

热力学性质计算与相图分析

本节将介绍JAMIP在热力学性质计算与相图分析方面的计算流程，主要包含以下内容：

- 热力学性质计算：
 - 批量计算 Sn-O-P 体系结构
- 相图分析：
 - 搜索二元体系的稳定结构，并绘制 convex hull 图
 - 计算结构分解路径，搜索三元体系的稳定结构，并绘制 triangle zone 图

JAMIP计算准备工作

1. 程序准备

为进行JAMIP计算，您需要准备以下程序：

- VASP可执行程序与赝势库
- 安装JAMIP程序包
- 进行三元相图绘制时，需额外安装 python-ternary 依赖，可执行 `pip install python-ternary` 完成

2. 输入文件

本示例中以 Sn-O, Sn-O-P 体系[1]为例，所涉及到的竞争相和元素结构文件(POSCAR)及计算结果(OUTCAR)已经存于Input目录下。并且，在 Sn-O 和 Sn-O-P 体系中，相图分析所需的各竞争相对应的形成能数据（注，提取OUTCAR中的输出信息）已经保存在csv格式的文件中，该部分数据可从JAMIP数据库或从OUTCAR文件中导出，用户可自行测试计算结果，提取结构文件中对应POSCAR即可。

注：Scripts文件夹内提供计算形成能和生成相应csv文件格式的脚本，用户可在Scripts目录下运行。

```
>>> python formation_energy_SnOP.py # 生成 'Sn-O-P.csv' 文件
>>> python formation_energy_SnO.py # 生成 'Sn-O.csv' 文件
```

Input目录层次：

```
| -Sn-O.csv
| -Sn-O-P.csv
| -Sn-O
|   | -competing_phase
|   | -element
| -Sn-O-P
|   | -competing_phase
|   | -element
```

3. 计算参数设置

当用户使用JAMIP计算材料的形成能时，可以参考以下配置文件：

`input.py` (任务提交文件，可通过执行 `jp -i vasp` 生成)

```
vasp.program = $VASP_PATH           # VASP可执行文件路径，需要用户添加
vasp.potential = $VASP_PSUEDOPATH    # VASP赝势库路径，需要用户添加
vasp.tasks = 'relax scf'             # 只需进行结构优化和自洽场计算
vasp.xc_func = 'pbe'

vasp.force = 1e-3                    # force convergence
vasp.energy = 1e-6                   # energy convergence
vasp.cutoff = 520                    # encut setting
vasp.kpoints = 0.189                 # kspacing
```

4. 形成能计算

本示例需要计算结构的形成能，计算公式如下（不考虑化学势）：

$$\Delta E_f = E_{tot} - \sum E_{element}$$

JAMIP相图分析

注：用户可以选择在交互模型下进行下面流程，或者直接执行Input文件夹内部脚本

1. 二元相图分析

本示例使用Input目录的 `Sn-0.csv` 文件或用户自行生成csv文件。

注：用户可执行Scripts目录下 `phase-2d.py` 脚本，完成下面流程。

```
>>> from jamip.utils.phase import PhaseAnalysis

# 初始化PhaseAnalysis类，指定读取的csv文件和包含的元素
>>> phase = PhaseAnalysis.from_csv('Sn-0.csv', ['Sn', 'O'])

>>> phase.element      # 元素信息
['Sn', 'O']

>>> phase.format       # 输入结构的化学式信息
['Sn506' 'Sn02' 'Sn0' 'Sn' 'O']

>>> phase.component    # 输入结构的组分(按元素信息排列)
[[5. 6.]
 [1. 2.]
 [1. 1.]
 [1. 0.]
 [0. 1.]]
```

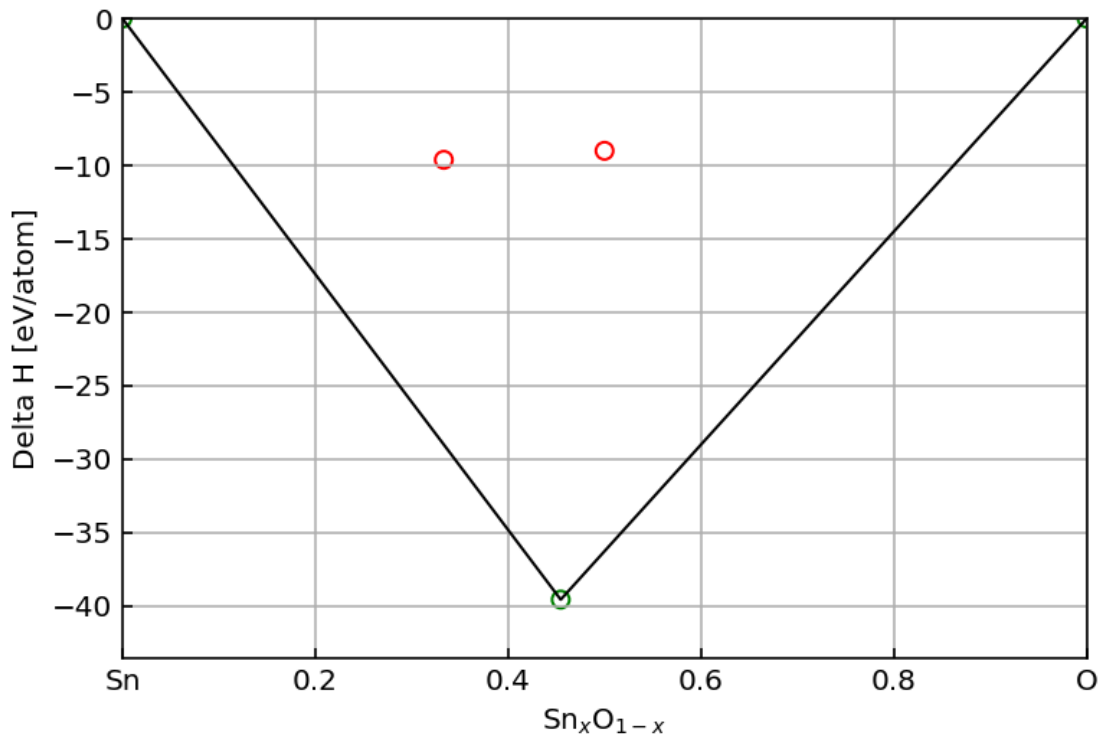
```

>>> phase.normalization_component # 元素在原子中的占比(按元素信息排列)
[[0.45454545 0.54545455]
 [0.33333333 0.66666667]
 [0.5         0.5         ]
 [1.         0.         ]
 [0.         1.         ]]

>>> stables = phase.convex_hull() # 获取二元相的稳定结构(基态结构), 返回其在输入结构中的序数
>>> stables
[4 0 3]
>>> phase.format[stables]
['O' 'Sn506' 'Sn']

>>> phase.convex_hull_diagram() # 绘制二元相图

```



2. 三元相图分析

2.1 计算结构分解路径

JAMIP计算三元以上(包括三元)分解路径需要用户提供该体系下全部的稳定相, 用户可使用Input目录下的 `Sn-O-P.csv` 文件或自行生成csv文件。

decompotion_path_output(input, stable, maximum=5, per='atom')

- `input: Union[str,list]` 输入结构的化学式或组分列表, 如: `'CsPbI3'` 或 `[1, 1, 3]` (与类属性 `element` 的顺序一致)
- `stable: list` 指定输入结构中的稳定相, 若不输入则指定全部结构

- `maximum: int, default: 5` 输出的分解路径数
- `per: str, default: 'atom'` 输出分解路径的配平方式 `'atom'` : 根据分子式中的元素占比配平, `'formula'` : 根据分子式配平

注: 用户可执行Scripts目录下 `decomposition.py` 脚本, 完成下面流程。

```
>>> from jamip.utils.phase import PhaseAnalysis

# Sn-O-P
>>> phase = PhaseAnalysis.from_csv('Sn-O-P.csv', ['Sn', 'O', 'P'])
>>> stable = phase.triangle_zone()

# Expect energy : 分解产物能量和 产物1 系数1 产物2 系数2 ...
# Expect energy 越低表示表示目标结构越稳定
# 以Sn5P2010为例, 分解路径为: Sn5P2010 -> 1/6(Sn5O6) + (1/6)Sn + Sn4P209

# 生成Sn3P208分解路径
>>> phase.decompotion_path_output('Sn3P208', stable, maximum=3)
Decomposition Path of Sn3P208 (per atom)
Expect energy : -156.4998      P          0.0081  Sn4P209  0.7895  SnP207   0.2024
Expect energy : -155.0584     P205     0.1346  Sn4P209  0.8654
Expect energy : -144.1550     O         0.0962  P         0.0385  Sn4P209  0.8654

# 生成Sn5P2010分解路径
>>> phase.decompotion_path_output('Sn5P2010', stable, maximum=5, per='formula')
Decomposition Path of Sn5P2010 (per formula)
Expect energy : -151.2468     Sn5O6    0.1667  Sn         0.1667  Sn4P209  1.0000
Expect energy : -146.9816     O         1.0000  Sn         1.0000  Sn4P209  1.0000
Expect energy : -146.0756     Sn5O6    0.2381  P          0.0952  Sn4P209  0.9524
Expect energy : -85.4133      Sn5O6    0.5000  Sn         1.5000  SnP207   1.0000
Expect energy : -72.6177      O         3.0000  Sn         4.0000  SnP207   1.0000
```

2.2 三元稳定相相图

得到三元分解路径后, 用户可根据结果生成三元相图, 可使用Input目录下的 `Sn-O-P.csv` 文件或自行生成csv文件。

注: 用户可执行Scripts目录下 `phase-3d.py` 脚本, 完成下面流程。

```
>>> from jamip.utils.phase import PhaseAnalysis

>>> phase = PhaseAnalysis.from_csv('Sn-O-P.csv', ['Sn', 'O', 'P'])
# 初始化PhaseAnalysis类, 指定读取的csv文件和包含的元素

>>> phase.element
['Sn', 'O', 'P']
>>> phase.format
['Sn5O6' 'P203' 'Sn3P208' 'Sn5P2010' 'SnP' 'P205' 'SnP207' 'SnO2'
'Sn2P207' 'Sn4P209' 'SnO' 'Sn' 'O' 'P']

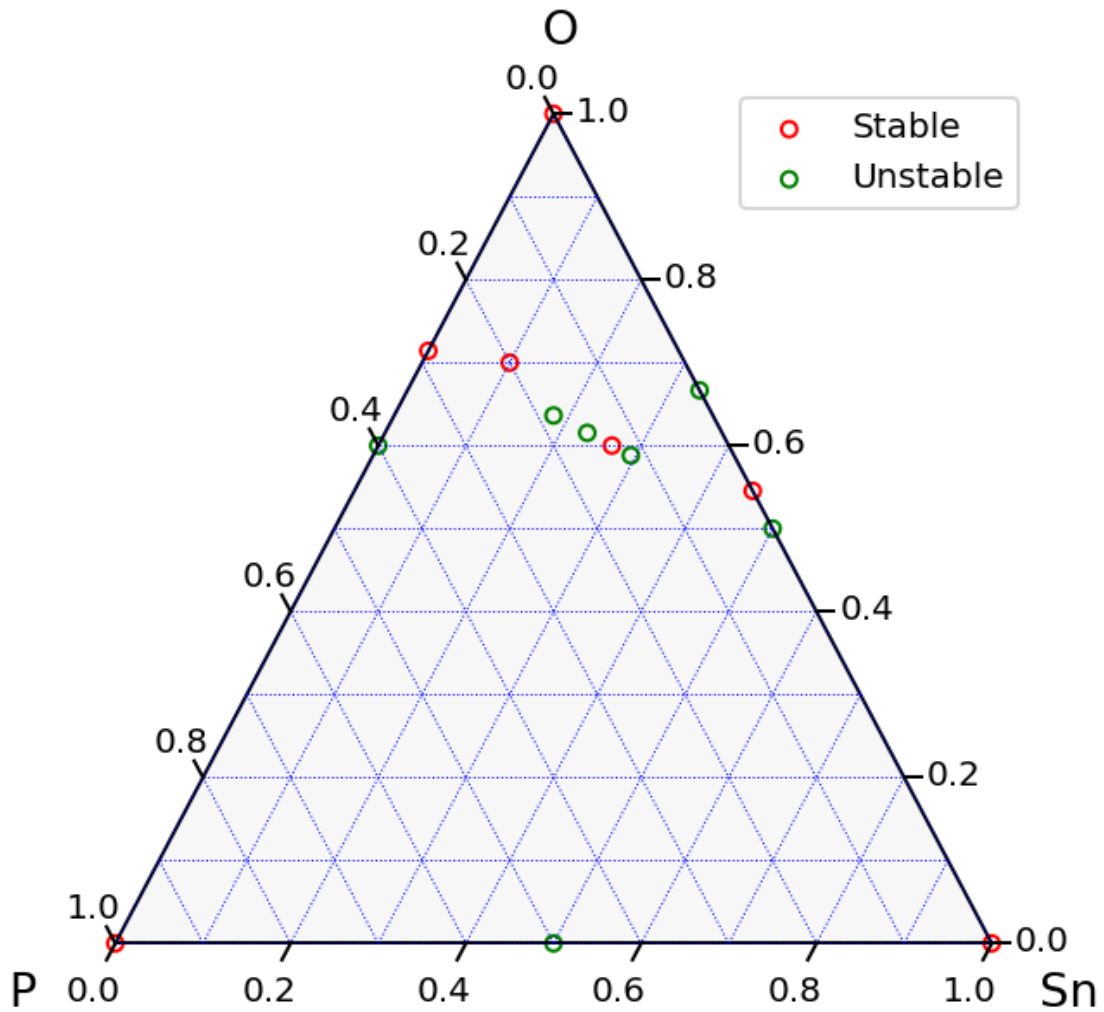
# component, normalization_component 用法与二元相图用法相同
```

```

>>> stables = phase.triangle_zone() # 获取三元相的稳定结构, 返回其在输入结构中的序数
>>> stables
[0, 5, 11, 12, 13, 9, 6]
>>> phase.format[stables]
['Sn5O6' 'P2O5' 'Sn' 'O' 'P' 'Sn4P2O9' 'SnP2O7']

>>> phase.triangle_zone_diagram() # 绘制三元相图

```



[1] "Dopability of divalent tin containing phosphates for p-type transparent conductors," T. Li, Y. Li, M. Faizan, H. Peng, and L. Zhang, Phys. Rev. Materials 3, 124606 (2019).