

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

Кафедра вычислительной физики  
и моделирования физических процессов

А. А. Цыганков

АТОМІСРЕХ. РУКОВОДСТВО ПО  
ИСПОЛЬЗОВАНИЮ

Казань-2020

# Содержание

Введение	4
Установка и компиляция	4
<b>1 Краткий обзор</b>	<b>8</b>
Ключевые аспекты . . . . .	8
Степени свободы . . . . .	8
Свойства . . . . .	9
Целевая функция . . . . .	10
<b>2 Использование. Основные принципы.</b>	<b>13</b>
Основной файл . . . . .	13
<b>3 Оптимизационные алгоритмы</b>	<b>14</b>
L-BFGS минимизатор . . . . .	15
Sra минимизатор . . . . .	16
Оптимизация через NLopt . . . . .	16
<b>4 Потенциалы</b>	<b>18</b>
Потенциал Леннард-Джонса . . . . .	20
Метод погруженного атома (EAM) . . . . .	21
Потенциал Терсоффа . . . . .	24
Потенциал "связи"(ABOP – Analytic Bond Order Potential) . . . . .	26
Модифицированный метод погруженного атома (MEAM) . . . . .	27
Табулированные потенциалы. Определение функций . . . . .	30
Функции экранирования . . . . .	31
Интерполяционные функции . . . . .	32
Сплайны . . . . .	33
Спецфункции . . . . .	34
Комбинация функций . . . . .	37
Пользовательские функции . . . . .	37
<b>5 Структуры</b>	<b>39</b>
Унарные кристаллические структуры . . . . .	40
Бинарные и тройные кристаллические структуры . . . . .	42
Пользовательские структуры . . . . .	44
Структуры с точечными дефектами . . . . .	47
Релаксация . . . . .	48
Деформации . . . . .	49

<b>6</b>	<b>Свойства</b>	<b>51</b>
	Структурные свойства . . . . .	51
	Объявляемые свойства . . . . .	54
<b>7</b>	<b>Примеры</b>	<b>54</b>
	<b>Список литературы</b>	<b>57</b>

# Введение

Atomicrex представляет собой вычислительный пакет, предназначенный для построения потенциалов взаимодействия элементов многочастичной системы. В качестве элементов могут быть атомы или коллоидные частицы. Пакет разработан коллективом ученых Чалмерского технического университета, Швеция. Особенностью данного пакета является то, что построение потенциалов происходит на основе известных значений физических характеристик рассматриваемой физической системы. В настоящем пособии демонстрируется процедура установки пакета "Atomicrex" на персональный компьютер, приводятся демонстрационные примеры по построению модельных потенциалов межчастичного взаимодействия. Для полноценного использования данного пакета потребуются знания языка XML (eXtensible Markup Language), а также опыт работы с операционными системами семейства Linux (здесь используется Ubuntu 18.04). Пакет позволяет использовать конструкции языка программирования Python. И, наконец, поддерживаются следующие типы (семейства) потенциалов:

- Потенциал Леннарда-Джонса.
- Модель погруженного атома (Embedded Atom Model) и его модификация (Modified Embedded Atom Model).
- Потенциал Терсоффа.
- Потенциал "связи" (АВОР – analytic bond-order potential).
- Потенциалы, заданные в табличном виде.
- Потенциалы, аналитический вид которых задается пользователем.

Официальный сайт вычислительного пакета: <https://www.atomicrex.org/index.html>

В данном учебном пособии будет использоваться следующая маркировка текста:

- Текст, выделенный **синим**, представляет примеры команд при установке пакета и примеры программ.
- Текст на **зеленом фоне** – команды, используемые в программе.
- Текст на **оранжевом фоне** – информация, на которую следует обратить внимание.
- Текст в **белой рамке** – относительный путь к примерам кода, который можно запускать и использовать.

## Установка и компиляция

Исходный код программы доступен в репозитории GitLab. Его можно скачать оттуда с помощью команды `git`:

```
git clone https://gitlab.com/atomicrex/atomicrex.git
```

После выполнения этого действия в текущей директории появится папка с названием "atomicrex" содержащая исходный код программы. Для сборки пакета понадобятся следующие программные продукты и библиотеки:

1. Компиляторы C++ и Fortran,
2. Система сборки CMake,
3. Системный инструмент `xxd`,
4. Внешние библиотеки `boost` (версия 1.67.0 или выше), `libxml2`, `muparser`, `nlopt`,
5. Интерпретатор Python с модулями `Sphinx` и `Numpy`.

В операционной системе Ubuntu Linux все эти программы могут быть установлены следующими командами:

```
sudo apt-get install \  
python-dev \  
python-sphinx \  
python-numpy \  
libboost-dev \  
libxml2-dev \  
libnlopt-dev \  
libmuparser-dev \  
build-essential \  
gfortran \  
vim-common \  
cmake
```

Для того, чтобы скомпилировать исходный код программы, нужно создать новую директорию `build`, перейти в нее, и вызвать систему сборки CMake. Это можно сделать следующими командами:

```
mkdir build  
cd build  
cmake ..
```

При успешном выполнении CMake будет создан файл Makefile. Компиляция исходного кода Atomicrex может быть запущена с помощью команды:

```
$ make
```

Для создания пользовательской инструкции (для полноценной работы программы необязательно), необходимо указать соответствующий параметр:

```
$ make userguide
```

Сгенерированные HTML файлы находятся в директории: `doc/userguide/build`. Если CMake не находит библиотеки, то их нужно указать вручную. Для этого есть несколько вариантов. Так, например, можно указать параметр флага `-D` при команде CMake:

```
$ cmake -D NLOPT_INCLUDE_DIR=../path_to_success/  
-D NLOPT_LIBRARY=../path_to_success/
```

Кроме того, можно вызвать интерактивную версию CMake, выполнив:

```
$ ccmake
```

Заметьте, что обычно необходимо переключиться в «расширенный режим», чтобы получить доступ к настройкам пути для библиотек.

По умолчанию atomicrex компилируется как релизная версия, что подразумевает включение оптимизации. Если это нежелательно, например, с целью отладки, можно запросить отладочную версию, выполнив:

```
cmake -D CMAKE_BUILD_TYPE=Debug
```

В этом контексте также может быть полезно включить подробный вывод во время компиляции, что достигается запуском make следующим образом:

```
$ make VERBOSE=1
```

Если поддерживается установленный компилятор C++, atomicrex автоматически использует преимущества распараллеливания OpenMP. Чтобы явно отключить эту функцию, нужно сделать следующее:

```
$ cmake -D USE_OPENMP=OFF
```

Кроме того, есть несколько дополнительных функций. Использование библиотеки нелинейной оптимизации NLOpt можно отключить с помощью `ENABLE_NLOPT`

Библиотека нелинейной оптимизации NLOpt должна быть скомпилирована как библиотека (общая или разделяемая). В противном случае вы можете получить

следующее сообщение об ошибке, такое, как `.rodata.str1.1`, которое нельзя использовать при создании общего объекта / ошибки добавления символов: неверное значение. Чтобы обойти эту ошибку, скомпилируйте `NLopt`, используя опцию `-enable-shared`. Интерфейс Python может быть активирован или деактивирован с помощью опции `ENABLE_PYTHON_INTERFACE`.

# 1 Краткий обзор

Atomicrex работает с XML-файлами, в которых содержатся все команды, определяемые пользователем. Работа программы условно разделяется на два этапа: этап обучения и выходной этап. На этапе обучения выбранные степени свободы (параметры) модели, в основном, это межатомный потенциал, варьируются таким образом, что прогнозируемые свойства (энергии, силы, упругие постоянные и пр.) должны наиболее точно соответствовать целевым значениям. Этап обучения является необязательным и может быть пропущен, если не указывать элемент `<fitting>` в файле задания.

За этапом обучения следует выводной этап. Здесь можно рассчитать дополнительные свойства рассматриваемой физической системы, которые не были включены во время обучения. Это позволяет разделить имеющиеся данные на обучающий набор и тестовый набор. Последний набор свойств позволяет оценить качество модели.

**Ключевые аспекты.** Atomicrex работает с двумя основными типами объектов: потенциалами и структурами.

- Потенциал состоит из уравнения, его задающего, и набора параметров, позволяющих рассчитать полную энергию и силы для данной структуры.
- Структура состоит из ячейки моделирования с периодическими граничными условиями или же без неё, и списка атомов с координатами. Каждый из потенциалов и структур представляет собой набор степеней свободы и свойств, определение которых будет дано ниже.

**Степени свободы.** Степень свободы – это параметр потенциала, который может изменяться во время процесса построения для того, чтобы потенциал удовлетворял экспериментальным величинам при моделировании. Следует отметить, что степени свободы потенциала изменяется во время процесса обучения, чтобы минимизировать целевую функцию, в то время как степени свободы структуры релаксируются, приводя ее в состояние с наименьшей потенциальной энергией. Существуют разные классы степеней свободы: в самом простом случае это скалярная переменная (один параметр), например, параметр  $\sigma$  потенциала Леннард-Джонса или постоянная решётки. Кроме того, существуют многомерные степени свободы, которые управляют более сложными параметрами, например, координаты атомов структуры или коэффициенты в сплайнах при сглаживании функции. Также в программе это могут быть когезионная энергия, модуль всестороннего сжатия, тензор напряжений, атомные силы, и другие.

Каждому типу потенциала или структуры соответствует определённый набор степеней свободы. Пользователь должен указать начальное значение в фай-



ле задания для каждого из них. По умолчанию они являются статическими, то есть их значение не изменяется во время работы. Однако, используя парный тег `<fit-dof>`, пользователь может включить выбранные степени свободы в процесс обучения, как описано в разделе "Потенциалы".

Соответственно, релаксация степеней свободы структуры может быть включена с помощью парного тега `<fit-dof>` в файле задания, как описано в этом разделе(вставить). Если релаксация включена для структурной степени свободы, её значение будет изменяться таким образом, чтобы потенциальная энергия структуры была минимальной.

**Свойства.** Свойствами называют здесь величины, которые могут быть рассчитаны из степеней свободы структуры или потенциала, то есть, они являются функционалами степеней свободы.<sup>1</sup> Каждый тип структуры или потенциала обеспечивает определённый набор свойств. По умолчанию рассчитываются немногие из них, такие как потенциальная энергия и равновесный объем структуры. Как описано в разделе (Свойства), пользователь может включить вычисление дополнительных свойств, которые должны быть включены в подбор и / или в окончательный вывод, с помощью элемента `<property>`. При включении свойства в процесс построения потенциала необходимо указать для него целевое значение. В самом простом случае это может быть просто число, например, энергии структуры, или это могут быть многомерные данные, например векторы силы, действующие на атомы структуры. Каждое подходящее свойство даёт вклад в целевую функцию, которая минимизируется во время тренировочного процесса в соответствии с его весом, как описано ниже.

Обратите внимание, что свойства рассчитываются после релаксации степеней свободы структуры. Это подразумевает, что, если релаксация включена для определённых степеней свободы, процесс обучения состоит из двух вложенных проходов минимизации: каждый раз, когда целевая функция для потенциала в процессе обучения оценивается во внешнем цикле оптимизации, энергия всех структур сначала минимизируется относительно структурных степеней свободы.

Третий шаг вариации может существовать, если разрешена релаксация положений атомов. В этом случае координаты атомов релаксируются каждый раз, когда рассчитывается полная энергия для данной пробной конфигурации степеней свободы, для которых разрешена релаксация. Имейте это в виду, когда вычислительное время, требуемое для определённого прогона, становится неожиданно большим.

---

<sup>1</sup>Обратите внимание, что в случае структурных свойств расчёт их может быть довольно сложным и обычно включает в себя оценку функций потенциалов. Иногда, напротив, свойства потенциалов могут быть напрямую получены из его степеней свободы. Эта сложность эффективно устраняется внутренней структурой программы и обычно не влияет на пользователя.

Для некоторых свойств есть взаимно однозначное отношение между свойством и соответствующей степенью свободы. Например, параметр решётки структуры определяет межатомное расстояние. С другой стороны, может потребоваться подобрать интервал равновесной решётки к некоторому экспериментальному значению. Таким образом, структура решетки выставляет как параметр решетки в качестве его степени свободы, так и свойства параметра решётки. Если свойство включено в построение, релаксация будет включена для соответствующей степени свободы, даже если элемент `<relax>` не был указан.

**Целевая функция.** Целевая функция - это основная величина, вычисляемая с помощью `atomicrex`. Её можно записать в общем виде:

$$\chi^2 = \underbrace{\sum_G \bar{w}_G}_{\text{structure group}} \underbrace{\sum_S \bar{w}_{GS}}_{\text{structures}} \underbrace{\sum_P \bar{w}_{GSP} r_{GSP}}_{\text{properties}} \quad (1)$$

с

$$\bar{w}_{GSP} = w_{GSP} / \sum_{P'}^{properties} w_{GSP'} \quad (2)$$

$$\bar{w}_{GS} = w_{GS} / \sum_{S'}^{properties} w_{GS'} \quad (3)$$

$$\bar{w}_G = w_G / \sum_{G'}^{properties} w_{G'}, \quad (4)$$

где  $\bar{w}_G$ ,  $\bar{w}_{GS}$ ,  $\bar{w}_{GSP}$  обозначают относительные весовые факторы для структурных групп, структур, свойств соответственно, которые могут быть назначены через команду `<relative-weight>`.

**Важно!** Структурная группа может содержать как сами структуры, так и подгруппы, что позволяет создавать сложные конструкции. В таких случаях при вычислении весовых коэффициентов подгруппы группы рассматриваются как эквивалентные структурам в той же группе, то есть веса подгрупп включаются в вычисление весовых коэффициентов структуры  $w_{GS}$ . Это реализуется рекурсивно и гарантируется, что все веса нормированы на единицу:

$$\sum_{GSP} \bar{w}_{GSP} = 1, \sum_{GS} \bar{w}_{GS} = 1, \sum_G \bar{w}_G = 1 \quad (5)$$

Остаточное значение  $r_{GSP}$  рассчитывается для каждого свойства из прогнозируемого  $A^{predicted}$  и целевых значений  $A^{target}$  в соответствии с указанным `residual-style`,

который устанавливается через одноименный атрибут элемента свойств. Например, команда `squared` подразумевает

$$r_{GSP} = \left[ (A_i^{predicted} - A_i^{target}) / \delta_{GSP} \right]^2 \quad (6)$$

Здесь  $\delta_{GSP}$  – точность, которая для свойства определена заранее. Заметьте, что не все настройки точности вызывают этот параметр, см. параграф "Свойства". Обоснование этой модели вложенных весов заключается в следующем. При тренировке межатомных потенциалов определенные структуры обычно более важны, чем другие. Например, в случае кремния алмазная структура, естественно, должна иметь больший вес, чем, скажем, гранецентрированная кубическая структура, тогда как в случае алюминия или золота применяется противоположный сценарий. Для достижения этого баланса можно установить веса отдельных структур ( $\bar{w}_{GS}$ ), а также веса структурных групп ( $w_G$ ). Обратите внимание, что вес структуры и группы структур нормализуется индивидуально (см. выше), так что соотношение между структурами остается неизменным при изменении весов отдельных свойств.

**Важно!** Все контролируемые пользователем весовые коэффициенты,  $\bar{w}_G$ ,  $\bar{w}_{GS}$ ,  $\bar{w}_{GSP}$ , а также параметры точности (допуска),  $\delta_{GSP}$  по умолчанию равно единице. Если вы предпочитаете работать с моделью весов одного уровня (один и только один весовой коэффициент на свойство), он должен работать только со значениями  $w_{GSP}$  (команды `relative-weight` и `<property>`) и оставить другие веса  $w_G$  и  $w_{GS}$ , а также атрибут допуска  $\delta_{GSP}$  нетронутым.

Аналогично, также желательно использовать «интуитивные» веса, чтобы выразить, например, тот факт, что когезионная энергия определенной структуры должна быть «в три раза более важной», чем модуль всестороннего сжатия. Эти свойства, однако, имеют очень разные единицы и, следовательно, различия между прогнозируемыми и целевыми значениями  $A^{predicted} - A^{target}$  могут быть очень разными. Обычно неудобно корректировать веса свойств вручную, чтобы исправить этот дисбаланс. Можно частично исправить ситуацию, используя `squared-relative` стиль, который, однако, налагает нормализацию в смысле абсолютного значения свойства.

Более тонкий подход заключается в использовании параметра допуска  $\delta$ , который позволяет указать приемлемый диапазон для каждого свойства. Естественно, он несет ту же единицу, что и свойство, так что все остатки безразмерны. Это позволяет достаточно разумно использовать этот параметр. Например, приемлемо устанавливать точность для когезионной энергии, скажем  $\delta = 0.1$  эВ целевым значением, тогда как для упругой постоянной  $\delta = 5$  ГПа может быть приемлемо. Задание допусков с помощью атрибута `tolerance` следует выполнять при настрой-

ке базы данных подгонки, после чего параметры  $\delta$  можно оставить, как есть. Впоследствии настройка баланса соответствия происходит путем корректировки весов отдельных структур, структурных групп и свойств. Начальные настройки таковы (они могут быть выбраны пользователем через атрибут `<tolerance-preset>`):

- универсальные `uniform`:  $\delta_i = 1$  для всех параметров.
- сбалансированные `balanced`:

Единица	эВ	эВ/атом	ГПа	Å	мэВ	мэВ/Å	эВ/Å
Точность	0.2	0.01	2.0	0.005	10	10	0.01

- Повышенной точности для энергий `accurate-energies` :

Единица	эВ	эВ/атом	ГПа	Å	мэВ	мэВ/Å	эВ/Å
Точность	0.1	0.005	2.0	0.005	5	10	0.01

## 2 Использование. Основные принципы.

Atomicrex запускается с помощью следующей команды:

```
./atomicrex <имя файла без угловых скобок с расширением XML>
```

**Основной файл.** Файл задания контролирует работу программы atomicrex. Он определяет потенциал, который будет установлен, и атомные структуры и их свойства, которые используются для процесса построения. Общая структура его такая:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<job>
  <!-- основные настройки -->
  <name>...</name>
  <verbosity>...</verbosity>
  <real-precision>...</real-precision>
  <validate-potentials />

  <!-- укажите различные типы атомов -->
  <atom-types>
    ...
  </atom-types>

  <!-- определите процедуру оптимизации -->
  <fitting>
    ...
  </fitting>

  <!-- определите форму потенциала и его параметры -->
  <potentials>
    ...
  </potentials>

  <!-- определите структуры и свойства -->
  <structures>
```

```
...
</structures>
</job>
```

После заголовочного тега XML корневой элемент `<job>` содержит несколько тегов и разделов. Здесь несколько основных настроек:

- Опциональный блок `<name>` определяет имя отдельной части рабочего файла.
- Детальность вывода контролируется с помощью тега `<verbosity>`, который поддерживает настройки: `none`, `minimum`, `maximum`, `debug`.
- Точность ввода/вывода с плавающей точкой контролируется с помощью команды `<real-precision>`, аргумент – целое число.
- Тег `<validate-potentials>` используется для отдельных целей и обычно используется только во время отладки и отслеживания ошибок. Он обеспечивает численную оценку сил и сравнивает результат с результатами расчета сил, реализованными в исходном коде. Эта команда позволяет проверить правильность последней реализации. Однако, поскольку численный расчет почти всегда гораздо медленнее, рекомендуется тег не активировать намеренно.

Затем следует блок `<atom-types>`, в котором пользователь должен указать типы атомов в этом формате:

```
<atom-types>
  <species mass='69.723' atomic-number='31'>Ga</species>
  <species mass='22.989' atomic-number='11'>N</species>
</atom-types>
```

Атрибуты `mass`, `atomic-number` опциональны. Основная часть входных параметров затем помещается в трех заключительных разделах, которые определяют потенциалы, структуры, включенные в построение, и общие параметры, которые управляют этим процессом. Процедура оптимизации указана в блоке `<fitting>`, который описан в разделе "Оптимизационные алгоритмы". Блоки `<potentials>` и `<structures>` описаны в разделах "Потенциалы" и "Структуры".

### 3 Оптимизационные алгоритмы

Блок `<fitting>` во входном файле программы контролирует процедуру оптимизации. Он может быть дополнен двумя опциональными атрибутами как показано на этом примере:

```
<fitting enabled="true" output-interval="100">
  ...
</fitting>
```

Здесь атрибут `enabled` может быть использован для выборочного включения или отключения (`True / False`) оптимизации модели без удаления целого блока. `output-interval` устанавливает число итераций, после которых выводится на экран текущий статус (параметры потенциала, данные о свойствах и др.). В настоящее время `atomicrnx` поддерживает напрямую следующие оптимизационные алгоритмы:

- Минимизатор с ограниченной памятью Бройдена-Флетчера-Голдфарба-Шэнно (L-BFGS).
- Sра минимизатор.
- Алгоритмы из библиотеки `NLopt`.

Есть ещё несколько необязательных параметров, которые предоставляются в виде атрибутов в теге минимизатора (примеры приведены ниже). Они позволяют изменять критерии, приводящие к прекращению цикла минимизации. Первый параметр `conv-threshold` определяет условие для максимального изменения целевой функции между последовательными итерациями  $k$  и  $k + 1$ . Минимизация останавливается, если

$$\frac{\chi_k^2 - \chi_{k+1}^2}{\max(|\chi_k^2|, \chi_{k+1}^2, 1)} \leq c_t p_m, \quad (7)$$

где  $c_t$  задается параметром `conv-threshold` и  $p_m$  – машинная точность, которая автоматически генерируется. По умолчанию равна  $10^{-5}$ . Если число итераций превышает значение `max-iter`, оптимизация останавливается. Начальное значение равно 100. Градиент целевой функции оценивается численно в точке  $x_0$  с вычислением значения функции  $\chi^2$  между двумя равноудаленными малыми точками:

$$\left. \frac{d\chi^2}{dx} \right|_{x_0} \approx \frac{\chi^2(x_0 + \epsilon) - \chi^2(x_0 - \epsilon)}{2\epsilon} + O(\epsilon^2) \quad (8)$$

Ширина шага  $\epsilon$  определяется с использованием атрибута `gradient-epsilon` и по умолчанию равна  $10^{-6}$ .

**L-BFGS минимизатор.** В состав `atomicrnx` входит реализация минимизатора Бройдена-Флетчера-Голдфарба-Шэнно (L-BFGS) с ограниченной памятью, который является популярным квазиньютоновским методом с поддержкой ограничений. Здесь не будет его описания, будет рассмотрен лишь механизм его работы. Активируется тегом `BFGS` в блоке `<fitting>` в основном входном файле со следующими опциями:

```
<fitting>
  <BFGS conv-threshold="1e-16" max-iter="200" gradient-epsilon="1e-8" />
</fitting>
```

где начальные значения дополнительных параметров переопределены.

**Spa минимизатор.** Минимизатор Spa на самом деле не является отдельным алгоритмом оптимизации, а работает как оболочка. Он генерирует случайные начальные наборы параметров, а затем вызывает другой алгоритм минимизации для оптимизации начального набора параметров. Следующий блок кода иллюстрирует использование этого минимизатора:

```
<fitting>
  <spa max-iter="25" seed="120815">
    <BFGS conv-threshold="1e-10" max-iter="50" />
  </spa>
</fitting>
```

Атрибут `max-iter` определяет число раз, когда генерируется новый начальный набор параметров. Атрибут `seeds` позволяет установить начальное значение генератора случайных чисел. Здесь используется минимизатор L-BFGS, описанный здесь (вставить ссылку). Заметьте, что максимальное число итераций для последнего установлено на довольно низкое значение. Эта процедура рекомендуется, чтобы избежать много итераций во внутреннем цикле на пошаговую оптимизацию набора параметров, а отбирать большее количество различных исходных наборов параметров. Если был найден набор параметров, целевая функция которого наиболее удовлетворяет, чем для любого из ранее найденных наборов, новый набор записывается в файл и, в зависимости от значения тега `<verbosity>`, выводится.

**Внимание!** При использовании минимизатора Spa настоятельно рекомендуется также определять границы для таких свойств, как параметры решетки. В противном случае в процессе минимизации свойство может быстро разойтись и привести к необоснованным результатам. Границы свойств устанавливаются с помощью ключевых слов `min` и `max` соответствующего элемента.

Обратите внимание, что оптимизаторы, выполняющие «глобальный» поиск минимума, такие, как минимизатор Spa, а также глобальные алгоритмы из библиотеки NLopt, обычно не очень тщательно минимизируют отдельные точки. Поэтому обычно настоятельно рекомендуется оптимизировать минимумы из этих алгоритмов, дополнительно используя обычные (локальные) минимизаторы с более строгими критериями сходимости.

**Оптимизация через NLopt.** У Atomicrex есть интерфейс с библиотекой с открытым исходным кодом NLopt для нелинейной оптимизации, которая предоставляет большое количество алгоритмов как для глобальной оптимизации, так и для



локальной оптимизации без производных и локальных градиентных методов. Эти алгоритмы могут быть вызваны, как показано в следующем примере. Полный пример можно найти здесь (вставить ссылку):

```
<fitting>
  <nlopt algorithm="LN_NELDERMEAD" stopval="1e-10"/>
</fitting>
```

Алгоритм выбирается с использованием обязательного атрибута `algorithm`, который может принимать любое из следующих строковых значений:

Алгоритмы, основанные на локальных градиентах:

- `LD_LBFGS`: с ограниченной памятью Бройдена-Флетчера-Голдфарба-Шэнно.
- `LD_MMA`: метод движущихся асимптот (ММА) и консервативное выпуклое сепарабельное приближение (CCSA).

Локальные алгоритмы, не содержащие производных:

- `LN_NELDERMEAD`: симплекс-метод Нелдера-Мида.
- `LN_SBPLX`: Sbpplx (основан на Subplex).

Глобальные алгоритмы:

- `GN_CRSS2_LM`: Контролируемый случайный поиск (CRS) с локальной мутацией.
- `GN_ESCH`: ESCH (эволюционный алгоритм).
- `GN_DIRECT`: DIRECT (метод разделяющих прямоугольников – DИviding RECTangles) и "локально определенный"DIRECT-L.
- `GN_ISRES`: ISRES (улучшенная ранжированная стратегия стохастической эволюции).

**Внимание!** Обратите внимание, что некоторые алгоритмы в NLopt, в частности, большинство алгоритмов глобальной оптимизации, не поддерживают оптимизацию без ограничений и будут возвращать ошибку, если пользователь не указал конечные нижние и верхние границы. Границы для потенциальных параметров устанавливаются с использованием атрибутов `min` и `max` элементов блока `<fit-dof>`, как описано в разделе (Потенциалы). Кроме того, настоятельно рекомендуется определить границы для таких свойств, как параметры решетки.

В противном случае пригонка может быстро разойтись и привести к необоснованным результатам. Границы свойств устанавливаются с помощью ключевых слов `min` и `max` соответствующего элемента.

Элемент `<nlopt>` может быть дополнительно изменён с использованием нескольких необязательных атрибутов:

- `seed` (целое): начальное значение для генератора псевдослучайных чисел. Критерии остановки<sup>2</sup>:
- `stopval` (с точкой): когда целевая функция  $\leq$  `stopval`.
- `maxeval` (целое): когда количество операций расчетов функции функций превысит `maxeval`.
- `maxtime` (с точкой): когда время оптимизации (в секундах) превысит `maxtime`.
- `ftol_rel` (с точкой): когда шаг оптимизации изменяет значение целевой функции менее чем на `ftol_rel` умноженное на абсолютное значение функции.
- `ftol_abs` (с точкой): когда шаг оптимизации (или оценка оптимального значения) изменяет значение функции менее чем на `ftol_abs`.
- `xtol_rel` (с точкой): когда шаг оптимизации изменяет каждый параметр менее чем на `xtol_rel` умноженное на абсолютное значение параметра.

## 4 Потенциалы

`atomicrex` поддерживает большое число потенциалов, которые представлены в конце этого подраздела и для каждого из них сделан разбор. Тип потенциала определяется тегом `<potentials>` в основном входном файле. В нем может быть несколько подблоков, которые определяют несколько разных потенциалов, тем самым можно объединить (наложить) несколько типов разных потенциалов. Специальные блоки для этого тега имеют несколько общих атрибутов; атрибут `id` используется для установки уникального идентификатора для каждого потенциала; `species-a` и `species-b` показывают, с какими типами атомов имеет дело потенциал (указано в теге `<atom-types>`).

`<potentials>`

```
<eam id="Al" species-a="Al" species-b="Al">
```

---

<sup>2</sup>Эти критерии следует рассматривать только в том случае, если вы используете нелинейную оптимизацию через `Nlopt`.

```

...
</eam>
<eam id="SiAl" species-a="Si" species-b="Al">
...
</eam>
<abop id="Si" species-a="Si" species-b="Si">
...
</eam>
</potentials>

```

Каждый потенциал содержит набор параметров, представляющие степени свободы. Пользователь должен указать, какие параметры могут изменяться в процессе оптимизации (и, таким образом, являются «динамическими» степенями свободы). Это достигается включением блока `<fit-dof>` внутри соответствующего потенциального блока, который содержит список элементов, названных параметром, с которым они связаны. Эти элементы имеют несколько атрибутов:

- Атрибут `enabled` внутри тега позволяет переключать с динамического (False) на статический (True). Это позволяет включить или исключить определенный параметр из соответствия.
- Атрибут `reset` указывает, сбрасывается ли значение степени свободы потенциала к начальному значению, указанному пользователем каждый раз, когда выполняется новый шаг релаксации.
- Атрибуты `min` и `max` указывают диапазон, в пределах которого разрешено варьировать параметр.
- Атрибут `tag` используется некоторыми функциями, такими как потенциалы Терсоффа и "связи" (Analytic Bond-Order Potential), чтобы связать параметры с определенными смешанными взаимодействиями.

Блок `<fit-dof>` примерно выглядит так:

```

<fit-dof>
  <twomu enabled="True" min="0.1" max="2.0">
  <beta enabled="False">
</fit-dof>

```

Типы потенциалов:

- Потенциал Леннард-Джонса.

- Метод погруженного атома (ЕАМ).
- Потенциал Терсоффа.
- Потенциал "связи"(АВОР).
- Метод погруженного атома (МЕАМ).
- Табулированные потенциалы (ЕАМ и МЕАМ).
- Определение функций:
  - Скрининговые функции.
  - Интерполяционные функции.
  - Спецфункции.
  - Комбинация функций.
  - Функции, определенные пользователем.

**Потенциал Леннард-Джонса.** Потенциал Леннард-Джонса имеет следующую форму:

$$V(r) = 4\epsilon \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \quad (9)$$

и

$$E = \sum_{ij} V(r_{ij}) \quad (10)$$

Здесь,  $\epsilon$  и  $\sigma$  – параметры, определяющие энергию и радиус взаимодействия соответственно. Потенциал обрывается за пределами определенной границы. Пример кода:

```
<lennard-jones id="Lennard-Jones" species-a="*" species-b="*">
  <sigma> 2.5 </sigma>
  <epsilon> 0.5 </epsilon>
  <cutoff> 10.0 </cutoff>
  <fit-dof>
    <sigma/>
    <epsilon/>
  </fit-dof>
</lennard-jones>
```

Элементы и атрибуты:

- `<sigma>`, `<epsilon>` – начальные значения  $\sigma$  и  $\epsilon$ .

- `<cutoff>` – радиус обрезания.

**Метод погруженного атома (EAM).** Существует множество различных типов потенциалов в методе погруженного атома (EAM): приклеивающий потенциал, теория эффективной среды, форма Финниса-Синклера (FS) и др. Полная энергия этих парных функций может быть записана в следующей форме:

$$E = \sum_{ij} V(r_{ij}) + \sum_i F(\rho_i), \quad (11)$$

где

$$\rho_i = \sum_j \rho(r_{ij}) \quad (12)$$

Потенциалы могут быть совершенно разными в зависимости от функциональной зависимости, выбранных для парной части  $V$ , функции погружения  $F$  и электронной плотности  $\rho$ . В настоящей реализации возможно составление различных форм потенциалов с использованием математического анализатора. Это позволяет не только копировать любую из исходных форм, но и определять практически любые другие. Это демонстрируется построением потенциала EAM с пользовательскими функциями, который использует форму потенциала из [32], а также в примере, в котором устанавливается простой потенциал EAM для алюминия<sup>3</sup>.

**Замечание:** опуская член функции погружения, можно использовать EAM потенциал для построения произвольных парных потенциалов.

Следующий блок кода иллюстрирует определение простой функциональной формы. Двумя основными подэлементами блока `<eam>` являются `<mapping>` и `<functions>`. Блок `<functions>` содержит определения различных функций и параметров. Каждой определенной функции должен быть присвоен идентификатор с помощью атрибута `id`. Идентификатор используется в блоке `<mapping>` для привязки функций к определенным типам взаимодействия (парный потенциал  $V$  и электронная плотность  $\rho$ ) или к типу атома (функция погружения  $F$ ).

```
<eam id="Al" species-a="*" species-b="*">
  <export-eam-file resolution='500' rho-range-factor='2.0'>
    output.eam.fs</export-eam-file>

  <mapping>
    <pair-interaction species-a="*" species-b="*" function="V" />
    <electron-density species-a="*" species-b="*" function="rho" />
```

<sup>3</sup>См. пример для алюминия в директории `examples/fittingEAMpotential`.

```

    <embedding-energy species-"*" function="F" />
</mapping>

```

```

<functions>

```

```

  <user-function id="V">

```

```

    <input-var>r</input-var>

```

```

    <expression> A*exp(-lambda*r) </expression>

```

```

    <derivative> -lambda*A*exp(-lambda*r) </derivative>

```

```

    <param name="A">500</param>

```

```

    <param name="lambda">2.73</param>

```

```

    <fit-dof> <A/> <lambda/> </fit-dof>

```

```

    <screening>

```

```

      <user-function id="V_screening">

```

```

        <cutoff>6.5</cutoff>

```

```

        <input-var>r</input-var>

```

```

        <expression>

```

```

          1 - 1/(1 + ((r - cutoff) / h)^4)

```

```

        </expression>

```

```

        <derivative>

```

```

          4 * h^4 * (r - cutoff)^3 / ((h^4 + (r - cutoff)^4)^2)

```

```

        </derivative>

```

```

        <param name="h">3</param>

```

```

      </user-function>

```

```

    </screening>

```

```

  </user-function>

```

```

<user-function id="rho">

```

```

  <input-var>r</input-var>

```

```

  <expression> exp(-twomu*r) </expression>

```

```

  <derivative> -twomu*exp(-twomu*r) </derivative>

```

```

  <param name="twomu">1.14</param>

```

```

  <fit-dof> <twomu/> </fit-dof>

```

```

  <screening>

```

```

    <user-function id="rho_screening">

```

```

      <cutoff>6.5</cutoff>

```

```

      <input-var>r</input-var>

```

```

      <expression>

```

```

        1 - 1/(1 + ((r - cutoff) / h)^4)

```

```

      </expression>

```

```

    <derivative>
      4 * h^4 * (r - cutoff)^3 / ((h^4 + (r - cutoff)^4)^2)
    </derivative>
    <param name="h">3</param>
  </user-function>
</screening>
</user-function>

<user-function id="F">
  <input-var>rho</input-var>
  <expression> -D*sqrt(rho) </expression>
  <derivative> -D/(2*sqrt(rho)) </derivative>
  <param name="D">8</param>
  <fit-dof> <D/> </fit-dof>
</user-function>
</functions>
</eam>

```

### Атрибуты и элементы.

- `<mapping>`: Этот блок определяет отображение функций, определенных в блоке `<functions>`, на разные типы и пары атомов.
  - `<pair-interaction>`: Включите парное взаимодействие, используя атрибуты `species-a` и `species-b`, чтобы указать типы атомов, между которыми оно будет осуществляться, и атрибут `functions`, чтобы указать идентификатор функции из объявленных существующих в файле - блок `<functions>`. Звездочка может указываться, когда строится потенциал взаимодействия для одинаковых типов атомов, либо если строится один потенциал (можно прописать и по идентификатору атома).
  - `<electron-density>`: Назначьте функцию электронной плотности, используя атрибуты `species-a` и `species-b`, чтобы указать используемые типы атомов, и атрибут функции, чтобы указать идентификатор функции. Идентификатор функции должен точно соответствовать одной из функций, определенных в блоке `<functions>`.
  - `<embedding-energy>`: Назначьте функцию погружения, используя атрибут `species`, чтобы указать используемый тип атома, и атрибут `function`, чтобы указать идентификатор функции. Идентификатор функции должен точно соответствовать одной из функций, определенных в блоке `<functions>`.

- `<functions>`: Этот блок содержит определения различных функций и параметров. Определение функции описано здесь (вставить ссылку). Каждой определенной функции должен быть присвоен идентификатор с помощью атрибута `id`. Идентификатор используется в блоке `<mapping>` для привязки функции к определенному типу взаимодействия (парный потенциал и электронная плотность) или типу атома (функция погружения).
- `<export-eam-file>`: Здесь указывается имя файла, в который записываются табличные функции потенциала EAM. Полученный файл имеет формат "eam/fs" для вычислительной среды LAMMPS и будет содержать данные табличных функций для всех элементов, для которых определена как минимум функция погружения, и их попарные комбинации. Разрешение (количество точек) сгенерированных табличных функций можно регулировать с помощью необязательного атрибута `resolution` (значение по умолчанию: 500). Диапазон функции погружения, которая должна быть занесена в таблицу и экспортирована, автоматически определяется путем нахождения максимальной электронной плотности, встречающейся в любой из структур текущей операции по построению. Эта максимальная плотность электронов умножается на дополнительный коэффициент для определения диапазона экспорта. Коэффициент, который по умолчанию равен 2.0, можно перезаписать, указав необязательный атрибут `rho-range-factor`. Обратите внимание, что массы и атомные номера, которые записываются в файл EAM, должны быть указаны в элементах `<species>` в списке `<atom-types>` задания (по умолчанию: файл не записывается).

**Потенциал Терсоффа.** Потенциал Терсоффа в AtomicSx может быть экспортирован в вычислительную среду LAMMPS. Эта версия поддерживает многокомпонентные системы со всеми параметрами, зависящими от типов включенных атомов, то есть двухчастичных членов, зависящих от типов атомов  $i$  и  $j$ , тогда как трехчастичные члены зависят от типов атомов  $i$ ,  $j$  и  $k$ . Обратите внимание, что в АВОР потенциале трехчастичные члены обычно зависят только от типов атомов  $i$  и  $k$ . Если вы хотите получить такое поведение, используйте форму АВОР. Потенциал Терсоффа записывается в следующей форме:

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} V_{ij}, \quad (13)$$

с

$$V_{ij} = f_c(r_{ij}) [f_R(r_{ij}) + b_{ij} f_A(r_{ij})]. \quad (14)$$

Здесь  $V_R(r) = A * \exp(-\lambda_1 r)$  и  $V_A(r) = -B * \exp(-\lambda_2 r)$  отталкивающая и притягивающая части потенциальных ветвей соответственно,  $f_c(r_{ij})$  – функция отсечки



и синусоидально затухающая в интервале  $(R - D; R + D)$ , за пределами которого исчезает. Трехчастичный вклад возникает из-за параметра "связи":

$$b_{ij} = (1 + \beta^n \zeta_{ij}^n)^{-\frac{1}{2n}}, \quad (15)$$

где

$$\zeta_{ij} = \sum_{k \neq i, j} f_c(r_{ij}) g(\theta_{ijk}) \exp[\lambda_3^m (r_{ij} - r_{ik})^m] \quad (16)$$

угловая зависимость обусловлена фактором:

$$g(\theta) = \gamma_{ijk} \left( 1 + \frac{c^2}{d^2} - \frac{c^2}{d^2 + (h - \cos\theta)^2} \right) \quad (17)$$

Следующий фрагмент кода, вставленный в блок `<potentials>`, иллюстрирует определение этого типа потенциала во входном файле:

```
<tersoff id="Co" species-a="*" species-b="*">
  <param-file>Co.tersoff</param-file>
  <fit-dof>
    <A tag="CoCoCo" enabled="true" />
    <B tag="CoCoCo" enabled="true" />
    <lambda1 tag="CoCoCo" enabled="true" />
    <lambda2 tag="CoCoCo" enabled="true" />
    <gamma tag="CoCoCo" enabled="true"/>
    <lambda3 tag="CoCoCo" enabled="true" />
    <c tag="CoCoCo" enabled="false" />
    <d tag="CoCoCo" enabled="false" />
    <theta0 tag="CoCoCo" enabled="false" />
  </fit-dof>
</tersoff>
```

**Внимание!** Заметьте, что атрибуты `species-a` и `species-b` всегда должны быть установлены на `*`. Назначение типов атомов выполняется с использованием информации в атрибуте тега `tag`.

### Элементы и атрибуты.

- `<param-file>`: имя файла с входными параметрами в формате LAMMPS/Tersoff.
- `<export-potential>` (опционально): имя файла, в который записываются параметры потенциала в формате LAMMPS/Tersoff (по умолчанию: файл не записывается).

**Потенциал "связи" (АВОР).** Потенциал АВОР – Analytic Bond Order Potential представляет собой переформулировку потенциала Терсоффа, реализованного в Atomicrex и LAMMPS. Разница заключается в двухчастичных параметрах  $A$ ,  $B$ ,  $\lambda_1$  и  $\lambda_2$ , которые были переписаны с точки зрения более строгих рассуждений,  $r_0$ ,  $D_0$ ,  $\beta$  и  $S$ , которые связаны со свойствами димеров и отношением Полинга. Реализация поддерживает многокомпонентные системы со всеми параметрами, зависящими от типов участвующих атомов, то есть двухчастичные члены, зависящие от типов атомов  $i$  и  $j$ , тогда как трехчастичные члены зависят от типов атомов  $i$ ,  $j$ ,  $k$ . Обратите внимание, что трехчастичные члены обычно зависят только от типов атомов  $i$  и  $k$  [13, 14, 16, 18]. Такое поведение можно получить, установив для атрибута классического режима значение True (по умолчанию: False).

В этом формате потенциал записывается так:

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} V_{ij}, \quad (18)$$

с

$$V_{ij} = f_c(r_{ij}) [f_R(r_{ij}) + b_{ij} f_A(r_{ij})]. \quad (19)$$

Здесь  $V_R(r) = \frac{D_0}{S-1} \exp[-\beta \sqrt{2S}(r-r_0)]$  и  $V_A(r) = -\frac{SD_0}{S-1} \exp(-\beta \sqrt{2/S}(r-r_0))$  отталкивающая и притягивающая части потенциальных ветвей соответственно,  $f_c(r_{ij})$  – функция отсечки и синусоидально затухающая в интервале  $(R-D; R+D)$ , за пределами которого исчезает. Трехчастичный вклад возникает из-за параметра порядка связи

$$b_{ij} = (1 + \beta^n \zeta_{ij}^n)^{-\frac{1}{2n}}, \quad (20)$$

где

$$\zeta_{ij} = \sum_{k \neq i, j} f_c(r_{ij}) g(\theta_{ijk}) \exp[\lambda_3^m (r_{ij} - r_{ik})^m] \quad (21)$$

угловая зависимость обусловлена фактором:

$$g(\theta) = \gamma_{ijk} \left( 1 + \frac{c^2}{d^2} - \frac{c^2}{d^2 + (h + \cos\theta)^2} \right) \quad (22)$$

Файлы параметров записываются в формате LAMMPS/Tersoff. В выходных файлах параметры  $r_0$ ,  $D_0$ ,  $\beta$  и  $S$  включены в блоки комментариев.

Обратите внимание, что параметр  $h$  в форме АВОР имеет знак, противоположный знаку в формулировке Терсоффа. Следующий фрагмент кода, который будет вставлен в блок потенциалов, иллюстрирует определение этого типа потенциала во входном файле.

```

<abop id="Si" species-a="*" species-b="*">
  <param-file>ABOP_potential.tersoff</param-file>
  <export-potential>ABOP_potential_fitted.tersoff</export-potential>
  <fit-dof>
    <r0 tag="SiSiSi" enabled="true" />
    <D0 tag="SiSiSi" enabled="true" />
    <beta tag="SiSiSi" enabled="true" />
    <S tag="SiSiSi" enabled="true" />
    <gamma tag="SiSiSi" enabled="true" />
    <c tag="SiSiSi" enabled="true" />
    <d tag="SiSiSi" enabled="true" />
    <h tag="SiSiSi" enabled="true" />
    <twomu tag="SiSiSi" enabled="false" />
    <beta2 tag="SiSiSi" enabled="false" />
    <powern tag="SiSiSi" enabled="false" />
  </fit-dof>
</abop>

```

**Внимание!** Обратите внимание, что атрибуты `species-a` и `species-b` всегда должны быть установлены на `*`. Назначение типов атомов выполняется с использованием информации в атрибуте `tag`.

### Элементы и атрибуты.

- `<param-file>`: имя файла с входными параметрами в формате LAMMPS/Tersoff.
- `<export-potential>` (опционально): имя файла, в который записываются параметры потенциала в формате LAMMPS/Tersoff [по умолчанию: файл не записывается].
- `<classic-mode>` (опционально): принудительная «классическая» связь трехчастичных параметров, которые затем зависят только от типов атомов  $i$  и  $k$ , а не  $i$ ,  $j$  и  $k$  (по умолчанию: False).

**Модифицированный метод погруженного атома (МЕАМ).** Схема модифицированного метода погруженного атома (МЕАМ) была разработана как обобщение метода погруженного атома. Потенциал модифицированного метода погруженного атома (МЕАМ) был разработан как обобщение метода погруженного атома [33]. В отличие от последнего, потенциалы МЕАМ включают асферические взаимодействия (зависящие от углов), которые входят через член электронной плотности. В результате эти потенциалы могут описывать направленные связи,

которая наиболее очевидна в ковалентных материалах, таких как кремний и алмаз, а также, например, в объемно-центрированных кубических решетках металлов [34].

Хотя изначально были приняты простые аналитические формы и угловые взаимодействия были ограничены первыми ближайшими соседями, МЕАМ может быть легко расширен для включения большего числа соседей и поддержки более общих функциональных зависимостей (например, сплайны[36]). Согласно классификации Карлссона[1] потенциалы МЕАМ попадают в категорию кластерных функционалов, которая также включает, например, потенциал "связи".

Общая формула для полной энергии:

$$E = \sum_{ij} V(r_{ij}) + \sum_i F(\rho_i), \quad (23)$$

где угловая зависимость входит через член электронной плотности<sup>4</sup>:

$$\rho_i = \sum_j \rho(r_{ij}) + \sum_{jk} f(r_{ij})f(r_{ik})g(\cos(\theta_{ijk})). \quad (24)$$

Как и в случае потенциалов ЕАМ, были предложены различные функциональные зависимости для  $V$ ,  $F$ ,  $\rho$ ,  $f$  и  $g$ . В настоящей реализации возможно составление функциональных зависимостей с использованием математического анализатора. Это позволяет не только копировать любую из исходных форм, но и определять практически любые произвольные другие. Это демонстрируется построением потенциала МЕАМ с пользовательскими функциями.<sup>5</sup>

Следующий блок кода иллюстрирует определение простой функциональной зависимости – в нем почти всё то же самое, что и в ЕАМ методе. Двумя основными подэлементами блока `<meam>` являются `mapping` и `<functions>`. Блок `<functions>` содержит определения различных функций и параметров, как подробно описано в разделе, посвященном спецификации функциональных зависимостей (вставить ссылку). Каждой определенной функции должен быть присвоен идентификатор с помощью атрибута `id`. Идентификатор используется в блоке `<mapping>` для привязки функций к определенным типам взаимодействия (парная часть  $V$ , электронная плотность  $\rho$  наряду с опорными функциями  $f$  и  $g$ ) или типу атома (функция погружения  $F$ ).

<sup>4</sup>Второе слагаемое в суммировании представляет собой ключевое отличие по сравнению с форматом ЕАМ.

<sup>5</sup>См. пример в директории `examples/potential_MEAM`

```

<meam id="Al" species-a="*" species-b="*">
  <export-functions>Si_meam_out</export-functions>

  <mapping>
    <pair-interaction species-a='*' species-b='*' function='V' />
    <electron-density species-a='*' species-b='*' function='rho' />
    <f-function species-a='*' species-b='*' function='f' />
    <g-function species-a='*' species-b='*' species-c='*' function='g' />
    <embedding-energy species='*' function='F' />
  </mapping>

  <functions>
    <spline id='V'>
      <derivative-left>-42.66967</derivative-left>
      <cutoff>4.5</cutoff>
      <nodes>
        <node x='1.500000000' y=' 6.92994' enabled='true' />
        <node x='1.833333333' y='-0.43995' enabled='true' />
        <node x='2.166666667' y='-1.70123' enabled='true' />
        <node x='2.500000000' y='-1.62473' enabled='true' />
        <node x='2.833333333' y='-0.99696' enabled='true' />
        <node x='3.166666667' y='-0.27391' enabled='true' />
        <node x='3.500000000' y='-0.02499' enabled='true' />
        <node x='3.833333333' y='-0.01784' enabled='true' />
        <node x='4.166666667' y='-0.00961' enabled='true' />
        <node x='4.500000000' y=' 0.0      ' enabled='true' />
      </nodes>
    </spline>
    ...
  </functions>
</meam>

```

### Элементы и атрибуты.

- **<mapping>**: Этот блок определяет отображение функций, определенных в блоке **<functions>**, на разные типы и пары атомов.
  - **<pair-interaction>**: включите парное взаимодействие, используя атрибуты **species-a** и **species-b**, чтобы указать типы атомов, между которыми оно будет осуществляться, и атрибут **functions**, чтобы указать идентификатор функции из объявленных существующих в файле - блок **<functions>**.

- `<electron-density>`: назначьте функцию электронной плотности, используя атрибуты `species-a` и `species-b`, чтобы указать используемые типы атомов, и атрибут `function`, чтобы указать идентификатор функции. Идентификатор функции должен точно соответствовать одной из функций, определенных в блоке `<functions>`.
- `<embedding-energy>`: назначьте функцию погружения, используя атрибут `species`, чтобы указать используемый тип атома, и атрибут `function`, чтобы указать идентификатор функции. Идентификатор функции должен точно соответствовать одной из функций, определенных в блоке `<functions>`.
- `<f-function>`: назначьте вспомогательную попарную функцию, которая входит в угловую зависимую часть расчета электронной плотности, используя атрибуты `<species-a>` и `<species-b>`, чтобы указать используемые типы атомов, и атрибут `functions`, чтобы указать идентификатор функции. Идентификатор функции должен точно соответствовать одной из функций, определенных в блоке `<functions>`.
- `<g-function>`: назначьте вспомогательную угловую функцию, которая входит в угловую зависимую часть расчета электронной плотности, используя атрибуты `species-a`, `species-b` и `species-c`, чтобы указать используемые типы атомов, и атрибут `function`, чтобы указать идентификатор функции. Идентификатор функции должен точно соответствовать одной из функций, определенных в блоке `<functions>`.
- `<embedding-energy>`: назначьте функцию погружения, используя атрибут `species`, чтобы указать используемый тип атома, и атрибут `function`, чтобы указать идентификатор функции. Идентификатор функции должен точно соответствовать одной из функций, определенных в блоке `<functions>`.
- `<functions>`: этот блок содержит определения различных функций и параметров. Каждой определенной функции должен быть присвоен идентификатор с помощью атрибута `id`. Идентификатор используется в блоке `<mapping>` для привязки функции к определенному типу взаимодействия (парный потенциал и электронная плотность) или типу атома (функция погружения).
- `<export-functions>`: имя файла, в который записываются параметры потенциала в формате LAMMPS (по умолчанию: файл не записывается).

**Табулированные потенциалы.** Смотрите раздел EAM и MEAM потенциалы.

**Определение функций.** Функции используются для определения зависимости потенциалов от различных параметров, например, потенциал в методе погруженного атома (EAM). Atomicrex предоставляет несколько предопределенных функ-

ций, а также механизм для определения (почти) произвольных зависимостей. Для получения информации о том, как и где следует задавать функции в зависимости от выбранного типа потенциала, обратитесь к соответствующему параграфу. Все функциональные элементы имеют атрибут `id`, который используется для идентификации соответствующей функции, например, для назначения функций различным типам взаимодействия для потенциалов ЕАМ через блок `<mapping>`.

**Функции экранирования.** Несколько предустановленных функций предназначены для использования в качестве функций «экранирования» / отсечки. Они удовлетворяют условиям:  $f(0) = 1$  и  $f(r_c) = 0$  с гладкими производными на границах, в основном<sup>6</sup>. Это функции следующего типа – коэффициент нормировки опущен для ясности:

- `<exp-A>`: экспоненциальная функция экранирования,

$$f(r; r_c) = \exp[1/(r - r_c)], \quad (25)$$

её объявление происходит следующим образом:

```
<exp-A>
  <cutoff> 5.5 </cutoff>
</exp-A>
```

Здесь `<cutoff>` =  $r_c$ .

- `<exp-B>`: ещё один тип экспоненциальной функции экранирования,

$$f(r; r_c; n; \alpha; r_c^{(i)}) = \exp[-\text{sign}(n) * \alpha / (1 - x^n)], \quad (26)$$

где  $x = (r - r_c^{(i)}) / (r_c - r_c^{(i)})$ . Она объявляется следующим образом:

```
<exp-B>
  <cutoff>      5.5 </cutoff>
  <rc>         2.8 </rc>
  <alpha>      1.0 </alpha>
  <exponent> -3.0 </exponent>
</exp-B>
```

Здесь `<cutoff>` =  $r_c$ , `<rc>` =  $r_c^{(i)}$  (внутренний радиус отсечки, пример на рисунке ниже), `<alpha>` =  $\alpha$ , `<exponent>` =  $n$ .

---

<sup>6</sup>Некоторые потенциалы, такие как потенциалы Терсоффа и АВОР, имеют встроенные функции отсечки, которые обрабатываются непосредственно программой.

- `<exp-gaussian>`: этот элемент определяет функцию распределения Гаусса, умноженную на экспоненциальную функцию экранирования:

$$f(r; r_c; n; \alpha; \sigma) = \exp \left[ \frac{-\text{sign}(n) * \alpha}{(1 - (r/r_c)^n)} \right] * \exp \left[ \frac{-r^2}{2\sigma^2} \right] / (\sigma\sqrt{2\pi}) \quad (27)$$

Пример объявления функции:

```
<exp-gaussian>
  <cutoff>      5.5 </cutoff>
  <stddev>     2.8 </stddev>
  <alpha>      1.0 </alpha>
  <exponent>  -3.0 </exponent>
</exp-gaussian>
```

Здесь `cutoff` =  $r_c$ , `stddev` =  $\sigma$ , `alpha` =  $\alpha$ , `exponent` =  $n$ .

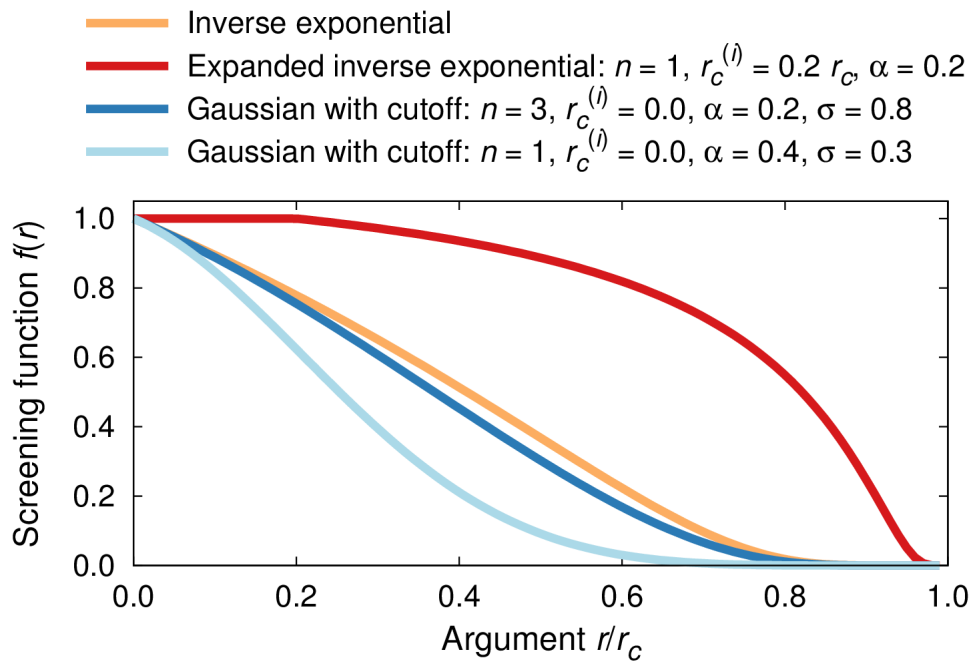


Рис. 1: Различные варианты экспоненциальных функций отсечки.

**Интерполяционные функции.** Этот набор предназначен для интерполяции функций:

- `<poly>`: полиномиальная функция:

$$g(x) = \sum_{n=0}^P a_n x^n, \quad (28)$$



с граничными условиями  $f(0) = 1$  и  $f(r_c) = 0$ , где  $r_c$  – расстояние отсечки. Объявление функции выглядит следующим образом:

```
<poly>
  <cutoff>5.5</cutoff>
  <coefficients>
    <coeff n="1" value="2.3" enabled="True" reset="False" min="1.0"
      max="7.0">
    <coeff n="2" value="1.1" enabled="True" reset="False" min="0.0"
      max="4.0">
    <coeff n="3" value="0.6" enabled="True" reset="False" min="0.0"
      max="4.0">
  </coefficients>
</poly>
```

Здесь `<cutoff>` определяет расстояние отсечки. Блок `<coefficients>` определяет коэффициенты. Значение  $a_n$  устанавливается через атрибут `value`, `n` определяет  $n$ . Атрибуты `enabled`, `reset`, `min` и `max` реализуют обычные задачи степеней свободы здесь (см. блок `<fit-dof>` в разделе "Потенциалы"). Коэффициент нулевого порядка  $a_0$  не может быть установлен пользователем изначально, так как он фиксируется граничными условиями.

В коде задан кубический полином  $y = x_0 + 2.3x + 1.1x^2 + 0.6x^3$  с вариациями каждого из коэффициентов в своем интервале, обозначенном соответствующими тегами (`min`; `max`).

- `<spline>`: кубический сплайн:

```
<spline>
  <cutoff>5.5</cutoff>
  <nodes>
    <node x="0" y="10" enabled="true" />
    <node x="0.2" y="-0.3" enabled="true" />
    <node x="0.5" y=" 0.1" enabled="true" />
    <node x="0.7" y="-0.1" enabled="true" />
    <node x="0.85" y=" 0.4" enabled="true" />
    <node x="5.5" y=" 0" enabled="true" />
  </nodes>
</spline>
```

Здесь `<cutoff>` определяет расстояние отсечки. Блок `<nodes>` определяет узловые точки сплайна. Каждому узлу соответствует пара точек с координатами  $(x, y)$ , предоставленной атрибутами `x` и `y`. Атрибуты `enabled`, `reset`, `min` и `max` отвечают за обычные задачи степеней свободы, описанных здесь (см. блок `<fit-dof>` в разделе "Потенциалы"). Заметьте, что только значение  $y$  изменяется во время оптимизации, а  $x$  остается постоянным.

Можно указать первые производные в конечных точках, используя необязательные атрибуты параметров, `derivative-left` и `derivative-right`, следующим образом:

```
<spline>
  <derivative-left> 123.0 </derivative-left>
  <derivative-right> -456.0 </derivative-right>
  ...
</spline>
```

Если эти атрибуты не указаны, они по умолчанию равны нулю. Пример использования сплайнов можно найти в примере построения MEAM потенциала (директория `examples/potential_MEAM`).

- `<constant>`: обычная константа.

```
<constant> 2.1 </constant>
```

**Спецфункции.** Atomicrex включает в себя несколько функций, которые непосредственно подходят, например, в качестве функциональных зависимостей для парных потенциалов.

- `<morse-A>`: Оригинальный потенциал Морзе (тип A):

$$V(r) = D_0 [\exp(-2\alpha(r - r_0)) - 2 * \exp(-\alpha(r - r_0))], \quad (29)$$

где параметры  $D_0$  и  $r_0$  определяют когезионную энергию и оптимальную длину связи соответственно, в то время, как параметр  $\alpha$  влияет на кривизну функции, соответственно, на жесткость. Обратите внимание, что потенциал Морзе не имеет встроенной отсечки, поэтому он должен сочетаться с функцией экранирования. Следующий блок иллюстрирует объявление стандартного потенциала Морзе во входном файле. Также приводится пример того, как задать параметры потенциала (`<fit-dof>` блок) и вставку функции экранирования через блок `<screening>`:

```

<morse-A>
  <D0>    0.3 </D0>
  <r0>    3.0 </r0>
  <beta>  1.8 </beta>
  <fit-dof>
    <D0    enabled="true" />
    <r0    enabled="true" />
    <beta  enabled="false" />
  </fit-dof>

  <screening>
    <exp-B>
      <cutoff>    5.5 </cutoff>
      <rc>        0.0 </rc>
      <alpha>     1.0 </alpha>
      <exponent> -3.0 </exponent>
      <fit-dof>
        <alpha    enabled="false" />
        <exponent enabled="false" />
      </fit-dof>
    </exp-B>
  </screening>
</morse-A>

```

- **<morse-B>**: обобщенный вариант потенциала Морзе (вариант В) использованный в примере АВОР потенциала, также в некоторых примерах (ЕАМ) потенциалов. Определяется как:

$$V(r) = \underbrace{\frac{D_0}{S-1} \exp\left(-\beta\sqrt{2S}(r-r_0)\right)}_{V_R} - \underbrace{\frac{D_0 S}{S-1} \exp\left(\beta\sqrt{2/S}(r-r_0)\right)}_{V_A} + \delta. \quad (30)$$

Как и в исходном потенциале Морзе, параметры  $D_0$  и  $r_0$  определяют когезионную энергию и оптимальную длину связи соответственно. Параметр  $\beta$  влияет на кривизну функции, следовательно, на жесткость. Параметр  $S$  изменяет ангармонизм и может быть установлен через соотношение Полинга. Весь потенциал может быть сдвинут с помощью параметра  $\delta$ . Случай  $S = 2$  и  $\delta = 0$ , соответствует оригинальному потенциалу Морзе.

```
<morse-B>
```

```

<D0> 0.3 </D0>
<r0> 3.0 </r0>
<beta> 1.8 </beta>
<S> 2.1 </S>
</morse-B>

```

- `<morse-C>`: обобщенный потенциал (вариант C), использованный в потенциале Терсоффа также доступен. Определяется как:

$$\underbrace{A * \exp(-\lambda r)}_{V_R} - \underbrace{B * \exp(-\mu r)}_{V_A} + \delta \quad (31)$$

Заметьте, что эта форма потенциала обычно наиболее предпочтительна, поскольку параметры имеют непосредственную физическую интерпретацию. Таким образом, параметры  $A$  и  $B$  в форме `<morse-C>` могут изменяться в очень большом диапазоне, иногда вызывая нестабильность, которая может сильно мешать процессу построения. Формат блока `<morse-C>` эквивалентен определению функций `<morse-A>` и `<morse-B>`.

- `<gaussian>`: Функция Гаусса определяется как:

$$V(r) = a * \exp(-\eta(r - \mu)^2) \quad (32)$$

с тремя параметрами  $a$  (префактор),  $\eta$ ,  $\mu$ . Обратите внимание, что функция Гаусса не имеет «встроенной» отсечки, но - как и для любой функции - вы все равно можете указать жесткую отсечку для необязательного атрибута `cutoff` или применить функцию экранирования с использованием элемента `<screening>`. Блок кода выглядит так:

```

<gaussian>
  <prefactor> 1.0 </prefactor>
  <eta> 3.0 </eta>
  <mu> 1.8 </mu>
  <fit-dof>
    <prefactor enabled="true" />
    <eta enabled="false" />
    <mu enabled="true" />
  </fit-dof>
</gaussian>

```

**Комбинация функций.** Кроме того, Atomicrex предоставляет две вспомогательные функции, `<sum>` и `<product>`, которые позволяют объединять два или более блока функций без необходимости явной интеграции их в один. Например, можно добавить две функции экранирования, используя следующий пример:

```
<sum>
  <exp-A>
    <cutoff> 5.5 </cutoff>
  </exp-A>

  <exp-B>
    <cutoff> 5.5 </cutoff>
    <rc> 2.8 </rc>
    <alpha> 1.0 </alpha>
    <exponent> -3.0 </exponent>
  </exp-B>
</sum>
```

Операция `<sum>` используется в примере, демонстрирующем построение ЕАМ потенциала с пользовательскими функциями с помощью объединения двух парных потенциалов.

**Пользовательские функции.** Atomicrex ссылается на библиотеку математического анализатора `mparser`, которая обеспечивает огромную гибкость для определения различных функций. Следующий блок входного файла иллюстрирует определение фактически двух таких пользовательских функций (одна для парного потенциала `V` и одна для функции экранирования `rho_screening`):

```
<user-function id="V">
  <input-var> r </input-var>
  <expression> A*exp(-lambda*r) </expression>
  <derivative> -lambda*A*exp(-lambda*r) </derivative>
  <param name="A"> 500 </param>
  <param name="lambda"> 2.73 </param>
  <fit-dof>
    <A/>
    <lambda/>
  </fit-dof>

  <screening>
```

```

<user-function id="rho_screening">
  <cutoff>6.5</cutoff>
  <input-var>r</input-var>
  <expression> 1 - 1/(1 + ((r - cutoff) / h)^4) </expression>
  <derivative>
    4 * h^4 * (r-cutoff)^3 / ((h^4 + (r-cutoff)^4)^2)
  </derivative>
  <param name="h">3</param>
</user-function>
</screening>
</user-function>

```

Здесь были использованы следующие элементы:

- `<input-var>`: идентификатор входной переменной, используемый в определении функции.
- `<expression>`: объявленная функция, где идентификатор, указанный в `<input-var>`, используется в качестве аргумента функции.
- `<derivative>`: производная, определенная в блоке `<expression>`.
- `<param>`: введите параметр, который появляется в определении функции. Имя параметра указывается с помощью атрибута `name`. Обычно существует несколько параметров.
- `<fit-dof>`: этот элемент используется для объявления параметров, которые должны быть скорректированы в процессе оптимизации. Параметры предоставляются в виде списка подэлементов блока `<fit-dof>`, где именем элемента является просто `name`, предоставленное через атрибут `name` в элементе `<param>`.
- `<screening>`: этот блок позволяет вставить функцию экранирования (или отсечки), которая обеспечивает контейнер (нужна помощь в переводе).

Полный входной файл можно найти в примере, демонстрирующем построение потенциала ЕАМ с помощью пользовательских функций.

## 5 Структуры

Структуры являются одной из двух основных частей в `atomicrex`, другая является потенциалом. Они вставляются в блок `<structures>` входного файла.

Доступна довольно большая база данных изначально определенных структур; кроме того, произвольные структуры могут быть включены либо с использованием конструкции `<user-structure>`, либо с помощью входных файлов внешней структуры. Структуры связаны с набором свойств, которые вычисляются на этапе подгонки или вывода в зависимости от настроек в файле задания. Элемент `<group>` позволяет создавать группы структур, что очень удобно при обработке большого количества входных структур.

Каждой структуре может быть назначен индивидуальный вес, который может быть установлен с помощью атрибута `relative-weight`. Вес по умолчанию равен единице.

Можно экспортировать окончательную структуру после процесса подгонки в файл данных. Следующий фрагмент кода иллюстрирует команду для этой функции во входном файле:

```
<output file="structure.dump" format="lammps-dump"/>
```

Это создает файл выходных данных LAMMPS с именем «structure.dump». Возможные форматы файлов: «lammps-dump» и «POSCAR» (из VASP). Эта функция также используется в примере по расчету свойства, указанного пользователем, с использованием структуры с точечными дефектами для визуализации созданной там вакансии.

**Предопределенные структуры.** Для удобства и эффективности вычислений многие распространенные кристаллические структуры были реализованы непосредственно в `atomicrex`. Эти структуры представлены в списке ниже. Как правило, они могут быть идентифицированы с использованием либо обычной аббревиатуры (например, FCC для гранцентрированной кубической решетки), либо заглавной английской буквы и цифры (например, A1<sup>7</sup> для FCC).

**Димеры.** `<dimer>`: Димер имеет одну степень свободы, а именно межатомное расстояние `<atom-distance>`. Следующий фрагмент кода иллюстрирует определение димера во входном файле:

```
<dimer-structure id="SiC_dimer">  
  <atom-type-A> Si </atom-type-A>  
  <atom-type-B> C </atom-type-B>
```

---

<sup>7</sup> Данная система обозначений является дополнением к уже существующим и описывает пространственные группы решеток. Система введена впервые в Германии в журнале *Zeitschrift für Kristallographie – Crystalline Materials* в период с 1913 по 1928 годы. При задании типа решетки будут показаны оба варианта кода, так что это не сильно влияет на восприятие текста.

```
<atom-distance> 1.75 </atom_distance>
</dimer-structure>
```

### Унарные кристаллические структуры.

- `<fcc-lattice>`, `<A1-lattice>`. Это обозначение для гранецентрированной кубической решетки – ГЦК (FCC). Имеется только одна степень свободы – постоянная решетки.

```
<fcc-lattice id="Cu_fcc">
  <atom-type> Cu </atom-type>
  <lattice-parameter> 3.615 </lattice-parameter>
</fcc-lattice>
```

- `<bcc-lattice>`, `<A2-lattice>`. Это обозначение для объемноцентрированной кубической решетки – ОЦК (FCC). Имеется только одна степень свободы – постоянная решетки.

```
<bcc-lattice id="Mo_bcc">
  <atom-type> Mo </atom-type>
  <lattice-parameter> 3.147 </lattice-parameter>
</bcc-lattice>
```

- `<hcp-lattice>`, `<A3-lattice>`. Это обозначение для гексагональной плотноупакованной решетки – ГПУ (HCP, 2H). Имеется две степени свободы – постоянная решетки и отношение  $c/a$ .

```
<hcp-lattice id="Ti_hcp">
  <atom-type> Ti </atom-type>
  <lattice-parameter> 2.95 </lattice-parameter>
  <ca-ratio> 1.588 </ca-ratio>
</hcp-lattice>
```

- `<dhcp-lattice>`. Это обозначение для двойной гексагональной плотноупакованной решетки – ДГПУ (DHCP, 4H). Имеется две степени свободы – постоянная решетки и отношение  $c/a$ . Структура ДГПУ тесно связана со структурой ГПУ. В то время как ГПУ соответствует последовательности стеков АВ | АВ |..., ДГПУ эквивалентен последовательности стеков АВАС | АВАС |...



```
<dhcp-lattice id="Ti_hcp">
  <atom-type> Ti </atom-type>
  <lattice-parameter> 2.95 </lattice-parameter>
  <ca-ratio> 3.176 </ca-ratio>
</dhcp-lattice>
```

- `<betaSn-lattice>`, `<A5-lattice>`. Это обозначение для решетки  $\beta$ -олова. Структура имеет две степени свободы (в плоскости и вне ее) – постоянная решетки и отношение  $c/a$ . Структура бета-олова в основном относится к элементам группы IV, таким как кремний, германий.

```
<betaSn-lattice id="Si_betatin">
  <atom-type> Si </atom-type>
  <lattice-parameter> 3.1 </lattice-parameter>
  <ca-ratio> 0.9 </ca-ratio>
</betaSn-lattice>
```

- `<sc-lattice>`, `<A5-lattice>`<sup>8</sup>. Это обозначение для простой кубической решетки. Есть одна степень свободы – постоянная решетки.

```
<sc-lattice id="Si_sc">
  <atom-type> Si </atom-type>
  <lattice-parameter> 2.6 </lattice-parameter>
</sc-lattice>
```

- `<diamond-lattice>`, `<A4-lattice>`. Это обозначение для решетки типа алмаза. Структура имеет одну степень свободы.

```
<diamond-lattice id="Si_dia">
  <atom-type> Si </atom-type>
  <lattice-parameter> 5.1 </lattice-parameter>
</diamond-lattice>
```

- `omega-lattice`. Это обозначение для структуры омега-решетки. Она имеет две степени свободы – постоянная решетки и отношение  $c/a$ . Омега-структура в первую очередь реализуется для переходных металлов, таких как Ti, Zr

---

<sup>8</sup>На первый взгляд может показаться, что здесь ошибка из-за повтора с предыдущим пунктом. На самом деле ошибки нет, это просто частный случай решетки A5 с отношением  $c/a = 1$ .

или Hf, которые имеют как низкотемпературную гексагональную плотноупакованную, так и высокотемпературную объемно-центрированную кубическую фазу.

```
<omega-lattice id="Ti_omega">  
  <atom-type> Ti </atom-type>  
  <lattice-parameter> 4.57 </lattice-parameter>  
  <ca-ratio> 0.618 </ca-ratio>  
</omega-lattice>
```

### Бинарные и тройные кристаллические структуры.

- `<B1-lattice>`, `<NaCl-lattice>`. Это обозначение для решетки хлорида натрия (NaCl, B1). Структура имеет одну степень свободы – постоянную решетки.

```
<B1-lattice id="ZnO_rocksalt">  
  <atom-type-A> Zn </atom-type-A>  
  <atom-type-B> O </atom-type-B>  
  <lattice-parameter> 2.75 </lattice-parameter>  
</B1-lattice>
```

- `<B2-lattice>`, `<CsCl-lattice>`. Это обозначение для решетки типа хлорида цезия (CsCl, B2). Структура имеет одну степень свободы – постоянную решетки.

```
<B2-lattice id="ZnO_caesiumchloride">  
  <atom-type-A> Zn </atom-type-A>  
  <atom-type-B> O </atom-type-B>  
  <lattice-parameter> 2.64 </lattice-parameter>  
</B2-lattice>
```

- `<B3-lattice>`, `<zincblende-lattice>`. Это обозначение для решетки типа цинковой обманки – сфалерита (ZnS, B3). Структура имеет одну степень свободы – постоянную решетки.

```
<B3-lattice id="ZnO_zincblende">  
  <atom-type-A> Zn </atom-type-A>  
  <atom-type-B> O </atom-type-B>  
  <lattice-parameter> 4.55 </lattice-parameter>  
</B3-lattice>
```

- `<B4-lattice>`, `<wurtzite-lattice>`. Это обозначение для решетки Вюртцита – ещё одна разновидность кристалла ZnS (B4). Имеется три степени свободы: постоянная решетки, отношение  $c/a$  и внутренний параметр  $u$ <sup>9</sup>, для идеальной структуры равный 0.375.

```
<B4-lattice id="ZnO_wurtzite">
  <atom-type-A> Zn </atom-type-A>
  <atom-type-B> O </atom-type-B>
  <lattice-parameter> 3.242 </lattice-parameter>
  <ca-ratio> 1.6 <ca-ratio>
  <u-parameter> 0.38 <u-parameter>
</B4-lattice>
```

- `<Bh-lattice>`, `<WC-lattice>`. Это обозначение для решетки типа карбида вольфрама (Bh). Структура имеет две степени свободы: постоянная решетки и отношение  $c/a$ .

```
<Bh-lattice id="WC">
  <atom-type-A> W </atom-type-A>
  <atom-type-B> C </atom-type-B>
  <lattice-parameter> 2.907 </lattice-parameter>
  <ca-ratio> 0.97 <ca-ratio>
</Bh-lattice>
```

- `<C1-lattice>`, `<fluorite-lattice>`. Это обозначение для решетки типа флюорита – плавиновый шпат ( $CaF_2$ , C1). Структура имеет одну степень свободы – постоянная решетки.

```
<C1-lattice id="CaF2">
  <atom-type-A> Ca </atom-type-A>
  <atom-type-B> F </atom-type-B>
  <lattice-parameter> 5.462 </lattice-parameter>
</C1-lattice>
```

- `<C15-lattice>`. Примеры типа решетки C15:  $Cu_2Mg$ ,  $Fe_2Y$  – с формулой  $A_2B$ . Структура имеет одну степень свободы – постоянная решетки.

---

<sup>9</sup>Параметр  $u$  обозначает абсолютную величину смещения атома ( в кристаллографии – рассеивающего центра). Более подробную информацию можно найти по ключевым словам фактор Дебая-Валлера

```
<C15-lattice id="Fe2Y">
  <atom-type-A> Fe </atom-type-A>
  <atom-type-B> Y </atom-type-B>
  <lattice-parameter> 3.0 </lattice-parameter>
</C15-lattice>
```

- `<D8a-lattice>`. Это обозначение для решетки типа D8a – с формулой  $A_{23}B_6$  ( $Mn_{23}Th_6, Fe_{23}Y_6$ ). Структура имеет одну степень свободы – постоянная решетки.

```
<D8a-lattice id="Fe23Y6">
  <atom-type-A> Fe </atom-type-A>
  <atom-type-B> Y </atom-type-B>
  <lattice-parameter> 4.5</lattice-parameter>
</D8a-lattice>
```

- `<L12-lattice>`. Это обозначение для решетки типа  $L1_2$  (например,  $Ni_3Al, Cu_3Au$ ). Структура имеет одну степень свободы – постоянная решетки.

```
<L12-lattice id="Fe3Pt">
  <atom-type-A> Fe </atom-type-A>
  <atom-type-B> Pt </atom-type-B>
  <lattice-parameter> 3.74 </lattice-parameter>
</L12-lattice>
```

- `<L10-lattice>`. Это обозначение для решетки типа  $L1_0$  ( $NiAl, CuAu$ ). Структура имеет одну степень свободы – постоянная решетки.

```
<L10-lattice id="FePt">
  <atom-type-A> Fe </atom-type-A>
  <atom-type-B> Pt </atom-type-B>
  <lattice-parameter> 3.872 </lattice-parameter>
  <ca-ratio> 0.973 <ca-ratio>
</L10-lattice>
```

**Пользовательские структуры.** В дополнение к заранее определенным структурам, описанным выше, `atomicsex` позволяет подключать определенные пользователем структуры. Они могут быть представлены либо непосредственно во входном файле, либо через другой внешний файл.

Определение пользовательской структуры во входном файле. Структуры могут быть указаны непосредственно во входном файле с помощью элемента `<user-structure>`, который настраивается с использованием нескольких дополнительных элементов. Ниже показан пример для структуры каменной соли GaN:

```
<user-structure id="B1">
  <abc x="true" y="true" z="true" />
  <cell>
    <a1 x="4.22" y="0" z="0" />
    <a2 x="0" y="4.22" z="0" />
    <a3 x="0" y="0" z="4.22" />
    <atoms>
      <atom type="Ga" x="0.1" y="0.0" z="0.0" reduced="true" />
      <atom type="Ga" x="0.56" y="0.5" z="0.2" reduced="true"/>
      <atom type="Ga" x="0.5" y="0.0" z="0.5" reduced="true" />
      <atom type="Ga" x="0.0" y="0.5" z="0.5" reduced="true" />

      <atom type="N" x="0.5" y="0.2" z="0.0" reduced="true" />
      <atom type="N" x="0.0" y="0.5" z="0.0" reduced="true" />
      <atom type="N" x="0.0" y="0.0" z="0.5" reduced="true" />
      <atom type="N" x="0.5" y="0.6" z="0.8" reduced="true" />
    </atoms>
  </cell>

  <relax-dof>
    <atom-coordinates/>
  </relax-dof>
</user-structure>
```

Элементы имеют следующее значение:

- `<abc>`: устанавливает периодические граничные условия по трем декартовым направлениям  $x$ ,  $y$ ,  $z$ .
- `<cell>`: устанавливает размеры ячейки с помощью параметров  $a_1$ ,  $a_2$ ,  $a_3$ .
- `<atoms>`: элемент для списка атомов, установленный для элементов `<atom>`.
- `<atom>`: определите атом, в котором его тип задается с помощью атрибута `<atom>`, а позиции устанавливаются через атрибуты  $x$ ,  $y$ ,  $z$ . Атрибут `reduced` используется для выбора способа задания положений атомов: если

в абсолютных единицах через прямое задание координат через  $x$ ,  $y$ ,  $z$ , то используется атрибут `(reduced="True")`, или в единицах постоянных решетки / нормированных на них `(reduced="True")`

Блок `<relax-dof>` в основном отвечает за релаксацию степеней свободы. Если структура определена через `<user-structure>` то релаксируют только координаты атомов (но не размеры ячейки), что достигается путем вставки элемента `<atom-coordinates>`.

В настоящем примере постоянная решетки непосредственно включена в блок `<cell>`. Также можно установить постоянную решетки путем применения деформации, и в этом случае постоянная решетки входит в качестве коэффициента масштабирования. Это продемонстрировано в этом примере (ссылка на релаксацию пользовательской структуры).

**Пользовательские структуры из внешних файлов.** Это особенно полезно при работе с большим количеством конфигураций, например, для согласования сил. В настоящее время `atomicrex` поддерживает входные файлы в формате дампа LAMMPS и модифицированный формат POSCAR, который также содержит атомные силы.

Добавление других файлов во входной является простым и просто требует вставки элемента, специфичного для каждого формата. Чтобы включить структуру в формат дампа LAMMPS, используется элемент `<lammps-dump>`:

```
<user-structure id="my_structure_from_lammps">
  <lammps-dump>my_lammps_file.dump</lammps-dump>
</user-structure>
```

Для файла формата POSCAR (VASP) `<poscar-file>`:

```
<user-structure id="my_structure_from_somewhere_else">
  <poscar-file>my_POSCAR_file</poscar-file>
</user-structure>
```

**Дамп файлы LAMMPS.** Формат дампа LAMMPS позволяет предоставить несколько возможностей. `Atomicrex` анализирует поля файлов, связанные с положением (`x/y/z`, `xu/yu/zu` и `xs/ys/zs`), силой `fx/fy/fz`, типа `<type>` и идентификатора `id`. Кроме того, он пытается прочитать данные смещения атомов `dx/dy/dz`, которые используются для вычисления определенных силовых констант.

```
ITEM: TIMESTEP
```

```
0
```

```
ITEM: NUMBER OF ATOMS
```

108

ITEM: BOX BOUNDS pp pp pp

0 8.7915

0 15.2273

0.160821 14.1956

ITEM: ATOMS id xu yu zu fx fy fz

1 0.44 0 0.160821222 -2.309895 -0.016848 -0.000000

2 1.46524925 2.53788615 0.160821222 0.101288 0.468496 -0.000000

...

**Структуры с точечными дефектами.** Для удобства и эффективности вычислений в структурах с точечными дефектами некоторые операции были реализованы непосредственно в Atomicrex. Следующий фрагмент кода иллюстрирует определение точечного дефекта во входном файле:

```
<point-defect id="Vac">
  <atom-type-A>Fe</atom-type-A>
  <lattice-type>bcc</lattice-type>
  <cell-size>4</cell-size>
  <lattice-parameter> 2.86 </lattice-parameter>
  <defects>
    <vacancy index="1" />
    <substitutional index="1" atom-type="Y" />
    <interstitial x="0.5625" y="0.5625" z="0.5625" atom-type="Y" />
  </defects>
  <properties>
    <lattice-parameter equalto="bcc-Fe.lattice-parameter"/>
  </properties>
</point-defect>
```

Здесь определяется суперячейка, содержащая точечный дефект. Атрибуты:

- `<atom-type-A>`: тип атома в суперячейке.
- `<lattice-type>`: тип кристаллической решетки. Доступно bcc (ОЦК), fcc (ГЦК), hcp (ГПУ). Структуры с точечными дефектами основываются на определениях суперячейки.
- `<cell-size>`: размеры суперячейки, 4 соответствует кубу из  $4*4*4=64$  элементарных ячеек, объединенных в одну.
- `<lattice-parameter>`: параметр решетки суперячейки.

- `<ca-ratio>`: отношение  $c/a$  (только для ГПУ решетки).

Блок `<defects>` добавляет точечные дефекты к суперячейке. Возможны 3 разных типа дефектов. Вакансия имеет атрибут `<index>`, который определяет, какой атом будет удален из суперячейки. Замещающий атом создается путем изменения типа атома атома с индексом на тип атома. Создание промежуточного атома требует позицию промежуточного элемента ( $x, y, z$ ) в нормированных координатах элементарной ячейки и типа добавляемого атома. Ограничений на количество и тип дефектов нет, это позволяет создавать все возможные комплексы точечных дефектов.

Вычисление энергий образования дефектов или энергий связи между дефектами может быть удобно выполнено с использованием вычисленных величин (см. следующий раздел - "Свойства" и пункт "Объявленные свойства"). Энергии точечных дефектов часто чувствительны к значениям постоянной решетки. Поэтому удобно устанавливать постоянную решетки, равной релаксированной постоянной решетки соответствующего типа решетки. Это можно сделать с помощью атрибута `equalto` свойства `lattice-parameter`. Использование структуры точечного дефекта продемонстрировано в примере `examples/derived_properties`.

**Релаксация.** Есть возможность релаксировать структуры во время процесса построения. Релаксация координат атомов возможна для любой структуры, но релаксация других степеней свободы, таких, как постоянная решетки, отношение  $c/a$  для пользовательских структур невозможна. Релаксация степеней свободы возможна с помощью блока `<relax-dof>`:

```
<relax-dof>
  <atom-coordinates relax="false"/>
  <lattice-parameter reset="false" />
</relax-dof>
```

В этом примере релаксация параметра решетки активна, в то время как релаксация координат атома явно деактивируется с помощью флага `relax`. Флаг `reset` контролирует, сбрасывается ли степень свободы к своему начальному значению до релаксации. Это делается по умолчанию из-за того, что неправильные наборы параметров могут привести к значительному искажению структуры. Если искажение не устранено, релаксация при следующем наборе параметров может занять много времени или даже провалиться. Если наборы параметров нестрогие, может быть лучше начать каждое последующую релаксацию из предыдущего состояния. В этом случае опция команды `reset` устанавливается на "false". Это может ускорить процесс релаксации.

Релаксация степени свободы активируется автоматически, если она связана со



степенью свободы, используемой в процессе построения. Пример с релаксацией параметра решетки, объявленной также и степенью свободы.

```
<properties>
  <lattice-parameter fit='true' target='3.615' min='3' max='4' />
</properties>
```

Обратите внимание на определение верхней и нижней границ для изменения параметра решетки. Это важно, чтобы убедиться, что расстояние между атомами не превышает расстояния отсечки потенциала.

**Деформации.** Может быть полезным подвергнуть определенную пользователем структуру аффинному преобразованию, например, изменению объема или деформации ячейки. Это может быть удобно выполнено с использованием механизма, предусмотренного элементами `<deformation>` и `<deformations>`. Сначала необходимо определить деформацию, это достигается путем вставки блока `<deformations>` вне блока `<structures>`, который может содержать несколько элементов `<deformation>`, которые должны быть обозначены с использованием атрибута `id`. Каждый элемент `<deformation>` определяет отдельную деформацию, которая будет использоваться позже при определении структур. Элемент `<deformation>` может содержать любую комбинацию элементов `<scale>` и `<matrix>`, как показано следующим кодом:

```
<deformations>
  <deformation id="volume_strain">
    <scale> 1.05 </scale>
  </deformation>

  <deformation id="tetragonal_strain">
    <matrix>
      <scale> 3.615 </scale>
      <s1 x="1.05" y="0.0" z="0.0"/>
      <s2 x="0.0" y="1.0" z="0.0"/>
      <s3 x="0.0" y="0.0" z="1.0"/>
    </matrix>
  </deformation>

  ...
</deformations>
```

Первая деформация изменяет один параметр ячейки в 1,05 раза и, таким образом, предполагает деформацию объема на 5%. Второй элемент деформации определяет тетрагональную деформацию вдоль оси  $x$  через блок `<matrix>`; кроме того,

он предписывает изменение масштаба всех параметров ячейки в 3,615 раз. Эти деформации могут затем использоваться в определении структур, как показано в следующем фрагменте кода:

```
<structures>
  <user-structure id="conventional_fcc_cell">
    <pbcs x="true" y="true" z="true" />
    <cell>
      <a1 x="1" y="0" z="0" />
      <a2 x="0" y="1" z="0" />
      <a3 x="0" y="0" z="1" />
      <atoms>
        <atom type="Cu" x="0.0" y="0.0" z="0.0" reduced="true" />
        <atom type="Cu" x="0.0" y="0.5" z="0.5" reduced="true" />
        <atom type="Cu" x="0.5" y="0.0" z="0.5" reduced="true" />
        <atom type="Cu" x="0.5" y="0.5" z="0.0" reduced="true" />
      </atoms>
    </cell>
    <deformation> tetragonal_strain </deformation>
  </user-structure>
</structures>
```

Здесь использована деформация с именем `tetragonal_`, объявленная через элемент `<deformation>` внутри конструкции `<user-structure>`. Конечные параметры ячейки следующие:

$$h = 3.615 \begin{pmatrix} 1.05 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (33)$$

Практический пример кода `examples/imposing_a_deformation`.

**Группировка структур.** Во время выполнения программы часто удобно назначать структуры различным группам, например, для разделения входных данных на данные для построения и тестовую базу данных. Это может быть достигнуто с помощью элемента `<group>`. В следующем примере структуры в первой группе не оцениваются во время фазы построения (`exclude-fit = 'true'`), в то время как структуры во второй группе имеют вид (`exclude-fit = 'true'`, поведение по умолчанию); структуры в обеих группах включены во время вывода (`exclude-output = 'false'`):

```

<group exclude-fit="true" exclude-output="false">
  <B1-lattice id="B1_ZnS">
    ...
  </B1-lattice>

  <B2-lattice id="B2_ZnS">
    ...
  </B2-lattice>
</group>

<group exclude-fit="false" exclude-output="false">
  <B3-lattice id="B3_ZnS">
    ...
  </B3-lattice>

  <C1-lattice id="C1_ZnS">
    ...
  </C1-lattice>
</group>

```

## 6 Свойства

**Структурные свойства.** Учитывая структуру и соответствующую модель взаимодействия, можно рассчитать различные свойства. Atomicrex обеспечивает функциональность для вычисления большого количества различных свойств, включая когезионную энергию, параметры решетки, упругие свойства или атомные силы. Эти свойства можно сравнить с целевыми значениями и добавить с соответствующим весом к целевой функции. Пример кода:

```

<structures>
  <fcc-lattice id="FCC">
    <atom-type>Al</atom-type>
    <lattice-parameter>4.0</lattice-parameter>
    <properties>
      <atomic-energy relax='true' fit='true' target='-3.36'
        relative-weight='100' residual-style='squared' />
    </properties>
  ...
</structures>

```

Здесь ГЦК решетка определена для алюминия с начальным параметром решетки 4.0. Когезионная энергия на атом (`atomic-energy`) указывается как свойство.

Atomicrex различает скалярные свойства и векторные свойства. Большинство свойств, перечисленных ниже, является скалярным. Наиболее распространенным векторным свойством являются атомные силы. Поведение свойства можно настроить с помощью атрибутов элемента `<property>`:

- `<fit>`: Если для этого атрибута установлено значение `true`, свойство включается во время построения и добавляется к целевой функции.
- `target` (только для скаляров): устанавливает целевое значение для свойства.
- `min`, `max` (только для скаляров): эти атрибуты позволяют ограничить структурную релаксацию, которая останавливается, если границы, определенные здесь, превышены. Это может быть полезно для предотвращения сбоя алгоритмов оптимизации (в частности, глобальных оптимизаторов).
- `relative-weight`: этот атрибут устанавливает вес, свойство которого влияет на целевую функцию; Это соответствует параметру  $w_P$  в определении целевой функции.
- `residual-style`: этот атрибут определяет способ вычисления остатка, связанного с этим свойством, который входит в сумму целевой функции. Возможные значения:

– `squared`

$$r = (\Delta A / \delta)^2 \quad (34)$$

– `squared-relative`

$$r = (\Delta A / A^{target})^2 \quad (35)$$

– `absolute-diff`

$$r = (|\Delta A / \delta|)^2 \quad (36)$$

Здесь,  $\Delta = A^{predicted} - A^{target}$ , где  $A^{predicted}$  и  $A^{target}$  вычисленное и целевое значения свойств соответственно. параметр  $\delta$  это значение атрибута `tolerance`, определенное для свойства.

- `tolerance`: параметр точности используется при расчете остатка (сравните с `residual-style`): Начальное значение 1.0. Больше информации об этом можно найти в разделе "Краткий обзор – Целевая функция".
- `relax`: если для этого атрибута установлено значение "True структура подвергается релаксации до расчета свойства.

- **output**: если для этого атрибута установлено значение "True" свойство записывается в стандартный вывод во время фазы вывода, а также во время фазы построения, если значение параметра **fit** "True".
- **output-all** (только для векторных свойств): если для этого атрибута установлено значение "True" будут напечатаны все компоненты вектора, а не только остаток всего вектора.

В следующей таблице представлен обзор различных свойств, которые могут быть вычислены (и указаны в процессе построения модели). В дополнение к перечисленным свойствам также могут быть установлены различные параметры, связанные с предварительно определенными кристаллическими структурами, такие, как **<atom-distance>** (димерная структура), **<lattice-parameter>**, **<ca-ratio>**, **<u-parameter>** (структура вюрцита) и прочих.

Элемент	Тип	Комментарий
<b>&lt;atomic-energy&gt;</b>	Скаляр	Энергия, приходящаяся на один атом – когезионная энергия (в эВ/атом)
<b>&lt;atomic-forces&gt;</b>	Вектор	Атомные силы (в эВ/Å)
<b>&lt;bulk-modulus&gt;</b>	Скаляр	Модуль всестороннего сжатия (в ГПа)
<b>&lt;pressure&gt;</b>	Скаляр	Гидростатическое давление (в Бар)
<b>&lt;pxx&gt;</b>	Скаляр	<i>xx</i> компонента тензора напряжения (1-ая компонента в записи Фойгта)
<b>&lt;pyy&gt;</b>	Скаляр	<i>yy</i> компонента тензора напряжения (2-ая компонента в записи Фойгта)
<b>&lt;pzz&gt;</b>	Скаляр	<i>zz</i> компонента тензора напряжения (3-ая компонента в записи Фойгта)
<b>&lt;pyz&gt;</b>	Скаляр	<i>yz</i> компонента тензора напряжения (4-ая компонента в записи Фойгта)
<b>&lt;pxz&gt;</b>	Скаляр	<i>xz</i> компонента тензора напряжения (5-ая компонента в записи Фойгта)
<b>&lt;pxy&gt;</b>	Скаляр	<i>yz</i> компонента тензора напряжения (6-ая компонента в записи Фойгта)
<b>&lt;Cij&gt;</b>	Скаляр	Тензор жесткости в записи Фойгта

Расчет упругих свойств, в частности, упругих постоянных как «закрепленного иона», так и «релаксированного иона» для гексагональной решетки, продемонстрирован в примерах по расчету упругих свойств меди:

Путь: [examples/elastic\\_constants\\_Cu](#)

и по расчету упругих постоянных "закрепленного" и релаксированного иона.

Путь: [examples/elastic\\_constants\\_Si](#).

**Объявляемые (пользователем) свойства.** Объявляемые свойства не связаны напрямую со структурой, но могут быть рассчитаны по свойствам одной или нескольких структур. Это позволяет рассчитывать такие величины, как энергия образования дефектов, поверхностные энергии и разности энергий между различными кристаллическими структурами. Даже сложные свойства, такие как дислокации ядер и фононные частоты, могут быть в принципе рассчитаны.

```
<derived-properties>
  <derived-property id="bccFe-Ev" unit="eV" fit="true" target="2.1"
    relative-weight="1" equation="[Vac.total-energy] -
    127*[bcc-Fe.atomic-energy]" "/>
</derived-properties>
```

Здесь энергия образования вакансии в ОЦК железе рассчитывается путем вычитания атомной энергии 127 атомов железа из полной энергии суперячейки, содержащей вакансию. Объявляемые свойства определяются в отдельном блоке xml под определениями структуры. Они могут быть сгруппированы так же, как структуры. Использование производных свойств демонстрируется в этом примере – вычисление произвольного свойства используя структуру с точечным дефектом, путь: [examples/derived\\_properties](#).

Сравнивая с "нормальными" свойствами, объявляемые свойства имеют два дополнительных атрибута:

- **id**: идентификатор.
- **equation**: Уравнение анализируется математическим парсером и используется для вычисления значения производного свойства. Свойства структуры указываются с использованием формата `[structure.property]`.

## 7 Примеры

Atomicrex содержит много различных примеров, находящихся в директории [examples](#), включая наборы входных и справочных выходных данных. Эти примеры можно использовать для тестирования кода, например, во время отладки или после компиляции на новой платформе. Здесь будет только описано, какие примеры доступны, и где они находятся:

- Метод погруженного атома (EAM) с пользовательскими функциями.  
Путь: [examples/potential\\_EAM](#)
- Модифицированный метод погруженного атома с пользовательскими функциями.  
Путь: [examples/potential\\_MEAM](#)

- Потенциал Стиллинджера-Вебера с пользовательскими функциями.  
Путь: `examples/potential_SW`
- Построение парного потенциала с пользовательскими функциями.  
Путь: `examples/fitting_pair_potential`
- Метод погруженного атома для алюминия.  
Путь: `examples/fitting_EAM_potential`
- Построение потенциала аналитического пеорядка связи (АВОР).  
Путь: `examples/fitting_ABOR_potential`
- Построение потенциала с использованием Spa минимизатора.  
Путь: `examples/fitting_Spa_minimizer`
- Построение потенциала с использованием NLopt библиотеки.  
Путь: `examples/fitting_NLopt_minimizer`
- Вычисление объявленного пользователем свойства с использованием структуры с точечным дефектом.  
Путь: `examples/derived_properties`
- Использование деформаций со структурами, определенными пользователем.  
Путь: `examples/imposing_a_deformation`
- Упругие свойства меди  
Путь: `examples/elastic_constants_Cu`
- Упругие свойства нитрида галлия.  
Путь: `examples/elastic_constants_GaN`
- Упругие постоянные "фиксированного" и отрелаксированного" ионов.  
Путь: `examples/elastic_constants_Si`
- Добавление библиотечных структур (Python)  
Путь: `examples/python_interface`
- Добавление структур из ASE (Python)  
Путь: `examples/python_interface`
- Построение потенциала, основы (Python)  
Путь: `examples/python_basic_fitting`
- ASE калькулятор (Python)  
Путь: `examples/python_ase_calculator`

- Перечень команд для интерфейса Python.



## Список литературы

- [1] A. E. Carlsson, Beyond Pair Potentials in Transition Metals and Semiconductors, Solid State Physics 43, 1 (1990); series published by Academic Press, edited by H. Ehrenreich and D. Turnbull doi:10.1088/0965-0393/8/6/305
- [2] F. H. Stillinger and T. A. Weber, Computer simulation of local order in condensed phases of silicon, Phys. Rev. B 31, 5262 (1984); doi:10.1103/PhysRevB.31.5262
- [3] D. Frenkel, Simulations: the dark side, arXiv:1211.4440 [cond-mat.stat-mech];
- [4] G. C. Abell, Empirical chemical pseudopotential theory of molecular and metallic bonding, Phys. Rev. B 31, 6148 (1985); doi:10.1103/PhysRevB.31.6184
- [5] J. Tersoff, New Empirical Model for the Structural Properties of Silicon, Phys. Rev. Lett. 56, 632 (1986); doi:10.1103/PhysRevLett.56.632
- [6] J. Tersoff, New empirical approach for the structure and energy of covalent systems, Phys. Rev. B 37, 6991 (1988); doi:10.1103/PhysRevB.37.6991
- [7] J. Tersoff, Empirical interatomic potential for silicon with improved elastic properties, Phys. Rev. B 38, 9902 (1988); doi:10.1103/PhysRevB.38.9902
- [8] J. Tersoff, Empirical Interatomic Potential for Carbon, with Applications to Amorphous Carbon, Phys. Rev. Lett. 61, 2879 (1988); doi:10.1103/PhysRevLett.61.2879
- [9] J. Tersoff, Modeling solid-state chemistry: Interatomic potentials for multicomponent systems, Phys. Rev. B. 39, 5566 (1989); doi:10.1103/PhysRevB.39.5566
- [10] D. W. Brenner, Relationship between the Embedded-Atom Method and Tersoff Potentials, Phys. Rev. Lett. 63, 1022 (1989); doi:10.1103/PhysRevLett.63.1022
- [11] D. W. Brenner, Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films, Phys. Rev. B 42, 9458 (1990); doi:10.1103/PhysRevB.42.9458
- [12] D. W. Brenner, O. A. Shenderov, J. A. Harrison, S. J. Stuart, B. Ni and S. B. Sinnott, A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons, J. Phys. Cond. Matter 14, 783 (2002); doi:10.1088/0953-8984/14/4/312

- [13] K. Albe, K. Nordlund, and R. S. Averback, Modeling the metal-semiconductor interaction: Analytical bond-order potential for platinum-carbon, *Phys. Rev. B* 65, 195124 (2002); doi:10.1103/PhysRevB.65.195124
- [14] K. Albe, K. Nordlund, J. Nord, and A. Kuronen, Modeling of compound semiconductors: Analytical bond-order potential for Ga, As, and GaAs, *Phys. Rev. B* 66, 035205 (2002); doi:10.1103/PhysRevB.66.035205
- [15] J. Nord, K. Albe, P. Erhart, and K. Nordlund, Modelling of compound semiconductors: Analytical bond-order potential for gallium, nitrogen and gallium nitride, *J. Phys. Cond. Matter* 15, 5649 (2003); doi:10.1088/0953-8984/15/32/3245
- [16] P. Erhart and K. Albe, Analytical Potential for Atomistic Simulations of Silicon and Silicon Carbide, *Phys. Rev. B* 71, 035211 (2005); doi:10.1103/PhysRevB.71.035211
- [17] P. Erhart, N. Juslin, O. Goy, K. Nordlund, R. Müller, and K. Albe, Analytic bond-order potential for atomistic simulations of zinc oxide, *J. Phys. Cond. Matter* 18, 6585 (2006); doi:10.1088/0953-8984/18/29/003
- [18] N. Juslin, P. Erhart, P. Träskelin, J. Nord, K. Henriksson, E. Salonen, K. Nordlund, and K. Albe, Analytical interatomic potential for modeling nonequilibrium processes in the W-C-H system, *J. Appl. Phys.* 98, 123520 (2005); doi:10.1063/1.2149492
- [19] M. Müller, P. Erhart, and K. Albe, Analytic bond-order potential for bcc and fcc iron - comparison with established embedded-atom method potentials, *J. Phys. Cond. Matter* 19, 326220 (2007); doi:10.1088/0953-8984/19/32/326220
- [20] M. Müller, P. Erhart, and K. Albe, Thermodynamics of L10 ordering in FePt nanoparticles studied by Monte Carlo simulations based on an analytic bond-order potential, *Phys. Rev. B* 76, 155412 (2007); doi:10.1103/PhysRevB.76.155412
- [21] M. V. G. Petisme, M. A. Gren, and G. Wahnström, Molecular dynamics simulation of WC/WC grain boundary sliding resistance in WC-Co cemented carbides at high temperature, *Internat. J. Refractory Met. Hard Mater.* 49, 75 (2015); doi:10.1016/j.ijrmhm.2014.07.037
- [22] M. J. Stott and E. Zaremba, Quasiatoms: An approach to atoms in nonuniform electronic systems, *Phys. Rev. B* 22, 1564 (1980); doi:10.1103/PhysRevB.22.1564
- [23] M. J. Puska, R. M. Nieminen, and M. Manninen, Atoms embedded in an electron gas: Immersion energies, *Phys. Rev. B* 24, 3037 (1981); doi:10.1103/PhysRevB.24.3037

- [24] J. K. Nørskov, Covalent effects in the effective-medium theory of chemical binding: Hydrogen heats of solution in the 3d metals, *Phys. Rev. B* 26, 2875 (1982); doi:10.1103/PhysRevB.26.2875
- [25] M. W. Finnis and J. E. Sinclair, A simple empirical N-body potential for transition metals, *Phil. Mag. A* 50, 45 (1984); doi:10.1080/01418618408244210
- [26] M. S. Daw and M. I. Baskes, Semiempirical, Quantum Mechanical Calculation of Hydrogen Embrittlement in Metals, *Phys. Rev. Lett.*, 50, 1285 (1983); doi:10.1103/PhysRevLett.50.1285
- [27] M.S. Daw and M. I. Baskes, Embedded-atom method: Derivation and application to impurities, surfaces and other defects in metals, *Phys. Rev. B* 29, 6443 (1984); doi:10.1103/PhysRevB.29.6443
- [28] M. S. Daw, S. M. Foiles, and M. I. Baskes, Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys, *Phys. Rev. B* 33, 7983 (1986); doi:10.1103/PhysRevB.33.7983
- [29] M. S. Daw, S. M. Foiles, and M. I. Baskes, The embedded-atom method - A review of theory and applications, *Mater. Sci. Rep.* 9, 251 (1993); doi:10.1016/0920-2307(93)90001-U
- [30] F. Ercolessi, E. Tosatti, and M. Parrinello, Au (100) Surface Reconstruction, *Phys. Rev. Lett.* 57, 719 (1986); doi:10.1103/PhysRevLett.57.719
- [31] F. Ercolessi and J. B. Adams, Interatomic potentials from first-principles calculations: the force-matching method, *Europhys. Lett.* 26, 583 (1994); doi:10.1209/0295-5075/26/8/005
- [32] Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, A. F. Voter, and J. D. Kress, Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations, *Phys. Rev. B* 63, 224106 (2001); doi:10.1103/PhysRevB.63.224106
- [33] M. Baskes, Application of the Embedded-Atom Method to Covalent Materials: A Semiempirical Potential for Silicon, *Phys. Rev. Lett.* 59, 2666 (1987); doi:10.1103/PhysRevLett.59.2666
- [34] M. Baskes, Modified embedded-atom potentials for cubic materials and impurities, *Phys. Rev. B* 46, 2727 (1992); doi:10.1103/PhysRevB.46.2727

- [35] B.-J. Lee, M. I. Baskes, H. Kim, and Y. K. Cho, Second nearest-neighbor modified embedded atom method potentials for bcc transition metals, *Phys. Rev. B* 64, 184102 (2001); doi:10.1103/PhysRevB.64.184102
- [36] T. J. Lenosky, B. Sadigh, Babak, E. Alonso, V. V. Bulatov, T. Diaz de la Rubia, J. Kim, A. F. Voter, and J. D. Kress, Highly optimized empirical potential model of silicon, *Modelling Simul. Mater. Sci. Eng.* 8, 825 (2000); doi:10.1088/0965-0393/8/6/305