

ИСПОЛЬЗОВАНИЕ ЭФФЕКТИВНЫХ АЛГОРИТМОВ ОПТИМИЗАЦИИ И СУПЕР-КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ ПРИ ПОСТРОЕНИИ НЕСТРУКТУРИРОВАННЫХ РАСЧЕТНЫХ СЕТОК МЕТОДОМ МОЛЕКУЛЯРНО-ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ

А.Л. Железнякова

Институт проблем механики им. А.Ю. Ишлинского Российской академии наук, Россия, Москва, 119526, проспект Вернадского, д.101-1

Аннотация

Представлена параллельная реализация компьютерного кода, предназначенного для построения двумерных и трехмерных, однородных и адаптивных неструктурированных расчетных сеток для различных задач механики сплошной среды с использованием классических методов молекулярной динамики. Предложены и реализованы различные методы ускорения вычислений, основанные на модификации исходной модели взаимодействующих частиц и методы алгоритмической оптимизации, не требующие модификации модели. Применявшиеся методы оптимизации направлены главным образом на уменьшение расчетного времени. Созданы альтернативные алгоритмы распараллеливания с использованием библиотеки параллельного программирования MPI и технологии OpenMP.

MOLECULAR DYNAMICS-BASED UNSTRUCTURED GRID GENERATION USING SUPER-COMPUTING TECHNOLOGIES AND EFFICIENT OPTIMIZATION ALGORITHMS

Alexandra L. Zheleznyakova

Institute for Problems in Mechanics Russian Academy of Sciences, Russia, Moscow, 119526

A parallel implementation of the molecular dynamics-based unstructured grid generation algorithm for continuum mechanics problems is presented. Various methods for speeding up computation based on modification of particle-particle interaction model and algorithm optimization techniques are proposed and implemented. Optimization methods are mainly applied to reduce computation time. Alternative parallel algorithms using OpenMP and MPI parallel technologies are developed.

1. ВВЕДЕНИЕ

Главная идея молекулярно-динамического метода построения двумерных и трехмерных, однородных и адаптивных неструктурированных расчетных сеток, основные положения которого изложены в предшествующих работах (см., например, [1]), состоит в том, что узлы будущей сетки это одноименно заряженные, взаимодействующие между собой частицы, с зарядами q_i , и массами m_i (рис. 1).

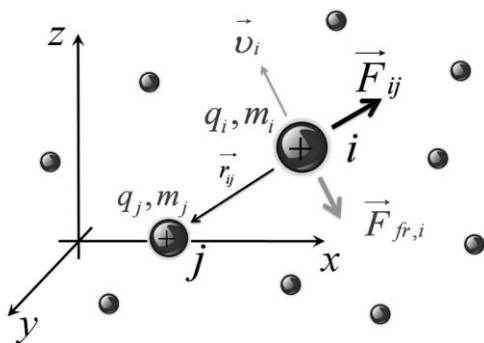


Рис. 1. Схема взаимодействия частиц между собой

Отталкивающая сила, действующая на каждую частицу, со стороны остальных частиц системы имеет электрическую природу:

$$\vec{F}_{ij} = -C \frac{q_i q_j}{|\vec{r}_{ij}|^k} \cdot \frac{\vec{r}_{ij}}{|\vec{r}_{ij}|}, \quad k \geq 1, \quad |\vec{r}_{ij}| = l, \quad (1)$$

здесь l – длина радиуса-вектора или расстояние между зарядами, C – постоянная нормировки.

Кроме сил взаимодействия в системе действуют силы сопротивления. Основной закон динамики для системы подвижных узлов можно записать в следующем виде:

$$m_i \frac{dv_i}{dt} = \vec{F}_i + \vec{F}_{fr,i} + \vec{F}_{i,wall}, \quad i = 1, \dots, N, \quad (2)$$

где первое слагаемое – равнодействующая сил, действующих на i -ю подвижную частицу со стороны остальных подвижных частиц $\vec{F}_i = \sum_{j=1}^N \vec{F}_{ij}$, второе – сила сопротивления, третье – равнодействующая сил, действующих на i -ю частицу со стороны границ расчетной области. Исходную систему из N уравнений замыкаем дифференциальными уравнениями для скорости, в результате получаем (в проекциях на оси координат):

$$\begin{cases} m_i \frac{dv_{ai}}{dt} = F_{ai} + F_{fr,ai} + F_{ai,wall}, & i = 1, \dots, N \\ v_{ai} = dr_{ai}/dt, \end{cases} \quad (3)$$

$\alpha = x, y, z$

Моделирование динамики системы одновременно заряженных, попарно взаимодействующих частиц или узлов, проводится путем интегрирования уравнений движения Ньютона на каждом временном шаге для каждой частицы.

Численное интегрирование уравнений движения (3) проводится по алгоритму Верле [2]:

1) В начальный момент времени $t = 0$ задается число моделируемых частиц N , их массы m_i , заряды q_i . Координаты подвижных частиц в начальный момент времени задаются случайным образом. В момент времени $t = 0$ частицы покоятся.

2) В момент времени отличный от нуля, значения компонентов скорости и координат частицы используются для определения проекций сил, действующих на частицу, и проекций равнодействующей силы.

3) При $t \neq 0$ равнодействующая всех сил, действующих на i -ю частицу заставляет ее двигаться с ускорением:

$$a_{ai}(t) = m_i^{-1} \Sigma_{ai}(t), \quad (4)$$

$$\alpha = x, y, z.$$

4) Зная ускорение, скорость и положение частицы в текущий момент времени можно вычислить скорость и новое положение частицы в следующий момент времени:

$$v_{ai}(t + \delta t) = v_{ai}(t) + a_{ai}(t)\delta t,$$

$$r_{ai}(t + \delta t) = r_{ai}(t) + v_{ai}(t + \delta t)\delta t, \quad (5)$$

$$\alpha = x, y, z.$$

Расчет траекторий в процессе молекулярно-динамического моделирования на каждом временном шаге (для достижения системой состояния близкого к равновесному необходимо не менее 500 шагов по времени) требует использования существенных компьютерных ресурсов, кроме того, исключительная вычислительная сложность поставленной задачи определяется особенностями рассматриваемой системы (рис. 2):

- 1) Большая размерность сеток применяемых в промышленном моделировании. Число узлов $N \sim 10^6 - 10^8$.
- 2) Сложная форма расчетных областей.
- 3) Большая неоднородность распределения узлов расчетной сетки со сгущениями в зонах больших градиентов.

Единственный способ решения поставленной задачи за разумное время – использование методов ускорения вычислений.

2. ОПТИМИЗАЦИЯ АЛГОРИТМОВ МОЛЕКУЛЯРНО-ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ

Описанная модель взаимодействующих частиц позволяет применить методы для уменьшения вычислительной сложности. В работе использовались различные методы ускорения вычислений, основанные на модификации исходной модели взаимодействующих частиц и методы алгоритмической оптимизации, не требующие модификации модели. Применявшиеся методы оптимизации были направлены главным образом на уменьшение расчетного времени. Наиболее трудоемкий этап рассматриваемого алгоритма это расчет сил, действующих в системе.

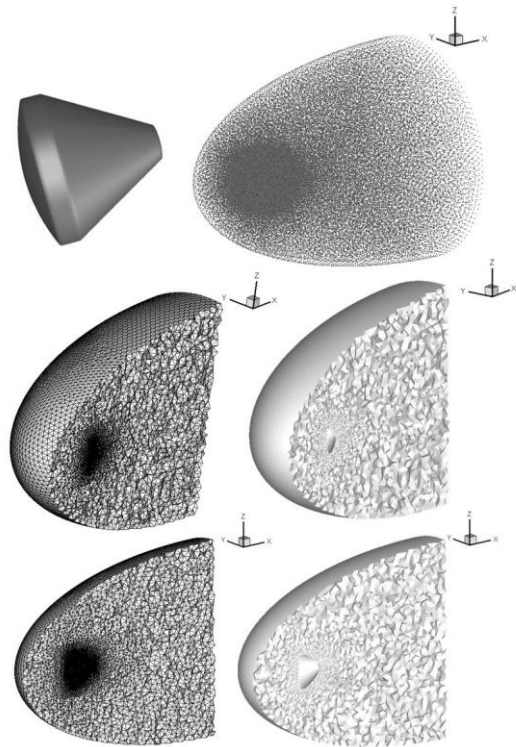


Рис. 2. Трехмерная адаптивная неструктурированная расчетная сетка около спускаемого космического аппарата Fire II, построенная с использованием молекулярно-динамического моделирования

Трудоемкость вычисления сил парного взаимодействия возрастает пропорционально квадрату числа частиц (рис. 3). Вычислительную сложность данной процедуры можно уменьшить, используя максимальный радиус взаимодействия R_{max} , когда i -я частица вступает во взаимодействие только со своими ближайшими соседями (с частицами находящимися не дальше максимального радиуса взаимодействия). На рис. 3 с помощью связей показано, сколько парных сил взаимодействия нужно вычислить с введением и без введения R_{max} .

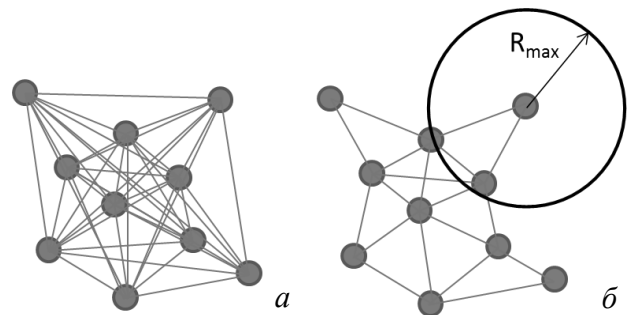


Рис. 3. Число парных сил взаимодействия, которые необходимо вычислить: а) без использования максимального радиуса взаимодействия; б) с использованием R_{max}

Еще один метод, применявшийся для ускорения молекулярно-динамического моделирования, был основан на использовании короткодействующих сил взаимодействия между частицами. При использовании сил Кулоновского отталкивания, которые убывают

достаточно медленно (как квадрат расстояния между частицами $F_{ij} \sim 1/|\vec{r}_{ij}|^2$), число частиц вступающих во взаимодействие с каждой i -й частицей будет очень большим. В этом случае почти все частицы расчетной области попарно взаимодействуют друг с другом. Если использовать короткодействующие силы, которые убывают пропорционально кубу, четвертой, пятой и т.д. степени расстояния между частицами ($F_{ij} \sim 1/|\vec{r}_{ij}|^k, k > 3$), то отдельная i -я частица будет взаимодействовать только с небольшим числом ближайших соседей и на относительно небольшом расстоянии (несоизмеримом с размерами расчетной области), силы взаимодействия будут настолько малы, что ими можно пренебречь. Следует особо отметить, что рассматриваемая система частиц не применяется для моделирования конкретной среды или физического процесса, а для решения поставленной в данной работе задачи (построение расчетных сеток) имеется возможность использования различных сил взаимодействия вида (1) между парами частиц. Изложенные выше методы, позволяющие существенно снизить трудоемкость вычисления сил парного взаимодействия, основаны на модификации исходной математической модели взаимодействующих частиц.

Использование “списка ближайших соседей” является еще одним подходом к ускорению вычислений и широко применяется при молекулярно-динамическом моделировании. Суть метода заключается в создании списка узлов, находящихся на расстоянии не далее R_{NBout} и вступающих во взаимодействие с i -й подвижной частицей. Список обновляется с периодичностью 3 – 30 шагов по времени в зависимости от подвижности частиц (подвижность определяется расстоянием, на которое перемещается частица за временной шаг). Особенностью рассматриваемой системы является то, что в начальные моменты времени частицы достаточно подвижны и перемещаются по расчетной области на значительные расстояния, а на завершающем этапе молекулярно-динамического моделирования узлы совершают лишь незначительные колебания около своих положений равновесия. Хранение информации о соседних узлах позволяет не вычислять расстояние между взаимодействующими частицами на каждом временном шаге и избежать затратной процедуры перебора всех возможных пар частиц.

Методы, основанные на использовании короткодействующих сил и введении максимального радиуса взаимодействия, не имеют этого преимущества, т.к. без вычисления расстояний между всеми парами частиц системы невозможно определить взаимодействуют частицы отдельной пары между собой или нет. При использовании данных двух подходов, компьютерные ресурсы удастся сэкономить за счет вычисления гораздо меньшего числа парных сил взаимодействия. Наиболее эффективным является совместное использование рассмотренных методов ускорения вычислений.

Если не затрагивать математическую модель рассматриваемой системы частиц, то для уменьшения сложности вычисления сил парного взаимодействия

между частицами можно предпринять следующее. Для наглядности представим силы парного взаимодействия, действующие в системе из N частиц в виде матрицы размером $N \times N$ (рис. 4).

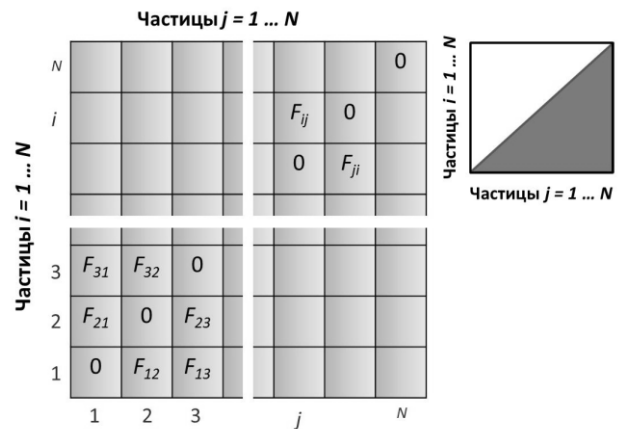


Рис. 4. Матрица сил парных взаимодействий между частицами

Здесь элемент с индексом (i, j) содержит силу взаимодействия i -й частицы с j -й частицей. Диагональные элементы этой матрицы равны нулю, т.к. частица не взаимодействует сама с собой. Аналогичные матрицы можно составить для трех компонент сил. Если воспользоваться третьим законом Ньютона ($\vec{F}_{i,j} = -\vec{F}_{j,i}$), то можно увидеть, что данная матрица является антисимметричной и любую парную силу можно вычислять только один раз, т.е. достаточно вычислить все элементы под главной диагональю или над главной диагональю. Это уменьшает трудоемкость вычисления парных сил в два раза. Для элементов под главной диагональю $j > i$ (см. рис. 4, справа).

Несмотря на то, что оптимизация исходных алгоритмов с использованием рассмотренных выше технологий позволила существенно снизить время проведения молекулярно-динамического моделирования (в 5 – 10 раз в зависимости от конкретной задачи), для расчета динамики больших систем ($N > 10^6$) за приемлемое время необходимо создавать альтернативные алгоритмы распараллеливания для расчетов с использованием современных высокопроизводительных вычислительных систем [3–5].

3. ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ МОЛЕКУЛЯРНО-ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ

Перед созданием параллельного алгоритма необходимо оценить масштабируемость задачи – насколько эффективно она будет распараллеливаться. Численное интегрирование уравнений движения включает следующие наиболее трудоемкие этапы:

1) Инициализация системы (задание координат и скоростей частиц в начальный момент времени, определение границ расчетной области).

2) Расчет равнодействующей силы, которая определяется суммой парных взаимодействий i -й частицы со всеми остальными частицами, силой сопротивления и силой взаимодействия с границами расчетной области.

3) Расчет параметров движения и новых координат частиц.

Последние два этапа выполняются на каждом временном шаге. Для данной системы расчет сил и траекторий занимает более 85% общего времени, но силы, действующие на каждую частицу и, следовательно, траектории, могут быть вычислены независимо на разных процессорах. Результаты оценки масштабируемости исходного последовательного кода, полученные в работе, показали, что параллельная реализация должна обладать высокой эффективностью, что также подтверждается проведенными расчетами.

В работе созданы две параллельные реализации исходного последовательного алгоритма с использованием библиотеки параллельного программирования MPI и технологии OpenMP. Параллельные программы написаны на языке FORTRAN 90.

3.1. Использование библиотеки параллельного программирования MPI в молекулярно-динамическом моделировании

Message Passing Interface (интерфейс передачи сообщений) – это библиотека подпрограмм обмена сообщениями и других подпрограмм, необходимых для организации обмена. Главное преимущество этой технологии – возможность использования высокопроизводительных вычислительных систем с любой архитектурой (с общей и распределенной памятью) [3 – 5]. В работе использовалась самая распространённая бесплатная реализация MPI – MPICH версии MPICH2-1.2.1p1.

Хотя в стандарте MPI предусматривается наличие более чем 120 подпрограмм (в языке Fortran они оформлены в виде процедур), для создания эффективного параллельного приложения бывает вполне достаточно десятка. Ниже перечислены основные процедуры библиотеки MPI, которые использовались при создании параллельного кода:

1. MPI_INIT (ierr);
2. MPI_COMM_SIZE (mpi_comm, nprocs, ierr);
3. MPI_COMM_RANK (mpi_comm, myrank, ierr);
4. MPI_BCAST (data, count, mpi_datatype, root, mpi_comm, ierr);
5. MPI_SCATTERV (data, countV, idisp, mpi_datatype, data_loc, count, mpi_datatype_loc, root, mpi_comm, ierr);
6. MPI_GATHERV (data_loc, count, mpi_datatype_loc, data, countV, idisp, mpi_datatype, root, mpi_comm, ierr);
7. MPI_FINALIZE (ierr).

Процедуры инициализации MPI окружения MPI_INIT, MPI_COMM_SIZE, MPI_COMM_RANK создают группу процессов и область взаимодействия (коммуникатор mpi_comm), определяют число процессов коммуникатора mpi_comm и записывают его в переменную nprocs, присваивают переменной myrank номер процесса от 0 до nprocs-1.

Процедуры коллективного обмена MPI_BCAST, MPI_SCATTERV, MPI_GATHERV осуществляют взаимодействие процессов и обмен данными между ними. Подпрограмма MPI_BCAST рассылает вектор данных data типа mpi_datatype от главного процесса под номером root всем остальным процессам комму-

никатора mpi_comm. Вектор данных data имеет размер count.

Подпрограмма MPI_SCATTERV разбивает сообщение (data) типа mpi_datatype от главного процесса под номером root на части произвольного размера и рассылает *i*-ю часть данных процессу с номером *i* (в том числе и самому себе). Начало расположения элементов блока, посылаемого *i*-му процессу, задается в массиве смещений idisp, а число посылаемых элементов в массиве countV. Например, если data это вектор, содержащий 100 элементов и эти данные нужно равномерно распределить между nprocs = 5 процессами, то countV=(20, 20, 20, 20, 20), idisp=(0, 20, 40, 60, 80). Таким образом,

процесс 0 получит часть данных data(1:20) которая запишется в data_loc(1:20);

процесс 1 получит часть данных data(21:40) которая запишется в data_loc(1:20);

процесс 2 получит часть данных data(41:60) которая запишется в data_loc(1:20);

процесс 3 получит часть данных data(61:80) которая запишется в data_loc(1:20);

процесс 4 получит часть данных data(81:100) которая запишется в data_loc(1:20).

Процедура MPI_GATHERV является обратной по отношению к MPI_SCATTERV и собирает блоки данных с разным числом элементов от каждого процесса. Количество элементов принимаемых от каждого процесса задается с помощью массива countV. Размещение отдельных блоков данных в главном процессе root, производится с помощью того же массива смещений idisp.

Последняя процедура MPI_FINALIZE закрывает все MPI-процессы и ликвидирует все области связи. В языке Fortran у всех рассмотренных процедур аргумент ierr – это код ошибки. Если процедура выполнена без ошибки параметр ierr = 0.

Структурную схему параллельной программы можно упрощенно описать следующим образом:

1) Включение заголовочных файлов MPI, инициализация MPI окружения. Инициализация системы взаимодействующих частиц (задание координат и скоростей частиц в начальный момент времени) осуществляется главным процессом (процессором) с локальным номером 0.

2) На каждом шаге временного цикла передаем всем процессам (с использованием процедуры MPI_BCAST) информацию о координатах всех частиц системы. Определяем число и индексы частиц, которые будет обштитывать каждый процесс из условия равномерной загрузки всех процессоров (каждый процесс обштитывает приблизительно одинаковое число частиц). С использованием подпрограммы MPI_SCATTERV разбиваем массивы координат и скоростей на блоки и распределяем эти данные между процессами.

3) Расчет равнодействующей силы, действующей на каждую из частиц и параметров траектории (новые координаты и скорости) проводится независимо для каждой группы частиц на разных процессорах или ядрах.

4) Обновленные значения параметров траектории с использованием MPI_GATHERV собираем и передаем главному процессу.

Этапы со второго по четвертый повторяются в цикле на каждом временном шаге.

3.2. Применение технологии OpenMP в молекулярно-динамическом моделировании

OpenMP – это интерфейс прикладного программирования для создания многопоточных приложений, предназначенных для параллельных вычислительных систем с общей памятью. OpenMP состоит из набора директив для компиляторов и библиотек специальных функций [3 – 5].

При использовании данной технологии код исходной последовательной программы можно распараллелить без существенной модификации с помощью установки специальных директив (в начале и в конце главного временного цикла) и вызова библиотечных функций, которые указывают компилятору способы распределения вычислений и данных между потоками. Главной идеей OpenMP является распараллеливание циклов, обычно несущих основную вычислительную нагрузку. Недостаток OpenMP в том, что он работает только в системах с общей памятью.

4. ТЕСТИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ КОДОВ

Два параллельных кода, использующих библиотеку параллельного программирования MPI и технологию OpenMP тестировались на вычислительной системе с общей памятью – четырехъядерном процессоре Intel Core i7. Моделировались системы из $N = 5 \times 10^5$ и 10^6 частиц. На рис. 5 показаны данные по относительному ускорению расчетов (по отношению к однопроцессорному варианту расчета) для OpenMP и MPI реализаций, а так же выполнено сравнение эффективности параллельных алгоритмов, использующих библиотеку параллельного программирования MPI и технологию OpenMP.

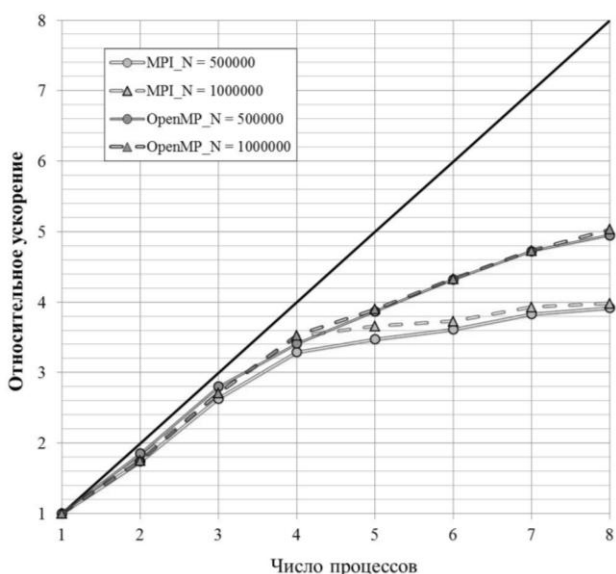


Рис. 5. Результаты тестирования производительности параллельных кодов

Сплошная линия соответствует случаю стопроцентного распараллеливания. Из графиков видно, что с ростом числа частиц эффективность распараллеливания возрастает в любом случае.

Хотя для рассматриваемой задачи и данной системы использование технологии OpenMP на многопроцессорных вычислительных системах с общей памятью более эффективно, чем библиотеки MPI (причем выигрыш в эффективности возрастает с ростом числа процессов от 1 до 8), применение библиотеки MPI позволяет проводить вычисления с использованием кластерных систем – наиболее мощных вычислительных ресурсов.

Применение современных технологий параллельного программирования дает возможность на несколько порядков увеличить размер моделируемой системы.

ЗАКЛЮЧЕНИЕ

В данной работе реализованы различные методы оптимизации алгоритмов молекулярно-динамического моделирования, основанные на модификации исходной модели взаимодействующих частиц и методы алгоритмической оптимизации, не требующие модификации модели. Созданы две параллельные реализации исходного последовательного кода предназначенного для построения двумерных и трехмерных, однородных и адаптивных неструктурированных расчетных сеток для различных задач механики сплошной среды с использованием библиотеки параллельного программирования MPI и технологии Open MP. Выполнена оценка эффективности параллельных алгоритмов по отношению к однопроцессорному варианту расчета для OpenMP и MPI реализаций. Проведено сравнение коэффициентов относительного ускорения при использовании MPI и OpenMP технологий. Для рассматриваемой задачи и данной системы с общей памятью (Intel Core i7) использование технологии OpenMP более эффективно, чем библиотеки MPI, причем выигрыш в эффективности возрастает с ростом числа процессов от 1 до 8.

Применение современных технологий параллельного программирования дает возможность на несколько порядков увеличить размер моделируемой системы. С ростом числа частиц эффективность распараллеливания возрастает.

Разработка специализированных методов, алгоритмов и высокопроизводительных программных средств для построения неструктурированных расчетных сеток является актуальной проблемой, так как для большинства современных задач вычислительной газовой динамики, представляющих практический интерес, форма расчетной области очень сложна. Поэтому процесс построения расчетной сетки для геометрии, максимально приближенной к реальной, весьма трудоемок и требует больших временных затрат. В работах [6 – 9] приводятся результаты численного моделирования обтекания перспективных летательных аппаратов сложной формы с использованием неструктурированных расчетных сеток.

Работа выполнена в Лаборатории радиационной газовой динамики ИПМех РАН в рамках проекта РФФИ № 10-01-00544 (разработка пространственной модели

движения химически реагирующего газа), гранта Президента РФ № МК-5324.2014.1 для государственной поддержки молодых российских ученых – кандидатов наук, в рамках международного Российско-Итальянского проекта РФФИ № 09-08-92422-КЭа (разработка компьютерных радиационно-столкновительных моделей) и программы сотрудничества РАН и CNR, а также в рамках Программы фундаментальных исследований РАН (создание моделей физико-химической кинетики высокотемпературных газовых потоков) и Программы министерства образования и науки Российской Федерации РНПВШ 2.1.1/4693 (создание гибридных радиационно-столкновительных моделей аэрофизики).

Автор выражает благодарность своему научному руководителю, чл.-корр. РАН, проф. С.Т. Суржикову и А.А. Горбунову

СПИСОК ЛИТЕРАТУРЫ

1. Железнякова А.Л., Суржиков С.Т. “Построение пространственных неструктурированных сеток для задач аэротермодинамики методом молекулярной динамики.” *Доклады Академии наук*. Т.439. №1 (2011): 42–47.
2. Verlet, L., “Computer experiments on classical fluids. I. Thermodynamic properties of Lennard–Jones molecules.” *Phys. Rev.* 159 (1967): 98–103.
3. Немнюгин С.А. “Средства программирования для многопроцессорных вычислительных систем.” Интернет – университет информационных технологий.
4. Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем. – СПб.: БХВ-Петербург, 2002. 400 с.
5. Гергель В.П. Теория и практика параллельных вычислений. Интернет – университет информационных технологий. – М: Бином, 2007. 424 с.
6. Котов М.А., Кузенов В.В. Создание сложных поверхностей гиперзвуковых летательных аппаратов системами САПР // Физико-химическая кинетика в газовой динамике. 2010. Т.9. <http://www.chemphys.edu.ru/pdf/2010-01-12-039.pdf>
7. Железнякова А.Л., Суржиков С.Т. Поле течения около космического аппарата Fire II под углом атаки // Физико-химическая кинетика в газовой динамике. 2010. Т.9. <http://www.chemphys.edu.ru/pdf/2010-01-12-020.pdf>
8. Железнякова А.Л., Кузенов В.В., Петрусев А.С., Суржиков С.Т. Расчет аэротермодинамики двух типов моделей спускаемых космических аппаратов // Физико-химическая кинетика в газовой динамике. 2010. Т.9. <http://www.chemphys.edu.ru/pdf/2010-01-12-025.pdf>
- Железнякова А.Л., Суржиков С.Т. Численное моделирование гиперзвукового обтекания модели летательного аппарата X-43 // Физико-химическая кинетика в газовой динамике. 2011. Т.11. <http://www.chemphys.edu.ru/pdf/2011-02-01-030.pdf>