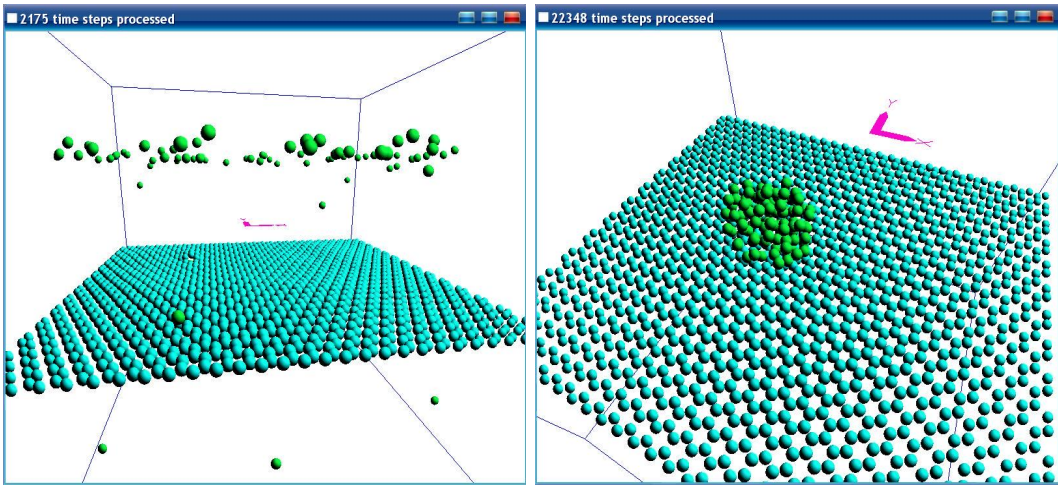
SurfaceGrowth

модель, структура кода, инструкция пользователю

Николай Проданов, май 2010



Содержание

1 О чем этот документ

Данный документ является описанием приложения SurfaceGrowth – программного продукта, предназначенного для моделирования трения металлических наночастиц на поверхности графена и других явлений методом молекулярной динамики и реализованного с использованием технологии NVIDIA® CUDA™. Изложение начинается с физических принципов, лежащих в основе функционирования программы, и заканчивается работой с интерфейсом приложения и некоторыми результатами. Краткое содержание основных разделов:

2 **Введение** – представлена постановка задачи.

3 **Модель** – кратко знакомит с принципами молекулярной динамики и методами, использованными в программе SurfaceGrowth.

4 **Структура** – представлены архитектура приложения SurfaceGrowth и описание кода.

5 **Инструкция пользователю** – даны указания по установке и использованию приложения SurfaceGrowth.

6 **Результаты** – описаны некоторые результаты, полученные с использованием SurfaceGrowth.

До начала использования приложения рекомендуется последовательно прочитать по крайней мере первые 5 разделов документа, поскольку пользовательский интерфейс предполагает знание того, какое значение имеет тот или иной параметр для функционирования программы.

2 Введение

Изучение поверхностных явлений на атомарном уровне вызывает значительный интерес на протяжении последних двух десятилетий, что обусловлено развитием соответствующих экспериментальных методик, например, атомной силовой микроскопии, а также интенсивной миниатюризацией электронных и других устройств [1]. Кроме исследования традиционных явлений, например, нанесения покрытий или напыления тонких пленок, довольно быстро развивается новая область нанотехнологии – *нанотрибология*, рассматривающая трение и износ поверхностей на атомарном уровне [2, 3].

При исследовании нанотрибологических явлений часто требуются атомарно-гладкие поверхности. В связи с этим высокоориентированный пиролитический графит (ВОПГ) занимает особенное место в нанотрибологии, поскольку является материалом, позволяющим относительно легко получить атомарно-гладкие поверхности. Эксперименты по изучению трения графита на наноуровне открыли ряд интересных явлений (см., например, ссылки в [4]). Отметим лишь опыт, имеющий непосредственное значение для программы SurfaceGrowth: в работе [5] исследовалось трение металлических (Sb) наночастиц, выращенных на поверхности ВОПГ. Авторы обнаружили существование так называемой фрикционной дуальности: некоторые частицы проявляли конечное трение, возрастающее линейно с ростом площади поверхности контакта, а для других частиц имело место состояние скольжения без трения.

Отсутствие в литературе достоверного объяснения фрикционной дуальности указывает на необходимость теоретического исследования трения металлических наночастиц по поверхности графита. Сложность и многофакторность нанотрибологических задач делает фактически неосуществимым построение достоверной аналитической теории рассматриваемых явлений. Поэтому компьютерное моделирование, в частности метод молекулярной динамики (МД), является незаменимым инструментом теоретического изучения трения и износа на атомарном уровне [2, 6].

Описываемая в данном документе программа SurfaceGrowth является первым шагом на пути всестороннего теоретического исследования трения металлических наночастиц по поверхности графита с использованием метода классической МД. Желание приблизить модель к экспериментальным условиям обусловило использование реалистичных потенциалов для межатомных взаимодействий. Данный факт, а также возможность рассмотрения относительно больших систем привели к значительным вычислительным затратам и к необходимости использования параллельных вычислений. В SurfaceGrowth они основываются на тех-

нологии NVIDIA® CUDA™.

SurfaceGrowth предоставляет возможность изучения движения металлических наночастиц по поверхности графена – одного слоя атомов углерода, упакованных в решетку, составленную из "пчелиных сот" (касательно свойств графена см., например, ссылки в [7]). Несмотря на то, что графит состоит из множества слоев графена, рассмотрение одного слоя может быть полезным в качестве первого приближения для исследования трения наночастиц на поверхности графита. SurfaceGrowth предоставляет выбор одного из шести металлов, имеющих гранецентрированную кубическую (ГЦК) решетку в объемном состоянии (при нормальных условиях): меди (Cu), серебра (Ag), золота (Au), никеля (Ni), алюминия (Al) и свинца (Pb). В процессе моделирования сдвига частиц, состоящих из атомов выбранного металла, производится измерение различных параметров: полного импульса системы, полной и потенциальной энергий, температуры, скорости и положения центра масс, размеров наночастицы, сил трения и сдвига, действующих на частицу, а также функции радиального распределения, позволяющей следить за структурой наночастицы. Кроме того, систему можно визуализировать встроенными инструментами или с использованием программы Visual Molecular Dynamics (VMD) [8], доступной для свободного использования и также предусматривающей возможность использования технологии NVIDIA® CUDA™.

Кроме сдвига в SurfaceGrowth предусмотрены режимы моделирования объемных свойств выбранного металла (режим "`Bulk`"), а также напыления на графен (режим "`Surface Growth`"). Первый режим предназначен для проверки правильности расчетов для металлов, в частности значений энергии, полного импульса и атомной структуры. Второй режим является первым шагом на пути полного воспроизведения экспериментов, в которых наночастицы получают напылением. В данной реализации SurfaceGrowth в режиме `Surface Growth` производится просто наблюдение за процессом напыления атомов металла на слой графена без возможности дальнейшего сдвига полученных структур.

3 Модель

3.1 Принципы молекулярной динамики

Суть атомистического компьютерного моделирования методом классической молекулярной динамики (МД) довольно проста: имея набор начальных условий и способ описания межатомных сил, необходимо проинтегрировать ньютоновские классические уравнения движения одним из нескольких стандартных методов. Моделирования дают новые атомные координаты, скорости и силы, и за поведением атомов можно следить в режиме реального времени с помощью изображений-анимаций, а также можно производить измерения различных параметров системы. Блок-схема метода представлена на Рис. 1.

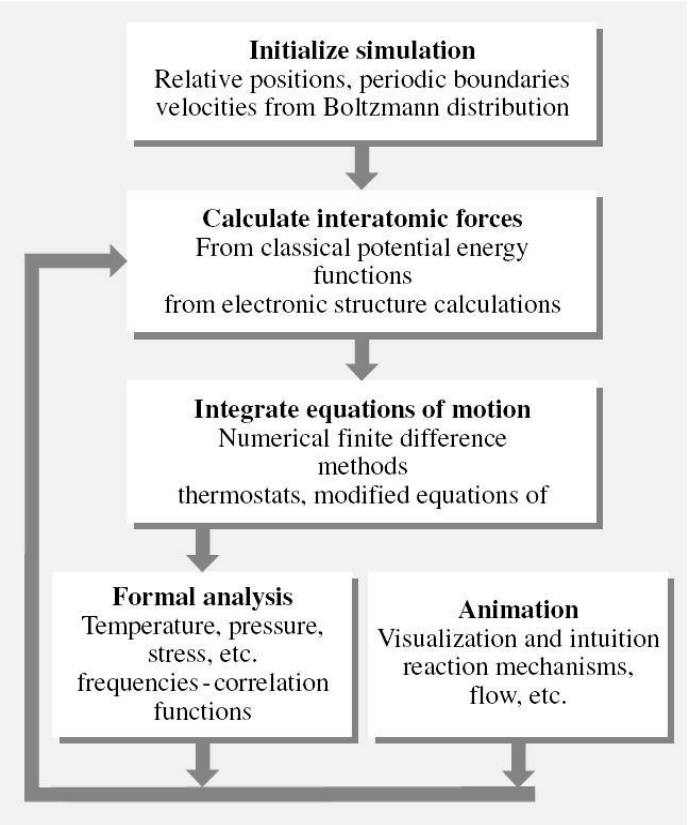


Рис. 1: Блок-схема метода классической МД (из работы [9]).

Однако эффективное использование метода МД требует понимания многих деталей, не являющихся очевидными в указанном простом анализе. Их подробное описание и алгоритмы, используемые в классической

МД, можно найти в книгах [10,11,12,13]. В данном документе многие детали, например, периодические границы, назначение безразмерных единиц и т. д. не представлены, и читателю предлагаем обратиться к указанным выше источникам. Мы коснемся вопросов, являющихся принципиальными при реализации МД на графических процессорах (graphic processing units, GPU), и, таким образом, имеющих большое влияние на строение приложения SurfaceGrowth.

Уравнения Ньютона, дающие ускорение каждого атома в ответ на действие приложенной к нему силы, имеют вид:

$$\mathbf{F} = m\mathbf{a}, \tag{1}$$

$$-\nabla V = m(\partial^2 \mathbf{r}/\partial t^2), \tag{2}$$

где $\mathbf{F}$ – сила, действующая на каждый атом, m – масса атома, $\mathbf{a}$ – ускорение атома, V – потенциальная энергия атома, $\mathbf{r}$ – пространственная координата атома и t – время. В начале моделирования вычисляются силы, действующие на каждый атом. Затем атомы двигаются малый промежуток времени Δt (называемый временным шагом) в ответ на действие приложенных сил, что сопровождается изменением атомных координат, скоростей и ускорений. Описанный процесс повторяется для определенного числа временных шагов, выбранного пользователем.

Преимущество этого метода в том, что он дает возможность следить за индивидуальным движением всех атомов в режиме реального времени. Недостатком является то, что временные масштабы очень ограничены (от пико- до наносекунд). К тому же размеры системы, которые можно рассматривать на данный момент, ограничены числом в 10^6 – 10^8 атомов, которое хотя и впечатляет, однако все еще далеко отстоит от реальных систем, содержащих 10^{23} атомов и более. Следовательно, хотя МД моделирования используются для изучения трения на атомарном уровне и сделали хорошую работу для обеспечения понимания этого процесса, однако они все еще ограничены пространственными и временными масштабами, которые значительно меньше большинства экспериментальных значений. Интенсивное развитие параллельных вычислений, в частности с использованием GPU, может значительно ослабить указанные ограничения.

Для атомистического моделирования материала необходимы математические выражения его потенциальной энергии (V в (2)). В программе SurfaceGrowth используются эмпирические потенциалы, в которых предполагается, что потенциальная энергия атомов может быть представлена как функция только их относительных положений. Подробности даны

в разделе 3.3.1. Здесь же стоит отметить, что для вычисления потенциальной энергии атома и соответственно силы, действующей на него, необходимо рассчитать расстояния до всех других атомов. Вследствие отсутствия дальнодействующих вкладов для многих потенциалов можно опустить рассмотрение удаленных соседей для данного атома, и часто вводится радиус отсечки r_c. Атомы, расстояние до которых менее r_c, дают вклад в энергию и силу для рассматриваемого атома, а остальные атомы – нет. Однако простой перебор всех атомов и сравнение расстояния до них с r_c – так называемый метод *всех пар* (см. левую часть Рис. 2) – приводит к тому, что количество требуемых вычислений растет пропорционально N^2, где N – число атомов. Это делает метод всех пар непригодным для крупных систем.

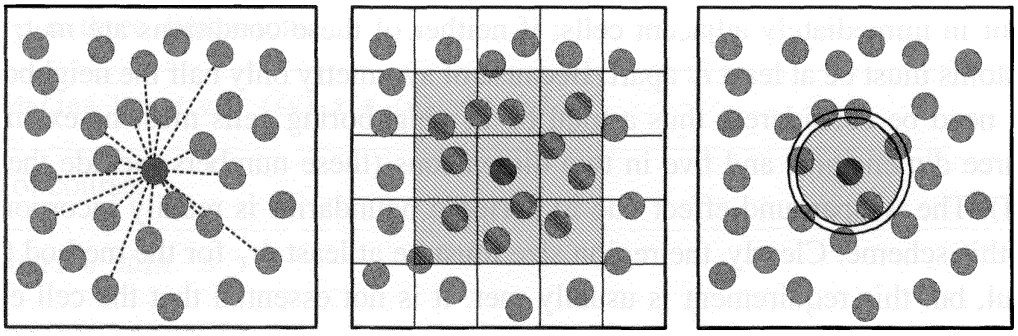


Рис. 2: Схематическое изображение подходов к расчету взаимодействий: все пары, разделение на ячейки, список соседей (из работы [12]).

Разработаны другие методики, в частности метод *ячеек* и использование *списка соседей* (соответственно центральный и правый рисунки на Рис. 2), дающие линейное возрастание вычислительных затрат с N.

В первом методе область моделирования разбивается на ячейки размером не менее r_c. Тогда, если атомы попадают в ячейку исходя из их текущих координат, то очевидно, что взаимодействия возможны только между атомами, находящимися или в одинаковой ячейке, или в непосредственно соседних. Если ни одно из этих условий не удовлетворяется, то расстояние между атомами равно как минимум r_c, и такие атомы не дают вклада в силу и энергию, и их можно не рассматривать.

В методе списка соседей учитывается то, что только малая доля атомов, перебираемых в методе ячеек, лежит внутри радиуса взаимодействия. Поэтому сначала, используя метод ячеек, строится список пар соседей, не только дающих вклад в силу и энергию, но также и некоторого числа атомов, лежащих непосредственно вне радиуса взаимодействия.

Для этого радиус отсечки r_c при сравнении межатомных расстояний заменяется на

$$r_n = r_c + \Delta r. \tag{3}$$

А далее, основываясь на том, что микроскопическая среда изменяется медленно, данный список используется на протяжении нескольких последовательных временных шагов. Список соседей остается пригодным на протяжении обычно 10 – 20 временных шагов, даже для относительно небольших Δr. Тот факт, что список содержит пары атомов, лежащие вне радиуса взаимодействия, гарантирует, что на протяжении данной последовательности временных шагов не могут появиться новые взаимодействующие пары, которые не занесены в список соседей. Один из возможных критериев обновления списка может быть основан на наблюдении за максимальной скоростью атомов на каждом шаге, ожидая, пока не выполнится условие:

$$\sum_{\text{steps}} (\max_i |\mathbf{v}_i|) > \frac{\Delta r}{2\Delta t}, \tag{4}$$

после чего производится обновление списка. Важно отметить, что поскольку список обновляется не каждый шаг, то обычно использование списка соседей дает прирост производительности в несколько раз (для некоторых задач до 7) по сравнению с методом ячеек.

3.2 Реализация МД на GPU

Начиная с этого раздела и далее в документе будут употребляться два термина – *хост* (host), означающий центральный процессор (CPU) и его адресное пространство памяти, и *девайс* (device) – обозначает GPU и соответствующее адресное пространство памяти.

В литературе можно найти ряд работ с описанием реализации классической МД на GPU. В частности, в работах [14, 15, 16] представлены алгоритмы и результаты моделирования простых жидкостей, атомы которых взаимодействуют друг с другом посредством парного потенциала Леннарда-Джноса (ЛД). Полученные результаты во всех указанных работах показывают в среднем ускорение от 10 до нескольких десятков раз по сравнению с одним CPU (во всех работах использовались CPU приблизительно одинаковой производительности и GPU NVIDIA® GeForce™ 8800 GTX). Однако следует отметить особенности реализаций алгоритмов. В [14] расчет взаимодействий основывается на методе ячеек, и алгоритм полностью реализован на девайсе, без промежуточного копирования данных на хост. Формирование ячеек (называемое “binning”)

происходит на GPU с использованием методики двойного буффера. Авторы [14] сообщают о времени расчета, равном 20 ms для 100000 частиц, полученном на GPU GeForce 8800 GTX. В [15], несмотря на использование списка соседей, а не метода ячеек как в [14], и также выполнения кода на девайсе, получено самое низкое ускорение. Это связано с использованием метода всех пар, а не ячеек для формирования списка соседей, а также с неоптимальным использованием памяти (не учтены "соединенные запросы" к глобальной памяти, "coalesced reads"). В [16] представлены алгоритмы, так же как и в [15] основанные на использовании списка соседей, но в которых в отличие от [15] список соседей формируется с использованием ячеек, и учтены тонкости работы с памятью. Авторы отмечают, что их алгоритм работает приблизительно в 3 раза быстрее, чем в работе [14], давая менее 10 ms за один шаг для 100000 частиц на GeForce 8800 GTX. Однако существенным недостатком работы [16] является то, что не весь код выполняется на GPU. А именно, binning происходит на хосте, т. е. координаты атомов копируются с девайса на хост, а затем после выполнения binning обратно на девайс, что занимает дополнительное время, которое для больших систем может быть существенным. Авторы объясняют свой подход тем, что для реализации методики двойного буффера формирования ячеек, реализованной в [14], у них не было времени. А реализация альтернативного метода, который возможен, но не описан в их работе, требует использования атомных функций, которые не поддерживаются GPU серии 8800 GTX.

В SurfaceGrowth, исходя из рассмотренных результатов, выбрано использование списка соседей, и за основу взяты алгоритмы, описанные в [16]. Но поскольку разработка приложения производилась на GPU NVIDIA® GeForce™ GTX 260, поддерживающей атомные функции, то формирование ячеек (binning) производится на GPU с использованием специально разработанного алгоритма, основывающегося на атомных операциях (см. раздел 4.2.2). Это позволяет избежать копирования координат на хост и способствует ускорению работы приложения. В работе [16] также присутствует сортировка координат атомов, что позволяет ускорить алгоритм для больших систем. В данной реализации SurfaceGrowth сортировка отсутствует.

3.3 Описание системы

В данном разделе представлены особенности модели, лежащей в основе работы SurfaceGrowth.

3.3.1 Потенциалы взаимодействия

Как отмечалось выше, в классической МД ключевую роль для моделирования того или иного материала играет потенциальная энергия взаимодействия атомов V (см. уравнение (2)). Для металлов в SurfaceGrowth используется эмпирический многочастичный потенциал, основанный на методе внедренного атома (embedded atom method, EAM) и реализованный в работе [17]. Он разработан для моделирования сплавов и полностью выражен через аналитические функции в отличие от первых версий EAM, где для функции внедрения (embedding function) использовались кубические сплайны [13].

В EAM потенциальная энергия V кристалла может быть выражена следующим образом [17]:

$$V_{\text{eam}} = \frac{1}{2} \sum_{i,j,i \neq j} \phi_{ij}(r_{ij}) + \sum_{i} F_i(\rho_i), \tag{5}$$

где ϕ_{ij} представляет парную энергию между атомами i и j, разделенными расстоянием r_{ij}, а F_i означает энергию внедрения, необходимую, чтобы поместить атом i в локальную позицию с электронной плотностью ρ_i. ρ_i рассчитывается следующим образом:

$$\rho_i = \sum_{j,j \neq i} f_j(r_{ij}), \tag{6}$$

где $f_j(r_{ij})$ – электронная плотность в месте расположения атома i, происходящая от атома j на расстоянии r_{ij}.

Обобщенный парный потенциал имеет вид

$$\phi(r) = \frac{A\exp\left[-\alpha\left(\frac{r}{r_{\text{e}}} - 1\right)\right]}{1 + \left(\frac{r}{r_{\text{e}}} - \kappa\right)^{20}} - \frac{B\exp\left[-\beta\left(\frac{r}{r_{\text{e}}} - 1\right)\right]}{1 + \left(\frac{r}{r_{\text{e}}} - \lambda\right)^{20}}, \tag{7}$$

где r_e – равновесное расстояние между ближайшими соседями, A, B, α, β – подгоночные параметры и κ, λ – два дополнительных параметра для отсечки.

Функция электронной плотности имеет такую же форму, как и притягивающий член в парном потенциале с теми же значениями β и λ, т. е.

$$f(r) = \frac{f_{\text{e}}\exp\left[-\beta\left(\frac{r}{r_{\text{e}}} - 1\right)\right]}{1 + \left(\frac{r}{r_{\text{e}}} - \lambda\right)^{20}}. \tag{8}$$

Чтобы функция внедрения F работала для широкого диапазона значений электронной плотности, используются три уравнения для каждого из следующих интервалов: $\rho < \rho_{\text{n}}$, $\rho_{\text{n}} \leq \rho < \rho_{\text{o}}$ и $\rho_{\text{o}} \leq \rho$. Используя

$\rho_{\text{n}} = 0.85\rho_{\text{e}}$, где ρ_{e} – равновесная электронная плотность, можно быть уверенным, что все равновесные свойства могут быть получены в диапазоне электронных плотностей $\rho_{\text{n}} \leq \rho < \rho_{\text{o}}$. Для гладкого изменения энергии внедрения требуются уравнения, дающие одинаковые значения функции и ее производной на границах интервалов. Эти уравнения имеют вид:

$$F(\rho) = \sum_{i=0}^{3} F_{\text{n}i} \left(\frac{\rho}{\rho_{\text{n}}} - 1 \right)^i, \rho < \rho_{\text{n}}, \rho_{\text{n}} = 0.85\rho_{\text{e}}, \tag{9}$$

$$F(\rho) = \sum_{i=0}^{3} F_i \left(\frac{\rho}{\rho_{\text{e}}} - 1 \right)^i, \rho_{\text{n}} \leq \rho < \rho_{\text{o}}, \rho_{\text{o}} = 1.15\rho_{\text{e}}, \tag{10}$$

$$F(\rho) = F_{\text{e}} \left[1 - \ln\left(\frac{\rho}{\rho_{\text{e}}} \right) \right] \left(\frac{\rho}{\rho_{\text{e}}} \right)^{\eta}, \rho_{\text{o}} \leq \rho. \tag{11}$$

Значения параметров, входящих во все выписанные уравнения, для 16 металлов можно найти в работе [17], или для шести металлов, перечисленных в разделе 2, – в коде SurfaceGrowth. Величины плотностей и масс металлов в SurfaceGrowth взяты из книги [18], где также можно найти другие характеристики металлов.

Сила, действующая на металлический атом k со стороны других атомов металла, дается выражением:

$$\mathbf{f}_k = -\frac{\partial V_{\text{eam}}}{\partial \mathbf{r}_k} = -\sum_{j \neq k} \frac{\mathrm{d}\phi(r_{kj})}{\mathrm{d}r} \hat{\mathbf{r}}_{kj} - \sum_{i=1}^{N_{\text{m}}} \frac{\partial F_i}{\partial \rho_i} \frac{\partial \rho_i}{\partial \mathbf{r}_k}, \tag{12}$$

где N_{m} – полное число атомов металла, $\hat{\mathbf{r}}_{kj}$ – единичный вектор, направленный от атома j к атому k. Выражения для производных в (12) имеют вид:

$$\frac{\mathrm{d}\phi}{\mathrm{d}r} = -\frac{A\exp\left[-\alpha\left(\frac{r}{r_{\text{e}}} - 1\right)\right]}{1 + \left(\frac{r}{r_{\text{e}}} - \kappa\right)^{20}} \left[\alpha + \frac{20\left(\frac{r}{r_{\text{e}}} - \kappa\right)^{19}}{1 + \left(\frac{r}{r_{\text{e}}} - \kappa\right)^{20}} \right] \frac{1}{r_{\text{e}}} + \frac{B\exp\left[-\beta\left(\frac{r}{r_{\text{e}}} - 1\right)\right]}{1 + \left(\frac{r}{r_{\text{e}}} - \lambda\right)^{20}} \left[\beta + \frac{20\left(\frac{r}{r_{\text{e}}} - \lambda\right)^{19}}{1 + \left(\frac{r}{r_{\text{e}}} - \lambda\right)^{20}} \right] \frac{1}{r_{\text{e}}}, \tag{13}$$

$$\sum_{i=1}^{N_{\text{m}}} \frac{\partial F_i}{\partial \rho_i} \frac{\partial \rho_i}{\partial \mathbf{r}_k} = \sum_{i=1, i \neq k}^{N_{\text{m}}} \frac{\mathrm{d}f(r_{ki})}{\mathrm{d}r} \left(\frac{\partial F_i}{\partial \rho_i} + \frac{\partial F_k}{\partial \rho_k} \right) \hat{\mathbf{r}}_{ki}, \tag{14}$$

$$\frac{\mathrm{d}f}{\mathrm{d}r} = -\frac{f_{\text{e}}\exp\left[-\beta\left(\frac{r}{r_{\text{e}}} - 1\right)\right]}{1 + \left(\frac{r}{r_{\text{e}}} - \lambda\right)^{20}} \left[\beta + \frac{20\left(\frac{r}{r_{\text{e}}} - \lambda\right)^{19}}{1 + \left(\frac{r}{r_{\text{e}}} - \lambda\right)^{20}} \right] \frac{1}{r_{\text{e}}}, \tag{15}$$

$$\frac{\partial F}{\partial \rho} = \frac{1}{\rho_{\mathrm{n}}}\left[F_{\mathrm{n}1} + 2F_{\mathrm{n}2}\left(\frac{\rho}{\rho_{\mathrm{n}}} - 1\right) + 3F_{\mathrm{n}3}\left(\frac{\rho}{\rho_{\mathrm{n}}} - 1\right)^2\right], \rho < \rho_{\mathrm{n}}, \rho_{\mathrm{n}} = 0.85\rho_{\mathrm{e}}, \tag{16}$$

$$\frac{\partial F}{\partial \rho} = \frac{1}{\rho_{\mathrm{e}}}\left[F_1 + 2F_2\left(\frac{\rho}{\rho_{\mathrm{e}}} - 1\right) + 3F_3\left(\frac{\rho}{\rho_{\mathrm{e}}} - 1\right)^2\right], \rho_{\mathrm{n}} \leq \rho < \rho_{\mathrm{o}}, \rho_{\mathrm{o}} = 1.15\rho_{\mathrm{e}}, \tag{17}$$

$$\frac{\partial F}{\partial \rho} = -\frac{F_{\mathrm{e}}\eta}{\rho_{\mathrm{e}}}\left(\frac{\rho}{\rho_{\mathrm{e}}}\right)^{\eta-1} \ln\left(\frac{\rho}{\rho_{\mathrm{e}}}\right)^{\eta}, \rho_{\mathrm{o}} \leq \rho. \tag{18}$$

Поскольку выражения для потенциальной энергии металла (5) и силы, действующей на атом металла (12), состоят из двух частей – парной, зависящей от относительного положения атомов, и EAM части, зависящей от электронных плотностей всех атомов, отличных от данного, то расчет V_{eam} и силы производится в два этапа. Вначале рассчитывается электронная плотность для каждого атома, а затем – указанные величины.

Потенциальная энергия V_C взаимодействия атомов углерода в слое графена описывается пружинным потенциалом из работы [19] следующей формы:

$$V_C = \frac{1}{2}\sum_{i-j}\mu_r(r_{ij} - r_0)^2 + \frac{1}{2}\sum_{i-j-k}\mu_\theta r_0^2(\theta_{ijk} - \theta_0)^2 + \frac{1}{2}\sum_{i-(j,k,l)}\mu_p\left(\delta z_i - \frac{\delta z_j + \delta z_k + \delta z_l}{3}\right)^2. \tag{19}$$

Суммирование производится по ближайшим соседям, их парам и триплетам, а значения параметров можно найти в [19]. Здесь только выпишем выражение для силы, действующей на атом i углерода со стороны других атомов углерода в слое:

$$\mathbf{f}_i = -\sum_{i-j}\mu_r(r_{ij} - r_0)\frac{\mathbf{r}_{ij}}{r_{ij}} + \sum_{j\neq i, k\neq i}\mu_\theta r_0^2(\theta_{jik} - \theta_0)\left[1 - \left(\frac{\mathbf{r}_{ji}\mathbf{r}_{ki}}{r_{ji}r_{ki}}\right)^2\right]^{-\frac{1}{2}} \times \frac{\left(1 - \frac{r_{ji}}{r_{ki}}\cos\theta_{jik}\right)\mathbf{r}_{ik} + \left(1 - \frac{r_{ki}}{r_{ji}}\cos\theta_{jik}\right)\mathbf{r}_{ij}}{r_{ji}r_{ki}} +$$

$$\sum_{j\neq i,k\neq i} \mu_\theta r_0^2(\theta_{ijk}-\theta_0)\left[1-\left(\frac{\mathbf{r}_{ij}\mathbf{r}_{kj}}{r_{ij}r_{kj}}\right)^2\right]^{-\frac{1}{2}}\times \left[\frac{\mathbf{r}_{kj}}{r_{ij}r_{kj}}-\frac{(\mathbf{r}_{ij}\mathbf{r}_{kj})}{r_{ij}^3 r_{kj}}\mathbf{r}_{ij}\right]- \sum_{j,k,l}\frac{2}{3}\mu_p\left[2\delta z_i-(\delta z_j+\delta z_k+\delta z_l)\right]-\frac{1}{9}\mu_p\sum_{m,n}(\delta z_m+\delta z_n)\,. \quad (20)$$

В последней сумме индексы m, n означают соседей атома i, следующих за ближайшими. Таким образом, при расчете силы из пружинного потенциала нужно рассматривать не только трех ближайших соседа в слое графена, а и соседей ближайших соседей, что учтено в коде программы.

Взаимодействие металл-углерод описывается потенциалом Леннарда-Джонса:

$$V_{LJ}=\begin{cases} 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12}-\left(\frac{\sigma}{r}\right)^{6}\right], & r<r_{\mathrm{c}} \\ 0, & r\geq r_{\mathrm{c}} \end{cases}, \quad (21)$$

где по умолчанию значения параметров ε и σ выбраны как в работе [19], однако интерфейс приложения позволяет задавать эти величины. Расстояние отсечки равно $r_c = 2.5\sigma$. Отметим, что для EAM потенциала используется расстояние отсечки $r_c = 1.45a$, a – постоянная решетки металла.

3.3.2 Безразмерные единицы

В SurfaceGrowth используются следующие единицы измерения:

массы: масса углерода $m_0 = 19.9441 \cdot 10^{-27}$ kg;

длины: длина ковалентной связи в графене $a_0 = 1.42$ Å;

времени $t_0 = 0.2$ ps;

энергии: $\varepsilon_0 = \frac{m_0 a^2}{t_0^2} = 6.275049 \cdot 10^{-2}$ eV;

температуры: 1 K безразмерный равен 0.001373;

силы: 1 nN безразмерный равен 14.123984.

3.3.3 Контроль температуры

Обычная МД отличается от большинства экспериментальных исследований тем, что в моделировании фиксированными являются полная энергия E и объем V, а не температура T и давление P. В терминах стати-

стической механики обычная МД дает средние по микроканоническому ансамблю NVE (N – количество молекул), в то время как эксперименты с постоянной температурой соответствуют каноническому ансамблю NVT.

Для проведения моделирований, близких к экспериментам, необходимо использовать канонический ансамбль, и, следовательно, поддерживать постоянную температуру. Поскольку внешними силами над системой выполняется работа, то поддержание постоянной температуры означает выбор способа отвода излишка тепла или соединение системы с термостатом. В SurfaceGrowth используется термостат Берендсена [20], который хотя и не дает траектории истинного канонического ансамбля, однако прост в реализации.

Термостат Берендсена предполагает, что температура T системы экспоненциально приближается к требуемой температуре T_0 с временем релаксации τ_T, т. е.

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{T_0 - T}{\tau_T}. \tag{22}$$

В дискретной форме (22) имеет вид:

$$\frac{T^{n+1} - T^n}{h} = \frac{T^n}{\tau_T}\left(\frac{T_0}{T^n} - 1\right), \tag{23}$$

где h – временной шаг, T^n и T^{n+1} – температура на текущем и на следующем временном шаге соответственно. Связь между T и кинетической энергией системы имеет вид:

$$T = \frac{2}{3Nk_\mathrm{B}} \sum_{i=1}^{N} \frac{m_i v_i^2}{2}, \tag{24}$$

здесь N – число частиц, k_B – постоянная Больцмана, m_i и v_i – соответственно масса и скорость частицы. Обозначая скорость частицы на текущем и на следующем шаге через $\mathbf{v}_i^n$ и $\mathbf{v}_i^{n+1}$, и подставляя уравнение (24) в (23), получаем (предполагая частицы одного сорта):

$$\frac{\sum_{i=1}^{N} (\mathbf{v}_i^{n+1})^2 - \sum_{i=1}^{N} (\mathbf{v}_i^n)^2}{h} = \sum_{i=1}^{N} \frac{(\mathbf{v}_i^n)^2}{\tau_T}\left(\frac{T_0}{T^n} - 1\right). \tag{25}$$

Для частицы i, можно записать:

$$(\mathbf{v}_i^{n+1})^2 = (\mathbf{v}_i^n)^2 \left[1 + \frac{h}{\tau_T}\left(\frac{T_0}{T^n} - 1\right)\right]. \tag{26}$$

Таким образом, реализация термостата Берендсена сводится к перемасштабированию скоростей, т. е. $v_i^{n+1} = \beta v_i^n$. Однако множитель β не является постоянным, а зависит от текущего значения температуры:

$$\beta = \sqrt{1 + \gamma(\frac{T_0}{T^n} - 1)}, \tag{27}$$

где $\gamma = \frac{h}{\tau_T} \in [0; 1]$. Если текущая температура T^n выше требуемой T_0, то $\beta < 1$ и v_i^{n+1} уменьшается. А если $T^n < T_0$, то $\beta > 1$ и v_i^{n+1} увеличивается. Отметим, что γ характеризует скорость диссипации тепла. Чем ближе γ к 1, т. е. чем меньше τ_T, тем быстрее тепло диссипируется и, следовательно, сильнее связь системы с тепловой ванной. В SurfaceGrowth выбрано значение $\gamma = 0.005$, соответствующее относительно слабой связи с термостатом.

Стоит отметить, что масштабирование скоростей может повлиять на распределение энергии в системе. Чтобы уменьшить это влияние в программе используется реализация термостата Берендсена, описанная в работе [13], где скорости перемножаются не каждый временной шаг, а каждые k шагов, а в промежуточные шаги система интегрируется без масштабирования. В SurfaceGrowth $k = 25$, чтобы установилось "равновесие" между потенциальной и кинетической энергиями системы.

Уравнения движения интегрируются методом Верле (leapfrog method), описанным в [12]. Величину временного шага можно выбирать, она указывается в безразмерном виде.

3.3.4 Режимы

Как отмечалось в разделе 2, в SurfaceGrowth моделирование проводится в одном из трех режимов: `Bulk`, `Surface Growth`, `Shear`. Они отличаются друг от друга составом атомов, их начальным расположением, ходом расчетов и возможностями измерений. Отметим, что всем трем режимам присущи общие и специфичные параметры, необходимые для проведения моделирования и которые задает пользователь, (данный вопрос рассматривается в разделе 5.2). Также возможно измерять как параметры, присутствующие во всех режимах (и называемые "общими параметрами"), так и специфичные для каждого из режимов (см. раздел 3.3.5). Рассмотрим особенности каждого из режимов.

Режим `Bulk`. Данный режим предназначен для моделирования объемных свойств металла. Используются только атомы металла (без графена), они располагаются в вершинах ГЦК решетки, по всем координат-

ным направлениям (x, y, z) приложены периодические граничные условия. Термостат отсутствует, т. е. поддерживаются условия микроканонического статистического ансамбля с постоянными N (число атомов), V (объем образца), E (полная энергия системы). В режиме `Bulk` моделирование протекает в равновесных условиях, и можно производить измерение общих параметров и функции радиального распределения (radial distribution function, RDF), подробности измерений даны в следующем разделе 3.3.5. Назначение режима `Bulk` – проверка правильности общего алгоритма МД и расчета EAM потенциала – энергии связи и атомных конфигураций, а также правильности работы reduction. Это обусловлено тем, что в условиях данного режима полная энергия должна быть постоянной, полный импульс должен быть равен 0, и эти значения должны сохраняться по ходу вычислений.

Режим `Surface Growth`. Первоначально планировалось, что программа SurfaceGrowth будет полностью воспроизводить эксперименты по исследованию трения металлических наночастиц, в которых наночастицы получаются напылением атомов металла на поверхность ВОПГ, а затем сдвигаются с помощью атомного силового микроскопа (АСМ) [5]. Однако по ряду причин на текущем этапе разработки программы данная цель не реализована, и введены два режима `Surface Growth` и `Shear`. О `Shear` будет рассказано далее. В режиме же `Surface Growth` можно исследовать напыление атомов выбранного металла на слой графена в зависимости от энергии напыления атомов металла, от температуры, плотности потока напыляемых атомов и от параметров взаимодействия металл-углерод.

Алгоритм протекания расчетов в режиме `Surface Growth` состоит в следующем. Задаются размеры графенового слоя и число атомов металла. Сначала в области моделирования присутствует только слой графена, расположенный параллельно плоскости xy, а атомы металла находятся вне области моделирования и не взаимодействуют между собой. Периодические границы приложены в плоскости xy. Крайние атомы по периметру в слое графена жестко закреплены, чтобы слой не смещался в вертикальном направлении, параллельном оси z. После периода уравновешивания (equilibration), длительность которого задается пользователем, и во время которого достигается заданное пользователем значение температуры слоя графена, через заданное пользователем число временных шагов происходит введение атомов металла в область моделирования над слоем графена группами, размер которых также задается пользователем. Введенные атомы имеют случайные координаты в плос-

кости xy, их координата z определяется размерами области моделирования, описанными в разделе 4.2.2. Компоненты x, y начальных скоростей внедренных атомов металла равны 0, а z компонента направлена к слою графена (против оси z) и ее значение определяется выражением:

$$v_{0z} = \sqrt{\frac{2\varepsilon_{\text{dep}}}{m}}, \tag{28}$$

где ε_{dep} – энергия напыляемых атомов, задаваемая пользователем, m – масса атома металла.

В режиме `Surface Growth` можно производить измерение общих параметров. Отметим, что термостат Берендсена, используемый для отвода тепла, прикладывается только к атомам углерода, и слишком интенсивная бомбардировка слоя графена приведет к его перегреву и разрушению. Поэтому данный режим требует очень тщательного подбора значений параметров, поскольку высокие значения энергии атомов или плотности потока приводят к разрушению слоя графена.

Режим `Shear`. В данном режиме происходит исследование трения наночастиц. Пользователь задает размеры слоя графена и число атомов металла. Как и в предыдущем режиме слой графена лежит в плоскости xy, в которой приложены периодические границы, и крайние атомы по периметру в слое графена жестко закреплены, чтобы слой не смещался в вертикальном направлении, параллельном оси z. Атомы металла размещаются над слоем графена в вершинах идеальной ГЦК решетки, особенности построения начальной конфигурации для металла в данном режиме представлены в разделе 4.2.2. Начальные скорости атомов металла равны 0. На протяжении периода уравновешивания (equilibration), задаваемого пользователем, система эволюционирует без отвода тепла, т. е. термостат не прикладывается ни к атомам металла, ни к атомам углерода. На этом этапе атомы перестраиваются в конфигурацию, соответствующую минимуму потенциальной энергии, при этом происходит нагрев системы вследствие выделения энергии связи. Если взаимодействие металл-графен много слабее, чем металл-металл, то конфигурация с минимальной энергией имеет форму шара. Однако поскольку площадь контакта сферических наночастиц с графеном довольно мала, и в экспериментах наночастицы не сферические, то желательно остановить формирование сферы в подходящий момент времени. В SurfaceGrowth эта остановка производится путем приложения термостата Берендсена как к металлу, так и к графену после времени уравновешивания. Термостат прикладывается на протяжении времени, заданного пользователем,

а затем тепло отводится только от графена до конца моделирования. Сразу после выключения термостата к атомам наночастицы, расположенным левее ее центра масс (по оси x), начинает прикладываться сила вдоль направления x, имитируя сдвиг с помощью АСМ, имеющий место в экспериментах. Данная сила увеличивается каждый временной шаг на величину, заданную пользователем, пока скорость центра масс наночастицы не достигнет значения, равного 0.005 (в безразмерных единицах). После этого к атомам уже прикладывается постоянная сила, и частица сдвигается под действием практически постоянной силы.

В режиме `Shear` можно производить измерение общих параметров, а также различных характеристик наночастицы: скорости и положения центра масс, сил трения и сдвига, размеров, а также структуры, исходя из RDF. Стоит отметить, что в данном режиме важным является выбор длительности уравновешивания, определяющий форму и структуру наночастицы, и, соответственно, площадь контакта. Так, слишком короткий период приведет к наночастице с кристаллической структурой и прямоугольной формы. Также отметим, что эти периоды зависят от типа металла, поскольку для тяжелых металлов, например свинца, атомы перестраиваются намного медленнее легких (например, алюминия).

3.3.5 Измерения

Как указывалось в предыдущих разделах, в приложении можно производить измерения различных параметров системы: общих параметров и характеристик наночастицы. Если пользователь сделал соответствующий выбор, то общие параметры выводятся в текстовый файл через указанное пользователем число временных шагов. Это число также определяет интервал, по которому производится усреденение величин. Используя текстовый файл с данными, можно строить графики зависимостей величин с помощью различных плоттеров (Grapher, Origin, Sigma Plot и т. д.). К общим параметрам относятся (приводимые ниже названия соответствуют заголовкам столбцов в файле с результатами):

- stepCnt – количество временных шагов, прошедших с начала моделирования;
- impulse – полный импульс системы;
- totEn(eV), totEn.rms(eV) – полная энергия системы и ее среднеквадратичное отклонение (в электрон-вольтах) за период усреднения. Полная энергия равна сумме потенциальной и кинетической энергий всех атомов;

- potEn(eV), potEn.rms(eV) – потенциальная энергия системы и ее среднеквадратичное отклонение (в электрон-вольтах). Равна сумме потенциальных энергий всех атомов;

- Tempr(K), T.rms(K) – температура системы и ее среднеквадратичное отклонение (в градусах Кельвина) за период усреднения. Рассчитывается, исходя из кинетической энергии атомов с использованием уравнения (24). Также в режиме `Shear` при расчете суммы квадратов скоростей вычитается скорость центра масс из скорости каждого атома металла, чтобы исключить этот вклад в температуру;

- oneStep(ms) – средняя длительность расчета одного временного шага в миллисекундах.

Большая часть характеристик наночастицы выводится в тот же текстовый файл, что и общие параметры. К характеристикам относятся:

- Veloc_CM – скорость центра масс (ЦМ) наночастицы вдоль направления x (в безразмерных единицах);

- CM(angstr) – координата ЦМ наночастицы вдоль направления x (в ангстремах). Начало координат находится в центре области моделирования;

- friction(nN) – сила трения, действующая на частицу (в наноньютонах). Она равна x компоненте силы, действующей на атомы металла со стороны атомов углерода;

- sizex(angstr), sizey(angstr), sizez(angstr) – размеры частицы вдоль направлений x, y, z (в ангстремах). Определяются как разности максимального и минимального значений координат атомов вдоль соответствующего направления;

- shearForce(nN) – полная сила сдвига, действующая на наночастицу (в наноньютонах). Равна произведению значения силы сдвига, действующей на один атом, на число атомов, лежащих левее ЦМ наночастицы.

Измерения характеристик наночастицы производится во всех режимах, если задано отличное от нуля число атомов металла. При нулевом числе атомов металла эти характеристики приравниваются нулю. Однако нужно понимать, что полный смысл указанные характеристики имеют только лишь в режиме `Shear`. В режиме `Bulk` вся система

фактически отвечает наночастице, параметры Veloc_CM, CM(angstr) дают еще одну возможность проверить правильность работы reduction, поскольку их значения должны быть близки к нулю, а sizex(angstr), sizey(angstr), sizez(angstr) показывают размеры системы. В режиме же `Surface Growth` в большинстве случаев характеристики наночатицы не несут смысла. Только если в результате напыления образуется сплошной слой, то только после этого можно использовать указанные величины для его характеристики.

Кроме указанных параметров наночастицы также можно наблюдать за ее структурой путем измерения функции радиального распределения $g(r)$ (radial distribution function, RDF). RDF показывает вероятность найти соседний атом на определенном расстоянии относительно данного атома (подробности описаны в [12], на с. 90). В SurfaceGrowth пользователь задает шаг, через который производится измерение $g(r)$ по специальному алгоритму. После проведения 100 таких измерений полученные результаты усредняются и выводятся в отдельный файл в виде гистограммы, показывающей частоту попадания атома в один из 200 интервалов, на которые разбивается расстояние величиной приблизительно в 7.5 Å. Также стоит отметить, что в режимах `Bulk` и `Shear` RDF является полноценной характеристикой. Однако в режиме `Surface Growth` она в большинстве случаев не является точной характеристикой структуры металла.

Кроме указанных характеристик программа SurfaceGrowth позволяет периодически (интервал задается пользователем) сохранять координаты всех атомов в формате Protein Data Bank (PDB), в файл с расширением .pdb с целью дальнейшей визуализации системы. Формат PDB поддерживается программами, специализирующимися на визуализации молекул. В частности для визуализации системы на основе .pdb файлов, полученных с помощью SurfaceGrowth, рекомендуется использовать программу Visual Molecular Dynamics (VMD) [8], доступной для свободного использования и также предусматривающей возможность использования технологии NVIDIA® CUDA™ (подробности даны в разделе 5).

4 Структура

Структуру программы рассмотрим с точек зрения

- особенностей интерфейса и функциональных возможностей;
- структуры кода.

4.1 Особенности интерфейса и функциональных возможностей

SurfaceGrowth – оконное приложение, рассчитанное на работу в операционной системе Microsoft Windows XP и более старших версиях Windows. Оно имеет простой интерфейс, реализованный в виде диалогового окна с использованием Windows API. Интерфейс позволяет выбирать режим моделирования (`Bulk`, `Surface Growth`, `Shear`), металл, и вводить соответствующие параметры. Кроме того можно сохранять результаты измерений в файлы и выбирать возможность вывода изображения системы в OpenGL (OGL) окне во время расчетов. Важно отметить, что при выводе изображения системы в OGL окне расчеты замедляются в несколько раз (от 3 до 8). Это связано с тем, что список соседей формируется каждый временной шаг, а не раз в 10 – 20 шагов, как в режиме без OGL окна.

В режиме без OGL окна есть возможность создавать резервные копии системы (back up), которые при непредвиденном прерывании расчетов дают шанс начать вычисления с места, наиболее близкого к моменту прерывания. С этой целью в корневом каталоге на диске C поочередно каждые 1000 временных шагов создаются 2 файла bckup0.sugr, bckup1.sugr. Наличие двух файлов повышает вероятность запуска с резервной копии, поскольку учитывает факт, что прерывание программы может произойти во время записи в файл. Отметим, что back up еще находится на стадии тестирования, и запуск с back up после прерывания счета не всегда работает, особенно для больших систем.

Научная направленность приложения обусловила отсутствие стремления предоставить удобный интерфейс. Основное внимание при разработке программы было сфокусировано на функционировании расчетов. В режиме с OGL окном есть возможность паузы вычислений и обычного выхода из приложения, а также за ходом вычислений можно следить по числу шагов, выводимых в заголовке окна. Но режим OGL не дает возможность моделировать системы больших размеров, и его следует использовать как демонстрационный режим. Поэтому для серьезных вычислений следует использовать режим без OGL. Однако в этом режи-

ме отсутствует возможность паузы и безаварийного выхода из приложения, и за вычислениями можно следить только по файлам результатов, что представляет некоторые неудобства для пользователя. Режим паузы можно реализовать, выбрав сохранение back up файлов, затем выключения приложения, и начала вычислений после прерывания с back up. Однако, как отмечалось, запуск с back up не всегда работает и им желательно пользоваться только в крайних случаях.

Подробные инструкции по работе с интерфейсом даны в разделе 5.

4.2 Структура кода

В данном разделе представлена организация решения и основные функции в коде.

4.2.1 Решение

Приложение SurfaceGrowth реализовано в Microsoft Visual Studio 2008 с использованием CUDA Toolkit 2.3. Решение (solution) SurfaceGrowth содержит один проект SurfaceGrowth, для его работы также необходимо установить CUDA SDK, т. к. используется часть заголовочных файлов из SDK. Проект содержит файлы:

- три C++ файла: render_particles.cpp, shaders.cpp, SurfaceGrowth.cpp. Первые два файла взяты из примера "particles" из CUDA SDK и содержат код, отвечающий за отображение системы в OpenGL окне. Файл SurfaceGrowth.cpp содержит код, отвечающий за пользовательский интерфейс и вызов функций-оберток (wrappers);

- один CUDA файл: SurfaceGrowth.cu, содержит реализацию функций-оберток (wrappers) и ядер, и несколько функций, вызываемых на хосте;

- шесть загловочных файлов: render_particles.h, shaders.h – содержат объявления, необходимые для вывода системы в OpenGL окне; resource.h – объявления ресурсов; SurfaceGrowthProto.h – объявлены функции, вызываемые на хосте в SurfaceGrowth.cpp, и определен массив структур, содержащий параметры EAM потенциала для шести металлов; SurfaceGrowth.h – содержит необходимые заголовочные файлы, объявление структуры `SimParams` и прототипы функций-оберток (wrappers); ComputeDefs.h – содержит макросы, используемые при вычислениях. Большая часть этих макросов взята из кода, сопровождающего книгу [12];

- один файл ресурсов: SurfaceGrowth.rc – содержит описание диалогового окна.

4.2.2 Код

Данный раздел содержит только основные вопросы. Его назначение – помочь разобраться в коде, а не представить исчерпывающее описание.

Сразу стоит отметить, что одну из ключевых ролей играет структура `SimParams`, объявленная в SurfaceGrowth.h. Она содержит около 100 параметров, необходимых для работы приложения, обеспечивая связь между пользовательским интерфейсом, хостом и девайсом благодаря наличию глобального хост-экземпляра `g_hSimParams`, определенного в SurfaceGrowth.cpp, и глобального девайс-экземпляра `dparams`, объявленного в SurfaceGrowth.cu и находящегося в constant памяти GPU. Экземпляр `dparams` вызывается практически в каждом ядре, поэтому для ускорения доступа к его данным он помещен в constant память. Также именно благодаря `SimParams` есть возможность восстановления вычислений. В заголовочном файле достаточно комментариев, чтобы разобраться с назначением каждого параметра, поэтому здесь не приводится эта информация.

Стоит отметить, что размеры области моделирования, хранящиеся в переменной `g_hSimParams.region` и вычисляемые на хосте в `SetParams()`, определяются режимом моделирования. Так, в режиме Bulk размеры области моделирования равны произведению числу элементарных ячеек `g_hSimParams.initUcell.x(y,z)` вдоль данного направления, которое задается пользователем, на размер элементарной ячейки металла. Этот размер определяется как длина ребра куба, объем которого равен $m/(4\rho)$, где m – масса атома металла, ρ – плотность металла, а четверка учитывает, что в элементарной ячейке ГЦК решетки находятся 4 атома. В режимах `Surface Growth` и `Shear` размеры области моделирования вдоль осей x и y определяются одинаково в зависимости от размеров слоя графена. Эти размеры равны

$$\texttt{g\_hSimParams.region.x} = 8a_C \cos(\pi/6) \cdot \texttt{g\_hSimParams.initUcell.x},$$
$$\texttt{g\_hSimParams.region.y} = 6a_C \cdot \texttt{g\_hSimParams.initUcell.y}, \quad (29)$$

где $a_C = 1.42$ Å – длина ковалентной связи в графене. Вертикальная же длина области моделирования `g_hSimParams.region.z` зависит от режима и от числа атомов металла. Если металла нет, то размер равен трем длинам ячеек, используемых в binning при построении списка соседей. Если металл есть, то в обоих режимах расстояние под слоем графена

определяется переменной `g_hSimParams.cellShiftZ`, определяющей число ячеек под слоем графена. В режиме `Surface Growth` расстояние над слоем графена равно 1.8 умножить на длину ребра куба, который можно составить из данного числа атомов металла, упаковав их в ГЦК решетку (плюс длина постоянной решетки). В режиме `Shear` расстояние над слоем графена равно указанному числу без множителя 1.8. Общее значение `g_hSimParams.region.z` равно сумме расстояний под и над слоем графена.

Также важную роль играют массивы для координат на хосте – `g_hr`, и девайсе – `g_dr`. Определение сорта атома (металл или углерод) производится по индексу атома в указанных массивах. Атомы, занимающие позиции от 0 до `g_hSimParams.nMolMe-1` – атомы металла, а индексы от `g_hSimParams.nMolMe` до `g_hSimParams.nMol-1` – атомы углерода. Нужно заметить, что в отличие от скоростей и ускорений имеющих тип `float3`, координаты имеют тип `float4`. Это сделано преднамеренно, т. к. в `w`-компоненте координат сохраняется потенциальная энергия взаимодействия, которая рассчитывается при вычислении сил.

Хост код. Для устранения сложности работы и отладки кода, отвечающего за интерфейс, в SurfaceGrowth.cpp отсутствует явный вызов CUDA функций. Они вызываются неявно из функций-оберток (wrappers), имена которых заканчиваются на W. Они объявлены в SurfaceGrowth.h и реализованы в SurfaceGrowth.cu. Большая часть оберток отвечает за инициализацию CUDA или OpenGL interoperability, а также за работу с буферами. Во всех обертках есть обработка исключений путем анализа значения, возвращаемого функцией `cudaGetLastError()`. Если возникло исключение, то запрашивается его описание, пользователю выводится сообщение об ошибке с описанием, и приложение закрывается. Однако поскольку исключения перехватываются не после каждого ядра, то сообщения могут не отражать истинную причину ошибки. Также отметим отсутствие обработки ошибок, связанных с работой файловых функций. Рассмотрим следующие функции-обертки:

- `SetParametersW` – копирует значения из `g_hSimParams` в `dparams`;
- `InitCoordsW` – в зависимости от режима инициализирует координаты;
- `DoComputationsGLW` – вызывает все ядра, необходимые для продвижения системы на один шаг в режиме отображения системы в OpenGL окне;

- `DoComputationsW` – вызывает все ядра, необходимые для продвижения системы на один шаг в режиме без отображения системы в OpenGL окне.

Как отмечалось выше, SurfaceGrowthProto.h содержит объявления функций, вызываемых на хосте. Большинство из этих функций производят инициализацию параметров и структуры `g_hSimParams`. Рассмотрим некоторые из функций:

- `EamInit()` – вызывается в `SetParams()`, производит инициализацию EAM параметров, а именно копирует значения в структуру `g_hSimParams`;
- `GrapheneInit()` – вызывается в `SetParams()`, также инициализирует параметры, используемые при расчете сил в слое графена;
- `SetParams()` – инициализирует большинство параметров, входящих, в `g_hSimParams`;
- `SetupJob()` – выделяет память на хосте для координат, скоростей и ускорений, инициализирует координаты, скорости и ускорения, определяет имя файла для результатов и открывает этот файл.

На хосте также вызываются стандартные функции WinAPI, работающие с пользовательским интерфейсом, и функции OpenGL. Они не описываются в данном документе.

Ядра. Ядра (kernels) вызываются из оберток для параллельного выполнения некоторой задачи. Имена ядер заканчиваются на K, и для них нет отдельного объявления прототипов. Ядра, реализованные в приложении, можно разделить на три группы:

- работающие с координатами и с интегрированием уравнений движения;
- использующиеся для расчета сил;
- выполняющие измерение характеристик системы.

Практически для всех функций первых двух групп (кроме `InitFcc CoordsK(float4 *pos)`, `InitSlabCoordsK(float4 *pos)`, `InitGraphene`

`CoordsK(float4 *pos)`) количество блоков и потоков определяется принципами построения ячеек для формирования списка соседей, описанными в работе [16]. А именно, каждый атом обрабатывается отдельным потоком, поэтому число потоков, *выполняющих полезные вычисления*, равно полному числу атомов `g_hSimParams.nMol`. Число блоков (размер сетки блоков) определяется числом ячеек `g_hSimParams.cells`, используемых при построении списка соседей (см. раздел 3.1), которое определяется размером области моделирования `g_hSimParams.region` и расстоянием отсечки выбранного металла `g_hSimParams.rCutEam` (см. функцию `SetParams()`). Поскольку заранее неизвестно, сколько атомов будет находиться в ячейке, то пришлось ввести константу `BLOCK_SIZE` (она определена в SurfaceGrowth.h), определяющую количество потоков в блоке и, соответственно, максимальное число атомов в ячейке. Выбранное значение 64 учитывает все 6 металлов. *Полное* число потоков равно произведению `BLOCK_SIZE` на количество блоков, и это число может значительно превышать число потоков, выполняющих полезные вычисления. Это связано с тем, что максимальное число атомов зависит от размера ячеек, определяемого постоянной решетки металла a, и для некоторых металлов с малым a (и, соответственно, с малыми ячейками), например никеля, можно было бы взять `BLOCK_SIZE` равным 48. Но т. к. приходится учитывать другие металлы с большими a, например свинец, то пришлось подобрать значение, оптимальное для всех 6 металлов. Но при `BLOCK_SIZE`=64 для никеля много потоков будут работать с "виртуальными" атомами. Данный подход к построению блоков накладывает ограничения на размеры системы, для которых количество блоков не должно превышать их максимального числа для используемой видеокарты.

Для третьей группы ядер количество блоков определяется полным числом атомов, размер блока одинаков для всех функций и равен 512 потоков.

Рассмотрим ядра первой группы – работающие с координатами или с уравнениями движения:

- `InitFccCoordsK(float4 *pos)` – вызывается из `InitCoordsW` в режиме `Bulk`, размещает атомы металла в вершинах ГЦК (face centered cubic, FCC) решетки. Металл разбивается на полоски, состоящие в плоскости xy из одной элементарной ячейки, а вдоль направления z – из `g_hSimParams.initUcell.z` элементарных ячеек, это число задается пользователем. Каждый поток в блоке генерирует координаты для каждого из четырех атомов, расположенных в одной ячейке из указанной полоски элементарных ячеек;

- `InitSlabCoordsK(float4 *pos)` – вызывается из `InitCoordsW` в ре-

жиме `Shear`, задает начальные координаты атомов металла – располагает атомы металла в вершинах ГЦК решетки над слоем графена. Здесь используются двумерные индексы для блоков, их количество равно числу элементарных ячеек металла вдоль x и y направлений (это число специально рассчитывается в обертке). Количество потоков в блоке равно числу слоев. Как и в предыдущем случае, каждый поток генерирует координаты для четырех атомов в одной элементарной ячейке решетки;

- `InitGrapheneCoordsK(float4 *pos)` – вызывается из `InitCoordsW` в режимах `Surface Growth` и `Shear`, дает начальные координаты слоя графена. Каждый блок содержит 32 потока, каждый поток генерирует координаты для одного атома, количество блоков равно числу атомов углерода, разделенному на 32;

- `LeapfrogStepK(int part, float4 *dr, float3 *dv, float3 *da)` – интегрируются уравнения движения методом Верле. Каждый поток вычисляет данные для одного атома. Стоит отметить, что атомы, расположенные по периметру слоя графена, пропускаются, чтобы обеспечить их неподвижность (жесткость). Для этого сравниваются координаты атома со специально рассчитанными значениями;

- `ApplyBoundaryCondK(float4 *dr)` – вызывается при вычислении одного шага, прикладывает периодические граничные условия к каждому атому;

- `ApplyBerendsenThermostat(float3 *dv, real *vvSum, int step Count)` – прикладывает термостат Берендсена, умножая скорость атома на множитель из уравнения (27) из раздела 3.3.3. В самом ядре производится проверка типа атома, это приводит к divergent warps. Отметим, что обычно функция, отвечающая за термостат, располагается сразу непосредственно за функцией отвечающей за расчет сил (в нашем случае это `ComputeForcesK`). Однако по техническим причинам в SurfaceGrowth термостат прикладывается после расчета суммы квадратов скоростей. Но такая архитектура принципиально не влияет на результаты моделирований;

- `InsertAtomsK(float4 *dr, float3 *dv, int nMolDeposited, int nMolToDeposit)` – вызывается в режиме `Surface Growth`, предназначена для вставки атомов, имитирующей напыление. В функции происходит сравнение индекса атома с количеством напыленных атомов и с количеством атомов, которые должны быть напылены

в данный момент. Если индекс атома попадает в этот интервал, то его вертикальной координате присваивается значение `0.49f * dparams.region.z`, и задается начальная скорость, определяемая уравнением (28). Если же индекс атома не удовлетворяет указанным условиям, то он остается вне области моделирования с нулевой скоростью;

- `ApplyShearK(float4 *dr, float3 *da, real shear, real centerOfMassX, uint *numOfSharedMols)` – вызывается в режиме `Shear` после периодов уравновешивания и охлаждения. Это ядро прикладывает силу сдвига к атому в направлении x путем добавления к x компоненте ускорения соответствующего значения. Эта функция выбирает нужные атомы, лежащие левее x координаты центра масс наночастицы. Также в функции производится расчет числа таких атомов, которое используется для расчета полной силы сдвига, действующей на частицу;

- `SetColorK(float4 *color)` – эта функция не относится к вычислениям, она задает цвета атомов, которые будут выводиться в OpenGL окне.

Рассмотрим ядра второй группы – использующиеся при расчете сил:

- `BinAtomsIntoCellsK(float4 *dr, int *CELL, uint *molsInCells)` – это ядро, используя специально разработанный для GPU алгоритм, производит binning, т. е. определение, в какой ячейке находится атом. Как указывалось, число блоков равно числу ячеек, и индекс блока соответствует номеру ячейки. В отличие от других ядер, в которых каждый поток на протяжении своей работы отвечает только за один атом, в данном ядре каждый поток последовательно работает с несколькими атомами. А именно, каждый блок последовательно, каждую итерацию, определяемую переменной `count`, считывает по `BLOCK_SIZE` атомов. Каждый поток определяет индекс ячейки, в которой находится один из `BLOCK_SIZE` атомов и сравнивает с индексом своего блока, т. е. с индексом своей ячейки. Если эти индексы совпадают, то поток, используя атомную функцию `atomicAdd`, узнает свободное место в массиве `CELL` и записывает в него индекс атома. Таким образом, в конце концов формируется массив `CELL`, содержащий индексы атомов, находящихся в каждой из ячеек. Структура данного массива описана в [16];

- `BuildNebrListK(float4 *dr, int *CELL, int *NN, int *NBL)` – производит построение списка соседей с использованием Алгоритма 4 из работы [16];
- `EamComputeRhoK(real *rho, float4 *dr, int *NN, int *NBL)` – рассчитывает электронную плотность, используемую при расчете EAM силы. За основу взят Алгоритм 2 из той же работы [16];
- `ComputeForcesK(float3 *a, float4 *dr, int *NN, int *NBL, real *rho, real *fForce)` – рассчитывает силы, действующие между всеми атомами. Для расчета EAM сил между атомами металла и ЛД силы для пар металл-углерод за основу взят Алгоритм 2 из работы [16]. Вследствие необходимости учета дальних соседей, силы между атомами углерода в слое графена рассчитываются по специально разработанному алгоритму.

В основе практически всех ядер третьей группы, отвечающих за измерения, лежит алгоритм reduction. В данной программе используется общая схема, представленная в CUDA Programming Guide (с. 111), где функция `calculatePartialSum` реализована с использованием алгоритма, представленного в видеолекции “Control Flow.m4v” из курса “ECE 498AL” университета Illinois [22]. В принципе, все измерения можно было реализовать с помощью одной функции. Однако из-за различий в деталях вычислений многих величин принято решение по реализации отдельных функций для каждого измерения. К третьей группе относятся следующие ядра:

- `ComputeVSumK(float3 *dv, float3 *hlpArray)` – рассчитывает полный импульс системы. Отметим, что поскольку в безразмерных единицах масса углерода равна 1, то при суммировании для углерода используются только скорости. Для атомов металла учитывается масса;
- `ComputeVvSumK(float3 *dv, float3 *hlpArray, real cmVelX)` – находит сумму квадратов скоростей атомов, необходимую для вычисления кинетической энергии. Также для атомов металла учитывается масса, и в режиме сдвига из скорости каждого атома металла вычитается скорость центра масс наночастицы;
- `ComputeVvMaxK(float3 *dv, float3 *hlpArray)` – определяет максимальный квадрат скорости, используемый для проверки необходимости обновления списка соседей, исходя из формулы (4) раздела 3.1;

- `ComputePotEnergyK(float4 *dr, float3 *hlpArray)` – рассчитывает полную потенциальную энергию как сумму величин, сохраненных в `w` компонентах координат;

- `ComputeCenterOfMassK(float4 *dr, float3 *hlpArray)` – рассчитывает координаты центра масс наночастицы, используя стандартное выражение $\sum_i \mathbf{r}_i/N_m$, где N_m – число атомов металла (деление производится на хосте, а не в ядре), $\mathbf{r}_i$ – координата атома. Масса атомов не фигурирует, поскольку суммирование ведется только по атомам металла;

- `ComputeCmVelK(float3 *dv, float3 *hlpArray)` – определяет скорость центра масс наночастицы, используя выражение $\sum_i \mathbf{v}_i/N_m$, где N_m – число атомов металла (деление производится на хосте, а не в ядре), $\mathbf{v}_i$ – скорость атома. Масса атомов не фигурирует, поскольку суммирование ведется только по атомам металла;

- `ComputeParticleSizeK(float4 *dr, float3 *hlpArray, int min_max)` – в зависимости от значения `min_max` находит атом с минимальным или максимальным значением координат и сохраняет это значение в нулевой элемент массива `hlpArray`;

- `ComputeNetForceK(real *hlpArray)` – рассчитывает x компоненту полной силы, действующей на атомы металла со стороны атомов углерода. Это сила трения;

- `EvalRdfK(float4 *dr, int *CELL, int *histRdf, int countRdf)` – рассчитывает функцию радиального распределения $g(r)$ (RDF). В отличие от всех вышеперечисленных функций, выполняющих измерения, это единственное ядро, не использующее reduction. Оно основывается на Алгоритме 4 из работы [16] для построения списка соседей.

Кроме оберток, некоторых хост функций и ядер в SurfaceGrowth.cu есть несколько `__device__` функций. Они предназначены для расчета EAM функций ϕ, f, F и их производных по формулам, представленным в разделе 3.3.1.

5 Инструкция пользователю

В данном разделе даны указания по установке приложения и использованию его пользовательского интерфейса.

5.1 Установка

SurfaceGrowth поставляется в виде одного установочного файла. Установку приложения можно производить даже если система, на которую производится установка, *не удовлетворяет* следующим условиям, необходимым для работы программы:

- наличие GPU с Compute Capability 1.2 или выше. Это обусловлено использованием атомных функций, большинство из которых требует по крайней мере Compute Capability 1.1. Однако для atomicExch, используемой в некоторых ядрах с reduction, необходима версия Compute Capability 1.2 и выше;
- наличие 32-разрядной операционной системы Window XP или более старших версий Windows;
- наличие установленного программного обеспечения: CUDA Driver и CUDA Toolkit (желательно версии 2.3 или выше).

Эти условия должны быть выполнены все одновременно. Если система не удовлетворяет хотя бы одному из требований, то установка должна произойти успешно, но приложение не будет запускаться по причинам, которые будут указываться при попытках запуска. В частности, первое условие проверяется в приложении, и пользователю в случае GPU с неподходящей версией Compute Capability будет выведено соответствующее сообщение. Однако установку можно произвести с целью ознакомления с документацией приложения и просмотра некоторых результатов его работы. Для работы с кодом приложения необходимо еще установить CUDA SDK в папку, предлагаемую по умолчанию, поскольку в решении используется часть файлов из SDK.

Установка происходит по стандартной для многих приложений схеме, поэтому здесь детали не описываются.

После установки в каталоге, указанном пользователем (или оставленном по умолчанию), создаются следующие папки:

- `bin` – содержит Release версию приложения – SurfaceGrowth.exe, и некоторые dll файлы для работы OpenGL;

- `doc` – содержит описание программы – SurfaceGrowth_doc_rus.pdf и статьи, которые могут быть полезны для понимания работы приложения (работы [16, 17, 19]);
- `src` – содержит код приложения (решение Microsoft Visual Studio 2008);
- `samples` – примеры результатов, описанных в следующем разделе 6. Также vmd.rc для получения анимаций с помощью VMD.

Запустить приложение можно щелкнув по SurfaceGrowth.exe (используя ярлык программы или из меню “Пуск -> Все Программы -> Nikolay Prodanov”), или собрав проект в Visual Studio и запустив его на выполнение.

5.2 Интерфейс

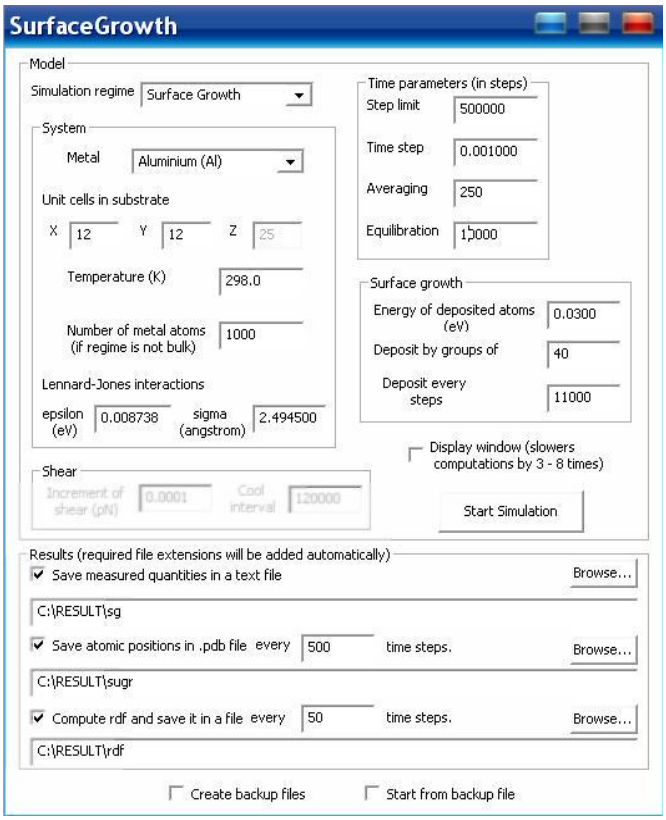


Рис. 3: Диалоговое окно SurfaceGrowth.

После запуска SurfaceGrowth должно появиться диалоговое окно, представленное на Рис. 3, которое позволяет пользователю задать параметры. Существуют параметры, общие для всех трех режимов, а некоторые параметры нужны только в каждом из режимов. Задавать параметры можно только до начала вычислений, в SurfaceGrowth не предусмотрено интерактивное изменение параметров.

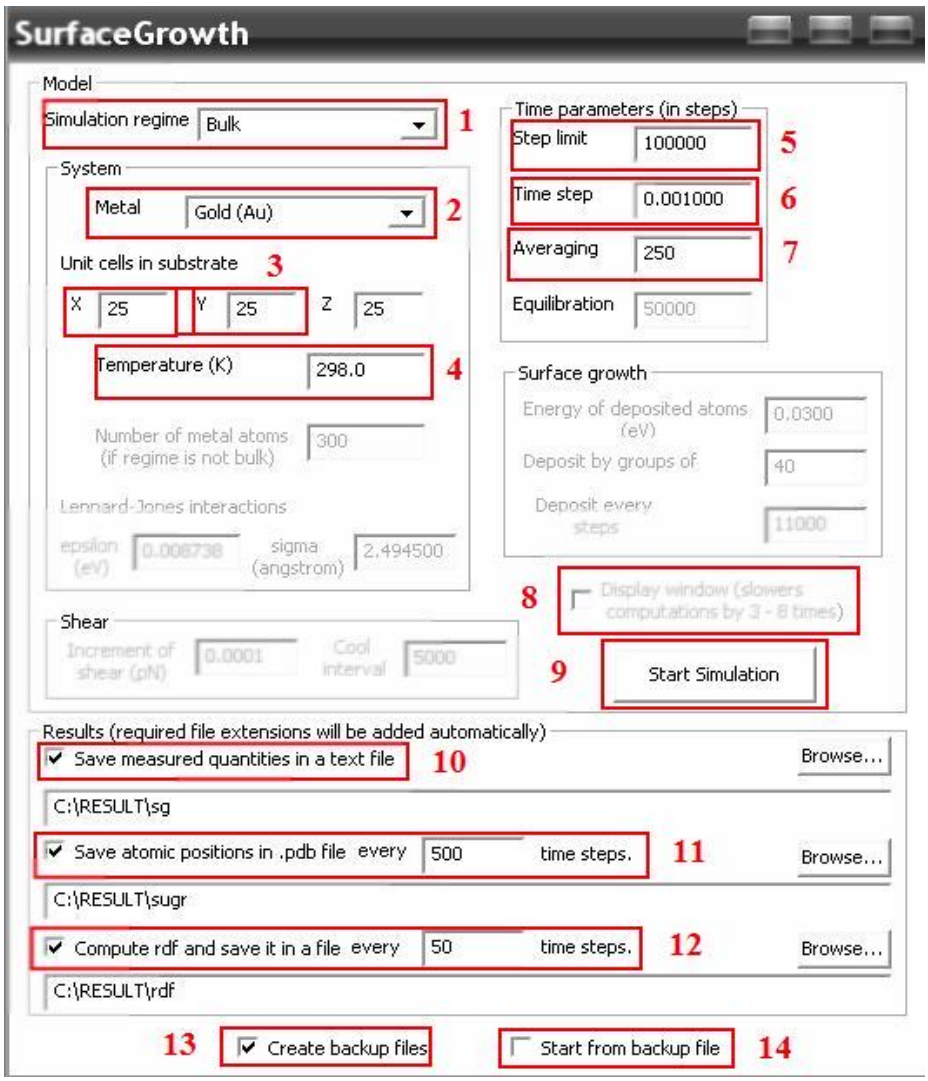


Рис. 4: Параметры, общие для всех режимов.

Рассмотрим параметры, общие для всех режимов (номера элементов в списке соответствуют нумерации на Рис. 4):

- 1 – выбор одного из 3х режимов. В зависимости от выбранного

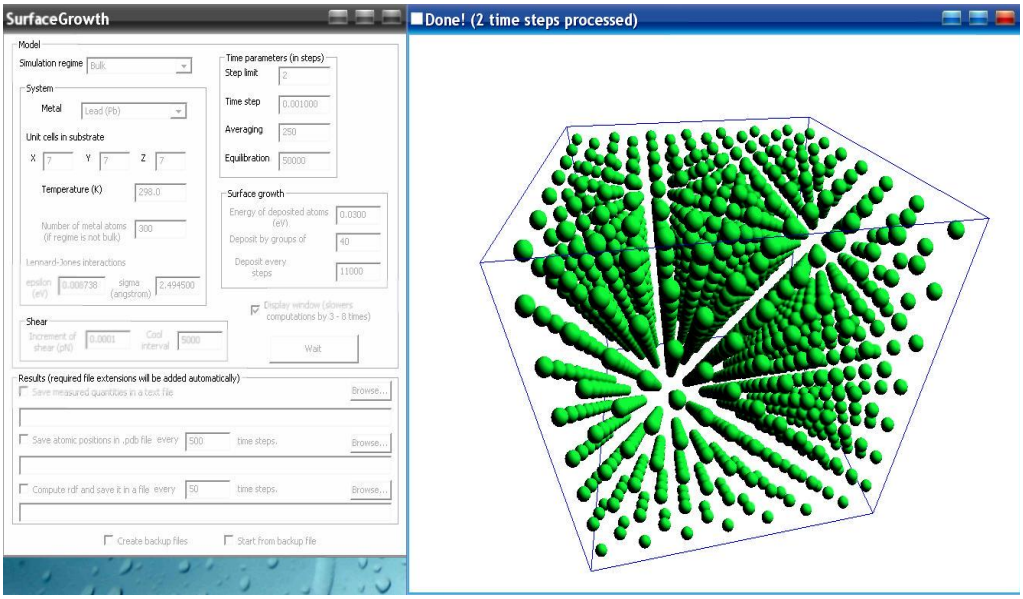


Рис. 5: OpenGL окно в режиме `Bulk`.

режима часть элементов окна, необходимая для данного режима, активируется, а часть становится неактивной;

- 2 – выбор одного из 6 металлов;
- 3 – число элементарных ячеек вдоль направлений x и y. В режимах `Surface Growth` и `Shear` элементарная ячейка, используемая при трансляции для построения слоя графена, содержит 32 атома;
- 4 – температура в градусах Кельвина;
- 5 – длительность моделирования, выраженная в количестве временных шагов;
- 6 – временной шаг в безразмерном виде. Значение 0.001 соответствует 0.2 fs;
- 7 – промежуток времени, по которому производится усреднение измеряемых величин, выраженный в количестве временных шагов. Через это количество шагов производится запись в файл результатов;
- 8 – флаг, определяющий, показывать или нет систему в OpenGL окне в режиме реального времени (Рис. 5). Как уже отмечалось,

установка флага приводит к существенному замедлению вычислений (от 3 до 8 раз), поэтому OpenGL окно стоит использовать только для малых систем (до 10000 атомов) в демонстрационных целях. При наличии окна система отображается внутри куба, длина ребра которого определяется максимальной длиной области моделирования в плоскости xy. При этом можно использовать следующие возможности:

  - вращать систему вокруг осей x, y используя движение мыши с нажатой левой кнопкой (LB);
  - вращать систему вокруг оси z, используя движение мыши с нажатой LB+Alt;
  - сдвигать систему, используя движение мыши с комбинацией LB+Shift, или с нажатой средней кнопкой мыши (MB);
  - масштабировать, Ctrl+LB или LB+MB;
  - делать паузу, нажав пробел.

  Кроме того, можно следить за ходом вычислений по заголовку OpenGL окна, а также выходить из приложения во время вычислений обычным способом, нажав крестик в верхнем правом углу окна. Однако при наличии OpenGL окна отсутствует возможность восстановления расчетов (back up);

- 9 – нажатие по этой кнопке начинает расчеты. При этом становятся неактивными все элементы управления диалогового окна, а надпись на кнопке изменяется на “Wait”. Если выбран показ системы в OpenGL окне, то эта надпись не изменяется по окончании расчетов, а нужно смотреть на заголовок OpenGL окна. Если OpenGL окна нет, то в конце вычислений на кнопке высвечивается “Done! You can exit”;

- 10 – флаг, определяющий сохранять или нет результаты в файл. Если он установлен, то нужно выбрать файл, куда будут сохраняться результаты (нажав Browse, выбрав путь расположения и указав имя файла). Отметим, что в имени файла следует задать только начало и *не нужно* указывать расширение, т. к. после начала вычислений в имени файла будет сохранены значения некоторых параметров (в зависимости от режима) и нужное расширение добавится автоматически;

- 11 – установка этого флага дает возможность сохранять результаты в pdb (Protein Data Bank) файл с целью дальнейшей визуализации

системы. Как и в предыдущем случае, если флаг установлен, нужно выбрать путь и имя файла. Однако рекомендуется имя файла оставить по умолчанию (“sugr_”), а для пути на диске C создать папку RESULT, и сохранять файлы в эту папку. Также следует указать количество шагов, через которое будет сохраняться файл. Это число не должно быть слишком малым (лучше значение 500 или 1000 в зависимости от длительности моделирования), т. к. запись в файл для больших систем может занимать много времени. Данный флаг рекомендуется использовать без OpenGL окна. Причина указанного выбора пути и имени файла в следующем. С программой SurfaceGrowth поставляется файл vmd.rc (расположенный в папке `samples`). Этот файл определяет поведение программы Visual Molecular Dynamics (VMD) [8] при ее запуске. Программа VMD, о которой мы уже упоминали ранее, предназначена для визуализации органических и неорганических молекул, координаты атомов которых заданы в различных форматах, в частности PDB. VMD предоставляет очень широкие возможности по визуализации, например, можно создавать видеоанимации. Если установить VMD, то в ее папке также будет файл vmd.rc. Рекомендуется его переименовать, и скопировать в эту папку файл vmd.rc, поставляемый с SurfaceGrowth. Тогда, если в папке “C:\ RESULT\” содержится последовательность pdb файлов, полученных с помощью SurfaceGrowth, например, sugr_0.pdb, sugr_1.pdb и т. д., то при запуске VMD они автоматически откроются в VMD, и можно будет наблюдать анимации системы в VMD. Описанным способом получены видео анимации, поставляемые с SurfaceGrowth, а также мгновенные снимки системы в следующем разделе 6;

- 12 – установка этого флага дает возможность производить измерения функции радиального распределения RDF и сохранять их в файл. Кроме указания пути и имени файла нужно также задать число шагов, через которое будут производиться измерения. Всего производится 100 измерений, после чего создается файл. Таким образом, если задано, что измерения производить каждые 50 шагов, то файл будет создаваться каждые 5000 шагов, отражая RDF за этот промежуток времени;

- 13 – установка этого флага приводит к созданию двух back up файлов (C:\bckup0.sugr, C:\bckup1.sugr) во время моделирования, в которые каждые 1000 шагов поочередно записывается вся информация, необходимая для работы приложения, а именно, структура

`SimParams`, массивы координат, скоростей и ускорений. В случае прерывания расчетов, установка следующего флага дает шанс возобновить вычисления. Отметим, что если моделирование дошло до конца и приложение было закрыто с использованием крестика в правом верхнем углу, то указанные два файла будут удалены. Если же приложение не закончило нормально работу, то эти два файла остаются на диске C. Также отметим, что back up возможен только без OpenGL;

- 14 – этот флаг указывает на желание возобновить прерванный счет с использованием back up файлов. При установке этого флага становятся неактивными все элементы управления, кроме флага выбора файла результатов (10 на Рис. 4) и кнопки "Start Simulation". Если не выбрано размещение файла результатов, а до прерывания он создавался, то такой файл и PDB файлы, если они создавались до прерывания, будут созданы в каталоге, из которого запущена программа. Отметим, что при нажатии кнопки "Start Simulation" элементы диалогового окна *не отображают* значения параметров, используемых в расчетах. Также отметим, что back up не всегда работает, особенно для больших систем при многократном прерывании счета.

Теперь рассмотрим элементы управления, специфичные для каждого из трех режимов. Эти элементы активны только при выборе соответствующего режима.

В режиме `Bulk` используются фактически только общие элементы управления, перечисленные выше (см. Рис. 6). Исключением является число элементарных ячеек вдоль направления z, обведенное на Рис. 6.

В режиме `Surface Growth` добавляются 6 специфичных параметров (номера в списке соответствуют цифрам на Рис. 7):

- 1 – задает длительность периода уравновешивания (equilibration), в количестве временных шагов. На протяжении этого промежутка времени в области моделирования находится только слой графена, к которому прикладывается термостат. Данный период предназначен для установления требуемой температуры, его длительность зависит от размеров слоя графена и определяется эмпирически;

- 2 – число атомов металла. Число атомов углерода равно произведению количества элементарных ячеек вдоль направлений x, y, умноженному на 32;

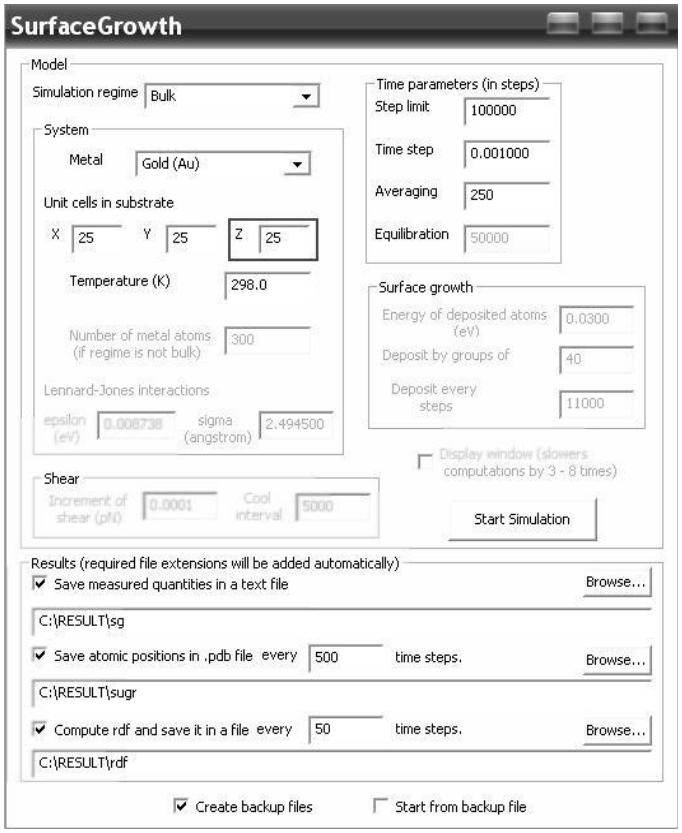


Рис. 6: Параметры режима `Bulk`.

- 3 – параметры потенциала Леннарда-Джонса, отвечающего за взаимодействивя металл-углерод. Значение `epsilon` следует подбирать с учетом энергии налетающих атомов и определяется эмпирически. Так, слишком малые значения приведут к отсутствию осаждения, поскольку атомы будут отражаться или испаряться с поверхности. Слишком большие значения приведут к малой подвижности осажденных атомов. А если энергия осаждения велика, то это может привести к дополнительному ускорению атомов и к разрушению слоя графена вследствие столкновений с быстрыми атомами. В целом, для наблюдения напыления значение `epsilon` должно быть чуть выше энергии осаждения. Значение `sigma` можно оставить по умолчанию. Отсутствие ограничений на эти параметры дает широкие возможности для изучения влияния величины энергии взаимодействия металл-подложка на характер напыления;

- 4 – энергия осаждаемых атомов, определяет начальное значение

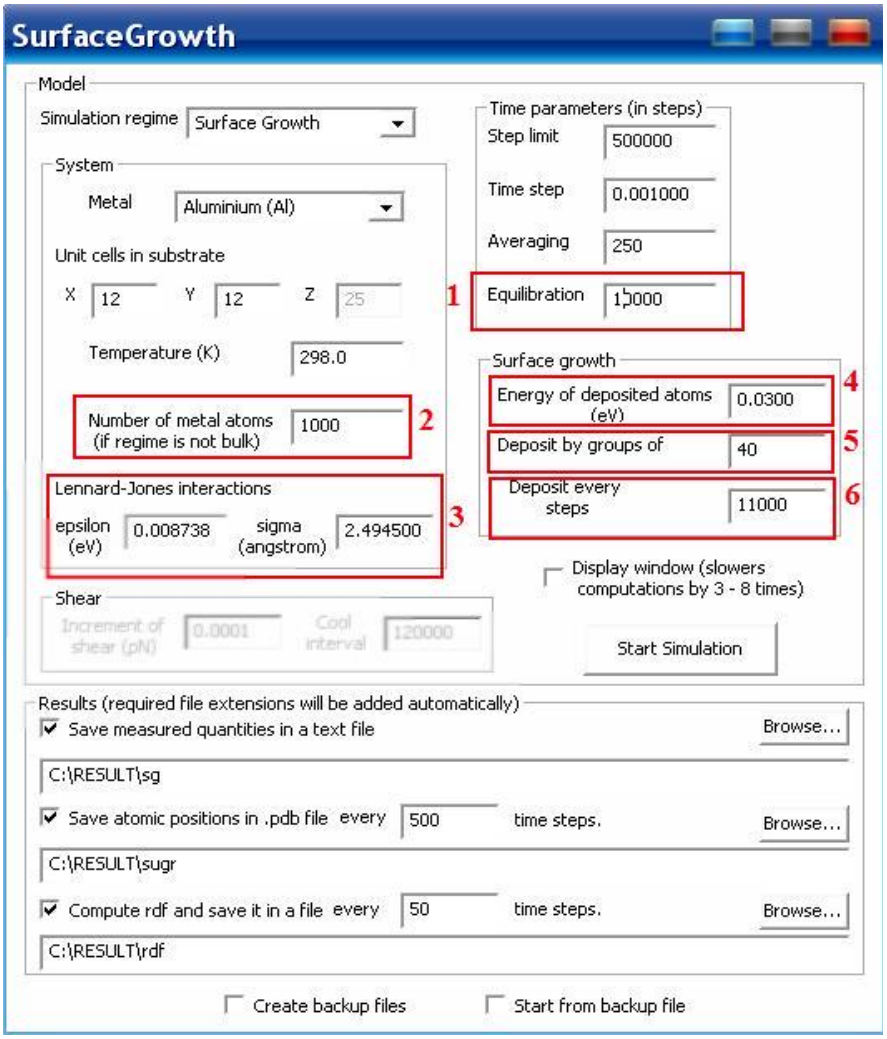


Рис. 7: Параметры режима `Surface Growth`.

вертикальной компоненты скорости атомов металла. Значения подбираются эмпирически. Слишком большие энергии приводят к разрушению слоя графена;

- 5 – определяет число атомов металла, которые одновременно появляются в области моделирования. Этот параметр вместе со следующим, определяет плотность потока осаждаемых атомов. Значение подбирается эмпирически, в зависимости от общего числа атомов металла, размеров области моделирования и следующего параметра. Завышенные значения приводят к разрушению слоя графена вследствие перегрева или столкновений с очень быстрыми атома-

ми;

- 6 – время, через которое появляются группы атомов, размер которых задается предыдущим параметром. Этот параметр подбирается эмпирически, в зависимости от общего числа атомов и размера группы. Так, если это время будет слишком мало для данного размера группы, то вследствие того, что осаждаемые атомы взаимодействуют между собой, они будут сталкиваться друг с другом, ускоряться и разрушать слой графена. Однако слишком большие времена приведут к очень долгому напылению. Так, если указано число атомов металла 20000, размер группы 50 атомов, то группа атомов будет вводиться в область моделирования $20000/50 = 400$ раз. Если указать время, через которое будет происходить ввод группы атомов (рассматриваемый параметр 6), равным 10000, то длительность напыления составит $400 \cdot 10000 = 4000000$ временных шагов, что может быть слишком большим (на NVIDIA® GeForce™ GTX 260 расчеты указанного числа шагов могут занять более 10 часов, в зависимости от размера слоя графена). Поэтому следует экспериментировать со значениями, учитывая цели исследований и имеющиеся в наличие оборудование и время.

В режиме `Shear` добавляются 5 специфичных параметров (номера в списке соответствуют цифрам на Рис. 8):

- 1 – период уравновешивания (временных шагов). На протяжении указанного числа временных шагов система предоставляется самой себе, термостат не прикладывается. Атомы металла перестраиваются в конфигурацию с минимальной потенциальной энергией, при этом происходит нагрев системы (см. раздел 3.3.4);

- 2 – количество атомов металла. Отметим, что число атомов углерода равно произведению количества элементарных ячеек вдоль направлений x, y, умноженному на 32;

- 3 – как и в предыдущем режиме это параметры потенциала Леннарда-Джонса, отвечающего за взаимодействив металл-углерод. Они дают возможность изучать движение наночастицы для различных величин взаимодействий со слоем графена;

- 4 – величина, определяющая, как быстро увеличивается сила сдвига, действующая на атомы наночастицы. Как уже отмечалось в разделе 3.3.4, сдвиг имитируется приложением силы вдоль оси x к атомам, расположенным левее центра масс (ЦМ) частицы. Поскольку

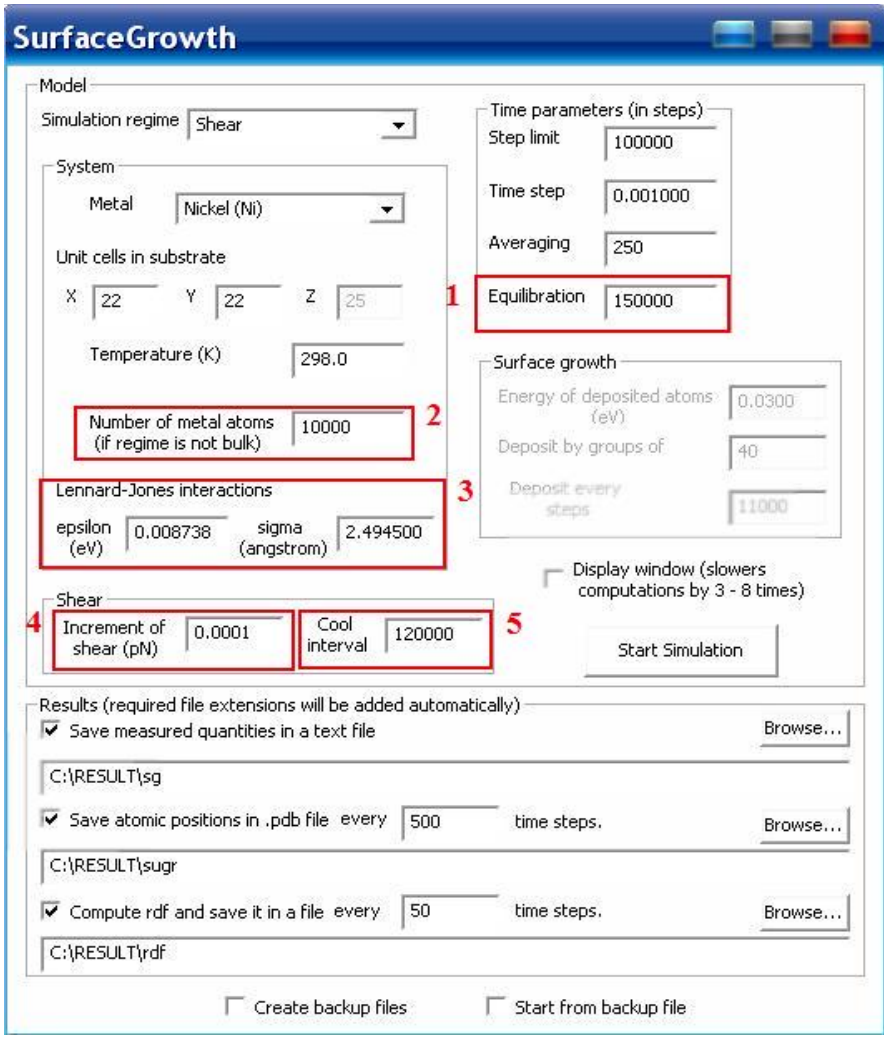


Рис. 8: Параметры режима `Shear`.

неизвестно, какая сила нужна для сдвига частицы (так называемая статическая сила трения), то в SurfaceGrowth выбран подход, согласно которому сила сдвига, прикладываетмая к каждому из указанных атомов, увеличивается пошагово, пока скорость ЦМ не достигнет определенного значения (выбранного равным 0.005). Параметр 4 как раз и определяет величину шага, с которым увеличивается сила;

- 5 – длительность охлаждения, т. е. промежутка времени, в течение которого к атомам металла прикладывается термостат Берендсена (см. раздел 3.3.4).

6 Результаты

Данный раздел содержит результаты, полученные в каждом из трех режимов работы программы для различных металлов. Результаты получены на GPU NVIDIA® GeForce™ GTX 260 для относительно небольших систем, т. к. целью не является демонстрация предельных возможностей программы. Программа должна давать схожие результаты при запусках на других GPU с параметрами, указанными в соответствующих разделах. При других комбинациях параметров не гарантируется адекватная работа программы (особенно в режиме `Surface Growth`), но можно поэкспериментировать.

Результаты получены в режиме без окна OpenGL. Поставляемые с приложением видео анимации (в папке `samples`) и мгновенные снимки получены в VMD. Примеры мгновенных снимков с использованием окна OpenGL представлены на титульном листе данного документа.

Имя текстового файла в зависимости от режима формируется из части, заданной пользователем, и части, содержащей значения параметров, определяемых режимом моделирования. Результаты в файле организованы в виде столбцов с заголовками, описанными в разделе 3.3.5. Также в самом верху файла после всех столбцов справа записывается величина временного шага и некоторых других параметров, а если приложение закончило работу нормально, то в конце файла записывается полное время счета в секундах (в режиме без OpenGL). Для отображения столбцов в блокноте нужно в меню “Формат” убрать галочку “Перенос по словам”.

Описываемые файлы результатов можно найти в папке `samples`. Отметим, что в ней представлены не все файлы для RDF, а только выборочные, и отсутствуют pdb файлы, с помощью которых были получены видео анимации. Для режима `Bulk` анимации не представлены.

6.1 Результаты для `Bulk` Au

В данном разделе описаны результаты моделирования системы в режиме `Bulk`, состоящей из $24 \times 24 \times 24$ элементарных ячеек или 55296 атомов золота, рассматривавшейся на протяжении 100000 временных шагов. Мгновенный снимок системы дан на Рис. 9, часть результатов из файла представлена на Рис. 10. В данном режиме первая часть имени файла результатов по умолчанию имеет вид “sg_blk_”, а затем имя содержит число элементарных ячеек, количество атомов металла, период усреднения, частоту генерации pdb файлов, начальную температуру и тип металла.

Из Рис. 10 видно, что полный импульс системы (impulse, обозначен

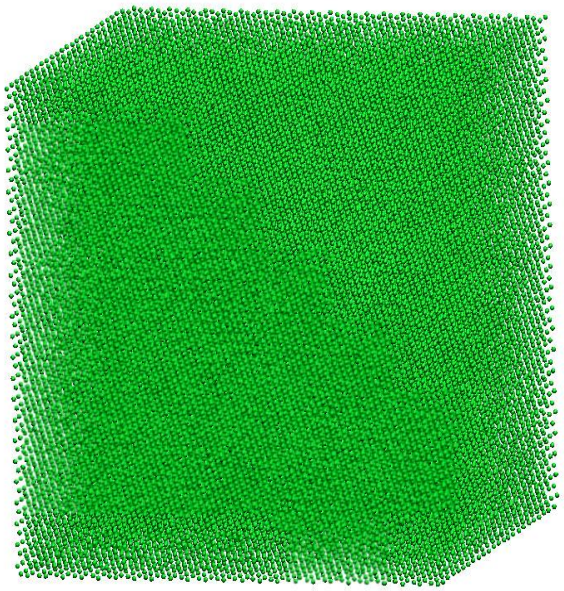
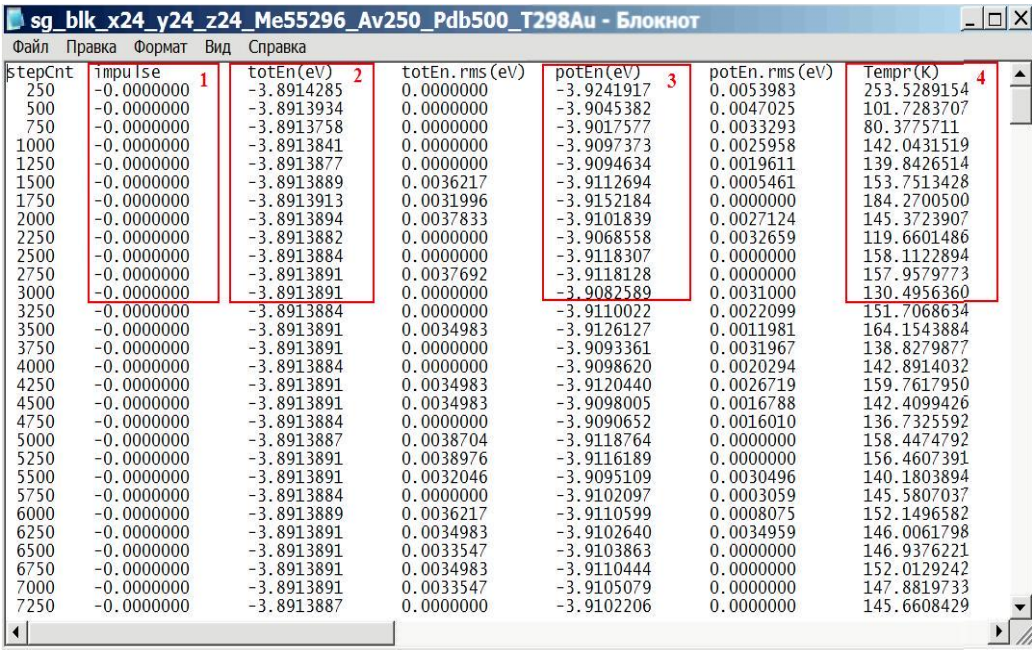


Рис. 9: Мгновенный снимок 55296 атомов золота в режиме `Bulk`.

числом 1) на протяжении начала моделирования равен нулю, что может свидетельствовать о правильности расчетов. Отметим, что вследствие накопления ошибок округления после 20000 шагов импульс начинает незначительно отклоняться от 0. Полная энергия системы (totEn(eV), 2) также сохраняет свое значение с хорошей точностью (до пятого знака после запятой включительно), что и должно быть в микроканоническом ансамбле. Значение потенциальной энергии (potEn(eV), 3) отражает энергию связи металла. Заметим, что температура (Tempr(K), 4) значительно упала относительно заданного значения 298 K, что связано с переходом части кинетической энергии атомов в потенциальную в NVE ансамбле [12]. На Рис. 10 не представлены результаты для длительности расчета одного шага, которая составила около 30 ms, а также скорости центра масс и силы трения, равных нулю. Координата центра масс системы также близка к нулю. Величины sizex(angstr), sizey(angstr), sizez(angstr) отражают размеры системы, которые приблизительно одинаковы и равны 96,3 Å. Полное время моделирования на GPU NVIDIA® GeForce™ GTX 260 составило немного меньше часа, а именно 3526 s.

На Рис. 11 показан вид функции радиального распределения $g(r)$ (RDF), полученной после первых 5000 шагов. Видно, что он представляет собой серию довольно острых пиков различной длины, отражаю-



sg_blk_x24_y24_z24_Me55296_Av250_Pdb500_T298Au - Блокнот

Файл Правка Формат Вид Справка

stepCnt	impulse	1	totEn(eV)	2	totEn.rms(eV)	potEn(eV)	3	potEn.rms(eV)	Tempr(K)	4
250	-0.0000000		-3.8914285		0.0000000	-3.9241917		0.0053983	253.5289154	
500	-0.0000000		-3.8913934		0.0000000	-3.9045382		0.0047025	101.7283707	
750	-0.0000000		-3.8913758		0.0000000	-3.9017577		0.0033293	80.3775711	
1000	-0.0000000		-3.8913841		0.0000000	-3.9097373		0.0025958	142.0431519	
1250	-0.0000000		-3.8913877		0.0000000	-3.9094634		0.0019611	139.8426514	
1500	-0.0000000		-3.8913889		0.0036217	-3.9112694		0.0005461	153.7513428	
1750	-0.0000000		-3.8913913		0.0031996	-3.9152184		0.0000000	184.2700500	
2000	-0.0000000		-3.8913894		0.0037833	-3.9101839		0.0027124	145.3723907	
2250	-0.0000000		-3.8913882		0.0000000	-3.9068558		0.0032659	119.6601486	
2500	-0.0000000		-3.8913884		0.0000000	-3.9118307		0.0000000	158.1122894	
2750	-0.0000000		-3.8913891		0.0037692	-3.9118128		0.0000000	157.9579773	
3000	-0.0000000		-3.8913891		0.0000000	-3.9082589		0.0031000	130.4956360	
3250	-0.0000000		-3.8913884		0.0000000	-3.9110022		0.0022099	151.7068634	
3500	-0.0000000		-3.8913891		0.0034983	-3.9126127		0.0011981	164.1543884	
3750	-0.0000000		-3.8913891		0.0000000	-3.9093361		0.0031967	138.8279877	
4000	-0.0000000		-3.8913884		0.0000000	-3.9098620		0.0020294	142.8914032	
4250	-0.0000000		-3.8913891		0.0034983	-3.9120440		0.0026719	159.7617950	
4500	-0.0000000		-3.8913891		0.0034983	-3.9098005		0.0016788	142.4099426	
4750	-0.0000000		-3.8913884		0.0000000	-3.9090652		0.0016010	136.7325592	
5000	-0.0000000		-3.8913887		0.0038704	-3.9118764		0.0000000	158.4474792	
5250	-0.0000000		-3.8913891		0.0038976	-3.9116189		0.0000000	156.4607391	
5500	-0.0000000		-3.8913891		0.0032046	-3.9095109		0.0030496	140.1803894	
5750	-0.0000000		-3.8913884		0.0000000	-3.9102097		0.0003059	145.5807037	
6000	-0.0000000		-3.8913889		0.0036217	-3.9110599		0.0008075	152.1496582	
6250	-0.0000000		-3.8913891		0.0034983	-3.9102640		0.0034959	146.0061798	
6500	-0.0000000		-3.8913891		0.0033547	-3.9103863		0.0000000	146.9376221	
6750	-0.0000000		-3.8913891		0.0034983	-3.9110444		0.0000000	152.0129242	
7000	-0.0000000		-3.8913891		0.0033547	-3.9105079		0.0000000	147.8819733	
7250	-0.0000000		-3.8913887		0.0000000	-3.9102206		0.0000000	145.6608429	

Рис. 10: Часть результатов, полученных в режиме `Bulk` для 55296 атомов золота.

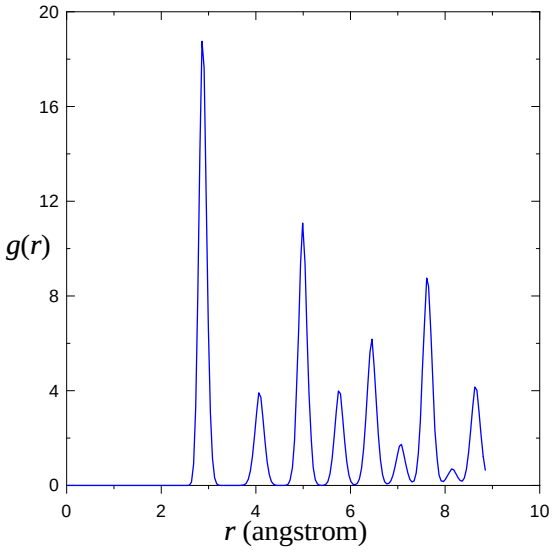


Рис. 11: Функция радиального распределения, полученная в режиме `Bulk` для золота после 5000 шагов.

щих кристаллическую структуру исследуемой системы. Горизонтальная ось показывает расстояние от данного атома (расположенного в точке с

$r = 0$) до соседних атомов. Положение первого пика дает расстояние до ближайшего соседа, которое задается параметром r_e в EAM потенциале. Для золота оно равно около 2.89 Å. Положение второго пика совпадает с расстоянием до второго соседа, которое равно постоянной решетки золота (около 4.07 Å). Отметим, что со временем положение пиков не изменяется, поэтому на Рис. 11 приведен только один график. Таким образом, полученная RDF правильно отражает структуру исследуемой системы. В качестве эксперимента можно повысить температуру металла намного выше точки плавления (значения которой см., например, в [18]) и посмотреть на вид RDF.

6.2 Результаты для `Surface Growth` Al

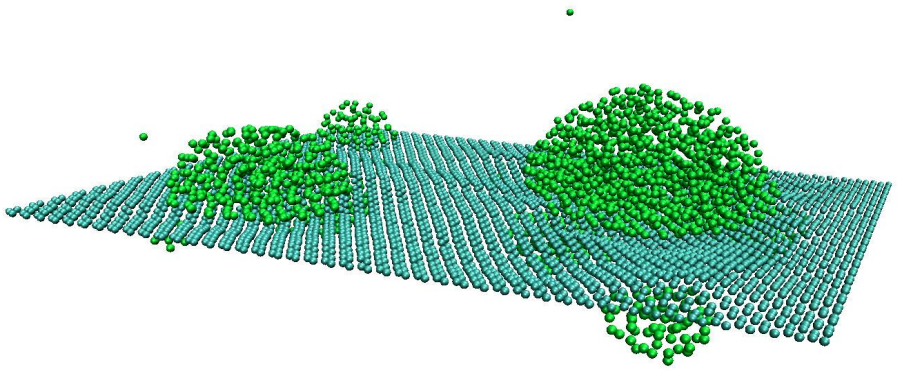


Рис. 12: Мгновенный снимок алюминия `Surface Growth`.

Кратко рассмотрим результаты моделирования небольшой системы в режиме `Surface Growth`, состоящей из 11×11 элементарных ячеек графена (или $11 \times 11 \times 32$=3872 атомов углерода), на который напыляется 2000 атомов алюминия. Мгновенный снимок системы в конце моделирования после 800000 шагов дан на Рис. 12. В данном режиме первая часть имени файла результатов по умолчанию имеет вид “sg_”, а затем имя содержит число элементарных ячеек, количество атомов металла, период уравновешивания, частоту напыления, размер группы напыляемых атомов, частоту генерации pdb файлов, начальную температуру и тип металла.

Как отмечалось, для данного режима очень важно правильно подобрать параметры. Мы использовали следующие параметры: энергия напыления 0.03 eV, параметры ЛД потенциала: epsilonLJ = 0.05 eV, sigmaLJ = 2.494500 angstrom. Атомы напылялись группами по 20 штук через

5000 временных шагов. Это обусловило то, что группы генерировались $2000/20 = 100$ раз, и длительность напыления составила 5000·100=500000 временных шагов. Мы не будем описывать результаты измерений, при желании читатель может сам проанализировать поведение различных величин, построив соответствующие графики зависимости. Отметим, что задав нулевое число атомов металла, импульс не будет равен нулю, поскольку крайние атомы графена закреплены. Как отмечалось, RDF в нашем случае особого смысла на протяжении большей части периода моделирования не несет, поэтому здесь не приводится. Отметим, что в результате напыления образовались островки алюминия полусферической формы (Рис. 12). Это указывает на то, что для выбранных параметров напыление происходит по механизму Вольмера-Вебера [23]. Часть атомов также осаждается под слоем графена, из-за периодических граничных условий. Процесс напыления можно наблюдать из видеоанимаций, длительность расчетов составила около 6000 s.

6.3 Результаты для `Shear` Ni

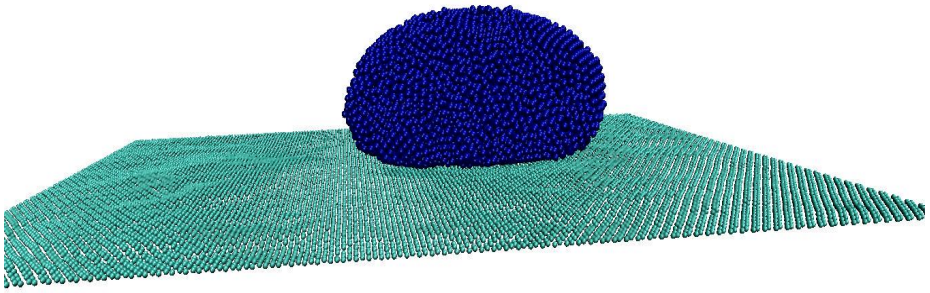


Рис. 13: Мгновенный снимок никелевой наночастицы в режиме `Shear`.

Рассмотрим результаты моделирования системы в режиме `Shear`, состоящей из 22×22 элементарных ячеек графена (или $22 \times 22 \times 32$=15488 атомов углерода), на котором находятся 10000 атомов никеля, так что общий размер системы 25488 атомов. Мгновенный снимок системы при сдвиге наночастицы дан на Рис. 13. В режиме `Shear` первая часть имени файла результатов по умолчанию имеет вид “sg_sh_”, а затем имя содержит число элементарных ячеек, количество атомов металла, периоды уравновешивания, охлаждения, усреднения, частоту генерации pdb

файлов, начальную температуру и тип металла. Отметим, что выбранные значения параметров достаточны для исследования только начала сдвига, что можно также заметить на видео.

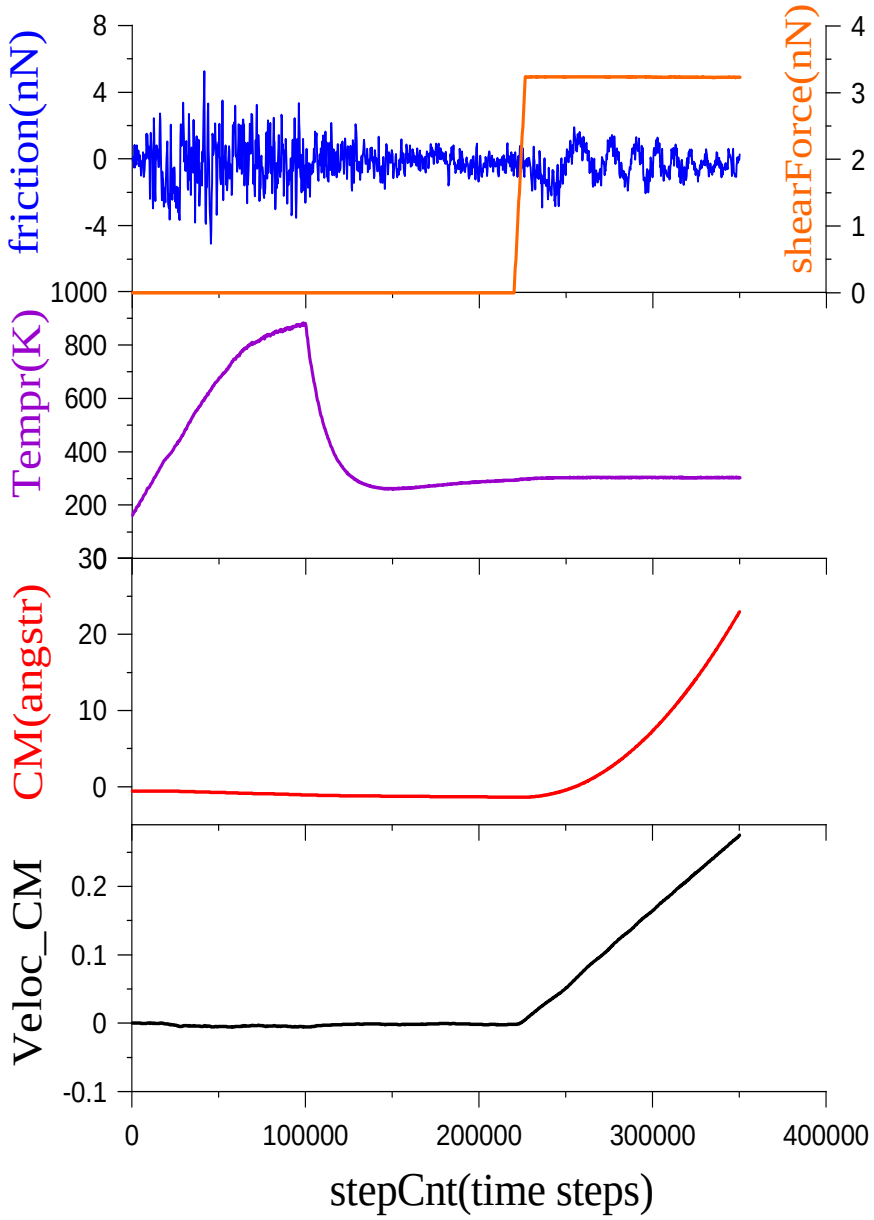


Рис. 14: Временные зависимости различных величин, полученные при моделировании в режиме `Shear` для никеля.

Рассмотрим подробнее некоторые измерения. На Рис. 14 представ-

лены временные зависимости скорости и координаты центра масс (ЦМ) наночастицы, температуры, сил трения и сдвига, действующих на частицу. Видно, что в период уравновешивания, который равен 100000 шагов, вследствие перегруппировки атомов и отсутствия термостата температура системы повышается до 900 K. В этот период сила трения имеет хаотический вид. После 100000 шагов прикладывается термостат ко всем атомам на протяжении 120000 шагов (stepCool). При этом температура экспоненциально релаксирует к заданному значению в 298 K. За эти периоды формируется наночастица размерами $8 \times 5.9 \times 3.9$ nm. После 220000 шагов, по окончании охлаждения, термостат остается приложенным только к графену, а к наночастице начинает прикладывать сила сдвига, которая быстро увеличивается (см. верхний график). Скорость и координата ЦМ наночастицы увеличиваются соответственно линейно и квадаратично. Можно отметить, что зависимость силы трения, действующей на частицу, от времени имеет пилообразный вид, что является характерным для так называемого прерывистого движения (stick-slip motion), часто наблюдаемого в нанотрибологических экспериментах [2,6]. Этот режим также проиллюстрирован на Рис. 15, где показана зависимость силы трения от координаты ЦМ. Причина прерывистого трения не является очевидной, и необходимы дальнейшие исследования, чтобы ее выяснить.

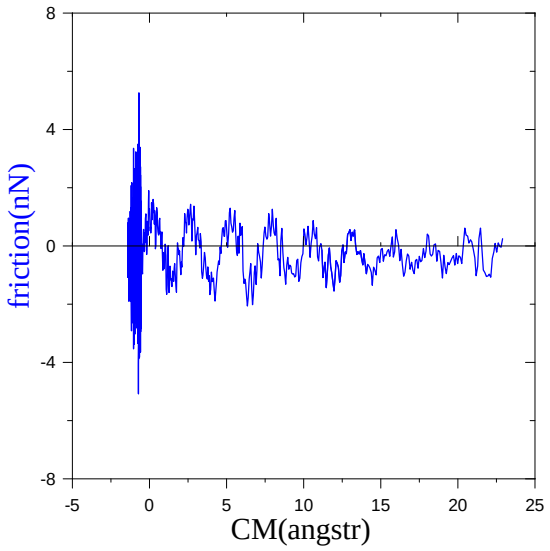


Рис. 15: Зависимости силы трения от положения центра масс наночастицы в режиме `Shear` для никеля.

На Рис. 16 представлена функция радиального распределения сразу

по окончании периода уравновешивания (синий график), и сразу перед началом сдвига (за 5000 временных шагов). Видно, что в отличие от режима `Bulk` (где для никеля, как и для золота наблюдаются острые пики), RDF не имеет серии четко выраженных пиков, что указывает на аморфную структуру наночастицы. Отметим, что после охлаждения (красный график) первый пик становится более выраженным, и второй "горб" разделяется на несколько "горбов", что может указывать на некоторое упорядочение атомов при снижении температуры. Длительность расчета 350000 временных шагов составила 6415 s.

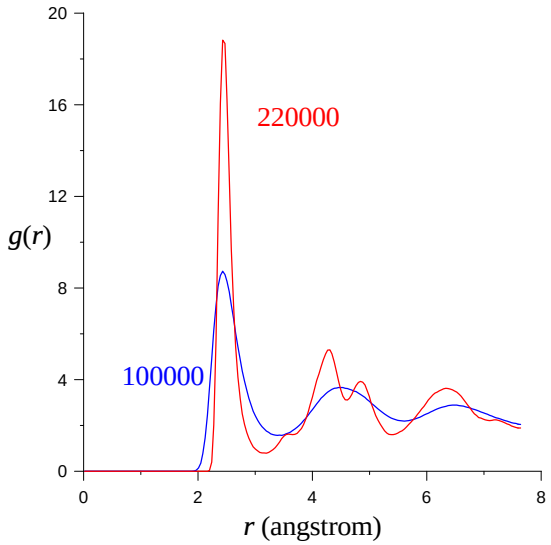


Рис. 16: Функция радиального распределения, полученная в режиме `Shear` для Ni после 100000 и 220000 шагов.

Представленные выше результаты для всех режимов имеют только лишь демонстративную цель. Поэтому вопросы существования фрикционной дуальности, отмеченной в разделе 2, и другие вопросы, связанные с физикой процессов, в данном документе не обсуждаются. Читателю рекомендуется поэкспериментировать со значениями параметров, с целью изучения поведения систем в различных условиях.

Список литературы

[1] B. Bhushan (Ed.), Springer Handbook of Nanotechnology, Springer, Berlin, 2004.

[2] B. Bhushan (Ed.), Nanotribology and Nanomechanics, Springer, Berlin, 2005.

[3] E. Gnecco, E. Meyer (Eds.) Fundamentals of Friction and Wear on the Nanoscale, Springer, Berlin, Germany, 2007.

[4] A.V. Khomenko, N.V. Prodanov, Carbon **48** (2010) 1234.

[5] D. Dietzel, C. Ritter, T. Mönninghoff, H. Fuchs, A. Schirmeisen, U.D. Schwarz, Phys. Rev. Lett. **101** (2008) 125505.

[6] A.V. Khomenko, N.V. Prodanov, Condens. Matter Phys. **11** (2008) 615.

[7] N.V. Prodanov, A.V. Khomenko, Surf. Sci. **604** (2010) 730.

[8] W. Humphrey, A. Dalke, K. Schulten, J. Molec. Graphics **14** (1996) 33 (http://www.ks.uiuc.edu/Research/vmd/).

[9] S.J. Heo, S.B. Sinnott, D.W. Brenner, J.A. Harrison, In: B. Bhushan (Ed.), Nanotribology and Nanomechanics, Springer, Berlin, 2005, p. 623.

[10] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Oxford, 1987.

[11] D. Frenkel, B. Smit, Understanding Molecular Simulation, Academic Press, London, 2002.

[12] D.C. Rapaport, The Art of Molecular Dynamics Simulation, 2nd ed., Cambridge University Press, Cambridge, 2004.

[13] M. Griebel, S. Knapek, G. Zumbusch, Numerical Simulation in Molecular Dynamics, Springer, Berlin, Heidelberg, 2007 (http://wissrech.ins.uni-bonn.de/research/projects/tremolo/).

[14] J.A. van Meel, A. Arnold, D. Frenkel, S.F.P. Zwart, R.G. Belleman, Mol. Sim. **34** (2008) 259.

[15] W. Liu, B. Schmidt, G. Voss, W. Muller-Wittig, Comp. Phys. Commun. **179** (2008) 634.

[16] J.A. Anderson, C.D. Lorenz, A. Travesset, J. Comp. Phys. **227** (2008) 5342.

[17] X.W. Zhou, H.N.G. Wadley, R.A. Johnson, D.J. Larson, N. Tabat, A. Cerezo, A.K. Petford-Long, G.D.W. Smith, P.H. Clifton, R.L. Martens, T.F. Kelly, Acta Mater. **49** (2001) 4005.

[18] ASM Handbook, Properties and Selection: Nonferrous Alloys and Special-Purpose Materials, V. 2, ASM International, 1992.

[19] N. Sasaki, K. Kobayashi, M. Tsukada, Phys. Rev. B **54** (1996) 2138.

[20] H. Berendsen, J. Postma, W. van Gunsteren, A. Di Nola, J. Haak, J. Chem. Phys. **81** (1984) 3684.

[21] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, P. Bourne, Nucleic Acids Research **28** (2000) 235 http://www.rcsb.org/pdb.

[22] http://developer.nvidia.com/object/cuda_training.html

[23] G.H. Gilmer, H. Huang, C. Roland, Comput. Mater. Sci. **12** (1998) 354.