

ИСПОЛЬЗОВАНИЕ АЛГЕБРАИЧЕСКОГО И ГЕОМЕТРИЧЕСКОГО ПОДХОДОВ К ДЕКОМПОЗИЦИИ РАСЧЕТНОЙ ОБЛАСТИ ПРИ МОДЕЛИРОВАНИИ ГАЗОДИСПЕРСНЫХ СРЕД

*С. В. Лашкин, А. С. Козелков, В. Ю. Герасимов, Д. К. Зеленский,
С. Н. Полищук, А. В. Ялозо*

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, г. Саров

Течения газа с частицами встречаются во многих промышленных приложениях, таких, как сжигание твердого и жидкого топлива, распылительная сушка и охлаждение, пескоструйная обработка, пневмотранспорт сыпучих материалов, циклонное сепарирование и т. д.

При моделировании движения твердых частиц в непрерывной среде, возникающих в задачах переноса вулканической пыли, переноса примесей в течении теплоносителей в реакторах АЭС [1–3] и т. д., применяются два подхода: Эйлера–Лагранжев и Эйлеров–Эйлеров. Реализация Эйлеров–Эйлерова подхода требует введения дополнительных уравнений для твердых частиц, имеющих разные параметры (диаметр, плотность и т. д.), что требует большого количества компьютерных ресурсов и, как следствие, значительного увеличения времени счета. Эйлеров–Лагранжев подход лишен этого недостатка и позволяет моделировать частицы с различными параметрами и получать при этом детальную информацию о взаимодействии частиц друг с другом, стенками, а также с непрерывной фазой. Одним из существенных ограничений данного подхода является его параллельная реализация.

Если рассматривать чистый Лагранжев подход (без учета непрерывной фазы), то его особенности полностью изучены и не представляют особых трудностей в реализации, в том числе и параллельной. Однако при введении непрерывной фазы, включающей использование сеточной модели, появляются существенные сложности в параллельной реализации Эйлеров–Лагранжева подхода. Трудность заключается в наличии достаточно больших параллельных массивов, учитывающих взаимодействие частиц между собой и непрерывной средой. На сегодняшний момент не существует идеального способа решения проблемы эффективной параллельной реализации и при моделировании Эйлеров–Лагранжевой многофазности применяют два взаимоисключающих подхода: алгебраический и геометрический. Главной особенностью алгебраического подхода является равномерное распределение частиц по счетным процессорам, называемым распределением по времени. Прямо противоположный подход – геометрический, где расчет частицы проводится на том же процессоре, в котором находится ячейка сетки, называют распределением «по месту». Алгоритмы хорошо представлены и описаны только при бессеточном Лагранжевом моделировании дискретных фаз, но в присутствии сетки и с наличием непрерывной среды проблемы применения данных подходов, как показывает изученная литература, практически не исследованы.

В данном докладе рассматриваются вопросы решения задач движения твердых частиц в непрерывной среде – Эйлеров–Лагранжев подход. Особое внимание уделено проблеме параллельной реализации алгебраического и геометрического подходов. Проведено сравнение эффективности применения подходов на тестовых задачах, как по общему времени счета задачи, так и по необходимой памяти.

Методика для численного моделирования течений с частицами

Для моделирования многофазных смесей, состоящих из жидкостей или газа и взвешенных в них твердых частиц или капель, используется ряд подходов, таких как континуальный, траекторный, кинетический и многие другие [4]. Каждый из данных подходов основан на совместном использовании сеточной модели и модели лагранжевых частиц. В рамках комплекса ЛОГОС, разработана структура памяти частиц на неструктурированной эйлеровой сетке. Предполагается реализовать методику расчета многофазных течений на неструктурированной сетке на основе модели лагранжевых частиц.

Модели расчета течения

Обсуждается лагранжево описание движения дисперсной фазы. На данный момент реализованы три возможных режима движения частиц после вхождения в расчетную область:

- 1) псевдодвижение;
- 2) движение по потоку;
- 3) движение под действием внешних сил.

Кинематическое соотношение, позволяющее вычислить радиус-вектор центра масс частицы одинаково для всех трех режимов:

$$\frac{d\vec{r}_p}{dt} = \vec{v}_p, \quad (1)$$

где \vec{r}_p , \vec{v}_p – координаты и скорость частицы с индексом « p ».

Режимы отличаются между собой только в записи правой части уравнений импульса.

Для режима «псевдодвижение» вновь появляющиеся частицы имеют свою начальную скорость и благодаря этой скорости движутся. Учитывается влияние внешних сил (влияние гравитации) на изменение скорости.

Уравнение импульса записывается следующим образом:

$$\frac{d\vec{v}_p}{dt} = \frac{\vec{g}(\rho_p - \rho)}{\rho_p}, \quad (2)$$

где ρ_p , \vec{v}_p – плотность, скорость частицы с индексом « p », ρ плотность непрерывной фазы, \vec{g} – ускорение свободного падения.

Для движения «по потоку» вновь появляющиеся частицы не имеют своей начальной скорости, то есть находятся в начальный момент в состоянии покоя. После входа в расчетную область все частицы имеют скорость, равную скорости потока в ячейке, где находится частица, что требует локализации частицы в данный момент времени.

$$\vec{v}_p = \vec{v}, \quad (3)$$

где \vec{v}_p – плотность, диаметр, масса, скорость частицы с индексом « p », ρ , \vec{v} – плотность и скорость непрерывной фазы в ячейке, где находится частица.

Для движения «с учетом всех сил» вновь появляющиеся частицы имеют свою начальную скорость.

Учитываются все внешние силы (такие, как сила гравитации) и сила сопротивления движению. Сила сопротивления движению со стороны непрерывной фазы имеет следующий вид [5]:

$$F_D = \frac{18\mu}{\rho_p d_p^2} \frac{C_D \text{Re}}{24}; \quad \text{Re} = \frac{\rho_p d_p |\vec{v}_p - \vec{v}|}{\mu}, \quad (4)$$

где ρ_p , d_p , \vec{v}_p – плотность, диаметр, скорость частицы с индексом « p », μ – плотность, вязкость, скорость непрерывной фазы в ячейке, где находится частица, C_D – коэффициент сопротивления среды.

Коэффициент сопротивления среды C_D вычисляется по формулам в зависимости от типа используемого сопротивления. При дальнейшем рассмотрении будем предполагать, что все частицы имеют сферическую форму.

Сферический закон сопротивления [6]:

$$C_D = a_1 + \frac{a_2}{\text{Re}} + \frac{a_3}{\text{Re}^2}, \quad (5)$$

где коэффициенты известны в зависимости от числа Рейнольдса.

Несферический закон сопротивления[7]:

Для несферических частиц. Зависит от формы моделируемой частицы.

Фактор формы для частиц вычисляется по формуле:

$$\phi = \frac{s}{S}, \quad (6)$$

где s – площадь поверхности сферы, имеющей такой же объем, как и частица, S – площадь поверхности частицы.

$$C_D = \frac{24}{\text{Re}} \left(1 + b_1 \text{Re}_{sph}^{b_2} \right) + \frac{b_3 \text{Re}_{sph}}{b_4 + \text{Re}_{sph}}; \quad (7)$$

$$\begin{cases} b_1 = \exp(2,3288 - 6,4581\phi + 2,4486\phi^2), \\ b_2 = 0,0964 + 0,5565\phi, \\ b_3 = \exp(2,3288 - 13,8944\phi + 18,4222\phi^2 - 10,2599\phi^3), \\ b_4 = \exp(2,3288 + 12,2584\phi - 20,7322\phi^2 + 15,8855\phi^3), \end{cases} \quad (7.1)$$

где Re_{sph} вычисляется с использованием диаметра сферы одинакового объема с частицей.

Закон сопротивления Стокса–Кунигэма[8]:

$$F_D = \frac{18\mu}{\rho_p d_p^2 C_c}; \quad C_c = 1 + \frac{2\lambda}{d_p} \left(1,257 + 0,4 \exp\left(\frac{-1,1d_p}{2\lambda}\right) \right), \quad (8)$$

где λ – средняя длина свободного пробега молекулы.

Динамический закон сопротивления[9]:

$$C_D = \begin{cases} 0,424, & \text{Re} \geq 1000, \\ \frac{24}{\text{Re}} \left(1 + \frac{1}{6} \text{Re}^{2/3} \right), & \text{Re} \leq 1000, \end{cases} \quad (9)$$

Сила сопротивления движению F_D со стороны непрерывной фазы может быть записана и в следующем виде[10]:

$$\begin{cases} F_D = \frac{1}{\tau} \\ \tau = \frac{m_p}{3\pi\mu d_p} \end{cases} \rightarrow F_D = \frac{3\pi\mu d_p}{m_p}, \quad (10)$$

где d_p , m_p – диаметр, масса частицы с индексом « p », μ – вязкость непрерывной фазы в ячейке, где находится частица.

Для такого типа движения уравнение импульса записывается следующим образом [5]:

$$\frac{d\vec{v}_p}{dt} = F_D (\vec{v} - \vec{v}_p) + \frac{\vec{g}(\rho_p - \rho)}{\rho_p}, \quad (11)$$

где ρ_p , \vec{v}_p – плотность, скорость частицы с индексом « p », ρ , \vec{v} – плотность и скорость непрерывной фазы в ячейке, где находится частица, \vec{g} – ускорение свободного падения, F_D – сила сопротивления.

Обработка взаимодействия «частица»-«граница ячейки»

Обрабатывается несколько типов взаимодействия «частица»-«граница ячейки»:

а) внутренняя грань.

Частицы проходят сквозь грань без изменений.

б) Внешняя грань.

1) Грань вылета.

Если координата частицы оказывается за пределами грани вылета, то она удаляется из расчетных, далее считается, что она покинула расчетную зону.

2) Грань с прилипанием.

Если частица оказывается на такой грани, то после этого она перестает двигаться. Для других частиц эта частица все еще существует. Так что взаимодействие ее с другими частицами остается.

3) Грань с отскоком.

Частица, попадающая на эту грань, изменяет свою скорость согласно алгоритму:

- находим нормаль к границе \vec{n} ;
- вычисляем нормальную и тангенциальную составляющие скорости;
- по формулам восстановления скоростей:

$$\begin{cases} \vec{v}'_n = -k_n \vec{v}_n \\ \vec{v}'_t = k \vec{v}_t \end{cases} \rightarrow \vec{v}' = \vec{v}'_n + \vec{v}'_t. \quad (12)$$

4) Грань с зеркальным отражением.

Частица, попадающая на эту грань, изменяет направление своей скорости, но не модуль, согласно алгоритму:

1. Находится угол α между старым направлением скорости \vec{t} и нормалью к границе \vec{n} .
2. Далее составляется и решается система уравнений:

$$\begin{pmatrix} t_x & t_y & t_z \\ [\vec{n}\vec{t}]_x & [\vec{n}\vec{t}]_y & [\vec{n}\vec{t}]_z \\ n_x & n_y & n_z \end{pmatrix} \vec{t}' = \begin{pmatrix} -\cos 2\alpha \\ 0 \\ -\cos \alpha \end{pmatrix}. \quad (13)$$

3. Получив \vec{t}' , получаем новое направление и значение скорости.

Влияние частиц на движение среды

Влияние частиц на среду, в которой они движутся, учитывается при помощи источников в правую часть уравнений Навье–Стокса в тех ячейках, где побывала частица на своем пути.

Источники вычисляются в каждой ячейке, через которую прошла хотя бы одна частица за данный временной период, по следующим формулам:

$$\begin{cases} \vec{S}_{V,Cell} = \sum_p \frac{m_{p,init} \vec{v}_{p,old} - m_{p,new} \vec{v}_{p,new}}{\tau_t}; \\ S_{P,Cell} = \sum_p \frac{m_{p,init} - m_{p,new}}{\tau_t}; \\ S_{E,Cell} = \sum_p \frac{E_{p,init} - E_{p,new}}{\tau_t}, \end{cases} \quad (14)$$

где $m_{p,init}$, $\vec{v}_{p,old}$, $E_{p,init}$ – параметры частицы: масса, скорость, энергия в начале временного шага, $m_{p,new}$, $\vec{v}_{p,new}$, $E_{p,new}$ – параметры частицы в конце шага, τ_t – время, которое частица « p » находилась в данной ячейке.

Модель инжекторов частиц

Необходимо отметить, что для реализации счетного алгоритма, как в скалярном режиме, так и в параллельном, необходимо добавлять частицы на процессорах. Добавляемые частицы записываются в конец массивов.

Типы инжекторов:

- а) одиночная частица;
- б) группа частиц;
- в) частицы, расположенные на определенной поверхности или «патче»;
- г) частицы, расположенные в конусе;
- д) частицы, расположенные в усеченном конусе;
- е) частицы, расположенные в зазоре между внешним и внутренним конусами.

Начальное расположение частиц

Если задается одиночная частица, то задаются координаты частицы.

Координаты группы частиц задаются линейным распределением через минимальные и максимальные координаты (X, Y, Z) . Приведем формулу для X .

$$x_i = x_{\min} + \frac{x_{\max} - x_{\min}}{N-1} i; \quad i = \overline{0, N-1}. \quad (15)$$

Координаты группы частиц могут задаваться также с помощью трех фигур:

- а) конус:
 - характеризуется осью, диаметром и углом при вершине;
- б) усеченный конус:
 - характеризуется осью, диаметром, фактором усечения и углом при вершине;
- в) форсунка (возможно усечение):
 - характеризуется осью, диаметрами (внутренним и внешним), фактором усечения и внутренним углом при вершине.

Особенности параллельной реализации. Параллельная реализация ПК ЛОГОС

В комплексе ЛОГОС расчетная область описывается неструктурированной эйлеровой сеткой. При декомпозиции задачи на процессоры производится геометрическое разбиение сетки на процессорные блоки. В результате чего на каждом процессоре формируется фрагмент расчетной сетки. На границах смежности процессоров вводятся слои обменных ячеек, которые являются фиктивными (для счета) и используются в межпроцессорных обменах. Общая идеология и организация данных обменов подробно описана в работе [11].

Существующие методы параллельной реализации расчета движения частиц

Хотя число модельных частиц всегда на много порядков меньше количества частиц, образующих исходную среду, тем не менее, в абсолютном выражении оно обычно достаточно велико. Это связано с тем, что точность методов частиц возрастает с ростом количества частиц N , хотя и не по линейному закону. По мере совершенствования компьютеров типичное количество частиц, исполь-

зуемых в рядовых расчетах, возрастало в пределах $N = 10^3 - 10^6$. Трудоемкость чисто лагранжевых методов оценивается величиной $O(N^2)$. Хорошие алгоритмы комбинированных методов частиц-в-ячейках обычно являются экономичными, и их трудоемкость имеет порядок $O(N)$ или $O(N \ln N)$.

Отсюда можно заключить, что для всех методов частиц характерны большие объемы вычислительной работы. Для их реализации нужны компьютеры с большими памятью и быстродействием. На трехмерных и двумерных нестационарных задачах необходимые вычислительные ресурсы достигают уровня современных суперкомпьютеров. Потому до недавнего времени методы частиц не получали широкого распространения на практике и развивались относительно небольшим числом групп вычислителей.

Важное технологическое достоинство этих методов заключается в том, что движение частиц в каждой ячейке эйлеровой сетки определяется только локальными значениями полей и не зависит от координат и скоростей других частиц. Именно независимость движения частиц друг от друга на лагранжевом этапе лежит в основе распараллеливания алгоритма.

Распараллеливание позволяет использовать современные многопроцессорные компьютеры и существенно сократить время решения задачи в расчете на одну частицу. Это, в свою очередь, дает возможность увеличить количество модельных частиц и тем самым повысить точность расчетов, что необходимо для исследования тонких физических процессов.

Однако при использовании многопроцессорных компьютеров для моделирования методами частиц-в-ячейках возникает проблема равномерной загрузки процессоров. Это связано с тем, что в них массивы данных делятся на две группы. Одна содержит данные о частицах — координаты, компоненты скоростей, массы, заряды и другие характеристики частиц. Объем этих массивов пропорционален количеству модельных частиц. Другая часть — массивы значений сеточных функций плотности, давления, напряженностей электрических и магнитных полей и т. п. Если при распараллеливании на каждый процессор отводятся одинаковые части сеточной области, то объемы вычислений на отдельных процессорах могут сильно различаться из-за разного количества частиц, приходящихся на них. Если же сеточную область в начальный момент разделить так, чтобы уравнять количества частиц, приходящиеся на каждый процессор, то в процессе вычислений из-за движения частиц из одной подобласти в другую нагрузка на отдельные процессоры также может существенно отклоняться от средней. Поэтому для оптимизации расчетов движения частиц на многопроцессорных компьютерах необходимо использовать специальные приемы балансировки загрузки [12–13].

Важной особенностью рассматриваемых задач является расчет частиц. Обычно расчет перемещения частиц занимает гораздо большее время, чем расчет уравнений для непрерывной фазы. В ряде задач расчет методом частиц может занимать до 90 % общего времени решения задачи.

Расчетом частицы будем называть вычисление расчетных переменных, характеризующих частицу, для фиксированной частицы. Все частицы связаны с ячейками исходной сетки задачи. По координатам частицы однозначно можно определить номер ячейки сетки, в которой эта частица находится в текущий момент моделирования.

Для проведения расчета частицы необходимы значения расчетных переменных частицы с предыдущего временного шага моделирования, а также значения из некоторого числа ячеек исходной сетки задачи.

Для параллельного выполнения расчета частиц необходимо решить следующие задачи:

- 1) распределить частицы по процессорам вычислительной системы, на которой решается задача;
- 2) обеспечить расчет частиц всеми необходимыми исходными данными, которые используются при расчете частиц;

3) разместить вычисленные при расчете частиц значения в требуемых ячейках.

Однако есть три существенных момента, которые необходимо учитывать:

- 1) число частиц меняется в процессе решения задачи;
- 2) расчет каждой частицы незначительно, но отличается (по времени) от расчета других частиц;
- 3) частицы в процессе счета могут перемещаться из ячейки в ячейку.

Расчет частицы можно проводить на том же процессоре, в котором находится ячейка сетки, содержащая данную частицу. Далее такое распределение будем называть распределением «по месту» [14–16].

1) Второй подход к распределению частиц учитывает время счета задачи на предыдущем шаге решения и общее число частиц. Далее такое распределение будем называть распределением «по времени» [17–19].

2) Частный случай такого подхода состоит в том, чтобы, зная общее количество частиц, равномерно распределить их по всем процессорам, на которых решается задача. Такое распределение частиц в дальнейшем будем называть равномерным распределением [19].

Все вышеназванные методы имеют свои недостатки и преимущества в зависимости от поставленной задачи. Далее будет проведено тестирование эффективности равномерного (алгебраического) и геометрического подходов.

Организация межпроцессорных обменов для различных подходов к распределению частиц при параллельной реализации

В модели лагранжевых частиц дополнительно к первому (обмен величинами в ячейках) используется второй тип обменов – обмен частицами.

Реализация обмена частицами при различных подходах к распределению частиц на процессоры существенно отличается.

Организация межпроцессорных обменов для «геометрического» распределения частиц

Рассмотрим сначала межпроцессорные обмены при геометрическом распределении частиц, то есть когда частицы распределяются на тот процессор, которому принадлежит ячейка, в которой находится частица.

Для того чтобы передать частицу при таком распределении, необходимо знать номер частицы в массиве частиц и номер процессора, на котором находится новая ячейка.

После расчета движения всех частиц на итерации выполняем процедуру проверки флага у всех частиц. И, если флаг частицы выставлен на передачу, добавляем номера частиц в массив адресатов с соответствующим номером процессора.

После этого запускается процедура обмена частицами между процессорами. Когда частица оказывается на новом процессоре, флаг передачи снимается.

После завершения процедуры обмена все частицы с флагом на передачу удаляются со старых процессоров.

Организация межпроцессорных обменов для «алгебраического» распределения частиц

Намного более трудоемкой операцией выглядит реализация «алгебраического» подхода к распределению частиц. Суть в том, что частицы распределяются вне зависимости от того, где они находятся в геометрическом плане. Распределение производится равномерно по всем процессорам. Но так как для решения задачи движения квазичастицы необходимы данные о непрерывной фазе, то необходимо организовать передачу на процессор, где находится частица, следующих параметров:

- геометрия,
- физические параметры для всех ячеек, находящихся в соседстве через ребро с ячейкой, в которой находится частица.

Для оптимизации времени и количества пересылок возможно копировать ячейки с заданным числом соседних слоев.

После начального распределения частиц организуется процедура передачи и хранения тех ячеек (с соседями через ребро), в которых первоначально оказываются частицы.

На каждом шаге интегрирования проводится проверка наличия всех ячеек, которые необходимы для корректного расчета задачи движения частиц. Если каких-то ячеек не хватает, то запускается процедура пересылки соответствующих ячеек и их соседей.

Алгоритм пересылки проходит следующие этапы:

- пересылка источников (для всех обменных ячеек);
- проверка наличия ячеек (геометрически и физически).

Если в обменном массиве на процессоре, который отвечает за частицу, нужная ячейка отсутствует, то ставится флаг полной передачи ячейки, если же ячейка есть, то флаг изменяется на передачу только физических параметров.

• Запускается процедура удаления ненужных ячеек, тех, которые не участвуют в дальнейшем перемещении частицы и не входят в один из слоев соседей.

• Номера необходимых ячеек с флагами отправляются на необходимый процессор. Проводится процедура сборки массива отправки согласно флагам передачи с ячейками под номерами из запроса и соседними ячейками (из соседних слоев).

• Пересылаются геометрические и физические характеристики ячейки для ячеек с флагом «полная передача».

• Пересылаются необходимые грани.

• Для ячеек с флагом «передача физических параметров» передача физических параметров.

Все эти параметры сохраняются на процессоре в специально созданных для этого объектах. После чего организуется максимально быстрый доступ информации о нужной ячейке через эти объекты.

Литература

1. Costa A., Macedonio G., Folch A. A three-dimensional Eulerian model for transport and deposition of volcanic ashes // *Earth and Planetary Science Lett.* 2006. Vol. 241. P. 634–647.
2. Macedonio G., Costa A., Longo A. A computer model for volcanic ashfallout and assessment of subsequent hazard // *Computers & Geosciences.* 2005. Vol. 31. P. 837–845.
3. Молчанов П. А. Моделирование пристеночной плазмы сферического токамака MAST: автореферат диссертации. Санкт-Петербург, 2010.
4. Волков К. Н., Емельянов В. Н. Течения газа с частицами. М.: ФИЗМАТЛИТ, 2008.
5. The ANSYS FLUENT Theory Guide. Chapter 16. 2010.
6. Morsi S. A., Alexander A. J. An Investigation of Particle Trajectories in Two-Phase Flow Systems // *J. Fluid Mech.* 1972. 55(2). September 26. P. 193–208.
7. Haider A., Levenspiel O. Drag Coefficient and Terminal Velocity of Spherical and Nonspherical Particles // *Powder Technology.* 1989. Vol. 58. P. 63–70.
8. Ounis H., Ahmadi G., McLaughlin J. B. Brownian Diffusion of Submicrometer Particles in the Viscous Sublayer // *Journal of Colloid and Interface Science.* 1991. 143(1). P. 66–277.
9. Liu A. B., Mather D., Reitz R. D. Modeling the Effects of Drop Drag and Breakup on Fuel Sprays // SAE Technical Paper 930072. SAE. 1993.
10. Ветошкин А. Г. Теоретические основы защиты окружающей среды. [Электронный ресурс]. Режим доступа: http://mobile.uchebniki.ws/14650717/ekologiya/gravitatsionnoe_osazhdenie_chastits_aerозoley.
11. Григорьев Ю. Н., Вшивков В. А., Федорук М. П. Численное моделирование методами частиц-в-ячейках. Новосибирск: Изд-во СО РАН, 2004.
12. Vshivkov V. A., Kraeva M. A., Malyshkin V. E. Parallel Implementation of the Particle-in-Cell Method // *Programming and Computer Software.* 1997. Vol. 23, N 2. P. 87–97.

13. Старченко А. В., Ибраев Г. М. Опыт создания вычислительного метакластера на базе кластерных систем Томского научного центра / Под ред. Старченко // IV Сибирская школа-семинар по параллельным вычислениям. Томск, 9–11 октября 2007 г. С. 61–78.

14. Minty E., Davey R., Simpson A., Henty D. Decomposing the Potentially Parallel – The Edinburgh Parallel Computing Centre. [Электронный ресурс]. Режим доступа: <http://www.epcc.ed.ac.uk>.

15. Трунов А. А., Старченко А. В., Турчановский И. Ю., Шкляев В. А. Параллельная реализация алгоритма решения задачи транспортировке пучка заряженных частиц / Под ред. Старченко // IV Сибирская школа-семинар по параллельным вычислениям. Томск, 9–11 октября 2007 г. С. 234–241.

16. Хокни Р., Иствуд Дж. Численное моделирование методом частиц. М.: Мир, 1987.

17. Берендеев Е. А., Ефимова А. А. Реализация эффективных параллельных вычислений при моделировании больших задач физики плазмы методом частиц в ячейках // Труды Международ. научной конф. «Параллельные вычислительные технологии» (ПаВТ-2012). Новосибирск, 2012. Р. 380–385.

18. Андрианов А. Н., Ефимкин К. Н. Подход к параллельной реализации метода частиц в ячейках: Препринт № 9. М.: ИПМ им. М. В. Келдыша, 2009. [Электронный ресурс]. Режим доступа: <http://library.keldysh.ru/preprint.asp?id=2009-9>.

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ РАЗДЕЛЕННОГО И СОВМЕЩЕННОГО ПОДХОДОВ ЧИСЛЕННОГО РЕШЕНИЯ УРАВНЕНИЙ НАВЬЕ – СТОКСА В ЗАДАЧАХ ГИДРОДИНАМИКИ

С. В. Лашкин, А. С. Козелков, А. В. Ялозо, Н. В. Тарасова

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, г. Саров

Эффективность решения большинства задач вычислительной гидродинамики зависит от выбора алгоритма решения системы уравнений Навье–Стокса, который может быть основан на решении полной системы уравнений Навье–Стокса без расщепления или на алгоритме расщепления по физическим процессам, подразумевающим решение отдельного уравнения для давления.

Аспекты выбора того или иного подхода рассмотрены в многочисленных статьях и монографиях, и существенно зависят от физики моделируемого течения, необходимой точности расчета и имеющихся вычислительных ресурсов. Однако всем алгоритмам присуща одна и та же проблема, а именно скорость сходимости зависит от количества ячеек вычислительной сетки. Даже несмотря на успешное решение некоторых проблем, связанных с решателями систем линейных уравнений (введение многосеточных технологий), параллельными вычислениями, геометрической инициализации и декомпозицией расчетной области), проблемы сходимости все еще решены не полностью. Это касается обоих подходов к дискретизации системы уравнений Навье–Стокса.

Последние успехи в этом направлении связаны с разработкой совмещенного алгоритма, основанного на алгоритме расщепления системы уравнений Навье–Стокса с последующим формированием общей системы алгебраических уравнений [1, 2]. Совмещение уравнений для компонент вектора скорости и давления осуществляется за счет введения неявных слагаемых для градиента давления и массового потока в уравнения сохранения количества движения и неразрывности. Получаемые таким образом неявные коэффициенты суммируются в связанную диагонально-доминантную матрицу. В немногочисленных работах, посвященных разработке и сравнению совмещенного и разделенного алгоритмов, показывается впечатляющая скорость сходимости совмещенного решателя и почти одинаковое время расчета одного контрольного объема, не зависящего от размера сетки, что для разделенного решателя недостижимо. Однако преимущество разделенного решателя показано на простых тестовых задачах и при решении промышленно-ориентированных задач, где опре-