「极化晶体电荷 PCC 计算程序」用户手册

Contents

[1. 简介 2](#_Toc75444265)

[1.1. 程序说明 2](#_Toc75444266)

[1.2. 理论背景 2](#_Toc75444267)

[1.3. 程序结构 2](#_Toc75444268)

[2. 用法 3](#_Toc75444269)

[2.1. 准备结构 3](#_Toc75444270)

[2.2. 修改程序参数 3](#_Toc75444271)

[2.3. 记录对称性操作的文件格式 4](#_Toc75444272)

[2.4. 安装其它程序 4](#_Toc75444273)

[3. 程序结构 5](#_Toc75444274)

[3.1. 主程序 PCC.py 5](#_Toc75444275)

[3.2. 流程控制脚本 batch.sh 5](#_Toc75444276)

[4. 操作流程 5](#_Toc75444277)

[5. 程序结构 7](#_Toc75444278)

# 简介

## 程序说明

本程序名为“极化晶体电荷 PCC 计算程序”，英文名 ePCC，是一个用于计算极化晶体电荷 PCC 的通用程序。编程语言为 Python 和 Shell。

开发人员是上海大学硕士生范文斌（fanwenbin@shu.edu.cn），导师为上海大学理学院量子与分子结构国际中心（ICQMS）李永乐副教授（[yongleli@shu.edu.cn](mailto:yongleli@shu.edu.cn)）。

本程序无图形化界面，运行于 Linux 中。通常将其布置在超算中，以便高效计算。要求 Python 版本大于 3，必须安装 numpy、matplotlib 库。

本手册更新于2021年6月24日。

## 理论背景

静电相互作用是分子动力学（molecular dynamics, MD）中势能的重要组成部分之一。目前，多数原子电荷都是用于描述真空中的分子。对于晶体中的原子，有 Bader、Hirshfeld 电荷等空间划分方式，这些原子电荷通常仅定性正确。

为了精确描述晶体中的原子电荷，李永乐副教授于 2018 年提出了极化晶体电荷（polarized crystal charge, PCC），基于极化特定蛋白电荷（polarized protein-specific charge, PPC）改进发展而来。PCC 首次应用在甘氨酸晶体中，结果发表在 J. Phys. Chem. Lett. 2019, 10, 6, 1319–1324。

PCC 方法的主要思想为，用点电荷模拟体系所处的真实电荷环境。使用密度泛函理论（density functional theory, DFT）计算真空中晶胞的 RESP 电荷。将体系扩胞，其它晶胞内的所有原子均用点电荷表示，赋予其中心晶胞的电荷。重新计算 RESP 电荷，将中心晶胞的电荷赋予其它点电荷。如此迭代计算 RESP 电荷，直到电荷收敛。

本手册所指的“计算”均为 DFT 计算。

## 程序结构

本程序包含如下文件，这些文件所在的目录称为根目录。

1. PCC.py　核心程序，用于查找对称性、扩胞等操作。
2. PCC\_para.py　参数文件，用户按需修改。
3. batch.sh　Shell 脚本，用于调用 Python 及第三方程序。

# 用法

本程序调用 Gaussian 与 Multiwfn 联合计算 PCC，将整个计算过程全部自动化处理。下面叙述程序使用过程。

## 准备结构

本程序目前仅支持 VASP 的结构格式与文件名，如 POSCAR 和 CONTCAR。该结构文件与程序文件放在根目录下。

对于非周期性的计算，分子必须保持完整。那些处在盒子边界上的分子会被边界截断，所以必须补全这些分子才能确保计算正确。下面叙述处理结构的过程。

VESTA 是日本开发的晶体可视化软件，同时具有强大的建模、格式转换功能。使用 VESTA 载入分子结构，推荐含有对称性的结构文件，如 .cif 格式。载入后，屏幕上显示整个晶胞及所有原子。由于对称性的原因，部分原子或分子会重复出现。为确保正确描述晶体电荷分布（保留足够多的重复单元），同时减小计算量（保留必要的重复单元），所以删除多余的重复分子，使得体系的原子数大于等于晶胞内原子数，同时确保分子完整、确保中心晶胞被包埋充分。导出为 P1 格式，选择导出屏幕上显示的所有原子，该格式为 VASP 的坐标，将其改名为 POSCAR 或 CONTCAR 即可。

## 修改程序参数

程序运行前，必须按照实际需求修改参数文件 PCC\_para.py，且将其与其它文件放置在同一目录。下面简述程序参数及其含义。

1. Nim　一维数组、整数、长度为 3，表示三个方向的镜像（image）个数。如 Nim=[3,3,3] 表示将会有 个镜像。
2. space\_group　整数，表示空间群编号，对应 ./space\_group/SO\_<space\_group> 文件。其格式在后文叙述。
3. DFT 计算参数。
   1. charge 和 sm，整数，对应系统的净电荷和自旋多重度。如果 POSCAR 里有多余的分子，注意整体净电荷可能不为 0。通常二者为 0、1。
   2. gau\_memory 和 gau\_nprocs，整数，指内存和核心数，会被写在 .gjf 文件头，如 %mem=<gau\_memory>GB 和 %nprocs=<gau\_nprocs>。
   3. gau\_method 和 gau\_basis，字符串，指理论方法和基组。方法写在关键词行，如gau\_method/genecp，基组写在文件末尾 -C -H -O -N 0\n<gau\_basis>，部分重原子（如 Mn、Br）默认使用 SDD 赝势和基组。
   4. gau\_maxcycle 和 gau\_conv，整数，用于自洽场（self-consistent field, SCF）迭代。前者指最大迭代次数，后者指收敛限，会被写在关键词行中，如 SCF(maxcycle=<gau\_maxcycle>, conver=<gau\_conv>)。通常 conver=8 对于大体系过于严格，建议大体系设置为 6，意思是当能量变化小于 a.u. 即为收敛。
   5. gau\_symm，布尔值，是否启用对称性。真，则无变化；否，则在关键词行写入 nosymm。
4. max\_diff\_crt，浮点数，指 PCC 的收敛限，两次电荷变化的最大值若小于此值即为收敛，停止计算。若 gau\_conv=6，建议此值设置为 1e-4，gau\_conv=8，则此值建议为 1e-5。
5. lcp2k，布尔值，已弃用。真，则输出 CP2K 格式的 RESP 约束文件。

## 记录对称性操作的文件格式

所有对称性操作均存放于 ./space\_group/ 目录中，文件名是 SO\_<空间群号>。该文件分三部分，

1. 第一行是注释，不会被读取，后面所有行均会被读取。
2. 第二行是对称操作的个数，不包含本身的对称操作，空一行。
3. 接下来 5 行一组，记录了对称矩阵（3行，）及平移向量（1行，），空一行。

下面举个例子，113 号空间群的最后一个对称操作为 ，其分数坐标下的变换可记作

那么这个对称操作应写成

0 -1 0 \n -1 0 0 \n 0 0 1 \n 0.5 0.5 0.0 \n\n

## 安装其它程序

本程序 ePCC 需调用其它程序进行 DFT 与 RESP 计算，必须安装好 Gaussian 与 Multiwfn 并配置良好。

将 Gaussian 解压，配置环境变量，创建 scratch 文件夹。版本必须大于等于 16A。

下载并解压 Multiwfn（http://sobereva.com/multiwfn），配置环境变量，修改配置文件中的核心数、formchk 路径。版本必须大于等于 3.7。

# 程序结构

## 主程序 PCC.py

本程序运行时需要一个参数，可以是 p 或任何非负整数，直接接在运行命令后面，如 python PCC.py <opt>。

所有选项均需要读取参数（read\_para）、读取结构文件（read\_vasp）。

选项 p 对应画图，绘制两张图，所有 PCC 演化过程（charge\_evolution.svg）和每一步之间的电荷变化对数图（charge\_evolution\_difference.svg）。

选项 0 表示初次计算，将坐标等信息写成 Gaussian 格式（.gjf），判断对称性并输出对称原子，输出使用 Multiwfn 计算 RESP 电荷的流程。

选项正整数 i 表示迭代计算，程序会创建目录 i，读取上一步的电荷、将体系超胞（中心晶胞意外的原子用点电荷表示），输出 RESP 流程、画图（等价于选项 p）。

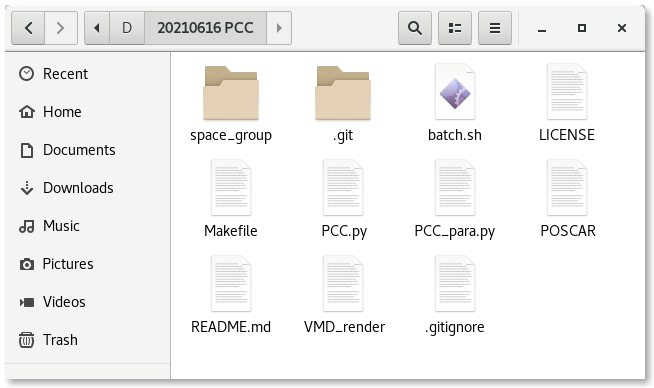
## 流程控制脚本 batch.sh

此文件使用 Shell 语言编写，仅兼容 bash 格式，用于控制整个计算流程。

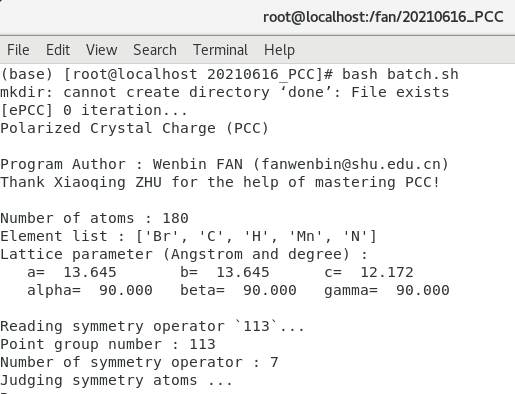
本脚本会从 0 开始迭代至 100，依次调用 PCC.py 生成输入文件等，同时调用 Gaussian 等程序计算并判断收敛情况。

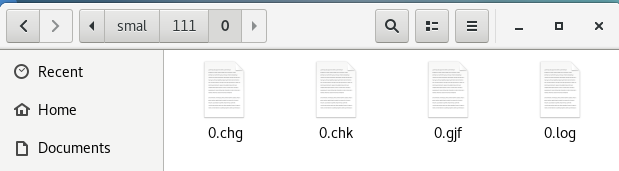
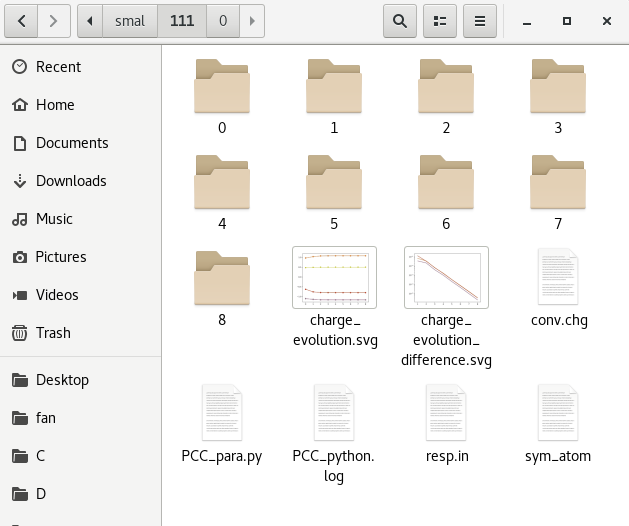
PCC 计算收敛后，所有文件将会被移至 done 文件夹。

# 操作流程

本程序无图形化界面，可在终端中直接运行。受限于本地资源，通常放在计算集群中运行，与本地运行方式一致。

准备结构文件 CONTCAR 或 POSCAR，准备对称操作 SO\_ 文件，准备主程序文件。准备好 Python 环境、Gaussian 和 Multiwfn 程序。修改 PCC\_para.py 参数文件。

打开终端，在终端中运行 bash batch.sh 即可使用本程序。屏幕上会输出当前迭代编号（shell 输出）、程序作者、体系信息（元素、晶格参数）、点群序号等。

初次计算（第 0 次）计算完成后，在文件夹 0 中会产生如下四个文件 chg、chk、gjf、log，分别是电荷、DFT 计算的检查点文件、Gaussian 输入文件、Gaussian 计算日志。每次迭代都会在对应的文件夹里产生同样的四个文件。.chg 是 RESP 原子电荷，记录了各个原子的电荷。

右图是经过了 8 次迭代后收敛的根目录，包括了 9 个文件夹、两幅图、收敛后的电荷（conv.chg）、RESP 参数、对称原子等。

# 程序结构

