
eDoping

发行版本 0.4.2

Jianbo Zhu, Jingyu Li, Peng-Fei Liu

2025 年 08 月 29 日

相关文档

1	主要功能	2
2	程序安装	2
3	快速开始	4
3.1	文件结构	4
3.2	能量计算	5
3.3	化学势计算	5
3.4	修正项	5
3.5	后处理	7
4	点缺陷形成能计算	7
4.1	缺陷晶胞构建与体系能量计算	8
4.2	化学势计算与数据库使用	8
5	命令行使用参考	10
6	输入/输出文件	14
6.1	EDOPING.in	14
6.2	EDOPING.log	15
6.3	EDOPING.dat	16
6.4	EDOPING.cmpot	16
7	文章引用	16
	索引	17



1 主要功能

- 流程化带电点缺陷的形成能计算
- 完全基于命令行，无需编程和复杂的脚本
- 可以进行多组元体系的元素化学势估计
- 从 OQMD/MaterialsProject 快速获取竞争相结构和形成能
- 缺陷形成能的修正项计算
- 自洽费米能级的估计

2 程序安装

eDoping 程序包基于 python3 软件，确保它已经被正确安装。如果可以联网，最简单的方式就是通过 pip 安装（或更新）eDoping 程序包：

```
$ pip install -U eDoping
```

如果无法联网或者对源代码感兴趣，程序包从 Github 下载，或者利用 git：

```
$ git clone https://github.com/JianboHIT/eDoping.git
```

对于中国大陆地区，也可以从 Gitee 获取源码，与上面类似：

```
$ git clone https://gitee.com/joulehit/eDoping.git
```

下载源码后，进入文件夹（如果是压缩文件需要先解压缩），确保网络通畅，我们可以通过 pip（或 pip3）工具来自动安装程序以及所有依赖的库（实际上只依赖于 numpy 和 sicpy）：

```
$ pip install .
```

安装完成，我们就可以通过 edp 命令来使用 eDoping 程序包了。通过 -h（或者 --help）选项可以打印帮助信息，同时检查安装是否成功：

```
$ edp -h
usage: edp [-h] [-v] [-q] Subcommand ...

Point Defect Formation Energy Calculation - v0.4.0

options:
```

(续下页)

```

-h, --help          show this help message and exit
-v, --verbosity     increase output verbosity
-q, --quiet         only show key output

Tips:
  Subcommand      Description
  cal             Calculate defect formation energy
  energy          Read final energy from OUTCAR
  ewald           Read Ewald from OUTCAR
  volume          Read volume from OUTCAR
  epsilon         Read epsilon from OUTCAR
  evbm            Read VBM from EIGENVAL
  fixchg          Produce charge-fixed inputs
  boxhyd          Place a single hydrogen atom in the box
  refine          Refine the unit cell
  replace         Replace atoms X by Y
  groupby         Group atoms by radial distribution function
  diff            Show difference between two POSCAR
  query           Fetch data from OQMD website
  chempot         Calculate chemical potential
  trlevel         Calculate transition levels
  scfermi         Calculate sc-fermi level
  fzfermi         Calculate fz-fermi level

>>>>>>>>> Citation of EDOPING <<<<<<<<<<
If you have used EDOPING, please cite the following article:
J. Zhu, J. Li, P. Liu, et al, eDoping: A high-throughput software
package for evaluating point defect doping limits in semiconductor
and insulator materials, Materials Today Physics, 55 (2025) 101754.
DOI: 10.1016/j.mtphys.2025.101754

```

我们可以进一步查看子命令的帮助信息:

```

$ edp replace -h
usage: edp replace [-h] [-p fa fb fc] [-i FILENAME] [-o FILENAME] X Y

positional arguments:
  X                      Name of previous atom
  Y                      Name of present atom

options:
  -h, --help            show this help message and exit
  -p fa fb fc, --position fa fb fc
                        Fractional coordinates of new interstitial atom
  -i FILENAME, --input FILENAME
                        Input filename(default: POSCAR)
  -o FILENAME, --output FILENAME
                        Output filename(default: POSCAR)

```

至此，我们就已经成功地安装了 eDoping 程序包。

可选地: python 作为一种解释性编成语言，因此每次运行都需要完全地加载相应的环境或者虚拟环境。对于个人设备这是非常方便的，但是对于大型公共计算平台，这就尤为不便了。一个解决办法就是将程序打包成独立可执行程序，这样它就能和普通的程序一样不再依赖 python 环境。我们程序开发之初就有考虑到这个问题，因此严格控制对于第三方库的依赖，尽量以 python 的标准库来实现。我们在程序源码包中，包含了一个 standalone 的文件夹，其中包含一个 compile_for_linux.sh 脚本，可以帮助我们完成独立可执行程序的构建。这里，我们需要准备一个干净的 python 虚拟环境，并安装 pyinstaller 和其它的 eDoping 依赖库，然后运行下面

的命令:

```
$ cd standalone
$ bash compile_for_linux.sh
```

当脚本运行成功后, 在 `standalone/dist` 中就得到了一个可执行程序 `edp`, 可以把它移动到任何需要的位置, 方便日常工作。

3 快速开始

确保已正确安装 **eDoping** 程序包 (详细参考[程序安装](#) 部分), 可以通过 `edp -h` 打印帮助信息。密度泛函计算以 VASP 软件为例, 理论上也可采用其它计算能量的软件, 但是目前的接口并未完全支持, 有待后续完善。

3.1 文件结构

以计算 NbFeSb 带有 Mn 和 Ni 间隙的缺陷为例 (详见 Github/Gitee 仓库中的 `examples/`), 这里我们推荐按照如下目录形式组织文件 (前置的编号方便 Tab 键快速补全):

```
NbFeSb_Interstitials
├── 1.perfect
├── 2-1.defect-Mn_i
│   ├── charge_+1
│   ├── charge_+2
│   ├── charge_+3
│   ├── charge_-1
│   ├── charge_-2
│   ├── charge_-3
│   ├── charge_0
│   └── relax
├── 2-2.defect-Ni_i
│   ├── charge_+1
│   ├── charge_+2
│   ├── charge_+3
│   ├── charge_-1
│   ├── charge_-2
│   ├── charge_-3
│   ├── charge_0
│   └── relax
├── 3.phases
│   ├── NbFeSb_with_Mn
│   ├── NbFeSb_with_Ni
│   ├── elemental_Fe
│   ├── elemental_Mn
│   ├── elemental_Nb
│   ├── elemental_Ni
│   └── elemental_Sb
├── 4-1.corr-dielectric
├── 4-2.corr-hydrogen
├── EDOPING.Mn_i.in
└── EDOPING.Ni_i.in
```

3.2 能量计算

这是缺陷计算的核心耗时部分，需要调用 VASP 软件计算得到完美晶胞和所有价态下缺陷胞的能量（注意确保所有结构被合理弛豫至收敛），即 1.perfect 和 2-X.defect-XX 目录下所有的子文件夹。

为了模拟不同电荷量的缺陷，需要在 INCAR 文件中通过 NELECT 参数设置体系总电子数目。为简化该过程，可以使用 `edp fixchg` 命令从不带电的计算文件夹自动生成带电缺陷的计算文件夹 `charge_+1`、`charge_-1` 等：

```
$ edp fixchg -i charge_0 +1 -1 +2 -2 +3 -3
```

这里 `charge_0` 中包含结构不带电时自洽计算所需的文件，上述命令会完全复制该文件夹为 `charge_+1`、`charge_-1` 等文件夹，并修改其中 INCAR 文件中的 NELECT 参数使其净电荷等于给定值。

3.3 化学势计算

在缺陷形成能计算中，原子化学势项用于表达缺陷晶胞与完美晶胞之间原子种类和数目不守恒导致的能量变化。一种原子的化学势由两部分构成，即 $\mu_i = \mu_i^\ominus + \Delta\mu_i$ ，其中 μ_i^\ominus 表示该元素单质中平均每个原子的能量，这部分需要通过理论计算单质材料或者实验手段获得；而对于 $\Delta\mu_i$ ，需要通过热力学稳定性条件获得其范围。严格来说，我们需要计算所有潜在竞争相的形成能，具体的竞争相通过查询数据库（比如 OQMD，MaterialsProject，AFLOW 等）获得。

这里，我们提供了一个查询命令 `edp query`，可以直接从 OQMD 数据库获取所有竞争相的形成能（从 v0.4 开始支持 MaterialsProject 数据库，详见 `edp query` 介绍）。比如，对于含有 Mn 原子缺陷的 NbFeSb，可以通过如下命令获得 $E_{\text{hull}} < 0.01$ eV/atom 的所有竞争相的形成能（在 3.phases/NbFeSb_with_Mn 目录下运行）：

```
$ edp query NbFeSb -x Mn --ehull 0.01
```

结果保存在文件 `EDOPING.cmpot` 中。这里如果省略 `--ehull` 选项，则会获取所有稳定和亚稳竞争相的能量。此外，可以通过 `-s/--structure` 选项同时下载所有竞争相的结构文件（POSCAR 格式），从而进行更加精准的能量计算：

```
$ edp query NbFeSb -x Mn --ehull 0.01 --structure
```

然后根据计算的结果手动准备 `EDOPING.cmpot` 文件。基于 `EDOPING.cmpot` 文件，就可以通过 `edp chempot` 命令根据化学环境确定元素的化学势：

```
$ edp chempot -n
```

注意这里的 `-n`（或者 `--norm`）选项表示文件中的形成能单位是 eV/atom。如果形成能是对应组分下的晶胞总能，则不需要该选项。`EDOPING.cmpot` 文件会被自动读取，不同环境下的化学势（ $\Delta\mu_i$ ）会打印在屏幕上。

3.4 修正项

在点缺陷计算时，由于有限尺寸的限制通常需要对获得的形成能进行修正，即公式中的 E_{corr} 项。各种修正项中，镜像电荷修正需要额外提供介电常数和马德龙常数。如果需要考虑该修正机制，可以通过下面的步骤通过 VASP 计算得到介电常数和马德龙常数。

在 VASP 中，针对完美结构的原始晶胞，可以参考下面的 INCAR 参数来获得介电常数（参考 4-1.corr-dielectric/INCAR）：

```

Global Parameters
ISTART = 0          (Read existing wavefunction; if there)
ISPIN  = 1          (Non-Spin polarised DFT)
LREAL  = .FALSE.    (Projection operators: automatic)
ENCUT  = 500        (Cut-off energy for plane wave basis set, in eV)
PREC   = Accurate   (Precision level)
LWAVE  = .FALSE.    (Write WAVECAR or not)
LCHARG = .FALSE.    (Write CHGCAR or not)

Static Calculation
NSW    = 1
IBRION = 8
ISMEAR = 0          (gaussian smearing method)
SIGMA  = 0.01       (please check the width of the smearing)
NELM   = 60         (Max electronic SCF steps)
EDIFF  = 1E-08      (SCF energy convergence; in eV)

Macroscopic Dielectric Tensor
LEPSILON = .TRUE.
LPEAD    = .TRUE.

```

等待计算完成后，可以通过 `edp epsilon` 命令打印 OUTCAR 文件中介电常数的信息：

```

$ edp epsilon -f 4-1.corr-dielectric/OUTCAR
HEAD OF MICROSCOPIC STATIC DIELECTRIC TENSOR (INDEPENDENT PARTICLE, excluding Hartree-
↪and local field effects)
-----
25.438394      0.000000      -0.000000
0.000000      25.438394      0.000000
-0.000000     -0.000000      25.438394
-----

MACROSCOPIC STATIC DIELECTRIC TENSOR (including local field effects in DFT)
-----
24.482055      0.000000      -0.000000
0.000000      24.482055      -0.000000
-0.000000      0.000000      24.482055
-----

MACROSCOPIC STATIC DIELECTRIC TENSOR (including local field effects in DFT)
-----
24.482055      0.000000      -0.000000
0.000000      24.482055      -0.000000
-0.000000      0.000000      24.482055
-----

MACROSCOPIC STATIC DIELECTRIC TENSOR IONIC CONTRIBUTION
-----
19.549608     -0.000000     -0.000000
-0.000000     19.549608      0.000000
-0.000000      0.000000     19.549608
-----

```

从显示结果看到，离子贡献的介电常数为 19.55（最后一个张量），电子贡献的介电常数为 24.48（倒数第二个张量），因此总介电常数为 43.93。

对于马德隆常数，我们可以在和超胞同等尺寸的晶胞中放置一个单氢原子，VASP 自洽计算后 OUTCAR 文件中会包含相应的马德龙常数。这里提供了 `edp boxhyd` 命令从超胞 POSCAR 文件产生仅包含单氢原子的

同尺寸 POSCAR 文件（在 4-2.corr-hydrogen 目录下运行，其中包含超胞 POSCAR 文件）：

```
$ edp boxhyd
```

运行后，得到 POSCAR.H 文件，对其进行自洽计算，通过 `edp ewald` 命令可以获得马德龙常数：

```
$ edp ewald -f 4-2.corr-hydrogen/OUTCAR
Final (absolute) Ewald: 1.7152
```

即该超胞的马德隆常数为 1.7152。

3.5 后处理

根据前面的信息准备 `EDOPING.in` 文件如下：

```
DPERFECT = 1.perfect
DDEFECT  = 2-1.defect-Mn_i
CMPOT    = 0 -9.0147
VALENCE  = -3 -2 -1 0 1 2 3
# PREFIX = charge_
# DDNAME  = auto
EVBM     = inf
ECBM     = inf
PENERGY  = inf
PVOLUME  = inf
EWALD    = 1.7152
EPSILON  = 44.03
BFTYPE   = 2
EMIN     = -1
EMAX     = 2
NPTS     = 3001
```

然后调用 `edp cal` 命令进行计算：

```
$ edp cal -i EDOPING.in
```

运行结束后，会生成 `EDOPING.log` 和 `EDOPING.dat` 文件，分别记录了程序的运行日志和计算结果。

4 点缺陷形成能计算

在第一性原理的计算框架下，这里所有的计算都围绕能量（或者也被称为焓值）计算进行。对于一个带电量为 q 的缺陷 D ，其形成能定义为：

$$\Delta H_D^q(E_F) = E_D^q - E_{perfect} - \sum_i n_i \mu_i + qE_F + E_{corr}$$

这里， E_D^q 表示带电量为 q 的缺陷 D 的超胞的能量， $E_{perfect}$ 表示对应的完美超胞的能量， μ_i 表示形成缺陷过程中失去 ($n_i < 0$) 或者加入 ($n_i > 0$) 的原子的化学势， n_i 为对应的原子数量， E_F 是实际缺陷体系的费米能级， E_{corr} 是一些能量修正项，比如来自于周期边界条件的影响、静电势的变化等等。通常情况下，我们不能够准确定位体系费米能级的位置，但是能够知道它位于带隙附近。因此，我们通常是给出 $\Delta H_D^q - E_F$ 关系曲线，因此这里我们将形成能表示为费米能级的函数。接下来，我们将逐步解释其它每一项的计算，以及最终的数据处理过程。

4.1 缺陷晶胞构建与体系能量计算

完美晶胞的能量比较容易获得，因此我们这里将从缺陷结构的能量计算谈起。我们首先考虑单一点缺陷的晶胞结构构建，包括空位、置换和间隙。对于空位和间隙缺陷，通常我们可以直接手动修改 POSCAR 文件获得缺陷结构，由于这个过程中我们不需要改变原子位置列表顺序。对于取代缺陷，可以利用 `edp replace` 从 POSCAR 文件来构建结构。也可以借助一些晶体学可视化工具来辅助我们产生缺陷结构，比如免费的 VESTA 软件。

当我们需要考虑更加复杂的缺陷时，可能的超胞结构构型数量将急剧增长，利用结构的对称性我们能够有效减小所需的计算量。对于比较简单情况，我们可以利用结构可视化程序进行观察分析，排除对称等价的缺陷，但是对于复杂的结构我们就很难处理了。另外，专门处理这方面问题的软件和程序也非常有限。在我们的软件中，集成了一个 `edp groupby` 命令，可以用来辅助我们筛选出不等价的缺陷。当我们需要在一个已包含缺陷的结构上需要再引入一个缺陷时，我们舍弃了从传统的对称性来考虑等价性，而是从近邻的环境进行分析，将具有相似环境的原子归为一组，从而找出具有代表性的缺陷。由于点缺陷的局域特性，近邻分析可能是一种更加直接有效的方式来确定候选复合缺陷构型的方式。

当缺陷结构构造好后，我们可以通过 `edp diff` 命令来对比原始的晶胞和当前晶胞的差异，确保我们构造的构型是我们想要的。

参见：

- `replace` - 产生原子取代结构
- `groupby` - 不等价原子位置分析
- `diff` - 晶体结构对比与分析

当缺陷结构构建好后，我们将需要花费一定的时间来弛豫晶胞的结构，从而获得收敛的能量值。而且，我们需要改变每种缺陷结构体系的电子数目，来模拟不同的带电情况（VASP 程序 INCAR 中的 NELECT 参数），得到相应的能量值。

参见：

- `fixchg` - 准备不同电荷数的计算文件

对于 VASP 软件，如果结构优化/自洽计算正常结束，我们可以通过 `grep` 命令配合 `tail` 命令从 OUTCAR 读取能量：

```
$ grep 'energy without entropy' OUTCAR | tail -n 1
energy without entropy=          -755.64631647 energy(sigma->0) =          -755.65114440
```

这个例子中，体系的能量值为 -755.646 eV。也可以通过 `edp energy` 命令从 OUTCAR 文件中读取能量值。

参见：

- `energy` - 从 OUTCAR 读取体系能量值。

4.2 化学势计算与数据库使用

在我们完成缺陷结构的构建和相关的计算后，应该可以注意到一个重要的事情：缺陷结构和相应的完美结构很难保持原子数目的守恒。为了评估缺陷的形成能，我们就必须要消除原子本身的能量的差异，也就是我们这里所说的化学势。一个直接的想法是，我们可以用相应的单质材料计算来评估单个原子的能量。然而事实却是，这是一种非常粗糙的评估，伴随有严重的系统误差。我们可以想象，我们目标化合物中原子的能量，一定是低于单质中原子的能量，否则我们的目标化合物将会分解成单质来降低系统的能量。这里，一般将化合物中原子的能量称为化学势 μ_i ，将单质中原子的能量称为标准化学势 μ_i^\ominus ，然后有 $\mu_i = \mu_i^\ominus + \Delta\mu_i$ ，这里我们的目标就是确定 $\Delta\mu_i$ 的大小。遗憾的是，目前没有办法来给出一个确切的 $\Delta\mu_i$ 值，我们能作的就是进行范围估计，然后根据具体的实验环境进一步确定其值的大小。

按照我们前面的讨论，我们可以明确的知道一定有

$$\Delta\mu_i < 0$$

另外一方面，按照能量守恒，我们知道化合物中所有元素的内能改变量就是该化合物的形成焓 ΔH_{comp} 也就是

$$\sum_i c_i \cdot \Delta\mu_i = \Delta H_{comp}$$

这里，假设 $c_1 + c_2 + \dots + c_N = 1$ ，而且 ΔH_{comp} 为平均每个原子的形成焓。我们由此可以确定 $\Delta\mu_i$ 的下边界：

$$c_i \cdot \Delta\mu_i > \Delta H_{comp}$$

在实验中，称 $\Delta\mu_i = 0$ 时的 μ_i 为 “rich”，称 $\Delta\mu_i = \Delta H_{comp}/c_i$ 时的 μ_i 为 “poor”。

对于二元化合物，我们不难注意到，当一种元子的化学势为 “rich” 时，另外一种原子的化学势必然为 “poor”。因此，我们通常会给出两种原子分别为 “rich” 的情况来计算缺陷形成能，反映了化学环境从一个极端到另外一个极端的情况，真实的实验情况必然介于这两个极限情况之间。

随着元素种类增加到三种时，“poor” 和 “rich” 的概念就比较复杂了，因为当一种原子为 “rich” 时，另外两种原子的情况我们并不能确定，我们不得不进行细致的分类讨论，从而尽可能的接近实验环境。

尽管如此，这个范围依然太粗糙了。目前，最有效的进一步缩小化学势范围的办法就是考虑加入竞争相的考虑。按照我们前面的分析不难想到，目标化合物中各原子的化学势之和必然小于竞争性的形成焓，否则实验中就应该是形成更 “稳定” 的竞争相而不我们的目标相。由此我们可以引入一系列的不等式约束：

$$\sum_i c_{j,i} \cdot \Delta\mu_{j,i} \leq \Delta H_{comp,j}$$

这里的角标 j 表示第 j 竞争相。在这一系列的不等式约束下，化学势的范围会更加精细，可行域的形状也会变得更加复杂。

在我们的程序设计中，摒弃了对可行域形状的讨论，而是将注意力直接放在了每种元素的化学势取值范围上。尤其对于多组元化合物，当元素种类为 N 时，其可行域的维度为 $N-1$ ，由于第二相对可行域的裁剪，使其形状变得及其复杂。此时我们没有精力去关注所有顶角的情况，而且希望直接地知道某种关心元素的化学势范围。我们的程序正是为此开发了 `edp chempot` 命令，来直接地获取不同元素的化学势取值范围。

手动处理大量竞争相是一个费力耗时的过程，因此我们提供了 `edp query` 命令来，能够从数据库直接获取所有竞争相结构文件。同时，我们还可以从数据库同时拉取竞争相的形成焓，方便我们检查自己的计算结果。另外一方面，在第一性原理的计算框架下，体系的能量值是依赖于赝势和计算程序的，但是物质的形成焓具有较好的稳定性。当我们对精度的要求不高时，或者进行初步试探时，我们完全可以利用数据库的竞争相形成焓来确定元素化学势的范围，加速我们的工作进程。

参见：

- `chempot` - 根据化合物和竞争相形成焓估计原子的化学势
- `query` - 从数据库获取竞争相结构和形成焓

警告： 由于数据库的高通量计算缘故，形成焓精度非常有限，因此只建议作为初步探索使用，我们无法对数据库获取到的数据可靠性作任何保证。此外，此功能的开发主要是为了方便大家交流学习，如有任何侵权行为，我们会立即关停此功能。

5 命令行使用参考

我们可以通过 `edp -h` 来查看所有支持的命令，一般命令的使用格式为：

```
$ edp [-v|-q] <command> --option1 --option2 [inputfile]
```

这里的 `-v` 选项可以增加屏幕的显示信息，而 `-q` 选项会尽量抑制屏幕的显示信息。我们可以通过子命令的 `-h` 选项来查看支持的操作，比如查看 `edp chempot` 命令支持的选项：

```
$ edp chempot -h
```

接下来我们将介绍支持的子命令 (以字母表顺序排序)：

boxhyd

产生仅包含单氢原子的同尺寸 POSCAR 文件。

cal

根据配置文件 (由 `-i/--input` 选项指定, 默认为 *EDOPING.in*) 计算缺陷形成能随费米能级的变化。

chempot

求解元素化学势的范围

这里我们需要准备一个输入文件 (默认文件名为 *EDOPING.cmpot*), 第一行需要以 ‘#’ 号开始, 然后依次是每种元素的名称, 以空格分隔。接下来是所有考虑的化合物的元素配比, 以及相应的能量值。这里, 第一个出现的化合物 (也就是文件的第二行) 会被程序认定为目标化合物, 也就是我们的基体相物质。

重要提醒: 在处理元素配比和能量时, 由于个人习惯以及不同数据库的格式规范差异, 我们需要非常小心这里的归一化相关的问题:

- 元素配比格式: (I) 晶胞中每种原子数目 (2) 最简原子数比 (3) 归一化比例
- 化合物的焓值表示: (A) 晶胞的总焓值 (B) 平均每个原子的焓值
- 焓值的参考: (I) 绝对焓值, 即计算程序中给出的焓值 (II) 形成焓, 即相对与对应单质的焓值差

在程序内部, 我们实际上是在处理类似下面的式子:

$$\frac{1}{C} \sum_i c_i \cdot \mu_i \leq \mu$$

这里, i 代指不同的化合物, c_i 是输入文件的元素配比, μ 是输入文件的化合物焓值; 如果使用了 `-n` (`--norm`) 选项, 则 $C = \sum_i c_i$, 否则 $C = 1$ 。简单来说, 为了得到正确的结果, 对于 (1+B) 和 (2+B) 情况需要指定 `-n` (`--norm`) 选项, 而对于 (1+A) 和 (3+B) 情况则需要避免该选项。由于缺少必要的信息, 我们无法处理 (2+A) 和 (3+A) 的情况, 需要使用者进行必要的数据处理。

至于焓值的参考问题, 基本原则就是: 最终求解化学势的参考就是初始给定化合物焓值的参考。如果提供的都是 (I) 绝对的焓值, 那么给出的就是绝对化学势; 如果提供的都是 (II) 形成焓, 那么给出的元素化学势和对应单质的差值。

由于 *EDOPING.in* 文件需要指定绝对化学势, 为了简化计算流程, 可以通过 `--refs` 选项指定每个单质元素的标准化学势, 从而最终获得相应的绝对化学势。在程序内部, 实际上就是将最后求解的化学势数值加上该参考值。

在 0.3 版本加入: `--refs` 选项。

diff

对比两个具有相同基矢 POSCAR 的原子增减情况, 可以用于检查点缺陷。以含有 Mn 间隙的 NbFeSb 超胞为例, 结果如下:

```
$ cd examples/NbFeSb_Interstitials
$ edp diff 1.perfect/POSCAR 2-1.defect-Mn_i/relax/POSCAR
  No.      f_a      f_b      f_c      previous      present
i 1        0.1250    0.1250    0.1250    Vac1          Mn1
```

这里，i 表示间隙型缺陷（v 表示空位，s 表示取代）。

energy

从 OUTCAR 文件读取最后一步的能量。

epsilon

从 OUTCAR 文件读取并打印各项介电常数。

ewald

从 OUTCAR 文件读取并打印马德龙常数。

fixchg

自动生成带电缺陷的计算文件夹，通过 `-i/--inputdir` 选项指定不带电结构自洽计算的文件夹（默认为 `charge_0`）。这里实际上会从 **POTCAR** 文件计算体系的净电子数，然后根据给定的体系电荷量自动计算体系的电子数（**NELECT** 参数）。因此，推荐在准备好不带电结构的计算文件夹后，运行该命令生成带电缺陷的计算文件夹，然后再进行批量提交。

groupby

利用径向分布函数对 **POSCAR** 中的原子进行分组，可以用于寻找不等价位置的复合缺陷。比如，我们想在含有 **Mn** 间隙缺陷的 **NbFeSb** 超胞中再引入一个 **Fe** 空位。在超胞中通常有很多个 **Fe** 原子，逐个计算每个位点的情况是非常耗时的。一个有效的简化策略就是，按照每个 **Fe** 原子的近邻环境对它们进行分组，对于同一个组内的 **Fe** 原子，它们理应具有相似的缺陷行为。如下所示，**POSCAR** 是一个包含 **Mn** 间隙原子的 **NbFeSb** 超胞：

```
$ cd examples/NbFeSb_Interstitials/2-1.defect-Mn_i/relax/
$ edp groupby -f POSCAR Fe
Group #1: Fe1, Fe2, Fe9, Fe11, Fe17, Fe21
Group #2: Fe3, Fe4, Fe5, Fe6, Fe10, Fe12, Fe13, Fe15, Fe18, Fe19, Fe22, Fe23
Group #3: Fe7, Fe8, Fe14, Fe16, Fe20, Fe24
Group #4: Fe25, Fe26, Fe27, Fe28, Fe29, Fe30, Fe31, Fe32

=====
No. |      Group #1      |      Group #2      |      Group #3      |      Group #4
---+-----+-----+-----+-----+
0 | (0.0, 'Fe', 1) | (0.0, 'Fe', 1) | (0.0, 'Fe', 1) | (0.0, 'Fe', 1)
1 | (2.6, 'Nb', 4) | (2.6, 'Nb', 4) | (2.6, 'Nb', 4) | (2.6, 'Nb', 4)
2 | (2.6, 'Sb', 4) | (2.6, 'Sb', 4) | (2.6, 'Sb', 4) | (2.6, 'Sb', 4)
3 | (3.0, 'Mn', 1) | (4.2, 'Fe', 12) | (4.2, 'Fe', 12) | (4.2, 'Fe', 12)
4 | (4.2, 'Fe', 12) | (4.9, 'Nb', 12) | (4.9, 'Nb', 12) | (4.9, 'Nb', 12)
5 | (4.9, 'Nb', 12) | (4.9, 'Sb', 12) | (4.9, 'Sb', 12) | (4.9, 'Sb', 12)
6 | (4.9, 'Sb', 12) | (6.0, 'Fe', 6) | (6.0, 'Fe', 6) | (5.2, 'Mn', 1)
7 | (6.0, 'Fe', 6) | (6.5, 'Nb', 12) | (6.5, 'Nb', 12) | (6.0, 'Fe', 6)
8 | (6.5, 'Nb', 12) | (6.5, 'Sb', 12) | (6.5, 'Sb', 12) | (6.5, 'Nb', 12)
9 | (6.5, 'Sb', 12) | (6.7, 'Mn', 2) | (7.3, 'Fe', 24) | (6.5, 'Sb', 12)
10 | (7.3, 'Fe', 24) | (7.3, 'Fe', 24) | (7.7, 'Nb', 16) | (7.3, 'Fe', 24)
11 | (7.7, 'Nb', 16) | (7.7, 'Nb', 16) | (7.7, 'Sb', 16) | (7.7, 'Nb', 16)
12 | (7.7, 'Sb', 16) | (7.7, 'Sb', 16) | (8.4, 'Fe', 12) | (7.7, 'Sb', 16)
13 | (8.4, 'Fe', 12) | (8.4, 'Fe', 12) | (8.8, 'Nb', 24) | (8.4, 'Fe', 12)
14 | (8.8, 'Nb', 24) | (8.8, 'Nb', 24) | (8.8, 'Sb', 24) | (8.8, 'Nb', 24)
15 | (8.8, 'Sb', 24) | (8.8, 'Sb', 24) | (8.9, 'Mn', 4) | (8.8, 'Sb', 24)
16 | (8.9, 'Mn', 1) | (9.4, 'Fe', 24) | (9.4, 'Fe', 24) | (9.4, 'Fe', 24)
17 | (9.4, 'Fe', 24) | (9.8, 'Nb', 12) | (9.8, 'Nb', 12) | (9.8, 'Nb', 12)
```

(续下页)

(接上页)

```
18 | (9.8, 'Nb', 12) | (9.8, 'Sb', 12) | (9.8, 'Sb', 12) | (9.8, 'Sb', 12)
19 | (9.8, 'Sb', 12) | (10.3, 'Fe', 8) | (10.3, 'Fe', 8) | (9.9, 'Mn', 3)
20 | (10.3, 'Fe', 8) | (10.6, 'Nb', 24) | (10.6, 'Nb', 24) | (10.3, 'Fe', 8)
21 | (10.6, 'Nb', 24) | (10.6, 'Sb', 24) | (10.6, 'Sb', 24) | (10.6, 'Nb', 24)
22 | (10.6, 'Sb', 24) | (10.7, 'Mn', 2) | (11.1, 'Fe', 48) | (10.6, 'Sb', 24)
23 | (11.1, 'Fe', 48) | (11.1, 'Fe', 48) | (11.4, 'Nb', 36) | (11.1, 'Fe', 48)
24 | (11.4, 'Nb', 36) | (11.4, 'Nb', 36) | (11.4, 'Sb', 36) | (11.4, 'Nb', 36)
25 | (11.4, 'Sb', 36) | (11.4, 'Sb', 36) | (11.9, 'Fe', 6) | (11.4, 'Sb', 36)
26 | (11.9, 'Fe', 6) | (11.9, 'Fe', 6) | (12.2, 'Nb', 12) | (11.9, 'Fe', 6)
27 | (12.2, 'Nb', 12) | (12.2, 'Nb', 12) | (12.2, 'Sb', 12) | (12.2, 'Nb', 12)
28 | (12.2, 'Sb', 12) | (12.2, 'Sb', 12) | (12.3, 'Mn', 4) | (12.2, 'Sb', 12)
29 | (12.3, 'Mn', 4) | (12.6, 'Fe', 36) | (12.6, 'Fe', 36) | (12.6, 'Fe', 36)
30 | (12.6, 'Fe', 36) | (12.9, 'Nb', 28) | (12.9, 'Nb', 28) | (12.9, 'Nb', 28)
=====
```

可以看到，32 个 Fe 原子可以被分成 4 组。元组内的三项分别是距离、原子类别和数量。比如，Fe1, Fe2, Fe9, Fe11, Fe17, Fe21 都属于组 #1，它们在 12.6 Angstrom 范围内具有着完全相同近邻原子。具体地，它们与最近的一个 Mn 原子距离 3.0 Angstrom，与第二近邻的 Mn 原子距离 8.9 Angstrom。为了更加清晰地聚焦于 Mn 原子，可以使用 `--grep Mn` 选项只保留含有 Mn 原子的行：

```
$ edp groupby -f POSCAR Fe --grep Mn
Group #1: Fe1, Fe2, Fe9, Fe11, Fe17, Fe21
Group #2: Fe3, Fe4, Fe5, Fe6, Fe10, Fe12, Fe13, Fe15, Fe18, Fe19, Fe22, Fe23
Group #3: Fe7, Fe8, Fe14, Fe16, Fe20, Fe24
Group #4: Fe25, Fe26, Fe27, Fe28, Fe29, Fe30, Fe31, Fe32

=====
No. |      Group #1      |      Group #2      |      Group #3      |      Group #4
-----+-----+-----+-----+-----
3 | (3.0, 'Mn', 1) | (4.2, 'Fe', 12) | (4.2, 'Fe', 12) | (4.2, 'Fe', 12)
6 | (4.9, 'Sb', 12) | (6.0, 'Fe', 6) | (6.0, 'Fe', 6) | (5.2, 'Mn', 1)
9 | (6.5, 'Sb', 12) | (6.7, 'Mn', 2) | (7.3, 'Fe', 24) | (6.5, 'Sb', 12)
15 | (8.8, 'Sb', 24) | (8.8, 'Sb', 24) | (8.9, 'Mn', 4) | (8.8, 'Sb', 24)
16 | (8.9, 'Mn', 1) | (9.4, 'Fe', 24) | (9.4, 'Fe', 24) | (9.4, 'Fe', 24)
19 | (9.8, 'Sb', 12) | (10.3, 'Fe', 8) | (10.3, 'Fe', 8) | (9.9, 'Mn', 3)
22 | (10.6, 'Sb', 24) | (10.7, 'Mn', 2) | (11.1, 'Fe', 48) | (10.6, 'Sb', 24)
28 | (12.2, 'Sb', 12) | (12.2, 'Sb', 12) | (12.3, 'Mn', 4) | (12.2, 'Sb', 12)
29 | (12.3, 'Mn', 4) | (12.6, 'Fe', 36) | (12.6, 'Fe', 36) | (12.6, 'Fe', 36)
=====
```

query

从公开材料数据库获取竞争相信息。通过 `--backend` 选项指定数据库的名称：

- OQMD (默认值): The Open Quantum Materials Database (<https://www.oqmd.org>)
- MP : The Materials Project (<https://next-gen.materialsproject.org>)

通过该命令我们可以得到用于化学势估计的输入数据文件 *EDOPING.cmpot*，其中给定的最简原子比和化合物平均每个原子的形成焓。因此，为了得到正确的化学势 $\Delta\mu_i$ ，在使用 *edp chempot* 进行计算时需要添加 `-n (--norm)` 选项。

提示： 使用 Materials Project 数据库前必须正确配置 MP_API_KEY 环境变量（在 [Materials Project - API 页面](#) 注册并获取 API key），将下面命令中的 `<your_api_key>` 替换为你的 API key：

```
$ export MP_API_KEY=<your_api_key>
```

对于私有化平台，可以将其添加到 `~/.bashrc` 文件中避免反复配置。

小技巧：使用时确保网络畅通，且受制于数据库的访问频率限制，不建议在短时间内反复多次使用。通常情况下，可以先到数据库官网进行查询，具有更好的可视化结果，然后再通过该命令进行数据获取。

在 0.4 版本加入：支持 Materials Project 后端数据库。

refine

对晶体结构进行调整，目前支持的操作如下：

- `-t/--transform`: 根据给定的变换矩阵进行扩胞。变换矩阵通常是一个 3x3 的整数矩阵，输入时 9 个组元以空格分隔排列成一行，比如 `-t "2 0 0 0 2 0 0 0 2"`。该选项也支持 3 个元素和 1 个元素的格式，分别表示对角矩阵和缩放的单位矩阵。下面的三条命令是等价的：

```
$ edp refine -t "2 0 0 0 2 0 0 0 2"
$ edp refine -t "2 2 2"
$ edp refine -t 2
```

此外，如果不提供值，软件会尝试从当前目录下的 `TRANSMAT.in` 文件读取变换矩阵（兼容 `vaspkit` 软件格式）：

```
$ edp refine -t
```

重要：eDoping 软件内部使用行向量矩阵的表示晶胞基矢，这与 VASP、Quantum Espresso、vaspkit 等软件保持一致，但是与 VESTA、phonopy 等软件采用的列向量形式不同。它们的转变矩阵相差一个转置操作。

- `-c/--cubicize`: 在给定的原子数附近进行立方体扩胞。比如：

```
$ edp refine -c 100
```

这表示寻找原子数靠近 100 且最接近立方体的超胞。

- `-d/--displace`: 移动晶胞中的原子。通常需要提供四个参数，依次表示需要移动的原子索引，沿着笛卡尔坐标系 x/y/z 方向的位移量（以埃为单位，注意：**不是分数坐标**）。比如：

```
$ edp refine -d Sb3 0 0 0.1
```

这表示将第 3 个 Sb 原子沿着 z 方向移动 0.1 埃。

在 0.4 版本加入：支持 refine 命令。

replace

替换 POSCAR 文件中的原子。通常需要两个位置变量以此表示旧的原子和新的原子。特别地，当新原子指定为 `Vac` 时表示删除原子，构造空位缺陷；当旧原子指定为 `Vac` 时表示仅插入原子，构造间隙缺陷。构造间隙缺陷时，必须通过 `-p/--position` 选项指定间隙原子的位置。

例子 1：将 POSCAR 中的 Fe2 替换为 Co，并将新结构保存为 POSCAR-sub：

```
$ edp replace -o POSCAR-sub Fe2 Co
```

例子 2：在 POSCAR 中构建 Fe2 位置的空位缺陷，并将新结构保存为 POSCAR-vac：

```
$ edp replace -o POSCAR-vac Fe2 Vac
```

例子 3: 在 POSCAR 中 (0.125, 0.125, 0.125) 处插入 Co 间隙原子, 并将新结构保存为 POSCAR-int :

```
$ edp replace -o POSCAR-int Vac Co -p 0.125 0.125 0.125
```

可以通过 `edp diff` 检查 POSCAR 文件的变化:

```
$ edp diff POSCAR POSCAR-int
```

在 0.3 版本加入: 支持通过 Vac 字符构造空位缺陷和间隙缺陷。

6 输入/输出文件

6.1 EDOPING.in

这是 `edp cal` 命令的输入文件, 用于指定点实施缺陷计算的一些配置。关键字推荐使用大写字母 (大小写不敏感特性还在实验中)。# 号开头的行是注释行, 会被程序忽略。行内的 # 号之后的内容也会被忽略。

DPERFECT

完美基体超胞自洽计算的目录路径。

DDEFECT

缺陷超胞自洽计算的顶级目录路径, 其中包含不同带电情况自洽计算的子目录。

CMPOT

增减原子的化学势 μ_i ($= \mu_i^\ominus + \Delta\mu_i$), 以空格分隔, 是一个包含偶数个值的序列, 交替表示移除的加入的原子的化学势。比如对于 Nb 被 Ta 取代 (即移除 Nb 原子, 加入 Ta 原子), 则设置为:

```
CMPOT = mu_Nb mu-Ta
```

特别地, 对于间隙和空位, 可以认为同时地移除或者加入一个化学势为 0 的原子, 比如对于 Nb 空位缺陷:

```
CMPOT = mu_Nb 0
```

对于 Nb 间隙缺陷:

```
CMPOT = 0 mu_Nb
```

VALENCE

缺陷原子的电荷值, 以空格分隔。注意, 电子本身带负电, 这意味着在 VASP 的 INCAR 文件中, 增加 NELECT 值对应更负的电荷值。

DDNAME

每个电荷值 (*VALENCE*) 自洽计算的子目录名称, 默认值为 `auto`, 通过组合 *PREFIX* 和 *VALENCE* 自动生成, 这里价态会保留前置的 “+/-” 号。比如, 如果 *VALENCE* = `-1 0 1`, 且 *PREFIX* = `charge_`, 那么当 *DDNAME* = `auto` 时, 等价于 *DDNAME* = `charge_-1 charge_0 charge_+1`。或者, 可以直接指定子目录名称, 用空格分隔 (目前不允许在子目录名称中含有空格)。

DREFER

指定含有缺陷的超胞所在的目录, 主要用于和完美晶胞对比确认缺陷位置。默认值为 `auto`, 程序自动从 0 价缺陷目录中读取缺陷超胞结构。如果提供, 则程序会在 *DDNAME* / *DREFER* 目录下寻找缺陷超胞的 POSCAR 文件。

在 0.3 版本加入: *DREFER* 参数。

PREFIX

不同电荷值的自洽计算子目录的前缀，默认值为 `charge_`。

EVBM

价带顶能量。默认值为 `inf`，程序自动从 `DDEFECT` 目录下的 `EIGENVAL` 文件读取。

ECBM

导带底能量。默认值为 `inf`，程序自动从 `DDEFECT` 目录下的 `EIGENVAL` 文件读取。

PENERGY

完美晶胞的总能量。默认值为 `inf`，程序自动从 `DPERFECT` 目录下的 `OUTCAR` 文件读取。

PVOLUME

完美晶胞的体积。默认值为 `inf`，程序自动从 `DPERFECT` 目录下的 `OUTCAR` 文件读取。

EWALD

马德龙常数。默认值为 0，表示禁用镜像电荷修正项。

EPSILON

介电常数。默认值为 `inf`，表示禁用镜像电荷修正项。

ICCOEF

镜像电荷修正项可以改写为 $E_{IC} = C_{IC} \cdot q^2$ ，可以直接通过 `ICCOEF` 指定系数 C_{IC} 。默认值为 `inf`，由 `EPSILON` 和 `EWALD` 自动计算，即：

$$C_{IC} = \left[1 - \frac{1}{3} \left(1 - \frac{1}{\epsilon} \right) \right] \frac{E_{wald}}{2\epsilon}$$

PADIFF

电势对齐修正项可以改写为 $E_{PA} = q \cdot \Delta V$ ，可以直接通过 `PADIFF`（和 `VALENCE` 长度相等的列表）指定电势差 ΔV 。默认值为 `[inf, ...]`，自动读取 `OUTCAR` 文件中距离缺陷最远位置处的电势差值。

BFTYPE

能带填充修正机制的类型。默认值为 0，表示禁用能带填充修正项。1 表示仅修正导带，-1 表示仅修正价带，2 表示同时修正导带和价带。

EMIN

费米能级的下边界（以 `EVBM` 为基准），默认值为 -1。

EMAX

费米能级的上边界（以 `EVBM` 为基准），默认值为 2。

NPTS

费米能级采样点数，默认值为 1001。

6.2 EDOPING.log

`edp cal` 命令的运行日志，和屏幕输出一样。

6.3 EDOPING.dat

`edp cal` 命令的计算结果文件，包含不同电荷值缺陷的形成能。它包含 $Nq + 3$ 列，以空格分隔，第一列是费米能级（以 *EVBM* 为基准），第二列是缺陷形成能，第三列是对应的电荷值，后面各列依次是不同电荷值的缺陷形成能。第一行是注释行，包含列名称，最后两个值分别表示晶胞体积和简并因子。一个例子如下：

```
# Ef, Eformation, q, q_-3, q_-2, q_-1, q_+0, q_+1, q_+2, q_+3; 1689.3500 1
-1.0000 -1.6816 3.0000 5.8115 4.3378 2.8985 1.4866 0.0987 -0.8285 -1.6816
-0.9990 -1.6786 3.0000 5.8085 4.3358 2.8975 1.4866 0.0997 -0.8265 -1.6786
-0.9980 -1.6756 3.0000 5.8055 4.3338 2.8965 1.4866 0.1007 -0.8245 -1.6756
-0.9970 -1.6726 3.0000 5.8025 4.3318 2.8955 1.4866 0.1017 -0.8225 -1.6726
-0.9960 -1.6696 3.0000 5.7995 4.3298 2.8945 1.4866 0.1027 -0.8205 -1.6696
-0.9950 -1.6666 3.0000 5.7965 4.3278 2.8935 1.4866 0.1037 -0.8185 -1.6666
-0.9940 -1.6636 3.0000 5.7935 4.3258 2.8925 1.4866 0.1047 -0.8165 -1.6636
-0.9930 -1.6606 3.0000 5.7905 4.3238 2.8915 1.4866 0.1057 -0.8145 -1.6606
-0.9920 -1.6576 3.0000 5.7875 4.3218 2.8905 1.4866 0.1067 -0.8125 -1.6576
...
```

6.4 EDOPING.cmpot

`edp chempot` 命令的输入文件，用于指定化合物的配比和形成能。以下面的 Mn 掺杂 NbFeSb 体系为例解释文件格式：第一行以 # 号开始，包含体系的元素名称。第二行是目标化合物 NbFeSb 的配比及形成能，然后是所有考虑的竞争相的元素配比及形成能。关于元素配比和形成能归一化的问题，详见 `edp chempot` 命令。

```
# Nb Fe Sb Mn
1 1 1 0 -0.350468735
0 1 0 0 0.0
0 0 1 0 -1.94641666666871563e-05
0 1 2 0 -0.036502481666666733
3 0 1 0 -0.28675903625
1 0 2 0 -0.279502903333333
1 0 0 2 -0.1441571941954
...
```

7 文章引用

如果此软件以及文档给您的工作提供了帮助，请引用我们的文章，这对我们很重要，非常感谢！

[1] J. Zhu, J. Li, Z. Ti, L. Wang, Y. Shen, L. Wei, X. Liu, X. Chen, P. Liu, J. Sui, Y. Zhang, eDoping: A high-throughput software package for evaluating point defect doping limits in semiconductor and insulator materials, *Materials Today Physics*, 55 (2025) 101754, <https://doi.org/10.1016/j.mtphys.2025.101754>.

[2] J. Li, J. Zhu, Z. Ti, W. Zhai, L. Wei, C. Zhang, P. Liu, Y. Zhang, Synergistic defect engineering for improving n-type NbFeSb thermoelectric performance through high-throughput computations, *Journal of Materials Chemistry A*, 10 (46) (2022) 24598-24610, <https://doi.org/10.1039/d2ta07142h>.

索引

B

BFTYPE
 命令行选项, 15
boxhyd
 命令行选项, 10

C

cal
 命令行选项, 10
chempot
 命令行选项, 10
CMPOT
 命令行选项, 14

D

DDEFECT
 命令行选项, 14
DDNAME
 命令行选项, 14
diff
 命令行选项, 10
DPERFECT
 命令行选项, 14
DREFER
 命令行选项, 14

E

ECBM
 命令行选项, 15
EMAX
 命令行选项, 15
EMIN
 命令行选项, 15
energy
 命令行选项, 11
EPSILON
 命令行选项, 15
epsilon
 命令行选项, 11
EVBM
 命令行选项, 15
EWALD
 命令行选项, 15
ewald
 命令行选项, 11

F

fixchg
 命令行选项, 11

G

groupby
 命令行选项, 11

I

ICCOEF
 命令行选项, 15

N

NPTS
 命令行选项, 15

P

PADIFF
 命令行选项, 15
PENERGY
 命令行选项, 15
PREFIX
 命令行选项, 14
PVOLUME
 命令行选项, 15

Q

query
 命令行选项, 12

R

refine
 命令行选项, 13
replace
 命令行选项, 13

V

VALENCE
 命令行选项, 14
命令行选项
 BFTYPE, 15
 boxhyd, 10
 cal, 10
 chempot, 10
 CMPOT, 14
 DDEFECT, 14
 DDNAME, 14
 diff, 10
 DPERFECT, 14
 DREFER, 14
 ECBM, 15
 EMAX, 15
 EMIN, 15

energy, 11
EPSILON, 15
epsilon, 11
EVBM, 15
EWALD, 15
ewald, 11
fixchg, 11
groupby, 11
ICCOEF, 15
NPTS, 15
PADIFF, 15
PENERGY, 15
PREFIX, 14
PVOLUME, 15
query, 12
refine, 13
replace, 13
VALENCE, 14