|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
|  | | | **编 号** | | **指标1-2** | |  |
| **密 级** | | **公开** | |  |
| **页　　数** | | **47** | |  |
|  | | | | | | | |
| |  |  | | --- | --- | | **文件名称** | **金属基复合材料基体/增强体/界面的物性/变形参数与温度相关性计算模块技术报告** | | | | | | | | |
|  | | | | | | | |
|  | **编制单位** |  | | **上海交通大学** | |  | |
| **编　　写** |  | | **龚鸣宇** | |
| **校　　对** |  | | **曹贺** | |
| **审　　核** |  | | **赵蕾** | |
| **批　　准** |  | | **苏益士** | |
|  | | | | | | | |
| **上海交通大学 材料科学与工程学院** | | | | | | | |

**目 录**

[一、 概述 1](#_Toc217893525)

[二、 研究目的及指标 2](#_Toc217893526)

[2.1 研究目的 2](#_Toc217893527)

[2.2 项目指标 2](#_Toc217893528)

[三、 研究内容 3](#_Toc217893529)

[3.1 模块介绍 3](#_Toc217893530)

[3.1.1 单晶体系内聚能计算 4](#_Toc217893531)

[3.1.2 缺陷体系形成能计算 4](#_Toc217893532)

[3.1.3 界面体系界面能计算 4](#_Toc217893533)

[3.1.4 位错体系核心能计算 5](#_Toc217893534)

[3.1.5 最小能量路径（MEP）体系迁移垒计算 5](#_Toc217893535)

[3.2 算例描述 6](#_Toc217893536)

[3.2.1 Al-Mg合金 6](#_Toc217893537)

[3.2.2 Ti/TiB2复合材料 14](#_Toc217893538)

[四、 结论 20](#_Toc217893539)

[五、 附件 21](#_Toc217893540)

[5.1 《金属基复合材料原子尺度构型设计软件》著作权登记证书…………………………………………………..21](#_Toc217893541)

[5.2 《金属基复合材料原子尺度构型设计软件》第三方测评报告…………………………………………….…….22](#_Toc217893542)

[六、 附表 31](#_Toc217893543)

[6.1 “金属基复合材料界面/增强体/界面的物性/变形参数与温度相关性计算模块”源代码 31](#_Toc217893544)

# 概述

金属基复合材料的宏观力学性能与服役可靠性高度依赖于基体、增强体及界面相的物性参数（如内聚能、界面能）与变形参数（如缺陷形成能、位错核心能、迁移垒），而这些参数具有显著的温度依赖性—温度变化会引发原子间相互作用势垒重构、晶体结构热振动加剧及界面结合状态演化，直接导致参数数值的动态波动。在开展金属基复合材料温度相关的原子尺度模拟研究时，需基于LAMMPS开源程序开展大量分子静力学（MS）与分子动力学（MD）模拟，生成海量包含原子坐标、能量、应力、位移等信息的输出文件。传统数据处理方式依赖人工编程或商业软件，存在流程碎片化、计算标准不统一、数据提取效率低、易引入人为误差等问题，尤其难以满足多温度点、多体系（基体/增强体/界面）、多参数同步计算的规模化需求，严重制约了温度依赖性构-效关系模型的精准构建。

本研究开发了金属基复合材料基体/增强体/界面物性与变形参数温度相关性专用计算模块，核心功能为自动化处理LAMMPS分子静力学与分子动力学计算输出文件，实现多类型温度依赖性关键参数的标准化、精准化提取与计算。该模块基于面向对象设计理念，构建了“数据解析-参数计算-结果校验-输出可视化”全流程自动化框架，通过集成标准化的物理模型与计算算法，有效解决了传统数据处理的技术瓶颈，为系统揭示温度对金属基复合材料微观参数的调控规律、构建“温度-微观参数-宏观性能”关联模型提供关键技术支撑。

# 研究目的及指标

## 研究目的

金属基复合材料中基体、增强体、界面的物性及变形参数随温度变化而变化，涉及大量原子尺度计算及其数据提取工作，为此开发金属基复合材料基体/增强体/界面的物性/变形参数与温度相关性计算模块处理“LAMMPS开源程序”的分子静力学与分子动力学计算输出文件，分别实现单晶计算结果的内聚能计算、缺陷计算结果的形成能计算、界面计算结果的界面能计算、位错计算结果的核心能计算、MEP计算结果的迁移垒计算。

## 项目指标

项目技术指标，如表1所示。

1. 项目技术指标

|  |  |  |
| --- | --- | --- |
| **指标名称** | **完成时**  **指标值/状态** | **考核方式（方法）及**  **评价手段** |
| 指标1. 金属基复合材 料强度/韧性/耐热的 多尺度集成计算模型 | 金属基复合材料界面与缺陷、**基体/增强体/界面的温度相关物性/变形参数**、构型基元组分热稳定性、构型基元晶体塑性、构型基元热-力耦合特性的多尺度计算模块5 项 | 提供技术报告，  专家评审考核 |

# 研究内容

## 模块介绍

本模块可快速提取“LAMMPS开源程序”产生的金属基复合材料中基体、增强体、界面分子静力学及分子动力学计算输出文件中的关键信息，得到与温度相关的物性/变形参数，其运行逻辑如图1所示，包括内聚能计算、形成能计算、界面能计算、核心能计算及迁移垒计算5个功能。内聚能计算功能读取单晶的静态/动态优化松弛结果，形成能计算功能读取点缺陷（空位、间隙原子、替换原装等）与体缺陷（析出相、增强相等）的静态/动态优化松弛结果，界面能计算读取界面的静态/动态优化松弛结果，核心能计算功能读取线缺陷（位错、阶错等）的静态/动态优化松弛结果，迁移垒计算功能读取最小能量路径结果。

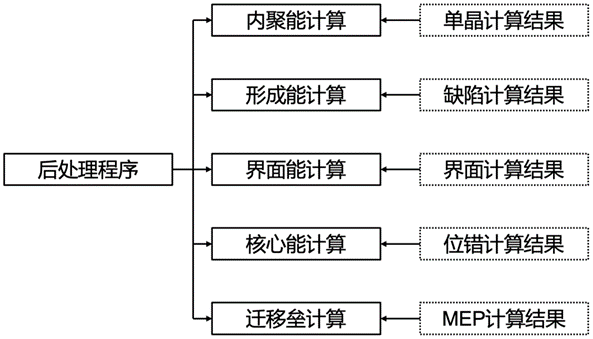


图1 金属基复合材料基体/增强体/界面的物性/变形参数与温度相关性计算模块运行逻辑

**3.1.1 单晶体系内聚能计算**

针对基体与增强体单晶体系的LAMMPS输出文件（含能量文件、原子坐标文件），模块可自动解析不同温度下的体系总能量、原子数量及模拟胞尺寸等核心数据。基于内聚能的物理定义（单位体积/单位原子的结合能，反映原子间结合牢固程度），集成标准化计算模型：内聚能=（体系总能量-单个原子孤立态能量总和）/原子数量（或模拟胞体积），自动扣除温度场引发的热振动能量修正项，精准计算不同温度下基体与增强体的内聚能参数。该参数是评估材料热稳定性与抗高温失效能力的核心指标，为筛选高温服役适配的基体/增强体体系提供数据支撑。

**3.1.2 缺陷体系形成能计算**

针对含空位、间隙原子、位错等缺陷的LAMMPS模拟输出文件，模块可自动区分完美晶体与缺陷晶体的能量数据，通过引入温度相关的晶格振动修正系数，实现缺陷形成能的精准计算。其核心算法基于：缺陷形成能=缺陷体系总能量-完美晶体总能量+缺陷原子数×完美晶体中单个原子平均能量，同时自动校正不同温度下模拟胞热膨胀导致的体积效应。该功能可系统获取不同温度下基体、增强体及界面区域各类缺陷的形成能数据，为揭示温度诱导缺陷萌生、演化及聚集的能量驱动机制提供核心依据。

**3.1.3 界面体系界面能计算**

针对基体-增强体、基体-反应层等多类界面体系的LAMMPS输出文件，模块可自动解析界面两侧不同相区的能量贡献与原子分布信息。采用“剥离法”结合温度修正模型计算界面能，核心流程为：首先计算界面复合体系总能量，再分别计算相同温度下单独基体相、增强体相（或反应层相）的能量（保持与复合体系相同的尺寸与边界条件），通过公式：界面能=（界面复合体系总能量-基体相能量-增强体相能量）/界面面积，自动扣除温度引发的界面原子热扩散能量损耗，精准获取不同温度下界面能数值。界面能是评估界面结合强度与热稳定性的关键参数，其温度依赖性数据为预测高温服役下界面脱粘失效风险提供核心支撑。

**3.1.4 位错体系核心能计算**

针对位错演化相关的LAMMPS输出文件，模块可通过原子中心对称参数（CSP）识别位错核心区域原子，自动分割位错核心区与基体区的能量数据。基于连续介质力学与原子尺度能量分解模型，集成位错核心能计算算法，扣除温度导致的基体区热振动能量背景，精准提取不同温度下位错核心区的额外能量（即位错核心能）。该参数直接反映位错运动的能量势垒，其温度依赖性规律为解析高温下材料塑性变形机制（如位错滑移、交滑移）的温度敏感性提供关键数据。

**3.1.5 最小能量路径（MEP）体系迁移垒计算**

针对缺陷迁移、界面扩散等过程的LAMMPS微动弹性带（NEB）模拟输出文件，模块可自动解析不同温度下各迁移路径节点的能量数据，识别能量极值点（势垒峰与势阱）。基于最小能量路径理论，通过二次插值拟合能量-路径曲线，精准计算不同温度下缺陷迁移或界面原子扩散的迁移垒（势垒峰与势阱的能量差值），同时输出迁移路径的能量分布图谱。迁移垒的温度依赖性数据是揭示高温下缺陷动态演化速率与界面扩散机制的核心依据，为预测材料高温蠕变与时效软化行为提供关键参数。

## 算例描述

本部分以常见的铝镁（Al-Mg）合金与硼化钛增强钛基（Ti/TiB2）复合材料的微观模拟为例，展示程序对分子静力学（MS）与动力学（MD）模拟结果处理的准确性。

**3.2.1 Al-Mg合金**

该合金中，Al为基体，Mg为置换型的溶质原子，整体上存在孪生与位移两种塑性变形机制。通过前处理模块，自动生成MS所需的结构能量最小化及微弹性带（NEB）法计算过渡态的基本输入文件，MD所需的加载的基本输入文件以及Al-Mg体系的势函数文件。该体系下，所有的MS和MD涉及能量最小化的条件一致，时间步为0.001 ps。

* Al单晶结构

对Al的初始晶体模型进行结构优化，可得图2的可视化结果。晶体结构为面心立方（FCC）结构，在可视化软件OVITO的共近邻原子分析（CNA）中对应绿色，晶格常数为a = 4.032 Å。由后处理模块直接得到每个原子的内聚能为-3.36 eV，与已有计算结果一致。

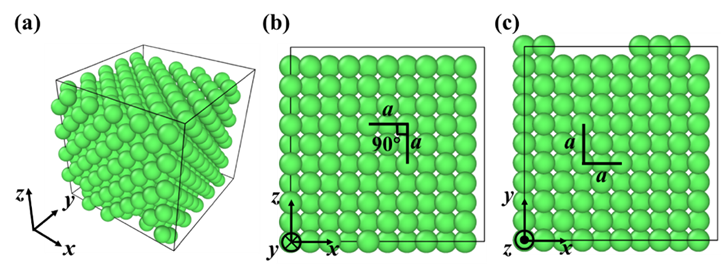


图2 Al的晶体结构：（a）三维视角；（b）主视图视角；（c）俯视图视角

* Mg单晶结构

对Mg的初始晶体模型进行结构优化，可得图3的可视化结果。晶体结构为密排六方（HCP）结构，在OVITO的CNA分析中对应红色，晶格常数为a = 3.196 Å，c = 5.197 Å。由后处理模块直接得到每个原子的内聚能为-1.51 eV，与已有计算结果一致。

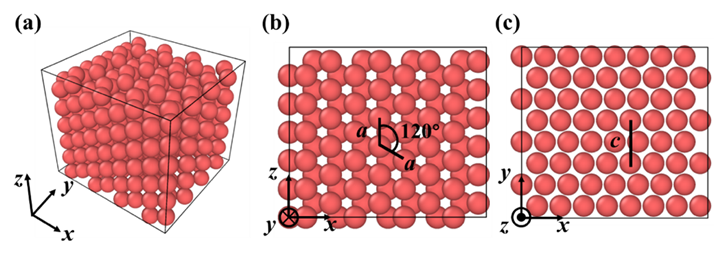


图3 Mg的晶体结构：（a）三维视角；（b）主视图视角；（c）俯视图视角

* Al基体中的Mg溶质原子

在Al中用Mg原子替换某个位置上的原子，该置换型溶质原子会改变原本晶格点位置的成键环境，引起能量的变化。后处理模块直接得到笛卡尔坐标系下溶质原子的位置 (0.00, 2.02, 2.02) Å，与可视化结果（图4（a））直接判定的位置 (0.00, 2.016, 2.016) Å一致，空位的形成能为0.55 eV。溶质引起的能量变化从三维视角的图4（b）分析，两种原子在此处的能量相比于各自的单晶水平均有所降低，原因是Al-Al键比Al-Mg键弱，此处形成了更稳定的键合环境，符合已有的第一性原理计算结果。

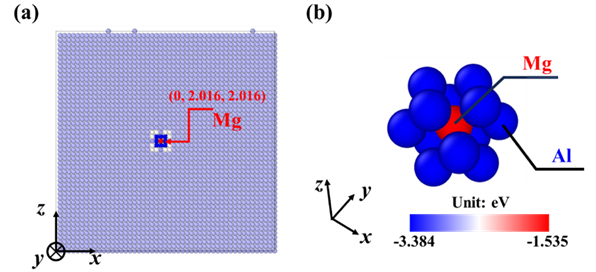


图4 Al基体中的Mg溶质原子：（a）空间位置；（b）对周围原子的影响

* Al/Mg异质界面

随着Mg在基体中含量的上升，合金微观组织中不再局限于点缺陷形式的置换原子，而是会出现一定尺寸的HCP相，与Al基体形成异质界面。Al和Mg各自的密排面可以形成半共格界面并满足取向关系：(111)Al//(0001)Mg，[11-1]Al//[1-100]Mg。进行了结构的能量最小化后，可以得到图5（a）的界面结构。界面层在y方向上的坐标处在-3.00 Å与2.00 Å之间，具有图5（b）的失配位错网络，基本的位错为Shockley位错，位错包围的区域通过CNA分析可以观察到晶体结构堆垛方式的交替变化。由后处理模块直接得界面能为288.78 mJ/m2，界面的中心位置在0.61 Å，处在可视化人为标定的实际范围内。

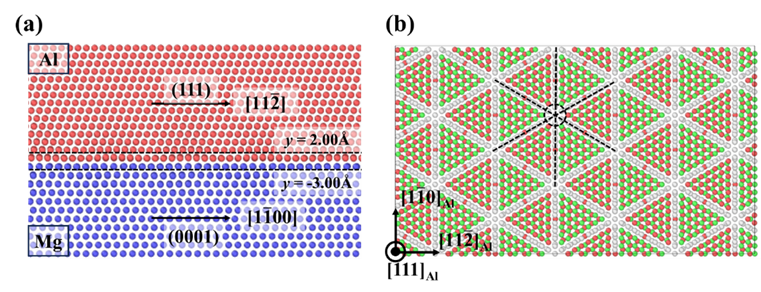


图5 Al/Mg异质界面：（a）界面取向及位置；（b）界面层失配位错网络

* Al孪晶界面

FCC相Al的层错能较高，通常不容易产生孪晶，但是在极端情况包括急剧的应力与温度变化，在应力集中区域可能会形成生长与形变孪晶，不仅自身提供了塑性，还具有存储与钉扎位错的能力。图6（a）展示了Al中最典型的(111)共格孪晶界，能量最小化后界面层在y方向上的坐标处在-0.19 Å。后处理模块直接得到界面能为78.73 mJ/m2，界面位置在-0.21 Å，与人为标定的结果处于同一原子层。图6（b）显示两侧晶体取向过渡时，共格孪晶面上的原子比其单晶状态稳定，成键环境的改变在相邻区域形成了一定的高能区域，远离界面的区域仍旧处于单晶的能量水平，与已有的计算结果一致。

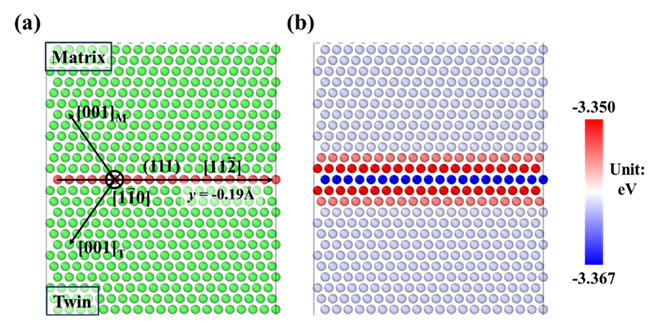


图6 Al(111)共格孪晶界：（a）界面取向及位置；（b）界面区域的能量变化

* Al单晶中的位错

FCC金属中，位错承担着大部分塑性变形的作用，因此对位错需要有足够的认识。在密排(111)面上，全位错1/2<110>倾向于分解成被两个1/6<112>Shockley位错包围局部层错的形式进行滑移，呈现出扩展位错的组态。对于一个1/2<110>全位错，用前处理模块生成的输入文件进行能量最小化，可以得到符合该理论预期的位错核结构（图7）。将输出文件送入后处理模块中，可直接得到位错核心能量为0.37 eV/Å，位错核的宽度为13.10 Å，位错线的中心位置在x = 3.196 Å，y = 5.197 Å处。

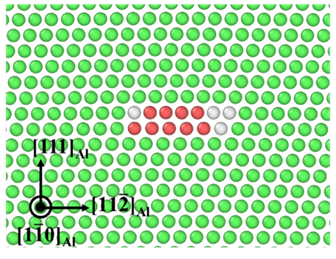


图7 Al单晶中的1/2<110>位错，能量最小化后形成扩展位错

* 溶质偏析后的位错

合金材料中，溶质不可能始终处于理想均匀的分布状态，而是会倾向于扩散到某些缺陷的位置，起到稳定体系能量的作用。因此在3.1.6部分得到结构的基础上，用Mg原子替换掉其中一个位错区域的原子，考察溶质存在对位错结构的影响，可以得到图8的结构优化结果。通过后处理模块得到位错核心能量为0.29 eV/Å，位错核的宽度为11.37 Å，位错线的中心位置在x = 13.09 Å，y = 0.09 Å处。可以发现溶质偏析到位错区域之后，位错的核心宽度变窄，结构更加紧密了，且核心能量降低，符合溶质稳定缺陷的预期。

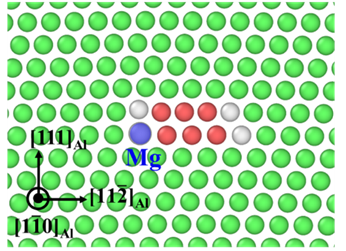


图8 Mg溶质存在的1/2<110>位错，扩展位错核结构

* 加载下的位错迁移

材料起到结构作用的关键场合在于承受外部加载，此时微观尺度上会激发位错的运动，以提供后续强度或者塑性相关的响应。为了设计出更可靠的结构材料，要对位错发生迁移的应力条件有充分的认识。通过前处理模块生成的MD输入文件可以研究一定加载下的位错迁移行为，该加载在NVT系综下进行，温度设定为10K，应变率为1×10-5，每1000时间步执行一次应变以保证有足够的时间进行微观结构的相应。整个过程记录的应力应变曲线如图9所示，其中有4个标注点对应位错迁移的状态。状态1和2之间位错并没有发生指定方向的迁移，说明没有达到临界分切应力（CRSS）。状态2-4可以发现位错持续发生指定方向上的运动，说明状态2对应的应力就是CRSS，大小为11MPa，与文献中记录的结果处于同一量级。这也说明从状态2开始材料进入了屈服阶段，通过塑性变形的启动进行了应力释放，与实际实验现象相符合，说明了这套程序的准确性。

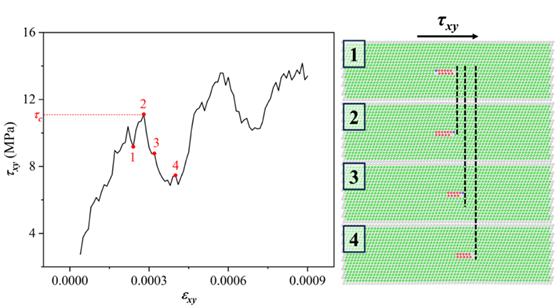


图9 含位错的Al中恒定剪切率下的应力变化与位错迁移情况：左侧曲线上的标注点对应右侧的迁移情况

* 过渡态计算研究位错迁移

位错的迁移壁垒也可以通过NEB法计算过渡态能量来获取。前处理模块可用于进行基于NEB法的计算：以位错迁移1/2<110>距离，即移动一个柏氏矢量的前后作为初末态（迁移距离约化为1.0），设定生成包括这两个状态在内的21个镜像进行计算，得到的结果如图10所示。以初态为参考状态，每个状态的相对能量构成数据点，手动计算得到的能垒大小为2.06 meV/nm。后处理模块直接得到这个NEB过程的能垒为2×10-3 eV，与手动计算的结果一致，说明了该模块处理过渡态计算的方便性与准确性。



图10 NEB法计算位错迁移的能垒

**3.2.2 Ti/TiB2复合材料**

该材料为金属基复合材料，Ti为基体，TiB2为增强体。除了典型的空位点缺陷以及位错外，存在着位错与增强体发生交互的情形。通过前处理模块，自动生成MS所需的结构能量最小化的基本输入文件，MD所需的加载的基本输入文件以及Ti-B体系的势函数文件。该体系下，所有的MS和MD涉及能量最小化的条件一致，时间步为0.001 ps。

* TiB2单晶结构

对TiB2的初始晶体模型进行结构优化，可得图11的可视化结果。晶体结构为六方结构，B层和Ti层交替堆垛。六方结构对应晶格常数为a = 3.050 Å，c = 3.197 Å。由后处理模块直接得到B原子的内聚能为-7.580 eV，Ti原子的内聚能为-4.497 eV，与已有计算结果一致。

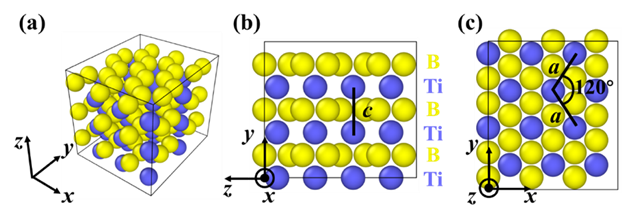


图11 TiB2的晶体结构：（a）三维视角；（b）左视图视角；（c）俯视图视角

* Ti单晶结构

对Ti的初始晶体模型进行结构优化，可得图12的可视化结果。晶体结构为HCP结构，在OVITO的CNA分析中对应红色，晶格常数为a = 2.945 Å，c = 4.687 Å。由后处理模块直接得到每个原子的内聚能为-4.873 eV，与已有计算结果一致。

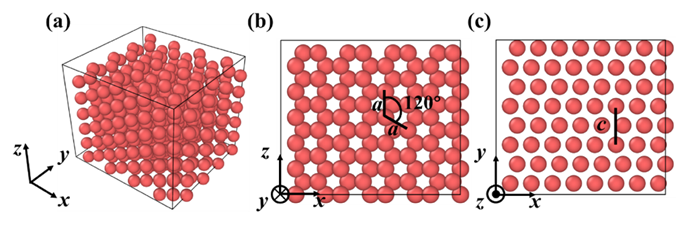


图12 Ti的晶体结构：（a）三维视角；（b）左视图视角；（c）俯视图视角

* Ti基体中的空位

之前在合金体系中考虑了溶质原子这类置换型点缺陷，会对局部的环境造成影响。在单晶中，为了实现热力学平衡，空位是更为普遍的缺陷。在应力集中区域如位错和晶界位置，空位的出现可以实现应力释放以及提供位点以帮助缺陷迁移，对空位的理解可以加深对其他缺陷迁移机制乃至变形机制的理解。随机挖掉Ti晶格中的一个原子，使其成为空的晶格点，通过前处理模块生成的输入文件，得到图13（a）的结构。后处理模块直接得到笛卡尔坐标系下空位的位置 (8.90, 9.52, 10.32) Å，与可视化结果（图13（a））直接判定的位置 (8.97, 9.70, 10.31) Å一致，空位的形成能为1.78 eV。空位引起的能量变化从三维视角的图13（b）分析，相比于单晶的内聚能，空位周围原子的能量更高，原因是减少了参与成键的原子，该晶格点周围12个原子的等效配位数降低，这一点通过本套程序可以准确捕捉到。

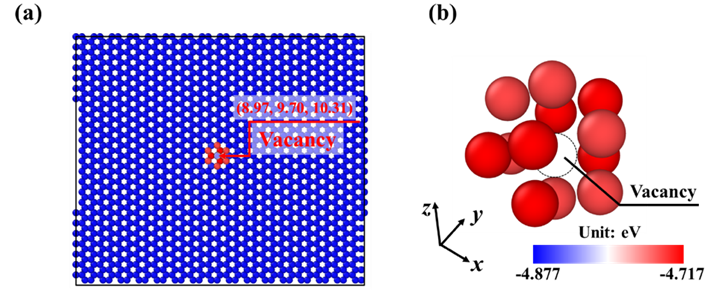


图13 Ti单晶中的空位：（a）空间位置；（b）对周围原子的影响

* Ti/TiB2异质界面

复合材料中存在基体与增强体，势必会形成两相的界面过渡区域，TiB2增强体与Ti基体形成异质界面。两相的基面可以形成半共格界面并满足一定的取向关系：(0001)Ti//(0001)Ti，[1-100]Ti//[1-100]Ti。进行了结构的能量最小化后，可以得到图14（a）的界面结构。界面层在y方向上的坐标处在-1.70 Å与0.40 Å之间，具有图14（b）的失配位错网络，位错网络勾勒的区域通过能量数值的映射可观察到明显的花纹变化。由后处理模块直接得界面能为1666.11 mJ/m2，界面的中心位置在-1.65 Å，处在可视化人为标定的实际范围内。

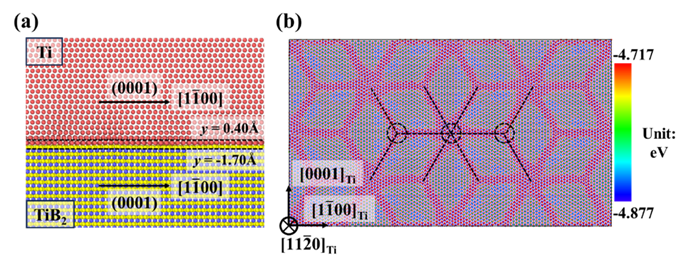


图14 Ti/TiB2异质界面：（a）界面取向及位置；（b）界面层失配位错网络

* Ti单晶中的位错

HCP金属中，位错承担着非<c>方向塑性变形的作用，对其他塑性变形缺陷如变形孪晶有交互作用，因此对位错需要有足够的认识。对于Ti，在柱面(1-100)面上可存在<a>位错的运动以提供相应方向的塑性变形，柏氏矢量为1/3[11-20]。用前处理模块生成的输入文件进行能量最小化，沿用图15的能量映射范围，可以得到相应的位错核结构。将输出文件送入后处理模块中，可直接得到位错核心能量为1.05 eV/Å，位错核的宽度为34.78 Å，位错线的中心位置在x = -0.67 Å，y = 1.12 Å处。

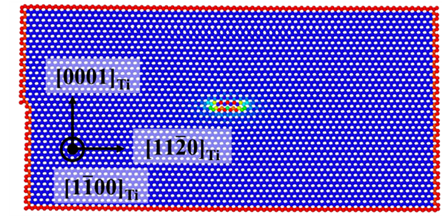


图15 Ti单晶中的1/3[11-20]位错核结构

* 加载下的位错迁移

以柱面<a>位错为例，通过前处理模块生成的MD输入文件可以研究一定加载下的位错迁移行为，该加载在NVT系综下进行，温度设定为10K，每1000时间步执行一次应变以保证有足够的时间进行微观结构的相应。整个过程记录的应力应变曲线如图16所示，其中有4个标注点对应位错迁移的状态。从状态1开始，位错持续发生指定方向上的运动，说明状态1对应的应力就是CRSS，大小为0.86 GPa。同时也说明从状态3开始材料应力下降是因为进入了屈服阶段，通过塑性变形的启动进行了应力释放，与实际实验现象相符合，并且该模拟的模型边缘为三维方向的固定边界条件，位错在接近边界后被阻碍导致的应力积累也可以捕捉到，说明了这套程序对相关力学现象模拟的准确性。

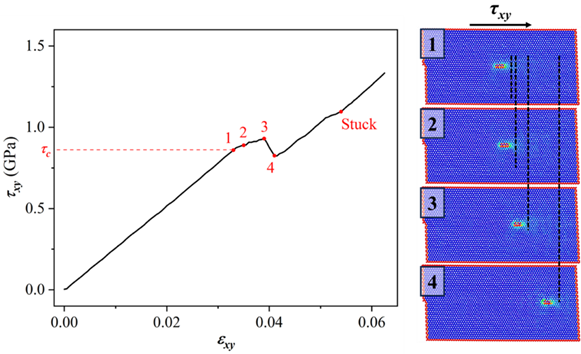


图16 含位错的Ti中恒定剪切率下的应力变化与柱面位错的迁移情况：左侧曲线上的标注点对应右侧的迁移情况

* 位错与增强体颗粒的交互

对于金属基复合材料，不可避免会产生可动位错与增强体颗粒的交互。对于TiB2颗粒，其相较于Ti基体比较硬，不容易被位错直接切过，所以会发生位错绕过并留下位错环的情况。对柱面位错的滑移面使用周期性边界条件并设定固定步数一百万步，使得位错可以反复与增强体颗粒发生交互，容易观测到对应的现象。初始模型如图17（a）所示，柱面位错线和增强体颗粒保持了适当的距离。在这个状态下进行切应力的加载，得到了整个模拟体系的应力应变曲线（图17（b）），可从其中的一部分数据点判断位错和增强体交互的情况。

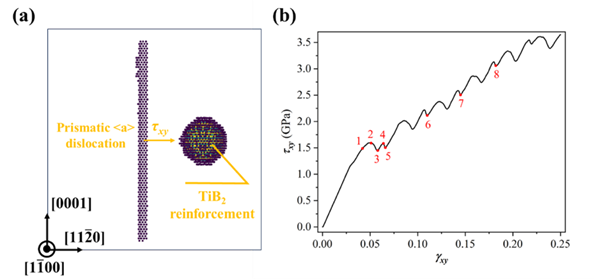


图17 Ti柱面<a>位错和TiB2增强体的交互：（a）加载前的初始状态；（b）应力-应变曲线

在曲线上提取8个数据点，各自对应的状态如图18所示。状态1是位错开始和增强体接触的时刻，因为颗粒的强度相对较高，所以位错的运动被阻碍，应力仍旧继续上升到状态2。此时位错被钉扎在颗粒区域的长度已经达到了饱和值，其余部分可以绕过阻碍继续运动，应力得到部分释放。到了状态3，可动部分的位错线几乎和运动方向平行，无法再作进一步滑移，为了整体能够继续运动，被钉扎的部分必须脱离原本的位错线，因此需要额外的应力使得位错线如同状态4一样绕在颗粒上。状态5对应的情况即为正负位错线接触形成了第一个位错环，此时整根位错线都绕过了增强体，因为周期性条件，又恢复了和状态1接近的情况。这些情况表明，应力每经历偶数次下降后即将上升的时刻对应位错环的形成，新的一轮交互即将开始。状态6-8对应着形成了2 ~ 4个位错环的形成，因为复杂的交互应力变化，会形成少量缺陷团簇。这些都符合实验中观测到的力学现象，说明通过这套程序的设置输入可以准确地模拟这一复杂过程。

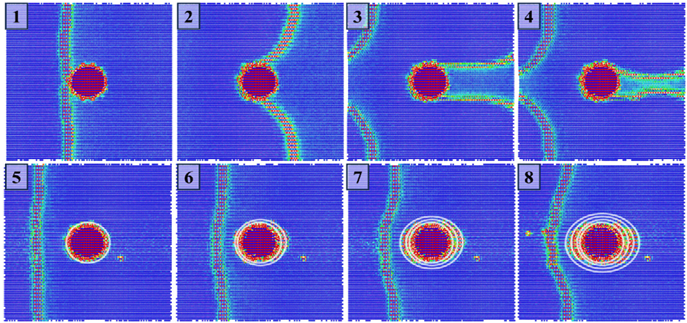


图18 加载曲线上标注点对应的交互状态

# 结论

（1）开发的计算模块通过对LAMMPS计算输出文件进行处理，实现了单晶计算结果的内聚能计算、缺陷计算结果的形成能计算、界面计算结果的界面能计算、位错计算结果的核心能计算、MEP计算结果的迁移垒计算；

（2）开发的计算模块成功用于Al-Mg合金以及Ti/TiBx复合材料的微观模拟结果的数据提取，通过后处理模块得到原子内聚能、界面能等数据，与已有计算结果一致，获取的计算数据准确性好。

# 附件

## 《金属基复合材料原子尺度构型设计软件》著作权登记证书

金属基复合材料原子尺度设计，已申请计算机软件著作权：金属基复合材料原子尺度构型设计软件V1.0，证书如下：

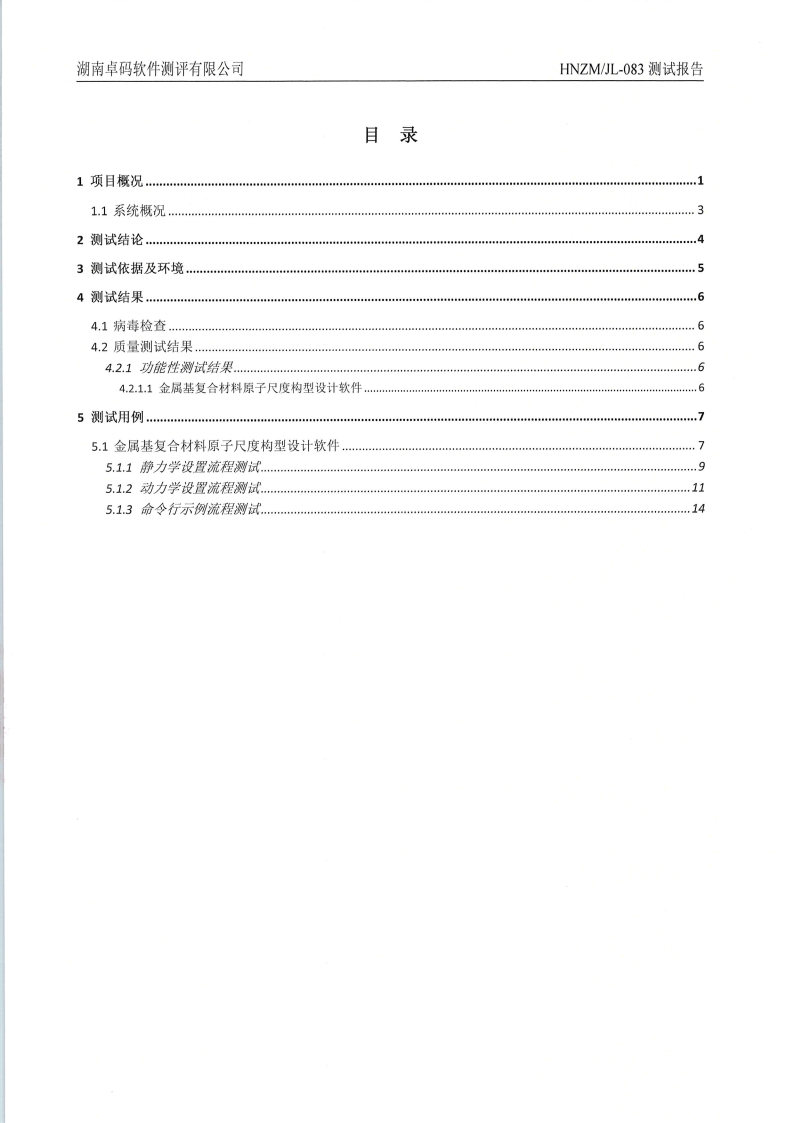
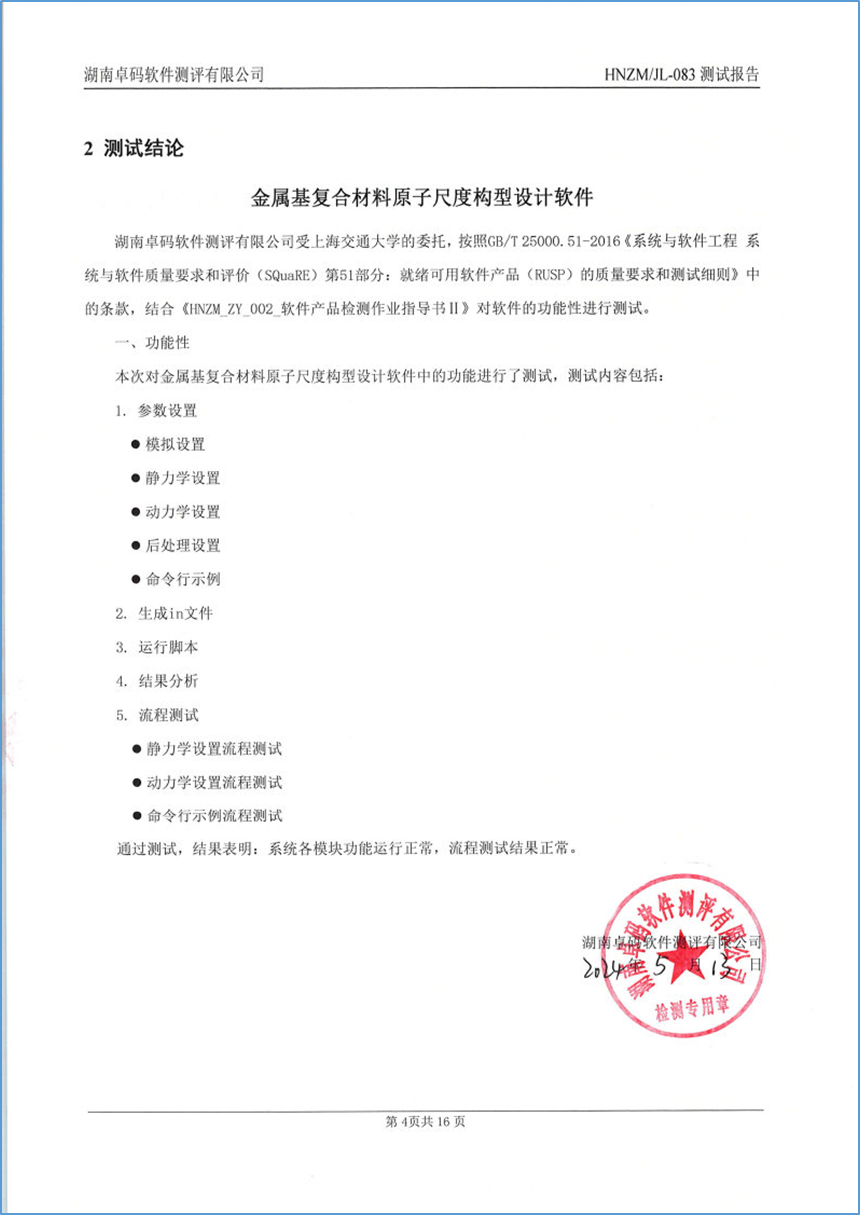
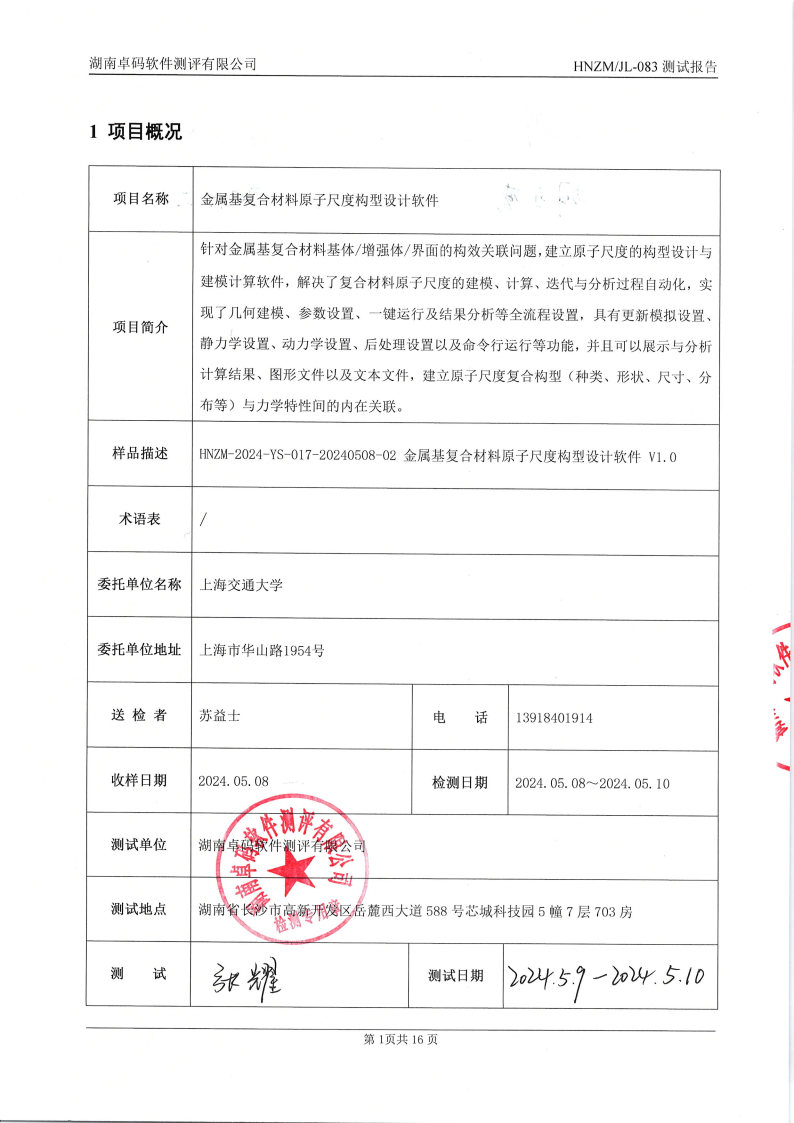
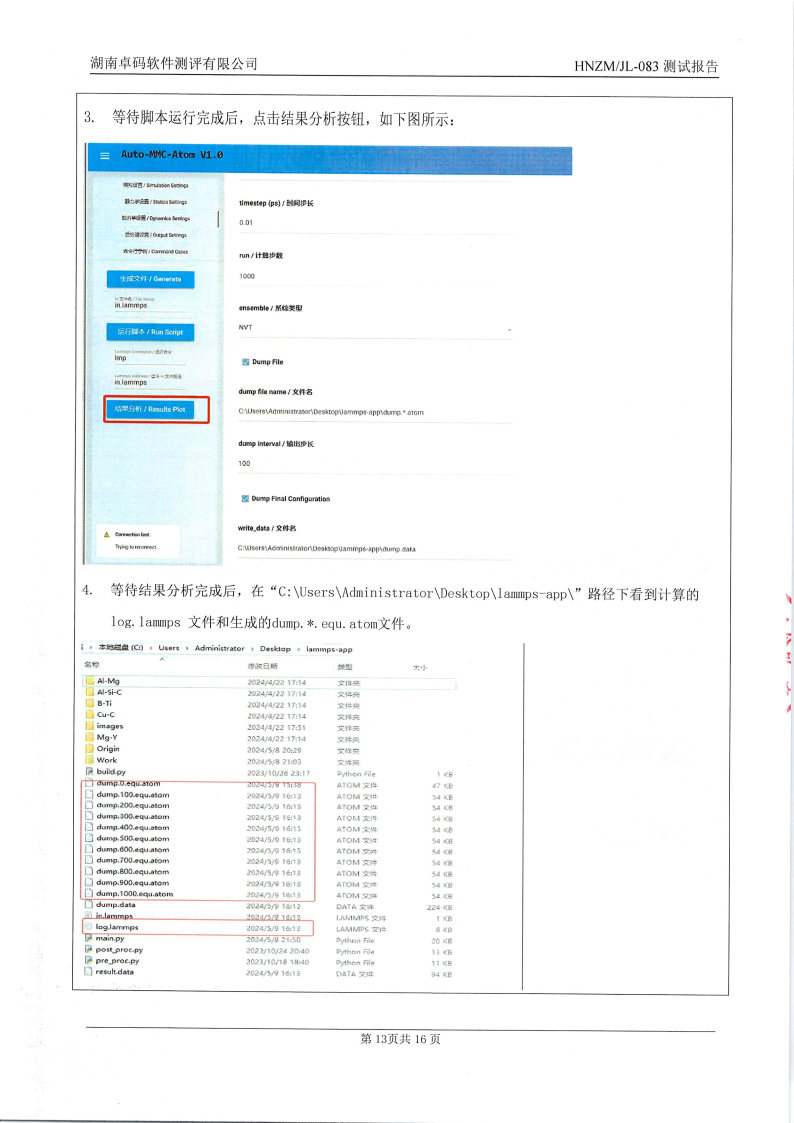
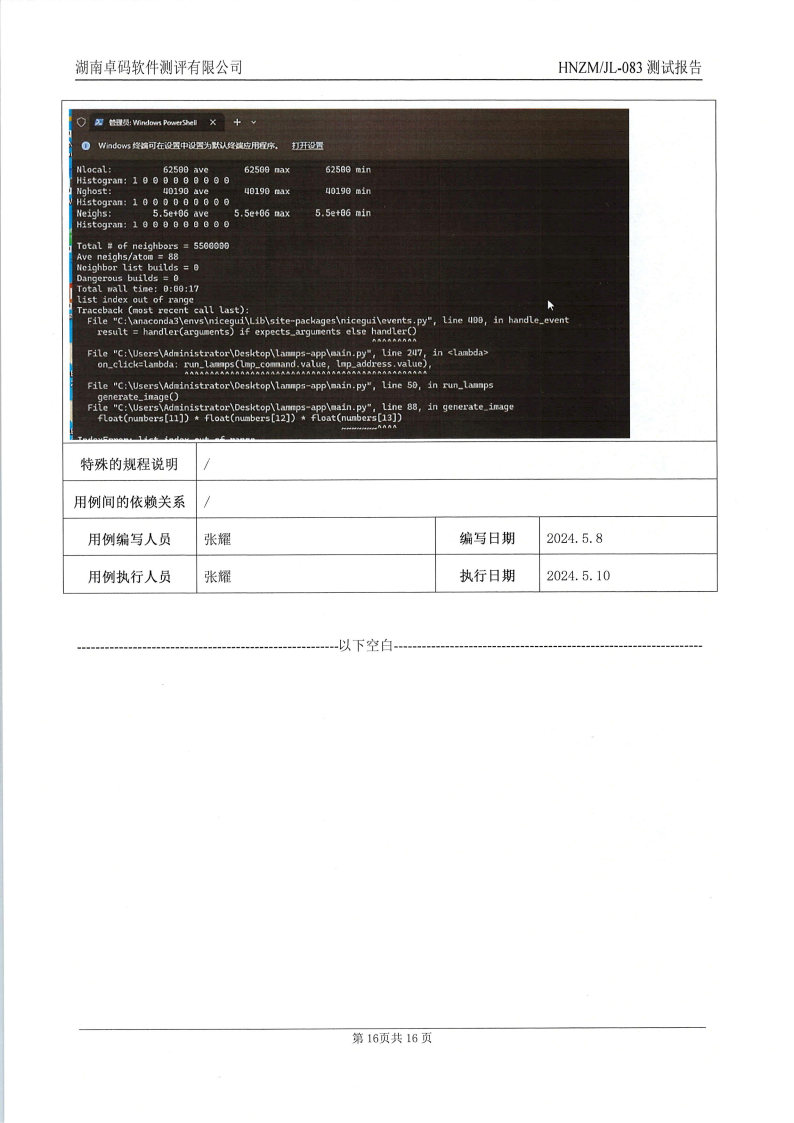
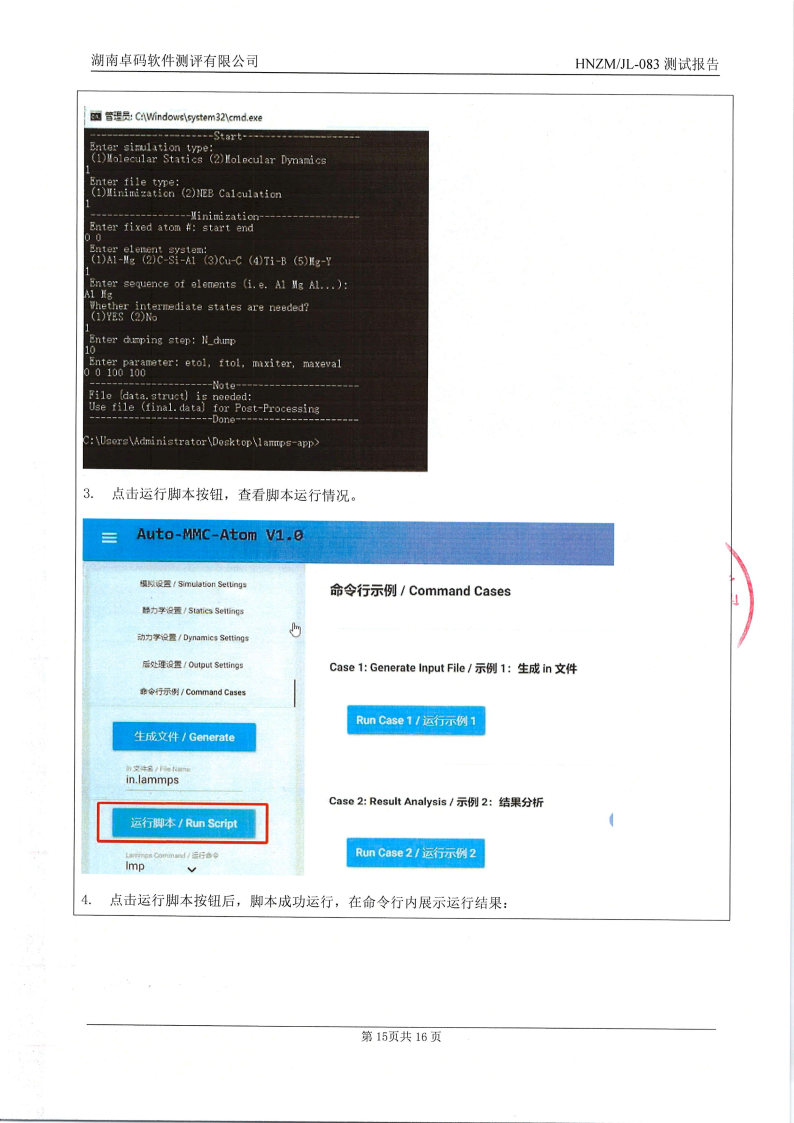
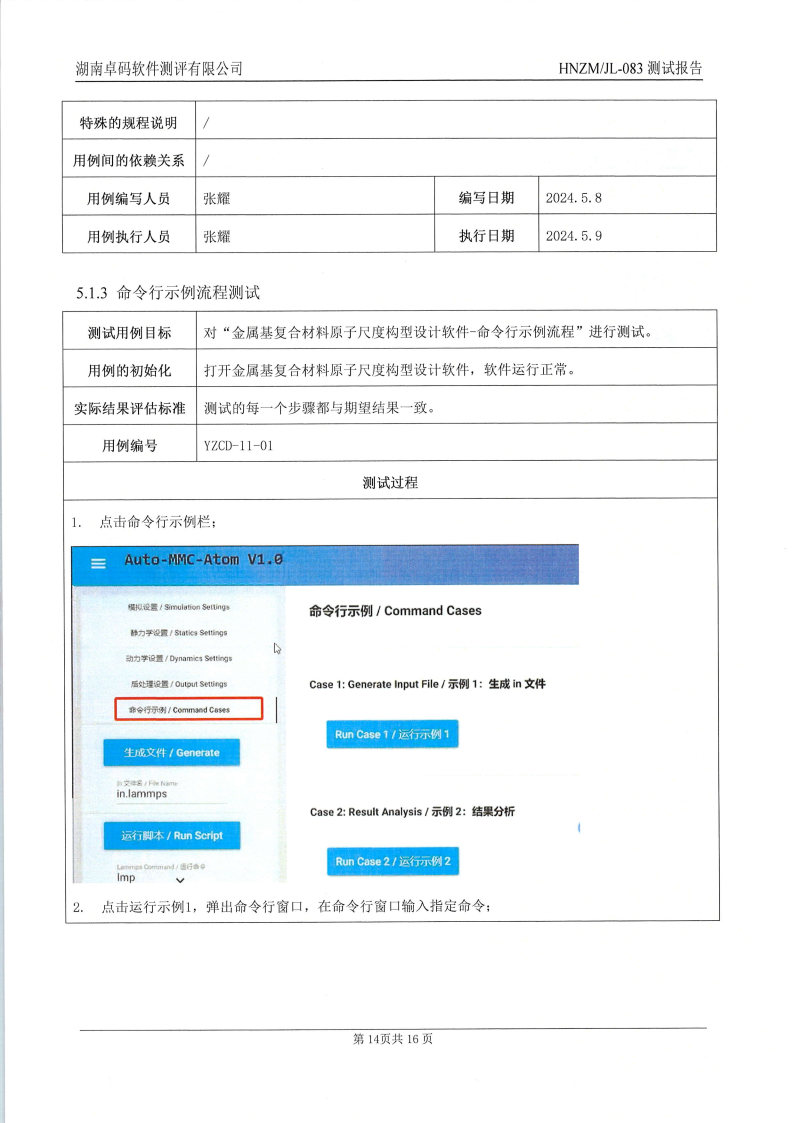


《金属基复合材料原子尺度构型设计软件》登记证书

## 《金属基复合材料原子尺度构型设计软件》第三方测评报告

委托湖南卓码软件测评有限公司，按照GB/T 25000.51-2016《系统与软件工程系统与软件质量要求和评价（SQuaRE）第51部分：就绪可用软件产品（RUSP）的质量要求和测试细则》中的条款，结合《HNZM\_ZY\_002\_软件产品检测作业指导书Ⅱ》对软件的功能性和性能效率进行测试。软件测试报告部分内容如下：



# 附表

## “金属基复合材料界面/增强体/界面的物性/变形参数与温度相关性计算模块”源代码

c\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

c programed by Mingyu on 02/16/23 in SJTU

c post-processing of LAMMPS data

c\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

program post\_processing

implicit real\*8 (a-h,o-z)

implicit integer (i-n)

write(\*,\*)'Select function:'

write(\*,\*)'(1)Calculate cohesive energy'

write(\*,\*)'(2)Calculate defect formation energy'

write(\*,\*)'(3)Calculate interface energy (horizontal)'

write(\*,\*)'(4)Calculate dislocation core energy'

write(\*,\*)'(5)Calculate kinetic barrier'

read(\*,\*)ifunction

write(\*,\*)'-------------------Running-------------------'

if (ifunction.eq.1) then

call COHESIVE\_ENERGY('final.data')

elseif (ifunction.eq.2) then

call DEFECT\_ENERGY('final.data',pot\_def)

write(\*,'(A,f8.2,A)')' Formation energy: ',pot\_def,' eV'

elseif (ifunction.eq.3) then

call INTERFACE\_ENERGY('final.data',pot\_int)

write(\*,'(A,f8.2,A)')' Interface energy: ',pot\_int,' mJ/m^2'

elseif (ifunction.eq.4) then

call CORE\_ENERGY('final.data',pot\_core,rr\_core)

write(\*,'(A,f8.2,A)')' Core width: ',rr\_core,' Å'

write(\*,'(A,f8.2,A)')' Core energy: ',pot\_core,' eV/Å'

elseif (ifunction.eq.5) then

call NEB\_BARRIER('log.lammps',pot\_bar)

write(\*,'(A,e10.2,A)')' Kinetic barrier: ',pot\_bar,' eV'

endif

write(\*,\*)'---------------------End---------------------'

end

subroutine CALC\_LINE(filename,nlines)

implicit real\*8 (a-h,o-z)

implicit integer (i-n)

character\*16 filename

open(unit=1,file=filename,status='unknown')

nlines=0

do

read(1,\*, END=1000)

nlines=nlines+1

end do

1000 close(1)

end

SUBROUTINE COHESIVE\_ENERGY(filename)

implicit real\*8 (a-h,o-z)

implicit integer (i-n)

parameter (n20=20)

parameter (nn=5000000)

dimension ntype(nn),xr(3,nn),pr(nn)

dimension pot\_tot(n20),ncnt(n20),pot\_coh(n20)

character\*16 filename

open(unit=1,file=filename,status='unknown')

read(1,\*)

read(1,\*)

read(1,\*)

read(1,\*)nrmax

read(1,\*)

read(1,\*)xrmin,xrmax

read(1,\*)yrmin,yrmax

read(1,\*)zrmin,zrmax

read(1,\*)

ntype\_max=0

do i=1,nrmax

read(1,\*)ntmp,ntype(ntmp),xr(1,ntmp),xr(2,ntmp),xr(3,ntmp),

$ pr(ntmp)

if (ntype(ntmp).gt.ntype\_max) then

ntype\_max=ntype(ntmp)

endif

enddo

close(1)

do i=1,ntype\_max

pot\_tot(i)=0.0

ncnt(i)=0

enddo

do i=1,nrmax

pot\_tot(ntype(i))=pot\_tot(ntype(i))+pr(i)

ncnt(ntype(i))=ncnt(ntype(i))+1

enddo

do i=1,ntype\_max

pot\_coh(i)=pot\_tot(i)/ncnt(i)

enddo

if (ntype\_max.eq.1) then

write(\*,'(A,A)')' Cohesive energy for element #1',' is:'

write(\*,'(f8.4)')pot\_coh(1)

else

n1=NINT(LOG10(REAL(ntype\_max)))+1

write(\*,'(A,i<n1>,A)')' Cohesive energy for element #1-#',

$ ntype\_max,' is:'

write(\*,'(<ntype\_max>f13.8,A)')(pot\_coh(i),i=1,ntype\_max),

$ ' Unit: eV'

endif

END

SUBROUTINE DEFECT\_ENERGY(filename,pot\_def)

implicit real\*8 (a-h,o-z)

implicit integer (i-n)

parameter (n20=20)

parameter (nn=5000000)

dimension ntype(nn),xr(3,nn),pr(nn)

dimension pot\_ref(n20)

character\*16 filename

open(unit=1,file=filename,status='unknown')

read(1,\*)

read(1,\*)

read(1,\*)

read(1,\*)nrmax

read(1,\*)

read(1,\*)xrmin,xrmax

read(1,\*)yrmin,yrmax

read(1,\*)zrmin,zrmax

read(1,\*)

ntype\_max=0

do i=1,nrmax

read(1,\*)ntmp,ntype(ntmp),xr(1,ntmp),xr(2,ntmp),xr(3,ntmp),

$ pr(ntmp)

if (ntype(ntmp).gt.ntype\_max) then

ntype\_max=ntype(ntmp)

endif

enddo

close(1)

if (ntype\_max.eq.1) then ! get ewf pot

write(\*,\*)'Enter cohesive energy for element #1'

read(\*,\*)pot\_ref(1)

else

n1=NINT(LOG10(REAL(ntype\_max)))+1

write(\*,'(A,i<n1>)')' Enter cohesive energy for element #1-#',

$ ntype\_max

read(\*,\*)(pot\_ref(i),i=1,ntype\_max)

endif ! end get

do i=1,nrmax ! calculate excess pot

pr(i)=pr(i)-pot\_ref(ntype(i))

enddo ! end calculate

dummy1=-99999999.0 ! find defect

dummy2=-99999999.0

dtmp=(xrmax-xrmin)/1000.0

do i=100,900

xtmp\_min=xrmin+dtmp\*(i-1)

xtmp\_max=xrmin+dtmp\*i

dummy3=0.0

do j=1,nrmax

if (xr(1,j).ge.xtmp\_min .and. xr(1,j).lt.xtmp\_max .and.

$ xr(2,j).ge.0.9\*yrmin+0.1\*yrmax .and.

$ xr(2,j).le.0.1\*yrmin+0.9\*yrmax) then

dummy3=dummy3+pr(j)

endif

enddo

if (dummy3.gt.dummy1) then

dummy2=dummy1

dummy1=dummy3

x\_mark2=x\_mark1

x\_mark1=(xtmp\_min+xtmp\_max)/2.0

elseif (dummy3.gt.dummy2 .and. dummy3.le.dummy1) then

dummy2=dummy3

x\_mark2=(xtmp\_min+xtmp\_max)/2.0

endif

enddo

xr\_def=(x\_mark1+x\_mark2)/2.0

dummy1=-99999999.0

dummy2=-99999999.0

dtmp=(yrmax-yrmin)/1000.0

do i=100,900

ytmp\_min=yrmin+dtmp\*(i-1)

ytmp\_max=yrmin+dtmp\*i

dummy3=0.0

do j=1,nrmax

if (xr(2,j).ge.ytmp\_min .and. xr(2,j).lt.ytmp\_max .and.

$ xr(1,j).ge.0.9\*xrmin+0.1\*xrmax .and.

$ xr(1,j).le.0.1\*xrmin+0.9\*xrmax) then

dummy3=dummy3+pr(j)

endif

enddo

if (dummy3.gt.dummy1) then

dummy2=dummy1

dummy1=dummy3

y\_mark2=y\_mark1

y\_mark1=(ytmp\_min+ytmp\_max)/2.0

elseif (dummy3.gt.dummy2 .and. dummy3.le.dummy1) then

dummy2=dummy3

y\_mark2=(ytmp\_min+ytmp\_max)/2.0

endif

enddo

yr\_def=(y\_mark1+y\_mark2)/2.0

dummy1=-99999999.0

dummy2=-99999999.0

dtmp=(zrmax-zrmin)/1000.0

do i=1,1000

ztmp\_min=zrmin+dtmp\*(i-1)

ztmp\_max=zrmin+dtmp\*i

dummy3=0.0

do j=1,nrmax

if (xr(3,j).ge.ztmp\_min .and. xr(3,j).lt.ztmp\_max .and.

$ xr(1,j).ge.0.9\*xrmin+0.1\*xrmax .and.

$ xr(1,j).le.0.1\*xrmin+0.9\*xrmax .and.

$ xr(2,j).ge.0.9\*yrmin+0.1\*yrmax .and.

$ xr(2,j).le.0.1\*yrmin+0.9\*yrmax) then

dummy3=dummy3+pr(j)

endif

enddo

if (dummy3.gt.dummy1) then

dummy2=dummy1

dummy1=dummy3

z\_mark2=z\_mark1

z\_mark1=(ztmp\_min+ztmp\_max)/2.0

elseif (dummy3.gt.dummy2 .and. dummy3.le.dummy1) then

dummy2=dummy3

z\_mark2=(ztmp\_min+ztmp\_max)/2.0

endif

enddo

zr\_def=(z\_mark1+z\_mark2)/2.0

write(\*,'(A,f8.2,A,f8.2,A,f8.2,A)')' Position of defect: x=',

$ xr\_def,' Å, y=',yr\_def,' Å, z=',

$ zr\_def,' Å' ! end find

xbound\_min=(xr\_def+xrmin)/2.0

xbound\_max=(xr\_def+xrmax)/2.0

ybound\_min=(yr\_def+yrmin)/2.0

ybound\_max=(yr\_def+yrmax)/2.0

pot\_def=0.0

do i=1,nrmax

if (xr(1,i).ge.xbound\_min .and. xr(1,i).le.xbound\_max .and.

$ xr(2,i).ge.ybound\_min .and. xr(2,i).le.ybound\_max) then

pot\_def=pot\_def+pr(i)

endif

enddo

END

SUBROUTINE INTERFACE\_ENERGY(filename,pot\_int)

implicit real\*8 (a-h,o-z)

implicit integer (i-n)

parameter (n20=20)

parameter (nn=5000000)

dimension ntype(nn),xr(3,nn),pr(nn)

dimension pot\_ref(n20)

character\*16 filename

open(unit=1,file=filename,status='unknown')

read(1,\*)

read(1,\*)

read(1,\*)

read(1,\*)nrmax

read(1,\*)

read(1,\*)xrmin,xrmax

read(1,\*)yrmin,yrmax

read(1,\*)zrmin,zrmax

read(1,\*)

ntype\_max=0

do i=1,nrmax

read(1,\*)ntmp,ntype(ntmp),xr(1,ntmp),xr(2,ntmp),xr(3,ntmp),

$ pr(ntmp)

if (ntype(ntmp).gt.ntype\_max) then

ntype\_max=ntype(ntmp)

endif

enddo

close(1)

if (ntype\_max.eq.1) then ! get coh pot

write(\*,\*)'Enter cohesive energy for element #1'

read(\*,\*)pot\_ref(1)

else

n1=NINT(LOG10(REAL(ntype\_max)))+1

write(\*,'(A,i<n1>)')' Enter cohesive energy for element #1-#',

$ ntype\_max

read(\*,\*)(pot\_ref(i),i=1,ntype\_max)

endif ! end get

do i=1,nrmax ! calculate excess pot

pr(i)=pr(i)-pot\_ref(ntype(i))

enddo ! end calculate

dummy1=-99999999.0 ! find interface

dummy2=-99999999.0

dtmp=(yrmax-yrmin)/1000.0

do i=100,900

ytmp\_min=yrmin+dtmp\*(i-1)

ytmp\_max=yrmin+dtmp\*i

dummy3=0.0

do j=1,nrmax

if (xr(2,j).ge.ytmp\_min .and. xr(2,j).lt.ytmp\_max) then

dummy3=dummy3+pr(j)

endif

enddo

if (dummy3.gt.dummy1) then

dummy2=dummy1

dummy1=dummy3

y\_mark2=y\_mark1

y\_mark1=(ytmp\_min+ytmp\_max)/2.0

elseif (dummy3.gt.dummy2 .and. dummy3.le.dummy1) then

dummy2=dummy3

y\_mark2=(ytmp\_min+ytmp\_max)/2.0

endif

enddo

yr\_int=(y\_mark1+y\_mark2)/2.0

write(\*,'(A,f8.2,A)')' Position of interface: y=',

$ yr\_int,' Å' ! end find

ybound\_min=(yr\_int+yrmin)/2.0

ybound\_max=(yr\_int+yrmax)/2.0

pot\_tot=0.0

do i=1,nrmax

if (xr(2,i).ge.ybound\_min .and. xr(2,i).le.ybound\_max) then

pot\_tot=pot\_tot+pr(i)

endif

enddo

pot\_int=pot\_tot/(xrmax-xrmin)/(zrmax-zrmin)\*16021.8

END

SUBROUTINE CORE\_ENERGY(filename,pot\_core,rr\_core)

implicit real\*8 (a-h,o-z)

implicit integer (i-n)

parameter (n20=20)

parameter (n200=200)

parameter (nn=5000000)

dimension ntype(nn),xr(3,nn),pr(nn)

dimension pot\_ref(n20)

dimension rr(n200),pot\_line(n200)

character\*16 filename

open(unit=1,file=filename,status='unknown')

read(1,\*)

read(1,\*)

read(1,\*)

read(1,\*)nrmax

read(1,\*)

read(1,\*)xrmin,xrmax

read(1,\*)yrmin,yrmax

read(1,\*)zrmin,zrmax

read(1,\*)

ntype\_max=0

do i=1,nrmax

read(1,\*)ntmp,ntype(ntmp),xr(1,ntmp),xr(2,ntmp),xr(3,ntmp),

$ pr(ntmp)

if (ntype(ntmp).gt.ntype\_max) then

ntype\_max=ntype(ntmp)

endif

enddo

close(1)

if (ntype\_max.eq.1) then ! get coh pot

write(\*,\*)'Enter cohesive energy for element #1'

read(\*,\*)pot\_ref(1)

else

n1=NINT(LOG10(REAL(ntype\_max)))+1

write(\*,'(A,i<n1>)')' Enter cohesive energy for element #1-#',

$ ntype\_max

read(\*,\*)(pot\_ref(i),i=1,ntype\_max)

endif ! end get

do i=1,nrmax ! calculate excess pot

pr(i)=pr(i)-pot\_ref(ntype(i))

enddo ! end calculate

dummy1=-99999999.0 ! find dislocation

dummy2=-99999999.0

dtmp=(xrmax-xrmin)/1000.0

do i=100,900

xtmp\_min=xrmin+dtmp\*(i-1)

xtmp\_max=xrmin+dtmp\*i

dummy3=0.0

do j=1,nrmax

if (xr(1,j).ge.xtmp\_min .and. xr(1,j).lt.xtmp\_max .and.

$ xr(2,j).ge.0.9\*yrmin+0.1\*yrmax .and.

$ xr(2,j).le.0.1\*yrmin+0.9\*yrmax) then

dummy3=dummy3+pr(j)

endif

enddo

if (dummy3.gt.dummy1) then

dummy2=dummy1

dummy1=dummy3

x\_mark2=x\_mark1

x\_mark1=(xtmp\_min+xtmp\_max)/2.0

elseif (dummy3.gt.dummy2 .and. dummy3.le.dummy1) then

dummy2=dummy3

x\_mark2=(xtmp\_min+xtmp\_max)/2.0

endif

enddo

xr\_dis=(x\_mark1+x\_mark2)/2.0

dummy1=-99999999.0

dummy2=-99999999.0

dtmp=(yrmax-yrmin)/1000.0

do i=1,1000

ytmp\_min=yrmin+dtmp\*(i-1)

ytmp\_max=yrmin+dtmp\*i

dummy3=0.0

do j=1,nrmax

if (xr(2,j).ge.ytmp\_min .and. xr(2,j).lt.ytmp\_max .and.

$ xr(1,j).ge.0.9\*xrmin+0.1\*xrmax .and.

$ xr(1,j).le.0.1\*xrmin+0.9\*xrmax) then

dummy3=dummy3+pr(j)

endif

enddo

if (dummy3.gt.dummy1) then

dummy2=dummy1

dummy1=dummy3

y\_mark2=y\_mark1

y\_mark1=(ytmp\_min+ytmp\_max)/2.0

elseif (dummy3.gt.dummy2 .and. dummy3.le.dummy1) then

dummy2=dummy3

y\_mark2=(ytmp\_min+ytmp\_max)/2.0

endif

enddo

yr\_dis=(y\_mark1+y\_mark2)/2.0

write(\*,'(A,f8.2,A,f8.2,A)')' Position of dislocation: x=',

$ xr\_dis,' Å, y=',yr\_dis,' Å' ! end find

dummy1=xrmax-xr\_dis ! get E-Ln(r) data

dummy2=xr\_dis-xrmin

dummy3=yrmax-yr\_dis

dummy4=yr\_dis-yrmin

rr\_max=0.5\*MIN(dummy1,dummy2,dummy3,dummy4)

do i=0,100

rr\_tmp=1.0+rr\_max/100.0\*i

rr(i)=LOG(rr\_tmp)

pot\_line(i)=0.0

do j=1,nrmax

dummy=SQRT((xr(1,j)-xr\_dis)\*\*2+(xr(2,j)-yr\_dis)\*\*2)

if (dummy.le.rr\_tmp) then

pot\_line(i)=pot\_line(i)+pr(j)/(zrmax-zrmin)

endif

enddo

enddo ! end get

tk\_ref=(pot\_line(100)-pot\_line(98))/(rr(100)-rr(98))

do i=99,0,-1

tk\_tmp=(pot\_line(i+1)-pot\_line(i-1))/(rr(i+1)-rr(i-1))

dummy=ABS((tk\_tmp-tk\_ref)/tk\_ref)

if (dummy.le.0.25) then

tk\_ref=tk\_tmp

else

rr\_core=EXP(rr(i))

pot\_core=pot\_line(i)

go to 2000

endif

enddo

2000 END

SUBROUTINE NEB\_BARRIER(filename,pot\_bar)

implicit real\*8 (a-h,o-z)

implicit integer (i-n)

parameter (n20=100)

dimension pot\_ref(n20)

character\*16 filename

write(\*,\*)'Enter number of states'

read(\*,\*)istate

call CALC\_LINE(filename,nlines)

open(unit=1,file=filename,status='unknown')

do i=1,nlines-1

read(1,\*)

enddo

read(1,\*)tmp1,tmp2,tmp3,

$ tmp4,tmp5,tmp6,

$ tmp7,tmp8,tmp9,

$ (pot\_ref(j),j=1,2\*istate)

pot\_tmp\_min=99999999

pot\_tmp\_max=-99999999

do i=1,istate

if (pot\_ref(2\*i).lt.pot\_tmp\_min) then

pot\_tmp\_min=pot\_ref(2\*i)

endif

if (pot\_ref(2\*i).gt.pot\_tmp\_max) then

pot\_tmp\_max=pot\_ref(2\*i)

endif

enddo

pot\_bar=pot\_tmp\_max-pot\_tmp\_min

c write(\*,'(<istate>f8.2)')(pot\_ref(j),j=1,2\*istate,2)

c write(\*,'(<istate>f8.3)')(pot\_ref(j)-pot\_ref(2),j=2,2\*istate,2)

END