

ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ МНОГОЯДЕРНЫХ СОПРОЦЕССОРОВ ПРИ СУПЕРКОМПЬЮТЕРНОМ СТАТИСТИЧЕСКОМ МОДЕЛИРОВАНИИ ЭЛЕКТРОННЫХ ЛАВИН¹

М.А. Марченко

Для моделирования развития электронных лавин в газе разработаны трехмерный параллельный алгоритм метода Монте-Карло и программа ELSHOW, реализованная с использованием комбинирования принципов крупно- и мелкозернистого параллелизма. Для реализации параллельных вычислений на высокопроизводительных гибридных вычислительных системах с сопроцессорами Intel Xeon Phi используется хорошо зарекомендовавшая себя библиотека PARMONC. Применение разработанной технологии распараллеливания существенно уменьшает вычислительную трудоемкость оценки таких интегральных характеристик, как число частиц в лавине, коэффициент ударной ионизации, скорость дрейфа и др.

Ключевые слова: электронная лавина, метод Монте-Карло, распараллеливание, суперкомпьютер.

Введение

Для моделирования электронной лавины в газе необходимо решать уравнение Больцмана, с этой целью нами используется метод Монте-Карло [1, 2]. Он позволяет учесть влияние «маловероятных процессов», что практически невозможно для других моделей (например, BOLSIG+ [3]). При всех достоинствах данного метода нужно обратить особое внимание на то, что приходится держать в памяти ЭВМ координаты в шестимерном фазовом пространстве (x, y, z, V_x, V_y, V_z) всех электронов лавины, количество которых растет экспоненциально со временем [4]. Частично эту проблему решает известная лексикографическая схема «ветвления» траекторий. Практически достаточный выигрыш во времени расчетов позволяет здесь получить использование технологии распараллеливания, что и было реализовано в представляемой программе ELSHOW (ELectron SHOWer). При моделировании влияние собственного электрического поля электронов и ионов лавины на внешнее электрическое поле не учитывалось, т.е. исследовалась только начальная стадия развития лавины (до формирования критической лавины, когда собственное поле лавины уравнивается с внешним электрическим полем).

Основной проблемой использования метода Монте-Карло для моделирования ионизационного размножения электронов является лавинообразное нарастание количества электронов и ионов (моделируемых частиц, для каждой из которых необходимо решать уравнения движения). Для решения этой проблемы можно использовать различные методы укрупнения (например, когда моделируемая частица представляется в виде облака, содержащего в себе несколько элементарных частиц), что приводит к ухудшению точности моделирования. Целесообразно, однако, учитывать траектории каждого отдельно взятого

¹ Статья рекомендована к публикации программным комитетом Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма – 2013».

электрона и использовать для моделирования эффективные технологии распараллеливания на многопроцессорных высокопроизводительных вычислительных системах [2, 4].

Алгоритм метода Монте-Карло для моделирования развития электронных лавин подробно описан в работе [5]. В настоящей работе делается акцент на технологии распараллеливания статистического моделирования на высокопроизводительных вычислительных системах с разными архитектурами.

Статья организована следующим образом. В разделе 1 описан алгоритм статистического моделирования развития электронных лавин в газе. В разделе 2 для разработанного алгоритма статистического моделирования представлены технологии распараллеливания для разных видов многопроцессорных вычислительных систем. В разделе 3 представлены результаты статистического моделирования электронных лавин и показан эффект от распараллеливания. В заключении сформулированы направления дальнейших исследований по численному статистическому моделированию электронных лавин.

1. Алгоритм статистического моделирования развития электронных лавин

1.1. Общее описание алгоритма

Параллельный трехмерный алгоритм статистического моделирования, реализованный в программе ELSHOW, учитывает ускорение электронов в электрическом поле, процессы упругого рассеяния электронов на молекулах газа и ионизации, а также возбуждения. Для этого используются сечения 24-х типов взаимодействий для азота, приведенные в [3, 6].

Рассматривается открытая система с внешним электрическим полем, напряженность которого $\mathbf{E} = (0, 0, -E_z)$ считается постоянной. В ходе моделирования с катода из точки $\mathbf{r} = (x, y, z) = (0, 0, 0)$ в момент времени $t = 0$ происходит выброс n_0 электронов с нулевыми энергиями. Затем прослеживаются траектории движения каждого из электронов, а также всех вторичных электронов, образовавшихся в результате ионизации, до достижения времени t_{\max} . С этой целью делаются одинаковые шаги Δt по времени такие, чтобы при этом «прямолинейный» пробег Δl был меньше 0,4 % длины свободного пробега для любых энергий (см. далее). За время Δt электрон с энергией T_{i-1} движется из точки \mathbf{r}_{i-1} до точки \mathbf{r}_i , где i – номер шага. При этом координаты и скорости частицы с зарядом e и массой m_0 изменяются следующим образом [2]:

$$\mathbf{r}_i = \mathbf{r}_{i-1} + \mathbf{V}_{i-1} \Delta t - e \mathbf{E} \Delta t^2 / (2 m_0), \quad \mathbf{V}_i = \mathbf{V}_{i-1} - e \mathbf{E} \Delta t / m_0.$$

В конце каждого временного шага разыгрывается столкновение с вероятностью

$$P = 1 - \exp(-\mathbf{b}_t(T_i) N \Delta l),$$

где \mathbf{b}_t – полное микроскопическое сечение взаимодействий, N – концентрация частиц газа. Затем разыгрывается тип взаимодействия в соответствии с сечениями упругого рассеяния, возбуждения и ионизации.

Упругое рассеяние не меняет энергию частицы, а направление движения определяется согласно дифференциальным сечениям [7, 8]. При этом азимутальный угол выбирается равномерно на отрезке $[0, 2\pi]$.

Для всех видов возбуждения энергия уменьшается на величину потенциала возбуждения, а направление движения разыгрывается также, как и для упругого рассеяния.

При ионизации энергия налетающего электрона T сначала уменьшается на величину потенциала ионизации T_{bd} , а затем остаток делится [2] между двумя вылетающими электронами: $T^{(1)} = T_1 + T_2$, $T_1 = T - \omega T_{bd}$, $T_2 = (\omega - 1) T_{bd}$, где ω – доля переданной энергии. Из простых кинематических уравнений (законов сохранения импульса и энергии) можно получить направления движения образующихся в результате ионизации электронов. Они задаются углами вылета θ_1 и θ_2 :

$$\sin^2 \theta_1 = T_2 / T^{(1)}, \sin^2 \theta_2 = T_1 / T^{(1)}.$$

Азимутальные углы связаны между собой соотношением $\varphi_1 = \varphi_2 + \pi$, причем один из них выбирается равновероятно на отрезке $[0, 2\pi]$.

1.2. Вычисление сечений

Для ускорения расчетов имеющаяся эмпирическая таблица сечений была дополнена значениями с помощью линейной интерполяции. Область энергий была поделена на несколько частей, в каждой из которых выбран свой достаточно маленький шаг, общий для всех типов взаимодействия. Значения сечений с таким шагом по энергии занесены в новую таблицу. Этот подход позволяет сделать поиск необходимого интервала в массиве сечений с помощью операции взятия целой части без длительного перебора. Окончательно требуемые значения получаются линейной интерполяцией в этом интервале.

1.3. Моделирование ионизации

Рассмотрим подробнее процедуру моделирования ω согласно плотности [2]:

$$\psi_{\omega}(u, T) = A [(3u + 4) \arctg b + b / (1 + b^2) + 2b(u - 4) / (1 + b^2)^2] / (\zeta u^3),$$

$$u \in [1, \zeta].$$

Здесь A – константа, $b^2 = \zeta - u$, $\zeta = T / T_{bd}$ – энергия налетающего электрона в единицах T_{bd} . Поскольку интервал, на котором функция $\psi_{\omega}(u)$ не равна нулю, зависит от энергии, перейдем к моделированию другой случайной величины

$$\gamma = (\omega - 1) / (\zeta - 1)$$

с плотностью распределения

$$\psi_{\gamma}(w, T) = A (\zeta - 1) [(3u + 4) \arctg b + b / (1 + b^2) + 2b(u - 4) / (1 + b^2)^2] / (\zeta u^3),$$

$$w \in [0, 1],$$

где $u = w(\zeta - 1) + 1$, $b^2 = (\zeta - 1)(1 - u)$. Эта функция была заатабулирована для набора энергий $\{T_j\}$.

Для промежуточных значений $T \in (T_{j-1}, T_j)$ плотность можно определить с помощью линейной интерполяции по параметру

$$\psi_{\gamma}(w, T) = \psi_{\gamma}(w, T_{j-1}) (T_j - T) / (T_j - T_{j-1}) + \psi_{\gamma}(w, T_j) (T - T_{j-1}) / (T_j - T_{j-1}).$$

При этом моделирование величины γ осуществляется методом суперпозиции [1] с вероятностью $P_1 = (T_j - T) / (T_j - T_{j-1})$ согласно плотности $\psi_{\gamma}(w, T_{j-1})$, а с вероятностью $P_2 = (T - T_{j-1}) / (T_j - T_{j-1})$ – соответственно $\psi_{\gamma}(w, T_j)$. Затем обратным преобразованием получаем $\omega = \gamma(\zeta - 1) + 1$.

Аналогичная процедура использовалась при моделировании углов рассеяния по таблицам дифференциальных сечений для упругого и неупругого рассеяния.

1.4. Лексикографическая схема

В программе ELSHOW моделирование индивидуальных соударений построено таким образом, что прослеживается путь частицы от момента вылета с катода до времени t_{\max} .

При этом в массив фактически записываются лишь вторичные частицы, которые образовались в результате ионизации. Затем моделируется путь одной из вторичных частиц до достижения времени t_{\max} , и так далее до тех пор, пока все вторичные частицы не будут рассмотрены. Такая схема расчета называется лексикографической. Она не требует больших объемов памяти для хранения всех частиц. Кроме того, предусмотрена возможность использования метода укрупненных соударений, в котором на определенном шаге по времени производится процедура удаления электронов с вероятностью P_u из рассмотрения («русская рулетка»). Для оставшихся частиц статистический «вес» умножается на величину $1/(1 - P_u)$. Получаемые при этом весовые оценки являются статистически несмещенными с конечной дисперсией, которая также оценивается.

1.5. Функционалы, погрешность, построение гистограммы

Программа ELSHOW в результате своей работы выдает для момента времени t_{\max} значения числа частиц n , положения центра масс $\mathbf{r}_c = (\langle x \rangle, \langle y \rangle, \langle z \rangle)$, средней скорости $\langle V_z \rangle$, средней энергии $\langle \epsilon \rangle$, гистограммы плотности частиц, а также их среднеквадратические погрешности. Это позволяет вычислить различные характеристики лавины такие, как скорость центра масс $V_c = \langle z \rangle / t_{\max}$ и скорость дрейфа $V_{dr} = \langle V_z \rangle$. Коэффициенты поперечной D_T и продольной диффузии D_L находятся путем определения по гистограмме соответствующих диффузионных радиусов [3]. Коэффициент ударной ионизации α определялся с помощью формулы [9]:

$$\alpha = (V_c - (V_c^2 - 4 \nu_i)^{1/2}) / (2 D_L),$$

где $\nu_i = \ln(n/n_0) / t_{\max}$ — частота ионизации. Более того, все числовые характеристики можно получить и для нескольких промежуточных значений времени, что дает возможность исследовать их динамику.

Гистограммой оценивалась плотность частиц, как функция расстояния r от оси OZ . При выборе шага гистограммы учитывалось, что в оптимальном варианте детерминированная и статистическая погрешности функциональной оценки должны быть приближенно равными. Для наиболее важной «осевой» ячейки было получено соотношение для шага $\Delta r = (8 D_T t)^{1/2} (2 n)^{-1/6}$.

1.6. Выбор шага

Шаг Δt был выбран пропорциональным $0.004 / \max(\sigma_t(T) (2T/m)^{1/2})$. Это соответствует тому, что Δl меньше случайного пробега с вероятностью $P_{pr} = 0,996$. Такое жесткое условие продиктовано тем, что требуется получать результаты с высокой точностью. С помощью метода зависимых траекторий была проведена серия расчетов, в которых использовалось два разных датчика случайных чисел. Один из них применялся только при розыгрыше столкновения, а другой – во всех остальных формулах распределенным способом. Это позволило скоррелировать траектории электронов для различных значений Δt . Наши расчеты показали, что при $P_{pr} = 0,996$ основные транспортные характеристики практически не отличаются для шагов Δt и $\Delta t/2$, т.е. указанный шаг Δt является удовлетворительным.

Указанный выше способ оценки подходящего шага Δt был успешно проверен с помощью тестового альтернативного расчета для $E = 0$ и среднего ненулевого значения начальной энергии, в котором моделировалась экспоненциально распределенная длина Δl свободного пробега с соответствующим пересчетом времени. Разности искомых величин

для основного («пошагового») и вспомогательного («экспоненциального») вариантов расчета оказались статистически незначимыми. При этом также был реализован распределительный способ использования псевдослучайных чисел.

2. Технологии распараллеливания статистического моделирования

Вероятностные модели, описывающие развитие электронных лавин, обладают большой вычислительной сложностью. Тем не менее, такого рода модели могут быть успешно реализованы с использованием распараллеливания на многопроцессорных вычислительных системах, например, с графическими ускорителями [10]. В программе ELSHOW распараллеливание осуществлялось в рамках единого подхода к реализации распределенного статистического моделирования, предложенного в работе [11]. Применяемая методология позволяет добиться высокой эффективности в использовании гибридных вычислительных систем и существенного уменьшения трудоемкости вычислений: в наших расчетах моделирование развития лавин электронов проводилось вплоть до 10^9 частиц без ухудшения точности оценки функционалов.

2.1. Метод распределенного статистического моделирования

Метод распределенного статистического моделирования состоит в распределении моделирования независимых реализаций по вычислительным ядрам с периодическим осреднением полученных выборочных значений по статистически эффективной формуле

$$\bar{\zeta} = \left[\sum_{m=1}^M l_m \right]^{-1} \sum_{m=1}^M l_m \bar{\zeta}_m. \quad (1)$$

Здесь $\bar{\zeta}$ — искомое выборочное среднее значение, M — общее число ядер, l_m — объем выборки, полученной на m -м ядре, $\bar{\zeta}_m$ — соответствующее m -му ядру выборочное среднее значение.

Очевидно, что главным критерием осуществимости такой параллельной реализации является возможность «поместить» данные вычислительной программы для моделирования реализаций в оперативную память каждого ядра. Подчеркнем, что в целях распределенного статистического моделирования допустимо использовать вычислительные ядра с разной производительностью. Важно, чтобы пересылка данных на центральное ядро и соответствующий прием данных осуществлялись в асинхронном режиме.

2.2. Параллельный генератор псевдослучайных чисел

Как правило, при параллельной реализации необходимый объем выборки базовых случайных чисел очень велик, поэтому целесообразно использовать длиннопериодные псевдослучайные последовательности. А именно, для решения «больших» задач по методу Монте-Карло предлагается использовать генератор следующего вида:

$$u_0 = 1, u_n \equiv u_{n-1} A \pmod{2^{128}}, \alpha_n = u_n 2^{-128}, n = 1, 2, \dots \quad (2)$$

$$A \equiv 5^{100109} \pmod{2^{128}}.$$

Получаемая последовательность псевдослучайных чисел является периодической, длина ее периода равна 2^{126} .

Для реализации статистического моделирования на независимых вычислительных ядрах предлагается следующий *распределительный способ использования базовых псевдослучайных чисел* (термин предложен член-корр. Г.А. Михайловым). Базовая последовательность предварительно разбивается на подпоследовательности, начинающиеся с чисел

$$\{\alpha_{m\mu}\}, m = 0, 1, 2, \dots,$$

где μ — длина «прыжка», после чего полученные таким образом подпоследовательности распределяются по разным ядрам. Значение «прыжка» генератора должно выбираться так, чтобы такого количества псевдослучайных чисел хватало для моделирования на каждом ядре. Легко показать, что для метода вычетов начальные значения указанных подпоследовательностей получаются по формуле

$$u_{(m+1)\mu} = u_{m\mu} A_{\mu} \pmod{2^{128}}, \alpha_{m\mu} = u_{m\mu} 2^{-128}, m = 0, 1, \dots \quad (3)$$

$$A_{\mu} \equiv A^{\mu} \pmod{2^{128}}.$$

Отметим, что длина периода генератора позволяет независимым образом распределять псевдослучайные числа по реализациям на практически неограниченное число вычислительных ядер. Параллельный генератор, приведенный выше, успешно используется в ряде институтов СО РАН на протяжении последних десяти лет.

2.3. Библиотека PARMONC

С целью унификации использования распределенного статистического моделирования при решении широкого круга задач методом Монте-Карло автором разработана и внедрена программная библиотека **PARMONC** (**PAR**allel **MON**te **C**arlo) [12]. Библиотека PARMONC установлена на кластерах Сибирского суперкомпьютерного центра (ЦКП ССКЦ СО РАН) и может использоваться на вычислительных системах с аналогичной архитектурой. Библиотека предназначена для распараллеливания программ, написанных на языках Fortran или C, причем библиотечные подпрограммы применяются без явного использования команд MPI. Инструкции по использованию библиотеки приведены в [13].

Возможность применения библиотеки PARMONC определяется «естественной» крупноблочной фрагментированностью программ статистического моделирования. Общая структура такого рода программ, в упрощенном виде, представлена на рис. 1 (в нотации языка C++).

```

void main( void ) {
    long int i, L;
    TypeRL RL, SUBT;
    SUBT = 0.0;
    // цикл по реализациям
    for( i = 0; i < L; i++ ) {
        // далее операторы, вычисляющие реализацию RL
        ...
        SUBT= SUBT + RL;
    }
    SUBT = SUBT / L;
}

```

Рис. 1. Общая структура программ статистического моделирования

Здесь L – общее число независимых реализаций случайного объекта, задаваемых композитным типом данных $TypeRL$ (допускается поэлементное суммирование переменных такого типа); реализации RL моделируются внутри цикла по переменной i . Полученные таким образом реализации RL (статистически независимые в совокупности) добавляются к «счетчику» $SUBT$ и по выходу из цикла осредняются, что дает статистическую оценку искомого математического ожидания случайного объекта.

При распараллеливании последовательных программ с помощью PARMONC определяется процедура *realization* (моделирующая подпрограмма), возвращающая одну реализацию (возвращение осуществляется через аргумент процедуры). При этом считается, что моделирующая подпрограмма использует потоки псевдослучайных чисел, генерируемых внешней по отношению к ней подпрограммой. Цикл по независимым реализациям и финальное осреднение заменяются вызовом библиотечной процедуры следующего вида:

```
parmoncc( realization, L, SUBT, ... );
```

Здесь имя моделирующей подпрограммы и общее число независимых реализаций передаются в библиотечную процедуру *parmoncc* в качестве входных аргументов; выборочное среднее будет возвращаться в переменную $SUBT$; для простоты остальные аргументы процедуры *parmoncc* опущены и заменены многоточием. Процедура *parmoncc* автоматически распределяет моделирование независимых реализаций по вычислительным ядрам. Все остальные операторы пользовательской программы остаются без изменений. На рис. 2 представлен модифицированный код, пригодный для компиляции и сборки с помощью библиотеки PARMONC.

```
void main( void ) {
    TypeRL SUBT;
    parmoncc (realization, L, SUBT , ... );
}

void realization( TypeRL RL ){
    // далее операторы моделирующей подпрограммы
    ...
    // вычисленная реализация возвращается в переменную RL
}
```

Рис. 2. Модифицированный код программы, пригодный для компиляции и сборки с помощью библиотеки PARMONC

В процессе распределенных вычислений на каждом ядре используются потоки независимых псевдослучайных чисел, получаемые в результате работы подпрограммы, реализующей параллельный генератор. В процедуре *realization* библиотечный параллельный генератор вызывается следующим образом:

$$a = rnd128(),$$

Здесь a — очередное псевдослучайное число, равномерно распределенное в интервале от нуля до единицы. Инициализация параллельного генератора выполняется автоматически при запуске программы, скомпилированной и собранной с помощью библиотеки PARMONC.

Следует упомянуть о практически важной возможности коррелирования результатов различных расчетов для одной и той же задачи, когда варьируется лишь ряд ее параметров. При использовании библиотеки PARMONC это делается на основе

использования одних и тех же псевдослучайных чисел в каждом из расчетов. Предлагаемый выше параллельный генератор позволяет распределять псевдослучайные числа также и по отдельным траекториям в каждом из расчетов. С использованием такого подхода проводились многократные численные расчеты по подбору параметров нашей вероятностной модели, в частности, величины временного шага в модели.

2.4. Комбинирование крупнозернистого и мелкозернистого распараллеливания

Для рассматриваемой задачи *методика крупнозернистого распараллеливания* заключается в распределении моделирования отдельных реализаций электронной лавины по отдельным ядрам. С помощью библиотеки PARMONC такого рода распределение делается автоматически. С одной стороны, среднее время моделирования отдельной реализации может быть достаточно большим (в некоторых расчетах оно достигало нескольких суток). С другой стороны, при крупнозернистом распараллеливании возникают проблемы, связанные с обработкой быстро растущего числа частиц в лавине.

Расчеты по методике крупнозернистого распараллеливания производились на кластере НКС-30Т в ЦКП ССКЦ СО РАН. Объем оперативной памяти, доступный каждому CPU ядру на узле, составляет от 4 до 8 ГБ. Таких объемов оперативной памяти вполне достаточно для реализации методики распараллеливания. Количество используемых ядер варьировалось в пределах от 128 до 512.

Методика мелкозернистого распараллеливания для рассматриваемой задачи заключается в моделировании отдельной реализации электронной лавины на одном многоядерном процессоре, например, на графическом сопроцессоре или акселераторе Intel Xeon Phi. С этой целью при реализации лексикографической схемы моделирование развития условно-независимых «ветвей» дерева (т.е. части лавины частиц) от вторичных частиц, появляющихся при актах ионизации, распределяется по разным ядрам сопроцессора. Естественно, необходимо сбалансированно распределять моделирование «ветвей» по ядрам сопроцессора с целью недопущения простоя ядер, таким образом, осуществляя балансировку вычислительной нагрузки. Такая балансировка (т.е. пересылка массивов вторичных частиц), очевидно, требует затрат машинного времени.

При лавинообразном росте числа частиц в реализации лавины следует также передавать «лишние» частицы в память CPU ядра для временного размещения в «магазине» памяти и при окончании моделирования всех «ветвей» на сопроцессоре, догружать частицы в память сопроцессора из «магазина» для дальнейшей обработки.

Ясно, что при таком распараллеливании необходимо распределять псевдослучайные числа также и по отдельным ядрам сопроцессора. Применение методики мелкозернистого распараллеливания помогает обрабатывать быстро растущее число частиц в лавине.

Поскольку узлы гибридного кластера имеют в своем составе разные вычислители (CPU ядра и сопроцессоры), то целесообразно комбинировать крупно- и мелкозернистое распараллеливание следующим образом. На каждом вычислительном узле часть CPU ядер будет моделировать реализации согласно методике крупнозернистого распараллеливания, в то время как сопроцессоры узла (и «прикрепленные» к ним CPU ядра) будут моделировать реализации лавины по методике мелкозернистого распараллеливания. Таким образом, при использовании такой *методики*

комбинированного распараллеливания вычислительные ресурсы гибридного узла будут задействованы максимально полно.

В следующем разделе приведены результаты численных расчетов с целью сравнения предложенных методик распараллеливания. Полученные результаты позволяют сделать вывод о целесообразности применения комбинированной методики распараллеливания на основе использования сопроцессоров Intel Xeon Phi для моделирования развития электронных лавин.

3. Результаты статистического моделирования

3.1. Результаты моделирования электронных лавин

Расчеты показали, что данная вычислительная модель хорошо согласуется с теоретическими и экспериментальными данными при $E_z/p < 200$ В/(см·Торр). Во всех расчетах относительная погрешность результатов ELSHOW не превосходит 2,6 %.

На рис. 3а представлены графики для скорости дрейфа, за которую принималась среднестатистическая скорость по ансамблю частиц $V_{dr} = \langle V_z \rangle$. На рисунке видно, что при $E_z/p > 200$ В/(см·Торр) наблюдается хорошее совпадение полученных с помощью ELSHOW данных с результатами расчетов с помощью программы BOLSIG+ [3]. Это, по всей видимости, связано с тем, что в ELSHOW используются преимущественно те же сечения взаимодействий, что и в BOLSIG+. Функция $3,3 \cdot 10^6 (E_z/p)^{1/2}$ (рис. 3а, штриховая линия), построенная по данным экспериментов [4], при больших E_z/p лежит ниже результатов ELSHOW (разница доходит до 21 %). Значения V_{dr} (рис. 3а, «точки»), полученные в [15], выше данных ELSHOW на 9–16 %.

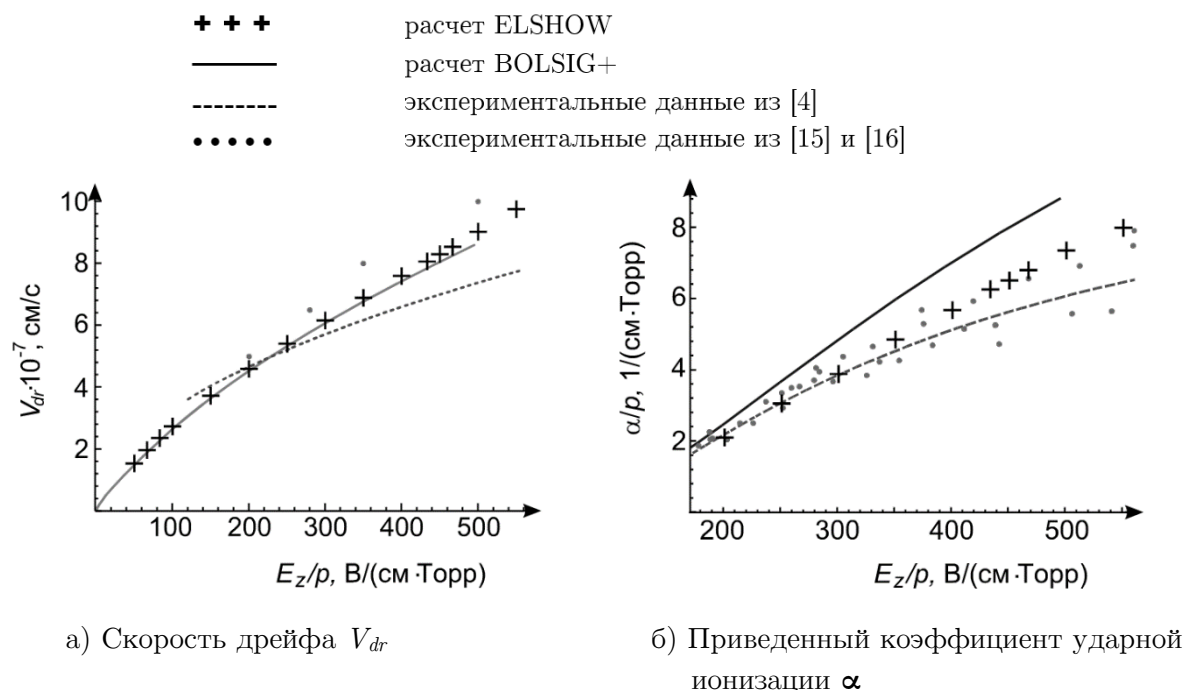


Рис. 3. Сравнение расчетных и экспериментальных данных

На рис. 3б представлены графики для приведенного коэффициента ударной ионизации α . Из рисунка видно, что при $E_z/p > 200$ В/(см·Торр) результаты ELSHOW лежат

ближе к экспериментальным данным ([4]—штриховая линия и [16]—«точки»), чем у результатов BOLSIG+. Функция $12 \cdot \exp(-342 p/E_z)$, построенная по результатам экспериментов (штриховая линия), лежит ниже значений ELSHOW до 19 %.

Важно отметить, что решение уравнения Больцмана с помощью метода Монте-Карло позволяет учесть маловероятные события при развитии лавины. Так, на рис. 4 приведены «портреты» электронных лавин, полученные в расчетах с помощью ELSHOW, в азоте при $p = 300$ Торр, $E_z/p = 50$ В/(см·Торр), $t_{\max} = 90$ нс (рис. 4а) и 500 В/(см·Торр), $t_{\max} = 0.055$ нс (рис. 4б). Из рисунков видно, что при внешнем поле порядка пробивного ($1.2 E_{np} = 15$ кВ/см) лавина имеет практически правильную сферическую форму (рис. 4а). При увеличении напряженности поля до значений около $12 E_{np}$ сильное влияние на форму лавины оказывает высокоэнергетичная часть функции распределения электронов по энергии. Помимо четко выраженной сферической формы имеются «ветвления», которые искажают форму лавины (рис. 4б). Этими соображениями, по-видимому, объясняется расхождение в значениях приведенного коэффициента ударной ионизации между результатами ELSHOW и BOLSIG+ (рис 3б).

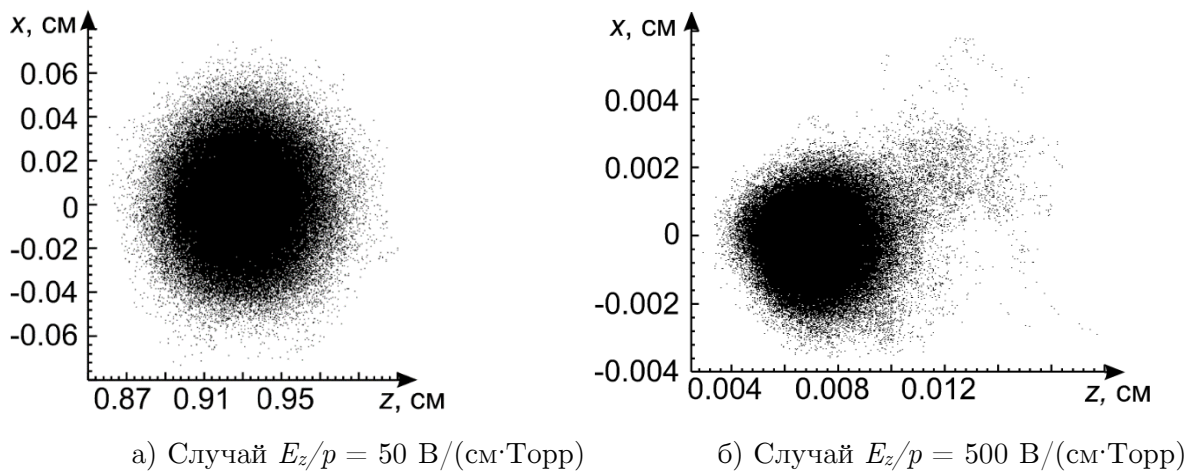


Рис. 4. «Портреты» электронных лавин, полученные в расчетах с помощью ELSHOW

3.2. Результаты распараллеливания на многоядерных сопроцессорах

Расчеты по методике комбинированного распараллеливания производились на прототипе суперкомпьютера МВС-10П в МСЦ РАН. На этом кластере на каждом вычислительном узле доступны два 8 ядерных процессора Intel Xeon E5-2690 и два 60 ядерных сопроцессора Intel Xeon Phi SE10X. Объем оперативной памяти, доступный каждому ядру на узле, составляет 4 ГБ. На каждом сопроцессоре для моделирования доступно примерно 8 ГБ оперативной памяти (за вычетом объема оперативной памяти, требуемой операционной системе сопроцессора). Таких объемов оперативной памяти вполне достаточно для реализации описанных выше методик распараллеливания. При программной реализации использовалась технология т.н. offload-режима исполнения задач на сопроцессоре [14].

Для одной из физических постановок задачи сравнивался эффект от применения методики крупнозернистого распараллеливания с эффектом от применения методики комбинированного распараллеливания (на одном вычислительном узле). При вычислениях варьировалась только величина t_{\max} , что давало в конце моделирования различное число частиц в лавине (большему значению времени соответствовало большее число частиц).

Для каждого сочетания методики распараллеливания и величины t_{\max} оценивались значения функции $L = L(t)$, где $L(t)$ - среднее количество реализаций лавины частиц, полученных к моменту машинного времени t .

Как показали расчеты, для $t_{\max} = 0.01$ нс в лавине образуется в среднем примерно 10^6 частиц, а при $t_{\max} = 1$ нс – в среднем около $8 \cdot 10^7$ частиц. В программе, реализующей методику комбинированного распараллеливания, при числе частиц в лавине более 10^6 намеренно производилось периодическое перераспределение частиц между ядрами сопроцессора и «прикрепленного» CPU ядра.

На рис. 5 представлено сравнение эффективности двух методик распараллеливания. По горизонтальной оси отложено машинное время t в секундах, по вертикальной оси – соответствующее количество реализаций L . Линия с ромбами — крупнозернистое распараллеливание, $t_{\max} = 0,01$ нс; линия с кругами — крупнозернистое распараллеливание, $t_{\max} = 1$ нс; линия с квадратами — комбинированное распараллеливание, $t_{\max} = 0,01$ нс; линия с треугольниками — комбинированное распараллеливание, $t_{\max} = 1$ нс.

Как видно из рис. 5, для одной и той же величины t_{\max} методика комбинированного распараллеливания позволяет получить значительно большее число реализаций в течение заданного машинного времени. Таким образом, при большой величине t_{\max} методика комбинированного распараллеливания предпочтительнее методики крупнозернистого распараллеливания (несмотря на затраты машинного времени на балансировку вычислительной нагрузки).

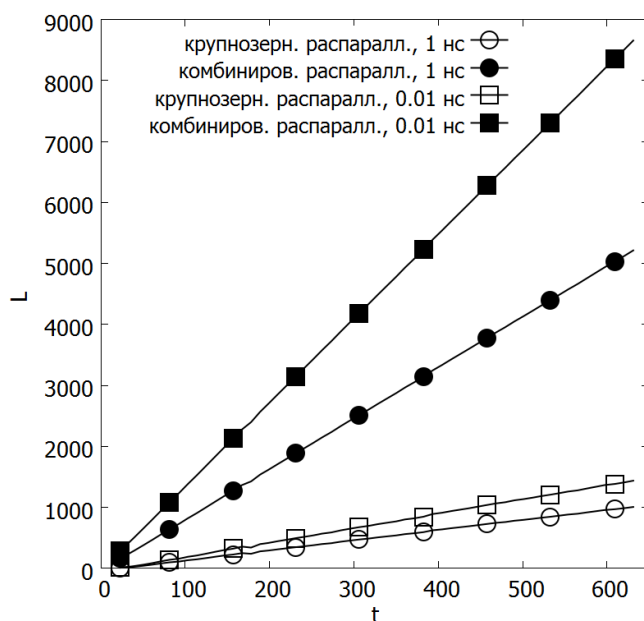


Рис. 5. Сравнение эффективности двух методик распараллеливания

Заключение

В работе представлены трехмерный параллельный алгоритм метода Монте-Карло и программа ELSHOW, предназначенные для моделирования развития электронных лавин в газе. В численных экспериментах показывается удовлетворительное соответствие результатов расчетов с помощью ELSHOW известным из литературы экспериментальными данными и результатам расчетов с помощью других пакетов программ. Изучается эффективность применения различных методик распараллеливания (крупнозернистого и

мелкозернистого параллелизма и их комбинации), делается вывод о целесообразности применения предложенной комбинированной методики распараллеливания на основе использования сопроцессоров Intel Xeon Phi для моделирования развития электронных лавин.

В дальнейшем предполагается усовершенствовать разработанный параллельный алгоритм с целью моделирования облака электронов начиная с момента, когда собственное электрическое поле электронов и ионов лавины становится сравнимым с внешним электрическим полем [4]. С целью повышения эффективности моделирования следует, возможно, отказаться от лексикографической схемы «ветвления» траекторий и использовать «метод поколений» [1]. Следует также оценивать влияние высокоэнергичных электронов на развитие пробоя в газе [2, 4]. Эта задача связана с оценкой т.н. «редких» событий и требует применения специальных методик статистического моделирования. Для таких постановок статистическое моделирование на гибридных суперкомпьютерах с использованием методики комбинированного распараллеливания представляется весьма перспективным.

Настоящая работа проводилась при финансовой поддержке грантов РФФИ №№ 13-07-00589, 13-01-00746, 12-01-00034, 12-01-00727; МИП №№ 39, 47, 126, 130 СО РАН.

Пользуясь случаем, автор выражает признательность гл. специалисту ЦКП ССКЦ СО РАН Н.В. Кучину за помощь и плодотворные обсуждения.

Литература

1. Ермаков, С.М. Курс статистического моделирования / С.М. Ермаков, Г.А. Михайлов — М.: Наука, 1976. — 320 с.
2. Аккерман, А.Ф. Моделирование траекторий заряженных частиц в веществе / А.Ф. Аккерман — М.: Энергоатомиздат, 1991. — 200 с.
3. Hagelaar, G.J.M. Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models / G.J.M. Hagelaar, L.C. Pitchford // Plasma Sources Sci. Technol. — 2005. — Vol. 14. — P. 722–733.
4. Королёв, Ю.Д. Физика импульсного пробоя газов / Ю.Д. Королев, Г.А. Месяц. — М.: Наука. Гл. ред. физ.-мат. лит., 1991. — 224 с.
5. Параллельная реализация метода Монте-Карло для моделирования развития электронных лавин в газе / Г.З. Лотова, М.А. Марченко, Г.А. Михайлов и др. // Известия высших учебных заведений. Физика. — 2013.
6. Itikawa, Y. Cross Sections for Collisions of Electrons and Photons with Nitrogen Molecules. / Y. Itikawa, M. Hayashi, A. Ichimura, K. Onda, K. Sakimoto, K. Takayanagi, M. Nakamura, H. Nishimura, T. Takayanagi // J. Phys. Chem. Ref. Data — 1986. — Vol. 15, No. 3. P. 985–1010.
7. Okhrimovskyy, A. Electron anisotropic scattering in gases: A formula for Monte Carlo simulations / A. Okhrimovskyy, A. Bogaerts, R. Gijbels // Phys. Rev. E. — 2002. — Vol. 65, No. 037402. — P. 1–4.
8. Sun, W. Detailed theoretical and experimental analysis of low-energy electron-N₂ scattering / W. Sun, M.A. Morrison, W.A. Isaacs, W.K. Trail, D.T. Alle, R.J. Gulley, M.J. Brennan, S.J. Buckman // Phys. Rev. A. — 1995. — Vol. 52, No. 2. — P. 1229–1256.

9. Tagashira, H. The development of electron avalanches in argon at high E/N values. II. Boltzmann equation analysis / H. Tagashira, Y. Sakai, S. Sakamoto // J. Phys. D: Appl. Phys. — 1977 — Vol. 10. — P. 1051.
10. Жуковский, М.Е. Математическое моделирование радиационной эмиссии электронов на гибридных суперкомпьютерах / М.Е. Жуковский, Р.В. Усков // Вычислительные методы и программирование. — 2012. — Т. 13, № 1. — С. 189–197.
11. Марченко, М.А. Распределенные вычисления по методу Монте-Карло / М.А. Марченко, Г.А. Михайлов // Автоматика и телемеханика. — 2007. — Вып. 5. — С. 157–170.
12. Марченко, М.А. Библиотека PARMONC для решения «больших задач по методу Монте-Карло» / М.А. Марченко // Вестник ННГУ. — 2012. — № 5. — С. 392–397.
13. Марченко, М.А. Библиотека PARMONC на сайте ЦКП ССКЦ СО РАН / М.А. Марченко. URL: <http://www2.sccc.ru/SORAN-INTEL/paper/2011/parmonc.htm> (дата обращения: 19.08.2013).
14. Jeffers, J. Intel Xeon Phi Coprocessor High - Performance Programming. / J. Jeffers, J. Reinders — Elsevier, 2013. — 432 p.
15. Lisovskiy, V. Electron drift velocity in argon, nitrogen, hydrogen, oxygen and ammonia in strong electric fields determined from rf breakdown curves / V. Lisovskiy, J.P. Booth, K. Landry, D. Douai, V. Cassagne, V. Yegorenko // J. Phys. D: Appl. Phys. — 2006. — Vol. 39. — P. 660–665.
16. Dutton, J. A survey on electron swarm data / J. Dutton // J. Phys. Chem. Ref. Data. — 1975. — Vol. 4, No. 3. — P. 577–851.

Марченко Михаил Александрович, к.ф.-м.н., ученый секретарь, Институт вычислительной математики и математической геофизики СО РАН, доцент кафедры вычислительной математики, механико-математический факультет, Новосибирский государственный университет (Новосибирск, Российская Федерация), mam@osmf.sccc.ru.

EFFECTIVE USE OF MULTICORE COPROCESSORS IN SUPERCOMPUTER STOCHASTIC SIMULATION OF ELECTRON AVALANCHES

M.A. Marchenko, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, Novosibirsk State University (Novosibirsk, Russian Federation)

Three-dimensional parallel Monte Carlo algorithm for modelling the electron avalanches in gases is developed. Parallel Implementation is made on supercomputers with MPP architecture and on hybrid supercomputers with Intel Xeon Phi coprocessors. The well-working library PARMONC is used to implement parallel computations. The use of the library enables fast calculation of functionals such as the number of particles in avalanche, first Townsend coefficient, drift velocity, etc.

Keywords: electron avalanche, Monte Carlo method, parallelization, supercomputer.

References

1. Ermakov S.M., Mikhailov G.A. Kurs statisticheskogo modelirovaniya [Course of stochastic simulation]. Moscow, Nauka, 1976. 320 p.
2. Akkerman A.F. Modelirovanie traektorij zaryazhennykh chastic v veshestve [Simulation of trajectories of charged particles in medium]. Moscow, Energoatomizdat, 1991. 200 p.
3. Hagelaar G.J.M., Pitchford L.C. Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models. Plasma Sources Sci. Technol. 2005. Vol. 14. P. 722–733.
4. Korolev Ju.D., Mesyats G.A. Physics of impulse breakdown in gases. Moscow, Nauka, 1991. 224 p.
5. G.Z. Lotova, M.A. Marchenko, G.A. Mikhailov, et al. Parallel realization of Monte Carlo method for modelling of electron avalanches in gases // Izvestiya vyshykh uchebnykh zavedeniy. 2013.
6. Itikawa Y., Hayashi M., Ichimura A., et al. Cross Sections for Collisions of Electrons and Photons with Nitrogen Molecules. // J. Phys. Chem. Ref. Data. 1986. Vol. 15, No. 3. P. 985–1010.
7. Okhrimovskyy A., Bogaerts A., Gijbels R. Electron anisotropic scattering in gases: A formula for Monte Carlo simulations // Phys. Rev. E. 2002. Vol. 65, No. 037402. P. 1–4.
8. Sun W., Morrison M.A., Isaacs W.A., et al. Detailed theoretical and experimental analysis of low-energy electron-N₂ scattering // Phys. Rev. A. 1995. Vol. 52, No. 2. P. 1229–1256.
9. Tagashira H., Sakai Y., Sakamoto S. The development of electron avalanches in argon at high E/N values. II. Boltzmann equation analysis // J. Phys. D: Appl. Phys. 1977. Vol. 10. P. 1051.
10. Zhukovskiy M.E., Uskov R.V. Mathematical modeling of radiative electron emission using hybrid supercomputers // Numerical methods and programming. 2012. Vol. 13, P. 271–279.
11. Marchenko M.A., Mikhailov G.A. Distributed computing by the Monte Carlo method. // Automation and Remote Control. 2007. Vol. 68, Iss. 5, P. 888–900.
12. Marchenko M. PARMONC - A Software Library for Massively Parallel Stochastic Simulation. // LNCS. 2011. Vol. 6873. P. 302–315.
13. Marchenko M.A. Page of PARMONC on the web site of Siberian Supercomputer Center. URL: <http://www2.sccc.ru/SORAN-INTEL/paper/2011/parmonc.htm> (accessed: 19.08.2013).
14. Jeffers J., Reinders J. Intel Xeon Phi Coprocessor High-Performance Programming. Elsevier, 2013. 432 p.
15. Lisovskiy V., Booth J.P., Landry K., et al. Electron drift velocity in argon, nitrogen, hydrogen, oxygen and ammonia in strong electric fields determined from rf breakdown curves // J. Phys. D: Appl. Phys. 2006. Vol. 39. P. 660–665.
16. Dutton J. A survey on electron swarm data // J. Phys. Chem. Ref. Data. 1975. Vol. 4, No. 3. P. 577–851.

Поступила в редакцию 20 августа 2013 г.