

VaspCZ: 一个提高效率的 VASP 计算 辅助程序

张正德^{1,2} 谈蒙露^{1,2} 任翠兰¹ 怀平^{1,3,4}

1(中国科学院上海应用物理研究所 上海 201800)

2(中国科学院大学 北京 100049)

3(上海科技大学 物质科学与技术学院 上海 201210)

4(中国科学院上海高等研究院 上海光源科学中心 上海 201204)

摘要 VASP(Vienna Ab-initio Simulation Package)是基于密度泛函理论的通用材料计算模拟软件。随着高性能计算平台的广泛使用和高通量计算的兴起, VASP 计算任务的复杂性和数据量也大幅度提升。为了提高材料理论计算的方便程度,减少大量重复的指令和结果检查的复杂性,本文提供了一个基于 Python 语言的 VASP 计算辅助程序 VaspCZ,能够大幅提高研究者的科研效率。该程序包含软件部分和 API 部分,软件部分提供了命令行用户界面,基于基本的 Linux 命令即可一键完成 VASP 计算的相关操作,为无编程语言基础的研究人员提供了便利。应用程序接口部分服务于具有 Python 语言基础的研究者,提供了底层库,使得自定义计算(如高通量计算、编写高级应用等)更为简单快捷。本程序包可以显著提高 VASP 的计算效率,并已经通过 Github 开源项目公开,提供了详细的使用文档、使用示例和 API 接口,有望能为广大材料理论研究者提供便利。

关键词 第一性原理, VASP, 辅助程序, 电子结构计算, 过渡态计算

中图分类号 TL341

DOI: 10.11889/j.0253-3219.2020.hjs.43.030501

VaspCZ: an efficient VASP computation assistant program

ZHANG Zhengde^{1,2} TAN Menglu^{1,2} REN Cuilan¹ HUAI Ping^{1,3,4}

1(Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China)

2(University of Chinese Academy of Sciences, Beijing 100049, China)

3(School of Physical Science and Technology, ShanghaiTech University, Shanghai 201210, China)

4(Shanghai Synchrotron Radiation Facility, Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201204, China)

Abstract [Background] With the improvement of computing power and the rise of high-throughput computing, researchers usually need to perform a large number of repeated instructions to submit scientific tasks and further check their computational results during material theoretical calculations. Generally, these instructions are simple, cumbersome and time consuming. [Purpose] This study aims to improve the efficiency of scientific calculations by using VASP code. [Methods] The material theory calculation assistant program, VaspCZ, developed by using Python languages, is summarized to form a general github open source project, which provides detailed user documents,

中国科学院战略先导专项(No.XDA02040100)、国家自然科学基金(No.11605273)资助

第一作者: 张正德, 男, 1992年出生, 2011年毕业于北京航空航天大学, 现为博士研究生, 研究领域为材料科学

通信作者: 怀平, E-mail: huaiping@sinap.ac.cn

收稿日期: 2019-10-31, 修回日期: 2020-01-06

Supported by the Strategically Leading Program of Chinese Academy of Sciences (No.XDA02040100), National Natural Science Foundation of China (No.11605273)

First author: ZHANG Zhengde, male, born in 1992, graduated from Beihang University in 2011, doctoral student, focusing on materials science

Corresponding author: HUAI Ping, E-mail: huaiping@sinap.ac.cn

Received date: 2019-10-31, revised date: 2020-01-06

examples and application programming interfaces (API). **[Results]** VaspCZ, an efficient VASP computation assistant program includes two parts: the software part and API part. The software part provides the command line user interface, which enabled the researchers to complete the VASP calculations with the basic Linux command. The API part provides the bottom layer library to realize the custom calculation (such as high-throughput calculation, writing advanced application, etc.) for the researchers with python experiences. **[Conclusions]** It is proved by practice that the efficiency of theoretical calculation of materials using VASP can be significantly improved by aid of this assistant program.

Key words First-principles, VASP, assistant program, electronic structure calculation, transient state calculation

VASP (Vienna Ab-initio Simulation Package) 是 20 世纪 90 年代, 由 Vienna 大学 Hafner 小组开发的、进行材料电子结构计算和量子力学-分子动力学模拟的商用软件^[1-3]。VASP 基于密度泛函理论对 Kohn-Shan 方程进行迭代求解, 从头计算求解薛定谔方程, 在计算材料的结构参数、状态方程、电子性质、光学性质、磁学性质等各项实践中, 预测了很多先于实验的正确结果, 取得了很大的成功。目前, 该程序包已更新到 5.4.4 版, 在其基础上, 世界各地的科学家也开发了诸多补充算法和辅助程序, 以扩展其计算能力、提高计算精度或提升计算效率。

2000 年, Henkelman 等^[4] 发布了 VTST (Transition State Tools for VASP) 工具包, 提供了寻找鞍点和计算过渡态的源码和脚本。作为 VASP 的计算能力的补充, 其中的 CI-NEB 方法^[5-6] 和 Dimer 方法^[7], 已经成为目前计算体相扩散、晶界扩散、催化反应的重要工具之一。Mathew 等^[8] 于 2014 年在平面波 DFT 编码 VASP 中实现了隐式溶剂化模型, 描述了溶质和溶剂间的静电、空化和色散效应。他们发布的 VASP 插件 VASPsol 提供了一种有效的计算溶剂化对分子、晶体表面和反应势垒的影响的方法^[9]。PyLab 在 github 中开源了一个处理 VASP 文件的 python 库 VASPy^[10], 提供了绘制电荷密度、等高线、差分电荷密度、原子振动分析等功能。2018 年, Stolaroff 等^[11] 发布了开源程序 PyDEF2.0, 可在 GUI 用户界面自动执行 VASP 计算, 并提供了较为强大的态密度和能带图绘制工具。2018 年, Kundu 等^[12] 发布了 PASTA (Python Algorithms for Searching Transition states), 提供了搜索过渡态的 Python 算法。Rutter 等^[13] 发布了 C2x 工具, 提供了针对 CASTEP (Cambridge Sequential Total Energy Package) 输入的便捷准备功能和电子结构可视化功能。还有很多辅助程序可能流通于各研究小组内部。

当前, 随着计算能力的提升和高通量计算的兴起, 使用命令行方式操作 VASP 进行材料的理论研究越来越繁琐, 研究者通常需要输入大量重复的指

令提交计算任务、检查计算结果。为了提高科研效率, 我们开发了 VaspCZ 辅助程序。该程序开发了 Linux 用户界面以满足在超算平台快速提交任务和便捷检查的需求; 此外, 开发了 Python 库 VaspCZ.zzzlib, 以满足更高层次的快速编写上层应用的需求。本文将系统地阐述 VaspCZ 的框架设计、模块组成及其功能, 并给出了各部分的使用示例, 最后提供了获取程序的便捷途径。

1 程序总体设计

VaspCZ 程序基于 Linux 系统, 用 Python 语言编写而成, 该程序框架如图 1 所示。该对 Linux 系统无额外要求, Python 需使用 3.6 及以上版本, 支持库需安装 numpy 以实现高性能数值计算。程序分为两部分: 顶层是软件部分, 底层是 API 部分。

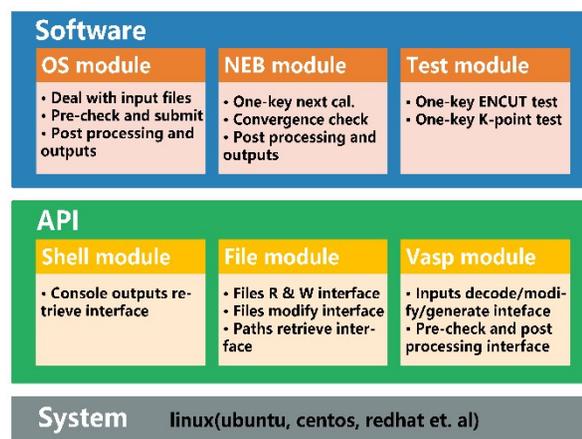


图 1 VaspCZ 程序框架图

Fig.1 Frameworks of VaspCZ

顶层的软件部分为了满足快速提交任务、便捷检查的需求, 针对结构优化、静态计算、过渡态计算和测试计算设计了三个模块: OS (Optimization and Static) 模块、NEB (Nudged Elastic Band) 模块和 Test 模块。

底层的 API 部分是顶层的 python 库, 分为 shell、File 和 Vasp 模块, 安装 VaspCZ 时库自动安装, 使用

时 import VaspCZ.zzzlib, 为编写顶层应用提供便捷的接口。

1.1 顶层:软件部分

软件部分提供了如图2所示的Linux命令行用户界面,终端输入快捷键vcz启动程序后,在主界面输入数字选择模块,而后在该模块界面选择功能即可。

软件部分分为OS模块、NEB模块和Test模块。

```

1  |-----|
2  |          |
3  |          |
4  |          |
5  |          |
6  |          |
7  |          |
8  |          |
9  |          |
10 |          |
11 | Input: 1 |
12 | OS module selected |
13 |-----|
14 |          |
15 |          |
16 |          |
17 |          |
18 |          |
19 |          |
20 |          |
21 |          |
22 |          |
23 |          |
24 |          |
25 |-----|
26 | Input:   |

```

VaspCZ interface

(1) Opt and Sta module
(2) NEB module
(3) Test module
(0) Exit

by: Zhengde Zhang (drivener@163.com)

Optimization and Static calculation

(1) Generate inputs (example)
(2) Generate INCAR for Sta
(3) Generate POTCAR
(4) Generate KPOINTS
(5) Generate Vasp.sh
(6) Vasp Keep Inputs
(7) Vasp Pre-check and Qsub
(8) Check Results
(0) Back

图2 VaspCZ软件主界面和OS模块界面

Fig.2 The main and OS module user interface of VaspCZ.

1.1.1 OS模块

OS模块提供了快捷进行结构优化(Optimization)计算和静态(Static)计算以及检查结果的功能,各功能如表1所示。

表1 OS模块功能
Table 1 List of OS module

| 标签 Label | 功能 Function |
|-------------|--|
| 1.1 | 产生Vasp输入文件(示例) Generate inputs (example) |
| 1.2 | 生成静态计算INCAR Generate INCAR for Static calculation |
| 1.3 | 产生赝势文件POTCAR Generate POTCAR |
| 1.4 | 产生网格文件KPOINTS Generate grid file KPOINTS |
| 1.5 | 产生提交任务脚本Vasp.sh Generate script Vasp.sh for submitting jobs |
| 1.6 | 仅保留VASP输入文件 Only keep inputs of VASP |
| 1.7 | 前检查并提交任务 Pre-check and submit jobs |
| 1.8 | 后处理并打印计算结果 Post process and check results |

其中,标签使用模块名.功能名的格式代表了具体的功能,如1.1代表第1个模块(OS模块)的第1个功能,调用此功能只需在终端依次输入:vcz-1-1。

功能1.1,产生示例的Vasp输入文件,会在当前目录下产生5个文件(Vasp计算所需文件):INCAR、POSCAR、POTCAR、KPOINTS和计算平台PBS任务系统提交计算任务的脚本Vasp.sh。

功能1.2,将当前路径的结构优化INCAR一键修改为静态计算的INCAR。

功能1.3,自动产生匹配POSCAR中元素的Vasp计算所需赝势文件POTCAR,或指定元素和赝势类型产生POTCAR。

功能1.4,输入网格和方法(或使用默认值),一键产生Vasp计算所需网格文件KPOINTS。

功能1.5,输入节点数、核数和任务名(或使用默认值),一键产生PBS系统提交任务的脚本Vasp.sh。

功能1.6,删除当前目录其他所有文件和文件夹,仅保留Vasp的5个输入文件(INCAR、POSCAR、POTCAR、KPOINTS和Vasp.sh),用于计算出现问题时重算。选择该功能后可输入文件名添加需要额外保留的文件。

功能1.7,准备好输入文件后,一键进行前检查,检查INCAR、POSCAR和POTCAR是否匹配,检查通过后将打印检查信息,并提交任务。

功能1.8,一键检查当前目录及所有子目录下的结构优化和静态计算的结果,如OUTCAR或者log中有错误(ERROR)或警告(WARNING)将提示所在位置。检查内容包括:当前路径、能量、离子步数、磁矩、POSCAR和CONTCAR原子之间的距离、原子最大受力。

1.1.2 NEB模块

NEB模块提供了便捷地使用NEB(Nudged Elastic Band)方法计算过渡态和检查结果的功能。本程序假设计算遵循如图3所示的过渡态计算的一般流程:先进行结构优化,而后静态计算,最后过渡态计算,如需再振动分析。

用NEB方法计算过渡态需要知道过渡态的初态和末态,如催化反应中原子吸附脱附前后的结构、扩散研究中原子扩散前后的结构等。为保证过渡态计算的准确性,初态和末态需要分别在文件夹[WORKDIR]/ini/Opt和[WORKDIR]/fin/Opt文件夹内进行结构优化,将原子弛豫到比较稳定的结构。确定初末态的稳定结构后,分别在文件夹[WORKDIR]/ini和[WORKDIR]/fin文件夹内进行静态计算,以获得更加准确的能量。而后将初态的稳定结构作为初始点,末态的稳定结构作为末态点,

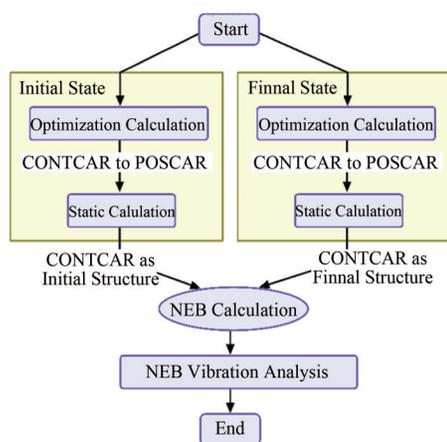


图3 过渡态计算的一般流程图

Fig.3 General flow charts of NEB calculation

使用 vtst 工具插入中间态,用 NEB 方法在当前工作目录[WORKDIR]下进行过渡态搜索计算。最后如果需要,在当前文件夹通过振动分析计算出该过渡过程原子的有效跃迁频率。遵循此过渡态计算的一般过程而设计的 NEB 模块各功能如表 2 所示。

表 2 NEB 模块功能
Table 2 List of NEB module

| 标签 Label | 功能 Function |
|----------|--|
| 2.1 | 一键结构优化到静态计算 One-key from optimization to static calculation |
| 2.2 | 一键静态计算到过渡态计算 One-key from static cal. to NEB calculation |
| 2.3 | 过渡态振动分析 Transition state vibration analysis |
| 2.4 | 仅保留结构优化输入文件 Only keep inputs of optimization |
| 2.5 | 仅保留过渡态输入数据 Only keep inputs of static calculation |
| 2.6 | 检查过渡态受力情况 Check the stress of transition state |
| 2.7 | 检查过渡态各态原子距离 Check the atomic distance of transition state |
| 2.8 | 检查过渡态计算结果 Check the results of transition state |
| 2.9 | 检查过渡态振动分析结果 Check the results of vibration analysis |

功能 2.1,在初末态的结构优化完成后,一键检查结果、拷贝 CONTICAR 为 POSICAR、设置新的 INCAR、前检查、提交静态计算任务。

功能 2.2,在初末态的静态计算完成后,一键检查结果、根据初末态的稳定结构的原子距离差之和/0.8 自动插入中间态、设置新的 INCAR、提交过渡态计算任务。

功能 2.3,在过渡态计算完成后,自动读取初态、末态、鞍点态的结构,确定迁移的原子,修改 POSICAR 文件,设置新的 INCAR,提交振动分析任务,计算迁移原子在初态、末态、鞍点态时的振动频率。

功能 2.4,在过渡态计算出现错误或其他原因需要重新计算时,回滚到第一步初末态结构优化阶段。删除当前目录下的所有文件和文件夹,仅保留 ini/Opt/下和 fin/Opt 下的 5 个输入文件 (INCAR, POSICAR, POTICAR, KPOINTS 和 Vasp.sh),而后可使用功能 2.1 重新提交任务。

功能 2.5,在过渡态计算出现错误或其他原因需要重新计算时,回滚到 NEB 计算阶段。删除目录下的文件和文件夹,仅保留 ini/和 fin/文件夹下所有内容,而后可使用功能 2.2 重新提交任务。

功能 2.6,过渡态计算中或计算结束不收敛时,检查受力情况。一键打印各个离子步时各个中间态的原子最大受力及他们之和。根据原子最大受力和最小时对应的步数,快捷查找过渡态搜索中较为合理的中间结构,进行进一步分析和计算。

功能 2.7,过渡态计算中或计算结束不收敛或计算完成后,检查各中间态原子距离。快捷查看过渡态搜索中是否有某个中间态弛豫到不可预测的结构导致计算不收敛。

功能 2.8,在过渡态计算中或完成后,检查当前目录及所有子目录下的 NEB 计算结果(忽略静态计算和结构优化),如 OUTICAR 或者 log 中有错误 (ERROR)或警告 (WARNING)将提示所在位置,检查完成后输出结果。输出结果包含过渡态计算路径、各中间态的编号、体系总能、势垒等信息。

功能 2.9,在过渡态振动分析完成后,检查当前目录及所有子目录下的原子振动频率(即尝试频率)并计算有效频率。

1.1.3 Test 模块

Test 模块提供了一键截断能测试和 K 点测试功能。通常,一个体系在大规模进行计算和分析之前,需要进行截断能测试和 K 点测试确定合适的 ENCUT 设置和 KPOINTS 设置。本程序提供的 Test 模块各功能如表 3 所示。

表 3 Test 模块功能
Table 3 List of Test module

| 标签 Label | 功能 Function |
|----------|-------------------------------|
| 3.1 | 截断能测试 One-key ENCUT test |
| 3.2 | K 点测试 One-key K-point test |

做截断能测试的目的是选取一个合适的截断能,截断能决定了VASP计算过程中被作为赝势处理的电子波函数的范围。截断能太小,计算得到的体系总能不可信,截断能太大,计算中迭代需要花费大量资源。

功能3.1,提供了一键进行截断能测试得到合理的截断能测试的功能。做K点测试的目的是选取一个合适的KPOINTS设置,K点决定了VASP计算过程中倒空间的网格分隔点数,体系越大,合适的K点网格一般越小。

功能3.2,提供了一键进行K点测试得到合理的K点设置的功能。

1.2 底层:API部分

API部分为有python基础的研究者提供了本程序通用功能的接口。通过库便捷调用相关功能,以实现自定义计算和高通量计算等功能。库名:VaspCZ.zzzlib,包含:Shell模块、File模块和Vasp模块,该库在安装本程序时自动安装,使用本库在python交互界面或.py文件中导入库即可:

```
1 import VaspCZ.zzzlib as zzd
```

API部分的Shell模块提供了便捷获取控制台输出的接口,可使用本模块实现程序部分的功能。File模块提供了便捷的文件处理接口。Vasp该模块提供了编写高级Vasp计算辅助程序的底层接口。各模块详细的接口函数、传入参数、功能和返回值见API说明文档^[14]。

2 应用实例

将通过三个实例说明VaspCZ软件部分的使用和输出结果,并通过两个代码实例说明如何使用API部分减少代码量,实现高通量计算。

2.1 软件部分实例

实例在安装本程序时自动获得,安装默认目录为:[HOME]/bin/VaspCZ/examples。

例1:一键检查结构优化和静态计算结果

cd进入实例文件下的1.8文件夹,依次输入vcz-1-8,即可运行OS模块的功能1.8:后检查并打印结果。

输出结果如图4所示,自动识别当前目录及所有子目录是否存在结构优化或静态计算,存在则判断是否完成,并输出当前的离子步和电子步;自动检查是否有警告或错误,如有则打印警告或错误类型及内容;自动检查各路径下计算的体系总能、步数、磁性、距离和原子最大受力。

例2:一键检查过渡态计算和振动分析结果

| | | | | | | |
|-----|-----------------------------------|-----------|-----|--------|--------|--------|
| 01. | Path:./Te-Energies/Fe106Te1V1_6nn | Done | 10F | 12 | RMM | |
| 02. | Path:./Te-Energies/Fe108 | Done | 1F | 43 | RMM | |
| 03. | Path:./Te-Energies/Fe106Te1V1_2nn | Done | 12F | 12 | RMM | |
| 04. | Path:./Te-Energies/Fe107Te1 | Done | 8F | 7 | RMM | |
| 05. | Path:./Te-Energies/Fe106Te1V1 | Done | 15F | 9 | RMM | |
| 06. | Path:./Te-Energies/Fe106Te1V1_5nn | Done | 10F | 10 | RMM | |
| 07. | Path:./Te-Energies/Fe106Te1V1_1nn | Done | 15F | 9 | RMM | |
| 08. | Path:./Te-Energies/Fe107V1 | Done | 8F | 11 | RMM | |
| 09. | Path:./Te-Energies/Fe106Te1V1_4nn | Done | 12F | 12 | RMM | |
| 10. | Path:./Te-Energies/Fe106Te1V1_3nn | Done | 12F | 12 | RMM | |
| 11. | 无警告或错误 | | | | | |
| 12. | 路径 | 能量 | 步数 | mag | dist | RMS |
| 13. | .../Fe106Te1V1_6nn | -862.8140 | 10 | 0.0000 | 0.4859 | 0.0082 |
| 14. | ...-Energies/Fe108 | -879.4933 | 1 | 0.0000 | 0.0000 | 0.0000 |
| 15. | .../Fe106Te1V1_2nn | -862.7920 | 12 | 0.0000 | 0.4653 | 0.0080 |
| 16. | ...nergies/Fe107Te1 | -873.2808 | 8 | 0.0000 | 0.4551 | 0.0000 |
| 17. | ...gies/Fe106Te1V1 | -863.5515 | 15 | 0.0000 | 0.4876 | 0.0000 |
| 18. | .../Fe106Te1V1_5nn | -862.8245 | 10 | 0.0000 | 0.4708 | 0.0062 |
| 19. | .../Fe106Te1V1_1nn | -863.5515 | 15 | 0.0000 | 0.4876 | 0.0000 |
| 20. | ...nergies/Fe107V1 | -869.0110 | 8 | 0.0000 | 0.1263 | 0.0000 |
| 21. | .../Fe106Te1V1_4nn | -862.7821 | 12 | 0.0000 | 0.5201 | 0.0093 |
| 22. | .../Fe106Te1V1_3nn | -862.8325 | 12 | 0.0000 | 0.5003 | 0.0093 |

图4 功能1.8:后检查并打印输出示例

Fig.4 Function 1.8: post check and print example

同理,在examples/2.6-2.9文件夹下调用功能2.8可以一键检查过渡态计算结果,调用功能2.9可以一键检查振动分析结果。该文件夹下的实例是fcc-Fe中Fe原子向其第一近邻空位扩散的过渡态计算(该计算中未考虑自旋极化)。

功能2.8检查过渡态计算结果如图5所示,自动判断当前目录及所有子目录中是否存在过渡态计算,判断该计算是否完成并输出;自动检查是否在警告或错误;而后打印当前的过渡态计算结果。第一列IMAGE0是初态,IMAGE4是末态,IMAGE1、2、3是中间态,第二列显示出中间态原子的最大受力,第三列是各态的体系总能,第四列是以初态能量为原点的相对能量,IMAGE2的能量差最高,该中间态即为鞍点态(或称为过渡态),因此fcc-Fe中Fe向其第一近邻空位扩散的势垒为1.39 eV。

| | | | | |
|-----|---|----------|-------------|----------|
| 01. | NEBCheckBegins... --func=3 | | | |
| 02. | Path:./ | NEB计算完成! | | |
| 03. | log和OUTCAR中无警告 | | | |
| 04. | Path:[HOME]/bin/VaspCZ/examples/2.6-2.9./ | | | |
| 05. | IMAGE | RMS | Energy | Barrier |
| 06. | 0 | 0.000000 | -869.011000 | 0.000000 |
| 07. | 1 | 0.007191 | -868.512500 | 0.498500 |
| 08. | 2 | 0.004424 | -867.616700 | 1.394300 |
| 09. | 3 | 0.007191 | -868.512500 | 0.498500 |
| 10. | 4 | 0.000000 | -869.011000 | 0.000000 |
| 11. | NEBCheck done! | | | |

图5 功能2.8:检查过渡态输出结果

Fig.5 Function 2.8: check the NEB results

2.9功能检查过渡态振动分析结果如图6所示,[True, True, False]表示是否分析了初态、鞍点态和末态的原子的振动。由于fcc-Fe自扩散的初态和末态是相同的,因此不必分析末态原子的振动。输出表明:扩散原子在初态中三个自由度上的尝试频率(Attempt Frequency)分别为:6.59 THz、6.16 THz和4.99 THz,而扩散原子在鞍点态中三个自由度上的尝试频率分别为:6.92 THz、4.67 THz和5.82 THz,其中,第三个方向上由f/i标识表示出该方向的振动为

虚频, 是不稳定的。过渡态的有效频率(Effective Frequency)是所有原子在初态时的尝试频率之积与在过渡态时尝试频率之积的商, 简化为扩散原子在初态时的尝试频率之积与在过渡态时尝试频率之积的商(不计虚频)^[15], 计算得该过渡态的有效频率为 6.28 THz。

```

01. ----- 开始检查振动分析结果 -----
02. [HOME]/bin/VaspCZ/examples/2.6-2.9/vib_analysis [True, True, False]
03. [HOME]/bin/VaspCZ/examples/2.6-2.9/vib_analysis
04. 1 f = 6.592458 THz 41.421637 2PiTHz 219.900730 cm-1 27.264225 meV
05. 2 f = 6.167487 THz 38.751465 2PiTHz 205.725224 cm-1 25.506686 meV
06. 3 f = 4.988518 THz 31.343783 2PiTHz 166.399047 cm-1 20.630860 meV
07. 1 f = 6.915127 THz 43.449027 2PiTHz 230.663815 cm-1 28.598678 meV
08. 2 f = 4.670165 THz 29.343510 2PiTHz 155.779920 cm-1 19.314255 meV
09. 3 f/i= 5.816549 THz 36.546458 2PiTHz 194.019199 cm-1 24.055323 meV
10. [HOME]/bin/VaspCZ/examples/2.6-2.9/vib_analysis forward frequency: 6.2805
11. ----- 检查结束 -----

```

图 6 功能 2.9: 检查过渡态振动分析输出结果

Fig.6 Function 2.9: check the vibration results of transient states

例 3: 过渡态收敛性

能量收敛但力不收敛是过渡态计算中很可能出现的收敛问题, 不收敛有可能是初末态结构不合理所致, 也可能是在计算中结构弛豫到不可预测的结果所致。进入 examples 文件夹, 调用 2.6 功能可以对力不收敛的过渡态计算进行分析。

该示例的各中间态在各离子步下时的原子最大受力如图 5 所示。中间态共三个: 01、02 和 03。由第 2~4 列看出, 随着计算进行到第 8 离子步时, 各中间态的原子最大受力减小到小于设置的收敛标准 ($0.01 \text{ eV} \cdot \text{Å}^{-1}$) 以下, 该过渡态计算收敛成功。某些力不收敛的计算表现为在第 7 步将收敛的时候, 原子最大受力不降反升, 最终计算到 100 步都没有收敛, 此时较为合理的中间态结构不是 100 步时的结构, 而是第 7 步的结构。使用功能 2.6 根据最大受力和最小时所在的离子步判断哪一步的结构较为合理, 而后取出该合理的结构进行进一步分析和计算。

```

01. CheckNEBRMS runs...
02. Step 01-RMS 02-RMS 03-RMS 01+02+03
03. 1 1.037104 2.021187 1.037073 4.095364
04. 2 0.536016 0.547372 0.536028 1.619416
05. 3 0.161509 0.171973 0.161456 0.494938
06. 4 0.107206 0.100033 0.107196 0.314435
07. 5 0.066828 0.045721 0.066828 0.179377
08. 6 0.035165 0.021158 0.035164 0.091487
09. 7 0.014496 0.008725 0.014491 0.037712
10. 8 0.007191 0.004424 0.007191 0.018806
11. CheckNEBRMS Done!

```

图 7 功能 2.6: 检查未达到收敛标准的过渡态

Fig.7 Function 2.6: check the unconvergent NEB results

2.2 API 部分实例

例 4 说明了如何调用 VaspCZ.zzzlib 库降低代码量完成自定义计算, 例 5 说明了如何调用该库实现高通量计算。

例 4: 14 行代码实现一键截断能测试程序

进行截断能测试的原理是在 INCAR 中分别设置截断能分为 200 eV、250 eV、300 eV、... 700 eV, 进行结构优化计算。一般来说截断能越高, 体系总能越准确, 但资源消耗也越大。随着截断能增加, 体系能量相差小于 0.001 eV 每原子即可确定合适的截断能。程序设计思路为: 当前目录准备好 VASP 输入文件, 每个截断能创建一个文件夹, 拷贝输入文件到文件夹内, 修改 INCAR 中的截断能设置, 进行前检查并提交任务, 计算结束用软件部分 OS 模块功能 1.8 后检查确定合适的截断能。实现脚本为 Vasp_ENCUT_Test.py, 代码在 VaspCZ 安装目录/examples/API/example_1 下, 在命令行界面输入: python3 Vasp_ENCUT_Test.py 即可运行代码实现本功能。

本例中, 代码使用了 API 部分的 File 模块的 2.2 接口 substituteData() 函数和 Vasp 模块的 3.7 接口 check_and_qsub() 函数, 将近百行代码缩短到 14 行, 大大简化了开发难度。

例 5: 批量提交结构优化计算任务

本例中计算任务为: 对不同的溶质元素 (Al、As、Bi、Co、Cr、Ga、Ge、Ir、Mo、Nb、Ni、P、Pb、Rh、Ru、Sb、Si、Sn、Tc、Ti、V、W、Zn、Cu、Mn、Ag、Au、Cd、Hf、Hg、In、Os、Pd、Pt、Sc、Ta、Tl、Zr、Re) 进行高通量计算筛选, 需要获得不同元素作为溶质时在 bcc-Fe 中空位辅助下进行扩散的扩散势垒, 如 Wu 等^[15]所述。程序设计思路为: 通过 NEB 获得扩散势垒, 需要执行如图 3 的过渡态一般过程: 先结构优化, 而后静态计算, 最后过渡态计算。编写程序一键提交 39 种溶质元素初态和末态的结构优化共 78 个计算任务, 而后使用软件部分 NEB 模块功能 2.1 和 2.2 一键进行下一步静态计算和过渡态计算。计算完成后使用功能 2.8 后检查一键获得全部扩散势垒。实现脚本为 Vasp_Batch.p, 代码在 VaspCZ 安装目录/examples/API/example_2 下, 在命令行界面输入: python3 Vasp_Batch.py 即可运行代码实现本功能。

本例中, 使用了 VaspCZ.zzzlib 库实现高通量计算, 十分方便。其他的高级功能实现见 VaspCZ 安装目录/sourcecode 文件夹下。

3 程序可用性

本程序已在 github 免费开源, 开源许可^[16]。安装使用说明文档见网页^[17]。所需 python 及环境安装教程见网页^[18]。

4 结语

为了满足科研人员在使用VASP进行高通量计算时快速提交任务,批量检查结果的需求,我们设计了VaspCZ软件,该软件的设计在包含三个模块的同时,提供了LINUX环境下的用户界面,是使用VASP在超算平台进行计算的高效辅助程序。VaspCZ软件同时提供了API接口,为研究者提供了编写高级应用(如VASP高通量计算)更便捷的Python库。本程序使用简单、便捷,有望为使用VASP软件进行计算的广大计算材料研究者带来更好的体验。

参考文献

- Kresse G, Hafner J. Abinitio molecular-dynamics for liquid-metals[J]. *Physical Review B*, 1993, **47**(1): 558–561.
- Kresse G, Hafner J. Ab-initio molecular-dynamics simulation of the liquid-metal amorphous-semiconductor transition in germanium[J]. *Physical Review B*, 1994, **49**(20): 14251–14269. DOI: 10.1103/physrevb.49.14251.
- Kresse G, Furthmüller J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set [J]. *Physical Review B*, 1996, **54**(16): 11169–11186. DOI: 10.1103/physrevb.54.11169.
- Group J. Transition State Tools for VASP [EB/OL]. 2019-10-10. <http://theory.cm.utexas.edu/vtsttools/index.html>.
- Henkelman G, Jónsson H. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points[J]. *The Journal of Chemical Physics*, 2000, **113**(22): 9978 – 9985. DOI: 10.1063/1.1323224.
- Henkelman G, Uberuaga B P, Jónsson H. A climbing image nudged elastic band method for finding saddle points and minimum energy paths[J]. *The Journal of Chemical Physics*, 2000, **113**(22): 9901 – 9904. DOI: 10.1063/1.1329672.
- Henkelman G, Jónsson H. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives[J]. *The Journal of Chemical Physics*, 1999, **111**(15): 7010–7022. DOI: 10.1063/1.480097.
- Mathew K, Sundararaman R, Letchworth-Weaver K, *et al.* Implicit solvation model for density-functional study of nanocrystal surfaces and reaction pathways[J]. *The Journal of Chemical Physics*, 2014, **140**(8): 084106. DOI: 10.1063/1.4865107.
- Mathew K, Chaitanya Kolluru V S, *et al.* VASPsol: implicit solvation and electrolyte model for density-functional theory[EB/OL]. 2019-10-10. <https://github.com/henniggroup/VASPsol>.
- PyLab. VASPy manipulating VASP files with Python[EB/OL]. 2019-10-10. <https://github.com/PytLab/VASPy>.
- Stoliaroff A, Jobic S, Latouche C. PyDEF 2.0: an easy to use post-treatment software for publishable charts featuring a graphical user interface[J]. *Journal of Computational Chemistry*, 2018, **39**(26): 2251 – 2261. DOI: 10.1002/jcc.25543.
- Kundu S, Bhattacharjee S, Lee S C, *et al.* PASTA: python algorithms for searching transition states[J]. *Computer Physics Communications*, 2018, **233**: 261 – 268. DOI: 10.1016/j.cpc.2018.06.026.
- Rutter M J. C2x: a tool for visualisation and input preparation for Castep and other electronic structure codes [J]. *Computer Physics Communications*, 2018, **225**: 174–179. DOI: 10.1016/j.cpc.2017.12.008.
- Zhang Z D. The API documentation for VaspCZ [EB/OL]. 2019-10-10. https://github.com/zhangzhengde0225/VaspCZ/blob/master/docs/VaspCZ_python_API.md.
- Wu H, Mayeshiba T, Morgan D. High-throughput ab-initio dilute solute diffusion database[J]. *Scientific Data*, 2016, **3**: 160054. DOI: 10.1038/sdata.2016.54.
- Zhang Z D. The LICENSE for VaspCZ[EB/OL]. 2019-10-10. <https://github.com/zhangzhengde0225/VaspCZ/blob/master/LICENSE.md>.
- Zhang Z D. The README for VaspCZ[EB/OL]. 2019-10-10. <https://github.com/zhangzhengde0225/VaspCZ>.
- Zhang Z D. The install tutorial of python3[EB/OL]. 2019-10-10. https://github.com/zhangzhengde0225/VaspCZ/blob/master/docs/python3_install_tutorial.md.