

## ОРИГИНАЛЬНАЯ СТАТЬЯ

УДК 519.6: 532.5

<https://doi.org/10.26907/2541-7746.2025.1.99-114>Реализация быстрых алгоритмов при моделировании  
двумерных течений вихревыми методамиЕ.П. Рятина, И.К. Марчевский , А.О. Колганова, Д.Ю. Кобзарь*Московский государственный технический университет им. Н. Э. Баумана, г. Москва, Россия* *iliamarchevsky@mail.ru*

## Аннотация

Вихревые методы вычислительной гидродинамики широко используются в инженерной практике для решения задач моделирования течений и оценки нестационарных гидродинамических нагрузок, действующих на тела в потоке. Основным преимуществом вихревых методов является возможность выполнения расчета при относительно небольших затратах вычислительных ресурсов, при этом область их применения весьма узка: моделирование дозвуковых несжимаемых однофазных нетеплопроводных течений. Однако при повышении степени дискретизации вычислительная сложность основных операций существенно возрастает, как и затраты машинной памяти, если использовать «прямые» алгоритмы. Для основных трудоемких операций, таких как вычисление конвективных скоростей вихревых частиц и решение граничного интегрального уравнения, разработаны и реализованы приближенные быстрые алгоритмы, обладающие квазилинейной сложностью. Представлено описание быстрых алгоритмов, обсуждаются их модификации для решения указанных задач, исследована эффективность их реализаций. Показано, что использование быстрых алгоритмов позволяет достичь ускорения расчетов в сотни раз при количестве вихревых частиц порядка миллиона.

**Ключевые слова:** вихревой метод, быстрый алгоритм, метод Барнса – Хата, быстрый метод мультиполей, задача  $N$  тел, граничное интегральное уравнение

---

**Для цитирования:** Рятина Е.П., Марчевский И.К., Колганова А.О., Кобзарь Д.Ю. Реализация быстрых алгоритмов при моделировании двумерных течений вихревыми методами // Учен. зап. Казан. ун-та. Сер. Физ.-матем. науки. 2025. Т. 167, кн. 1. С. 99–114.  
<https://doi.org/10.26907/2541-7746.2025.1.99-114>.

---

## ORIGINAL ARTICLE

<https://doi.org/10.26907/2541-7746.2025.1.99-114>

## Implementation of fast algorithms for 2D flow simulation using vortex particle methods

E.P. Ryatina, I.K. Marchevsky , A.O. Kolganova, D.Yu. Kobzar

*Bauman Moscow State Technical University, Moscow, Russia*

 [iliamarchevsky@mail.ru](mailto:iliamarchevsky@mail.ru)

### Abstract

Vortex particle methods of computational hydrodynamics are widely employed by engineers to solve the problems of flow simulation and estimation of unsteady hydrodynamic loads acting on bodies. The main advantage of such methods is a relatively low computational cost, but their applicability is limited to subsonic incompressible single-phase non-heat-conducting flows. If high-order discretization is required, the usage of direct algorithms leads to a significant increase in computational complexity and memory demand. To overcome this limitation, approximate fast algorithms of quasilinear computational complexity were developed and implemented for the most time-consuming operations, such as the computation of convective velocities and the solution of the boundary integral equation. The general principles of fast algorithms were described. Their modifications for the problems mentioned above were discussed, and their efficiency was evaluated. The results obtained show that the application of fast algorithms enables a computational speedup of up to several hundred times for around a million vortex particles.

**Keywords:** vortex method, fast algorithm, Barnes–Hut method, fast multipole method,  $N$ -body problem, boundary integral equation

---

**For citation:** Ryatina E.P., Marchevsky I.K., Kolganova A.O., Kobzar D.Yu. Implementation of fast algorithms for 2D flow simulation using vortex particle methods. *Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki*, 2025, vol. 167, no. 1, pp. 99–114. <https://doi.org/10.26907/2541-7746.2025.1.99-114>. (In Russian)

---

### Введение

В настоящее время для решения задач вычислительной гидродинамики, таких как моделирование обтекания двумерных профилей и трехмерных тел, оценка стационарных и нестационарных гидродинамических нагрузок, действующих на подвижные или неподвижные конструкции, и т. д., преимущественно применяются сеточные методы. Однако при рассмотрении низкоскоростных, существенно дозвуковых течений однофазных сред, когда влиянием сжимаемости можно пренебречь, эффективной альтернативой сеточным (с точки зрения времени вычислений) могут являться лагранжевы вихревые методы [1–6], особенно при моделировании внешнего обтекания профилей или тел, которые являются подвижными и/или деформируемыми. При этом движение обтекаемых поверхностей может

происходить как по заранее заданному закону, так и под действием гидродинамических сил; в последнем случае речь идет о рассмотрении сопряженных задач гидроупругости.

Суть вихревых методов заключается в переходе от классической формулировки уравнений Навье – Стокса в переменных «скорость – давление» к завихренности как первичной расчетной величине. Распределение завихренности в области течения задают в виде множества  $N$  вихревых частиц, которые в двумерном случае представляют собой бесконечные вихревые нити (или вихревые домены), ортогональные плоскости течения. Их интенсивности (циркуляции), в зависимости от рассматриваемой модификации вихревых методов, могут быть как постоянными, так и изменяющимися во времени. В рамках обсуждаемого здесь метода вязких вихревых доменов [3, 7, 8] использован первый подход.

Обтекаемая поверхность в потоке в соответствии с идеями, восходящими еще к Н. Е. Жуковскому и даже Г. Гельмгольцу, может быть заменена тонким вихревым слоем с заранее неизвестной интенсивностью, удовлетворяющей некоторому интегральному уравнению, которое выражает граничное условие на обтекаемом профиле: условие прилипания, если рассматривается вязкая среда, или условие непротекания, если моделируется течение идеальной среды.

При решении задач моделирования поперечного обтекания элементов конструкций, имеющих значительное удлинение, вместо решения пространственной задачи можно рассмотреть одну или набор значительно более простых плоских задач о расчете обтекания профилей отдельных сечений исходного тела. В настоящей работе будем рассматривать именно алгоритмы решения двумерных задач.

Если моделируется течение вязкой жидкости, описываемое уравнениями Навье – Стокса, завихренность, содержащаяся в вихревом слое на профиле, следует рассматривать как свободную. Это означает, что будучи сгенерированной вблизи профиля, далее она становится «частью» вихревого следа, т. е. распределения завихренности в области течения [3]. Эволюция вихревого следа моделируется движением большого числа вихревых частиц согласно законам гидродинамики вязких сред. При этом диффузия отдельных вихревых частиц, по крайней мере в методе вязких вихревых доменов, не моделируется. Воспроизведение диффузии макроскопических («физических») вихрей обеспечивается путем их моделирования большим числом вихревых частиц, которые, в свою очередь, движутся по специальным образом построенному полю скоростей – это т. н. метод диффузионной скорости [3, 7–9].

При моделировании обтекания подвижных профилей в дополнение к свободному вихревому слою вводят присоединенный вихревой слой и присоединенный слой источников, интенсивности которых определяются только лишь скоростями движения точек обтекаемой поверхности [3].

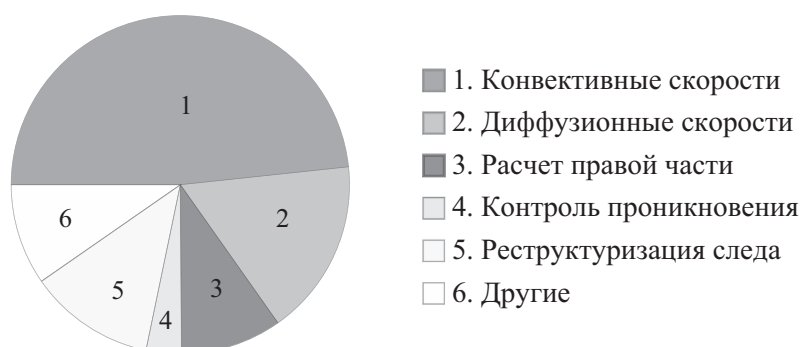
При повышении степени дискретизации число вихревых частиц может достигать сотен тысяч и даже миллионов, что резко снижает эффективность вихревых методов, если все вычислительные алгоритмы реализовать тривиальным образом, рассчитывая взаимодействие каждой частицы с каждой (в отличие от других методов частиц, к примеру, SPH, вихревые частицы обладают дальним действием, и влияние каждой частицы распространяется на всю область течения). Для решения этой проблемы была предложена идея построения семейства приближенных быстрых алгоритмов, позволяющих значительно снизить вычислительную сложность основных операций вихревого метода. Целями настоящей работы являются программная реализация названных алгоритмов и исследование их эффективности при решении модельных задач.

## 1. Об эффективности вихревых методов

Не ставя задачу полного и детального описания алгоритма вихревых методов (для этого можно обратиться к ранее упомянутым монографиям [1–5], обзору [6], а также описаниям реализации метода вязких вихревых доменов в коде VM2D [10–12]), приведем лишь краткую схему алгоритма моделирования течения среды вокруг профиля или системы профилей. Ниже перечислены основные блоки операций алгоритма, выполняемые на каждом шаге расчета по времени.

- А. Генерация завихренности на обтекаемой поверхности; математически – решение граничного интегрального уравнения (ГИУ). Помимо собственно решения ГИУ, для этого требуется решение таких вычислительно сложных задач, как аппроксимация самого оператора и расчет правой части (если профиль неподвижен – первое выполняется однократно).
- В. Моделирование эволюции завихренности в области течения; математически – решение уравнения переноса завихренности. В рамках лагранжева подхода для этого выполняются расчет конвективных и диффузионных скоростей вихревых частиц и их перемещение по суммарному полю. Также выполняются реструктуризация вихревого следа, позволяющая сократить число частиц в потоке, моделирование движения профиля (если требуется) и контроль проникновения вихревых частиц внутрь профиля.
- С. При необходимости – восстановление поля скоростей и поля давления в области течения в точках, заданных пользователем; расчет гидродинамических нагрузок, действующих на обтекаемые профили.

Задачи из каждого указанного блока существенно различаются как по подходам и алгоритмам их решения, так и по вычислительной сложности. Для наглядности на рис. 1 представлена диаграмма, показывающая соотношения времен, затрачиваемых на выполнение перечисленных операций, при рассмотрении типичной ситуации моделирования обтекания неподвижного профиля, когда все операции реализованы тривиальным образом. Видно, что наиболее трудоемким является расчет конвективных скоростей вихревых частиц.



**Рис. 1.** Распределение времени расчета по основным операциям вихревого метода при решении типичной задачи моделирования обтекания неподвижного профиля

**Fig. 1.** Distribution of computational times for the main operations of the vortex method when solving a standard problem of flow simulation around a stationary airfoil

Граница профиля обычно аппроксимируется многоугольником, состоящим из прямолинейных участков – панелей. Дискретным аналогом ГИУ относительно интенсивности вихревого слоя на профиле является система линейных алгебраических уравнений с полностью заполненной матрицей, размерность которой пропорциональна числу панелей (суммарно на всех профилях, если речь идет о моделировании обтекания системы профилей). Столбец правой части вычисляется как суперпозиция влияний на панели со стороны набегающего потока, всей завихренности в области течения и присоединенных слоев вихрей и источников (если профили подвижны).

Оценка вклада влияния завихренности в области течения – наиболее трудозатратная операция с учетом того, что количество вихревых частиц в следе может быть весьма значительным. Если рассматриваются система из нескольких профилей или один профиль, но имеющий сложную геометрическую форму, размерность матрицы системы также может стать довольно большой, поэтому в общем случае и вычисление коэффициентов матрицы, и расчет правой части, а также собственно решение системы уравнений становятся невозможными за разумное время [13].

Рассмотрим операции из второго блока описанного алгоритма. Процедура восстановления поля скоростей основана на использовании обобщенного закона Био – Савара, такие скорости называют «конвективными». Скорости необходимо вычислять по крайней мере во всех точках расположения вихревых частиц. Задача расчета конвективных скоростей аналогична гравитационной задаче  $N$  тел и имеет, таким образом, квадратичную по числу частиц вычислительную сложность. Отметим, что вычисление правой части линейной системы также можно рассматривать как расчет влияния вихревых частиц на панели профиля, поэтому вычислительная сложность этой операции пропорциональна произведению числа частиц на число панелей. Расчет компонентов матрицы линейной системы также фактически представляет собой вычисление влияния каждой панели с единичной завихренностью на ней на каждую.

При использовании метода вязких вихревых доменов для учета влияния вязкости необходимо рассчитывать и диффузионные скорости вихревых частиц, обусловленные влиянием других частиц и обтекаемой поверхности. Сложность этой операции также является квадратичной по числу частиц. Похожим образом обстоит дело и с реструктуризацией вихревого следа, а также контролем проникновения частиц. Здесь, конечно, речь не идет о законе Био – Савара, но все эти операции также суть расчет взаимодействия частиц между собой и/или с панелями профиля.

Операции из третьего блока алгоритма также обладают квадратичной сложностью. Исключение составляет расчет гидродинамических нагрузок, действующих на обтекаемые профили: в работе [14] представлены интегральные выражения для них, зависящие от интенсивности свободного вихревого слоя; вычислительная сложность такой операции пренебрежимо мала. В то же время восстановление скоростей и давления предполагает обращение к закону Био – Савара, т. е. дополнительно требует учета влияния всех вихревых частиц. Таким образом, сложность этой операции пропорциональна произведению числа частиц на число точек вычисления скоростей и давления.

Основные операции алгоритма вихревого метода, выполняемые на каждом расчетном шаге по времени, и оценки по порядку величины сложности их прямой реализации приведены в табл. 1.

Здесь  $N$  – число вихревых частиц, моделирующих завихренность в области течения;  $M$  – суммарное число панелей на границах профилей;  $P$  – число точек вычисления давления и скорости.

Если число частиц и панелей невелико (первых – не более десятков тысяч, вторых – до нескольких тысяч), то, несмотря на квадратичную вычислительную сложность рассмотренных операций, время их выполнения на современных ЭВМ оказывается весьма малым и может быть практически линейно уменьшено за счет применения технологий параллельных вычислений, наиболее эффективной из которых является перенос вычислительной работы на графические ускорители (GPU). Однако в задачах, представляющих интерес, может потребоваться довольно высокое разрешение, в результате количество частиц  $N$  может достигать сотен тысяч и даже нескольких миллионов, а число панелей на профилях  $M$  – десятков тысяч. Прямое решение указанных подзадач, наиболее характерной из которых является расчет конвективных скоростей вихрей по закону Био – Савара, становится при этом неприемлемо долгим и, по существу, сводит на нет все преимущества вихревых методов. Использование технологий параллельных вычислений (в данном случае – «экстенсивный» путь) при этом не позволяет решить проблему: простая оценка показывает, что при повышении степени дискретизации по пространству в два раза и двукратном измельчении шага по времени общая трудоемкость алгоритма решения нестационарной задачи возрастает в тридцать два раза.

**Табл. 1.** Сложность (по порядку величины) основных операций вихревых методов при прямой реализации алгоритмов

**Table 1.** Computational complexity (by the order of magnitude) for the main operations of vortex methods in case of direct algorithm implementation

| Блок | Операция                 | Сложность            |
|------|--------------------------|----------------------|
| А    | Формирование матрицы     | $M^2$                |
|      | Расчет правой части      | $NM$                 |
|      | Решение системы          | от $M^2$ до $M^3$    |
| В    | Конвективные скорости    | $(N + M)N$           |
|      | Диффузионные скорости    | $(N + M)N$           |
|      | Перемещение частиц и тел | $N + M$              |
|      | Контроль проникновения   | $NM$                 |
|      | Реструктуризация следа   | $N^2$                |
| С    | Расчет $V$ и $p$         | $(M + N)P$           |
|      | Расчет нагрузок          | от $M$ до $(N + M)M$ |

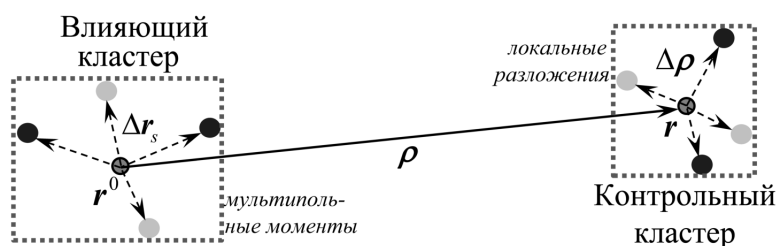
Принципиального снижения вычислительной сложности операций можно достичь путем применения приближенных быстрых алгоритмов, обладающих квазилинейной вычислительной сложностью вместо квадратичной.

Обсуждаемый здесь быстрый алгоритм разработан для решения вышеупомянутых задач и основан на объединении идей хорошо известного метода Барнса – Хата [15] и быстрого метода мультиполей [16, 17]. Предлагаемый алгоритм [18] можно рассматривать как

дальнейшее развитие метода [19] с учетом бóльшего количества слагаемых в мультипольном и локальном разложениях. Метод может быть адаптирован как к решению задачи расчета скоростей вихревых частиц и восстановления поля скоростей в целом, так и к расчету правой части, а также к итерационному решению системы линейных уравнений, аппроксимирующей граничное интегральное уравнение.

## 2. Быстрый алгоритм расчета конвективных скоростей вихревых частиц

Метод Барнса–Хата основан на иерархическом разделении области течения на под-области, что, по существу, означает построение в расчетной области  $k$ -d дерева; таким образом, все частицы объединяются в кластеры (по геометрическому принципу). Если при расчете влияния на вихревые частицы, находящиеся в контрольном кластере, влияющий кластер расположен достаточно далеко, его влияние можно с высокой точностью заменить суммированием нескольких первых членов разложения функции скорости по отрицательным степеням расстояния  $|\rho|$  между центрами кластеров. Такое разложение называют мультипольным. При этом нет необходимости рассчитывать это влияние в точках расположения всех частиц, а достаточно сделать это лишь в центре контрольного кластера  $\mathbf{r}$ . После этого для расчета скоростей расположенных в нем частиц необходимо выполнить локальное разложение всех членов упомянутого мультипольного разложения (фактически – по формуле Тейлора), считая расстояния  $\Delta\rho$  от центра кластера до отдельных частиц малыми величинами (рис. 2). Таким образом, вычислительная сложность всего алгоритма расчета конвективных скоростей составляет величину порядка  $O(N \log N)$  вместо  $O(N^2)$  по числу частиц в вихревом следе.



**Рис. 2.** Взаимодействие двух кластеров вихревых частиц

**Fig. 2.** Interaction between two clusters of vortex particles

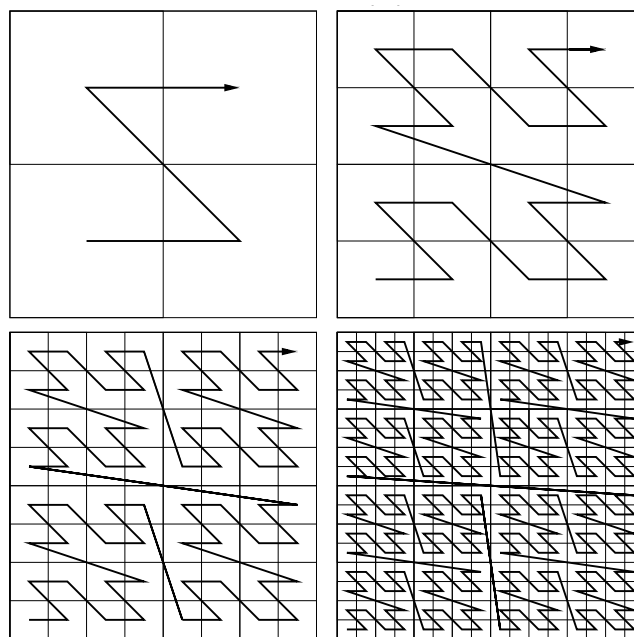
Отметим, что при реализации алгоритма для GPU, используя технологию программирования Nvidia CUDA, производить объединение частиц в контрольные кластеры явным образом не нужно; роль «контрольных кластеров» там играют блоки частиц (в терминологии CUDA), размер которых пропорционален размеру варпа (32). При этом в каждый блок должны входить частицы, расположенные в пространстве близко друг к другу, что достигается в результате их предварительной сортировки по Z-кривой (мортонским кодам).

Если влияющий кластер расположен близко к контрольному, то его влияние на все точки наблюдения рассчитывается напрямую по закону Био–Савара без использования каких-либо приближений и упрощений:

$$\mathbf{V}(\mathbf{r}) = \sum_{i=1}^{N_{\text{inf}}} \frac{\Gamma_i}{2\pi} \frac{\mathbf{k} \times (\mathbf{r} - \mathbf{r}_i)}{|\mathbf{r} - \mathbf{r}_i|^2},$$

где  $\mathbf{r}$  – радиус-вектор точки наблюдения;  $\mathbf{k}$  – единичный вектор, ортогональный плоскости течения;  $N_{\text{inf}}$  – число вихревых частиц, содержащихся во влияющем кластере;  $\mathbf{r}_i$  и  $\Gamma_i$  – положения и циркуляции этих частиц.

Наиболее эффективным способом построения дерева является использование заполняющих пространство кривых, например, кривой Мортон, позволяющих отобразить многомерные данные (ячейки дерева и вихревые частицы) в упорядоченный линейный массив (рис. 3).



**Рис. 3.** Кривая Мортон

**Fig. 3.** Morton curve

Такой алгоритм является нерекурсивным и хорошо масштабируемым. В дополнение к этому известны алгоритмы [20], позволяющие независимо и параллельно обрабатывать все ячейки дерева на всех уровнях. Затем производится обход дерева снизу вверх для расчета всех параметров ячеек – мультипольных моментов. Далее для каждой ячейки нижнего уровня (т.е. отдельных вихревых частиц или кластеров, если дерево строится лишь до определенной глубины) выполняется обход дерева сверху вниз; при этом определяются далеко расположенные ячейки, влияние которых может быть рассчитано приближенно, а также близко расположенные ячейки (нижнего уровня), влияние которых рассчитывается напрямую по закону Био – Савара. Все необходимые формулы для расчета мультипольных моментов, их сдвига к новой точке приведения, а также вычисления коэффициентов локальных разложений имеются в работе [18]. Их реализации в виде прототипов программ с исходными кодами доступны по ссылке <https://github.com/vortexmethods/fastm>.

На рис. 4 приведено сравнение времени расчета конвективных скоростей вихревых частиц с помощью прямого метода и разработанного быстрого алгоритма. Расчеты проводились в параллельном режиме: для реализации алгоритма с использованием технологии OpenMP – на 18-ядерном центральном процессоре Intel i9-10980XE; для реализации с использованием технологии Nvidia CUDA – на графическом ускорителе Nvidia Titan V.



Последняя является результатом существенной переработки подхода и кода [21], использованного первоначально для решения на GPU гравитационной задачи  $N$  тел.

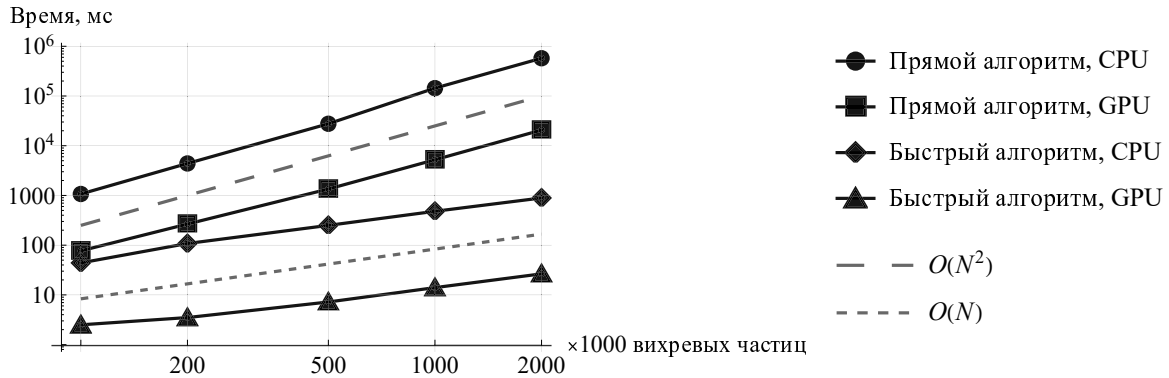


Рис. 4. Время расчета конвективных скоростей вихревых частиц

Fig. 4. Computational time for the convective velocities of vortex particles

Видно, что реализация и прямого, и быстрого алгоритмов на GPU позволяет получить ускорение примерно в тридцать раз по сравнению с расчетами на центральном процессоре. Использование быстрого алгоритма позволяет достичь ускорения в несколько сотен раз при значениях  $N$  порядка  $10^6$ .

### 3. Быстрый алгоритм для решения граничного интегрального уравнения

Интенсивность свободного вихревого слоя может быть найдена из решения граничного интегрального уравнения (ГИУ), выражающего условие прилипания. Наиболее эффективным подходом представляется использование т.н.  $T$ -модели [22, 23], основанной на равенстве касательных компонент скорости среды на границе профиля  $K$  и собственной скорости профиля, что математически означает рассмотрение ГИУ относительно интенсивности вихревого слоя  $\gamma(\mathbf{r}) = \gamma(\mathbf{r})\mathbf{k}$

$$\oint_K Q(\mathbf{r}, \boldsymbol{\xi}) \gamma(\boldsymbol{\xi}) d\ell_{\boldsymbol{\xi}} - \frac{\gamma(\mathbf{r})}{2} = f(\mathbf{r}), \quad \mathbf{r} \in K,$$

ядро которого отличается от функции расчета скоростей по закону Био–Савара лишь скалярным умножением на орт касательной в точке наблюдения:

$$Q(\mathbf{r}, \boldsymbol{\xi}) = \frac{\mathbf{k} \times (\mathbf{r} - \boldsymbol{\xi})}{2\pi|\mathbf{r} - \boldsymbol{\xi}|^2} \cdot \boldsymbol{\tau}(\mathbf{r}).$$

Отметим, что такое ядро является ограниченным для гладких профилей и абсолютно интегрируемым для профилей с угловыми точками или острыми кромками.

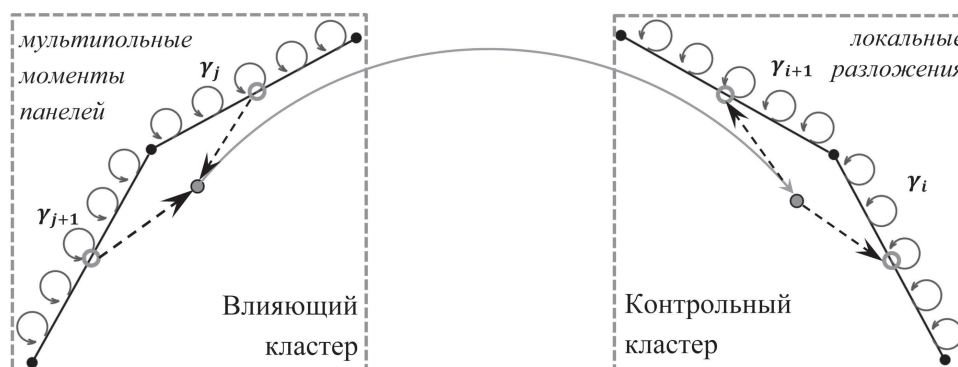
Если рассматривается задача моделирования внешнего обтекания профиля или системы профилей, такое ГИУ имеет бесконечное множество решений; для выделения единственного решения следует задать величины интеграла от решения вдоль каждого профиля:

$$\oint_{K_p} \gamma(\boldsymbol{\xi}) d\ell_{\boldsymbol{\xi}} = \Gamma_p, \quad p = 1, \dots, C,$$

где  $C$  — количество контуров.

Для решения ГИУ разработана иерархия численных схем на основе метода Галеркина [22,23], из которых оптимальной по точности/сложности при моделировании обтекания гладких профилей является схема  $\mathcal{T}^1$ , основанная на кусочно-линейном представлении решения  $\gamma(\mathbf{r})$  на панелях профиля. Матрица соответствующей линейной системы, полученной из условия ортогональности невязки системе проекционных кусочно-линейных функций, имеет размерность  $(2M + C) \times (2M + C)$ . Следует отметить, что если рассматривается один профиль или система из нескольких профилей, неподвижных друг относительно друга, то трудоемкость операции решения СЛАУ на каждом расчетном шаге можно значительно снизить: коэффициенты матрицы системы остаются постоянными, поэтому она может быть сформирована и обращена однократно в начале расчета, а на каждом шаге следует лишь умножать обратную матрицу на новый вектор правой части. В противном случае, если с течением времени взаимное расположение профилей изменяется, на каждом шаге необходимо решать систему с новой матрицей. При достаточно подробной дискретизации профилей трудоемкость данной операции многократно возрастает.

Вычисление каждого коэффициента  $a_{ij}$  матрицы системы, за исключением последних  $C$  строк и столбцов, можно трактовать как расчет влияния  $j$ -й панели с заданным распределением завихренности на  $i$ -ю (контрольную) панель. Таким образом, требуется производить интегрирование функции влияния, причем дважды – по влияющей и контрольной панелям, с постоянными и линейными базисными и проекционными функциями. Первое реализуется за счет введения мультипольных моментов панелей, которые можно вычислить, если суммирование по частицам заменить на интегрирование вдоль панели. Второе реализуется переходом от интегрирования самой функции влияния к интегрированию ее локального разложения (с необходимыми весами – проекционными функциями). Панели, далеко расположенные друг от друга, могут быть объединены в кластеры, и их влияние может быть вычислено приближенно (рис. 5).



**Рис. 5.** Влияние кластеров панелей профиля  
**Fig. 5.** Influence of the clusters of airfoil panels

Для решения линейной системы можно использовать итерационные алгоритмы. При выполнении каждой итерации происходит умножение матрицы системы на некоторый вектор, который можно формально считать заданным распределением завихренности. В силу приведенных выше соображений это умножение можно рассматривать как форму вычисления вихревого влияния и, значит, выполнять приближенно, используя фактически

тот же быстрый алгоритм Барнса – Хата/мультиполей, что и для расчета конвективных скоростей.

Приведем основные шаги быстрого алгоритма матрично-векторного умножения.

1. Построение  $k$ -d дерева на основе фрактальной кривой Мортонa (по центрам панелей профиля).
2. Расчет мультипольных моментов панелей; отличие от расчета скоростей вихревых частиц заключается в том, что панели характеризуются не только монопольным моментом – содержащейся в них суммарной завихренностью, но и моментами старших порядков, характеризующими ее распределение вдоль панели.
3. Расчет мультипольных моментов всех ячеек дерева путем суммирования моментов содержащихся в них панелей (для ячеек нижнего уровня, т.е. листьев дерева) или ячеек-потомков с предварительным приведением их моментов к центру родительской ячейки.
4. Для каждой ячейки нижнего уровня выполняем:
  - обход дерева от корня к листьям: определение дальних и ближних ячеек;
  - для дальних ячеек: расчет (путем накопления) коэффициентов локального разложения;
  - для ячеек ближней зоны: расчет их влияния напрямую;
  - расчет приближенного влияния дальней зоны с использованием накопленных коэффициентов локального разложения.

Вычисление столбца правой части системы линейных уравнений – это расчет влияния вихревых частиц на панели профиля, который может быть выполнен приближенно с использованием комбинации описанных алгоритмов. Необходимо построить два дерева (по частицам и панелям), затем влияние далеко расположенных кластеров частиц рассчитывать по быстрому алгоритму, выполнив при этом интегрирование по контрольным панелям с требуемыми проекционными функциями.

При рассмотрении различных итерационных алгоритмов решения линейных систем наиболее эффективным оказался обобщенный метод минимальных невязок (GMRES) с дополнительно реализованной процедурой предобуславливания. Время решения системы линейных уравнений при достаточно подробной дискретизации профиля представлено в табл. 2. В качестве тестовой была взята задача о моделировании обтекания системы из 9 близко расположенных круговых профилей, каждый из которых был разбит сначала на 1 000, а потом на 4 000 панелей. Были рассмотрены три способа решения линейной системы: метод исключения Гаусса, реализованный в библиотеке Intel MKL и имеющий кубическую вычислительную сложность; итерационный метод GMRES (с вычислением всей матрицы и ее прямым умножением на вектор), имеющий квадратичную вычислительную сложность каждой итерации, а также метод GMRES с выполнением матрично-векторного умножения по предлагаемому быстрому алгоритму, имеющий квазилинейную вычислительную сложность итерации. Расчеты проводились в параллельном режиме на 18-ядерном процессоре Intel i9-10980XE, для первых двух методов указаны время формирования матрицы и собственно время решения СЛАУ.

**Табл. 2.** Время решения системы линейных уравнений**Table 2.** Computation time for solving the system of linear equations

| Метод               | $9 \times 1\,000$ панелей | $9 \times 4\,000$ панелей |
|---------------------|---------------------------|---------------------------|
| Гаусс (MKL)         | $1.81 + 5.27$ с           | $29.3 + 268$ с            |
| GMRES               | $1.81 + 0.61$ с           | $29.3 + 8.80$ с           |
| GMRES + быстрый м-д | $0.20$ с                  | $0.68$ с                  |

Ускорение матрично-векторного умножения за счет использования быстрого алгоритма составляет десятки и даже сотни раз. Следует отметить, что помимо этого также достигается существенная экономия машинной памяти, поскольку при использовании быстрого алгоритма необходимо хранить лишь ту часть коэффициентов матрицы, которая вычисляется точно. Например, при решении указанных задач для хранения полной матрицы системы потребовалось 2.6 Гб и 41.5 Гб оперативной памяти соответственно, тогда как при использовании быстрого метода при количестве панелей, превышающем 5 000, оказалось достаточным явного вычисления и хранения менее 1 % компонент, и эта доля быстро уменьшается с ростом размерности системы.

### Заключение

Реализован быстрый алгоритм квазилинейной вычислительной сложности на основе метода Барнса – Хата с использованием мультипольных и локальных разложений для задач расчета конвективных скоростей вихревых частиц. Мультипольное разложение, содержащее достаточное количество членов, позволило с высокой точностью рассчитать влияние кластеров вихревых частиц на достаточно удаленную точку наблюдения. Локальные разложения мультипольных слагаемых позволили также объединять точки наблюдения в кластеры и ограничиться вычислением членов мультипольного разложения в их центрах. Использование быстрого алгоритма позволило достичь ускорения в сотни раз по сравнению с прямым расчетом. Разработаны параллельные реализации алгоритма расчета конвективных скоростей частиц для многоядерных процессоров по технологии OpenMP и для графических ускорителей с использованием технологии Nvidia CUDA.

Выполнена адаптация быстрого алгоритма к решению граничного интегрального уравнения с применением расчетных схем повышенной точности. Модификация основана на учете мультипольных моментов панелей и интегрального характера обеспечения граничного условия. Помимо значительного уменьшения времени проведения расчетов, не потребовалось вычислять и хранить полную матрицу системы, что привело к существенной экономии машинной памяти.

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

**Conflicts of Interest.** The authors declare no conflicts of interest.

### Литература

1. *Lewis R.I.* Vortex Element Methods for Fluid Dynamic Analysis of Engineering Systems. Cambridge Engine Technology Ser. 1. Cambridge, New York, NY: Cambridge Univ. Press, 1991. xxii, 566 p. <https://doi.org/10.1017/CBO9780511529542>.
2. *Cottet G.-H., Koumoutsakos P.D.* Vortex Methods: Theory and Practice. Cambridge: Cambridge Univ. Press, 2000. xiv, 313 p. <https://doi.org/10.1017/CBO9780511526442>.
3. *Андронов П.Р., Гувернюк С.В., Дынникова Г.Я.* Вихревые методы расчета нестационарных гидродинамических нагрузок. М.: МГУ, 2006. 184 с.
4. *Головкин М.А., Головкин В.А., Каляев В.М.* Вопросы вихревой гидромеханики. М.: Физматлит, 2009. 265 с.
5. *Branlard E.* Wind Turbine Aerodynamics and Vorticity-Based Methods: Fundamentals and Recent Applications. Ser.: Research Topics in Wind Energy. V. 7. Cham: Springer, 2017. xxxi, 632 p. <https://doi.org/10.1007/978-3-319-55164-7>.
6. *Mimeau C., Mortazavi I.* A review of vortex methods and their applications: From creation to recent advances // *Fluids*. 2021. V. 6, No 2. Art. 68. <https://doi.org/10.3390/fluids6020068>.
7. *Гувернюк С.В., Дынникова Г.Я.* Моделирование обтекания колеблющегося профиля методом вязких вихревых доменов // *Изв. РАН. МЖГ*. 2007. № 1. С. 3–14.
8. *Дынникова Г.Я.* Лагранжев подход к решению нестационарных уравнений Навье – Стокса // *ДАН*. 2004. Т. 399, № 1. С. 42–46.
9. *Ogami Y., Akamatsu T.* Viscous flow simulation using the discrete vortex model — the diffusion velocity method // *Comput. Fluids*. 1991. V. 19, Nos 3–4. P. 433–441. [https://doi.org/10.1016/0045-7930\(91\)90068-S](https://doi.org/10.1016/0045-7930(91)90068-S).
10. *Kuzmina K., Marchevsky I., Ryatina E.* VM2D: Open source code for 2D incompressible flow simulation by using vortex methods // *Parallel Computational Technologies. PCT 2018: Proc. 12th Int. Conf. / Ed. by L. Sokolinsky, M. Zymbler. Ser.: Communications in Computer and Information Science. V. 910. Cham: Springer, 2018. P. 251–265. [https://doi.org/10.1007/978-3-319-99673-8\\_18](https://doi.org/10.1007/978-3-319-99673-8_18).*
11. *Kuzmina K., Marchevsky I., Soldatova I., Izmailova Y.* On the scope of Lagrangian vortex methods for two-dimensional flow simulations and the POD technique application for data storing and analyzing // *Entropy*. 2021. V. 23, No 1. Art. 118. <https://doi.org/10.3390/e23010118>.
12. *Marchevsky I., Sokol K., Ryatina E., Izmailova Y.* The VM2D open source code for two-dimensional incompressible flow simulation by using fully Lagrangian vortex particle methods // *Axioms*. 2023. V. 12, No 3. Art. 248. <https://doi.org/10.3390/axioms12030248>.
13. *Kolganova A., Marchevsky I.* Parallel implementation of fast algorithms in the vortex particle method // *Parallel Computational Technologies. PCT 2023: Proc. 17th Int. Conf. / Ed. by L. Sokolinsky, M. Zymbler. Ser.: Communications in Computer and Information Science. V. 1868. Cham: Springer, 2023. P. 197–211. [https://doi.org/10.1007/978-3-031-38864-4\\_14](https://doi.org/10.1007/978-3-031-38864-4_14).*
14. *Дынникова Г.Я.* Аналог интегралов Бернулли и Коши – Лагранжа для нестационарного вихревого течения идеальной несжимаемой жидкости // *Изв. РАН. МЖГ*. 2000. № 1. С. 31–41.
15. *Barnes J., Hut P.* A hierarchical  $O(N \log N)$  force-calculation algorithm // *Nature*. 1986. V. 324, No 6096. P. 446–449. <https://doi.org/10.1038/324446a0>.
16. *Greengard L., Rokhlin V.* A fast algorithm for particle simulations // *J. Comput. Phys*. 1987. V. 73, No 2. P. 325–348. [https://doi.org/10.1016/0021-9991\(87\)90140-9](https://doi.org/10.1016/0021-9991(87)90140-9).

17. Carrier J., Greengard L., Rokhlin V. A fast adaptive multipole algorithm for particle simulations // *SIAM J. Sci. Stat. Comput.* 1988. V. 9, No 4. P. 669–686.  
<https://doi.org/10.1137/0909044>.
18. Marchevsky I., Ryatina E., Kolganova A. Fast Barnes–Hut-based algorithm in 2D vortex method of computational hydrodynamics // *Comput. Fluids*. 2023. V. 266. Art. 106018.  
<https://doi.org/10.1016/j.compfluid.2023.106018>.
19. Дынникова Г.Я. Использование быстрого метода решения задачи  $N$  тел при вихревом моделировании течений // *Журн. вычисл. матем. и матем. физ.* 2009. Т. 49, № 8. С. 1458–1465.
20. Karras T. Maximizing parallelism in the construction of BVHs, octrees, and  $k$ -d trees // *Proc. Eurographics/ACM SIGGRAPH Symp. on High Performance Graphics. Collect.: High-Performance Graphics 2012*. Eurographics Assoc., 2012. P. 33–37.  
<https://doi.org/10.2312/EGGH/HPG12/033-037>.
21. Burtscher M., Pingali K. An efficient CUDA implementation of the tree-based Barnes Hut  $n$ -body algorithm // *GPU Computing Gems Emerald Edition* / Ed. by Wen-mei W. Hwu. Ch. 6. Morgan Kaufmann Publ., 2011. P. 75–92. <https://doi.org/10.1016/B978-0-12-384988-5.00006-1>.
22. Kempka S.N., Glass M.W., Peery J.S., Strickland J.H., Ingber M.S. Accuracy considerations for implementing velocity boundary conditions in vorticity formulations. SANDIA Report No. SAND96-0583, UC-700. Sandia Natl. Lab., 1996. 52 p. <https://doi.org/10.2172/242701>.
23. Марчевский И.К., Сокол К.С., Измайлова Ю.А. Т-схемы для математического моделирования генерации завихренности на гладких профилях в вихревых методах // *Вестн. МГТУ им. Н.Э. Баумана. Сер. Ест. науки*. 2022. № 6 (105). С. 33–59.  
<https://doi.org/10.18698/1812-3368-2022-6-33-59>.

## References

1. Lewis R.I. *Vortex Element Methods for Fluid Dynamic Analysis of Engineering Systems*. Cambridge Engine Technology Ser. 1. Cambridge, New York, NY, Cambridge Univ. Press, 1991. xxii, 566 p. <https://doi.org/10.1017/CBO9780511529542>.
2. Cottet G.-H., Koumoutsakos P.D. *Vortex Methods: Theory and Practice*. Cambridge, Cambridge Univ. Press, 2000. xiv, 313 p. <https://doi.org/10.1017/CBO9780511526442>.
3. Andronov P.R., Guvernyuk S.V., Dynnikova G.Ya. *Vikhrevye metody rascheta nestatsionarnykh gidrodinamicheskikh nagruzok* [Vortex Methods for Non-Stationary Hydrodynamic Loads Calculation]. Moscow, MGU, 2006. 184 p. (In Russian)
4. Golovkin M.A., Golovkin V.A., Kalyavkin V.M. *Voprosy vikhrevoi gidromekhaniki* [Issues of Vortex Hydromechanics]. Moscow, Fizmatlit, 2009. 265 p. (In Russian)
5. Branlard E. *Wind Turbine Aerodynamics and Vorticity-Based Methods: Fundamentals and Recent Applications*. Ser.: Research Topics in Wind Energy. Vol. 7. Cham, Springer, 2017. xxxi, 632 p. <https://doi.org/10.1007/978-3-319-55164-7>.
6. Mimeau C., Mortazavi I. A review of vortex methods and their applications: From creation to recent advances. *Fluids*, 2021, vol. 6, no. 2, art. 68. <https://doi.org/10.3390/fluids6020068>.
7. Guvernyuk S.V., Dynnikova G.Ya. Modeling the flow past an oscillating airfoil by the method of viscous vortex domains. *Fluid Dyn.*, 2007, vol. 42, no. 1, pp. 1–11.  
<https://doi.org/10.1134/S0015462807010012>.
8. Dynnikova G.Ya. The Lagrangian approach to solving the time-dependent Navier–Stokes equations. *Dokl. Phys.*, 2004, vol. 49, no. 11, pp. 648–652. <https://doi.org/10.1134/1.1831530>.

9. Ogami Y., Akamatsu T. Viscous flow simulation using the discrete vortex model — the diffusion velocity method. *Comput. Fluids*, 1991, vol.19, nos. 3–4, pp. 433–441. [https://doi.org/10.1016/0045-7930\(91\)90068-S](https://doi.org/10.1016/0045-7930(91)90068-S).
10. Kuzmina K., Marchevsky I., Ryatina E. VM2D: Open source code for 2D incompressible flow simulation by using vortex methods. *Parallel Computational Technologies. PCT 2018: Proc. 12th Int. Conf.* Sokolinsky L., Zymbler M. (Eds.). Ser.: Communications in Computer and Information Science. Vol. 910. Cham, Springer, 2018, pp. 251–265. [https://doi.org/10.1007/978-3-319-99673-8\\_18](https://doi.org/10.1007/978-3-319-99673-8_18).
11. Kuzmina K., Marchevsky I., Soldatova I., Izmailova Y. On the scope of Lagrangian vortex methods for two-dimensional flow simulations and the POD technique application for data storing and analyzing. *Entropy*, 2021, vol. 23, no. 1, art. 118. <https://doi.org/10.3390/e23010118>.
12. Marchevsky I., Sokol K., Ryatina E., Izmailova Y. The VM2D open source code for two-dimensional incompressible flow simulation by using fully Lagrangian vortex particle methods. *Axioms*, 2023, vol. 12, no. 3, art. 248. <https://doi.org/10.3390/axioms12030248>.
13. Kolganova A., Marchevsky I. Parallel implementation of fast algorithms in the vortex particle method. *Parallel Computational Technologies. PCT 2023: Proc. 17th Int. Conf.* Sokolinsky L., Zymbler M. (Eds.). Ser.: Communications in Computer and Information Science. Vol. 1868. Cham, Springer, 2023, pp. 197–211. [https://doi.org/10.1007/978-3-031-38864-4\\_14](https://doi.org/10.1007/978-3-031-38864-4_14).
14. Dynnikova G.Ya. An analog of the Bernoulli and Cauchy-Lagrange integrals for a time-dependent vortex flow of an ideal incompressible fluid. *Fluid Dyn.*, 2000, vol. 35, no. 1, pp. 24–32. <https://doi.org/10.1007/BF02698782>.
15. Barnes J., Hut P. A hierarchical  $O(N \log N)$  force-calculation algorithm. *Nature*, 1986, vol. 324, no. 6096, pp. 446–449. <https://doi.org/10.1038/324446a0>.
16. Greengard L., Rokhlin V. A fast algorithm for particle simulations. *J. Comput. Phys.*, 1987, vol. 73, no. 2, pp. 325–348. [https://doi.org/10.1016/0021-9991\(87\)90140-9](https://doi.org/10.1016/0021-9991(87)90140-9).
17. Carrier J., Greengard L., Rokhlin V. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Stat. Comput.*, 1988, vol. 9, no. 4, pp. 669–686. <https://doi.org/10.1137/0909044>.
18. Marchevsky I., Ryatina E., Kolganova A. Fast Barnes–Hut-based algorithm in 2D vortex method of computational hydrodynamics. *Comput. Fluids*, 2023, vol. 266, art. 106018. <https://doi.org/10.1016/j.compfluid.2023.106018>.
19. Dynnikova G.Ya. Fast technique for solving the  $N$ -body problem in flow simulation by vortex methods. *Comput. Math. Math. Phys.*, 2009, vol. 49, no. 8, pp. 1389–1396. <https://doi.org/10.1134/S0965542509080090>.
20. Karras T. Maximizing parallelism in the construction of BVHs, octrees, and  $k$ -d trees. *Proc. Eurographics/ACM SIGGRAPH Symp. on High Performance Graphics*. Collect.: High-Performance Graphics 2012. Eurographics Assoc., 2012, pp. 33–37. <https://doi.org/10.2312/EGGH/HPG12/033-037>.
21. Burtscher M., Pingali K. An efficient CUDA implementation of the tree-based Barnes Hut  $n$ -body algorithm. In: Hwu Wen-mei W. (Ed.) *GPU Computing Gems Emerald Edition*. Ch. 6. Morgan Kaufmann Publ., 2011, pp. 75–92. <https://doi.org/10.1016/B978-0-12-384988-5.00006-1>.
22. Kempka S.N., Glass M.W., Peery J.S., Strickland J.H., Ingber M.S. *Accuracy considerations for implementing velocity boundary conditions in vorticity formulations*. SANDIA Report No. SAND96-0583, UC-700. Sandia Natl. Lab., 1996. 52 p. <https://doi.org/10.2172/242701>.

23. Marchevsky I.K., Sokol K.S., Izmailova Yu.A. *T*-schemes for mathematical modelling of vorticity generation on smooths airfoils in vortex particle methods. *Vestn. MGTU im. N.E. Baumana. Ser. Estestv. Nauki*, 2022, no. 6 (105), pp. 33–59.  
<https://doi.org/10.18698/1812-3368-2022-6-33-59>. (In Russian)

### Информация об авторах

**Евгения Павловна Рятина**, аспирант кафедры «Прикладная математика», Московский государственный технический университет им. Н.Э. Баумана

E-mail: [evgeniya.ryatina@yandex.ru](mailto:evgeniya.ryatina@yandex.ru)

ORCID: <https://orcid.org/0000-0002-7533-3796>

**Илья Константинович Марчевский**, доктор физико-математических наук, доцент, профессор кафедры «Прикладная математика», Московский государственный технический университет им. Н.Э. Баумана

E-mail: [iliamarchevsky@mail.ru](mailto:iliamarchevsky@mail.ru)

ORCID: <https://orcid.org/0000-0003-4899-4828>

**Александра Олеговна Колганова**, студент кафедры «Прикладная математика», Московский государственный технический университет им. Н.Э. Баумана

E-mail: [kolganova.ao@mail.ru](mailto:kolganova.ao@mail.ru)

**Дарья Юрьевна Кобзарь**, студент кафедры «Прикладная математика», Московский государственный технический университет им. Н.Э. Баумана

E-mail: [cobzardash@yandex.ru](mailto:cobzardash@yandex.ru)

### Author Information

**Evgeniya P. Ryatina**, Postgraduate Student, Department of Applied Mathematics, Bauman Moscow State Technical University

E-mail: [evgeniya.ryatina@yandex.ru](mailto:evgeniya.ryatina@yandex.ru)

ORCID: <https://orcid.org/0000-0002-7533-3796>

**Ilya K. Marchevsky**, Dr. Sci. (Physics and Mathematics), Professor, Department of Applied Mathematics, Bauman Moscow State Technical University

E-mail: [iliamarchevsky@mail.ru](mailto:iliamarchevsky@mail.ru)

ORCID: <https://orcid.org/0000-0003-4899-4828>

**Aleksandra O. Kolganova**, Student, Department of Applied Mathematics, Bauman Moscow State Technical University

E-mail: [kolganova.ao@mail.ru](mailto:kolganova.ao@mail.ru)

**Dar'ya Yu. Kobzar**, Student, Department of Applied Mathematics, Bauman Moscow State Technical University

E-mail: [cobzardash@yandex.ru](mailto:cobzardash@yandex.ru)

Поступила в редакцию 31.07.2024

Принята к публикации 16.12.2024

Received July 31, 2024

Accepted December 16, 2024