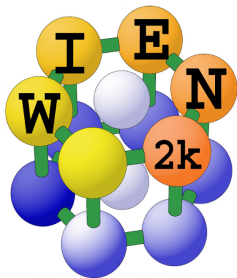


# Thermodynamics of Point Defects in Solids with WIEN2k and Spinney

Marco Arrigoni

Institute of Materials Chemistry TU Wien

26th WIEN2k Workshop 17.08.2019



IMC

[marco.arrigoni@tuwien.ac.at](mailto:marco.arrigoni@tuwien.ac.at)

# Why Model Point Defects in Solids?

- At the thermodynamic equilibrium, any solid system will possess a given amount of point defects.
- The presence of defects can noticeably affect the electronic, optical and transport properties of materials.
- Such effects can be beneficial for device applications or detrimental.
- The vast majority of electronic and optoelectronic devices are based on doped semiconductors. The amount of point defects necessary for considerably affecting the properties of a semiconductor material can be very small ( $\approx 1\text{ppm}$ ).
- Effectiveness of doping can be hindered by the formation of compensating intrinsic defects.

# Why Model Point Defects in Solids?

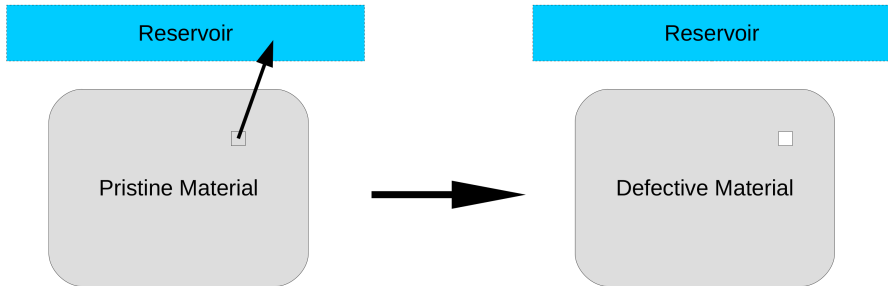
- The ability to control the concentrations of defects in materials determines the possibility of using such materials in technological devices.
- Understanding the physics and chemistry of point defects and their interaction in a material can guide the development of devices based on that material.
- The experimental characterization of point defects in solids is challenging and time consuming.
- First-principles modeling offers a convenient method for an atomistic description of point defects, complementing experimental studies and predicting new materials of interest.
- Calculations based on DFT are nowadays routinely employed but are challenging. The accuracy of first-principles calculations of defect properties is still far from the one of ground state properties.

# The Dilute Limit

The vast majority of first-principles calculations of systems with point defects aim to consider the **dilute limit**: defect concentrations  $\rightarrow 0$ , no defect-defect interactions.

Interest in this limit since in technological applications we usually deal with very small defect concentrations.

In this limit, we would ideally like to consider the introduction of an isolated defect on the host material:



# The Defect Formation Energy

A key quantity that describes such process is the **defect formation energy**: the change in grand potential due to the defect introduction.

$$\Delta G_f(X; q) = G(X; q) - G(\text{bulk}) - \sum_i n_i \mu_i + q \mu_e (+E_{\text{corr}}) \quad (1)$$

- $G(X; q)$ : the energy of the system with a defect  $X$  in charge state  $q$
- $G(\text{bulk})$ : the energy of the pristine system
- $n_i$ : number of (neutral) atoms added ( $n_i > 0$ ) or removed ( $n_i < 0$ ) from the host material
- $\mu_i$ : chemical potential of atom  $i$  in the reservoir,  $\mu_e$ : chemical potential of the electron in the reservoir
- $E_{\text{corr}}$ : a term that aims to correct for the inevitable errors that occur when we do actual calculations.

# The Defect Formation Energy

Example: Formation of a fully ionized N vacancy in GaN:

$$\Delta G_f(\text{Vac}_N; 3) = G(\text{Vac}_N; 3) - G(\text{GaN}) + \mu_N + 3\mu_e(+E_{\text{corr}}) \quad (2)$$

- In the process, one N atom is removed from the system.
- MO picture: pristine system has  $N_0$  bonding (valence) states that can accommodate  $2N_0$  valence electrons.
- After removing a N atom ( $[\text{He}] 2s^2 2p^3$ ), we have  $N_0 - 4$  bonding states than can accommodate  $2N_0 - 8$  valence electrons.
- The number of valence electrons after removing a N atom is  $2N_0 - 5 \rightarrow (2N_0 - 5) - (2N_0 - 8) = 3$  valence electrons must be placed in antibonding states.
- Such states are induced by the defect. They can lie close to the conduction band minimum or can lie deep within the band gap. For some process these 3 electrons are removed from these states, leaving the N vacancy in the completely ionized state with charge +3. In such case, the N vacancy acts as a donor.

# Thermodynamic Transition Levels

Consider the case of P doping in Si: a typical shallow donor.

Long-range behavior potential ionized P:  $V_I(\mathbf{r}) \approx \frac{e}{4\pi\kappa\epsilon_0 r}$ .

$\kappa$  is the dielectric constant of the host material.

In a single-conduction-band model, we can write the Schrödinger equation for an electron in the doped material as:

$$\left( -\frac{\hbar^2}{2m_c^*} \nabla^2 - \frac{e^2}{4\pi\kappa\epsilon_0 r} \right) F(\mathbf{r}) = (E - E_c)F(\mathbf{r}) \quad (4)$$

- $m_c^*$ : conduction band effective mass at  $\mathbf{k} = 0$ .
- $E_c$ : conduction band minimum.
- $F(\mathbf{r})$ : envelope function.

# Thermodynamic Transition Levels

Analogous to Schrödinger equation of the hydrogen atom:

$$E_n = E_c + E_n(H) \frac{m_c^*}{m\kappa^2} \quad (5)$$

$E_n(H)$ : energy levels hydrogen atom.

- A defective level has a series of “internal” excitation levels. They can be detected with spectroscopy techniques.
- More interesting for the electrical behavior is when this defective level is ionized: one electron is donated to the host material.
- Most point defects, can be ionized more than once. For devices, we are mostly interested in the temperature-induced ionization.

**Thermodynamic Transition Level** between charge state  $q$  and  $q'$ : the value of  $\mu_e$  for which defect  $X$  in charge state  $q$  has the same formation energy as in charge state  $q'$ :

$$\Delta G_f(X; q) = \Delta G_f(X; q') \longrightarrow \mu_e = \frac{G(X, q) - G(X, q')}{q' - q} \equiv \epsilon(q/q') \quad (6)$$



# Defect Concentrations

Consider the formation of defect  $X$  with formation energy:  $\Delta G_f(X; q)$ . We want to find the concentration of defect  $X$  at equilibrium.

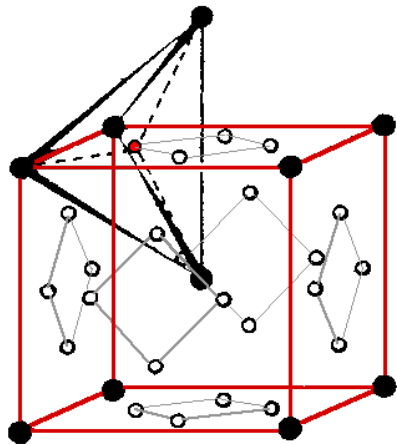
- In the dilute limit defects do not interact:  
 $\Delta G_f(n_X X; q) = n_X \Delta G_f(X; q)$ .
- Our system is in contact with a reservoir that keeps  $\mu_i, T, p$  constant.
- Under such conditions the appropriate thermodynamic potential is the grand potential:  $\Phi = G - \sum_i \mu_i n_i$  which is minimized at the equilibrium.

$$n_X \Delta G_f(X; q) = \Delta G_f(n_X X; q) = \Phi(n_X X; q) - \Phi(\text{bulk}) - TS_{conf}(n_X) \quad (8)$$

$S_{conf}$  is the contribution of the configurational entropy to the grand potential of the defective system.

# Defect Concentrations

Suppose that defect  $X$  can be placed in  $g$  equivalent lattice sites in the unit cell of the host material.



- E.g. an impurity can occupy 24 equivalent tetrahedral interstitial sites in a BCC conventional cell.
- Considering a crystal made of  $N$  primitive BCC cells, we have that the defect  $X$  can be placed on  $Ng = 12N$  sites.
- In each of these sites, the defect can have  $\theta_X$  equivalent configurations with the same formation energy (e.g. due to spin degeneracy).

# Defect Concentrations

We can thus place  $n_X$  defects on  $Ng$  sites, considering that each site has a degeneracy of  $\theta_X$ , in:

$$\Omega_X = \theta_X^{n_X} \binom{Ng}{n_X} \quad (9)$$

ways.

The configurational entropy is then:

$$S_{conf} = k_B \ln \Omega_X \quad (10)$$

Using Stirling's approximation:

$$S_{conf} = k_B (n_X \ln \theta_X + (Ng) \ln Ng - n_X \ln n_X - (Ng - n_X) \ln (Ng - n_X))$$

In contact with the reservoir,  $n_X$  is such that  $\Phi(n_X X; q)$  is minimized:

$$0 = \frac{\partial \Phi(n_X X; q)}{\partial n_X} = \Delta G_f(X; q) - k_B T \ln \left( \theta_X \frac{Ng - n_X}{n_X} \right) \quad (11)$$

# Defect Concentrations

We obtain then the defect concentration:

$$c_X = \frac{n_X}{N} = \frac{\theta_X g}{\exp\left(\frac{\Delta G_f(X; q)}{k_B T}\right) + \theta_X} \quad (12)$$

For common temperatures and formation energies  $\Delta G_f(X; q) \gg k_B T$  thus:

$$c_X = \theta_X g e^{-\frac{\Delta G_f(X; q)}{k_B T}} \quad (13)$$

More than one defect type, assuming mutually independence for the placement of different defect types (since  $n_X \ll N$ ):

$$\Omega = \Omega_X \Omega_Y \cdots \Omega_Z \rightarrow S_{conf} = S_{conf}(n_X) + S_{conf}(n_Y) + \cdots + S_{conf}(n_Z)$$

$$c_X = \theta_X g e^{-\frac{\Delta G_f(X; q)}{k_B T}} \quad c_Y = \theta_Y g e^{-\frac{\Delta G_f(Y; q')}{k_B T}} \quad \dots$$

# The Spinney Code

- The Spinney code is a collection of Python 3 modules aiming to process first-principles calculations of point defects in solids.
- All core routines accept built-in Python objects making the code independent on a particular ab-initio code or non-standard Python library.
- A high level interface, the `PointDefect` **class**, based on the `Atoms` **class** of the `ase` library, allows for a straightforward calculation of defect formation energies of charged defects, corrected for electrostatic finite-size effects, from raw first-principles data.
- The code allows to calculate defect and carriers concentrations at equilibrium in the dilute limit and implements a series of tools able to aid in all the computational steps needed in order to calculate these quantities.

# Calculate the Defect Formation Energy in Practice

$\Delta G_f(X; q)$  allows calculation of defect concentrations and transition levels which ultimately determine the carrier concentrations in the material.

- Calculate  $\Delta G_f(X; q)$ : estimating the various quantities using Density Functional Theory.
- From electronic structure calculations, we obtain the Born-Oppenheimer electronic energy of the system:  $G \rightarrow E$ .
- Finite temperatures are of more interest:  
$$G = E + F_{el} + F_{vib} + \dots + pV.$$
- Free energy contributions might be important, however they are expensive to calculate. Usual approximation:  $G \rightarrow E$  for solid species, free energy contributions included only for gas species.

# The Supercell Method

Most popular method for calculating  $\Delta G_f(X; q)$ . A defect is placed in a portion of the crystal consisting of several primitive cells. Periodic boundary conditions are applied.

- **Pro:** Highly efficient and simple implementation with plane-wave-based methods.
- **Pro:** Easy, but might be expensive, control of convergence with supercell size.
- **Pro:** Accurate description of the periodic defect-free system.
- **Con:** Instead of an isolated defect, we obtain a periodic array of defects: spurious interactions.
- **Con:** We need to apply a *a posteriori* correction scheme to amend such effects.
- **Pro:** Most spurious effects can be efficiently avoided by increasing the supercell size.

# Building Supercells with WIEN2k

Simple supercells can be built using the `supercell` program: E.g.  $2 \times 2 \times 2$  supercell of cubic BN.

```
[marrigoni@c006-Bb c-BN]$ x supercell
Program generates supercell from a WIEN struct file.

Filename of struct file:
c-BN.struct

Number of cells in x direction:
2
Number of cells in y direction:
2
Number of cells in z direction:
2
Optional shift all atoms by the same amount (fractional coordinates).
Please enter x shift:
0
Please enter y shift:
0
Please enter z shift:
0

Current structure has lattice type F
Enter your target lattice type: (P,B,F)
P
Target lattice type will be P

Add vacuum in x-direction for surface-slab [bohr]:
0
Add vacuum in y-direction for surface-slab [bohr]:
0
Add vacuum in z-direction for surface slab [bohr]:
0

Supercell generated successfully.
Stored in struct file: c-BN_super.struct

You may need to replace an atom by an impurity or distort the positions, ....
0.004u 0.006s 0:17.88 0.0%      0+0k 8+80to 0pf+0w
[marrigoni@c006-Bb c-BN]$ ls
:log c-BN.struct c-BN_super.struct fort.21 supercell.def
[marrigoni@c006-Bb c-BN]$
```



# Building Supercells with WIEN2k

- Lower the symmetry of the supercell: modify `c-BN_super.struct`.

```
ATOM 7: X=0.00000000 Y=0.50000000 Z=0.50000000
      MULT= 1          ISPLIT= 2
B      NPT= 781 R0=0.00010000 RMT= 1.2200 Z: 5.0
LOCAL ROT MATRIX: 1.0000000 0.0000000 0.0000000
                  0.0000000 1.0000000 0.0000000
                  0.0000000 0.0000000 1.0000000
ATOM 8: X=0.50000000 Y=0.50000000 Z=0.50000000
      MULT= 1          ISPLIT= 2
B1     NPT= 781 R0=0.00010000 RMT= 1.2200 Z: 5.0
LOCAL ROT MATRIX: 1.0000000 0.0000000 0.0000000
                  0.0000000 1.0000000 0.0000000
                  0.0000000 0.0000000 1.0000000
ATOM 9: X=0.25000000 Y=0.25000000 Z=0.00000000
```

- We have renamed atom 8 as B1 since we want to model a B vacancy in cubic BN.
- Copy `c-BN_super.struct` to `c-BN.struct` and run `sgroup` to find the space group of the supercell.

```
[marrigoni@c006-Bb c-BN]$ x sgroup
0.001u 0.003s 0:00.01 0.0% 0+0k 0+8io 0pf+0w
[marrigoni@c006-Bb c-BN]$
```

# Building Supercells with WIEN2k

- Copy `c-BN.struct_sgroup` to `c-BN.struct` and initialize the calculation:

```
[marrigoni@c006-Bb c-BN]$ init -b -sp -lvns 6 -gmax 20 -numk 64
```

- We will need the Coulomb potential for the electrostatic finite-size corrections (that we will see later). You need to modify the `c-BN.in0` file: replace `NR2V` with `R2V`. This will print the `c-BN.vcoul` file containing the Coulomb potential after the calculation is run.

```
TOT XC_PBE (XC_LDA,XC_PBESOL,XC_WC,XC_MBJ,XC_SCAN)
NR2V IFFT (R2V)
90 90 90 2.00 1 NCON 9 # min IFFT-parameters, enhancement factor, iprint, NCON
```

- Run the calculation for the  $2 \times 2 \times 2$  pristine supercell of cubic BN.
- Files that we will need for further processing:
  - `c-BN.scf`
  - `c-BN.struct`
  - `c-BN.vcoul`

# Building Supercells with WIEN2k

- Defective system: B vacancy in charge state  $q = -3$ .
- Create a new folder (e.g. defective) and copy the `c-BN_super.struct` file to `defective.struct`.
- Remove a B atom (for example ATOM 8) and reduce the total number of atoms by 1 in the `defective.struct` file.

```
BN_cubic
P LATTICE,NONEQUIV. ATOMS 6
MODE OF CALC=RELA unit=bohr
13.701898 13.701898 13.701898 90.000000 90.000000 90.000000
ATOM 1: X=0.00000000 Y=0.00000000 Z=0.00000000
      MULT= 1 ISPLIT= 2
B      NPT= 781 R0=0.00010000 RMT= 1.2200 Z: 5.0
LOCAL ROT MATRIX: 1.0000000 0.0000000 0.0000000
                  0.0000000 1.0000000 0.0000000
                  0.0000000 0.0000000 1.0000000
ATOM 2: X=0.50000000 Y=0.00000000 Z=0.00000000
      MULT= 1 ISPLIT= 2
B      NPT= 781 R0=0.00010000 RMT= 1.2200 Z: 5.0
LOCAL ROT MATRIX: 1.0000000 0.0000000 0.0000000
                  0.0000000 1.0000000 0.0000000
                  0.0000000 0.0000000 1.0000000
ATOM 3: X=0.00000000 Y=0.50000000 Z=0.00000000
      MULT= 1 ISPLIT= 2
B      NPT= 781 R0=0.00010000 RMT= 1.2200 Z: 5.0
LOCAL ROT MATRIX: 1.0000000 0.0000000 0.0000000
                  0.0000000 1.0000000 0.0000000
                  0.0000000 0.0000000 1.0000000
```

# Building Supercells with WIEN2k

- Run `x sgroup` and copy `defective.struct_sgroup` to `defective.struct`.
- Modify the `defective.in0` file as before.
- We want a charged system: we must modify `defective.in2c` and `defective.inm`.
- Add 3 valence electrons in `defective.in2c`: 253  $\rightarrow$  256

```
TOT          (TOT, FOR, QTL, EFG, FERMI)
-12.0   256.0   0.50 0.05 1  EMIN, NE, ESEPERMIN, ESEPER0, iqtlsave
TETRA    0.000   (GAUSS, ROOT, TEMP, TETRA, ALL eval)
```

- In `defective.inm`, modify `MSR1` from 0.0 to 3.0 to avoid that WIEN2k associates the extra charge to core-leaking.

```
MSR1  3.0  YES  (BROYD/PRATT, BG charge (-1 for core hole), norm)
0.20   mixing FACTOR for BROYD/PRATT scheme
1.00  1.00  PW and CLM-scaling factors
9999  8     idum, HISTORY
```

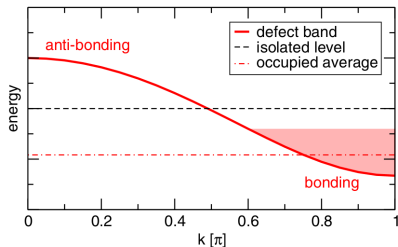
- Run the calculation *including* structure optimization (`runsp_lapw -min`).

# Quantum-Mechanical Finite-Size Effects

Due to periodic boundary conditions, the Kohn-Shame orbitals of the defective level will overlap with the neighboring cells creating a defect band.

This gives spurious contribution to  $E$  especially if defective levels with partial occupancy are considered.

From **C. Freysoldt et al., Rev. Mod. Phys. 86, 253 (2014)**:

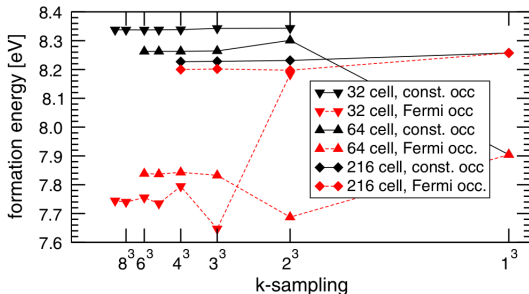


- At 0K, levels with the lowest energy are occupied.
- For a dispersive defective band, this causes an attraction between the defects.
- The wider the defect band, the larger the effect.

# Quantum-Mechanical Finite-Size Effects

For localized states,  $\psi(\mathbf{r}) \approx e^{-r}$ . For supercells with edge length  $L$ , the spurious energy due to wavefunction overlap scales as  $e^{-L}$ .

Increasing the supercell size to reasonable dimensions avoids the issue.  $V_{acC}$  in diamond (from **C. Freysoldt et al., Rev. Mod. Phys. 86, 253 (2014)**).



For delocalized states, the spatial extent of the defect wave function covers several primitive cells. However spurious effects can be avoided if only the  $\Gamma$  point is sampled. This is an accurate enough sampling if the supercell is large enough.

# Elastic Finite-Size Effects

The introduction of a defect induces some strain on the host material.

Atomic positions atom  $k$  in the pristine system:  $\mathbf{R}_k$ .

Atomic position atom  $k$  in the defective system:  $\mathbf{R}_k + \mathbf{u}_k$ .

Taylor expansion lattice energy of the defective system up to second order:

$$V(\{\mathbf{R}_k + \mathbf{u}_k\}) = V(\{\mathbf{R}_k\}) + \sum_{\alpha,k} \sum_{\beta,l} \Phi_{\alpha,\beta}(k;l) u_{\alpha,k} u_{\beta,l} \quad (14)$$

$\alpha, \beta$  range over the Cartesian coordinates.

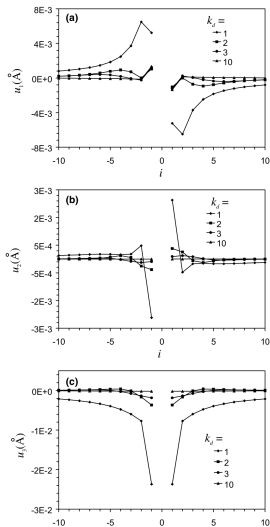
The force on atom  $k$  along  $\alpha$  is:

$$f_{\alpha}(k) = -\frac{\partial V}{\partial u_{\alpha,k}} = -\sum_{\beta,l} \Phi_{\alpha,\beta}(k;l) u_{\beta,l} \quad (15)$$

In matrix form:

$$\mathbf{f} = -\Phi \mathbf{u} \longleftrightarrow \mathbf{u} = -\mathbf{G} \mathbf{f} \quad (16)$$

## G: Lattice Green's function (B. Yang and V. K. Tewary, Int. J. Solids and Structures 42, 4722 (2005))



- $Vac_{Si}$  in Si.
- $i, j, k_d$ : index unit cell along  $x, y, -z$
- Defect at  $i = j = 0$ .
- In a bulk-like region:  $k_d = 10$
- The displacement field goes to zero fast from the defect as  $1/r^2$ .
- Defect-defect lattice energy scales as  $1/L^3$ .
- Use large enough supercell.



# Electrostatic Finite-Size Effects

- Monopole Potential

$$\psi(\mathbf{r}) = \frac{Q}{r}$$

- Dipole Potential

$$\psi(\mathbf{r}) = \frac{\mathbf{p} \cdot \mathbf{r}}{r^3}$$

- Quadrupole Potential

$$\psi(\mathbf{r}) = \frac{\mathbf{r} \cdot \Theta \cdot \mathbf{r}}{r^5}$$

$\Theta$ : the traceless quadrupole tensor.

Electrostatic interactions can be very long-ranged if the charge distribution has low multipole terms.

Electrostatic interaction energy  $\rho(\mathbf{r})$  and its images in PBC:

- Monopole:  $E \sim L^{-1}$
- Dipole:  $E \sim L^{-3}$
- Quadrupole:  $E \sim L^{-5}$

For charged defects the situation is then problematic as the lowest term is the monopole one.

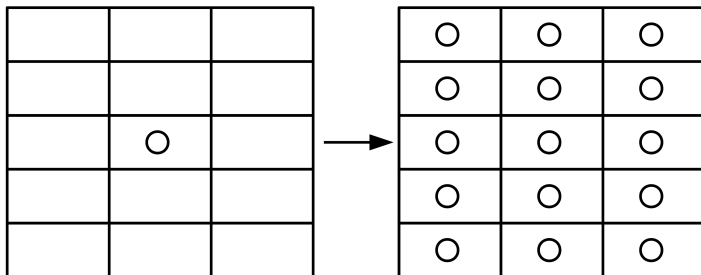
Energy of the defective supercell,  $E(X; q; L)$  converges very slowly for  $L \rightarrow \infty$ . Fitting the formation energies as a function of  $L$ , one finds (**Castleton et al., PRB 73, 035215 (2006)**):

$$E(X; q; L) = E(X; q; L \rightarrow \infty) + aL^{-1} + bL^{-3} \quad (17)$$

But extracting the  $L \rightarrow \infty$  in this way is very expensive.

Fortunately, it is possible to obtain convergent results is relatively small supercell adding a  $E_{corr}$  to  $\Delta E_f(X; q)$ .

The nature of the problem:



- Isolated charge distribution:

$$\nabla^2 \psi_{iso}(\mathbf{r}) = -4\pi \rho_{iso}(\mathbf{r}) \quad \psi_{iso}(\mathbf{r}) \rightarrow 0 \text{ as } r \rightarrow \infty$$

- Periodic charge distribution:

$$\nabla^2 \psi_{per}(\mathbf{r}) = -4\pi \rho_{per}(\mathbf{r}) \quad \psi_{per}(0, y, z) = \psi_{per}(L, y, z) \text{ etc.}$$

# Electrostatic Finite-Size Effects

The solution of the Poisson's equation will be different according to the boundary conditions.

- Isolated charge distribution:

$$\psi_{iso}(\mathbf{r}) = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}'$$

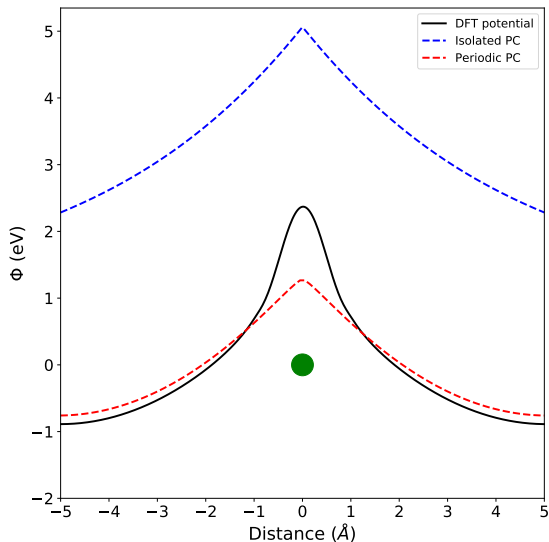
- Periodic charge distribution:

$$\psi_{per}(\mathbf{r}) = 4\pi \sum_{\mathbf{G} \neq 0} \frac{\tilde{\rho}(\mathbf{G})}{G^2} e^{i\mathbf{G} \cdot \mathbf{r}}$$

- In a supercell calculation we are interested in  $\psi_{iso}$  (dilute limit) but we obtain  $\psi_{per}$ . This affects the total energy of the system.

$$E = \frac{1}{2} \int_V \rho(\mathbf{r}') \psi(\mathbf{r}') d\mathbf{r}'$$

# Potential Alignment



- Alignment problems: average potential of a system with PBC is zero ( $\mathbf{G} = (0, 0, 0)$  Fourier coefficient of  $\psi_{per}(\mathbf{r})$  set to zero).
- $e^-$  eigenvalues, in particular  $E_{VBM}$ , in pristine and charged supercell have different references.
- Far from  $\rho$ , DFT and model potential differ by a constant: alignment term.

# Jellium Background

Moreover, the electrostatic energy of charged cells with PBC diverges: we must keep the unit cell neutral. If the cell charge density is  $\rho(\mathbf{r})$  and:

$$q = \int_V \rho(\mathbf{r}) d\mathbf{r} \quad (18)$$

But in PBC, using Poisson's equation and Gauss' law:

$$4\pi \int_V \rho_{per}(\mathbf{r}) d\mathbf{r} = \int_{\partial V} \frac{\partial \psi_{per}(\mathbf{r})}{\partial \mathbf{n}} dS = 0 \text{ with PBC} \quad (19)$$

So we must have:

$$\rho_{per}(\mathbf{r}) = \rho(\mathbf{r}) - \frac{q}{V} \quad (20)$$

$-\frac{q}{V}$  is the uniform jellium background which is added to the unit cell to make it neutral. Now the electrostatic energy is finite, however we have additional artifacts.

Ideal procedure for correcting electrostatic finite-site effects in supercell calculations of solids:

- Identify defect-induced charge density in the simulation cell:  $\rho_d(\mathbf{r})$ .  
E.g. difference electronic  $\rho$  between defective charged and defective neutral system.
- Solve Poisson's equation  $\nabla(\kappa(\mathbf{r})\nabla\psi(\mathbf{r})) = -4\pi\rho_d(\mathbf{r})$  with open and periodic boundary conditions.
- Obtain corrective potential:  $\psi_{corr}(\mathbf{r}) = \psi_{iso}(\mathbf{r}) - \psi_{per}(\mathbf{r})$
- Calculate correction on the electrostatic energy:  
$$E_{corr} = \frac{1}{2} \int_V \rho_d(\mathbf{r})\psi_{corr}(\mathbf{r})d\mathbf{r}$$
- Calculate the alignment term between model  $\rho$ , charged and pristine systems:  $\Delta V_{q/b}$
- Correct for the defect formation energy by adding:  $E_{corr} + q\Delta V_{q/b}$

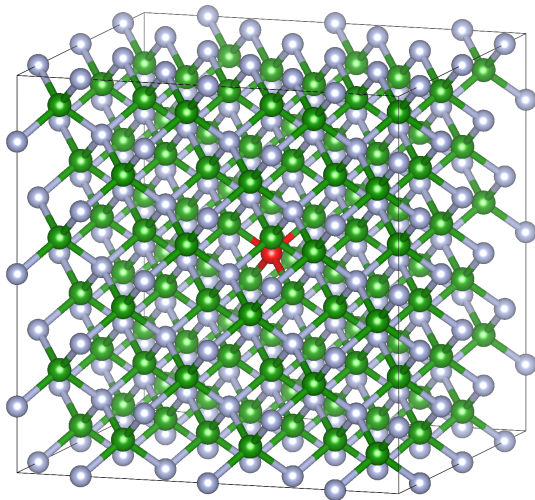
- This process can be done, both self-consistently within the DFT code or *a posteriori* (see e.g. **I. Dabo et al., PRB 77, 115139 (2008)** and **T. R. Durrant et al., J. Chem. Phys. 149, 024103 (2018)**). However, Poisson's solvers are complicated to implement on general domains and the calculations can be time consuming.
- The approach most commonly employed consist in finding good simple models for  $\rho_d(\mathbf{r})$ , e.g. a Gaussian charge distribution (**C. Freysoldt et al., PRL 102, 016402 (2009)**) and calculate analytically most of the terms.
- Simple model: point charges. Afterall, if  $\rho_d$  is spherically symmetric, its potential outside from it is a point charge potential. A system of point charges in a jellium neutralizing background has an energy  $E_{corr} = \frac{q^2\alpha}{2\kappa L}$ .  $\alpha$  is the Madelung's constant: depends only on the supercell shape.



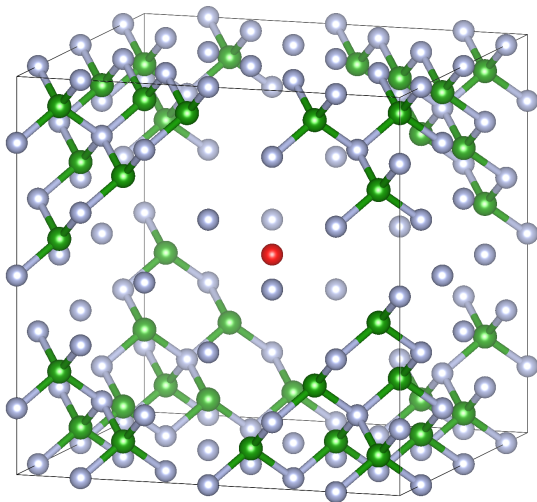
## Y. Kumagai and F. Oba, PRB 89, 195295 (2014)

- Charge distribution model: point charges.
- Potential alignment term: from atomic-site potentials for atoms in a sampling region.
- Sampling region: region between the Wigner-Seitz cell and its largest inscribed sphere.
- Pros:
  - 1 Easy to implement
  - 2 Fast execution
  - 3 Electrostatic potential value at the atomic sites is less affected by the presence of the defect → more reliable alignment.
  - 4 Straightforward extension to anisotropic systems (**G. Fischerauer, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 44, 6 (1997)**).
  - 5 Tested successfully on different materials.

## Sampling Region



## Sampling Region



Potential periodic array of point charges with charge  $q$ :

$$\begin{aligned} \psi_{per}^{PC}(\mathbf{r}) = & q \left( \frac{1}{\sqrt{\det(\boldsymbol{\kappa})}} \sum_{\mathbf{n}} \frac{\text{erfc}(\gamma \sqrt{(\mathbf{n} - \mathbf{r}) \cdot \boldsymbol{\kappa}^{-1} \cdot (\mathbf{n} - \mathbf{r})})}{\sqrt{(\mathbf{n} - \mathbf{r}) \cdot \boldsymbol{\kappa}^{-1} \cdot (\mathbf{n} - \mathbf{r})}} \right. \\ & \left. + \sum_{\mathbf{G} \neq 0} \frac{4\pi \exp(-\mathbf{G} \cdot \boldsymbol{\kappa} \cdot \mathbf{G} / 4\gamma^2)}{V \mathbf{G} \cdot \boldsymbol{\kappa} \cdot \mathbf{G}} - \frac{\pi q}{V\gamma^2} \right) \end{aligned} \quad (21)$$

$\gamma$ : Ewald parameter.

Potential alignment term:

$$\Delta V_{PC,q/b}(\mathbf{r}_i) = \psi_{defect,q}(\mathbf{r}_i) - \psi_{pristine}(\mathbf{r}_i) - \psi_{per}^{PC}(\mathbf{r}_i) \quad (22)$$

for  $\mathbf{r}_i$  in the sampling region.

Correction energy term:

$$E_{corr} = -\frac{q}{2} \lim_{r \rightarrow 0} \psi_{per}^{PC}(\mathbf{r}) \quad (23)$$

The method of Kumagai and Oba is implemented in Spinney as a stand-alone module that can be imported as:

```
from spinney.defects.kumagai import KumagaiCorr
```

- Consider the B  $-3$  vacancy in cubic BN modeled with the  $2 \times 2 \times 2$  supercell.
- From WIEN2k we obtained the `.scf`, `.struct`, `.vcoul` files for pristine and defective system.
- The defective system was obtained by removing B with fractional coordinates (0.5, 0.5, 0.5).
- We have to provide some data to `KumagaiCorr` **class** in order to compute  $E_{corr}$  and  $\Delta V_{PC,q/b}$ .
- The `ase` library can help extract most of the needed data from the `.struct` files.

---

```
class KumagaiCorr(Ewald):  
    def __init__(self, cell, positions_defective,  
                positions_pristine, defect_position,  
                defect_formal_charge,  
                dielectric_constant,  
                dft_core_potential_def,  
                dft_core_potential_prist,  
                direct_cutoff=10,  
                reciprocal_cutoff=1, alpha=None,  
                length_units='Angstrom',  
                energy_units='eV', tol_en=1e-6,  
                min_steps=2, tol_dist=1e-2):
```

---

# Method of Kumagai and Oba with Spinney

```
import numpy as np
from spinney.defects.kumagai import KumagaiCorr
import ase.io
struct_pris = ase.io.read('c-BN.struct',
                          format='struct')
struct_def = ase.io.read('defective.struct',
                          format='struct')
# Note: internal units in ase are Angstrom
cell = struct_pris.get_cell()
positions_defective = struct_def.get_scaled_positions()
positions_pristine = struct_pris.get_scaled_positions()

defect_position = np.array([0.5, 0.5, 0.5]) # the B atom we removed
defect_formal_charge = -3 # formal charge of the defect
dielectric_constant = 6.92 # calculated in another calculation

# extract the atomic-site potentials from the .vcoul files
# convert to eV
from spinney.io.wien2k import extract_potential_at_core_wien2k
dft_core_potential_def = extract_potential_at_core_wien2k('defective.struct',
                                                           'defective.vcoul')* 13.6056981
dft_core_potential_pris = extract_potential_at_core_wien2k('c-BN.struct',
                                                           'c-BN.vcoul')* 13.6056981

# create the object for the correction scheme
Corr = KumagaiCorr(cell, positions_defective, positions_pristine,
                   defect_position, defect_formal_charge, dielectric_constant,
                   dft_core_potential_def, dft_core_potential_pris)
```

---

```
# Correction energy eq. (23)
corr_ene = Corr.get_correction_energy()
# takes around 2 ms

# alignment potential eq. (22)
pot = Corr.get_potential_alignment()
# takes around 0.1 s

# overall correction energy to add to
# uncorrected defect formation energy
# E_corr = corr_ene + defect_formal_charge*pot
E_corr = Corr.get_net_correction_energy()
```

---



# Method of Kumagai and Oba with Spinney

- If we want to obtain the defect formation energy already corrected, use the **class** `PointDefect`.  
`from spinney.structures.pointdefect import PointDefect`
- Takes as argument the ase `Atoms` object representing the defective system **with attached an** ase `Calculator`. ase does not have a calculator for WIEN2k, but for our goal, we just need the total electronic energy of the system.
- `spinney.structures.pointdefect` has the **class** `DummyAseCalculator` with the method `set_total_energy(value)` which allow us to insert the system total energy.
- To obtain a ase `Atoms` object with attached a calculator that keeps the interface of `PointDefect` use:

# Method of Kumagai and Oba with Spinney

---

```
from spinney.io.wien2k import prepare_ase_atoms_wien2k

# total energy calculated by WIEN2k, in Ry
total_energy_Ry = -5100.65085079
calc_obj = prepare_ase_atoms_wien2k('c-BN.struct', 'c-BN.scf')
calc_obj.get_total_energy() # in eV, we keep ase internal units
```

---

We need to prepare the data needed for calculating the formation energy of the B -3 vacancy.

---

```
# total energy calculated by WIEN2k for the defective system
total_energy_Ry_def = -5048.05765754
calc_obj_def = prepare_ase_atoms_wien2k('defective.struct', 'defective.scf')

# total energy bulk trigonal B
# we use this value for the chemical potential of B
total_energy_Ry_B = -1789.16328336
calc_B = prepare_ase_atoms_wien2k('B.struct', 'B.scf')
mu_B = calc_B.get_total_energy()/calc_B.get_number_of_atoms()

# Valence band maximum pristine system
vbm = 0.6979264665 * 13.6056981
```

---

# Method of Kumagai and Oba with Spinney

We now have all the data for calculating the formation energy.

---

```
from spinney.structures.pointdefect import PointDefect

pd = PointDefect(calc_obj_def)
pd.set_defect_position(defect_position)
pd.set_defect_charge(defect_formal_charge)
pd.set_pristine_system(calc_obj)

# Values of the chemical potentials to be used in the calculations
# since we need only the value for B, we leave the one for N equal to None
pd.set_chemical_potential_values({'B' : mu_B, 'N' : None})

pd.set_vbm(vbm)
pd.set_dielectric_tensor(dielectric_constant)

# choose the Kumagai-Oba method for electrostatic finite-size corrections
pd.set_finite_size_correction_scheme('ko')
# add the atomic-site potentials needed by this method
pd.add_correction_scheme_data(potential_pristine=dft_core_potential_pris ,
                             potential_defective=dft_core_potential_def)

# calculate formation energy
form_ene = pd.get_defect_formation_energy() # 10.89 eV, without E_corr
form_ene_corr = pd.get_defect_formation_energy(include_corr=True) # 14.40, with E_corr
```

---

# Giving Values to the Chemical Potentials

- $\Delta G_f(X; q)$  values are determined by the chemical potentials of the atomic species and electrons affected by the creation of the point defect.
- In our grand canonical scheme, their value is fixed by the reservoir. In practice, the values depend on the environmental conditions: connection between calculations and experiments.

E.g.  $A_xB_y$  compound. Thermodynamic stability requires that the chemical potentials of A and B must have a value such that:

- $x\mu_A + y\mu_B = \mu_{A_xB_y}$
- $\mu_A \leq \mu_A(A)$
- $\mu_B \leq \mu_B(B)$
- $a\mu_A + b\mu_B \leq \mu_{A_aB_b}$

Optimization problem: find the possible ranges of  $\mu_A$  and  $\mu_B$ .

# Choosing the Competing Phases

- Several compounds made with atoms A and B might exist.
- ① Consider only the most known compounds and calculate their total energies.
- ② Make an exhaustive search using online databases of ab-initio calculations (such as AFLOW, Materials Project, etc.)
- Using databases seems desirable: no need to redo the calculations for the competing phases, which can be time-consuming for systems with a complex chemistry
- The quality of the calculations in the databases should be carefully assessed.
- Calculating the convex hull of the system is a good starting point for identifying the most stable phases, and redo only a small amount of calculations, if needed.

Find the convex hull for the Ga-N system by accessing the AFLOW database.

- Use the AFLOW REST API to retrieve the data for the Ga-N system.
- Save the data in two text files.
- One file must have two columns: the molar fraction of one component (for example N) and the formation energy of that compound with respect to a reference bulk Ga and N (whose values are found on the database).
- Ideally the references should reflect the chemistry of the system:  $\alpha$ -Ga and gas N<sub>2</sub>.

# Calculating Convex Hull with Spinney

```
# # These are the composition-formation-energy-per-atom data points
# # obtained from the aflowlib database:
# # "http://afwlib.duke.edu/AFLOWDATA/ICSD_WEB
# # For the Ga-N system. These use PBE + U with U = 3.9 eV
# # on Ga d orbitals.
# #
# # The formation energies are obtained from the reference atoms
# # alpha-Ga and N (N % = 0 and 1, respectively).
# # The Ga system was chosen in the space group 64 (Cmca)],
# # using PAM PBE pseudopotentials.
# # The identifiers relative to these compounds are found in the
# # file "all_compounds_Ga_N_ids.txt" in the same order used here.
# #
# # N %      Form. Energy per atom
# #
5.00000000000000000000e-01      -1.907819999999999849e+00
5.00000000000000000000e-01      -2.363570000000000171e+00
5.00000000000000000000e-01      -2.362620000000000609e+00
5.00000000000000000000e-01      -1.908169999999999222e+00
5.00000000000000000000e-01      -2.363520000000000287e+00
5.00000000000000000000e-01      -1.908269999999999689e+00
5.00000000000000000000e-01      -2.363220000000000098e+00
5.00000000000000000000e-01      -1.908269999999999689e+00
5.00000000000000000000e-01      -2.3633199999999999865e+00
5.00000000000000000000e-01      -2.363269999999999982e+00
5.00000000000000000000e-01      -2.3633199999999999865e+00
5.00000000000000000000e-01      -2.369394999999999918e+00
5.00000000000000000000e-01      -2.369545000000000456e+00
5.00000000000000000000e-01      -2.369120000000000559e+00
5.00000000000000000000e-01      -2.369570000000000398e+00
5.00000000000000000000e-01      -2.369744999999999990e+00
5.00000000000000000000e-01      -2.368845000000000312e+00
5.00000000000000000000e-01      -2.369570000000000398e+00
5.00000000000000000000e-01      -2.369545000000000456e+00
5.00000000000000000000e-01      -2.369449999999999801e+00
5.00000000000000000000e-01      -2.369245000000000267e+00
5.00000000000000000000e-01      -2.369520000000000515e+00
5.00000000000000000000e-01      -2.369295000000000151e+00
5.00000000000000000000e-01      -2.369495000000000573e+00
5.00000000000000000000e-01      -2.369295000000000151e+00
5.00000000000000000000e-01      -2.369744999999999874e+00
5.00000000000000000000e-01      -2.369495000000000573e+00
```

- For each compound in this file, the second file should contain the database identifiers for that compound. (Not mandatory file)

# Calculating Convex Hull with Spinney

---

```
from spinney.thermodynamics.convex_hull import BinaryConvexHull

# molar fractions of N
data_energies = 'all_compounds_Ga-N.txt'
data_ids = 'all_compounds_Ga-N_ids.txt'

all_points = np.loadtxt(data_energies, dtype=np.float64)
identifiers = np.loadtxt(data_ids, dtype=str)

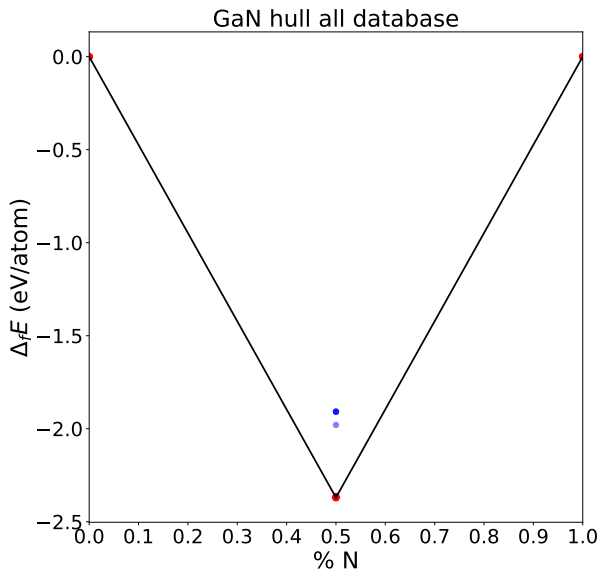
gan_hull = BinaryConvexHull(all_points, identifiers)
gan_hull.plot_hull('GaN hull all database', 'N')

# print the identifiers for the compounds on the convex hull
print('Identifiers compounds on the hull: ')
for ix in gan_hull.candidates_identifiers:
    print(ix)
```

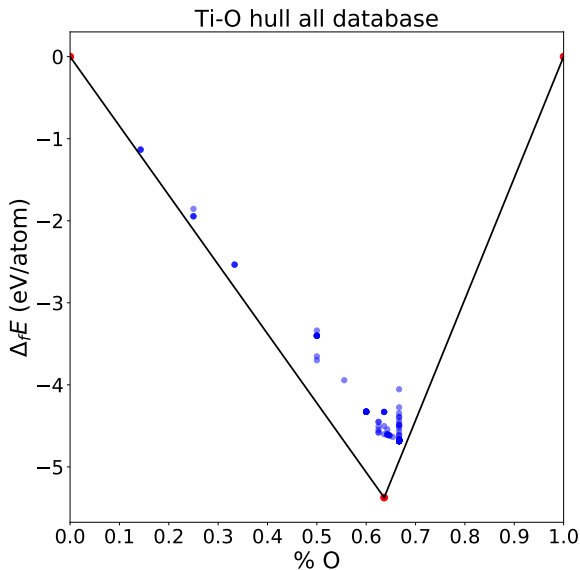
---



# Calculating Convex Hull with Spinney



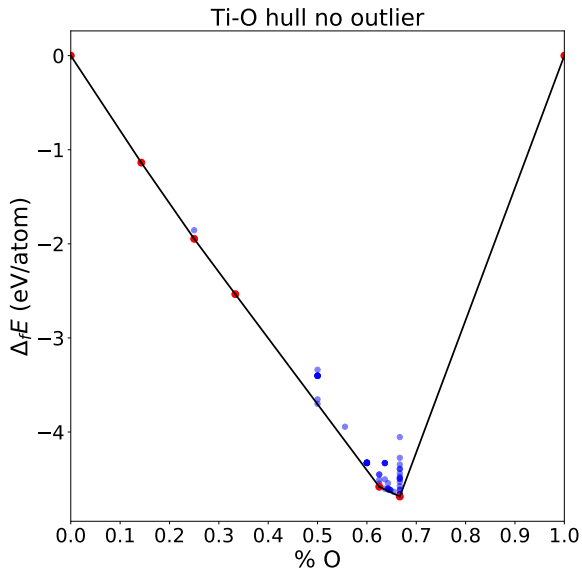
# System with a more complex chemistry: Ti-O



# System with a more complex chemistry: Ti-O

There is clearly an outlier in the data. We remove it.

# System with a more complex chemistry: Ti-O



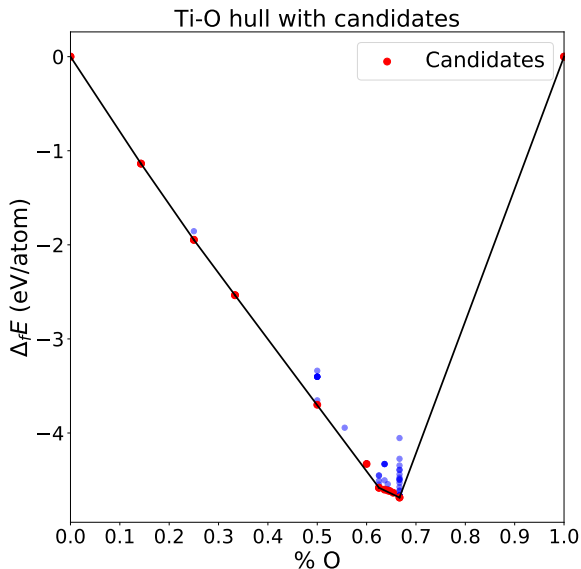
# System with a more complex chemistry: Ti-O

It looks much better, but known compounds as TiO and Ti<sub>2</sub>O<sub>3</sub> are still missing.

We instruct the `BinaryConvexHull` object to include in the candidate compounds those whose energy is within 0.1 eV above the hull:

```
ti_o_hull.add_candidates_within_tolerance(0.1)
```

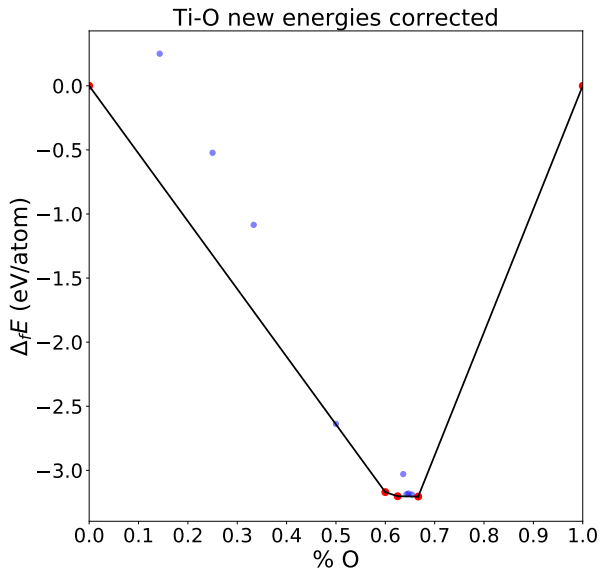
# System with a more complex chemistry: Ti-O



# System with a more complex chemistry: Ti-O

- This looks more satisfying. We can retrieve the candidates from the database (if we supplied proper identifiers) for further processing:
- `for ix in ti_o_hull.candidates_identifiers: print(ix)`
- For example, we could apply the method described in **A. Jain et al., PRB 84, 045115 (2011)** to mix GGA + U and GGA in order to obtain more accurate formation energies. GGA + U describes better systems with strongly correlated electrons, as transition-metal oxides; while GGA describes better metallic systems. Their energies are however not directly comparable.

# System with a more complex chemistry: Ti-O





# System with a more complex chemistry: Ti-O

- This looks reasonable; unfortunately TiO is still not on the hull, even though it is within a  $300k_B$  from it. This is a very reasonable agreement since we are considering systems at 0K.
- Bottom line: online repositories are a very powerful tool, but they should be used with care when dealing with complex systems. The **class** `BinaryConvexHull` offers a simple interface which can help with the task.

# Finding the allowed chemical potentials ranges with Spinney

- Consider the Ti-O system. Suppose we are interested in finding the physical limit of the chemical potentials  $\mu_{Ti}$  and  $\mu_O$  for the  $TiO_2$  anatase system.
- After using the **class** `BinaryConvexHull` we know which are the most stable competing phases:  $TiO$ ,  $Ti_2O_3$  and  $Ti_3O_5$ .
- We need to solve the optimization problem:

$$\begin{aligned}\mu_{Ti} + 2\mu_O &= \mu_{TiO_2}(anatase) \\ \mu_O &\leq \frac{1}{2}\mu_{O_2}(gas) \\ \mu_{Ti} &\leq \mu_{Ti}(bulk) \\ \mu_{Ti} + \mu_O &\leq \mu_{TiO} \\ 2\mu_{Ti} + 3\mu_O &\leq \mu_{Ti_2O_3} \\ 3\mu_{Ti} + 5\mu_O &\leq \mu_{Ti_3O_5}\end{aligned}\tag{24}$$

# Finding the allowed chemical potentials ranges with Spinney

- Its solution gives the possible values of the chemical potentials for which  $\text{TiO}_2$  anatase is thermodynamically stable. We can solve it using the Range **class** in `spinney.thermodynamics.chempots`

---

```
class Range:
    def __init__(self,
                coeff_equalities, const_equalities,
                coeff_inequalities, const_inequalities,
                bounds):
```

---

- With:  
coeff\_equalities = ((1,2))  
const\_equalities = ( $\mu_{\text{TiO}_2}$ , )  
coeff\_inequalities = ((1, 1), (2, 3), (3,5))  
const\_inequalities = ( $\mu_{\text{TiO}}$ ,  $\mu_{\text{Ti}_2\text{O}_3}$ ,  $\mu_{\text{Ti}_3\text{O}_5}$ )  
bounds = ((None,  $\mu_{\text{Ti}}(\text{bulk})$ ), (None,  $1/2\mu_{\text{O}_2}$ ))

---

```
from spinney.thermodynamics.chempots import Range
```

```
mu_range = Range(coeff_equalities, const_equalities,  
                 coeff_ineq, const_ineq,  
                 (bounds_o, bounds_ti))
```

```
# chemical potential ranges on an absolute scale
```

```
mu_range.find_variables_extrema()  
print(mu_range.variables_extrema)
```

---

- We find that:

$$\Delta\mu_{O} \equiv \mu_{O} - \frac{1}{2}\mu_{O_2}(\textit{gas}) \in [-2.81, 0] \quad (25)$$

$$\Delta\mu_{Ti} \equiv \mu_{Ti} - \mu_{Ti}(\textit{bulk}) \in [-9.48, -3.85]$$

- While  $TiO_2$  anatase can exist in equilibrium with  $O_2$ , it cannot with bulk Ti: compound of the Ti-O system richer in Ti (as  $Ti_2O_3$ ) would start to precipitate.
- Calculating the defect formation energy in the Ti-rich limit is unphysical!

# Including (some) Thermal Effects

- The value of  $\mu$  is determined by the environment. Experimentally it is possible to modify the  $\mu$  of gas species by tuning their partial pressure and T.
- We can take into account T and p effects on the defect formation energy by mixing DFT energies with experimental values: Shomate equation plus ideal gas model:

$$\mu(T, p) = \mu(0, p^\circ) + [G(T, p^\circ) - G(0, p^\circ)] + k_B T \ln\left(\frac{p}{p^\circ}\right)$$

- The module `thermodynamics.chempots` of `spinney` implements this model for some gas molecules.

---

```
from spinney.thermodynamics.chempots import OxygenChemPot

mu_O = OxygenChemPot(energy_units='eV', pressure_units='Pa')
mu_O_0K = 0.5*dft_energy_o2
T_range = [300, 500, 100]
p_range = [1e-5, 1e-3, 1, 1e3]
mu_O_values = mu_O.get_ideal_gas_chemical_potential_Shomate(mu_O_0K,
                                                            p_range, T_range)
# mu_O_values in a 2D numpy array of shape (len(T_range), len(p_range))
# containing the calculated values of mu_O
```

---

# The Electron Chemical Potential

- As for the atomic species,  $\mu_e$  is fixed by the environment; *i.e.* the reservoir.
- The system must be overall neutral considering all charged species.
- Free electrons:

$$n_0 = \int_{E_c}^{\infty} \omega(E) \frac{1}{e^{(E-\mu_e)/k_B T} + 1} dE$$

- Free holes:

$$p_0 = \int_{-\infty}^{E_{VBM}} \omega(E) \frac{1}{e^{(\mu_e-E)/k_B T} + 1} dE$$

- Ionized donors will generate extra  $n_0$  and ionized acceptors extra  $p_0$
- Neutrality condition:

$$\sum_{X,q} q c_{X,q}(\mu_e) + p_0(\mu_e) - n_0(\mu_e) = 0$$

- Solving this equation for  $\mu_e$  gives the value of the chemical potential of the electron.

# Calculating Equilibrium Defects and Carrier Concentrations

- We now have all the ingredients for calculating the equilibrium defects and carrier concentrations.
- Take the example of GaN, we consider only the intrinsic defects  $V_{acGa}$  and  $V_{acN}$ : intrinsic acceptor and donor, respectively.
- Small (72-atoms) supercell: only for illustration purposes.
- To gain insight on the system, we plot the defect formation energy as a function of  $\mu_e$ .
- We can use `spinney.defects.diagrams.Diagram`.

---

```
class Diagram:  
  def __init__(self, defects_dictionary, gap_range,  
              extended_gap_range=None,  
              electron_mu=None):
```

---

# Finding Charge Transition Levels with Spinney

```
defects_dictionary = {  
    'Vac_Ga': {0 : ef_ga_0, -1 : ef_ga_m1, ... },  
    'Vac_N': {0 : ef_n_0, ... }  
}
```

Formation energy values calculated for  $\mu_e = E_{VBM}$ .

gap\_range = (0, E\_g)

- LDA/GGA functionals underestimate the band gap.
- Transition levels in fairly good agreement with more accurate functionals (e.g. hybrids) might be obtained by simply aligning LDA/GGA band edges to the other functional's one: **J. L. Lyons et al., npj Computational Materials 12, (2017)**

extended\_gap\_range = (-vbm\_offset, -vbm\_offset + E\_g\_corr)



---

```
from spinney.defects.diagrams import Diagram
```

```
# offset VBM PBE and HSE
```

```
vbm_offset = 0.85
```

```
E_g = 1.71 # PBE
```

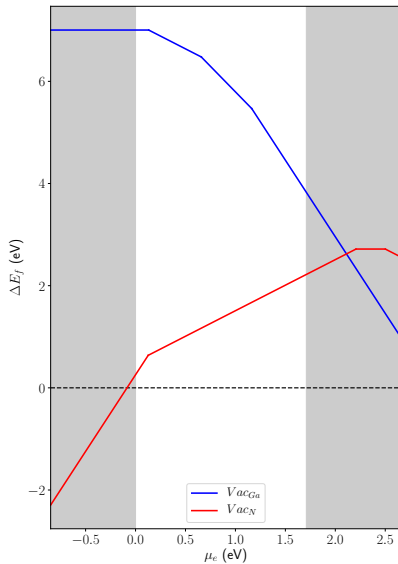
```
E_g_corr = 3.51 # HSE06
```

```
diagram = Diagram(defects_dictionary ,  
                  (0, E_g), # E_vbm for formation energies  
                  (-vbm_offset, -vbm_offset + E_g_corr))
```

```
diagram.plot(colors_dict={'Vac_Ga': 'blue', 'Vac_N': 'red'},  
             legend=True)
```

---

## Ga-rich



# Finding Charge Transition Levels with Spinney

Find charge transition levels

---

```
diagram.defects['Vac_Ga'].transition_levels
#Returns the transition levels for the Ga vacancy:
>>> defaultdict(dict,
                  {1: {0: -0.844698},
                   0: {1: -0.844698, -1: -0.2494978},
                   -1: {0: -0.2494978, -2: 0.2430350},
                   -2: {-1: 0.2430350, -3: 0.4755775},
                   -3: {-2: 0.4755775}}})

# Same thing for N vacancy
diagram.defects['Vac_N'].transition_levels
```

---

# Finding Defects Concentrations with Spinney

Equilibrium defect and carriers concentrations can be found by using the `class EquilibriumConcentrations` in `spinney.defects.concentration`.

---

```
class EquilibriumConcentrations:
    def __init__(self, charge_states,
                 form_energy_vbm, vbm, e_gap, N_conc,
                 dos, N_eff, T_range, g=None, units_energy='eV',
                 dos_down=None):
```

```
    charge_states = {'Vac_Ga':[-3, -2, -1, 0, 1],
                     'Vac_N':[-1, 0, 1, 2, 3]}
```

```
    form_energy_vbm = {
        'Vac_Ga': [val_m3, val_m2, val_m1,
                  val_0, val_1],
        'Vac_N': [valn_m1, valn_0, valn_1,
                  valn_2, valn_3]
    }
```

# Finding Defects Concentrations with Spinney

```
# sites for defects and electrons, holes
```

```
# concentrations per cell, in cm-3
```

```
N_conc = {'Vac_Ga':g_Ga,  
          'Vac_N':g_N,  
          'electron': conc_factor,  
          'hole':conc_factor}
```

```
# DOS must be consistent with the
```

```
# concentrations of electrons and holes
```

```
# in N_conc
```

```
# it is convenient to align the DOS
```

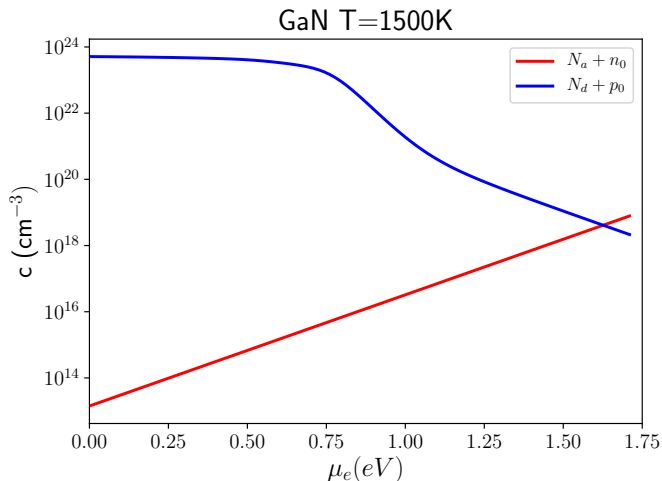
```
dos[:,0] -= hse_offset
```

```
T_range=np.linspace(300, 1500, 50)
```

```
eq_conc = EquilibriumConcentrations(charge_states,  
                                     form_energy_vbm,  
                                     vbm, E_g, N_conc,  
                                     dos, 0, T_range=T_range)
```

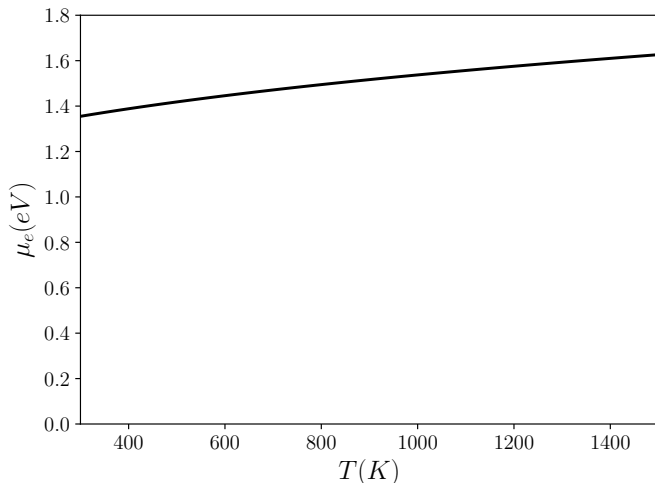
# Finding Defects Concentrations with Spinney

```
eq_conc.equilibrium_fermi_level[-1] - eq_conc.vbm  
>>> 1.626
```



# Finding Defects Concentrations with Spinney

```
plt.plot(T_range ,  
         eq_conc.equilibrium_fermi_level - eq_conc.vbm)
```

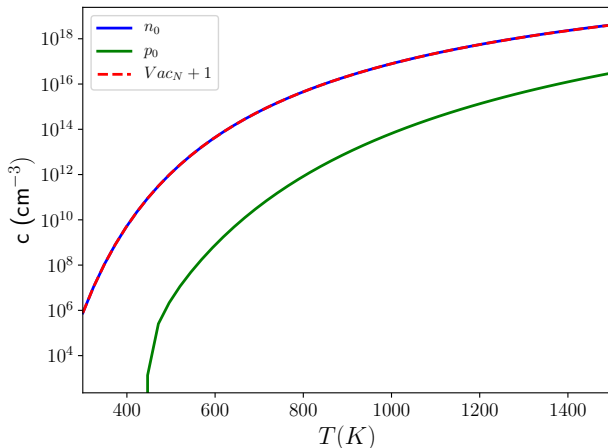


# Finding Defects Concentrations with Spinney

`eq_conc.equilibrium_defect_concentrations`

Dictionary: for each defect and charge states reports the equilibrium defect concentration for each  $T$  in  $T\_range$ .

`eq_conc.equilibrium_electron_concentrations`

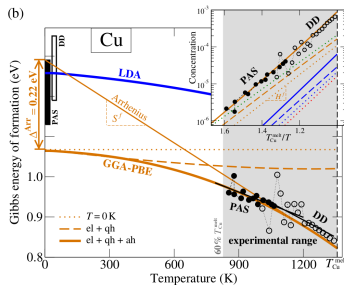
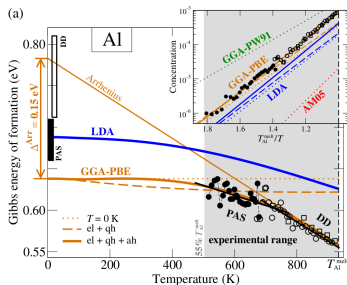




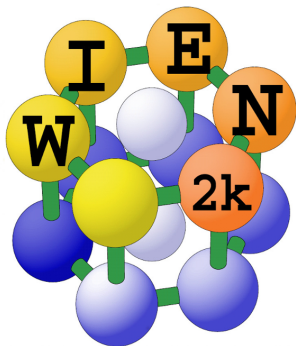
# What's more?

- Inclusion of thermal effects

- 1 Harmonic vibrational free energy impacts on  $\Delta G_f(X; q)$ .  
**T. S. Bjørheim, M. Arrigoni et al., PCCP 17, 20765 (2015),  
Chemistry of Materials 28, 1363 (2016)**
- 2 Even anharmonic terms **A. Glensk et al., PRX 4, 011018 (2014)**



- Beyond the dilute limit: defect complexes.



IMC

*marco.arrigoni@tuwien.ac.at*

The End