

УДК 532.5+519.6

ТРЕХМЕРНОЕ МОДЕЛИРОВАНИЕ ДИНАМИКИ ДЕФОРМИРУЕМЫХ КАПЕЛЬ ЭМУЛЬСИИ МЕТОДОМ ГРАНИЧНЫХ ЭЛЕМЕНТОВ И БЫСТРЫМ МЕТОДОМ МУЛЬТИПОЛЕЙ НА ГЕТЕРОГЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

О. А. Абрамова¹, Ю. А. Иткулова¹, Н. А. Гумеров², И. Ш. Ахатов³

Прямое моделирование взаимодействия большого количества деформируемых капель необходимо для более точного предсказания реологических свойств и микроструктуры систем “жидкость–жидкость”. Рассмотрена математическая модель трехмерного течения смеси двух ньютоновских жидкостей каплевой структуры в неограниченной области при малых числах Рейнольдса. Разработан и реализован эффективный подход для расчета динамики большого количества деформируемых капель. Этот подход основан на методе граничных элементов для трехмерных задач, ускорение которого произведено как за счет высокоэффективного масштабируемого алгоритма (FMM), так и за счет использования гетерогенных вычислительных архитектур (многоядерные CPU и графические процессоры). Все это позволяет напрямую рассчитывать взаимодействие десятков тысяч деформируемых капель, что было подтверждено тестовыми и демонстрационными расчетами. Разработанный метод может быть использован для решения широкого класса задач, связанных с течениями эмульсий в микро- и наномасштабах, и для исследования микроструктуры течений.

Ключевые слова: деформируемые капли, уравнения Стокса, метод граничных элементов, быстрый метод мультиполей, параллельные вычисления, графические процессоры.

1. Введение. Изучение взаимодействия большого количества деформируемых капель необходимо для более точного предсказания реологических свойств и микроструктуры систем “жидкость–жидкость”, встречающихся в нефтяной, пищевой промышленности и других сферах экономики. Основной целью данной работы является разработка эффективного численного инструмента для исследования поведения трехмерных течений Стокса двухфазных жидкостей в случае, когда одна из жидкостей присутствует в другой в виде капель. В настоящей статье численная методика основывается на методе граничных элементов (МГЭ) с применением быстрого метода мультиполей (FMM – Fast Multipole Method) для случая произвольных деформируемых границ. Кроме того, реализованные алгоритмы были ускорены с использованием многоядерных CPU и графических процессоров (GPU).

Моделирование динамики капель одной жидкости в объеме другой жидкости для конечных чисел Рейнольдса может также проводиться с использованием методов, отличных от МГЭ. В статье [19] представлены результаты исследования течения большого количества деформируемых клеток крови в каналах различных форм методом спектральных элементов. В работах [1, 10] для решения уравнений Навье–Стокса, описывающих движение двух жидкостей, используется метод конечных разностей в комбинации с “front tracking method” для описания границы раздела между жидкостями. В цитируемых работах исследованы динамика периодической ячейки с 27 каплями и взаимодействие двух близко расположенных капель в сдвиговом потоке [1] и под действием силы тяжести [10] между двумя параллельными пластинами. Несмотря на преимущества использования полной системы уравнений движения жидкостей, применение этих методов ограничивает размер исследуемой области. Кроме того, МГЭ лучше описывает границу капель и позволяет детально исследовать их деформацию.

Метод граничных элементов для течений Стокса изложен в [11] и успешно применялся в ряде работ для расчета динамики и взаимодействия капель и частиц в дисперсных течениях. Некоторые результаты моделирования течений концентрированных эмульсий в сдвиговом потоке с использованием этого метода

¹ Башкирский государственный университет, Центр микро- и наномасштабной динамики дисперсных систем (ЦМНДДС, БашГУ); ул. Заки Валиди, 32, 450076, г. Уфа; О. А. Абрамова, стажер-исследователь, e-mail: abramovacmnds@gmail.com; Ю. А. Иткулова, стажер-исследователь, e-mail: Itkulova.Yulia@bashedu.ru

² Университет штата Мэриленд, США, Институт передовых компьютерных исследований (UMIACS), Room 3305, A.V. Williams Building, College Park, MD 20742; professor, e-mail: gumerov@umiacs.umd.edu

³ Университет штата Северная Дакота, США (NDSU), NDSU Dept 2490, 210 Dolve Hall, P.O. Box 6050, Fargo, ND 58108; professor, e-mail: Iskander.Akhatov@ndsu.edu

представлены в работах [9, 21]; кроме того, изучалось поведение эмульсий под действием силы тяжести [20]. Однако такие расчеты занимали существенное время для относительно небольшого количества капель. Это обусловлено тем, что МГЭ сводится к решению на каждом временном шаге плотной системы линейных алгебраических уравнений (СЛАУ) относительно $3N$ неизвестных, где N — число расчетных узлов. Для систем из тысяч капель, поверхность каждой из которых содержит сотни треугольных элементов, значение N может достигать миллиона. Прямые методы решения с вычислительной сложностью $O(N^3)$ в таких случаях неприменимы. Использование эффективных итеративных методов снижает вычислительную сложность до $O(N_{\text{iter}}N^2)$, где $N_{\text{iter}} \ll N$ — количество итераций и $O(N^2)$ — сложность одного умножения матрицы на вектор. Таким образом, для больших N решение СЛАУ происходит не так быстро, поскольку время вычислений возрастает пропорционально квадрату N . Основной особенностью реализованного численного подхода является применение алгоритма FMM, имеющего сложность $O(N)$ для заданной точности матрично-векторных произведений (МВП), возникающих при моделировании течений Стокса, что приводит к $O(N_{\text{iter}}N)$ вычислительной сложности всего алгоритма.

Метод FMM впервые был представлен в статье [3] для суммирования электростатического и гравитационного потенциалов (функция Грина для уравнения Лапласа) в двух и трех измерениях. В работе [15] впервые было предложено использование FMM для ускорения решения уравнений Стокса, однако был рассмотрен только случай твердых сферических частиц. Использование FMM в МГЭ для областей произвольной формы было опубликовано в [17]. В этой работе, как и в [16], проблема быстрого суммирования фундаментальных решений уравнений Стокса сводится к суммированию фундаментальных решений уравнения Лапласа. Отметим, что это более быстрый метод, чем “kernel-independent FMM” [2, 18], который также может применяться к решению уравнений Стокса. Последнее обстоятельство связано с более эффективной реализацией операторов трансляции для уравнений Лапласа (декомпозиция на вращения и коаксиальную трансляцию [4, 5]), чем через общее матричное представление [18]. Кроме того, в работе [22] МГЭ для системы капель был ускорен с помощью мультипольных разложений и операторов трансляций типичных для FMM, но иерархический FMM так и не был реализован.

Алгоритм FMM может быть достаточно эффективно распараллелен на гетерогенных вычислительных системах. Первая реализация FMM на графических процессорах была опубликована в статье [4], где было показано, что FMM для трехмерного уравнения Лапласа может быть ускорен в 30–60 раз и время счета одного МВП для системы с миллионом частиц составило 1 секунду на одном GPU. Этот подход получил дальнейшее развитие, и в работах [6, 7] авторы представили реализацию масштабируемого FMM на гетерогенном вычислительном кластере с несколькими узлами, каждый из которых оснащен многоядерными CPU и несколькими GPU. В этом случае МВП для миллиарда частиц занимало всего 10 секунд на 32 узлах. Демонстрация решения уравнений Стокса при помощи модификации метода [2] для течения крови (более 200 миллионов красных кровяных телец) на больших CPU- и CPU/GPU-кластерах представлена в [12]. В настоящей работе все расчеты проводятся на вычислительной машине с одной графической картой и используется CPU/GPU-параллелизм, что позволяет напрямую моделировать динамику системы с десятком тысяч деформируемых капель в стоковом потоке за разумное время.

Первая попытка авторов ускорить МГЭ с помощью GPU была предпринята в статье [24]. Несмотря на достигнутое ускорение, размер задач, решаемых в рамках этого подхода, ограничен в связи с нехваткой памяти и с тем, что сложность алгоритма составляла $O(N^2)$. Как упомянуто выше, применение FMM приводит к тому, что вычислительная сложность снижается до $O(N)$ и размер необходимой памяти становится пропорционален N . Новизна данной работы заключается в разработке и демонстрации алгоритма, отличного от [12], позволяющего рассчитывать движение большого количества деформируемых капель. Несмотря на то что составные части алгоритма (метод граничных элементов для капель произвольной формы в стоковом течении и быстрый метод мультиполей) были реализованы и протестированы в цитируемых работах, представленное в настоящей работе объединение этих методов, реализация на гетерогенной архитектуре и исследование производительности и возможностей, которые дает алгоритм для исследования капельных течений, произведено впервые.

2. Математическая модель. Рассматривается задача динамики деформируемых капель одной ньютоновской жидкости (индекс 2) в неограниченном потоке другой жидкости (индекс 1) при малых числах Рейнольдса. Схематичное изображение исследуемого процесса представлено на рис. 1.

Кроме того, аналогичная задача была решена для ограниченных областей. В этом случае соответствующие граничные условия поставлены на заданных границах [23]. Движение жидкостей описывается уравнениями Стокса

$$\nabla \cdot \boldsymbol{\sigma}_i = -\nabla p_i + \mu_i \nabla^2 \mathbf{u}_i = \mathbf{0}, \quad \nabla \cdot \mathbf{u}_i = 0, \quad i = 1, 2,$$

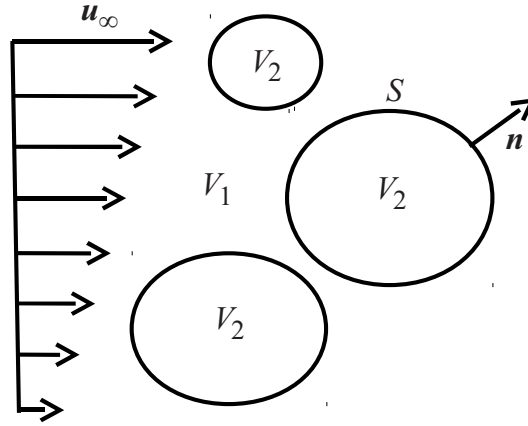


Рис. 1. Схематическое изображение исследуемой задачи: динамика капель в неограниченной области (обозначен линейный профиль сдвигового течения)

с условиями на границе раздела двух жидкостей S для скорости $\mathbf{u}$ и нормального напряжения $\mathbf{f} = \boldsymbol{\sigma} \cdot \mathbf{n}$:

$$\begin{aligned} \mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}, \quad \mathbf{f} = \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{f}_1 - \mathbf{f}_2 = \mathbf{f}\mathbf{n}, \\ \mathbf{f} = \gamma(\nabla \cdot \mathbf{n}) + (\rho_1 - \rho_2)(\mathbf{g} \cdot \mathbf{x}), \quad \mathbf{x} \in S. \end{aligned} \quad (1)$$

Здесь $\boldsymbol{\sigma}$ — тензор напряжений; ρ — плотность; μ — динамическая вязкость; $\mathbf{x}$ — радиус-вектор рассматриваемой точки; $\mathbf{g}$ — ускорение свободного падения; γ — коэффициент поверхностного натяжения; p — давление, которое включает в себя гидростатический компонент; $\mathbf{n}$ — нормаль к поверхности, направленная в жидкость 1; k — средняя кривизна поверхности. В случае неограниченной области для внешней жидкости задается условие на бесконечности: $\mathbf{u}_1(\mathbf{x}) \rightarrow \mathbf{u}_\infty(\mathbf{x})$.

Кинематическое условие, описывающее движение точек поверхности капель, записывается в виде

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}), \quad \mathbf{x} \in S. \quad (2)$$

3. Гранично-интегральная формулировка. Задача решается методом граничных элементов, который заключается в переходе от уравнений в частных производных, описывающих поведение неизвестной функции внутри и на границе области, к интегральному уравнению, связывающему только граничные значения, и в поиске численного решения этого уравнения. Далее значения искомой функции в произвольных точках расчетной области можно определить из интегрального уравнения с использованием найденных решений на границе. Таким образом, размерность задачи уменьшается на единицу, что делает МГЭ достаточно эффективным при моделировании трехмерных задач в бесконечных областях.

Интегральное представление поля скоростей внутри и вне капель может быть записано в следующем виде [13]:

$$\left. \begin{aligned} \mathbf{y} \in V_1, \quad \mathbf{u}(\mathbf{y}) - 2\mathbf{u}_\infty(\mathbf{y}) \\ \mathbf{y} \in V_2, \quad \lambda \mathbf{u}(\mathbf{y}) \\ \mathbf{y} \in S, \quad \frac{1+\lambda}{2} \mathbf{u}(\mathbf{y}) - \mathbf{u}_\infty(\mathbf{y}) \end{aligned} \right\} = \int_S \left\{ -\frac{1}{\mu_1} \mathbf{G}(\mathbf{y}, \mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) + (\lambda - 1) [\mathbf{T}(\mathbf{y}, \mathbf{x}) \cdot \mathbf{n}(\mathbf{x})] \cdot \mathbf{u}(\mathbf{x}) \right\} dS(\mathbf{x}). \quad (3)$$

Здесь V_1 и V_2 — объемы, занимаемые соответственно внешней и внутренней жидкостями, и $\mathbf{f}(\mathbf{x})$ определяется из уравнения (1). Нормаль в данном случае направлена из жидкости 2 в жидкость 1. Заметим, что скорость $\mathbf{u}(\mathbf{x})$ находится как решение последнего уравнения в (3), которое является интегральным уравнением Фредгольма 2-го рода с сингулярным ядром. Символами $\mathbf{G}$ и $\mathbf{T}$ обозначены тензоры второго ранга (стокслет) и третьего ранга (стресслет), компоненты которых в прямоугольной декартовой системе координат имеют вид

$$G_{ij}(\mathbf{y}, \mathbf{x}) = \frac{1}{8\pi} \left(\frac{\delta_{ij}}{r} + \frac{r_i r_j}{r^3} \right), \quad T_{ijk}(\mathbf{y}, \mathbf{x}) = -\frac{3}{4\pi} \frac{r_i r_j r_k}{r^5}, \quad r_i = y_i - x_i, \quad i, j, k = 1, 2, 3,$$

где δ_{ij} — символ Кронекера. Итак, если $\mathbf{u}$ и $\mathbf{f}$ известны на границе области, то поле скорости $\mathbf{u}(\mathbf{y})$ может быть определено в любой точке области.

Поверхность каждой капли дискретизируется треугольной сеткой. Для вычисления поверхностных интегралов используются квадратурные формулы [23]. Сингулярные интегралы определяются на основе известных интегральных тождеств для течений Стокса [13, 20, 23]. Для вычисления средней кривизны поверхности в каждом узле сетки применяется метод параболической аппроксимации [20, 23]. Методом коллокаций в узлах сетки последнее уравнение в граничных интегралах (3) сводится к линейной системе относительно неизвестных компонентов скорости на границе:

$$A \mathbf{x} = \mathbf{b}, \quad (4)$$

где A — расчетная матрица системы, $\mathbf{x}$ — вектор неизвестных и $\mathbf{b}$ — правая часть системы. Таким образом, при моделировании динамики эмульсии с общим числом узлов сетки на поверхности капель N требуется находить решение СЛАУ размером $3N \times 3N$ на каждом временном шаге.

Для стабилизации сетки в процессе моделирования динамики капель правая часть условия (2) модифицируется следующим образом:

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}) + \mathbf{w}(\mathbf{x}), \quad \mathbf{w} \cdot \mathbf{n} = 0, \quad \mathbf{x} \in S,$$

где $\mathbf{u}(\mathbf{x})$ — скорость жидкости, определяемая из решения эллиптической граничной задачи, описанной выше; $\mathbf{w}(\mathbf{x})$ — локально определяемая поправка, которую можно задавать произвольно для сохранения более или менее равномерного распределения узлов сетки на поверхности капель. Поскольку $\mathbf{w} \cdot \mathbf{n} = 0$, такая модификация изменяет только тангенциальную скорость движения узлов сетки вдоль поверхности и не влияет на представление всей поверхности этими точками. В статье [9] для $\mathbf{w}(\mathbf{x})$ предложена формула, которая в упрощенном виде (без коррекции для близлежащих капель) используется для тестовых расчетов, представленных в настоящей работе.

Интегрирование по времени проводится методом Адамса–Башфорта–Моултона (АБМ) 6-го порядка (предиктор–корректор), для которого значения функции на начальном отрезке времени находятся методом Рунге–Кутты 4-го порядка (РК4). Шаг по времени выбирается в соответствии с условием численной устойчивости (см. детали в [23]).

4. Гетерогенный быстрый метод мультиполей для уравнений Стокса. В случае небольшого числа капель при решении СЛАУ (4) применяются стандартные прямые методы, но при увеличении масштаба задачи их использование затрудняется. Это связано с тем, что размер необходимой памяти пропорционален квадрату числа узлов сетки. Кроме того, при их увеличении возрастает время вычислений, что делает невозможным расчеты для систем с количеством капель $M \geq 10^2$.

Эту проблему можно решить путем использования итерационных методов, которые существенно снижают затраты памяти и времени. Одними из наиболее эффективных методов решения СЛАУ являются методы проектирования на подпространства Крылова, в частности метод обобщенных минимальных невязок (GMRES — Generalized Minimal RESidual) [14], который и используется в данной работе. Известно, что сходимость метода GMRES можно достаточно эффективно ускорить за счет использования предобусловливателя, но в данной работе требуется всего 5–7 итераций для сходимости процесса с погрешностью 10^{-6} . Количество итераций может меняться в зависимости от распределения капель в пространстве и их расположения относительно друг друга, но не превышает 10 для представленных ниже тестовых расчетов.

Однако основную сложность составляет необходимость вычисления МВП на каждой итерации, что для СЛАУ большой размерности затруднительно. Применение FMM снижает вычислительную сложность МВП для плотных матриц с $O(N^2)$ до $O(N \log N)$ или $O(N)$. Кроме того, FMM является экономичным в плане использования памяти, поскольку нет необходимости хранить матрицу A .

4.1. Основы быстрого метода мультиполей. Метод FMM применяется для вычисления полей сил и потенциалов в задачах N тел, когда требуется вычислить все парные взаимодействия. Например, для суммирования фундаментальных решений уравнений Лапласа в трех измерениях этот метод широко используется в задачах астрофизики, молекулярной динамики, электростатики и др. Системная матрица

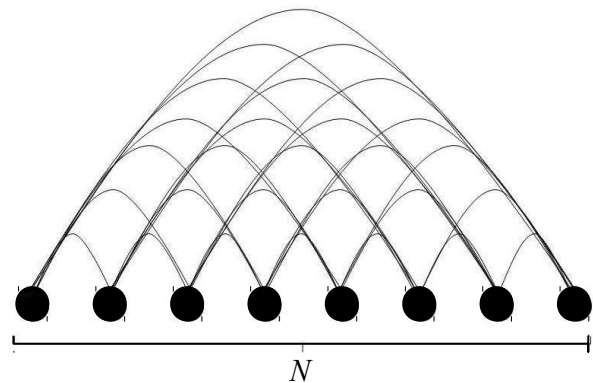


Рис. 2. Стандартный МГЭ-подход.
Вычислительная сложность $O(N^2)$

МГЭ представляет собой также матрицу “взаимодействия” каждого узла со всеми остальными узлами сетки (рис. 2).

Основная задача FMM — вычисление потенциалов, генерируемых множеством “источников”, которые оцениваются в множестве точек — “приемников”: $\phi_j = \phi(\mathbf{y}_j) = \sum_{i=1}^N q_i \Phi(\mathbf{y}_j, \mathbf{x}_i)$, $j = 1, \dots, M$, $\mathbf{x}_i, \mathbf{y}_j \in R^d$, где $\mathbf{x}_i$ — источники, $\mathbf{y}_j$ — приемники, ϕ_j — потенциал множества источников интенсивности q_i , d — размерность задачи, $\Phi(\mathbf{y}_j, \mathbf{x}_i)$ — некоторая функция, называемая ядром.

Кроме того, метод FMM может применяться для вычисления градиента потенциала в точках-приемниках:

$$\nabla \phi_j = \nabla_{\mathbf{y}} \phi(\mathbf{y})|_{\mathbf{y}=\mathbf{y}_j} = \sum_{i=1}^N q_i \nabla_{\mathbf{y}} \Phi(\mathbf{y}, \mathbf{x}_i)|_{\mathbf{y}=\mathbf{y}_j}.$$

Метод FMM используется при вычислении МВП для матриц специального вида, элементы которых находятся через потенциалы (ядра), и основывается на использовании иерархической структуры данных, математической теории разложения и трансляции функций. Основные идеи FMM можно описать следующим образом. Пусть функцию $\Phi(\mathbf{y}, \mathbf{x}_i)$ можно разложить в локальный (регулярный) или мультипольный (сингулярный) ряд следующим образом:

$$\begin{aligned} \Phi(\mathbf{y}, \mathbf{x}_i) &= \sum_{m=1}^{\infty} a_m(\mathbf{x}_i - \mathbf{x}_*) R_m(\mathbf{y} - \mathbf{x}_*) = \sum_{m=1}^{p^2} a_m(\mathbf{x}_i - \mathbf{x}_*) R_m(\mathbf{y} - \mathbf{x}_*) + \epsilon(p), \quad |\mathbf{y} - \mathbf{x}_*| < |\mathbf{x}_i - \mathbf{x}_*|, \\ \Phi(\mathbf{y}, \mathbf{x}_i) &= \sum_{m=1}^{\infty} b_m(\mathbf{x}_i - \mathbf{x}_*) S_m(\mathbf{y} - \mathbf{x}_*) = \sum_{m=1}^{p^2} b_m(\mathbf{x}_i - \mathbf{x}_*) S_m(\mathbf{y} - \mathbf{x}_*) + \epsilon(p), \quad |\mathbf{y} - \mathbf{x}_*| > |\mathbf{x}_i - \mathbf{x}_*|. \end{aligned}$$

Здесь $\mathbf{x}_*$ — центр разложения, R_m и S_m — базисные функции локального и мультипольного разложений соответственно, a_m , b_m — коэффициенты разложений, p — параметр усечения, p^2 — число элементов усеченного разложения потенциала по базису. В данной работе реализована многоуровневая версия FMM, в которой кроме построения S - и R -разложений используются также понятия операторов трансляций.

Под трансляцией понимается переход от представления функции в одной системе координат с центром в $\mathbf{x}_*$ к представлению функции в другой системе координат с центром в $\mathbf{x}_* + \mathbf{t}$. В FMM различают три типа операторов трансляций, относящихся к коэффициентам разложений в различных базисах:

$$\begin{aligned} R_m(\mathbf{y} - \mathbf{x}_* + \mathbf{t}) &= \sum_{l=1}^{p^2} (R|R)_{lm}(\mathbf{t}) R_l(\mathbf{y} - \mathbf{x}_*), \\ S_m(\mathbf{y} - \mathbf{x}_* + \mathbf{t}) &= \sum_{l=1}^{p^2} (S|S)_{lm}(\mathbf{t}) S_l(\mathbf{y} - \mathbf{x}_*), \\ S_m(\mathbf{y} - \mathbf{x}_* + \mathbf{t}) &= \sum_{l=1}^{p^2} (S|R)_{lm}(\mathbf{t}) R_l(\mathbf{y} - \mathbf{x}_*). \end{aligned}$$

Здесь $\mathbf{t}$ — вектор трансляции, $(R|R)_{lm}(\mathbf{t})$, $(S|S)_{lm}(\mathbf{t})$ и $(S|R)_{lm}(\mathbf{t})$ — коэффициенты $R|R$ (регулярное по регулярному базису), $S|S$ (сингулярное по сингулярному базису) и $S|R$ (сингулярное по регулярному базису) переразложений, образующие матрицы операторов трансляции $R|R$, $S|S$ и $S|R$ соответственно размером $p^2 \times p^2$.

Схема многоуровневого FMM и блок-схема алгоритма показаны на рис. 3 и 5. Кратко для трехмерного пространства его можно описать следующим образом. Пусть вся расчетная область со всеми точками (источниками и приемниками) содержится в единичном кубе, который разбивается как восьмеричное дерево. На первом уровне область будет разбита на 8 кубов (ячеек, боксов) и так далее до уровня $l_{\max}$ так, что на каждом уровне l дерева область будет содержать 8^l ячеек. Ячейки могут содержать источники, приемники или вообще быть пустыми, тогда они не учитываются в расчетах. Таким образом, FMM не зависит от распределения расчетных точек и не требует наличия регулярной сетки. Боксы, содержащие источники, образуют свою иерархию (слева на рис. 3), боксы с приемниками — свою (справа на рис. 3). Формируется структура данных для распределения по ячейкам расчетных точек на каждом уровне, списки соседей и т.д.

Алгоритм состоит из трех основных шагов (рис. 3 и 4):

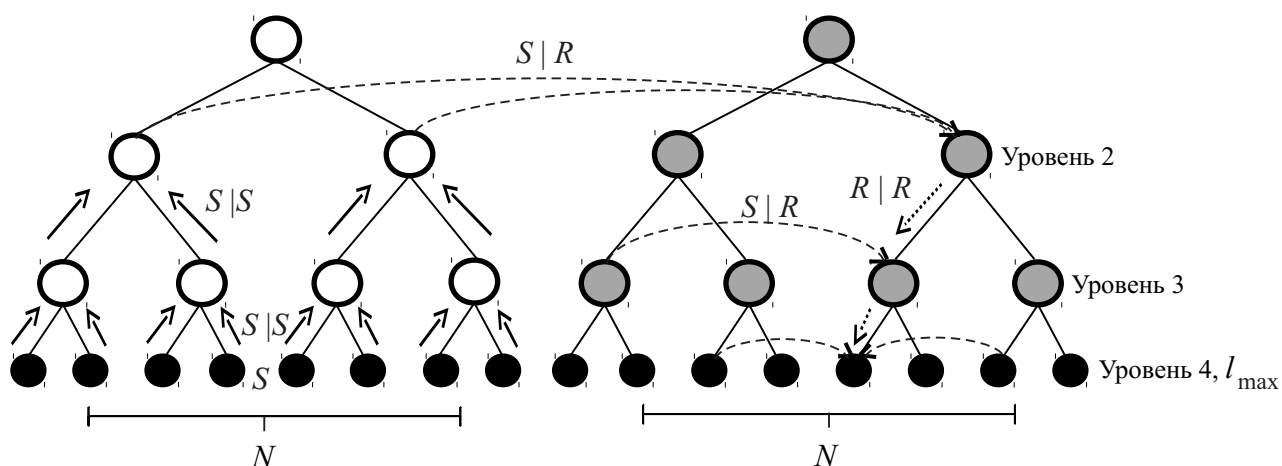


Рис. 3. Схема алгоритма многоуровневого FMM. Вычислительная сложность $O(N \log N)$

— восхождение (“upward pass”) от уровня $l_{\max}$ до $l = 2$: вычисление коэффициентов S -разложений на уровне $l_{\max}$ относительно центров боксов для каждого источника и суммирование для источников внутри каждого бокса (рис. 4а); полученные разложения переводятся с уровня на уровень до $l = 2$ с помощью $S|S$ -трансляций (рис. 4б); коэффициенты переразложений на каждом уровне относительно центров боксов хранятся в памяти;

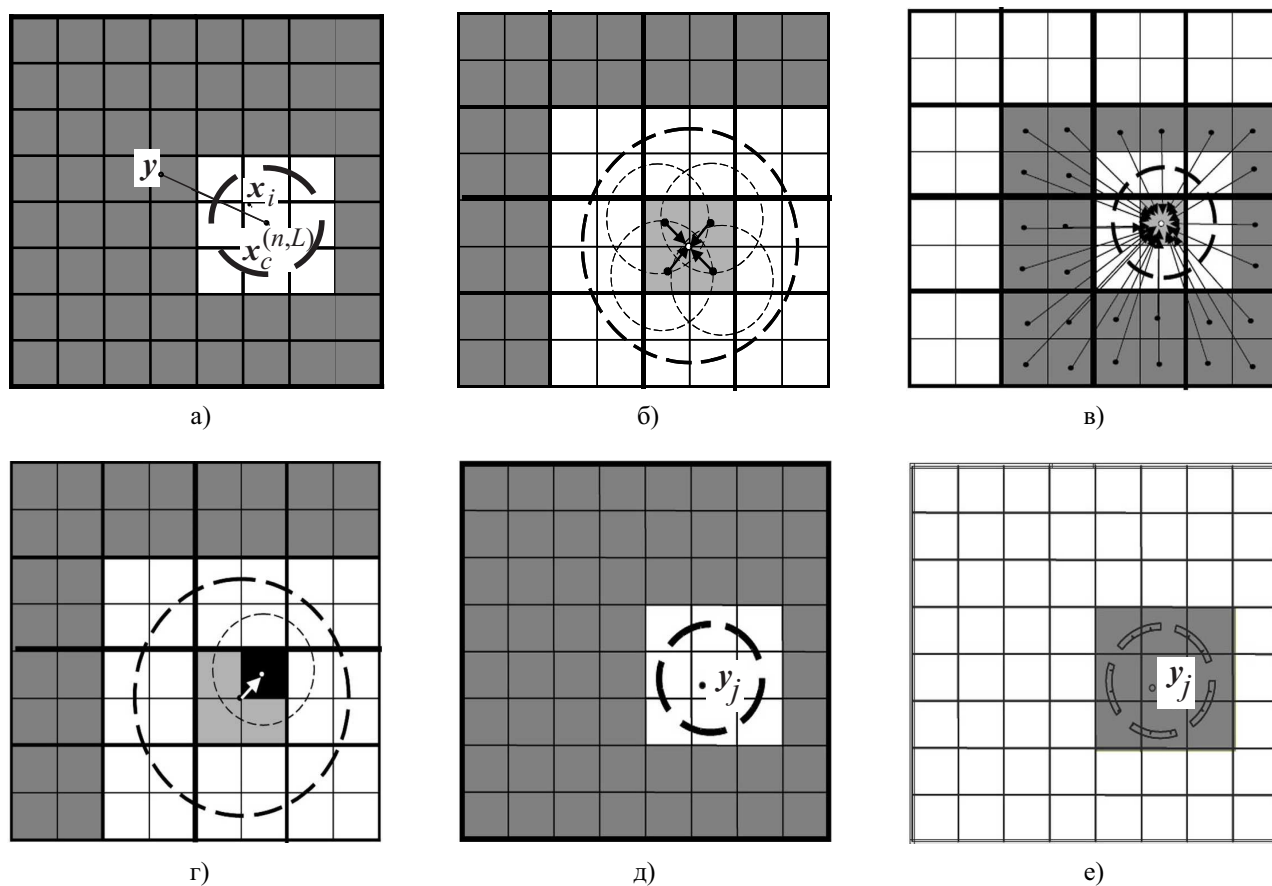


Рис. 4. Схемы трех основных шагов алгоритма многоуровневого FMM: восхождение (а, б), нисхождение (в, г), окончательное суммирование (д, е). Размерность пространства $d = 2$, $l_{\max} = 4$

— нисхождение (“downward pass”) с уровня $l = 2$ до $l_{\max}$: $S|R$ -трансляция и суммирование коэффициентов мультипольных S -разложений в коэффициенты R -разложений, причем трансляции производятся из боксов, которые не являются соседями для родительского бокса в иерархической структуре данных для

оцениваемой ячейки (рис. 4в); $R|R$ -трансляция локальных разложений от родительских ячеек с предыдущего уровня (рис. 4г) и суммирование с коэффициентами, полученными с помощью $S|R$;

— окончательное суммирование (“final summation”) на уровне $l_{\max}$: оценка R -разложения для каждого бокса с приемниками (рис. 4д) и прямое суммирование вклада источников из соседних боксов (рис. 4е).

Более детально с основами FMM можно ознакомиться в [4, 5].

4.2. Быстрый метод мультиполей для уравнений Стокса. В данной работе применяется FMM, предложенный в статьях [16, 17], в которых суммирование фундаментальных решений уравнений Стокса (3) сводится к суммированию фундаментальных решений трехмерного уравнения Лапласа. Поле скоростей стокслета с моментом $\mathbf{f}$, $\mathbf{v}(\mathbf{y}, \mathbf{x}) = \mathbf{G}(\mathbf{y}, \mathbf{x}) \cdot \mathbf{f}$ можно представить в виде

$$\mathbf{v} = \mathbf{f} \frac{1}{r} + (\mathbf{f} \cdot \mathbf{r}) \frac{\mathbf{r}}{r^3} = \sum_{k=1}^3 \left[\mathbf{i}_k \frac{f_k}{r} - y_k \nabla_y \frac{f_k}{r} \right] + \nabla_y \frac{(\mathbf{f} \cdot \mathbf{x})}{r}, \quad \mathbf{r} = \mathbf{y} - \mathbf{x}.$$

Поле скоростей стресслета с моментами $\mathbf{u}$ и $\mathbf{n}$, $\mathbf{v}(\mathbf{y}, \mathbf{x}) = (\mathbf{T}(\mathbf{y}, \mathbf{x}) \cdot \mathbf{n}) \cdot \mathbf{u}$ можно представить в виде

$$\mathbf{v} = -3 \frac{\mathbf{r}(\mathbf{u} \cdot \mathbf{r})(\mathbf{n} \cdot \mathbf{r})}{r^5} = \sum_{k=1}^3 \left[-\mathbf{i}_k \frac{(\mathbf{d}_k \cdot \mathbf{r})}{r^3} + y_k \nabla_y \frac{(\mathbf{d}_k \cdot \mathbf{r})}{r^3} \right] - \nabla_y \frac{(\mathbf{c} \cdot \mathbf{r})}{r^3},$$

$$\mathbf{d}_k = \frac{1}{2} (\mathbf{n} u_k + \mathbf{u} n_k), \quad \mathbf{c} = \frac{1}{2} [\mathbf{n}(\mathbf{u} \cdot \mathbf{x}) + \mathbf{u}(\mathbf{n} \cdot \mathbf{x})],$$

где y_k , f_k , n_k и u_k — компоненты соответствующих векторов в декартовой системе координат, $\mathbf{i}_k$ — базисные векторы. Таким образом, общее поле можно представить в форме

$$\mathbf{v} = \sum_{k=1}^3 (\mathbf{i}_k \Phi_k - y_k \nabla_y \Phi_k) + \nabla_y \Phi_0, \quad \Phi_0 = \frac{(\mathbf{f} \cdot \mathbf{x})}{r} - \frac{(\mathbf{c} \cdot \mathbf{r})}{r^3}, \quad \Phi_k = \frac{f_k}{r} - \frac{(\mathbf{d}_k \cdot \mathbf{r})}{r^3}, \quad k = 1, 2, 3.$$

Следовательно, вычисление интегралов в (3) сводится к суммированию монополей и диполей уравнения Лапласа и вычислению градиента этих сумм. Можно отметить, что суммирование требует четырех вызовов FMM для независимых функций Φ_k , $k = 0, 1, 2, 3$. На основе этой декомпозиции для решения уравнений Стокса используются уже готовые компоненты FMM для трехмерного уравнения Лапласа [4].

4.3. Реализация быстрого метода мультиполей на гетерогенных вычислительных системах. Для ускорения расчетов в настоящей работе используются графические процессоры. Реализация FMM на GPU была предложена в статье [4]. Подробный анализ алгоритма [6] показал, что более эффективно он может быть реализован на гетерогенных вычислительных архитектурах, состоящих из многоядерных центральных процессоров и графических процессоров. Особенность FMM состоит в формальном представлении расчетной матрицы A линейной системы (4) следующим образом: $A = A_{\text{sparse}} + A_{\text{dense}}$, где A_{sparse} — разреженная матрица, в которой учитываются взаимодействия узлов, расположенных на уровне $l_{\max}$ в соседних боксах, а A_{dense} — плотная матрица, учитывающая все остальные взаимодействия. Произведение $A_{\text{sparse}} \mathbf{x}$ вычисляется напрямую, а произведение $A_{\text{dense}} \mathbf{x}$ вычисляется при помощи разложений и операторов трансляций функций (рис. 5).

В [4] показано, что в связи с особенностями архитектуры графических процессоров произведение $A_{\text{sparse}} \mathbf{x}$ может быть реализовано достаточно эффективно на GPU с ускорением до 100 раз в сравнении с одним ядром CPU, в то время как реализация $A_{\text{dense}} \mathbf{x}$ на GPU позволяет сократить время вычисления всего в несколько раз. Последнее обусловлено тем, что используемая в FMM структура данных достаточно сложна и требует частого обращения к глобальной памяти GPU, а также ограниченным размером локальной памяти GPU, что недостаточно для хранения всех необходимых операторов трансляций. В то же время, использование CPU с P ядрами делает возможным достаточно простую параллелизацию операции $A_{\text{dense}} \mathbf{x}$ на CPU с использованием OpenMP. Причем эффективность такого подхода близка к 100%, т.е. на P ядрах достигается ускорение практически в P раз. Кроме того, возможность обрабатывать на GPU кластеры большого числа частиц снижает глубину иерархического дерева структуры данных, что обеспечивает дополнительное ускорение алгоритма.

Отметим особенности реализации модуля вычисления произведения разреженной матрицы $A_{\text{sparse}} \mathbf{x}$ на GPU с использованием технологии CUDA:

— все необходимые для вычислений массивы хранятся в глобальной памяти GPU, включая массивы размером $3N$ с координатами узлов сетки (массивы для источников и приемников для данной динамической задачи совпадают), списки соседей и геометрические характеристики сетки (массивы площадей и нормалей);

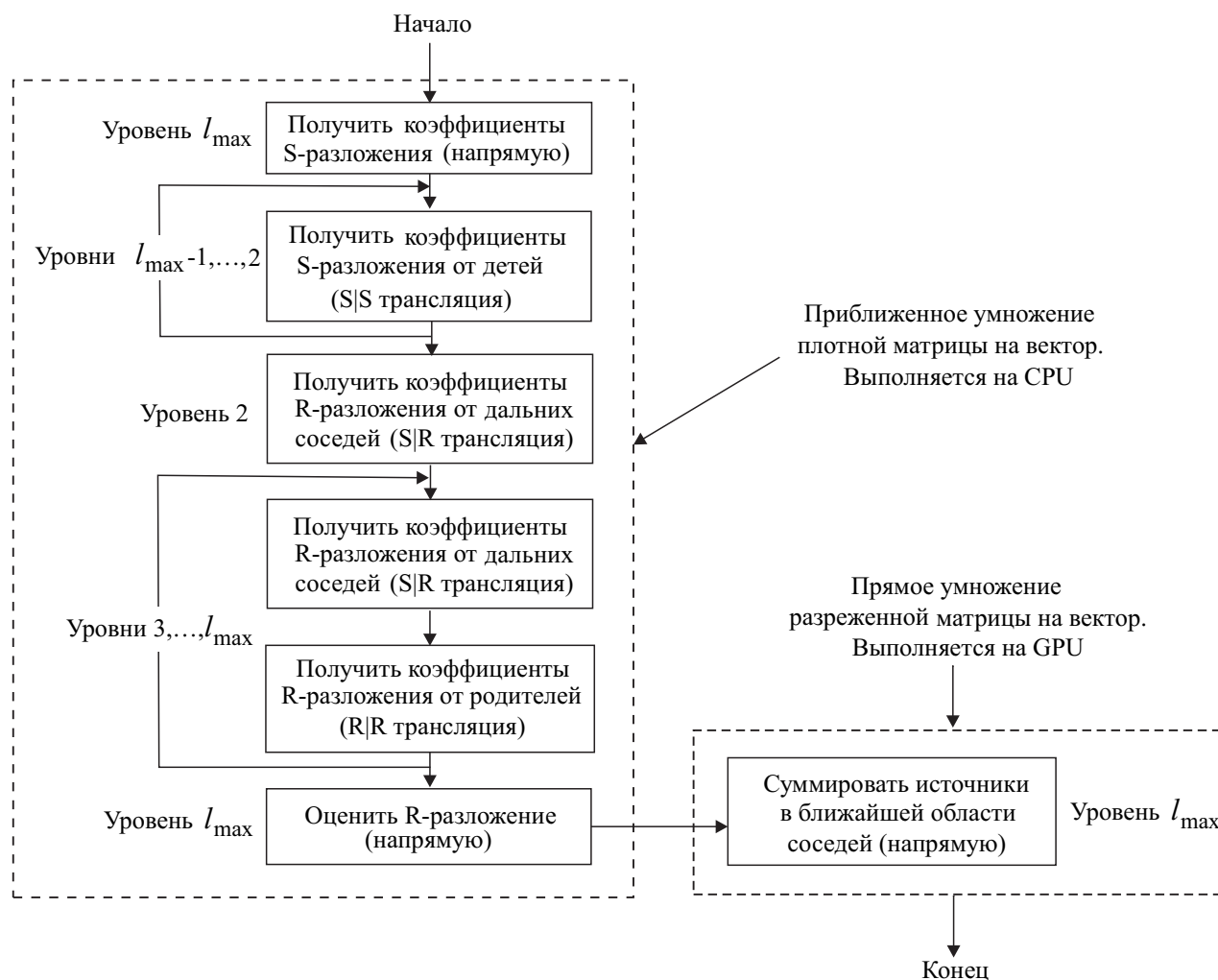


Рис. 5. Блок-схема алгоритма FMM

- количество потоков в блоке задавалось равным 256; при проведении вычислительных экспериментов было установлено, что для различных размеров задачи этот размер блока является оптимальным;
- каждый блок потоков обрабатывает только один непустой бокс с приемниками; каждый поток в блоке копирует необходимые координаты приемника из глобальной памяти в разделяемую память;
- каждый поток вычисляет свою часть результирующего вектора, которая по окончании вычислений копируется в глобальную память GPU для получения полного вектора решения.

Подобный подход к распараллеливанию алгоритма многоуровневого FMM, используемый в настоящей работе, был предложен и протестирован в статье [6]. В цитируемой работе структура данных формировалась на GPU с помощью нового алгоритма, который, несмотря на ускорение, существенно ограничен размером глобальной памяти GPU. Для задачи, решаемой в настоящей работе, такого ускорения не требуется и структура данных может вычисляться на CPU, поскольку для итеративного решения СЛАУ и вычисления сингулярных частей интегралов с учетом того, что один вызов FMM для решения уравнений Стокса соответствует четырем вызовам FMM для ядра Лапласа, количество вызовов вычислительной части FMM для одного и того же пространственного распределения расчетных узлов составляет приблизительно 50. Таким образом, в течение одного шага по времени происходит амортизация формирования структуры данных, что не приводит к существенному увеличению общего времени счета.

5. Результаты тестовых расчетов. Все программные модули (кроме многоуровневого FMM) разработаны в среде Matlab. Расчеты проводились на персональном компьютере, оснащённом CPU Intel Xeon 5660, 2.8 GHz, 12 GB RAM с 12 физическими и 12 виртуальными ядрами и GPU NVIDIA Tesla C2050, 3 GB глобальной памяти. Все тесты представлены для операций с двойной точностью.

5.1. Результаты тестирования модуля гетерогенного FMM. Для того чтобы подобрать оптимальные параметры FMM применительно к рассматриваемой в данной статье задаче, подпрограммы суммирования разложений фундаментальных решений уравнений Стокса тщательно протестированы. На

рис. 6 показаны графики зависимости относительной погрешности FMM для суммирования стокслетов и стресслетов от размера задачи при различных значениях параметра p . Вообще говоря, не только точность, но и скорость выполнения алгоритма сильно зависит от p (табл. 1). Для последующих расчетов течения эмульсии достаточно выбирать $p = 8$, так как при этом значении достигается баланс между получаемой точностью алгоритма и временем его выполнения. Таким образом, точность используемого FMM составила порядка 10^{-5} . На общую погрешность численного подхода, применяемого в данном исследовании, также влияют погрешности вычисления геометрических характеристик, сингулярных частей интегралов, точности итеративного решателя и схемы интегрирования по времени.

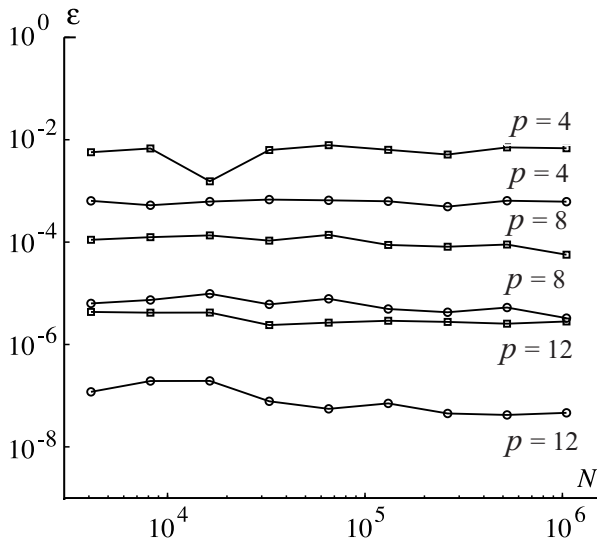


Рис. 6. Относительная погрешность (ϵ) FMM для стокслетов (отмечено на графиках линиями с кружками) и стресслетов (отмечено на графиках линиями с квадратами) при различных значениях параметра p

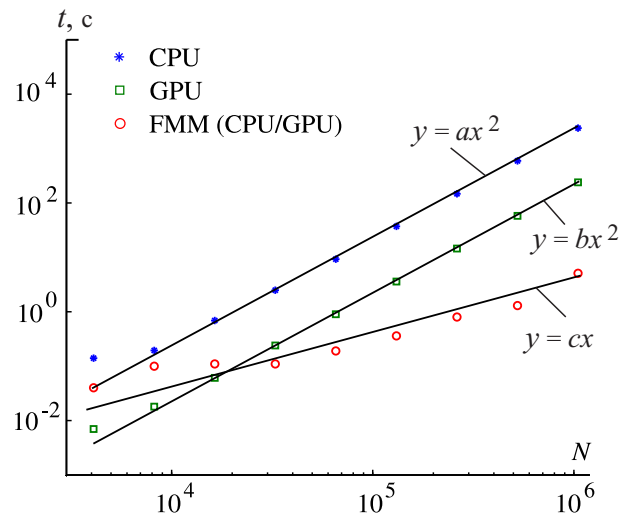


Рис. 7. Время выполнения подпрограмм МВП для стокслетов, реализованных на CPU, GPU и с помощью гетерогенного FMM

Таблица 1

Время одного вызова FMM для стокслетов и стресслетов

N	p	$l_{\max}$	Формирование структуры данных, с	Стокслеты		Стресслеты	
				Время вычисления $A_{\text{dense}}\mathbf{x}$, с	Время вычисления $A_{\text{sparse}}\mathbf{x}$, с	Время вычисления $A_{\text{dense}}\mathbf{x}$, с	Время вычисления $A_{\text{sparse}}\mathbf{x}$, с
4 096	4	3	0.021	0.016	0.002	0.02	0.0018
32 768	4	3	0.03	0.02	0.015	0.02	0.014
131 072	4	4	0.28	0.18	0.053	0.19	0.053
524 288	4	5	1.02	0.96	0.236	0.91	0.225
1 048 576	4	5	1.33	1.09	0.465	1.09	0.44
4 096	8	2	0.014	0.013	0.0028	0.011	0.0028
32 768	8	3	0.03	0.06	0.018	0.074	0.017
131 072	8	4	0.17	0.58	0.053	0.613	0.053
524 288	8	5	0.99	4.36	0.24	4.5	0.225
1 048 576	8	5	1.39	4.41	0.466	4.6	0.43
4 096	12	3	0.02	0.27	0.002	0.26	0.0018
32 768	12	3	0.034	0.32	0.018	0.31	0.017
131 072	12	4	0.15	1.53	0.053	1.52	0.053
524 288	12	5	1.02	12.2	0.236	12.15	0.225
1 048 576	12	5	1.4	12.48	0.46	12.497	0.45

В табл. 1 приведены результаты замеров времени выполнения отдельных частей алгоритма FMM при нескольких значениях параметра p для различных размеров задач. Значительное изменение времени при

Таблица 2

Сравнение времени выполнения МВП прямым методом счета на CPU и на GPU и МВП с применением FMM (гетерогенный CPU/GPU алгоритм) для стокслетов и стресслетов, $p = 8$

N	Оптимальный $l_{\max}$	Архитектура	Стокслеты		Стресслеты	
			Время вычисления, с	Ускорение	Время вычисления, с	Ускорение
4 096	2	CPU	0.14	1	0.08	1
		GPU	0.007	20	0.006	13.3
		FMM CPU/GPU	0.04	3.5	0.035	2.3
8 192	3	CPU	0.195	1	0.227	1
		GPU	0.018	10.8	0.017	13.4
		FMM CPU/GPU	0.1	2	0.12	1.9
16 384	3	CPU	0.689	1	0.61	1
		GPU	0.061	11.3	0.062	9.8
		FMM CPU/GPU	0.11	6.2	0.09	6.7
32 768	3	CPU	2.486	1	2.18	1
		GPU	0.24	10.4	0.23	9.5
		FMM CPU/GPU	0.11	22.6	0.11	19.8
65 536	3	CPU	9.272	1	8.57	1
		GPU	0.9	10.2	0.83	10.3
		FMM CPU/GPU	0.19	48.8	0.24	35.7
131 072	3	CPU	37.12	1	33.54	1
		GPU	3.6	10.3	3.5	9.58
		FMM CPU/GPU	0.36	103	0.43	78
262 144	4	CPU	147.78	1	135.7	1
		GPU	14.5	10.19	14.1	9.6
		FMM CPU/GPU	0.8	185	0.87	156
524 288	4	CPU	592.48	1	545.3	1
		GPU	58	10.22	56	9.73
		FMM CPU/GPU	1.3	456	1.3	419.4
1 048 576	5	CPU	2379.3	1	2222	1
		GPU	239	9.96	225	9.87
		FMM CPU/GPU	5.1	466	5.3	419.2

увеличении p происходит за счет возрастания времени расчета произведения плотной матрицы дальних взаимодействий. Это происходит в связи с увеличением количества членов в разложении потенциалов при вычислении операторов трансляций и разложений функций. Кроме того, большое значение имеет выбор максимального уровня разбиения иерархического дерева. Как видно из таблиц, чтобы достичь оптимальных результатов, необходимо для каждого размера задачи подбирать свой уровень разбиения. От глубины дерева зависит, какая часть вычислительной нагрузки будет распределена на CPU, а какая на GPU. Чем больше значение $l_{\max}$, тем меньше вычислений выполняется на GPU и больше рассчитывается трансляций на CPU.

В табл. 2 представлено время счета МВП для различных размеров матриц. Выполнено сравнение подпрограмм, реализованных на CPU без оптимизаций и распараллеливания, на GPU с применением

технологии CUDA (при замерах времени учитывалась пересылка данных), а также с применением гетерогенного FMM (без учета формирования структуры данных). Графики зависимости времени выполнения подпрограмм для суммирования стоксетов, реализованных на различных архитектурах, от размера исследуемой задачи показаны на рис. 7. Видно, что сложность МВП, реализованного на CPU или GPU, квадратичная, а с использованием гетерогенного FMM линейная.

Из данных, приведенных в табл. 1 и 2, следует, что применение FMM к задачам малого размера приводит к сравнительно небольшому ускорению, хотя требуется реализовать достаточно сложный алгоритм FMM. Поэтому в данной работе для проведения тестовых расчетов для небольшого количества капель используется МГЭ без модификаций, а для расчетов большого количества капель — МГЭ с гетерогенным FMM.

5.2. Тестовые расчеты динамики разбавленной эмульсии. Реализованные методы протестированы для случая одиночной сферической капли в неограниченном потоке и проведено сравнение с аналитическим решением для обтекания жидкой сферы. Относительная погрешность в норме L_∞ , определяемая для численного решения $\mathbf{u}^*$ и аналитического $\mathbf{u}$ как $\delta(\mathbf{u}^*) = \frac{\|\mathbf{u}^* - \mathbf{u}\|_\infty}{\|\mathbf{u}\|_\infty}$, где $\|\mathbf{u}\|_\infty = \max |u_i|$, $i = 1, \dots, N$, для $N_\Delta = 642$ составила 1–1.5% для $\lambda = 2.5$. При увеличении количества узлов сетки на поверхности ошибка уменьшалась. Кроме того, проведено исследование формы и положения деформируемых капель при различных отношениях вязкостей и различных капиллярных числах $Ca = \frac{\mu_1 a G}{\gamma}$, где a — радиус недеформируемой капли и G — скорость сдвига. Выполненные расчеты показали хорошее согласование с экспериментальными данными и опубликованными численными результатами [23].

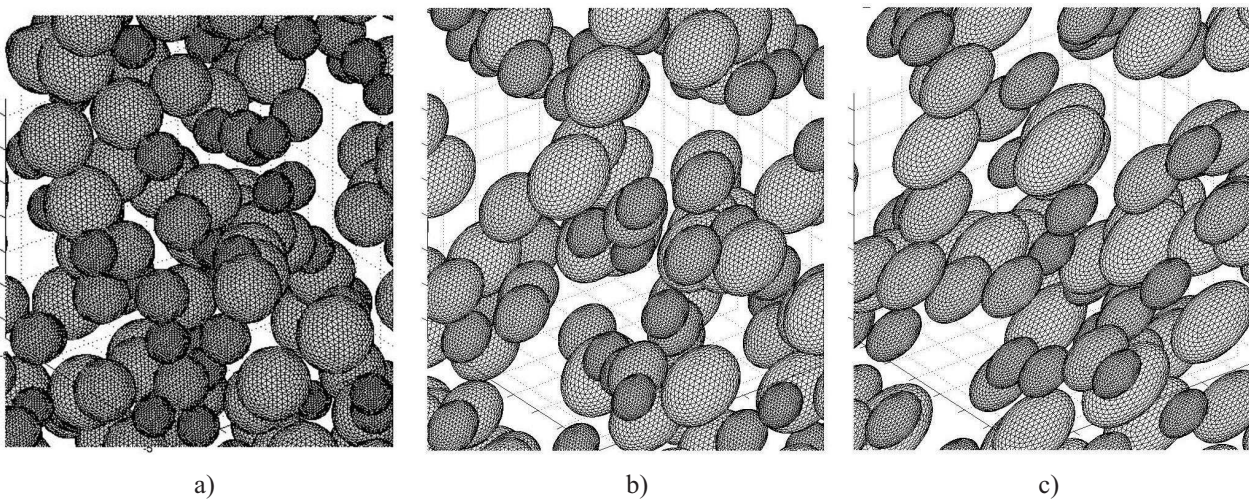


Рис. 8. 512 каплей в сдвиговом потоке в различные моменты времени, $N_\Delta = 642$, $Ca = 0.8$, $\lambda = 2$, $\alpha = 1.5 \times 10^{-2}$ (часть расчетной области): а) $t = 0$, б) $t = 0.5$, в) $t = 1$

Тестовые расчеты проводились для полидисперсной смеси изначально сферических капель с равномерным распределением по радиусам от $a_{\min}$ до $a_{\max}$, $\frac{a_{\max}}{a_{\min}} = 2$, $\lambda = 0.6$, и капиллярным числом $Ca = 1$ (рис. 8–10). На рис. 9 показано начальное распределение сферических капель в кубе со стороной $500 a_{\min}$. Капли равномерным случайным образом распределены внутри куба. Капли, расстояние между центрами которых не превышало 1.3 суммы их радиусов, удалялись. После этого расчетная область содержала $M = 10\,934$ капли, что соответствует объемному содержанию $\alpha = 1.4 \times 10^{-3}$. Поверхность каждой капли дискретизировалась сеткой с $N_\Delta = 162$ вершинами. Таким образом, в данном примере общее количество расчетных узлов составило $N = M \times N_\Delta = 1\,771\,308$ и общее количество неизвестных в СЛАУ $3N = 5\,313\,924$. В работах [8, 9, 20] показано, что дискретизация поверхности капли относительно небольшим числом элементов приводит к хорошим результатам, согласующимся с экспериментальными данными, для капель в сдвиговых потоках, когда форма капли достаточно гладкая. Для сильных деформаций, приводящих, например, к дроблению, может возникнуть необходимость в большей дискретизации, что требует дополнительных исследований, выходящих за рамки данной работы.

На рис. 10 изображен фрагмент расчетной области $(-10; 10)^3$ для случая с 10 934 каплями в момент безразмерного времени $t = 1$, связанного с размерным временем как $t = t_{\text{nondim}} = \frac{\gamma t_{\text{dim}}}{\mu_1 a_{\min}}$.

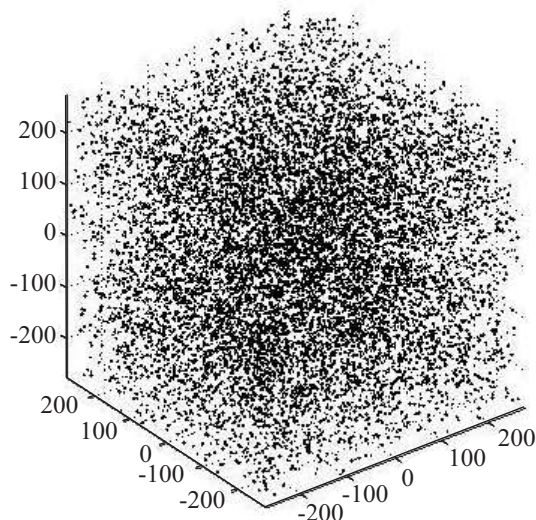


Рис. 9. Начальное распределение $M = 10\,934$ капель

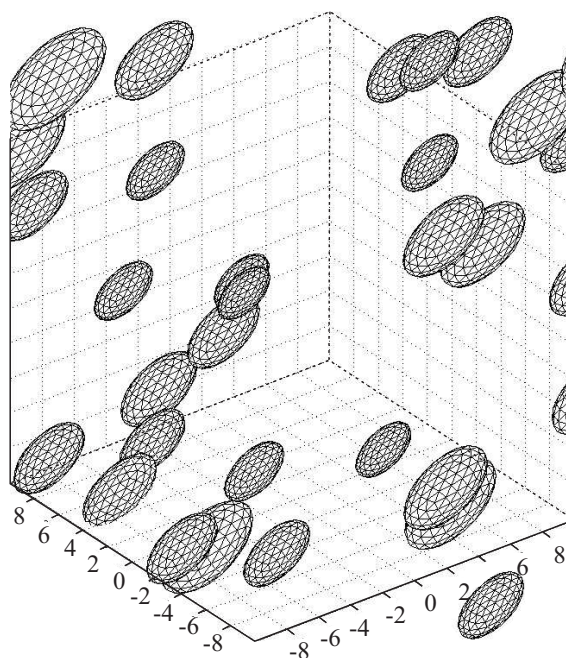


Рис. 10. Фрагмент расчетной области на 100 шаге по времени для 10 934 капель в сдвиговом потоке, $N_\Delta = 162$, $Ca = 1$, $\lambda = 0.6$, $t = 1$, $\alpha = 1.4 \times 10^{-3}$

Рисунок 8 иллюстрирует динамику 512 капель с $N_\Delta = 642$ и $\alpha = 1.6 \times 10^{-2}$ в различные моменты безразмерного времени $t = 0$, $t = 0.5$, $t = 1$. Временной шаг интегрирования для обоих случаев выбирался равным $\Delta t = 0.01$. Изначально сферические капли с течением времени вытягиваются по потоку. В этих тестовых примерах рассматривались жидкости с одинаковыми плотностями. Положение в пространстве и деформация капель для заданных капиллярных чисел и соотношения вязкостей те же, что и для тестового случая с одной каплей, и хорошо согласуются с соответствующими экспериментальными данными и численными результатами [8].

Общее время счета процесса для 100 шагов по времени (для 10 934 капель) составило около 5 часов, а один вызов FMM для уравнений Стокса — 7 секунд. Следует отметить, что на каждом временном шаге модуль FMM вызывался около 11 раз; кроме того производилось вычисление всех геометрических характеристик капель. Исследовалась зависимость времени расчета матрично-векторного произведения, одного временного шага и 100 временных итераций от количества капель (рис. 11). Во всех случаях время вычислений растет по линейному закону при увеличении количества узлов сетки, что показывает хорошую масштабируемость используемого алгоритма.

6. Заключение. Разработан и реализован эффективный подход для расчета динамики большого количества деформируемых капель в стоксовом режиме. Этот подход основан на методе граничных элементов для трехмерных задач, ускорение которого произведено как за счет высокоэффективного масштабируемого алгоритма FMM, так и за счет использования гетерогенных вычислительных архитектур (многоядерные CPU и графические процессоры). Все это позволяет напрямую рассчитывать взаимодействие десятков тысяч деформируемых капель. Разра-

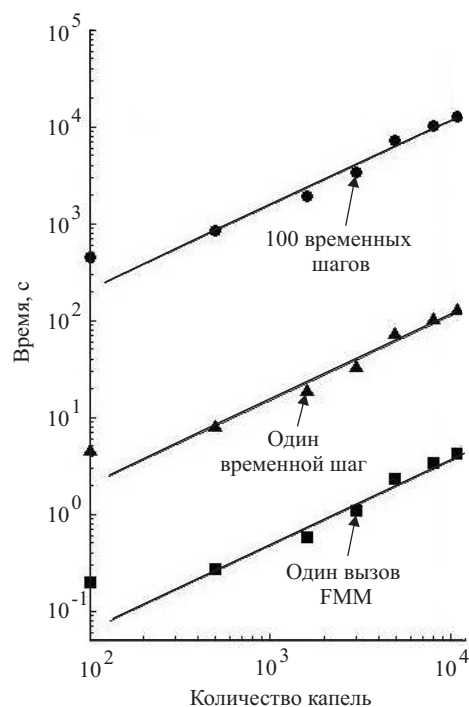


Рис. 11. Время расчета матрично-векторного произведения, одного временного шага и 100 временных итераций в зависимости от количества капель в логарифмической шкале

ботанный метод может быть использован для решения широкого класса задач, связанных с течениями эмульсий в микро- и наномасштабах. Он может быть использован также и для установления замыкающих соотношений для моделирования двухфазных течений “жидкость–жидкость” на основе континуального подхода в макромасштабах.

Работа выполнена при поддержке Министерства образования и науки Российской Федерации (грант № 11.G34.31.0040). Часть программного обеспечения, использованного при разработке основного кода, описанного в настоящей статье, любезно предоставлена компанией Fantalgo, LLC (USA).

СПИСОК ЛИТЕРАТУРЫ

1. Bayareh M., Mortazavi S. Three-dimensional numerical simulation of drops suspended in simple shear flow at finite Reynolds numbers // J. Multiphase Flow. 2011. **37**. 1315–1330.
2. Biros G., Ying L., Zorin D. A fast solver for the Stokes equations with distributed forces in complex geometries // J. Comp. Phys. 2003. **193**. 317–348.
3. Greengard L., Rokhlin V. A fast algorithm for particle simulations // J. Comp. Phys. 1987. **73**. 325–348.
4. Gumerov N.A., Duraiswami R. Fast multipole methods on graphics processors // J. Comput. Phys. 2008. **227**, N 18. 8290–8313.
5. Gumerov N.A., Duraiswami R. Fast multipole methods for the Helmholtz equation in three dimensions. Oxford: Elsevier, 2005.
6. Hu Q., Gumerov N.A., Duraiswami R. Scalable fast multipole methods on distributed heterogeneous architectures // Proc. of International Conference on High Performance Computing. Article N 36. New York: ACM Press, 2011.
7. Hu Q., Gumerov N.A., Duraiswami R. Scalable distributed fast multipole methods // Proc. 14th International Conference on High Performance Computing and Communications. New York: IEEE Press, 2012. 270–279.
8. Kennedy M.R., Pozrikidis C., Skalak R. Motion and deformation of liquid drops, and the rheology of dilute emulsions in simple shear flow // Computers and Fluids. 1994. **23**, N 2. 251–278.
9. Loewenberg M., Hinch E.J. Numerical simulation of a concentrated emulsion in shear flow // J. Fluid Mech. 1996. **321**. 395–419.
10. Nourbakhsh A., Mortazavi S. The lateral migration of a drop under gravity between two parallel plates at finite Reynolds numbers // J. Applied Fluid Mech. 2012. **5**, N 1. 11–21.
11. Pozrikidis C. Boundary integral and singularity methods for linearized viscous flow. Cambridge: Cambridge Univ. Press, 1992.
12. Rahimian A., Lashuk I., Veerapaneni S.K., Chandramowlishwaran A., Malhotra D., Moon L., Sampath R., Shringarpure A., Vetter J., Vuduc R., Zorin D., Biros G. Petascale direct numerical simulation of blood flow on 200K cores and heterogeneous architectures // Proc. Supercomputing'10. New Orleans, 2010. 1–11.
13. Rallison J.M., Acrivos A. A numerical study of the deformation and burst of a viscous drop in an extensional flow // J. Fluid Mech. 1978. **89**, N 1. 91–200.
14. Saad Y. Iterative methods for sparse linear system. Philadelphia: SIAM, 2000.
15. Sangani A.S., Mo G. An $O(N)$ algorithm for Stokes and Laplace interactions of particles // Phys. Fluids. 1996. **8**, N 8. 1990–2010.
16. Tornberg A.K., Greengard L. A fast multipole method for the three-dimensional Stokes equations // J. Comput. Phys. 2008. **227**, N 3. 1613–1619.
17. Wang H., Lei T., Li J., Huang J., Yao Z. A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations // Int. J. Num. Meth. Engng. 2007. **70**. 812–839.
18. Ying L., Biros G., Zorin D. A kernel-independent adaptive fast multipole algorithm in two and three dimensions // J. Comp. Phys. 2004. **196**. 591–626.
19. Zhao H., Isfahani A.H.G., Olson L.N., Freund J.B. A spectral boundary integral method for flowing blood cells // J. Comp. Phys. 2010. **229**. 3726–3744.
20. Zinchenko A.Z., Rother M.A., Davis R.H. A novel boundary-integral algorithm for viscous interaction of deformable drops // Phys. Fluids. 1997. **9**, N 6. 1493–1511.
21. Zinchenko A.Z., Davis R.H. Large-scale simulations of concentrated emulsion flows // Phil. Trans. R. Soc. Lond. A. 2003. **361**. 813–845.
22. Zinchenko A.Z., Davis R.H. A multipole-accelerated algorithm for close interaction of slightly deformable drops // J. Comp. Phys. 2005. **207**. 695–735.
23. Абрамова О.А., Итжулова Ю.А., Гумеров Н.А. Моделирование трехмерного движения деформируемых капель в стоковом режиме методом граничных элементов // Вычисл. мех. сплош. сред. 2013. **6**, № 2. 214–223.
24. Солнышкина О.А., Итжулова Ю.А., Гумеров Н.А. Ускорение расчетов на графических процессорах при исследовании течения Стокса методом граничных элементов // Вестник Уфимского гос. авиационного техн. ун-та. 2013. **17**, № 2 (55). 92–100.

Поступила в редакцию
31.07.2013