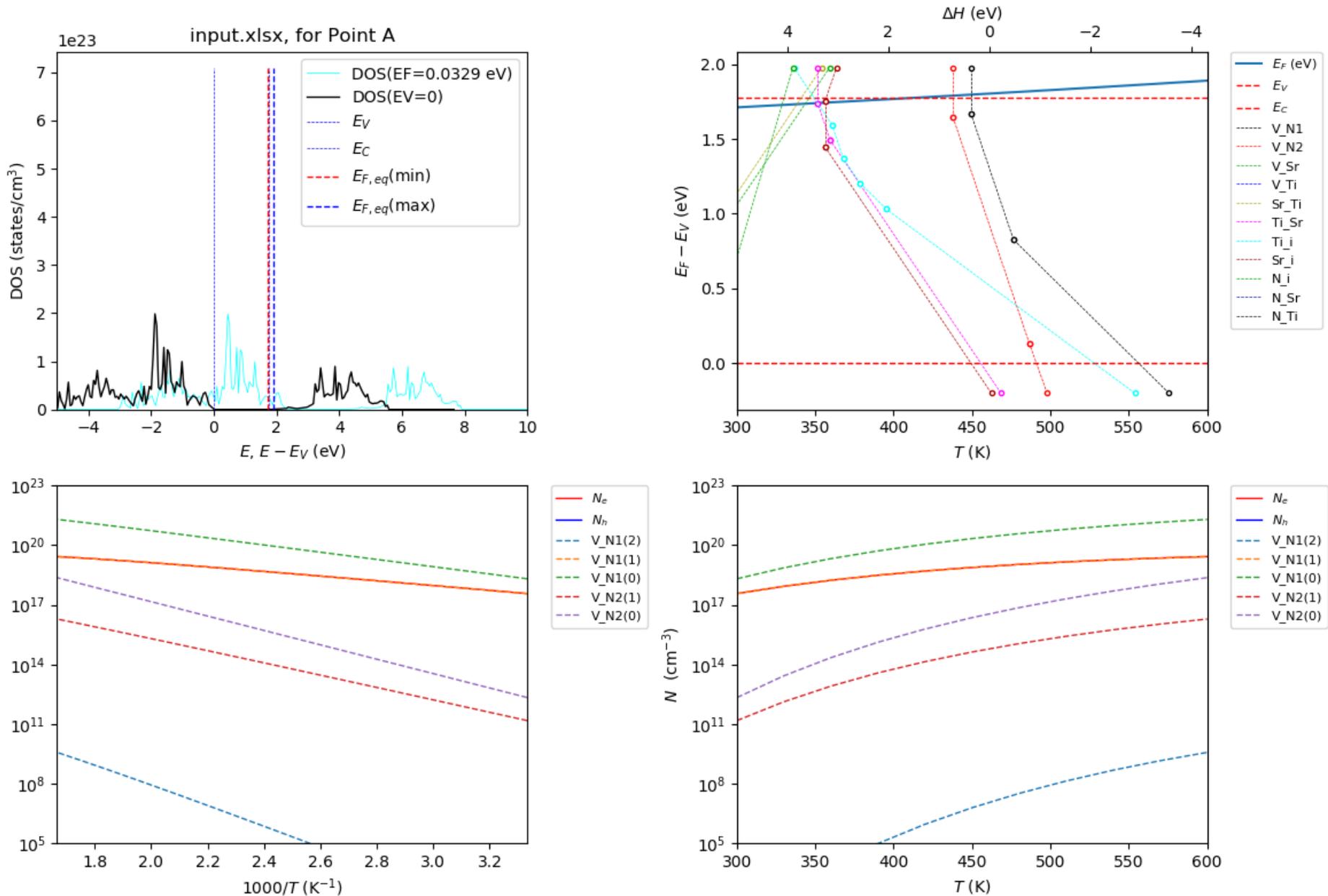
python vasp_defect.py EF min

$$\Delta S = 5k_B, \Delta G(E_F) = \Delta H(E_F) - T\Delta S, N_d(E_F) = N_{site,d} \exp\left[-\frac{\Delta G_d(E_F)}{k_B T}\right] / \sum_{d \text{ at site}} \exp\left[-\frac{\Delta G_d(E_F)}{k_B T}\right]$$



Point Aにおける欠陥濃度等の温度依存性: SrTiN₂の例

python vasp_defect.py T

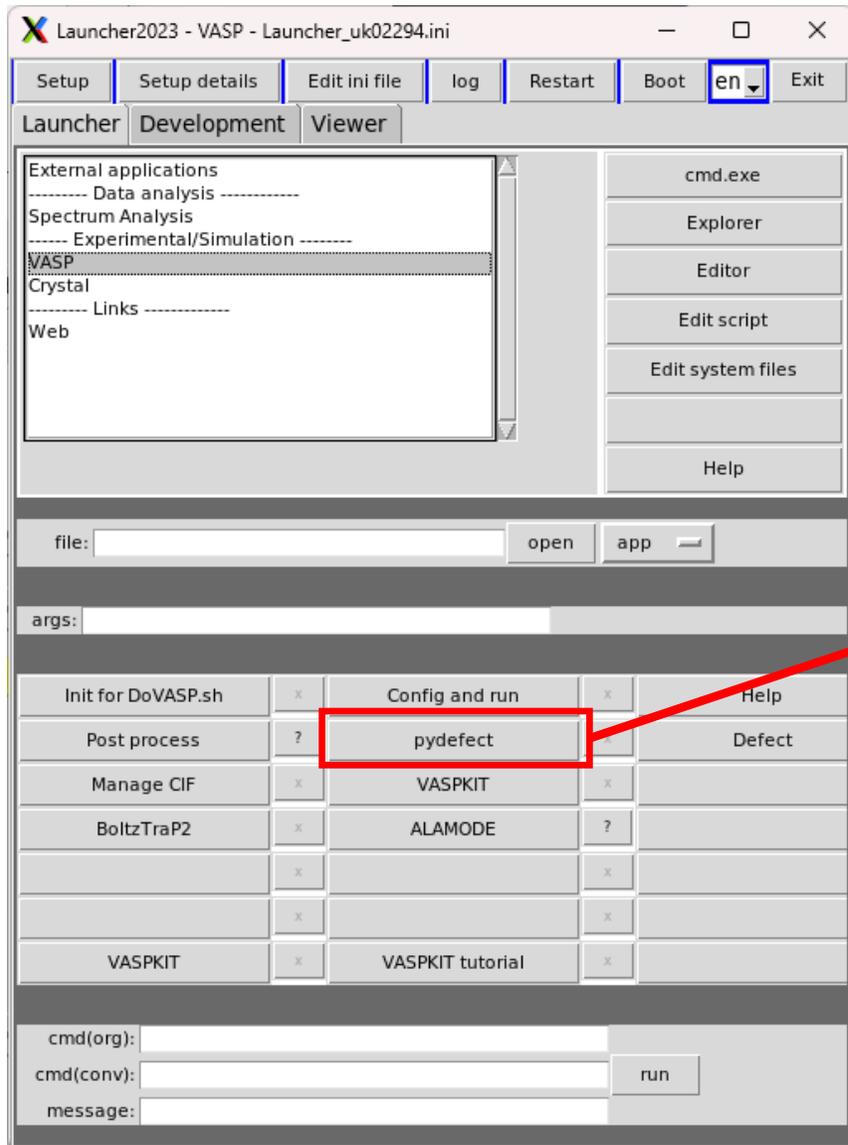


キャリア濃度、欠陥濃度の計算 vlauncher/defect

流れ：

1. pydefectタブ: pydefectの計算結果defect_energy_summary.jsonからinput.xlsxをつくる
2. plot dH and Nタブ: input.xlsxを表示 (vasp_defect.py)

vlauncher: VASP=>Defect



計算のルートディレクトリ、
CsClなどを選ぶ

vlauncher: VASP=>Defect

pydefect Plot dH and N

pydefect JSON: /gs/bs/tga-kklab/user/tkamiya/CsCl/defect/defect_energy_summa

CONTCAR (ideal, relaxed structure): b/user/tkamiya/CsCl/unitcell/structure_o

POSCAR (defect):

input.xlsx: /gs/bs/tga-kklab/user/tkamiya/CsCl/input.xlsx

Step: (1) 'Make in input.xlsxを作成: 'Edit input.xlsx', (3) 'load summary file'

View pydefect JSON Make input.xlsx Edit input.xlsx edit log file

edit summary file load summary file

dEcorr: eV dEVBM: eV dE

EV(ideal): eV EC(ideal):

vasprun.xml: path app

View vasprun.xml

post process explorer cmd VESTA Chesta (Windows)

Defect/defect_energy_summary.jsonが選択される

理想構造の緩和済 CONTCAR unitcell/structure_opt/CONTCAR
が選択される

欠陥supercellのPOSCAR: defect以下のPOSCARを適当に選ぶ

input.xlsxが選択される (計算ルートディレクトリの下)

流れ:

1. 欠陥supercellから欠陥サイトの数を計算し、input.xlsxに入力
理想構造のPOSCARは欠損、格子間などの特定に使用

↑: Make input.xlsxの後、確認
(TSUBAMEにはappがない。
Windowsから開く)

input.xlsx

赤枠はdefect_energy_summary.jsonから読み込む

Version2	Atom	Site	q	dS/kB	N0	Ndoped	A	B			dH0	dEVBM	dEpc
	V	Cl1	0	5	62	0	4.305878	4.305878			4.305878	0	0
	V	Cl1	-1	5	62	0	9.396214	9.396214			9.221148	0.011657	0.163408
	V	Cl1	1	5	62	0	0.166152	0.166152			0.033966	-0.03122	0.163408
	V	Cs1	0	5	62	0	4.66922	4.66922			4.66922	0	0
	V	Cs1	-1	5	62	0	4.871655	4.871655			4.652788	0.055459	0.163408
	V	Cs1	1	5	62	0	4.733593	4.733593			4.746544	-0.17636	0.163408

↑Version2: 識別子なので必須

**N0は欠陥supercell内の欠陥サイト数:
input.xlsx作成後に確認すること**

dSは、 k_B で除した欠陥生成エントロピー
不明な場合は 5 ~ 10 程度

Atom: 元素名

Site: サイト名

欠陥名は Atom_{Site} (V_{Cl1} など)

q: 欠陥電荷

N0: 欠陥supercell内の欠陥サイト数

Boltzmann分布から欠陥濃度を計算する際に使用 $N_{\text{defect}} = N0 \exp(-\Delta H_f/kT)$

A, B, ...: 化学ポテンシャル図から取り出した相境界点の化学ポテンシャル

dH0: ΔH_0

dEVBM, dEpc: pydefectによる補正 (VBM補正と、電荷/ダイポール補正とバンドフィリング補正の和)

input.xlsx

赤枠はdefect_energy_summary.jsonから読み込む

Version2	Atom	Site	q	dS/kB	N0	Ndoped	A	B	dn(Cs)	dn(Cl)	dH0	dEVBM	dEpc	mu(Cs)@A	mu(Cl)@A	mu(Cs)@B	mu(Cl)@B
	V	Cl1	0	5	64	0	4.3059	0.4256	0	-1	4.3059	0	0	-3.8803	0	0	-3.8803
	V	Cl1	-1	5	64	0	9.3962	5.5159	0	-1	9.2211	0.0117	0.16341	-3.8803	0	0	-3.8803
	V	Cl1	1	5	64	0	0.1662	-3.714	0	-1	0.034	-0.031	0.16341	-3.8803	0	0	-3.8803
	V	Cs1	0	5	64	0	0.7889	4.6692	-1	0	4.6692	0	0	-3.8803	0	0	-3.8803
	V	Cs1	-1	5	64	0	0.9914	4.8717	-1	0	4.6528	0.0555	0.16341	-3.8803	0	0	-3.8803
	V	Cs1	1	5	64	0	0.8533	4.7336	-1	0	4.7465	-0.176	0.16341	-3.8803	0	0	-3.8803

↑Version2:
識別子なので必須

**N0は欠陥supercell内の欠陥サイト数:
input.xlsx作成後に確認すること**

dSは、 k_B で除した欠陥生成エントロピー
不明な場合は 5 ~ 10 程度

Atom: 元素名

Site: サイト名

欠陥名は Atom_{Site} (V_{Cl1} など)

q: 欠陥電荷

N0: 欠陥supercell内の欠陥サイト数

Boltzmann分布から欠陥濃度を計算する際に使用 $N_{\text{defect}} = N0 \exp(-\Delta H_f/kT)$

A, B, ...: 化学ポテンシャル図から取り出した相境界点の ΔH_f ($dH0+dEVBM+dEpc+\sum(\delta n_i \delta \mu_i)$)

dn(X), mu(X)@A: 理想構造と欠陥構造の原子Xの加減数、A点におけるXの化学ポテンシャル

dH0: 欠陥反応の全エネルギー差 ΔH_0

dEVBM, dEpc: pydefectによる補正 (VBM補正と、電荷/ダイポール補正とバンドフィリング補正の和)

vlauncher: VASP=>Defect: ここは不要

VASP: Defect calculation

pydefect Plot dH and N

pydefect JSON: path app

POSCAR (ideal): path app

POSCAR (defect): path app

input.xlsx: path app

Step: (1) 'Make input.xlsx', (2) Correct N0 by 'Edit input.xlsx', (3) 'load summary file'

View pydefect JSON Make input.xlsx Edit input.xlsx edit log file

edit summary file **load summary file**

dEcorr: eV dEVBM: eV dE(BF): eV

EV(ideal): eV EC(ideal): eV Eg(ideal): eV

vasprun.xml: path app

View vasprun.xml

post process explorer cmd VESTA Chesta (Windows)

確認用。↓の欄にはdefect_energy_summary.jsonから読み込んだ値が表示される

←↑: 単なる確認用。vasprun.xmlをXML Viewerで表示する

vlauncher: input.xlsxプロット

VASP: Defect calculation

pydefect Plot dH and N

Ideal model INCAR: /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_o

理想構造のINCAR: unitcell/structure_opt/INCAR

Defect model INCAR:

欠陥supercellのINCAR:

input.xlsx: /gs/bs/tga-kklab/user/tkamiya/CsCl/input.xlsx

defect以下のINCARを適当に選ぶ

index of chemical potential (iPoint): 0

化学ポテンシャルの番号
(0 => A, 1 => B, など)

plot mode: min

Electron T (T0): 300.0 K default

Defect T (Tdef): 300.0 K default

電子平衡濃度、欠陥平衡濃度

Defect EF (EFdef): eq eV default

欠陥平衡のEF (eqであれば、計算した平衡フェルミ準位)

For axis mode=EF:

dEF range: -0.2 default -0.2 eV default nEF 101 default

For axis mode=T

T range: 300.0 default -600.0 K default nT: 11 default

plot dH(defect) vs EF

plot dH(defect) vs T

edit log file

Chesta (Windows)

プロット

↑pydefectでは使えない

post process

explorer

cmd

VESTA

Chesta (Windows)

Vlauncher/vasp_plot.py: input.xlsxプロット

```
python3@login1
V_Cs1 :   -1    5    62     0    4.87166    4.87166
V_Cs1 :    1    5    62     0    4.73359    4.73359

To be analyzed for Point A
T(defects): 300.0 K

VASP files :
CAR dir   : /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt
INCAR     : /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt/INCAR
POSCAR    : /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt/POSCAR
CONTCAR   : /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt/CONTCAR
OUTCAR    : /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt/OUTCAR
EIGENVAL  : /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt/EIGENVA
L
DOSCAR    : /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt/DOSCAR
POSCAR(defect): /gs/bs/tga-kklab/user/tkamiya/CsCl/defect/Va_Cl1_-1/POSCAR

Crystal structure from [/gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_opt/POSCAR]
cell:  4.12300000  4.12300000  4.12300000 A  90.000000  90.000000  90.000000
volume:  70.087409 A^3

Check atom sites
from POSCAR:
Cs      64
Cl      63
from /gs/bs/tga-kklab/user/tkamiya/CsCl/input.xlsx:
Cl1    62.000000
d= Cl {'Cs': 64, 'Cl': 63}
Cs1    62.000000
d= Cs {'Cs': 64, 'Cl': 63}

=====
Error: Site numbers mismatch between POSCAR and /gs/bs/tga-kklab/user/tkamiya/CsCl/input.xlsx
Cl (62.0 != 63), Cs (62.0 != 64),
=====

Choose continue: Enter continue to disregard this error
Choose stop : Enter stop to terminate this run, and correct site numbers in /gs/bs/tga-kklab/user/tkamiya/CsCl/input.xlsx so as to corresponds to POSCAR
coninue or stop>>█
```

POSCARから予測した欠陥サイト数は Cs:64、 Cl:63
input.xlsxのサイト数は Cl:62、 Cs:62

POSCARから予測した欠陥サイト数とinput.xlsxのサイト数が違う場合、
“stop”で中止してinput.xlsxを修正するか、
“continue”と入力して継続する必要がある

input.xlsx

Version2	Atom	Site	q	dS/kB	N0	Ndoped	A	B			dH0	dEVBM	dEpc
	V	Cl1	0	5	62	0	4.305878	4.305878			4.305878	0	0
	V	Cl1	-1	5	62	0	9.396214	9.396214			9.221148	0.011657	0.163408
	V	Cl1	1	5	62	0	0.166152	0.166152			0.033966	-0.03122	0.163408
	V	Cs1	0	5	62	0	4.66922	4.66922			4.66922	0	0
	V	Cs1	-1	5	62	0	4.871655	4.871655			4.652788	0.055459	0.163408
	V	Cs1	1	5	62	0	4.733593	4.733593			4.746544	-0.17636	0.163408

64に修正

Version2	Atom	Site	q	dS/kB	N0	Ndoped	A	B			dH0	dEVBM	dEpc
	V	Cl1	0	5	64	0	4.305878	4.305878			4.305878	0	0
	V	Cl1	-1	5	64	0	9.396214	9.396214			9.221148	0.011657	0.163408
	V	Cl1	1	5	64	0	0.166152	0.166152			0.033966	-0.03122	0.163408
	V	Cs1	0	5	64	0	4.66922	4.66922			4.66922	0	0
	V	Cs1	-1	5	64	0	4.871655	4.871655			4.652788	0.055459	0.163408
	V	Cs1	1	5	64	0	4.733593	4.733593			4.746544	-0.17636	0.163408

vlauncher: input.xlsxプロット

VASP: Defect calculation

pydefect **Plot dH and N**

Ideal model INCAR: /gs/bs/tga-kklab/user/tkamiya/CsCl/unitcell/structure_o path app

Defect model INCAR: path app

input.xlsx: /gs/bs/tga-kklab/user/tkamiya/CsCl/input.xlsx path app

index of chemical potential (iPoint): 0 default

plot mode: min

Electron T (T0): 300.0 K default

Defect T (Tdef): 300.0 K default

Defect EF (EFdef): eq eV default

For axis mode=EF:

dEF range: -0.2 default - 0.2 eV default nEF 101 default

For axis mode=T

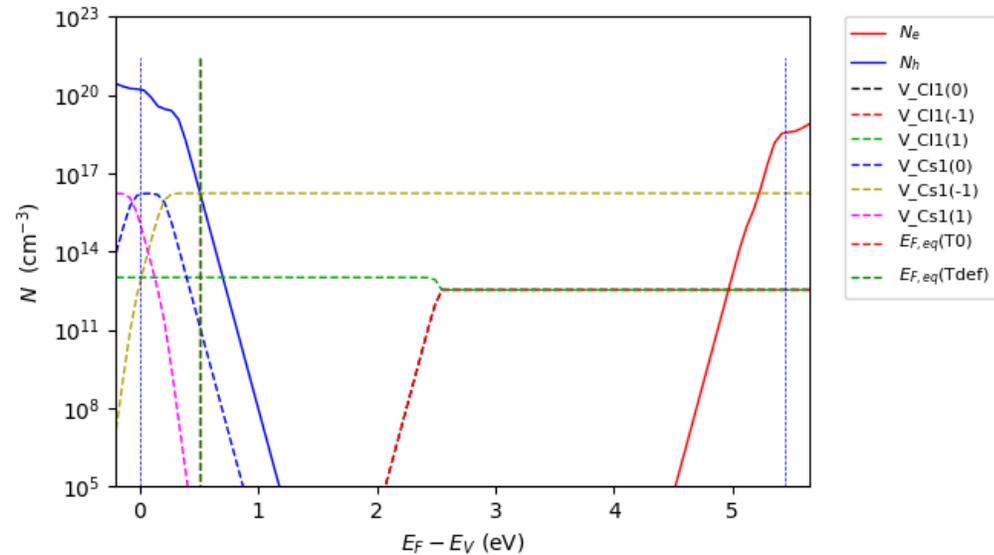
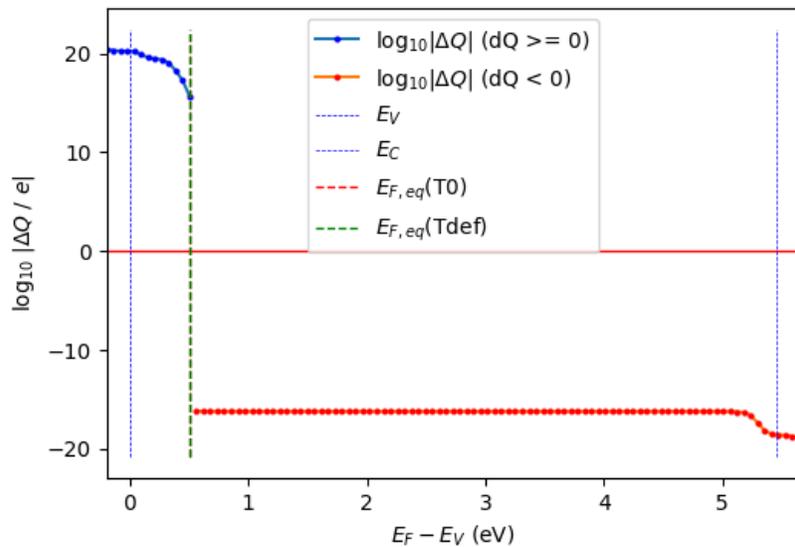
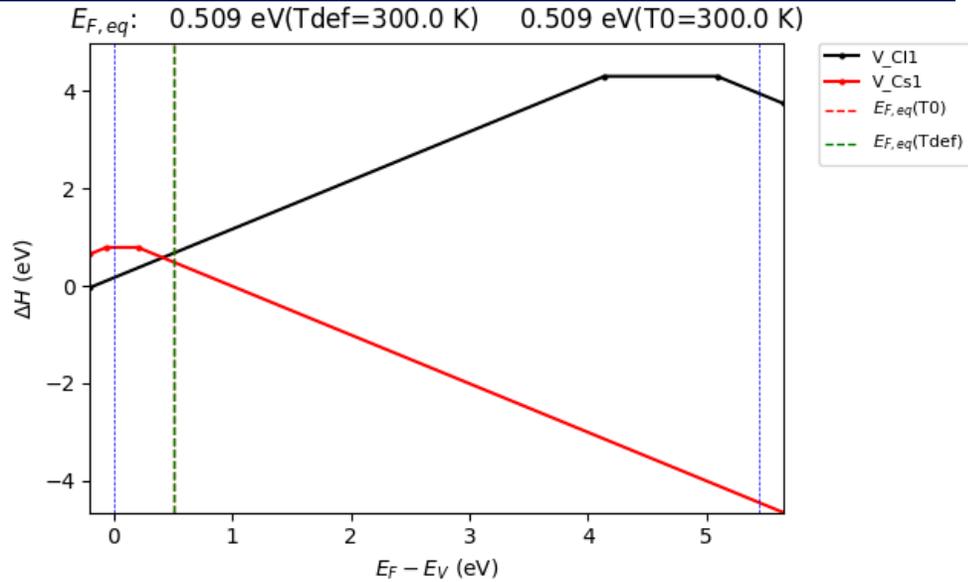
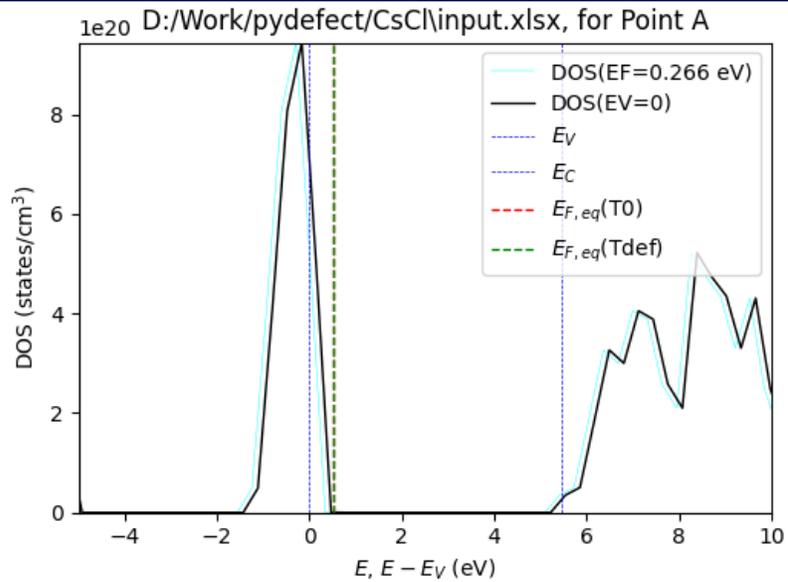
T range: 300.0 default - 600.0 K default nT: 11 default

plot dH(defect) vs EF plot dH(defect) vs T edit log file Chesta (Windows)

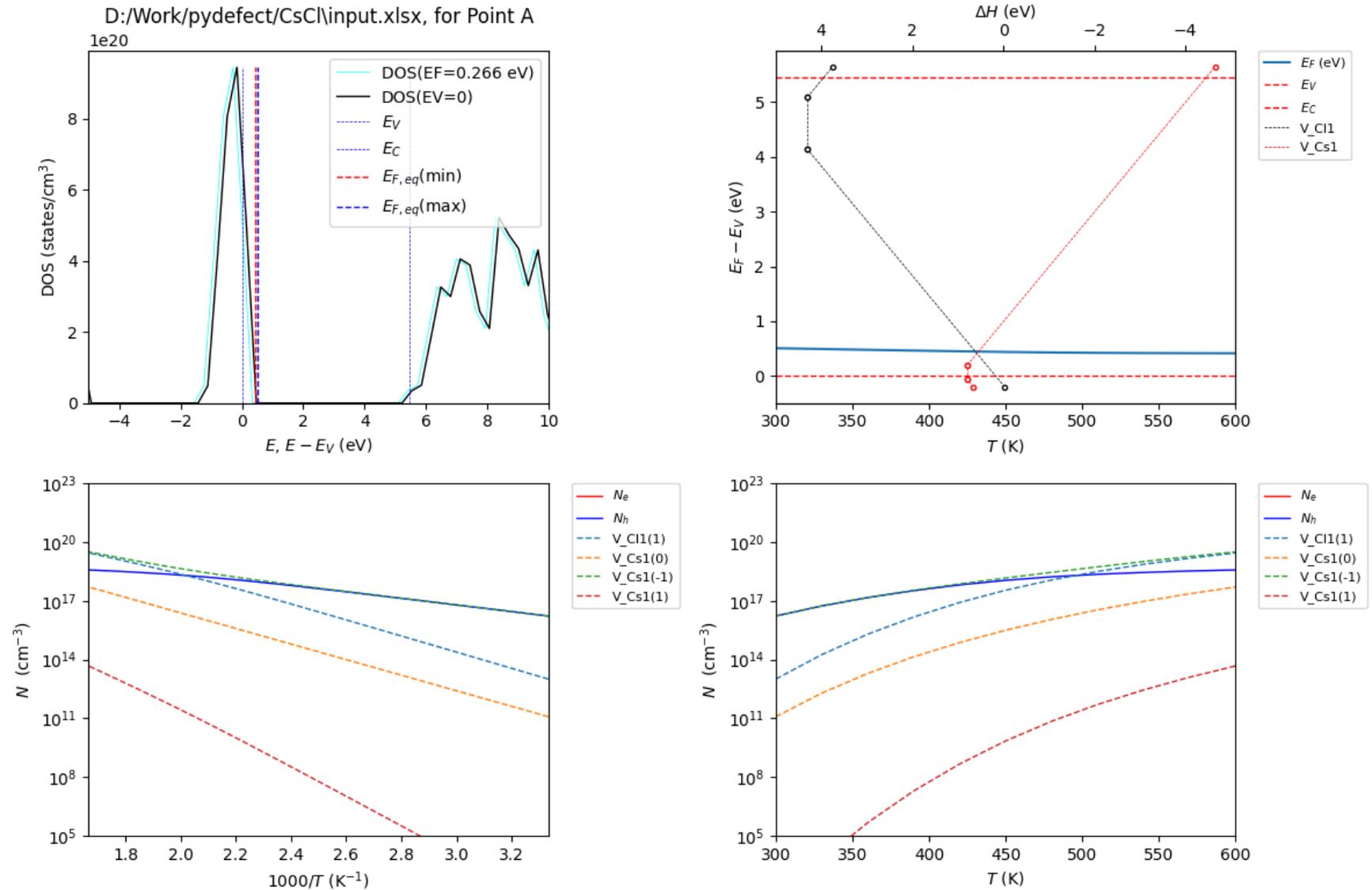
post process explorer cmd VESTA Chesta (Windows)

Input.xlsxを修正して再プロット

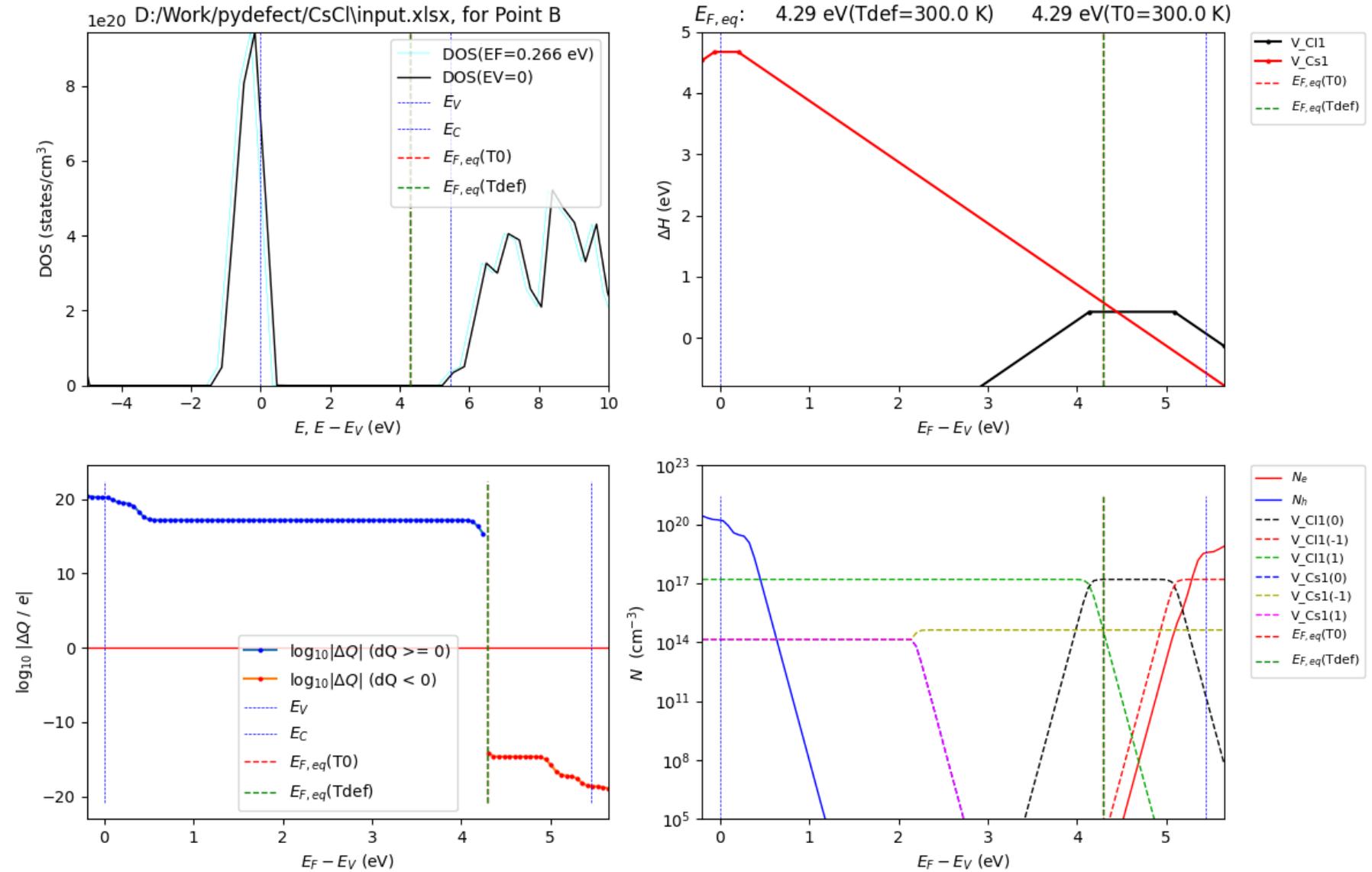
vlauncher: plot dH vs EF at A



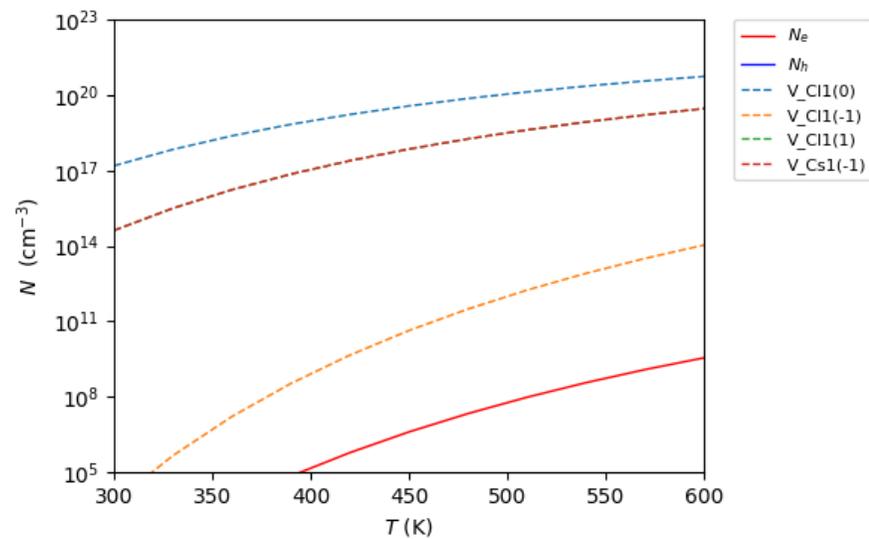
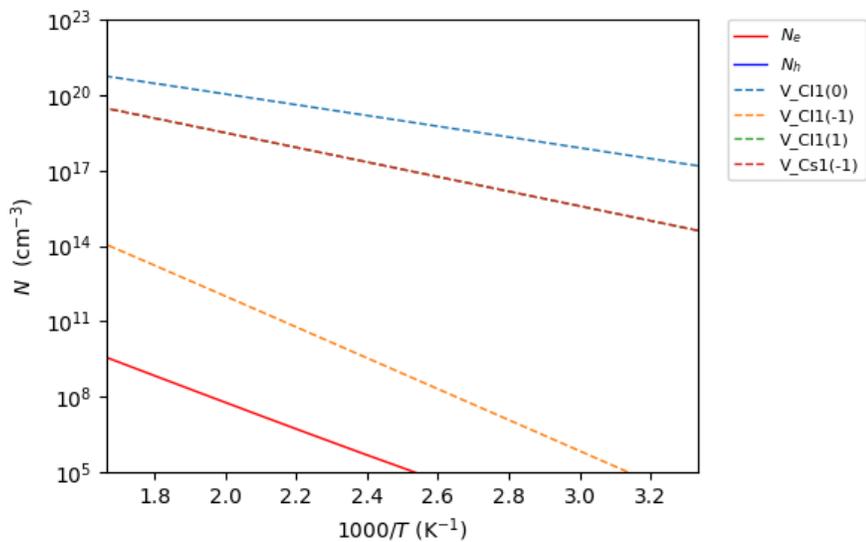
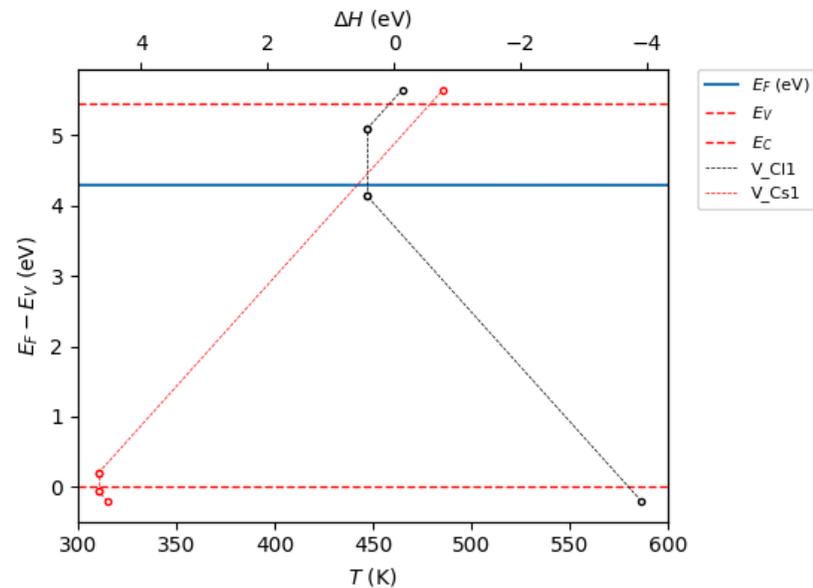
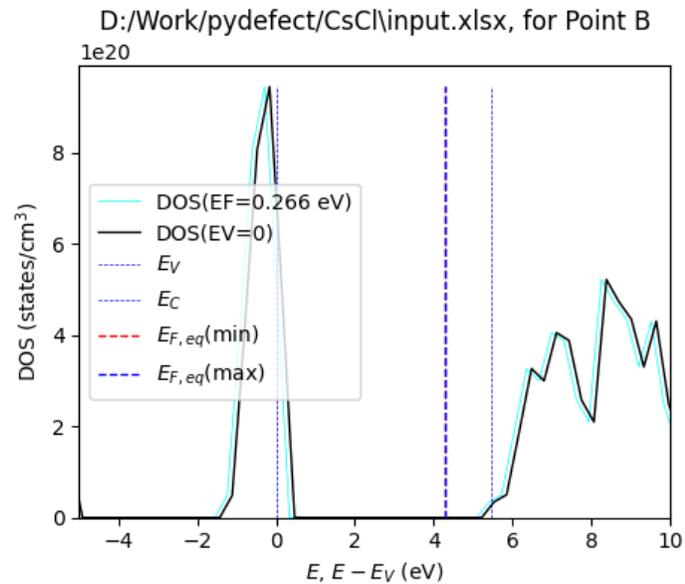
vlauncher: plot dH vs T at A



vlauncher: plot dH vs EF at B



vlauncher: plot dH vs T at B



Pydefect on TSUBAME

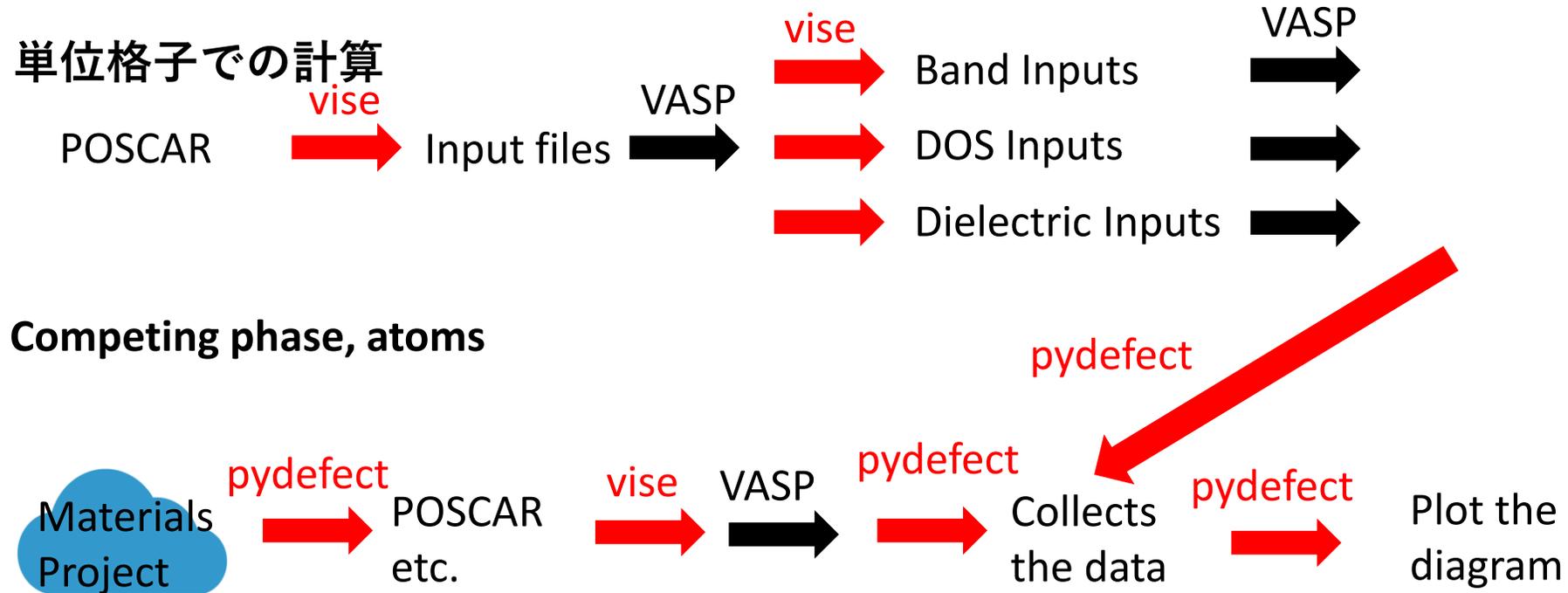
(240528)

M2 Minamishima Haruto

(240604)

Modified by T. Kamiya

計算の流れ



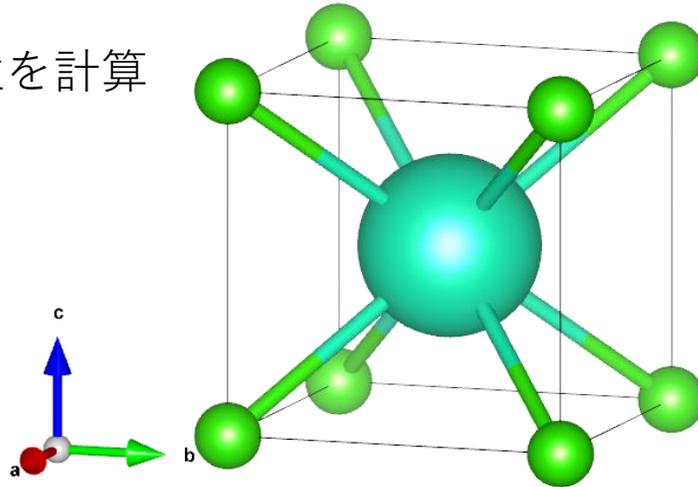
やること自体は

<https://kumagai-group.github.io/pydefect/tutorial.html>

の通りです。

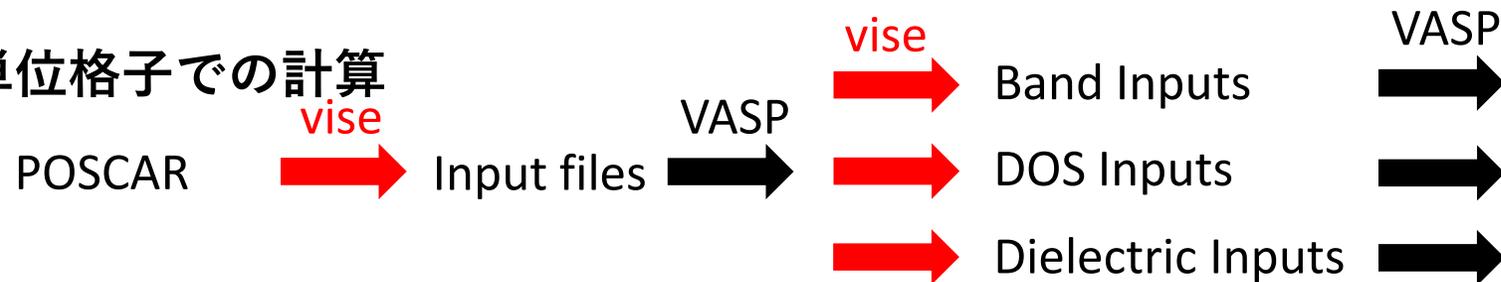
テスト計算: CsCl

CsClの欠陥特性を計算

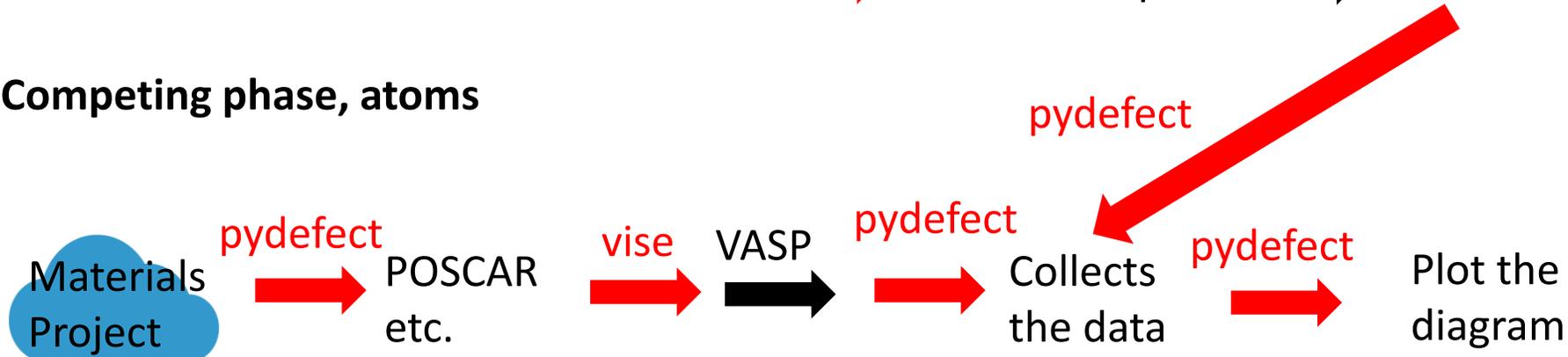


<https://unit.aist.go.jp/ifm-ri/ja/groups/db-nomura/common/STRUC-CIF/CsCl.cif>

単位格子での計算



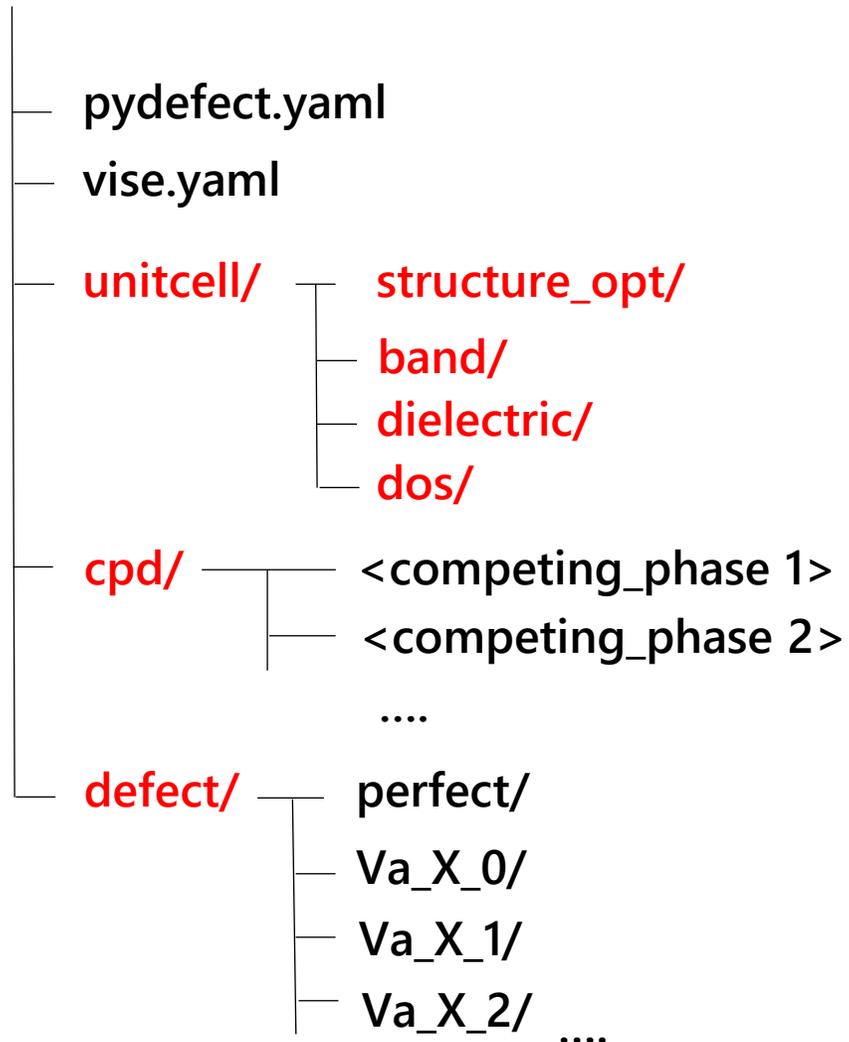
Competing phase, atoms



前提とするファイル構造

赤字のディレクトリは自分で作成する 例: <project_name>: CsCl

<project_name>



コマンド

pydefect

pydefect_vasp

pydefect_util

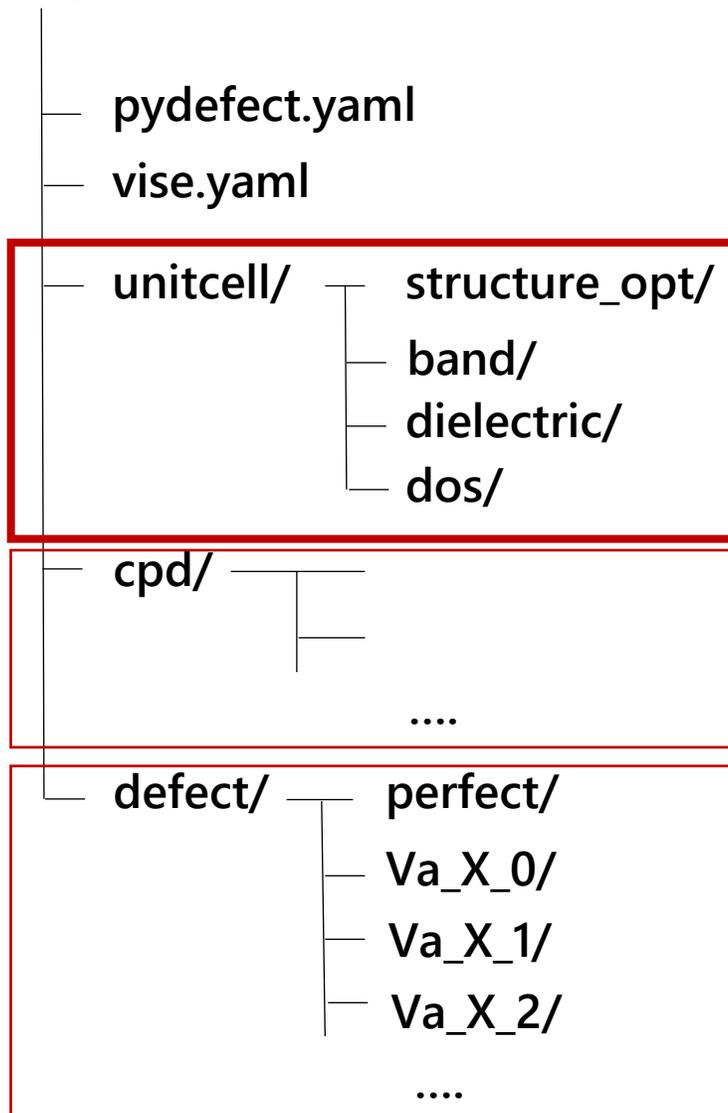
pydefect_vasp_util

pydefect_print

viseコード (pydefectに入ってる)

理想構造の計算: unitcell

<project_name>



<project_name>ディレクトリ作成

```
$ mkdir CsCl
```

理想構造ディレクトリ unitcell

```
$ mkdir unitcell
```

```
$ chdir unitcell
```

POSCARを<project_name>にしておく

```
$ get_poscar CsCl
```

構造緩和:

```
$ mkdir structure_opt
```

```
$ chdir structure_opt
```

```
$ cp ../../POSCAR .
```

入力ファイル作成

```
$ vise vs -x pbesol -t structure_opt -uis NSW 100 KPAR 1
```

-x: 汎関数を指定 (pbe0, hseなどもOK)

-t: 計算するtask。今は構造最適化用の設定

-uis NSW: NSWの値を設定

初期値20は小さすぎるので変更

KPAR: 1にしたほうが無難

https://kumagai-group.github.io/vise/tutorial_input_set.html

計算実行: qvasp

job投入用のファイルは /gs/bs/tga-kklab/bin/{qvasp, vasp-info} にある

cpvaspコマンドでコピー (基本的に変える必要はない)

```
$ cpvasp
```

```
以下と同等: $ cp /gs/bs/tga-kklab/bin/qvasp
```

```
$ cp /gs/bs/tga-kklab/bin/vasp-info .
```

実行

```
$ ./qvasp
```

```
Your job 236910 ("job-name") has been submitted
```

jobの状態を確認

```
$ qstat
```

```
job-ID  prior  name          user      state submit/start at   queue          jclass      slots ja-task-ID
```

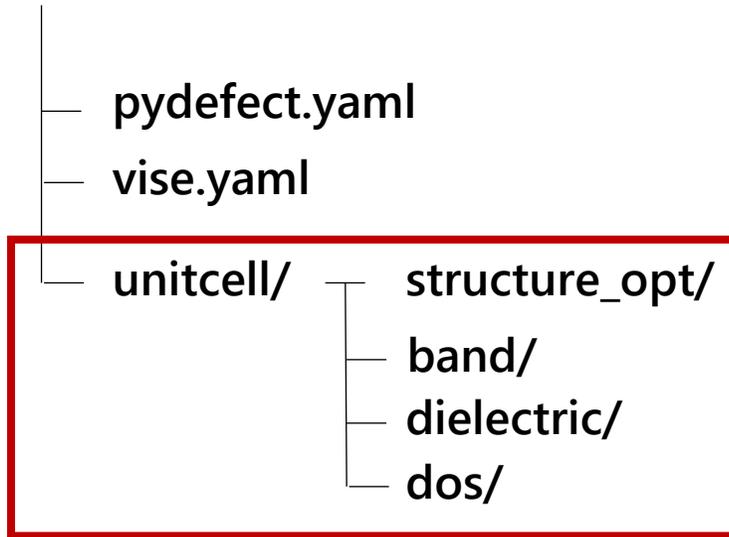
```
-----  
236910 0.55256 job-name    uk02294   r    06/04/2024 13:35:57 all.q@r18n5
```

```
  r: running
```

```
  qw: queue waiting
```

電子構造・誘電率計算 : unitcell

<project_name>



- structure_optが終了したら
band/ dielectric/ dos/ を作り、それぞれのディレクトリで以下のコマンドを実行
\$ vise vs -x pbesol -t band -pd ../structure_opt
\$ vise vs -x pbesol -t dielectric_dfpt -pd ../structure_opt
\$ vise vs -x pbesol -t dos -uis LVTOT True LAECHG True KPAR 1 -pd ../structure_opt
-pd: 指定したディレクトリの計算結果から入力ファイルを作る
(CONTCARをもってくるなど)
- 各ディレクトリでVASPの計算を実行

計算のまとめ、Band/DOSプロット

計算結果をまとめた unitcell.yaml を作る

unitcell/ に移動

```
$ chdir ..
```

```
$ pydefect_vasp u -vb band/vasprun.xml -ob band/OUTCAR -odc dielectric/OUTCAR -odi  
dielectric/OUTCAR -n CsCl
```

- **unitcell.yaml**が生成される

バンドの描画と band.pdfの生成

band/ディレクトリに移動

```
$ cd band
```

```
$ vise pb
```

- 縦軸は-y <下限> <上限>で設定できる

DOSの描画と dos.pdfの生成

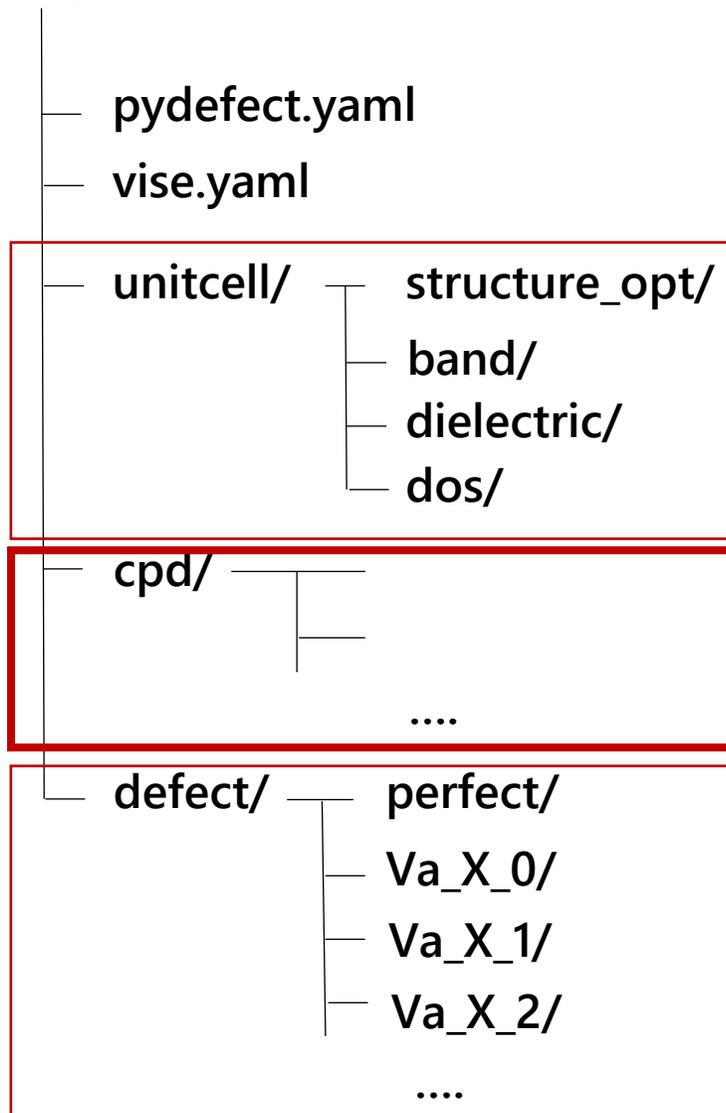
dos/ディレクトリに移動

```
$ cd ../dos
```

```
$ vise pd
```

競合相ディレクトリの作成

<project_name>



cpd/ディレクトリを作成し、移動

```
$ mkdir cpd
```

```
$ chdir cpd
```

Materials Projectから競合相の候補物質を取得

```
$ pydefect_vasp mp -e Cs Cl --e_above_hull 0.0005
```

-e: 含めたい元素

ドーピングする元素も入れる

--e_above_hull: 熱力学的凸包からどれだけ

離れている物質を取得するかを指定

0.0005 [eV] はDocumentで示されていた値

準安定相の計算をする場合はその物質自身が取得

される程度まで値を大きくする

競合相の計算

Al2O3_mp-1143/
Al_mp-134/
Mg149Al_mp-1185596/
Mg17Al12_mp-2151/
MgAl2O4_mp-3536/
MgAl2_mp-1094116/
MgO_mp-1265/
Mg_mp-1056702/
mol_O2/

左のようなディレクトリが生成されているはず

- これらの中にはPOSCARとprior_info.yamlが入っている
- **親物質**自身がリストに入っているので、シンボリックリンクで置き換える
- `$ rm -r CsCl_mp-22865`
- `$ ln -s ../unitcell/structure_opt CsCl_unitcell`

まとめて入力ファイルを作成

```
$ for i in *_*/;do cd $i; vise vs -uis ENCUT 400.0 NSW 300 -x pbesol ; cd ../;done
```

ENCUT: 用いている元素のENMAXのうち最大のもので統一するために設定
struct_optとエネルギーを比較できる必要がある。
ほとんどの場合、400.0でも十分 (なはず)
NSW: 大きいセルを持つ物質が候補にある場合、大きくする(300程度)

すべての物質に対してVASPで計算を実行

```
$ for i in *_*/;do cd $i; cpvasp; ./qvasp; cd ../;done
```

競合相の計算

cpd/ で実行

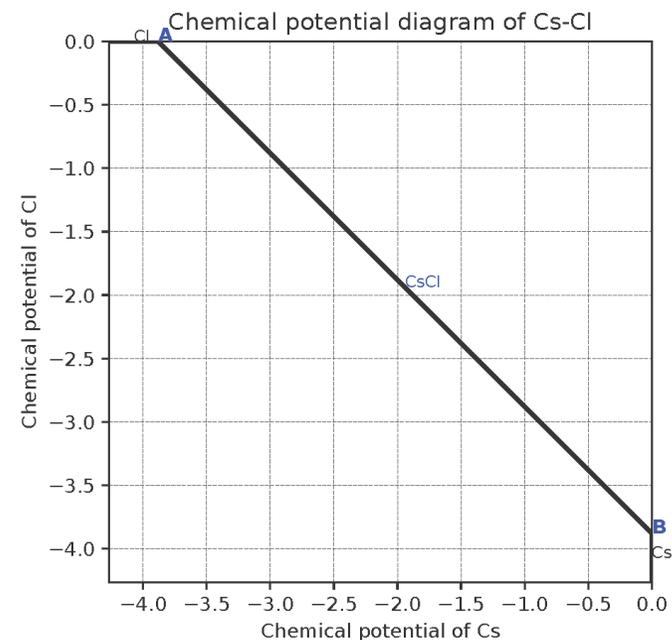
結果のパーズ

```
$ pydefect_vasp mce -d *_*/  
    composition_energies.yamlが生成される
```

```
$ pydefect sre  
    relative_energies.yamlとstandard_energies.yamlが生成される
```

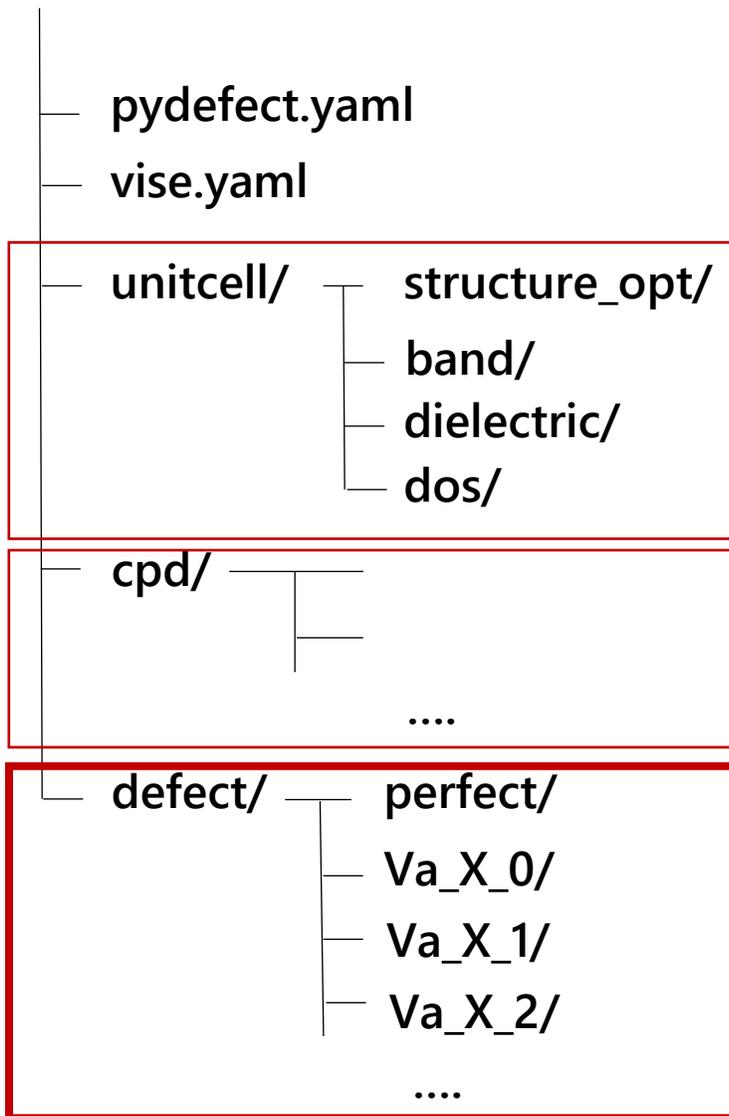
相図のプロット

```
$ pydefect cv -t CsCl  
$ pydefect pc  
    chemical potential diagramが描画される  
    含まれる元素が4以上になると描画されない  
    (計算自体は実行可能)$II
```



欠陥計算 (native defect)

<project_name>



defect/ ディレクトリを作成、移動

```
$ mkdir defect
```

```
$ chdir defect
```

スーパーセルの作成

```
$ pydefect s -p
```

```
../unitcell/structure_opt/CONTCAR
```

SPOSCARが生成される

大きさはpydefectが適切に設定する

指定したい場合は--matrixで指定する

```
$ pydefect s -p
```

```
../unitcell/structure_opt/CONTCAR
```

```
--matrix 2 1 1
```

supercell_info.jsonにsupercellや欠陥の情報が
入っている

確認:

```
$ pydefect_print supercell_info.json
```

欠陥計算

defect/で実行

```
$ pydefect ds
```

ドーピングの場合: `$ pydefect ds -d Ca`

欠陥の入った構造の生成

```
$ pydefect_vasp de
```

各構造ごとにディレクトリが生成され、
POSCARとdefect_entry.jsonが入っている

VASPの入力ファイルの生成

```
$ for i in */;do cd $i; vise vs -t defect -x pbesol -uis NSW 300 ; cd ../;done
```

-x pbesol: 公式のドキュメントから抜けているが必須
これまでの計算と汎関数をそろえること

```
(-uis ENCUT 400.0)
```

VASPを実行

```
$ for i in */;do cd $i; cpvasp; ./qvasp; cd ../;done
```

計算結果のパーズ

```
$ pydefect_vasp cr -d *_*/ perfect
```

欠陥計算: 補正

defect/で実行

計算結果の補正: pydefectでは拡張型fnv補正を使う
(点電荷補正とバンドフィリングもする)

```
$ pydefect efnv -d *_*/ -pcr perfect/calc_results.json -u ../unitcell/unitcell.yaml  
各ディレクトリにcorrection.pdfが生成される
```

欠陥構造の解析

```
$ pydefect dsi -d *_*/  
defect_structure_info.jsonが生成される
```

解析用のvesta用ファイルも生成できる

```
$ pydefect_util dvf -d *_*
```

バンド端関係の補正

```
$ pydefect_vasp pbes -d perfect  
$ pydefect_vasp beoi -d *_* -pbes perfect/perfect_band_edge_state.json  
$ pydefect bes -d *_*/ -pbes perfect/perfect_band_edge_state.json
```

欠陥計算: プロット

各欠陥のエネルギーを抽出

```
$ pydefect dei -d *_*/ -pcr perfect/calc_results.json -u ../unitcell/unitcell.yaml -s ../cpd/standard_energies.yaml
```

各ディレクトリにdefect_energy_info.yamlができる

欠陥生成エネルギーのsummaryを作る

```
$ pydefect des -d *_*/ -u ../unitcell/unitcell.yaml -pbes perfect/perfect_band_edge_state.json -t ../cpd/target_vertices.yaml
```

defect_energy_summary.jsonができる

採用するchemical potential、生成エネルギー、エネルギーの補正值が入っている

計算結果のsummaryを作る

```
$ pydefect cs -d *_*/ -pcr perfect/calc_results.json
```

欠陥生成エネルギー図のプロット

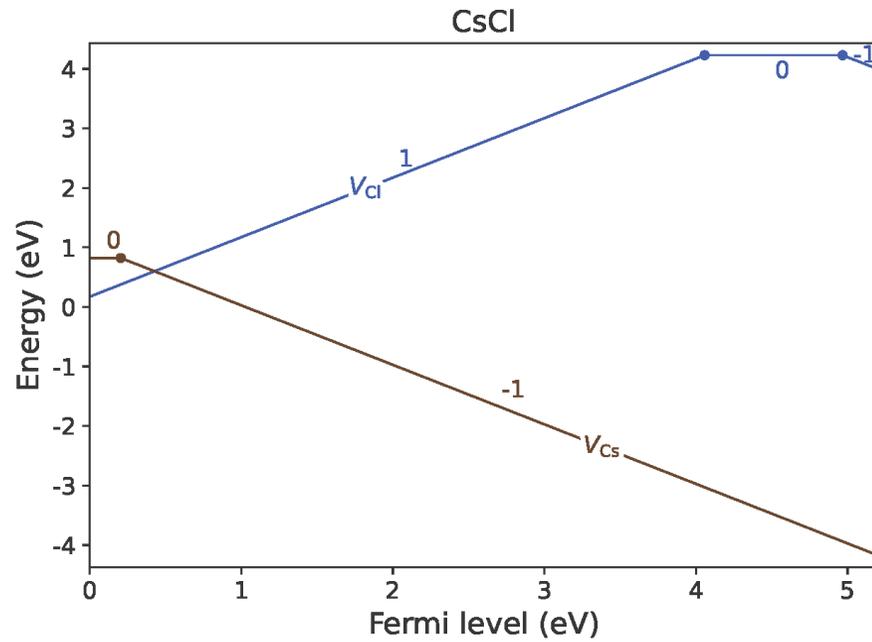
```
$ pydefect pe -d defect_energy_summary.json -l A
```

-lで化学ポテンシャル図のどの点を取るかを決めている

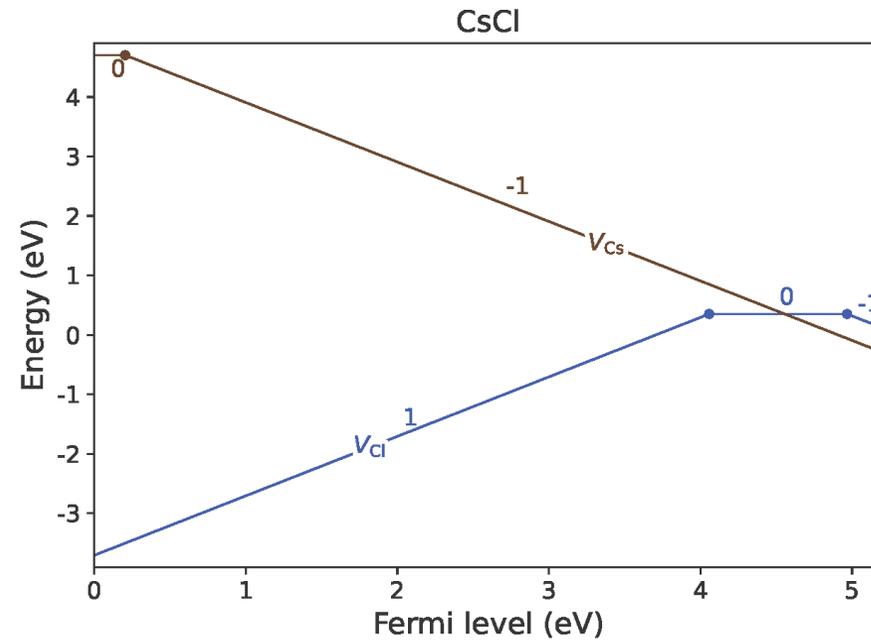
energy_A.pdfを出力

計算結果

point A



point B



その他

欠陥計算: 格子間欠陥

- 格子間の欠陥に関して
 - `pydefect`では、結晶中の電荷密度のうち局所最小を取る場所を探索し、格子間サイトとして提案してくれる機能がある
 - 探索には、`unitcell/dos`ディレクトリで以下のコマンドを実行すればできる
`$ pydefect_vasp le -v AECCAR{0,2} -i all_electron_charge`
 - `volumetric_data_local_extrema.json`に結果が格納される
 - `supercell_info.json`ができたあと、`defect/`で
`$ pydefect_util ai --local_extrema ../unitcell/dos/volumetric_data_local_extrema.json -i 1 2`
を実行すると`supercell_info.json`に格子間サイトの情報を書き込んでくれる
 - 消したい場合は
`$ pydefect pi -i 1 -s supercell_info.json`
とすることで消せる
 - `-i`で格子間サイトの番号を指定する
- 設定終了後に改めて
`$ pydefect ds`
を実行することで`defect_in.yaml`に欠陥の情報が書き込まれる

欠陥計算: ドーピング

- ドーピング
 - ドーパントを含んだ競合相の計算とドーパントが入った欠陥構造の計算が必要
- 競合相
 - `$ pydefect_vasp mp -e Cs Cl --e_above_hull 0.0005`
で競合相の構造を取得する際に、`-e`でドーパントの元素も指定しておく
- 欠陥構造の生成
 - `$ pydefect ds`
を実行する際に、`-d`オプションでドーパントの元素を指定する
 - 欠陥サイトや価数まで含めて`pydefect`が決定してくれる
 - 欠陥のエネルギー抽出や`summary`を作る際などに競合相の情報を渡すが、その時にドーパントの元素を含んだ情報を渡すようにする
 - ドーパントの情報が入っていないとエラーになる

Tips

- `vise vs`に関して
 - `INCAR`を生成してくれるが、`NSW`を小さくする傾向があり、収束しないことがある
 - また、`KPAR=3`のようなそもそも`VASP`が実行できないパラメータを設定されることもある
- 安全のため、常に`-uis KPAR 1 NSW 300`をつけておくと無駄な計算を減らせる
- `vise.yaml`に書いておくと自動的に設定してくれるため、設定しておくことを推奨

```
user_incar_settings:  
  NSW: 300  
  KPAR: 1
```

POTCARの設定に関して:メモ

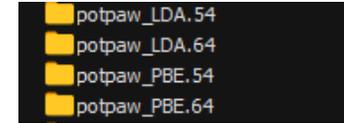
- POTCARが入っているディレクトリの名前を画像のように変更する
pymatgenがこの形式のディレクトリ名じゃないと読めないらしい

- 画像の状態で

```
$ pimg config -p <> <>
```

を実行し、パスを通してもうまくいかない

- 生成されるディレクトリ名が想定と異なり、
potpaw_LDA.64とpotpaw_PBE.64というディレクトリのみ生成される
→この事象は原因不明、中身自体は想定通りになっている
potpaw_LDA.64とpotpaw_PBE.64は使えないと割り切る？
どのみちPBEsolの入力ファイルをviseで生成しようとするのでPBE.54の
POTCARを持ってくるため気にしなくてよさそう



- potpaw_LDA.54とpotpaw_PBE.54の下に直接各元素のPOTCARが入っているファイル構成になっている必要がある
 - tkProg/tkdbにあるディレクトリはpotpaw_PBE54/elementsの下にあるため直す
 - 直せば想定通りに生成された

pymatgen公式の案内:

<https://pymatgen.org/installation.html#potcar-setup>