
通用进化算法晶体结构预测

A.R. Oganov, C.W. Glass, A.O. Lyakhov, Q. Zhu, G.-R. Qian, H.T. Stokes,
P.V. Bushlanov, Z. Allahyari, P. Graf, V. Stevanovic, F. Therrien, S.V. Lepeshkin,
Z.-H. Wang

with contributions from

M. Davari, A.I. Samtsevich, E.V. Tikhonov, R. Agarwal, X. Dong, M.S. Rakin,
P. Pertierra, Z. Raza, M.A. Salvado, D. Dong, Q.-F. Zeng

用户手册

Version 10.5, 2021 年 7 月 7 日.

© A.R. Oganov, with sections by Q. Zhu, M.S. Rakin, G.-R. Qian and A. Samtsevich

<http://uspex-team.org>

目录

1 USPEX 的特点, 目标和历史	5
1.1 概述	5
1.2 USPEX 的特点	7
1.3 有关 USPEX 主要原理及应用的文章	10
1.4 版本历史	11
2 初始化	15
2.1 如何获得 USPEX	15
2.2 版权引用	15
2.3 错误报告	15
2.4 运行 USPEX 所需的计算机配置	15
2.5 USPEX 的外部接口代码	15
2.6 如何安装 USPEX	17
2.7 如何运行 USPEX	17
2.8 执行 USPEX 示例: 简明教程	18
2.8.1 测试 USPEX-Python 指令	19
2.8.2 运行示例 EX13-3D_special_quasirandom_structure_TiCoO ₂	19
2.8.3 借助外部代码运行示例	20
2.8.4 检查结果	21
3 输入与输出文件	22
3.1 输入文件	22
3.1.1 Specific 文件夹	22
3.2 输出文件	25
4 输入文件 INPUT.txt 的设置	30
4.1 计算类型	30
4.2 种群	39
4.3 适者生存和选择	40

4.4	结构产生和变异操作	41
4.5	约束条件	45
4.6	晶胞	47
4.7	重启作业	48
4.8	详述 从头算法	49
4.9	指纹设置	52
4.10	反种子设置	53
4.11	空间群的判定	55
4.12	给开发者的关键参数	55
4.13	不常使用的关键参数	57
5	特例的额外输入设置	60
5.1	变成分代码	60
5.2	热电导率优化	62
5.2.1	相关软件的安装	62
5.2.2	在 INPUT.txt 里设置热电导率计算	63
5.2.3	输出	65
5.3	分子晶体和聚合物	65
5.3.1	分子晶体, calculationType=310/311	65
5.3.2	聚合物晶体, calculationType=110 (“线性链状模型”)	66
5.3.3	经典力场的额外输入	66
5.3.4	如何准备 MOL 文件	68
5.4	表面	68
5.5	二维晶体	71
5.5.1	二维晶体, 计算类型 =-200/-201	71
5.5.2	定组分	71
5.5.3	变组分	71
5.5.4	二维晶体预测结果	72
5.6	团簇	73
5.7	进化准动力学命令	73

5.8	粒子群算法 (PSO) 指令	76
6	相变路径预测	78
6.1	Pmpaths: 预测固体-固体相变机制的几何方法	78
6.2	变胞微动弹性带 (VCNEB) 法	80
6.3	VCNEB 的 Input 选项	80
6.4	如何设置 VC-NEB 计算的初始路径	87
6.5	过渡态路径取样 (TPS) 方法	89
6.6	TPS 方法的 Input 设置	89
7	在线工具	94
7.1	结构表征	94
7.2	性质计算	94
7.3	分子晶体	94
7.4	表面	95
7.5	其它	95
8	常见问题	96
8.1	如何将结果可视化?	96
8.2	如何避免陷阱?	96
8.3	什么是单块计算?	96
8.4	如何基于已知片段预测结构	97
8.5	如何使用种子技术?	98
8.6	如何控制成分?	99
8.7	如何从本地机器向远程集群设置免密连接?	100
8.8	如何创建一个计算使用的工作提交脚本?	101
8.8.1	步骤 1: 配置 Submission/文件夹中的文件	102
9	附录	110
9.1	例子目录	110
9.2	测试运行	113

9.3	INPUT.txt 输入文件示例	114
9.3.1	定成分预测 (calculationType=300):	114
9.3.2	变成分预测 (calculationType=301):	115
9.3.3	进化赝动力学模拟 (calculationMethod=META):	116
9.3.4	VCNEB 模拟 (calculationMethod=VCNEB):	117
9.4	空间群列表	118
9.5	层群列表	119
9.6	平面群列表	120
9.7	点群列表	121
9.8	USPEX 中使用的共价原子半径表	122
9.9	USPEX 使用的默认化合价表	124
9.10	USPEX 使用的默认 goodBonds 表	125
	Bibliography	126

1 USPEX 的特点, 目标和历史

1.1 概述

USPEX(Universal Structure Predictor: Evolutionary Xtallography 的缩写) 是一种基于进化算法开发的晶体结构预测方法和程序。“uspekx”在俄语译为“成功”, 这象征着这种预测方法的高成功率和取得的丰硕的研究成果。在新材料理论发现领域, USPEX 发挥着重要作用。USPEX 的基本特点如下:

自 2004 年起, 开展 USPEX 项目的主要目的是发展非经验晶体结构预测方法并开发程序。除了这种非经验式的结构搜索外, USPEX 也能够预测大量亚稳结构, 亦能够基于先验知识进行模拟。2010 年, USPEX 开始大量涉及其它结构搜索问题, 并于 2012 年起在程序中包含了许多互补的方法。

晶体结构预测问题非常古老, 而且一直是理论晶体化学的重要核心问题。John Maddox¹ 在 1988 年写道:

“在物理学中, 一个令人遗憾的问题是我们知道了晶体的化学组分但却不能预测它们的晶体结构, 哪怕是最简单的晶体。预测固体(如冰)的晶体结构仍然是人力所不能及的”。

显然, 结构预测的首要问题是如何进行全局优化, 即在一系列结构中找到自由能(每摩尔)最小的那个。在这里, 我们首先讨论一下潜在结构的数量。我们考虑一种简化的情况, 一个体积为 V , 有 N 个相同原子的固定立方晶胞。为了进一步简化, 我们假设原子均离散分布于分辨率为 δ 的网格节点上。这样, 原子间的坐标组合数目 C 会变得有限化:

$$C = \frac{1}{(V/\delta^3)} \frac{(V/\delta^3)!}{[(V/\delta^3) - N]!N!} \quad (1)$$

如果将 δ 设为典型键长的一个分数值(例如, $\delta=1 \text{ \AA}$), 由方程 1 给出的组合将会是对结构自由能局部极小值数量的一种合理估计。如果原子类型不同, 则不同结构的数量将显著增加。假设一个典型的原子体积 $\sim 10 \text{ \AA}^3$, 并考虑 Stirling 公式 ($n! \approx \sqrt{2\pi n}(n/e)^n$), 那么在一个有 10 原子的单胞内单质 A (化合物 AB) 体系的可能结构数为 10^{11} (10^{14}); 若单胞内有 20 个或 30 个原子, 则可能结构数分别为 10^{25} (10^{30}) 和 10^{39} (10^{47})。我们可以看到, 这些数目十分巨大, 甚至是总原子数 $N \sim 10$ 的小体系实际处理起来也不可能实现。更糟的是, 随着 N 的增加系统复杂性会呈指数增加。很明显, 除了最简单的含有 $\sim 1-5$ 个原子的单胞体系以外, 在自由能表面上逐点搜索所有结构是不可行的。

USPEX^{2;3} 采用进化算法进行结构搜索, 该结构搜索算法由 A.R. Oganov 和 C.W. Glass 主导设计, A.O. Lyakhov, Q. Zhu, G.-R. Qian, P.V. Bushlanov, Z. Allahyari, S.V. Lepeshkin 和 A.I. Samtsevich 等不断发展。它的高效性在于精心设计的结构变异操作方式, 其可靠性取决于在进化算法中运用了目前最先进的从头算模拟。进化算法的最大优势是不需要任何体系特定知识(除了化学组分之外)并且可以进行自我完善 - 即, 在后代种群中, 越来越好的结构会被发现并用于产生新结构。这样可以帮助我们快速定位到能量(性质)面上的潜在区域(图 1)。而且, 基于变异操作方式的灵活特性, 人们也很容易把一些附加的特点融入到进化算法中。

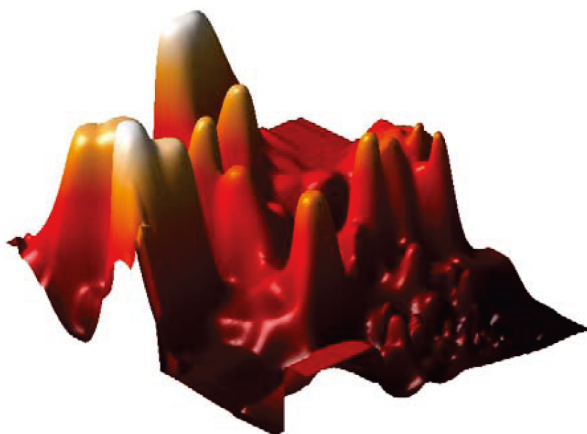


图 1: Au_8Pd_4 简化面的 2D 投影图, 显示了某一具有大量低能量结构的聚集区。此图来源于 Oganov and Valle (2009)。

我们开发 USPEX 的主要动力之一是 2004 年^{4,5} 发现了 MgSiO_3 的后钙钛矿结构 (图 2), 这一发现显著改变了地球内部结构模型。在 2005 年的中期, 我们开发出了 USPEX 的第一个内部工作版本。2010 年 9 月, 当 USPEX 公开发布时, 用户数近 200。近些年, USPEX 用户数量增长速度很快 - 在 2012 年 5 月我们拥有 800 用户, 在 2014 年 12 月用户数超过 2100, 在 2018 年 12 月用户数超过 4500。

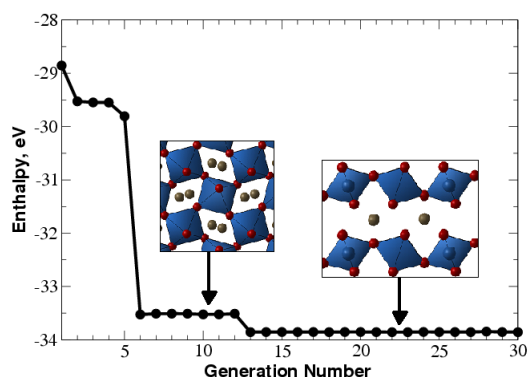


图 2: 120 GPa 下的 MgSiO_3 晶体结构预测 (20 原子/晶胞)。上图展示了最优结构焓值随着代数增加的变化。在第六代和第十二代间最优结构为钙钛矿, 但在第十三代全局最优结构 (后钙钛矿) 被发现。这次模拟计算是在 2005 年我们利用 USPEX 的第一个版本并结合从头算完成的。在不使用任何实验数据的背景下, 它证明了 USPEX 可以在一次模拟中同时找到稳定结构和低能量的亚稳结构。每一代都包含 30 个结构。我们利用 USPEX 的第一个版本进行了约 10 次结构预测, 该图显示的是最慢的一次结果, 即使这样速度也相当快。

USPEX 的极大推广应归功于它的高效性和可靠性。在第一次无机晶体结构预测盲测 (The First Blind Test of Inorganic Crystal Structure Prediction⁷) 中, USPEX 的效率和可靠性都超过了其他方法 (如模拟退火和随机抽样等)。随机抽样 (由 Freeman 和 Schmidt 于 1993 年和 1996 年率先用于晶体结构预测, 并于 2006 年被 Pickard⁶ 以 AIRSS 的名字重新使用) 是最简单的, 也是成功率最低和计算成本最高的结构预测方法。即使在小体系中, 如有 8 个原子的 GaAs 单胞内, USPEX 相对于随机抽样方法的优势也是巨大的 (随机抽样需要平均针对 500 个结构进行弛豫来发现基态, 而 USPEX 仅仅需要对约 30 个结构进行弛豫 (图 3))。由于结构搜索的复杂性与体系大小具有指数相

关性, 因此 USPEX 的优势也会随着体系的增大而变得巨大。例如: 由随机抽样预测得到的 SnH_4 的稳定结构⁶ 中三分之二都被证明不稳定⁷; 同样, 使用随机抽样⁸ 对 N^9 和 SnH_4 的预测也被证实不正确 (对 USPEX¹⁰ 和随机抽样¹¹ 的预测结构进行比较)。

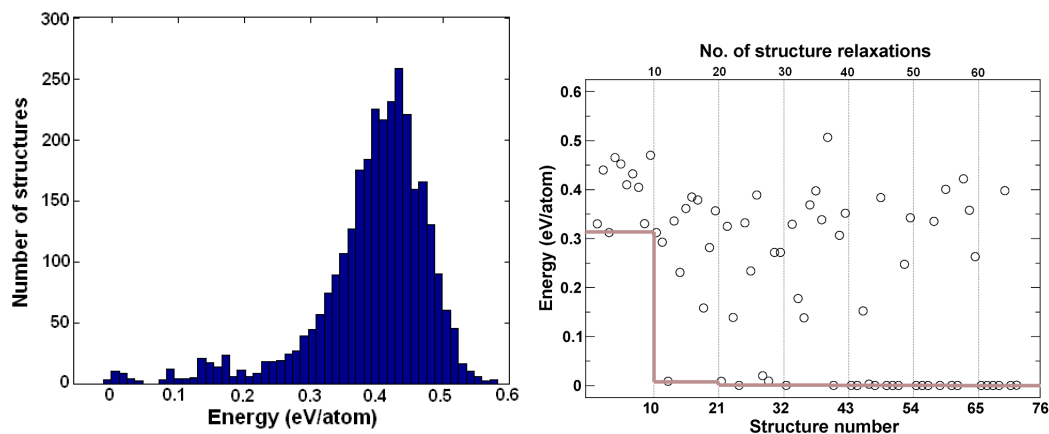


图 3: GaAs 结构预测。a) 随机结构的能量分布图, b) 一个进化模拟进程图 (细垂直线表示结构代数, 而灰线则表现随代数变化结构最低能量的变化趋势。所有的能量都是基态结构能量的相对值, 进化模拟中每代会有 10 个结构。此外, 上一代具有最低能量的结构会延续到下一代)。

对于原子数较多的体系, 随机抽样倾向于产生几乎具有相同能量的完全无序结构, 这样结构预测的成功率几乎降低到为零。从预测 MgSiO_3 后钙钛矿结构 (超胞中有 40 个原子) 的例子可以看出: 在进行 120,000 次结构弛豫后, 随机抽样仍然没有找到正确的结构, 而 USPEX 只需进行几百次结构弛豫 (Fig. 4)。

USPEX 可以很容易地基于随机抽样预测结构, 但大多数情况下仅仅只是为了对体系进行测试。同样, 基于粒子群优化 (PSO) 算法开发的团簇及晶体结构预测方法 (由 A.I. Boldyrev 开发, Wang、Lu、Zhu 和 Ma 重新改写) 也可以在 USPEX 实现。在 USPEX 原有代码的基础上, 我们进行了少量编程改进了 PSO (croPSO) 算法用于晶体结构预测; 结构预测测试结果表明该 croPSO 方法优于之前所有的 PSO 方法。但实际上, 所有基于 PSO 算法开发的晶体结构预测程序均远不如 USPEX 高效可靠。如果有人想尝试, 我们认为 PSO 方法更适用于测试。结合 Martoňák 赝动力学和 Oganov-Glass 进化算法开发的进化赝动力学方法¹² 是一种非常强大的新方法, 它是对进化算法的补充。进化赝动力学方法可以获得全局最优结构和低能量亚稳结构, 甚至可用于寻找可能的相转变路径。为了更详细地研究相变机制, 其他方法: 变胞 NEB 法¹³ 和转变路径方法¹⁴ 也已嵌入到当前 USPEX 版本中¹⁵。

1.2 USPEX 的特点

- 仅给定化学组分, 预测稳定和亚稳结构。可同时预测稳定组分及结构。
- 结合已知信息进行结构预测:
 - 根据实验已知晶胞参数, 或固定晶胞形状, 又或固定晶胞体积进行约束性结构搜索 (详

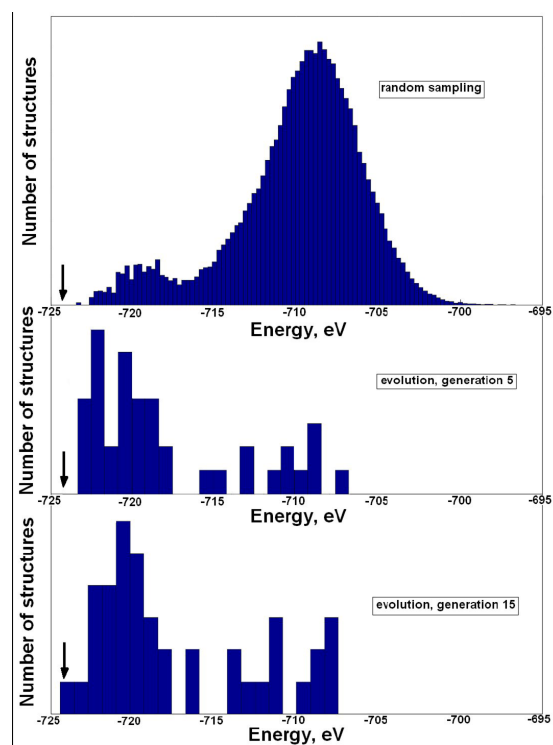


图 4: 能量面取样: 基于 MgSiO_3 后钙钛矿结构 (40 原子/晶胞) 晶格常数, 分别采用随机抽样和 USPEX 方法进行结构搜索。局域优化结构的能量如图所示。随机抽样共生成 1.2×10^5 个结构 (没有一种结构与基态结构一致)。USPEX 的搜索, 每一代包括 40 个结构, 在 15 代内发现基态结构。基态结构的能量为箭头标示处。这张图显示, 包含在进化搜索中的“学习”驱动发现低能量结构。

见 4.6);

– 基于已知或假想结构进行结构搜索 (详见 8.5);

– 依据预设分子进行结构搜索 (详见 5.3)。

- 有效的束缚技术, 可以消除搜索空间中的非物理的和多余的区域。晶胞精简技术 (Oganov & Glass, 2008)。
- 指纹函数技术。(Oganov & Valle, 2009; Lyakhov, Oganov, Valle, 2010)。详见 4.9。
- 通过完全随机的方式, 或使用空间群和晶胞分裂技术产生初始结构 (Lyakhov, Oganov, Valle, 2010)。基于强大的拓扑结构技术产生初始结构 (Bushlanov, Blatov, Oganov, 2018)。
- 结果在线分析: 空间群的确定 (输出 CIF 文件) (详见 4.11), 计算硬度、有序参量等。
- 预测纳米颗粒和表面重构结构。详见 5.4。
- 续算功能。可以从任何终止的时刻开启续算 (详见 4.7)。
- 基于 STM4 代码 (由 M. Valle 开发) 对 USPEX 结果文件进行可视化和分析, 该代码与 USPEX 完全对接 (详见 8.1)。
- USPEX 与 VASP、SIESTA、GULP、LAMMPS、DMACRYS、CP2K、Abinit、CRYSTAL、Quantum Espresso、FHI-aims、ATK、CASTEP、Tinker、MOPAC 等代码都有接口, 详见 ??。
- 可以从本地的工作站向远程集群和超算提交作业。详见 8.8。
- 可以选择使用 USPEX 算法 (默认算法), 随机取样法, 修正粒子群优化法 (5.8), 进化赝动力学法 (5.7) 以及类最小跳跃法来进行结构预测。可以使用进化赝动力学法, 变胞 NEB 法 (6.2) 和 TPS 法 (6.5) 预测相变路径。USPEX 同时提供一种快速的几何映射方法预测最优可能的相变机制。
- 除了能量外, USPEX 还可以选择以其他物理性质作为优化函数 - 例如, 硬度 (Lyakhov & Oganov, 2011)、密度 (Zhu 等, 2011)、带隙和介电常数 (Zeng 等, 2014) 以及许多其它性质。
- 为了方便开发者和用户, USPEX 是一个基于 MATLAB 的代码同时它也能在 Octave (一个类似于 MATLAB 环境的免费软件) ——用户不需要进行任何编译, 为了加强 MATLAB 版本的兼容性, 只使用了基本的 MATLAB 命令。USPEX 代码已经在 MATLAB2012 版至 2015 版和 Octave 3.4 (新的 Octave 版本将不被支持) 进行了开发和测试。
- 从 9.4.1 版本开始, USPEX 开发了基于 Python 指令的作业运行模块 (USPEX-Python), 提供了许多有用的命令行选项。

1.3 有关 USPEX 主要原理及应用的文 章

1. Oganov A.R., Glass C.W. (2006). Crystal structure prediction using evolutionary algorithms: principles and applications. *J. Chem. Phys.*, **124**, 244704.
2. Oganov A.R., Stokes H., Valle M. (2011). How evolutionary crystal structure prediction works — and why. *Acc. Chem. Res.*, **44**, 227–237.
3. Lyakhov A.O., Oganov A.R., Stokes H., Zhu Q. (2013). New developments in evolutionary structure prediction algorithm USPEX. *Comp. Phys. Comm.*, **184**, 1172–1182.
4. Zhu Q., Oganov A.R., Glass C.W., Stokes H. (2012). Constrained evolutionary algorithm for structure prediction of molecular crystals: methodology and applications. *Acta Cryst. B*, **68**, 215–226.
5. Zhu Q., Li L., Oganov A.R., Allen P.B. (2013). Evolutionary method for predicting surface reconstructions with variable stoichiometry. *Phys. Rev. B*, **87**, 195317.
6. Zhu, Q., Sharma V., Oganov A.R., Ramprasad R. (2014). Predicting polymeric crystal structures by evolutionary algorithms. *J. Chem. Phys.*, **141**, 154102.
7. Lyakhov A.O., Oganov A.R., Valle M. (2010). Crystal structure prediction using evolutionary approach. In: Modern methods of crystal structure prediction (ed: A.R. Oganov), Berlin: Wiley-VCH.
8. Oganov A.R., Ma Y., Lyakhov A.O., Valle M., Gatti C. (2010). Evolutionary crystal structure prediction as a method for the discovery of minerals and materials. *Rev. Mineral. Geochem.*, **71**, 271–298.
9. Duan, D., Liu, Y., Tian, F., Li, D., Huang, X., Zhao, Z., Yu, H., Liu, B., Tian, W., Cui, T. (2014). Pressure-induced metallization of dense $(\text{H}_2\text{S})_2\text{H}_2$ with high- T_c superconductivity. *Sci. Rep.*, **4**.
10. Bushlanov, P. V., Blatov, V. A., Oganov, A. R. (2019). Topology-based crystal structure generator. *Computer Physics Communications*, **236**, 1-7.
11. Lepeshkin S.V., Baturin V.S., Uspenskii Yu.A., Oganov A.R. (2019). Method for simultaneous prediction of atomic structure of nanoclusters in a wide area of compositions. *J. Phys. Chem. Lett.*, **10**, 102-106.
12. Lepeshkin S.V., Baturin V.S., Uspenskii Yu.A., Oganov A.R. (2019). Method for simultaneous prediction of atomic structure of nanoclusters in a wide area of compositions. *J. Phys. Chem. Lett.*, **10**, 102-106.
13. Allahyari, Z., Oganov, A. R. (2018). Multi-objective optimization as a tool for material design. *Handbook of Materials Modeling*, 1-15.

1.4 版本历史

- v.1 — 无局域优化进化算法。实空间描述, 与 VASP 接口。试验版 (2004.10)。
- v.2 — 嵌入 CMA-ES (CMA-ES 是一种强大的全局优化方法, 由 N. Hansen 设计)。试验版 (2005.1)。
- v.3 — 有局域优化进化算法。
- v.3.1 — 工作版本, 序列化。重大改进。
- 3.1.4-3.1.5 — 第一个产品版本。主要依赖于遗传操作。2005.5。
- 3.1.8 — 自动调整 k 点网格。15/10/2005。
- 3.1.11 — 从任意一代开始重新计算。测试版。04 /11/2005。
- 3.1.12 — 产品版本, 基于 v.3.1.11, 可变化的片段移动变异。11/11/2005。
- 3.1.13 — 自动缩放体积。29/11/2005。
- 3.1.14 — 种子技术。29/11/2005 (6/12/2005 调试)。
- v.3.2 — 大规模并行版本。
- v.4 — 统一并行/串行版本。
- 4.1.1 — 晶格变异。20/12/2005 (10/01/2006 调试)。
- 4.2.1 — 与 SIESTA 接口。初始群的数量可以不同于后代种群。24/01/2006 (20/04/2006 调试)。
- 4.2.3 — 可选择是否优化最优结构。完全调试过的大规模并行版本。25/04/2006。
- 4.4.1 — 与 GULP 接口。08/05/2006
- v.5 — 经过重写和调试过的版本, 代码实现模块化。.
- 5.1.1 — 原子类型突变, 代码的互通性, 从 INPUT_EA.txt 读取输入参数。20/12/2006。
- 5.2.1 — SIESTA 接口用于 Z 矩阵, 旋转变异操作。01/03/2007。
- v.6 — 产品版本。
- 6.1.3 — 可以高效地完成大体系的硬约束。07/06/2007。
- 6.2 — 开发者版本。
- 6.3.1-6.3.2 — 引入晶胞对角线角度约束。重写作业远程提交代码。改进输入文件格式。进一步标准测试。07/12/2007。
- 6.3.3 — X-com 网格界面 (S. Tikhonov 和 S. Sobolev 共同参与)。05/03/2008。
- 6.4.1 — 指纹函数功能。07/04/2008。
- 6.4.4 — 空间群识别。快速指纹 (参考表)。05/05/2008。
- 6.5.1 — 针对大体系的分裂晶胞法。更容易的远程提交作业。可变数量的最佳结构 (能量群)。16/07/2008。
- 6.6.1 — 非常稳定的版本, 改进了指纹和晶胞分裂技术。13/08/2008。
- 6.6.3 — 实现基于多种亲代的遗传。01/10/2008。
- 6.6.4 — 增加亲代参与遗传的阈值。03/10/2008。
- 6.6.6 — 第一次实现多元指纹功能。04/12/2008。

6.6.7, 6.7.1 和 6.7.2 — 利用准熵测量种群的多样性。10/12/2008。

v.7 — 产品版本, 含有变化学成分功能。

7.1.1-7.1.7 — 序列化改进版本。版本 7.1.7 已提供给约 200 位用户。变化学组分已部分编码, 修复了许多常见错误, 改进了一些针对能量面的技巧。改进了晶胞分裂技术, 实现了赝亚晶格方法。实现了多元指纹函数 (较一元指纹函数更加敏感)。28/04/2009 (最终版 28/05/2009)。

7.2.5 — 第一个实现变化学组分的版本。引入了变换操作以及组分熵。06/09/2009。

7.2.7 — 充分调试, 改进续算功能, 改进种子技术, 在结构弛豫中引入扰动。25/09/2009, 在 7.2.8/9 版本中进一步改进。

7.3.0 — 指纹函数引入变化学成分代码。引入一种“公平”算法用于产生第一代组分。22/10/2009。

7.4.1 — 引入基于局部有序坐标突变¹⁶, 遗传和变换操作也是基于局部有序度。引入硬度计算程序并实现以硬度和密度作为优化函数。04/01/2010。

7.4.2 — 实现基于局部有序函数的多亲代遗传。15/01/2010。

7.4.3 — 嵌入一种新的优化函数 (最大化结构有序度和种群多样性)。引入反种子技术和消除性参数 `volTimeConst`, `volBestHowMany`。24/01/2010。

v.8 — 产品版本, 写入新优化函数。

8.1.1-8.2.8 — 开发版本。局部有序与坐标变异算子, 软模变异运算, 硬度的计算与优化, 优化介质的敏感性。预测纳米粒子与表面结构, 实现点群, 软件整体表现提升。可选用 PSO 模拟, (由于 PSO 的低效率, 不建议在实际应用中使用 PSO 模拟——所以只作测试用途)。参数 `goodBonds` 转变为矩阵用以建立纳米粒子。22/09/2010。

8.3.1 — 调试 PSO 算法, 优化介电常数, 清除输入。08/10/2010。

8.3.2 — 为集群引入连通性检查 (非常有用), 改进 `dynamicalBestHM=2` 选项, 以及用于产生一代纯软模变异结构。改进集群的指纹功能。与 Quantum Espresso 和 CP2K 接口。11/10/2010。

8.4 — 开发版本集合版, 包含改进的反种子功能和纳米团簇计算上做了一些改进, 建立表面重构、伪准动态和分子晶体分支程序。

8.5.0 — 第一代随机结构的初始化是使用 H. Stokes 添加的空间群代码。准动力学新的表达方式添加到一个单独的代码中。改正了 `varcomp`, `antiseeds`, `nanoparticles` 和计算硬度代码中的一些错误。18/03/2011。

8.5.1 — 工作界面的大量调试。对定晶胞, 变成分和亚晶胞实现空间群初始化。20/04/2011。

8.6.0 — H. Stokes 的工作为软件添加空间群判定程序。为分子晶体合并更新后代码 (包括空间群初始化)。修正了一个 SIESTA 中错误 (感谢 D. Skachkov)。06/05/2011。

8.6.1-8.7.2 — 开发版本, 相当稳健。改进了定晶胞内的对称初始化。优化介电常数 (使用 GULP? 和 VASP), 带隙 (使用 VASP) 和费米能级的 DOS 图 (VASP)。启用图形输出。改进的软模变异运算 (使用更好的标准模式与定向简并) 和遗传运算 (对实验物质使用能量规则相关性系数和余弦公式)。现在大多数变量有默认值, 它减少用户输入量。减短并改进格式日志。13/11/2011。

8.7.5 — 修正变成分的代码中错误, 图形输出许多额外数据, 如用于可变成分计算的原子近似体积。增加将结构提取为凸包图的功能, 以便进行后期处理。21/03/2012。

v.9 — 产品版本, 增加客户体验, 加入新类型的功能, 在该领域设置新标准。

9.0.0 — 将进化准动力学和 `vc-NEB` 代码添加到 USPEX 软件包, 增加准动力学的张量的版本, 添加额外的数据和后处理工具, 清理代码的输出。去除 INPUT 中的部分参数。改进软模变异。2012 年 4 月。

- 9.1.0 — 发布版本。清理并整合。用户人数超过 800 人。发布于 28/05/2012。
- 9.2.0 — 运行 GEM (广义进化准动力学)。持续开发 GEM 代码。空间群测定限度作为新的输入参数。改进大量置换变异参数。2012.7-2012.8。
- 9.2.1-9.2.3 — 改进 GEM, 种群多样性和超胞尺寸, 改进选择模式。2012.9-2012.10。
- 9.2.4-9.2.6 — (9.2.4 发行版本)。多数输入参数采用智能默认值。改进的对称集群初始化能力。为纳米粒子增强有序性。新的参数用以调整空间群测定的限度。修正几个小错误。可以优化新的性质 (准熵)。完全集成的 vc-NEB 的代码。2012.11-2012.12。
- 9.2.7. — 发布版本。在没有结构弛豫的情况下进行合金的有序优化, (在不依托于特定的准随机机制而是通常用的普通随机机制的情况下, 更易于产生准随机结构)。改进对称代 (这点对于定晶胞计算非常重要)。在固定晶胞计算过程中, 可以指定一个晶胞参数, 不一定非是一个 3×3 矩阵, 也可以是一排六个值 (三个以埃为单位的长度值和三个以度为单位的角度值)。为了置换数目的最大化 (参数 `howManySwaps`), 引入了智能默认值。添加新的测试, 清除当前参数并启用旧参数。与 CASTEP 接口 (感谢 Z. Raza, X. Dong 和 AL)。用户人数达到 1160 人。30/12/2012
- 9.3.0-9.3.3 — 修正在一个在随机对称结构代中出现的错误 (这个错误出现于 9.2.7 版本)。显著简化输入和输出。创建 `OUTPUT.txt` 用以存放重要信息。对分子晶体进行晶体分裂。通过允许用户设定初始成分来改进变成分计算与 LAMMPS 和 CASTEP 的接口。添加新的测试案例。20/03/2013。
- 9.3.4 — 发布版本, 并做了清理。25/03/2013。
- 9.3.5 — 添加并调试了预测二维晶体的代码。19/04/2013。
- 9.3.6 — 集成和调试了 GEM。改进变成分计算。二维晶体包含平面种群。29/04/2013。
- 9.3.8 — 一维聚合物晶体中添加平面群描述, 改进变化学计量比的表面结构预测。19/06/2013。
- 9.3.9 — 发布版本。显著改进的版本, 加强用户体验, 新功能 (如二维晶体, GEM) 表现良好, 改进变成分算法 (同时支持单块计算, 即定组分但改变单胞内原子个数)、全功能表面计算, 新类型优化 (可优化介电常数、键长和新的介电材料的品质因数)。筹备在新的测试中与 LAMMPS 和 ATK 接口。通过微小调试不断更新 (最后调试于 10/02/2014)。
- 9.4.1 — 主要升级, 极大地提高用户使用性便利性 (自动预估体积和每种案例的变异运算操作的比例), 新功能 (弹性性质的优化和 Chen 氏硬度模型, 预测聚合物的结构、反组分、自动统计分析和改进的种子技术), 第一次发布 GEM, 提供一系列 USPEX 的计算实例, 测试案例, 说明文件。用户人数超过 2100 人。发布于 30/12/2014。
- 9.4.2 — 发布版本, 兼容 Octave 3.4 中的错误。重新写入凸包图代码。与 MOPAC 接口。启用默认参数 `goodBonds`, `valences`, `IonDistances`。三元、四元和更复杂成分的变成分搜索表现良好。用户超过 2200 人。发行于 21/03/2015。
- 9.4.3 — 发布版本。包括一系列的代码优化 (这些更新将会加快代码的运行速度), 与 MOPAC 的接口, 改进了文件说明。这个版本几乎没有 bug 出现, 此版本将会在版本 10. 中被进一步提高。发布于 10/08/2015。
- 9.4.4 — 发布版本。解决了确定空间群以及用户反馈的问题, 改进了说明文档并丰富了实例, 完全兼容 Octave 3.4 以及部分兼容 Octave3.6/3.8/4.0。这一版本基本没有 bug, 是 USPEX 的一个里程碑。发布于 05/10/2015。
- 10.1 — 试验版, 发布于 01/08/2018。
- 10.2 — 发布版本。以编译过的程序发布, 因此用户不再需要 Matlab。没有已知的错误并包含了大量的新特点和新改进。可随机依据拓扑结构产生初始结构并能自动控制输入参数, 极大地加速了计算速度。能够预测磁性材料。引入了多种新的优化函数: 磁矩、双折射率、热电品质因子 ZT、断裂韧性。能够对多种性质进行 Pareto 优化。USPEX 与 Gaussian、MOPAC、DFTB、ORCA、FHI-aims、ABINIT 等代码接口。利用 SPGLIB 确定晶体结构对称性。计算物理性质时, 可以选择性引入对称性操作从而简化计算并得到可靠结果。利用 80 种平面群产生初始二维结构。能够变组分预测二维材料。对于表面结构预测, 我们能够考虑所有可能的表面超胞类型并可输出表面结构相图。变

胞 NEB 方法被充分改进 (计算速度提高好几倍)。引入 V. Stevanovic 路径用于预测最可能的相变机制 (可以直接作为变胞 NEB 代码的输入)。发布于 19/01/2019。

10.4 — 局部改进, 修正漏洞, 添加了一些新的功能。Local 和 remote 模式提交作业的方式被改进。VCNEB 和 PSO 的代码被改进。与 QE、DMACRYS、VASP 和 DFTB+ 的接口被改进。添加了与 Abinit 和 CRYSTAL 的接口。添加了 Half-metallicity fitness。添加了 Mazhnik-Oganov 硬度与裂纹耐受度模型。所有 python 脚本的语法与 python3 保持一致。

10.5 — 修正了一些漏洞。改进了与 DFTB+ 和 GULP5.2 的接口。添加了“USPEX -u”功能, 该命令可以直接升级 USPEX 至最新版本而无需下载整个安装包。添加了由机器学习模型计算弹性常数的功能 (optType=1201-1207), 该方法基于图形卷积神经网络, 例子 35 展示了如何使用该功能。

2 初始化

2.1 如何获得 USPEX

USPEX 是一款开源软件，可以从以下网址下载：

<http://uspex-team.org>

在下载页面，用户需要注册申请，收到密码后即可下载 USPEX。我们提供 USPEX 源代码包，示例和手册文件。

2.2 版权引用

使用 USPEX 开展工作时，用户在发表研究或者做报告中必须引用 USPEX，例如下面这种方式：

“Crystal structure prediction was performed using the USPEX code^{2;12;17}, based on an evolutionary algorithm developed by Oganov, Glass, Lyakhov and Zhu and featuring local optimization, real-space representation and flexible physically motivated variation operators”.

用户可以查阅 `OUTPUT.txt` 文件后知悉有关 USPEX 程序的重要文献。

2.3 错误报告

像其它代码一样，USPEX 在运行过程中可能存在错误。如果你在模拟中发现异常，你可以将错误反馈到 USPEX 团队：

<https://groups.google.com/forum/#!forum/uspex>

当你反馈错误的时候，请详细地描述你的问题并附上 `INPUT.txt`、`OUTPUT.txt`、`log` 和其他相关文件。你也可以将问题或错误描述发至 (bugreport@uspex-team.org)。

2.4 运行 USPEX 所需的计算机配置

USPEX 可以在任何基于 unix 开发的平台上（Linux, Unix 或 Mac OS X）安装及运行 - 只需要一个 CPU 并已安装 Python 即可。USPEX 使用远程作业机制，可以连接任何远端机进行计算。

2.5 USPEX 的外部接口代码

USPEX 产生的结构需要借助外部接口程序进行结构优化以及性质计算。USPEX 会对已优化结构的性质进行排序并据此产生新的结构，然后再借助外部接口软件对这些新结构进行结构优化以及性质

计算, 然后再进行性质排序并再产生新结构...。我们利用已知的完备的从头算 (或经典力场) 程序进行结构优化和能量计算。当前, 与 USPEX 接口的软件有:

- VASP — <https://www.vasp.at/>
- SIESTA — <http://departments.icmab.es/leem/siesta/>
- GULP — <http://nanochemistry.curtin.edu.au/gulp/>
- LAMMPS — <http://lammms.sandia.gov/>
- DMACRYS — <http://www.chem.ucl.ac.uk/basictechorg/dmacrys/index.html>
- CP2K — <http://www.cp2k.org/>
- Quantum Espresso — <http://www.quantum-espresso.org/>
- FHI-aims — <https://aimsclub.fhi-berlin.mpg.de/>
- Abinit — <https://www.abinit.org>
- CRYSTAL — <https://www.crystal.unito.it/>
- ATK — <http://quantumwise.com/>
- CASTEP — <http://www.castep.org/>
- Tinker — <http://dasher.wustl.edu/tinker/>
- MOPAC — <http://openmopac.net/>
- Gaussian — <http://www.gaussian.com/index.htm>
- ORCA — <https://orcaforum.cec.mpg.de>
- BoltzTrap — http://www.imc.tuwien.ac.at/forschungsbereich_theoretische_chemie/forschungsgruppen/prof_dr_gkh_madsen_theoretical_materials_chemistry/boltztrap/
- DFTB — <http://www.dftb-plus.info>

选择这些程序的原因有: 1) 优化结构的效率; 2) 可靠性; 3) 通用性。当然, 还有其它很多程序也满足这些标准, USPEX 在未来将开发与它们的接口。

2.6 如何安装 USPEX

USPEX 版本 10.2 已使用 Matlab 编译器进行编译，可以作为单机软件使用。因此，使用这个版本时，你不再需要安装 Matlab。作为替代，当你运行该版本时，你需要在目标服务器上安装 Matlab Runtime 软件 (MCR-R2016b)。USPEX 软件包里已包含合适的 MCR。再次提醒，你需要同时安装 MCR 和 USPEX 软件。

当你下载 USPEX 软件包后，需要解压并运行以下命令安装 USPEX 到个人或系统帐户下：

```
sh install.sh
```

安装者不要求具有 root 权限。安装 USPEX 版本 10.2 及 MCR-R2016b 有以下四个步骤：

- 选择安装方法（图形界面安装或终端安装）；
- 同意 USPEX 和 MCR 相关条款；
- 选择 USPEX 和 MCR 安装路径（如果在 root 目录安装，用户需要 root 权限）；
- 设置环境变量（在终端安装时，环境变量会自动设置）。

在最后一个步骤中，如果用户选择终端安装，环境变量会被自动写入 .bashrc (bash shell 系统) 或 .cshrc (C shell 系统)。对于图形界面安装，用户需要提供环境变量信息以确保 USPEX 能够正常运行。例如：

```
For Bash shell system, add these lines in ~/.bashrc or ~/.profile or /etc/profile:  
export PATH=/home/user/bin/USPEX/application/archive:$PATH  
export USPEXPath=/home/user/bin/USPEX/application/archive/src  
export MCRROOT=usr/local/MATLAB/MATLAB_Runtime
```

```
For C shell system, add these lines in ~/.cshrc or ~/.profile or /etc/profile:  
setenv PATH "/home/user/bin/USPEX/application/archive:$PATH"  
setenv USPEXPath "/home/user/bin/USPEX/application/archive/src"  
setenv MCRROOT usr/local/MATLAB/MATLAB_Runtime
```

*** 请注意 MCRROOT 路径需要在 v91 文件夹前。*** 我们并不建议在 root 帐户安装程序。

2.7 如何运行 USPEX

为了运行 USPEX，你必须已安装 Python 并且在计算节点上有可运行的外部代码用于结构弛豫和计算结构能量 (USPEX 支持的代码详见章节 2.5)。开始你的计算时，根据你计算的体系来从示例中找到类似的例子（见附录 9.1），然后开始编辑 INPUT.txt 文件。INPUT.txt 文件中各参数的解释详见 4 章节。然后，将外部代码做结构弛豫需要的文件放在 Specific/文件夹下，以 vasp 为例，这样的文件有 INCAR_1, INCAR_2, ..., INCAR_N, 和 POTCAR_A, POTCAR_B, ..., 这里的 A, B, ... 指的是与 POTCAR 文件相对应的化学元素名称。

另外一件非常重要的事情是准备执行外接程序的脚本。USPEX 允许用户自行决定提交作业的方式：本地和远程。请阅读关键词 `whichCluster` 和章节 8.8。

执行 USPEX 的命令：

```
USPEX -r
```

`log` 文件用于记录模拟过程中的信息，如果有错误的话（如果你愿意报告软件的漏洞，请把 `log` 文件发送给我们）。`OUTPUT.txt` 文件包含有每一代计算和分析的详细信息。

对于 USPEX 使用者，我们提供了一些人性化的选项：

- `-v, --version`: 显示程序的版本号并退出
- `-h, --help`: 显示帮助信息并退出
- `-e, --example`: 显示 USPEX 例子的细节。如果没有值或者是 'all'，则所有的例子都将被显示。
- `-c NUM, --copy=NUM`: 复制 INPUT 文件并指明例子所在文件夹。
- `-r, --run`: 运行 USPEX 计算
- `--clean`: 清理计算文件夹
- `--list`: 显示已存在计算作业的地址。

当进行大规模并行计算时，用户需要根据用户的电脑配置文件。

有两种提交作业的方式：（1）本地提交；（2）远程提交，取决于你是否将 *ab initio* 计算任务提交在与你运行 USPEX 相同的服务器，或者发送至远程的超级计算机。参见关键字 `whichCluster` 以及 8.8 章节。

2.8 执行 USPEX 示例：简明教程

下载并安装 USPEX 软件后，我们就可以运行 USPEX 示例。这些示例的详细信息参见附录 9.1。运行这些示例（除 EX13 外）的外接代码显示如下：

- GULP: EX02, EX03, EX08, EX12, EX15 (VCNEB), EX16, EX18, EX21 (META), EX22 (Generalized META), EX23
- VASP: EX01, EX07, EX09, EX14 (META), EX17, EX19, EX24, EX29, EX30, EX31
- LAMMPS: EX04, EX26 (TPS)
- ATK: EX05
- CASTEP: EX06

- DMACRYS: EX10
- Tinker: EX11
- MOPAC: EX20
- DFTB: EX25
- FHIaims: EX27, EX28

现在，我们开始体验 USPEX：

2.8.1 测试 USPEX-Python 指令

我们可以使用以下命令显示当前 USPEX 版本信息：>> USPEX -v

如果 USPEX 运行正常，你应该得到以下信息。

```
USPEX Version 10.2 (19/01/2019)
```

2.8.2 运行示例 EX13-3D_special_quasirandom_structure_TiCoO₂

示例 13 (EX13) 不需要借助任何外部代码，我们可以通过运行这个例子来熟悉 USPEX 的运行步骤。运行这个实例大约会花费 ~30 分钟左右的时间。我们首先创建一个测试文件夹，将示例文件复制过来，然后通过 USPEX Python 指令集运行，命令如下：

```
>> mkdir EX13
>> cd EX13
>> USPEX -c 13
>> USPEX -r
```

在运行示例的同时，我们可以了解 EX13 更多的细节。在 EX13 中，我们通过进化算法优化结构有序度。所以，在 INPUT.txt 中，我们设置：

```
USPEX : calculationMethod
-4    : optType
```

我们使用以下参数：

```
300   : calculationType
```

```
% atomType
Co Ti 0
% EndAtomType
```

```
% numSpecies
```

```
16 16 64
% EndNumSpecies
```

这些参数表示我们正在研究 $\text{Co}_{16}\text{Ti}_{16}\text{O}_{64}$ 体系。

由于我们不需要外部代码，我们可以简单设置：

```
% abinitioCode
0
% ENDabinit
```

种子文件放置在 `Seeds/POSCARS`，即初始的 $\text{Ti}_{16}\text{Co}_{16}\text{O}_{64}$ 结构。

当你发现文件 `USPEX_IS_DONE` 时，恭喜你，你已经成功完成了第一个示例。接下来，我们将运行那些需要外部代码的示例。

2.8.3 借助外部代码运行示例

这一步中，我们建议运行将 GULP 或 VASP 作为外部代码的示例，如 EX02 或 EX01。首先利用 USPEX 指令获得示例 EX02 的信息，然后创建一个单独的文件夹并复制示例文件，指令如下：

```
>> mkdir EX02
>> cd EX02
>> USPEX -c 2
```

在示例 EX02 中，我们使用 GULP 作为外部代码，因此设置：

```
% abinitioCode
3 3 3 3
% ENDabinit
```

在不能批量提交作业的系统中运行一系列计算任务，我们需要改变 `INPUT.txt` 中的以下参数：

```
1      : whichCluster
1      : numParallelCalcs
```

在示例 `INPUT.txt` 文件中，`whichCluster=QSH`，QSH 是我们内部团队所使用集群的名字。用户可以以相同的方式定义自己的集群。详细信息参见 8.8。

在 `INPUT.txt` 中，我们没有指定怎么运行 GULP，因为我们假定所有用户使用相同的命令：

```
% commandExecutable
gulp < input > output
% EndExecutable
```

但是要确保这个命令是在你的计算机上执行的。如果你想使用 VASP 运行 EX01, 你应该这样设置:

```
% abinitioCode
1 1 1 1
% ENDabinit

% commandExecutable
mpirun -np 8 vasp
% EndExecutable
```

如果 `commandExecutable` 设置错误, 启动 USPEX 计算就会失败。请注意, 根据你自己电脑或超级电脑的基本情况, 你需要适当修改 `submitJob_remote.py` 和 `checkStatus_remote.py` 文件。

当每个参数都被正确设置后, 我们就可以通过 USPEX 指令进行计算了。执行命令:

```
>> USPEX -r
```

2.8.4 检查结果

开始计算后, 你可以检查 `results1/` 文件夹中的 OUTPUT 文件。现在, 你有了使用 USPEX 运行简单计算的经验。要想了解关于 USPEX 更详细的信息, 请仔细阅读本手册余下内容。分析结果时, 我们认为对结构进行可视化是很重要的 (可视化方法, 详见 8.1)。

3 输入与输出文件

输入/输出文件与进行结构弛豫的外部代码相关。我们一般采用多步结构弛豫策略。显然，为了保证结构能量排序的正确性，最终结构及能量信息必须是高质量的。大多数由 USPEX 产生的新结构与局域最优结构相差较大，但是对这些初始结构进行高质量的结构弛豫是耗时的。如果最初的结构弛豫使用较为粗略的收敛参数，只有最后一步的弛豫才进行高精度的计算，则可以避免耗时长的问题。第一步的结构弛豫可以使用耗时少的计算方法或较为粗糙的计算参数（基组， k 点采样，赝势）或不同的近似方法（力场 *vs.* LDA *vs.* GGA）或不同结构弛豫的代码。我们强烈建议初始优化时，保持体积不变，仅优化晶胞形状和原子位置。当固定体积进行优化时，你不必担心平面波计算中的 Pulay 应力 - 使用小基组是合适的。然而，当优化体积时，你需要设置一个高质量的基组进行计算。对于初期的结构弛豫，你不必设置大的 k 点，但是不要忘记在最终的结构弛豫时增加 k 点，以获得精确的能量。利用你的智慧，成为一个专家，并记住一个低质量的结构弛豫会毁掉你的计算。

3.1 输入文件

假设计算所在文件目录是 `~/StructurePrediction`，当运行 USPEX 时，这个目录下将会包含：

- 文件 `INPUT.txt`，详见 4 章节。
- Subdirectory `~/StructurePrediction/Specific/`，该文件夹里包含 VASP 或 SIE-STA 或 GULP (*etc.*) 等外部代码的可执行文件，这些文件以数字进行编码 - `INCAR_1`, `INCAR_2`, ... 和赝势文件。
- Subdirectory `~/StructurePrediction/Seeds`，该文件中包含种子结构以及推荐/不推荐化学组分。其中列出了成分/反成分。种子结构以 VASP5 POSCARS 格式书写并放置在 `POSCARS` 或者 `POSCARS_gen` (`gen` 是结构代数) 文件中。化学成分文件 `compositions` 和 `Anti-compositions` 用来控制变成成分计算。
- Subdirectory `~/StructurePrediction/AntiSeeds` - 你可以在这个文件夹中放置在计算中希望避免产生的结构。
-

3.1.1 Specific 文件夹

经过数字标记的用于结构弛豫的可执行文件放在 `~/StructurePrediction/Specific/` 文件夹。

- 对于 VASP，文件 `INCAR_1`、`INCAR_2` 等定义了进行结构弛豫和总能计算的参数（我们推荐至少进行三次计算），相应的 `POTCAR_*` 等是赝势文件。例如，利用 `INCAR_1` 和 `INCAR_2` 在体积固定的情况下对结构的原子位置和晶胞常数进行粗略优化，`INCAR_3` 是在恒定压力下以中等精度对结构进行充分地结构弛豫，`INCAR_4` 是进行较为精确的结构优化。请记住，更高精度的

结构弛豫均是基于上一步优化结果的。所预测材料体系包含的相关元素的 POTCAR 文件也需放在 Specific/文件夹，比如 POTCAR_C、POTCAR_0 等

- 对于 SIESTA，你需要准备赝势文件和输入文件 `input_1.fdf`、`input_2.fdf` 等
- 对于 GULP，你需要准备文件 `goptions_1`、`goptions_2...` 以及 `ginput_1`、`ginput_2...`。前者定义优化性质，后者定义优化参数（原子间相互作用势、压力、温度和优化迭代次数等）。
- 对于 DMACRYS，基本的控制参数需要写入 `fort.22` 文件；经典力场写入 `fit.pots` 文件；内部分子间的最大键长写入 `cutoff` 文件。
- 对于 CASTEP，结构文件写入 `cell_1`、`cell_2` 等文件，而 `param_1`、`param_2` 等文件控制计算参数由。此外，还需要提供相应元素的赝势文件。
- 对于 CP2K，用户需要提供正常的 CP2K 输入文件 `cp2k_options_1`、`cp2k_options_2` 等。在这些输入文件中，除了原子坐标和晶胞参数外，其它参数信息必须完整。原子坐标和晶胞参数会写在 USPEX 的 INPUT 文件的最后一行“&END FORCE_EVAL”。你需要对计算项目以 USPEX 文进行命名，这是因为 USPEX 运行时需要从 `USPEX-1.cell` 和 `USPEX-pos-1.xyz` 文件中读取输出文件。和 VASP 类似，我们建议是至少进行三步弛 - 首先是固定晶格（仅优化原子位置），然后是充分的几何结构优化。
- 对于 Quantum Espresso，你需提供 `qEspresso_options_1`、`qEspresso_options_2` 等文件。在这些输入文件中，除了原子坐标、晶胞参数及 k 点信息外，其它输入参数信息必须完整。原子坐标、晶胞参数及 k 点信息会写在 USPEX 的 INPUT 文件中。我们建议进行多步弛豫。例如，利用 `qEspresso_options_1` 实现仅固定晶胞参数而优化原子位置的粗略计算，利用 `qEspresso_options_2` 基于一般精度条件在固定外压下进行结构弛豫，`qEspresso_options_3` 是高精度的几何结构优化。

为了正确运行 USPEX，你需要正确编辑 Specific 文件夹中的文件。在这里，我们以 VASP 作为例子：

- 你必须对最终的结构进行充分地几何结构优化弛豫并获得准确的总能。这样做的目的是确保结构间的能量排序是正确的（你可以查看输出文件 `E_series.pdf`）。
- 正确准备 POTCAR 文件。
- 对每一个结构设置至少三步弛豫，即至少有三个 INCAR 文件：`INCAR_1`、`INCAR_2`、`INCAR_3` 等。通常我们会进行 4 - 5 步弛豫。
- 通常，USPEX 产生的初始结构并不是局域最优的。在这种情况下，利用 `INCAR_1, 2` 固定晶胞体积（`ISIF=4`）优化原子位置和晶胞形状，利用 `INCAR_3,4` 中进行充分地几何结构优化（`ISIF=3`），利用 `INCAR_5` 进行一次非常精确的单元能计算（`ISIF=2`，`NSW=0`）。注意：当你固定晶胞进行结构预测时，或在进行进化赝动力学模拟（全结构弛豫除外）时，你必须在所有的 INCAR 文件中设置 `ISIF=2`。

- 当晶胞体积固定时，你可以使用默认的平面波截断能。而当你在 INCAR 中使用 ISIF=3，你必须将截断能增加 30-40%，否则你会得到一个大的 Pulay 应力。同样的，收敛精度可以从低变高，例如：在 INCAR_1 中使用 EDIFF=1e-2 和 EDIFFG=1e-1，然后逐步提高收敛精度，在 INCAR_4 中一般设置 EDIFF=1e-4 和 EDIFFG=1e-3。为了保证对结构进行较为充分的弛豫，timesteps(NSW) 要设置大一些；但也不能太大以避免在一些不好的结构上浪费太多计算时间。如果你的计算体系越大，NSW 就应该越大。
- 选择一个有效的弛豫算法可以节省很多时间，在 VASP 中，我们推荐在开始弛豫时使用共轭梯度算法 (IBRION=2, POTIM=0.02)，而当结构接近局部最优时，可更改参数至 IBRION=1 和 POTIM=0.3。
- 即使你研究的是绝缘体系，USPEX 产生的许多构型将会是金属性的。所以为了得到好的预测结果，你必须对体系使用“金属化”处理（这对于金属和绝缘体同样适用）。因此，我们推荐 ISMEAR=1。对于金属体系，使用 ISMEAR=1, SIGMA=0.1-0.2。对于绝缘材料体系，我们推荐 ISMEAR=1, SIGMA 从 0.1 (INCAR_1) 逐渐降至 0.05。

这里，我们提供一个关于如何设置 INCAR 文件的示例。在该材料体系中，每个单胞中有 16 个 C 原子，默认 POTCAR 中的 ENCUT=400 eV：

INCAR_1:	INCAR_2:	INCAR_3:
PREC=LOW	PREC=NORMAL	PREC=NORMAL
EDIFF=1e-2	EDIFF=1e-3	EDIFF=1e-3
EDIFFG=1e-1	EDIFFG=1e-2	EDIFFG=1e-2
NSW=65	NSW=55	ENCUT=520.0
ISIF=4	ISIF=4	NSW=65
IBRION=2	IBRION=1	ISIF=3
POTIM=0.02	POTIM=0.30	IBRION=2
ISMEAR=1	ISMEAR=1	POTIM=0.02
SIGMA=0.10	SIGMA=0.08	ISMEAR=1
		SIGMA=0.07
INCAR_4:		ISMEAR=1
PREC=NORMAL		SIGMA=0.06
EDIFF=1e-4		
EDIFFG=1e-3	INCAR_5:	
ENCUT=600.0	PREC=NORMAL	
NSW=55	EDIFF=1e-4	
ISIF=3	EDIFFG=1e-3	
IBRION=1	ENCUT=600.0	
POTIM=0.30	NSW=0	

```
ISIF=2
IBRION=2
POTIM=0.02
ISMEAR=1
SIGMA=0.05
```

设置进化赝动力学模拟文件（`calculationMethod=META`）与进行 USPEX 预测（`calculationMethod=USPEX`）时基本一致。如果你想固定晶胞体积，你需要设置 `ISIF=2`。如果需要充分地结构弛豫，你需要设置 `ISIF=3`。如果你的设置如下：

```
% abinitioCode
1 1 1 (1 1)
% ENDabinit
```

那么这意味着在你的五个 INCAR 文件中，`ISIF` 依次分别是 “2 2 2 3 3”。

不同于 USPEX，VC-NEB 方法并不需要利用外部代码进行结构弛豫，只需要利用外部代码计算获得的力。以 VASP 的 INCAR 文件为例，我们需要设置 `NSW=0` 以避免结构弛豫，同时需要设置 `ISIF=2` 或者 `3` 获得原子上的力以及应力张量。我们建议设置 `PREC=Accurate` 从而对力和应力有一个好的估计以加速 VC-NEB 的计算。一个针对 VC-NEB 的 INCAR 文件例子如下：

```
INCAR_1:
PREC=Accurate
EDIFF=1e-4
EDIFFG=1e-3
ENCUT=600.0
NSW=0
ISIF=2
IBRION=2
POTIM=0.02
ISMEAR=1
SIGMA=0.05
```

3.2 输出文件

输出文件一般在 `result*` 文件夹里。当查看 `Individual` 文件里罗列的结构空间群时，请注意 USPEX 经常会低估结构的空群，这是由于由于结构预测时进行结构弛豫的精度仍不是特别高以及利用相对较高的精度去确定空群。你应该可视化所预测的结构。为了获得确信的群，你可以用较低的精度重新确定空群（但是这种方式可能很危险），或者以更高的精度对结构进行弛豫。

~/StructurePrediction/results1 包含以下文件：

- `OUTPUT.txt` — 依据 USPEX 输入变量，罗列预测产生的结构和它们的特征值。

- **Parameters.txt** — INPUT.txt 文件的复制，供参考。
- **Individuals** — 列出每个 USPEX 产生结构的详细信息（能量，晶胞体积，空间群，该结构的产生方式，用于计算能量的 k 点网格，结构有序度等）。文件 BESTIndividuals 给出每一代最好结构的信息。Individuals 文件如下：

Gen	ID	Origin	Composition	Enthalpy (eV)	Volume (Å ³)	Density (g/cm ³)	Fitness	KPOINTS	SYMM	Q_entr	A_order	S_order
1	1	Random	[4 8 16]	-655.062	201.062	4.700	-655.062	[1 1 1]	1	0.140	1.209	2.632
1	2	Random	[4 8 16]	-650.378	206.675	4.572	-650.378	[1 1 1]	1	0.195	1.050	2.142
1	3	Random	[4 8 16]	-646.184	203.354	4.647	-646.184	[1 1 1]	1	0.229	0.922	1.746
1	4	Random	[4 8 16]	-649.459	198.097	4.770	-649.459	[1 1 1]	9	0.128	0.958	2.171
1	5	Random	[4 8 16]	-648.352	202.711	4.662	-648.352	[1 1 1]	2	0.154	1.014	2.148
1	6	Random	[4 8 16]	-643.161	206.442	4.577	-643.161	[1 1 1]	1	0.234	0.946	1.766
1	7	Random	[4 8 16]	-647.678	207.119	4.562	-647.678	[1 1 1]	1	0.224	1.108	2.106
1	8	Random	[4 8 16]	-644.482	203.844	4.636	-644.482	[1 1 1]	1	0.215	0.952	1.857
1	9	Random	[4 8 16]	-647.287	204.762	4.615	-647.287	[1 1 1]	40	0.136	1.142	2.563
1	10	Random	[4 8 16]	-649.459	198.097	4.770	-649.459	[1 1 1]	9	0.128	0.958	2.171

- **convex_hull** — 仅限变成分计算，给出了所有热力学稳定结构的成分和它们的焓值（每原子）。示例如下：

```

---- generation 1 -----
10 0 -8.5889
 0 14 -8.5893
11 3 -8.7679
---- generation 2 -----
10 0 -8.5889
 0 14 -8.5893
11 3 -8.8204
---- generation 3 -----
10 0 -8.5889
 0 14 -8.5893
12 4 -8.9945
.....

```

- **gatheredPOSCARS** — 所有经过结构优化后的结构（以 VASP5 POSCAR 的格式）。例如：

```

EA1 9.346 8.002 2.688 90.000 90.000 90.000 Sym.group: 1
1.0000
 9.346156 0.000000 0.000000
 0.000000 8.002181 0.000000
 0.000000 0.000000 2.688367
Mg Al O
4 8 16
Direct
 0.487956 0.503856 0.516443
 0.777565 0.007329 0.016443
 0.987956 0.507329 0.016443
 0.277565 0.003856 0.516443
 0.016944 0.178753 0.016443
 0.019294 0.833730 0.516443
 0.746227 0.333730 0.516443
 0.748577 0.678753 0.016443
 0.516944 0.832431 0.516443
 0.519294 0.177455 0.016443
 0.246227 0.677455 0.016443
 0.248577 0.332431 0.516443
 0.416676 0.241774 0.516443
 0.559871 0.674713 0.016443
 0.205650 0.174713 0.016443
 0.348845 0.741774 0.516443
 0.613957 0.380343 0.016443

```

```

0.804054    0.542164    0.516443
0.113957    0.630842    0.516443
0.304054    0.469021    0.016443
0.848845    0.269411    0.016443
0.705650    0.836472    0.516443
0.059871    0.336472    0.516443
0.916676    0.769411    0.016443
0.651564    0.130842    0.516443
0.461467    0.969021    0.016443
0.151564    0.880343    0.016443
0.961467    0.042164    0.516443
EA2      9.487  4.757  4.580 90.243 90.188 89.349 Sym.group:  1
1.0000

```

```

9.486893    0.000000    0.000000
0.054041    4.756769    0.000000
-0.014991  -0.019246    4.579857

```

```

Mg  Al  O
4   8  16

```

```

Direct
0.499837    0.633752    0.011361
0.500082    0.131390    0.482012
0.813573    0.257696    0.494520
0.326111    0.625491    0.501746
0.995267    0.254346    0.992293
0.160822    0.689054    0.001270
0.995907    0.760753    0.498354
0.159742    0.192300    0.491958
0.811206    0.761223    0.997857
0.325692    0.125479    0.987935
0.656355    0.695175    0.503322
0.656596    0.199917    0.991605
0.487990    0.763078    0.627771
0.845518    0.645378    0.347890
0.623474    0.895186    0.185946
0.616379    0.395875    0.308861
0.093745    0.991831    0.185467
0.092669    0.494591    0.309957
0.847697    0.118765    0.113434
0.475636    0.251449    0.875207
0.327510    0.787484    0.116764
0.720411    0.975740    0.706398
0.200804    0.880147    0.683027
0.975416    0.612789    0.852917
0.986131    0.108285    0.644081
0.204805    0.364607    0.830780
0.718464    0.496262    0.817031
0.323904    0.257705    0.340590
.....

```

- **BESTgatheredPOSCARS** — 每一代中最好的结构。
- **gatheredPOSCARS_unrelaxed** — 由 USPEX 产生，但未经过结构弛豫的结构。
- **enthalpies_complete.dat** — 给出每一个结构每一步弛豫的焓值。
- **origin** — 显示所有结构的产生方式及相关信息。例如:

```

ID  Origin  Enthalpy  Parent-E  Parent-ID
1   Random  -23.395  -23.395  [ 0]
2   Random  -23.228  -23.228  [ 0]
3   Random  -23.078  -23.078  [ 0]
4   Random  -23.195  -23.195  [ 0]
5   Random  -23.155  -23.155  [ 0]
6   Random  -22.970  -22.970  [ 0]
7   Random  -23.131  -23.131  [ 0]
8   Random  -23.017  -23.017  [ 0]
9   Random  -23.117  -23.117  [ 0]
10  Random  -23.195  -23.195  [ 0]
.....

```

- gatheredPOSCARS_order — gatheredPOSCARS 的内容基本相同，额外给出了每个原子的局域有序度参量值 (Ref. 16)。例如：

```
EA1 9.346 8.002 2.688 90.000 90.000 90.000 Sym.group: 1
1.0000
9.346156 0.000000 0.000000
0.000000 8.002181 0.000000
0.000000 0.000000 2.688367
Mg Al O
4 8 16
Direct
0.487956 0.503856 0.516443 1.1399
0.777565 0.007329 0.016443 1.1399
0.987956 0.507329 0.016443 1.1399
0.277565 0.003856 0.516443 1.1399
0.016944 0.178753 0.016443 1.1915
0.019294 0.833730 0.516443 1.2474
0.746227 0.333730 0.516443 1.2474
0.748577 0.678753 0.016443 1.1915
0.516944 0.832431 0.516443 1.1915
0.519294 0.177455 0.016443 1.2474
0.246227 0.677455 0.016443 1.2474
0.248577 0.332431 0.516443 1.1915
0.416676 0.241774 0.516443 1.2914
0.559871 0.674713 0.016443 1.1408
0.205650 0.174713 0.016443 1.1408
0.348845 0.741774 0.516443 1.2914
0.613957 0.380343 0.016443 1.2355
0.804054 0.542164 0.516443 1.2161
0.113957 0.630842 0.516443 1.2355
0.304054 0.469021 0.016443 1.2161
0.848845 0.269411 0.016443 1.2914
0.705650 0.836472 0.516443 1.1408
0.059871 0.336472 0.516443 1.1408
0.916676 0.769411 0.016443 1.2914
0.651564 0.130842 0.516443 1.2355
0.461467 0.969021 0.016443 1.2161
0.151564 0.880343 0.016443 1.2355
0.961467 0.042164 0.516443 1.2161
EA2 9.487 4.757 4.580 90.243 90.188 89.349 Sym.group: 1
1.0000
9.486893 0.000000 0.000000
0.054041 4.756769 0.000000
-0.014991 -0.019246 4.579857
Mg Al O
4 8 16
Direct
0.499837 0.633752 0.011361 0.9368
0.500082 0.131390 0.482012 1.0250
0.813573 0.257696 0.494520 0.9805
0.326111 0.625491 0.501746 0.9437
0.995267 0.254346 0.992293 0.9241
0.160822 0.689054 0.001270 1.1731
0.995907 0.760753 0.498354 0.9696
0.159742 0.192300 0.491958 1.2666
0.811206 0.761223 0.997857 1.0215
0.325692 0.125479 0.987935 1.0353
0.656355 0.695175 0.503322 1.2291
0.656596 0.199917 0.991605 1.2547
0.487990 0.763078 0.627771 1.1199
0.845518 0.645378 0.347890 0.9725
0.623474 0.895186 0.185946 0.9990
0.616379 0.395875 0.308861 1.0141
0.093745 0.991831 0.185467 1.1451
0.092669 0.494591 0.309957 1.1164
0.847697 0.118765 0.113434 0.8821
0.475636 0.251449 0.875207 1.0609
0.327510 0.787484 0.116764 1.0451
0.720411 0.975740 0.706398 0.9539
0.200804 0.880147 0.683027 0.9398
0.975416 0.612789 0.852917 1.1191
```

```

0.986131 0.108285 0.644081 0.9803
0.204805 0.364607 0.830780 1.0915
0.718464 0.496262 0.817031 1.0663
0.323904 0.257705 0.340590 1.1471
.....

```

- `goodStructures_POSCARS` and `extended_convex_hull_POSCARS` — 这两个文件分别与定成分和变成分预测对应。这两个文件列出了能量较优的结构，以稳定性递减的顺序展示：从最稳定的结构开始，到最不稳定的。
- `compositionStatistic` — 总计所有结构的化学成分并依据产生这些结构的方法统计结构数量。例如：

```

      Comp/Ratio      Total      Random Heredity Mutation Seeds  COPEX  Best/Convex
[ 0.0000 1.0000 ] 30( 63)      21      8      1      0      0      33
      0      8      4( 4)      2      2      0      0      0      0
      0      9      2( 2)      2      0      0      0      0      0
      0     10      5( 5)      3      2      0      0      0      0
      0     11      2( 2)      0      1      1      0      0      0
      0     12      6( 35)      6      0      0      0      0      29
      0     13      2( 3)      1      1      0      0      0      1
      0     14      0( 0)      0      0      0      0      0      0
      0     15      2( 2)      2      0      0      0      0      0
      0     16      5( 8)      5      0      0      0      0      3
      0      3      1( 1)      0      1      0      0      0      0
      0      4      1( 1)      0      1      0      0      0      0
[ 0.1250 0.8750 ] 52( 52)      32      9     11      0      0      0
      1      7     20( 20)      4      8      8      0      0      0
      2     14     32( 32)      28      1      3      0      0      0
[ 0.1111 0.8889 ] 25( 25)      9     14      2      0      0      0
      1      8     25( 25)      9     14      2      0      0      0
[ 0.1000 0.9000 ] 16( 16)      9      5      2      0      0      0
      1      9     16( 16)      9      5      2      0      0      0
[ 0.0909 0.9091 ] 11( 11)      5      4      2      0      0      0
      1     10     11( 11)      5      4      2      0      0      0
[ 0.0833 0.9167 ] 17( 17)      8      2      7      0      0      0
      1     11     14( 14)      8      2      4      0      0      0
      2     22      3( 3)      0      0      3      0      0      0
.....

```

- graphical files (*.pdf) — 结果快速分析文件：
 - `Energy_vs_N.pdf` (`Fitness_vs_N.pdf`) — 能量（适应度）与结构序号的函数；
 - `Energy_vs_Volume.pdf` — 能量与体积的函数；
 - `Variation-Operators.pdf` — 子代与亲代的能量对比；不同的颜色对应不同的运算方法（此图可用于评估不同的变异运算的结果）；
 - `E_series.pdf` — 弛豫的第 i 步和第 $i+1$ 步的能量关系；有助于发现问题和改进弛豫文件输入。
 - 对于可变成分有另外的图 `extendedConvexHull.pdf`，用于显示形成焓与组分的函数。
 - `compositionStatistic.pdf` — visualization of `compositionStatistic` file.
 - `Surface_Diagram.pdf` — file (only for variable-composition surface structure predictions) showing surface phase diagram.

4 输入文件 `INPUT.txt` 的设置

典型的 `INPUT.txt` 文件在附录 9.3 中给出。下面我们介绍输入中最重要的参数。许多参数都有可靠的默认值，当你在输入文件中遗漏掉它们时，这些默认值就变得有用（这就允许用户可以有非常短的输入文件！）。那些没有预设默认值的参数就需要用户根据具体的体系进行设置。用户可以查阅在线工具 http://uspeX-team.org/online_utilities/，这些小工具可以帮助用户更好的准备 `INPUT.txt` 文件，涉及分子体系的文件以及分析一些结果。手册的章节 7 简要地讨论了这些工具用法。

4.1 计算类型

▷ *variable* `calculationMethod`

Meaning: 计算方法

可选值（特性）：

- USPEX — 进化算法晶体结构预测
- META — 进化赝动力学模拟
- VCNEB — 用变晶胞微动弹性方法确定相变路径
- PSO — 修正的 PSO 算法
- TPS — 相变路径抽样方法
- MINHOP — 最小跳跃方法
- COPEX — 协同式进化算法晶体结构预测，用于可靠地预测三元体系变成成分结构模拟

Default: USPEX

Format:

USPEX: `calculationMethod`

▷ *variable* `calculationType`

Meaning: 计算类型:

即，明确预测体晶或纳米团簇或表面结构等，该参数由四部分组成，包括 结构维度、体系涉及分子与否和 成分可变与否，以及 “s” or “S” 表示自旋状态选项:

- 结构维度:
“3” — 体材料

“2” — 表面, “-2” — 二维晶体

“1” — 聚合物

“0” — 纳米粒子

- 分子状态:

“0” — 无分子体系

“1” — 分子体系

- ? 在计算中化学组成的变化:

“0” — 定成分

“1” — 变成分

- magnetic calculation :

“s” or “S” – 考虑自旋计算

Default: 300

Format:

301: calculationType

注意: 如果 calculationType=310, 意味着, 你将预测分子晶体, 然后 USPEX 希望你提供含有所有的分子类型的分子几何体的 MOL_1, MOL_2, ...这些分子将被安置在新的几何结构中。有效的选择: 300 (s300), 301 (s301), 310, 311, 000(s000), 200(s200), 201(s201), -200(-s200), -201 (-s201), 110。

▷ *variable* **optType**

Meaning: 用户可以通过这个值来定义优化函数。对于焓值（自由能）和体积，默认设置是寻找最小值；对于其它 optTypes，默认设置是寻找最大值。你可以根据需求自主更改寻找最大还是最小值。你也可以用某个目标值来优化性质（例如，对太阳能光伏来说，寻找带隙接近于 1.34 eV 的材料是很有吸引力的）。

可能的值（特征）:

Value	Number	Description
enthalpy	1	预测稳定结构
volume	2	最小体积（每原子） （找到最密结构）
hardness	3	最大硬度 （找到最硬相）
struc_order	4	最有序结构 （发现最有序结构）
density	5	最大密度
diel_sus	6	最大介电常数 （仅适用于 VASP 和 GULP）
bandgap	7	最大带隙 （仅适用于 VASP）
diel_gap	8	最大电容量 （仅适用于 VASP）
mag_moment	9	最大磁化强度（例如，每体积磁矩最大） （仅适用于 VASP）
quasientropy	10	最大结构准熵值
birefringence	11	最大双折射率（最大及最小反射率差值最大）
HalfMetalicity	12	maximization halfmetalicity parameter (only for VASP)
ZT	14	最大热电品质因子
Fphon	17	最小自由能（有限温度）

与弹性力学相关的性质 (“11**”):

值	序号	说明
K, 体积弹性模量	1101	体弹性模量最大化
G, 剪切模量	1102	剪切模量最大化
E, 杨氏模量	1103	杨氏模量最大化
ν , 泊松比	1104	泊松比最大化
G/K, Pugh 模量比	1105	Pugh 模量比最大化
Hv, 维氏硬度	1106	维氏硬度最大化
Kg, 断裂韧性	1107	断裂韧性最大化
D, 德拜温度	1108	德拜温度最大化
Vm, 声速	1109	声速最大化
S-波速	1110	S-波速最大化
P-波速	1111	P-波速最大化

注意: 与弹性力学相关的性质的计算仅仅适用于 VASP (从 VASP5.1 开始) 和 GULP。对于 VASP 用户, 需要在 *Specific/*文件夹中增加至少一个 *INCAR_** 文件, *INCAR_** 文件应该包括的参数有 *IBRION=6*, *ISIF≥3* 和 *NFREE=4*。体积弹性模量剪切模量和杨氏模量的估计值是 Voigh-Reuss-Hill (VRH) 的平均值。注意这样算出的弹性性质数值上可能不稳定, 用户需要小心检查结果。用户也可以使用另外一种方法计算弹性模量 — machine learning 模型。

Users also can perform the calculation of the elastic properties using Machine Learning (ML) approach, where the elastic moduli are computed using a graph convolutional neural network¹⁸ and then hardness and fracture toughness are evaluated with the Mazhnik-Oganov model¹⁹.

ML-based elasticity-related properties ("12**"):

Value	Number	Description
K, Bulk Modulus	1201	maximization of bulk modulus
G, Shear Modulus	1202	maximization of shear modulus
E, Young's Modulus	1203	maximization of Young's modulus
ν , Poisson's ratio	1204	maximization of Poisson's ratio
G/K, Pugh's modulus ratio	1205	maximization of Pugh's modulus ratio
Hv, Vickers hardness	1206	maximization of Vickers hardness
Kg, Fracture toughness	1207	maximization of fracture toughness

注意: 断裂韧性的公式只对绝缘体有效 (以及 carbides, borides, hydrides)

Default: enthalpy

Format:

```
% optType
enthalpy (equivalent to Min_enthalpy)
% EndOptType

另外一个例子: % optType
Min_(bandgap-1.34)^2
% EndOptType
```

需要注意的是在后一个例子里, 我们优化的 *optType* 变量是一个数学表达式。数学表达式应该在圆括号内, 同时应该是有效的 MATLAB 表达形式。完整的表达 (包括 *min_* and *max_* 在内) 应该是一个整体, 不要在整个表达式里填入空格。例如 “(bandgap - 1.3) ^2” 或者 *min_* (bandgap-3) 都是无效的形式。

在当前 USPEX 版本, 我们增加了多目标函数 (Pareto) 优化功能, *optType* 变量设置可以如下:

```
% optType
3 1
% EndOptType
```

或者

```
% optType
max_Hardness enthalpy
% EndOptType
```

在进行多目标函数优化时，有必要先仅基于焓值进行结构搜索，然后再进行多种性质的 Pareto 优化。相关设置如下：

```
% optType
3 6 [5]
% EndOptType
```

```
或者% optType
max_Hardness min_Diel_sus [5]
% EndOptType
```

该设置表示，你将按照最大硬度以及最小介电常数进行结构搜索，前 5 代结构搜索是仅基于焓值进行优化的，从第 6 代结构搜索开始，切换至 Pareto 优化。

Fig. 5显示了以最大硬度作为优化函数（optType=hardness），寻找最硬的 TiO_2 晶体结构。结果显示， TiO_2 结构可能的最大硬度仅为 14GPa²⁰，反驳了 Dubrovinsky（2001）关于 TiO_2 是超硬材料的结论²¹。这个例子表明通过一个简单的 USPEX 结构预测就可能解决长期存在的争论。

Fig. 6是以最大硬度和最小焓值同时作为优化函数进行结构预测⁷，显示了结构稳定性和硬度的相互制约关系。

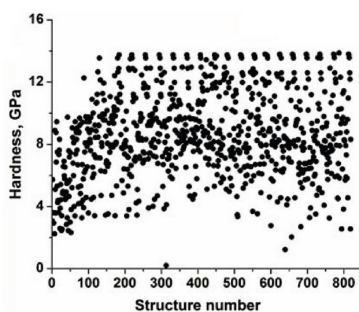


图 5: Predictions of the hardest structure of TiO_2 .

注意: 如果 optType=bandgap（带隙）或 diel_gap（电介质带隙），这里的带隙我们使用的是一个扩展函数，这个扩展函数对于金属表现出连续性。一也就是说， $\Delta E_g - g(E_F)/N$ ， ΔE_g 是带隙， $g(E_F)$ 是在费米能级处的态密度（对金属而言），而 N 是单胞内的原子数。由于这个扩展函数的连

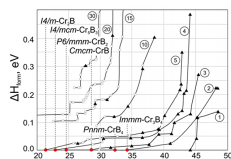


图 6: Pareto optimization of hardness and stability in the Cr-B system, showing several Pareto fronts⁷.

续性，与带隙相关量的全局最大化甚至可以用于解决金属的问题。对金属而言，这个值等于费米能级处的态密度。对于半导体和绝缘体而言这个值等于带隙。

▷ variable `atomType`

Meaning: 描述每种原子的类型。

Default: 无，必须明确给出

Format:

如果你喜欢用门捷列夫元素周期表的原子序数，请设置：

```
% atomType
12 14 8
% EndAtomType
```

或者你可以使用元素符号，如下：

```
% atomType
Mg Si O
% EndAtomType
```

再或者，你可以用元素名，设置：

```
% atomType
Magnesium Silicon Oxygen
% EndAtomType
```

▷ variable `numSpecies`

Meaning: 描述每种类型原子的数目。

Default: 无，必须明确给出

Format:

```
% numSpecies
4 4 12
% EndNumSpecies
```

这表示第一种类型有 4 个原子，第二种类型有 4 个原子，第三种类型有 12 个原子。

Notes: 对于变成分的计算，你需要列出如下所示的组成构建块：

```
% numSpecies
2 0 3
0 1 1
% EndNumSpecies
```

这表示第一个几何结构单元的分子式是 A_2C_3 ，第二个几何结构单元的分子式是 BC ， A 、 B 、 C 在原子类型块中已经解释过。所有结构都满足公式 $xA_2C_3 + yBC$ with $x, y = (0,1,2,\dots)$ — 或者是 $A_{2x}B_yC_{3x+y}$ 。如果你想做 A-B-C 体系中所有可能组成的预测，应该这样设置：

```
% numSpecies
1 0 0
0 1 0
0 0 1
% EndNumSpecies
```

你也可以用一个化学式的计量比数来做定成分的变成分计算。这种情况下，你需要设置 (`calculationType=301`)，化学组分，以及最大和最小原子个数。如下为 A_2BC_4 化合物的例子：

```
% numSpecies
2 1 4
% EndNumSpecies
```

14: minAt (单胞中最小原子个数) 28: maxAt (单胞中最大原子个数)

▷ *variable* `magRatio`

Meaning: 对非自旋极化 (NM), FM-LS、FM-HS、AFM-L、AFM-H、FM-LH 和 AF-LH 所对应的不同类型磁性质的结构的初始比例。仅 VASP 支持此功能

Default: 0.1, 0.9/4, 0.9/4, 0.9/4, 0.9/4, 0, 0

Format:

```
% magRatio
1/8 1/8 1/8 1/8 1/8 0 0
% EndMagRatio
```

这意味着产生 NM、FM-LS、FM-HS、AFM-L 和 AFM-H 结构的比率均为 20% (这里不是 $1/8$ ，已经重新调整为 1)。没有 FM-LH 和 AF-LH 磁化态的结构产生。

注意:

(1) 总的来说，磁化率 `magRatio` 可以大于 1，其比率将会自动调整为 1。(2) 在 USPEX 中的含义和初始磁化率的值：

- NM — 非自旋极化；
- FM-LS — 低自旋铁磁；
- FM-HS — 高自旋铁磁；
- AFM-L — 低自旋反铁磁；

- AFM-H — 高自旋反铁磁;
- FM-LH — 低/高自旋混合铁磁;
- AF-LH — 低/高自旋混合反铁磁。

(3) 对 HM (非自旋极化), 所有原子的初始磁化值均为 0。对自旋状态和高自旋状态原子的初始磁化值分别可以设置为 `MAGMOM = 1` 和 `4`。对低/高自旋混合状态, 每个原子的磁化值可以被随意地设置为 `MAGMOM = 1` 或 `4`。

(4) 当单胞中有奇数个原子时, 将不会产生 AFM 类型结构。

(5) 磁化率 `magRatio` 亦适用于在自旋突变操作中的突变值。

▷ variable `ldaU`

Meaning: 使用 DFT+U 方法, 指定每种原子的 Hubbard U 值。仅 VASP 支持此功能

Default: 0 对应每种类型的原子

Format:

```
% ldaU
4 0
% EndLdaU
```

▷ variable `ExternalPressure`

Meaning: 指定你所要计算体系的外部压力, 单位是 GPa。

Default: 0

Format:

```
100 : ExternalPressure
```

Note: 从 USPEX 9.4.1 起, 压力的值 (单位 GPa) 可以直接在 `INPUT.txt` 文件中设置。不需要在弛豫优化文件 `Specific/` 再次指定。

▷ variable `valences`

Meaning: 描述每类原子的化合价。这仅用于评估结合的硬度, 而结合硬度又用来计算近似的动力学矩阵 (对软模变异) 和晶体的硬度。

Default: USPEX 有一系列的系统默认化合价 (见附录 9.9)。然而, 需要注意的是, 对一些元素 (例如 N、S、W、Fe 和 Cr 等), 有许多可能的化合价。除非你要计算硬度, 否则这没有问题, 你可以用系统默认化合价。如果你要做硬度计算, 你必须明确给出化合价。

Format:

```
% valences
2 4 2
```

```
% EndValences
```

▷ *variable* `goodBonds`

Meaning: 在矩阵中指定两原子间最低的键价，大于这些键价的化学键被认为是重要的化学键。该参数的设置与原子距离矩阵 `IonDistances`（如下）一样，是一个上三角形式的方阵。`goodBonds` 仅用来计算硬度和软模变异。我们可以用公式

$$\text{goodBonds} = \frac{\text{valence}}{\text{max_coordination_number}}$$
 计算 `goodBonds`。

Default: USPEX 可以给出 `goodBonds` 合理的默认估值，你可以在输出文件 `OUTPUT.txt` 中得到这些数值。对大多数情况，用默认值已经足够，但是对于硬度的计算，你必须仔细地审查这些数值，或者手动设置。更多细节，请参考附录 9.10

Format:

```
% goodBonds
10.0 10.0 0.2
0.0 10.0 0.5
0.0 0.0 10.0
% EndGoodBonds
```

注意: 矩阵的维度必须与原子种类数或原子名称数相等。如果只有一种原子，那么矩阵就填充该数字。上面的矩阵解读如下：当可被认为是成键时，Mg-Mg 间的距离要足够短使其键价为 10 或者更大，Mg-Mg 键的距离必须足够短。对 Mg-Si 键，Si-Si 键和 O-O 键也有同样要求（通过使用这种专一的标准，我们有效地排除了来自软模变异和硬度计算的交互作用），然而在用于硬度和软模变异计算时 Mg-O 键的键价取为 0.2 或者更大，Si-O 键的键价取为 0.5 或者更大。

▷ *variable* `checkMolecules`

Meaning: 打开或关闭原始分子的再弛豫 (`MOL_1`、`MOL_2` 等) 是否完整。可用于分子晶体 (`calculationType=310, 311`)。

可能的数值（整数）：

- 0 — 未经核实，被认为是损坏了的结构或是合并的分子（我们强烈建议不使用这种类型）。
- 1 — 执行检查命令，所有损坏或是合并的分子均被遗弃。

Default: 1

Format:

```
1: 核实分子
```

▷ *variable* `checkConnectivity`

Meaning: 执行或不执行硬度计算以及软模变异。

可能的数值（整数）：

- 0 — 不检查原子关联性，不进行硬度计算。
- 1 — 考虑原子间的关联性，进行硬度计算。

Default: 默认值：

1: `checkConnectivity`

Format:

▷ *variable* `fitLimit`

Meaning: 对于最大值: 如果所获得 fitness 的最小值比 `fitLimit` 的值小, 那么 USPEX 的计算将会在第一代之后结束。对于最小值: 如果所获得 fitness 的最小值比 `fitLimit` 的值大, 那么 USPEX 的计算将会在第一代之后结束。

Default: 无默认值, 用户需要自己设定。

Format:

10 : `fitLimit`

4.2 种群

▷ *variable* `populationSize`

Meaning: 每一代中的结构数。如果需要的化, 第一代可以单独设置。

Default: 当 N 是晶胞中的原子数时 (或者是变成分结构中的 `maxAt` 最大原子数), $2 \times N$ 四舍五入最接近 10, 那么上限值是 60。通常可以依靠系统默认设置。

Format:

20 : `populationSize`

▷ *variable* `initialPopSize`

Meaning: 第一代中的结构数。

Default: 等于 `populationSize`。

Format:

20 : `initialPopSize`

Note: 在许多情况下，我们建议这两个参数值相等。有时（尤其是在变成分计算中）指定 `initialPopSize` 值大于 `initialPopSize` 值更为有用。设置一个很小的初始群也是可能的，当你想要完全从几个种子结构产生第一代时这样设置就变得很有用。

▷ *variable* `numGenerations`

Meaning: 最大模拟代数。当 `stopCrit` 的最优结构保持不变时，模拟可以较早地结束。

Default: 100

Format:

```
50 : numGenerations
```

▷ *variable* `stopCrit`

Meaning: 当停止标准代数中的最优结构不再变化，或者是代数已满的话—无论哪个先发生，模拟就会停止。

Default: 定成分运行的总原子数，变成分运行的最大原子数 `maxAt`。

Format:

```
20 : stopCrit
```

4.3 适者生存和选择

▷ *variable* `bestFrac`

Meaning: 目前这一代中用于产生下一代的比例。

Default: 0.7

Format:

```
0.7 : bestFrac
```

注意: 这是一个非常重要的参数，值在 0.5–0.8 之间比较合理。

▷ *variable* `keepBestHM`

Meaning: 定义有多少个最优结构将会在下一代中存活。

Default: $0.15 \times \text{populationSize}$

Format:

```
3 : keepBestHM
```

▷ variable **reoptOld**

Meaning: 定义存活的结构是否需要优化. 如果 `reoptOld=0`, 这些结构不会在下一代中重新优化; 如果 `reoptOld=1`, 这些结构将会在下一代中重新优化. 通常设定 `reoptOld=0` 比较合理

Default: 0

Format:

```
1 : reoptOld
```

4.4 结构产生和变异操作

▷ variable **symmetries**

Meaning: 晶体可能的空间群, 二维晶体/表面的平面群组, 或者是团簇的点群. 在该列表中用随机选择的对称群, 用随机生成的晶格常数和原子坐标产生确定结构数. 在这个过程中, 从普通位置上将生成特殊的威科夫 (Wyckoff) 位置 (Fig. 7)。

Default:

- 对三维晶体: 2-230
- 对二维晶体/表面: 2-17
- 团簇: E C2 D2 C4 C3 C6 T S2 Ch1 Cv2 S4 S6 Ch3 Th Ch2 Dh2 Ch4 D3 Ch6 O D4 Cv3 D6 Td Cv4 Dd3 Cv6 Oh Dd2 Dh3 Dh4 Dh6 Oh C5 S5 S10 Cv5 Ch5 D5 Dd5 Dh5 I Ih

Format:

```
% symmetries
195-198 200 215-230
% EndSymmetries
```

▷ variable **fracGene**

Meaning: 通过遗传所产生的结构所占的百分比; 0.1 means 10%, etc.

Default: 0.5

Format:

```
0.5 : fracGene
```

▷ variable **fracRand**

Meaning: 通过空间群随机产生的结构所占的百分比。

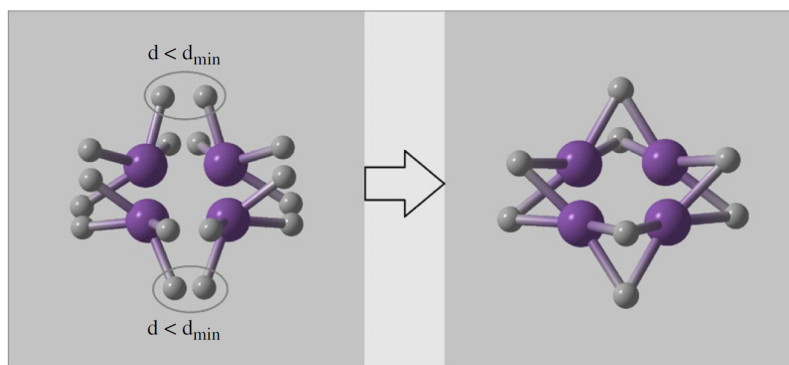


图 7: Example of merging atoms onto special Wyckoff positions (from Ref. ¹²).

Default: 0.2

Format:

```
0.20 : fracRand
```

▷ variable `fracTopRand`

Meaning: 通过拓扑随机器产生的结构所占的百分比。

Default: 0.2

Format:

```
0.20 : fracTopRand
```

▷ variable `fracPerm`

Meaning: 通过置换所产生的结构所占的百分比；0.1 means 10%, etc.

Default: 0.1 如果有超过一种类型的原子/分子；0 其他情况。

Format:

```
0.1 : fracPerm
```

▷ variable `fracAtomsMut`

Meaning: 指定通过软模变异或交换所产生的机构所占的百分比。

Default: 0.1

Format:

```
0.1 : fracAtomsMut
```

注意: 你可以通过指定 `softMutTill` 用软模变异或协同变异。

▷ variable `fracRotMut`

Meaning: 通过分子取向变异所产生的结构所占的百分比。0.1 代表 10%，等等；0.1 means 10%, etc.

Default: 0.1 对于分子晶体； 0 其他情况

Format:

0.1 : `fracRotMut`

▷ variable `fracLatMut`

Meaning: 由晶格变异所获得的结构的百分比。0.1 代表 10%，等等。；0.1 means 10%, etc.

Default: 0 固定晶胞预测； 0.1 其他情况

Format:

0.1 : `fracLatMut`

注意: 如果所有分数的总和 (`fracGene + fracRand + fracPerm + ...`) 不等于 1, 它们将会被重新调节。

▷ variable `fracSpinMut`

Meaning: 通过由自旋突变所产生的结构所占的百分比，0.1 代表 10%，等等。

Default: 0.1

Format:

0.2 : `fracSpin`

注意:

(1) 当运行一个自旋状态突变时，将会产生不同于初始磁性状态的新状态。例如，在突变后，FM-L 态绝对不会仍处在 FM-L 态，而可能会停留在 NM 和 FM-H 状态等等。对 AFM 状态，在突变后可能仍处在 AFM 态，但是可能会伴随有原子的自旋向上或自旋向下的调整。(2) 新的状态突变的比率由 `magRatio` (磁性比率) 决定。

▷ variable `howManySwaps`

Meaning: 对置换来说，互换原子对的数量由在 1 和 `howManySwaps` 之间的统一分配所随机决定。

Default: $0.5 \times (\text{可能的互换原子对的最大值})$. 如果原子 N_a 和 N_b , 原子 N_c 和 N_d 发生了互换, 那么可能的互换原子对的总数是 $\min(N_a, N_b) + \min(N_c, N_d)$, 并且 `howManySwaps` 的默认设置为 $0.5 \times [\min(N_a, N_b) + \min(N_c, N_d)]$. 在许多情况下, 采用默认值是一个不错的选择。

Format:

5 : `howManySwaps`

▷ variable `specificSwaps`

Meaning: 明确在置换中允许哪种原子类型发生互换。

Default: 空行，这意味着没有明确的互换，所有的原子都发生了置换。

Format:

```
% specificSwaps
1 2
% EndSpecific
```

注意: 在这种情况下，第一种类型的原子可以与第二种类型的原子发生互换。如果你想要尝试所有可能的互换，在该关键块里留一空行，或者删除这一部分。

▷ *variable* `mutationDegree`

Meaning: 在软模变异的位移的最大值的单位是 Å. 设置了软模变异和协同变异的最大位移矢量使其最大位移等于 `mutationDegree`。

Default: $3 \times$ (平均原子半径)

Format:

```
2.5 : mutationDegree
```

▷ *variable* `mutationRate`

Meaning: 晶格变异的应变矩阵的标准偏差的极小值。应变矩阵部分从高斯分布中随机选择，并且只允许取-1 到 1 之间的值。晶格变异的本质上是把准动力学的思想融入到了我们的方法中^{22?}，新的结构是由一些异质结构的晶胞扭曲产生的。不像在准动力学中，在我们的方法中，变形是不累积的，所以弹性部分可以足够大去获得新结构。

Default: 0.5

Format:

```
0.5 : mutationRate
```

晶格变异与较弱的软模变异结合在一起时一个好选择:

▷ *variable* `DisplaceInLatmutation`

Meaning: 将软模变异指定为晶格变异的一部分，并且设置最大位移的量级为 Å.

Default: 1.0

Format:

```
1.0 : DisplaceInLatmutation
```

▷ variable `AutoFrac`

Meaning: 程序根据计算的演变，自动分配对变量算符所占的百分比（参数控制）。这使得计算速度加快了两倍左右。当要选择用户自定义变异操作时，设置 `AutoFrac=0`。

Default: 0

Format:

1 : `AutoFrac`

4.5 约束条件

相同的结构可以再坐标系中以无限的方式展现出来（“模数不变性”）。这些大量相同的选择将会导致非常扁平的晶胞单元，从而为结构弛豫和能量计算带来问题（例如需要很大数量的 k 点）。约束条件在结晶学中已经熟知，那就是晶胞角度在 60° 到 120° 之间，不会移出任何一个多余的和有问题的晶胞（例如：晶胞中允许的 $\alpha = \beta = \gamma \sim 120^\circ$ 实际上是扁平的）。因此我们发明了一种特殊的方法：用最短的晶格矢量来获得特殊的晶胞形状^{23?}。如果至少有一个晶格矢量的投影在其他的晶格矢量上或者在对立面上的斜的晶格矢量的长度大于（通过模比较）该晶格矢量长度的一半，就会发生变形。也就是说，对于 \mathbf{a} 和 \mathbf{b} 电子对，或者是 \mathbf{c} 和 $(\mathbf{a} + \mathbf{b})$ 电子对，有以下标准：

$$\left| \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{b}|} \right| > \frac{|\mathbf{b}|}{2} \quad (2)$$

$$\left| \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|} \right| > \frac{|\mathbf{a}|}{2} \quad (3)$$

$$\left| \frac{\mathbf{c} \cdot (\mathbf{a} + \mathbf{b})}{|\mathbf{c}|} \right| > \frac{|\mathbf{c}|}{2} \quad (4)$$

$$\left| \frac{\mathbf{c} \cdot (\mathbf{a} + \mathbf{b})}{|\mathbf{a} + \mathbf{b}|} \right| > \frac{|\mathbf{a} + \mathbf{b}|}{2} \quad (5)$$

例如，在标准2中，新的矢量 \mathbf{a}^* 值等于：

$$\mathbf{a}^* = \mathbf{a} - \text{ceil} \left(\frac{|\mathbf{a} \cdot \mathbf{b}|}{|\mathbf{b}|^2} \right) \text{sign}(\mathbf{a} \cdot \mathbf{b})\mathbf{b} \quad (6)$$

这种变形迭代进行，完全避免了劣质的晶胞形状，从而解决了问题。在该变形中，原子坐标部分发生了改变，以保证初始结构和改变后的结构是完全相同的（在变形过程中，原子的笛卡尔坐标保持不变）。

▷ variable `minVectorLength`

Meaning: 设置最新产生的结构的晶胞参数的最小长度。

Default: $1.8 \times$ 最大原子的共价直径。对于分子晶体 (*calculationType* = 310, 311) 默认最大值是 $1.8 \times \max(\text{MolCenters})$ 。

Format:

```
2.0 : minVectorLength
```

当原子间距离足够小时，通常使用的计算方法（如赝势，PAW，LAPW 和其他参数化的力场）将不会起作用。这种情况需要避免，你可以用上三角形式的离子距离方阵 *IonDistances* 指定每对原子之间的最小距离：

▷ *variable* [IonDistances](#)

Meaning: 在不同的原子类型之间设置最小原子距离矩阵。小于离子距离的数值在物理意义上是没有意义的，需要严格避免。

Default: A 和 B 原子之间的离子距离被估计为 $0.22 \times (V_A^{1/3} + V_B^{1/3})$ ，但是不会超过 1.2 \AA ，在分子类晶体计算中是 $0.45 \times (V_A^{1/3} + V_B^{1/3})$ ，此处 V_A 和 V_B 是 USPEX 中所估计得到的 A 原子和 B 原子的默认体积值。

Format:

```
% IonDistances
1.0 1.0 0.8
0.0 1.0 0.8
0.0 0.0 1.0
% EndDistances
```

注意：矩阵的大小必须等于原子种类数。如果上面例子中的化合物是 MgSiO_3 ，那么矩阵应该按如下解释：在一个新产生的结构中允许的 Mg-Mg 距离为 1.0 \AA ，Mg-Si 距离、Si-Si 距离和 O-O 距离也是 1.0 \AA ，而最小的 Mg-O 距离和 Si-O 距离是 0.8 \AA 。你可以把特定体系的更多信息放进这个键长矩阵，举例来说，如果你知道 Mg 原子倾向离得很远，在你的体系中最近会小于 3 \AA ，你可以列入该信息。然而需要注意的是最小距离越大，计算产生符合这些约束条件的结构就越困难（特别是大体系）。所以尽量折中，并且谨记离子距离必须远远小于实际的键长。

▷ *variable* [constraint_enhancement](#)

Meaning: 可以用 *IonDistances* 矩阵（通过变为 *constraint_enhancement* 的倍数）来严格约束计算产生的对称随机结构（适用于所有变异操作，未增强的离子距离 *IonDistances* 矩阵依然可以应用）。允许你应用离子距离矩阵中严格的约束条件（通过乘以增强约束条件）。确保你在知道自己正在用的是什么意义的时候使用。

Default: 1

Format:

```
1 : constraint_enhancement
```

对分子晶体，下面的关键块体非常重要：

▷ *variable* **MolCenters**

Meaning: 分子中心之间的最小距离矩阵。任何小于那些明确给出的分子的大重叠的距离，在物理意义上都是不合理的，需要严格避免。

Default: 零矩阵适用于非分子计算，分子晶体没有默认（必须明确给出）。

Format:

```
% MolCenters
5.5 7.7
0.0 9.7
% EndMol
```

注意：在上面的例子中，有两种分子类型。在所有产生的结构中，第一种类型的分子的几何中心之间的距离必须至少为 5.5 Å（A-A 距离），第一种类型和第二种类型分子几何中心之间的距离— 7.7 Å（A-B 距离），第二种分子几何中心之间的距离— 9.7 Å（B-B 距离）。

4.6 晶胞

在指定条件下，我们给定合适的晶胞体积大小有利于去产生新晶体结构（弛豫前）。这个给定的初始体积可以用 **Latticevalues** 输入：

▷ *variable* **Latticevalues**

Meaning: 明确给出单胞或抑制晶格参数的初始体积。

Default: 对于晶胞体积不需要明确给出值—USPEX 有一个强大的计算体系可以给出任何压力下的合理的估计值。

Format:

```
% Latticevalues
125.00
% Endvalues
```

Notes: (1) 该体积仅用于初始的猜测，并且只影响第一代，每个结构都经过充分优化，且采用与最低（自由）能相对应的体积。这个关键部分还有其他用途：当你知道晶格参数时（例如从实验中），你就可以在晶格参数 **Latticevalues** 中使用三元矩阵（计算类型 300/310）和二元矩阵（计算类型-200）明确指定，而不是使用单胞体积，例如：


```
% Latticevalues
7.49 0.0 0.0
0.0 9.71 0.0
0.0 0.0 7.07
% Endvalues
```

你也可以通过列出 a 、 b 、 c 、 α 、 β 和 γ 值来指定晶胞参数：

```
% Latticevalues
10.1 8.4 12.5 90.0 101.3 90.0
% Endvalues
```

Attention: if you do a calculation with a fixed monoclinic cell, please use setting with special angle β (standard setting).

对于二维晶体 ($\text{calculationType} = -200$)，你只需要列出参数 a 、 b 和 γ 即可：

```
% Latticevalues
10.1 8.4 90.0
% Endvalues
```

注意：如果你要做定成分的单斜晶胞的计算，则要做专门的 β 角度设置（标准设置）。

(2) 对于变成分计算，需要明确给出计算搜索空间最后一个的体积。例如：

```
% Latticevalues
12.5 14.0 11.0
% Endvalues
```

(3) `Latticevalues` 参数不需要用户必须去设定，在 USPEX 中我们用了一个专门算法去较精确的估计它的初始值。（适用于所有需要输入初始晶胞体积的计算类型 `calculationType: 3**`, 2D-crystals, 110, 000, 你也可以用在线程序 http://uspex-team.org/online_utilities/volume_estimation 去估算晶胞体积。用户也可以手动输入晶胞体积。

(4) 如果你研究压力下的分子晶体，有时你可能需要稍微增加初始晶胞体积，为了能够用随机对称算子产生结构。

4.7 重启作业

如果某些作业运行出错，你想要在它停止的点重新开始计算—或者是更早的点。如果你想做的就是从停止的那一点处重新开始运行，你不需要改变任何设置（所有信息都会被存储在 `*.mat` 文件中），删掉 `still_reading` 文件，再次运行 USPEX 就足够了。

如果你想要在一个特殊结果 `results` 文件夹中的特殊的一代重新开始，那么给出 `pickUpGen =` 你想要重新开始的代数。`pickUpFolder =` number of `results` 文件夹的编号（例如 `results1` 为 1, `results2`, ...）也是从你想要重新开始的那里开始。如果 `pickUpGen=0` 就是要重新开始计算。三个

参数的默认值都为 0。例如，想要从第 10 代开始在 *results5* 文件夹中重新启动运算，需设置：

```
10 : pickUpGen
5 : pickUpFolder
```

4.8 详述 从头算法

USPEX 采用强大的两级并行框架，是并行计算程序中的佼佼者。并行第一级是在结构弛豫代码里执行，并行第二级是把计算分配给了同群中的个体（因为同一种群内的结构之间是彼此独立的）。

首先，你需要指定你想使用哪个代码去进行结构弛豫和目标函数计算：

▷ *variable* `abinitioCode`

Meaning: 指定每一步结构弛豫的代码类型

Default: 1 for every optimization step (VASP)

Format:

```
% abinitioCode
3 2 2 1 1
% ENDabinit
```

Alternative Format:

```
% abinitioCode
1 1 1 1 1 (14)
% ENDabinit
```

这里的区别在于第二种方式中有一步或者几步的计算被包含在括号中。这对于优化物理性质非常有用：对于包含在括号中的计算，结构会被对称化并被表示为标准晶格设置形式。这样在计算中就可以充分利用晶格对称性从而使得性质的计算更加容易和稳定。在对称化和性质计算之前，结构需要被充分弛豫。

Note 1: 焴取自括号前的最后一步。

Note 2: 数字表示每一步的结构弛豫用到的外部程序：

1 — VASP	5 — ORCA
2 — SIESTA	6 — DMACRYS
3 — GULP	7 — CP2K
4 — LAMMPS	8 — Quantum Espresso

9 — FHI-aims	15 — DFTB
10 — ATK	16 — Gaussian
11 — CASTEP	(only for clusters)
12 — Tinker	18 — Abinit
13 — MOPAC	19 — CRYSTAL
14 — BoltzTraP	

▷ variable **KresolStart**

Meaning: 给出 k 点网格倒易空间的分辨率 (单位: $2\pi\text{\AA}^{-1}$).

Default: 线性的从 0.2 到 0.08

Format:

```
% KresolStart
0.2 0.16 0.12 0.08
% Kresolend
```

注意: 你可以输入几个数值 (每一步结构弛豫对应一个数), 开始时粗略些 (也就是值大一些), 到最后, 精确度高, 这样就明显地加快了计算速度。特别是对金属而言, 因为需要很多 k 点。如果你想用 VASP 或 QuantumEspresso, 那么这个模块非常重要 (GULP 完全没必要用它, 而对于 SIESTA, 你必须在 SIESTA 输入文件里定义 **KresolStart**)。

对于团簇, 二维晶体和表面, 你必须明确给定团簇 (或表面) 真空范围的厚度:

▷ variable **vacuumSize**

Meaning: 定义在结构周围增加的空间的数量 (在毗邻的单胞内邻近团簇之间的最近邻距离, \AA)。仅适用于表面, 二维晶体和纳米微粒。

Default: 10 每一步弛豫设为 10\AA

Format:

```
% vacuumSize
10 10 15 20 20
% EndVacuumSize
```

▷ variable **numParallelCalcs**

Meaning: 指出在并行计算中结构弛豫的数目。

Default: 1

Format:

```
10 : numParallelCalcs
```

你需要提供提交作业文件或者你正在使用的准则/模式下的可执行文件的名字。

▷ *variable* `commandExecutable`

Meaning: 明确给定提交作业文件名或在给定准则下的可执行文件名。

Default: 无默认值，需要用户自己设定。

Format:

```
% commandExecutable
gulp < input > output
mpirun -np 8 vasp > out
mpirun -np 8 vasp > out
mpirun -np 8 vasp > out
% EndExecutable
```

注意: 每一行都对应于一个弛豫阶段——第一行描述了第一个弛豫阶段的执行，等等。例如：从头算准则等于“3 1 1”意味着第一个弛豫阶段是执行 GULP，接下来的阶段会通过 VASP 用命令“`mpirun-np 8 vasp > out`”用 VASP 命令执行。如果在可执行命令中只显示出一行，那么所有的弛豫阶段都执行相同的命令。

实际上在你远程提交作业模式下几乎所有平台都可以使用 USPEX，你需要的是在你的工作站有 MATLAB/Octave 运行。在这种情况下，你的工作站准备开始输入（包括 jobs），然后提交到远程计算节点上，计算完成时查看结果并分析，准备新的输入文件。传出和来往的数据不是很大，所以网络不需要非常快。当然，作业提交依赖于机器。

▷ *variable* `whichCluster`

Meaning: 明确给定作业提交的类型。

Possible values (integer):

- 0 — 无作业脚本；
- 1 — 本地提交；
- 2 — 远程提交。

Default: 0

Format:

```
1 : whichCluster
```

▷ variable `remoteFolder`

Meaning: 计算即将进行的远程超级计算机上的文件夹。只有当 `whichCluster=2` 时, 这个文件夹才会被激活。

Default: 没有

Format:

```
Blind_test : remoteFolder
```

注意: 在远程提交文件夹中有一个相似的参数设置 — `homeFolder`, 计算的真实路径实际上是 `*homeFolder*/*remoteFolder*/CalcFolder*`。在此处, `x=1, 2, 3,...`。

▷ variable `PhaseDiagram`

Meaning: 使计算类型 =300 和 301 的相图的计算能够进行。这给出了一个想法 (比较粗劣, 仅仅是一个大致的结果): 给出新相可能会在较高或较低压力下, 同时粗劣的给出一个转变压力。

Default: 0

Format:

```
1 : PhaseDiagram
```

4.9 指纹设置

关于指纹函数概念和方法, 请阅读 (Oganov & Valle, 2009¹⁶)。

▷ variable `RmaxFing`

Meaning: 截断距离 (单位 Å)。

Default: 10.0

Format:

```
10.0 : RmaxFing
```

▷ variable `deltaFing`

Meaning: 是指纹函数的离散化 (单位 Å)。

Default: 0.08

Format:

0.10 : `deltaFing`

▷ *variable* `sigmaFing`

Meaning: 是原子间距的高斯扩展。

Default: 0.03

Format:

0.05 : `sigmaFing`

`toleranceFing` (默认值为 0.003) 和 `toleranceBestHM` (默认值为 0.02) 指定了结构之间的最小余弦距离, 证明它们是不相同的——分别用于产生子代和适者生存。它们依赖于结构弛豫的精度和体系的物理性质 (例如对于排序问题, 分别属于不同结构的指纹非常相似, 并且参数公差也设得非常小) 只有当 `dynamicalBestHM=1` 时 (我们已不用了), `toleranceBestHM` 才是有用的, 所以这个参数基本上被废弃。

4.10 反种子设置

反种子技术已经在 USPEX 中开发实施了, 所有基于对结构不利的想法都已被屏蔽以保证模拟没有陷入局部最小值。在这儿, 和时间有关的适应度是所有总的真实焓值 (或其他的适应性质) 和跟历史有关的项, 它是能量抽样面上叠加的高斯势能之和:

$$f = f_0 + \sum_a W_a \exp\left(-\frac{d_{ia}^2}{2\sigma_a^2}\right),$$

在这里, f 是适应度 (f_0 — 真实适应度, f — 跟历史有关的适应度), W_a 是高度, 是高斯展宽。在我们的方法中, 高斯参数的变化依赖于种群多样性, 能量在每一代中扩展。

有三种使用该技术的方法。第一种, 将你认为不利的结构放置在反种子文件夹 `AntiSeeds` 中。例如它可能是基态结构—在这种情况下, USPEX 将会尽力找到第二个最低焓值的结构。

在第二种和第三种方法中, 你不需要明确给定反种子结构—USPEX 要么用所有的抽样结构作为反种子 (经过很好的测试, 是被推荐的方法), 要么是每一代中的最优结构。你需要给定以下的一些设置:

▷ *variable* `antiSeedsActivation`

Meaning: 明确从哪一代开始反种子模式开始。当 `antiSeedsActivation = N > 0` 时, 从第 N 代开始, 高斯势加入到所有的结构中, 当 $N < 0$ 时, 高斯势仅加入到每一代的最优结构中, 从第 N 代开始。当 $N = 0$ 时, 高斯势仅加入到放在反种子文件夹中的结构中。如果你不想用反种子, 那么给定非常大的 `antiSeedsActivation` 值 (例如:5000) 和 `antiSeedsMax=0.0`。

Default: 5000

Format:

```
1 : antiSeedsActivation
```

▷ *variable* `antiSeedsMax`

Meaning: 明确高斯的高度, 以每一代中的焓值的平均方差为单位 (算出来的值仅在 `bestFrac` 之间, 例如在准双亲之间)。我们推荐 `antiSeedsMax=0.01`。

Default: 0.000

Format:

```
0.005 : antiSeedsMax
```

▷ *variable* `antiSeedsSigma`

Meaning: 明确高斯的宽度, 以每一代中结构之间的平均距离为单位 (算出来的值仅在 `bestFrac` 之间, 例如在准双亲之间), 我们建议 `antiSeedsSigma=0.005`。

Default: 0.001

Format:

```
0.005 : antiSeedsSigma
```

Fig. 8 是一个应用反种子技术的例子。

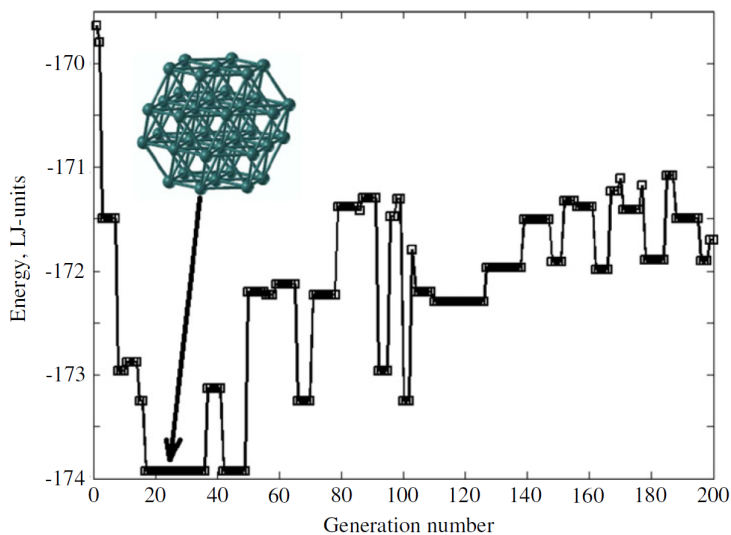


图 8: 应用反种子的有 38 个原子的 Lennard-Jones 集群计算的例子。每一代中的最优结构的能量已被标绘出。我们可以清晰地看到, 算法没有陷在任何一个候选最小值很长时间, 并且很快发现了基态。在这儿我们用 `antiSeedsActivation=1`, `antiSeedsMax=0.01`, `antiSeedsSigma=0.001`。

4.11 空间群的判定

▷ variable `doSpaceGroup`

Meaning: 判定空间群，还以结晶学的 *.CIF 格式（当你要准备发表的时候这会更简单一点，但是需要注意的是：如果弛豫不是非常精确，或者是如果对于对称探测器设置了非常严格的公差，那么有时候空间群可能是不确定的）写在输出文件中。多亏了由 H.T. Stokes 提供的强大的对称准则，这个选项才得以被使用。

Default: 1, if `calculationType=3**` (300, 301, 310,311 — bulk crystals) and 0 otherwise.

Format:

1 : `doSpaceGroup` (0 - 没有空间群, 1 - 判定空间群)

▷ variable `SymTolerance`

Meaning: 用 H.T. Stokes 的寻找对称性代码精确地确定对称性。可以设置为一个数字（单位为 Å），或者是 `high | medium | low` (= 0.04 | 0.08 | 0.15)。

Default: `medium`

Format:

`medium` : `SymTolerance`

4.12 给开发者的关键参数

▷ variable `repeatForStatistics`

Meaning: USPEX 自动运行的数量。USPEX 模拟是随机的，用同样的输入参数再次模拟，得到的结果是必然不同的。然而最终的结果—基态值—是相同的（也是我们所期望的）。为了达到它所运行的步数和化学空间中的轨道数在每次运行时也是不同的。为了比较不同的算法，你必须收集一些统计资料—不要只依赖于单次运行（它可能是好运的，也可能不是好运的...USPEX 不能只依赖于好运）。这个选项仅是开发者感兴趣，而且仅对用简单的势函数（如 GULP）收集统计资料有意义。

Default: 1 (*i.e.*, 也就是说，没有统计资料将会被收集)

Format:

20 : `repeatForStatistics`

▷ variable `stopFitness`

Meaning: 设置适应度的值就可以当达到适应度值 \leq `stopFitness` 时，计算停止。

Default: 没有默认值，需要用户自己设定。

Format:

90.912 : stopFitness

注意: 当 stopFitness 值给定之后，统计资料才能够自动分析。为了收集 stopFitness 的可获取的统计资料建议 repeatForStatistics 模块部分设置的值 >。输出样本如下:

```
Number of files to be processed: 20
Target enthalpy: 90.912

Generation: 23  Number: 1326  Enthalpy: 90.9119  Mat-file: /home/USPEX/01/results1/USPEX.mat
Generation: 22  Number: 1224  Enthalpy: 90.9119  Mat-file: /home/USPEX/02/results1/USPEX.mat
Generation: 60  Number: 3451  Enthalpy: 90.9119  Mat-file: /home/USPEX/03/results1/USPEX.mat
Generation: 30  Number: 1739  Enthalpy: 90.9119  Mat-file: /home/USPEX/04/results1/USPEX.mat
Generation: 17  Number: 956   Enthalpy: 90.9119  Mat-file: /home/USPEX/05/results1/USPEX.mat
Generation: 36  Number: 2055  Enthalpy: 90.9119  Mat-file: /home/USPEX/06/results1/USPEX.mat
Generation: 35  Number: 1987  Enthalpy: 90.9119  Mat-file: /home/USPEX/07/results1/USPEX.mat
Generation: 22  Number: 1241  Enthalpy: 90.9119  Mat-file: /home/USPEX/08/results1/USPEX.mat
Generation: 18  Number: 1002  Enthalpy: 90.9119  Mat-file: /home/USPEX/09/results1/USPEX.mat
Generation: 29  Number: 1641  Enthalpy: 90.9119  Mat-file: /home/USPEX/10/results1/USPEX.mat
Generation: 21  Number: 1197  Enthalpy: 90.9119  Mat-file: /home/USPEX/11/results1/USPEX.mat
Generation: 27  Number: 1542  Enthalpy: 90.9119  Mat-file: /home/USPEX/12/results1/USPEX.mat
Generation: 44  Number: 2519  Enthalpy: 90.9119  Mat-file: /home/USPEX/13/results1/USPEX.mat
Generation: 32  Number: 1821  Enthalpy: 90.9119  Mat-file: /home/USPEX/14/results1/USPEX.mat
Generation: 15  Number: 835   Enthalpy: 90.9119  Mat-file: /home/USPEX/15/results1/USPEX.mat
Generation: 43  Number: 2477  Enthalpy: 90.9119  Mat-file: /home/USPEX/16/results1/USPEX.mat
Generation: 40  Number: 2278  Enthalpy: 90.9119  Mat-file: /home/USPEX/17/results1/USPEX.mat
Generation: 24  Number: 1358  Enthalpy: 90.9119  Mat-file: /home/USPEX/18/results1/USPEX.mat
Generation: 14  Number: 757   Enthalpy: 90.9119  Mat-file: /home/USPEX/19/results1/USPEX.mat
Generation: 27  Number: 1532  Enthalpy: 90.9119  Mat-file: /home/USPEX/20/results1/USPEX.mat

Found structures numbers : 1326 1224 3451 1739 956 2055 1987 1241 1002 1641 1197 1542 2519 1821 835 2477 2278 1358 757 1532
Found generations numbers: 23 22 60 30 17 36 35 22 18 29 21 27 44 32 15 43 40 24 14 27

Success rate: 100 percent
Average number of generations to get E=90.912: 29
Average number of structures to get E=90.912: 1647
Standard deviation: 670
```

▷ variable `fixRndSeed`

Meaning: USPEX 计算的随机种子数。对于相同的随机种子，USPEX 将会产生相同的结果。

Default: 0

Format:

-2000 : fixRndSeed

▷ variable `collectForces`

Meaning: 收集 USPEX 计算产生的所有弛豫优化的信息，包括在结构优化中的原子上的力，原子位置，晶格常数和应力张量。这个信息存储在 `FORCE.mat` 文件中。仅 VASP 支持此功能。

Default: 0

Format:

```
1 : collectForces
```

4.13 不常使用的关键参数

▷ *variable* `mutationRate`

Meaning: 晶格变异应变矩阵组分的标准偏差。应变矩阵组分是从高斯分布中随机选择的，而且只允许取-1 到 1 之间的值。晶格变异基本上是将 metadynamics 的思路融入到了我们的方法当中^{22?}，即是通过建立一些晶胞的变形来发现新的结构类型。与 metadynamics 不同的是，晶胞变形在我们的方法中不会累计，因此应变分量能确保足够大以获得新晶体结构。

Default: 0.5

Format:

```
0.5 : mutationRate
```

最好把晶格变异和小的软膜变异相结合:

▷ *variable* `mutationDegree`

Meaning: 软膜变异中的最大位移距离的单位是埃。mutationDegree 用于设定软膜或晶格变异的具体位移向量值。

Default: 3×(平均原子半径)

Format:

```
2.5 : mutationDegree
```

▷ *variable* `ordering_active`

Meaning: 通过局部有序参数打开有偏见的变量算符。

Default: 1

Format:

```
1 : ordering_active
```

▷ *variable* `symmetrize`

Meaning: 把所有结构转换为可以适应对称性的结晶学设置。

Default: 0

Format:

```
1 : symmetrize
```

▷ variable `valenceElectr`

Meaning: 每种原子类型的价电子数。

Default: 所有原子的值都是常量，我们已经将它们制成了表，不需要特别设置。

Format:

```
% valenceElectr
2 6
% EndValenceElectr
```

▷ variable `percSliceShift`

Meaning: 在所有维度内进行块移动（在遗传中用到）的概率, 1.0 代表 100%.

Default: 1.0

Format:

```
0.5 : percSliceShift
```

▷ variable `maxDistHeredity`

Meaning: 给定参与遗传的结构之间的最大余弦距离。它指定了与该结构能够匹配的能量面半径。

Default: 0.5

Format:

```
0.5 : maxDistHeredity
```

▷ variable `manyParents`

Meaning: 明确是否应该要超过两片（或超过两个亲代结构）被用于遗传。这对大体系来说是有用的。

可能的值 (整数):

0 — 仅用两个亲代，每个一个片段。

1 — 许多结构被用作亲代，每个一个片段。

2 — 两个结构被用于亲代，许多片段（用参数 `minSlice` 和 `maxSlice` 动态决定）可以从每个亲代中独立选择。

3 — 两个结构被用于亲代，许多片段（用参数 `minSlice` 和 `maxSlice` 不断变化决定）是从有特定设置的晶胞中切下来。这是适用于大体系的较好的选择。例如：我们将两个结构切成近似相等厚度

的薄片，然后从亲代 1 中选择偶数片，从亲代 2 中选择奇数片，制成多层的“三明”或是“斑马”。

Default: 0

Format:

3 : manyParents

minSlice, maxSlice: 决定了切片的最小和最大厚度，单位为 Å，他们将会从亲代中删掉来参加子代结构的产生。我们期望这些薄片足够薄，能够携带一些亲代的信息（但是不是太薄，以至于对遗传没有影响）。这些参数的合适值分别大约是 1 和 6 Å。

对于团簇，你可以直接给定参与遗传的亲代数（但是我们发现这几乎没有用）：

▷ *variable* **numberparents**

Meaning: 定义了团簇中遗传的亲代数。

Default: 2

Format:

2 : numberparents

5 特例的额外输入设置

5.1 变成分代码

切换为变成分模式，必须：

1. 设置 301、311 或 201 : 计算类型。
2. 在 `numSpecies` 下设置结构组成 (参照 `numSpecies` 的说明书)。
3. 用 `Latticevalues` 设置每种类型原子 (或每个组成块) 粗略的原子体积。
4. 设置下面的 *varcomp-only* 选项:

▷ variable `firstGeneMax`

Meaning: 第一代抽取多少不同的组分。如果是 0, 这个数值就等于 `initialPopSize/4`。对于二元的, 要求 `firstGeneMax=11`, 对于三元, 需要更高的值, 比如 30。

Default: 11

Format:

```
10 : firstGeneMax
```

▷ variable `minAt`

Meaning: 第一代单胞的最小原子数 (对于 `calculationType=301/201/300`) 或分子数 (对于 `calculationType=311`)。

Default: 无默认值

Format:

```
10 : minAt
```

▷ variable `maxAt`

Meaning: 第一代单胞的最大原子数 (对于 `calculationType=301/201/300` 或 META 计算) 或分子数 (对于 `calculationType=311`)。

Default: No default

Format:

```
20 : maxAt
```

▷ variable `fracTrans`

Meaning: 通过原子替换获得的结构所占的百分比。这个过程中，一个随机选取的原子被系统中其他化学成分原子取代—新的化学组成由默认随机选取，或者你可以在 `specificTrans` 下定义它，就像明确的置换。

Default: 0.1

Format:

```
0.1 : fracTrans
```

▷ *variable* `howManyTrans`

Meaning: 结构中被替换原子的最大百分比 (0.1 = 10%)。要替换的原子的百分比从 0 到 `howManyTrans` 随机选取。

Default: 0.2

Format:

```
0.2 : howManyTrans
```

▷ *variable* `specificTrans`

Meaning: 允许替换的种类。

Default: 空行 (没有特定的替换)

Format:

```
% specificTrans
1 2
% EndTransSpecific
```

注意: 在这种情况下，原子 1 可以被转变成原子 2，反之原子 2 也可以被转变成原子 1。如果你想尝试所有的变形可能，在这一关键块下留下空行。

变成分案例在运行时，参数 `keepBestHM` 有新含义—凸包 (比如，多成分系统的热力学稳定状态) 图上的所有结构保留，连同 `convex hull` 附近的一些亚稳结构—总量是 `keepBestHM`。

对于变成分的运行，明智地设置第一代尤其重要。选择适当大的初始代的大小 `initialPopSize`。为不同组成的 `firstGeneMax` 选取合理大的数值作为第一代的样品。最后，`minAt` 和 `maxAt` 的差异不超过 2 倍，而且不同的体系大小你可能需要少量的计算：比如，4-8, 8-16, 16-30 原子，等等。

给 VASP 用户一个附加意见—如果你想完成一个变成分的运行，拿 Na-Cl 体系来说，你需要确定 `INPUT.txt` 中给出了正确的原子类型，并且将赝势文件 `POTCAR_Na` 和 `POTCAR_Cl` 放入文件夹 `~/StructurePrediction/Specific/`。USPEX 将识别每个原子并且适当选取每个原子的 `POTCAR` 文件进行计算。图 Fig. 9 展示了 USPEX 发现的且被实验证实²⁴ 的热力学稳定的氯化钠。

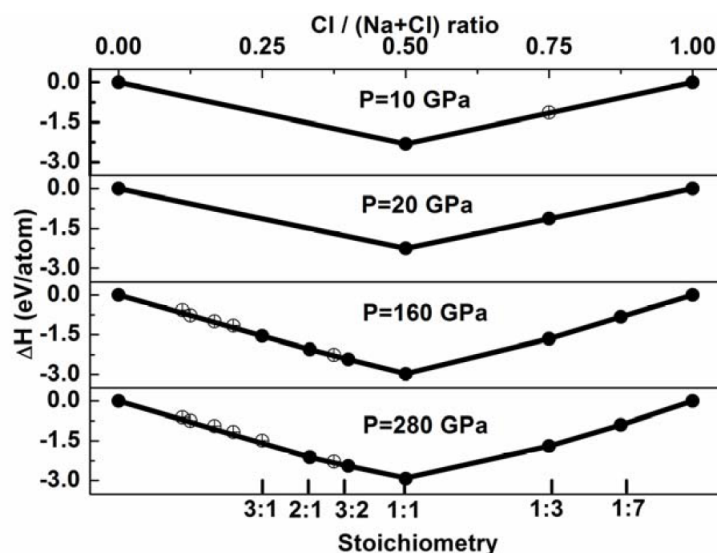


图 9: Convex hull diagram for Na-Cl system at selected pressures. Solid circles represent stable compounds; open circles — metastable compounds.

5.2 热电导率优化

寻找具有高热电导率的材料为可持续能源提供了一个综合解决的方案。因此，寻找如何优化热电导率变得非常必要。品质因数 ZT 可以用来衡量一种材料的热电导率，它的定义是：

$$ZT = \frac{\sigma S^2 T}{\kappa_E + \kappa_L}, \quad (7)$$

其中 σ 是电导率， S 是塞贝克系数， T 是温度， κ_E and κ_L 分别表示电子热导率和晶格热导率。除了 κ_L ，假设在一个恒定的弛豫时间内，其余参数的值都是可以通过使用代码 BoltzTraP²⁵ 来计算一系列的弛豫优化（比如使用 VASP）的结构来得到的。

5.2.1 相关软件的安装

通过 USPEX 进行热电导率优化需要其他软件的支持。

1. BoltzTraP²⁵ 的源代码。可以从 <http://www.icams.de/content/departments/cmat/boltztrap/> 下载。请参考源代码附带的手册来进行安装。
2. Python and python 库。USPEX 的接口，VASP 和 BoltzTraP²⁵ 都在 Anaconda Python 2.7.10 经过了测试。Anaconda Python 2.7.10 可以在 <https://www.continuum.io/downloads> 进行下载。除了 Python 标准库之外，还需要以下库：
 - numpy, version 1.4.1 或者以上。
 - scipy, version 0.11.0 或者以上。

- ase, version 3.8.1 或者以上。
- pyspglib (任意版本)
- matplotlib (任意版本)

有了 Anaconda Python, 大量的库可以通过命令 `conda` 被安装。以下命令行是安装进行热电优化库的一个例子。

```
$ conda install numpy scipy matplotlib
```

```
$ pip install ase pyspglib
```

在一个安装 `pip` 的 Python 发行版里, 我们可以通过输入以下命令行来安装要求的软件包。

```
$ pip install numpy scipy matplotlib ase pyspglib
```

5.2.2 在 INPUT.txt 里设置热电导率计算

对于热电导率优化, `optType` 应该被设置成 14: :

```
14 : optType
```

因为热电性质的计算针对的是已经弛豫的结构, 因此一系列的结构弛豫应该在此之前完成。目前 VASP 是唯一一个可以与 USPEX 和 BoltzTraP²⁵ 结构的代码。特定的设置如下:

```
% abinitioCode
```

```
1 1 1 1 14
```

```
% ENDabinit
```

```
% KresolStart
```

```
0.12 0.10 0.07 0.05 0.05
```

```
% Kresolend
```

在以上的例子中, 结构弛豫通过一系列的 VASP 计算来完成 (一共 4 步), 每增加一步都对应着更密的 `k` 点。注意由于兼容性的问题, 最后一个 `KresolStart` 数值将不会被使用。

NOTE: 正如在 Section 4.8, `abinitioCode` keyblock 中提到的, 括号是用于对结构施加对称性以用于进一步的性质计算。

热电导率的优化只对 `calculationType` 300 和 301 开放。为了使用这个方法, 需要进行以下设置。为了更深一步理解它们, 用户可以参考提出 BoltzTrap 的论文²⁵。

```
800.0 : BoltzTraP_T_max
```

```
50.0 : BoltzTraP_T_delta
```

```
0.15 : BoltzTraP_efcut
```

```
300.0 : TE_T_interest
```

```
0.45 : TE_threshold
```

```
ZT : TE_goal
```


▷ variable BoltzTraP_T_max

Meaning: BoltzTraP 计算的最大温度 (K)。

Default: 800.0

Format:

800.0 : BoltzTraP_T_max

▷ variable BoltzTraP_T_delta

Meaning: BoltzTraP 里温度 (K) 的增加值。

Default: 50

Format:

50.0 : BoltzTraP_T_delta

▷ variable BoltzTraP_T_efcut

Meaning: 这个变量表示的是 BoltzTraP 计算里化学式 μ (eV) 的间隔

Default: 0.15

Format:

0.2 : BoltzTraP_efcut

▷ variable TE_T_interest

Meaning: 在使用 USPEX 进行热电斜率优化的过程中, 温度 (K) 一直被监视和追踪这个应该是热电材料的工作温度。

Default: 300.0

Format:

300.0 : TE_T_interest

▷ variable TE_threshold

Meaning: 这个参数是用来减少优化时产生的存储, 同时设置 ZT 值得下限。存储 BoltzTraP 产生的所有的信息将会占用大量的磁盘空间。那些 ZT 值低于设定值 `TE_threshold` 的结构将会被遗弃

Default: 0.5

Format:

0.45 : TE_threshold

▷ *variable* `TE_goal`

Meaning: ZT 的组成将会被优化，因为用来定义 ZT 的物理量是张量，用户需要去选择 ZT 的组成来进行优化。目前支持的选项是追踪 ZT 或者是对角元 ZT_{xx} , ZT_{yy} , 和 ZT_{zz} 。

Default: `ZT`

Format:

`ZT : TE_goal`

5.2.3 输出

热电导率的优化的输出文件在文件夹 `results-folder` 的子文件夹 `TEproperties` 里。 ZT 的最大值和对应的 μ 列在 `summary.txt` 中，第一列代表结构的 ID。有了 ID，这个结构就可以在文件 `gatheredPOSCARS` 找到。`keepBest` 没有被列在 `summary.txt` 中，`summary.txt` 中的一小段样板列举如下：

```
#
# ID      mu_max_1  ZT_max_1  mu_max_2  ZT_max_2
  1  0.019900  0.314997 -0.186600  0.294976
  2  0.152270  0.913590  0.002270  0.864951
  3 -0.021300  4.951155  0.152200  3.244481
  4 -0.194630  0.677683  0.054370  0.628948
  6  0.024120  6.144226  0.053620  0.914606
  7  0.030460  0.887808  0.041960  0.831734
  9  0.167860  0.479234  0.159360  0.446431
 10  0.133260  0.893694  0.144760  0.822560
 12  0.063250  0.903193  0.075750  0.843479
 13  0.044820  1.793899  0.004820  1.026578
```

个体的热电导性质（例如 σ , S , κ_E 和功率因子 σS^2 ）可以用对应文件附加的结构 ID 号在文件夹 `TEproperties` 中找到。

注意，由于热电性质的计算中会出现数值的不稳定，有些结果或许不正确，在这种情况下 ZT 的值可能是正常值得 10 倍或者上百倍。为了解决这个问题，在这个数大于设定的上限例如 10，就会被乘以-1。这样这些反常的值将不会应该优化的结果。

5.3 分子晶体和聚合物

5.3.1 分子晶体, `calculationType=310/311`

对于一个分子晶体，`MOL_1` 文件描述分子结构开始构建的结构部分。这个文件也定义如果分子可活动的哪个扭转角会突变。这个文件和它的格式跟 SIESTA' s `Z_Matrix` 文件（`MOL_1` 给出原子的笛

卡尔坐标，反之，Z_Matrix 文件从键长、键角和扭转角定义原子位置）不同。Z_Matrix 文件是利用 MOL_1 文件给的信息创建的，即：键长和所有必要的角度都是用笛卡尔坐标计算的。重要的键长和角度用来创造 Z_Matrix—这恰好是 5-7 列列举的。我们看一下苯 C₆H₆ 的 MOL_1 文件：

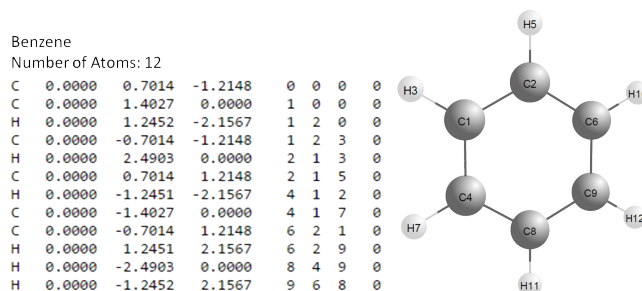


图 10: Sample of MOL_1 file and illustration of the corresponding molecular structure.

第一个原子是 H，它的坐标定义与其他原子无关（“0 0 0”）。

第二个原子是 C，它在 Z_matrix 的坐标（在分子坐标系中）仅根据它到第一个原子（即，上面描述的 H）的距离来设置，但没有角度—（“1 0 0”）。

第三个原子是 C，它的坐标将根据它到第二个原子的距离来设置，键角 3-2-1，但不是扭转角—因此我们用“2 1 0”。

第四个原子是 C，它的坐标根据它到第三个原子的距离设置，键角 4-3-2，扭转角 4-3-2-1—因此，我们用“3 2 1”等等...直到最后第 12 个原子，H，根据它到第七个原子（C）的距离定义，键角 12-7-6，扭转角 12-7-6-11—因此是“7-6-11”。

最后一列是扭转角灵活度的标示。例如，在 C4 中，扭转角由 4-3-2-1 定义。理想情况下，对于前 3 个原子来说这个标示应该是 1，其他原子的是 0，如果存在其他活动的扭转角，需在这一列说明是 1。

5.3.2 聚合物晶体, calculationType=110 (“线性链状模型”)

对于聚合物，MOL_1 文件用来表示单体单元的几何结构，和分子晶体一样，除了要在最后一行列出活性原子，如下图列出的 PVDF（聚偏二氟乙烯）的 MOL_1 文件：

5.3.3 经典力场的额外输入

上面的 MOL_1 文件适用于 USPEX 的一般情况。然而，一些基于编码的经典力场需要额外的信息。例如，GULP 需要列出化学分类和电荷。阿司匹林的 MOL_1 文件的写法如下：

```
Aspirin_charge
Number of atoms: 21
H_1      0.2310      3.5173      4.8778      0 0 0 1      0.412884
```

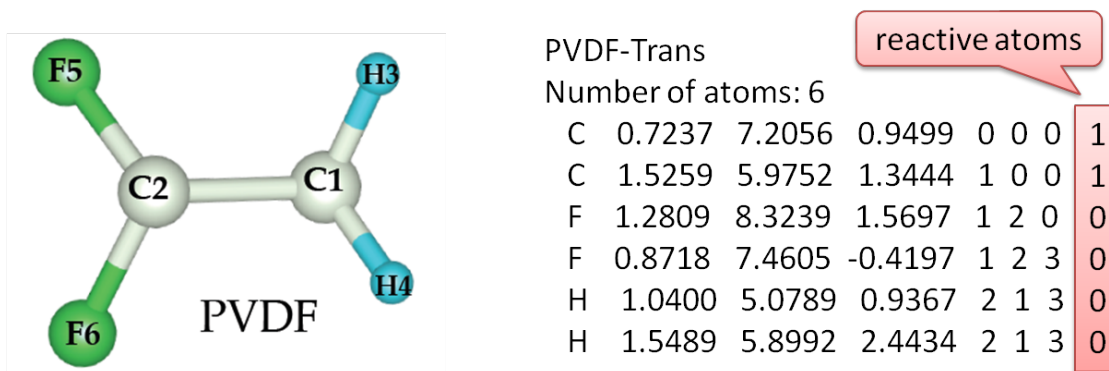


图 11: Sample of MOL_1 file of PVDF and illustration of the corresponding monomeric structure.

```

O_R      0.7821      4.3219      4.9649      1 0 0 1      -0.676228
C_R      0.4427      5.0883      6.0081      2 1 0 1      0.558537
O_2     -0.5272      4.5691      6.6020      3 2 1 0     -0.658770
C_R      1.0228      6.3146      6.3896      3 2 4 0      0.116677
C_R      2.1330      6.8588      5.6931      5 3 2 0      0.311483
C_R      0.4810      7.0546      7.4740      5 3 6 0     -0.119320
O_R      2.8023      6.2292      4.6938      6 5 3 0     -0.574557
C_R      2.6211      8.1356      6.0277      6 5 8 0     -0.083091
C_R      0.9966      8.3146      7.8237      7 5 3 0     -0.103442
H_2     -0.3083      6.6848      8.0128      7 5 10 0     0.198534
C_R      3.6352      5.1872      4.9079      8 6 5 0      0.609295
C_R      2.0623      8.8613      7.0940      9 6 5 0     -0.119297
H_2      3.3963      8.5283      5.4906      9 6 13 0     0.174332
H_2      0.5866      8.8412      8.6013     10 7 13 0    0.205960
O_2      3.9094      4.7941      6.0632     12 8 6 0    -0.588433
C_3      4.2281      4.5327      3.7638     12 8 16 0   -0.271542
H_2      2.4227      9.7890      7.3367     13 9 10 0   0.196738
H_2      3.4269      4.1906      3.1183     17 12 8 0   0.151315
H_2      4.8283      3.6848      4.0792     17 12 19 0  0.131198
H_2      4.8498      5.2464      3.2337     17 12 19 0  0.127726

```

这里，标题的关键词电荷是告诉程序在最后一行读取电荷。

结合 Tinker 一起计算，附加行用来列出原子标签，如下所示：

```

Urea
Number of atoms: 8

```

```

C    0.000000    0.000000    0.000000    0 0 0 1 189
O    0.000000    0.000000    1.214915    1 0 0 1 190
N    1.137403    0.000000   -0.685090    1 2 0 1 191
N   -1.137403    0.000000   -0.685090    1 2 3 0 191
H    1.194247    0.000000   -1.683663    4 1 3 0 192
H   -1.194247    0.000000   -1.683663    4 1 3 0 192
H    1.998063    0.000000   -0.138116    2 1 3 0 192
H   -1.998063    0.000000   -0.138116    2 1 3 0 192

```

5.3.4 如何准备 MOL 文件

有许多程序可以生成 Zmatrix style 文件，如 Molden、Avogadro 等。有经验的用户或许有自己的方法来准备这些文件。为了用户的方便，我们创建了一个在线工具，允许用户仅用 XYZ 格式的文件来生成 USPEX-style MOL 文件。这个工具的网址为 <http://han.ess.sunysb.edu/zmatrix>。

5.4 表面

为了完成表面重构的预测，必须：

- 指定 200 或 201 : 计算类型。
- 准备包含基底的 VASP5 POSCAR 格式的文件，如图 11 12所示。
- 列举以下参数：

```

% symmetries
2-17
% endSymmetries

```

注意：如果对称标签 `symmetries` 出现，USPEX 将尝试使用平面组生成结构。

▷ *variable* `thicknessS`

Meaning: 表面区域的厚度。只有这个区域允许吸附原子。

Default: 2.0 Å

Format:

```
3.5 : thicknessS
```

▷ *variable* `thicknessB`

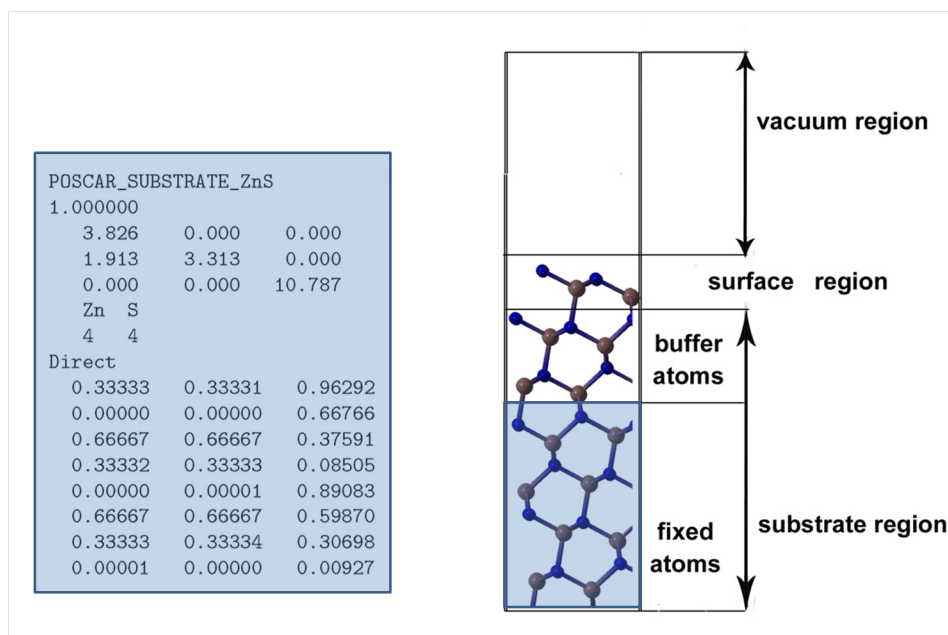


图 12: USPEX 使用的表面模型。注意 POSCAR_SUBSTRATE 应该准确代表没有真空的晶体的几何信息。如果输入中有大量的真空区域, 程序将自动删除它并提供一个名为 POSCAR_SUBSTRATE_NEW 的新文件, 这个文件必须在计算中用到 (重命名为 POSCAR_SUBSTRATE)

Meaning: 基底的缓冲区域的厚度。这个区域是 POSCAR_SUBSTRATE 的一部分, 可以松弛。

Default: 3.0 Å

Format:

3.0 : thicknessB

▷ *variable* **reconstruct**

Meaning: 表面单元的最大增值, 允许复杂重建。

Default: 1

Format:

1 : reconstruct

USPEX 也提供研究表面原子变成分搜索的可能 (在 USPEX 10.1 发布)。在这种情况下, 稳定的表面重建是由化学势²⁶控制的。典型的输入设置如下:

```

*****
*      TYPE OF RUN AND SYSTEM      *
*****
USPEX : calculationMethod (USPEX, VCNEB, META)

```

```
201      : calculationType (dimension: 0-3; molecule: 0/1; varcomp: 0/1)
```

```
% atomType
```

```
Si 0
```

```
% EndAtomType
```

```
% numSpecies
```

```
2 4
```

```
% EndNumSpecies
```

这里我们列出 1×1 晶胞表面原子的最大数。

```
*****
```

```
*                SURFACES                *
```

```
*****
```

```
% symmetries
```

```
2-17
```

```
% endSymmetries
```

```
% StoichiometryStart
```

```
1 2
```

```
% StoichiometryEnd
```

这定义晶体的化学计量数。

```
-23.7563 : E_AB (AmBn化合物的DFT能量, eV/化学式)
```

```
-5.4254  : Mu_A (元素A的DFT能量, eV/原子)
```

```
-4.9300  : Mu_B (元素B的DFT能量, in eV/atom)
```

```
3.5      : thicknessS (表面区域的厚度, 默认2  $\text{\AA}$ )
```

```
3.0      : thicknessB (基底缓冲区域的厚度, 默认3  $\text{\AA}$ )
```

```
4        : reconstruct (晶胞的最大倍数)
```

现在, USPEX 支持如下的表面变成分计算:

- 基本的表面重建 (如 C 在 diamond(100) 表面)。
- 二元化合物的表面重建 (如 GaN(0001) 表面)。
- 包含不同物种的基本表面的重建 (如 PdO 在 Pd(100) 表面)。

5.5 二维晶体

5.5.1 二维晶体，计算类型 =-200/-201

对于二维晶体，USPEX 目前可以进行二元的固定组分或变组分的计算（分别对应于-200 和-201），三元及以上的还在开发。不同于表面计算（200/201），我们既不需要端点参考能也不需要衬底，而只需设置二维初始结构的厚度即可，例如：

```
3.0: thicknessS
```

表示我们是在 3.0 埃厚度层内产生的二维初始结构。经验上我们将弛豫后的二维结构厚度限定在 `thicknessS+3` 范围内。更重要的是，我们还需给定二维晶体一个足够大的真空层（默认数值为 10 埃，但这通常不够）。和表面计算对称性（2-17 平面群）设置不同，二维晶体的对称性我们用 2-80 的层面群来描述，如附录 9.5 中所列。

```
% symmetries
2-80
% endSymmetries
```

5.5.2 定组分

定组分的化学配比我们用 `numSpecies` 来设定，同时用 `minAt` 和 `maxAt` 参数来设定原子数范围。举例来说，我们研究 B_2O_3 的二维晶体体系，体系原子数为 4-20，也就是说我们需要在同一个 USPEX 计算作业中考虑原子数比例为 2:3, 4:6, 6:9 和 8: 12 的二维结构：

```
% numSpecies
2 3
% EndNumSpecies
```

```
% atomType
B O
% EndAtomType
```

```
4 : minAt
20 : maxAt
```

5.5.3 变组分

对于二维晶体的全变组分结构搜索，类比于三维晶体预测，我们通过设定 `numSpecies` 矩阵来实现：


```
% numSpecies
1 0
0 1
% EndNumSpecies
```

注意：因为我们设定了弛豫后二维晶体结构厚度的上限，因此我们会舍弃掉那些在弛豫后厚度大于 $\text{thicknessS}+3$ 的二维结构。对于全变组分的结构搜索计算，我们可能需要设定一个较大的 `initialPopSize`（建议至少两倍于 `populationSize`）：

```
90 : populationSize
180 : initialPopSize
100 : numGeneration
```

5.5.4 二维晶体预测结果

对于-200 计算类型，可以在 `goodStructures` 文件中找到好的预测后的二维晶体结构，下图中有一个输出算例。对于-201 变组分计算类型，我们可以在 `extendedConvexHull.pdf` 中找到对应的 `convexhull` 图，在 `extended_convex_hull` 文件中找到所有结构的信息（稳定结构优先排序），如下图所示。Sn-S 体系的变组分二维晶体预测算例可以在 USPEX 程序中找到（EX30）。

ID	Compositions	Enthalpies (eV/atom)	Thickness A	Surface_area (A ²)	Spec_surf_area (m ² /g)	fitness (^o)	SYMM
143	[4]	-9.2236	0.0000	10.5528	2645.5711	-9.2236	191
78	[4]	-8.7221	0.0000	11.8822	2978.8604	-8.7221	123
11	[4]	-8.5367	0.0004	12.7939	3207.4204	-8.5367	187
41	[4]	-8.4279	0.0001	12.2276	3065.4564	-8.4279	65
69	[2]	-8.2782	0.0001	10.8403	5435.2904	-8.2782	1
72	[2]	-8.2473	0.0002	8.5149	4269.3632	-8.2473	1
150	[6]	-8.2250	2.1192	9.7086	1622.6223	-8.2250	51
76	[2]	-8.2040	0.0000	9.6220	4824.4432	-8.2040	1
37	[4]	-8.0326	1.5786	6.3675	1596.3332	-8.0326	191

```
It contains all the information in extendedConvexHull.pdf
# X axis: Composition
# Y axis: Y = (E(AxBy)-x*E(A)-y*E(B))/(x+y)
# Fitness: its distance to the convex hull (eV/block)
```

ID	Compositions	Enthalpies (eV/atom)	Thickness (A)	Fitness (eV/block)	SYMM	X	Y (eV/atom)
3024	[1 4]	-4.8128	1.7647	-0.0003	1	0.800	-0.2531
1316	[8 0]	-9.2265	4.3598	0.0000	67	0.000	0.0000
2913	[0 4]	-3.3930	3.3321	0.0000	65	1.000	0.0000
306	[1 6]	-4.4059	4.3380	0.0009	1	0.857	-0.1796
790	[1 8]	-4.1802	4.3734	0.0014	3	0.889	-0.1391
1216	[1 10]	-4.0366	3.3797	0.0016	1	0.909	-0.1133
2378	[1 4]	-4.7956	1.2762	0.0168	3	0.800	-0.2359
1412	[0 8]	-3.3720	4.1494	0.0210	10	1.000	0.0210
1162	[2 6]	-5.0662	2.3035	0.0221	2	0.750	-0.2149
2105	[2 4]	-5.5105	2.7845	0.0377	12	0.667	-0.1730

5.6 团簇

为了预测团簇结构，计算类型设定为 000

注意: 不要忘记设置 `vacuumSize`，它应该足够大从而确保避免因周期性造成的团簇间相互作用。

5.7 进化准动力学命令

这对于找全局最小值是一个非常强大的方法，以及开始结构中潜在的动力学接近的低能量亚稳结构。开始的结构必须是高质量的且在 `POSCAR_1` 中给出。进化准动力学只能在 VASP 和 GULP 中使用。

为了运行进化准动力学程序，必须：

1. 设置

`META` : 计算方法

`300` : 计算类型

2. 在你的文件夹的 VASP5 版本中建立 `POSCAR_1` 文件（进化准动力学需要一个好的初始结构，在感兴趣的压力下弛豫）。

3. 设定数量大小（这个例子中，这是每一步软模变异的数量）：

`30` : `populationSize`

4. 设定压力：

▷ *variable* `ExternalPressure`

Meaning: 你想在哪个压力下执行计算，单位 GPa.

Default: 无默认值

Format:

`10` : `ExternalPressure` (GPa)

5. 设置准动力学选项：

▷ *variable* `GaussianWidth`

Meaning: 每个加入到能量面以加速相变的高斯宽度。好的经验是从 $0.10-0.15L$ 选一个值，这里 L 是单胞的最小长度，单位为 Angstroms。

Default: $0.10 \times L$ (Å)

Format:

`0.80` : `GaussianWidth`

▷ *variable* `GaussianHeight`

Meaning: 每个加入到能量面以加速相变的高斯高度。好的经验 (Martoňák *et al.*, 2005) 是选一个接近 $L(\delta h)^2 G$, L 是单胞的平均长度, 单位为 Angstroms, δh 是高斯宽度, G 是剪切模量, 单位为 kbars。

Default: $1000 \times (0.10 \times L)^2 \times L = 10 \times L^3$ ($\text{\AA}^3 \text{kbar}$)

Format:

```
2000 : GaussianHeight
```

▷ *variable* `FullRelax`

Meaning: 这样的准动力学仅在固定晶胞中弛豫结构。为了分析, 我们需要进行完整的结构弛豫 (即弛豫晶胞)。

- `FullRelax=0` — 没有完全弛豫要执行 (非常快速的选择, 但对结果分析造成不便)。
- `FullRelax=1` — 仅一代中的最好结构要完全弛豫也很快, 有时充足)。
- `FullRelax=2` — 所有的等价结构完全弛豫 (依旧很快, 只比 `FullRelax=1` 时慢 ~ 2 倍, 但提供了更多的内涵信息。大多数案例都需要它)。

Default: 2

Format:

```
2 : FullRelax
```

为了完全弛豫, 当执行进化准动力学计算时, `abinitioCode` 的格式略有不同, 比如:

```
abinitioCode
3 3 3 3 (3 3)
ENDabinit
```

在上面的例子中, 一个固定晶胞的弛豫有 4 个阶段, 完全弛豫有 2 个 (括号中)。记住, 在最后的固定晶胞的弛豫阶段, 压力张量必须精确—这才能驱动准动力学过程。目前只有 VASP, SIESTA 和 GULP 支持此计算。

注意: 在进化准动力学 (`calculationMethod = META`) 中使用圆括号的含义与在进化优化结构 (`calculationMethod = USPEX`) 时使用不一样的。在优化结构时, 括号的含义是完全对称化结构并只考虑性质 (忽视能量)。而准动力学计算中并没有对称化和性质计算, 括号的含义是完全结构优化 (包括对晶胞的弛豫)。

▷ *variable* `maxVectorLength`

Meaning: 与 `minVectorLength` 一样，是基础晶胞在进化准动力学模拟时的长度边界值（注意在与一般计算中 `minVectorLength` 含义不同，`maxVectorLength` 仅在进化准动力学中使用）。当任何基础晶胞的长度比 `minVectorLength` 或比 `maxVectorLength` 大，我们会在准动力学中添加一个夸张的修正力，这驱使晶胞进化朝“好”的方向发展。当所有基础晶胞的长度在“好的”范围时，这个修正力为零。

Default: No default

Format:

12.0 : `minVectorLength`

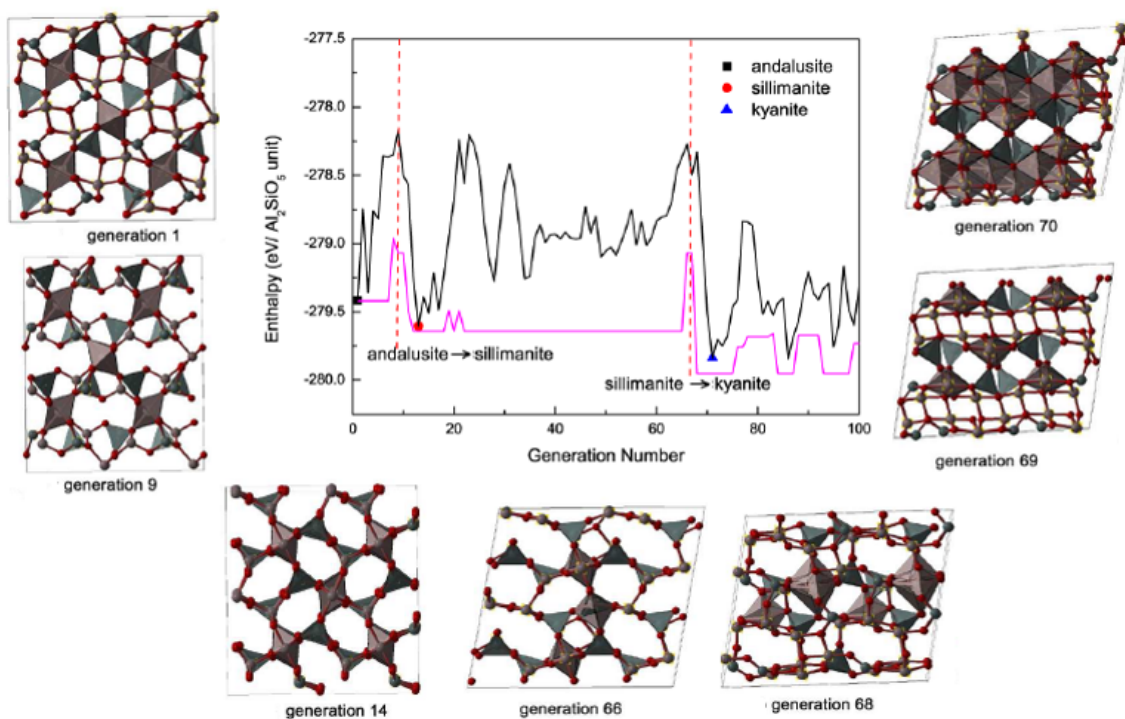


图 13: 10GPa 时压缩红柱石 (Al_2SiO_5) 过程中的焓演变 (黑线: 常量 h 下最好结构的焓; 红线: 最好的结构完全弛豫后的焓)。得到结构的顺序为: 第 1 代 (红柱石) \rightarrow 第 9 代 (硅线石) \rightarrow 第 14 代 \rightarrow 第 66 代 \rightarrow 第 68 代 \rightarrow 第 69 代 \rightarrow 第 70 代 (蓝晶石)。

当运行准动力学时，附加文件可在 `results1` 文件夹中找到，更重要的是：

- `force.dat` — 分析晶胞上的力，内部的 (`f_c`) 和来自高斯的 (`f_g`)；
- `presten` — 压力张量；
- `lattice.dat` — 模拟过程中晶胞形状的改变；
- `enthalpies` and `enthalpies_relaxed` — 固定晶胞每一步结构的焓和完全弛豫后的焓；

- `gatheredPOSCARS` and `gatheredPOSCARS_relaxed` — 固定晶胞的结构和完全弛豫后的结构。

图 Fig. 13展示了进化准动力学应用的例子：从一个 Al_2SiO_5 同质多形体（红柱石）开始，我们得到另外两个已知的同质多形体（蓝晶石和硅线石）和重要的相变机理。

5.8 粒子群算法（PSO）指令

在晶体和簇结构预测领域，一些算法对于小体系非常成功。粒子群算法，在这个领域由 Boldyrev²⁷ 所倡导，是进化算法的特殊情形，大量的（集群）候选值（称为“微粒”）根据一些简单的公式移动到搜索空间。微粒的运动由它们在搜索空间中自己最显著的位置决定，和全体集群最显著的位置一样。首先，微粒的坐标 χ 和‘速度’ v 随机产生。随后每一步，位置和速度都根据以下公式更新：

$$\begin{aligned} v'_i &= \omega \cdot v_i + \varphi_p \cdot r_p \cdot (p_i - \chi_i) + \varphi_g \cdot r_g \cdot (g - \chi_i), \\ \chi'_i &= \chi_i + v'_i. \end{aligned} \quad (8)$$

这里 ω , φ_p 和 φ_g 是加权因子，控制 PSO 算法的行为和效率； r_p 和 r_g 是 $[0; 1]$ 范围内的随机数，分别在每一步为每一个微粒产生； p_i 是微粒 i 最显著的位置， g 是整个集群最显著的位置。

这种算法，尽管它简单但可以工作²⁷。改进之前安装运行^{27;28} 的关键：（1）搜索空间的度量（单独在坐标系中绘制各个晶体结构不重要）；（2）用 PSO 进化结构的方法，也就是变异操作。

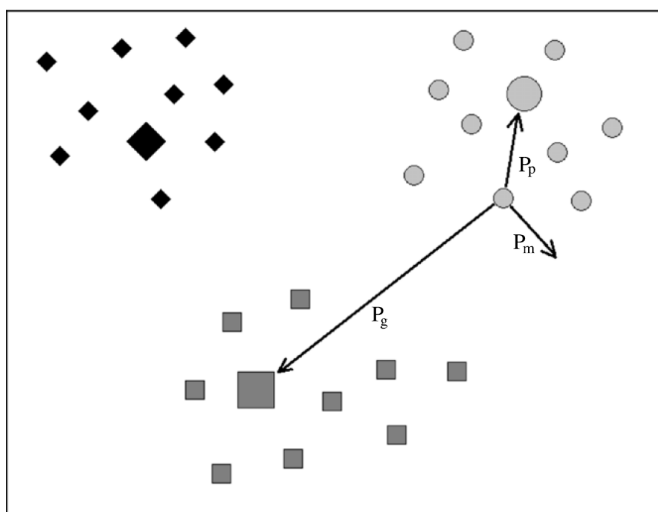


图 14: 三个个体数量 10 代之后的的 PSO-USPEX 混合算法图例（用菱形、正方形和圆形标记）。每个微粒的最好位置都用放大的符号标记。最好的结构是大的正方形。圆形标记的结构可突变，用其历史中最好的位置（大圆）产生子代或通过遗传操作以 P_m , P_p 和 P_g 的概率产生全部种群（大正方形）的最好位置。

通过直接由原子坐标和两个结构的参数决定的速度 v_i (8) 来进化微粒是不可行的。我们的问题是用指纹距离¹⁶ 作为能谱最原始的坐标，且 USPEX 用来进化 PSO 微粒的变量作为进化种群结构最有效公正的方法。即，微粒要么突变（模拟自由移动），要么是以最佳位置参与遗传或最佳的种群位置

(模拟 PSO 朝这些方向移动) 参与遗传。代替了在所有的移动的每一步都加上一些权重 (see Eq. 8), 我们通过以下描述的可能性一次性加上:

$$P_m = \frac{\omega}{\Sigma}; \quad P_p = \frac{\varphi_p \cdot r_p \cdot D_p}{\Sigma}; \quad P_g = \frac{\varphi_g \cdot r_g \cdot D_g}{\Sigma};$$

$$\Sigma = \omega + \varphi_p \cdot r_p \cdot D_p + \varphi_g \cdot r_g \cdot D_g,$$
(9)

其中 D_p 是一个微粒现在和最佳位置间的指纹间距, D_g 是微粒现在位置和全部种群最佳位置的指纹间距。我们的测试在几个不同的系统进行, 表明这种方法 (我们称之为 “cor-PSO”, 即校正 PSO) 是比较成功的。而且比 PSO 之前的版本好, 但在成功率和效率方面仍不能和 USPEX 算法【3,41】相比。

下面的变量对 `calculationMethod=PSO` 是唯一的:

▷ *variable* `PSO_softMut`

Meaning: 软模变异的权重 (公式 eq. 9 中的 ω)

Default: 1

Format:

1 : PSO_softMut

▷ *variable* `PSO_BestStruc`

Meaning: 用同样 PSO 微粒的最好位置遗传的权重 (φ_p in eq. 9)。

Default: 1

Format:

1 : PSO_BestStruc

▷ *variable* `PSO_BestEver`

Meaning: 由全局遗传最好 PSO 的微粒 (φ_g 的权重 eq. 9)。

Default: 1

Format:

1 : PSO_BestEver

6 相变路径预测

相变决定材料许多方面的行为。因此，很有必要揭示结构相变可能存在的机理。

6.1 Pmpaths: 预测固体-固体相变机制的几何方法

这个由 V. Stevanovic, P. Graf 和 F. Therrien 开发, 并进一步由 A. Samtsevich 拓展的工具, 可以用来解决这样一个难题: 如何找到两个给定晶体结构之间最佳的映射路径。这直观的对应于最简单的相变机制。Pmpaths 会排序列出两个给定晶体结构之间的不同路径。然后通过 VCNEB 方法优化这些路径 (详见下节)。该算法首先通过优化晶胞的“重叠”来搜索结构晶格的映射路径, 并且对晶格矢量进行置换, 使得坐标系旋转轴的最佳重叠尽可能大。这是后续分析和确保找到在两个晶体结构的原子位置间最佳的映射路径的起点。

这个工具依赖于这些 Python 数据库:

```
munkres
scikit_learn
spglib
cython
```

这些数据库都可以通过 pip 安装。为了运行 pmpaths 工具, 请尝试:

```
python3 run_pmpaths.py -t 4 -z traj_A2B -n 20 -A POSCAR_A -B POSCAR_B -b 3.6 --numtraj=5
```

帮助文档:

```
Usage: run_pmpaths.py [options]
```

Options:

```
-h, --help          show this help message and exit
-A A, --A=A         poscar 1
-B B, --B=B         poscar 2
-t OUTPUT_TILES, --tiles=OUTPUT_TILES
                    how many cells to tile in output
-H, --hlst          perform HLST fitting
-v VERBOSE, --verbose=VERBOSE
                    verbosity
-z TRAJDIR, --trajdir=TRAJDIR
                    where to dump trajectory files
-c MIN_CLUSTER_SIZE, --min_cluster_size=MIN_CLUSTER_SIZE
                    minimum size of atom clusters
-s SHIFTING, --shifting=SHIFTING
                    control shift of inequiv atoms to origin (0:none,
                    1:one of each inequivalent subgroup,2:all)
-b BOND_LEN, --bond_len=BOND_LEN
                    bond length
-n FRAMES, --frames=FRAMES
```

```
                                how many frames in trajectory
-y, --nocheck_syms             don't check syms
-u, --nocheck_ucells           don't check more than one unit cell pairing, but DO do
                                gruberization
-w, --use_given_ucells         use given unit cells as is, no gruberization
-d, --noucell-dist             don't include unit cell vector movement in distance
                                measure
-f, --get-fast                 use bonding to specifically search for FAST
-e TOL, --tol=TOL             tolerance for coordination calcs
--numtraj=N                    , where N is the number of pathways that will be in the output
```

运行 `pmpaths.py` 之后的输出是一个文件夹 (除了 `stdout` 中的输出之外) (“`traj_A2B`”), 里面包含映射路径上的 n 个图像 (“`images`”). 我们推荐不只寻找一条几何上的最优路径, 而是寻找 N 条 (N 5-10) 几何最优路径。记住这些路径仅仅给出了两个晶体结构间的线性插值, 这些插值点需要做局域结构优化。这就需要使用到下一节中介绍的 VCNEB 方法。我们的 VCNEB 代码可以直接使用 `pmpaths utility` 的输出。

6.2 变胞微动弹性带 (VCNEB) 法

预测相变机制可以认为是一个两端的问题, 算法必须定位在中间状态。变胞微动弹性带 (NEB)^{29;30;31} 方法是解决两端问题广泛运用的技术, 一个有效而稳健的方法来寻找反应路径和鞍点, 在“最小能量路径” (MEP) 上的两个端点之间的势能面。该 NEB 方法已成功应用于分子化学反应和缺陷迁移, 尤其它可以提供相变过程给定的初始和最终状态间的能量势垒。不幸的是, 大多数用 NEB 方法解决的问题都是在单胞的限定下, 这排除了它用于相变的可能 (这涉及沿相变路径变异的单胞)。

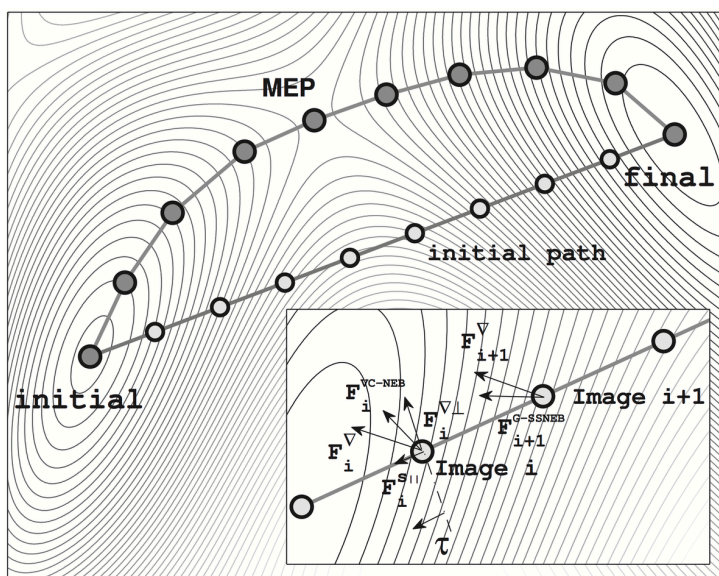


图 15: 2D 焓表面模型的最小能量路径 (有灰色圆的线) 和初始路径。影像 i 用 VC-NEB 方法的力示于插图。 F_i^∇ 是梯度方向的潜在力。 F_i^{\perp} 和 F_i^∇ 分别是的横向分量和弹性力。

变胞 NEB (VC-NEB) 方法¹³, 我们用不同的公式来完善, 平等对待单胞和原子坐标, 在恒定压力下在膨胀的结构空间运行。以第一性原理为框架的 VC-NEB 方法已经被添加到 USPEX 代码中。VC-NEB 方法是探索更大结构空间相变过程两个端点激活路径的常用工具。VCNEB 方法中路径上的所有结构都被示为 “Image”。

6.3 VCNEB 的 Input 选项

VCNEB 方法目前只能用 VASP, GULP 和 Quantum Espresso 指令。

为了利用 VCNEB 模块, 必须:

1. 设定

```
VCNEB: calculationMethod
```

2. 以 VASP5 文件格式在你的文件夹中创建 Images 文件 (VCNEB 需要至少两个结构, 初始和最终相, 来进行相变路径预测)。

3. 设置以下 VCNEB 选项:

▷ variable `vcnebType`

Meaning: 说明 VCNEB 计算的类型。这个变量包含 3 个目录: *calculation option*, *Image number variability*, 和 *spring constant variability*:

- calculation option:
 - “1” — 使用 VCNEB 方法;
 - “2” — 结构弛豫时不使用 VCNEB 计算;
- Variable-Image-Number method:
 - “0” — VCNEB 的 Images 数量固定;
 - “1” — VCNEB 的 Images 数量可变;
- variability of spring constant:
 - “0” — 固定弹性常数;
 - “1” — 弹性常数可变;

Default: 110

Format:

111 : `vcnebType`

注意: 如果 `vcnebType=111`, 即, VCNEB 计算的 Images 值可变, 弹性常数可变。我们强烈建议用户用 VCNEB 方法研究重建相变时使用可变的 Images 值。

▷ variable `numImages`

Meaning: 执行计算初始的 Images 值。

Default: 9

Format:

13 : `numImages`

▷ variable `numSteps`

Meaning: 执行 VCNEB 计算的最大步数。

Default: 600

Format:

500 : numSteps

注意: (1) 当 numSteps=-1, 只会产生初始路径而不再做路径优化。(2) VCNEB 路径通常收敛很慢。我们要求 numSteps 最少设为 500。

▷ *variable* `optReadImages`

Meaning: Images 文件的参考选项: :

- “0” — 需要所有 images (numImages) 并写入在 Images 文件;
- “1” — 只需要初始和最终的 images, 可以在 Images 中看的;
- “2” — 初始、最终和任何特定的中间 Images 都能在 Images 文件中看的。

Default: 2

Format:

1: optReadImages

注意: 所有选项中, 初始和最终的 images 必须指定。自动线性插值法被应用于在选项 1 和 2 中产生初始 Images。

▷ *variable* `optimizerType`

Meaning: 结构弛豫的优化算法选项:

- “1” — 最陡下降法 (SD);
- “2” — 快速惯性弛豫算法 (FIRE, Fast Inertial Relaxation Engine) 算法³²。

Default: 1 (SD) 对于 VCNEB 计算; 2 (FIRE) 对于结构弛豫

Format:

1 : optimizerType

▷ *variable* `optRelaxType`

Meaning: 结构弛豫类型:

- “1” — 晶胞固定, 仅弛豫原子位置, 跟传统 NEB 方法一样;
- “2” — 仅弛豫晶格 (仅用于测试);
- “3” — 原子位置和晶格完全弛豫。

Default: 3

Format:

3 : optRelaxType

▷ *variable* `dt`

Meaning: 结构弛豫的时间步长。

Default: 0.05

Format:

0.1 : dt

注意: 如果 `dt` 很小, 计算就会很慢。如果 `dt` 很大, 计算将不稳定且产生没有意义的路径。

▷ *variable* `ConvThreshold`

Meaning: images 中均方根力 (RMS, Root Mean Square forces) 弛豫的标准条件。

Default: 0.003 eV/Å

Format:

0.005 : ConvThreshold

▷ *variable* `VarPathLength`

Meaning: 变 Images 方法的标准路径长度。当两个相邻 images 的长度比 `VarPathLength` 的 1.5 倍还大, 新的 image 将用线性插入法加入两个 images 间; 当小于 0.5 倍, 第二个 image 将会移动。

Default: 初始 Images 间的平均路径长度

Format:

0.3 : VarPathLength

▷ *variable* `K_min`

Meaning: 最小弹性常数, 仅用于变弹性常数 VCNEB (单位 eV/Å²)。

Default: 5

Format:

3 : K_min

▷ *variable* `K_max`

Meaning: 最大弹性常数，仅用于变弹性常数 VCNEB (单位 $\text{eV}/\text{\AA}^2$).

Default: 5

Format:

6 : K_max

▷ *variable* `Kconstant`

Meaning: 弹性常数，仅用于固定弹性常数 VCNEB (单位 $\text{eV}/\text{\AA}^2$).

Default: 5

Format:

4 : Kconstant

▷ *variable* `optFreezing`

Meaning: 冻结 Images 结构的选项。当 ConvThreshold 被激活时 Image 结构将被冷冻:

- “0” — 任何时候都不冷冻 Images。
- “1” — 当 ConvThreshold 被激活时冷冻 Image。

Default: 0

Format:

1 : optFreezing

▷ *variable* `optMethodCIDI`

Meaning: Climbing-Image(CI) 和 Descending-Image(DI) 方法的选项。这种方法只有在你已经有一个合理且收敛的路径时才建议使用。CI/DI 方法的选项包括:

- “0” — 不使用 CI/DI 方法;
- “1” — CI 方法，只有最高能量或者用户提供的过渡态 (TS) 会被用于 CI;
- “-1” — DI 方法，只有最低能量或者用户提供的局域最小态 (LM) 会被用于 DI;
- “2” — 混合 CI/DI 方法，需要提供 TS 或者 LM 态的序列号。

Default: 0

Format:

1 : optMethodCIDI

▷ *variable* `startCIDISStep`

Meaning: CI/DI 方法的起始步数，只有当 `optMethodCIDI=1` 的时候才有效

Default: 100

Format:

```
200 : startCIDISStep
```

▷ *variable* `pickupImages`

Meaning: CI/DI 方法中用到的 image 的个数

Default: 过渡态或者局域最小态的 image 的个数

Format:

```
% pickupImages  
9 11 17  
% EndPickupImages
```

注意: 在这种情况下，第 9，第 11，第 17 个 image 会被应用在 CI/DI 方法中。过渡态的 image 和局域最小态的 image 会被自动分别应用于 CI 和 DI 方法。

▷ *variable* `FormatType`

Meaning: 路径输出文件中结构的格式，位于 `results1/PATH/` 路径下。路径结构的输出格式包括：

- “1” — XCRYSDEN 格式 (.xsf 文件)；
- “2” — VASP POSCAR；
- “3” — 晶格的 XYZ 格式。

Default: 2

Format:

```
1 : FormatType
```

▷ *variable* `PrintStep`

Meaning: 每 `PrintStep` 步将 VCNEB 的重启文件保存在 `results1/STEP/` 路径下。

Default: 1

Format:

```
10 : PrintStep
```

注意: 对使用力场软件如 GULP 的用户, 我们建议设置 $\text{PrintStep}=10$ 以减少保存重启文件的时间消耗。

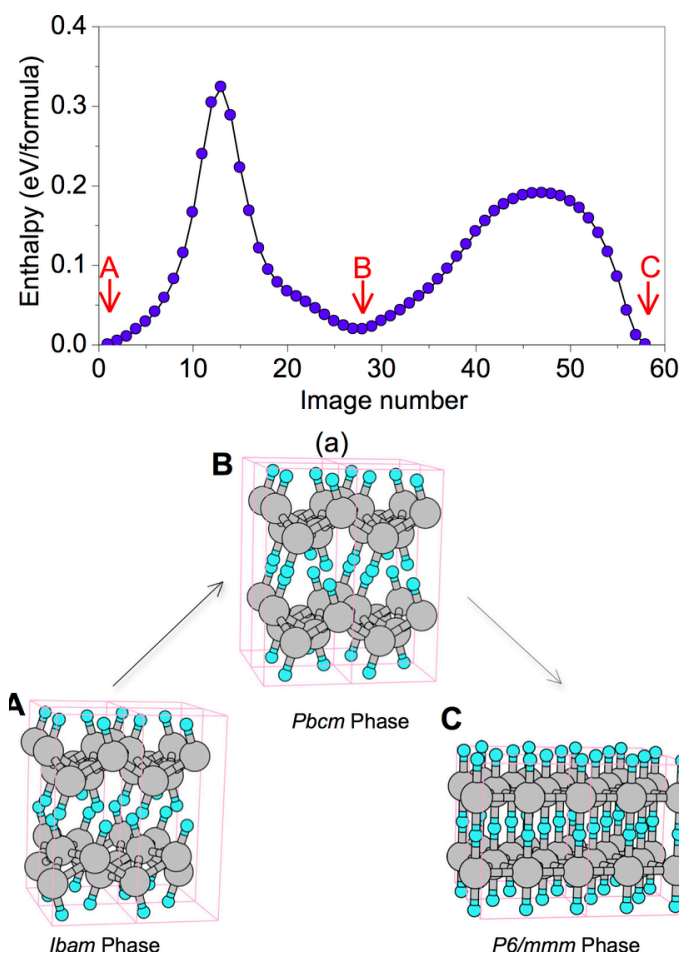


图 16: BH 在 168 GPa 时的 $Ibam \rightarrow P6/mmm$ 相变¹³。出现 Pbcm 中间相。Ibam \rightarrow Pbcm 和 Pbcm \rightarrow P6/mmm 片段上的鞍点的势垒分别为 0.32、0.19 eV/f.u.

图 Fig. 16显示了使用 VCNEB 方法的一个例子: 在 168 GPa 下的 BH 由 Ibam 相转变为 P6/mmm 相的相变机理和能垒, 我们获得了一个 Pbcm 的过渡态。

6.4 如何设置 VC-NEB 计算的初始路径

VCNEB 方法对于寻找相变路径非常有效，但是必须小心地准备初始路径。在 VCNEB 的计算中，晶胞旋转发生在初始和最终的结构附近，这里初始和最终的图像包含许多相同结构。我们使用改进的 `Variable-Image-Number` 方法将会自动阻止晶胞旋转，这会节约很多时间。

另外，你也可以在使用 VCNEB 方法产生初始图像集之前使用避免旋转的技术。一般的具有欧拉角 $R(\phi, \theta, \psi)$ 的 3×3 的旋转矩阵和晶格镜像操作符 $M(x, y, z)$ 矩阵将会被定义。在 VCNEB 运算之前，将会在欧拉角空间和镜像操作符空间中使用全局数值搜索寻找最小晶胞转变距离 Δh :

$$\Delta h = |h_{initial} - R(\phi, \theta, \psi)M(x, y, z)h_{final}|. \quad (10)$$

最后的图像 $\text{image}\tilde{h}_{final}$ 的避免旋转晶格向量将会作为终点的图像:

$$\tilde{h}_{final} = R(\phi, \theta, \psi)M(x, y, z)h_{final}. \quad (11)$$

更重要的是，我们需要阻止在指定初始和最终图像的原子分数坐标 \mathbf{r}_v 时的随机性（正确地反应初始和最终结构的位置）。否则的话，计算将很难收敛或者相同的路径被重复发现，如图 17 中所展示的。对于更复杂的体系，如果你没有一个好的初始路径，你将会得到一些不合理甚至是混乱的路径。对最小化来自两个端点图像的原子间距离的全局数值搜索可以帮助 VCNEB 方法重新分配原子序列。在 VCNEB 计算前自动创建模型路径对于算法的稳定性和收敛性十分关键，是研究大型复杂系统的前提。

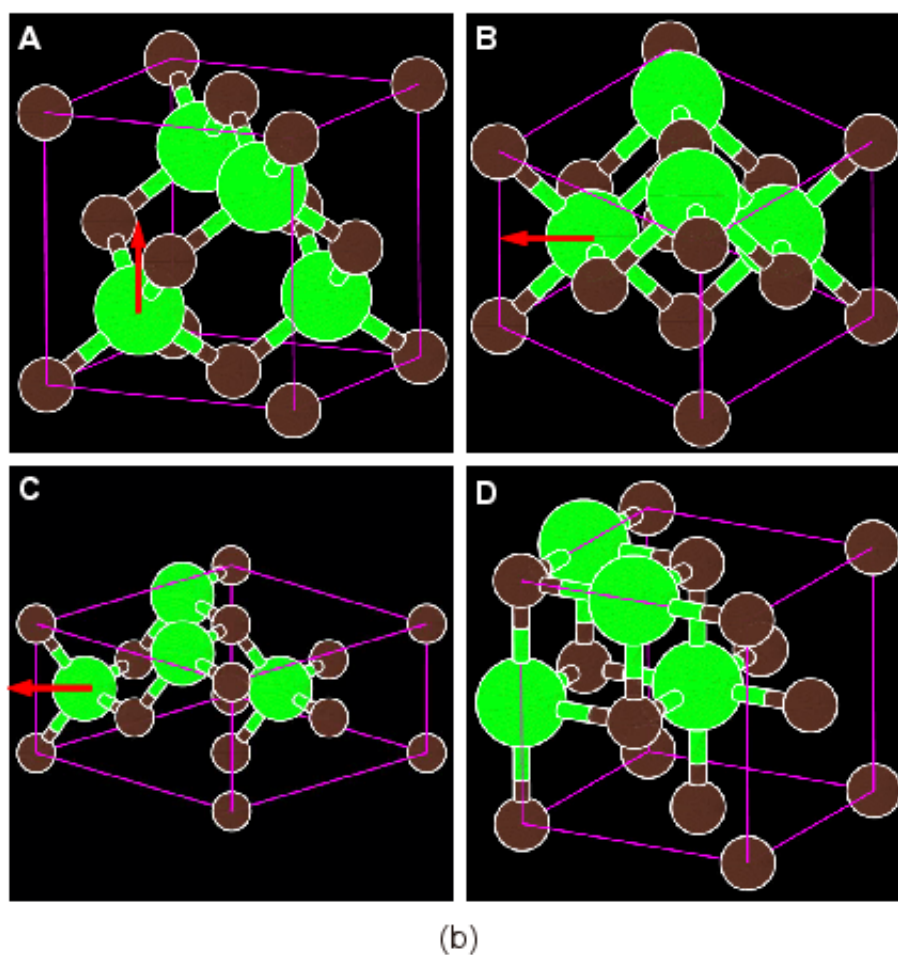
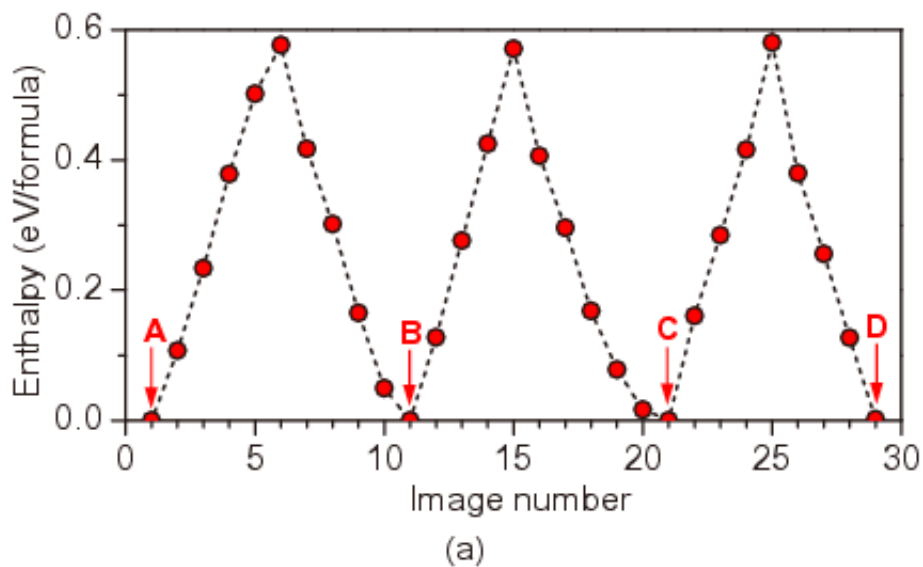


图 17: 当设置“不好”的初始 Image 文件会发现相同的路径。该路径是 GaN 在 45 GPa 静压下 B3→B1 的相变。在 Images-11 和-21, 单斜晶胞中的 B1 和 B3 结构在 MEP 搜索时被发现。Ga 原子在相变过程中沿箭头方向移动。

6.5 过渡态路径取样 (TPS) 方法

过渡态采样 (TPS) 是一种模拟稀有事件的方法: 一个稀有事件的典型例子是一个系统从一个稳态过渡到另一个稳态; 这种情况的发生是由于稀有扰动, 而这往往很难在分子动力学模拟的典型尺度内被观察到。这类例子包括蛋白质的折叠, 化学反应和成核。TPS 是研究这类现象的有效方法。简单地说, TPS 是在过渡轨迹 (每一个都是分子动力学轨迹) 上进行蒙特卡洛采样的一种技术。

6.6 TPS 方法的 Input 设置

目前 TPS 方法值只能和 LAMMPS 和 CP2K 软件一起使用。

为了打开 TPS 模式, 你必须:

1. 设置

```
TPS : calculationMethod
```

2. 在 Seeds/文件夹中创建分子动力学重启文件 `lammps.restart` (for LAMMPS) 或者 `cp2k.restart` (for CP2K)。

3. 创建一个序参数计算脚本或者使用 USPEX 默认的 fingerprints (详细说明见下文)。

4. 指定以下 TPS 选项:

▷ *variable* `numIterations`

Meaning: TPS 步数的最大值.

Default: 1000

Format:

```
500 : numIterations
```

▷ *variable* `speciesSymbol`

Meaning: 描述所有化学种类 (原子或分子) 的特性.

Default: 无默认值

Format:

```
% speciesSymbol  
CH4 Si 0  
% EndSpeciesSymbol
```

注意: 在 TPS 中, 我们用 `speciesSymbol` 和 `numSpecies` 代替 `atomType` 来表示分子组成的种类。比如, 我们可以用:

```
% numSpecies
1 648 1296
% EndNumSpecies

% speciesSymbol
CH4 Si 0
% EndSpeciesSymbol
```

来研究甲烷分子 (CH₄) 在二氧化硅沸石 (SiO₂) 中的扩散。在这个例子中, 甲烷分子将被看做整体写入 LAMMPS 的 MD input 文件中。

▷ *variable* `mass`

Meaning: 每个种类的质量。

Default: 默认值和种类有关

Format:

```
% mass
16.000
% EndMass
```

▷ *variable* `amplitudeShoot`

Meaning: 在 shooting 运算中为 A→B 和 B→A 方向分配的动量的振幅

Default: 0.1 0.1

Format:

```
% amplitudeShoot
0.12 0.12
% EndAmplitudeShoot
```

▷ *variable* `magnitudeShoot`

Meaning: 当获取 MD 轨迹成功或失败时, 分别增加或减少分配的动量振幅的大小。

Default: 1.05 1.05

Format:

```
% magnitudeShoot
1.0 1.05
% EndMagnitudeShoot
```

注意: 除非用户对特殊体系有经验, 否则我们强烈建议用这个选项自动适应动量振幅的分配。

▷ *variable* `shiftRatio`

Meaning: shooter 操作成功后又进行移动操作的比例。

Default: 0.1

Format:

```
0.5 : shiftRatio
```

▷ *variable* `orderParaType`

Meaning: 区分不同相的有序参数计算方法:

- “0” — 用户定义计算有序参数的方法, 需在 `cmdOrderParameter` 提供一个脚本;
- “1” — 指纹法 (Oganov & Valle, 2009).

Default: No default

Format:

```
0 : orderParaType
```

▷ *variable* `opCriteria`

Meaning: 这里有两个值分别说明对与初始和最终态相似的容许程度。如果一个或两个轨迹导致不同的最终结构, 这样的轨迹在 TPS 法中将被遗弃。

Default: 无默认值

Format:

```
% opCriteria
-0.1 0.1
% EndOpCriteria
```

Note: 如果用指纹法, 体系越大, 就该有越严格的环境。例如, 当研究 fcc→hcp 的相变时, 如果有 1000 个原子时, 应设 `opCriteria=[0.995, 0.995]`; 当超过 3000 个原子时, `opCriteria=[0.998, 0.998]`。

▷ *variable* `cmdOrderParameter`

Meaning: 用户定义要求计算有序参数。如果用指纹法时就不用。

Default: 无默认值

Format:

```
% cmdOrderParameter
./extractOp.sh
% EndCmdOrderParameter
```

▷ variable `cmdEnthalpyTempture`

Meaning: 用户自定义命令从 MD 结果提取焓和温度。

Default: No default

Format:

```
% cmdEnthalpyTempture
./extractHT.sh
% EndCmdEnthalpyTempture
```

▷ variable `orderParameterFile`

Meaning: 一个 MD 计算的文件，存储有序参数历史以便 TPS 读取。

Default: fp.dat

Format:

```
op.dat : orderParameterFile
```

▷ variable `enthalpyTemptureFile`

Meaning: 一个 MD 计算的文件，存储焓和稳定历史以便 TPS 读取或用户定义脚本的分析。

Default: HT.dat

Format:

```
HT0.dat : enthalpyTemptureFile
```

▷ variable `trajectoryFile`

Meaning: 存储 MD 轨迹文件。这个文件名需和 MD 轨迹的 output 文件名计算一致。

Default: traj.dat

Format:

```
trajectory.xyz : trajectoryFile
```

▷ variable `MDrestartFile`

Meaning: 存储计算 MD 重启文件以便 TPS 读取。这个名字需和 MD 新建文件在计算中的名字一

致。

Default: traj.restart

Format:

lammps0.restart : MDrestartFile

7 在线工具

我们创建了许多有用的线上工具，可以用于准备 USPEX 输入文件和后处理。这些工具位于：

http://uspex-team.org/online_utilities

下面你可以找到有关每个在线工具的信息。

7.1 结构表征

在这里我们有 5 个工具：

- [Fingerprints](#) — 该工具用于计算和画出指纹函数，一种晶体结构描述符，一种与对关联函数和衍射模式相关的一维函数。它不依赖于原子坐标的绝对值，而仅仅依赖于原子间距离。对原子位置小的扰动不会对指纹函数有明显的影响，它是数值稳定的。
- [Multifingerprint](#) — 该工具计算平均准熵，A-order（平均原子序参量）和 S-order（整个结构的序参量）。它也可以筛选 cosine 距离差大于等于 0.003 的特殊结构，识别这些结构的对称性并将他们列在 `uniq_gatheredPOSCARS` 文件中。
- [POSCAR2CIF](#) — 基于 POSCAR 文件确定结构空间群并生成 CIF 文件。
- [CIF2POSCAR](#) — 基于 CIF 文件生成 POSCAR 文件。
- [XSF2POSCAR](#) — 利用 XSF（XCRYSDEN）文件生成 POSCAR 文件。

7.2 性质计算

这里我们有 2 个工具：

- [Hardness](#) — 这一工具基于 Lyakhov-Oganov 模型计算硬度。
- [EELS](#) — 这一工具用于计算电子能量损失谱，EELS 是 Priya Johari 编写的。

7.3 分子晶体

在这里我们有 2 个工具：

- [MOL precheck](#) — 该工具允许你在运行含有 `calculationType=310/311/110` 的 USPEX 计算器之前查阅 `MOL_1` 文件。
- [Zmatrix](#) — 该工具可将 XYZ 文件转换为 USPEX `MOL_1` 文件。

7.4 表面

[Substrate](#) — 基底——该程序可以结合 POSCAR/CIF 文件，密勒指数，层厚度和偏移生成基底结构。所产生的 POSCAR 文件可用于 `calculationType=200/201` 作为表面计算的基底。

7.5 其它

我们有以下工具:

- [Input generator](#) — USPEXINPUT.txt 生成器。该实用程序可以帮助初学者建立正确的 USPEX 输入文件。
- [Volume estimation](#) — 对于 USPEX (或对于 INPUT.txt 文件) 该功能可以估计非分子晶体与分子晶体的体积。
- [USPEX manual](#) — 本手册的在线版本。
- [USPEX examples](#) — 查阅 USPEX 示例。

8 常见问题

8.1 如何将结果可视化?

USPEX 会产生一大批数据 (结构, 能量), 后处理或者数据分析极其重要。然而, 手动分析这些数据十分无聊而且耗时间。USPEX 受益于由 Mario Valle 特别为 USPEX 研发的界面, 该界面可以使用他的 STM4 可视化工具³³ 来读取并可视化 USPEX 的输出文件, 相关功能包括在数个分钟内便可以分析上千种结构, 决定结构-性质的关联, 分析算法的性能, 量化能量地形图, 最高水平的可视化结构, 决定空间群, 甚至包括生成显示模拟过程的视频! Fig. 18显示了几张典型的由 STM4 生成的图片。为了使用 STM4, 你需要在你的计算机上安装 AVS / Express。AVS / Express 不是公开的域, 需要授权才能使用。STM4 可以<http://mariovalle.name/STM4>获得。

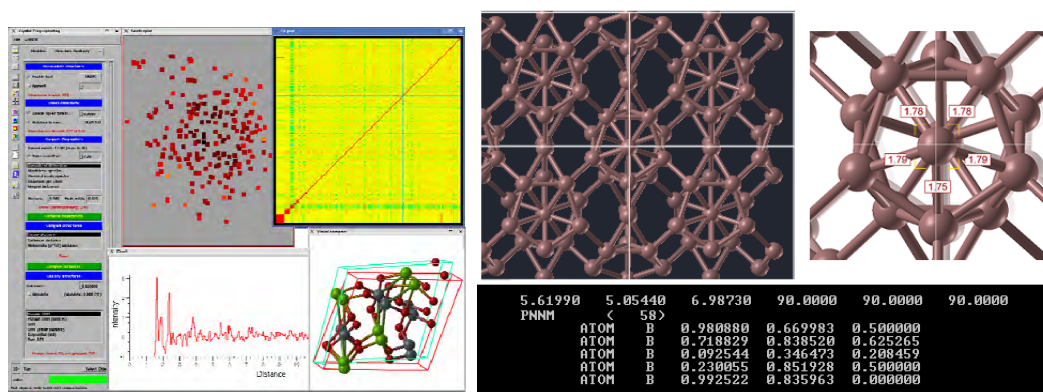


图 18: STM4 interface for USPEX.

另外, 你也可以使用其他软件来可视化 USPEX 的结果, 例如, VESTA, 该软件可以直接查看 USPEX 产生的结构。

8.2 如何避免陷阱?

首先, 使用足够大的族群数。第二, USPEX 默认采用一种非常强大的 fingerprint 方法。任何可以增加族群多样性的手段都可以减少陷于局部最小的风险。为保证你的模拟没有调入陷阱, 使用不同的参数进行第二遍模拟是有用的。其它有效避免陷阱的方法包括反 antiseed 技术。

8.3 什么是单块计算?

单块特征于 USPEX 9.3.9 时被引进, 使得 USPEX 可以对含有不同数量化学式单元的具有相同成分的化合物进行结构预测。例如:

```
% atomType
Si 0
```

```
% EndAtomType

% numSpecies
1 2
% EndNumSpecies

12 : minAt
24 : maxAt
```

这意味着我们采样的化合物 SiO_2 (原子比 1:2) 的结构中包含有 12-24 不等数量的原子。

从 USPEX 9.4.1 开始, 单块功能已经从 `calculationType = 301/311` 变为 `300/310`。其他参数的设置是相同的, 用户只需要设置 `minAt`、`maxAt` 和 `numSpecies` 这些关键参数。目前, 用户可以在 300, 310 和-200 中使用这一特性。

8.4 如何基于已知片段预测结构

有时候你已经知道你的结构是基于一些结构基元的堆叠。你可以告诉程序只根据这些“伪分子”来产生结构。同时, 你仍然可以使用标准的方法来产生子一代结构(如遗传和变异), 标准方法将结构作为原子晶体对待, 因此会打破已预先定义的结构基元。为激活相关程序, 你只需要常规的 `INPUT.txt` 文件,

```
% atomType
B
% EndAtomType

% numSpecices
48
% EndNumSpecices
```

假设你想根据硼二十面体产生结构, 需要一个额外的 `MOL_1` 文件:

```
B12_[4]
Number of atoms: 12
B -1.591325 -0.618615 -0.217220 0 0 0 0
B -0.574110 -0.095870 -1.619670 1 0 0 0
B -1.211325 1.134555 -0.455665 2 1 0 0
B -1.158010 0.414740 1.203810 3 2 1 0
B -0.487865 -1.260560 1.065420 4 3 1 0
B -0.126980 -1.576135 -0.679575 5 4 1 0
```

```

B  0.487845  1.260560 -1.065430  3 2 1  0
B  0.127000  1.576130  0.679585  4 3 1  0
B  0.574120  0.095870  1.619670  5 4 1  0
B  1.211315 -1.134555  0.455685  6 5 1  0
B  1.158015 -0.414735 -1.203800  2 6 1  0
B  1.591320  0.618615  0.217195  7 8 3  0

```

该文件的格式在5.3中有说明。唯一的区别是我们需要在文件的头部写入一些额外信息。如上例所示，该处的信息说明 4 个 MOL_1 (B12) 将会被使用，这与在 INPUT.txt 文件中描述的 48 个 B 原子是一致的。这里的一致性是必须的，否则的话片段功能就不会被激活，即使你放置了 MOL_x 文件。如果你的设置是正确的，你会在 OUTPUT.txt 中找到关于该特征的信息。

对于二维晶体，这种情况下你需要去指明两种厚度。

```
4.0 : thicknessS (it specifies the overall thickness of 2D crystal )
```

```
0.0 : thicknessB (it specifies the thickness of the molecular centers)
```

目前，可以再模块 300 和-200 中使用这一功能。注意你无法同时使用单块体和片段功能。强烈推荐，晶胞参数已知的情况下，在定胞模块中使用这一功能。

8.5 如何使用种子技术？

这种技术是有用的，如果不是从随机结构开始，你会更愿意输入一些你已经知道的化合物或相关材料结构。在 VASP5 格式中的级联 POSCAR 文件的格式如下，为下一代结构创建 Seeds/POSCARS 文件，或为一次 USPEX 计算的特殊代创建 Seeds/POSCARS_gen(gen 指代号)。在此文件名中不要遗忘字母“S”。

例如：

```

EA33 2.69006 5.50602 4.82874 55.2408 73.8275 60.7535 no SG
1.0
2.690100 0.000000 0.000000
2.690100 4.804100 0.000000
1.344900 2.402100 3.967100
Mg Al O
1 2 4
Direct
0.799190 0.567840 0.859590
0.793520 0.230950 0.544750
0.793540 0.916090 0.174450
0.050972 0.816060 0.859610
0.172230 0.194810 0.859600
0.438250 0.655170 0.406880
0.438230 0.202440 0.312330
EA34 7.61073 2.85726 2.85725 60.0001 79.1809 79.1805 no SG

```

```

1.0
7.610700 0.000000 0.000000
0.536350 2.806500 0.000000
0.536330 1.352000 2.459300
Mg Al O
1 2 4
Direct
0.708910 0.507440 0.068339
0.374050 0.285730 0.846630
0.023663 0.069185 0.630090
0.889560 0.780560 0.341460
0.350470 0.626920 0.187820
0.597290 0.211310 0.772210
0.116440 0.371590 0.932500

```

在计算中的任何时候都可以加入种子。每一代开始时 USPEX 将尝试读取种子文件的两个类型。相应的信息将被记录在 `results1/Seeds_history` 中，种子文件（`POSCARS` 或 `POSCARS_gen`）将为被作为 `POSCARS_gen` 保存在种子 `Seeds/`文件中。

当种子加入时，我们建议用户检查 `results1/Seeds_history` 和警告文件。如果你的种子有问题时，可能会有一个警告信息“Meet a problem when reading Seeds - ...”。当一个错误出现在种子文件，比如是缺失了一行，错误出现后的结构将不会被添加进去。

注意：确保你指定的所有原子符号在每一结构的 6 号线位置。例如，添加 $P6_3/mc$ H_2 结构到 H-O 变成分计算，你要编辑的文件：

```

H_I-P63/mc
1
4.754726 -2.74514 0.000000
-0.00000 5.490285 0.000000
0.000000 0.000000 4.508715
H
16
Direct
...
(the atomic positions information is omitted here)

```

8.6 如何控制成分？

对于可变组分和单块计算，当计算开始时，它便会生成一个带有所有可能组成的种子文件 `Seeds/compositions`，其中遗传代码从随机结构发生器中随机选取组成。你可以编辑该文件，只留下你最感兴趣的成分——只有这些成分才会被用于在接下来代中的随机结构产生。`Seeds/compositions` 文件中列出了每个胞中每种原子的数量，例如，对于 C-O 体系：

```

8 0
0 8
2 4

```

意味着你会对随机产生 C_8 、 O_8 和 C_2O_4 结构感兴趣。由于遗传变异和原子置换，其他的成分也将被采样。

当想要产生含有指定组成的结构时，可以使用 `anticompositions` 的特点—将所有不想要的组成输入至 `Seeds/Anti-compositions` 文件中。主要有 3 种可行方式：

1. 对于所有同一比例不需要的成分，可以写下某一成分比以禁止某些成分。例如，可以使用 “1 2 0” 禁止所有的以相同的比例组成的成分，如 “1 2 1”、“2 4 2” 和 “3 6 3” 等。
2. 只为特定的成分，而不是其他具有同一成分比的组成。可以用一个负号标识组成。例如，可以使用 “-3 2 0” 或 “3 -2 0” 禁止 “3 2 0” 的成分，但不禁止 “6 4 0” 或 “9 6 0” 组成。（注意：“3 2 -0” 这种情况下无意义）。
3. 对于所有的单一/二元/三元化合物，如果你不想对所有的单一/二元/三元化合物进行采样，请在 `Anti-compositions` 文件中写下关键词 `single/binary/ternary`。

例如：

```
single
binary
1 1 2
-2 2 1
```

如果你不太确定你正在做什么，请将 `Anti-compositions` 文件置空。更多关于你不想要组分的信息，你可以查看下 `results1/compositionStatistic`。

注意：

- 即使在计算开始之前 `compositions` 或 `Anti-compositions` 文件已存在，他们会被忽略。`Anti-compositions` 文件会被重命名为 `Anti-compositions-back` 备份文件。因此，在计算开始之后请编辑 `compositions` 或 `Anti-compositions` 文件。
- 也请注意，在带有组合块的 USPEX 计算中，组合物通常意味着这些块的数量。因此，为了确保 `Anti-compositions` 文件的格式正确，请首先检查组成 `compositions` 文件。

8.7 如何从本地机器向远程集群设置免密连接？

有两种方式来解决这个问题：

1. 免密 SSH 登录。（注意，本小结基于文章 http://linuxproblem.org/art_9.html）

你的目标：你想使用 OpenSSH 实现自动作业提交，因此你需要一个从 `hostA` 或者 `userA` 自动登录到

hostB 或者 userB。在这个过程中你不想输入任何密码，因为你想要在 shell 脚本中调用 SSH。

如何去做：首先以 userA 的身份登录 hostA，接着产生一对验证密钥。不需要输入密码：

```
userA@hostA:~> ssh-keygen -t rsa Generating public/private rsa key
pair.
Enter file in which to save the key (/home/userA/.ssh/id_rsa):
Created directory '/home/userA/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/userA/.ssh/id_rsa.
Your public key has been saved in /home/userA/.ssh/id_rsa.pub.
The key fingerprint is:
3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 userA@hostA
```

然后以 userB 的身份登录 hostB，使用 ssh 去产生一个目录（目录可能本身已经存在，但是没关系）

```
userA@hostA:~> ssh -p portB userB@hostB 'mkdir -p .ssh' userB@hostB's
password:
```

最后将 A 的新公共密钥附加给 hostB 的 userB：.ssh/authorized_keys 同时最后一次输入 B 的密码：

```
userA@hostA:~> cat .ssh/id_rsa.pub | ssh -p portB userB@hostB 'cat >>
.ssh/authorized_keys'.
userB@hostB's password:
```

从现在开始，你就可以避免从 hostA 里 userA 的身份登录 hostB 里 userB 时输入密码。

```
userA@hostA:~> ssh -p portB userB@hostB
```

2. 你需要将你本地机器（目录 ./ssh 或 ./ssh2）的公共密钥拷贝到远程集群上，这里是你需要执行的命令列表：

```
local # ssh-keygen -t dsa
local # scp ~/.ssh2/id_dsa.pub oganov@palu.cscs.ch:~/.ssh/tmp.pub
remote # cd ~/.ssh/
remote # ssh-keygen -f tmp.pub -i >> authorized_keys
remote # rm tmp.pub
```

8.8 如何创建一个计算使用的工作提交脚本？

为了创建作业提交脚本，我们希望用户了解一些基本的 python 编程知识和你的作业提交系统。

有两种工作脚本提交模式：**本地提交**或**远程提交**，这取决于你是将 *ab initio* 计算提交到你运行 USPEX 和 MATLAB 的本地机器还是远程超级计算机。

8.8.1 步骤 1: 配置 Submission/文件夹中的文件

案例一：本地提交。

请在 INPUT.txt 文件中编辑以下条目：

```
1 : whichCluster (0: no-job-script, 1: local submission, 2: remote submission)
```

然后，需要在你的本地机器上运行 ssh 服务器。USPEX 会连接它并通过 ssh 运行 ab-initio 代码。然后到 Submission/文件夹下，这里你需要编辑 submitJob_local.py 和 checkStatus_local.py 这两个文件。

你可以在这些文件中找到详细的说明。在一般情况下，你只需告诉 USPEX 如何提交工作和检查工作是否已经完成。

文件 submitJob_local.py 如下：

```
from subprocess import check_output
import re
import sys

def submitJob_local(index : int, commnadExecutable : str) -> int:
    """
    This routine is to submit job locally
    One needs to do a little edit based on your own case.

    Step 1: to prepare the job script which is required by your supercomputer
    Step 2: to submit the job with the command like qsub, bsub, llsubmit, .etc.
    Step 3: to get the jobID from the screen message
    :return: job ID
    """

    # Step 1
    myrun_content = ''
    myrun_content += '#!/bin/sh\n'
    myrun_content += '#SBATCH -o out\n'
    myrun_content += '#SBATCH -p cpu\n'
    myrun_content += '#SBATCH -J USPEX-' + str(index) + '\n'
    myrun_content += '#SBATCH -t 06:00:00\n'
    myrun_content += '#SBATCH -N 1\n'
```

```
myrun_content += '#SBATCH -n 8\n'
# myrun_content += 'cd ${PBS_O_WORKDIR}\n' check this, must have /cephfs
# suffix with SBATCH in my case
myrun_content += 'mpirun vasp_std > log\n'
with open('myrun', 'w') as fp:
    fp.write(myrun_content)

# Step 2 # It will output some message on the screen like
# '2350873.nano.cfn.bnl.local'
output = str(check_output('sbatch myrun', shell = True))

# Step 3
# Here we parse job ID from the output of previous command
jobNumber = int(re.findall(r'\d+', output)[0])
return jobNumber

if __name__ == '__main__':
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('-i', dest='index', type=int)
    parser.add_argument('-c', dest='commnadExecutable', type=str)
    args = parser.parse_args()

    jobNumber=submitJob_local(index=args.index,commnadExecutable= \
        args.commnadExecutable)
    print('CALLBACK ' + str(jobNumber))
```

文件 checkStatus_local.py 如下:

```
import argparse
import glob
import os

from subprocess import check_output

_author_ = 'etikhonov'
```



```

def checkStatus_local(jobID : int) -> bool:
    """
    This function is to check if the submitted job is done or not
    One needs to do a little edit based on your own case.
    1 : whichCluster (0: no-job-script, 1: local submission, 2: remote submission)
    Step1: the command to check job by ID.
    Step2: to find the keywords from screen message to determine if the job is done
    Below is just a sample:
    -----
    Job id                Name                User                Time Use S Queue
    -----
    2455453.nano          USPEX                qzhu                02:28:42 R cfn_gen04
    -----

    If the job is still running, it will show as above.

    If there is no key words like 'R/Q Cfn_gen04', it indicates the job is done.
    :param jobID:
    :return: doneOr
    """

    # Step 1
    output = str(check_output('qstat {}'.format(jobID), shell=True))
    # Step 2
    doneOr = True
    if ' R ' in output or ' Q ' in output:
        doneOr = False
    if doneOr:
        for file in glob.glob('USPEX*'):
            os.remove(file) # to remove the log file
    return doneOr

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument('-j', dest='jobID', type=int)
    args = parser.parse_args()

    isDone = checkStatus_local(jobID=args.jobID)

```

```
print('CALLBACK ' + str(int(isDone)))
```

案例二：远程提交。

在文件 INPUT.txt 编辑：

```
2      : whichCluster (default 0, 1: local submission; 2: remote submission)
```

然后，进入 Submission/文件夹，修改以下两个文件：

submitJob_remote.py and checkStatus_remote.py

文件 submitJob_remote.py 如下：

```
import argparse
import os
import re

from subprocess import check_output

def submitJob_remote(workingDir : str, index : int, commandExecutable : str) -> int:
    """
    This routine is to submit job to remote cluster
    One needs to do a little edit based on your own case.
    Step 1: to prepare the job script which is required by your supercomputer
    Step 2: to submit the job with the command like qsub, bsub, llsubmit, .etc.
    Step 3: to get the jobID from the screen message

    :param workingDir: working directory on remote machine
    :param index: index of the structure.
    :param commandExecutable: command executable for current step of optimization
    :return:
    """

    # Step 1
    # Specify the PATH to put your calculation folder
    Home = '/home/etikhonov' # 'pwd' of your home directory of your remote machine
    Address = 'rurik' # your target server: ssh alias or username@address
    Path = Home + '/' + workingDir + '/CalcFold' + str(index) # Just keep it
```

```
run_content = ''
run_content += '#!/bin/sh\n'
run_content += '#SBATCH -o out\n'
run_content += '#SBATCH -p cpu\n'
run_content += '#SBATCH -J USPEX-' + str(index) + '\n'
run_content += '#SBATCH -t 06:00:00\n'
run_content += '#SBATCH -N 1\n'
run_content += '#SBATCH -n 8\n'
run_content += 'cd /cephfs'+ Path + '\n'
run_content += commandExecutable + '\n'

with open('myrun', 'w') as fp:
    fp.write(run_content)

# Create the remote directory
# Please change the ssh/scp command if necessary.
try:
    os.system('ssh -i ~/.ssh/id_rsa ' + Address + ' mkdir -p ' + Path)
except:
    pass

# Copy calculation files
# add private key -i ~/.ssh/id_rsa if necessary
os.system('scp POSCAR ' + Address + ':' + Path)
os.system('scp INCAR ' + Address + ':' + Path)
os.system('scp POTCAR ' + Address + ':' + Path)
os.system('scp KPOINTS ' + Address + ':' + Path)
os.system('scp myrun ' + Address + ':' + Path)

# Step 2
# Run command
output = str(check_output('ssh -i ~/.ssh/id_rsa ' + Address + ' qsub ' \
+ Path + '/myrun', shell=True))

# Step 3
# Here we parse job ID from the output of previous command
jobNumber = int(re.findall(r'\d+', output)[0])
return jobNumber
```

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument('-i', dest='index', type=int)
    parser.add_argument('-c', dest='commnadExecutable', type=str)
    parser.add_argument('-f', dest='workingDir', type=str)
    args = parser.parse_args()

    jobNumber = submitJob_remote(workingDir=args.workingDir, index=args.index, \
    commnadExecutable=args.commnadExecutable)
    print('CALLBACK ' + str(jobNumber))
```

文件 `checkStatus_remote.py` 如下:

```
import argparse
import os

from subprocess import check_output

def checkStatus_remote(jobID : int, workingDir : str, index : int) -> bool:
    """
    This routine is to check if the submitted job is done or not
    One needs to do a little edit based on your own case.
    Step1: Specify the PATH to put your calculation folder
    Step2: Check JobID, the exact command to check job by jobID
    :param jobID:
    :param index:
    :param workingDir:
    :return:
    """
    # Step 1
    Home = '/home/etikhonov' # 'pwd' of your home directory of your remote machine
    Address = 'rurik' # Your target supercomputer: username@address or ssh alias
    # example of address: user@somedomain.edu -p 2222
    Path = Home + '/' + workingDir + '/CalcFold' + str(index) # just keep it

    # Step 2
    output = str(check_output('ssh ' + Address + ' qstat ' + str(jobID), shell=True))
    # If you using full adress without ssh alias,
    # you must provide valid ssh private key like there:
    # output = str(check_output('ssh -i ~/.ssh/id_rsa ' + Address + \
    # ' /usr/bin/qstat ' + str(jobID), shell=True))

    if not ' R ' in output or not ' Q ' in output:
        doneOr = True
        # [nothing, nothing] = unix(['scp -i ~/.ssh/id_rsa ' Address ':' \
        # Path '/OUTCAR ./']) %OUTCAR is not necessary by default
        # For reading enthalpy/energy
        os.system('scp ' + Address + ':' + Path + '/OSZICAR ./')
```

```
    # For reading structural info
    os.system('scp ' + Address + ':' + Path + '/CONTCAR ./')
    # Edit ssh command as above!
else:
    doneOr = False
return doneOr

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument('-j', dest='jobID', type=int)
    parser.add_argument('-i', dest='index', type=int)
    parser.add_argument('-f', dest='workingDir', type=str)
    args = parser.parse_args()

    isDone = checkStatus_remote(jobID=args.jobID, \
        workingDir=args.workingDir, index=args.index)
    print('CALLBACK ' + str(int(isDone)))
```

9 附录

9.1 例子目录

- EX01-3D_Si_vasp: 零压力下的硅（每个单胞中含 8 个原子）。使用 VASP, PBE96 泛函进行变胞 DFT 计算。感谢 G. Kresse 和他的 PAW 文件 (POTCAR) 对我们的贡献。
- EX02-3D_MgAl2O4_gulp: 在 100 GPa 下的 MgAl_2O_4 （每一个单胞中有 28 个原子）。使用 Buckingham 势, GULP 代码进行变胞计算。如果需要很可靠的结果, 最好用 *ab initio* 算法。
- EX03-3D-const_cell_MgSiO3_gulp: 这个例子向我们展示在知道单胞参数时如何进行结构预测。含有 Buckingham 势, GULP 代码的 MgSiO_3 （每个单胞有 20 个原子）。晶胞参数与后钙钛矿一致。后钙钛矿的发现 (*Oganov & Ono, Nature 2004; Murakami et al., Science 2004*) 在地球科学领域是一个重大突破。
- EX04-3D_C_lammps: 这个例子展示了如何使用结合了 LAMMPS 码的 USPEX 进行晶体结构预测。在这个简单的例子中: 有 8 个碳原子, 采用 Tersoff 势。
- EX05-3D_Si_atk: 这是一个使用密度泛函紧束缚近似和 ATK 代码对每一单胞含 8 个原子的硅进行晶体结构预测的例子。
- EX06-3D_C_castep: 在 0 GPa 下, 使用 CASTEP 代码对每个单胞含 8 个原子的碳的晶体结构进行基于 DFT 的预测。
- EX07-2D_Si_vasp: 使用 DFT 和 VASP 预测硅的二维晶体。简单而有效。
- EX08-0D_LJ_gulp: 纳米结构预测。含有 30 个原子的 Lenard-Jones 纳米颗粒, GULP 代码。
- EX09-3D-molecules_CH4_vasp: 甲烷分子晶体预测, 压力 20 GPa, DFT, VASP。分子在文件 MOL_1 中描述。
- EX10-3D-molecules_CH4_dmacrys: 常压, 带有 8 个分子的单胞, 采用力场和 DMACRYS 代码。在文件 MOL_1 中描述分子, 但要注意其 DMACRYS 计算略有不同的格式。请把可执行文件 dmacrys、neighcrys-pp 和 neighcrys-vv 放在文件夹 Specific/。
- EX11-3D-molecules_urea_tinker: 常压, 2 个分子的单胞, 采用力场和 TINKER 代码。在文件 MOL_1 中描述分子。
- EX12-3D_special_quasirandom_structure_TiCoO: 含有伪 “Mo” 和 “B” 原子, GULP 和变组成 USPEX (*Lyakhov and Oganov, 2010*) 的 Lenard-Jones 二元体系。
- EX13-3D_special_quasirandom_structure_TiCoO: USPEX 可以很容易地找到最无序 (或最有序) 合金结构。这里表示为 $\text{Ti}_x\text{Co}_{(1-x)}\text{O}$ 。你需要在 Seeds/POSCARS 指定初始结构和仅使用置换操作。在这种情况下, 不需要使用任何外部代码。在这个例子中, 我们在无弛豫的情况下 (`abinitioCode = 0`) 优化 (最小化) 结构有序度 (*Oganov and Valle (2009); Lyakhov, Oganov,*

Valle (2010))。种子结构 (Ti-Co-O 结构的超晶胞) 被置换找到结构的最小/最大化有序度。在这种情况下最小化有序度, 可以得到“特殊准随机结构”的通用版本。

- **EX14-GeneralizedMetadynamics_Si_vasp**: 关于开始用一个简单的种子结构搜索复杂低能量结构的简单例子 (Zhu et al, 2013)。硅, 最多为 16 个原子的单胞, DFT, VASP。要特别注意到 INCAR 文件。最重要的是, 只要保留你在这里看到的文件, 只改变 ENCUT, SIGMA, 进化准动力学不仅预测低能量的结构, 同时也给出了晶体结构之间的转换机制。
- **EX15-VCNEB_Ar_gulp**: 氩气, 在 0 GPa 压力模型系统中采用变胞微动弹性带 (VCNEB: Qian et al., 2013) 计算 fcc-hcp 转变。Lenard-Jones 势, GULP 代码。
- **EX16-USPEX-performance_SrTiO3_gulp**: SrTiO₃ (50 原子/单胞) 在零压力下。使用 Buckingham 势, GULP 代码进行可变单胞计算。从这个例子中可以看到, 即使是这样一个比较大的系统, USPEX 代码仍有 >90% 的成功率和显著的效率。与此相反, 通过 Zurek 和 Lonie 使用相同的体系, 相同的势能报道的成功率只有 7-12%。显然, USPEX 优于我们通过 Zurek 和 Lonie 的实现方法。我们已经见证了我们的代码的性能在更大的系统也很优异。
- **EX17-3D_DebyeTemp_C_vasp**: 对弹性相关性能优化的例子 (体积和剪切模量, 泊松比, Chen-Niu 硬度, 德拜温度) 在这个例子中, 我们使用 VASP 代码最大限度地提高碳的德拜温度。
- **EX18-3D_varcomp_ZnOH_gulp**: 正如你所知道的, USPEX 对于变成分的计算具有独特的能力。这个例子为我们展现了一个非常具有挑战性的情况—对于三元体系 Zn-O-H 的变成分计算。这个计算在 GULP 代码中使用 ReaxFF 力场。USPEX 可以为任何成分组成进行计算—例如, 在其它范围内可实现的四元和五元体系。当然, 成分越多, 计算越昂贵 (和越有风险)。目前没有任何参考结果。
- **EX19-Surface-boron111**: 预测 alpha-B 的 (111) 表面重构, 采用变原子数 (Zhou et al., Phys. Rev. Lett. 113, 176101 (2014))。
- **EX20-0D-Cluster-C60-MOPAC**: 使用 MOPAC 进行 C₆₀ 团簇结构的预测 (000)。
- **EX21-META_MgO_gulp**: 进化准动力学方法, 采用 GULP 代码和 Buckingham 势, 用含有 8 个原子的 MgO 初始结构是盐岩型结构, 进化准动力学方法找到许多能量低的结构和结构之间的关系。
- **EX22-GEM_MgO_gulp**: 通用进化准动力学方法, 采用 GULP 代码和 Buckingham 势, 用含有 8 个原子的单胞, 这个计算是允许增加到 16 个原子, 通用进化准动力学方法找到许多能量低的结构和结构之间的关系。
- **EX23-MgO-Surface**: 预测 MgO 的表面重构结构。
- **EX24-SingleBlock_Magnetic_Fe3C_VASP**: 结合 VASP 预测定成分预测三维 Fe₃C 磁性结构的预测。

- **EX25-3D-C8-DFTB**: 基于 3ob-3-1 基组, 结合 DFTB 预测 C 结构 (8 原子/晶胞)。在当前参数设置下, 石墨和金刚石的能量差似乎被高估了。
- **EX26-Ar-TPS**: 在 1 个标准大气压及 40 K 下, 利用 TPS 模拟 hcp-和 fcc-Ar 固体间的相变, Ar 结构中含有 8000 原子。
- **EX27-3D-P2-FHIaims**: 结合 FHIaims 定成分预测三维 P 结构 (2 原子/晶胞)。
- **EX28-0D-Cluster-Cu9-FHIaims**: 结合 FHIaims 预测零维 Cu9 团簇结构。
- **EX29-Si-gap-maximize-singleblock**: 基于 meta-GGA 泛函, 预测具有最大带隙的三维 Si 结构。
- **EX30-Pd-111-oxidation**: 在 Pd(111) 表面上预测含氧重构结构。
- **EX31-varcomp-SnS-VASP**: 变组分预测二维 Sn-S 稳定结构。
- **EX32-pmpaths**: 五种不同的金刚石-石墨转换路径, 由 pmpaths 产生。
- **EX33-carbon-QE**: 利用 Quantum Espresso 辅助预测具有最低能量的碳结构 (8 原子/晶胞), 计算模块 (300)。
- **EX34-Glycine-Molecular-QE**: 利用 Quantum Espresso 辅助预测甘氨酸分子晶体 (N 分子/晶胞), 计算模块 (310)。

9.2 测试运行

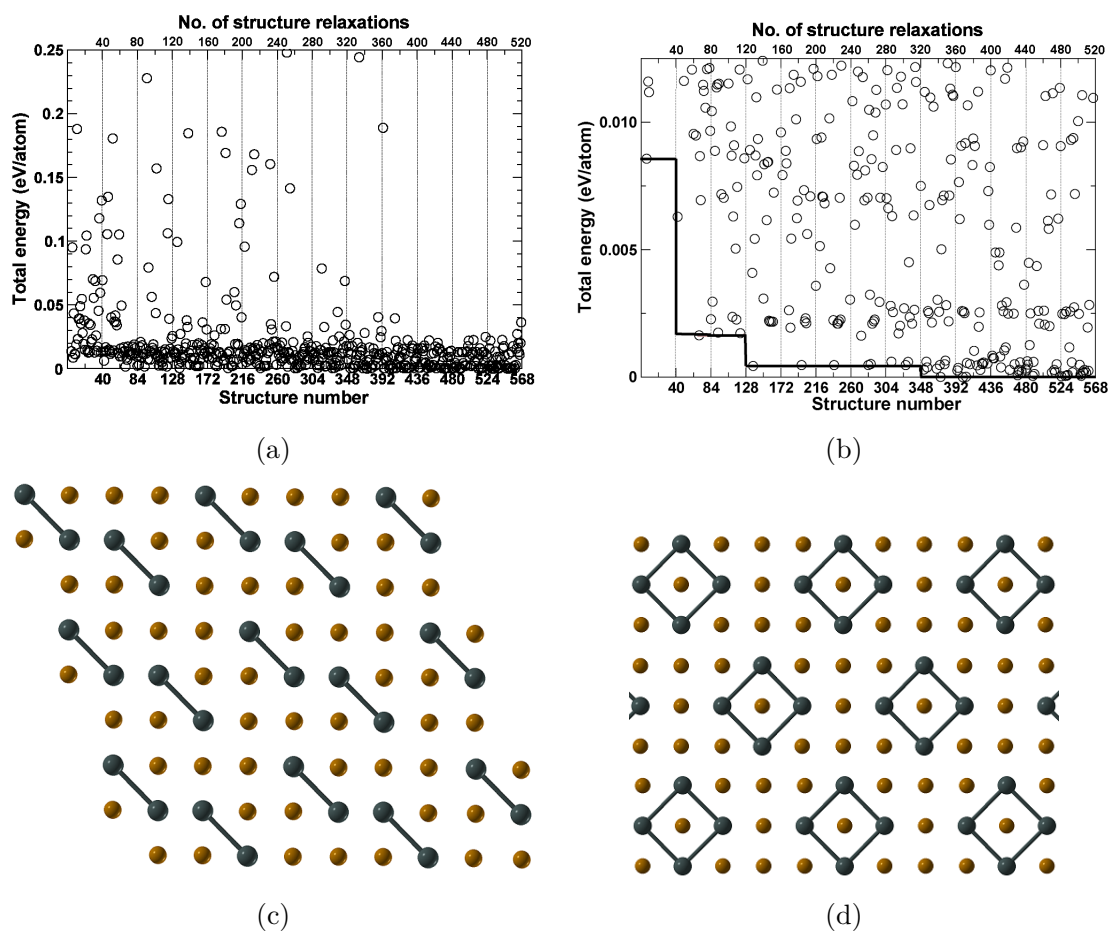


图 19: 对 Au_8Pd_4 进化结构搜索。a, b—总能量进化 (为了清晰, (b) 可以在同一数据集能量最低的区域急剧上升), c—在我们的进化模拟中发现的能量最低结构, d—利用准噶尔集群扩展的计算发现的能量最低结构。请注意我们的结构 (c) 是这种化合物已知的最低能量结构。这建立了我们方法的权威性 (即使是旧的 2007 版)

9.3 INPUT.txt 输入文件示例

9.3.1 定成分预测 (calculationType=300):

```
1 PARAMETERS EVOLUTIONARY ALGORITHM
2 % Example of the short input, using most options as defaults
3
4 % atomType
5 Mg Al O
6 % EndAtomType
7
8 % numSpecies
9 2 4 8
10 % EndNumSpecies
11
12 50 : numGenerations
13 50.0 : ExternalPressure
14
15 % abinitioCode
16 3 3 3 3 3
17 % ENDabinit
18
19 % commandExecutable
20 gulp < input > output
21 % EndExecutable
```

9.3.2 变成分预测 (calculationType=301):

```
1 USPEX : calculationMethod (USPEX, VCNEB, META)
2 301   : calculationType (dimension: 0-3; molecule: 0/1; varcomp: 0/1)
3 1     : AutoFrac
4
5 % atomType
6 Mo B
7 % EndAtomType
8
9 % numSpecies
10 1 0
11 0 1
12 % EndNumSpecies
13
14 80    : populationSize
15 200   : initialPopSize
16 60    : numGenerations
17 20    : stopCrit
18
19 11    : firstGeneMax
20 8     : minAt
21 18    : maxAt
22
23 % abinitioCode
24 3 3 3
25 % ENDabinit
26
27 % commandExecutable
28 gulp < input > output
29 % EndExecutable
```

9.3.3 进化赝动力学模拟 (calculationMethod=META):

```
1 META : calculationMethod (USPEX, VCNEB, META)
2 301 : calculationType (dimension: 0-3; molecule: 0/1; varcomp: 0/1)
3
4 % valences
5 4
6 % endValences
7
8 % IonDistances
9 1.2
10 % EndDistances
11
12 0.0001 : ExternalPressure
13
14 16 : maxAt
15 2.0 : minVectorLength
16 8.0 : maxVectorLength
17
18 15 : populationSize
19 40 : numGenerations
20 3.0 : mutationDegree
21 250.0 : GaussianHeight
22 0.3 : GaussianWidth
23 2 : FullRelax
24
25 abinitioCode
26 1 1 1 (1 1)
27 ENDabinit
28
29 % KresolStart
30 0.12 0.10 0.09 0.10 0.08
31 % Kresolend
32
33 % commandExecutable
34 mpirun -np 4 vasp > log
35 % EndExecutable
```

9.3.4 VCNEB 模拟 (calculationMethod=VCNEB):

```
1 VCNEB : calculationMethod
3 % numSpecies
4
5 % EndNumSpecies
7 % atomType
8 Ar
9 % EndAtomType
11 0.0 : ExternalPressure
13 111 : vcnebType
14 15 : numImages
15 500 : numSteps
16 1 : optimizerType
17 2 : optReadImages
18 3 : optRelaxType
19 0.25 : dt
20 0.003 : ConvThreshold
21
22 0.3 : VarPathLength
23 3 : K_min
24 6 : K_max
25 0 : optFreezing
26 0 : optMethodCIDI
27
28 2 : FormatType
29 10 : PrintStep
31 abinitioCode
32 3
33 ENDabinit
35 % commandExecutable
36 gulp < input > output
37 % EndExecutable
```

9.4 空间群列表

1	<i>P1</i>	2	<i>P-1</i>	3	<i>P2</i>	4	<i>P2₁</i>
5	<i>C2 (A2)*</i>	6	<i>Pm</i>	7	<i>Pc (Pa)*</i>	8	<i>Cm (Am)*</i>
9	<i>Cc (Aa)*</i>	10	<i>P2/m</i>	11	<i>P2₁/m</i>	12	<i>C2/m (A2/m)*</i>
13	<i>P2/c (P2/a)*</i>	14	<i>P2₁/c (P2₁/a)*</i>	15	<i>C2/c (A2/a)*</i>	16	<i>P222</i>
17	<i>P222₁</i>	18	<i>P2₁2₁2</i>	19	<i>P2₁2₁2₁</i>	20	<i>C222₁</i>
21	<i>C222</i>	22	<i>F222</i>	23	<i>I222</i>	24	<i>I2₁2₁2₁</i>
25	<i>Pmm2</i>	26	<i>Pmc2₁</i>	27	<i>Pcc2</i>	28	<i>Pma2</i>
29	<i>Pca2₁</i>	30	<i>Pnc2</i>	31	<i>Pmn2₁</i>	32	<i>Pba2</i>
33	<i>Pna2₁</i>	34	<i>Pnn2</i>	35	<i>Cmm2</i>	36	<i>Cmc2₁</i>
37	<i>Ccc2</i>	38	<i>Amm2 (C2mm)*</i>	39	<i>Aem2 (C2mb)*</i>	40	<i>Ama2 (C2cm)*</i>
41	<i>Aea2 (C2cb)*</i>	42	<i>Fmm2</i>	43	<i>Fdd2</i>	44	<i>Imm2</i>
45	<i>Iba2</i>	46	<i>Ima2</i>	47	<i>Pmmm</i>	48	<i>Pnnn</i>
49	<i>Pccm</i>	50	<i>Pban</i>	51	<i>Pmma</i>	52	<i>Pnna</i>
53	<i>Pmna</i>	54	<i>Pcca</i>	55	<i>Pbam</i>	56	<i>Pccn</i>
57	<i>Pbcm</i>	58	<i>Pnmm</i>	59	<i>Pmmn</i>	60	<i>Pbcn</i>
61	<i>Pbca</i>	62	<i>Pnma</i>	63	<i>Cmcm</i>	64	<i>Cmce (Cmca)*</i>
65	<i>Cmmm</i>	66	<i>Cccm</i>	67	<i>Cmme (Cmma)*</i>	68	<i>Ccce (Ccca)*</i>
69	<i>Fmmm</i>	70	<i>Fddd</i>	71	<i>Immm</i>	72	<i>Ibam</i>
73	<i>Ibca</i>	74	<i>Imma</i>	75	<i>P4</i>	76	<i>P4₁</i>
77	<i>P4₂</i>	78	<i>P4₃</i>	79	<i>I4</i>	80	<i>I4₁</i>
81	<i>P-4</i>	82	<i>I-4</i>	83	<i>P4/m</i>	84	<i>P4₂/m</i>
85	<i>P4/n</i>	86	<i>P4₂/n</i>	87	<i>I4/m</i>	88	<i>I4₁/a</i>
89	<i>P422</i>	90	<i>P42₁2</i>	91	<i>P4₁22</i>	92	<i>P4₁2₁2</i>
93	<i>P4₂22</i>	94	<i>P4₂2₁2</i>	95	<i>P4₃22</i>	96	<i>P4₃2₁2</i>
97	<i>I422</i>	98	<i>I4₁22</i>	99	<i>P4mm</i>	100	<i>P4bm</i>
101	<i>P4₂cm</i>	102	<i>P4₂nm</i>	103	<i>P4cc</i>	104	<i>P4nc</i>
105	<i>P4₂mc</i>	106	<i>P4₂bc</i>	107	<i>I4mm</i>	108	<i>I4cm</i>
109	<i>I4₁md</i>	110	<i>I4₁cd</i>	111	<i>P-42m</i>	112	<i>P-42c</i>
113	<i>P-42₁m</i>	114	<i>P-42₁c</i>	115	<i>P-4m2</i>	116	<i>P-4c2</i>
117	<i>P-4b2</i>	118	<i>P-4n2</i>	119	<i>I-4m2</i>	120	<i>I-4c2</i>
121	<i>I-42m</i>	122	<i>I-42d</i>	123	<i>P4/mmm</i>	124	<i>P4/mcc</i>
125	<i>P4/nbm</i>	126	<i>P4/nnc</i>	127	<i>P4/mbm</i>	128	<i>P4/mnc</i>
129	<i>P4/nmm</i>	130	<i>P4/ncc</i>	131	<i>P4₂/mmc</i>	132	<i>P4₂/mcm</i>
133	<i>P4₂/nbc</i>	134	<i>P4₂/nmm</i>	135	<i>P4₂/mbc</i>	136	<i>P4₂/mnm</i>
137	<i>P4₂/nmc</i>	138	<i>P4₂/ncm</i>	139	<i>I4/mmm</i>	140	<i>I4/mcm</i>
141	<i>I4₁/amd</i>	142	<i>I4₁/acd</i>	143	<i>P3</i>	144	<i>P3₁</i>
145	<i>P3₂</i>	146	<i>R3</i>	147	<i>P-3</i>	148	<i>R-3</i>
149	<i>P312</i>	150	<i>P321</i>	151	<i>P3₁12</i>	152	<i>P3₁21</i>
153	<i>P3₂12</i>	154	<i>P3₂21</i>	155	<i>R32</i>	156	<i>P3m1</i>
157	<i>P31m</i>	158	<i>P3c1</i>	159	<i>P31c</i>	160	<i>R3m</i>
161	<i>R3c</i>	162	<i>P-31m</i>	163	<i>P-31c</i>	164	<i>P-3m1</i>
165	<i>P-3c1</i>	166	<i>R-3m</i>	167	<i>R-3c</i>	168	<i>P6</i>
169	<i>P6₁</i>	170	<i>P6₅</i>	171	<i>P6₂</i>	172	<i>P6₄</i>
173	<i>P6₃</i>	174	<i>P-6</i>	175	<i>P6/m</i>	176	<i>P6₃/m</i>
177	<i>P622</i>	178	<i>P6₁22</i>	179	<i>P6₅22</i>	180	<i>P6₂22</i>
181	<i>P6₄22</i>	182	<i>P6₃22</i>	183	<i>P6mm</i>	184	<i>P6cc</i>
185	<i>P6₃cm</i>	186	<i>P6₃mc</i>	187	<i>P-6m2</i>	188	<i>P-6c2</i>
189	<i>P-62m</i>	190	<i>P-62c</i>	191	<i>P6/mmm</i>	192	<i>P6/mcc</i>
193	<i>P6₃/mcm</i>	194	<i>P6₃/mmc</i>	195	<i>P23</i>	196	<i>F23</i>
197	<i>I23</i>	198	<i>P2₁3</i>	199	<i>I2₁3</i>	200	<i>Pm-3</i>
201	<i>Pn-3</i>	202	<i>Fm-3</i>	203	<i>Fd-3</i>	204	<i>Im-3</i>
205	<i>Pa-3</i>	206	<i>Ia-3</i>	207	<i>P432</i>	208	<i>P4₂32</i>
209	<i>F432</i>	210	<i>F4₁32</i>	211	<i>I432</i>	212	<i>P4₃32</i>
213	<i>P4₁32</i>	214	<i>I4₁32</i>	215	<i>P-43m</i>	216	<i>F-43m</i>
217	<i>I-43m</i>	218	<i>P-43n</i>	219	<i>F-43c</i>	220	<i>I-43d</i>
221	<i>Pm-3m</i>	222	<i>Pn-3n</i>	223	<i>Pm-3n</i>	224	<i>Pn-3m</i>
225	<i>Fm-3m</i>	226	<i>Fm-3c</i>	227	<i>Fd-3m</i>	228	<i>Fd-3c</i>
229	<i>Im-3m</i>	230	<i>Ia-3d</i>				

9.5 层群列表

Triclinic							
1	$p1$	2	$p-1$				
Monoclinic / inclined							
3	$p112$	4	$p11m$	5	$p11a$	6	$p112/m$
7	$p112/a$						
Triclinic / orthogonal							
8	$p211$	9	$p2_111$	10	$c211$	11	$pm11$
12	$pb11$	13	$cm11$	14	$p2/m11$	15	$p2_1/m11$
16	$p2/b11$	17	$p2_1/b11$	18	$c2/m11$		
Orthorhombic							
19	$p222$	20	$p2_122$	21	$p2_12_12$	22	$c222$
23	$pmm2$	24	$pma2$	25	$pba2$	26	$cmm2$
27	$pm2m$	28	$pm2_1b$	29	$pb2_1m$	30	$pb2b$
31	$pm2a$	32	$pm2_1a$	33	$pb2_1a$	34	$pb2n$
35	$cm2m$	36	$cm2e$	37	$pmmm$	38	$pmaa$
39	$pban$	40	$pmam$	41	$pmma$	42	$pman$
43	$pbaa$	44	$pbam$	45	$pbma$	46	$pmmn$
47	$cmmm$	48	$cmme$				
Tetragonal							
49	$p4$	50	$p-4$	51	$p4/m$	52	$p4/n$
53	$p422$	54	$p42_12$	55	$p4mm$	56	$p4bn$
57	$p-4m2$	58	$p-42_1m$	59	$p-4m2$	60	$p-4b2$
61	$p4/mmm$	62	$p4/nbm$	63	$p4/mbn$	64	$p4/nmm$
Trigonal							
65	$p3$	66	$p-3$	67	$p312$	68	$p321$
69	$p3m1$	70	$p31m$	71	$p-31m$	72	$p-3m1$
Hexagonal							
73	$p6$	74	$p-6$	75	$p6/m$	76	$p622$
77	$p6mm$	78	$p-m2$	79	$p-62m$	80	$p6/mmm$

*括号里是在 USPEX 代码里使用的非标准空间群标号。

9.6 平面群列表

Number	Group
1	p1
2	p2
3	pm
4	pg
5	cm
6	pmm
7	pmg
8	pgg
9	cmm
10	p4
11	p4m
12	p4g
13	p3
14	p3m1
15	p31m
16	p6
17	p6m

9.7 点群列表

罗列出了在 Schönflies 和 Hermann-Maugin (国际) 符号中的所有晶体和最重要非晶体点群。

点群列表:

Hermann-Maugin	Schönflies	In USPEX
1	C ₁	C1 or E
2	C ₂	C2
222	D ₂	D2
4	C ₄	C4
3	C ₃	C3
6	C ₆	C6
23	T	T
$\bar{1}$	S ₂	S2
M	C _{1h}	Ch1
mm2	C _{2v}	Cv2
$\bar{2}$	S ₄	S4
$\bar{3}$	S ₆	S6
$\bar{6}$	C _{3h}	Ch3
m $\bar{3}$	T _h	Th
2/m	C _{2h}	Ch2
mmm	D _{2h}	Dh2
4/m	C _{4h}	Ch4
32	D ₃	D3
6/m	C _{6h}	Ch6
432	O	O
422	D ₄	D4
3m	C _{3v}	Cv3
622	D ₆	D6
$\bar{4}3m$	T _d	Td
4mm	C _{4v}	Cv4
$\bar{3}m$	D _{3d}	Dd3
6mm	C _{6v}	Cv6
m $\bar{3}m$	O _h	Oh
$\bar{4}2m$	D _{2d}	Dd2
$\bar{6}2m$	D _{3h}	Dh3
4/mmm	D _{4h}	Dh4
6/mmm	D _{6h}	Dh6
m $\bar{3}m$	O _h	Oh

Important non-crystallographic point groups

Hermann-Maugin	Schönflies	In USPEX
5	C ₅	C5
5/m	S ₅	S5
$\bar{5}$	S ₁₀	S10
5m	C _{v5v}	Cv5
$\bar{10}$	Ch _{5h}	Ch5
52	D ₅	D5
$\bar{5}m$	D _{5d}	Dd5
$\bar{10}2m$	D _{5h}	Dh5
532	I	I
$\bar{5}3m$	I _h	Ih

9.8 USPEX 中使用的共价原子半径表

USPEX 中所使用的原子半径 (单位 Å) (主要用于计算硬度 *etc.*):

Z	Element	radius	Z	Element	radius	Z	Element	radius
1	H	0.31	30	Zn	1.22	63	Eu	1.98
2	He	0.28	31	Ga	1.22	64	Gd	1.96
3	Li	1.28	32	Ge	1.20	65	Tb	1.94
4	Be	0.96	33	As	1.19	66	Dy	1.92
5	B	0.84	34	Se	1.20	67	Ho	1.92
6	Csp ³	0.76	35	Br	1.20	68	Er	1.89
	Csp ²	0.73	36	Kr	1.16	69	Tm	1.90
	Csp	0.69	37	Rb	2.20	70	Yb	1.87
7	N	0.71	38	Sr	1.95	71	Lu	1.87
8	O	0.66	39	Y	1.90	72	Hf	1.75
9	F	0.57	40	Zr	1.75	73	Ta	1.70
10	Ne	0.58	41	Nb	1.64	74	W	1.62
11	Na	1.66	42	Mo	1.54	75	Re	1.51
12	Mg	1.41	43	Tc	1.47	76	Os	1.44
13	Al	1.21	44	Ru	1.46	77	Ir	1.41
14	Si	1.11	45	Rh	1.42	78	Pt	1.36
15	P	1.07	46	Pd	1.39	79	Au	1.36
16	S	1.05	47	Ag	1.45	80	Hg	1.32
17	Cl	1.02	48	Cd	1.44	81	Tl	1.45
18	Ar	1.06	49	In	1.42	82	Pb	1.46
19	K	2.03	50	Sn	1.39	83	Bi	1.48
20	Ca	1.76	51	Sb	1.39	84	Po	1.40
21	Sc	1.70	52	Te	1.38	85	At	1.50
22	Ti	1.60	53	I	1.39	86	Rn	1.50
23	V	1.53	54	Xe	1.40	87	Fr	2.60
24	Cr	1.39	55	Cs	2.44	88	Ra	2.21
25	Mn l.s.	1.39	56	Ba	2.15	89	Ac	2.15
	h.s	1.61	57	La	2.07	90	Th	2.06
26	Fe l.s	1.32	58	Ce	2.04	91	Pa	2.00
	h.s.	1.52	59	Pr	2.03	92	U	1.96
27	Co l.s.	1.26	60	Nd	2.01	93	Np	1.90
	h.s.	1.50	61	Pm	1.99	94	Pu	1.87
28	Ni	1.24	62	Sm	1.98	95	Am	1.80
29	Cu	1.32				96	Cm	1.69

Source: Cordero *et al.*, Dalton Trans. 2832-2838, 2008³⁴.

9.9 USPEX 使用的默认化合价表

USPEX 中使用的化合价 *valences* 表 (主要用于硬度计算 *etc.*):

Z	Element	valence	Z	Element	valence	Z	Element	valence
1	H	1	35	Br	1	69	Tm	3
2	He	0.5	36	Kr	0.5	70	Yb	3
3	Li	1	37	Rb	1	71	Lu	3
4	Be	2	38	Sr	2	72	Hf	4
5	B	3	39	Y	3	73	Ta	5
6	C	4	40	Zr	4	74	W	4
7	N	3	41	Nb	5	75	Re	4
8	O	2	42	Mo	4	76	Os	4
9	F	1	43	Tc	4	77	Ir	4
10	Ne	0.5	44	Ru	4	78	Pt	4
11	Na	1	45	Rh	4	79	Au	1
12	Mg	2	46	Pd	4	80	Hg	2
13	Al	3	47	Ag	1	81	Tl	3
14	Si	4	48	Cd	2	82	Pb	4
15	P	3	49	In	3	83	Bi	3
16	S	2	50	Sn	4	84	Po	2
17	Cl	1	51	Sb	3	85	At	1
18	Ar	0.5	52	Te	2	86	Rn	0.5
19	K	1	53	I	1	87	Fr	1
20	Ca	2	54	Xe	0.5	88	Ra	2
21	Sc	3	55	Cs	1	89	Ac	3
22	Ti	4	56	Ba	2	90	Th	4
23	V	4	57	La	3	91	Pa	4
24	Cr	3	58	Ce	4	92	U	4
25	Mn	4	59	Pr	3	93	Np	4
26	Fe	3	60	Nd	3	94	Pu	4
27	Co	3	61	Pm	3	95	Am	4
28	Ni	2	62	Sm	3	96	Cm	4
29	Cu	2	63	Eu	3	97	Bk	4
30	Zn	2	64	Gd	3	98	Cf	4
31	Ga	3	65	Tb	3	99	Es	4
32	Ge	4	66	Dy	3	100	FM	4
33	As	3	67	Ho	3	101	Md	4
34	Se	2	68	Er	3	102	No	4

9.10 USPEX 使用的默认 goodBonds 表

USPEX 中使用的 goodBonds 表, 主要用于硬度计算 *etc.*:

Z	Element	goodBonds	Z	Element	goodBonds	Z	Element	goodBonds
1	H	0.20	35	Br	0.10	69	Tm	0.20
2	He	0.05	36	Kr	0.05	70	Yb	0.20
3	Li	0.10	37	Rb	0.05	71	Lu	0.20
4	Be	0.20	38	Sr	0.10	72	Hf	0.30
5	B	0.30	39	Y	0.20	73	Ta	0.40
6	C	0.50	40	Zr	0.30	74	W	0.30
7	N	0.50	41	Nb	0.35	75	Re	0.30
8	O	0.30	42	Mo	0.30	76	Os	0.30
9	F	0.10	43	Tc	0.30	77	Ir	0.30
10	Ne	0.05	44	Ru	0.30	78	Pt	0.30
11	Na	0.05	45	Rh	0.30	79	Au	0.05
12	Mg	0.10	46	Pd	0.30	80	Hg	0.10
13	Al	0.20	47	Ag	0.05	81	Tl	0.20
14	Si	0.30	48	Cd	0.10	82	Pb	0.30
15	P	0.30	49	In	0.20	83	Bi	0.20
16	S	0.20	50	Sn	0.30	84	Po	0.20
17	Cl	0.10	51	Sb	0.20	85	At	0.10
18	Ar	0.05	52	Te	0.20	86	Rn	0.05
19	K	0.05	53	I	0.10	87	Fr	0.05
20	Ca	0.10	54	Xe	0.05	88	Ra	0.10
21	Sc	0.20	55	Cs	0.05	89	Ac	0.20
22	Ti	0.30	56	Ba	0.10	90	Th	0.30
23	V	0.30	57	La	0.20	91	Pa	0.30
24	Cr	0.25	58	Ce	0.30	92	U	0.30
25	Mn	0.30	59	Pr	0.20	93	Np	0.30
26	Fe	0.25	60	Nd	0.20	94	Pu	0.30
27	Co	0.25	61	Pm	0.20	95	Am	0.30
28	Ni	0.15	62	Sm	0.20	96	Cm	0.30
29	Cu	0.10	63	Eu	0.20	97	Bk	0.30
30	Zn	0.10	64	Gd	0.20	98	Cf	0.30
31	Ga	0.25	65	Tb	0.20	99	Es	0.30
32	Ge	0.50	66	Dy	0.20	100	FM	0.30
33	As	0.35	67	Ho	0.20	101	Md	0.30
34	Se	0.20	68	Er	0.20	102	No	0.30

Bibliography

- [1] J. Maddox. Crystals from first principles. *Nature*, 335:201, 1988.
- [2] A.R. Oganov and C.W. Glass. Crystal structure prediction using ab initio evolutionary techniques: Principles and applications. *The Journal of Chemical Physics*, 124:244704, 2006.
- [3] C.W. Glass, A.R. Oganov, and N. Hansen. USPEX — evolutionary crystal structure prediction. *Comp. Phys. Comm.*, 175:713–720, 2006.
- [4] A.R. Oganov and S. Ono. Theoretical and experimental evidence for a post-perovskite phase of MgSiO₃ in Earth's D" layer. *Nature*, 430(6998):445–448, July 2004.
- [5] M. Murakami, K. Hirose, K. Kawamura, N. Sata, and Y. Ohishi. Post-perovskite phase transition in MgSiO₃. *Science*, 304(5672):855–858, 2004.
- [6] C.J. Pickard and R.J. Needs. High-pressure phases of silane. *Phys. Rev. Lett.*, 97:045504, Jul 2006.
- [7] M. Martinez-Canales, A.R. Oganov, Y. Ma, Y. Yan, A.O. Lyakhov, and A. Bergara. Novel structures and superconductivity of silane under pressure. *Phys. Rev. Lett.*, 102:087005, Feb 2009.
- [8] Y. Ma, A.R. Oganov, Y. Xie, Z. Li, and J. Kotakoski. Novel high pressure structures of polymeric nitrogen. *Phys. Rev. Lett.*, 102:065501, 2009.
- [9] C.J. Pickard and R.J. Needs. High-pressure phases of nitrogen. *Phys. Rev. Lett.*, 102:125702, Mar 2009.
- [10] G. Gao, A.R. Oganov, P. Li, Z. Li, H. Wang, T. Cui, Y. Ma, A. Bergara, A.O. Lyakhov, T. Iitaka, and G. Zou. High-pressure crystal structures and superconductivity of stannane (SnH₄). *Proceedings of the National Academy of Sciences*, 107(4):1317–1320, 2010.
- [11] C.J. Pickard and R.J. Needs. Structures at high pressure from random searching. *physica status solidi (b)*, 246(3):536–540, 2009.
- [12] A.O. Lyakhov, A.R. Oganov, H.T. Stokes, and Q. Zhu. New developments in evolutionary structure prediction algorithm USPEX. *Comp. Phys. Comm.*, 184:1172–1182, 2013.
- [13] G.R. Qian, X. Dong, X.-F. Zhou, Y. Tian, A.R. Oganov, and H.-T. Wang. Variable cell nudged elastic band method for studying solid-solid structural phase transitions. *Computer Physics Communications*, 184(9):2111–2118, 2013.
- [14] C. Dellago, P.G. Bolhuis, F.S. Csajka, and D. Chandler. Transition path sampling and the calculation of rate constants. *The Journal of Chemical Physics*, 108(5):1964–1977, 1998.
- [15] S.E. Boulfelfel, A.R. Oganov, and S. Leoni. Understanding the nature of "superhard graphite". *Scientific Reports*, 2(471):1–9, 2012.
- [16] A.R. Oganov and M. Valle. How to quantify energy landscapes of solids. *The Journal of Chemical Physics*, 130:104504, 2009.
- [17] A.R. Oganov, A.O. Lyakhov, and M. Valle. How evolutionary crystal structure prediction works — and why. *Accounts of Chemical Research*, 44(3):227–237, 2011.

- [18] Efim Mazhnik and Artem R. Oganov. Application of machine learning methods for predicting new superhard materials. *Journal of Applied Physics*, 128(7):075102, 2020.
- [19] Efim Mazhnik and Artem R. Oganov. A model of hardness and fracture toughness of solids. *Journal of Applied Physics*, 126(12):125109, 2019.
- [20] A.R. Oganov and A.O. Lyakhov. Towards the theory of hardness of materials. *Journal of Superhard Materials*, 32(3):143–147, 2010.
- [21] L.S. Dubrovinsky, N.A. Dubrovinskaia, V. Swamy, J. Muscat, N.M. Harrison, R. Ahuja, B. Holm, and B. Johansson. Materials science: The hardest known oxide. *Nature*, 410(6829):653–654, 2001.
- [22] R. Martoňák, A. Laio, M. Bernasconi, C. Ceriani, P. Raiteri, F. Zipoli, and M. Parrinello. Simulation of structural phase transitions by metadynamics. *Z. Krist.*, 220:489–498, 2005.
- [23] A.R. Oganov and C.W. Glass. Evolutionary crystal structure prediction as a tool in materials design. *Journal of Physics: Condensed Matter*, 20(6):064210, 2008.
- [24] W. Zhang, A.R. Oganov, A.F. Goncharov, Q. Zhu, S.E. Boulfelfel, A.O. Lyakhov, E. Stavrou, M. Somayazulu, V.B. Prakapenka, and Z. Konopkova. Unexpected stable stoichiometries of sodium chlorides. *Science*, 342(6165):1502–1505, 2013.
- [25] Georg KH Madsen and David J Singh. Boltztrap. a code for calculating band-structure dependent quantities. *Comput. Phys. Commun.*, 175(1):67–71, Jul 2006.
- [26] Q. Zhu, L. Li, A.R. Oganov, and P.B. Allen. Evolutionary method for predicting surface reconstructions with variable stoichiometry. *Phys. Rev. B*, 87:195317, May 2013.
- [27] S.T. Call, D.Yu. Zubarev, and A.I. Boldyrev. Global minimum structure searches via particle swarm optimization. *Journal of Computational Chemistry*, 28(7):1177–1186, 2007.
- [28] Y. Wang, J. Lv, L. Zhu, and Y. Ma. Crystal structure prediction via particle-swarm optimization. *Phys. Rev. B*, 82:094116, Sep 2010.
- [29] G. Mills, H. Jónsson, and G.K. Schenter. Reversible work transition state theory: application to dissociative adsorption of hydrogen. *Surf. Sci.*, 324(2):305–337, 1995.
- [30] G. Henkelman, B.P. Uberuaga, and H. Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *J. Chem. Phys.*, 113(22):9901–9904, 2000.
- [31] G. Henkelman and H. Jónsson. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *J. Chem. Phys.*, 113(22):9978–9985, 2000.
- [32] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch. Structural relaxation made simple. *Phys. Rev. Lett.*, 97(17):170201, 2006.
- [33] M. Valle. STM3: a chemistry visualization platform. *Z. Krist.*, 220:585–588, 2005.
- [34] B. Cordero, V. Gomez, A.E. Platero-Prats, M. Reves, J. Echeverria, E. Cremades, F. Barragan, and S. Alvarez. Covalent radii revisited. *Dalton Trans.*, 21:2832–2838, 2008.