

УДК 004.92+532.5

doi 10.26089/NumMet.v16r117

О РЕАЛИЗАЦИИ ВИХРЕВЫХ МЕТОДОВ МОДЕЛИРОВАНИЯ ДВУМЕРНЫХ ТЕЧЕНИЙ НЕСЖИМАЕМОЙ СРЕДЫ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ CUDA

С. Р. Гречкин-Погребняков¹, К. С. Кузьмина², И. К. Марчевский³

Исследована возможность ускорения вычислений в методе вихревых элементов — бессеточном лагранжевом методе вычислительной гидродинамики — за счет использования графических ускорителей. Реализован алгоритм, основанный на авторской модификации метода вихревых элементов, позволяющий проводить все необходимые вычисления непосредственно на графическом ускорителе с использованием технологии CUDA. Скорость решения типичной задачи на одном ускорителе GeForce GTX 970 или Tesla C2050 оказывается соизмеримой со скоростью решения аналогичной задачи на кластере, содержащем 30–40 ядер, при использовании технологии MPI. Полученные результаты говорят о высокой эффективности применения графических ускорителей при решении задач гидродинамики вихревыми методами.

Ключевые слова: графические ускорители, технология CUDA, параллельные вычисления, уравнения Навье–Стокса, вязкая несжимаемая среда, метод вязких вихревых доменов, метод дискретных вихрей, задача Блазиуса.

1. Введение. Вычислительный эксперимент в механике сплошной среды занимает сегодня центральное место в исследовании физических процессов и технических систем. Всё возрастающие требования к точности моделирования при сохранении приемлемого времени выполнения расчетов служат на протяжении последних десятилетий мощным стимулом развития как численных методов, так и вычислительной техники.

Задачи вычислительной гидро- и газодинамики, а также сопряженные задачи гидроаэроупругости являются одними из наиболее сложных с вычислительной точки зрения в силу нелинейности исходных математических моделей и особенностей их численного анализа (в частности, необходимости расчета течений в пограничных слоях, в которых на малом расстоянии происходит значительное изменение параметров течения). Выбор подхода к решению задачи вычислительной гидромеханики не всегда является очевидным; существует значительное количество различных методов, которые упрощенно можно разделить на три класса: эйлеровы, в которых расчет ведется на неподвижной сетке; лагранжевы, в которых сетка движется вместе с частицами среды (либо вообще отсутствует в явном виде); и гибридные эйлерово-лагранжевые подходы, в определенной мере соединяющие в себе преимущества методов обоих классов. При решении задач гидроупругости предпочтительными являются лагранжевы и гибридные методы, поскольку область, занятая жидкостью или газом, изменяется на каждом шаге расчета: ее форма зависит от деформации и перемещения обтекаемых тел, которые, в свою очередь, определяются действующими на них гидродинамическими нагрузками.

Существует весьма узкий, однако важный с практической точки зрения класс задач, предполагающий моделирование течений с малыми дозвуковыми скоростями, когда влиянием сжимаемости среды можно пренебречь. Если область, в которой завихренность отлична от нуля, сравнительно невелика (что обычно наблюдается при моделировании внешнего обтекания тел), то для решения таких задач могут быть особенно эффективны бессеточные лагранжевы вихревые методы: в случае решения плоских задач — метод дискретных вихрей [1, 2] и метод вязких вихревых доменов [3]; для моделирования пространственных течений — метод вихревых рамок [2, 4], метод вихревых отрезков [5] и их модификации.

Вопросам ускорения вычислений в вихревых методах посвящено достаточное количество работ; основные пути повышения скорости счета — использование приближенных быстрых методов [6] и применение параллельных вычислительных алгоритмов с использованием технологии MPI [7, 8]. Получаемые

¹ Московский государственный технический университет им. Н. Э. Баумана, факультет фундаментальных наук, ул. 2-я Бауманская, д. 5, 105005, Москва; студент, e-mail: grechkin.pogrebnyakov@gmail.com

² Московский государственный технический университет им. Н. Э. Баумана, факультет фундаментальных наук, ул. 2-я Бауманская, д. 5, 105005, Москва; студент, e-mail: kuz-ksen-serg@yandex.ru

³ Московский государственный технический университет им. Н. Э. Баумана, факультет фундаментальных наук, ул. 2-я Бауманская, д. 5, 105005, Москва; доцент, e-mail: iliamarchevsky@mail.ru

результаты расчетов показывают хорошую масштабируемость алгоритмов вихревых методов. В работе [8] показано, что при распараллеливании всех основных операций удается добиться ускорения в 20–30 раз при проведении расчетов на 64-ядерном кластере. В то же время публикации, посвященные систематическим исследованиям в области проведения расчетов вихревыми методами с использованием графических ускорителей и технологии nVidia CUDA, практически отсутствуют; отдельные сведения приведены в [9, 10]. Отметим, что в активно развиваемых за рубежом гибридных методах расчеты на графических ускорителях применяются весьма успешно [11].

Целью настоящей работы является разработка модификации алгоритма метода вязких вихревых доменов, адаптированной под использование графических ускорителей и предназначенной для расчета двумерных течений несжимаемой среды и моделирования обтекания профилей, и исследование эффективности ее применения.

2. Постановка задачи. Рассматривается двумерная задача о моделировании внешнего обтекания жесткого неподвижного профиля потоком вязкой несжимаемой среды постоянной плотности, движение которой описывается уравнением неразрывности и уравнением Навье–Стокса:

$$\nabla \cdot \mathbf{V} = 0, \quad (1)$$

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\frac{\nabla p}{\rho} + \nu \Delta \mathbf{V}. \quad (2)$$

Здесь $\mathbf{V} = \mathbf{V}(\mathbf{r}, t)$ — скорость среды, $p = p(\mathbf{r}, t)$ — давление, $\rho = \text{const}$ — плотность, $\nu = \text{const}$ — кинематическая вязкость, ∇ и Δ — дифференциальные операторы Гамильтона и Лапласа. Начальное распределение скоростей в среде полагается известным; в качестве граничных условий используются условие затухания возмущений на бесконечности: $\mathbf{V}(\mathbf{r}, t) \rightarrow \mathbf{V}_\infty$, $p(\mathbf{r}, t) \rightarrow p_\infty$ при $|\mathbf{r}| \rightarrow \infty$, и условие прилипания на границе K профиля: $\mathbf{V}(\mathbf{r}, t) = 0$ при $\mathbf{r} \in K$.

При проведении практических расчетов в зависимости от решаемой задачи могут представлять интерес как нестационарное поле скоростей среды и поле давления в ней, так и распределение давления по обтекаемому профилю, а также интегральные величины действующих на профиль со стороны потока нагрузок — лобового сопротивления, подъемной силы и аэродинамического момента.

3. Краткое описание вихревых методов вычислительной гидродинамики. Использование вихревых методов предполагает переход к завихренности $\Omega = \nabla \times \mathbf{V}$ как к первичной расчетной величине, при этом в плоских задачах гидродинамики вектор Ω имеет лишь одну ненулевую компоненту. Поле скоростей может быть восстановлено по известному распределению завихренности при помощи закона Био–Савара

$$\mathbf{V}(\mathbf{r}, t) = \mathbf{V}_\infty + \iint_S \frac{\Omega(\boldsymbol{\xi}, t) \times (\mathbf{r} - \boldsymbol{\xi})}{|\mathbf{r} - \boldsymbol{\xi}|^2} dS_\xi, \quad (3)$$

где интеграл берется по области течения S ; при этом уравнение неразрывности (1) выполняется автоматически. Для расчета давления наиболее удобен аналог интегралов Бернулли и Коши–Лагранжа [12], при этом для вычисления интегральных значений гидродинамических нагрузок, действующих на обтекаемый профиль (главного вектора сил и момента), можно использовать весьма простые выражения, также приведенные в [12].

В терминах завихренности уравнение (2) принимает простой вид

$$\left. \frac{D\Omega}{Dt} \right|_{\mathbf{V}} = \nu \Delta \Omega, \quad (4)$$

где $\left. \frac{D}{Dt} \right|_{\mathbf{V}} = \frac{\partial}{\partial t} + (\mathbf{V} \cdot \nabla)$ — субстанциональная (материальная) производная. В случае моделирования течения идеальной (невязкой) несжимаемой среды уравнение (4) выражает собой закон Гельмгольца, в соответствии с которым завихренность “вморожена” в жидкость, а эволюция завихренности, таким образом, представляет собой ее перенос вместе с частицами жидкости.

Отметим, что во многих инженерных приложениях, в частности при моделировании обтекания профилей, имеющих острые кромки и угловые точки, при проведении моделирования течений и расчете нагрузок можно ограничиться моделью идеальной жидкости. Данный подход положен в основу известного метода дискретных вихрей (МДВ) [1, 2], активно развиваемого еще с 60-х годов и хорошо зарекомендовавшего себя в целом ряде отраслей.

Если влияние вязкости существенно и требуется ее корректный учет в расчете, то возможен расчет так называемой диффузионной скорости $\mathbf{W} = -\nu \frac{\nabla \Omega}{\Omega}$, эффективный способ вычисления которой разработан

в методе вязких вихревых доменов (ВВД) [3, 12]. С учетом диффузионной скорости уравнение (4) можно привести к виду

$$\frac{D\Omega}{Dt} \Big|_{\mathbf{V}+\mathbf{W}} = 0, \tag{5}$$

где по аналогии с субстанциональной производной введено обозначение $\frac{D}{Dt} \Big|_{\mathbf{V}+\mathbf{W}} = \frac{\partial}{\partial t} + ((\mathbf{V} + \mathbf{W}) \cdot \nabla)$.

Механический смысл уравнения (5) заключается в том, что в области течения S происходит перенос имеющейся завихренности со скоростью $\mathbf{V} + \mathbf{W}$, при этом “новая” завихренность внутри области течения не образуется. Генерация завихренности происходит лишь на границе области течения, т.е. на обтекаемом профиле.

Вихревые методы относятся к так называемым методам частиц (particle methods) [13], при этом в качестве таких частиц выступают элементарные поля завихренности — вихревые элементы, при помощи которых моделируется распределение завихренности в области течения. Общее количество N вихревых элементов может быть достаточно большим и достигать десятков-сотен тысяч и даже миллионов; сами вихревые частицы в методах МДВ и ВВД представляют собой круглые вихри постоянного достаточно малого радиуса ε .

Положение каждого вихревого элемента в пространстве характеризуется радиус-вектором \mathbf{r}_i , а содержащаяся в нем завихренность — циркуляцией Γ_i , $i = 1, \dots, N$. При этом влияние вихревых элементов на скорость среды в произвольной точке \mathbf{r} вычисляется в соответствии с дискретным аналогом закона Био–Савара (3)

$$\mathbf{V}(\mathbf{r}, t) = \mathbf{V}_\infty + \sum_{i=1}^N \frac{\Gamma_i}{2\pi} \frac{\mathbf{k} \times (\mathbf{r} - \mathbf{r}_i)}{\max\{|\mathbf{r} - \mathbf{r}_i|^2, \varepsilon^2\}}, \tag{6}$$

где \mathbf{k} — единичный вектор, перпендикулярный плоскости течения.

Основной смысл вихревых методов состоит в переходе от решения уравнения Навье–Стокса (2), или, после преобразований, уравнения в частных производных (4) или (5), к решению системы обыкновенных дифференциальных уравнений, описывающих движение вихревых элементов. Действительно, рассмотренный выше “механический смысл” уравнения (5) позволяет считать, что циркуляции вихревых элементов остаются постоянными, а их перемещение происходит со скоростью $\mathbf{V} + \mathbf{W}$:

$$\frac{d\Gamma_i}{dt} = 0, \quad \frac{d\mathbf{r}_i}{dt} = \mathbf{V}(\mathbf{r}_i) + \mathbf{W}(\mathbf{r}_i), \quad i = 1, \dots, N. \tag{7}$$

Здесь $\mathbf{V}(\mathbf{r}_i)$ — скорость среды в точке расположения i -го вихревого элемента, вычисляемая по формуле (6), $\mathbf{W}(\mathbf{r}_i)$ — диффузионная скорость в этой точке.

Отметим, что возможен и альтернативный подход — считать положения вихревых элементов неизменными, а их циркуляции пересчитывать так, чтобы моделировать перенос завихренности. На такой интерпретации основан класс численных методов, называемых в литературе методами “вихрь в ячейке” (vortex-in-cell) [14].

Граничное условие на бесконечности в вихревых методах выполняется автоматически, при этом не требуется искусственно ограничивать расчетную область, как это делается в сеточных методах. Выполнение граничного условия прилипания на профиле (или условия непротекания, если моделируется течение невязкой среды) обеспечивается генерацией на нем вихревого слоя на каждом шаге расчета по времени, также моделируемого набором вихревых элементов. Этот вихревой слой является присоединенным, если обтекание невязкое, т.е. вихревые элементы, его моделирующие, остаются на своих местах и заменяются новыми на следующем шаге расчета, и является свободным при обтекании профиля вязкой жидкостью. Последнее означает, что вихревые элементы “сходят в поток” и становятся частью вихревого следа, образующегося вблизи обтекаемого профиля и позади него.

Для определения интенсивности вихревого слоя на профиле и последующего расчета циркуляций вихревых элементов можно применять различные методы и подходы, сводящиеся к решению системы линейных алгебраических уравнений, размерность которой соответствует густоте дискретизации профиля. Наибольшую эффективность, особенно применительно к методу ВВД, показал метод, основанный на обеспечении равенства нулю среднего значения касательной компоненты скорости на панелях — отрезках ломаной, аппроксимирующей профиль [15]; он во многих случаях превосходит по точности подход, обычно используемый в МДВ, на один–два порядка при одинаковой дискретизации профиля.

4. Программная реализация и оценка вычислительной сложности вихревых методов. Моделирование течений вихревыми методами включает в себя последовательное выполнение ряда операций

алгоритма на каждом шаге расчета (рис. 1). Обычно для установления режима обтекания профиля требуется выполнение нескольких тысяч и даже десятков тысяч временных шагов. Это количество определяется формой профиля, его ориентацией по отношению к набегающему потоку, тщательностью дискретизации профиля (более тщательная дискретизация требует более мелкого шага во времени) и рядом других параметров.

Особенности решаемых задач и конкретных применяемых алгоритмов могут влиять на реализацию операций из указанных блоков (в частности, некоторые из блоков могут отсутствовать). Будем считать, что на каждом шаге расчета на профиле генерируется n вихревых элементов, общее количество вихрей в области течения равно N ($n \ll N$). В практических расчетах n обычно составляет от нескольких сотен до нескольких тысяч, N может сильно варьироваться от задачи к задаче и составляет от нескольких десятков тысяч до нескольких сотен тысяч. Отметим, что вычисление скорости, генерируемой одним вихревым элементом в произвольной точке области течения, как следует из анализа формулы (6), требует выполнения 6 арифметических операций умножения/деления.

Далее оценим трудоемкости отдельных блоков алгоритма, указанных на рис. 1.

Блок генерации завихренности. В данном блоке производится решение системы линейных алгебраических уравнений и определяются циркуляции вихревых элементов, генерируемых на данном шаге расчета. Матрица системы вне зависимости от используемого метода определяется лишь формой обтекаемого профиля, поэтому в случае обтекания недеформируемого профиля она остается постоянной. Формирование матрицы системы и ее обращение производятся однократно перед началом счета, поэтому трудоемкость этих операций в расчет не принимается. На каждом шаге производится лишь вычисление правой части системы, основанное на определении скоростей среды в n точках профиля. Отыскание решения системы с матрицей порядка n сводится к умножению обратной матрицы на вектор правой части. Таким образом, суммарная трудоемкость блока генерации завихренности приближенно равна $(6N + n)n \approx 6Nn$.

Блок вычисления скоростей точек среды является наиболее трудоемким; здесь необходимо вычислить скорости всех вихревых элементов, находящихся в области течения. Данная задача аналогична задаче N тел, ее вычислительная сложность при непосредственном вычислении всех парных влияний вихревых элементов друг на друга составляет $6N^2$.

Блок вычисления диффузионных скоростей. Вычислительную сложность блока вычисления диффузионных скоростей оценить труднее, поскольку она зависит от особенностей реализации алгоритма и конкретных значений параметров расчетной схемы, однако поскольку диффузионные скорости также определяются парными влияниями вихревых элементов и влиянием на них обтекаемого профиля, вычислительная сложность алгоритма в первом приближении также пропорциональна N^2 . Результаты практических расчетов показывают, что этот блок является в 1.5–2.0 раза более трудоемким, чем предыдущий.

Блок вычисления нагрузок. Как упоминалось выше, в зависимости от решаемой задачи могут применяться различные алгоритмы вычисления гидродинамических нагрузок. В простейшем случае, когда вычисляются лишь интегральные нагрузки, зависящие только от распределения давления на профиле, сложность данного алгоритма крайне низка и пропорциональна n ; учет силы вязкого трения требует дополнительных затрат, при этом сложность становится пропорциональной n^2 . Расчет распределения давления и касательных напряжений на профиле будет иметь сложность, пропорциональную Nn .

Блок эволюции вихревого следа реализует смещение всех вихревых элементов со скоростями $\mathbf{V}_i + \mathbf{W}_i$, найденными ранее. Таким образом, его вычислительная трудоемкость низка и пропорциональна N .

Блок контроля непротекания необходим, поскольку в ходе расчета отдельные вихревые элементы могут проникать внутрь профиля. В данном блоке производится их программное отыскание и “извлечение” из профиля. Вычислительная сложность данной операции также в первом приближении пропорциональна n^2 .

Блок реструктуризации вихревого следа обеспечивает объединение близкорасположенных вихревых элементов в один, а также исключение из расчетной схемы вихревых элементов, удалившихся от обтекаемого профиля на значительное расстояние (обычно 10–20 характерных размеров профиля), поскольку они практически не оказывают влияния на параметры течения вблизи профиля. Вычислительная сложность алгоритма, как показывают расчеты, пропорциональна N^2 , при этом коэффициент пропорциональности



Рис. 1. Блок-схема программы

сравнительно мал (меньше значения 6, соответствующего второму блоку).

5. Параллельная реализация вихревых методов. Созданный к настоящему времени программный комплекс POLARA [16] позволяет производить расчеты обтекания профилей произвольной формы потоком вязкой несжимаемой среды. Разработанные параллельные реализации основных операций алгоритма позволяют существенно сократить время выполнения расчетов на многопроцессорных вычислительных комплексах с использованием библиотеки MPI.

Для повышения эффективности работы программного комплекса на системах с общей памятью была исследована возможность применения для этих же целей технологии OpenMP. С ее помощью были распараллелены те же три наиболее трудозатратные операции, что и при использовании MPI. При проведении расчетов на вычислительных машинах с общей памятью на 2–8 вычислительных ядрах было установлено, что выигрыш по времени, достигаемый за счет использования OpenMP, не превышает нескольких процентов по сравнению с применением MPI. В связи с этим данная возможность практически не использовалась, поскольку применение MPI позволяет сделать программу универсальной — она может работать как на системах с общей, так и с распределенной памятью (кластерах). В последнем случае эффективность распараллеливания, очевидно, снижается, но такая возможность необходима для решения сложных задач, когда в расчетной схеме присутствуют десятки-сотни тысяч вихревых элементов.

Примером такой вычислительно сложной задачи может служить тестовая задача Блазиуса о расчете обтекания тонкой пластинки вязким потоком. Данная задача (для случая бесконечно тонкой и бесконечно длинной пластинки) имеет точное аналитическое решение, с которым можно сравнить результаты расчета. Характерный вид вихревого следа в пограничном слое на пластинке показан на рис. 2 [17].

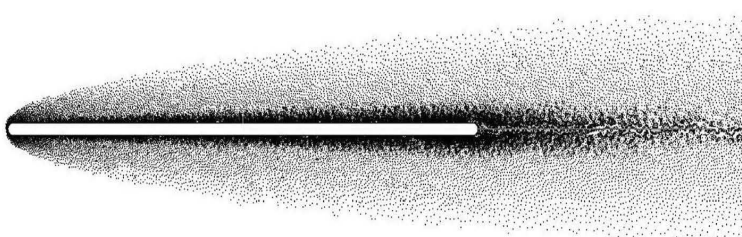


Рис. 2. Вид вихревого следа вблизи пластинки. Точки — положения вихревых элементов

В расчетах рассматривалась пластинка с относительной толщиной 2% и полукруглыми законцовками; число Рейнольдса, вычисленное по длине пластинки, составляло 10^3 , вихревой след в установившемся режиме состоял примерно из 120 000 вихревых элементов. Результаты расчета компонент вектора скорости в сечении пограничного слоя, отстоящем от переднего края пластинки на одну треть ее длины, в сравнении с точным аналитическим решением показаны на рис. 3. Отметим, что аналитическое решение является автомодельным, поэтому положение сечения может быть выбрано произвольным образом; необходимо лишь, чтобы оно отстояло на некоторое расстояние от задней кромки пластинки конечной длины, рассматриваемой в расчете.

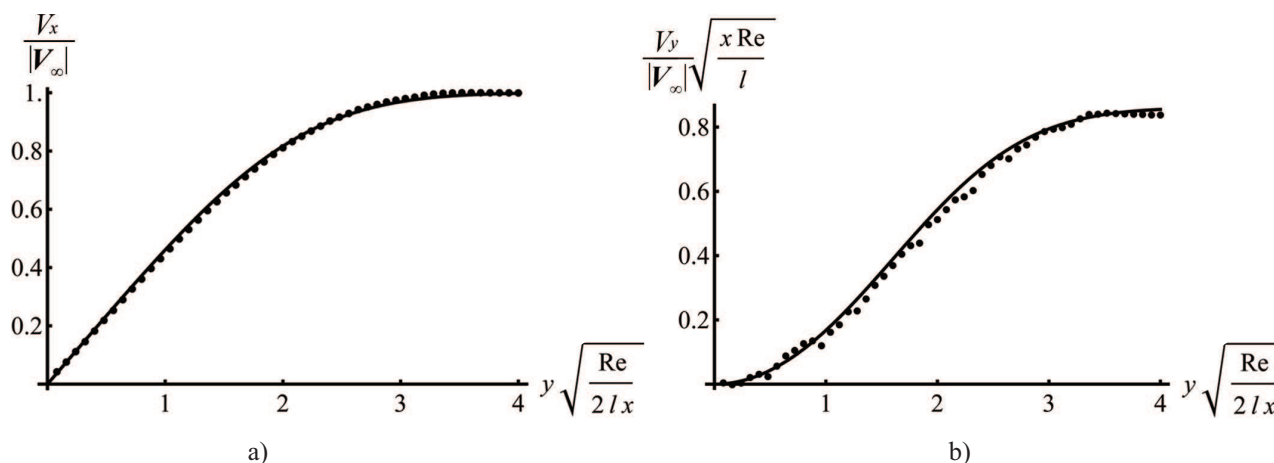


Рис. 3. Распределение продольной (а) и поперечной (б) компоненты скорости в пограничном слое в сечении $x = 0.33$. Сплошная линия — точное аналитическое решение

В качестве путей дальнейшего ускорения вычислений можно отметить два направления. Первое — это использование приближенных быстрых методов расчета вихревого влияния [16], для которых построена весьма точная оценка вычислительной сложности [18], позволяющая оптимальным образом подобрать

параметры алгоритма и снизить вычислительную сложность вычисления вихревого влияния с $O(N^2)$ до $O(N \log N)$. Оценки и результаты практических расчетов [19] показывают, что ускорение за счет использования быстрого метода достигает нескольких десятков раз при сохранении приемлемой точности.

Второй подход предполагает использование для проведения расчета возможностей графических ускорителей и технологии nVidia CUDA. Отметим, что данные подходы не являются взаимоисключающими; применительно к гравитационным задачам N тел известны эффективные реализации приближенных быстрых методов на GPU [20]. В настоящей работе ограничимся переносом на GPU алгоритма, в котором парные влияния вихревых элементов рассчитываются непосредственно.

6. Особенности использования технологии CUDA. Важной особенностью вихревых методов, позволяющей эффективно использовать возможности графических ускорителей (GPU), являются малые затраты оперативной памяти при проведении расчета, поскольку в данном случае не требуется строить сетку и хранить значения неизвестных во всех ее узлах. Оценим необходимые затраты оперативной памяти для работы алгоритма на рассмотренном выше примере решения задачи Блазиуса.

Для хранения обратной матрицы для расчета циркуляций генерируемых на профиле вихрей требуется около 50 Мб памяти (при генерации 2600 ВЭ на профиле на каждом шаге). В разработанном алгоритме для каждого вихревого элемента требуется хранить его циркуляцию, координаты, вектор скорости, а также характерное расстояние от него до ближайших к нему вихрей, следовательно, требуется $48N$ байт оперативной памяти, где N — количество ВЭ в области течения. В тестовых расчетах N не превышало 330 тридцати тысяч, следовательно, затраты памяти на их хранение составляют около 15 Мб. При выполнении каждого шага в памяти хранятся как “старые”, так и “новые” параметры вихрей, поэтому требуется дополнительно 15 Мб памяти. Если для интегрирования системы (7), описывающей движение вихрей, будет использоваться многостадийный метод Рунге–Кутты, то это потребует дополнительных 15 Мб оперативной памяти в расчете на каждую промежуточную стадию. В данном примере использовался явный метод Эйлера.

Таблица 1

Затраты оперативной памяти

| Назначение | Оценка | Экспериментальные данные |
|-------------------------------|------------------|--------------------------|
| Матрица системы | 51 Мб | 52 Мб |
| Параметры ВЭ | 2×15 Мб | 2×16 Мб |
| Системные затраты при запуске | 30 Мб | 30 Мб |
| Пиковое потребление | 111 Мб | 114 Мб |

Некоторое количество оперативной памяти требуется также на хранение параметров обтекаемого профиля (не более нескольких сотен килобайт), а также “служебной информации”. Порядка 30 Мб в памяти GPU резервируется системой при запуске любой программы, использующей технологию CUDA. Таким образом, для успешной работы программы в данном примере достаточно менее 120 Мб свободной оперативной памяти. Отметим, что приведенные оценки являются по сути оценками сверху: значения $n = 2600$ и $N = 330\,000$ являются избыточными для большинства практических задач, поэтому реальные затраты оперативной памяти на практике скорее всего окажутся существенно ниже. В табл. 1 показано сравнение полученных оценок потребления оперативной памяти с ее реальным расходом при работе программы, полученном с помощью утилиты MSI Afterburner.

Таким образом, оценка потребности в оперативной памяти $\text{Mem} = 8n^2 + 2 \times 48N$ (байт) является весьма точной.

Столь низкие затраты оперативной памяти упрощают пересылку данных между оперативной памятью компьютера и оперативной памятью графического ускорителя, однако время на их выполнение все равно оказывается существенным, поэтому была разработана реализация алгоритма вихревого метода, в которой все вычисления полностью перенесены на GPU, а пересылки данных между GPU и компьютером происходят лишь при сохранении промежуточных результатов.

На практике сохранение текущего состояния вихревого следа в файл производится обычно раз в несколько десятков или даже сотен шагов; значения гидродинамических нагрузок, действующих на тело, вычисляются на каждом шаге, однако они представляют собой лишь три действительных числа, пересылка которых не вызывает проблем. Различная статистика расчета и прочая “телеметрическая” информация в силу ее небольшого объема (обычно суммарно несколько десятков-сотен килобайт, реже — мегабайты)

может сохраняться в оперативной памяти GPU и пересылаться в память компьютера для дальнейшего сохранения в файл либо один раз в конце расчета, либо регулярно, но редко, скажем, 1 раз за 1000 шагов.

Анализ используемых алгоритмов показал, что для их эффективного исполнения на GPU требуется не слишком значительная их переработка; в частности, практически все операции были реализованы без использования операторов условного перехода `if...else`. Кроме того, в целях оптимизации программы все выполняемые операции, ранее разделенные на 7 блоков (рис. 1), были сведены в три крупных блока, названных по наиболее трудоемкой операции: блок генерации, блок скоростей и блок реструктуризации.

Определение скоростей, т.е. расчеты парных влияний вихревых элементов (ВЭ) друг на друга, могут выполняться независимо, что делает очевидным подход к распараллеливанию данной операции. Каждое CUDA-ядро вычисляет скорость одного ВЭ, используя разделяемую память, в которую поочередно помещаются блоки из N_b ВЭ. При этом общее количество ВЭ увеличивается до кратного N_b путем добавления “фиктивных” ВЭ с нулевой циркуляцией. Аналогично производится расчет диффузионных скоростей.

При *генерации* ВЭ на профиле каждое CUDA-ядро вычисляет скорость среды в точке рождения аналогично вычислению парных влияний ВЭ.

При *реструктуризации* вихревого следа (включающей в себя перемещение ВЭ, контроль непротекания и собственно реструктуризацию — объединение близкорасположенных ВЭ в один) для каждого вихревого элемента необходимо проверять несколько условий и в случае их выполнения проводить над данным ВЭ некоторые операции. Использование оператора ветвления в программах, использующих GPGPU, значительно снижает скорость выполнения расчета, поэтому для минимизации негативных последствий необходимо, во-первых, категорически избегать вложенных ветвлений, а во-вторых, проектировать алгоритм так, чтобы вероятность выполнения одной из веток была значительно выше другой и вычислительная сложность менее вероятного варианта была многократно меньшей по сравнению с основным. Кроме того, ветвления алгоритма не критичны в случае крайне низкой сложности каждого из путей.

В связи с указанными особенностями *контроль непротекания* реализован в два этапа: на первом этапе для каждого ВЭ проверяется условие попадания в габаритный прямоугольник профиля (прямоугольник со сторонами, параллельными осям координат, в который вписан обтекаемый профиль); попавшие в него запоминаются в отдельном массиве — таким образом, вероятность непопадания в прямоугольник выше вероятности попадания, а вычислительная сложность обеих веток незначительна, что соответствует приведенным критериям эффективного применения операторов ветвления в CUDA-программах. Второй этап выполняется лишь для ВЭ, оказавшихся внутри габаритного прямоугольника; для каждого из них проверяется пересечение траектории его движения на последнем шаге с панелями профиля — участками ломаной, аппроксимирующей профиль.

Операция *объединения близкорасположенных вихревых элементов* из тех же соображений разделена на два этапа. На первом этапе для каждого ВЭ происходит поиск первого подходящего для объединения вихревого элемента, а номера ВЭ, образующих эту пару, запоминаются в отдельном массиве. Таким образом, для каждого ядра при выполнении этого этапа возможны два состояния — либо продолжается поиск пары, либо вычисления завершены, причем первый вариант более вероятен. Это приводит лишь к небольшому замедлению всего алгоритма. Второй этап объединения предполагает внесение изменений в параметры ВЭ, поэтому для соблюдения условия согласованности данных он проводится в последовательном режиме (но выполняется на GPU) и состоит в объединении пар ВЭ из служебного массива.

На заключительной стадии блока реструктуризации из расчета исключаются ВЭ, удалившиеся на значительное расстояние от профиля, и происходит перенумерация массива ВЭ для вытеснения из него удаленных ранее вихрей. Этот этап также проводится в последовательном режиме и имеет сравнительно малую трудоемкость.

Операция *вычисления гидродинамических нагрузок* при проведении расчета на GPU получается пренебрежимо малозатратной, поскольку все основные вычисления (при вычислении интегральных нагрузок на профиль и силы вязкого трения) производятся одновременно с вычислением парных влияний ВЭ и их диффузионных скоростей.

Таким образом, операция вычисления скоростей ВЭ исключительно эффективно распараллеливается на GPU, эффективность реализации остальных операций несколько ниже; незначительная часть кода, имеющая минимальную вычислительную сложность, выполняется на GPU в последовательном режиме.

В настоящей работе расчеты на GPU проводились с использованием графических ускорителей Tesla C2050 и GeForce GTX 970, имеющих 448 и 1644 ядра соответственно и по 14 мультипроцессоров.

Опыт проведения расчетов показывает, что в вихревых методах нет необходимости выполнения всех вычислений с двойной точностью. Как показывает практика, двойная точность обязательна лишь в отдельных операциях внутри блока генерации вихревых элементов на профиле и в блоке реструктуриза-

ции. Это позволяет существенным образом сократить затраты на проведение расчетов на GPU, поскольку производительность графических ускорителей при проведении расчетов с одинарной точностью выше (во многих случаях — в несколько раз), чем при использовании двойной точности.

Ниже в табл. 2 и 3 приведены затраты времени на проведение 3 000 шагов одного и того же расчета с одинарной и двойной точностью; на обтекаемом профиле на каждом шаге рождалось $n = 2576$ вихревых элементов, их количество в следе нарастало от 0 до примерно 50 000 и в среднем составляло около 43 000.

Таблица 2
Время расчета на GPU Tesla C2050 / GeForce GTX 970, одинарная точность

| | Время счета (в секундах) при различных размерах блока | | | |
|-----------------------|---|-----------|-----------|-----------|
| | 32 | 64 | 128 | 256 |
| Блок генерации (Б.1) | 525/231 | 538/232 | 535/228 | 535/229 |
| Блок скоростей (Б.2) | 3459/1750 | 2242/1809 | 2203/1787 | 2220/1769 |
| Блок реструкт-и (Б.3) | 4026/1871 | 2573/1610 | 2365/1610 | 2455/1659 |
| Общее время расчета | 8010/3852 | 5353/3651 | 5103/3625 | 5210/3657 |

Таблица 3
Время расчета на GPU Tesla C2050 / GeForce GTX 970, двойная точность

| | Время счета (в секундах) при различных размерах блока | | | |
|-----------------------|---|-----------|-----------|-----------|
| | 32 | 64 | 128 | 256 |
| Блок генерации (Б.1) | 671/944 | 678/997 | 719/995 | 730/1005 |
| Блок скоростей (Б.2) | 5319/6744 | 3694/6857 | 3612/6878 | 3772/6873 |
| Блок реструкт-и (Б.3) | 3863/1868 | 2512/1636 | 2337/1669 | 2428/1725 |
| Общее время расчета | 9853/9556 | 6884/9490 | 6668/9542 | 6930/9603 |

Анализ данных, приведенных в табл. 2 и 3, показывает также, что в графических ускорителях GeForce последнего поколения обработка ветвлений реализована более эффективно (блок Б.3), однако они уступают сравнительно старым ускорителям Tesla при выполнении с двойной точностью операций блоков Б.1 и Б.2, где ветвления практически отсутствуют. При этом время выполнения блока Б.3 оказывается практически неизменным для одинарной и двойной точности.

При использовании GPU Tesla C2050 общее время выполнения расчетов с одинарной и двойной точностью различается на 25–30%, в зависимости от размера блока. Проведенные вычислительные эксперименты показали, что оптимальным является размер блока 128, его изменение до 64 или 128 приводит к снижению производительности на 3–5%. Слишком малый размер блока (например, 32) приводит к 1.5-кратному увеличению времени счета.

Проведение расчетов на GPU GeForce GTX 970 показывает существенно меньшую зависимость времени вычислений от размера блока, что можно объяснить особенностью архитектуры последнего поколения GPU: общая DRAM-память является кэшируемой. Однако в данном случае проведение расчетов с одинарной точностью (кроме упоминавшихся отдельных операций блоков Б.1 и Б.3) позволяет сократить время выполнения блоков Б.1 и Б.2, в которых практически отсутствуют ветвления, более чем в 3.8 раза. С учетом того, что время выполнения блока Б.3 при этом остается практически неизменным, общее время расчета с одинарной точностью оказывается в 2.6 раза меньше, чем время расчета с двойной точностью. Эту возможность исключительно целесообразно использовать при моделировании течений вихревыми методами.

В результате решения описанной выше задачи Блазиуса по моделированию обтекания пластинки получены следующие соотношения трудоемкостей основных операций (рис. 4), отражающие сделанные выше оценки.

Для оценки общей эффективности применения графических ускорителей при проведении расчетов вихревыми методами рассмотренная задача о моделировании обтекания пластинки была решена без использования GPU в последовательном режиме (1 ядро), а также в параллельном режиме с использованием

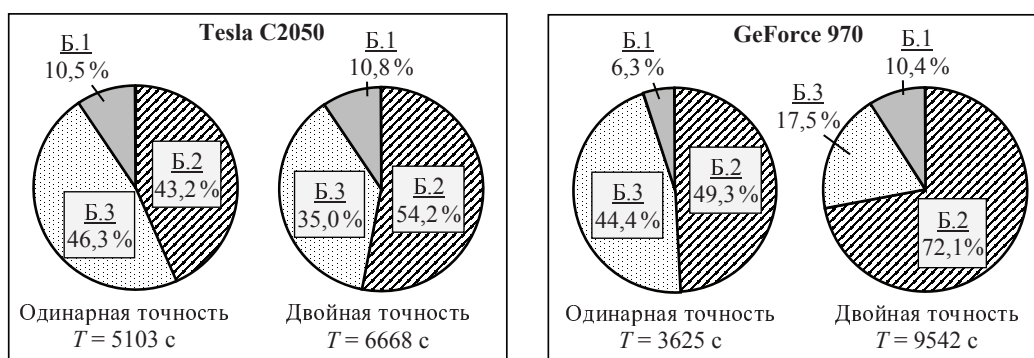


Рис. 4. Соотношение трудоемкостей укрупненных блоков при проведении расчета на GPU Tesla C2050 (слева) и GeForce GTX 970 (справа) с одинарной и двойной точностью

технологии MPI на 4-ядерном процессоре и на 16-ядерном кластере (характеристики CPU и кластера приведены в [7]). Затраты времени на выполнение одного шага расчета при $N \approx 50\,000$ приведены в табл. 4.

Таблица 4
Время выполнения одного шага расчета при $N \approx 50\,000$

| CPU | | Кластер, 4 узла | Tesla C2050 | | GeForce 970 | |
|---------|--------|-----------------|-------------|--------|-------------|--------|
| 1 ядро | 4 ядра | CPU, 16 ядер | float | double | float | double |
| 211.2 с | 57.1 с | 17.6 с | 2.18 с | 2.84 с | 1.57 с | 4.10 с |

Как видно из представленных результатов, использование возможностей графического ускорителя Tesla C2050 позволяет в режиме одинарной точности сократить время выполнения шага расчета более чем в 8 раз по сравнению с расчетом на 16-ядерном кластере и почти в 100 раз по сравнению с последовательным расчетом на CPU. При проведении расчета на графическом ускорителе GeForce GTX 970 тоже в режиме одинарной точности ускорение составляет примерно 11 и 135 раз соответственно, т.е. время выполнения шага снижается дополнительно на треть, несмотря на существенно большее количество CUDA-ядер по сравнению с Tesla C2050. Данный эффект объясняется тем, что ускорители Tesla оптимизированы для выполнения расчетов, а видеокарты GeForce относятся к “бытовым”. Их производительность, как следует из представленных результатов, при переходе к двойной точности существенно (более чем в 2.6 раза) снижается и заметно уступает картам Tesla.

Представляет также интерес сравнение скорости решения задачи на GPU с использованием разработанного алгоритма и скорости решения той же задачи на CPU, когда расчет ведется с использованием быстрого метода [6] при его последовательной и параллельной реализациях [16, 19]. Параметр точности быстрого метода был выбран равным $\theta = 0.2$; для выбора оптимальной глубины дерева использовалась оценка трудоемкости соответствующего алгоритма, полученная в [18]. Затраты времени на выполнение одного шага расчета при $N \approx 50\,000$ приведены в табл. 5.

Таблица 5
Время выполнения одного шага расчета при $N \approx 50\,000$
(на CPU — быстрый метод)

| CPU | | Кластер, 4 узла | Tesla C2050 | | GeForce 970 | |
|---------|--------|-----------------|-------------|--------|-------------|--------|
| 1 ядро | 4 ядра | CPU, 16 ядер | float | double | float | double |
| 19.69 с | 7.92 с | 3.44 с | 2.18 с | 2.84 с | 1.57 с | 4.10 с |

Из табл. 5 видно, что использование GPU позволяет получить более чем 10-кратное ускорение по сравнению с последовательным расчетом на CPU быстрым методом (12.5 раз для GeForce 970 с одинарной точностью) и более чем 2-кратное ускорение по сравнению с MPI-реализацией быстрого метода при использовании 16 вычислительных ядер. Лишь расчет с двойной точностью на карте GeForce 970 оказывается на 20% медленнее, чем параллельная реализация быстрого метода на 16-ядерном кластере.

7. Выводы. Рассмотрен подход к ускорению вычислений в вихревых методах при моделировании обтекания профиля несжимаемой средой за счет применения технологии nVidia CUDA. Разработан алгоритм, позволяющий проводить все вычисления на графическом ускорителе. Получены и подтверждены оценки затрат оперативной памяти и трудоемкости отдельных операций алгоритма. Достигнутые значения ускорения позволяют существенно расширить применение вихревых методов и повысить точность решения задач путем повышения количества вихревых элементов при сохранении сравнительно малого времени расчета.

Проведено решение тестовой задачи Блазиуса о моделировании обтекания тонкой пластинки; результаты расчетов хорошо согласуются с точным аналитическим решением и известными результатами вычислительных экспериментов.

Работа выполнена при финансовой поддержке гранта Президента РФ по государственной поддержке молодых российских ученых — кандидатов наук (проект МК–3705.2014.8).

Статья рекомендована к публикации Программным комитетом Международной научной конференции “Параллельные вычислительные технологии” (ПаВТ-2015; <http://agora.guru.ru/pavt2015>).

СПИСОК ЛИТЕРАТУРЫ

1. *Belotserkovsky S.M., Lifanov I.K.* Method of discrete vortices. Boca Raton: CRC Press, 1994.
2. *Лифанов И.К.* Метод сингулярных уравнений и численный эксперимент (в математической физике, аэродинамике, теории упругости и деформации волн). М.: Янус, 1995.
3. *Дынкикова Г.Я.* Вихревые методы исследования нестационарных течений вязкой несжимаемой жидкости: Дис. ... докт. физ.-мат. наук. М., 2011.
4. *Апарин В.А., Дворак А.В.* Метод дискретных вихрей с замкнутыми вихревыми рамками // Труды ВВИА им. проф. Н.Е. Жуковского. 1986. Вып. 1313. 424–432.
5. *Марчевский И.К., Щеглов Г.А.* Модель симметричного вортон-отрезка для численного моделирования пространственных течений идеальной несжимаемой среды // Вестник МГТУ им. Н.Э. Баумана. Серия: Естественные науки. 2008. № 4. 62–71.
6. *Дынкикова Г.Я.* Использование быстрого метода решения “задачи N тел” при вихревом моделировании течений // Журнал вычислительной математики и математической физики. 2009. 49, № 8. 1458–1456.
7. *Лужин В.В., Марчевский И.К., Морева В.С., Попов А.Ю., Шатовалов К.Л., Щеглов Г.А.* Учебно-экспериментальный вычислительный кластер. Ч. 2. Примеры решения задач // Вестник МГТУ им. Н.Э. Баумана. Естественные науки. 2012. № 4. 82–102.
8. *Марчевский И.К., Щеглов Г.А.* Применение параллельных алгоритмов при решении задач гидродинамики методом вихревых элементов // Вычислительные методы и программирование. 2010. 11. 105–110.
9. *Дынкикова Г.Я., Сыроватский Д.А.* Трехмерное бессеточное моделирование нестационарных течений жидкости на суперкомпьютерах гибридной архитектуры с графическими ускорителями // Ломоносовские чтения. Тезисы. 2012. 71–72.
10. *Дынкикова Г.Я., Сыроватский Д.А.* Численное моделирование трехмерного нестационарного обтекания идеальной жидкостью тонких несущих поверхностей бессеточным методом дипольных доменов // Современные проблемы аэрогидродинамики: Тезисы докладов XVII школы-семинара. М.: Изд-во Моск. ун-та, 2014. 55.
11. *van Rees W.M., Rossinelli D., Hadjidakas P., Koumoutsakos P.* High performance CPU/GPU multiresolution Poisson solver // Parallel Computing: Accelerating Computational Science and Engineering (CSE). Advances in Parallel Computing. Vol. 25. Amsterdam: IOS Press, 2014. 481–490.
12. *Андронов П.Р., Гувернюк С.В., Дынкикова Г.Я.* Вихревые методы расчета нестационарных гидродинамических нагрузок. М.: Изд-во Моск. ун-та, 2006.
13. *Li S., Liu W.K.* Meshfree particle methods. Berlin: Springer, 2007.
14. *Christiansen I.P.* Numerical simulation of Hydrodynamics by the Method of Point Vortices // J. Comp. Phys. 1973. 13, N 3. 363–379.
15. *Moreva V.S., Marchevsky I.K.* Vortex element method for 2D flow simulation with tangent velocity components on airfoil surface // Proc. 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012). Vienna: Vienna Univ. Technol., 2012. 5952–5965.
16. *Марчевский И.К., Морева В.С.* Параллельный программный комплекс POLARA для моделирования обтекания профилей и исследования расчетных схем метода вихревых элементов // Параллельные вычислительные технологии (ПаВТ-2012): труды Международной научной конференции. Новосибирск: Ин-т вычисл. матем. и вычисл. геофизики, 2012. 236–247.
17. *Макарова М.Е., Марчевский И.К., Морева В.С.* Моделирование обтекания тонкой пластинки с использованием модифицированной схемы метода вихревых элементов // Наука и образование. Т. 9. М.: МГТУ им. Н.Э. Баумана, 2013. 233–242.
18. *Кузьмина К.С., Марчевский И.К.* Оценка трудоемкости быстрого метода расчета вихревого влияния в методе вихревых элементов // Наука и образование. Т. 10. М.: МГТУ им. Н.Э. Баумана, 2013. 399–414.

19. Кузьмина К.С., Марчевский И.К. Об ускорении вычислений при решении двумерных сопряженных задач гидроупругости вихревыми методами // Вестник Пермского Национального исследовательского политехнического университета. Аэрокосмическая техника. 2014. № 39. 145–163.
20. Yokota R., Barba L.A. Hierarchical N -body simulations with autotuning for heterogeneous systems // Computing in Science and Engineering. 2012. 14, N 3. 30–39.

Поступила в редакцию
10.02.2015

An Implementation of Vortex Methods for Modeling 2D Incompressible Flows Using the CUDA Technology

S. R. Grechkin-Pogrebnyakov¹, K. S. Kuzmina², and I. K. Marchevsky³

¹ Bauman Moscow State Technical University, Faculty of Fundamental Sciences; ulitsa 2-ya Baumanskaya 5, Moscow, 105005, Russia; Student, e-mail: grechkin.pogrebnyakov@gmail.com

² Bauman Moscow State Technical University, Faculty of Fundamental Sciences; ulitsa 2-ya Baumanskaya 5, Moscow, 105005, Russia; Student, e-mail: kuz-ksen-serg@yandex.ru

³ Bauman Moscow State Technical University, Faculty of Fundamental Sciences; ulitsa 2-ya Baumanskaya 5, Moscow, 105005, Russia; Ph.D., Associate Professor, e-mail: iliamarchevsky@mail.ru

Received February 10, 2015

Abstract: The possibility of computation speedup in the vortex element method (a meshfree Lagrangian method of computational fluid dynamics) using the graphics accelerators is studied. An algorithm based on the authors' modification of the vortex element method is implemented; this algorithm allows one to perform all the necessary computations directly on a GPU using the CUDA technology. The speed of solving a typical problem on a single GeForce GTX 970 or Tesla C2050 accelerator is comparable with the speed of solving a similar problem on a cluster containing 30–40 cores with the use of the MPI technology. The numerical results obtained confirm a high efficiency of using graphics accelerators when solving the problems of hydrodynamics with vortex methods.

Keywords: graphics processor unit (GPU), CUDA technology, parallel computing, Navier–Stokes equations, viscous incompressible flow, method of viscous vortex domains, discrete vortex method, Blasius problem.

References

1. S. M. Belotserkovsky and I. K. Lifanov, *Method of Discrete Vortices* (CRC Press, Boca Raton, 1994).
2. I. K. Lifanov, *Singular Integral Equations and Discrete Vortices* (Yanus, Moscow, 1995; VSP, Utrecht, 1996).
3. G. Ya. Dynnikova, *Vortex Methods to Study Unsteady Flows of Viscous Incompressible Fluids*, Doctoral Dissertation in Mathematics and Physics (Zhukovsky Air Force Engineering Acad., Moscow, 2011).
4. V. A. Aparinov and A. V. Dvorak, "Method of Discrete Vortices with Closed Vortex Frames," Tr. VVIA im. Prof. Zhukovskogo, No. 1313, 424–432 (1986).
5. I. K. Marchevskii and G. A. Shcheglov, "Model of Symmetrical Vortex-Segment for Numerical Modeling of 3D Flows of Ideal Incompressible Medium," Vestn. Bauman Mosk. Tekh. Univ., Ser.: Natural Sci., No. 4, 62–71 (2008).
6. G. Ya. Dynnikova, "Fast Technique for Solving the N -Body Problem in Flow Simulation by Vortex Methods," Zh. Vychisl. Mat. Mat. Fiz. **49** (8), 1458–1465 (2009) [Comput. Math. Math. Phys. **49** (8), 1389–1396 (2009)].
7. V. V. Lukin, I. K. Marchevskii, V. S. Moreva, et al., "Computing Cluster for Training and Experiments. Part 2. Examples of Solving Problems," Vestn. Bauman Mosk. Tekh. Univ., Ser.: Natural Sci., No. 4, 82–102 (2012).
8. I. K. Marchevsky and G. A. Scheglov, "Application of Parallel Algorithms for Solving Hydrodynamic Problems by the Vortex Element Method," Vychisl. Metody Programm. **11** (1), 105–110 (2010).

9. G. Ya. Dynnikova and D. A. Syrovatskii, "Three-Dimensional Meshfree Simulation of Unsteady Flows on Hybrid Supercomputing Systems with Graphics Accelerators," in *Proc. Lomonosov Readings, Moscow, April 22–24, 2012* (Mosk. Gos. Univ., Moscow, 2012), pp. 71–72.
10. G. Ya. Dynnikova and D. A. Syrovatskii, "Numerical Simulation of Unsteady Flow by an Ideal Fluid around Thin Lifting Surfaces Using the Meshfree Methods of Dipole Domains," in *Proc. 17th Conference on Modern Problems of Aerohydrodynamics, Sochi, Russia, August 20–30, 2014* (Mosk. Gos. Univ., Moscow, 2014), p. 55.
11. W. M. Van Rees, D. Rossinelli, P. Hadjidoukas, and P. Koumoutsakos, "High Performance CPU/GPU Multiresolution Poisson Solver," in *Parallel Computing: Accelerating Computational Science and Engineering (CSE). Advances in Parallel Computing* (IOS Press, Amsterdam, 2014), Vol. 25, pp. 481–490.
12. P. R. Andronov, S. V. Guvernyuk, and G. Ya. Dynnikova, *Vortex Methods of Calculation of Unsteady Hydrodynamic Loads* (Mosk. Gos. Univ., Moscow, 2006) [in Russian].
13. S. Li and W. K. Liu, *Meshfree Particle Methods* (Springer, Berlin, 2004).
14. I. P. Christiansen, "Numerical Simulation of Hydrodynamics by the Method of Point Vortices," *J. Comp. Phys.* **13** (3), 363–379 (1973).
15. V. S. Moreva and I. K. Marchevsky, "Vortex Element Method for 2D Flow Simulation with Tangent Velocity Components on Airfoil Surface," in *Proc. 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), Vienna, Austria, September 10–14, 2012* (Vienna Univ. Technol., Vienna, 2012), pp. 5952–5965.
16. I. K. Marchevsky and V. S. Moreva, "The POLARA Package to Model the Profile Flow and to Study the Computational Scheme of the Vortex Element Method," in *Proc. Int. Conf. on Parallel Computational Technologies, Novosibirsk, Russia, March 26–30, 2012* (Inst. Comput. Math. Math. Geophys., Novosibirsk, 2012), pp. 236–247.
17. M. E. Makarova, I. K. Marchevsky, and V. S. Moreva, "Flow Simulation around a Thin Plate Using a Modified Numerical Scheme of the Vortex Element Method," in *Science and Education* (Bauman Moscow Tech. Univ., Moscow, 2013), Vol. 9, pp. 233–242.
18. K. S. Kuzmina and I. K. Marchevsky, "Estimation of Computational Complexity of the Fast Numerical Algorithm for Calculating Vortex Influence in the Vortex Element Method," in *Science and Education* (Bauman Moscow Tech. Univ., Moscow, 2013), Vol. 10, pp. 399–414.
19. K. S. Kuzmina and I. K. Marchevsky, "On Computation Speeding up when Solving Two-Dimensional Hydroelastic Coupled Problems by Using Vortex Methods," *Vestn. Perm Politekh. Univ., Ser.: Aerokosm. Tekh.*, No. 39, 145–163 (2014).
20. R. Yokota and L. A. Barba, "Hierarchical N -Body Simulations with Autotuning for Heterogeneous Systems," *Comput. Sci. Eng.* **14** (3), 30–39 (2012).