

РАСПРЕДЕЛЕННОЕ ЧИСЛЕННОЕ СТАТИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ВЕТВЯЩИХСЯ ПРОЦЕССОВ С УЧЕТОМ АРХИТЕКТУРЫ СУПЕР-ЭВМ

М. А. Марченко^{1, 2}

¹ *Институт вычислительной математики и математической геофизики СО РАН, 630090, Новосибирск*

² *Новосибирский государственный университет, 630090, Новосибирск*

УДК 519.6 + 519.21 + 519.245 + 537.523

В работе предлагается методика распараллеливания алгоритмов численного статистического моделирования траекторий ветвящихся случайных процессов с учетом архитектуры гибридных вычислительных системах. Приводится формула для оценки ускорения от распараллеливания. Используются распределительный способ получения псевдослучайных чисел и методика распределенного численного статистического моделирования. Разработанная методика апробирована при моделировании процесса развития электронных лавин в газе.

Ключевые слова: ветвящиеся процессы, численное статистическое моделирование, параллельные алгоритмы, суперкомпьютеры, электронные лавины.

Введение

В работе предлагается вычислительная технология распараллеливания алгоритмов численного статистического моделирования траекторий ветвящихся процессов [1] на гибридных вычислительных системах. Новизна предлагаемого подхода заключается в эффективном применении распределительного способа получения псевдослучайных чисел и методики распределенного численного статистического моделирования с одновременным использованием основных процессоров и сопроцессоров гибридной вычислительной системы.

Разработанная нами методика апробирована при моделировании процесса развития электронных лавин в газе, причем известные способы «огрубления» моделирования эволюции ансамбля частиц (методы «русской рулетки», «утолщенных траекторий», «групповых соударений» и др. [2]) не применялись.

1 Моделирование траекторий ветвящихся процессов

Ограничимся классом ветвящихся процессов, определяемых следующей схемой [1]:

- в момент времени $t = 0$ в некоторой области Ω в d -мерном пространстве согласно плотности распределения $p_0(x)$ рождаются n_0 независимых частиц в точках $x_i^{(0)}$ ($i = 1, 2, \dots, n_0$), характеризующих состояние соответствующих частиц;
- в следующий момент времени (соответственно шагу Δt) каждая из существующих на предыдущем шаге частиц либо погибает с вероятностью $0 \leq g(x_i^{(0)}) < 1$, либо с вероятностью $1 - g(x_i^{(0)})$ выживает, после чего определяется случайное число ее потомков m согласно дискретного распределения $P(m = k | x_i^{(0)})$, $0 \leq k \leq m^*$ (величина m^* фиксирована); при $m \geq 1$ в точках y_1, y_2, \dots, y_m , имеющих совместную плотность распределения $p(x_i^{(0)} \rightarrow y_1, y_2, \dots, y_m) / (1 - g(x_i^{(0)}))$, моделируются порожденные ею вторичные частицы; каждая из выживших частиц далее меняет свое состояние согласно плотности распределения $r(x_i^{(0)} \rightarrow x')$;

- обработка всех частиц (первичных и вторичных) на следующем шаге производится аналогично изложенному в предыдущем пункте.

В процессе моделирования траектории в определенные моменты времени по набору частиц из соответствующего поколения (скажем, с номером k) вычисляются заданные случайные оценки $\zeta^{(k)}$, дающие вклад при вычислении выборочных средних $L^{-1} \sum \zeta^{(k)}$ для оценки требуемых функционалов (скорость распространения эпидемии в медицинских задачах, коэффициента размножения электронов в задачах расчета реакторов и др.). Предполагается, что $\zeta^{(k)}$ является аддитивной функцией переменных $x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}$ ($n = n(k)$):

$$\zeta^{(k)} = \sum_{i=1}^n f(x_i^{(k)})$$

В лексикографической схеме (при обходе дерева «в глубину») в памяти выделяется место под массив — т.н. магазин частиц, в каждом элементе которого хранится текущее состояние частицы. После «рождения» новая частица помещается в магазин для дальнейшей обработки, а до конца обрабатывается только одна траектория частицы. После того, как произведено моделирование всех переходов для рассматриваемой частицы, приступают к обработке других частиц из магазина, причем целесообразно все время выбирать частицы с конца магазина (т.е. действовать по методу LIFO). Для каждой выбранной частицы моделируются оставшиеся ей переходы по той же методике, т.е. все вторичные частицы также помещаются в магазин. Обработка дерева завершается, когда будут выбраны все частицы из магазина.

Ясно, что при использовании лексикографической схемы размер магазина соответствует максимальному числу переходов в дереве, равному $t_{max}/\Delta t$. Следовательно, использование лексикографической схемы предпочтительнее счета по поколениям, так как число переходов заранее известно и до начала расчетов можно зарезервировать необходимый объем машинной памяти.

Вычислительная трудоемкость рассматриваемого алгоритма численного статистического моделирования деревьев частиц (как для счета по поколениям, так и для лексикографической схемы) является достаточно большой, поскольку среднее время на обработку всего дерева зависит от величины t_{max} по экспоненциальному закону:

$$t_1 = C_1 n_0 \exp(C_2 t_{max}), \quad C_1, C_2 > 0 \quad (1)$$

Множитель n_0 при экспоненте объясняется тем, что для каждой начальной частицы соответствующие ей деревья моделируются независимо. Кроме того, при моделировании деревьев большой глубины величина дисперсии оценок функционалов для больших значений времени может быть большой. Ясно, что такого рода численные алгоритмы практически нереализуемы без применения технологий распараллеливания и использования современных высокопроизводительных вычислительных систем.

2 Методы распараллеливания численного статистического моделирования траекторий ветвящихся процессов

1. Под массивно-параллельной вычислительной системой будем понимать однородную (из одинаковых процессоров или вычислительных ядер) систему с распределенной памятью, где у каждого процессора (вычислительного ядра) — своя оперативная память. Процессоры при этом обычно объединены в вычислительные узлы, причем в самой вычислительной системе вычислительные узлы могут иметь различную производительность. Под **гибридной вычислительной системой** будем понимать систему, состоящую из вычислительных узлов, на каждом из которых расположены основные многоядерные процессоры (CPU) и сопроцессоры, например, Intel Xeon Phi или Nvidia GPU.

2. Методику распределенного статистического моделирования, в которой моделирование независимых реализаций дерева частиц распределяется по разным процессорам (вычислительным ядрам), будем называть **крупноблочным распараллеливанием**. Такая методика эффективно реализуется на массивно-параллельных вычислительных системах.

3. Более «глубокое» распараллеливание заключается в моделировании отдельной реализации дерева частиц совместно на нескольких ядрах. А именно, моделирование условно независимых поддеревьев от вторичных частиц, появившихся к некоторому моменту времени, распределяется по разным ядрам. Будем называть такую методику **мелкоблочным распараллеливанием**. Ее применение помогает обрабатывать быстро растущее число частиц, тем самым ускоряя расчет реализации дерева, и использовать бо́льший объем оперативной памяти для размещения массивов частиц.

Для реализации мелкоблочного распараллеливания можно совместно использовать вычислительные ядра процессоров (одного или нескольких) массивно-параллельной вычислительной системы, а также ядра основных процессоров и сопроцессоров гибридных вычислительных систем.

Допустим, что для заданного (основного) процессора выделено N дополнительных процессоров; например, на вычислительном узле к одному ядру основного процессора «прикрепляются» N ядер сопроцессора. Моделирование одной реализации дерева определим следующим образом:

1. на основном процессоре производится моделирование дерева до определенного момента времени t_p ; пусть «накопленное» к этому моменту число частиц $n(t_p) \approx n_p(N + 1)$, $n_p \geq 1$;
 2. на дополнительные процессоры передаются примерно $n_p N$ частиц, на основном процессоре при этом также остается примерно n_p частиц;
 3. на основном и дополнительных процессорах начинается моделирование поддеревьев из соответствующих частиц;
 4. в определенные моменты времени на каждом процессоре производится расчет функционалов;
 5. в конце моделирования на всех процессорах массивы с рассчитанными функционалами передаются на основной процессор, где производится окончательный расчет функционалов для смоделированной реализации.
4. На каждом вычислительном узле гибридной вычислительной системы целесообразно комбинировать крупно- и мелкоблочное распараллеливание:
- одна часть CPU-ядер на процессорах узла будет моделировать реализации согласно методике крупноблочного распараллеливания;
 - другая часть CPU-ядер на процессорах узла и «прикрепленных» к ним ядер сопроцессоров будут моделировать реализации дерева по методике мелкоблочного распараллеливания.

При использовании такой методики **комбинированного распараллеливания** вычислительные ресурсы будут задействованы максимально полно.

5. Выведем формулу для ускорения при комбинированном распараллеливании на гибридной вычислительной системе. Производительность ядра основного процессора примем за единицу, производительность ядер дополнительных процессоров обозначим как α . Для современных сопроцессоров, как правило, $\alpha < 1$.

Пусть из общего числа $M = M_1 + M_2$ ядер основных процессоров на M_1 производится моделирование по методике крупноблочного распараллеливания, а к каждому из M_2 ядер «прикреплено» по N ядер сопроцессора и на них производится моделирование по методике мелкоблочного распараллеливания; каждое из M_2 ядер назовем **основным ядром**.

Представим среднее время моделирования одной реализации на основном ядре в виде суммы

$$t_1 = \tau_1 + \tau_2,$$

где τ_1 — время, затраченное на моделирования $n_p(N + 1)$ частиц (п. 1 упомянутого выше алгоритма), τ_2 — время на обработку остальной части дерева. Предполагается, что $\tau_1 \ll \tau_2$, т.е. переход к мелкоблочному распараллеливанию осуществляется достаточно быстро сравнительно с общим временем счета.

При мелкоблочном распараллеливании среднее машинное время на получение одной реализации равно

$$t'_1 = \tau_1 + \frac{\tau_2}{(N + 1)\alpha} + \tau_s + \tau_b, \quad (2)$$

где τ_s — машинное время на передачу массивов с функционалами из памяти сопроцессора в память основного процессора, их получение и «объединение» с соответствующими массивами на основном процессоре, τ_b — машинное время на возможную синхронизацию и обмен частицами при балансировке нагрузки между основными ядрами и ядрами сопроцессора.

В (2) второе слагаемое соответствует времени на параллельное моделирование условно независимых поддеревьев на основном ядре и ядрах сопроцессора. Поскольку каждому ядру при распараллеливании «достанется» n_p начальных частиц и процедура распараллеливания не меняет вероятностных характеристик реализаций ветвящегося процесса, то, в силу (1), время моделирования лавины уменьшится в среднем в

$N + 1$ раз. Поскольку время счета до синхронизации определяется самым медленным процессом, а для ядер сопроцессора, как правило, $\alpha < 1$, то максимальное время будет достигнуто именно на них.

Если учесть, что $\tau_1 \ll \tau_2$, а также пренебречь величинами τ_s и τ_b , то ускорение от применения мелкоблочного распараллеливания на одном ядре можно оценить величиной

$$S = t_1/t'_1 \approx \alpha(N + 1)$$

В частности, эта величина выражает максимальный выигрыш от применения методики комбинированного распараллеливания по сравнению с методикой крупноблочного в предельном случае (по видимому, практически нереальном), когда $M_1 = 0$, т.е. к каждому основному ядру «прикреплен» сопроцессор.

Для всей вычислительной системы при использовании только основных процессоров при крупноблочном распараллеливании за машинное время T будет получено в среднем $L = MT/t_1$ реализаций, а при использовании методики комбинированного распараллеливания будет получено

$$L' = M_1T/t_1 + M_2T/t'_1 \approx M_1T/t_1 + M_2T\alpha(N + 1)/t_1$$

реализаций. Таким образом, величина выигрыша от применения методики комбинированного распараллеливания по сравнению с методикой крупноблочного L'/L приближенно равна

$$g = \frac{1 + \alpha\beta(N + 1)}{1 + \beta}, \quad \beta = M_2/M_1 \quad (3)$$

Следовательно, ускорение от применения методики комбинированного распараллеливания выражается величиной

$$S \approx M \frac{1 + \alpha\beta(N + 1)}{1 + \beta}.$$

3 Распределительный способ получения псевдослучайных чисел и методика распределенного численного статистического моделирования для различных архитектур вычислительных систем

1. Для получения базовой последовательности псевдослучайных чисел $\{\alpha_n\}$ предлагается использовать следующий 128-битный конгруэнтный генератор псевдослучайных чисел (метод вычетов) [4]:

$$u_0 = 1, \quad u_n \equiv u_{n-1}A \pmod{2^{128}}, \quad \alpha_n = u_n 2^{-128}, \quad n = 1, 2, \dots, \quad (4)$$

где множитель A определяется выражением $A \equiv 5^{100109} \pmod{2^{128}}$.

2. При реализации методики распределенного статистического моделирования целесообразно определять различные подпоследовательности базовых псевдослучайных чисел для вычислительных экспериментов (конкретных расчетов для выбранной задачи), процессоров, реализаций (отдельных случайных испытаний) и их элементов. Такой способ можно назвать **распределительным**, так как он особенно удобен для распределенных вычислений.

Базовая последовательность $\{\alpha_n\}$ предварительно разделяется на подпоследовательности заданной длины μ , начинающиеся с чисел $\alpha_{k\mu}$, $k = 0, 1, 2, \dots$. Значение μ должно выбираться из тех соображений, что μ псевдослучайных чисел практически достаточно для реализации соответствующего укрупненного блока. Для метода вычетов начальное значение k -й подпоследовательностей получается по формуле

$$u_{(k+1)\mu} \equiv u_{k\mu}A^\mu \pmod{2^{128}}, \quad \alpha_{k\mu} = u_{k\mu} 2^{-128}, \quad k = 0, 1, 2, \dots \quad (5)$$

Такая методика распределения псевдослучайных чисел позволяет определять одинаковые подпоследовательности целиком для каждого вычислительного эксперимента, процессора, реализации и ее элементов. Ее использование улучшает корреляционную зависимость между результатами расчетов для разных вариантов задач, что особенно полезно при параметрическом анализе вероятностных моделей.

3. Для получения псевдослучайных чисел при моделировании по методике комбинированного распараллеливания на гибридной вычислительной системе предлагается применять следующую процедуру:

- базовая последовательность $\{\alpha_n\}$ разделяется на подпоследовательности для вычислительных экспериментов (обозначим их длину как n_e);

- подпоследовательности для вычислительных экспериментов разделяются на подпоследовательности для вычислительных узлов (обозначим их длину как n_n);
- подпоследовательности для вычислительных узлов разделяются на подпоследовательности для групп вычислительных ядер (обозначим их длину как n_g);
- подпоследовательности для групп вычислительных ядер разделяются на подпоследовательности для вычислительных ядер (обозначим их длину как n_c);
- подпоследовательности для вычислительных ядер разделяются на подпоследовательности для реализаций (обозначим их длину как n_r);
- подпоследовательности для реализаций разделяются на подпоследовательности для элементов реализации (обозначим их длину как n_l).

Таким образом, предлагаемая методика позволяет эффективно организовать проведение коррелированных расчетов при моделировании деревьев частиц с учетом архитектуры гибридных вычислительных систем. Отметим, что эту методику можно также использовать при моделировании на массивно-параллельных вычислительных системах.

Подчеркнем, что при моделировании элементов реализации дерева частиц, распределение которых сильно меняются при изменении параметров задачи, целесообразно использовать псевдослучайные числа из подпоследовательностей для элементов реализации, в частности, это важно при применении метода исключения [3].

4. Для практической реализации алгоритмов моделирования траекторий ветвящихся процессов предлагается использовать библиотеку PARMONC [5], предназначенную для запуска приложений распределенного численного статистического моделирования на массивно-параллельных и гибридных вычислительных системах с использованием сопроцессоров Intel Xeon Phi. В настоящее время библиотека установлена на вычислительных кластерах ЦКП «Сибирский суперкомпьютерный центр» СО РАН [6].

При использовании библиотеки PARMONC алгоритм, использующий методику крупноблочного распараллеливания, на массивно-параллельной вычислительной системе можно исполнять с помощью библиотечных процедур `parmonc1d` или `tparmonc1d`; методику комбинированного распараллеливания на гибридной вычислительной системе — с помощью процедур `parmonc1d_h` или `tparmonc1d_h` [7]. В каждой из выбранных процедур соответствующий вариант распределительного способа получения псевдослучайных чисел автоматически реализуется путем использования библиотечных функций `rnd128` (параллельный 128-битный генератор), `rnd128e1` (параллельный 128-битный генератор с выбором номера моделируемого элемента реализации). Библиотечные процедуры позволяют проводить коррелированные расчеты с использованием распределительного способа получения псевдослучайных чисел путем явного выбора пользователем номера подпоследовательности для вычислительных экспериментов.

4 Апробация предложенных методов распараллеливания при моделировании процесса развития электронных лавин в газе

1. Рассматриваемая ниже вероятностная модель процесса развития электронных лавин в газе описана в работе [8].

В ходе моделирования с катода из точки $r = (0, 0, 0)$ в момент времени $t = 0$ происходит эмиссия n_0 электронов с нулевыми энергиями. Траектории движения каждого из электронов прослеживаются либо до достижения времени t_{max} , либо до достижения уровня $z = d$; с этой целью делаются одинаковые шаги Δt по времени.

За один шаг по времени электрон с энергией T_{i-1} перемещается из точки r_{i-1} в точку r_i , где i — номер шага, а его координата и скорость $V_i = (V_x, V_y, V_z)$ изменяются по формулам

$$r_i = r_{i-1} + V_{i-1}\Delta t - eE(\Delta t)^2/(2m_0), \quad V_i = V_{i-1} - eE\Delta t/m_0, \quad (6)$$

где e — заряд электрона, m_0 — его масса; здесь учитывается ускорение электронов в электрическом поле.

В конце каждого временного шага разыгрывается столкновение с вероятностью

$$P = 1 - \exp\left(-\sigma_{tot}(T_i)N\Delta\ell\right), \quad (7)$$

где σ_{tot} — полное микроскопическое сечение взаимодействий, $\Delta\ell = |r_i - r_{i-1}|$. Затем разыгрывается тип взаимодействия в соответствии с заданными сечениями упругого рассеяния, возбуждения и ионизации; для этого используются сечения 24-х типов взаимодействий для азота. Параметры частицы после взаимодействия определяются в зависимости от его типа, при ионизации определяются параметры вторичной частицы [8].

В соответствии с описанными выше особенностями вычисления элементов реализации и значений функционалов, моделирование осуществляется по следующим шагам:

1. инициализация начального состояния лавины в момент времени $t = 0$: в магазин помещается n_0 частиц с параметрами $x = y = z = V_x = V_y = V_z = t = 0$;
2. выбор из магазина очередной частицы $\{x, y, z, V_x, V_y, V_z, t\}$; установка параметра $\tau = t$; если в магазине не осталось частиц — переход к п. 8;
3. вычисление новых координаты и скорости частицы согласно (6); установка параметра $\tau = \tau + \Delta t$; если $\tau \geq t_{max}$ или $z \geq d$ — переход к п. 2;
4. розыгрыш столкновения с вероятностью (7);
5. розыгрыш типа взаимодействия в соответствии с заданными сечениями упругого рассеяния, возбуждения и ионизации; изменение скорости частицы в соответствие с типом взаимодействия; при рождении новой частицы — помещение ее в магазин;
6. вычисление функционалов в момент времени τ , при необходимости;
7. переход к п. 2;
8. окончание моделирования реализации.

2. Расчеты по методике комбинированного распараллеливания производились на кластере МВС-10П в МСЦ РАН, где на каждом вычислительном узле доступны два 8 ядерных процессора Intel Xeon E5-2690 (2.9 GHz) и два 61 ядерных сопроцессора Intel Xeon Phi SE10X (1.1 GHz) [9]. При вычислениях использовалась библиотека PARMONC [5] и модифицированная для сопроцессоров Intel Xeon Phi параллельная программа ELSHOW [8].

Физические параметры задачи: расстояние между катодом и анодом $d = 0.7$ см, давление газа $P = 300$ Торр, z -компонента напряженности электрического поля $E_z = -1.65 \cdot 10^5$ В/см; параметры для моделирования деревьев: $n_0 = 1$, $\Delta t = 10^{-7}$ нс. Рассматривались значения $t_{max} = 4 \cdot 10^{-11}$ нс и $t_{max} = 6 \cdot 10^{-11}$ нс. Функционалы оценивались в 10 узлах по времени с шагом $0.1 \cdot t_{max}$.

Особенности моделирования одной реализации лавины по методике мелкоблочного распараллеливания:

- моделирование на основном ядре производилось до накопления примерно 620 частиц (п. 1 алгоритма мелкоблочного распараллеливания); это занимало в среднем менее 1 с машинного времени;
- накопленные вторичные частицы (примерно 610 частиц) отправлялись на сопроцессор, около 10 частиц оставалось на основном ядре (п. 2 алгоритма мелкоблочного распараллеливания); магазины частиц располагались в общей для всех сопроцессоров области памяти;
- массивы с функционалами располагались на сопроцессоре в общей для всех его ядер области памяти (п. 3 алгоритма мелкоблочного распараллеливания);
- массивы с функционалами отправлялись с сопроцессора на основной процессор после каждой смоделированной реализации (п. 4 алгоритма мелкоблочного распараллеливания); объем пересылаемых данных составил примерно 14 Mb;
- метод «русской рулетки» не применялся.

На одном вычислительном узле сравнивались методики крупноблочного и комбинированного распараллеливания; для каждого сочетания методики распараллеливания и величины t_{max} оценивались значения функции $L = L(t_{max})$, где $L(t_{max})$ — среднее число реализаций лавины частиц, полученных к моменту машинного времени t_{max} . При крупноблочном распараллеливании использовались только 16 ядер двух

Таблица 1: Сравнение двух методик распараллеливания.

t_{max}	$4 \cdot 10^{-11}$ нс	$6 \cdot 10^{-11}$ нс
$\langle n(t_{max}) \rangle$	$\approx 3.3 \cdot 10^4$	$\approx 5.1 \cdot 10^6$
t_1	37.2 с	5697.1 с
$t_1^{(comb)}$	8.3 с	1291.3 с

основных процессоров: $M = 16$; при комбинированном распараллеливании помимо основного процессора использовались два сопроцессора: $M_1 = 14$, $M_2 = 2$, $N = 61$.

В таблице 1 представлены данные расчетов для двух значений t_{max} . Приводится среднее число частиц в лавине к моменту времени t_{max} ; среднее время моделирования одной реализации: t_1 — для крупноблочной методики распараллеливания, $t_1^{(comb)}$ — для комбинированной. Для каждой методики распараллеливания величина t_1 рассчитывалась по формуле $t_1 = T(L(t_{max}))/L(t_{max})$, где $T(L)$ — машинное время, затраченное на моделирование L реализаций.

Отметим что в формуле (3) для использованной при расчетах гибридной вычислительной системы $M_2/M_1 = 2/14 = 1/7$ и величина $\alpha \approx 0.4$ (т.е. ядро сопроцессора рассчитывает часть «дерева» примерно в 2.5 медленнее, чем основное ядро); практически достигнутая величина выигрыша составила примерно 4.4. Заметим, что при таких значениях параметров максимальная («нереальная») величина выигрыша равна $62 \cdot 0.4 \approx 25$.

Результаты проведенного сравнения показывают целесообразность применения методики комбинированного распараллеливания для численного статистического моделирования траекторий ветвящихся процессов.

Список литературы

- [1] Ермаков С. М., Некруткин В. В., Сипин А. С. Случайные процессы для решения классических уравнений математической физики. М.: Наука, 1984.
- [2] Кольчужкин А. М., Учайкин В. В. Введение в теорию прохождения частиц через вещество. М.: Атомиздат, 1978.
- [3] Михайлов Г. А., Войтишек А. В. Численное статистическое моделирование. Методы Монте-Карло. М.: Издательский центр «Академия», 2006. 368 с.
- [4] Марченко М. А., Михайлов Г. А. Распределенные вычисления по методу Монте-Карло // Автоматика и телемеханика. 2007. № 5. С. 157–170.
- [5] Marchenko M. A. PARMONC — A Software Library for Massively Parallel Stochastic Simulation // Lecture Notes in Computer Science. 2011. Vol. 6873. P. 302–315.
- [6] Центр коллективного пользования «Сибирский суперкомпьютерный центр» (ЦКП ССКЦ СО РАН): [Электронный ресурс]. Новосибирск. 2001–2016. <http://www.sccc.icmmg.nsc.ru>.
- [7] Марченко М. А. Документация для библиотеки PARMONC: [Электронный ресурс] // Центр коллективного пользования «Сибирский Суперкомпьютерный центр». <http://www2.sccc.ru/PPP/Mat-Libr/parmonc.pdf>.
- [8] Lotova G. Z., Marchenko M. A., Mikhailov G. A. et al. Numerical statistical modelling algorithms for electron avalanches in gases // Russian Journal of Numerical Analysis and Mathematical Modelling. 2014. Vol. 29, no. 4. P. 251–263.
- [9] Межведомственный суперкомпьютерный центр Российской академии наук (МСЦ РАН): [Электронный ресурс]. Москва. 2017. <http://www.jscc.ru>.

Михаил Александрович Марченко — к.ф.-м.н., заведующий лабораторией Института вычислительной математики и математической геофизики СО РАН;
e-mail: marchenko@sccc.ru.

Дата поступления — 31 мая 2017 г.