

[index](#) | [modules](#) | [gitlab](#) | [page source](#)

Thermochemistry

ASE contains a `ase.thermochemistry` module that lets the user derive commonly desired thermodynamic quantities of molecules and crystalline solids from ASE output and some user-specified parameters. Four cases are currently handled by this module: the ideal-gas limit (in which translational and rotational degrees of freedom are taken into account), the hindered translator / hindered rotor model (used for adsorbates, in which two degrees of freedom are translational, one is rotational, and the remaining $3N-3$ are vibrational), the harmonic limit (generally used for adsorbates, in which all degrees of freedom are treated harmonically), and a crystalline solid model (in which a lattice of N atoms is treated as a system of $3N$ independent harmonic oscillators). The first three cases rely on good vibrational energies being fed to the calculators, which can be calculated with the `ase.vibrations` module. Likewise, the crystalline solid model depends on an accurate phonon density of states; this is readily calculated using the `ase.phonons` module.

Ideal-gas limit

The thermodynamic quantities of ideal gases are calculated by assuming that all spatial degrees of freedom are independent and separable into translational, rotational, and vibrational degrees of freedom. The `IdealGasThermo` class supports calculation of enthalpy (H), entropy (S), and Gibbs free energy (G), and has the interface listed below.

```
class ase.thermochemistry.IdealGasThermo(vib_energies, geometry, potentialenergy=0.0,
atoms=None, symmetrynumber=None, spin=None, natoms=None) [source]
```

Class for calculating thermodynamic properties of a molecule based on statistical mechanical treatments in the ideal gas approximation.

Inputs for enthalpy calculations:

vib_energies :list

a list of the vibrational energies of the molecule (e.g., from `ase.vibrations.Vibrations.get_energies`). The number of vibrations used is automatically calculated by the geometry and the number of atoms. If more are specified than are needed, then the lowest numbered vibrations are neglected. If either atoms or natoms is unspecified, then uses the entire list. Units are eV.

geometry : 'monatomic', 'linear', or 'nonlinear'

geometry of the molecule

potentialenergy :float

the potential energy in eV (e.g., from `atoms.get_potential_energy`) (if potentialenergy is unspecified, then the methods of this class can be interpreted as the energy corrections)

natoms :integer

the number of atoms, used along with 'geometry' to determine how many vibrations to use. (Not needed if an atoms object is supplied in 'atoms' or if the user desires the entire list of vibrations to be used.)

Extra inputs needed for entropy / free energy calculations:

atoms :an ASE atoms object

used to calculate rotational moments of inertia and molecular mass

symmetrynumber :integer

symmetry number of the molecule. See, for example, Table 10.1 and Appendix B of C. Cramer "Essentials of Computational Chemistry", 2nd Ed.

spin :float

the total electronic spin. (0 for molecules in which all electrons are paired, 0.5 for a free radical with a single unpaired electron, 1.0 for a triplet with two unpaired electrons, such as O₂.)

get_enthalpy(*temperature*, *verbose=True*) [\[source\]](#)

Returns the enthalpy, in eV, in the ideal gas approximation at a specified temperature (K).

get_entropy(*temperature*, *pressure*, *verbose=True*) [\[source\]](#)

Returns the entropy, in eV/K, in the ideal gas approximation at a specified temperature (K) and pressure (Pa).

get_gibbs_energy(*temperature*, *pressure*, *verbose=True*) [\[source\]](#)

Returns the Gibbs free energy, in eV, in the ideal gas approximation at a specified temperature (K) and pressure (Pa).

Example

The `IdealGasThermo` class would generally be called after an energy optimization and a vibrational analysis. The user needs to supply certain parameters if the entropy or free energy are desired, such as the geometry and symmetry number. An example on the nitrogen molecule is:

```

from ase.build import molecule
from ase.calculators.emt import EMT
from ase.optimize import QuasiNewton
from ase.vibrations import Vibrations
from ase.thermochemistry import IdealGasThermo

atoms = molecule('N2')
atoms.set_calculator(EMT())
dyn = QuasiNewton(atoms)
dyn.run(fmax=0.01)
potentialenergy = atoms.get_potential_energy()

vib = Vibrations(atoms)
vib.run()
vib_energies = vib.get_energies()

thermo = IdealGasThermo(vib_energies=vib_energies,
                        potentialenergy=potentialenergy,
                        atoms=atoms,
                        geometry='linear',
                        symmetrynumber=2, spin=0)
G = thermo.get_gibbs_energy(temperature=298.15, pressure=101325.)

```

This will give the thermodynamic summary output:

```

Enthalpy components at T = 298.15 K:
=====
E_pot          0.263 eV
E_ZPE          0.076 eV
Cv_trans (0->T) 0.039 eV
Cv_rot (0->T)   0.026 eV
Cv_vib (0->T)   0.000 eV
(C_v -> C_p)    0.026 eV
-----
H              0.429 eV
=====

Entropy components at T = 298.15 K and P = 101325.0 Pa:
=====
                        S                T*S
S_trans (1 bar)  0.0015590 eV/K        0.465 eV
S_rot           0.0004101 eV/K        0.122 eV
S_elec          0.0000000 eV/K        0.000 eV
S_vib           0.0000016 eV/K        0.000 eV
S (1 bar -> P)  -0.0000011 eV/K       -0.000 eV
-----
S              0.0019695 eV/K        0.587 eV
=====

Free energy components at T = 298.15 K and P = 101325.0 Pa:
=====
H              0.429 eV
-T*S          -0.587 eV
-----
G             -0.158 eV
=====

```

Hindered translator / hindered rotor model

The hindered translator / hindered rotor model bridges the gap between the 2D gas (i.e. free translator / free rotor) and the 2D lattice gas (i.e. harmonic oscillator). For an adsorbate containing N atoms, two degrees of freedom are treated as hindered translations in the two directions parallel to the surface, one degree of freedom is treated as a hindered rotation about the axis

perpendicular to the surface, and the remaining $3N-3$ degrees of freedom are treated as vibrations. The `HinderedThermo` class supports the calculation of internal energy, entropy, free energy, and zero point energy (included in the internal energy). All of the thermodynamic properties calculated here are at the standard state surface concentration (defined here such that a 2D ideal gas at that concentration has $2/3$ the translational entropy of a 3D ideal gas at 1 bar pressure, so that $\theta^0 = 0.012$ at 298 K for a surface with 10^{15} sites/cm²). This class returns the Helmholtz free energy; if the user assumes that the pV term (in $G = U + pV - TS$) is zero then this free energy can also be interpreted as the Gibbs free energy. This class depends on the user defined translation barrier (`trans_barrier_energy`) and rotational barrier (`rot_barrier_energy`) for the adsorbate to move on the surface in order to calculate the translational and rotational degrees of freedom. To calculate the vibrational degrees of freedom, all $3N$ vibrational energies must be supplied in the `vib_energies` list and the $3N-3$ largest vibrational energies are used to calculate the vibrational contribution; this is a list as can be generated with the `.get_energies()` method of `ase.vibrations.Vibrations`. The class `HinderedThermo` has the interface described below.

```
class ase.thermochemistry.HinderedThermo(vib_energies, trans_barrier_energy,
rot_barrier_energy, sitedensity, rotationalminima, potentialenergy=0.0, mass=None, inertia=None, atoms=None,
symmetrynumber=1) \[source\]
```

Class for calculating thermodynamic properties in the hindered translator and hindered rotor model where all but three degrees of freedom are treated as harmonic vibrations, two are treated as hindered translations, and one is treated as a hindered rotation.

Inputs:

vib_energies :list

a list of all the vibrational energies of the adsorbate (e.g., from `ase.vibrations.Vibrations.get_energies`). The number of energies should match the number of degrees of freedom of the adsorbate; i.e., $3 \cdot n$, where n is the number of atoms. Note that this class does not check that the user has supplied the correct number of energies. Units of energies are eV.

trans_barrier_energy :float

the translational energy barrier in eV. This is the barrier for an adsorbate to diffuse on the surface.

rot_barrier_energy :float

the rotational energy barrier in eV. This is the barrier for an adsorbate to rotate about an axis perpendicular to the surface.

sitedensity :float

density of surface sites in cm⁻²

rotationalminima :integer

the number of equivalent minima for an adsorbate's full rotation. For example, 6 for an adsorbate on an fcc(111) top site

potentialenergy :float

the potential energy in eV (e.g., from `atoms.get_potential_energy`) (if `potentialenergy` is unspecified, then the methods of this class can be interpreted as the energy corrections)

mass :float

the mass of the adsorbate in amu (if mass is unspecified, then it will be calculated from the `atoms` class)

inertia :float

the reduced moment of inertia of the adsorbate in $\text{amu} \cdot \text{\AA}^2$ (if inertia is unspecified, then it will be calculated from the `atoms` class)

atoms :an ASE atoms object

used to calculate rotational moments of inertia and molecular mass

symmetrynumber :integer

symmetry number of the adsorbate. This is the number of symmetric arms of the adsorbate and depends upon how it is bound to the surface. For example, propane bound through its end carbon has a symmetry number of 1 but propane bound through its middle carbon has a symmetry number of 2. (if `symmetrynumber` is unspecified, then the default is 1)

get_entropy(*temperature*, *verbose=True*) [\[source\]](#)

Returns the entropy, in eV/K, in the hindered translator and hindered rotor model at a specified temperature (K).

get_helmholtz_energy(*temperature*, *verbose=True*) [\[source\]](#)

Returns the Helmholtz free energy, in eV, in the hindered translator and hindered rotor model at a specified temperature (K).

get_internal_energy(*temperature*, *verbose=True*) [\[source\]](#)

Returns the internal energy (including the zero point energy), in eV, in the hindered translator and hindered rotor model at a specified temperature (K).

get_zero_point_energy(*verbose=True*) [\[source\]](#)

Returns the zero point energy, in eV, in the hindered translator and hindered rotor model

Example

The `HinderedThermo` class would generally be called after an energy optimization and a vibrational analysis. The user needs to supply certain parameters, such as the vibrational energies, translational energy barrier, rotational energy barrier, surface site density, number of equivalent minima in a full rotation, and the number of symmetric arms of the adsorbate as it rotates on the surface. The user also needs to supply either the mass of the adsorbate and the reduced moment

of inertia of the adsorbate as it rotates on the surface or the user can supply the atoms object from which the mass and an approximate reduced moment of inertia may be determined. An example for ethane on a platinum (111) surface is:

```
from ase.thermochemistry import HinderedThermo
from numpy import array

vibs = array([3049.060670,
              3040.796863,
              3001.661338,
              2997.961647,
              2866.153162,
              2750.855460,
              1436.792655,
              1431.413595,
              1415.952186,
              1395.726300,
              1358.412432,
              1335.922737,
              1167.009954,
              1142.126116,
              1013.918680,
              803.400098,
              783.026031,
              310.448278,
              136.112935,
              112.939853,
              103.926392,
              77.262869,
              60.278004,
              25.825447])
vib_energies = vibs / 8065.54429 # convert to eV from cm^-1
trans_barrier_energy = 0.049313 # eV
rot_barrier_energy = 0.017675 # eV
sitedensity = 1.5e15 # cm^-2
rotationalminima = 6
symmetrynumber = 1
mass = 30.07 # amu
inertia = 73.149 # amu Ang^-2

thermo = HinderedThermo(vib_energies=vib_energies,
                        trans_barrier_energy=trans_barrier_energy,
                        rot_barrier_energy=rot_barrier_energy,
                        sitedensity=sitedensity,
                        rotationalminima=rotationalminima,
                        symmetrynumber=symmetrynumber,
                        mass=mass,
                        inertia=inertia)

F = thermo.get_helmholtz_energy(temperature=298.15)
```

This will give the thermodynamic summary output:

Internal energy components at T = 298.15 K:

```
=====
E_pot          0.000 eV
E_trans        0.049 eV
E_rot          0.018 eV
E_vib          0.076 eV
E_ZPE          1.969 eV
-----
U              2.112 eV
=====
```

Entropy components at T = 298.15 K:

```
=====
                        S          T*S
S_trans      0.0005074 eV/K      0.151 eV
S_rot        0.0002287 eV/K      0.068 eV
S_vib        0.0005004 eV/K      0.149 eV
S_con        0.0005044 eV/K      0.150 eV
-----
S            0.0017409 eV/K      0.519 eV
=====
```

Free energy components at T = 298.15 K:

```
=====
U          2.112 eV
-T*S       -0.519 eV
-----
F          1.593 eV
=====
```

Harmonic limit

In the harmonic limit, all degrees of freedom are treated harmonically. The `HarmonicThermo` class supports the calculation of internal energy, entropy, and free energy. This class returns the Helmholtz free energy; if the user assumes the pV term (in $H = U + pV$) is zero this can also be interpreted as the Gibbs free energy. This class uses all of the energies given to it in the `vib_energies` list; this is a list as can be generated with the `.get_energies()` method of `ase.vibrations.Vibrations`, but the user should take care that all of these energies are real (non-imaginary). The class `HarmonicThermo` has the interface described below.

class `ase.thermochemistry.HarmonicThermo(vib_energies, potentialenergy=0.0)` [\[source\]](#)

Class for calculating thermodynamic properties in the approximation that all degrees of freedom are treated harmonically. Often used for adsorbates.

Inputs:

vib_energies :list

a list of the harmonic energies of the adsorbate (e.g., from `ase.vibrations.Vibrations.get_energies()`). The number of energies should match the number of degrees of freedom of the adsorbate; i.e., $3 \cdot n$, where n is the number of atoms. Note that this class does not check that the user has supplied the correct number of energies. Units of energies are eV.

potentialenergy :float

the potential energy in eV (e.g., from `atoms.get_potential_energy()`) (if `potentialenergy` is

unspecified, then the methods of this class can be interpreted as the energy corrections)

get_entropy(*temperature*, *verbose=True*) [\[source\]](#)

Returns the entropy, in eV/K, in the harmonic approximation at a specified temperature (K).

get_helmholtz_energy(*temperature*, *verbose=True*) [\[source\]](#)

Returns the Helmholtz free energy, in eV, in the harmonic approximation at a specified temperature (K).

get_internal_energy(*temperature*, *verbose=True*) [\[source\]](#)

Returns the internal energy, in eV, in the harmonic approximation at a specified temperature (K).

Crystals

In this model a crystalline solid is treated as a periodic system of independent harmonic oscillators. The `CrystalThermo` class supports the calculation of internal energy (U), entropy (S) and Helmholtz free energy (F), and has the interface listed below.

```
class ase.thermochemistry.CrystalThermo(phonon_DOS, phonon_energies, formula_units=None,
potentialenergy=0.0) \[source\]
```

Class for calculating thermodynamic properties of a crystalline solid in the approximation that a lattice of N atoms behaves as a system of $3N$ independent harmonic oscillators.

Inputs:

phonon_DOS :list

a list of the phonon density of states, where each value represents the phonon DOS at the vibrational energy value of the corresponding index in `phonon_energies`.

phonon_energies :list

a list of the range of vibrational energies ($\hbar\omega$) over which the phonon density of states has been evaluated. This list should be the same length as `phonon_DOS` and integrating `phonon_DOS` over `phonon_energies` should yield approximately $3N$, where N is the number of atoms per unit cell. If the first element of this list is zero-valued it will be deleted along with the first element of `phonon_DOS`. Units of vibrational energies are eV.

potentialenergy :float

the potential energy in eV (e.g., from `atoms.get_potential_energy`) (if `potentialenergy` is unspecified, then the methods of this class can be interpreted as the energy corrections)

formula_units :int

the number of formula units per unit cell. If unspecified, the thermodynamic quantities calculated will be listed on a per-unit-cell basis.

get_entropy(temperature, verbose=True) [\[source\]](#)

Returns the entropy, in eV/K, of crystalline solid at a specified temperature (K).

get_helmholtz_energy(temperature, verbose=True) [\[source\]](#)

Returns the Helmholtz free energy, in eV, of crystalline solid at a specified temperature (K).

get_internal_energy(temperature, verbose=True) [\[source\]](#)

Returns the internal energy, in eV, of crystalline solid at a specified temperature (K).

Example

The `CrystalThermo` class will generally be called after an energy optimization and a phonon vibrational analysis of the crystal. An example for bulk gold is:

```
from ase.spacegroup import crystal
from ase.calculators.emt import EMT
from ase.optimize import QuasiNewton
from ase.phonons import Phonons
from ase.thermochemistry import CrystalThermo

# Set up gold bulk and attach EMT calculator
a = 4.078
atoms = crystal('Au', (0., 0., 0.),
                spacegroup=225,
                cellpar=[a, a, a, 90, 90, 90],
                pbc=(1, 1, 1))

calc = EMT()
atoms.set_calculator(calc)
qn = QuasiNewton(atoms)
qn.run(fmax=0.05)
potentialenergy = atoms.get_potential_energy()

# Phonon analysis
N = 5
ph = Phonons(atoms, calc, supercell=(N, N, N), delta=0.05)
ph.run()
ph.read(acoustic=True)
phonon_energies, phonon_DOS = ph.dos(kpts=(40, 40, 40), npts=3000,
                                     delta=5e-4)

# Calculate the Helmholtz free energy
thermo = CrystalThermo(phonon_energies=phonon_energies,
                       phonon_DOS=phonon_DOS,
                       potentialenergy=potentialenergy,
                       formula_units=4)
F = thermo.get_helmholtz_energy(temperature=298.15)
```

This will give the thermodynamic summary output:

Internal energy components at T = 298.15 K,
on a per-formula-unit basis:

```
=====
E_pot          0.0022 eV
E_ZPE          0.0135 eV
E_phonon       0.0629 eV
-----
U              0.0786 eV
=====
```

Entropy components at T = 298.15 K,
on a per-formula-unit basis:

```
=====
                        S          T*S
-----
S              0.0005316 eV/K    0.1585 eV
=====
```

Helmholtz free energy components at T = 298.15 K,
on a per-formula-unit basis:

```
=====
U              0.0786 eV
-T*S          -0.1585 eV
-----
F             -0.0799 eV
=====
```

Background

Ideal gas. The conversion of electronic structure calculations to thermodynamic properties in the ideal-gas limit is well documented; see, for example, Chapter 10 of Cramer, 2004. The key equations used in the `IdealGasThermo` class are summarized here.

C.J. Cramer. *Essentials of Computational Chemistry*, Second Edition. Wiley, 2004.

The ideal-gas enthalpy is calculated from extrapolation of the energy at 0 K to the relevant temperature (for an ideal gas, the enthalpy is not a function of pressure):

$$H(T) = E_{\text{elec}} + E_{\text{ZPE}} + \int_0^T C_P \, dT$$

where the first two terms are the electronic energy and the zero-point energy, and the integral is over the constant-pressure heat capacity. The heat capacity is separable into translational, rotational, vibrational, and electronic parts (plus a term of k_B to switch from constant-volume to constant-pressure):

$$C_P = k_B + C_{V,\text{trans}} + C_{V,\text{rot}} + C_{V,\text{vib}} + C_{V,\text{elec}}$$

The translational heat capacity is $3/2 k_B$ for a 3-dimensional gas. The rotational heat capacity is 0 for a monatomic species, k_B for a linear molecule, and $3/2 k_B$ for a nonlinear molecule. In this module, the electronic component of the heat capacity is assumed to be 0. The vibrational heat

capacity contains $3N - 6$ degrees of freedom for nonlinear molecules and $3N - 5$ degrees of freedom for linear molecules (where N is the number of atoms). The integrated form of the vibrational heat capacity is:

$$\int_0^T C_{V,\text{vib}} dT = \sum_i^{\text{vib DOF}} \frac{\epsilon_i}{e^{\epsilon_i/k_B T} - 1}$$

where ϵ_i are the energies associated with the vibrational frequencies, $\epsilon_i = \hbar\omega_i$.

The ideal gas entropy can be calculated as a function of temperature and pressure as:

$$\begin{aligned} S(T, P) &= S(T, P^\circ) - k_B \ln \frac{P}{P^\circ} \\ &= S_{\text{trans}} + S_{\text{rot}} + S_{\text{elec}} + S_{\text{vib}} - k_B \ln \frac{P}{P^\circ} \end{aligned}$$

where the translational, rotational, electronic, and vibrational components are calculated as below. (Note that the translational component also includes components from the Stirling approximation, and that the vibrational degrees of freedom are enumerated the same as in the above.)

$$\begin{aligned} S_{\text{trans}} &= k_B \left\{ \ln \left[\left(\frac{2\pi M k_B T}{h^2} \right)^{3/2} \frac{k_B T}{P^\circ} \right] + \frac{5}{2} \right\} \\ S_{\text{rot}} &= \begin{cases} 0 & \text{, if monatomic} \\ k_B \left[\ln \left(\frac{8\pi^2 I k_B T}{\sigma h^2} \right) + 1 \right] & \text{, if linear} \\ k_B \left\{ \ln \left[\frac{\sqrt{\pi I_A I_B I_C}}{\sigma} \left(\frac{8\pi^2 k_B T}{h^2} \right)^{3/2} \right] + \frac{3}{2} \right\} & \text{, if nonlinear} \end{cases} \\ S_{\text{vib}} &= k_B \sum_i^{\text{vib DOF}} \left[\frac{\epsilon_i}{k_B T (e^{\epsilon_i/k_B T} - 1)} - \ln(1 - e^{-\epsilon_i/k_B T}) \right] \\ S_{\text{elec}} &= k_B \ln[2 \times (\text{total spin}) + 1] \end{aligned}$$

I_A through I_C are the three principle moments of inertia for a non-linear molecule. I is the degenerate moment of inertia for a linear molecule. σ is the symmetry number of the molecule.

The ideal-gas Gibbs free energy is then just calculated from the combination of the enthalpy and entropy:

$$G(T, P) = H(T) - T S(T, P)$$

Hindered translator / hindered rotor. The conversion of electronic structure calculations to thermodynamic properties in the hindered translator / hindered rotor model was developed for adsorbates on close packed surfaces and is documented by Sprowl, Campbell, and Arnadottir,

2016. The key equations used in the `HinderedThermo` class are summarized here.

L.H. Sprowl, C.T. Campbell, and L. Arnadottir. Hindered Translator and Hindered Rotor Models for Adsorbates: Partition Functions and Entropies. *J. Phys. Chem. C*, **2016**, 120 (18), pp 9719-9731.

L.H. Sprowl, C.T. Campbell, and L. Arnadottir. Correction to “Hindered Translator and Hindered Rotor Models for Adsorbates: Partition Functions and Entropies”. *J. Phys. Chem. C*, **2017**, 121 (17), pp 9655-9655.

C.T. Campbell, L.H. Sprowl, and L. Arnadottir. Equilibrium Constants and Rate Constants for Adsorbates: Two-Dimensional (2D) Ideal Gas, 2D Ideal Lattice Gas, and Ideal Hindered Translator Models. *J. Phys. Chem. C*, **2016**, 120 (19), pp 10283-10297.

The $3N - 3$ largest vibrational frequencies are used to calculate the vibrational contributions to the internal energy and the entropy. The remaining three degrees of freedom are calculated from two translational contributions and one rotational contribution of the adsorbate. The energy barriers for the adsorbate to translate and rotate on a close packed surface are used to calculate the translational and rotational frequencies, respectively. From the translational and rotational frequencies, the translational and rotational contributions to the internal energy and the entropy of the adsorbate are determined. The calculation of the translational frequency is:

$$\nu_{trans} = \sqrt{\frac{W_{trans}}{2mA}}$$

where W_{trans} is the translational energy barrier, m is the mass of the adsorbate, and A is the area per surface atom, or the inverse of the surface site density. The rotational frequency is calculated as:

$$\nu_{rot} = \frac{1}{2\pi} \sqrt{\frac{n^2 W_{rot}}{2I}}$$

where W_{rot} is the rotational energy barrier, n is the number of equivalent energy minima in a full rotation of the adsorbate, and I is the reduced moment of inertia of the adsorbate about its surface bond. Two variables are now introduced, a unitless temperature

$$T_i = \frac{kT}{h\nu_i}$$

and a unitless energy barrier

$$r_i = \frac{W_i}{h\nu_i}$$

to ease the internal energy and entropy calculations.

The internal energy of the adsorbate is calculated as:

$$U(T) = E_{\text{elec}} + E_{\text{ZPE}} + E_{\text{trans}} + E_{\text{rot}} + E_{\text{vib}}$$

where E_{trans} and E_{rot} are:

$$E_i = k_B T \left(\frac{1/T_i}{\exp[1/T_i] - 1} - \frac{1}{2} - \frac{1}{(2 + 16r_i) T_i} + \frac{r_i}{2T_i} \left(1 - \frac{I_1[r_i/2T_i]}{I_0[r_i/2T_i]} \right) \right)$$

where I_n is the n th-order modified Bessel function of the first kind. Similarly for the harmonic limit, E_{vib} is:

$$E_{\text{vib}} = k_B T \sum_i^{3N-3} \left(\frac{1/T_i}{\exp[1/T_i] - 1} \right)$$

The entropy of the adsorbate is calculated as:

$$S = S_{\text{trans}} + S_{\text{rot}} + S_{\text{vib}} + S_{\text{con}}$$

where S_{trans} and S_{rot} are:

$$S_i = k_B \left(\frac{1/T_i}{\exp[1/T_i] - 1} - \ln \left[1 - \exp \left[-\frac{1}{T_i} \right] \right] - \frac{1}{2} - \frac{r_i}{2T_i} \frac{I_1[r_i/2T_i]}{I_0[r_i/2T_i]} + \ln \left[\left(\frac{\pi r_i}{T_i} \right)^{1/2} I_0 \left[\frac{r_i}{2T_i} \right] \right] \right)$$

and S_{vib} is:

$$S_{\text{vib}} = k_B \sum_i^{3N-3} \left(\frac{1/T_i}{\exp[1/T_i] - 1} - \ln \left[1 - \exp \left[-\frac{1}{T_i} \right] \right] \right)$$

S_{con} is a concentration related entropy and is calculated as:

$$S_{\text{con}} = k_B \left(1 - \ln \left[A \left(\frac{N}{A} \right)^0 \right] \right)$$

where

$$\left(\frac{N}{A} \right)^0 = e^{1/3} \left(\frac{N_A \text{ 1 bar}}{RT} \right)$$

The Helmholtz free energy is calculated as:

$$F(T) = U(T) - T S(T)$$

If the user assumes that the pV term in $H = U + pV$ is negligible, then the Helmholtz free energy can be used to approximate the Gibbs free energy, as $G = F + pV$.

Harmonic limit. The conversion of electronic structure calculation information into thermodynamic properties is less established for adsorbates. However, the simplest approach often taken is to treat all $3N$ degrees of freedom of the adsorbate harmonically since the adsorbate often has no real translational or rotational degrees of freedom. This is the approach implemented in the `HarmonicThermo` class. Thus, the internal energy and entropy of the adsorbate are calculated as

$$U(T) = E_{\text{elec}} + E_{\text{ZPE}} + \sum_i^{\text{harm DOF}} \frac{\epsilon_i}{e^{\epsilon_i/k_B T} - 1}$$

$$S = k_B \sum_i^{\text{harm DOF}} \left[\frac{\epsilon_i}{k_B T (e^{\epsilon_i/k_B T} - 1)} - \ln(1 - e^{-\epsilon_i/k_B T}) \right]$$

and the Helmholtz free energy is calculated as

$$F(T) = U(T) - T S(T)$$

In this case, the number of harmonic energies (ϵ_i) used in the summation is generally $3N$, where N is the number of atoms in the adsorbate. If the user assumes that the pV term in $H = U + pV$ is negligible, then the Helmholtz free energy can be used to approximate the Gibbs free energy, as $G = F + pV$.

Crystalline solid The derivation of the partition function for a crystalline solid is fairly straightforward and can be found, for example, in Chapter 11 of McQuarrie, 2000.

D.A. McQuarrie. *Statistical Mechanics*. University Science Books, 2000.

The treatment implemented in the `CrystalThermo` class depends on introducing normal coordinates to the entire crystal and treating each atom in the lattice as an independent harmonic oscillator. This yields the partition function

$$Z = \prod_{j=1}^{3N} \left(\frac{e^{-\frac{1}{2}\epsilon_j/k_B T}}{1 - e^{-\epsilon_j/k_B T}} \right) e^{-E_{\text{elec}}/k_B T}$$

where ϵ_j are the $3N$ vibrational energy levels and E_{elec} is the electronic energy of the crystalline solid. Now, taking the logarithm of the partition function and replacing the resulting sum with an integral (assuming that the energy level spacing is essentially continuous) gives

$$-\ln Z = E_{\text{elec}}/k_B T + \int_0^\infty \left[\ln(1 - e^{-\epsilon/k_B T}) + \frac{\epsilon}{2k_B T} \right] \sigma(\epsilon) d\epsilon$$

Here $\sigma(\epsilon)$ represents the degeneracy or phonon density of states as a function of vibrational energy. Once this function has been determined (i.e. using the `ase.phonons` module), it is a simple matter to calculate the canonical ensemble thermodynamic quantities; namely the internal energy, the entropy and the Helmholtz free energy.

$$\begin{aligned}
 U(T) &= - \left(\frac{\partial \ln Z}{\partial \frac{1}{k_B T}} \right)_{N,V} \\
 &= E_{\text{elec}} + \int_0^\infty \left[\frac{\epsilon}{e^{\epsilon/k_B T} - 1} + \frac{\epsilon}{2} \right] \sigma(\epsilon) d\epsilon \\
 S(T) &= \frac{U}{T} + k_B \ln Z \\
 &= \int_0^\infty \left[\frac{\epsilon}{T} \frac{1}{e^{\epsilon/k_B T} - 1} - k_B \ln(1 - e^{-\epsilon/k_B T}) \right] \sigma(\epsilon) d\epsilon \\
 F(T) &= U(T) - T S(T, P)
 \end{aligned}$$