

VASP的热力学

提到VASP中的热力学计算，我们不得不去抱怨一下这个软件，跟高斯相比，VASP做的实在是寒碜。高斯有一个PDF文件（Thermochemistry in Gaussian）专门介绍了热力学的计算，而且频率计算之后，都会直接给你一些自由能之类的数据。与之相比，VASP就比较简单粗暴，直接给我们基本的必需的数据，用户则需要通过后处理，来实现热力学相关的计算。这一步其实难倒了很多的用户，有不少人没学过物理化学，统计相关的知识，甚至连配分函数都不知道是啥，看到公式更是脑子直接宕机，根本不知道怎么用VASP的结果去计算热力学的数据。最简单粗暴的方式，莫过于用现成的脚本，经过这么多年的发展，网上有很多后处理软件可以实现这一功能，帮助我们去计算不同温度压力下的熵以及吉布斯自由能，省去了头疼之烦恼。

在这里，本人分享一下通过ASE去计算气相分子，表面吸附物种热力学的脚本。模型采用（1）氨气分子，（2）NH₃在Ru(0001) 表面的吸附，以及（3）第一个N-H键断裂的过渡态。这三个模型基本涵盖了表面催化相关的热力学计算，希望大家可以通过脚本中的内容，去掌握热力学计算的关键细节，不至于停留在只会使用脚本的这一浅层的状态。

下面是一些关键的点：

1) 我们用到的是ASE的热力学模块；气相分子使用“Ideal-gas limit”，表面物种（吸附，过渡态）则使用“Harmonic limit”。如果你不会使用ASE，亦或者电脑上没有安装ASE，可以先问问AI学习怎么安装使用，比如通过Anaconda创建一个ASE的虚拟环境，激活环境并通过下面的命令安装。

```
conda install conda-forge::ase
```

也可以通过ASE官网的介绍进行安装：<https://ase-lib.org/>，还可以停止浏览后面的内容，使用其他你熟悉的软件。

2) 气相分子：分子是否是线性，对称数，电子的自旋需要作为输入参数，这个具体到你自己的研究对象，基础不好可以通过查表，问AI来获得。

3) VASP需要的输入有2部分：（1）正常DFT的优化或者单点计算用以得到体系的potential energy。（2）频率计算得到振动的波数。对于表面上的频率计算来说，大家常用的方式是把slab的原子固定住，只放开表面的原子。这不是一个强制性的做法，只是这样做的时候，可以节省很多的时间和精力。当然，我们也可以放开表面上那些与分子结合的原子。需要注意的是，DFT中相对能量至关重要，计算吸附能，反应热，或者反应能垒的时候，所涉及到的体系中表面上放开的原子要保持一致。

4) 这一个关键点是3)的延伸，针对于本脚本而言。计算文件夹的命名：气相分子后面必需加一个 _gas，表面吸附物种必需加 _surf，过渡态则必需是以 TS 开头。这样有助于区分气相和表面物种，并采用不同的热力学模块进行计算。文件夹中需要有OUTCAR，CONTCAR，以及 freq 子文件夹，子文件夹中需要有频率计算的OUTCAR。具体如下：

```

(base) qli@bigbrosci VASP_thermochemistry % ls
get_GS_species.py  list                NH3_gas/                NH3_surf/
slab/              TS1/
(base) qli@bigbrosci VASP_thermochemistry % ls *
get_GS_species.py  list

NH3_gas:
CONTCAR  freq/    OUTCAR  POSCAR

NH3_surf:
CONTCAR  freq/    OUTCAR  POSCAR

slab:
CONTCAR  OUTCAR  POSCAR

TS1:
CONTCAR  freq/    OUTCAR  POSCAR
(base) qli@bigbrosci VASP_thermochemistry % ls */freq
NH3_gas/freq:
CONTCAR  OUTCAR  POSCAR

NH3_surf/freq:
CONTCAR  OUTCAR  POSCAR

TS1/freq:
CONTCAR  OUTCAR  POSCAR

```

5) 一个非常值得注意的一点就是体系的自由度，VASP在计算分子体系的时候所得到的频率结果并不是全部都能用，需要剔除那些平动，转动对应的模式。这个需要根据分子的结构去判断。

6) 运行的时候只需要输入：python3 get_GS_species.py XXX XXX 是文件夹的名字。

```

(base) qli@bigbrosci VASP_thermochemistry % tail -n 5 get_GS_species.py
T = 600 # K
species = sys.argv[1]
G, S = get_gs(species,T)
S_j_per_k_mol = entropy_ev_per_k_to_j_per_k_mol(S)
print(G, 'eV', S, 'eV/K', S_j_per_k_mol, 'J·K-1 mol-1')
(base) qli@bigbrosci VASP_thermochemistry % python3 get_GS_species.py NH3_gas/
NH3_gas -19.76815458084255 eV 0.0022809014645747286 eV/K 220.07353535003693
J·K-1 mol-1
(base) qli@bigbrosci VASP_thermochemistry % python3 get_GS_species.py NH3_surf
NH3_surf -579.4379259222848 eV 0.0008406155810852475 eV/K 81.10707352903934
J·K-1 mol-1
(base) qli@bigbrosci VASP_thermochemistry % python3 get_GS_species.py TS1
TS1 -578.0446419173567 eV 0.0004744156036228784 eV/K 45.77414708003402 J·K-1
mol-1

```

第一列是600 K时体系的Gibbs Free Energy，第二，三列是单位分别为 eV/K和J·K-1 mol-1的entropy。其实我们需要的是G，S这里打印出来主要是跟NIST的数据库进行对比，验证下我们计算的准确性。数据库中 (<https://janaf.nist.gov/tables/H-083.html>) NH3在600 K时候的S (J·K-1 mol-1): 220.615，ASE的结果是：220.074。在600 K的时候，误差为：(220.615 - 220.074) * 600 / 96485 = 0.00336 eV。

7) 如果想要批量计算不同物种的自由能，我的做法是这样的，先创建一个list文件，里面包括了我们要计算的物种，也就是对应的文件夹的名字。然后用一个for循环去运行，如下：

```

(base) qli@bigbrosci VASP_thermochemistry % cat list
NH3_gas
NH3_surf
TS1
slab
(base) qli@bigbrosci VASP_thermochemistry % for i in $(cat list); do python3
get_GS_species.py $i; done
NH3_gas -19.76815458084255 eV 0.0022809014645747286 eV/K 220.07353535003693
J·K-1 mol-1
NH3_surf -579.4379259222848 eV 0.0008406155810852475 eV/K 81.10707352903934
J·K-1 mol-1
TS1 -578.0446419173567 eV 0.0004744156036228784 eV/K 45.77414708003402 J·K-1
mol-1
slab -559.66557958 eV 0 eV/K 0.0 J·K-1 mol-1

```

8) 这个脚本调用的是ASE的热力学模块: <https://ase-lib.org/ase/thermochemistry/thermochemistry.html>。文中的数据已经打包上传, 自己可以通过AI写一个脚本实现这个功能, 只要结果能对得上, 就OK了。如果不想自己动手动脑, 也可以直接购买我写的脚本 (20块钱, 加微信BigBroSci)。