

УДК 51-73

## РЕШЕНИЕ КИНЕТИЧЕСКОГО УРАВНЕНИЯ БОЛЬЦМАНА С ПОМОЩЬЮ ТЕТРАЭДРИЧЕСКИХ СЕТОК НА КЛАСТЕРНОЙ АРХИТЕКТУРЕ

Ю. Ю. Клосс<sup>1</sup>, Д. В. Мартынов<sup>2</sup>

Рассматривается способ решения кинетического уравнения Больцмана консервативным проекционным методом на многопроцессорной вычислительной системе с помощью технологии MPI. Для дискретизации координатного пространства используются тетраэдрические сетки. Приводится сравнение эффективности алгоритмов разбиения сетки между процессами. Построены графики зависимости роста производительности от числа потоков для схем первого и второго порядков точности.

**Ключевые слова:** уравнение Больцмана, проекционный метод, MPI, тетраэдрические сетки.

**1. Введение.** Кинетическое уравнение Больцмана описывает статистическое распределение молекул по скоростям в разреженном газе при заданных начальных и граничных условиях. Метод численного решения уравнения Больцмана приведен в работе [1] и носит название консервативного проекционного метода. Он применяется для вычисления интеграла столкновений в проблемно-моделирующей среде [2–5], разрабатываемой с 2002 г.

Неизвестной величиной в уравнении Больцмана является функция распределения по скоростям. Ее эволюция моделируется в процессе решения. Макропараметры газа, такие как плотность, температура, тензор давлений, потоки скорости и тепла, являются интегралами функции распределения, в общем случае зависящей от семи переменных: времени, трех координат в физическом пространстве и трех координат в скоростном пространстве. Процесс моделирования сложных технических устройств на одном процессоре, даже при условии, что оперативной памяти достаточно, может занять недели или месяцы. В таких случаях необходимо решать кинетическое уравнение Больцмана на многопроцессорных вычислительных системах.

Использование кластерной архитектуры должно быть эффективным и значительно ускорять работу проблемно-моделирующей среды. Для этого необходим алгоритм распараллеливания, способный при увеличении числа процессоров обеспечивать рост производительности, близкий к линейному.

В настоящей статье приводится алгоритм распараллеливания решения кинетического уравнения Больцмана и строится график зависимости роста производительности от числа работающих процессоров, которые связаны высокоскоростной сетью и обмениваются сообщениями с помощью технологии MPI.

**2. Численный метод решения уравнения Больцмана.** В случае одноатомного газа кинетическое уравнение Больцмана имеет вид

$$\frac{\partial f}{\partial t} + \xi \frac{\partial f}{\partial x} = I,$$

где  $I$  — пятимерный интеграл столкновений и  $f(t, x, \xi)$  — функция распределения, зависящая от времени  $t$ , трехмерной координаты  $x$  и трехмерной скорости  $\xi$ .

Для численного решения уравнения Больцмана необходимо перейти от непрерывной функции распределения к дискретной. В скоростном пространстве согласно проекционному методу [1] строится равномерная сетка, и для каждого узла скоростной сетки  $\xi_\gamma$  уравнение Больцмана переписывается в виде

$$\frac{\partial f_\gamma}{\partial t} + \xi_\gamma \frac{\partial f_\gamma}{\partial x} = I_\gamma(x, t).$$

Физическое пространство разбивается на малые объемы тетраэдрической формы, а функция распределения в каждом тетраэдре считается постоянной. Эффективным методом решения этой системы является метод расщепления по физическим процессам [6]. На малом по сравнению со средним временем

<sup>1</sup> Российский научный центр “Курчатовский институт” (РНЦ КИ), пл. Академика Курчатова, д. 1, 123182, Москва; Московский физико-технический институт (МФТИ), Институтский переулок, д. 9, 141700, Московская обл., г. Долгопрудный; нач. лаборатории, доцент, e-mail: kloss@mnpt.kiae.ru

<sup>2</sup> Московский физико-технический институт (МФТИ), Институтский переулок, д. 9, 141700, Московская обл., г. Долгопрудный; студент, e-mail: mart1989@list.ru

свободного пробега молекул шаге по времени  $\tau = t^{n+1} - t^n$  осуществляется расщепление на две стадии: расчет переноса молекул без соударений и расчет упругих столкновений. Обозначив функцию распределения в  $i$ -й ячейке координатного пространства после первого этапа через  $f_{i\gamma}^*$ , после второго этапа через  $f_{i\gamma}^{n+1}$  и решение на предыдущем временном слое через  $f_{i\gamma}^n$ , получим последовательность уравнений

$$\frac{f_{i\gamma}^* - f_{i\gamma}^n}{\tau} = \frac{1}{V_i} \sum_{k=1}^K (\xi_\gamma, n_{ik}) f_{ik\gamma}^n S_{ik}, \quad \frac{f_{i\gamma}^{n+1} - f_{i\gamma}^*}{\tau} = I_{i\gamma},$$

в которой  $n_{ik}$  — единичная нормаль к общей грани тетраэдров  $i$  и  $k$ ,  $S_{ik}$  — площадь грани и  $V_i$  — объем тетраэдра  $i$ .

Для определения дискретных значений  $I_{i\gamma}$  интеграла столкновений используется консервативный проекционный метод [1]. Основная идея метода заключается в рассмотрении столкновений двух молекул с определенными скоростями, прицельным расстоянием и азимутальным углом. С помощью законов кинематики вычисляются скорости после столкновения, которые в общем случае не попадают на построенную скоростную сетку. Значения физических величин, зависящих от скоростей после столкновения, рассчитываются с помощью степенной интерполяции по двум соседним скоростным узлам, которая устроена таким образом, чтобы выполнялись законы сохранения вещества, импульса и энергии и не нарушалось термодинамическое равновесие. После рассмотрения каждого столкновения вносятся соответствующие изменения в функцию распределения.

Ключевым при параллельных вычислениях свойством консервативного проекционного метода является изменение функции распределения только в рассматриваемом тетраэдре  $i$ . При координатном разбиении расчетной области каждый процесс рассчитывает столкновения молекул в отведенных ему тетраэдрах без дополнительной информации от других процессов, что приводит к эффективному распараллеливанию метода. Основная проблема состоит в организации параллельного решения уравнения переноса.

Для численного решения уравнения переноса необходимо для

каждой ячейки координатного пространства определить  $f_{ik\gamma}^n$  — функцию распределения либо на границе двух тетраэдров, либо на границе тетраэдра со стенкой. Для вычисления  $f_{ik\gamma}^n$  необходимо знать функцию распределения в соседних к данному тетраэдру. В зависимости от метода вычисления  $f_{ik\gamma}^n$  численная схема может менять порядок точности так, как показано на рис. 1.

При использовании схемы первого порядка точности для определения  $f_{ik\gamma}$  задействуются только два тетраэдра  $i$  и  $k$ :  $f_{ik\gamma} = \begin{cases} f_{i\gamma}, & (n_{ik}, \xi_\gamma) < 0, \\ f_{k\gamma}, & (n_{ik}, \xi_\gamma) > 0. \end{cases}$

При расчете с помощью разностной схемы второго порядка точности для определения  $f_{ik\gamma}$  необходимы функции распределения в четырех тетраэдрах  $i, k, a, b$  или  $i, k, c, d$  в зависимости от направления скорости  $\xi_\gamma$ :  $f_{ik\gamma} = \begin{cases} f_{i\gamma} + 0.5 \nabla f_{i\gamma} r_{ik}, & (n_{ik}, \xi_\gamma) < 0, \\ f_{k\gamma} + 0.5 \nabla f_{k\gamma} r_{ki}, & (n_{ik}, \xi_\gamma) > 0. \end{cases}$

Здесь  $r_{ik} = -r_{ki}$  — вектор, соединяющий центры тетраэдров  $i$  и  $k$ ;  $\nabla f_{i\gamma}, \nabla f_{k\gamma}$  — градиенты функции распределения в тетраэдрах  $i$  и  $k$  (для их определения и нужны значения функций распределения в дополнительных тетраэдрах  $a, b, c, d$ ).

Для расчета упругих столкновений применяется проекционных метод [1], знание функции распределения в соседних ячейках не требуется.

Итак, для численного решения уравнения Больцмана относительно функции распределения некоторого тетраэдра необходимо знать функцию распределения в соседних тетраэдрах при использовании

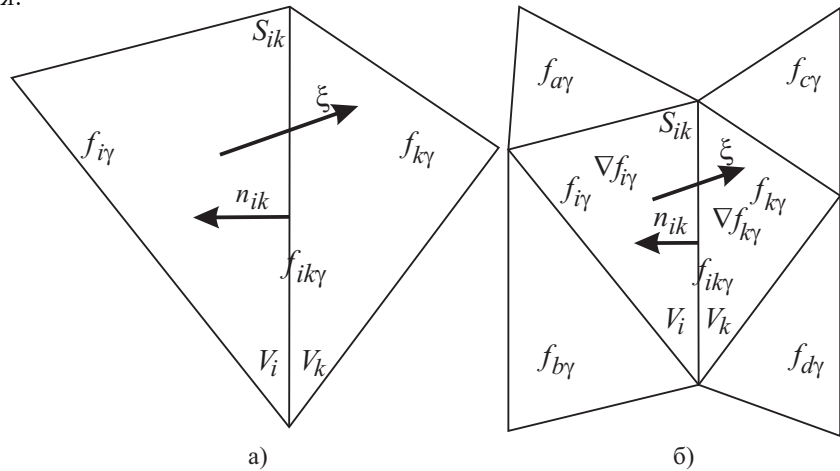


Рис. 1. Решение уравнения переноса: а) схема первого порядка точности; б) схема второго порядка точности

схемы первого порядка точности, а также функцию распределения во всех соседних к соседним тетраэдрам при использовании разностной схемы второго порядка точности. Этот вывод будет использован при разработке параллельного алгоритма проблемно-моделирующей среды.

**3. Параллельный алгоритм.** При разработке параллельного алгоритма численного решения уравнения Больцмана в основе лежали следующие предположения:

- 1) на одном вычислительном ядре работает один процесс;
- 2) все вычислительные ядра идентичны;
- 3) все вычислительные ядра связаны одинаковой сетью;
- 4) для одной итерации решения уравнения Больцмана необходимо одинаковое время для всех узлов пространственной сетки.

Для распараллеливания работы проблемно-моделирующей среды необходимо, чтобы каждый процесс  $p_i$ ,  $i = 1, \dots, n$ , где  $n$  — число процессов, проводил вычисления на своем множестве тетраэдров  $X_i$ ,  $X_\alpha \cap X_\beta = 0$ ,  $i, \alpha, \beta = 1, \dots, n$ ,  $\alpha \neq \beta$ , а в сумме эти множества должны образовывать всю координатную сетку  $X = \bigcup_i X_i$ . Если число тетраэдров в координатной сетке равно  $G = |X|$ , то для эффективного

распараллеливания необходимо, чтобы выполнялось условие  $\left\lceil \frac{G}{n} \right\rceil \leq |X_i| \leq \left\lfloor \frac{G}{n} \right\rfloor + 1$ .

При решении уравнения переноса необходимо знание функции распределения в соседних узлах. Обозначим через  $N$  оператор, который действует на некоторое множество тетраэдров  $\Psi$  так, что полученное множество  $N\Psi$  состоит из тетраэдров, являющихся соседями к тетраэдрам из множества  $\Psi$ . Для аппроксимации уравнения переноса с помощью схемы первого порядка на множестве  $\Psi$  необходимо знать функцию распределения в тетраэдрах из множества  $N\Psi \cup \Psi$ . Для достижения второго порядка точности необходима функция распределения в тетраэдрах из множества  $NN\Psi$ . Отметим, что  $NN\Psi \supseteq \Psi$ , поэтому  $NN\Psi \cup \Psi = NN\Psi$ . Таким образом, для решения уравнения переноса процесс  $p_i$  должен получить значение функции распределения в тетраэдрах из множества  $I_i = NX_i \cup X_i - X_i$  в случае схемы первого порядка точности и  $I_i = NNX_i \cup X_i - X_i$  в случае второго порядка точности от других процессов в виде MPI-сообщений.

За одну итерацию численного решения уравнения Больцмана процесс  $p_i$  рассчитывает перенос вещества и упругие столкновения на множестве тетраэдров  $X_i$ . Для этого процесс  $p_i$  принимает от остальных процессов  $p_j$ ,  $j = 1, \dots, i-1, i+1, \dots, n$ , функцию распределения в тетраэдрах из множества и посылает ему функцию распределения в тетраэдрах  $I_j \cap X_i$ . Этот процесс изображен на рис. 2.

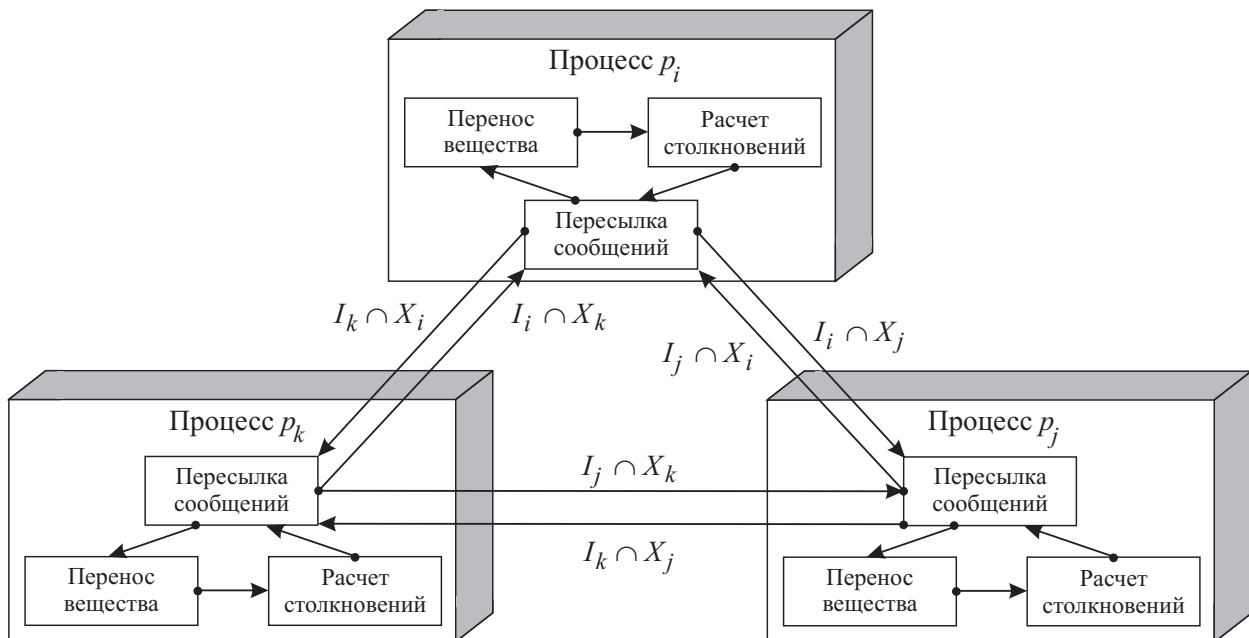


Рис. 2. Взаимодействие процессов

Процесс взаимодействия организован с помощью неблокируемых функций `MPI_Irecv` и `MPI_Isend`, которые синхронизируются с помощью функции `MPI_Waitall()`. С помощью функции `MPI_Irecv` процесс получает функцию распределения в тетраэдрах из множества  $I_i$ , а с помощью функции `MPI_Isend`

посылает другим процессам функцию распределения в тетраэдрах из множества  $O_i = \bigcup_{i \neq k} (I_k \cap X_i)$ .

При работе  $n$  процессов скорость работы программы увеличивается в  $a$  раз по сравнению с работой одного процесса. Оценим  $a$ . Время выполнения одного шага одним процессом равно  $t_1 = G(t_T + t_I)$ , где  $t_T$  — время решения уравнения переноса в одном тетраэдре и  $t_I$  — время расчета упругих столкновений в одном тетраэдре.

Время выполнения одного шага  $n$  процессами равно  $t_n = \max_{i=1, \dots, n} t_{p_i} = |X_i|(t_T + t_C) + |I_i|B/V$ , где  $B$  — размер функции распределения внутри одного тетраэдра и  $V$  — скорость передачи данных по сети.

Ускорение выполнения одного шага определяется через отношение времен:

$$a = \frac{t_1}{t_n} = \frac{G(t_T + t_C)}{\max_{i=1, \dots, n} (|X_i|(t_T + t_C) + |I_i|B/V)} = n \frac{1}{\max_{i=1, \dots, n} (|X_i|n/G + |I_i|B/(V|X_i|(t_T + t_C)))}$$

Полученная формула показывает, что сетка должна быть разбита поровну между процессами, т.е. должно выполняться условие  $\left\lfloor \frac{G}{n} \right\rfloor \leq |X_i| \leq \left\lfloor \frac{G}{n} \right\rfloor + 1$ . В таком случае формула упрощается, получаем

$$a = n \frac{1}{\max_{i=1, \dots, n} (1 + |I_i|nB/(VG(t_T + t_C)))}$$

Величины  $n, B, V, G$  являются константами, следовательно, для уменьшения времени пересылки сообщений необходимо разбивать координатную сетку так, чтобы число тетраэдров, функция распределения которых передается или принимается, должно было минимальным.

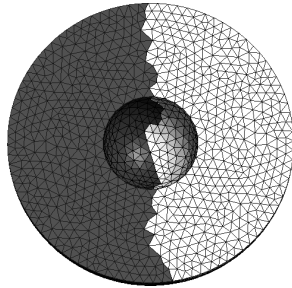
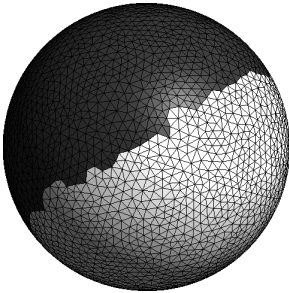


Рис. 3. Сетка в модели двух концентрических сфер, разбитая на две части алгоритмом Spectral

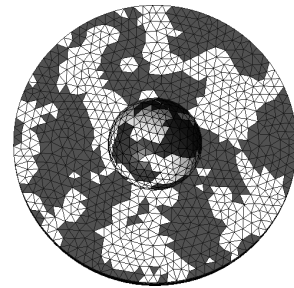


Рис. 4. Разбиение расчетной области на две части с помощью алгоритма Scattered

**4. Сравнение алгоритмов разбиения.** Оптимальный алгоритм разбиения сетки должен удовлетворять следующим условиям.

1. Все процессы решают уравнение Больцмана в одинаковом числе тетраэдров:

$$\left\lfloor \frac{G}{n} \right\rfloor \leq |X_i| \leq \left\lfloor \frac{G}{n} \right\rfloor + 1.$$

2. Число передаваемых и получаемых тетраэдров должно быть минимальным:  $|I_i| \rightarrow \min, |O_i| \rightarrow \min$ .

В табл. 1 приведено сравнение нескольких алгоритмов разбиения тетраэдрической сетки [7], построенной в модели двух концентрических сфер (рис. 3), на 8 частей.

Из табл. 1 можно сделать вывод, что все алгоритмы деления, за исключением алгоритма K-way, делят расчетную область оптимальным образом:  $\left\lfloor \frac{G}{n} \right\rfloor \leq |X_i| \leq \left\lfloor \frac{G}{n} \right\rfloor + 1, i = 1, \dots, n$ . В табл. 2 приведено количество тетраэдров, пересылаемых и получаемых процессами при решении уравнения переноса с помощью схемы первого порядка точности.

В табл. 3 приведено количество тетраэдров, пересылаемых и получаемых процессами при решении уравнения переноса с помощью схемы второго порядка точности.

Из табл. 2 и 3 можно сделать вывод, что алгоритмы Linear, Random и Scattered разбивают расчетную область так, что для каждого процесса  $p_i$  число тетраэдров из множества  $I_i$  и  $O_i$  приблизительно в 10 раз больше, чем число тетраэдров из тех же множеств при разбиении расчетной области алгоритмами Recursive, K-way, Multilevel Kernighan–Lin и Spectral. Это связано с тем, что алгоритмы Linear, Random и Scattered не группируют тетраэдры при разбиении, как показано на рис. 4.

Следовательно, оптимальными алгоритмами разбиения являются Recursive, Multilevel Kernighan–Lin и Spectral. Они разбивают расчетную область согласно условию  $\left\lfloor \frac{G}{n} \right\rfloor \leq |X_i| \leq \left\lfloor \frac{G}{n} \right\rfloor + 1, i = 1, \dots, n$ , и

Таблица 1

	Recursive	K-way	Multilevel-KL	Spectral	Linear	Random	Scattered
$G$	120463						
$G/n$	15057.875						
$ X_1 $	15058	15067	15058	15058	15058	15058	15058
$ X_2 $	15058	15084	15058	15058	15058	15058	15058
$ X_3 $	15058	15507	15058	15058	15058	15058	15058
$ X_4 $	15058	14853	15058	15058	15058	15058	15058
$ X_5 $	15058	14993	15058	15058	15058	15058	15058
$ X_6 $	15058	14825	15058	15058	15058	15058	15058
$ X_7 $	15058	14988	15058	15058	15058	15058	15058
$ X_8 $	15057	15146	15057	15057	15057	15057	15057
$ X_i _{\min}$	15057	14825	15057	15057	15057	15057	15057
$ X_i _{\max}$	15058	15507	15058	15058	15058	15058	15058

Таблица 2

	Recursive	K-way	Multilevel-KL	Spectral	Linear	Random	Scattered
$G$	120463						
$G/n$	15057.875						
$ I_1 $	954	1024	1473	1570	11041	11297	11726
$ O_1 $	966	1017	1462	1567	11019	11290	11722
$ I_2 $	1189	1006	1028	1111	11297	11704	11304
$ O_2 $	1172	1007	1030	1100	11302	11728	11294
$ I_3 $	1212	1512	956	1321	10407	11712	10403
$ O_3 $	1242	1523	951	1322	10385	11672	10448
$ I_4 $	1606	1424	1550	937	11425	11452	9852
$ O_4 $	1641	1410	1545	933	11415	11459	9827
$ I_5 $	1082	972	1076	977	11079	11602	11557
$ O_5 $	1034	972	1101	986	11068	11582	11506
$ I_6 $	1375	1435	1485	1447	11832	9974	11761
$ O_6 $	1370	1436	1473	1469	11881	9998	11765
$ I_7 $	1008	996	1206	968	10735	10473	12004
$ O_7 $	1001	996	1205	960	10706	10472	12054
$ I_8 $	1512	1452	1153	1468	10088	11831	11645
$ O_8 $	1512	1460	1160	1481	10128	11844	11636
$\sum_i  I_i $	9938	9821	9927	9818	87904	90045	90252
$\sum_i  O_i $	9938	9821	9927	9818	87904	90045	90252

содержат меньше всего тетраэдров в множествах  $I_i$  и  $O_i$ , т.е. при использовании таких алгоритмов будет тратиться меньше рабочего времени на пересылку функции распределения в виде MPI-сообщений.

**5. Эффективность параллельного алгоритма.** Для проверки эффективности описанного параллельного алгоритма вычисления в одной и той же расчетной области проводились на переменном числе ядер. Рассматривались две концентрические сферы; координатная сетка между ними строилась настолько подробной, чтобы задача еще могла считаться без использования виртуальной памяти на одном процес-

Таблица 3

	Recursive	K-way	Multilevel-KL	Spectral	Linear	Random	Scattered
$G$	120463						
$G/n$	15057.875						
$ I_1 $	2185	2055	3092	3172	24656	27417	22330
$ O_1 $	2130	2091	3120	3117	25425	26769	23538
$ I_2 $	1934	2039	2084	1878	23491	26639	23589
$ O_2 $	1978	2030	2033	1900	24384	26818	24738
$ I_3 $	2015	2904	2965	2632	27328	26632	25878
$ O_3 $	2032	2829	2973	2666	24796	26121	25890
$ I_4 $	2403	3044	1929	2222	25906	26115	27934
$ O_4 $	2385	3097	1938	2198	25660	26091	27291
$ I_5 $	2475	1977	2182	2925	25159	26304	26912
$ O_5 $	2500	2006	2199	2986	25342	26091	26591
$ I_6 $	2818	2948	3030	1948	27539	26553	26849
$ O_6 $	2786	2941	2987	1936	26915	26400	26326
$ I_7 $	3093	2015	2433	1996	24297	22730	27148
$ O_7 $	3087	1988	2434	1965	24609	23434	27045
$ I_8 $	3276	2963	2343	2951	22716	23884	27222
$ O_8 $	3301	2963	2374	2956	23961	24997	26443
$\sum_i  I_i $	20199	19945	20058	19724	201092	207077	207862
$\sum_i  O_i $	20199	19945	20058	19724	201092	207077	207862

соре с объемом оперативной памяти 8 Гб. Общее число тетраэдров в сетке получилось равным 88 000. Задача считалась с использованием  $n$  процессов,  $n \in [1, 100]$ . На рис. 5 приведены графики зависимости ускорения счета  $a = t_1/t_n$  в зависимости от числа процессов  $n$  в случае решения уравнения переноса с помощью схемы первого порядка точности (а) и схемы второго порядка точности (б). Расчетная область разбивалась с помощью алгоритма Spectral.

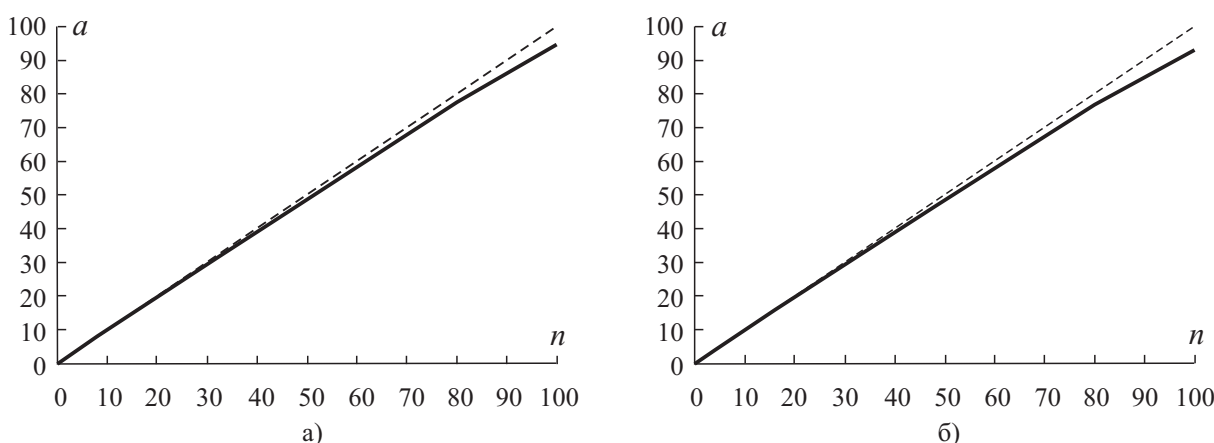


Рис. 5. Зависимость ускорения работы программы от числа работающих процессов

Пунктирной линией на графиках показана зависимость  $a(n) = n$ , т.е. максимально возможное увеличение производительности, если пренебречь влиянием кэша. Из рис. 5 видно, что полученные кривые  $a(n)$  как в случае использования схемы первого порядка точности, так и в случае использования схемы второго порядка точности дают существенные отклонения от кривой  $a(n) = n$ : для схемы первого порядка

точности  $a(100) = 95$ , для схемы второго порядка точности  $a(100) = 93$ . Это объясняется тем, что если взять  $n = 100$  при рассмотрении данной компьютерной модели, то число тетраэдров, в которых каждый процесс решает уравнение Больцмана, равно  $|X_i| \approx 880$ ,  $i = 1, \dots, 100$ . При этом число тетраэдров, из которых принимается и отправляется функция распределения, равно  $|I_i| \approx 200$  и  $|O_i| \approx 200$ ,  $i = 1, \dots, 100$ , для схемы первого порядка точности и  $|I_i| \approx 400$  и  $|O_i| \approx 400$ ,  $i = 1, \dots, 100$ , для схемы второго порядка точности. Таким образом, пересылка сообщений между процессами занимает значительное время и отрицательным образом влияет на производительность.

При моделировании прикладных устройств каждый процесс распараллеливания строится так, чтобы каждый вычислительный узел решал уравнение примерно на 3000–5000 тетраэдрах. Для рассмотренного примера это соответствует числу процессоров  $n = 16 \div 24$ . В этом диапазоне полученные зависимости увеличения производительности от числа процессов отклоняются от прямой  $a(n) = n$  менее, чем на 1%.

**Заключение.** В настоящей статье приведен алгоритм распараллеливания работы проблемно-моделирующей среды, решающей кинетическое уравнение Больцмана на тетраэдрических сетках. Сформулированы предположения, при которых строился алгоритм распараллеливания, показан процесс взаимодействия процессов и содержание пересылаемых сообщений для схем первого и второго порядка точности.

Сравнивались семь алгоритмов разбиения координатной сетки между процессами, сформулированы критерии оптимального алгоритма. На равные части координатную сетку разбивают все рассмотренные алгоритмы, за исключением алгоритма K-way. При использовании алгоритмов Linear, Random и Scattered приходится пересылать большие объемы данных через сообщения между процессами, поэтому на практике эти алгоритмы не применяются. Разбиение сетки алгоритмами Recursive, Multilevel Kernighan–Lin и Spectral привело примерно к одинаковым показателям согласно сформулированным критериям. Эти алгоритмы используются при моделировании прикладных устройств.

Эффективность алгоритма распараллеливания оценивалась с помощью графика зависимости ускорения работы проблемно-моделирующей среды от числа процессов, решающих задачу. В случае использования как схемы первого, так и схемы второго порядка точности график получился близким к линейному. Это доказывает пригодность алгоритма к эффективному применению в расчетах. Описанный алгоритм распараллеливания был успешно применен при исследовании термомолекулярных насосов Кнудсена, описанных в работах [8–10].

#### СПИСОК ЛИТЕРАТУРЫ

1. Черемисин Ф.Г. Консервативный метод вычисления интеграла столкновений Больцмана // Доклады РАН. 1997. **357**, №1. 53–56.
2. Додулад О.И., Клосс Ю.Ю., Мартынов Д.В., Rogozin O.A., Рябченков В.В., Черемисин Ф.Г., Шувалов П.В. Проблемно-моделирующая среда для расчета и анализа газокинетических процессов // Нано- и микросистемная техника. 2011. № 2. 12–17.
3. Клосс Ю.Ю., Черемисин Ф.Г., Шувалов П.В. Разработка программного солвера для решения задач динамики разреженного газа в кластерной архитектуре // Вестн. компьютерных и информационных технологий. 2011. № 5. 21–26.
4. Клосс Ю.Ю., Мартынов Д.В., Черемисин Ф.Г. Проблемно-моделирующая среда для решения кинетического уравнения Больцмана на тетраэдрических сетках // Вычислительные методы и программирование. 2011. **12**, № 1. 220–234.
5. Додулад О.И., Клосс Ю.Ю., Рябченков В.В., Черемисин Ф.Г. Система программных модулей для вычисления интеграла столкновений Больцмана // Вычислительные методы и программирование. 2011. **12**, № 1. 44–52.
6. Аристов В.В., Черемисин Ф.Г. Расщепление неоднородного кинетического оператора уравнения Больцмана // Докл. АН СССР. 1976. **231**, № 1. 49–52.
7. Hendrickson B., Leland R. The Chaco user's guide: version 2.0. Sandia Tech. Report SAND94–2692, 1994.
8. Клосс Ю.Ю., Мартынов Д.В., Черемисин Ф.Г. Компьютерное моделирование и анализ технических характеристик термомолекулярных микронасосов // Ж. техн. физики. 2011. **81**, № 7. 141–148.
9. Клосс Ю.Ю., Мартынов Д.В., Черемисин Ф.Г. Разработка методов компьютерного моделирования и анализа микронасоса Кнудсена // Информационные технологии. 2010. № 10. 30–35.
10. Аникин Ю.А., Клосс Ю.Ю., Мартынов Д.В., Черемисин Ф.Г. Компьютерное моделирование и анализ эксперимента Кнудсена 1910 года // Нано- и микросистемная техника. 2010. № 8. 6–14.

Поступила в редакцию  
23.01.2012