МАТРИЧНЫЕ МЕТОДЫ И ВЫЧИСЛЕНИЯ

МОСКВА 1999

Российская академия наук
Институт вычислительной математики

МАТРИЧНЫЕ МЕТОДЫ И ВЫЧИСЛЕНИЯ

Сборник научных трудов
под редакцией Е. Е. Тыртышникова

Москва 1999

УДК 519.6

Матричные методы и вычисления: Сборник научных трудов / Под редакцией Е. Е. Тыртышникова. — М.: Институт вычислительной математики РАН, 1999. — 208 с.

В сборник включены работы, посвященные быстрому приближенному матричному умножению, аппроксимационным и спектральным свойствам структурированных матриц и их связям с приложениями.

Сборник рассчитан на широкий круг научных работников, специализирующихся в области численных методов и разработки современного программного обеспечения.

Presented are the papers on fast approximate matrix multiplication, approximation and spectral properties of structured matrices and their applications.

The book is intended for wide audience of scientists, specialising on numerical methods and development of contemporary software.

ISBN 5-201-08803-1

275-летию
Российской
академии наук

ПРЕДИСЛОВИЕ

В настоящем сборнике публикуются труды нашего семинара „Матричные методы и вычисления“ в Институте вычислительной математики РАН. Семинар начал активно действовать не менее пяти лет тому назад. Излюбленные темы семинара — спектральные и аппроксимационные свойства структурированных матриц и их замечательные связи с различными областями математики и приложений.

Мы готовы, наконец, подвести некоторый итог нашим исследованиям по „сжатию“ больших плотных неструктурированных матриц — естественно, на основе аппроксимации их „хорошими“ структурированными матрицами. Это и есть главная тема сборника. Мою статью можно рассматривать как расширенное введение: пользуясь случаем, я решил собрать свои черновые заметки по методам быстрого умножения, которые давно уже использовались в том или ином виде для того, чтобы войти в проблему.

Все работы, включенные в сборник, выполнялись при поддержке РФФИ (грант 97-01-00155); большая часть работ была поддержана также фондом Volkswagen-Stiftung (грант VW I/71 493). Разработка эффективных методов для параллельных машин — в частности, МВС-100 и 1000 — была частично поддержана грантом Министерства науки (проект 0201.01.01). Мы признательны всем названным организациям.

С особой благодарностью я хочу отметить, что сборник выходит в свет за счет средств гранта РФФИ на поддержку научной школы члена-корреспондента РАН В. В. Воеводина, к которой мои коллеги и я имеем честь и удовольствие принадлежать.

Е. Е. Тыртышников

Методы быстрого умножения и решение уравнений*

Е. Е. Тыртышников

Аннотация

Цель работы — дать представление о подходах к быстрому приближенному умножению плотной неструктурированной матрицы на вектор. С точки зрения матричного анализа, во всех случаях ключевой идеей является аппроксимация матриц общего вида структурированными матрицами.

1. ВВЕДЕНИЕ. Попробуем посмотреть, какие существуют методы *приближенного* быстрого матрично-векторного умножения для плотных матриц достаточно общего вида, возникающих при решении интегральных уравнений теории потенциала в задачах электростатики, гидродинамики, акустики, электродинамики и т. п. Метод считается „быстрым“ для $n \times n$-матрицы, если он требует *арифметических действий и памяти* порядка $o(n^2)$. Все, что мы знаем о таких методах, возникло за последние 10–15 лет.

Мультипольный метод (Fast Multipole Method). Автором метода является В.Рохлин (V.Rokhlin); первая публикация была в 1985 году [44], а затем пошла целая серия работ по развитию, реализации и приложениям. Родственным является *метод кластеризации граничных элементов* (panel clustering method), предложенный Хакбушем и Новаком [25]; здесь, в частности, можно найти концепцию *дерева кластеров* (мы используем ее при построении мозаичных биений в методе скелетонов).

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155) и фонда Volkswagen-Stiftung (грант № VW I/71 493).

Интерполяционный метод. Задача сводится к аналогичной задаче с плотной матрицей, отвечающей (более мелкой) регулярной сетке. Размер этой матрицы больше, но она оказывается теплицевой или блочно теплицевой. Поэтому для нее существуют быстрые алгоритмы умножения, использующие быстрое преобразование Фурье.

Фактически эта идея рассматривалась Ю. М. Нечепуренко в 1985 году [4]. Она же получила многоуровневое развитие в работах А.Брандта (A. Brandt) [12]. При этом оказывается, что сетки могут и не быть регулярными.

Метод локальных волн. С помощью преобразований, дающих разложения по „локальным волнам" (wavelets), исходная матрица превращается в псевдоразреженную (то есть становится разреженной, если пренебречь относительно малыми элементами; А.Хартен (A.Harten) называет такие матрицы Т-разреженными). См., например, [10,15]. Но идея продолжает развиваться весьма активно — особенно последние 5–7 лет.

Метод скелетонов. Под (двумерным) скелетоном понимается матрица ранга 1. Обычно в матрице существуют довольно большие блоки, которые хорошо аппроксимируются суммой малого числа скелетонов. Для приближенного умножения предлагается использовать блочно малоранговые аппроксимации. Редкий случай — можно точно сказать, когда метод появился на свет: 21 января 1993 года. Этой датой помечен манускрипт с изложением идеи метода.* В не вполне осознанном виде идея, вероятно, витала и раньше; я бы отметил, что еще в 1979 году В.В.Воеводин рассматривал блочно малоранговые аппроксимации — правда, специального вида и в несколько ином контексте [1].

Но лишь в нынешнем году метод приобрел приятную завершенность — появился надежный алгоритм выбора „опорного креста" для неполной скелетонной аппроксимации. В результате не требуется вычислять все элементы матрицы, а лишь относительно малое их число — такое же по порядку, как память для хранения

*E.E.Tyrtyshnikov, Matrix approximation and cost-effective matrix-vector multiplication, Jan. 21, 1993, IVM RAN.

скелетонов.

Все названные методы являются *операторными* (другими словами, *безматричными*, так как исходная матрица в явном виде не участвует при выполнении умножения). Однако первые два метода являются таковыми в большей степени, так как исходная матрица в полном виде не появляется никогда. Локальные волны применяются двояко: как с участием исходной матрицы на начальном этапе, так и без нее. Метод скелетонов является *наиболее матричным*: он работает с элементами блоков исходной матрицы. Нам удалось, наконец, получить надежные алгоритмы аппроксимации блоков лишь по малой части их элементов; теоретическая основа для этого получена в работах [2,22].

Метод скелетонов выгодно отличается от других тем, что в нем естественным образом вычисляется *a posteriori* погрешность приближенного умножения. Явного описания способа *вычисления* такой оценки для других методов нигде не приводится.

Все методы разделяют элементы исходной матрицы на две группы, отвечающие *дальнодействию* и *близкодействию*. Это разделение зависит от вида оператора исходной задачи и, по-видимому, не зависит от вида базисных функций, способа дискретизации и моделирования поверхности (именно это декларируется в [18]). Поскольку все сингулярности находятся в зоне близкодействия, работа с ними должна проводиться традиционными (прежними) методами (то есть это не предмет беспокойства для методов быстрого умножения). Заметим, что в методе скелетонов при выделении зон дальнодействия полезно иметь, помимо матрицы коэффициентов, дополнительную информацию о расположении и нумерации „источников“ и „точек наблюдения“. Таким образом, матрица задачи рассматривается не как „черный ящик“, а скорее как „серый ящик“ — „серый“ потому, что объем дополнительной информации все же весьма умеренный. Все методы естественным образом связываются с идеями иерархического анализа.

Итак, мы имеем четыре метода (с рядом вариаций в каждом). Чтобы сравнить их, казалось бы, достаточно посмотреть на формулы для арифметических затрат и памяти:

$$\mathrm{Work}\,(\varepsilon, n) = ?, \qquad \mathrm{Memory}\,(\varepsilon, n) = ? \qquad (*)$$

в каждой из них: n — порядок матрицы A и ε — предписанная точность приближенного умножения. Вместо Ax вычисляется $\tilde{A}x$ и гарантируется, что

$$\|Ax - \tilde{A}x\| \leqslant \varepsilon \|x\|.$$

Для сравнения всюду нужно использовать одну и ту же (вероятно, евклидову) норму.

Однако эти определяющие формулы найти в существующих работах не очень просто. Хотя, казалось бы, есть почти все, чтобы их получить. Несмотря на полезность асимптотических оценок, заметим, что их недостаточно для сравнения с прагматической точки зрения. Еще один интересный вопрос: как выбирать ε? Вероятно, должны быть формулы $\varepsilon = \varepsilon(n)$ — в зависимости, конечно, от вида исходного оператора. Но не только от него. Здесь способ дискретизации (и т. п.) имеет решающее значение. Ни в одной из существующих работ таких формул не приводится.

При решении уравнений методы быстрого умножения используются для ускорения каждой отдельной итерации. Чтобы уменьшить общее число итераций, обычно строят предобусловливатели. Как правило, в работах по быстрому умножению проблема предобусловливания не рассматривается. Но часто она является не менее важной.

Наиболее простыми (концептуально) и, вероятно, наиболее полезными (на практике) являются *матричные* предобусловливатели (они строятся по всем или по части элементов исходной матрицы). Чаще всего при построении матричных предобусловливателей исходная матрица рассматривается как T-разреженная (для „не очень малого“ порога); в этом случае можно говорить о *T-предобусловливателях*. Обычно неплохо работают различные версии неполной треугольной и блочной факторизации (LU и IBF). Но для них нужно иметь *элементы* исходной матрицы. В определенном смысле, это проблема для всех методов быстрого умножения, так как они дают в итоге лишь *процедуру* применения матрицы $\tilde{A}$. Генерация большой совокупности элементов (до 10–20 % от общего числа) с помощью этой процедуры будет стоить дорого. Поэтому эти элементы следует вычислять *отдельно*: кроме процедуры быстрого умножения, необходимо еще иметь процедуру, эффективно вычисляющую предписанные элементы исходной матрицы. Более того, необходимо знать позиции предписанных элементов.

Все это требует вторжения в сам процесс генерации процедуры быстрого умножения. Задача, конечно, решаемая — но самым естественным образом именно в методе скелетонов.

Мне кажется, может возникнуть полезная обратная связь: при определении зоны дальнодействия могут приниматься во внимание параметры предобусловливателя. Разумно согласовывать затраты на генерацию процедур умножения и предобусловливания, но еще более важно согласовывать затраты в самих процедурах: вряд ли стоит стремиться умножать как можно быстрее, если все „тормозится“ предобусловливателем.

Теперь, когда необходимые акценты расставлены, мы готовы перейти к более предметному описанию методов. Чтобы не увязнуть в деталях, постараемся делать это на концептуальном уровне. Я надеюсь, это позволит читателю очень быстро получить представление сразу о всех существующих подходах. Всюду нас будет интересовать, насколько трудно получить формулы $(*)$, необходимые для прагматического сравнения разных методов быстрого умножения.

2. ПОСТАНОВКА ЗАДАЧИ. Вообще говоря, имеется интегральный оператор

$$v(x) = [\mathcal{K}u(y)](x) \equiv \int K(x,y)u(y)dy$$

и по функции u требуется найти функцию v. Чтобы провести вычисление, вводятся *дискретные аналоги* функций u и v.

Задача 1. Заданы значения $u(y)$ на какой-то сетке с узлами y_j и ищутся значения $v(x)$ на какой-то (возможно, другой) сетке с узлами x_i. Интеграл вычисляется по какой-либо квадратурной формуле. Требуется найти значения дискретной функции p (поле) по значениям дискретной функции q (заряд):

$$p_i = \sum_j K(x_i, y_j)q_j.$$

Задача 2. Заданы базисные функции $v_j(y)$ и по коэффициентам разложения

$$u(y) = \sum_j q_j v_j(y)$$

требуется найти

$$p_i = \sum_j q_j \int\limits_{\operatorname{supp} v_j} K(x, y) v_j(y)\, dy.$$

Каждый член суммы трактуется как поле от локального заряда, распределенного в области $\operatorname{supp} v_j$ с плотностью $q_j v_j(y)$. Если используются квадратуры, то эта задача легко сводится к задаче 1.

Задача 3. Требуется вычислить моменты

$$p_i = \sum_j q_j \int\limits_{\operatorname{supp} v_i} \int\limits_{\operatorname{supp} v_j} u_i(x) K(x, y) v_j(y)\, dy\, dx.$$

Часто для внешнего интегрирования используют квадратуры, и тогда задача сводится к задаче 2.

В контексте решения уравнений задача 1 отвечает простому квадратурному методу (Nystrom's method), задача 2 — методу коллокации, задача 3 — методу моментов (общему проекционному методу или, в частности, методу Галеркина).

3. МУЛЬТИПОЛЬНЫЙ МЕТОД

3.1. МУЛЬТИПОЛЬНОЕ И ЛОКАЛЬНОЕ РАЗЛОЖЕНИЯ. Пусть $K(x, y)$ — ядро интегрального оператора; $x, y \in \mathbb{R}^\nu$. Часто оказывается так, что если y ближе к z, чем к x, то

$$K(x, y) \approx K'_N(x, y) \equiv \sum_{l=0}^{N} \sum_{m \in \mathcal{M}_l} t_{lm}(y, z) T_{lm}(x, z),$$

и при этом

$$|K(x, y) - K'_N(x, y)| \leqslant c_N \left(\frac{|y - z|}{|x - z|} \right)^N.$$

Если же x ближе к z, чем к y, то выполняется двойственное соотношение

$$K(x, y) \approx K''_N(x, y) \equiv \sum_{l=0}^{N} \sum_{m \in \mathcal{M}_l} T_{lm}(y, z) t_{lm}(x, z),$$

и при этом

$$|K(x,y) - K''_N(x,y)| \leqslant c_N \left(\frac{|x-z|}{|y-z|}\right)^N.$$

Определение. Дальней зоной множества Y с центром $z \in Y$ называется множество

$$\mathcal{F}_q\,(Y,z) \equiv \{x:\ |x-z| \geqslant \frac{|y-z|}{q} \forall\ y \in Y\}.$$

Основная лемма. *Если* $y \in Y$ *и* $x \in \mathcal{F}_q\,(Y,z)$, *то*

$$|K(x,y) - K'_N(x,y)| \leqslant c_N q^N.$$

Если $x \in Y$ *и* $y \in \mathcal{F}_q\,(Y,z)$, *то*

$$|K(x,y) - K''_N(x,y)| \leqslant c_N q^N.$$

Следствие 1. Если в точках $y_j \in Y$ помещены заряды q_j, то поле от них

$$f(x) \equiv \sum_j K(x,y_j)q_j$$

в точке $x \in \mathcal{F}_q\,(Y,z)$ аппроксимируется *мультипольным разложением*

$$M(x) \equiv \sum_{l=0}^{N} \sum_{m \in \mathcal{M}_l} M_{lm} T_{lm}(x,z), \quad \text{где} \quad M_{lm} = \sum_j t_{lm}(y_j,z)q_j,$$

с оценкой погрешности

$$|f(x) - M(x)| \leqslant Q c_N q^N, \quad \text{где} \quad Q = \sum_j |q_j|.$$

Следствие 2. Если в точке $x \in Y$ вычисляется поле $f(x)$ от зарядов q_j в точках $y_j \in \mathcal{F}_q\,(Y,z)$, то оно аппроксимируется *локальным разложением*

$$L(x) \equiv \sum_{l=0}^{N} \sum_{m \in \mathcal{M}_l} L_{lm} t_{lm}(x,z), \quad \text{где} \quad L_{lm} = \sum_j T_{lm}(y_j,z)q_j,$$

с оценкой погрешности

$$|f(x) - L(x)| \leqslant Q c_N q^N, \quad \text{где} \quad Q = \sum_j |q_j|.$$

Аналогичные утверждения остаются в силе и в том случае, когда суммарный заряд Q распределен в некотором объеме или на части поверхности, принадлежащих области Y или $\mathcal{F}_q(Y, z)$.

Число N называется порядком мультипольного или локального разложения. Для многих приложений множество $\mathcal{M}_l$ содержит $2m+1$ индексов. В таких случаях разложение порядка N содержит $(N+1)^2$ членов.

3.2. ПРИМЕРЫ ЯДЕР И РАЗЛОЖЕНИЙ. Введем сферические координаты с центром в точке z. Для точек x и y будем писать r_x, θ_x, ϕ_x и r_y, θ_y, ϕ_y.

Пример 1. $K(x, y) = \frac{1}{|x-y|}$. Мультипольное и локальное разложения:

$$M(x) = \sum_{l=0}^{N} \sum_{m=-l}^{l} M_{lm} \frac{1}{r_x^{n+1}} Y_l^m(\theta_x, \phi_x),$$

$$L(x) = \sum_{l=0}^{N} \sum_{m=-l}^{l} L_{lm} r_x^n Y_l^m(\theta_x, \phi_x).$$

Пример 2. $K(x, y) = \frac{e^{ik|x-y|}}{|x-y|}$. Мультипольное и локальное разложения:

$$M(x) = \sum_{l=0}^{N} \sum_{m=-l}^{l} M_{lm}\, H_l^{(1)}(kr_x) Y_l^m(\theta_x, \phi_x),$$

$$L(x) = \sum_{l=0}^{N} \sum_{m=-l}^{l} L_{lm}\, J_l^{(1)}(kr_x) Y_l^m(\theta_x, \phi_x).$$

Они порождаются рядом

$$K(x, y) = 4\pi i k \sum_{l=0}^{\infty} \sum_{m=-l}^{l} J_l^{(1)}(kr_y) Y_l^{-m}(\theta_y, \phi_y)\, H_l^{(1)}(kr_x) Y_l^m(\theta_x, \phi_x).$$

3.3. ИЕРАРХИЯ КЛАСТЕРОВ И ТРАНСЛЯЦИОННЫЕ МАТРИЦЫ. В первых работах по мультипольным алгоритмам рассматривалась задача 1 (по проведенной выше классификации). Обобщения кажутся предметом очевидного развития. Но их все же необходимо делать — в интересах практики. Модифицированный мультипольный алгоритм для задачи 2 описан в [43]. В этой работе рассматривается конкретная задача электростатики. Однако алгоритм представлен на достаточно абстрактном (проблемно-независимом) уровне.

Все базисные функции (заряды) и точки наблюдения образуют один кластер, который разбивается на какое-то число меньших кластеров, те в свою очередь разбиваются на еще меньшие кластеры, и так далее. В результате возникает некоторая *иерархия кластеров* (ей соответствует дерево кластеров). Проще всего она порождается естественной иерархией вложенных кубов.

Куб уровня 0 — это куб, содержащий целиком распределенный заряд и все точки наблюдения. Для определенности пусть $\nu = 3$; тогда куб делится на 8 одинаковых кубов, принадлежащих уровню 1. И так далее. Каждый куб (d, j) в этой иерархии идентифицируется номером уровня d и своим индексом j среди кубов того же уровня. Обозначим через $J(d)$ множество индексов для кубов уровня d. Пусть уровень D является последним. Приведем список основных определений [43]:

$F(d, j)$	– индекс куба уровня $d - 1$, породившего куб (d, j).
$C(d, j)$	– множество индексов кубов уровня $d + 1$, порожденных кубом (d, j).
$N(d, j)$	– множество индексов кубов уровня d, которые являются соседями (имеют общую сторону или вершину) куба (d, j) или же *вторичными соседями*, то есть соседями его соседей $(j \in N(d, j))$.
$I(d, j)$	– так называемое *множество взаимодействия* для куба (d, j): каждый входящий в него куб является либо вторичным соседом для куба $(d-1, F(d, j))$, либо кубом того же уровня, возникающим при разбиении какого-либо соседа куба $(d - 1, F(d, j))$, но не принадлежащим множеству $N(d, j)$.

$q(d,j)$	– вектор коэффициентов разложения для тех базисных функций, носители которых имеют пересечение с кубом (d,j).
$p(d,j)$	– вектор значений поля в точках наблюдения, попавших в куб (d,j) (точки общей границы двух кубов одного уровня приписываются *к одному* из них).
$M(d,j)$	– вектор коэффициентов мультипольного разложения для заряда, распределенного в кубе (d,j).
$L(d,j)$	– вектор коэффициентов локального разложения для поля в кубе (d,j).

На предварительном этапе вычисляются *трансляционные матрицы*:

$$\begin{array}{lcccll}
ML\,(d,j,d',j') & : & L(d,j) & = & ML\,(d,j,d',j') & M(d',j');\\
LL\,(d,j,d',j') & : & L(d,j) & = & LL\,(d,j,d',j') & L(d',j');\\
MM\,(d,j,d',j') & : & M(d,j) & = & MM\,(d,j,d',j') & M(d',j');\\
QM\,(d,j,d',j') & : & M(d,j) & = & QM\,(d,j,d',j') & q(d',j');\\
LP\,(d,j,d',j') & : & p(d,j) & = & LP\,(d,j,d',j') & L(d',j');\\
QP\,(d,j,d',j') & : & p(d,j) & = & LP\,(d,j,d',j') & q(d',j').
\end{array}$$

Важно, конечно, что они вычисляются не для всех возможных пар (d,j) и (d',j'). Общее число трансляционных матриц будет порядка общего числа блоков.

3.4. МОДИФИЦИРОВАННЫЙ МУЛЬТИПОЛЬНЫЙ АЛГОРИТМ.

Ниже приводится модификация [43] быстрого мультипольного алгоритма [24].

1. *Вычисление мультипольных разложений для кубов нижнего уровня*:

$$M(D,j) = QM\,(D,j,D,j)q(D,j) \quad \forall\, j \in J(D).$$

2. *Вычисление мультипольных разложений для всех кубов*: для каждого d от $D-1$ до 2 вычислить

$$M(d,j) = \sum_{j \in C(d,j)} MM\,(d,j,d+1,j')M(d+1,j')\ \forall\, j \in J(d).$$

3. *Вычисление частичных локальных разложений поля (от заряда, распределенного в зоне взаимодействия каждого куба)*: для каждого d от 2 до D вычислить

$$L(d,j) = \sum_{(d',j')\in I(d,j)} ML\,(d,j,d',j')M(d',j')\;\forall\, j \in J(d).$$

4. *Трансляция во вложенные кубы и накопление частичных локальных разложений*: для каждого d от 3 до D вычислить

$$L(d,j) = L(d,j) + LL\,(d,j,d-1,F(d,j))L(d-1,F(d,j))\;\forall\, j \in J(d).$$

5. *Вычисление искомых значения поля*: $\forall\, j \in J(D)$

$$p(D,j) = LP\,(D,j)L(D,j) + \sum_{j'\in N(D,j)} QP\,(D,j,D,j')q(D,j).$$

Общее число операций в этом алгоритме пропорционально числу M самых мелких кубов. Но коэффициент пропорциональности не очень маленький. Как показано в [43], число умножений на матрицы MM и QM (LL и LP) не превышает $2M$. В то же время, число умножений на матрицы ML (число операций взаимодействия) не превышает $1998M$. Заметим, что умножение на матрицу MM, LL или ML требует порядка N^4 арифметических операций, где N — параметр, отвечающий за точность мультипольных и локальных разложений. При более-менее очевидных предположениях можно утверждать, что число операций при „мультипольном“ умножении матрицы на вектор зависит от размеров матрицы линейно.

Вероятно, версии мультипольного алгоритма с одним или двумя уровнями могут давать более оптимистические оценки арифметических затрат на каких-то практических задачах.

3.5. АДАПТИВНЫЙ МУЛЬТИПОЛЬНЫЙ АЛГОРИТМ. Во многих случаях заряд распределяется в исходном кубе крайне неравномерно (например, находится на какой-то кусочно-гладкой поверхности). В таких случаях в [16] предлагается не дробить те кубы, которые пересекают мало базисных носителей — меньше некоторого предписанного числа. Соответствующая адаптивная версия может иногда требовать большей работы, чем не адаптивная. Поэтому в

[43] предлагается более продвинутый подход — с отказом от построения „неэффективных“ мультипольных и локальных разложений и от построения таких разложений для любого куба, при разбиении которого лишь один куб *непустой*, то есть содержит часть заряда и точки наблюдения. Вот необходимые определения:

Адаптивный куб — любой непустой куб уровня D или же куб меньшего уровня, порождающий при разбиении не менее двух непустых подкубов.

Адаптивный подкуб — для куба (d, j) это вложенный в него адаптивный куб любого более высокого уровня, не являющийся частью какого-либо другого адаптивного куба, вложенного в исходный куб. Множество всех адаптивных подкубов куба (d, j) обозначается $C^A(d, j)$.

Адаптивный надкуб — для адаптивного куба (d, j) это единственный содержащий его адаптивный куб, для которого он является адаптивным подкубом. Обозначение: $F^A(d, j)$.

Адаптивное множество взаимодействия — это прежнее множество взаимодействия, из которого исключены пустые кубы, а каждый неадаптивный куб заменен своим адаптивным подкубом. Обозначение: $I^A(d, j)$.

Все ранее введенные множества модифицируются — из них выбрасываются пустые кубы. В множестве $N(d, j)$ любой неадаптивный куб заменяется его адаптивным подкубом. Ниже приводится алгоритм [43]:

1. Для всех непустых кубов (D, j):
   Если размер вектора $q(D, j)$ больше $(N+1)^2$, то
   $M(D, j) = QM\,(D, j, D, j) q(D, j)$.

2. Для каждого уровня d от $D-1$ до 2:
   Для всех адаптивных кубов (d, j):
   Если размер вектора $q(d, j)$ больше $(N+1)^2$, то
   Для всех кубов $(d', j') \in C^A(d, j)$:
   Если размер вектора $q(d', j')$ больше $(N+1)^2$, то
   $M(D, j) = M(d, j) + MM\,(d, j, d', j') M(d', j')$;
   если нет, то
   $M(D, j) = M(d, j) + QM\,(d, j, d', j') q(d', j')$.

3. Для каждого уровня d от 2 до D:
   Для всех кубов (d, j) таких, что $|I^A(d, j)| > 0$:
   Если $|C^A(d, j)| > 1$, то

$d' = d, \quad j' = j;$
если нет, то
(d', j') — единственный куб из $C^A(d, j)$.
Если размер вектора $p(d, j)$ больше $(N+1)^2$, то:
Для всех кубов $(d'', j'') \in I^A(d, j)$:
Если размер вектора $q(d'', j'')$ больше $(N+1)^2$, то
$L(d', j') = L(d', j') + ML\,(d', j', d'', j'')M(d'', j'');$
если нет, то
$L(d', j') = L(d', j') + QL\,(d', j', d'', j'')q(d'', j'').$
Если размер вектора $p(d, j)$ не больше $(N+1)^2$, то:
Для всех кубов $(d'', j'') \in I^A(d, j)$:
Если размер вектора $q(d'', j'')$ больше $(N+1)^2$, то
$p(d', j') = p(d', j') + MP\,(d', j', d'', j'')M(d'', j'');$
если нет, то
$p(d', j') = p(d', j') + QP\,(d', j', d'', j'')q(d'', j'').$

4. Для каждого уровня d от 3 до D:
Для всех адаптивных кубов (d, j):
Если размер вектора $p(d, j)$ больше $(N+1)^2$, то
$L(d, j) = LL\,(d, j, F^A(d, j))L_{F^A(d,j)};$
если нет и размер вектора $p_{F^A(d,j)}$ больше $(N+1)^2$, то
$p(d, j) = LP\,(d, j, F^A(d, j))L_{F^A(d,j)}.$

5. Для всех непустых кубов (D, j):
Если размер вектора $p(d, j)$ больше $(N+1)^2$, то
$p(D, j) = p(D, j) + \sum_{(d', j') \in N(d,j)} QP\,(D, j, d', j')q(d', j').$

Число операций в данном адаптивном алгоритме никогда не бывает больше, чем в исходном неадаптивном алгоритме.

3.6. „МЕДЛЕННЫЙ“ МУЛЬТИПОЛЬНЫЙ АЛГОРИТМ. Рассмотренные выше „быстрые“ мультипольные алгоритмы опирались на продвинутую иерархию кластеров. „Медленный“ мультипольный алгоритм использует один или два уровня этой иерархии. По мнению авторов работы [18], для матриц размера $\sim 10^5$ этого вполне достаточно.*

Предположим, что n базисных функций и n точек наблюдения разбиты на кластеры, принадлежащие областям $D_1, \ldots, D_M$. Для простоты будем считать, что число базисных функций и точек наблюдения для каждой области одно и то же и равно m

---

*„Лучшее — враг хорошего“.

($\Rightarrow \quad n = Mm$). Через $q(D_i)$ обозначаем вектор из коэффициентов разложения по базисным функциям области D_i, через $p(D_i)$ — вектор значений поля в точках наблюдения этой же области.

Для матрицы коэффициентов и векторов в данном случае возникают естественные блочные представления:

$$p \equiv \begin{bmatrix} p(D_1) \\ \dots \\ p(D_M) \end{bmatrix} = \begin{bmatrix} A_{11} & \dots & A_{1M} \\ \dots & \dots & \dots \\ A_{M1} & \dots & A_{MM} \end{bmatrix} \begin{bmatrix} p(D_1) \\ \dots \\ p(D_M) \end{bmatrix}.$$

В каждой области D_i фиксируем центр z_i мультипольных и локальных разложений. Фиксируем N — порядок разложения и некоторое $0 < q < 1$. Тогда для любой области D_i однозначно определяется ее дальняя зона $\mathcal{F}_q(D_i, z_i)$.

„Медленный“ мультипольный алгоритм:

1. Для всех областей D_i вычисляется мультипольное разложение порядка N.

2. Для всех пар (i, j) таких, что $D_j \subset \mathcal{F}_q(D_i, z_i)$, по мультипольному разложению для D_j вычисляется локальное разложение для D_i. Затем для каждой области формируется локальное разложение от всех областей ее дальней зоны (как сумма соответствующих локальных разложений).

3. Для всех областей D_i вычисляется $p(D_i)$ как сумма поля от дальней зоны (с использованием уже найденного локального разложения) и поля от ближней зоны (с помощью прямого умножения на соответствующие блоки матрицы A).

Фактически в данном алгоритме используется некоторое специальное приближенное представление матрицы A:

$$A \approx \tilde{A} = S + VTU.$$

Здесь $S = [S_{ij}]$ имеет блоки $S_{ij} = 0$, если $D_j \subset \mathcal{F}_q(D_i, z_i)$, и $S_{ij} = A_{ij}$ в противном случае. Матрица U имеет n столбцов и $(N+1)^2M$ строк. Матрица V имеет n строк и $(N+1)^2M$ столбцов. Квадратная матрица T отвечает конвертированию коэффициентов мультипольных разложений в коэффициенты локальных разложений с последующим суммированием для каждой отдельной области.

Если ввести подкластеры (второй уровень иерархии), то S заменяется аналогичным приближенным представлением. И так далее.

Теорема. *Для „медленного“ мультипольного алгоритма*

$$\|Aq - \tilde{A}q\|_\infty \leqslant c_N q^N \|Wq\|_1,$$

где W *— диагональная матрица* L_1*-норм базисных функций:*

$$W = \operatorname{diag}\{\|v_j\|_{L_1}\}_{j=1}^n.$$

По-видимому, для „быстрых“ мультипольных алгоритмов оценка погрешности останется такой же.

Предположим, что для любой области D_i в ее ближней зоне находится *конечное* число областей. Тогда очевидно, что если $M \sim \sqrt{n}$, то арифметические затраты имеют вид $\mathcal{O}\left(n^{\frac{3}{2}}\right)$.

Вероятно, в ближней зоне каждой области может находиться $\sim q^{-\nu}$ областей. Если ε — предписанная точность и $c_N \sim 1$, то q нужно выбрать так, чтобы $q^N \sim \varepsilon$. Поэтому

$$\operatorname{Work}(\varepsilon, n) = \mathcal{O}\left(\varepsilon^{-\frac{\nu}{N}} n^{\frac{3}{2}} + (N+1)^4 n\right).$$

Можно, конечно, зафиксировать q и взять $N \sim \log \varepsilon^{-1}$. Тогда

$$\operatorname{Work}(\varepsilon, n) = \mathcal{O}\left(n^{\frac{3}{2}} + \log^4 \varepsilon^{-1} n\right).$$

Все бы хорошо, но очень огорчает присутствие коэффициента $(N+1)^4$. Если $N \sim 10$, то в „медленном“ алгоритме мало смысла даже при $n \sim 10\,000$. Наверное, для таких „умеренных“ n нужно брать $N \sim 3$. Если $\nu = 3$, то законное желание иметь хотя бы два десятичных знака вроде бы делает алгоритм бесполезным?

4. ИНТЕРПОЛЯЦИОННЫЙ МЕТОД

4.1. МЕТОД РЕГУЛЯРНОЙ СЕТКИ. Если функция $f(x, y)$ достаточно гладкая при $x \in \Omega_x \subset \mathbb{R}^\nu$, $y \in \Omega_y \subset \mathbb{R}^\nu$, то значения $f(x_i, y_j)$ приближаются с помощью интерполяционных формул, построенных по значениям $f(s_k, t_l)$ той же функции на другой сетке:

$$f(x_i, y_j) \approx \sum_{kl} \alpha_{kl} f(s_k, t_l).$$

Потребуем, чтобы $\alpha_{kl} = c_k d_l$. Тогда интерполяционная формула получает матричную интерпретацию

$$A \approx \tilde{A} = S + VTU,$$

где $A = [f(x_i, y_j)]$ и $T = [f(s_k, t_l)]$.

Матрицы V и U содержат интерполяционные коэффициенты c_k и d_l, соответственно. Матрица S нужна для того, чтобы *исправить плохо приближенные значения.* Мы надеемся на то, что S будет разреженной матрицей.

Матрицы V и U, конечно, разреженные. При умножении V на вектор-столбец происходит обычная интерполяция. То, что происходит при умножении U на вектор-столбец, Брандт называет *антерполяцией* (adjoint interpolation).

Предположим, что $f(x, y) = F(|x - y|)$. Тогда в случае регулярной сетки и отвечающей ей *регулярной нумерации* узлов интерполяции матрица T оказывается *многоуровневой теплицевой матрицей.* Ради этого-то все и делалось: для умножения на T известны очень эффективные алгоритмы, использующие быстрое преобразование Фурье.

Если $f(x, y)$ имеет какой-то другой вид, то T может обладать какой-то другой полезной спецификой. Например, преобразование Фурье на нерегулярной сетке можно сводить к преобразованию Фурье на регулярной сетке [4,11].

4.2. ОСТАТОЧНЫЙ ЧЛЕН. Остаточный член одномерной интерполяционной формулы общеизвестен. Рассмотрим многомерное обобщение (двумерное см. в [4]).

Пусть $f(x) = f(x_1, \ldots, x_m)$ — функция от m скалярных компонент вектора x и $p(x) = p(x_1, \ldots, x_m)$ — интерполяционный многочлен для сетки

$$\mathcal{M} = \{x_1^{i_1}\}_{i_1=1}^{p_1} \times \ldots \{x_m^{i_m}\}_{i_m=1}^{p_m}.$$

Положим

$$D_{i_1,\ldots,i_k}(x) \equiv \frac{1}{p_{i_1}! \ldots p_{i_k}!} \frac{\partial^{p_{i_1}} \ldots \partial^{p_{i_k}}}{(\partial x_{i_1})^{p_{i_1}} \ldots (\partial x_{i_k})^{p_{i_k}}} f(x),$$

$$\Omega_{i_1,\ldots,i_k} \equiv \omega_{i_1}(x_{i_1})\ldots\omega_{i_k}(x_{i_k}), \quad \omega_i(t) = \prod_{j=1}^{p_i}(t - x_i^j).$$

Теорема. *Если* f *имеет непрерывные частные производные порядка* $p_1 + \cdots + p_m$, *то*

$$f(x) - p(x) = \sum_{k=1}^{m}(-1)^{(k-1)} \sum_{1\leqslant i_1<\cdots<i_k\leqslant m} E_{i_1,\ldots,i_k},$$

где

$$E_{i_1,\ldots,i_k} = D_{i_1,\ldots,i_k}(\xi_{i_1,\ldots,i_k})\ \Omega_{i_1,\ldots,i_k}.$$

Все точки $\xi_{i_1,\ldots,i_k}$ *принадлежат выпуклой оболочке, натянутой на* $\mathcal{M}$ *и* x.

Доказательство. При $m = 1$ мы получаем классическое представление остаточного члена одномерной лагранжевой интерполяции. Далее по индукции. Чтобы избежать сложной индексации, положим

$$z = (x_1, \ldots, x_{m-1}), \quad i = (i_1, \ldots, i_{m-1}), \quad z_i = (x_1^{i_1}, \ldots, x_{m-1}^{i_{m-1}});$$

$$y = x_m, \quad j = i_m, \quad y_j = x_m^j;$$

$$l_j(y) = \prod_{k=1,\ k\neq j}^{p_m} \frac{y - y_k}{y_j - y_k}, \quad \omega(y) = \prod_{j=1}^{p_m}(y - y_j);$$

$$L_i(z) = \prod_{t=1}^{m-1} \prod_{\substack{k_t=1\\ k_t\neq i_t}}^{p_t} \frac{x_t - x_t^{k_t}}{x_t^{i_t} - x_t^{k_t}}.$$

Тогда при некотором значении η

$$f(z,y) - p(z,y) = R + D_m(z,\eta)\ \omega(y),$$

где

$$R = \sum_{j=1}^{p_m} \left(f(z,y_j) - p(z,y_j)\right)\ l_j(y).$$

Согласно индуктивному предположению,

$$R = \sum_{i} \left(\sum_{j=1}^{p_m}(f(z_i,y_j) - p(z_i,y_j)l_j(y) \right)\ L_i(z) + E,$$

$$E = \sum_{k=1}^{m-1} (-1)^{k-1} \sum_{1 \leqslant i_1 < \dots < i_k \leqslant m-1} \sum_{j=1}^{p_m} D_{i_1,\dots,i_k}(\xi_{i_1,\dots,i_k}, y_j) l_j(y)\, \Omega_{i_1,\dots,i_k}.$$

Остается заметить, что при некотором ζ

$$\sum_{j=1}^{p_m} D_{i_1,\dots,i_k}(\xi_{i_1,\dots,i_k}, y_j) l_j(y) =$$

$$= D_{i_1,\dots,i_k}(\xi_{i_1,\dots,i_k}, y) - D_{i_1,\dots,i_k,m}(\xi_{i_1,\dots,\xi_k}, \zeta)\, \omega(y).$$

Теорема доказана. ■

Далее всегда предполагается, что точка x принадлежит выпуклой оболочке, натянутой на $\mathcal{M}$.

Следствие. Предположим, $p_1 = \dots = p_m \equiv N$ и сетка $\mathcal{M}$ является равномерной по каждому направлению с шагом $0 < h \leqslant 1$. Тогда если все частные производные любого порядка $k \leqslant mp$ ограничены по модулю константой C_k, то

$$|f(x) - p(x)| \leqslant C_N\, m\, h^N + C_{mN}\, (2^m - m)\, h^{2N}.$$

4.3. АСИМПТОТИЧЕСКИ ГЛАДКИЕ ФУНКЦИИ. Следуя Брандту, будем называть $f(x,y)$, $x, y \in \mathbb{R}^\nu$, *асимптотически гладкой* функцией, если для любого p

$$|\partial^p f(x,y)| \leqslant c_p |x-y|^{g-p},$$

где ∂^p обозначает любую смешанную производную порядка p и g не зависит от p. Однако при получении оценок нужно знать, как c_p может расти при увеличении p. Вот разумное (довольно общее) предположение [52]:

$$c_p \leqslant c\ d^p\, p!\, ,\ c, d > 0.$$

Например, если $f(x,y) = \frac{1}{|x-y|}$, то $g = -1$. „Прямое вычисление“ может дать оценку $c_p \leqslant 4^p\, p!$ [52]. Менее прямой путь в данном случае может дать более тонкую оценку* (если опереться на то, что f является гармонической функцией).

---

*С. А. Горейнов. Настоящий сборник.

Фиксируем $0 < q < 1$ и рассмотрим *зону дальнодействия*

$$\mathcal{F} = \{(x, y) : \quad |x - y| \geqslant q^{-1} h\}.$$

Теорема. *Предположим, что значение асимптотически гладкой функции $f(x, y)$ интерполируется многочленом $p(x, y)$ по узлам $\{x_i\}$ и $\{y_j\}$ равномерной сетки с шагом $0 < h \leqslant 1$. При этом x и y принадлежат выпуклым оболочкам X и Y , натянутым на соответствующие узлы, и $X \times Y \subset \mathcal{F}$. Пусть N — число интерполяционных узлов по каждому направлению. Тогда*

$$|f(x, y) - p(x, y)| \leqslant \gamma_N q^{N-g}.$$

Используя технику из [52], можно показать, что любая асимптотически гладкая функция обладает мультипольными разложениями.

4.4. СКОЛЬКО ЭТО СТОИТ? Итак, умножение на A заменяется умножением на $\tilde{A} = S + VTU$, где T — матрица регулярной сетки. По аналогии с предыдущими рассмотрениями, это — „медленный“ метод.

Пусть A — $n \times n$ и T — $n_T \times n_T$. Если T — многоуровневая теплицева матрица, то для хранения ее элементов нужно $\mathcal{O}(n_T)$ ячеек, а для умножения ее на вектор требуется $\mathcal{O}(n_T \log n_T)$ операций.

Чтобы запомнить интерполяционные коэффициенты, достаточно νN ячеек. Матрица U имеет N^ν ненулевых элементов в каждом столбце, матрица V имеет столько же ненулевых элементов в каждой строке. Число ненулей в матрице S, вероятно, должно быть порядка $q^{-\nu} n$. Таким образом, для метода регулярной сетки получаем

$$\mathrm{Work}\,(q, N, n) = \mathcal{O}\,\left((q^{-\nu} + N^\nu) n + n_T \log n_T\right).$$

Теорема. *Для интерполяционного метода регулярной сетки*

$$\|Au - \tilde{A}u\|_\infty \leqslant \gamma_N q^{N-g} \|u\|_1.$$

Если q фиксировано и $\varepsilon \sim q^N$, то $N \sim \log \varepsilon^{-1}$. Поэтому

$$\mathrm{Work}\,(\varepsilon, n) = \mathcal{O}\,(\log^\nu \varepsilon^{-1} n + n_T \log n_T).$$

При интегрировании по объему $n_T \sim n$. Вроде бы асимптотически Work $\sim n \log n$. Но в действительности не этот член главный! Скорее всего, $\varepsilon \sim n^{-\theta}$ для какого-то $\theta > 0$. Следовательно, Work $\sim n \log^{\nu} n$.

При интегрировании по поверхности $n_T \sim n^{\frac{\nu}{\nu-1}}$. В случае $\nu = 3$ при интегрировании по поверхности „медленный“ метод требует $\mathcal{O}(n^{\frac{3}{2}})$ операций и, возможно, не так уж плох.

В случае $\nu = 2$ вряд ли стоит применять „медленный“ метод при интегрировании по *произвольному* контуру. Тем не менее, путем интерполяции можно попытаться перейти на более грубую сетку. Если сделать это несколько раз, возникнет иерархия сеток („медленный“ алгоритм превратится в „быстрый“). В итоге можно уменьшить n_T или же вообще отказаться от теплицевости.

4.5. ПОЧТИ РЕГУЛЯРНОСТЬ. Предположим, что $n \times n$-матрица A отвечает не совсем уж произвольной сетке: пусть она отличается от регулярной лишь относительно малым числом $(o(n))$ узлов.

Матрица A оказывается *почти теплицевой*:

$$A = \begin{bmatrix} T & * \\ * & H \end{bmatrix}, \ \ T — (n - o(n)) \times (n - o(n)), \ \ H — o(n) \times o(n),$$

где T является (многоуровневой) теплицевой подматрицей.

В типичном случае $o(n) \sim \sqrt{n}$. Очевидно, для умножения достаточно $\sim n^{\frac{3}{2}}$ операций. То, что нас не устраивает, — это *память* для окаймления: $\sim n^{\frac{3}{2}}$.

Метод регулярной сетки для почти регулярной матрицы позволяет получить время $\sim n \log n$ *и память* $\sim n$.

В данном случае почти все столбцы матрицы U и почти все строки матрицы V будут иметь 1 в какой-то позиции и 0 во всех остальных. Поэтому „настоящих“ ненулей в матрицах V и U будет $\sim N^{\nu} o(n)$.

4.6. ВАРИАЦИОННАЯ ВЕРСИЯ. Помимо интерполяции, существуют, конечно, другие способы релегирования на регулярную сетку.

Задача „далеко-близко“. Пусть $y_j \in Y$. Для заданного $\varepsilon > 0$ найти коэффициенты γ_j, $1 \leqslant j \leqslant r$, такие, что

$$|G(x, y) - \sum_{j=1}^{r} \gamma_j G(x, y_j)| \leqslant \varepsilon \quad \forall\, x \in \mathcal{F}(Y, z).$$

Задача „близко-далеко“. Пусть $x_i \in Y$. Для заданного $\varepsilon > 0$ найти коэффициенты γ_i, $1 \leqslant i \leqslant r$, такие, что

$$|G(x,y) - \sum_{i=1}^{r} \gamma_i G(x_i, y)| \leqslant \varepsilon \quad \forall\, y \in \mathcal{F}(Y, z).$$

Чтобы решить задачу „далеко-близко“, построим мультипольное разложение

$$G(x,y) \approx G'(x,y) = \sum_{l=0}^{N} \sum_{m \in \mathcal{M}(l)} t_{lm}(y,z) T_{lm}(x,z)$$

и попытаемся выбрать γ_j из следующих уравнений:

$$\sum_{j=1}^{r} \gamma_j\, t_{lm}(y_j, z) = t_{lm}(y,z), \qquad 0 \leqslant l \leqslant N,\ m \in \mathcal{M}(l). \tag{1}$$

Чтобы решить задачу „близко-далеко“, ищем γ_i из уравнений

$$\sum_{i=1}^{r} \gamma_i\, T_{lm}(x_i, z) = T_{lm}(x,z), \qquad 0 \leqslant l \leqslant N,\ m \in \mathcal{M}(l). \tag{2}$$

Лемма 1. *Если система* (1) *или* (2) *имеет решение, то для задачи „далеко-близко“ или „близко-далеко“, соответственно, справедлива оценка*

$$|G(x,y) - \sum_{j=1}^{r} \gamma_j G(x, y_j)| \leqslant \Gamma_r\, c_N q^N, \quad \Gamma_r = \sum_{j=1}^{r} |\gamma_j|.$$

Теорема 1. *Если разрешимы системы* (1) *и* (2), *то для мультипольной версии метода регулярной сетки*

$$\|Ax - \tilde{A}x\|_\infty \leqslant (\Gamma_r + 1)\Gamma_r\, c_N q^N\, \|x\|_1.$$

Разрешимость систем (1) и (2) — это тонкий вопрос. Но даже если они разрешимы, они, скорее всего, плохо обусловлены, и, следовательно, величина Γ_r может оказаться чудовищно большой. Вероятно, нужно искать решения систем в смысле метода наименьших квадратов и при этом применять какую-то регуляризацию.

Лемма 2. *Пусть s-норма невязки системы* (1) (*или* (2)) *не больше δ. Тогда для задачи „далеко-близко“*

$$|G(x,y) - \sum_{j=1}^{r} \gamma_j G(x,y_j)| \leqslant \Gamma_r c_N q^N + \delta T_N^{(s')}, \qquad \frac{1}{s} + \frac{1}{s'} = 1,$$

где

$$T_N^{(s')} = \left(\sum_l \sum_m |T_{lm}(x,z)|^{s'} \right)^{\frac{1}{s'}}.$$

Для задачи „близко-далеко“ оценка имеет аналогичный вид, но всюду T нужно заменить на t.

Теорема 2. *В условиях и обозначениях леммы 2 для мультипольной версии метода регулярной сетки получаем*

$$\|Ax - \tilde{A}x\|_\infty \leqslant (\Gamma_r + 1)(\Gamma_r c_N q^N + \delta\, T_N)\|x\|_1,$$

где T_N — максимум из $T_N^{(s')}$ и $t_N^{(s')}$.

В принципе, можно пытаться решить задачи „далеко-близко“ и „близко-далеко“, не прибегая к мультипольным разложениям. Коэффициенты γ_j можно получить методом наименьших квадратов для любой *разумной* выборки точек в дальней зоне. (Идея кажется авантюрной, но такой же представлялась когда-то возможность аппроксимации блока скелетонами с использованием лишь малой части его элементов.)

4.7. „БЫСТРЫЕ“ АЛГОРИТМЫ. Понятно, что можно использовать иерархию сеток и строить соответствующие „быстрые“ алгоритмы. Отметим два возможных направления:

- Можно переходить к более мелкой сетке для того, чтобы умножать на S.

- Можно переходить к более крупной сетке, чтобы умножать на T. Для достаточно гладких ядер задача для более крупной сетки потребует меньших вычислительных затрат. В итоге может потребоваться лишь $\mathcal{O}(n)$ операций. Однако в случае сильно осциллирующих ядер необходимо как-то бороться с осцилляциями. Хорошая идея — каким-то образом включить их в вектор u. Как это можно сделать в многомерном случае — показано в [11]. В итоге на каждом шаге иерархии работа одна и та же. Общее число операций приобретает вид $\mathcal{O}(n \log n)$.

5. МЕТОД ЛОКАЛЬНЫХ ВОЛН

5.1. ТЕРМИНОЛОГИЯ. В течение последних 7–10 лет слово „wavelet“ стало одним из самых модных в западной литературе по вычислениям и обработке данных. В буквальном переводе, „wavelet“ — это „всплеск“ или „маленькая волна“. В отечественной литературе этому объекту уделялось мало внимания. Поэтому до сих пор нет устоявшегося адекватного термина. Можно встретить и „всплески“, и „вейвлеты“; мы будем называть их „локальными волнами“. Локальная волна должна обладать *двойным* свойством локальности: одновременно в „пространственной“ и в „частотной“ области. Еще один важный термин: „multiresolution analysis“. Возможные варианты перевода: „многоуровневый анализ“ или „мультичастотный анализ“. Более точным, вероятно, будет термин „иерархический анализ“.

Исследования по локальным волнам являются лишь некоторой (небольшой) частью иерархического анализа, органически связанного с многосеточными и многоуровневыми методами. Локальные волны наиболее успешно применяются в задачах обработки изображений и в схожих задачах, связанных с обработкой и хранением данных. Постепенно определяется и место локальных волн в численных методах.

Обилие сообщений о применении локальных волн в традиционных вычислительных задачах — это, скорее, дань моде. Пока не очень ясно, должны ли они претендовать на роль новых суперэффективных базисных функций в проекционных методах. Однако уже сейчас очевидно, что локальные волны могут успешно использоваться при построении адаптивных процедур (и, конечно, для сжатия числовых полей). Правильное их использование может привести к новым, исключительно эффективным численным методам.

5.2. ИЕРАРХИЧЕСКИЙ АНАЛИЗ. Приведем систему основных понятий иерархического анализа (см., например, [29]). Пусть F — функциональное пространство и V^k — векторное пространство размерности N_k.

Определение. Последовательность линейных операторов $\{D_k\}$, $k = 0, 1, \dots$, называется *последовательностью вложенных дискретизаций*, если

(1) $D_k : F \to V_k, \quad \mathrm{im}\, D_k = V^k$;
(2) $D_k f = 0 \quad \Rightarrow \quad D_{k-1} f = 0$.

С любой последовательностью вложенных дискретизаций ассоциируется *оператор сужения* (децимации)

$$D_k^{k-1}: \ V^k \to V^{k-1},$$

который однозначно определяется по следующему правилу:

$$D_k^{k-1} v = D_{k-1} f, \ \text{где}\ v = D_k f.$$

Обозначим через R_k какой-либо из правых обратных операторов для D_k:

$$R_k: \ V^k \to F, \quad D_k R_k = I_k$$

(I_k — единичный оператор в пространстве V^k). Будем называть R_k *оператором восстановления*.

Как только фиксирована какая-либо последовательность операторов восстановления, по ней однозначно определяются *операторы расширения* (предсказания, пролонгации):

$$P_{k-1}^k : V^{k-1} \to V^k, \quad P_{k-1}^k \equiv D_k R_{k-1}.$$

Теперь есть все, чтобы определить *иерархическую форму* вектора $v^k \in V^k$. Положим

$$v_{k-1} = D_k^{k-1} v_k$$

и запишем

$$v_k = P_{k-1}^k v_{k-1} + d_k, \quad \text{где} \quad e_k = (I_k - P_{k-1}^k D_k^{k-1}) v_k.$$

Соответствие $v_k \leftrightarrow \{d_k, e_{k-1}\}$ является взаимно-однозначным. По вектору v_{k-1} можно получить e_{k-1} и v_{k-2}, и так далее. В результате возникает взаимно-однозначное соответствие

$$v_k \leftrightarrow \hat{v}_k \equiv \{e_k,\ e_{k-1},\ \dots\ ,\ e_1,\ v_0\}.$$

Полученный таким образом кортеж векторов $\hat{v}_k$ называется *иерархической формой* вектора v_k.

Вектор v_k определяется N_k коэффициентами, а его иерархическая форма содержит $N_k + \ldots + N_0$ коэффициентов. Поэтому пока не очень ясно, зачем переходить к иерархической форме.

Прежде всего заметим, что $e_k \in \ker D_k^{k-1}$. Поэтому если выбрать какой-либо базис в $\ker D_k^{k-1}$, то вектор d_k, составленный из коэффициентов разложения e_k по этому базису, содержит лишь $N_k - N_{k-1}$ компонент. Будем писать $d_k = G_k e_k$, где G_k — оператор разложения по базису. Кортеж $\{d_k, \ldots, d_1, v_0\}$ будем называть *иерархическим представлением* вектора v_k.

В иерархическом представлении вектора столько же коэффициентов, сколько и в его исходном представлении. Однако если в исходном представлении все коэффициенты могут быть примерно одной величины, во многих случаях в иерархическом представлении относительно большой оказывается лишь малая их часть. Поэтому появляется возможность пренебречь остальными коэффициентами иерархического представления.

Нетрудно понять, почему это происходит. Согласно определению, вектор e_k есть ошибка аппроксимации вектора v_k вектором $P_{k-1}^k v_{k-1}$. Предположим, для примера, что компоненты v_k суть значения функции f в N_k узлах некоторой сетки на отрезке, а v_{k-1} суть значения той же функции в каких-то $N_{k-1} < N_k$ узлах той же самой сетки. В данном случае оператор восстановления можно построить с помощью любого интерполяционного правила. Тогда e_k будет иметь нулевые компоненты, отвечающие N_{k-1} узлам грубой сетки, а $N_k - N_{k-1}$ ненулевых компонент представляют собой ошибки интерполяции в соответствующих узлах. Эти ошибки будут малыми там, где функция обладает достаточной гладкостью. Относительно большие коэффициенты будут соответствовать „особым точкам“ функции.

Таких точек обычно бывает не очень много. Иерархическое представление ценно, во-первых, тем, что позволяет обнаружить такие точки. Во-вторых, оно позволяет оставить для хранения и дальнейшей обработки лишь „существенную“ информацию, или, другими словами, „сжать“ исходную информацию без заметной потери точности.

Иерархический анализ — это способ построения и перехода от исходных к иерархическим представлениям. Для заданной последовательности вложенных дискретизаций различные схемы иерархического анализа получаются путем выбора операторов восстановления и базисов в ядрах операторов сужения (децимации).

5.3. ЛОКАЛЬНЫЕ ВОЛНЫ. С точки зрения общей схемы иерархического анализа, локальные волны k-го уровня — это элементы базиса в ядре k-го оператора сужения. Однако, термин „wavelets“ (то есть „настоящие“ локальные волны) употребляется лишь в тех случаях, когда последовательность вложенных дискретизаций связана с равномерными регулярно-вложенными сетками.

Опишем некоторую достаточно общую схему [29,17], приводящую к биортогональным локальным волнам с компактным носителем. Рассмотрим равномерную сетку с узлами

$$x_i^k = ih_k, \ \ -\infty < i < \infty; \quad h_k = 2^{-k}h_0,$$

и будем считать, что V^k есть пространство функций, определенных на этой сетке. Пусть $f \in L^2(\mathbb{R})$ и оператор D_k определяется следующим образом:

$$(D_k f)_i = \frac{1}{h_k}\int\limits_{-\infty}^{\infty} f(x) w\left(\frac{x - x_i^k}{h_k}\right) dx, \quad -\infty < i < \infty,$$

где весовая функция $w(x)$ такова, что

$$w(x) \in L^2(\mathbb{R}), \qquad \int\limits_{-\infty}^{\infty} w(x) dx = 1.$$

Очевидно, мы получим последовательность вложенных дискретизаций, если для некоторых вещественных чисел $\alpha_0, \ldots, \alpha_m$

$$(D_{k-1}f)_i = \sum_{j=0}^{m} \alpha_j (D_k f)_{2i-j} \quad \forall\, i.$$

Учитывая определение операторов D_k, получаем

$$0 = \int\limits_{-\infty}^{\infty} \left(\frac{1}{h_{k-1}} w\left(\frac{x - x_i^{k-1}}{h_{k-1}}\right) - \frac{1}{h_k} \sum_{j=0}^{m} \alpha_j w\left(\frac{x - x_{2i-j}^k}{h_k}\right)\right) f(x) dx.$$

Следовательно, весовая функция $w(x)$ должна подчиняться уравнению

$$w(x) = 2\sum_{j=0}^{m} \alpha_j w(2x-j). \qquad (*)$$

Это так называемое *уравнение дилатации.*

В работах [19,17] показано, что при условиях

$$\sum_{j=0}^{m} \alpha_j = 1,$$

$$\sum_j \alpha_{2j} = \sum_j \alpha_{2j-1}$$

уравнение $(*)$ имеет решение $w(x) \in L^2(\mathbb{R})$ с носителем на отрезке длины m. Решение определено однозначно с точностью до сдвига. Поэтому если фиксировать его носитель — пусть это будет отрезок $[-m, 0]$, то решение единственно.

Соответствующий оператор сужения будет иметь вид

$$(D_k^{k-1}v)_i = \sum_j \alpha_j v_{2i-j} = \sum_l \alpha_{2i-l} v_l.$$

Ядро этого оператора одномерно, и ему принадлежит функция

$$(\mu_j^k)_i = (-1)^{i+1}\alpha_{2j-i-1}, \quad -\infty < i < \infty.$$

Чтобы построить оператор восстановления для D_k, рассмотрим дискретную функцию $\{\beta_l\}$ с конечным носителем, такую, что

$$\sum_l \beta_{2l} = \sum_l \beta_{2l-1} = 1, \quad \sum_l \alpha_l \beta_{l+2j} = \delta_{j,0},$$

и положим

$$(R_k v)(x) = \sum_i v_i \phi\left(\frac{x - x_i^k}{h_k}\right),$$

где $\phi(x)$ удовлетворяет уравнению дилатации

$$\phi(x) = \sum_l \beta_l \phi(2x - l)$$

и нормировано так, что

$$\int \phi(x)w(x)dx = 1.$$

Соответствующий оператор расширения $P^k_{k-1} = D_k R_{k-1}$ имеет вид

$$(P^k_{k-1}v)_i = \sum_j \beta_{i-2j} v_j.$$

При дополнительном условии $\beta_j = 2\alpha_j$ полученные выше биортогональные локальные волны превращаются в ортогональные D-волны (локальные волны Добешис (Daubechies)).

Краткое (но очень полезное!) введение в теорию локальных волн с обсуждением их аппроксимационных свойств и условий ортогональности можно найти в работе [48].

5.4. ПРИМЕНЕНИЯ К РЕШЕНИЮ ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ. Вполне естественным образом локальные волны применяются для построения быстрых алгоритмов реализации дискретных аналогов интегральных преобразований [10,30,12].

В этом качестве они могут весьма успешно использоваться при решении интегральных уравнений. Возникающие при дискретизации матрицы коэффициентов являются плотными, но после применения процедур иерархического анализа их удается преобразовать к таким матрицам, которые хорошо аппроксимируются разреженными матрицами. В ряде работ обсуждаются подходы, в которых для получения разреженных аппроксимаций не требуется иметь все элементы исходной плотной матрицы.

При решении алгебраических систем обычно используются итерационные методы, в которых главные затраты на каждой итерации приходятся на операцию умножения матрицы на вектор. Локальные волны дают возможность ускорить эту операцию. Однако во многих случаях само число итераций оказывается большим. Чтобы его уменьшить, требуются эффективные процедуры предобусловливания. Вероятно, с помощью локальных волн можно что-то сделать и в этой области.

5.5. ПРИМЕНЕНИЯ К РЕШЕНИЮ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ. Этот раздел можно считать отступлением от генеральной темы. При решении дифференциальных уравнений локальные волны позволяют, вообще говоря, „сжать“ вычисленные числовые поля. Однако их можно использовать и в самом процессе вычислений.

Рассмотрим эволюционное уравнение (см. [29,21])

$$\begin{aligned}&\partial_t u + L(x, \partial_x)u = f(x),\ x \in \Omega,\ t > 0;\\&u(x, 0) = u_0(x)\end{aligned}$$

с нулевыми граничными условиями.

Явная схема имеет вид

$$v^{n+1} = Av^n + g, \tag{3}$$

где $v_j^n \approx u(x_j, t_n)$, $g_j \approx \Delta t f(x_j); t_n = n\Delta t; \{x_j\}$ — некоторая сетка на Ω. Обозначим через W оператор перехода от вектора v^n к его иерархическому представлению $\hat{v}^n$. Тогда от (3) можно перейти к схеме „в частотной области“:

$$\hat{v}^{n+1} = \hat{A}\hat{v}^n + \hat{g}, \tag{4}$$

где

$$\hat{A} = WAW^{-1},$$
$$\hat{v}^{n+1} = Wv^{n+1}, \quad \hat{v}^n = Wv^n, \quad \hat{g} = Wg.$$

В работе [21] показано, что для одномерной гиперболической задачи сложность вычисления решения на $\sim N$-м временном слое (при использовании N узлов по x) от $O(N^2)$ может быть снижена до $O(N \log^3 N)$ (для одной и той же точности). Соответствующий алгоритм иерархического анализа основан на ортогональных D-волнах.

Если для простейшего уравнения переноса

$$u_t + u_x = 0$$

используется схема Лакса-Вендроффа

$$v_i^{n+1} = v_i^n - \frac{\lambda}{2}(v_{i+1}^n - v_{i-1}^n) + \frac{\lambda^2}{2}(v_{i+1}^n - 2v_i^n + v_{i-1}^n),$$

то A есть трехдиагональная матрица порядка N с $\sim 3N$ ненулевыми элементами. В данном случае $\hat{A}$ является *T-разреженной* (то есть становится разреженной, если пренебречь элементами меньше какого-то порога); ее Т-портрет $\mathsf{T}(\hat{A})$ имеет „спрутоподобную“ структуру заполнения и содержит порядка $\log N$ ненулей. Примечательно (на этом-то все и основано), что Т-портрет любой степени матрицы $\hat{A}$ имеет такую же структуру и такое же число ненулей. Поэтому удается эффективно вычислить $\mathsf{T}(\hat{A}^N)$ путем квадрирования — последовательного вычисления матриц

$$\mathsf{T}(\hat{A}^2),\ \mathsf{T}(\hat{A}^4) = \mathsf{T}((\mathsf{T}(\hat{A}^2))^2),\ \ldots .$$

Иной подход предлагается в [29]. Рассмотрим область индексов

$$\Gamma(v^n) \equiv \{(j,k) : |d_k^k(v^n)| \geqslant \varepsilon\},$$

характеризующих „информационно ценные“ коэффициенты иерархического представления, и расширенную область $\tilde{\Gamma}^n$, присоединяя к $\Gamma(v^n)$ сеточных соседей по j слева и справа и верхнего соседа по k. Вследствие конечной скорости распространения волн можно считать, что

$$\Gamma(v^{n+1}) \subset \tilde{\Gamma}^n.$$

Поэтому подлежат вычислению только те компоненты иерархического представления $\hat{v}^{n+1}$, которые попали в $\tilde{\Gamma}^n$. Все остальные компоненты сразу же полагаем равными нулю.

Можно, конечно, использовать локальные волны как базис для проекционных методов. При этом можно заранее предсказать, какие коэффициенты будут информационно ценными. Такой подход используется в работах [37,9] для решения уравнения Бюргерса

$$u_t + uu_x = \nu u_{xx}, \quad \nu > 0.$$

В случае $\nu = 0$ метод Галеркина, использующий локальные волны, может давать ложные осцилляции вблизи разрывов решения и иногда оказывается неустойчивым [29].

При решении нелинейных уравнений трудно (невозможно?) организовать вычисления лишь „в частотной области“. Поэтому кажется особенно привлекательной идея использования локальных волн *для динамической адаптации* исходных сеток или исходных базисных функций к особенностям решения. В работах [32,33] утверждается, что на этом пути можно, вообще говоря, добиваться такой же эффективности в простых случаях, как и в алгоритмах, переводящих вычисления в „частотную область“. Однако, оставаясь в сеточной области, мы не получаем специфических новых проблем, связанных с вычислением нелинейных членов.

6. МЕТОД СКЕЛЕТОНОВ

6.1. МОЗАИЧНЫЕ РАНГИ. Пусть A — матрица размеров $m \times n$. Если B — ее подматрица, то $\Gamma(B)$ — матрица рамеров $m \times n$, представляющая собой восполнение B нулями. Система блоков A_i называется *покрытием* A, если $A = \sum_i \Gamma(A_i)$. Если блоки не имеют общих элементов, то покрытие называется *мозаичным биением* матрицы A. Соответствующий ему мозаичный ранг матрицы A определяется формулой

$$\operatorname{mr} A = \sum_i \operatorname{mem} A_i/(m+n),$$

где (если блок A_i имеет размеры $m_i \times n_i$)

$$\operatorname{mem} A_i = \min\{m_i n_i, \operatorname{rank} A_i(m_i+n_i)\}.$$

С точки зрения экономии памяти и вычислений, роль мозаичного ранга аналогична роли классического ранга матрицы: если бы A имела ранг r, то число операций для умножения A на вектор (и память, чтобы хранить соответствующие скелетоны) имеет вид $r(m+n)$; то же самое сохраняется, если r — мозаичный ранг матрицы A.

При решении интегральных уравнений часто возникают невырожденные матрицы, имеющие относительно малый мозаичный ранг. Если n — порядок матрицы, то ее удается аппроксимировать матрицей с мозаичным рангом вида $\log n$ (или $\log^\gamma n$) [52]. Рассмотрим, например, интегральное уравнение с логарифмическим ядром

$$-\frac{1}{2\pi}\int_{\partial\Omega} \log|x-y| U(y)\,ds(y) = F(x), \quad x \in \partial\Omega.$$

Пусть $\partial\Omega$ — эллипс с полуосями $a = 1$ и $b = 0.5$. При использовании кусочно-постоянных базисных функций получаются матрицы с малым мозаичным ε-рангом ($\varepsilon = 10^{-4}$):

Порядок матрицы	512	1024	2048	4096	8192	16384	32768
Мозаичный ранг	63.46	71.48	78.44	86.84	93.80	100.76	106.50
Фактор сжатия	24.79%	13.96%	7.66%	4.24%	2.29%	1.23%	0.65%

Фактор сжатия — это отношение памяти для хранения скелетонов к полной памяти. При увеличении порядка в 2 раза фактор сжатия уменьшается примерно вдвое. Численные результаты, таким образом, вполне согласуются с теорией [52] (пожалуй, они даже более оптимистичны). Обратим внимание на существенность именно мозаичных биений: при стандартных биениях мозаичные ранги будут расти намного быстрее.

6.2. МОЗАИЧНОЕ ДЕРЕВО. Мозаичное биение строится с помощью *мозаичного дерева*, или *дерева кластеров*. Можно считать, что элемент a_{ij} в матрице характеризует „взаимодействие“ элементов сетки (базисных функций, конечных элементов) с номерами i и j. Поставим в соответствие элементу i точку x_i в пространстве (например, геометрический центр носителя базисной функции). Под *кластером* понимается совокупность каких-то точек x_i. Кластер нулевого уровня (корень дерева) — это все множество точек. Далее, каждый кластер разбивается на непересекающиеся подкластеры в соответствии с тем или иным методом сепарации. После какого-то разрешенного числа сепараций появляется дерево кластеров. Любые две вершины дерева кластеров определяют блок в матрице. Любое мозаичное биение определяется некоторым набором пар вершин. Таким образом, построение мозаичного биения можно рассматривать как комбинаторную задачу на мозаичном дереве. Цель — минимизировать итоговый мозаичный ранг.

Вот досточно разумный алгоритм. Мозаичное дерево просматривается начиная с корня. Для каждой пары вершин l-го уровня проверяется некоторый *признак сжимаемости* — ожидается, что соответствующий блок будет аппроксимирован малым числом скелетонов; такой блок сразу же включается в список блоков искомого мозаичного биения. Смысл — получить малый ранг для как можно более крупных блоков. С точки зрения теории здесь можно увидеть немало интересных вопросов. Однако в настоящее время я не вижу серьезной практической проблемы при построении мозаичного дерева и „хорошего“ мозаичного биения.

6.3. НЕПОЛНЫЕ СКЕЛЕТОННЫЕ АППРОКСИМАЦИИ. В принципе, скелетонную аппроксимацию для блока A_i можно найти с помощью метода бидиагонализации Ланцоша. При этом надо будет вычислить все элементы блока A_i. Однако в работах [2,22] показано, что существует *крест* из небольшого числа столбцов C и строк R, такой, что $A_i \approx CGR$ при выборе подходящей матрицы G. Пусть C и R содержат r столбцов и строк. Если *опорный блок* креста (матрица в пересечении C и R) имеет *наибольший объем* (модуль определителя) среди всех подматриц порядка r и $r+1$-е сингулярное число A_i меньше ε (другими словами, крест достаточно большой), то такой крест, вообще говоря, обеспечивает точность порядка ε. Еще требуются некоторые усилия, чтобы дать строгую формулировку и доказательство. Но уже сейчас мы используем это наблюдение на практике: оно позволяет вычислять лишь малую часть элементов исходной матрицы!

Ключевой для нас является следующая идея „улучшения“ креста. Пусть крест определяется строками $I = \{i_1, \ldots, i_r\}$ и столбцами $J = \{j_1, \ldots, j_r\}$. Находим в R и C подматрицы почти наибольшего объема — пусть они занимают столбцы $K = \{k_1, \ldots, k_r\}$ и строки $L = \{l_1, \ldots, l_r\}$. Тогда „улучшенный“ крест будет определяться строками $I \cup L$ и столбцами $J \cup K$. Практически требуется лишь малое число шагов „улучшения“, и стартовать можно, вообще говоря, с одной строки и одного столбца. Именно этот алгоритм использовался при получении таблицы из раздела 6.1. Более детальное обсуждение должно стать, очевидно, предметом отдельной работы.

Литература

1. Воеводин В. В. Об одном методе понижения порядка матриц при решении интегральных уравнений // *Численный анализ на ФОРТРАНе.* — М.: Изд-во МГУ. 1979. С. 21–26.

2. Горейнов С. А., Замарашкин Н. Л., Тыртышников Е. Е. Псевдоскелетные аппроксимации матриц // *Доклады РАН.* 1995. №343(2). С. 151–152.

3. Михайловский Н. Э. Мозаичные аппроксимации дискретных аналогов операторов Кальдерона-Зигмунда // *Математические заметки*. 1998. №63(1). С. 81–94.

4. Нечепуренко Ю. М. *Быстрые численно устойчивые алгоритмы для широкого класса линейных дискретных преобразований*: Препринт №92, ОВМ РАН, 1985.

5. Arandiga F., Candela V. *Multiresolution standard form of a matrix*. UCLA CAM Report 92-37, August 1992.

6. Arandiga F., Candela V., Donat R. *Fast multiresolution algorithms for solving linear equations: a comparative study*. UCLA CAM Report 92-52, December 1992.

7. Arandiga F., Donat R., Harten A. *Multiresolution based on weighted averages of the hat function*. UCLA CAM Report 93-34, September 1993.

8. Abgrall R., Harten A. *Multiresolution analysis on unstructured meshes: preliminary report*. UCLA CAM Report 94-26, July 1994.

9. Bacry E., Mallat S., Papanicolau G. A wavelet based space-time adaptive numerical methods for partial differential equations // *Math. Modelling and Numerical Analysis*. 1992. V. 26. P. 703–834.

10. Beylkin G., Coifman R., Rokhlin V. Fast wavelet transform and numerical algorithms. I // *Comm. Pure Appl. Math.* 1991. V. 44. P. 141–183.

11. Brandt A. Multilevel computations of integral transforms and particle interactions with oscillatory kernels // *Computer Physics Communications*. 1991. V. 65. P. 24–38.

12. Brandt A., Lubrecht A. A. Multilevel matrix multiplication and fast solution of integral equations // *J. Comput. Phys.* 1990. V. 90. P. 348–370.

13. Briggs W., Henson V. E. Wavelets and multigrid // *SIAM J. Sci. Comput.* 1993. V. 14(2). P. 506–510.

14. Cai W., Wang J.-Z. *Adaptive wavelet collocation methods for initial value boundary problems of nonlinear PDE's*. ICASE Report 93-48, July 1993.

15. Canning F. X. The impedance matrix localization (IML) method for moment-method calculations // *IEEE Antennas Propagat. Mag.* 1990. V. 32. P. 18–30.

16. Carrier J., Greengard L., Rokhlin V. A fast adaptive multipole algorithm for particle simulations // *SIAM J. Sci. Stat. Comput.* 1988. V. 9. P. 669–686.

17. Cohen A., Daubechies I., Feauveau J.-C. Biorthogonal bases of compactly supported wavelets // *Comm. Pure Appl. Math.* 1992. V. 45. P. 485–560.

18. Coifman R., Rokhlin V., Wandzura S. The fast multipole method for the wave equation: a pedestrian prescription // *IEEE Antennas Propagat. Mag.* V. 35(3). P. 7–12.

19. Daubechies I. Orthonormal bases of compactly supported wavelets // *Comm. Pure Appl. Math.* 1988. V. 41. P. 909–996.

20. Engheta N., Murphy W. D., Rokhlin V., Vassiliou M. S. The fast multipole method (FMM) electromagnetic scattering problems // *IEEE Trans. on Antennas and Propagat.* 1992. V. 40(6). P. 634–641.

21. Engquist B., Osher S., Zhong S. Fast wavelet algorithms for linear evolution equations // *SIAM J. Sci. Comput.* 1994. V. 15(4). P. 755–775.

22. Goreinov S. A., Tyrtyshnikov E. E., Zamarashkin N. L. A Theory of Pseudo-Skeleton Approximations // *Linear Algebra Appl.* 1997. V. 261. P. 1–21.

23. Goreinov S. A., Tyrtyshnikov E. E., Yeremin A. Yu. Matrix-Free Iterative Solution Strategies for Large Dense Linear Systems // *Numerical Linear Algebra with Applications.* 1997. V. 4(4). P. 273–294.

24. Greengard L., Rokhlin V. A fast algorithm for particle simulations // *J. Comput. Physics.* 1987. V. 73. P. 325–348.

25. Hackbusch W., Novak Z. P. On the fast matrix multiplication in the boundary element method by panel clustering // *Numer. Math.* 1989. V. 54(4). P. 463–491.

26. Harten A. Discrete multiresolution analysis and generalized wavelets // *J. Appl. Numer. Math.* 1993. V. 12. P. 153–193.

27. Harten A. *Multiresolution algorithms for the numerical computation of hyperbolic conservation laws.* UCLA CAM Report 93-03, March 1993.

28. Harten A. *Adaptive multiresolution schemes for shock computations.* UCLA CAM Report 93-06, April 1993.

29. Harten A. *Multiresolution representation and numerical algorithms: a brief review.* ICASE Report 94-59, October 1994.

30. Harten A., Yad-Shalom I. Fast multiresolution algorithms for matrix-vector multiplication // *SIAM J. Numer. Anal.* 1994. V. 31(4). P. 1191–1218.

31. Heil C. E., Walnut D. F. Continuous and discrete wavelet transforms // *SIAM Review.* 1989. V. 31(4). P. 628–666.

32. Jameson L. *On the wavelet optimized finite difference method.* ICASE Report 94-09, March 1994.

33. Jameson L. *On the spline-based wavelet differentiation matrix.* ICASE Report 93-80, November 1993.

34. Jameson L. *On the Daubechies-based wavelet differentiation matrix.* ICASE Report 93-95, December 1993.

35. Kalbasi K., Demarest K. R. A multilevel formulation of the method of moments // *IEEE Trans. on Antennas and Propagat.* 1993. V. 41(5). P. 589–599.

36. Kim H., Ling H. Wavelet analysis of radar echo from finite-size targets // *IEEE Trans. on Antennas and Propagat.* 1993. V. 41(2). P. 200–207.

37. Liandrat J., Tchamitchian Ph. *Resolution of the 1D regularized Burgers' equation using a spatial wavelet approximation algorithm and numerical results.* ICASE Report 90-83, December 1990.

38. Mallat S. Multiresolution approximation and wavelet orthogonal bases of $L^2(\mathbb{R})$ // *Trans. Amer. Math. Soc.* 1989. V. 315. P. 69–87.

39. Myagchilov S. V., Tyrtyshnikov E. E. A fast matrix-vector multiplier in discrete vortex method // *Russian J. Numer. Anal. Math. Modelling.* 1992. V. 7(4). P. 325–342.

40. Murphy W. D., Rokhlin V., Vassiliou M. S. Solving electromagnetic scattering problems at resonance frequences // *J. Appl. Phys.* 1990. V. 67. P. 6061–6065.

41. Nabors K., White J. Fastcap: a multipole accelerated 3-D capacitance extraction program // *IEEE Trans. on Computer-Aided Design of Integral Circuits and Systems.* 1991. V. 10. P. 1447–1459.

42. Nabors K., Kim S., White J. Fast capacitance extraction of general three-dimensional structures // *IEEE Trans. on Microwave Theory and Tech.* 1992. V. 40. P. 1496–1506.

43. Nabors K., Korsmeyer F. T., Leighton F. T., White J. *Preconditioned, Adaptive, Multipole-Accelerated Iterative Methods for Three-Dimensional Potential Integral Equations of the First Kind.* Dept. of Electrical Eng. and Computer Science, Massachusetts Institute of Technology, 1994.

44. Rokhlin V. Rapid solution of integral equations of classical potential theory // *J. Comput. Physics.* 1985. V. 60. P. 187–207.

45. Rokhlin V. Rapid solution of integral equations of scattering theory in two dimensions // *J. Comput. Physics.* 1990. V. 86. P. 414–439.

46. Schmidt K. E., Lee M. A. Implementing the fast multipole method in three dimensions // *J. of Statistical Physics.* 1991. V. 63(56). P. 1223–1235.

47. Steinberg B. Z., Leviatan Y. On the use of wavelet expansions in the method of moments // *IEEE Trans. on Antennas and Propagat.* 1993. V. 41(5). P. 610–619.

48. Strang G. Wavelets and dilation equations: a brief introduction // *SIAM Review.* 1989. V. 31. P. 614–627.

49. Strang G. Wavelet transforms versus Fourier transforms // *Bulletin or the Amer. Math. Soc.* 1993. V. 28(2). P. 288–305.

50. Sweldens W., Piessens R. Quadrature formulae and asymptotic error expansions for wavelet approximations of smooth functions // *SIAM J. Numer. Anal.* 1994. V. 31(4). P. 1240–1264.

51. Tyrtyshnikov E. E. Mosaic ranks and skeletons // *Lecture Notes in Computer Science 1196: Numerical Analysis and Its Applications. Proceedings of WNAA-96. Eds.: Vulkov L. et al.* Springer-Verlag. 1996. P. 505–516.

52. Tyrtyshnikov E. E. Mosaic–skeleton approximations // *Calcolo.* 1996. V. 33(1-2). P. 47–57.

Мозаично-скелетонные аппроксимации матриц, порожденных асимптотически гладкими и осцилляционными ядрами*

С. А. Горейнов

Аннотация

При решении интегральных уравнений теории потенциала возникает задача быстрого приближенного умножения на дискретный аналог интегрального оператора. Один из подходов к решению этой задачи, называемый мозаично-скелетонной аппроксимацией [8], связан с понятием асимптотически гладкой функции. Именно, если ядро интегрального уравнения принадлежит названному классу функций, приближенное матрично-векторное умножение на дискретизированный оператор размерности $n \times n$ *можно выполнить за* $\mathcal{O}(n \log n)$ *арифметических операций. В настоящей статье результаты [8] переносятся на случай некоторых осциллирующих ядер, именно, фундаментальных решений уравнения Гельмгольца в двумерном и трехмерном случаях, причем при более слабых ограничениях на сетку. Получены также новые оценки асимптотической гладкости некоторых ядер.*

1. ВВЕДЕНИЕ. Настоящая работа посвящена изучению одного из подходов для быстрого приближенного умножения на плотные неструктурированные матрицы. Такая задача возникает, например, при построении численного метода решения уравнения

$$\int_S K(x, y)\, u(y)\, dS_y = f(x),$$

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155) и фонда Volkswagen-Stiftung (грант № VW I/71 493).

которое каким-то образом дискретизируется (скажем, коллокацией или по методу Галеркина), и возникает линейная система $A_n x_n = b_n$, размерность n которой растет при увеличении точности дискретизации. Ключевой вопрос — как быстро растет объем памяти, необходимый для хранения дискретизированного оператора, и число арифметических действий, необходимое для решения задачи с заданной точностью, в зависимости от n.

Предположим, что для решения линейной системы используется какой-либо итерационный метод (возможно, с предобуславливанием). Тогда основные вычислительные затраты связаны с операцией матрично-векторного умножения на A_n, что приводит к следующей задаче матричной аппроксимации: найти „близкую" к A_n матрицу $\tilde{A}_n$, такую, что:

- матрично-векторное умножение $y = \tilde{A}_n x$ выполняется быстро;
- объем памяти, необходимый для хранения $\tilde{A}_n$, мал;
- погрешность в решении, допускаемая при замене A_n на $\tilde{A}_n$, сравнима с погрешностью дискретизации.

Говоря „быстро" и „мал", мы имеем в виду $o(n^2)$ арифметических операций и $o(n^2)$ ячеек памяти. Сложнее указать строгий критерий близости A_n и $\tilde{A}_n$, обеспечивающий допустимую погрешность в решении. В дальнейшем мы будем понимать под этим условие $\|A_n - \tilde{A}_n\|_F \leqslant \varepsilon \|A_n\|_F$, где ε не зависит от n, потому что для тех конкретных уравнений, с которыми мы имели дело, этого (при надлежащем выборе ε) в самом деле достаточно.

За последние 10–15 лет появилось довольно много подходов к решению такой задачи (см., например, [9]). С алгебраической точки зрения, большинство этих подходов явно или неявно базируются на одном и том же наблюдении: для весьма широкого класса ядер $K(x, y)$ и для всех достаточно больших n можно указать такое блочное разбиение матриц A_n, что подавляющая часть блоков будет иметь хорошее малоранговое приближение. Именно эта идея напрямую исследуется в [8] в случае, когда ядро является *асимптотически гладкой* функцией. Так, вслед за Брандтом [1], называют функцию $K(x, y)$, если

$$|D_y^p K(x, y)| \leqslant c_p \|x - y\|^{g-p}, \qquad \forall\, p \geqslant 0. \tag{1.1}$$

Здесь D_y^p обозначает оператор дифференцирования по y порядка p; c_p — положительная константа, зависящая от p; g — некоторая константа; норма $\|\cdot\|$ — евклидова.

В этом случае основным инструментом для получения оценок на ε-ранг блока служит формула Тейлора: в общем члене переменные x и y разделены, поэтому ранг суммы таких членов легко оценивается и не зависит от n, а остаточный член с учетом (1.1) определяет погрешность аппроксимации. Отметим, что теорема об аппроксимации всей матрицы следует из такого рода оценок только в том случае, когда делаются какие-либо предположения о характере зависимости c_p от p — не слишком обременительные, но приводящие к приемлемому результату. Мы будем пользоваться определением (2.1), принадлежащим Тыртышникову [8].

Главный результат работы [8] выглядит следующим образом. Пусть $S \subset \mathbb{R}^m$ — ограниченная область, в которой заданы наборы узлов $\{x(i)\}_{i=1}^n$ и $\{y(j)\}_{j=1}^n$, удовлетворяющие условию *квазиравномерности:*

$$\tau_1 \frac{\operatorname{mes}\Omega}{\operatorname{mes} S}\, n \;\leqslant\; \mu(\Omega) \;\leqslant\; \tau_2 \frac{\operatorname{mes}\Omega}{\operatorname{mes} S}\, n, \qquad \forall\, \Omega \in S.$$

Символом $\mu(\Omega)$ здесь обозначено число узлов, принадлежащих области Ω. Пусть, далее, $K(x, y)$ — асимптотически гладкая функция. Тогда для матрицы $A_n = [K\big(x(i), y(j)\big)]_{i,j=1}^n$ существует *мозаично-скелетонная* аппроксимация $\tilde{A}_n$, такая, что

$$\|A_n - \tilde{A}_n\|_F = \mathcal{O}(n\varepsilon),$$

$\tilde{A}_n$ занимает $\mathcal{O}(n \log n \log^m 1/\varepsilon)$ ячеек памяти и за столько же арифметических операций может быть умножена на вектор. Символ $\mathcal{O}$ относится здесь к пределу $n \to \infty$, а ε — произвольное достаточно малое положительное число.

Несколько изменив технику доказательства, нам удалось получить такую же теорему при ослабленном требовании к сеткам; достаточно, чтобы

$$\mu(\Omega) \;\leqslant\; \tau \frac{\operatorname{mes}\Omega}{\operatorname{mes} S}\, n, \qquad \forall\, \Omega \subset S.$$

Это означает, что в качестве S можно выбирать, например, многообразие меньшей размерности, чем m, с сохранением указанных оценок аппроксимации.

Впрочем, в некоторых случаях показатель степени в члене $\log^m 1/\varepsilon$ можно уменьшить до размерности многообразия S. Мы доказываем подобный результат для случая, когда S является простым несамопересекающимся *кусочно-целым* контуром. Имеется в виду следующее: пусть функция $y(\nu) : [0, 1] \mapsto \mathbb{C}$, параметризующая контур, такова, что для каждого интервала какого-либо конечного разбиения отрезка $[0, 1]$ $\Re y(\nu)$ и $\Im y(\nu)$ представляются рядами Тейлора с бесконечным радиусом сходимости, имеют отделенные от нуля производные и при $\nu \to \infty$ растут не быстрее экспоненты. Тогда для матриц, порожденных асимптотически гладкой по y функцией $K(x, y)$, справедливы приведенные выше оценки с $m = 1$. Эти результаты составляют содержание раздела 2. В разделе 3 исследуется асимптотическая гладкость некоторых конкретных функций; кроме того, доказано, что всякая гармоническая функция является асимптотически гладкой.

В разделах 4 и 5 сделана попытка использовать для получения интересующих нас оценок разложения, отличные от формулы Тейлора. В разделе 4 гармонические по переменному y ядра анализируются при помощи производящей функции для ультрасферических многочленов; в разделе 5 мы строим мозаично-скелетонные аппроксимации для фундаментальных решений уравнения Гельмгольца в двумерном и трехмерном случае при помощи теорем сложения для цилиндрических функций. В целях замкнутости изложения некоторые классические результаты, на которые мы опираемся, приведены с доказательствами.

Численные эксперименты в работе отсутствуют. Свидетельства того, что мозаично-скелетонные аппроксимации в принципе могут приводить к эффективным алгоритмам, имеются в [3]; в то же время, изучение асимптотических свойств конкретного алгоритма и сравнение с другими подходами есть, безусловно, предмет для отдельной работы.

1.1. ИСПОЛЬЗУЕМЫЕ ОБОЗНАЧЕНИЯ.

$|z| = \|z\|_2$ — евклидова норма вектора, или модуль числа z;

основание логарифма там, где это важно, считается равным двум: $\log x = \log_2 x$;

$\lfloor x \rfloor$ — наибольшее $n \in \mathbb{Z}$ такое, что $n \leqslant x$;

$\lceil x \rceil$ — наименьшее $n \in \mathbb{Z}$ такое, что $n \geqslant x$;

$z^{\underline{k}} = z\,(z-1)\,\cdots\,(z-k+1)$; $z^{\overline{k}} = z\,(z+1)\,\cdots\,(z+k-1)$ — символ Похгаммера;

$\binom{z}{k} = z^{\underline{k}}/k!$ — биномиальный коэффициент;

$\binom{z}{\nu} = z!/\nu!$ — „мультиномиальный“ коэффициент, соответствующий мультииндексу ν;

$\Gamma(z) = \int_0^\infty e^{-t}\, t^{z-1}\, dt, \quad \Re z > 0$ — гамма-функция Эйлера;

$B(x, y) = \int_0^1 t^{x-1}\,(1-t)^{y-1}\,dt, \quad \Re x > 0,\ \Re y > 0$ — бета-функция Эйлера;

const — положительная константа, не зависящая от изменяемых параметров.

2. АППРОКСИМАЦИЯ ДЛЯ АСИМПТОТИЧЕСКИ ГЛАДКИХ ЯДЕР. Для наших целей определение Брандта (1.1) слишком широко. В соответствии с [8] будем называть функцию $K(x, y)$: $S \times S \mapsto \mathbb{C}, \quad S \subset \mathbb{R}^m$, асимптотически гладкой по переменному y, если найдутся положительные числа c, d и вещественное число g такие, что

$$|D_y^p K(x, y)| \leqslant c\, d^p\, p!\, |x-y|^{g-p}, \qquad \forall\, p \geqslant 0, \tag{2.1}$$

где $D_y^p = \frac{\partial^p}{\partial y_1^{p_1} \partial y_2^{p_2} \dots \partial y_m^{p_m}}, \quad p_1 + p_2 + \dots + p_m = p, \quad x \neq y.$

Множество асимптотически гладких ядер непусто; таковым, например, является фундаментальное решение уравнения Лапласа (см. раздел 3). Однако функция $e^{i\kappa|x-y|}/|x-y|$, пример не менее важный, в это множество не входит.

Предположим все-таки, что функция $K(x, y)$ асимптотически гладкая. Тогда ε-ранг матрицы, порожденной этой функцией на паре сеток $\{x(i)\}$ и $\{y(j)\}$, которые хорошо друг от друга отделены, оказывается мал. Именно, пусть узлы $\{y(j)\}_{j=1}^{n_y}$ сосредоточены в кубе $C_a(\eta) = \{y : \|y - \eta\|_\infty \leqslant a/2\}$ и

$$\{x(i)\}_{i=1}^{n_x} \subset \{x : \|x - y\|_\infty \geqslant \sigma a \qquad \forall\, y \in C_a(\eta)\},$$

где $\sigma > dm/2$. Тогда имеет место

Лемма 1. $\forall\, p \in \mathbb{N}, \ p \geqslant g,$ справедливо равенство

$$A \equiv [K\big(x(i), y(j)\big)]_{i=1\,j=1}^{n_x\ n_y} = T_p + R_p,$$

где

$$\operatorname{rank} T_p \leqslant p^m,$$

$$\|R_p\|_F^2 \leqslant n_x n_y c^2\, (\sigma a)^{2g} \left(\frac{dm}{2\sigma}\right)^{2p}.$$

Доказательство. По формуле Тейлора для функции m переменных $y_1, y_2, \dots, y_m$ имеем

$$K(x,y) = \sum_{k=0}^{p-1} \frac{\left((y-\eta)^T \nabla_\xi\right)^k}{k!} K(x,\xi)\Big|_{\xi=\eta} + \frac{\left((y-\eta)^T \nabla_\xi\right)^p}{p!} K(x,\xi)\Big|_{\xi=\theta},$$

где $\theta - \eta = q(y-\eta)$ для некоторого $q \in (0,1)$. Первое слагаемое представляет собой вырожденное ядро, следовательно, ранг порожденной им матрицы, которую мы обозначим T_p, оценивается сверху рангом ядра и не зависит от n_x и n_y. Оценим сначала число слагаемых μ_{km} в правой части мультиномиальной формулы [4]

$$\left(\sum_{i=1}^m x_i\right)^k = \sum_{|\nu|=k} \frac{k!}{\nu_1! \cdots \nu_m!}\, x^\nu. \qquad (*)$$

Здесь $\nu = (\nu_1, \nu_2, \dots, \nu_m)$ — мультииндекс, $x^\nu = x_1^{\nu_1} \cdots x_m^{\nu_m}$. Фиксируя значение первой компоненты мультииндекса, получаем рекуррентную формулу $\mu_{km} = \sum_{i=0}^k \mu_{i,m-1}$. Кроме того, имеем $\mu_{0m} = \mu_{k1} = 1$. По индукции получаем $\mu_{km} = \binom{k+m-1}{m-1}$, и окончательно

$$\operatorname{rank} T_p \leqslant \sum_{k=0}^{p-1} \mu_{km} = \binom{p+m-1}{m} = p^m \frac{\prod_{k=0}^{m-1}\left(1+\frac{k}{p}\right)}{m!} \leqslant p^m.$$

Для остатка формулы Тейлора R в силу (2.1), $(*)$ имеем

$$|R(x,y)| \leqslant \frac{1}{p!} \sum_{|\nu|=p} \binom{p}{\nu} |(y-\eta)^\nu|\, c\, d^p\, p!\, (\sigma a)^{g-p} =$$

$$= c\, d^p\, (\sigma a)^{g-p} \|y-\eta\|_1^p \leqslant c \left(\frac{dm}{2\sigma}\right)^p (\sigma a)^g. \quad \blacksquare$$

Лемма 1, в сущности, утверждает, что блоки матрицы, соответствующие хорошо разделенным в пространстве наборам узлов, имеют хорошую малоранговую аппроксимацию (иногда называемую скелетонной, по ассоциации со скелетным разложением). Однако вся матрица соответствует ядру, переменные которого пробегают две *совпадающие* в пространстве области. Чтобы аппроксимировать всю матрицу, нам придется эти области (для простоты, кубы) дробить и применять лемму 1 для хорошо отделенных подкубов, в то время как аппроксимация близких или совпадающих подкубов будет происходить индуктивно, то есть последние снова дробятся на более мелкие, и так далее.

Для формального изложения этой идеи введем понятия *мозаичного разбиения* и *мозаичного ранга*.

Пусть $\hat{A}$ — подматрица матрицы A. Символом $\Pi(\hat{A})$ будем обозначать множество позиций, которые $\hat{A}$ занимает в A.

Определение 1. Конечное множество блоков $\{A_k\}$ матрицы $A \in \mathbb{C}^{m\times n}$ будем называть *покрытием* A, если

$$\bigcup_k \Pi(A_k) \supseteq \{(i,j)\colon\ 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n\},$$

и *мозаичным разбиением* A, если, дополнительно, $\bigcap_k \Pi(A_k) = \varnothing$.

Пусть выбрано какое-то покрытие данной матрицы A. Чтобы аппроксимировать A, мы предлагаем аппроксимировать каждый блок покрытия одним из „стандартных“ способов. Простейшему набору — хранение всех элементов блока и хранение скелетного разложения блока — отвечает

Определение 2. *Мозаичным рангом* матрицы $A \in \mathbb{C}^{m\times n}$, соответствующим некоторому покрытию, называется число

$$\operatorname{mr} A = \sum_k \operatorname{mem} A_k/(m+n),$$

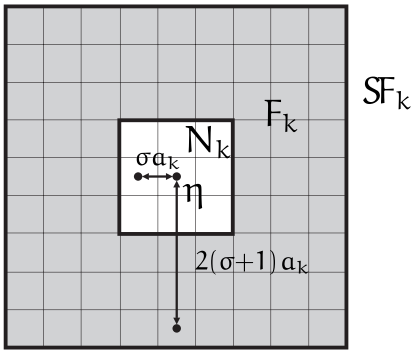
где сумма берется по всем блокам покрытия $A_k \in \mathbb{C}^{m_k\times n_k}$, и $\operatorname{mem} A_k = \min\,\{m_k n_k,\ (m_k+n_k)\operatorname{rank} A_k\}$.

Заметим, что число арифметических действий, необходимое для матрично-векторного умножения на A, и память, необходимая для хранения $A \in \mathbb{C}^{m\times n}$, равны соответствующим величинам для матрицы $m \times n$ (обычного) ранга $\operatorname{mr} A$.

Погрузим область S в куб со стороной α и, поделив каждое ребро куба пополам, разделим его на 2^m подкубов. Проделывая такую же операцию с подкубами, получим *иерархическое разбиение* исходного куба. Определим k-й уровень разбиения L_k как множество кубов с длиной стороны $a_k = \alpha\, 2^{-k}$; таких кубов будет, очевидно, 2^{km}. Центр куба k-го уровня, содержащего заданную точку x (или y), будем обозначать символом ξ_k (или η_k), а сам куб символом $C_k(\xi_k)$ (или $C_k(\eta_k)$). Для всякого куба k-го уровня с центром η

определим области (или „зоны“)

$$\begin{aligned} N_k(\eta) &= \{C_k(\xi) \in L_k : \quad \|\xi-\eta\|_\infty \leqslant \sigma a_k\}, \\ F_k(\eta) &= \{C_k(\xi) \in L_k : \quad 2(\sigma+1)a_k \geqslant \|\xi-\eta\|_\infty \geqslant (\sigma+1)a_k\}, \\ SF_k(\eta) &= \{C_k(\xi) \in L_k : \quad \|\xi-\eta\|_\infty > 2(\sigma+1)a_k\}, \end{aligned} \tag{2.2}$$



Зоны при $\sigma = 1$

где $\sigma \in \mathbb{N}$ — параметр, значение которого определится в дальнейшем. Зоны N_k будем называть „ближними“, F_k — „дальними“. Заметим, что $\forall\, y \in C_k(\eta)$, $\forall\, x \in F_k(\eta)$

$$\sigma a_k \leqslant \|x-y\|_\infty \leqslant (2\sigma+3)a_k.$$

Лемма 2. Пусть $s \in \mathbb{N}$ — последний уровень иерархического разбиения. Тогда $\forall\, x, y \in S$ либо $x \in N_s(\eta_s)$, либо $x \in F_k(\eta_k)$ для некоторого номера k, $1 < k \leqslant s$.

Доказательство. Пусть $x \notin N_s(\eta_s) \cup F_s(\eta_s)$, тогда $\|\xi_s - \eta_s\|_\infty > (\sigma+1)a_{s-1}$. Поскольку $\|\eta_s - \eta_{s-1}\|_\infty = a_s/2$, $\|\xi_s - \xi_{s-1}\|_\infty = a_s/2$, то по неравенству треугольника

$$\|\xi_{s-1} - \eta_{s-1}\|_\infty > (\sigma+\tfrac{1}{2})a_{s-1} \quad\Rightarrow\quad \|\xi_{s-1} - \eta_{s-1}\|_\infty \geqslant (\sigma+1)a_{s-1},$$

то есть $x \in F_{s-1}(\eta_{s-1}) \cup SF_{s-1}(\eta_{s-1})$. Если $x \in F_{s-1}(\eta_{s-1})$, то положим $k = s-1$; иначе $x \notin N_{s-1}(\eta_{s-1}) \cup F_{s-1}(\eta_{s-1})$, и наши выкладки можно повторить, уменьшив s на единицу. При этом формула $x \in SF_k(\eta_k)$ для всех k от s до 1 выполняться не может, поскольку $\|\xi_1 - \eta_1\|_\infty = a_1 \leqslant \sigma a_1$. ■

Таким образом, блоки матрицы, соответствующие зонам N_s и $\{F_k\}_{k=1}^s$ для всех кубов соответствующих уровней, определяют некоторое мозаичное покрытие. Построим матрицу $\tilde{A}_n$, используя скелетонные аппроксимации для блоков дальней зоны и точное представление для блоков ближней зоны. Допускаемую при этом погрешность и мозаичный ранг $\tilde{A}_n$ можно оценить следующим образом.

Теорема 1. Пусть матрица A_n порождается асимптотически гладким ядром $K(x, y)$ на сетках $\{x(i)\}_{i=1}^n$, $\{y(j)\}_{j=1}^n$, заданных на ограниченном множестве $S \subset \mathbb{R}^m$ и подчиненных условию

$$\mu(\Omega) \leqslant \tau \frac{\operatorname{mes}\Omega}{\operatorname{mes} S}\, n, \qquad \forall\, \Omega \subset S, \tag{2.3}$$

где τ — положительная константа, Ω — пересечение S с произвольным кубом из описанного выше разбиения, а $\mu(\Omega)$ обозначает число узлов сетки, принадлежащих Ω. Тогда $\forall\, \varepsilon,\ 0 < \varepsilon < \varepsilon_0,$ и $\forall\, n \in \mathbb{N},\ n > n_0$, существуют мозаично-скелетонные аппроксимации $\tilde{A}_n$, для которых

$$\begin{aligned} \operatorname{mr} \tilde{A}_n \; &< \; c_1 \log n \log^m \frac{1}{\varepsilon}\,, \\ \|A_n - \tilde{A}_n\|_F \; &< \; c_2\, n^{\gamma}\, \varepsilon, \end{aligned} \tag{2.4}$$

где

$$\gamma = \begin{cases} 1, & \text{если } m + 2g > 0, \\ \frac{|g|}{m} + \frac{1}{2}, & \text{если } m + 2g \leqslant 0. \end{cases}$$

В случае $m + 2g \leqslant 0$ существуют также аппроксимации, удовлетворяющие оценкам

$$\begin{aligned} \operatorname{mr} \tilde{A}_n \; &< \; c_1 \log^{m+1} n \log^m \frac{1}{\varepsilon}\,, \\ \|A_n - \tilde{A}_n\|_F \; &< \; c_2\, n\varepsilon. \end{aligned} \tag{2.5}$$

Константы c_1, c_2 положительны и не зависят от n и ε.

Доказательство. В силу (2.3) размеры всякого блока ближней зоны s-го уровня удовлетворяют неравенствам

$$n_y(s) \leqslant \hat{\tau} n 2^{-sm}, \quad n_x(s) \leqslant (2\sigma{+}1)^m \hat{\tau} n 2^{-sm},$$

где $\hat{\tau} = \tau\alpha^m / \operatorname{mes} S$. Отсюда мы усматриваем естественное ограничение на s:

$$\hat{\tau} n 2^{-sm} \geqslant 1 \quad \Rightarrow \quad s \leqslant \frac{1}{m} \log \hat{\tau} n.$$

Поскольку выбор s в нашей власти, возьмем $s = \left\lfloor \frac{1}{m} \log \hat{\tau} n \right\rfloor$; в этом случае объем памяти, необходимой для хранения элементов всех блоков ближней зоны,

$$\operatorname{mem}(N) \leqslant 2^{sm} n_x(s) n_y(s) \leqslant 2^m (2\sigma{+}1)^m \hat{\tau} n.$$

Для блоков дальней зоны будем использовать малоранговые приближения (например, наилучшие в смысле фробениусовой нормы, то есть частичные сингулярные разложения [3]). Если для фиксированного ранга используемые приближения обеспечивают наилучшую или сравнимую с таковой погрешность, можно использовать оценки погрешности тех аппроксимаций, которые исследуются в лемме 1.

Для всякого блока, соответствующего дальней зоне уровня k, возьмем число членов ряда Тейлора в аппроксимации по лемме 1 равным p_k. Положим $\sigma = \lceil dm \rceil$. Поскольку размеры блока дальней зоны ограничены формулами

$$n_y(k) \leqslant \hat{\tau} n 2^{-km}, \quad n_{fx}(k) \leqslant \big((4\sigma+5)^m - (2\sigma+1)^m\big)\hat{\tau} n 2^{-km},$$

то объем памяти, необходимый для хранения скелетонных приближений для всех блоков дальней зоны, оценивается как

$$\mathrm{mem}\,(F) \leqslant \sum_{k=1}^{s} p_k^m \big(n_{fx}(k) + n_y(k)\big) 2^{km} \leqslant \mathrm{const} \cdot \sigma^m n \sum_{k=1}^{s} p_k^m. \tag{2.6}$$

Погрешность аппроксимации всей матрицы равна погрешности аппроксимации блоков дальней зоны, поэтому в силу леммы 1

$$\begin{aligned} \|A_n - \tilde{A}_n\|_F^2 &\leqslant \sum_{k=1}^{s} n_{fx}(k)\, n_y(k)\, c^2 (\sigma a_k)^{2g}\, 2^{-2p_k}\, 2^{km} \\ &\leqslant \mathrm{const} \cdot \sigma^{m+2g}\, n^2 \sum_{k=1}^{s} 2^{-k(m+2g)} 2^{-2p_k}. \end{aligned} \tag{2.7}$$

Анализ правой части (2.7) существенно зависит от знака величины $m + 2g$. Положим $q = 2^{-m-2g}$, $p_k = \lceil \log \frac{1}{\varepsilon} \rceil$; тогда при $m + 2g > 0$ будет $q < 1$ и

$$\|A_n - \tilde{A}_n\|_F^2 \leqslant \mathrm{const} \cdot n^2 \sum_{k=1}^{s} q^k\, 2^{-2\log 1/\varepsilon} \leqslant \mathrm{const} \cdot (n\varepsilon)^2;$$

в то же время $\mathrm{mem}(F) \leqslant \mathrm{const} \cdot n s \log^m \dfrac{1}{\varepsilon}$, что и требовалось. Далее, при $m + 2g \leqslant 0$, $\;q = 2^{-m-2g} \geqslant 1$ имеем

$$\|A_n - \tilde{A}_n\|_F^2 \leqslant \mathrm{const} \cdot (n\varepsilon)^2 q^s \leqslant \mathrm{const} \cdot (n\varepsilon)^2 (\hat{\tau} n)^{\frac{2|g|}{m} - 1},$$

то есть неравенства (2.4) установлены.

Оценки (2.5) получаются из (2.6)-(2.7), если взять $p_k = \lceil k \log \frac{1}{\varepsilon} \rceil$; тогда сумма для погрешности аппроксимации ограничена по сравнению с $\sum_{k=1}^{\infty} q^k \varepsilon^{2k} \leqslant \frac{q\varepsilon^2}{1-q\varepsilon^2} \leqslant 2q\varepsilon^2$, как только $q\varepsilon^2 < 1/2$; соответствующая сумма для мозаичного ранга будет

$$\sum_{k=1}^{s} k^m \log^m \frac{1}{\varepsilon} \leqslant \text{const} \cdot s^{m+1} \log^m \frac{1}{\varepsilon} \leqslant \text{const} \cdot \log^{m+1} n \log^m \frac{1}{\varepsilon}.$$

■

Заметим, что при $m + 2g \leqslant 0$ оценку погрешности аппроксимации по формуле (2.4) можно заменой переменного ε привести к виду $\mathcal{O}(n\varepsilon)$. При одинаковой погрешности такого вида, фиксированном ε и $n \to \infty$ формулы (2.5) дают лучшую оценку мозаичного ранга лишь в том случае, когда $-g > \frac{m}{2} + m \log \frac{1}{\varepsilon}$.

Если $S \in \mathbb{R}^m$ — многообразие размерности, меньшей m, изложенный результат может быть усилен. Рассмотрим двумерный случай; пусть S — плоская ограниченная кривая, параметризованная функцией $y(\nu) : [0, 1] \mapsto \mathbb{C}$, $y(\nu) = y_1(\nu) + iy_2(\nu)$, причем отрезок $[0, 1]$ может быть разбит при помощи $J \geqslant 0$ *точек разрыва* так, что на каждом интервале разбиения выполнены следующие условия:
1. y_1 и y_2 суть целые функции ν, то есть представляются рядами Тейлора с бесконечным радиусом сходимости;
2. $|y_1'(\nu)| \geqslant \mu$, $|y_2'(\nu)| \geqslant \mu$ для некоторого положительного μ;
3. $y_{1,2}(\nu) = \mathcal{O}(e^{\beta|\nu|})$, $\nu \to \infty$, $\beta > 0$.

Будем называть такие кривые *кусочно-целыми*. Ограничения на кривую, как видим, довольно сильные; связано это с тем, что аналог леммы 1 об аппроксимации асимптотически гладкого ядра мы можем доказать лишь для такого класса кривых. Если сузить класс рассматриваемых ядер, например, до гармонических, условия на кривую могут быть отброшены (см. раздел 4).

Лемма 3. Пусть $y(\nu) = y_1(\nu) + iy_2(\nu)$ задает кусочно-целую кривую. Тогда во всякой точке υ отрезка $[0, 1]$, не совпадающей с точкой разрыва, справедливы оценки

$$|y_{1,2}^{(k)}(\upsilon)| \leqslant C^k, \quad \forall\, k \geqslant 1, \tag{2.8}$$

где $C > 0$ и не зависит от υ.

Доказательство. Неравенство Коши для коэффициентов степенного ряда дает

$$\frac{|y_1^{(k)}(\upsilon)|}{k!} \leqslant \frac{\max_{|\nu-\upsilon|=R} |y_1(\nu)|}{R^k} \leqslant \frac{\mathrm{const} \cdot e^{\beta R}}{R^k}, \quad \forall\, R > 0;$$

выбирая $R = k/\beta$ и применяя формулу Стирлинга, получаем (2.8). ■

Как и прежде, лемма о скелетонной аппроксимации в дальней зоне будет опираться на формулу Тейлора, но на этот раз для сложной функции $K(x, y(\nu))$. Для оценивания p-й производной такой функции нам потребуются вспомогательные построения.

Определение 3. Числом Стирлинга (второго рода) [4] будем называть функцию двух целых неотрицательных чисел, удовлетворяющую рекуррентному соотношению $\left\{ {n \atop k} \right\} = k\left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$ и начальным условиям $\left\{ {n \atop n} \right\} = 1$, $\left\{ {n \atop 0} \right\} = \delta_{n0}$. При $n < k$ положим $\left\{ {n \atop k} \right\} = 0$. „Комбинаторный“ смысл чисел Стирлинга таков: число способов разделить множество из n элементов на k непустых подмножеств равно $\left\{ {n \atop k} \right\}$.

Лемма 4. Величины $\alpha_n(\xi)$, $n = 0, 1, \dots$, определяемые по формуле $\alpha_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\} k!\, \xi^k$, где $\xi > 0$ произвольно, удовлетворяют неравенствам

$$\alpha_n \leqslant 2\, n!\, (2e\xi)^n.$$

Доказательство. Рассмотрим ряд $\sum_{n=0}^{\infty} \alpha_n\, \zeta^n/n!$, члены которого все положительны при $\zeta > 0$, и предположим его сходящимся для некоторого $\zeta_0 > 0$, возможно, зависящего от ξ. Тогда по теореме о повторных рядах

$$\sum_{n=0}^{\infty} \alpha_n \frac{\zeta_0^n}{n!} = \sum_{n=0}^{\infty} \sum_{k=0}^{n} \left\{ {n \atop k} \right\} \frac{\zeta_0^n}{n!} k!\, \xi^k = \sum_{k=0}^{\infty} \sum_{n=0}^{\infty} \left\{ {n \atop k} \right\} \frac{\zeta_0^n}{n!} k!\, \xi^k.$$

В силу равенства [4, с. 351]

$$(e^{\zeta} - 1)^k = k! \sum_{n=0}^{\infty} \left\{ {n \atop k} \right\} \frac{\zeta^n}{n!}, \qquad \forall\, \zeta \in \mathbb{C}$$

получаем

$$\sum_{n=0}^{\infty} \alpha_n \frac{\zeta_0^n}{n!} = \sum_{k=0}^{\infty} (e^{\zeta_0} - 1)^k\, \xi^k.$$

Последний ряд сходится при $(e^{\zeta_0}-1)\,\xi<1$; следовательно, исходный ряд сходится при том же условии и имеет ту же сумму. Нам будет достаточно следующей грубой оценки. Положим $\zeta_0 = \min\left(1,(2e\xi)^{-1}\right)$, тогда в силу неравенства $e^{\zeta}-1<\zeta e^{\zeta}$ общий член последнего ряда меньше 2^{-k}, и сумма ряда меньше 2. Поскольку общий член исходного ряда ограничен этой суммой, лемма доказана. ■

Перейдем к установлению оценок аппроксимации в дальней зоне, которую мы будем определять при помощи иерархического разбиения по лемме 2 какого-либо квадрата C_α с длиной стороны α, содержащего заданную кусочно-целую кривую S.

Лемма 5. Пусть ядро $K(x,y)$ асимптотически гладкое с параметрами c, d, g; $\eta\in C_\alpha$; узлы $\{y(j)\}_{j=1}^{n_y}\subset C_a(\eta)$ и $\{x(i)\}_{i=1}^{n_x}\subset\{x:\|x-y\|_\infty\geqslant\sigma a\quad\forall y\in C_a(\eta)\}$ лежат на кусочно-целой кривой, причем в квадрат $C_a(\eta)$ попадает не более J_0 точек разрыва, и $\sigma>\sigma_0=2deC/\mu$. Тогда $\forall\,p\in\mathbb{N},\ \ p\geqslant g,$ справедливо равенство

$$A\equiv[K\big(x(i),y(j)\big)]_{i=1}^{n_x}{}_{j=1}^{n_y}=T_p+R_p,$$

где

$$\operatorname{rank}T_p\leqslant(J_0+1)p,$$
$$\|R_p\|_F^2\leqslant 4c^2(J_0+1)^2n_xn_y\,(\sigma a)^{2g}\left(\frac{\sigma_0}{\sigma}\right)^{2p}.$$

Доказательство. Рассмотрим сначала какой-либо гладкий кусок контура в $C_a(\eta)$. Пусть υ — произвольная точка на соответствующем интервале I изменения параметра ν. Тогда $\forall\,\nu\in I$

$$K(x,y(\nu))=\sum_{k=0}^{p-1}\frac{D_\nu^kK(x,y(\nu))}{k!}\bigg|_{\nu=\upsilon}(\nu-\upsilon)^k+\frac{D_\nu^pK(x,y(\nu))}{p!}\bigg|_{\nu=\theta}(\nu-\upsilon)^p,$$

где $|\theta-\upsilon|<|\nu-\upsilon|$. Ранг матрицы, порожденной первым слагаемым, вне зависимости от n_x и n_y не превосходит p; перейдем к рассмотрению остатка $R(x,y)$. Прежде всего, по формуле конечных приращений имеем $a/2\geqslant\|y(\nu)-y(\upsilon)\|_\infty\geqslant\mu|\nu-\upsilon|$, так что

$$|R(x,y)|\leqslant\frac{|D_\nu^pK(x,y(\nu))|}{p!}\bigg|_{\nu=\theta}\left(\frac{a}{2\mu}\right)^p.$$

Далее, $\forall\, p \geqslant 1$ справедлива формула

$$D_\nu^p K(x, y(\nu)) = \sum_{k=1}^{p} \sum_{\nu} \Delta_y^\nu K(x,y) \sum_{\lambda}{}' y_{\nu_1}^{(\lambda_1)} \cdots y_{\nu_k}^{(\lambda_k)}, \tag{2.9}$$

где $\Delta_y^\nu = \frac{\partial^k}{\partial y_{\nu_1} \cdots \partial y_{\nu_k}}$; $\nu = (\nu_1, \ldots, \nu_k)$, $\lambda = (\lambda_1, \ldots, \lambda_k)$ — мультииндексы, подчиненные условиям

$$|\lambda| = p, \ \ \lambda_1, \ldots, \lambda_k \geqslant 1; \quad \nu_1, \ldots, \nu_k \in \{1, 2\}.$$

Штрих у внутренней суммы означает, что некоторые слагаемые в ней, возможно, повторяются; мы утверждаем лишь, что число слагаемых равно $\left\{ {p \atop k} \right\}$. Формула (2.9) легко доказывается индукцией по p, если воспользоваться рекуррентным соотношением для чисел Стирлинга.

Подставляя в (2.9) оценки лемм 3, 4, получаем

$$\begin{aligned} |D_\nu^p K(x, y(\nu))| &\leqslant \sum_{k=1}^{p} 2^k c d^k k!\, |x-y|^{g-k} \left\{ {p \atop k} \right\} C^p \\ &\leqslant c\, C^p |x-y|^g\, \alpha_p\!\left(\tfrac{2d}{|x-y|}\right) \leqslant 2cp!\, (4deC)^p\, |x-y|^{g-p} \end{aligned}$$

$\forall\, \nu \in I$ и, в частности, для $\nu = \theta$. Заменяя $|x-y|$ на σa, приходим к следующей оценке остатка:

$$|R(x,y)| \leqslant 2c\, \left(\frac{\sigma_0}{\sigma}\right)^p\, (\sigma a)^g. \tag{$**$}$$

Разобьем узлы $\{y(j)\}$ на $J_0 + 1$ групп, соответствующие гладким кускам кривой в квадрате $C_a(\eta)$, и переупорядочим $\{y(j)\}$ так, чтобы узлы каждой группы имели последовательные номера. Тогда при соответствующей перестановке столбцов матрицы A в ней возникнет $J_0 + 1$ блоков, каждый из которых может быть приближен матрицей ранга p с точностью, оцениваемой по $(**)$. Собирая скелетонные приближения каждого блока в матрицу T_p, получаем утверждение леммы. ■

Теорема 2. Пусть матрица A_n порождается асимптотически гладким ядром $K(x,y)$ на сетках $\{x(i)\}_{i=1}^n$, $\{y(j)\}_{j=1}^n$, заданных на кусочно-целой кривой S и подчиненных условию

$$\mu(\Gamma) \leqslant \tau \frac{\operatorname{mes} \Gamma}{\operatorname{mes} S} n, \qquad \forall\, \Gamma \subset S, \tag{2.10}$$

где Γ — кусок кривой S, ограниченный произвольным квадратом из иерархического разбиения. Тогда $\forall\,\varepsilon,\ 0<\varepsilon<\varepsilon_0,$ и $\forall\, n\in\mathbb{N}$, $n>n_0$, существуют мозаично-скелетонные аппроксимации $\tilde{A}_n$, для которых выполняются оценки

$$\begin{aligned}\mathrm{mr}\,\tilde{A}_n &< c_1\, J\log n\log\frac{1}{\varepsilon},\\ \|A_n-\tilde{A}_n\|_F &< c_2\, J^{3/2}\, n^{\gamma}\,\varepsilon,\end{aligned}$$

где

$$\gamma=\begin{cases}1, & \text{если } 1+2g>0,\\ |g|+\frac{1}{2}, & \text{если } 1+2g\leqslant 0.\end{cases}$$

В случае $1+2g\leqslant 0$ существуют также аппроксимации, удовлетворяющие оценкам

$$\begin{aligned}\mathrm{mr}\,\tilde{A}_n &< c_1\, J\log^2 n\log\frac{1}{\varepsilon},\\ \|A_n-\tilde{A}_n\|_F &< c_2\, J^{3/2}\, n\varepsilon.\end{aligned}$$

Доказательство отличается от приведенного выше для теоремы 1 тем, что квадраты одного уровня будут рассмотрены подробнее: далеко не все из них содержат кривую S. Оценим для начала число узлов $\{y(j)\}$ в l-м квадрате k-го уровня. Поскольку длина каждого гладкого куска кривой, ограниченного этим квадратом, ограничена по формуле $\mathrm{mes}\,\Gamma\leqslant\frac{\alpha\, 2^{-k}}{2\mu}\max_{\nu}\sqrt{y_1'(\nu)^2+y_2'(\nu)^2}$, то в силу (2.8)

$$n_{yl}(k)\leqslant(J+1)\hat{\tau}n2^{-k},\quad \text{где}\quad \hat{\tau}=\tau\,\frac{\alpha\, C}{\mu\ \mathrm{mes}\, S}.$$

Кроме того, $\sum_{l=1}^{2^{2k}} n_{yl}(k)=n$. Пусть $s=\lfloor\log\hat{\tau}n\rfloor$ — номер последнего уровня иерархического разбиения. Тогда число столбцов в l-м блоке ближней зоны (каждый из которых связан с одним из квадратов уровня s) не превосходит $n_{yl}(s)$; число строк $n_{xl}(s)\leqslant(2\sigma+1)^2\max_l n_{yl}(s)\leqslant 2(2\sigma+1)^2(J+1)$, и объем памяти, необходимый для хранения элементов всех блоков ближней зоны,

$$\begin{aligned}\mathrm{mem}\,(N)&\leqslant\sum_{l=1}^{2^{2s}} n_{xl}(s)n_{yl}(s)\leqslant\max_l n_{xl}(s)\sum_{l=1}^{2^{2s}} n_{yl}(s)\\ &\leqslant 2(2\sigma+1)^2(J+1)n.\end{aligned}$$

Перейдем к блокам дальней зоны. Число квадратов $(k+1)$-го уровня в дальней зоне квадрата k-го уровня $Q_k = \big((4\sigma{+}5)^2 - (2\sigma{+}1)^2\big)$, следовательно, число строк в блоке уровня k дальней зоны

$$n_{fxl}(k) \leqslant Q_k \max_l n_{yl}(k).$$

Поскольку каждый узел сетки попадает в Q_k (или меньшее количество) блоков уровня k, то

$$\sum_{l=1}^{2^{2k}} n_{fxl}(k) \leqslant Q_k n.$$

Пусть $\sigma = \lceil 2\sigma_0 \rceil$. Используя лемму 5, заключаем, что объем памяти для хранения аппроксимаций всех блоков дальней зоны

$$\begin{aligned} \mathrm{mem}\,(F) &\leqslant \sum_{k=1}^{s} \sum_{l=1}^{2^{2k}} (J+1)\, p_k \left(n_{fxl}(k) + n_{yl}(k)\right) \\ &\leqslant (J+1)\, n \sum_{k=1}^{s} (Q_k + 1) p_k \leqslant \mathrm{const} \cdot (J+1)\, \sigma^2 n \sum_{k=1}^{s} p_k; \end{aligned} \tag{2.11}$$

допускаемая при этом погрешность

$$\begin{aligned} \|A_n - \tilde{A}_n\|_F &\leqslant \sum_{k=1}^{s} \sum_{l=1}^{2^{2k}} 4c^2 (J+1)^2 n_{fxl}(k)\, n_{yl}(k)\, (\sigma a_k)^{2g} 2^{-2p_k} \\ &\leqslant \mathrm{const} \cdot (J+1)^2 \sigma^{2g} \sum_{k=1}^{s} \max_l n_{fxl}(k) 2^{-2kg} 2^{-2p_k} \sum_{l=1}^{2^{2k}} n_{yl}(k) \\ &\leqslant \mathrm{const} \cdot (J+1)^3 \sigma^{2+2g}\, n^2 \sum_{k=1}^{s} 2^{-k(1+2g)} 2^{-2p_k}. \end{aligned} \tag{2.12}$$

Ввиду совпадения оценок (2.11), (2.12) с (2.6), (2.7) при $m = 1$ с точностью до констант и такого же совпадения значений s будем считать доказательство законченным. ■

3. ОЦЕНКИ АСИМПТОТИЧЕСКОЙ ГЛАДКОСТИ

Нас будут интересовать оценки величин $|D_y^p K(x,y)|\,\big|_{y=\eta}$, $p \geqslant 0$, для конкретных примеров или классов ядер $K(x,y)$. Один из способов получения таких оценок связан с разложением $K(x,y)$ в степенной ряд в точке η.

Теорема 3. Пусть функция $K(x,y) : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$ — гармоническая по переменному y при $y \neq x$ и для некоторого g неравенство $|K(x,y)| \leqslant \text{const} \cdot |x-y|^g$ выполняется равномерно при $y \neq x$. Тогда $K(x,y)$ является асимптотически гладкой, именно, $\exists\, c > 0$ такое, что

$$|D_y^p K(x,y)| \leqslant c\,(2m)^{p/2}\, p^{\gamma}\, p!\, |x-y|^{g-p}, \quad \forall\, p \geqslant 0,$$

где

$$\gamma = \begin{cases} \frac{m}{2}, & \text{если } g > 0, \\ \frac{m}{2} + |g|, & \text{если } g \leqslant 0. \end{cases}$$

Доказательство. Для упрощения письма положим $\eta = 0$, поскольку $D_y^p K(x,y)|_{y=\eta} = D_\xi^p K(x, \xi+\eta)|_{\xi=0}$. В самом деле,

$$\tfrac{\partial}{\partial \xi_j} K(x, \xi+\eta) = e_j^T \nabla_y K(x,y)|_{y=\xi+\eta} = \tfrac{\partial}{\partial y_j} K(x,y)|_{y=\xi+\eta}.$$

При этом функция $K(x, \xi+\eta)$ остается гармонической по переменному ξ, $\forall\, \xi \neq x-\eta$. Как известно (см. раздел 4), функция $u(y)$, гармоническая в шаре $|y| < r_0$, разлагается в ряд вида $u(y) = \sum_{k=0}^{\infty} p_k(y)$, где $p_k(y)$ — однородный многочлен переменных y_1, $\ldots$, y_m степени k. Этот ряд сходится абсолютно и равномерно при $|y| \leqslant r$, $\forall\, r < r_0$. Коэффициенты многочленов p_k суть производные u, которые мы собираемся оценивать; иначе говоря, названный ряд можно трактовать как ряд Тейлора, в котором члены одной степени объединены. Естественно спросить, каким образом это объединение влияет на сходимость. Следующее утверждение, принадлежащее Хейману [5], дает вполне исчерпывающий ответ.

Теорема. *Пусть функция $u(y) : \mathbb{R}^m \mapsto \mathbb{R}$ удовлетворяет уравнению Лапласа в шаре $|y| < r_0$ и ни в каком шаре большего радиуса. Тогда m-кратный ряд Тейлора этой функции с центром в начале*

$$u(y) = \sum_{\nu} a_\nu\, y^\nu, \qquad a_\nu = \left.\frac{D^\nu u(y)}{\nu!}\right|_{y=0}, \tag{3.1}$$

где $D^\nu = \frac{\partial^{|\nu|}}{\partial y_1^{\nu_1} \cdots y_m^{\nu_m}}$, сходится абсолютно и равномерно при $|y| \leqslant r$, если $r < r_0/\sqrt{2}$. При условии $|y|\sqrt{2} < \rho < r_0$ общий член ряда допускает равномерную оценку

$$|a_\nu y^\nu| < \text{const} \cdot M_\rho\, p^{m/2} \left(\frac{|y|\sqrt{2}}{\rho} \right)^p, \tag{3.2}$$

где $p = |\nu|$, $M_\rho = \sup_{|y|=\rho} |u(y)|$. *Для некоторых функций указанного вида (например,* $u(y) = \Re(r_0 - y)^{-2}$, $y \in \mathbb{C}$*), существуют точки* y *на сфере радиуса* $r_0/\sqrt{2}$*, в которых ряд Тейлора расходится.* $\square$

Положим $u(y) = K(x, y + \eta)$, $r_0 = |x - \eta|$, $\rho = q r_0$, где $q \in (0, 1)$ пока произвольно. Тогда $\forall\, y : |y| = \rho$

$$\begin{aligned} |u(y)| &\leqslant \mathrm{const} \cdot (r_0 - \rho)^g = \mathrm{const} \cdot r_0^g \,(1-q)^g, \quad &\text{если } g \leqslant 0;\\ |u(y)| &\leqslant \mathrm{const} \cdot (r_0 + \rho)^g \leqslant \mathrm{const} \cdot r_0^g, \quad &\text{если } g > 0. \end{aligned}$$

Далее, пусть $y_1 = \dots = y_m = r/\sqrt{m}$, где r удовлетворяет неравенству $0 < r < \rho/\sqrt{2}$. Тогда в силу теоремы Хеймана ряд (3.1) сходится и, более того,

$$|a_\nu| < \mathrm{const} \cdot M_\rho \, p^{m/2} \,(2m)^{p/2} \, p! \, q^{-p} \, r_0^{-p}, \qquad p = |\nu|,$$

для любого допустимого q. Выбирая $q = 1 - 1/p$ для $p > 1$ и подставляя значение $r_0 = |x - \eta|$, получаем утверждение теоремы. $\blacksquare$

Для некоторых конкретных гармонических ядер результат теоремы 3 может быть существенно улучшен. Пусть, например, $x, y \in \mathbb{R}^2$, так что $K(x, y)$ является вещественной или мнимой частью регулярной функции, и производные этой функции легко вычисляются.

Предложение 1. Ядро $K(x, y) = \ln|x - y|$, где $x, y \in \mathbb{R}^2$, асимптотически гладкое, именно

$$|D_y^p K(x, y)| \leqslant (p-1)! \, |x - y|^{-p}, \qquad \forall\, p \geqslant 0.$$

Доказательство. Прежде всего, для ядер вида $\phi(y - x)$ формула (2.1) может быть переписана в виде

$$|D^p \phi(x)| \leqslant c \, d^p \, p! \, |x|^{g-p}, \qquad \forall\, p \geqslant 0,$$

в силу соотношения $D^p \phi(y - x) = (-1)^p \; D_\xi^p \phi(\xi)\big|_{\xi = y - x}$.

Рассмотрим функцию $\phi(z) = \ln\sqrt{x_1^2 + x_2^2}$ комплексного переменного $z = x_1 + i x_2$. Пусть $z_0 \neq 0$ фиксировано; проведем разрез от 0 до ∞ по лучу, не содержащему z_0 и не принадлежащему действительной оси. В полученной области выберем ту ветвь многозначной функции $\ln z$, которая совпадает с $\ln x_1$ при $z > 0$; обозначим ее

$f(z)$ и заметим, что $\Re f(z) = \phi(z)$. Пусть $D_1 = \partial/\partial x_1$, $D_2 = \partial/\partial x_2$. В силу условий Коши-Римана производная всякой регулярной функции $g(z) = u(z) + iv(z)$ в точке z_0 может быть вычислена по любой из следующих формул:

$$g' = D_1 u + iD_1 v, \tag{3.3a}$$

$$g' = D_2 v - iD_2 u. \tag{3.3b}$$

Продифференцируем p раз соотношение $f(z) = \phi(z) + i\psi(z)$, подставляя справа (3.3a) или (3.3b) так, чтобы конечная формула содержала $D^p\phi(z)$. Если общее число операторов D_2 в D^p равно k, то

$$(-1)^{\left\lfloor \frac{k+1}{2} \right\rfloor} D^p\phi(z)\Big|_{z=z_0} = \begin{cases} \Re f^{(p)}(z_0), & \text{если } k \text{ четно;} \\ \Im f^{(p)}(z_0), & \text{если } k \text{ нечетно.} \end{cases}$$

Следовательно, $|D^p\phi(z)| \leqslant |f^{(p)}(z_0)| = (p-1)!\,|z_0|^{-p}$; последнее потому, что формула $f^{(p)}(z) = (-1)^{p-1}(p-1)!\,z^{-p}$, верная при $z > 0$, стандартным образом может быть продолжена на всю рассматриваемую область. ■

Асимптотическая гладкость может быть также установлена при помощи подходящей замены координат, если градиент функции в новых переменных имеет простой вид. Например, для функции $\phi(x) = 1/|x|$ в двумерном случае естественно перейти к полярным координатам.

Предложение 2. Ядро $K(x, y) = 1/|x - y|$, где $x, y \in \mathbb{R}^2$, асимптотически гладкое, именно

$$|D_y^p K(x, y)| \leqslant 2^p\,p!\,|x - y|^{-p-1}, \qquad \forall\, p \geqslant 0.$$

Доказательство. Покажем сначала, что $D^p\frac{1}{r} = r^{-p-1}T_p(\phi)$, где $T_p(\phi)$ — некоторый тригонометрический многочлен от ϕ. По индукции получаем:

$$\begin{aligned} \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} D^p r^{-1} &= Q(\phi) \begin{bmatrix} 1 & 0 \\ 0 & r^{-1} \end{bmatrix} \begin{bmatrix} D_r \\ D_\phi \end{bmatrix} r^{-p-1}T_p(\phi) \\ &= Q(\phi) \begin{bmatrix} (-p-1)\,T_p(\phi) \\ T_p'(\phi) \end{bmatrix} r^{-p-2}, \end{aligned}$$

где $Q(\phi) = \left[\begin{smallmatrix}\cos\phi & -\sin\phi\\ \sin\phi & \cos\phi\end{smallmatrix}\right]$. То же самое может быть записано в „комплексной“ форме:

$$\begin{aligned} T_1(\phi) &= \pi_1(-e^{i\phi});\\ T_{p+1}(\phi) &= \pi_{p+1}\bigl(e^{i\phi}((-p-1)\,T_p + iT_p')\bigr). \end{aligned}$$

Символ $\pi_k(z)$ обозначает взятие вещественной или мнимой части $z \in \mathbb{C}$, на выбор (который обеспечивает нужную общность символа D^p).

Далее, запишем

$$T_{p+1}^{(q)} = \pi_{p+1}\left(e^{i\phi}\sum_{j=0}^{q+1}\left|\begin{matrix}q+1\\ j\end{matrix}\right|_p T_p^{(j)}\right), \qquad \forall\, q \geqslant 0, \tag{3.4}$$

и постараемся определить *квазибиномиальные* коэффициенты $\left|\begin{smallmatrix}n\\ k\end{smallmatrix}\right|_p$ так, чтобы равенство (3.4) оказалось верным. При $q = 0$ имеем $\left|\begin{smallmatrix}1\\ 0\end{smallmatrix}\right|_p = -p-1$, $\left|\begin{smallmatrix}1\\ 1\end{smallmatrix}\right|_p = i$. Потребуем, чтобы $\left|\begin{smallmatrix}n\\ k\end{smallmatrix}\right|_p = 0$ при $n < k$ или $k < 0$; тогда (3.4) доказывается индукцией по q при условии, что

$$\left|\begin{matrix}q+1\\ j\end{matrix}\right|_p = \left|\begin{matrix}q\\ j-1\end{matrix}\right|_p + i\left|\begin{matrix}q\\ j\end{matrix}\right|_p, \qquad j = 0, \ldots, q+1.$$

Наконец, из формул $T_1^{(q)} = \pi_1(-i^q e^{i\phi})$ и (3.4) индукцией по p следует, что

$$T_p^{(q)}(\phi) = a_p b_p^q\, \Pi_p(\phi) \qquad \forall\, p, q \geqslant 0, \tag{3.5}$$

где a_p, b_p — некоторые числа, а $\Pi_p(\phi)$ — тригонометрический многочлен, такой, что $|\Pi_p(\phi)| \leqslant 1 \quad \forall\,\phi$. В самом деле, $\Pi_1 = \pi_1(-e^{i\phi})$, $a_1 = 1$, $b_1 = i$, так что базис индукции выполнен; рассмотрим переход от шага p к шагу $p+1$.

В силу (3.4) и (3.5) имеем

$$\Pi_{p+1} = \pi_{p+1}(e^{i\phi}\Pi_p), \qquad T_{p+1}^{(q)} = a_p\, S_{q,p}(b_p)\, \Pi_{p+1},$$

где

$$S_{q,p}(z) \equiv \sum_{j=0}^{q+1}\left|\begin{matrix}q+1\\ j\end{matrix}\right|_p z^j = (z+i)^q S_{0,p}(z) = (z+i)^q(zi-p-1)$$

в силу рекуррентных соотношений для квазибиномиальных коэффициентов. Тем самым индуктивный шаг сделан, причем $a_{p+1} = a_p\,(b_p i - p - 1)$, $b_{p+1} = b_p + i$. Учитывая значения a_1, b_1, получаем $b_p = pi$, $a_p = (-1)^{p-1}(2p-1)!!$, так что (3.5) можно переписать в виде

$$T_p^{(q)}(\phi) = (-1)^{p-1}(2p-1)!!\,p^q i^q\,\Pi_p(\phi), \tag{3.5'}$$

откуда следует оценка $|T_p(\phi)| \leqslant (2p-1)!!$ и доказываемое утверждение. ■

4. АППРОКСИМАЦИЯ ДЛЯ ГАРМОНИЧЕСКИХ ЯДЕР. Если ядро $K(x, y)$ удовлетворяет условиям теоремы 3, то ε-ранг матрицы, порожденной этим ядром на паре отделенных сеток, можно оценить в соответствии с леммой 1. Между тем справедлив более сильный результат.

Лемма 6. Пусть функция $K(x, y) : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$ — гармоническая по переменному y при $y \neq x$ и для некоторого g неравенство $|K(x, y)| \leqslant \mathrm{const} \cdot |x - y|^g$ выполняется равномерно при $y \neq x$. Пусть узлы $\{y(j)\}_{j=1}^{n_y}$ сосредоточены в кубе

$$C_a(\eta) = \{y : \|y - \eta\|_\infty \leqslant a/2\};$$

$\{x(i)\}_{i=1}^{n_x}$ находятся в дальней зоне этого куба, $\{x(i)\}_{i=1}^{n_x} \subset F_a(\eta)$, причем $\sigma = \lceil 2\sqrt{m}\rceil$. Тогда $\forall\, p \in \mathbb{N}$, $p \geqslant 2(m-1)$, справедливо равенство

$$A \equiv [K\big(x(i), y(j)\big)]_{i=1}^{n_x}{}_{j=1}^{n_y} = T_p + R_p,$$

где

$$\begin{aligned} &\operatorname{rank} T_p \leqslant 2p^{m-1}, \\ &\|R_p\|_F^2 \leqslant \mathrm{const} \cdot n_x n_y\, m^g a^{2g}\, p^{2(m-1)}\, 2^{-2p}. \end{aligned} \tag{4.1}$$

Доказательство. Поскольку оператор Лапласа и условия на сетки инвариантны относительно сдвига начала координат, без потери общности можно положить $\eta = 0$.

Отправной точкой в наших рассуждениях будет представление $K(x, y)$ в виде *интеграла Пуассона* [5], справедливое $\forall\, y : |y| < r < \sigma a$. Именно,

$$K(x, y) = \frac{1}{c_m} \int\limits_{|\xi| = r} \frac{r^2 - |y|^2}{r|\xi - y|^m}\, K(x, \xi)\, d\sigma_\xi, \tag{4.2}$$

где $d\sigma_\xi$ — элемент площади поверхности сферы $|\xi| = r$ и $c_m = 2\pi^{m/2}/\Gamma(m/2)$. Непосредственным дифференцированием устанавливается, что ядро интеграла Пуассона

$$\mathcal{P}(\xi, y) = \frac{r^2 - |y|^2}{r|\xi - y|^m}$$

удовлетворяет уравнению Лапласа относительно переменного y, $\forall\, y \neq \xi$. Обозначим $|y| = \rho$, $t = \rho/r < 1$, $\xi^T y = r\rho\cos\gamma$ и перепишем $\mathcal{P}(\xi, y)$ в виде

$$\mathcal{P}(\xi, y) = \frac{1 - t^2}{r^{m-1}(1 - 2t\cos\gamma + t^2)^{m/2}}. \tag{4.3}$$

Разделение переменных ξ и y в формуле (4.3) возможно при помощи *ультрасферических многочленов*, называемых также многочленами Гегенбауэра [2, 7]. Эти многочлены имеют параметр $\lambda \in \mathbb{R}$ и могут быть определены рекуррентными соотношениями

$$\begin{aligned} &kP_k^{(\lambda)}(z) = 2\,(k+\lambda-1)zP_{k-1}^{(\lambda)}(z) \,-\, (k+2\lambda-2)P_{k-2}^{(\lambda)}(z);\\ &P_0^{(\lambda)}(z) = 1, \qquad P_1^{(\lambda)}(z) = 2\lambda z, \end{aligned} \tag{4.4}$$

или своей производящей функцией

$$\left(1 - 2tz + t^2\right)^{-\lambda} = \sum_{k=0}^{\infty} P_k^{(\lambda)}(z)\, t^k, \tag{4.5}$$

которой мы и собираемся воспользоваться. Ряд в правой части (4.5) сходится абсолютно и равномерно по $z : |z| \leqslant 1$ и $|t| \leqslant t_0 < 1$, что позволяет подставить его в (4.2), используя (4.3) и положив $\lambda = m/2$, $z = \cos\gamma$.

Относительно k-го члена этого ряда необходимо сделать два замечания. Во-первых, это однородная функция переменных $y_1, \ldots, y_m$ измерения k, что проверяется непосредственно. Во-вторых, с точностью до множителя, не зависящего от y,

$$P_k^{(\lambda)}(\cos\gamma)\, t^k \;=\; \sum_{j=0}^{k} a_{jk} \left(\frac{\sum_{i=1}^{m} \xi_i y_i}{\rho r}\right)^j \rho^k,$$

где a_{jk} — коэффициенты ультрасферического многочлена степени k. Из соотношений (4.4) легко следует, что $a_{jk} \neq 0$ лишь при четных значениях $k-j$; стало быть, k-*й член ряда (4.5) есть однородный многочлен переменных* $y_1, \ldots, y_m$ *степени* k.

То же самое верно для общего члена ряда

$$r^{m-1}\,\mathcal{P}(\xi, y) = \sum_{k=0}^{\infty} p_k(\cos\gamma)\,t^k, \tag{4.6}$$

где $p_k(z) = P_k^{(m/2)}(z) - P_{k-2}^{(m/2)}(z)$. В силу леммы Бернштейна об абсолютной величине производной алгебраического многочлена [7, с. 19], и леммы 7 последний ряд допускает дифференцирование по переменным $y_1, \ldots, y_m$. Следовательно, равенство

$$0 = \sum_{k=0}^{\infty} \Delta_y\bigl(p_k(\cos\gamma)t^k\bigr)$$

имеет место $\forall\, y, \xi:\ \rho < r$. Поскольку оператор Лапласа переводит однородный многочлен степени k в однородный многочлен степени $k-2$, индукцией по k получаем, что $p_k(\cos\gamma)\,t^k$ — гармонические по y. Более того, гармоническими оказываются также многочлены $P^{(m/2)}(\cos\gamma)\,t^k$, что нам потребуется в разделе 5.

Теперь мы можем оценить ранг вырожденного ядра T_p, получающегося при интегрировании первых p членов ряда (4.6), по формуле (4.1). В самом деле, в обозначениях, применявшихся при доказательстве леммы 1, однородный многочлен P_k степени k определяется $\mu_{km} = \binom{k+m-1}{m-1}$ коэффициентами. Оператор Лапласа переводит такой многочлен в однородный многочлен P_{k-2} степени $k-2$, коэффициенты которого суть линейные комбинации коэффициентов P_k. Соответствующая матрица имеет $\mu_{k-2,m}$ строк, которые в совокупности линейно независимы, потому что прообраз любого ненулевого P_{k-2} (строящийся конструктивно последовательным интегрированием в духе метода прогонки) есть однородный многочлен степени k. Поскольку линейная независимость набора многочленов эквивалентна линейной независимости набора векторов соответствующих коэффициентов, число линейно независимых гармонических однородных многочленов степени k совпадает с размерностью ядра упомянутой выше матрицы и, таким образом, равно $\mu_{km} - \mu_{k-2,m}$. Окончательно

$$\operatorname{rank} T_p \leqslant \sum_{k=0}^{p-1} (\mu_{km} - \mu_{k-2,m}) = \sum_{k=0}^{p-1} \tbinom{k+m-2}{m-2} + \tbinom{k+m-3}{m-2}$$
$$= \tbinom{p+m-2}{m-1} + \tbinom{p+m-3}{m-1} = \frac{p^{\overline{m-1}} + (p-1)^{\overline{m-1}}}{(m-1)!} \leqslant 2p^{m-1}.$$

Все преобразования биномиальных коэффициентов были сделаны с помощью дополнения по нижнему индексу и определяющего эти коэффициенты рекуррентного соотношения.

Для оценки остаточного члена нам потребуются две следующие леммы.

Лемма 7 (ср. [7, с. 178]). При $z \in [-1, 1]$

$$|P_k^{(\lambda)}(z)| \leqslant \binom{2\lambda + k - 1}{k}, \qquad \forall\, k \geqslant 0. \tag{4.7}$$

Доказательство. Положим $z = \cos\gamma$, $\gamma \in \mathbb{R}$. Общий член ряда

$$(1 - te^{i\gamma})^{-\lambda} = \sum_{k=0}^{\infty} \binom{\lambda + k - 1}{k} t^k e^{ik\gamma} \tag{4.8}$$

по абсолютной величине не превосходит соответствующего члена ряда

$$(1 - t)^{-\lambda} = \sum_{k=0}^{\infty} \binom{\lambda + k - 1}{k} t^k \tag{4.9}$$

при условии $0 < t < 1$. В силу тождества

$$1 - 2tz + t^2 = (1 - te^{i\gamma})(1 - te^{-i\gamma})$$

ряд (4.5) можно рассматривать как почленное произведение двух абсолютно сходящихся рядов типа (4.8), сгруппированное по степеням t. Применяя неравенство треугольника, получаем, что общий член (4.5) по абсолютной величине мажорируется общим членом ряда (4.9) с увеличенным вдвое значением λ. Последний ряд также следует трактовать как почленное произведение (4.9) на себя, сгруппированное по степеням t. ■

Лемма 8. При $t \in (0,1)$, $p,\ m \in \mathbb{N}$, $p \geqslant m$

$$\sum_{k=p}^{\infty} t^k\, k^{\underline{m}} \leqslant m!\, \frac{t^p}{1-t} \left(\frac{t}{1-t} + \frac{p}{m} \right)^m$$

Доказательство. Пусть $z = t/(1-t)$, тогда

$$\sum_{k=p}^{\infty} t^k\, k^{\underline{m}} = t^m \sum_{k=p}^{\infty} t^{k-m}\, k^{\underline{m}} = t^m \sum_{k=p}^{\infty} D_t^m\, t^k = t^m D_t^m \frac{t^p}{1-t} =$$

$$t^m \sum_{k=0}^{m} \tbinom{m}{k}\, p^{\underline{m-k}}\, t^{p-m+k}\, k!\, (1-t)^{-1-k} = m!\, \frac{t^p}{1-t} \sum_{k=0}^{m} \tbinom{p}{m-k}\, z^k.$$

Далее,

$$\sum_{k=0}^{m} \tbinom{p}{k}\, z^{m-k} = \sum_{k=0}^{m} \tbinom{m}{k}\, z^{m-k}\, \frac{p^{\underline{k}}}{m^{\underline{k}}} \leqslant \sum_{k=0}^{m} \tbinom{m}{k}\, z^{m-k} \frac{p^k}{m^k}. \quad \blacksquare$$

Теперь мы готовы к доказательству второй из формул (4.1). Объединяя (4.2) и (4.6), получаем

$$K(x,y) = \frac{1}{c_m r^{m-1}} \int\limits_{|\xi|=r} \sum_{k=0}^{\infty} p_k(\cos\gamma)\, t^k\, K(x,\xi)\, d\sigma_\xi,$$

откуда

$$|R_p(x,y)| \leqslant \sum_{k=p}^{\infty} t^k\, |p_k(\cos\gamma)|\, \max_{|\xi|=r} |K(x,\xi)|. \tag{4.10}$$

Начнем с последнего члена. Наименьший радиус шара, содержащего узлы $\{y(j)\}$, $\rho = a\sqrt{m}/2$. Далее, узлы $\{x(i)\}$ содержатся в области $(\sigma + \frac{1}{2})a \leqslant \|x\|_\infty \leqslant (2\sigma + \frac{5}{2})a$, поэтому при $t = \frac{1}{2}$

$$(\sigma + \tfrac{1}{2} - \sqrt{m})a \leqslant |x - \xi| \leqslant (2\sigma + \tfrac{5}{2} + \sqrt{m})a,$$

так что $|K(x,\xi)| \leqslant \mathrm{const} \cdot m^{g/2} a^g$.

Наконец, в силу лемм 7 и 8

$$\sum_{k=p}^{\infty} t^k\,|p_k(\cos\gamma)| \leqslant \frac{2t^{1-m}}{(m-1)!}\sum_{k=p}^{\infty} t^{k+m-1}(k+m-1)^{\underline{m-1}}$$
$$\leqslant 2mt^{1-m}\frac{t^{p+m-1}}{1-t}\left(\frac{t}{1-t}+\frac{p+m-1}{m-1}\right)^{m-1}$$
$$\leqslant \mathrm{const}\cdot 2^{-p}\left(2+\frac{p}{m-1}\right)^{m-1}. \quad \blacksquare$$

Теорема 4. Пусть матрица A_n порождается гармоническим по y ядром $K(x,y)$ на сетках $\{x(i)\}_{i=1}^n$, $\{y(j)\}_{j=1}^n$, заданных на ограниченном множестве $S\subset\mathbb{R}^m$ и подчиненных условию (2.3). Тогда $\forall\,\varepsilon,\ 0<\varepsilon<\varepsilon_0$, и $\forall\, n\in\mathbb{N},\ n>n_0$, существуют мозаично-скелетонные аппроксимации $\tilde A_n$, для которых

$$\begin{aligned} \mathrm{mr}\,\tilde A_n &< c_1\log n\,\log^{m-1}\frac{1}{\varepsilon}, \\ \|A_n-\tilde A_n\|_F &< c_2\, n^{\gamma}\,\varepsilon, \end{aligned} \tag{4.11}$$

где

$$\gamma=\begin{cases} 1, & \text{если } m+2g>0,\\ \frac{|g|}{m}+\frac{1}{2}, & \text{если } m+2g\leqslant 0.\end{cases}$$

Доказательство мало чем отличается от доказательства теоремы 1. Применяя лемму 6, получаем оценки

$$\mathrm{mem}\,(F)\leqslant \mathrm{const}\cdot\sum_{k=1}^{s} p^{m-1},$$

$$\|A_n-\tilde A_n\|_F^2\leqslant \mathrm{const}\cdot n^2\sum_{k=1}^{s}2^{-k(m+2g)}\,2^{-2p}\,p^{2(m-1)}.$$

Выберем теперь p так, чтобы оказалось верным неравенство $2^{-p}p^{m-1}\leqslant\varepsilon$. В силу леммы 9, это возможно при $p\leqslant \mathrm{const}\cdot\log\frac{1}{\varepsilon}$. $\blacksquare$

Лемма 9. Минимальное $p\in\mathbb{N}$, удовлетворяющее неравенству $2^{-p}\,p^{\kappa}\leqslant\varepsilon$, где $\kappa>0$, $\varepsilon\in(0,1)$, допускает оценку

$$p\leqslant\begin{cases} \mathrm{const}\cdot\left(\kappa\log\kappa+\log\frac{1}{\varepsilon}\right), & \text{если } \kappa\geqslant 2;\\ \mathrm{const}\cdot\left(\kappa+\log\frac{1}{\varepsilon}\right), & \text{если } \kappa<2.\end{cases} \tag{4.12}$$

Доказательство. Перейдем к эквивалентному неравенству

$$p \geqslant \kappa \log(ap), \qquad (***)$$

где $a = \varepsilon^{-1/\kappa} \geqslant 1$, и заметим, что в силу выпуклости логарифмической функции уравнение относительно $p \in \mathbb{R}$, $p \geqslant 1$ вида $ap = a\kappa \log(ap)$ либо не имеет решений и неравенство $(***)$ выполняется строго $\forall\, p \geqslant 1$, либо существует не более двух решений и интересующее нас $p \in \mathbb{N}$ оценивается по формуле $p \leqslant p_0 + 1$, где p_0 — большее из решений. Возведя уравнение в квадрат и поделив на $ap(a\kappa)^2$, получаем

$$\frac{ap}{(a\kappa)^2} = \frac{\log^2(ap)}{ap}.$$

Поскольку $\log^2 x/x < 2 \quad \forall\, x \geqslant 1$, приходим к неравенству $ap_0 < 2(a\kappa)^2$, которое после логарифмирования и подстановки в уравнение дает

$$p_0 < \kappa\left(1 + 2\log\kappa + 2\log\varepsilon^{-1/\kappa}\right). \quad \blacksquare$$

5. АППРОКСИМАЦИЯ ДЛЯ ОСЦИЛЛЯЦИОННЫХ ЯДЕР. Нас будут интересовать оценки мозаично-скелетонной аппроксимации для фундаментальных решений уравнения Гельмгольца в двумерном и трехмерном случаях, и в частности зависимость мозаичного ранга аппроксимации от волнового числа.

Пусть $x, y \in \mathbb{R}^m$, $|x| = r$, $|y| = \rho$, $x^T y = r\rho\cos\gamma$. Тогда справедливы следующие *теоремы сложения* [6, 10], связываемые с именами Сонина, Неймана, Гегенбауэра и Графа:

$$H_0^{(1)}(|x-y|) = \sum_{k=0}^{\infty} J_k(\rho)\, H_k^{(1)}(r)\, T_k(\cos\gamma), \quad m = 2; \qquad (5.1)$$

$$\frac{e^{i|x-y|}}{|x-y|} = i\pi \sum_{k=0}^{\infty} (k + \tfrac{1}{2})\, \frac{J_{k+\frac{1}{2}}(\rho)}{\sqrt{\rho}}\, \frac{H^{(1)}_{k+\frac{1}{2}}(r)}{\sqrt{r}}\, P_k(\cos\gamma), \quad m = 3. \qquad (5.2)$$

Здесь $J_k(z)$ — функции Бесселя; $H_k^{(1)}(z)$ — функции Ганкеля 1-го рода; $P_k(z) = P_k^{(1/2)}(z)$ — многочлены Лежандра; $T_k(z) = \lim_{\lambda\to 0} \frac{n}{2\lambda} P_k^{(\lambda)}(z)$ — многочлены Чебышёва. Ряды (5.1), (5.2) сходятся абсолютно и равномерно при $\rho \leqslant r_0 < r$.

Если произвести замену переменных $x = \tilde{x}\kappa$, $y = \tilde{y}\kappa$, где $\kappa > 0$ — волновое число, в левых частях формул (5.1), (5.2) возникают фундаментальные решения уравнения Гельмгольца; условия сходимости рядов в правых частях, очевидно, сохраняются.

Нам потребуются оценки остаточных членов разложений (5.1), (5.2), справедливые для *произвольного* положительного r. Если бы волновое число было фиксированным, мы могли бы при помощи линейной замены перевести область изменения переменных в окрестность нуля и воспользоваться асимптотическими представлениями типа

$$J_\nu(z) = \left(\frac{z}{2}\right)^\nu \frac{1 + \mathcal{O}(1)}{\Gamma(\nu+1)}, \qquad \nu \to \infty,$$

вытекающими из разложений функций Бесселя и Ганкеля в степенные ряды, или более сложными оценками [10, гл. VIII]. К сожалению, мы не нашли в литературе оценок этих функций, справедливых равномерно для всех положительных значений аргумента. Собственные усилия привели к следующим двум леммам.

Лемма 10. Пусть $\nu \geqslant 0$. Тогда

$$|J_\nu(\rho)| \leqslant \frac{1}{\Gamma(\nu+1)} \left(\frac{\rho}{2}\right)^\nu, \qquad \forall\, \rho > 0.$$

Доказательство сводится к грубой оценке модуля подынтегральной функции *представления Пуассона* [6] для функций Бесселя:

$$J_\nu(z) = \frac{(z/2)^\nu}{\sqrt{\pi}\,\Gamma(\nu+\frac{1}{2})} \int_{-1}^{1} (1-t^2)^{\nu-\frac{1}{2}} \cos zt\, dt.$$

Именно, $|\cos \rho t| \leqslant 1$ и

$$\int_0^1 (1-t^2)^{\nu-\frac{1}{2}}\, dt = \tfrac{1}{2}\,\mathrm{B}\left(\tfrac{1}{2}, \nu + \tfrac{1}{2}\right).$$

Осталось применить формулу дополнения для бета-функции. ■

Лемма 11. Пусть $\nu \geqslant 0$. Тогда

$$|H_\nu^{(1)}(r)| \leqslant \mathrm{const} \cdot \left(\frac{2}{r}\right)^k k!\, k e^r/\sqrt{r}, \qquad \forall\, r > 0,$$

где $k = \lceil \nu - \frac{1}{2} \rceil$.

Доказательство. Воспользуемся интегральным представлением Пуассона

$$H_\nu^{(1)}(z) = \sqrt{\frac{2}{\pi z}} \frac{e^{i(z-\pi\nu/2-\pi/4)}}{\Gamma\left(\nu+\frac{1}{2}\right)} \int_0^\infty e^{-t} t^{\nu-\frac{1}{2}} \left(1+\frac{it}{2z}\right)^{\nu-\frac{1}{2}} dt.$$

Поскольку при $\nu \geqslant \frac{1}{2}$

$$\left|\left(1+\frac{it}{2r}\right)^{\nu-\frac{1}{2}}\right| \leqslant \left(1+\frac{t}{2r}\right)^{\nu-\frac{1}{2}} \leqslant \sum_{j=0}^{k} \binom{k}{j} \frac{t^j}{(2r)^j} \tag{5.3}$$

и $\int_0^\infty e^{-t} t^{\nu-\frac{1}{2}+j}\, dt = \Gamma\left(\nu+j+\frac{1}{2}\right)$, имеем

$$|H_\nu^{(1)}(r)| \leqslant \sqrt{\frac{2}{\pi r}} \sum_{j=0}^{k} (\nu+\tfrac{1}{2})^{\bar{j}} \binom{k}{j} \tfrac{1}{(2r)^j} \leqslant \sqrt{\frac{2}{\pi r}}\ p_k\left(\tfrac{1}{2r}\right),$$

где $p_k(z) = \sum_{j=0}^{k} \binom{k}{j} (k+1)^{\bar{j}} z^j$. Если $\nu < \frac{1}{2}$, левая часть (5.3) оценивается единицей, так что выписанная оценка справедлива и в этом случае. Дело сводится к изучению значений многочленов $p_k(z)$ при $z>0$. Чтобы завершить доказательство, достаточно применить лемму 12. ■

Лемма 12. Многочлены $p_n(z) = \sum_{k=0}^{n} \binom{n}{k} (n+1)^{\bar{k}} z^k$ допускают оценку

$$p_n(z) \leqslant \mathrm{const} \cdot (4z)^n n!\, n e^{\frac{1}{2z}}, \qquad \forall\, z > 0.$$

Доказательство. Для сумм n членов *гипергеометрического типа*, то есть сумм вида $T_n = \sum_{k=0}^{n} t(n,k)$, где

$$t(n,k) = p(n,k) \frac{(\alpha_1 n + \beta_1 k + \gamma_1)! \cdots (\alpha_p n + \beta_p k + \gamma_p)!}{(a_1 n + b_1 k + c_1)! \cdots (a_q n + b_q k + c_q)!}\, w^n z^k,$$

где $p(n,k)$ — многочлен по n и k, величины α_j, β_j, a_j, b_j — целые константы, γ_j и c_j — произвольные константы, существует алгоритмизуемый метод получения коротких рекуррентных соотношений вида $\sigma_0(n)T_n + \sigma_1(n)T_{n+1} + \cdots + \sigma_l(n)T_{n+l} = 0$, где l не зависит от n, принадлежащий Госперу и Цайльбергеру [4, с. 230]. Не имея достаточно места для детального изложения, приведем конечный результат для сумм, которые определяют интересующие нас многочлены.*

$$\begin{aligned}
&p_0(z) = 1,\\
&p_1(z) = 1 + 2z,\\
&p_n(z) = 2z\,(2n-1)p_{n-1}(z) + p_{n-2}(z), \qquad n \geqslant 2.
\end{aligned} \tag{5.4}$$

Введем производящую функцию

$$\mathfrak{P}(w, z) = \sum_{n=0}^{\infty} p_n(z) \frac{w^n}{n!} \tag{5.5}$$

и будем обозначать дифференцирование по переменному w штрихом. Тогда, предполагая абсолютную и равномерную в области $|w| < \varepsilon$ сходимость ряда (5.5), запишем

$$\mathfrak{P}' = \sum_{n=1}^{\infty} p_n(z) \frac{w^{n-1}}{(n-1)!}, \quad \mathfrak{P}'' = \sum_{n=2}^{\infty} p_n(z) \frac{w^{n-2}}{(n-2)!}.$$

Подставляя в последний ряд уравнение (5.4), получаем

$$\begin{aligned}
\mathfrak{P}'' &= 2z \sum_{n=2}^{\infty} (2(n-2)+3) p_{n-1} \frac{w^{n-2}}{(n-2)!} + \mathfrak{P}\\
&= 4z \sum_{n=3}^{\infty} p_{n-1} \frac{w^{n-2}}{(n-3)!} + 6z \sum_{n=2}^{\infty} p_{n-1} \frac{w^{n-2}}{(n-2)!} + \mathfrak{P}\\
&= 4zw\mathfrak{P}'' + 6z\mathfrak{P}' + \mathfrak{P}.
\end{aligned}$$

Введем переменное $\nu = 1 - 4zw$ и запишем дифференциальное уравнение, которому удовлетворяет $\mathfrak{P}$, в виде

$$\nu P'' - 6zP' - P = 0.$$

Это уравнение однородно относительно P и его производных; заменой $P' = PQ$ оно сводится к следующему уравнению Риккати:

$$\nu(Q^2 + Q') - 6zQ = 1. \tag{5.6}$$

*Читатель, знакомый с системой Mathematica, сможет получить (5.4) при помощи компьютера.

Поиск частного решения в виде степенной функции переменного ν или линейной комбинации таких функций приводит к ответу $Q_1 = \nu^{-1/2} + 2z\nu^{-1}$. Достаточно проверить формулы

$$Q_1^2 = \nu^{-1} + 4z\nu^{-3/2} + 4z^2\nu^{-2},$$
$$Q_1' = 2z\nu^{-3/2} + 8z^2\nu^{-2}.$$

В силу начальных условий для $\mathfrak{P}$

$$\mathfrak{P}(0) = p_0 = 1,$$
$$\mathfrak{P}'(0) = p_1 = 1 + 2z$$

решение, соответствующее Q_1, есть $\mathfrak{P}$; таким образом, решение однородного уравнения, соответствующего (5.6), не требуется. Далее, уравнение $P' = PQ_1$ имеет общее решение $P = \frac{C}{\sqrt{\nu}}e^{-\sqrt{\nu}/(2z)}$. Подставляя начальные условия, получаем формулу

$$\mathfrak{P} = \frac{1}{\sqrt{\nu}}e^{\frac{1-\sqrt{\nu}}{2z}}.$$

Нам достаточно оценить производные $\mathfrak{P}$, как функции переменного ν, в единице. В окрестности единицы выделим регулярную ветвь $\mathfrak{P}(\nu)$, соответствующую главной ветви $\sqrt{\nu}$. В силу формулы Коши

$$e^{1/(2z)}\pi_n \equiv D_\nu^n\mathfrak{P}(1) = \frac{n!}{2\pi i}\int\limits_{|\nu-1|=\varepsilon}\frac{\mathfrak{P}(\nu)\,d\nu}{(\nu-1)^{n+1}}. \qquad (5.7)$$

При $\nu = 1 + \varepsilon e^{i\phi}$ имеем

$$e^{-1/(2z)}\mathfrak{P}(\nu) = \sum_{k=0}^{\infty}\left(-\tfrac{1}{2z}\right)^k\frac{\nu^{(k-1)/2}}{k!} = \sum_{k,m=0}^{\infty}\left(-\tfrac{1}{2z}\right)^k\binom{\frac{k-1}{2}}{m}\varepsilon^m e^{im\phi}/k!$$

Подставив это разложение в (5.7), получим

$$\pi_n = n!\sum_{k=0}^{\infty}\left(-\tfrac{1}{2z}\right)^k\binom{\frac{k-1}{2}}{n}\frac{1}{k!}.$$

Теперь мы можем получить оценку на радиус сходимости ряда

(5.5). В самом деле, при $|\zeta| < 1$

$$\sum_{n=0}^{\infty} \frac{\pi_n}{n!}\,\zeta^n = \sum_{k=0}^{\infty} \left(-\tfrac{1}{2z}\right)^k \frac{1}{k!} \sum_{n=0}^{\infty} \binom{\frac{k-1}{2}}{n} \zeta^n$$
$$= \sum_{k=0}^{\infty} \left(-\tfrac{1}{2z}\right)^k (1+\zeta)^{(k-1)/2}/k! = \frac{1}{\sqrt{1+\zeta}}\, e^{-\sqrt{1+\zeta}/(2z)}.$$

Заметим, что все члены ряда $\sum_{n=0}^{\infty} \frac{\pi_n}{n!}\,\zeta^n$ положительны при $\zeta < 0$; это следует из формул $p_n(z) > 0$ и $p_n(z) = (-4z)^n e^{1/(2z)} \pi_n$. Положим $\zeta = -1 + \varepsilon$ и, по аналогии с леммой 4, оценим общий член ряда его суммой:

$$|\pi_n| \leqslant n! \left(1 + \frac{\varepsilon}{1-\varepsilon}\right)^n \frac{1}{\sqrt{\varepsilon}}\, e^{-\sqrt{\varepsilon}/(2z)}.$$

Выбирая $\varepsilon = 1/n^2$ при $n > 1$, получаем оценку

$$e^{1/(2z)} |\pi_n| \leqslant \mathrm{const} \cdot n!\, n e^{(1-\frac{1}{n})/(2z)}$$

и утверждение леммы. ■

Собирая вместе результаты лемм 10, 11 и 8, получаем

Следствие. При $r = 2\rho$ остатки рядов (5.1), (5.2), начинающиеся с индекса $k = p$, при $p \geqslant m$ удовлетворяют неравенствам

$$\begin{aligned} |R_p(x,y)| &\leqslant \mathrm{const} \cdot p\, 2^{-p} e^r/\sqrt{r}, \qquad m = 2; \\ |R_p(x,y)| &\leqslant \mathrm{const} \cdot p^2\, 2^{-p} e^r/r, \qquad m = 3. \end{aligned} \tag{5.8}$$

Лемма 13. Пусть $x, y \in \mathbb{R}^m$, $m = 2, 3$, и

$$K(x,y) = \begin{cases} H_0^{(1)}(\kappa|x-y|), & m = 2; \\ e^{i\kappa|x-y|}/|x-y|, & m = 3. \end{cases} \tag{5.9}$$

Пусть $\kappa > 0$; узлы $\{y(j)\}_{j=1}^{n_y}$ сосредоточены в кубе $C_a(\eta)$; узлы $\{x(i)\}_{i=1}^{n_x}$ находятся в дальней зоне этого куба, $\{x(i)\}_{i=1}^{n_x} \subset F_a(\eta)$, причем $\sigma = 2$. Тогда $\forall\, p \in \mathbb{N}$, $p \geqslant m$, справедливо равенство

$$A \equiv [K\big(x(i), y(j)\big)]_{i=1\,j=1}^{n_x\ n_y} = T_p + R_p,$$

где

$$
\begin{aligned}
&\operatorname{rank} T_p \leqslant 2p^{m-1},\\
&\|R_p\|_F^2 \leqslant \mathrm{const}\cdot n_x n_y\, f(\kappa, a)\, p^{2(m-1)}\, 2^{-2p},\\
&f(\kappa, a) = \begin{cases} d^{2\kappa a}\,(\kappa a)^{-1}, & m = 2,\\ d^{2\kappa a}\, a^{-2}, & m = 3;\end{cases}\\
&d \leqslant e^{12}.
\end{aligned}
\tag{5.10}
$$

Доказательство. Ранг вырожденного ядра, соответствующего первым p членам разложений (5.1), (5.2), оценивается по первой формуле (5.10) благодаря тому, что $P_k(\cos\gamma)\rho^k$ есть гармонический однородный многочлен переменных $\{y_j\}_{j=1}^m$ (см. лемму 6).

Оценка остаточного члена производится следующим образом. Сдвигом начала координат в точку η обеспечиваем неравенства

$$
\left(\sigma+\tfrac{1}{2}\right)a \leqslant \|x\|_\infty \leqslant \left(2\sigma+\tfrac{5}{2}\right)a, \qquad \forall\, x \in F_a(0).
$$

Положим $\rho = a\frac{\sqrt{m}}{2}$, тогда $C_a(0) \in \{y : \|y\|_2 \leqslant \rho\}$ и $\|y\|_2 \leqslant \frac{1}{2}\|x\|_2$ $\forall\, y \in C_a(0)$, $x \in F_a(0)$; таким образом, применимы формулы (5.8). Делая в последних замену $r \to \kappa r$, $\rho \to \kappa\rho$ и полагая $d = \exp\left(\sqrt{m}(2\sigma+\frac{5}{2})\right)$, имеем $e^{\kappa r} \leqslant d^{\kappa a}$, $r \geqslant (\sigma+\frac{1}{2})\, a$ и вторую формулу (5.10). ■

Теорема 5. Пусть матрица A_n порождается ядром $K(x, y)$, заданным формулой (5.9), на сетках $\{x(i)\}_{i=1}^n$, $\{y(j)\}_{j=1}^n$, принадлежащих ограниченному множеству $S \subset \mathbb{R}^m$, $m = 2, 3$, и подчиненных условию (2.3). Пусть $\alpha_E = \kappa\alpha$, где α — диаметр множества S. Тогда $\forall\, \varepsilon$, $0 < \varepsilon < \varepsilon_0$, и $\forall\, n \in \mathbb{N}$, $n > n_0$, существуют мозаично-скелетонные аппроксимации $\tilde{A}_n$, для которых

$$
\begin{aligned}
&\operatorname{mr} \tilde{A}_n < c_1 \left(\alpha_E + \log \frac{1}{\varepsilon\sqrt{\alpha_E}}\right) \log n, \qquad m = 2,\\
&\operatorname{mr} \tilde{A}_n < c_1 \left(\alpha_E + \log \frac{1}{\varepsilon}\right)^2 \log n, \qquad m = 3;\\
&\|A_n - \tilde{A}_n\|_F < c_2\, n\varepsilon.
\end{aligned}
\tag{5.11}
$$

Доказательство. Следуя идее доказательства теоремы 1 и применяя лемму 13, получаем

$$\mathrm{mem}\,(F) \leqslant \mathrm{const} \cdot \sum_{k=1}^{s} p^{m-1};$$

$$\|A_n - \tilde{A}_n\|_F^2 \leqslant \mathrm{const} \cdot n^2 \sum_{k=1}^{s} 2^{-km}\, f(\kappa, \alpha\, 2^{-k})\, 2^{-2p}\, p^{2(m-1)}.$$

Последняя сумма имеет, по существу, одинаковый вид вне зависимости от m. Поскольку показательная функция монотонна, эту сумму можно интерпретировать как нижнюю сумму Дарбу и оценить ее через соответствующий интеграл. Именно, пусть $a = 2\alpha_E \ln d$, $\xi_k = 2^{-k}$, $\Delta\xi_k = \xi_{k-1} - \xi_k$. Тогда

$$\sum_{k=1}^{s} 2^{-k} e^{a2^{-k}} = \sum_{k=1}^{s} \Delta\xi_k\, e^{a\xi_k} \leqslant \int_0^1 e^{a\xi}\, d\xi = \frac{e^a - 1}{a}.$$

Выберем теперь p так, чтобы оказалось верным одно из неравенств

$$p^2\, 2^{-2p}/\alpha_E \leqslant \frac{a\varepsilon^2}{e^a - 1}, \qquad m = 2,$$

$$p^4 2^{-2p} \leqslant \frac{a\varepsilon^2}{e^a - 1}, \qquad m = 3,$$

которые обеспечивают последнюю оценку в (5.11). В силу леммы 9 и неравенства $e^a - 1 < ae^a$, справедливого при $a > 0$, это возможно при

$$p \leqslant \begin{cases} \mathrm{const} \cdot \left(\alpha_E + \log \dfrac{1}{\varepsilon\sqrt{\alpha_E}}\right), & m = 2, \\ \mathrm{const} \cdot \left(\alpha_E + \log \dfrac{1}{\varepsilon}\right), & m = 3, \end{cases}$$

откуда следует искомая оценка на мозаичный ранг. ■

Литература

1. Brandt A. Multilevel computations of integral transforms and particle interactions with oscillatory kernels // *Computer Physics Communications.* 1991. V. 65. P. 24–38.

2. Геронимус Я. Л. *Теория ортогональных многочленов.* — М.-Л.: ГИТТЛ, 1950.

3. Goreinov S. A., Tyrtyshnikov E. E., Yeremin A. Yu. Matrix-Free Iterative Solution Strategies for Large Dense Linear Systems // *Numerical Linear Algebra with Applications*. 1997. V. 4(4). P. 273–294.

4. Graham R. L., Knuth D. E., Patashnik O. *Concrete Mathematics.* 2nd edition. — Addison-Wesley, 1994.

5. Hayman W. K. Power series expansions for harmonic functions // *The Bulletin of the London Mathematical Society.* 1970. V. 2. P. 152–158.

6. Никифоров А. Ф., Уваров В. Б. *Основы теории специальных функций.* — М.: Наука, 1974.

7. Сегё Г. *Ортогональные многочлены.* — М.: Физматлит, 1962.

8. Tyrtyshnikov E. E. Mosaic-skeleton approximations // *Calcolo.* 1996. V. 33 (1-2). P. 47–57.

9. Тыртышников Е. Е. Методы быстрого умножения и решение уравнений // *Настоящий сборник.*

10. Watson G. N. *A treatise on the theory of Bessel functions.* 2nd edition. — Cambridge University Press, 1992.

Использование методов быстрого матричного умножения при решении интегральных уравнений теории потенциала*

М. С. Мартынов

Аннотация

Во многих задачах могут возникать линейные системы со специфическими, пусть даже плотными, матрицами, которые с помощью специальных средств могут быть умножены на вектор за число операций меньшее, чем $O(n^2)$. *В данной работе рассмотрены подходы к использованию мультипольного метода быстрого умножения для решения задач, возникающих в теории потенциала и в термальном анализе. Вначале изложены основные принципы мультипольного подхода, затем приведены непосредственно алгоритмы и оценки точности и эффективности различных модификаций мультипольного метода.*

1. ДЛЯ ЧЕГО НУЖНО БЫСТРОЕ МАТРИЧНОЕ УМНОЖЕНИЕ? Многие итерационные методы решения линейных систем вида $Ax = b$, например *обобщенный метод минимальных невязок* (GMRES), используют подпространства Крылова

$$K^i(A, \phi) \equiv span\{A^{i-1}\phi, ..., A\phi, \phi\},$$

где ϕ — некоторый (произвольный) ненулевой вектор. На каждой i-й итерации метод добавляет к имеющейся системе векторов новый вектор вида $A^i\phi$. Таким образом, на каждой итерации требуется выполнить одно умножение матрицы A на вектор.

---

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155) и фонда Volkswagen-Stiftung (грант № VW I/71 493).

Одно умножение $n \times n$-матрицы A на вектор b производится за $O(n^2)$ операций (строго говоря, за $n(2n-1)$ операций). При больших размерах матрицы A желание производить ее умножение на вектор за меньшее число операций вполне оправданно. Особенно, если матрица настолько велика, что не может поместиться целиком в оперативную память компьютера, и ее приходится сохранять на жестком диске и при каждом новом умножении считывать. Очевидно, что в общем случае оценка $O(n^2)$ не улучшаема. Однако во многих приложениях число операций имеет вид $o(n^2)$. Оно обеспечивается *методами быстрого матричного умножения.* Следует отметить, что метод быстрого умножения, о котором пойдет речь, — приближенный: он позволяет умножать матрицу на вектор с заданной точностью, от которой существенно зависит число выполняемых при умножении арифметических действий.

В данной работе рассмотрены подходы к использованию *мультипольного метода быстрого умножения* для решения задач, возникающих в теории потенциала и в термальном анализе. Вначале изложены основные принципы мультипольного подхода, затем приведены непосредственно алгоритмы и оценки точности и эффективности различных модификаций мультипольного метода. Далее описаны задачи и выведены формулы метода уже для конкретных задач. Приведены данные численных экспериментов, сравнение их с предварительными оценками и выводы о целесообразности использования метода для тех или иных задач.

2. МУЛЬТИПОЛЬНОЕ И ЛОКАЛЬНОЕ РАЗЛОЖЕНИЯ. Мультипольный подход был впервые описан В. Рохлиным (V. Rokhlin) в работе [1]. В этой статье шла речь о кулоновском взаимодействии в двумерном пространстве. Поэтому при изложении общей теории метода для наглядности будет использоваться „электрическая“ терминология, хотя, в принципе, метод применим для вычислений любых взаимодействий, обладающих свойствами, о которых будет рассказано ниже.

Пусть мы имеем дело с двумя типами объектов — зарядами и точками. Под зарядом мы подразумеваем источник некоторого взаимодействия, которое будем называть потенциалом. Заряд обозначим буквой q, и будем считать, что это обозначение включает как величину заряда (интенсивность взаимодействия), так и его пространственные координаты. Точкой будем считать просто точку в пространстве, обозначать ее будем буквой p, подразумевая под этим все координаты точки. Потенциал, создаваемый зарядом q в точке p, обозначим $K(q,p)$. Рассмотрим следующее расположение заряда и точки: пусть заряд q находится близко к началу координат (точке O) и расстояние от него до начала координат r_q, а точка p находится далеко от начала координат, на расстоянии r_p, таком, что $r_q \ll r_p$. Предположим наличие у потенциала $K(q,p)$ следующего свойства — пусть при данном условии ($r_q \ll r_p$) его можно представить в виде ряда

$$K(q,p) = \sum_{i=0}^{\infty} t_i^M(q) T_i^M(p),$$

где $\{t_i^M(q)\}_{i=0}^{\infty}$ — последовательность функций, зависящих только от параметров заряда, а $\{T_i^M(p)\}_{i=0}^{\infty}$ — последовательность функций, зависящих только от параметров точки.

Пусть данный ряд сходится следующим образом:

$$\varepsilon \equiv |K(q,p) - \sum_{i=0}^{k} t_i^M(q) T_i^M(p)| \leqslant c \left(\frac{r_q}{r_p}\right)^k,$$

где c — постоянная, $c < \infty$ и не зависит от других величин. Теперь, пусть у нас есть n зарядов, лежащих внутри сферы радиуса r_q. Каждый из них создает в точке p потенциал $K(q_j,p)$, $j = \overline{1,n}$, который с любой требуемой точностью можно представить в виде

$$K(q_j,p) \approx \sum_{i=0}^{k} t_i^M(q_j) T_i^M(p).$$

Тогда суммарный потенциал в точке p (обозначим его $\psi(p)$) будет выражаться следующим образом:

$$\psi(p) = \sum_{i=0}^{\infty} \Big(\sum_{j=0}^{n} t_i^M(q_j)\Big) T_i^M(p).$$

Обозначив $M_i \equiv \sum_{j=0}^{n} t_i^M(q_j)$, получим

$$\psi(p) = \sum_{i=0}^{\infty} M_i T_i^M(p) \approx \sum_{i=0}^{k} M_i T_i^M(p).$$

Это означает, что последовательность коэффициентов $\{M_j\}_{j=0}^{k}$ в совокупности с последовательностью функций $\{T_i^M(p)\}_{i=0}^{\infty}$ с требуемой точностью определяет потенциал, создаваемый данным множеством зарядов в любой точке, для которой выполняется условие $r_p \gg r_q$. Такая совокупность зарядов называется мультиполем и, в связи с этим, ряд вида $\psi(p) = \sum_{i=0}^{\infty} M_i T_i^M(p)$ называется *мультипольным разложением*, а M_i — *коэффициентами мультипольного разложения* или просто *мультипольными коэффициентами.* Для реализации мультипольного метода коэффициенты мультипольного разложения должны обладать двумя важными свойствами. Одно мы приведем прямо сейчас, другое будет описано несколько позже.

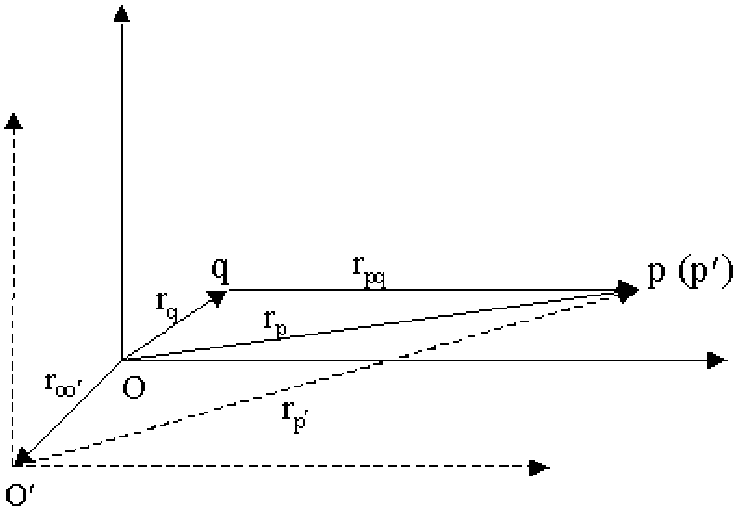


Рис. 1. Малое смещение начала координат

Предположим, что начало координат из точки O переносится в точку O' так, что расстояние $r_{oo'}$ между O и O' удовлетворяет условию $r_{oo'} + r_q \ll r_p{}'$. Тогда все заряды оказываются внутри сферы радиуса $r_q{}'$ с центром в новом начале координат. Так как $r_q{}' \leqslant r_{oo'} + r_q \ll r_p{}'$, то для нового расположения зарядов относительно точки O' существует мультипольное разложение потенциала в точке p с коэффициентами $\{M_i{}'\}_{i=0}^{\infty}$. При этом $\sum_{i=0}^{\infty} M_i T_i^M(p) = \sum_{i=0}^{\infty} M_i' T_i^M(p')$, где под p' подразумеваются новые координаты точки p.

Важное свойство мультипольных коэффициентов, необходимое для применения мультипольного метода, состоит в следующем: *в рассмотренном выше случае перемещения начала координат* *существует последовательность коэффициентов* $\{a_j^i\}_{i,j=0}^{\infty}$, *зависящих только от координат* O *и* O′ *и таких, что* $M_i{}' = \sum_{j=0}^{\infty} a_j^i M_j$. *При этом очевидно, что если* $\psi(p) = \sum_{i=0}^{k} M_i T_i^M(p)$, *то* $M_i' = \sum_{j=0}^{k} a_j^i M_j$.

Введем понятие *локального разложения.* Пусть теперь точка находится близко к началу координат, а заряд, наоборот, удален от него, то есть $r_q \gg r_p$. Пусть в этом случае существует ряд, сходящийся к значению потенциала в точке p:

$$K(q,p) = \sum_{i=0}^{\infty} t_i^L(q) T_i^L(p),$$

причем $|K(q,p) - \sum\limits_{i=0}^{k} t_i^L(q) T_i^L(p)| \leqslant c\left(\frac{r_p}{r_q}\right)^k$

Тогда в случае, когда у нас имеется n зарядов, таких, что $r_{q_j} \gg r_p$, $j = \overline{1,n}$, потенциал, создаваемый ими в точке p, вычисляется как

$$\psi(p) = \sum_{i=0}^{\infty} (\sum_{j=0}^{n} t_i^L(q_j)) T_i^L(p).$$

Введем следующее обозначение: $L_i = \sum_{j=0}^{n} t_i^L(q_j)$. Тогда $\psi(p) = \sum_{i=0}^{\infty} L_i T_i^L(p)$. Это значит, что, зная коэффициенты $\{L_i\}_{i=0}^{\infty}$, мы можем с любой заданной точностью определить значение потенциала, создаваемого удаленными зарядами в любой точке внутри сферы радиуса r_p с центром в начале координат. Данная сумма отражает распределение потенциала внутри некоторой области — локальное распределение, вследствие чего она названа *локальным разложением*, а коэффициенты $\{L_i\}_{i=0}^{\infty}$ — *коэффициентами локального разложения*, или *локальными коэффициентами.*

Локальные коэффициенты должны обладать свойством, аналогичным описанному выше свойству для мультипольных коэффициентов. Предположим, что начало координат сместилось из точки O в точку O′, причем $r_{oo'} + r_p \ll r_q$, то есть $r_{p'} \ll r_q$. Тогда потенциал в точке p (в новых координатах p′) выражается следующим образом:

$$\psi(p) = \sum_{i=0}^{\infty} L_i T_i^L(p) = \sum_{i=0}^{\infty} L_i' T_i(p').$$

Свойство локальных коэффициентов, нужное для построения мультипольного метода, состоит в следующем: *существует множество коэффициентов* $\{a_j^i\}_{i,j=0}^{\infty}$*, зависящих только от координат точки* O' *относительно точки* O*, таких, что* $L_i' = \sum\limits_{j=0}^{\infty} a_j^i L_j$*. При этом, если* $\psi(p) = \sum_{i=0}^{k} L_i T_i^L(p)$*, то* $L_i' = \sum_{j=0}^{k} a_j^i L_j$.

Введем теперь второе из упоминавшихся свойств локальных и мультипольных коэффициентов. Рассмотрим некоторое множество зарядов, находящееся внутри сферы радиуса r_q с центром в точке Q, удаленной от начала координат на расстояние r_{oq}, и некоторое множество точек, лежащих внутри сферы радиуса r_p с центром в начале координат. При этом $r_q \ll r_{oq}$ и $r_p \ll r_{oq}$. Мультипольные коэффициенты, вычисленные для потенциала, создаваемого множеством зарядов, характеризуют поле на достаточно большом расстоянии от точки Q, а локальные коэффициенты, вычисленные для сферы, содержащей точки, характеризуют поле внутри нее, создаваемое удаленными от нее зарядами. Тогда поле в точке p, лежащей возле начала координат, можно выразить следующим образом:

$$\psi(p) = \sum_{i=0}^{\infty} L_i T_i^L(p) = \sum_{i=0}^{\infty} M_i T_i^M(p).$$

Пусть при этом мультипольные и локальные коэффициенты связаны соотношением $L_i = \sum_{j=0}^{\infty} a_j^i M_j$*, где* $\{a_j^i\}_{i,j=0}^{\infty}$ *— коэффициенты, зависящие только от координат точки* Q *относительно точки* O*. При этом, если* $\psi(p) = \sum_{i=0}^{k} M_i T_i^M(p)$*, то* $L_i = \sum_{j=0}^{k} a_j^i M_j$.

Теперь все основные предположения относительно локальных и мультипольных разложений сделаны, и можно переходить непосредственно к описанию основных принципов мультипольного подхода.

3. ОСНОВЫ МУЛЬТИПОЛЬНОГО МЕТОДА. МЕДЛЕННЫЙ МУЛЬТИПОЛЬНЫЙ АЛГОРИТМ. Итак, пусть имеется кластер зарядов, укладывающийся в сферу радиуса r_q с центром в точке Q, и кластер точек, лежащий внутри сферы радиуса r_p с центром в точке P, причем расстояние между P и Q $r_{pq} \gg r_p + r_q$.

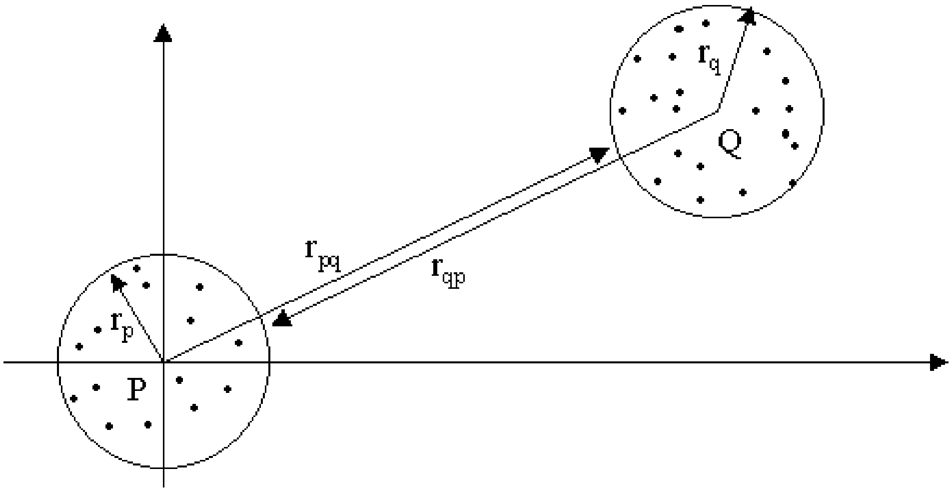


Рис. 2. Кластеры зарядов и точек

Предположим, что мы хотим приближенно определить поле в точках с некоторой заданной точностью ε. Так как для точности мультипольного разложения справедливо соотношение $\varepsilon \leqslant c(\frac{r_q}{r_{pq}})^k$, где k — число членов, до которого мы „урезаем“ мультипольное разложение, то возьмем $k \simeq \log_{(\frac{r_q}{r_{pq}})} \varepsilon$ и для каждого заряда вычислим k коэффициентов мультипольного разложения относительно точки Q. Далее просуммируем полученные коэффициенты и получим k мультипольных коэффициентов для потенциала, создаваемого всем кластером зарядов. Теперь применим к ним преобразование $\{M_i\}_{i=0}^{\infty} \to \{L_i\}_{i=0}^{\infty}$ и получим локальные коэффициенты для потенциала внутри сферы, ограничивающей кластер точек. При этом для произвольной точки кластера

$$|\psi(p) - \sum_{i=0}^{k} L_i T_i^L(p)| \leqslant c_1 \left(\frac{r_p}{r_{pq}}\right)^k + c_2 \left(\frac{r_q}{r_{qp}}\right)^k,$$

где r_{pq} — расстояние между точкой P и сферой с зарядами, а r_{qp} — расстояние между точкой Q и сферой с точками.

Таким образом, вычислив один раз k мультипольных коэффициентов для кластера зарядов, мы можем вычислять потенциал в любой достаточно удаленной от него точке с любой заданной точностью, причем, возможно, это будет осуществляться быстрее, чем вычисление при точном суммировании потенциалов отдельных зарядов. Рассмотренный принцип и лежит в основе мультипольного подхода.

Теперь будем рассматривать уже не абстрактный потенциал, а линейно зависящий от величины заряда: $\psi(y) = q\ K(x,y)$, где x, y — координаты, соответственно, заряда и точки, а q — величина заряда. Пусть у нас есть n зарядов величины $q_i,\ i = \overline{1,n}$, с пространственными координатами $x_i,\ i = \overline{1,n}$, и n точек с координатами $y_j,\ j = \overline{1,n}$, распределенных в некоторой области Ω. Тогда потенциал в i-й точке выражается следующей формулой:

$$\psi(y_i) = \sum_{j=0}^{n} q_j K(x_j, y_i).$$

Определим „вектор потенциалов“, i-м элементом которого является значение потенциала в i-й точке. Теперь операцию вычисления потенциалов в n точках можно представить в виде умножения матрицы на вектор:

$$A\bar{q} = \bar{\psi},$$

где

$$A = \begin{pmatrix} K(x_1,y_1) & \dots & K(x_n,y_1) \\ \vdots & K(x_j,y_i) & \vdots \\ K(x_1,y_n) & \dots & K(x_n,y_n) \end{pmatrix}, \quad \bar{q} = \begin{pmatrix} q_1 \\ \vdots \\ q_i \\ \vdots \\ q_n \end{pmatrix}, \quad \bar{\psi} = \begin{pmatrix} \psi(p_1) \\ \vdots \\ \psi(p_i) \\ \vdots \\ \psi(p_n) \end{pmatrix}.$$

Таким образом, если известна матрица A, потенциал в n точках вычисляется за $O(n^2)$ операций. Оказывается, что, используя мультипольный подход, число вычислений можно сократить. Приступим к описанию медленного мультипольного метода. Заключим область Ω в прямоугольник (или в куб, если рассматривается трехмерное пространство). Затем разобьем этот прямоугольник на $N \times M$ одинаковых прямоугольников (здесь N и M — число отрезков, на которые разбиты соответственно горизонтальные и вертикальные стороны прямоугольника). Будем считать каждый прямоугольник мультиполем и вычислим для него коэффициенты мультипольного разложения до k-го порядка. Для вычисления потенциала в нужных точках, для каждого содержащего эти точки прямоугольника следует найти локальные коэффициенты до

k-го порядка. Потенциал, вычисленный таким образом, является приближенным, и его точность $\varepsilon \leqslant c \cdot \max((\frac{a}{r_i})^k)$, $i = \overline{1, N \times M - 1}$, где a — диагональ прямоугольника, r_i — кратчайшее расстояние от прямоугольника с точками до i-го прямоугольника с зарядами. Очевидно, что для хорошей сходимости следует учитывать мультипольное воздействие только тех прямоугольников, для которых $a < r_i$. Потенциал, создаваемый остальными прямоугольниками, лучше считать точно. Таким образом, появляется понятие области взаимодействия для прямоугольника. Так как мы хотим определять потенциал с помощью мультипольного подхода, рассматривая каждый прямоугольник как мультиполь, то следует при этом учитывать, что потенциал, создаваемый прямоугольниками, не принадлежащими области взаимодействия, все равно приходится вычислять по точной формуле и потом прибавлять полученный результат к уже подсчитанному с помощью локальных коэффициентов. В случае, если прямоугольник разбит на квадраты (или на „почти квадраты“, т. е. $(a \leqslant 2a_x) \wedge (a \leqslant 2a_y)$, где a_x и a_y — размеры прямоугольника), зона взаимодействия для одного квадрата — это все остальные квадраты, не являющиеся его соседями (не имеющие с ним общих сторон) и не являющиеся его вторичными соседями (соседями его соседей). При этом для точности справедливо следующее ограничение: $\varepsilon \leqslant c(\frac{1}{2})^k$. То есть, если требуется точность ε, то k следует взять из соотношения $k \sim \log_2 \varepsilon^{-1}$. Медленный мультипольный алгоритм состоит в следующей последовательности действий:

1. Разбиение области на прямоугольники.

2. Вычисление мультипольных коэффициентов $\{M_i\}_{i=0}^k$ для каждого прямоугольника.

3. Вычисление коэффициентов локального разложения потенциала, создаваемого зарядами, находящимися в области взаимодействия, для каждого прямоугольника, с помощью преобразования $\{M_i\}_{i=0}^k \to \{L_i\}_{i=0}^k$.

4. Вычисление потенциала в точках с помощью локального разложения.

5. Вычисление в точках потенциала, создаваемого зарядами, не лежащими в зоне взаимодействия (ближними зарядами).

6. Суммирование потенциалов, вычисленных на этапах 4 и 5.

Теперь оценим число операций, производимых при умножении матрицы размера $n \times n$ на вектор длины n с помощью медленного мультипольного алгоритма. Будем вычислять количество операций соответственно приведенным выше этапам алгоритма.

1. Этот этап будем считать не требующим операций, аналогично тому, как мы не учитываем операции, производимые во время генерации матрицы при точном умножении. Можно считать этот этап предварительным.

2. Будем считать, что известны коэффициенты разложения для единичных зарядов, находящихся в точках расположения реальных зарядов. Тогда для вычисления мультипольных коэффициентов для прямоугольника нужно умножить коэффициенты для единичных зарядов на величины соответствующих зарядов и полученные коэффициенты сложить. Получается nk умножений и nk сложений — всего $2nk$ операций.

3. Трансляция $\{M_i\}_{i=0}^k \to \{L_i\}_{i=0}^k$ производится за $\sim k^2$ операций. Для одного прямоугольника максимальное количество прямоугольников в зоне взаимодействия $(N \cdot M - 9)$ (случай углового прямоугольника), следовательно, для одного прямоугольника производится $\sim (NM - 9)$ трансляций. Так как имеется $N \times M$ прямоугольников, то соответственно имеем уже $\sim NM(NM - 9)k^2$ операций.

4. Вычисление потенциала в одной точке $\sim k$ операций. При этом не учитываются затраты на вычисления значений функций $\{T_i^L(y_j)\}_{i=0,j=0}^{k,n}$, которые тоже можно отнести к предварительному этапу — будем считать, что времена предварительного этапа и генерации матрицы одного порядка. В этом случае для n точек имеем $\sim nk$ операций.

5. Максимальное для одного прямоугольника число прямоугольников, не принадлежащих области взаимодействия, — 25, включая его самого. Если заряды распределены равномерно, то число зарядов в одном прямоугольнике $\sim \frac{n}{NM}$ при условии $NM < n$. Значит, для одной точки число операций на 5-м этапе $\sim \frac{25n}{NM}$, а для n точек $\sim \frac{25n^2}{NM}$.

Общее число операций оценивается следующим образом:

$$\text{oper} \approx \frac{25n^2}{NM} + 3nk + NM(NM-9)k^2 \sim \frac{n^2}{NM} + nk + (NM)^2k^2.$$

Следует учитывать, что эта оценка справедлива только при $n > NM > 9$. N и M следует выбирать, минимизируя данную оценку с учетом указанных ограничений. Оптимизируя полученное выражение по NM, получим $NM \sim (\frac{n}{k})^{\frac{2}{3}}$, то есть $\text{oper} \sim n^{\frac{4}{3}}k^{\frac{2}{3}}$. Следовательно, при больших n можно выбрать $NM \sim (\frac{n}{k})^{\frac{2}{3}}$ и, таким образом, уменьшить число операций с $O(n^2)$ до $O(n^{\frac{4}{3}}k^{\frac{2}{3}})$ или, выражая то же самое через точность ε, $O(n^{\frac{4}{3}}\log_2^{\frac{2}{3}}\varepsilon^{-1})$.

Итак, мы выписали все этапы *медленного мультипольного алгоритма* и даже получили оценку для количества операций, выполняемых при его использовании. Следует отметить, что алгоритм не требует хранить всю матрицу $n \times n$ для своей работы. Объем памяти, требующийся алгоритму, следует оценивать в соответствии с его реализацией, однако проще получить верхнюю оценку:

1. $2MN \cdot k$ — память для хранения мультипольных и локальных коэффициентов.

2. $(MN)^2 \cdot k^2$ — память для хранения трансляционных коэффициентов.

3. nk — память под значения функций $\{T_i^L(y_j)\}_{i=0,j=0}^{k,n}$ (см. оценку числа операций для 4-го этапа алгоритма).

4. nk — память под хранение мультипольных коэффициентов для единичных зарядов с координатами реальных зарядов.

5. $\frac{25n^2}{NM}$ — память для хранения элементов матрицы, соответствующих точным вычислениям в мультипольном методе.

Суммируя, получаем

$$\text{mem} \leqslant \frac{25n^2}{NM} + 2nk + (NM)^2 \cdot k^2 + 2NM \cdot k.$$

Если брать $NM \sim (\frac{n}{k})^{\frac{2}{3}}$, то оценка для памяти сведется к следующему:

$$\text{mem} \leqslant 26n^{\frac{4}{3}}k^{\frac{2}{3}} + nk + 2n^{\frac{2}{3}}k^{\frac{1}{3}},$$

то есть $mem \sim n^{\frac{4}{3}} k^{\frac{2}{3}}$, что, вообще говоря, и следовало ожидать, исходя из оценки для количества операций. Очевидно, объем памяти можно уменьшить за счет некоторого увеличения числа операций. Здесь все зависит от имеющихся ресурсов и, как уже упоминалось, реализации алгоритма. Еще раз отметим, что заполнение описанной выше памяти производится до начала работы алгоритма и соответствует процессу генерации всей матрицы при обычном умножении.

Почему приведенный алгоритм был назван *медленным*, станет ясно, когда будет описан *быстрый мультипольный алгоритм.*

4. БЫСТРЫЙ МУЛЬТИПОЛЬНЫЙ АЛГОРИТМ. Основное различие между быстрым и медленным мультипольными алгоритмами можно упрощенно выразить в следующем: если в медленном мультипольном алгоритме все прямоугольники имеют одинаковый размер, то в быстром используются прямоугольники разных размеров. Приведем пример. Пусть используется медленный мультипольный алгоритм. Предположим, что мы вычисляем локальное разложение для одного прямоугольника, преобразовывая в коэффициенты локального разложения мультипольные коэффициенты четырех удаленных прямоугольников. Предположим также, что эти четыре прямоугольника расположены так, что могут быть объединены в один большой прямоугольник. Если его диагональ равна b, диагональ маленького прямоугольника равна a, а кратчайшее расстояние между ними r, и для них выполняется соотношение $a + b \ll r$, то, значит, мы можем применить мультипольный подход к этим двум прямоугольникам и преобразовывать в коэффициенты локального разложения уже мультипольные коэффициенты одного прямоугольника, а не четырех, сократив, таким образом, число операций в четыре раза. Данный пример, конечно, слишком груб и не учитывает массы факторов, о которых сейчас пойдет речь, но достаточно четко поясняет разницу между быст-

рым и медленным алгоритмами.

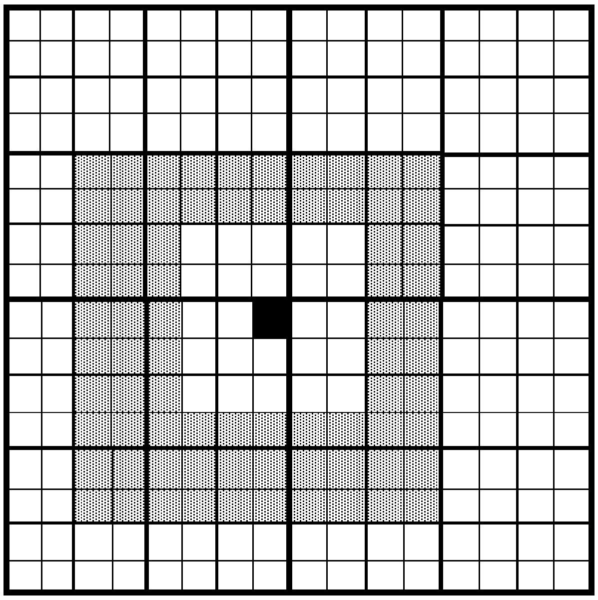


Рис. 3. Область взаимодействия (для черного прямоугольника)

Итак, введем следующее разбиение прямоугольной области. Разобьем область на ν прямоугольников, каждый из них разобьем еще на ν прямоугольников, и так λ раз. Число λ назовем числом уровней. В результате получается система вложенных прямоугольников, каждый из которых соответствует некоторому уровню и является родительским для ν прямоугольников следующего уровня. Введем новое понятие *области взаимодействия*, несколько отличное от приведенного при описании медленного мультипольного алгоритма. *Областью взаимодействия* для одного прямоугольника являются вторичные соседи его родительского прямоугольника и прямоугольники, порожденные соседями его родительского прямоугольника, но не являющиеся его соседями или его вторичными соседями. Пример области взаимодействия приведен на рис. 3. Для черного квадратика областью взаимодействия является заштрихованная область.

В быстром мультипольном алгоритме используются описанные ранее трансляции $M \to M'$ и $L \to L'$. Выпишем последовательность действий для алгоритма:

1. Вычисление мультипольных коэффициентов для всех прямоугольников уровня λ.

2. Объединение ν прямоугольников уровня λ в один прямоугольник уровня $\lambda - 1$. Выполняется $\nu^{\lambda-1}$ раз. С помощью трансляции $M \to M'$ вычисляются мультипольные коэффициенты для прямоугольников уровня $\lambda - 1$.

3. Шаг 2 повторяется с новым уровнем до получения мультипольных коэффициентов для всех прямоугольников 1-го уровня.

4. Локальным коэффициентам всех прямоугольников 1-го уровня присваиваются нулевые значения.

5. Вычисляются локальные коэффициенты для прямоугольников 2-го уровня. При этом используются трансляции $L \to L'$ при разбиении родительского прямоугольника, и $M \to L$ для вычисления потенциала, создаваемого прямоугольниками из области взаимодействия. Полученные в результате двух трансляций локальные коэффициенты суммируются. Следует отметить, что результатом вычислений на этом шаге являются значения локальных коэффициентов для вычисления потенциала, создаваемого внутри прямоугольника всеми зарядами, не принадлежащими его соседям и вторичным соседям.

6. Шаг 5 повторяется на каждом следующем уровне, пока не будут получены локальные коэффициенты для уровня λ.

7. С помощью локального разложения вычисляются значения потенциала во всех точках.

8. Во всех точках каждого прямоугольника на уровне λ вычисляются точные значения потенциала, создаваемого зарядами, принадлежащими его соседям и его вторичным соседям.

9. Вычисленные на шагах 7 и 8 значения потенциалов в точках суммируются.

Здесь, как и в медленном мультипольном алгоритме, подразумевается, что все трансляционные коэффициенты уже известны. На самом деле, в быстром методе используется меньшее число трансляционных коэффициентов, так как трансляции, выполняющиеся для каждого объединения или разбиения при переходе от одного уровня к другому, для всех прямоугольников одинаковы. Однако объем памяти, требуемый для работы метода, мы оценим позднее, а сейчас наиболее важным вопросом является оценка числа операций, за которое быстрый мультипольный алгоритм умножает матрицу на вектор. Как и в оценке для медленного алгоритма, будем оценивать число операций для каждого шага из приведенных выше.

1. Первый шаг, как и в медленном алгоритме, производится за $2nk$ операций.

2. Второй шаг для одного объединения включает в себя ν трансляций. Одна трансляция выполняется за $\sim k^2$ операций. Общее число объединений — $\nu^{\lambda-1}$. Таким образом, число операций на шаге 2 $\sim \nu^\lambda k^2$.

3. Так как процедура шага 2 выполняется на каждом уровне, вплоть до первого, то, следовательно, число операций по объединению прямоугольников равно сумме $\sum\limits_{i=2}^{\lambda} k^2\nu^i = \frac{\nu^{\lambda+1}-\nu}{\nu-1}k^2$.

4. Операции, выполняемые на этом шаге, учитывать не будем.

5. Число операций, выполняемых при трансляции $L \to L'$ для одного родительского прямоугольника $\sim \nu k^2$, соответственно для ν родительских прямоугольников $\nu^2 k^2$ операций, а для трансляции локальных коэффициентов с $(i-1)$-го уровня на i-й $\sim \nu^i k^2$ операций. При выполнении одной трансляции $M \to M'$ производится k^2 операций. Для одного прямоугольника при быстром алгоритме производится $\sim (9\nu - 25)$ трансляций, преобразующих мультипольные коэффициенты прямоугольников того же уровня, и 16 трансляций, преобразующих мультипольные коэффициенты прямоугольников предыдущего уровня. Итого, для ν^i прямоугольников i-го уровня производится $\sim 9\nu^{i+1}$ трансляций $M \to L$. Следовательно, на этом шаге для i-го уровня выполняется $\sim (9\nu^{i+1} + \nu^i)k^2$ операций.

6. Общее число операций, являющееся суммой числа операций шага 5 для каждого уровня, выражается суммой $\sum\limits_{i=1}^{\lambda}(9\nu^{i+1} +$

$+\nu^i)k^2$. Суммируя, получаем число операций

$$\text{oper} \sim \left(9\frac{\nu^{\lambda+2}-\nu^2}{\nu-1}+\frac{\nu^{\lambda+1}-\nu}{\nu-1}\right)k^2 = \frac{9\nu^2+\nu}{\nu-1}(\nu^\lambda-1)k^2.$$

Так как имеет смысл рассматривать ν как сравнительно небольшую константу (в двумерном случае ее обычно берут равной 4 или 9, в трехмерном — 8 или 27), то можно считать, что число операций $\sim \nu^\lambda k^2$.

7. На этом шаге, как и при медленном алгоритме, выполняется $\sim nk$ операций.

8. Аналогично медленному алгоритму, число операций $\sim \frac{25n^2}{\nu^\lambda}$.

9. Очевидно, что число операций на этом шаге $\sim n$.

Таким образом, для всего алгоритма, не считая предварительной части, справедлива следующая оценка числа операций:

$$\text{oper} \approx n + 3nk + \frac{25n^2}{\nu^\lambda} + \frac{9\nu^2+\nu}{\nu-1}(\nu^\lambda-1)k^2 + \frac{\nu}{\nu-1}(\nu^\lambda-1)k^2,$$

или

$$\text{oper} \approx n(1+3k) + \frac{25n^2}{\nu^\lambda} + \frac{9\nu^2+2\nu}{\nu-1}\nu^\lambda k^2.$$

Пренебрегая n по сравнению с nk, а также постоянными коэффициентами, в конечном итоге получим

$$\text{oper} \sim \nu^\lambda k^2 + \frac{n^2}{\nu^\lambda} + nk.$$

Здесь, как мы видим, величина ν^λ есть число прямоугольников на уровне λ, соответствующее выражению NM в аналогичной оценке для медленного алгоритма. По ней мы и будем минимизировать число операций. Получим $\nu^\lambda \sim \frac{n}{k}$. При этом $\text{oper} \sim nk$, то есть, используя быстрый мультипольный алгоритм, мы вместо $O(n^2)$ операций должны выполнить $O(nk)$ операций или, записав то же самое через желаемую точность, — $O(n\log_2(\varepsilon^{-1}))$.

Теперь оценим размер требуемой алгоритму памяти. Видимо, эта оценка должна совпадать с оценкой на число операций, однако это все же следует проверить. Рассмотрим все требующие памяти объекты:

1. Мультипольные коэффициенты хранятся для каждого прямоугольника каждого уровня. Для одного прямоугольника хранится k коэффициентов. Всего прямоугольников на всех уровнях — $\frac{\nu^\lambda-\nu}{\nu-1}$. Следовательно, памяти на мультипольные коэффициенты для прямоугольников требуется $\frac{\nu^{\lambda+1}-\nu}{\nu-1}k$. Также мультипольные коэффициенты хранятся для каждого заряда — это требует nk памяти. Получается $mem = (\frac{\nu^{\lambda+1}-\nu}{\nu-1} + n)k$.

2. Локальные коэффициенты хранятся для двух последних уровней. Максимальный объем памяти требуется при вычислении коэффициентов на уровне λ. Для одного прямоугольника — k коэффициентов. Прямоугольников на двух последних уровнях — $\nu^\lambda + \nu^{\lambda-1}$. Значит, $mem = (\nu^\lambda + \nu^{\lambda-1})k$.

3. Трансляционные коэффициенты для трансляций $M \to M'$. Так как все производящиеся при быстром алгоритме объединения прямоугольников i-го уровня, порождающие трансляции $M \to M'$, одинаковы, коэффициенты надо хранить только для одного такого объединения на каждом уровне. При трансляции, выполняемой для одного прямоугольника, задействуются k^2 трансляционных коэффициентов, следовательно, для объединения ν прямоугольников в один используются и, соответственно, хранятся νk^2 трансляционных коэффициентов. Суммирование по уровням дает $\lambda \cdot \nu k^2$ коэффициентов.

4. Трансляционные коэффициенты для трансляций $L \to L'$. Рассуждая аналогично пункту 3, получим $mem = \lambda\nu k^2$.

5. Трансляционные коэффициенты для трансляций $M \to L$. Здесь ситуация несколько сложнее. При вычислении локальных коэффициентов для одного прямоугольника после того, как он был образован разбиением родительского прямоугольника, выполняется, как было показано при подсчете операций, $9\nu - 25 + 16 = 9\nu - 9$ трансляций $M \to L$, для каждой из которых следует хранить k^2 коэффициентов. Однако каждый родительский прямоугольник образует ν прямоугольников. Следовательно, всего нужно хранить $\nu(9\nu - 9)k^2$ трансляционных коэффициентов.

6. Значения $\{T_i^L\}_{i=0}^k$ для всех точек занимают объем памяти kn.

7. Для выполняемых алгоритмом точных вычислений нужно хранить значения потенциала, создаваемого единичными зарядами с координатами реальных зарядов, принадлежащих прямоугольникам, близким к данному. Для одного прямоугольника при распределении зарядов и точек, близком к равномерному, таких значений $\sim \frac{25n^2}{\nu^{2\lambda}}$, а для всех прямоугольников уровня λ соответственно $\sim \frac{25n^2}{\nu^\lambda}$.

Суммируя по всем пунктам, получаем:

$$mem \approx \frac{25n^2}{\nu^\lambda} + (9\nu^2 - 9\nu + 2\lambda\nu)k^2 + (\nu^\lambda + \nu^{\lambda-1} + \frac{\nu^{\lambda+1} - \nu}{\nu - 1} + 2n)k.$$

Если $\nu^\lambda \sim n$, то $mem \sim nk$ или $mem \sim n \log_2 \varepsilon^{-1}$. Заметим, что память, требуемая на пункты 3–7, заполняется на предварительном этапе и при многократном умножении уже не обновляется.

Стоит прояснить „матричный“ смысл быстрого мультипольного метода. Фактически в методе происходит не что иное, как аппроксимация матрицы A суммой произведений блочно-разреженных матриц и разреженной матрицы. Поскольку, если хранить все вышеперечисленные коэффициенты, как матрицы, и вычислять все трансляции, умножая эти матрицы на соответствующие векторы, как, собственно, и осуществляется на практике, то станет очевидным, что в мультипольном методе матрица A представляется в виде $A \approx S+T$, где S — разреженная матрица, у которой ненулевые элементы совпадают с соответствующими элементами матрицы A, а матрица T — сумма $\sim \lambda$ произведений трансляционных матриц, каждая из которых является блочно-разреженной. Поэлементная точность такого приближения — ε. Для медленного алгоритма матрица T имеет вид $T = T^{L\Psi}T^{ML}T^{QM}$, где матрица T^{QM} переводит величину каждого заряда в k коэффициентов мультипольного разложения, матрица T^{ML} переводит мультипольные коэффициенты в локальные для каждого прямоугольника, а матрица $T^{L\Psi}$ переводит локальные коэффициенты в значения потенциала в точках. В данном случае матрица T^{ML} не будет блочно-разреженной, а будет просто плотной матрицей размера $\sim NMk \times NMk$, то есть матрицей размера меньшего, чем $n \times n$.

5. КОГДА ПРИМЕНЯТЬ МУЛЬТИПОЛЬНЫЙ МЕТОД? Рассмотрим применение обоих описанных выше алгоритмов. Начнем с медленного. Выпишем строго оптимальное значение для MN:

$$\mathrm{oper} \approx \frac{25n^2}{NM} + 3nk + NM(NM-9)k^2;$$

$$\mathrm{oper}'_{NM} = 0 \quad \Rightarrow \quad NM = \sqrt[3]{\frac{25}{2k^2}}\, n^{\frac{2}{3}} \quad \Rightarrow$$

$$\mathrm{oper} \approx (2^{\frac{1}{3}} + 2^{-\frac{2}{3}})25^{\frac{2}{3}} n^{\frac{4}{3}} k^{\frac{2}{3}} + 3nk - 9\left(\frac{25}{2k^2}\right)^{\frac{1}{3}} n^{\frac{2}{3}} k^2.$$

Посмотрим, при каких n число операций становится меньше n^2. Для этого запишем уравнение $\mathrm{oper} = n^2$, или

$$n^2 - (2^{\frac{1}{3}} + 2^{-\frac{2}{3}})25^{\frac{2}{3}} n^{\frac{4}{3}} k^{\frac{2}{3}} - 3nk + 9\left(\frac{25}{2k^2}\right)^{\frac{1}{3}} n^{\frac{2}{3}} k^2 = 0,$$

решив которое относительно n, получим зависимость n от k или от ε. На приведенном графике (рис. 4) можно увидеть, начиная с какого значения n имеет смысл применять медленный алгоритм при заданном k.

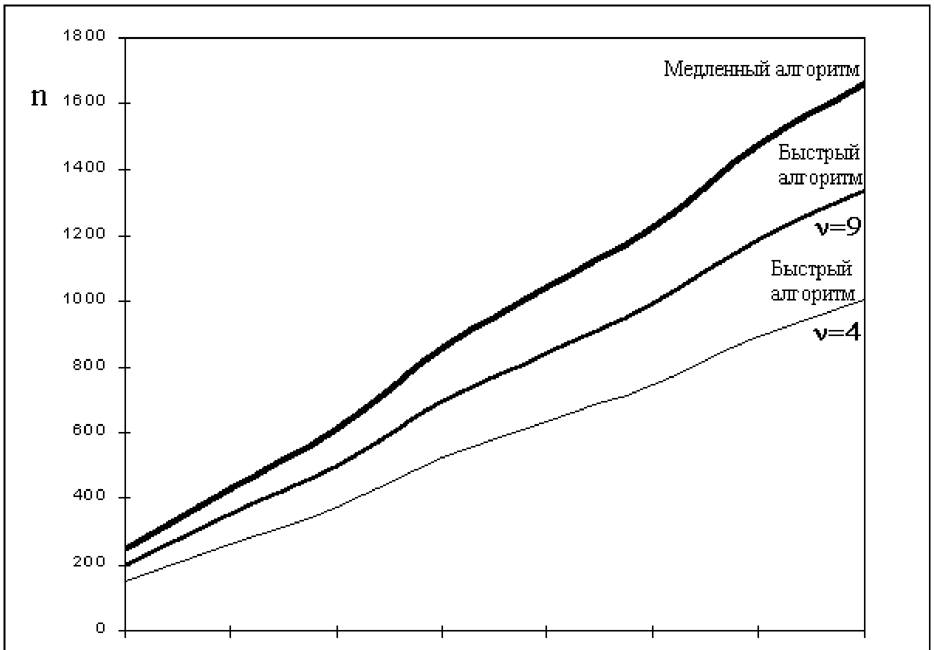


Рис. 4. Сравнение областей применения быстрого и медленного алгоритмов.

Аналогичные уравнения для быстрого алгоритма выглядят следующим образом:

$$\begin{aligned}
&\mathrm{oper} \approx n(1+3k) + \frac{25n^2}{\nu^\lambda} + \frac{9\nu^2+2\nu}{\nu-1}(\nu^\lambda-1)k^2;\\
&\mathrm{oper}'_{\nu^\lambda} = 0 \quad\Rightarrow\quad \nu^\lambda = \sqrt{\frac{25(\nu-1)}{9\nu^2+2\nu}}\cdot\frac{n}{k} \quad\Rightarrow\\
&\mathrm{oper} \approx \sqrt{\frac{25(9\nu^2+2\nu)}{\nu-1}}\,nk + n(1+3k) - \frac{9\nu^2+2\nu}{\nu-1}\,k^2;\\
&\mathrm{oper} = n^2 \Rightarrow n^2 - \sqrt{\frac{25(9\nu^2+2\nu)}{\nu-1}}\,nk - n(1+3k) + \frac{9\nu^2+2\nu}{\nu-1}\,k^2 = 0.
\end{aligned}$$

На рис. 4 приведены построенные для различных ν кривые, показывающие, при каких значениях n имеет смысл применять быстрый алгоритм для заданного k.

Для точности ε результаты удобнее отразить в табл. 1.

Таблица 1

ε	Минимальное n для медленного алгоритма	Минимальное n для быстрого алгоритма	
		$\nu = 4$	$\nu = 9$
10^{-1}	246	150	199
10^{-2}	431	262	348
10^{-3}	615	374	496
10^{-4}	861	523	694
10^{-5}	1046	634	842
10^{-6}	1230	746	990
10^{-7}	1446	895	1188
10^{-8}	1661	1007	1336

Таким образом, мы видим, что наиболее эффективным является быстрый мультипольный алгоритм с $\nu = 4$. В двумерном случае при желаемой точности $\varepsilon \sim 10^{-8}$ он становится эффективным уже при $n \sim 1500$ (по таблице $n = 1007$, но это значение нельзя считать точным, так как формула для подсчета числа операций уже приближенная). Для трехмерного случая результаты будут несколько иные (не принципиально, но тем не менее...), что связано с особенностями функций, используемых для мультипольного разложения. Однако об этом будет рассказано в следующей части данной работы, посвященной аспектам практического применения метода для вполне конкретных задач.

6. ПРИМЕНЕНИЕ МУЛЬТИПОЛЬНОГО МЕТОДА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ТЕОРИИ ПОТЕНЦИАЛА В ТРЕХМЕРНОМ ПРОСТРАНСТВЕ. Рассмотрим следующую задачу: в трехмерном пространстве выделен параллелепипед длины a, ширины b и высоты h, на верхней поверхности которого находятся заряженные пластины сложной конфигурации с распределенным по ним электрическим зарядом (см. пример на рис. 5). Известно значение потенциала внутри параллелепипеда, нужно найти поверхностную плотность заряда на пластинах $q(x)$, где x — координаты точки на пластине, то есть решить уравнение

$$\psi(y) = \int\limits_S q(x) \cdot \frac{1}{|x - y|}\, dS.$$

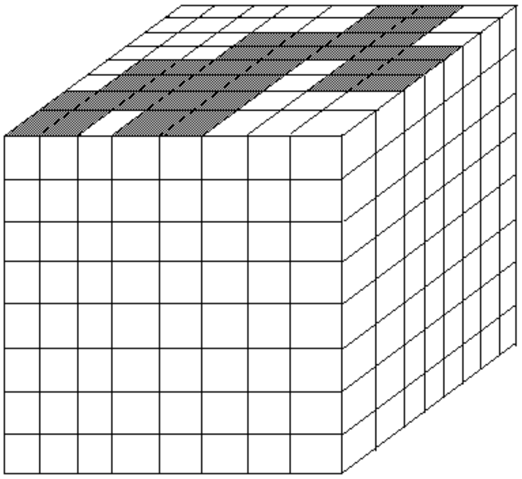


Рис. 5. Заряженные пластины на верхней грани параллелепипеда

Для численного решения задачи воспользуемся методом коллокации. Разобьем пластины на N маленьких квадратов, которые будем считать равномерно заряженными. Поверхностная плотность заряда на i-м квадратике — q_i. Выберем внутри параллелепипеда N точек, в которых известен потенциал ψ. Кулоновский потенциал, создаваемый в i-й точке j-м зарядом, вычисляется следующим образом:

$$\psi_{ij} = \int\limits_S \frac{q_j}{r_{ij}}\,dS = q_i \int\limits_S \frac{1}{r_{ij}}\,dS,$$

где S — площадь квадратика, а r_{ij} принимает значения расстояний от i-й точки до всех точек j-го квадратика. Общий потенциал в i-й точке $\psi_i = \sum_{j=1}^{N} \psi_{ij}$. Таким образом, вычисление потенциала в N точках сводится к умножению матрицы A на вектор $\bar{q}$, где

$$A = \begin{pmatrix} \int\limits_S \frac{1}{r_{11}}\,dS & \dots & \int\limits_S \frac{1}{r_{1N}}\,dS \\ \vdots & \int\limits_S \frac{1}{r_{ij}}\,dS & \vdots \\ \int\limits_S \frac{1}{r_{N1}}\,dS & \dots & \int\limits_S \frac{1}{r_{NN}}\,dS \end{pmatrix}, \quad \bar{q} = \begin{pmatrix} q_1 \\ \vdots \\ q_j \\ \vdots \\ q_N \end{pmatrix}.$$

Следовательно, если мы хотим найти плотность зарядов в виде вектора плотностей $\bar{q}$, нам необходимо решить матричное уравнение $A\bar{q} = \bar{\psi}$. Насколько известно из теории проекционно-сеточных методов, с возрастанием размерности данной системы повышается точность численного решения. В случае большой размерности удобно решать данную систему методами, опирающимися на пространства Крылова, а значит, возможно, окажется полезным применение мультипольного умножения.

Для начала рассмотрим случай точечных зарядов, находящихся в центрах пластин. Упрощенная задача будет выглядеть так:

$$A_s\bar{q}_s = \bar{\psi}, \quad \text{где } A_s = \begin{pmatrix} \frac{1}{r_{11}} & \cdots & \frac{1}{r_{1N}} \\ \vdots & \frac{1}{r_{ij}} & \vdots \\ \frac{1}{r_{N1}} & \cdots & \frac{1}{r_{NN}} \end{pmatrix}, \ \bar{q}_s = \begin{pmatrix} S \cdot q_1 \\ \vdots \\ S \cdot q_j \\ \vdots \\ S \cdot q_N \end{pmatrix}.$$

Рассмотрим один заряд величины q_s в точке Q, находящейся на расстоянии r_q от начала координат. Потенциал, создаваемый им в точке P, удаленной от начала координат на расстояние r_p, равен $\frac{q_s}{r_{pq}} = \frac{q_s}{\sqrt{r_q^2+r_p^2-2r_qr_p\cos\gamma}}$, где γ — угол между векторами $\bar{r}_p$ и $\bar{r}_q$, проведенными из начала координат соответственно в точки Q и P. Пусть $r_p > r_q$. Обозначим $r \equiv \frac{r_q}{r_p}$. Тогда $\frac{q_s}{r_{pq}} = \frac{q_s}{r_p}\frac{1}{\sqrt{1-2r\cos\gamma+r^2}}$. Вторая дробь правой части данного неравенства является производящей функцией для полиномов Лежандра. Это значит, что $\frac{q_s}{r_{pq}} = \frac{q_s}{r_p}\sum\limits_{n=0}^{\infty} r^n P_n^0(\cos\gamma)$, где $\{P_n^0\}_{n=0}^{\infty}$ — полиномы Лежандра. В конечном итоге для потенциала, создаваемого зарядом в точке P, получаем:

$$\psi_{qp} = \sum_{n=0}^{\infty} \frac{q_s r_q^n}{r_p^{n+1}} P_n^0(\cos\gamma). \tag{1}$$

Однако для полиномов Лежандра в этом случае справедливо следующее соотношение:

$$P_n^0(\cos\gamma) = \sum_{m=-n}^{n} Y_n^{-m}(\alpha,\beta)Y_n^m(\theta,\phi), \tag{2}$$

где (α,β) — угловые координаты точки Q, (θ,ϕ) — угловые координаты точки P, Y_n^m и Y_n^{-m} — сферические гармоники, определяющиеся следующим образом:

$$Y_n^m(\theta,\phi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \cdot P_n^{|m|}(\cos\theta)e^{im\phi}.$$

Тогда для потенциала в точке P имеем равенство:

$$\psi_{qp} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \frac{Y_n^m(\theta,\phi)}{r_p^{n+1}},$$

где

$$M_n^m = q_s r_q^n \cdot Y_n^{-m}(\alpha,\beta).$$

Это представление и будет мультипольным разложением для кулоновского потенциала в трехмерном пространстве. Таким образом, мультипольный метод в трехмерном пространстве будет использовать разложение по сферическим гармоникам.

Для реализации мультипольного метода будем опираться на изложенный во второй части общий подход. Для начала приведем необходимую оценку сходимости для мультипольного разложения. Одно из известных свойств полиномов Лежандра состоит в том, что если $|x| \leqslant 1$, то $P_n^0(x) \leqslant 1$. Отсюда следует, что

$$\left|\psi_{qp} - \sum_{n=0}^{p} \frac{q_s r_q^n}{r_p^{n+1}} P_n^0(\cos\gamma)\right| \leqslant \frac{q_s}{r_p - r_q}\left(\frac{r_q}{r_p}\right)^{p+1}. \tag{3}$$

Если рассматривать систему k зарядов, лежащих внутри сферы радиуса r_q с центром в начале координат, то соответствующие мультипольные коэффициенты для зарядов суммируются. То есть коэффициенты мультипольного разложения для всей сферы выражаются следующим образом:

$$M_n^m = \sum_{i=1}^{k} q_s^{(k)} (r_q^{(k)})^n \cdot Y_n^{-m}(\alpha^{(k)},\beta^{(k)}).$$

Для получения трансляционных матриц воспользуемся теоремами, доказанными в работе [2].

Теорема 1 (*L. Greengard, V. Rokhlin*). Пусть $Q = (\rho,\alpha,\beta)$,

$P = (r, \theta, \phi)$, $r > \rho$, $P - Q = (r', \theta', \phi')$. Тогда

$$\frac{Y_{n'}^{m'}(\theta', \phi')}{r'^{\ n'+1}} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{J_{m}^{m'} \cdot A_n^m \cdot A_{n'}^{m'} \cdot \rho^n \cdot Y_n^{-m}(\alpha, \beta)}{A_{n+n'}^{m+m'}} \times$$
$$\times \frac{Y_{n+n'}^{m+m'}(\theta, \phi)}{r^{n+n'+1}}, \qquad (4)$$

где

$$A_n^m = \frac{(-1)^n}{\sqrt{(n+m)!(n-m)!}}, \qquad (5)$$

$$J_m^{m'} = \begin{cases} (-1)^{\min(|m'|,|m|)}, & \text{если } m \cdot m' < 0; \\ 1, & \text{в остальных случаях.} \end{cases} \qquad (6)$$

Доказательство этой теоремы, как и двух других, опирается на свойство полиномов Лежандра (формула (2)) и на следующее свойство сферических функций:

$$\frac{Y_n^0(\theta, \phi)}{r^{n+1}} = A_n^0 \cdot \frac{\partial^n}{\partial z^n} \left(\frac{1}{r} \right),$$

и для $m > 0$

$$\frac{Y_n^m(\theta, \phi)}{r^{n+1}} = A_n^m \cdot \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right)^m \cdot \left(\frac{\partial^n}{\partial z^n} \right)^{n-m} \left(\frac{1}{r} \right),$$

$$\frac{Y_n^{-m}(\theta, \phi)}{r^{n+1}} = A_n^m \cdot \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right)^m \cdot \left(\frac{\partial^n}{\partial z^n} \right)^{n-m} \left(\frac{1}{r} \right),$$

где A_n^m определяется формулой (5). Используя теорему 1, найдем коэффициенты для трансляции $M \to M'$. Рассмотрим сферу радиуса a с центром в точке $Q = (\rho, \alpha, \beta)$. Мультипольное разложение для этой сферы дает в любой точке $P = (r, \theta, \phi)$ вне сферы потенциал

$$\psi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \frac{Y_n^m(\theta', \phi')}{r'^{n+1}},$$

где $P - Q = (r', \theta', \phi')$. Тогда для точки P, лежащей вне сферы радиуса $(a + \rho)$ с центром в начале координат, мультипольное разложение выглядит так:

$$\psi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} M'^k_j \frac{Y_j^k(\theta, \phi)}{r^{j+1}}.$$

Для того, чтобы получить трансляционные коэффициенты, подставим результат (4) в мультипольное разложение для сферы с центром в точке Q. Получим

$$\psi(P)=\sum_{n=0}^{\infty}\ \sum_{m=-n}^{n} M_n^m \sum_{n'=0}^{\infty}\ \sum_{m'=-n'}^{n'} \frac{J_m^{m'}\cdot A_{n'}^{m'}\cdot A_{n'}^{m}\cdot \rho^{n'}\cdot Y_{n'}^{-m'}(\alpha,\beta)}{A_{n'+n}^{m'+m}}\cdot\frac{Y_{n'+n}^{m'+m}(\theta,\phi)}{r^{n'+n+1}} \quad \Rightarrow$$

$$\psi(P)=\sum_{j=0}^{\infty}\ \sum_{k=-j}^{j}\frac{Y_j^k(\theta,\phi)}{r^{j+1}}\times$$

$$\times\sum_{n=0}^{j}\ \sum_{m=\max(k+n-j,-n)}^{\min(k+j-n,n)}\frac{M_{j-n}^{k-m}\cdot J_m^{k-m}\cdot\rho^n\cdot A_n^m\cdot A_{j-n}^{k-m}Y_n^{-m}(\alpha,\beta)}{A_j^k}.$$

Преобразование для суммы производится с помощью замены $j = n + n'$, $k = m + m'$. Окончательно получаем

$$M'^k_j=\sum_{n=0}^{j}\ \sum_{m=\max(k+n-j,-n)}^{\min(k+j-n,n)} M_{j-n}^{k-m}\times$$
$$\times\frac{\sqrt{(j+k)!(j-k)!}\, i^{|k|-|m|-|k-m|}\,Y_n^{-m}(\alpha,\beta)\rho^n}{\sqrt{(j-n+k-m)!(j-n-k+m)!(n+m)!(n-m)!}}. \tag{7}$$

Оценка сходимости в данном случае получается путем суммирования оценки (3) для всех зарядов. В итоге для сходимости получаем

$$\left|\psi(P)-\sum_{j=0}^{p}\sum_{k=-j}^{j}M'^k_j\frac{Y_j^k(\theta,\phi)}{r^{j+1}}\right|\leqslant\frac{\sum\limits_{i=1}^{N}\left|q_s^{(i)}\right|}{r-(a+\rho)}\left(\frac{a+\rho}{r}\right)^{p+1}. \tag{8}$$

При разложении потенциала по сферическим функциям до членов порядка $\frac{1}{r^p}$ получится сумма из $(p+1)^2$ слагаемых, т. е. для одного кластера зарядов нужно хранить $(p+1)^2$ мультипольных коэффициентов. Вернемся теперь к параллелепипеду, потенциал внутри которого мы желаем вычислить с помощью мультипольного метода. Снова, как и в описанном уже нулевом случае, выберем число ν (для трехмерного случая проще взять $\nu = 8$) и разобьем параллелепипед на ν частей. Для удобства будем считать параллелепипед кубом. Если реально это не куб — его всегда можно разбить на несколько кубов (или „почти кубов“) и применять иерархическое разбиение уже к ним. Итак, разбиваем куб на

ν кубов, каждый из них еще на ν кубов, и так λ раз. Получим ν^λ маленьких кубов. Для каждого из этих кубов подсчитаем коэффициенты мультипольного разложения, суммируя мультипольные коэффициенты, полученные для содержащихся в них зарядов, затем, как и в нулевом случае, будем объединять эти кубы в большие кубы предыдущего уровня, для вычисления мультипольных коэффициентов которых будем умножать векторы, составленные из мультипольных коэффициентов маленьких кубов, на соответствующие им трансляционные матрицы, элементы которых вычисляются с помощью формулы (7), и суммировать полученные результаты. Так как геометрически два объединения ничем не отличаются друг от друга, то для каждого объединения можно использовать одни и те же трансляционные матрицы. Очевидно, что для одного объединения требуется ν трансляционных матриц размера $(p+1)^2\times(p+1)^2$. В принципе они могут быть объединены в одну трансляционную матрицу, которая будет блочно-разреженной матрицей размера $\nu(p+1)^2 \times \nu(p+1)^2$. Теперь, получив мультипольные коэффициенты для всех кубов на всех уровнях, приступим к рассмотрению свойств локального разложения в условиях данной задачи. Рассмотрим снова заряд Q и точку P, но на этот раз $r_q > r_p$. В этом случае уравнение (1) примет вид

$$\psi_{qp} = \sum_{n=0}^{\infty} \frac{q_s r_p^n}{r_q^{n+1}} P_n^0(\cos\gamma), \tag{9}$$

или, учитывая соотношение (2),

$$\psi_{qp} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} L_n^m Y_n^{-m}(\alpha, \beta) r_p^n, \tag{10}$$

где

$$L_n^m = \frac{q_s Y_n^m(\theta, \phi)}{r_q^{n+1}}.$$

Эта сумма в данном случае является локальным разложением потенциала. Оценка (3) для локального разложения примет вид:

$$\left|\psi_{qp} - \sum_{n=0}^{p} \frac{q_s r_p^n}{r_q^{n+1}} P_n^0(\cos\gamma)\right| \leqslant \frac{q_s}{r_q - r_p} \left(\frac{r_p}{r_q}\right)^{p+1}. \tag{11}$$

Так как в мультипольном методе используются трансляции $L \to$ $\to L'$, нужно найти трансляционные матрицы для локальных коэффициентов, как это уже было сделано для мультипольных. Воспользуемся для этого другой теоремой из работы [2].

Теорема 2 (*L. Greengard, V. Rokhlin*). Рассмотрим точки $P =$ $= (r, \theta, \phi)$ и $Q = (\rho, \alpha, \beta)$. Пусть $P - Q = (r', \theta', \phi')$. Тогда

$$\begin{aligned} Y_{n'}^{m'}(\theta', \phi') r'^{n'} = \sum_{n=0}^{n'} \sum_{m=-n}^{n} \frac{J_{n,m}^{m'} \cdot A_n^m \cdot A_{n'-n}^{m'-m} \cdot \rho^n \cdot Y_n^m(\alpha, \beta)}{A_{n'}^{m'}} \times \\ \times Y_{n'-n}^{m'-m}(\theta, \phi) r^{n'-n}, \end{aligned} \tag{12}$$

где

$$J_{n,m}^{m'} = \begin{cases} (-1)^n(-1)^m, & \text{если } m \cdot m' > 0; \\ (-1)^n(-1)^{m'-m}, & \text{если } m \cdot m' < 0 \text{ и } |m'| < |m|; \\ (-1)^n & \text{в остальных случаях.} \end{cases} \tag{13}$$

Пусть имеется локальное разложение с центром в точке Q

$$\psi(P) = \sum_{n=0}^{p} \sum_{m=-n}^{n} L_n^m \cdot Y_n^m(\theta', \phi') \cdot r^n. \tag{14}$$

Тогда для точки P существует локальное разложение с центром в начале координат:

$$\psi(P) = \sum_{j=0}^{p} \sum_{k=-j}^{j} L'^k_j \cdot Y_j^k(\theta, \phi) \cdot r'^n.$$

Для того, чтобы найти связь между L и L', подставим выражение (12) в формулу (14).

$$\psi(P) = \sum_{n=0}^{p} \sum_{m=-n}^{n} L_n^m \cdot \sum_{n'=0}^{n} \sum_{m=-n'}^{n'} \frac{J_{n',m'}^{m} \cdot A_{n'}^{m'} \cdot A_{n-n'}^{m-m'} \cdot \rho^{n'} \cdot Y_{n'}^{m'}(\alpha,\beta)}{A_n^m} \times$$
$$\times Y_{n-n'}^{m-m'}(\theta,\phi) r^{n-n'} \quad \Rightarrow$$

$$\psi(P) = \sum_{j=0}^{p} \sum_{k=-j}^{j} Y_j^k(\theta,\phi) r^j \sum_{n=j}^{p} \sum_{m=\max(-n+j+k,-n)}^{\min(n-j+k,n)} L_n^m \times$$
$$\times \frac{J_{n-j,m-k}^{m} \cdot A_{n-j}^{m-k} \cdot A_j^k \cdot \rho^{n-j} \cdot Y_{n-j}^{m-k}(\alpha,\beta)}{A_n^m}.$$

Сумма преобразуется с помощью замены $j = n - n', k = m - m'$. Окончательно получим

$$L'^{k}_{j} = \sum_{n=j}^{p} \sum_{m=\max(-n+j+k,-n)}^{\min(n-j+k,n)} L_n^m \times$$
$$\times \frac{\sqrt{(n+m)!(n-m)!}\; i^{|m|-|k|-|m-k|}\, Y_{n-j}^{m-k}(\alpha,\beta)\rho^{n-j}}{\sqrt{(n-j+m-k)!(n-j-m+k)!(j+k)!(j-k)!}\,(-1)^{n-j}}. \tag{15}$$

Таким образом, с помощью этой формулы можно вычислять коэффициенты локального разложения для маленьких кубов, зная локальные коэффициенты для куба, в который они вложены. Как и в случае объединения, при разбиении кубов на более мелкие используются одни и те же трансляционные матрицы, поэтому для трансляции локальных коэффициентов достаточно хранить только ν трансляционных матриц размера $(p+1)^2 \times (p+1)^2$.

Перейдем теперь к третьей существенной составляющей мультипольного подхода — трансляции $M \to L$. Для начала приведем третью теорему из работы [2].

Теорема 3 (*L. Greengard, V. Rokhlin*). Рассмотрим точки $Q = (\rho, \alpha, \beta)$ и $P = (r, \theta, \phi)$, причем $P - Q = (r', \theta', \phi')$ и $r < \rho$. Тогда

$$\frac{Y_{n'}^{m'}(\theta',\phi')}{r'^{n+1}} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{J_{m}^{n',m'} \cdot A_n^m \cdot A_{n'}^{m'} \cdot Y_{n+n'}^{m'-m}(\alpha,\beta)}{\rho^{n+n'+1} \cdot A_{n+n'}^{m-m'}} Y_n^m(\theta,\phi)\, r^n, \tag{16}$$

где

$$J_{m}^{n',m'} = \begin{cases} (-1)^{n'}(-1)^{\min(|m'|,|m|)}, & \text{если } m \cdot m' > 0; \\ (-1)^{n'} & \text{в остальных случаях.} \end{cases} \tag{17}$$

Рассмотрим сферу радиуса a_q с центром в точке $Q = (\rho, \alpha, \beta)$ и сферу радиуса a_p с центром в начале координат, внутри которой лежит точка $P = (r, \theta, \phi)$, т. е. $r < a_p$, причем $\rho > a_q + a_p$. Тогда мультипольное разложение потенциала в точке P

$$\psi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \cdot \frac{Y_n^m(\theta', \phi')}{r'^{n+1}}. \tag{18}$$

Локальное разложение потенциала в этой точке

$$\psi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} L_n^m \cdot Y_j^k(\theta, \phi) \cdot r^j.$$

Для того, чтобы связать мультипольные и локальные коэффициенты, подставим выражение (16) в формулу (18). Получим

$$\psi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} \frac{J_{m'}^{n,m} \cdot A_{n'}^{m'} \cdot A_n^m \cdot Y_{n'+n}^{m-m'}(\alpha,\beta)}{\rho^{n'+n+1} \cdot A_{n'+n}^{m'-m}} \cdot Y_{n'}^{m'}(\theta, \phi) \cdot r^{n'} \Rightarrow$$

$$\psi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} Y_j^k(\theta, \phi) \cdot r^j \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{J_k^{n,m} \cdot A_j^k \cdot A_n^m \cdot Y_{j+n}^{m-k}(\alpha,\beta)}{\rho^{j+n+1} \cdot A_{j+n}^{k-m}} \cdot M_n^m.$$

Преобразование суммы производится с помощью замены $j = n'$, $k = m'$. Окончательно для локальных коэффициентов получаем:

$$L_j^k = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \frac{\sqrt{j+n+m-k)!(j+n-m+k)!}\ i^{|k-m|-|k|-|m|}\, Y_{n+j}^{m-k}(\alpha,\beta)}{\sqrt{(n+m)!(n-m)!(j+k)!(j-k)!}\ (-1)^n \rho^{j+n+1}}. \tag{19}$$

Оценка сходимости локального разложения получается из суммирования оценки (11) для всех зарядов и выглядит так:

$$\left| \psi(P) - \sum_{j=0}^{p} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j \right| \leqslant \frac{\sum_{i=1}^{N} |q_s^{(i)}|}{\rho - a_q - a_p} \left(\frac{a_p}{\rho - a_q} \right)^{p+1}. \tag{20}$$

Если взять $\psi(P) = \sum\limits_{n=0}^{p} \sum\limits_{m=-n}^{n} M_n^m \frac{Y_n^m(\theta',\phi')}{r'^{n+1}}$, то для локальных коэффициентов получим

$$L_j^k = \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{\sqrt{j+n+m-k)!(j+n-m+k)!}\ i^{|k-m|-|k|-|m|}\, Y_{n+j}^{m-k}(\alpha,\beta)}{\sqrt{(n+m)!(n-m)!(j+k)!(j-k)!}\ (-1)^n \rho^{j+n+1}} \cdot M_n^m.$$

Оценка ошибки в этом случае

$$\left|\sum_{n=0}^{p}\sum_{m=-n}^{n} M_n^m \frac{Y_n^m(\theta',\phi')}{r'^{n+1}} - \sum_{j=0}^{p}\sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta,\phi)\cdot r^j\right| \leqslant \frac{\sum\limits_{i=1}^{N}|q_s^{(i)}|}{\rho-a_q-a_p}\left(\frac{a_p}{\rho-a_q}\right)^{p+1}.$$

Ошибку для мультипольного разложения вычислим по формуле (8). Просуммировав две оценки, получим общую ошибку:

$$\begin{aligned}\left|\psi(P) - \sum_{j=0}^{p}\sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta,\phi)\cdot r^j\right| \leqslant \frac{\sum\limits_{i=1}^{N}|q_s^{(i)}|}{\rho-a_q-a_p}\left(\frac{a_p}{\rho-a_q}\right)^{p+1} + \\ + \frac{\sum\limits_{i=1}^{N}|q_s^{(i)}|}{\rho-a_q-a_p}\left(\frac{a_q}{\rho-a_p}\right)^{p+1}.\end{aligned} \tag{21}$$

В случае быстрого мультипольного алгоритма можно считать, что отношения $\left(\frac{a_p}{\rho-a_q}\right)$ и $\left(\frac{a_q}{\rho-a_p}\right)$ не превосходят $\frac{1}{2}$, и поэтому для точности метода ε верна следующая оценка:

$$\varepsilon \sim \left(\frac{1}{2}\right)^{p+1}. \tag{22}$$

Очевидно, что для каждой трансляции $M \to L$, происходящей при разбиении кубов, используются одни и те же коэффициенты, поэтому хранить нужно только ν наборов трансляционных матриц (по одному для каждого образующегося при разбиении куба). В набор входят 98 матриц для трансляции мультипольных коэффициентов кубов предыдущего уровня и $(27\nu - 125)$ матриц для трансляции мультипольных коэффициентов кубов текущего уровня. Все эти матрицы имеют размер $(p+1)^2 \times (p+1)^2$. В итоге получатся $27\nu(\nu-1)(p+1)^4$ трансляционных коэффициентов.

Вернемся теперь к первоначальной задаче с заряженными пластинами. Для того, чтобы в этом случае воспользоваться мультипольным методом, нужно получить коэффициенты мультипольного разложения для одного заряженного прямоугольника, взяв его центр за центр разложения. Если концентрация заряда q, то

$$M_n^m = q\int\limits_S \rho^n Y_n^{-m}(\alpha,\beta)\,dS.$$

Взяв начало координат в центре прямоугольника и выбрав координатные оси так, что прямоугольник лежит в плоскости XY, из определения сферической гармоники получаем

$$\int\limits_S \rho^n Y_n^{-m}(\alpha,\beta)\, dS = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}}\, P_n^{|m|}(0) \int\limits_S \rho^n e^{-im\phi}\, dS.$$

Для нечетных значений $n-|m|$: $P_n^{|m|}(0)=0$. Для четных значений $n-|m|$:
если $m \geqslant 0$, то

$$\rho^n e^{-im\phi} = (x^2+y^2)^{\frac{n-m}{2}}(x+iy)^m;$$

если $m<0$, то

$$\rho^n e^{-im\phi} = (x^2+y^2)^{\frac{n+m}{2}}(x-iy)^{-m}.$$

В итоге получим

$$\rho^n e^{-im\phi} = \sum_{j=0}^{\frac{n-|m|}{2}} \binom{j}{\frac{n-|m|}{2}} \sum_{k=0}^{|m|} (\mathrm{sign}(-m)i)^{|m|-k} \binom{k}{|m|} x^{2j+k} y^{n-(2j+k)}.$$

Так как интегрирование происходит по прямоугольнику, то

$$\int\limits_S x^{2j+k} y^{n-(2j+k)}\, dS = \int\limits_{-a}^{a} x^{2j+k} dx \int\limits_{-b}^{b} y^{n-(2j+k)} dy \quad \Rightarrow$$

$$\int\limits_S x^{2j+k} y^{n-(2j+k)}\, dS = \frac{\left(a^{2j+k+1}-(-a)^{2j+k+1}\right)\left(b^{n-(2j+k)+1}-(-b)^{n-(2j+k)+1}\right)}{\left(2j+k+1\right)\left(n-(2j+k)+1\right)},$$

где a и b — половины сторон прямоугольника. Отсюда получим окончательную формулу для M_n^m:

$$M_n^m = \sqrt{\tfrac{(n-|m|)!}{(n+|m|)!}}\, P_n^{|m|}(0) \sum_{j=0}^{\frac{n-|m|}{2}} \tbinom{j}{\frac{n-|m|}{2}} \sum_{k=0}^{|m|} (\mathrm{sign}(-m)i)^{|m|-k} \tbinom{k}{|m|} I_{2j+k,n-(2j+k)},$$

где

$$I_{j,k} \equiv \int\limits_S x^j y^k\, dS = \frac{(a^{j+1}-(-a)^{j+1})(b^{k+1}-(-b)^{k+1})}{(j+1)(k+1)}.$$

Теперь аппарат мультипольного метода для данной задачи полностью готов. Прямоугольные участки пластин становятся аналогами точечных зарядов. Сначала вычисляются мультипольные разложения для них, потом для кубов, в которые они вложены, затем все идет по ранее описанной схеме.

Конечно, в первую очередь нас интересует выигрыш в количестве операций, даваемый мультипольным алгоритмом, и объем памяти, занимаемый при этом. Сразу отметим, что порядок количества операций будет такой же, как и в двумерном случае, за исключением того, что изменится связь параметра k (количество членов в разложении) с точностью ε. Для нулевого случая имело место соотношение $\varepsilon \sim \left(\frac{1}{2}\right)^k$, откуда следовало $k \sim \log_2 \varepsilon^{-1}$. В трехмерном случае $k = (p+1)^2$, где p — порядок разложения, а для точности, в свою очередь, выполняется соотношение (22), из которого следует, что $(p+1) \sim \log_2 \varepsilon^{-1}$. Следовательно, для трехмерного случая $k \sim \log_2^2 \varepsilon^{-1}$ и число операций $oper \sim N \log_2^2 \varepsilon^{-1}$, где N — число прямоугольников, на которое оказались разбиты пластины. Однако для того, чтобы точно установить, при каких N алгоритм становится эффективным, следует все же выписать более точную формулу для подсчета операций и для объема памяти.

Начнем с памяти, так как некоторые ее составляющие были уже получены во время получения трансляционных матриц. Итак, память, идущая на хранение трансляционных матриц, равна

$$mem_{tr} = 2\nu(p+1)^4 + 27\nu(\nu-1)(p+1)^4 = \nu(27\nu-25)(p+1)^4.$$

На хранение начальных мультипольных коэффициентов для N заряженных прямоугольников нужно $N(p+1)^2$ элементов памяти. Помимо этого, нужно хранить $(p+1)^2$ мультипольных коэффициентов для каждого куба каждого уровня. Однако если заряды находятся только на одной грани куба, нужно хранить коэффициенты только для кубов, находящихся на верхнем слое. Будем считать, что на i-м уровне наверху находятся $\nu^{\frac{2}{3}i}$ кубов. Получим

$$mem_M = \sum_{i=1}^{\lambda} (p+1)^2 \nu^{\frac{2}{3}i} = \frac{\nu^{\frac{2}{3}(\lambda+1)} + \nu^{\frac{2}{3}}}{\nu^{\frac{2}{3}} - 1}(p+1)^2.$$

Локальные коэффициенты хранятся только для двух последовательных уровней. Максимальный объем памяти при этом

$$mem_L = \nu^{\lambda-1}(\nu+1)(p+1)^2.$$

На хранение значений сферических функций в N точках требуется память $mem_{N2} = N(p+1)^2$. На хранение точных значений потенциала для близких друг к другу кубов памяти расходуется несколько меньше, чем в общем случае, опять же из-за неравномерного распределения заряда:

$$mem_{pr} = \frac{75N^2}{\nu^\lambda}.$$

Всего получится объем памяти

$$\begin{aligned} mem = \nu(27\nu - 25)(p+1)^4 + \tfrac{\nu^{\frac{2}{3}(\lambda+1)} + \nu^{\frac{2}{3}}}{\nu^{\frac{2}{3}} - 1}(p+1)^2 + \\ + \nu^{\lambda-1}(\nu+1)(p+1)^2 + 2N(p+1)^2 + \tfrac{75N^2}{\nu^\lambda}, \end{aligned}$$

или приближенно

$$mem \approx 27\nu^2(p+1)^4 + \tfrac{\nu^{\frac{2}{3}}}{\nu^{\frac{2}{3}}-1}\nu^{\frac{2}{3}\lambda}(p+1)^2 + \nu^\lambda(p+1)^2 + 2N(p+1)^2 + \tfrac{75N^2}{\nu^\lambda}.$$

Не будем подробно останавливаться на отдельных составляющих, из которых складывается общее число операций; это подробно изложено в предыдущей части. Выпишем просто формулу с учетом размерности пространства и неравномерности распределения зарядов (распределение по плоскости заряженных квадратов, на которые разбиваются пластины, будем считать равномерным, а их концентрацию равной $N\nu^{-\frac{2}{3}\lambda}$)

$$\begin{aligned} oper \approx \tfrac{\left(54\nu^2 - 150\nu + \nu^{\frac{2}{3}}\right)}{\nu^{\frac{2}{3}}-1}(\nu^{\frac{2}{3}\lambda} + 1)(p+1)^4 + \\ + \tfrac{\nu}{\nu-1}(\nu^\lambda + 1)(p+1)^4 + \tfrac{75N^2}{\nu^\lambda} + 2N(p+1)^2. \end{aligned} \tag{23}$$

Минимизируем количество операций по ν^λ. Получим уравнение

$$\tfrac{\left(54\nu^2 - 150\nu + \nu^{\frac{2}{3}}\right)}{\nu^{\frac{2}{3}}-1}\tfrac{2}{3}(p+1)^4(\nu^\lambda)^{\frac{5}{3}} + \tfrac{\nu}{\nu-1}(p+1)^4(\nu^\lambda)^2 - 75N^2 = 0. \tag{24}$$

Если рассматривать порядок числа операций, то достаточно решить уравнение

$$(\nu^\lambda)^2 \sim N^2 \Rightarrow \nu^\lambda \sim N.$$

То есть результат, полученный в предыдущем разделе, останется справедливым и в данном случае. Что же касается точного решения уравнения (24), то его имеет смысл считать численно для конкретной реализации метода, потому что λ — целое число, и, следовательно, ν^λ принимает несколько дискретных значений, которые можно найти и с помощью приближенного $(p+1)^4(\nu^\lambda)^2-75N^2 \approx 0$, в результате решения которого возникают два значения для λ. Какое из них выбрать — определяется их подстановкой в формулу (23). Подстановка же приближенной оценки $\nu^\lambda \sim N$ в формулу (23) снова дает $oper \sim N(p+1)^2$.

Для того, чтобы выяснить, когда следует применять быстрый алгоритм, построим табл. 2, аналогичную табл. 1 из предыдущего раздела. Рассмотрим два наиболее удобных для реализации алгоритма значения для ν: $\nu=8$ и $\nu=27$ и для каждого из них получим оптимальные значения λ из соотношения $\lambda \approx \log_\nu\bigl(\sqrt{75}\frac{N}{(p+1)^2}\bigr)$. Для полученных значений будем решать уравнение $oper=N^2$ при определенных значениях точности ε. Будем считать, что $p=\log_2(\varepsilon^{-1})$. Получим следующие результаты:

Таблица 2

ε	$p+1$	Минимальное значение N, при котором алгоритм становится эффективным	
		$\nu = 8$	$\nu = 27$
10^{-1}	3	2185	5739
10^{-2}	7	8738	22955
10^{-3}	10	19660	51649
10^{-4}	13	41019	107761
10^{-5}	17	62136	163236
10^{-6}	20	87621	230188
10^{-7}	23	128397	337311
10^{-8}	27	164076	431043

Эти результаты нельзя считать точными в связи с тем, что не точна оценка числа операций, и они дают только приближенное представление о порядке N. Для наглядности отразим эти результаты на графике (см. рис. 6).

Для сравнения приведем теперь результаты численного эксперимента. С помощью мультипольного алгоритма вычислялся потенциал на равномерной сетке точек в единичном кубе. Заряженные пластины находились на верхней грани куба и представляли собой чередующиеся полосы. Точность определялась посредством вычисления разности точных значений, полученных интегрированием по каждому равномерно заряженному участку пластины, и значений потенциала, полученных с помощью мультипольного метода. Проще говоря, точностью считалось значение следующего выражения:

$$\varepsilon = \frac{\|\bar{\psi}_{\cdot} - \bar{\psi}_{\cdot}\|_2}{\|\bar{\psi}_{\cdot}\|_2}.$$

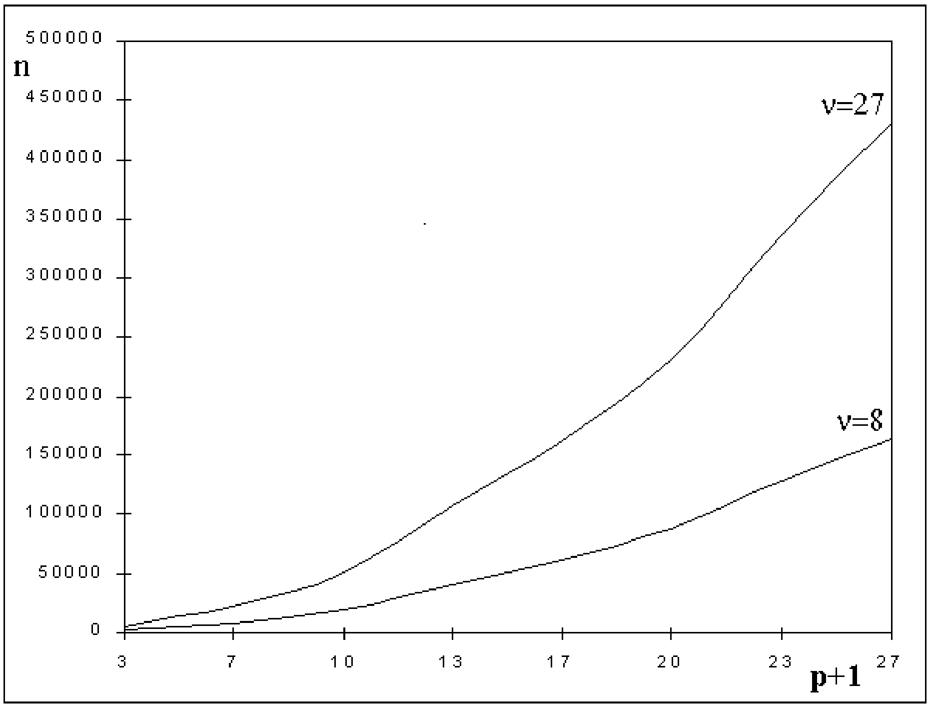


Рис. 6. График зависимости минимального размера матрицы, при котором применение мультипольного алгоритма имеет смысл, от порядка разложения (фактически от точности).

Аналогом числа операций служило время умножения матрицы на вектор, измерение которого осуществлялось с помощью машинного таймера. Время приводится с точностью до 5 с. Для наглядности в табл. 3 представлены отношения времен для точного ($t_{\cdot}$) и мультипольного ($t_{\cdot}$) умножений, и отношения соответствующих значений числа операций. В связи с тем, что большие матрицы не помещались в оперативную память компьютера, время точного умножения для них считалось приближенно, исходя из данных для небольших матриц. Также в таблице отражены отношения объемов памяти, использующихся при точном и при приближенном умножениях. Вычисления проводились при $\nu = 8$.

Таблица 3

N	$p+1$	λ	ε	$\frac{oper}{N^2}$	$t_{.}$	$t_{.}$	$\frac{t_{.}}{t_{.}}$	$\frac{mem}{N^2}$
5000	4	3	$\sim 10^{-4}$	0.66	10	20	2.00	0.17
10000	4	3	$\sim 10^{-4}$	0.28	55	30	0.55	0.16
20000	4	4	$\sim 10^{-4}$	0.15	220	55	0.25	0.02
30000	4	4	$\sim 10^{-4}$	0.08	495	80	0.16	0.02
40000	4	4	$\sim 10^{-4}$	0.05	880	105	0.12	0.02
50000	4	4	$\sim 10^{-4}$	0.04	1375	115	0.08	0.02
5000	6	2	$\sim 10^{-5}$	1.86	10	30	3.00	1.27
10000	6	3	$\sim 10^{-5}$	0.80	55	50	0.91	0.18
20000	6	3	$\sim 10^{-5}$	0.31	220	95	0.43	0.16
30000	6	3	$\sim 10^{-5}$	0.22	495	155	0.31	0.15
40000	6	4	$\sim 10^{-5}$	0.18	880	210	0.24	0.02
50000	6	4	$\sim 10^{-5}$	0.12	1375	265	0.19	0.02
5000	8	2	$\sim 10^{-6}$	3.32	10	70	7.00	1.47
10000	8	2	$\sim 10^{-6}$	1.72	55	105	1.91	1.25
20000	8	3	$\sim 10^{-6}$	0.66	220	185	0.81	0.17
30000	8	3	$\sim 10^{-6}$	0.38	495	235	0.47	0.16
40000	8	3	$\sim 10^{-6}$	0.28	880	305	0.35	0.16
50000	8	3	$\sim 10^{-6}$	0.23	1375	340	0.25	0.15
5000	10	2	$\sim 10^{-7}$	6.38	10	105	10.5	1.84
10000	10	2	$\sim 10^{-7}$	2.49	55	215	3.91	1.35
20000	10	3	$\sim 10^{-7}$	1.40	225	310	1.38	0.2
30000	10	3	$\sim 10^{-7}$	0.71	495	395	0.80	0.17
40000	10	3	$\sim 10^{-7}$	0.46	880	485	0.55	0.16
50000	10	3	$\sim 10^{-7}$	0.35	1375	525	0.38	0.16

Отметим, что приведенная теоретическая оценка для точности оказалась несколько хуже экспериментальной. Так, например, точность $\sim 10^{-7}$ достигалась уже при $p+1=10$. Приведем эти же результаты в виде графиков (см. рис. 7, 8).

Значения плотности заряда для квадратов, на которые разбивались пластины, выбирались случайным образом, однако все они были порядка $\frac{1}{S}$, где S — площадь одного квадрата.

Помимо экспериментов с умножением были проведены также эксперименты с использованием мультипольного умножения при решении уравнения $A\bar{q} = \bar{\psi}$ методом минимальных невязок (GMRES). При точности умножения $\sim 10^{-4}$ были получены решения с точностью $\sim 10^{-4}$. точность решения определялась следующим образом: сначала точно считался вектор $\bar{\psi} = A\bar{q}$, где вектор $\bar{q}$ выбирался произвольным, после чего численно решалось уравнение $A\bar{x} = \bar{\psi}$. Точность определялась как $\frac{\|\bar{q}-\bar{x}\|}{\|\bar{q}\|}$.

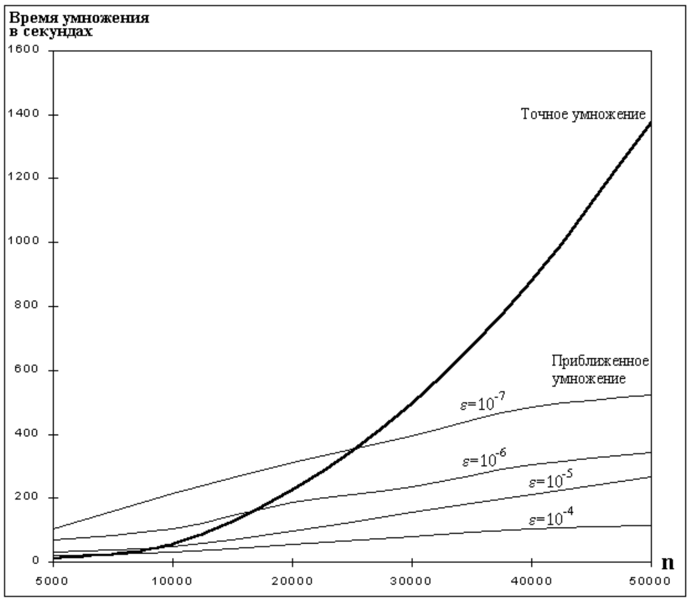


Рис. 7. Зависимость времени умножения от размера матрицы

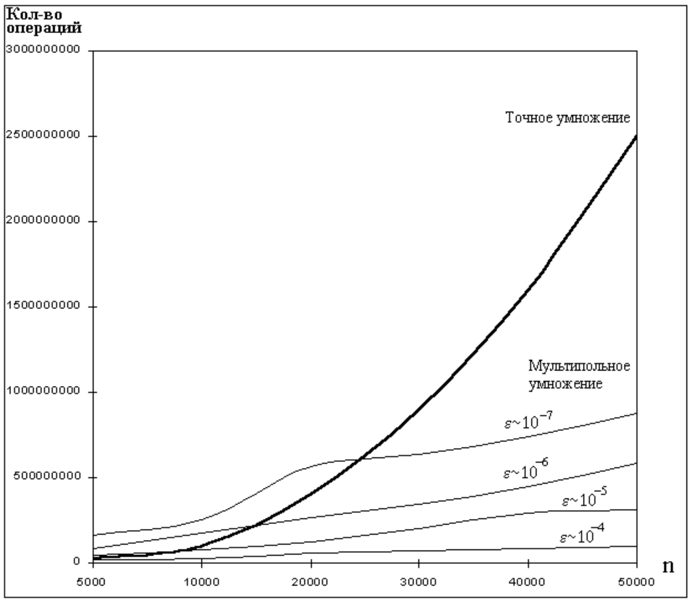


Рис. 8. Зависимость количества операций от размера матрицы

В заключение отметим, что для большой точности мультипольный алгоритм эффективен только для очень больших матриц, которые при его использовании все равно не помещаются в оперативной памяти обычного персонального компьютера. Однако для сравнительно небольшой точности он вполне эффективен и для не очень больших матриц.

7. ПРИМЕНЕНИЕ МУЛЬТИПОЛЬНОГО МЕТОДА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ТЕРМАЛЬНОГО АНАЛИЗА. Рассмотрим некоторый однородный теплопроводящий слой толщины h, на верхней поверхности которого находятся источники тепла. Источники будем считать плоскими, как и пластины в предыдущем разделе. На слой наложены два условия:

1) температура на его нижней границе постоянна;

2) поток тепла через верхнюю границу отсутствует.

Задача состоит в том, чтобы, зная распределение интенсивности источников, определить распределение температуры внутри некоторого участка слоя. В работе [3] для решения этой задачи используется следующий прием: предлагается от термальной задачи перейти к потенциальной. А именно, заменить температуру на потенциал (так как и то и другое — гармонические функции), интенсивность источников — на заряд, а теплопроводность — на диэлектрическую проницаемость. Теперь задача формулируется так: дана поверхностная плотность заряда на пластинах, надо найти потенциал внутри слоя. Два условия, накладываемые на потенциал, станут такими:

1) потенциал на нижней границе слоя равен нулю, то есть можно считать, что нижняя граница примыкает к проводнику;

2) производная потенциала по нормали у верхней поверхности равна нулю.

Построив такую аналогию, попробуем решить задачу с помощью мультипольного подхода. Для начала посмотрим, что из себя представляет такой потенциал.

Потенциал, создаваемый точечным зарядом в диэлектрике, соприкасающемся с проводником, строится методом изображений. Если заряд q находится на расстоянии h от нижней границы, то строим заряд $-q$ на расстоянии $-h$ от нижней границы, чтобы потенциал на ней стал равен нулю. Однако это создаст ненулевой поток через верхнюю границу, для устранения которого на расстоянии $2h$ от верхней границы построим новый заряд q. Он создаст ненулевой потенциал на нижней границе — его тоже нужно будет компенсировать... В итоге получим бесконечную цепочку зарядов, лежащих по обе стороны от слоя (рис. 9).

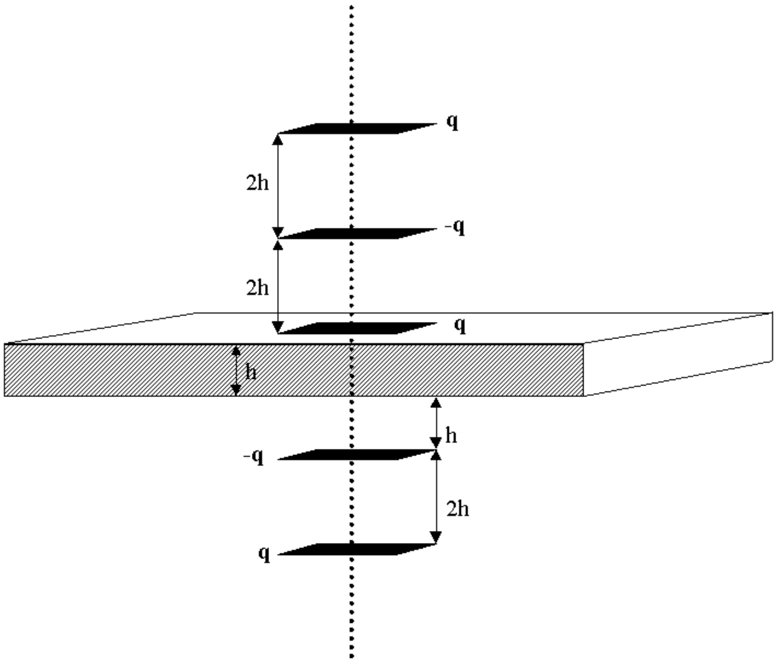


Рис. 9. Построение электрических изображений

Сразу отметим, что если требуется вычислить потенциал не в слое, а в параллелепипеде, на каждую боковую грань которого налагается одно из двух приведенных выше условий; потенциал строится посредством аналогичного отражения полученной цепочки по двум другим направлениям. Поэтому все, что будет далее предложено для бесконечного слоя, останется верным и для параллелепипеда.

Для вычисления потенциала в точке внутри слоя с координатами (x, y, z) нужно просуммировать ряд

$$q \cdot G(x, y, z|x_0, y_0, h) = q \cdot \frac{1}{2\pi\varepsilon} \left\{ \frac{1}{\sqrt{\xi^2+(\zeta-1)^2}} + \right.$$
$$+ \sum_{p=1}^{\infty} \left[-\frac{1}{\sqrt{\xi^2+(\zeta+4p-3)^2}} + \frac{1}{\sqrt{\xi^2+(\zeta+4p-1)^2}} - \right. \tag{25}$$
$$\left.\left. - \frac{1}{\sqrt{\xi^2+(\zeta-4p+1)^2}} + \frac{1}{\sqrt{\xi^2+(\zeta-4p-1)^2}} \right] \right\},$$

где ε — диэлектрическая проницаемость (в данном случае она же теплопроводность), $\zeta = \frac{z}{h}$, $\xi = \frac{\sqrt{(x_0-x)^2+(y_0-y)^2}}{h}$, (x_0, y_0, h) — координаты заряда. $G(x, y, z|x_0, y_0, h)$ есть не что иное, как функция Грина для данного потенциала. Таким образом, если поверхностная плотность заряда $q(x, y)$, то потенциал, создаваемый пластиной в точке, вычисляется по формуле

$$\psi(x, y, z) = \int\limits_S G(x, y, z|x_0, y_0, h)\, q(x_0, y_0)\, dx_0\, dy_0. \tag{26}$$

Поступим так же, как и в предыдущей задаче: разобьем пластины на N прямоугольных частей, которые будем считать равномерно заряженными. Тогда потенциал, создаваемый j-м прямоугольником в i-й точке, будет равен

$$\psi_{ij}(x, y, z) = q_j \cdot \int\limits_{S_j} G(x_i, y_i, z_i|x_0, y_0, h)\, dx_0\, dy_0,$$

где S_j — поверхность j-го прямоугольника. Таким образом, мы снова приходим к умножению матрицы на вектор:

$$\bar{\psi} = \begin{pmatrix} I_{11} & \dots & I_{1N} \\ \vdots & I_{ij} & \vdots \\ I_{1N} & \dots & I_{NN} \end{pmatrix} \times \begin{pmatrix} q_1 \\ \vdots \\ q_j \\ \vdots \\ q_N \end{pmatrix},$$

где

$$I_{i,j} = \int\limits_{S_j} G(x_i, y_i, z_i|x_0, y_0, h)\, dx_0\, dy_0.$$

Как в данном случае можно применить мультипольный алгоритм? Вообще говоря, точно так же, как и в предыдущей задаче. Так как все формулы для трансляций получены независимо от конкретного потенциала, то, значит, их можно применять для разложений любого потенциала. Поэтому достаточно получить мультипольные коэффициенты для каждого участка пластины, затем объединить их, преобразовывая при этом коэффициенты с помощью уже известных формул. Получить коэффициенты мультипольного разложения можно численно. Поскольку потенциал — гармоническая функция, он может быть приближен на сфере суммой сферических гармоник. Предположим, что мы построили сферу радиуса R_0 вокруг одной заряженной пластины и нашли такие коэффициенты $\{a_i^j\}$, $i = \overline{0,\infty}$, $j = \overline{-i,i}$, что в любой точке на сфере потенциал, создаваемый пластиной, равен $\sum\limits_{i=0}^{\infty}\sum\limits_{j=-i}^{i} a_i^j Y_i^j(\theta,\phi)$, где θ и ϕ — угловые координаты точки. Рассмотрим функцию $\frac{Y_i^j(\theta,\phi)}{R^{i+1}}$. Эта функция гармонична в любой области, не содержащей точку, в которой $R = 0$. Умножим ее на R_0^{i+1}. Полученная функция тоже гармонична, так как R_0 — константа. Следовательно, функция $\sum\limits_{i=0}^{\infty}\sum\limits_{j=-i}^{i} a_i^j Y_i^j(\theta,\phi)\left(\frac{R_0}{R}\right)^{i+1}$ тоже гармонична во всех точках вне сферы. То есть это гармоническая функция, значение которой на сфере совпадает со значением потенциала. В силу единственности решения уравнения Лапласа с условием Дирихле — вне сферы эта функция совпадает с потенциалом пластины. Ее мы и будем использовать в качестве мультипольного разложения. Мультипольные коэффициенты, таким образом, определяются как $M_n^m = a_n^m R_0^{n+1}$. В конечном итоге получим равенство

$$\int\limits_S G(R_0,\theta,\phi|x_0,y_0,h)dx_0dy_0 = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} a_n^m Y_n^m(\theta,\phi).$$

Таким образом, задача нахождения мультипольных коэффициентов сводится к задаче аппроксимации функции суммой сферических гармоник на сфере. Формула для нахождения мультипольных коэффициентов будет следующая:

$$M_n^m = a_n^m R_0^{n+1},$$

$$a_n^m = \int\limits_{-\pi}^{0}\int\limits_{0}^{2\pi} -Y_n^{-m*}(\theta,\phi)\sin(\theta)\cdot\left[\int\limits_S G(R_0,\theta,\phi|x_0,y_0,h)dx_0dy_0\right]d\theta d\phi.$$

Интегралы по сфере можно считать численно, взяв за центр сферы центр рассматриваемого участка. В качестве радиуса сферы имеет смысл взять удвоенную длину ребра куба уровня λ, то есть $R = \frac{h}{2^{\lambda-1}}$. Из теории сферических гармоник известно, что при приближении гармонической функции $F(R,\theta,\phi)$ на сфере с помощью сферических гармоник число членов в разложении определяется следующим образом: выбирается угол δ такой, что для любых точек на сфере (R, θ, ϕ) и (R, θ', ϕ'), таких, что угол между проведенными в них радиус-векторами меньше δ, выполняется неравенство

$$|F(R, \theta, \phi) - F(R, \theta', \phi')| < \frac{\varepsilon}{2},$$

где ε — желаемая ошибка. Тогда число членов в разложении, которое мы обозначим через p, определяется из неравенства

$$(p+1)\left(\frac{1+\cos\delta}{2}\right)^p < \frac{\varepsilon}{\max(F(\theta, \phi))},$$

где $\max(F(\theta, \phi))$ есть максимум модуля функции $F(R, \theta, \phi)$ на сфере. Данная оценка является достаточно общей. Значение p, полученное с ее помощью, может оказаться довольно большим. Экспериментальные результаты в этом случае значительно лучше. Рассмотрим случай, когда ребро куба в десять раз больше размера площадки, для которой ищутся мультипольные коэффициенты. Для этого случая экспериментально были получены значения точности разложения на сфере в нуле от числа членов в разложении $(p+1)$ для числа уровней λ, равного 3 и 4. Интегралы по сфере считались численно с относительной точностью $\sim 10^{-8}$. Затем вычислялись точные значения функции $G(0, 0, h|R, \theta, \phi)$ на сетке 1024×1024 и вычислялась разность между точными и приближенными значениями.

Таблица 4

ε	Число членов в мультипольном разложении	
	$\lambda = 3$	$\lambda = 4$
10^{-8}	20	16
10^{-7}	16	12
10^{-6}	13	10
10^{-4}	7	6
10^{-3}	5	5
10^{-2}	3	3
10^{-1}	2	2

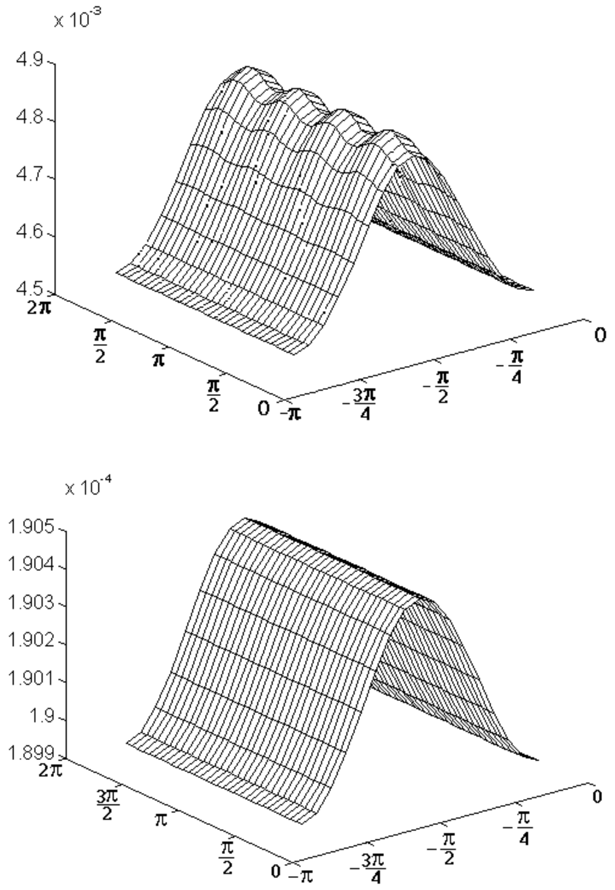


Рис. 10. Поле прямоугольной пластины на поверхности сферы

В табл. 4 под точностью ε подразумевается отношение

$$\varepsilon = \frac{\|\bar{G} - \bar{G}\|}{\|\bar{G}\|},$$

где $\bar{G}$ и $\bar{G}$ — векторы, составленные из точных и приближенных значений функции $G(0,0,h|R,\theta,\phi)$ на сфере.

На рис. 10 показана функция

$$\int\limits_{-\frac{a}{2}}^{\frac{a}{2}}\int\limits_{-\frac{a}{2}}^{\frac{a}{2}} G(R,\theta,\phi|x,y,h)dxdy,$$

где a — сторона заряженного квадрата, при $h = 16R$ и $a = \frac{R}{2}$ (верхний график) и $h = 16, a = \frac{R}{10}$ (нижний график). Как видно, при уменьшении a колебания значений на верхнем „гребне“ графика, обусловленные четырьмя углами квадрата, будут уменьшаться и при соотношении $a = \frac{R}{10}$ выродятся практически в прямую.

В качестве дополнения стоит упомянуть о способе быстрого суммирования ряда (25). В исходном виде ряд сходится медленно (как $\sum_{n=0}^{\infty}\frac{1}{n^2}$), однако можно существенно ускорить этот процесс, использовав следующий прием. Так как $G(x,y,0|x_0,y_0,h) = 0$, то будем суммировать ряд, порожденный разностью

$$G(x,y,z|x_0,y_0,h) - G(x,y,0|x_0,y_0,h),$$

сумма которого также равна $G(x,y,z|x_0,y_0,h)$. То есть вычислять будем сумму ряда

$$\begin{aligned}
&q\cdot G(x,y,z|x_0,y_0,h) = \\
&\quad = q\cdot\frac{1}{2\pi\varepsilon}\left\{\frac{1}{\sqrt{\xi^2+(\zeta-1)^2}}+\sum_{p=1}^{\infty}\left[-\frac{1}{\sqrt{\xi^2+(\zeta+4p-3)^2}}+\frac{1}{\sqrt{\xi^2+(\zeta+4p-1)^2}}-\right.\right.\\
&\qquad\qquad\left.\left.-\frac{1}{\sqrt{\xi^2+(\zeta-4p+1)^2}}+\frac{1}{\sqrt{\xi^2+(\zeta-4p-1)^2}}\right]\right\}\\
&\qquad -q\cdot\frac{1}{2\pi\varepsilon}\left\{\frac{1}{\sqrt{\xi^2+1}}+\sum_{p=1}^{\infty}\left[-\frac{1}{\sqrt{\xi^2+(4p-3)^2}}+\frac{1}{\sqrt{\xi^2+(4p-1)^2}}-\right.\right.\\
&\qquad\qquad\left.\left.-\frac{1}{\sqrt{\xi^2+(-4p+1)^2}}+\frac{1}{\sqrt{\xi^2+(-4p-1)^2}}\right]\right\}.
\end{aligned}$$

Полученный ряд будет сходиться значительно быстрее. Данный прием был использован при вычислении интегралов по сфере. При количестве слагаемых ~ 150 ряд сходился с точностью $\sim 10^{-10}$, в то время как исходный ряд сходится к такой точности при ~ 20000 слагаемых. Объясняется это тем, что чем дальше находятся изображения заряда от поверхности, тем слабее отличается поле, создаваемое ими в точке с координатами (x,y,z), $\;0 \leqslant z \leqslant h$, от поля, создаваемого ими в точке $(x, y, 0)$. Значит, суммы остаточных членов ряда для $G(x, y, 0|x_0, y_0, h)$ и ряда для $G(x, y, z|x_0, y_0, h)$ могут быть достаточно близки. Так как сумма остаточных членов ряда

$$q \cdot \frac{1}{2\pi\varepsilon}\left\{\frac{1}{\sqrt{\xi^2+1}} + \right.$$
$$\left. + \sum_{p=1}^{N}\left[-\frac{1}{\sqrt{\xi^2+(4p-3)^2}} + \frac{1}{\sqrt{\xi^2+(4p-1)^2}} - \frac{1}{\sqrt{\xi^2+(-4p+1)^2}} + \frac{1}{\sqrt{\xi^2+(-4p-1)^2}}\right]\right\}$$

равна

$$G(x, y, 0|x_0, y_0, h) - q \cdot \frac{1}{2\pi\varepsilon}\left\{\frac{1}{\sqrt{\xi^2+1}} + \right.$$
$$\left. + \sum_{p=1}^{N}\left[-\frac{1}{\sqrt{\xi^2+(4p-3)^2}} + \frac{1}{\sqrt{\xi^2+(4p-1)^2}} - \frac{1}{\sqrt{\xi^2+(-4p+1)^2}} + \frac{1}{\sqrt{\xi^2+(-4p-1)^2}}\right]\right\}$$
$$= -q \cdot \frac{1}{2\pi\varepsilon}\left\{\frac{1}{\sqrt{\xi^2+1}} + \right.$$
$$\left. + \sum_{p=1}^{N}\left[-\frac{1}{\sqrt{\xi^2+(4p-3)^2}} + \frac{1}{\sqrt{\xi^2+(4p-1)^2}} - \frac{1}{\sqrt{\xi^2+(-4p+1)^2}} + \frac{1}{\sqrt{\xi^2+(-4p-1)^2}}\right]\right\},$$

то из этого и следует, что

$$q \cdot G(x, y, z|x_0, y_0, h) \approx q \cdot \frac{1}{2\pi\varepsilon}\left\{\frac{1}{\sqrt{\xi^2+(\zeta-1)^2}} + \right.$$
$$\left. + \sum_{p=1}^{N}\left[-\frac{1}{\sqrt{\xi^2+(\zeta+4p-3)^2}} + \frac{1}{\sqrt{\xi^2+(\zeta+4p-1)^2}} - \frac{1}{\sqrt{\xi^2+(\zeta-4p+1)^2}} + \frac{1}{\sqrt{\xi^2+(\zeta-4p-1)^2}}\right]\right\}$$
$$- q \cdot \frac{1}{2\pi\varepsilon}\left\{\frac{1}{\sqrt{\xi^2+1}} + \right.$$
$$\left. + \sum_{p=1}^{N}\left[-\frac{1}{\sqrt{\xi^2+(4p-3)^2}} + \frac{1}{\sqrt{\xi^2+(4p-1)^2}} - \frac{1}{\sqrt{\xi^2+(-4p+1)^2}} + \frac{1}{\sqrt{\xi^2+(-4p-1)^2}}\right]\right\}.$$

Если разложить каждое слагаемое суммируемого ряда в ряд Тейлора по ζ, считая, что $\zeta < 1$ (а это действительно так), то, если аккуратно провести все выкладки, станет видно, что слагаемые полученного нами ряда убывают, как p^{-4}, то есть быстрее, чем слагаемые исходного ряда. С помощью этого приема можно значительно ускорить вычисление мультипольных коэффициентов. Полученные мультипольные коэффициенты подставляются в качестве начальных коэффициентов в алгоритм, после чего потенциал вычисляется так же, как и в предыдущей задаче, с таким же количеством операций и занимаемым объемом памяти.

Второй вариант использования мультипольного подхода, реально осуществленный для решения данной задачи, основан на прямом вычислении потенциала, создаваемого изображениями пластины, построенными над и под слоем. Выделим участок слоя размером $a \times a \times h$, на верхней поверхности которого лежит пластина. Для каждого изображения пластины, находящегося над и под слоем на расстоянии большем, чем $3a$ от поверхности, можно вычислить коэффициенты локального разложения относительно центра выделенного участка слоя. В самом деле, если рассматривать отдельно один участок пластины, который считаем равномерно заряженным, и все его изображения, то те из них, которые удалены от слоя на расстояние больше $2a$, создают в выделенном участке потенциал, пропорциональный плотности заряда. То есть если посчитать локальные коэффициенты для участка с единичной плотностью, можно находить локальные коэффициенты для любой плотности, просто домножая их на величину плотности. Это значит, что если мы, зная коэффициенты для разных участков пластины, с единичной плотностью, домножим их на значение плотности для каждого участка, затем просуммируем по всем участкам, мы получим коэффициенты локального разложения потенциала, создаваемого удаленными изображениями пластин. Потенциал, создаваемый близкими изображениями пластины, может быть вычислен с помощью медленного алгоритма, и, наконец, потенциал, создаваемый самой пластиной, может быть вычислен с помощью быстрого алгоритма. Итак, запишем последовательность действий такого алгоритма:

1. Разобьем пластину на N прямоугольных частей. Для каждой такой части существует бесконечное число изображений над и под слоем. Будем считать каждое изображение просто отдельным заряженным прямоугольником.

2. Для одного прямоугольника выберем все изображения, удаленные от слоя на расстояние больше a. Зададим некоторую относительную точность ε и вычислим для нее расстояние R такое, что $\frac{1}{R} < \varepsilon$. Рассмотрим j-е изображение i-го прямоугольника. Плотность заряда на нем — или $q^{(i)}$, или $-q^{(i)}$. Тогда потенциал, создаваемый им в точке $P = (r, \theta, \phi)$ внут-

ри слоя, может быть представлен в виде

$$\psi^{(j)}(P) = q^{(i)} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} L_n^{m(j)} Y_n^m(\theta, \phi) r^n,$$

где (r, θ, ϕ) — координаты точки P относительно центра рассматриваемого участка слоя. Тогда суммарный потенциал, создаваемый всеми удаленными изображениями, представляется в виде

$$\psi_{far}(P) \sum_{j=J_1}^{J_2} \psi^{(j)} = q^{(i)} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \left[\sum_{j=J_1}^{J_2} L_n^{m(j)} \right] Y_n^m(\theta, \phi) r^n,$$

то есть для каждого прямоугольника нужно посчитать коэффициенты

$$L_{n\ far}^{m(i)} = \sum_{j=J_1}^{J_2} L_n^{m(j)},$$

определяющие потенциал, создаваемый удаленными изображениями i-го прямоугольника. Расчет этих коэффициентов может быть осуществлен на предварительном этапе алгоритма, так как при многократном умножении будут использоваться одни и те же коэффициенты.

3. С помощью посчитанных коэффициентов можно вычислить коэффициенты локального разложения потенциала, создаваемого удаленными изображениями пластины, для всего выделенного участка слоя. Это потребует $2N(p+1)^2$ операций.

4. Для пластины проводится первый этап быстрого мультипольного алгоритма, а именно — считаются мультипольные коэффициенты для всех кубов.

5. Для ближних изображений пластины применяется медленный мультипольный алгоритм. Рассмотрим изображение пластины, находящееся над или под слоем на расстоянии d, $3a/\nu^{\frac{1}{3}} < d < 3a$. Разобьем изображение пластины так же точно, как и саму пластину на первом уровне разбиения при быстром алгоритме, то есть не менее, чем на $\nu^{\frac{2}{3}}$ частей. Если выделенный участок слоя — куб, а $\nu = 4$, то разобьем изображение

на четыре части. Мультипольные коэффициенты для таких частей могут отличаться от мультипольных коэффициентов соответствующих им частей пластины только знаком. Значит, их можно считать известными. Для каждой из таких частей изображения пластины можно посчитать преобразование $M \to L$ и найти локальные коэффициенты для каждого куба первого уровня разбиения. Если на первом уровне разбиения имеется ν кубов, то для одного изображения пластины это займет $\nu^{\frac{4}{3}}(p+1)^4$ операций. Это действие выполняется для всех попавших в диапазон $3\frac{a}{\nu^{\frac{1}{3}}} < d < 3a$ изображений.

6. Полученные для первого уровня локальные коэффициенты транслируются на второй уровень. Далее выполняются трансляции $M \to L$, как было описано в быстром алгоритме.

7. Шаг 5 выполняется для нового диапазона $3\frac{a}{\nu^{\frac{2}{3}}} < d < 3\frac{a}{\nu^{\frac{1}{3}}}$. Далее выполняется шаг 6 для второго уровня. Далее шаг 5 повторяется до тех пор, пока в выбираемом диапазоне не окажется ни одного изображения, а шаг 6 повторяется, как и в обычном быстром алгоритме, до уровня λ.

8. Как и в быстром алгоритме, для близких к пластине точек считаются точные значения потенциала.

9. Точные и приближенные значения суммируются.

Если для выделенного участка слоя $h < a$, то общая формула для оценки числа операций в этом случае такова:

$$oper_2 \sim 2\left(\frac{a}{h}\right)^5 (p+1)^4 + \nu^{\frac{4}{3}}\left(\frac{a}{h}\right)^4 (p+1)^4 + \\ + \left(\frac{a}{h}\right)^3 (p+1)^2 + 2N(p+1)^2 + oper_1,$$

где $oper_1$ — число операций, выполняемых быстрым мультипольным алгоритмом для расчета кулоновского потенциала, создаваемого заряженной пластиной, лежащей на верхней поверхности выделенного участка. В принципе, члены, добавленные в оценку числа операций, не зависят от N, за исключением члена $N(p+1)^2$, который не меняет оценки $oper \sim N\log_2^2 \varepsilon^{-1}$, однако при больших значениях $\frac{a}{h}$ число операций может достаточно сильно возрасти. Поэтому, безусловно, предпочтение следует отдать первому варианту применения мультипольного алгоритма, который вообще не увеличивает число операций, за исключением того, что при большой точности может потребоваться большее значение p, чем для второго варианта. К оценке для необходимого объема памяти данный алгоритм прибавляет $\approx 2(\frac{a}{h})^4(p+1)^4 + \nu^{\frac{4}{3}}(\frac{a}{h})^3(p+1)^4 + p^2$, что тоже достаточно много. Однако все-таки приведем табл. 5 и соответствующий график экспериментальных результатов (рис. 11), полученных с помощью данного алгоритма для $a = 5h$ и $a = 10h$ при $\nu = 8$. Все расчеты и измерения производились так же, как и для предыдущей задачи.

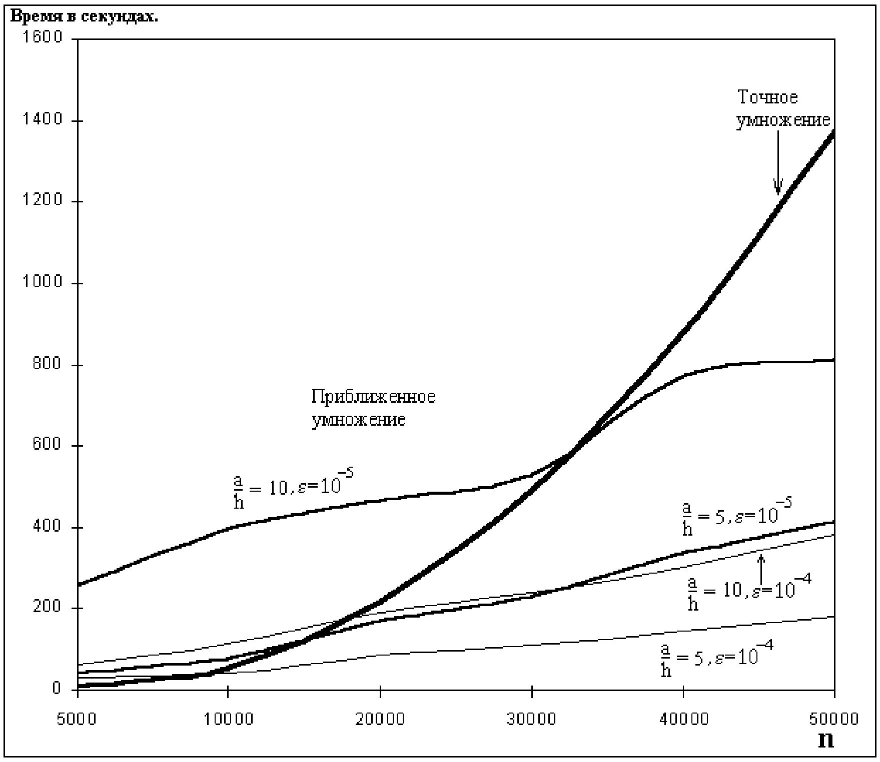


Рис. 11. Зависимость времени умножения от порядка матрицы

Таблица 5

$\frac{a}{h}$	$p+1$	ε	N	λ	$\frac{oper}{N^2}$	$\frac{mem}{N^2}$	$t_.$	$t_.$	$\frac{t_.}{t_.}$
10	6	$\sim 10^{-5}$	5000	2	20.54	3.15	10	255	25.50
10	6	$\sim 10^{-5}$	10000	3	5.47	0.65	55	395	7.18
10	6	$\sim 10^{-5}$	20000	3	1.48	0.28	220	465	2.11
10	6	$\sim 10^{-5}$	30000	3	0.74	0.21	495	530	1.07
10	6	$\sim 10^{-5}$	40000	4	0.48	0.05	880	770	0.88
10	6	$\sim 10^{-5}$	50000	4	0.31	0.04	1375	815	0.59
5	6	$\sim 10^{-5}$	5000	2	2.72	1.46	10	45	4.50
5	6	$\sim 10^{-5}$	10000	3	1.02	0.23	55	80	1.45
5	6	$\sim 10^{-5}$	20000	3	0.37	0.17	220	170	0.77
5	6	$\sim 10^{-5}$	30000	3	0.25	0.16	495	230	0.46
5	6	$\sim 10^{-5}$	40000	4	0.2	0.03	880	335	0.38
5	6	$\sim 10^{-5}$	50000	4	0.13	0.02	1375	415	0.30
10	4	$\sim 10^{-4}$	5000	3	4.36	0.55	10	65	6.50
10	4	$\sim 10^{-4}$	10000	3	1.2	0.25	55	115	2.09
10	4	$\sim 10^{-4}$	20000	4	0.38	0.05	220	195	0.89
10	4	$\sim 10^{-4}$	30000	4	0.18	0.03	495	240	0.48
10	4	$\sim 10^{-4}$	40000	4	0.11	0.03	880	305	0.35
10	4	$\sim 10^{-4}$	50000	4	0.08	0.02	1375	385	0.28
5	4	$\sim 10^{-4}$	10000	3	0.32	0.17	55	45	0.82
5	4	$\sim 10^{-4}$	20000	4	0.16	0.03	220	85	0.39
5	4	$\sim 10^{-4}$	30000	4	0.08	0.02	495	110	0.22
5	4	$\sim 10^{-4}$	40000	4	0.05	0.02	880	145	0.16
5	4	$\sim 10^{-4}$	50000	4	0.04	0.02	1375	180	0.13

Из таблицы видно, что, несмотря на недостатки, приведенный алгоритм тоже имеет смысл. Хорош он тем, что в принципе может быть применен к любому потенциалу, вычисляющемуся методом изображений. Хотя если с достаточной точностью удается суммой сферических гармоник аппроксимировать потенциал на сфере при достаточно небольшом p, то применение обычного быстрого алгоритма значительно эффективнее. Не стоит, конечно, забывать и про то, что при наличии условий на потенциал не только для верхней и нижней границы слоя, но и для четырех боковых граней число дополнительных операций увеличивается в $\sim (\frac{a}{h})^2$ раз, что уже много.

8. ПОДВЕДЕНИЕ ИТОГОВ. Итак, мы подробно изучили свойства мультипольного метода, быстрого мультипольного алгоритма и

рассмотрели некоторые области его применения. Были получены необходимые оценки, с помощью которых можно определять целесообразность использования мультипольного алгоритма при тех или иных условиях. Общий вывод относительно применения алгоритма может быть следующим: для матриц, порожденных гармоническим ядром интегрального уравнения в двумерном пространстве, оценка на число выполняемых операций в зависимости от размеров матрицы и достигаемой при этом точности такова:

$$oper \sim n \log_2 \varepsilon^{-1}.$$

Для матриц, порожденных гармоническим ядром интегрального уравнения в трехмерном пространстве, оценка на число операций в зависимости от точности умножения

$$oper \sim n \log_2^2 \varepsilon^{-1}.$$

Обе эти оценки достаточно хорошо подтверждены численными экспериментами (описание и результаты вычислений для задач на двумерном пространстве можно найти в работе [4]). Для сравнения приведем оценку на число операций, производимых при умножении матрицы на вектор, для *мозаично-скелетонного метода* (автор Е. Е. Тыртышников):

$$oper \sim n \log n \log_2^m \varepsilon^{-1},$$

где m — размерность пространства. Казалось бы, оценка несколько хуже, но метод мозаичного биения допускает работу со значительно более широким классом ядер — асимптотически гладкими ядрами, в то время как применение мультипольного метода на данный момент ограничено гармоническими ядрами. При этом мультипольный метод может быть использован пока только на пространстве размерности не выше трех, так как зависит от выбора системы функций, в то время как метод мозаичного биения в силу своей общности может быть использован для пространства любой размерности. Однако несмотря на упомянутые недостатки, мультипольный метод достаточно хорошо работает в своей области применения и, как показали численные эксперименты, вполне подходит для использования при численном решении весьма важного класса задач.

Литература

1. Rochlin V. Rapid solution of integral equations of classical potential theory // *J. Comput. Physics*. 1985. V. 60. P. 187–207.

2. Greengard L., Rokhlin V. The rapid evaluation of potential fields in three dimensions // *Lect. Notes Math.* 1988. V. 1360. P. 121–141.

3. Rizzoli V., Liparini A., Costanzo A., Frontini V. Three-dimensional computation of the thermal parameters of multiple-gate power FETs // *23rd European Microwave Conference Proceedings*. 1993. V. P3.1. P. 698–700.

4. Greengard L., Rokhlin V. A fast algorithm for particle simulations // *J. Comput. Physics*. 1987. V. 73. P. 325–348.

5. Тыртышников Е. Е. Методы быстрого умножения и решение уравнений // *Настоящий сборник.*

6. Nabors K., Korsmeyer F. T., Leighton F. T., White J. *Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional potential integral equations of the first kind.* Dept. of Electrical Eng. and Computer Science, Massachusetts Institute of Technology, 1994.

7. Tyrtyshnikov E. E. Mosaic-skeleton approximations // *Calcolo*. 1996. V. 33(1-2). P. 47–57.

8. Соболев С. Л. *Уравнения математической физики.* — М.: ГИТТЛ, 1954.

9. Newman J. N. Distributions of sources and normal dipoles over quadrilateral panel // *J. of Eng. Mathematics*. 1986. V. 20. P. 113–126.

Интерполяционный метод быстрого приближенного умножения матрицы, порожденной функцией на контуре*

И. Ю. Никольский

Аннотация

Рассмотрен метод быстрого матричного умножения в виде рекурсивного приближенного представления исходной матрицы $A \approx A_0 = S_0 + U_0 A_1 U_0^T,\ i = 1, ..., k$. *Порождающая функция, заданная на контуре, интерполируется на иерархии сеток в* $\mathbb{R}^2$. *Рассмотрен простейший вариант лагранжевой интерполяции и логарифмическая порождающая функция.*

Введение. Под *интерполяционным методом быстрого матричного умножения* будем понимать такое сведение задачи $y = Ax$ к задаче $\tilde{y} = \tilde{A}x$ с малой нормой $||y - \tilde{y}||$, когда элементы матрицы A получаются путем интерполяции элементами матрицы $\tilde{A}$.

Для иллюстрации возможных подходов рассмотрим несколько формальных задач из области быстрого умножения функционально порожденных матриц.

- Требуется умножить матрицу A, порожденную функцией на произвольном множестве точек в $\mathbb{R}^n$, на вектор. Имеется множество точек $\{\omega\}$, на котором та же функция порождает матрицу T. Матрицу T можно быстро умножить на вектор, например за $n \log_2 n$ операций.

- Есть набор матриц $\{A_k\}, k = 1, .., K$, и набор векторов $\{x_k\}$, для всех k фиксирована порождающая матрицу A функция

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155)

f. Для каждого k свое дискретное множество задания функции $f = f(\omega_i^{(k)}, \omega_j^{(k)})$, где $\omega_{i,j} \in \{\omega^{(k)}\}$. Причем области задания $\{\omega^{(k)}\}$, $k \geqslant 1$, и функция f „допускают“ интерполяцию значений функции f с $\omega^{(1)}$ на $\omega^{(k)}$ для любого $k > 1$. Вычисление порождающей функции f и, как следствие, элементов матрицы $a_{ij}^{(k)}$ требует много операций. Необходимо получить набор векторов $\{y_k\}$, $y_k = A_k x_k$.

Эти задачи допускают представление матрицы A в следующем виде:

$$A \cong S + UTV, \;\; \dim(A) = N, \;\; \dim(T) = n,$$

где матрица S — разреженная вырезка из матрицы A, в S попадают только те элементы A, которые плохо интерполируются. Матрицы U и V мы будем называть матрицами интерполяции и антерполяции соответственно, эти матрицы также разрежены, их элементы — коэффициенты интерполяции. T — или допускающая быстрое умножение (первый вариант), или $T = A_1$, то есть матрица, порожденная с множества $\{\omega^{(1)}\}$ (второй вариант).

Почему такое представление может давать преимущество перед обычным умножением матрицы A на вектор? Умножение S, U, V на вектор требует $O(N)$ операций. И если умножение T на вектор в первом случае требует в асимптотике по N меньше операций, чем $O(N^2)$, то умножение приближенного представления A на вектор „обходится нам дешевле“.*

Во второй задаче выигрыш по арифметическим операциям при умножении набора матриц на вектор имеет другую природу. В этом случае мы не вычисляем элементы матриц A_k для $k \geqslant 2$, и если затраты на вычисление функции f превосходят затраты на ее интерполяцию, то суммарное количество арифметических операций окажется меньше. Аналогично экономится и память, так как при плотном хранении набора матриц $\{A_k\}$ требуется Kn^2 ячеек памяти, а в приближенном формате $n^2 + (K-1)C_{\mathcal{J}}n$ ячеек, где $C_{\mathcal{J}}$ — константа интерполяции, не зависящая от n.

Естественно, наилучшими задачами для интерполяционного метода являются задачи, являющиеся „объединением“ двух вышеизложенных. Отметим, что мы не преследуем цели в данной статье

---

*Значения размерностей A и T зависят друг от друга, для простоты изложения асимптотические значения приведены к N.

привести полную картину возможных применений интерполяционного метода.

В дальнейшем нас будет интересовать только вариант первой задачи, причем тот случай, когда зависимость n от N квадратична ($n = O(N^2)$).

Размерности областей задания функции f для матриц A и T могут отличаться друг от друга. Например, функцию, заданную в $\mathbb{R}^2$, мы можем интерполировать с плоского множества на контур; или функцию, заданную в $\mathbb{R}^3$, можно интерполировать с куба на поверхность. Для простоты изложения интерполяцию с множества размерности m на множество размерности n обозначим $\mathbb{R}^n \to \mathbb{R}^m$. В случаях $\mathbb{R}^2 \to \mathbb{R}^2$, и даже $\mathbb{R}^2 \to \mathbb{R}^3$, равномерный выбор сетки и учет структуры T позволяют получить выигрыш. Количества операций в случае $\mathbb{R}^2 \to \mathbb{R}^3$ для точного и приближенного умножения связаны друг с другом следующим неравенством:

$$N^2 > \frac{3}{2} N^{\frac{3}{2}} \log_2 N,$$

где N — число точек дискретизации задачи на размерность.

В данной работе рассмотрен случай антерполяции с контура, заданного произвольной дискретизацией, на плоское множество. Простой подсчет показывает, что одного шага антерполяции не хватает для получения выигрыша, так как квадратичный рост размерности (количества точек дискретизации) сводит на нет логарифмический выигрыш от выбора равномерной сетки. Неравенство $N^2 < 2N^2 \log_2 N$ выполняется всегда для $N > 1$.

Покажем, что накладывание условий на контур типа „покрываемый узкой полосой“ или „малоотклоняющийся от окружности“ не дает желаемого „выигрыша“ по количеству операций. Для матриц, порожденных с контуров такого рода, попробуем построить приближенное умножение на вектор так, чтобы соотношения в условиях на геометрию задачи не зависели от N.

Возьмем равномерное пропорциональное разбиение прямоугольника $\Sigma = [a_1, a_2] \times [b_1, b_2]$, $a = a_2 - a_1$, $b = b_2 - b_1$, $\frac{a}{n_a} = \frac{b}{n_b}$ (n_a, n_b — число точек на сторону прямоугольника). Число точек на контуре обозначим N. Рассмотрим пропорциональное разбиение контура $N = {}_{\Gamma}(n_a + n_b)$, где C_Γ — коэффициент, характеризующий „заполнение“ контуром прямоугольника. Прямое умножение A на вектор требует N^2 операций. Матрица $A \approx UTV$ умножается на вектор за $n_a n_b \log_2(n_a n_b)$ операций. Следующая цепочка отношений

$$k = \frac{n_a}{n_b}, \ \ n_b = \frac{n_a}{k},$$

$$C_\Gamma^2\left(1 + \frac{1}{k}\right)^2 {n_a}^2 \geqslant C_\Gamma n_a^2 \log_2 \frac{n_a}{\sqrt{k}},$$

$$\big(1 + \frac{1}{k})^2 k + \frac{1}{C_\Gamma} \log_2 k \geqslant \frac{1}{C_\Gamma} \log_2 n_a,$$

$$k \geqslant \log_2 n_a,$$

показывает, что $k \neq \frac{a}{b}$ при изменении N. Итак, в геометрическое условие на контур должен входить параметр дискретизации задачи N. Следовательно, равномерной по N „узкой полосы“ не существует.

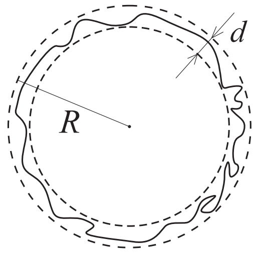
С другой стороны, для конкретной дискретизации задачи мы имеем условие на контур, позволяющее выяснить: получим ли мы выигрыш при антерполяции на равномерную сетку на прямоугольнике. Попробуем понять, в каких случаях можно умножить матрицу, порожденную с контура, удовлетворяющего такому условию, на вектор за $N \log_2 N$ операций. Последовательность выкладок

$$C_\Gamma(k+1) \log_2 C_\Gamma n_b(k+1) \sim k n_b^2 \log_2 k n_b^2,$$

$$C_\Gamma n_b(k+1) \sim k n_b^2, \ \ \ n_b \sim C_\Gamma,$$

показывает, что требуемое условие на дискретизацию принимает более жесткую форму в виде зависимости от коэффициента C_Γ. Данное ограничение может не позволить достичь приемлемой точности. Рассмотрим простой пример. Зададим контур в виде границы прямоугольника. Коэффициент заполнения $C_\Gamma = 2$, как следствие $n_b \sim 2$. Поэтому мы не можем рассматривать дискретизацию исходной задачи с числом точек более 2 на меньшую сторону прямоугольника. Что, естественно, слишком мало.

Не проходит и аналогичная попытка свести задачу для контуров, „мало“ отклоняющихся от окружности, к блочно-циркулянтной матрице с небольшим размером блока.



Умножение за $CN\log_2 N$ операций, что является достоинством семейства циркулянтных матриц, не удается проделать для любых N с константой C, не зависящей от N. Константа C есть функция от размера блока циркулянта, который, в свою очередь, есть функция от ширины кольца, покрывающего контур. Пусть $k = \frac{d}{R}$, d — ширина кольца, R — внешний радиус кольца, аналогично предыдущему случаю $k \sim \log_2 n_R$, а тогда нельзя ввести понятие „узкое кольцо“, равномерное по N.

Как и в предыдущем случае, анализ получения логарифмического ускорения приводит к той же проблеме. Чтобы получить логарифмическое ускорение, размер блока циркулянтной матрицы должен быть пропорционален C_Γ и, за исключением совпадения контура с окружностью, слишком мал для достаточной точности интерполяции.

Тем не менее мы предлагаем метод приближенного умножения матрицы A на вектор и в случае $\mathbb{R}^1 \to \mathbb{R}^2$. Для этого мы строим иерархическую последовательность полос $S^{(k)} \subset \mathbb{R}^2$ (рис. 1), покрывающих контур Γ, в виде объединения шаблонов $S^{(k)} = \bigcup_i^{n^{(k)}} \sigma_i^{(k)}$, что позволяет и достичь приемлемой точности при фиксированном N, и учесть свойства контура при росте N.

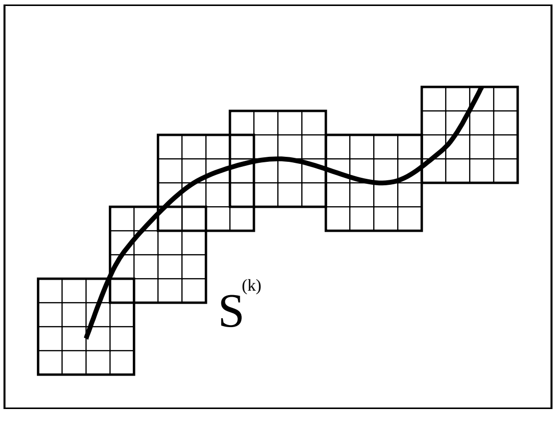
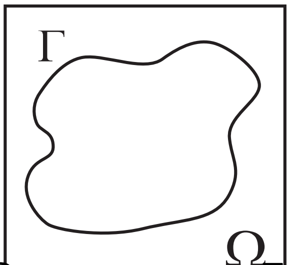


Рис. 1. Первый шаг интерполяции с $\mathbb{R}^1$ на $\mathbb{R}^2$

Формально эту задачу можно переформулировать как быстрое приближенное умножение структурированной матрицы на разреженный вектор, из-за получаемого логарифмического выигрыша, как и от применения быстрого преобразования Фурье (БПФ). Но в дальнейшем мы покажем, что метод работает и для неравномерных сеток. Поэтому решенная задача есть как минимум обобщение таким образом сформулированой задачи.

Идея интерполяционного метода в применении к умножению матрицы на вектор развивается с восьмидесятых годов нашего века. Основные подходы можно найти в работах, посвященных быстрым алгоритмам матричного умножения [1], [2], [5].

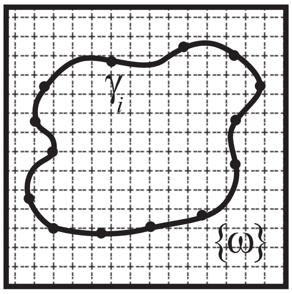
1. ПОСТАНОВКА ЗАДАЧИ. Рассмотрим функционально порожденную матрицу A, заданную произвольной дискретизацией $\{\gamma_i\}_{i=1}^N$ контура Γ в $\mathbb{R}^2$. То есть в $\mathbb{R}^2$ задана $f(x,y)$, а $A = [a_{ij}]_{i,j=1}^N$, где $a_{ij} = f(\gamma_i,\gamma_j)$. Рассмотрим квадрат $\Omega = [a,b]\times[a,b]$, $\Gamma \in \Omega$. Для умножения A на произвольный вектор x требуется $O(N^2)$ операций.



Определение. Будем называть $f(x,y) \in \mathbb{R}^\nu$ асимптотически гладкой функцией, если для $\forall p$

$$|\partial^p f(x,y)| \leqslant C d^p p! |x-y|^{g-p},$$

где ∂^p обозначает любую смешанную производную порядка p и C, d, g не зависят от p (см.[2], [4]).
Мы покажем, как сократить количество операций для асимптотически гладких функций путем интерполяции с Ω на Γ.



На Ω зададим равномерную сетку $\{\omega_i\}_1^n$, где n — число точек сетки на сторону квадрата. В точках сетки $\{\omega_i\}$ построим матрицу T, $t_{ij} = f(\omega_i,\omega_j)$. Из-за равномерности сетки $\{\omega\}$ полученная матрица T — дважды теплицева, и, следовательно, для умножения этой матрицы на вектор имеются быстрые алгоритмы на базе БПФ, позволяющие сократить умножение до $O(n^2\log_2 n)$ операций. Рассмотрим равномерный интерполяционный *шаблон* σ с порядком интерполяции m, заданный прямым произведением интерполяционных полиномов Лагранжа:

$$\alpha_{ij}(x^1,x^2) = l_i(x^1)l_j(x^2),\ \ l_i(x^{1,2}) = \prod_{\substack{k=0\\k\neq i}}^{m} \frac{x^{1,2}-x_k^{1,2}}{x_i^{1,2}-x_k^{1,2}}\,, \qquad (1)$$

где $(x^1,x^2) \in \mathbb{R}^2$, а x_k^i — узлы интерполяции.

В дальнейшем под шаблоном в зависимости от контекста мы будем понимать как набор узел – коэффициент интерполяции, так и геометрический объект (прямоугольник).

С помощью пар шаблонов проинтерполируем функцию $f(x,y)$:

$$f(\gamma_i,\gamma_j) \cong \sum_{k=1}^{m^2} \alpha_k^i \sum_{l=1}^{m^2} \beta_l^j f(\omega_k,\omega_l).$$

Отсюда можно переписать матричное равенство в следующем виде:

$$Ax = (S + UTV + E)x, \tag{2}$$

где S — разреженная матрица, U, V — матрицы интерполяции и антерполяции соответственно, E — матрица ошибки.

Элементы матриц интерполяции и антерполяции выписываются через интерполяционные коэффициенты следующим образом:

$$u_{ij} = \alpha_{L_1(j)L_2(j)}(x_i),$$
$$v_{ij} = \beta_{L_1(i)L_2(i)}(x_j),$$

где $L_1(i), L_2(i)$ — функции пересчета с глобальной сетки ω на локальную сетку шаблона.

При данном выборе интерполяции (точки x и y интерполируются одинаковым методом с одними и теми же параметрами) матрица $V = U^T$. Заметим, что матрицы интерполяции и антерполяции сильно разрежены, в строке матрицы U (столбец для V) ровно $(m+1)^2$ элементов. А так как размер матрицы $T \quad n$ может оказаться много больше порядка интерполяции m, то вычисление всех элементов матрицы не является необходимым условием для решения задачи. Геометрически этот факт интерпретируется следующим образом: в интерполяции принимают участие только точки полосы $S \ \omega_k = \{\tilde{\omega}_i\} \subset S$, и нет нужды умножать на все элементы матрицы T.

Введем дополнительные обозначения.

Пусть h — шаг равномерного шаблона (сетки), с которого интерполируется точка x. Соответственно, узлы шаблона задаются формулой $\vec{x_{ij}} = \vec{x_\sigma} + \overrightarrow{(ih, jh)}$, $i = 0, \ldots, m$, x^σ — абсолютная координата левого нижнего угла шаблона.

Будем называть *мерой центральности* и обозначать μ^* расстояние в шагах шаблона h, и с ее помощью задавать внутреннюю область шаблона, вне которой не могут лежать точки дискретной области задания функции, интерполируемые данным шаблоном.

Ей соответствует *мера нецентральности* $\mu = \frac{m-\mu^*}{2}$ шаблона.

Итак, если точка $x = (x^1, x^2)$ удовлетворяет условию

$$x^1_\sigma + \mu h \leqslant x^1 \leqslant x^1_\sigma + (m-\mu)h, \quad x^2_\sigma + \mu h \leqslant x^2 \leqslant x^2_\sigma + (m-\mu)h \quad (3)$$

для данного шаблона, то x может антерполироваться на него.

Будем называть *величиной ближней зоны* и обозначать R максимальное расстояние в шагах сетки h между точками, на которых функция f не интерполируется, а вычисляется явным образом. В матричной записи (2) этим элементам соответствует разреженная матрица S.

Таким образом, $f(x, y)$ не интерполируется с сетки ω, а вычисляется явно, если

$$|x - y| \leqslant Rh.$$

2. АЛГОРИТМ И ОЦЕНКИ. Рассмотрим иерархическое представление матрицы A в факторизованном виде

$$A = A_0 \simeq S_0 + U_0 A_1 U_0^T, \; A_{k-1} \simeq S_{k-1} + U_{k-1} A_k U_{k-1}^T. \quad (4)$$

Данное представление получается последовательной интерполяцией порождающей матрицу A функции $f(x, y)$ с контура Γ на полосу S_1, с полосы S_k на S_{k+1}. Пусть контур Γ задан дискретизацией по параметру ϕ

$$x = X(\phi), \; y = Y(\phi), \; 0 \leqslant \phi \leqslant 1,$$

и дискретизация контура задана множеством значений параметра $\{\phi_i\}_{i=1}^n$, где $\phi_i < \phi_{i+1}$. Полоса первого уровня получается из объединения шаблонов по следующему алгоритму.

Пусть первые k точек контура $\{\gamma_i\}_{i=1}^k$ покрыты объединением шаблонов $\bigcup_{j=1}^{l_k} \sigma_j$. Тогда $l_k + 1$-й шаблон строится с удовлетворением следующих двух условий ($x \in \sigma$ в нижеперечисленных условиях означает выполнение условий (3)):

- $\gamma_{k+1} \in \sigma_{l_k+1}$, следующая точка вне зависимости от своей принадлежности уже построенной части полосы S_1 принадлежит центральной части строящегося шаблона,
- $\sigma_{l_k+1} = \arg\ \max\ N_{k,k+q}(\sigma), \;\; N_{i,j}(\sigma) = \sum_{l=i}^{j} \nu_\sigma(\gamma_l),$ где

$\nu_\sigma(\gamma_l) = \nu_\sigma(\gamma_{l-1}) \times \begin{cases} 1, & \text{если } \gamma_l \in \sigma, \\ 0 & \text{— иначе,} \end{cases}$ $\nu_\sigma(\gamma_1) = 1$, так как из всех шаблонов выбирается шаблон, максимизирующий последовательное количество точек множества $\{\gamma_i\}_{i=k+1}^{k+q}$ (q — параметр, от которого зависит как предварительная работа по приближенной факторизации матрицы, так и фактические затраты на умножение).

Полоса S_k строится по полосе S_{k-1} формально как попарное объединение шаблонов k-го уровня. То есть для всех нечетных $0 < i \leqslant n^{(k-1)}$ шаблон $\sigma^{(k)}_{\frac{i+1}{2}}$ выбирается с центром посередине отрезка, соединяющего центры шаблонов σ_i^{k-1} и σ_{i+1}^{k-1}.

Утверждение. Пусть для центров шаблонов $k-1$-го уровня выполнено условие $|C_i^{(k-1)} - C_{i+1}^{(k-1)}| \leqslant m^{(k-1)} - 4\mu^{(k-1)}$ (рис. 2) для $i = \overline{1, N}$. Тогда

$$|C_i^{(k)} - C_{i+1}^{(k)}| \leqslant m^{(k)} - 4\mu^{(k)}, \qquad \forall i = 1, \ldots, \left[\tfrac{N}{2}\right].$$

Доказательство. Обозначим через C_1 и C_2 центры шаблонов k-го уровня, покрываемых шаблоном $S_i^{(k)}$, а через C_3 и C_4 — центры шаблонов, покрываемых шаблоном $S_{i+1}^{(k)}$. Простое преобразование

$$\begin{aligned} |C_i^{(k)} - C_{i+1}^{(k)}| &= \tfrac{1}{2}|C_1 - C_2 + C_2 - C_3 + C_3 - C_4 + C_2 - C_3| \\ &\leqslant 2(m^{(k-1)} - 4\mu^{(k-1)}) = m^{(k)} - 4\mu^{(k)} \end{aligned}$$

и доказывает утверждение. ■

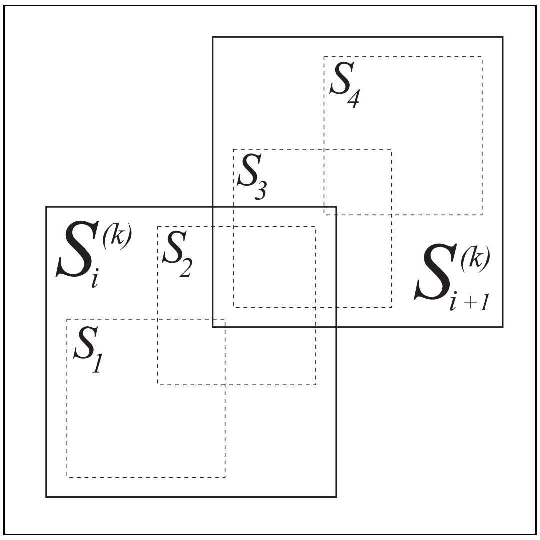


Рис. 2. Переход с $(k-1)$-го шага на k-й

Это утверждение дает нам третье условие на построение полосы первого уровня. При его выполнении построенное объединение шаблонов на k-м уровне покрывает контур и все полосы с 1 по $k-1$. Следовательно, мы получаем естественное заполнение матриц U_k и U_k^T в разреженном формате как построковую вырезку ($n^{(k)}$ строк — количество точек в k-й полосе) с фиксированным количеством элементов в строке $(m^{(k+1)}+1)^2$, что соответствует порядку интерполяции с k-го на $(k+1)$-й уровень.

Запишем идею умножения в виде алгоритма:

- переход от контура к полосе $\mathbb{R}^1 \to \mathbb{R}^2$: $v_1 = U_0^T x$;
- $v_i = U_{i-1}^T v_{i-1}$, $i = 1, \ldots, k-1$;
- $v_k = A_k v_{k-1}$;
- $y = v_{i-1}$, $v_{i-1} = U_{i-1} v_i$, $v_{i-1} = v_{i-1} + S_{i-1} y$, $i = k, \ldots, 2$;
- переход от полосы к контуру $\mathbb{R}^2 \to \mathbb{R}^1$: $y = S_0 x + U_0 v_1$.

Сказанного выше было бы достаточно для использования метода на практике, если бы мы могли позволить себе роскошь хранить вектор $v = U^T x$ в плотном формате. Но при переходе с контура на полосу размерность вектора увеличивается квадратично, соответственно размер вектора по порядку величины становится равным размеру матрицы, что, естественно, неприемлемо для любого метода быстрого умножения (помимо уменьшения количества операций, затрачиваемых на умножение, мы хотим получить и выигрыш по памяти).

Пусть вектор $v_k = V^k x$ хранится произвольным образом, его i-й элемент соответствует i-й точке произвольно занумерованной полосы k, тогда множество точек $\{\omega_i\}_{i=1}^{(m+1)^2}$ полосы $k+1$, необходимых для ее интерполяции, выбирается как множество точек шаблона σ, доставляющего минимум расстоянию от точки до своего центра:

$$\sigma = \arg \max_{\sigma \in \{\sigma_i^{(k+1)}\}} |x_i - C_\sigma|.$$

Следовательно, на момент перехода с $(k-1)$-го уровня на k-й необходимо хранить историю получения точек $(k-1)$-й полосы.

Теперь осталось подсчитать число операций и оценить ошибку $|a_{ij} - \tilde{a}_{ij}|$, где

$$\tilde{A} = S_0 + U_0(S_1 + U_1(...(S_k + U_k A_{k+1} U_k^T)...)U_1^T)U_0^T.$$

Число операций на i-м уровне

$$N_{flop}(i) = (C_i(R) + m^2)n^{(i)},$$

где $n^{(i)}$ — число точек в i-й полосе, $C_i(R)$ — константа, зависящая только от величины ближней зоны, m — порядок интерполяции. Общее количество арифметических операций

$$N_{flop} \leqslant (C(R) + m^2)\sum_{i=1}^{k} n^{(i)}, \qquad C(R) = \max_{1\leqslant i\leqslant k} C_i.$$

Уменьшение в два раза количества шаблонов при переходе с уровня на уровень, количество точек в объединении которых и есть $n^{(i)}$, позволяет ввести рекуррентное соотношение $n^{(i-1)} = \nu n^{(i)}$, где $1 < q \leqslant \nu < 2$. Количество шаблонов в первой полосе не может быть больше количества точек дискретизации контура, следовательно,

$$N_{flop} \leqslant (C(R) + m^2)\sum_{i=1}^{k} \frac{Nm^2}{\nu^{i-1}},$$

$$N_{flop} \leqslant Nm^2(C(R) + m^2)\frac{\nu^k - 1}{\nu - 1}. \tag{5}$$

Выбирая критерием останова рекуррентного выражения (4) следующее условие:

$$(n^{(k)})^2 = (C(R) + m^2)\sum_{i=1}^{k-1} n^{(i)},$$

равенство (5) можно переписать в виде, не зависящем от числа уровней иерархии:

$$N_{flop} = C(R, m)N\log_\nu N. \tag{6}$$

Оценку погрешности получаемой матрицы проведем в два этапа: на первом получим поэлементную оценку при переходе с уровня $k-1$ на уровень k, затем, мажорировав число уровней, мы сможем получить оценку на ошибку в элементе заданной матрицы.

Воспользовавшись теоремой об остаточном члене многомерной интерполяционной формулы (см. [2]), погрешность в элементе $a_{ij} = a_{ij}^{(l)}$ для шага l запишем в виде неравенства

$$|a_{ij} - p(a_{ij})| \leqslant \sum_{k=1}^{4} (-1)^{k-1} \sum_{1 \leqslant i_1 < \ldots < i_k \leqslant 4} D(i_1, \ldots, i_k),$$

где

$$D(i_1, \ldots, i_k) \equiv \frac{1}{p_{i_1}! \ldots p_{i_k}!} \frac{\partial^{p_{i_1}} \ldots \partial^{p_{i_m}}}{\partial x_{i_1}^{p_{i_1}} \ldots \partial x_{i_m}^{p_{i_m}}} f(\xi_{i_1, \ldots, i_m}) w_{i_1}(x_{i_1}) \ldots w_{i_m}(x_{i_m}),$$

$$w_i(t) = \prod_{j=1}^{p_i} (t - x_i^j).$$

Учтем симметрию по x_1, x_2, y_1, y_2; тогда неравенство перепишется в виде

$$\begin{aligned} |a_{ij} - p(a_{ij})| \leqslant\ & 4D(i_1) - 6D(i_1, i_2) + 4D(i_1, i_2, i_3) \\ & - D(i_1, i_2, i_3, i_4). \end{aligned} \tag{7}$$

Оценим члены неравенства по отдельности.

$$D(i_1) \leqslant \frac{1}{m!} h^m (m - \mu)! \mu! \frac{\partial^m f}{\partial x_1^m}$$

$$\leqslant \frac{(m - \mu)! \mu!}{m!} h^m C d^m m! \frac{1}{\min |x - y|^{m-g}},$$

где C, d, g — константы из определения асимптотически гладкой функции.

Аналогично получаем

$$D(i_1, i_2) \leqslant \frac{((m - \mu)! \mu!)^2}{(m!)^2} h^{2m} C d^{2m} (2m)! \frac{1}{\min |x - y|^{2m-g}},$$

$$D(i_1, i_2, i_3) \leqslant \frac{((m - \mu)! \mu!)^3}{(m!)^3} h^{3m} C d^{3m} (3m)! \frac{1}{\min |x - y|^{3m-g}},$$

$$D(i_1, i_2, i_3, i_4) \leqslant \frac{((m-\mu)!\mu!)^4}{(m!)^4} h^{4m} C d^{4m} (4m)! \frac{1}{\min|x-y|^{4m-g}}.$$

Воспользовавшись тем фактом, что при расстоянии меньшем, чем R узлов сетки, функция не интерполируется, а вычисляется явным образом, легко оценим $\min|x-y|$:

$$\min_{i_1,i_2,j_1,j_2} |x_{i_1 i_2} - y_{j_1 j_2}| = Rh.$$

Собирая все члены в (7), получим поэлементную оценку погрешности факторизации матрицы на одном шаге:

$$|a_{ij} - p(a_{ij})| \leqslant 15C((m-\mu)!\mu!)^4 \frac{(4m)!}{(m!)^4} h^{2m+g} R^{g-m}. \tag{8}$$

Для получения ошибки в исходной матрице осталось учесть количество уровней иерархии, изменение шага сетки при переходе с уровня на уровень, а также тот факт, что в интерполяции участвуют значения функции, сами заданные с ошибкой. Всего уровней $O(\log_\nu N)$, в интерполяции $a_{ij}^{(l-1)}$ участвуют m^4 элементов матрицы A_l, размер сетки увеличивается в 2 раза при переходе с уровня на уровень.

Имеем

$$|e_{ij}| \leqslant C_0(m, \mu, C) \log_\nu N m^{4\log_\nu N} (2^{m \log_\nu N} - 1) h_0^{2m+g} R^{g-m}.$$

Шаг начальной сетки можно оценить как $h \cong \frac{\operatorname{diam}\Omega}{N} = O(N)$. Выразив R через $\varepsilon = |e_{ij}|$, получаем количество арифметических затрат в виде утверждения следующей теоремы.

Теорема. Вышеописанным алгоритмом можно построить приближенную факторизацию матрицы A с ошибкой в элементе меньшей, чем ε, и позволяющую умножить матрицу A на вектор за

$$N_{flop} = O(\varepsilon^{-\frac{1}{m}} N \log_\nu N), \quad 1 < q \leqslant \nu \leqslant 2,$$

операций.

Литература

1. Нечепуренко Ю. М. *Быстрые численно устойчивые алгоритмы для широкого класса линейных дискретных преобразований.* — М.: 1985. Препринт ОВМ АН СССР №98.

2. Тыртышников Е. Е. Методы быстрого умножения и решение уравнений // *В настоящем сборнике.*

3. Горейнов С. А. Мозаично-скелетонные аппроксимации матриц, порожденных асимптотически гладкими и осцилляционными ядрами // *В настоящем сборнике.*

4. Tyrtyshnikov E. E. Mosaic-skeleton approximations // *Calcolo.* 1996. V. 33(1-2). P. 47–57.

5. Brandt A., Lubrecht A. A. Multilevel Matrix Multiplication and Fast Solution of Integral Equations // *Journal of Computational Physics.* V. 90. P. 348–370.

Применение структурированных матриц для решения уравнений Хартри-Фока*

И. В. Ибрагимов

Аннотация

Предложен новый подход к решению уравнений Хартри-Фока с использованием теплицевых матриц и базисов на основе регулярных конечных элементов.

Обсуждаются проблемы: выбора базиса, зависимости объема данных и числа арифметических операций от числа базисных функций, а также арифметическая сложность и точность вычисления двух- и четырехцентровых интегралов. Объем промежуточных и конечных данных зависит от размера базиса линейно, а арифметическая сложность задачи — как $N \log_2 N$*, где* N *— размер базиса. Подход реализован в программном комплексе. Получены результаты, которые хорошо согласуются с теорией.*

1. ВВЕДЕНИЕ. Актуальной задачей квантовой химии в настоящее время является решение уравнений Хартри-Фока [1]. Решением этих уравнений являются несколько минимальных собственных значений и соответствующих собственных функций эрмитового оператора. Разность таких собственных значений характеризует энергии излучения или поглощения электронов молекулярными системами. Собственные функции отвечают за статистические свойства электронов, такие, как вероятность нахождения электронов в пространстве вокруг ядер атомов при соответствующей энергии молекулы.

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155), фонда Миннауки (грант №0201.01.013) и фонда Volkswagen-Stiftung (грант № VW I/71 493).

Оператор Хартри-Фока задан в трехмерном пространстве и состоит из суммы трех операторов: T — кинетической энергии электронов; V_1 — потенциальной энергии электронов в поле ядер атомов; V_2 — потенциальной энергии электронов в поле, которое создается всеми остальными электронами. По порядку оператор T больше двух остальных операторов V_1 и V_2. Наибольший интерес представляет решение этой задачи для кристаллических структур, в которых граничные условия периодичны по каждой декартовой координате. Распространенным методом решения этой задачи является дискретизация методом Рица [1].

Введем обозначения: m — число электронных пар; N — число ядер электронов в искомой системе; N — размер базиса тестовых функций. Искомыми собственными функциями оператора Хартри-Фока будут

$$\psi_i = \sum_{j=1}^{N} c_{ji}\phi_j, \qquad i = 1, \ldots, m, \tag{1}$$

где c_{ji} — неизвестные коэффициенты, ϕ_j — тестовые функции. Вид дискретного оператора Хартри-Фока:

$$\sum_{p=1}^{N} (F_{pq} - \epsilon_k S_{pq}) c_{pk} = 0, \qquad q = 1, \ldots, N; k = 1, \ldots, m, \tag{2}$$

где

$$F_{pq} = < p|H|q > + \sum_{i,j=1; i\neq j}^{N} P_{ij}(< ij|pq > - \frac{1}{2} < ip|jq >),$$

$$< p|H|q > = \int \phi_p^*(\mathbf{r}) \left\{ -\frac{1}{2}\left(\frac{d^2}{dx^2} + \frac{d^2}{dy^2} + \frac{d^2}{dz^2}\right) + \sum_{i=1}^{N} \frac{\mathbf{Q}_i}{|\mathbf{R}_i - \mathbf{r}|} \right\} \phi_q(\mathbf{r}) d\mathbf{r}, \tag{3}$$

$$< ij|kl > = \int \frac{\phi_i^*(\mathbf{r}_1)\phi_j(\mathbf{r}_1)\phi_k^*(\mathbf{r}_2)\phi_l(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2, \tag{4}$$

$$S_{ij} = < i|j > = \int \phi_i^*(\mathbf{r})\phi_j(\mathbf{r}) d\mathbf{r} \qquad P_{ij} = \sum_{k=1}^{m} c_{ik}^* c_{jk}.$$

Необходимо решить эту систему нелинейных уравнений и найти c_{jk} и ϵ_k. В общем случае матрицы F_{pq} и S_{pq} — эрмитовы.

В настоящее время предложено множество наборов тестовых функций [1–6]. Условно типы базисов можно разделить на две категории: базисы на основе атомных орбиталей и базисы на основе конечных элементов (БКЭ).

Базисы на основе атомных орбиталей широко используются, в то же время БКЭ начали рассматриваться применительно к этой задаче сравнительно недавно [5, 7]. В этой статье будут рассмотрены именно БКЭ на сетке с одинаковыми ячейками. Оказалось, что на таких базисах можно достичь достаточной для большинства расчетов точности.

2. Базисы на основе регулярных конечных элементов.

Будем решать уравнение методом дискретизации Рица. Пусть D — область из $\mathbb{R}^3$ в виде параллелепипеда, причем все ядра атомов находятся на расстоянии не менее чем d от дополнения к D.

В области D задана трехмерная сетка с шагом h так, что по каждой декартовой координате внутри D помещается соответственно n_i, n_j, n_k отрезков. Эта сетка порождает набор одинаковых кубиков со стороной h. Пусть каждый такой кубик имеет трехмерный индекс i, j, k.

Пусть также все ядра атомов находятся только в центрах кубиков, тогда координаты ядер можно выразить через h и соответствующий трехмерный индекс кубика, в котором находится ядро. Будем обозначать вектор координат как R_{ijk}.

Базисные функции $\{S_{ijk}\}_{i,j,k=1}^{n_i,n_j,n_k} \in \mathbb{R}^p$ заданы как p-гладкие функции с конечным носителем [8]: S_{ijk} отлична от нуля в кубиках с индексами p_i, p_j, p_k, при $\max(|p_i - i|, |p_j - j|, |p_k - k|) \leqslant p$, в остальных кубиках S_{ijk} тождественно равна нулю.

На границе будем рассматривать периодические по каждой декартовой координате условия, хотя все дальнейшие выкладки легко обобщаются на нулевые граничные условия. Также пусть все S_{ijk} могут быть получены из любой другой $S_{i'j'k'}$ с помощью параллельного переноса.

В большинстве случаев будут рассмотрены еще два дополнительных условия: ортогональность всех S_{ijk} и симметричность всех S_{ijk} относительно трех плоскостей, которые проходят через центр кубика с индексом i, j, k и каждая из этих плоскостей параллельна одной из граней области D. Дискретизация в таком базисе возможна из-за эрмитовости исходного оператора [15]. Связанные с этим детали не будут рассмотрены в этой работе.

Так как базис состоит из функций с конечным носителем, то часть интегралов из формулы (2) будет тождественно равна нулю. Так, интегралы

$$\int_D S^*_{ijk}(\bar{r}) S_{i'j'k'}(\bar{r}) d\bar{r}, \tag{5}$$

$$-\frac{1}{2}\int_D S^*_{ijk}(\bar{r}) \left[\frac{d^2}{dx^2} + \frac{d^2}{dy^2} + \frac{d^2}{dz^2}\right] S_{i'j'k'}(\bar{r}) d\bar{r}, \tag{6}$$

$$\int_D S^*_{ijk}(\bar{r}) \left[\sum_{q=1}^{N} \frac{\mathbf{Q}_q}{\left|\bar{R}_q - \bar{r}\right|}\right] S_{i'j'k'}(\bar{r}) d\bar{r} \tag{7}$$

тождественно равны нулю в случае нулевых граничных условий, если $\max(|i - i'|, |j - j'|, |k - k'|) > p$, и в случае периодических по каждой декартовой координате граничных условий, если

$$\begin{aligned} \max(&\mathrm{MOD}(i - i' + n_i, n_i),\\ &\mathrm{MOD}(j - j' + n_j, n_j),\\ &\mathrm{MOD}(k - k' + n_k, n_k)) > p. \end{aligned}$$

$\mathrm{MOD}(a, b)$ — операция взятия остатка от деления a на b.

Для простоты изложения далее будем рассматривать только периодические граничные условия, а также оба дополнительных условия: ортогональность и симметричность всех S_{ijk}. Тогда (5) равно единице при $i = i'$, $j = j'$, $k = k'$ и нулю во всех остальных случаях. Для интегралов вида (6) и (7) выполняются равенства

$$\begin{aligned} &\int_D S^*_{ijk}(\bar{r}) \left[\frac{d^2}{dx^2} + \frac{d^2}{dy^2} + \frac{d^2}{dz^2}\right]\\ &S_{\mathrm{MOD}(i+i',n_i),\mathrm{MOD}(j+j',n_j),\mathrm{MOD}(k+k',n_k)}(\bar{r}) d\bar{r} =\\ &\int_D S^*_{i''j''k''}(\bar{r}) \left[\frac{d^2}{dx^2} + \frac{d^2}{dy^2} + \frac{d^2}{dz^2}\right]\\ &S_{\mathrm{MOD}(i''+i',n_i),\mathrm{MOD}(j''+j',n_j),\mathrm{MOD}(k''+k',n_k)}(\bar{r}) d\bar{r}; \end{aligned} \tag{8}$$

$$\int_D S^*_{ijk}(\bar r)\left[\frac{1}{\left|\bar R_{ijk}-\bar r\right|}\right] S_{MOD(i+i',n_i),MOD(j+j',n_j),MOD(k+k',n_k)}(\bar r)d\bar r = \int_D S^*_{i''j''k''}(\bar r)\left[\frac{1}{\left|\bar R_{i''j''k''}-\bar r\right|}\right] S_{MOD(i''+i',n_i),MOD(j''+j',n_j),MOD(k''+k',n_k)}(\bar r)d\bar r \qquad (9)$$

для любых i, i', i'', j, j', j'', k, k', k''. Заметим, что различных значений интегралов вида (8) будет только порядка p^3, а интегралов вида (9) — порядка $n_i n_j n_k$.

Более сложным будет вычисление интегралов вида (4), которых можно записать в виде

$$\int_D\int_D \frac{S^*_{i_1j_1k_1}(\bar r)S_{i_2j_2k_2}(\bar r)S^*_{i_3j_3k_3}(\bar r')S_{i_4j_4k_4}(\bar r')}{\bar r-\bar r'}d\bar r d\bar r'. \qquad (10)$$

Из-за того что базис построен на функциях с конечным носителем, интеграл (10) тождественно равен нулю либо при

$$\begin{aligned}\max(&MOD(i_1-i_2+n_i,n_i),\\ &MOD(j_1-j_2+n_j,n_j),\\ &MOD(k_1-k_2+n_k,n_k)) > p,\end{aligned} \qquad (11)$$

либо при

$$\begin{aligned}\max(&MOD(i_3-i_4+n_i,n_i),\\ &MOD(j_3-j_4+n_j,n_j),\\ &MOD(k_3-k_4+n_k,n_k)) > p.\end{aligned} \qquad (12)$$

А так как базисные функции симметричны, то выполняется

$$\int_D\int_D \frac{S^*_{i_1j_1k_1}(\bar r)S_{i_2j_2k_2}(\bar r)S^*_{i_3j_3k_3}(\bar r')S_{i_4j_4k_4}(\bar r')}{\bar r-\bar r'}d\bar r d\bar r' = \int_D\int_D \frac{S^*_{i_3j_3k_3}(\bar r)S_{i_4j_4k_4}(\bar r)S^*_{i_1j_1k_1}(\bar r')S_{i_2j_2k_2}(\bar r')}{\bar r-\bar r'}d\bar r d\bar r'.$$

Наконец, если любая базисная функция получается из другой параллельным переносом, то выполняется равенство

$$\int_D\int_D \frac{S^*_{i_1j_1k_1}(\bar{r})S_{i_2j_2k_2}(\bar{r})S^*_{i_3j_3k_3}(\bar{r}')S_{i_4j_4k_4}(\bar{r}')}{\bar{r}-\bar{r}'}d\bar{r}d\bar{r}' =$$
$$\int_D\int_D \frac{S^*_{i_5j_5k_5}(\bar{r})S_{i_6j_6k_6}(\bar{r})S^*_{i_7j_7k_7}(\bar{r}')S_{i_8j_8k_8}(\bar{r}')}{\bar{r}-\bar{r}'}d\bar{r}d\bar{r}'$$

при условиях

$$\begin{aligned}
MOD(i_1+i_3,n_i) &= MOD(i_5+i_7,n_i),\\
MOD(i_2+i_4,n_i) &= MOD(i_6+i_8,n_i),\\
MOD(j_1+j_3,n_j) &= MOD(j_5+j_7,n_j),\\
MOD(j_2+j_4,n_j) &= MOD(j_6+j_8,n_j),\\
MOD(k_1+k_3,n_k) &= MOD(k_5+k_7,n_k),\\
MOD(k_2+k_4,n_k) &= MOD(k_6+k_8,n_k).
\end{aligned}$$

Обозначим

$$t(i_2,j_2,k_2,i_3,j_3,k_3,i_4,j_4,k_4) =$$
$$\int_D\int_D \frac{S^*_{0,0,0}(\bar{r})S_{i_2j_2k_2}(\bar{r})S^*_{i_3j_3k_3}(\bar{r}')S_{i_4j_4k_4}(\bar{r}')}{\bar{r}-\bar{r}'}d\bar{r}d\bar{r}'.$$

Учитывая (11) и (12), легко заметить, что различных значений t будет порядка $p^6n_in_jn_k$. Коэффициент p^6 сильно завышен — из-за симметрии большинство таких интегралов равны между собой. В большинстве случаев мы будем ограничиваться сверху значением $p = 3$, для этого случая константа перед $n_in_jn_k$ будет равна 28.

Сделаем замену переменных во всех интегралах (6), (7) и (10) такую, чтобы в новых координатах шаг сетки тождественно равнялся бы единице. Тогда в новых координатах перед (7) и (10) появится коэффициент $\frac{1}{h}$, а перед (6) — коэффициент $\frac{1}{h^2}$. Также легко показать, что если все первые и вторые частные производные ограничены у S_{ijk}, то все интегралы в новой системе координат ограничены константой, которая не зависит от размеров задачи.

Подытожим полученные результаты. Для решения дискретизованных методом Рица уравнений Хартри-Фока необходимо вычислять три вида интегралов: (6), (7) и (10). Количество различных интегралов вида (6) не зависит от размера задачи. Более того, для кусочно-полиномиальных базисных функций такие интегралы могут быть без особого труда вычислены аналитически. Количество различных интегралов вида (7) и (10) линейно зависит от размера задачи.

2.1. ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ. Итак, необходимо вычислить интегралы вида

$$\int_\Omega \frac{s(\bar{r})}{\|\bar{r}-\bar{R}\|_2} d\bar{r} \tag{13}$$

и

$$\int_\Omega \int_\Omega \frac{s_1(\bar{r}_1)s_1(\bar{r}_2)}{\|\bar{r}_1-\bar{r}_2+\bar{t}\|_2} d\bar{r}_1 d\bar{r}_2, \tag{14}$$

где $\bar{r}$, $\bar{r}_1$, $\bar{r}_2$, $\bar{t}$, $\bar{R}$ — трехмерные векторы; Ω, — область в виде параллелепипеда, s, s_1, s_2 — функции, которые являются тензорным произведением функций одного аргумента вида

$$s(\bar{x}) = f_1(x_1)f_2(x_2)f_3(x_3).$$

Так как каждая $s(\bar{x})$ является произведением двух базисных функций, область задания которых — куб со стороной h, то область интегрирования Ω не может быть больше, чем куб со стороной h. Координаты векторов $\bar{R}$ и $\bar{t}$ принимают целочисленные значения от 0 до $n-1$, где n — размер задачи.

Заметим, что при условиях $R_x \leqslant h$, $R_y \leqslant h$, $R_z \leqslant h$ подынтегральная функция в (13) может принимать бесконечно большие по абсолютной величине значения. В то же время, при $\|\bar{R}\|_2 >> h$ подынтегральная функция ограничена, изменяется мало и значение интеграла (13) стремится к $\dfrac{1}{\|\bar{R}\|_2}\displaystyle\int_\Omega s(\bar{x})d\bar{x}$ при $h \to \infty$. Аналогичные рассуждения можно провести для интеграла (14), используя $\bar{t}$ вместо $\bar{R}$. Отметим еще несколько свойств функций s:

1) s — кусочно-полиномиальная функция;

2) так как s — это произведение двух базисных функций, а, по построению, базисные функции симметричны относительно $x_i = 0$, то s — симметрична относительно $x_1 = 0$, $x_2 = 0$ и $x_3 = 0$;

3) s на границе области интегрирования Ω обращается в нуль, за исключением случая, когда s является кусочно-постоянной.

Необходимо вычислять интегралы (13) и (14) с абсолютной точностью не менее $\frac{\eta}{N}$, где η — абсолютная точность вычисления собственных значений.

Далее будут рассмотрены несколько приемов, которые позволят вычислять такие интегралы с заданной точностью за разумное число арифметических операций.

Теорема 1. Для любых p и q, являющихся тензорным произведением функций от одной переменной и области Ω в виде параллелепипеда, существуют s и Ω' такие, что

$$\int_{\Omega}\int_{\Omega}\frac{p(\bar{x})q(\bar{y})}{\|\bar{x}-\bar{y}+\bar{t}\|_2}d\bar{x}d\bar{y}=\int_{\Omega'}\frac{z(\bar{r})}{\|\bar{r}-\bar{t}\|_2}d\bar{r}. \tag{15}$$

Доказательство. Интеграл

$$\int_0^h f_y(y)dy\int_0^h \omega(x-y)f_x(x)dx$$

при замене переменных $u=x-y$ и $v=x+y$ превращается в

$$\int_{-h}^0\omega(u)\int_{h+u}^{h-u}f_x(\frac{v+u}{2})f_y(\frac{v-u}{2})dvdu+ \\ \int_0^h\omega(u)\int_u^{2h-u}f_x(\frac{v+u}{2})f_y(\frac{v-u}{2})dvdu.$$

Обозначим

$$g(u)=\begin{cases} u\leqslant 0\ :\ \int_{h+u}^{h-u}f_x(\frac{v+u}{2})f_y(\frac{v-u}{2})dv, \\ u>0\ :\ \int_u^{2h-u}f_x(\frac{v+u}{2})f_y(\frac{v-u}{2})dv.\end{cases}$$

Тогда

$$\int_0^h f_y(y)dy\int_0^h\omega(x-y)f_x(x)dx=\int_{-h}^h\omega(u)g(u)du. \tag{16}$$

Пусть область ω задана в виде параллелепипеда со сторонами $h_1\times h_2\times h_3$. По условию $p(\bar{x})=p_1(x_1)p_2(x_2)p_3(x_3)$ и $q(\bar{y})=$

$q_1(y_1)q_2(y_2)q_3(y_3)$. Используя (16), получим (15), где Ω' — параллелепипед со сторонами $2h_1 \times 2h_2 \times 2h_3$. Что и требовалось доказать. ■

В большинстве случаев функцию z из (15) можно построить аналитически: если p и q — базисные сплайны степени k, то z — также базисный сплайн степени $2k+1$.

Итак, круг задач заметно упростился, и необходимо вычислять только интегралы вида (13), так как базисные функции можно выбрать так, чтобы неопределенные интегралы вида

$$\iiint s_1(\bar{x}+\bar{y})s_2(\bar{x}+\bar{y})d\bar{x}$$

вычислялись аналитически.

Лемма 1. Неопределенный интеграл $\iiint \frac{x_1^{n_1}x_2^{n_2}x_3^{n_3}}{R}dx_1dx_2dx_3$, где $R = \sqrt{x_1^2+x_2^2+x_3^2}$, а n_1, n_2, n_3 — целые положительные числа, представим в виде

$$p_1R + p_2\ln(x_1+R) + p_3\ln(x_2+R) + p_4\ln(x_3+R) + \\ p_5\mathrm{arctg}\left(\frac{x_1x_2}{x_3R}\right) + p_6\mathrm{arctg}\left(\frac{x_2x_3}{x_1R}\right) + p_7\mathrm{arctg}\left(\frac{x_1x_3}{x_2R}\right),$$

где $p_1,\dots,p_7$ — полиномы от x_1, x_2 и x_3 степени не выше

$$3\max(n_1,n_2,n_3)+2.$$

Доказательство леммы основано на непосредственном интегрировании и ввиду громоздкости не приводится.

Лемма говорит в пользу аналитического интегрирования, если базисные функции являются кусочно-полиномиальными. Но при $R \to \infty$ определенный интеграл стремится к $\frac{1}{R}$, в то время как неопределенный интеграл имеет порядок R^{3k+2}, k — максимальная степень полиномов в подынтегральной функции. Тогда абсолютная погрешность интегрирования будет не меньше $\eta R^{3(k+1)}$. При $k \geqslant 2$ такой подход становится практически неприемлемым для расчетов из-за большой погрешности.

При $k = 0,1$ аналитическое вычисление интегралов устойчиво при малых R и при $R < 10$ дает погрешность не более 10^{-9}. Для вычисления одного такого интеграла требуется около 200 арифметических операций и 56 вычислений иррациональных функций вида $\ln$, arctg, $\sqrt{\cdot}$.

К сожалению, уже при $k = 2$ аналитические формулы содержат полиномы до 8-й степени и число таких полиномов очень велико — порядка 150. Абсолютная точность интегрирования, полученная экспериментально (в качестве „точного“ значения было проведено численное интегрирование по формуле Гаусса), для $R = 0$ достигала только 10^{-3}, следовательно, при $R > 0$ точность может быть только хуже, что совершенно неприемлемо для наших целей.

В большинстве случаев функция $s(\bar{r})$ в (13) не принадлежит классу кусочно-линейных функций, поэтому необходимо было найти другой способ вычисления таких интегралов.

Использование формул Гаусса [9] без выбора шага требует огромных арифметических затрат в окрестности точки $||\bar{R}||_2 = 0$. Формулы Гаусса с выбором шага для трехмерных задач ненамного уменьшают число арифметических операций, поэтому предлагается приблизить $s(\bar{r})$ с помощью кусочно-линейных функций. На полученном разбиении каждый параллелепипед содержит аналитически вычислимый интеграл. Точность такого интегрирования — это произведение точности приближения $s(\bar{r})$ с помощью кусочно-линейных функций $l(\bar{r})$ и максимального значения интеграла $\int_\Omega \frac{d\bar{x}}{||\bar{R} - \bar{x}||_2}$ по всем возможным $\bar{R}$. Очевидно, что максимум достигается в случае, когда R — центр параллелепипеда Ω. Итак, получим окончательную оценку:

$$|s(\bar{r}) - f(\bar{r})| \int_\Omega \frac{d\bar{x}}{||\bar{R} - \bar{x}||_2}.$$

Величина $\int_\Omega \frac{d\bar{x}}{||\bar{R} - \bar{x}||_2}$ вычислима аналитически и ограничена сверху числом 3 для Ω в виде куба со сторонами $1 \times 1 \times 1$.

Можно использовать неравномерное разбиение для того, чтобы уменьшить $\max|s(\bar{r}) - f(\bar{r})|$. Эта задача очень простая, так как f_i и s_i — кусочно-полиномиальные функции. Более того, если $s_i(-h) = 0$, $s_i(h) = 0$, то

$$\int_{-h}^{h}\int_{-h}^{h}\int_{-h}^{h} \frac{s(\bar{x})}{||\bar{R} - \bar{x}||_2} dx_1 dx_2 dx_3 = \sum_{i,j,k=1}^{n} F_{ijk} p_{ijk}, \tag{17}$$

причем F_{ijk} зависят только от R и не зависят от s, а p_{ijk} зависят только от вида s и не зависят от R. Это достигается за счет того, что F_{ijk} — это величины интегралов, которые вычисляются аналитически, а p_{ijk} — их веса, которые зависят только от s.

Если необходимо посчитать несколько интегралов в одной области, то использование формулы (17) значительно упрощает это вычисление, так как наиболее трудоемкая часть по вычислению F_{ijk} выполняется один раз для нескольких интегралов.

Итак, мы показали, как вычислять интегралы, когда значение $||\bar{R}||_2$ находится в окрестности нуля, теперь необходимо понять, как вычислять интегралы, для которых $||\bar{R}||_2 \gg 0$.

Как уже было отмечено выше, для таких интегралов подынтегральная функция изменяется мало, поэтому было предложено разложить функцию $\dfrac{1}{||\bar{R}-\bar{x}||_2}$ в ряд Тейлора по всем декартовым координатам $\bar{x}$ в точке $\bar{R}$. В этом случае получается ряд по отрицательным степеням функции $||\bar{R}||_2$ с коэффициентами $p_i, i = 1, \ldots,$ в виде полиномов от координат $\bar{R}$ и $\bar{x}$. Первый коэффициент равен единице. Оценке на остаточный член для разложения до k-й степени соответствует выражение

$$\mathtt{const} \int_{-h}^{h} \int_{-h}^{h} \int_{-h}^{h} \frac{dx_1 dx_2 dx_3}{||\bar{R}||_2^{k+1}(k+1)!},$$

где $\mathtt{const}$ зависит только от вида базисной функции.

Был обнаружен также интересный факт: при таком разложении коэффициенты при четных степенях по $||\bar{R}||_2$ содержат только нечетные степени по координатам $\bar{x}$.

Если интегралы вида (13) при $||\bar{R}||_2 >> 0$ вычислять, используя такое разложение в ряд Тейлора, тогда (13) переписывается в виде:

$$\begin{aligned}\int_\Omega \frac{s(\bar{r})}{||\bar{R}||_2}\bar{r} + \int_\Omega \frac{s(\bar{r})p_2(\bar{r},\bar{R})}{||\bar{R}||_2^2}\bar{r} + \int_\Omega \frac{s(\bar{r})p_3(\bar{r},\bar{R})}{||\bar{R}||_2^3}\bar{r} + \cdots = \\ \frac{1}{||\bar{R}||_2}\int_\Omega s(\bar{r})d\bar{r} + \frac{1}{||\bar{R}||_2^2}\int_\Omega s(\bar{r})p_2(\bar{r},\bar{R})d\bar{r} + \\ \frac{1}{||\bar{R}||_2^3}\int_\Omega s(\bar{r})p_3(\bar{r},\bar{R})d\bar{r} + \ldots \qquad (18)\end{aligned}$$

Как видно, коэффициенты разложения по степеням $||\bar{R}||_2$ являются интегралами произведений базисных функций и некоторых полиномиальных функций и в большинстве случаев могут быть вычислены аналитически. Следовательно, все эти коэффициенты можно предварительно посчитать.

Более того, так как функции $s(\bar{r})$ симметричны по условию, то все интегралы при четных степенях по $||\bar{R}||_2$ обращаются в нуль. Оценка точности такого интегрирования:

$$\frac{1}{||\bar{R}||_2^k} \int_{\Omega} s(\bar{r}) p_k(\bar{r}) d\bar{r}.$$

В результате возникает интересная задача: есть два альтернативных алгоритма для вычисления интегралов, которые определяются параметром R и видом базисных функций. Каждый из этих алгоритмов зависит еще от одного параметра: для первого алгоритма это k_1 — число узлов в разбиении, а для второго k_2 — максимальная степень разложения в ряд Тейлора. Для каждого из этих алгоритмов существуют априорные функции $\epsilon_1(k_1, R)$ и $\epsilon_2(k_2, R)$, по которым можно определить точность вычисления такого интеграла для соответствующего алгоритма и априорные функции $f_1(k_1, R)$ и $f_2(k_2, R)$, по которым можно определить арифметические затраты, необходимые для вычисления этого интеграла с точностью $\epsilon_1(k_1, R)$ и $\epsilon_2(k_2, R)$ соответственно.

Для каждого интеграла важно правильно выбрать алгоритм и соответствующий номер k_1 или k_2 для вычисления интегралов с точностью не менее η.

Так как все эти функции монотонные по $||\bar{R}||_2$ на всей области задания, то решением этой задачи будет условие: *существует такое α, что при $||\bar{R}||_2 < \alpha$ используется первый алгоритм с постоянным значением k_1. В противном случае используется второй алгоритм с некоторыми значениями k_2.*

Как показала практика, при $||\bar{R}||_2 > \alpha$ для большинства интегралов достаточно точности, которая получается при $k_2 \leqslant 3$, то есть в разложении (18) используются только первые один или два ненулевых члена. При этом арифметические затраты на вычисление одного интеграла соответствуют константе порядка 20 арифметических операций.

2.2. СВОЙСТВА ДИСКРЕТИЗОВАННОГО ФУНКЦИОНАЛА ХАРТРИ-ФОКА. В настоящее время существует огромное количество алгоритмов для минимизации значений функций от нескольких переменных [10–11]. Поэтому вначале рассмотрим, как вычислять значение функции в точке и ее первые и вторые частные производные по всем аргументам, а только затем алгоритм минимизации. Также будем рассматривать приближенное вычисление вышеперечисленных операций при условии, что арифметические затраты и/или объем данных для временного и постоянного хранения будут значительно меньше, чем при точном вычислении. Это может потребоваться при построении предобусловливателя.

Пусть $N = n_i n_j n_k$ — размер задачи, m — как и прежде, число электронных пар. Каждая собственная функция для соответствующей электронной пары определена вектором $c_i \in \mathbb{C}^N$, $i = 1, \ldots, m$.

Число интегралов, необходимое для решения уравнений Хартри-Фока, — порядка N. Число неизвестных в задаче — Nm комплексных чисел.

2.2.1. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ДИСКРЕТИЗОВАННОГО ФУНКЦИОНАЛА ЭНЕРГИИ ХАРТРИ-ФОКА. Запишем выражение для энергии Хартри-Фока в матричном виде $E = T+V+V$ и рассмотрим отдельно каждый из этих членов.

Кинетическая энергия всех электронов:

$$T = \frac{1}{h^2} \sum_{s=1}^{m} c_s^* A c_s.$$

Здесь A определена только независимым от N числом параметров порядка p^3, более того, такие параметры не зависят от размера матрицы, а зависят только от вида базисных функций. A является трехуровневой положительно определенной ленточной матрицей [12], на диагоналях каждого такого уровня находятся одинаковые элементы и образуют циркулянты. Число таких диагоналей равно $(2p+1)^3$. Кроме этого, матрица симметрична.

Известно [12], что произвольный циркулянт (в том числе и многоуровневый) можно представить в виде произведения F^*DF, где F — матрица Фурье, D — диагональная матрица, а умножение вектора размера N на матрицу Фурье требует только порядка $N \log_2 N$ арифметических операций. Следовательно, вычисление кинетической энергии всех электронов потребует порядка $mN \min(\log_2 N, (2p+1)^3)$ арифметических операций.

При малых p выгоднее использовать явное умножение вектора на матрицу, а при больших — быстрое преобразование Фурье

(БПФ).

Потенциальная энергия всех электронов в поле всех атомных ядер:

$$V = -\frac{2}{h}\sum_{s=1}^{m} c_s^* B c_s.$$

Здесь B является трехуровневой положительно определенной ленточной матрицей с числом ненулевых диагоналей $(2p+1)^3$. Также матрицу B можно представить в виде суммы

$$B = \sum_{s=1}^{N} Q_s P(i_s, j_s, k_s) W P(n_i - i_s, n_j - j_s, n_k - k_s), \tag{19}$$

где W — трехуровневая положительно определенная ленточная матрица с числом ненулевых диагоналей $(2p+1)^3$, состоящая из интегралов (13), а матрицы $P(i,j,k)$ — это матрицы перестановок, которые сдвигают элемент с номером $(0,0,0)$ в позицию (i,j,k), индексы i_s, j_s, k_s соответствуют номеру кубика, в котором находится s-е ядро атома.

Вопрос об использовании вместо матрицы B представления (19) зависит от размера задачи, при больших N значительная часть внедиагональных элементов матрицы W стремится к нулю, и ими можно пренебречь. В этом случае суммарный объем памяти для хранения такого компактного представления и количества арифметических операций для умножения вектора на такую матрицу может быть значительно меньше, чем при использовании матрицы B. Правильный выбор можно сделать, только зная N и p. При больших N асимптотически число операций для вычисления V порядка Nm.

Потенциальная энергия всех электронов в поле, образованном самими электронами:

$$V = \frac{1}{h}\sum_{s,q=1}^{m} \left\{2c_s^* H^*(c_s) T H(c_q) c_q - c_s^* H^*(c_q) T H(c_s) c_q\right\}. \tag{20}$$

Здесь T — трехуровневая блочная симметричная циркулянтная матрица с неотрицательными элементами, составленными из значений интегралов вида (14); $H(c)$ — трехуровневая блочная ленточная матрица, элементы которой составлены из вектора c, причем (i, j, k)-й блок представляет собой вектор размера p^3, составленный из чисел $c_{i+i',j+j',k+k'}$, где $i', j', k' \in [0, p]$. При использовании БПФ арифметические затраты на вычисление V составляют порядка $\mathrm{const}\, m^2 N \log_2 N$, где $\mathrm{const} \simeq p^6$. Позже будут рассмотрены:

1) точное преобразование формулы (20), при котором для любого p будет выполняться $\mathrm{const} = 32$;

2) приближенное преобразование, при котором $\mathrm{const} \leqslant 32$.

2.2.2. ВЫЧИСЛЕНИЕ ПЕРВЫХ ПРОИЗВОДНЫХ ДИСКРЕТИЗОВАННОГО ФУНКЦИОНАЛА ЭНЕРГИИ ХАРТРИ-ФОКА. Вычисление первых производных дискретизованного функционала энергии Хартри-Фока алгоритмически ненамного отличается от вычисления самого функционала, поэтому здесь приведена только окончательная формула (в обозначениях предыдущей главы):

$$dE_i = \frac{2}{h^2} A c_i - \frac{4}{h} B c_i$$

$$\frac{4}{h} \sum_{s=1}^{m} \{+2H^*(c_i) T H(c_s) c_s - H^*(c_s) T H(c_s) c_i\} - \epsilon_i c_i, \tag{21}$$

где dE_i, $i = 1, \ldots, m$ — вектор первых производных функционала по координатам c_i-го вектора.

Число арифметических операций, которое требуется для вычисления всех таких векторов не больше аналогичного для самого значения дискретизованного функционала энергии Хартри-Фока.

2.2.3. ВЫЧИСЛЕНИЕ ВТОРЫХ ПРОИЗВОДНЫХ ДИСКРЕТИЗОВАННОГО ФУНКЦИОНАЛА ЭНЕРГИИ ХАРТРИ-ФОКА. Ввиду очень громоздкого выражения приведем вид матрицы вторых производных дискретизованного функционала энергии Хартри-Фока только для случая $p = 0$:

$$d^2 E_{ii} = \frac{2}{h^2} A - \frac{4}{h} B$$

$$+ \frac{4}{h} \{3\, \mathrm{diag}(c_i^*) T\, \mathrm{diag}(c_i) - \mathrm{diag}(T\, \mathrm{diag}(c_i^*) c_i)\} +$$

$$+ \frac{4}{h} \sum_{s=1}^{m} \{2\, \mathrm{diag}(T\, \mathrm{diag}(c_s^*) c_s) - \mathrm{diag}(c_s^*) T\, \mathrm{diag}(c_s)\} - \epsilon_i I,$$

$$d^2E_{ij} = \frac{4}{h}\{4\operatorname{diag}(c_i^*)T\operatorname{diag}(c_j)$$
$$-\operatorname{diag}(c_j^*)T\operatorname{diag}(c_i) - \operatorname{diag}(T\operatorname{diag}(c_i^*)c_j)\Big\}\,.$$

Здесь d^2E_{ii} и d^2E_{ij} — это соответственно диагональные и внедиагональные элементы блочной матрицы размеров $m \times m$.

Как видно, умножить вектор на такую матрицу можно за $m^2N\log_2 N$ операций, в то же время решение системы линейных уравнений с такой матрицей — это достаточно сложная задача, так как для этого придется использовать дополнительный итерационный процесс.

2.2.4. О ТОЧНЫХ ПРЕОБРАЗОВАНИЯХ ДИСКРЕТИЗОВАННОГО ФУНКЦИОНАЛА ЭНЕРГИИ ХАРТРИ-ФОКА. Пусть необходимо найти решение уравнений Хартри-Фока для m электронных пар на ортонормированном пространстве из N векторов $\{q_i\}$, причем $N - m < m$. Такая задача может возникнуть в минимизационных методах с релаксацией [11]. Тогда первые m векторов $\{q_i\}$ соответствуют решению уравнений Хартри-Фока на очередной итерации, а остальные $N - m$ векторов — направлениям, по которым предполагается улучшение решения. В этой задаче Nm неизвестных и $\dfrac{m(m+1)}{2}$ ограничений, связанных с ортонормированностью решения. Поэтому, по возможности, хочется упростить задачу.

Оказалось, что такую задачу можно привести к задаче с $(N - m)m$ неизвестными и соответственно $\dfrac{(N-m)(N-m+1)}{2}$ ограничениями. При $N - m << m$ этот подход обеспечит существенный выигрыш.

Наиболее интересными являются задачи, в которых значения m лежат в пределах от 10 до 50. Тогда существенный выигрыш будет наблюдаться при значениях $N - m$ в пределах от 1 до 10.

Заметим, что в случае квадратичного функционала это очевидно и решением будут $N - m$ старших собственных значений и соответствующие им собственные векторы. Интересно, что в случае функционала Хартри-Фока наблюдается похожая ситуация.

Будем использовать обозначения формул (1)–(4). Пусть необходимо найти векторы c_i, $i = 1, \ldots, m$, соответствующие минимуму E. Построим еще $N - m$ произвольных векторов $\{c_i\}$ так, чтобы полученная система из N векторов оставалась ортонормированной. Обозначим $C = (c_1, \dots, c_N)$.

Перейдем от задачи нахождения минимума E по первым m векторам c_i к задаче минимума того же E, но по $N - m$ старшим векторам c_i. Очевидно, что $CC^* = I$. Тогда матрицу $P = \{p_{ij}\}$ можно представить в виде $P = I - \sum\limits_{i=m+1}^{N} c_i c_i^*$. Подставим это в формулу (2):

$$E = 2 \sum_{p,q=1}^{N} < p|H|q > \left(\delta_{pq} - \sum_{k=m+1}^{N} c_{pk}^* c_{qk} \right) +$$

$$\sum_{p,q=1}^{N} \sum_{r,s=1}^{N} G_{pqrs} \left(\delta_{pq} - \sum_{k=m+1}^{N} c_{pk}^* c_{qk} \right) \left(\delta_{rs} - \sum_{l=m+1}^{N} c_{rl}^* c_{sl} \right) =$$

$$\sum_{p=1}^{N} (2 < p|H|q > + G_{pppp}) + \tag{22}$$

$$\sum_{p,q=1}^{N} \sum_{k=m+1}^{N} \left\{ -2 < p|H|q > + \sum_{r=1}^{N} (G_{pqrr} + G_{rrpq}) \right\} +$$

$$\sum_{p,q=1}^{N} \sum_{r,s=1}^{N} G_{pqrs} \left(\sum_{k=m+1}^{N} c_{pk}^* c_{qk} \right) \left(\sum_{l=m+1}^{N} c_{rl}^* c_{sl} \right). \tag{23}$$

Как видно, E состоит из двух слагаемых: (22) и (23), где слагаемое (22) является константой и на значение E не влияет, а слагаемое (23) — то же, что в исходной задаче, но содержит только $(N - m)m$ неизвестных.

Рассмотрим еще одну вспомогательную задачу: в области $D \in \mathbb{R}^3$ задана трехмерная сетка с шагом h размера $n_i \times n_j \times n_k$, на которой заданы кусочно-полиномиальные p-гладкие нормированные функции $\{S_{ijk}\}_{i,j,k=1}^{n_i,n_j,n_k}$ с конечным носителем с периодическими по каждой декартовой координате условиями. Каждая такая функция является тензорным произведением полиномов степени не выше $p + 1$.

Пусть $\phi = \sum\limits_{i=1}^{n_i} \sum\limits_{j=1}^{n_j} \sum\limits_{k=1}^{n_k} a_{ijk} S_{ijk}, \quad \psi = \sum\limits_{i=1}^{n_i} \sum\limits_{j=1}^{n_j} \sum\limits_{k=1}^{n_k} b_{ijk} S_{ijk}.$

Очевидно, что $\psi\phi \in \mathbb{R}^p$. Построим $\xi = \sum\limits_{i=1}^{n_i} \sum\limits_{j=1}^{n_j} \sum\limits_{k=1}^{n_k} \alpha_{ijk} S_{ijk}$ такую, чтобы значения $\psi\phi$ и ξ совпадали в узлах сетки. Так как $\xi \in \mathbb{R}^p$, то в узлах сетки равны нулю значения и все p частных производных функции $\psi\phi - \xi$. Так как степень полиномов в $\psi\phi - \xi$ на каждом из кубиков не выше $2p + 2$, то, накладывая $2p + 2$ ограничений, получаем, что $\psi\phi - \xi$ может быть разложена в виде

$$\psi\phi - \xi = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} \beta_{ijk} \mathcal{S}_{ijk},$$

где $\mathcal{S}_{ijk}$ задана только в кубике (i, j, k), тождественно равна нулю во всех остальных кубиках, является тензорным произведением трех одномерных функций $u_i(x)$, $u_j(y)$ и $u_k(z)$, каждая из которых полином степени $2p+2$, у которого значение и все $p-1$ производных равны нулю на границе кубика.

Такое преобразование позволяет любые ψ и ϕ представить в виде суммы с весами $n_i n_j n_k$ пар регулярных базисных функций, причем это точное преобразование. Используя такое преобразование, можно для любого p преобразовать формулу (20) так, чтобы размер блока матрицы T был только 8×8. В этом случае арифметические затраты на вычисление V составят только $32m^2N(\log_2 N + 16)$ операций.

2.2.5. ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ДИСКРЕТИЗОВАННОГО ФУНКЦИОНАЛА ЭНЕРГИИ ХАРТРИ-ФОКА. Обобщая предыдущий результат, можно:

- пренебречь величиной $\psi\phi - \xi$. Тогда арифметические затраты на вычисление V составят всего $\frac{1}{2}m^2 N \log_2 N$ операций;

- аппроксимировать $\psi\phi$ на более мелкой сетке с шагом $\frac{h}{K}$, $K = 2, 3, \ldots$. Тогда арифметические затраты на вычисление V составят порядка $\frac{1}{2}m^2 NK \log_2(NK)$ операций. Формула для оценки точности такой аппроксимации легко выводится для любых конкретных видов базисных функций, но в общем случае нам не удалось получить простого и понятного выражения.

Еще одно упрощение связано снова с членом V. Так как арифметические затраты на вычисление V и T составляют только $\mathtt{const} N \log_2 N$ операций против $\mathtt{const} m^2 N \log_2 N$ для V, то такое упрощение очень актуально. Идея заключается в следующем: приблизим T из (20) в виде $\alpha I + \beta u u^T$, где $u = (1, \ldots, 1)^T \in \mathbb{C}^N$. Тогда выражение (20) из-за ортонормированности решения перепишется в виде:

$$V = \frac{1}{h}(\alpha z^* z + \beta(2m-1)m), \tag{24}$$

где $z = \sum_{s=1}^{m} H(c_s)c_s$. Аналогично выражение (21) перепишется в виде

$$dE_i = \frac{2}{h^2}Ac_i - \frac{4}{h}Bc_i + \frac{4}{h}\alpha\,\mathrm{diag}(z)c_i + \frac{4}{h}\beta(2m-1) - \epsilon_i c_i.$$

Тогда арифметические затраты на вычисление как дискретизованного функционала Хартри-Фока, так и всех его первых частных производных сократятся до $\mathtt{const}N\log_2 N$ операций. Так как T — трехуровневый блочный симметричный циркулянт, то такое приближение можно выполнить в виде $min\|T - \alpha I - \beta uu^T\|_*$ во фробениусовой и спектральных нормах не более чем за $N\log_2 N$ арифметических операций.

На основе экспериментальных вычислений отношение

$$\frac{\|T - \alpha I - \beta uu^T\|_*}{\|T\|_*}$$

в зависимости от нормы, N и от вида базисных функций изменяется в пределах от 0.001 до 0.01. Это позволяет использовать такое приближение для грубых расчетов, вычисления начального приближения или для эффективного предобусловливания.

Изучим зависимости $\frac{\|T\|_F^2}{\|V\|_F^2}$ $\frac{\|T\|_F^2}{\|V\|_F^2}$.

Так как A — трехуровневая ленточная циркулянтная матрица, то $\|A\|_F^2 = \mathtt{const}N$, где $\mathtt{const}$ — сумма квадратов элементов матрицы A в любой строке, по построению эта величина порядка единицы. Тогда $\|T\|_F^2 \simeq \frac{n^3}{h^4}$. Из (19)

$$\|B\|_F^2 \leqslant \sum_{s=1}^{m} Q_s^2 \|W\|_F^2,$$

а по построению $||W||_F^2 \simeq n^2$. Учитывая, что n обратно пропорционально h, получим

$$\frac{||T||_F^2}{||V||_F^2} \geqslant Cn^3 = C_1 N, \tag{25}$$

где N — размер матриц T и V. Из этого следует, что при увеличении числа неизвестных в дискретной задаче оператор кинетической энергии становится большим по порядку по сравнению с V. Аналогично получается оценка для $\frac{||T||_F^2}{||V||_F^2}$. Так как $||V||_F^2 \leqslant ||T||_F^2$, следовательно, оператор кинетической энергии становится большим по порядку по сравнению и с нелинейной составляющей уравнений Хартри–Фока при росте размера матриц, то есть

$$\frac{||T||_F^2}{||V||_F^2} \geqslant Cn^3 = C_2 N. \tag{26}$$

Аналогичным образом можно получить оценки для таких отношений в спектральной норме. Тогда

$$\frac{||T||_2}{||V||_2} \geqslant C_3 n \qquad \frac{||T||_2}{||V||_2} \geqslant C_4 n. \tag{27}$$

Легко заметить, что эти оценки неулучшаемы.

2.3. РЕШЕНИЕ ИТЕРАЦИОННЫМИ МЕТОДАМИ ДИСКРЕТНЫХ УРАВНЕНИЙ ХАРТРИ-ФОКА. Для нахождения минимума энергии Хартри-Фока была применена стандартная процедура минимизации по Канторовичу [1, 10], обычно называемая в химической литературе методом самосогласования: на каждом шаге решается линейная проблема на собственные значения для оператора Хартри-Фока с использованием приближения с предыдущего шага для формирования V. В качестве начального приближения были взяты минимальные собственные значения и соответствующие им собственные векторы для линейной части оператора Хартри-Фока: $[T + V]\bar{\phi}_0 = \lambda\bar{\phi}_0$,

$[T + V + V(\bar{\phi}_{i-1})]\bar{\phi}_i = \lambda\bar{\phi}_i$, $i = 1, \ldots$.

Здесь $\bar{\phi}_i$ — набор уточняемых собственных векторов, которые соответствуют минимальным собственным значениям.

В работе [13] доказана сходимость такого метода для одной электронной пары и высказано предположение, что для нескольких электронных пар также будет наблюдаться сходимость.

В настоящее время наиболее популярными методами для нахождения одного или нескольких минимальных собственных значений и соответствующих собственных векторов эрмитовой матрицы являются методы Ланцоша с чебышевским ускорением в произвольном случае [14] и метод Давидсона [16], если известен хороший предобусловливатель. Из (25)–(27) оператор T^{-1} будет хорошим предобусловливателем для метода Давидсона, так как $\|T\|_F^2$ растет по сравнению с $\|V+V\|_F^2$ линейно относительно размера задачи. Так как T^{-1} — также трехуровневая циркулянтная матрица, то затраты на построение и умножение на такую матрицу будут порядка $N\log_2 N$, что значительно меньше $(m^2 N\log_2 N)$, чем умножение на весь оператор Хартри–Фока. Более того, в качестве предобусловливателя была взята симметричная трехуровневая циркулянтная матрица C, которая находилась из

$$\min_{C} \|T + V + \frac{4}{h}\alpha\,\mathrm{diag}(z) + \frac{4}{h}\beta(2m-1)I - C\|_F.$$

Очевидно, что C можно построить за порядка $N\log_2 N$ арифметических операций.

Легко показать, что если сдвинуть C на величину, много меньшую $\frac{4}{h}\beta(2m-1)$, то можно получить такую матрицу, у которой число обусловленности по $\|.\|_2$ будет не более N.

Как оказалось, отношение

$$\frac{\|T + V + \frac{4}{h}\alpha\,\mathrm{diag}(z) + s\frac{4}{h}\beta(2m-1)I - C\|_F}{\|C\|_F} \tag{28}$$

много меньше единицы и имеет порядок $\sqrt{N}$, то есть на более частой сетке это отношение меньше. В [16] показано, что сходимость алгоритма Давидсона зависит от этой величины.

Из этих соображений, для нахождения минимальных собственных значений и соответствующих собственных векторов эрмитовой матрицы, возникающей на каждом шаге процедуры минимизации по Канторовичу, был выбран метод Давидсона.

В настоящее время существует множество разновидностей этого метода, поэтому для ясности выпишем один из них, а затем объясним, почему был выбран именно этот метод.

Пусть $A \in \mathbb{C}^{N\times N}$ — эрмитова матрица, для которой необходимо найти минимальную собственную пару (ν, ψ). Предобусловливатель $P \simeq A$ такой, что $(P - \gamma I)$ легко обратимая. Пусть ψ_0 — начальное приближение к такому собственному вектору. Тогда:

$\nu_0 = \psi_0^* A \psi_0$,

$q_0 = \psi_0$,

$\forall i = 1, \dots,$ MAXSPACE:

$r_i = A\psi_{i-1} - \nu_{i-1}\psi_{i-1}$ — вычисление невязки i-й итерации,

если $\|r_i\|_F < \text{eps}$, то решение найдено, прекратить итерации.

Иначе:

$q_i = (P - I\nu_{i-1})^{-1} r_i$,

ортогонализация q_i ко всем предыдущим $q_0, \dots, q_{i-1}$,

$H = (q_0, \dots, q_i)^* A (q_0, \dots, q_i)$,

нахождение минимального собственного значения ν_i и соответствующего собственного вектора α_i матрицы H

$\psi_i = (q_0, \dots, q_i)\alpha_i$

Если $i =$ MAXSPACE, то $\psi_0 = \psi_i$, и перейти на шаг 1.

На каждой итерации необходимо хранить $i + 1$ вектор из $\mathbb{C}^N$, выполнять порядка $4iN$ арифметических операций во время ортогонализации и построения очередного приближения и два раза умножать вектор на матрицу.

Альтернативой этому методу может быть метод Давидсона с одним умножением матрицы на вектор на итерацию, но для него пришлось бы хранить дополнительно i векторов вида $A(q_0, \dots, q_i)$ и выполнять на iN арифметических операций больше во время ортогонализации.

Если решается приближенная задача, когда $A = T+V$, то арифметические затраты на умножение вектора на матрицу составляют от $3.5N$ до $13.5N$. Следовательно, использование алгоритма с одним умножением матрицы на вектор на итерацию не оправдано ни с точки зрения используемой памяти, ни с точки зрения выполняемых арифметических операций.

В то же время, если выполняется умножение вектора на исходный оператор Хартри-Фока без использования каких-либо упрощений, то число арифметических операций на одно такое умножение может достигать $500N$, в этом случае основной определяющей скорость стадией становится умножение матрицы на вектор. Возникает дилемма: один алгоритм требует в два раза меньше памяти, а другой — работает в два раза быстрее. Был выбран более медленный алгоритм, так как для большинства задач объем данных во время итераций был наиболее критическим ресурсом.

Для нахождения нескольких собственных векторов в настоящее время используются два подхода. Первый состоит в том, что выполняется тот же метод Давидсона, но вместо одного искомого вектора ищутся одновременно несколько векторов. Второй подход использует тот факт, что на каждой итерации решение ортогонализуется ко всем уже найденным собственным векторам.

Первый подход оказался неприемлемым в нашем случае, так как число итераций в таком блочном методе оставалось примерно таким же, а это влекло за собой увеличение объема используемой памяти в несколько раз. Поэтому использовался только второй подход — ортогонализация очередного приближения ко всем уже найденным собственным векторам.

Как оказалось, алгоритм Давидсона оказался подходящим для данной задачи, так как число итераций для тестовых задач было не более 50.

Было сделано еще одно упрощение, связанное с выбором начального приближения. Так как V достаточно хорошо приближается по формуле (24), то вначале решали упрощенную нелинейную задачу на собственные значения Хартри-Фока с таким приближенным оператором. Затем полученное приближение использовали для решения общей задачи Хартри-Фока. В этом случае число итераций при решении общей задачи Хартри-Фока уменьшалось в $2-3$ раза.

3. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ. Был написан программный комплекс для решения уравнений Хартри–Фока с использованием базиса на конечных элементах. Этот комплекс позволяет решать задачи на персональных компьютерах с использованием дисковой памяти.

Чтобы продемонстрировать эффективность этого подхода, был написан программный комплекс и проведены численные эксперименты на компьютере `Pentium-120` с 32 Мбайт оперативной памяти. Таким образом, большинство задач можно решать и на не очень мощном компьютере.

Затраты на оперативную память для задачи из N базисных функций были порядка $5N$ слов с плавающей точкой и для дисковой памяти — порядка $N(4M+C)$ слов с плавающей точкой, где C — величина в пределах от 10 до 100, которая зависит от скорости сходимости итерационного процесса.

Были проведены следующие эксперименты.

- Изучалась зависимость точности вычисления энергии образования, арифметических затрат, объема требуемой оперативной и дисковой памяти от числа базисных функций для нескольких хорошо известных молекул: H_2, CH_4, циклопропана и винилциклопропана. Эти эксперименты были направлены на то, чтобы продемонстрировать эффективность полученного алгоритма.

- Изучалось качество предобусловливателей и скорость сходимости алгоритма для решения этих же молекулярных систем.

- Изучалась зависимость энергии образования винилциклопропана от угла поворота вокруг связи, которая соединяет трехчленный цикл и двойную связь. Этот тест в настоящее время считается достаточно сложным, так как большинство квантовохимических программ предсказывают неправильную зависимость.

Базис был построен на кусочно-линейных базисных функциях. Итерации прекращали, когда невязка в L_2 достигала 10^{-7}.

В табл. 1 и 2 представлены зависимости объема оперативной и дисковой памяти и времени вычисления от числа базисных функций, для различных молекул. Обнаружено, что задачу с 12 электронами и с 262144 базисными функциями можно решить за приемлемое время.

Таблица 1. Зависимость объема оперативной (M) и дисковой (D) памяти (Мбайт) от числа базисных функций

Размер сетки	H_2		CH_4		C_3H_6	
	M	D	M	D	M	D
16^3	1.2	2	1.2	3	1.2	5
32^3	2.5	15	2.5	20	2.5	35
64^3	11.2	120	11.2	160	11.2	280

Таблица 2. Зависимость времени вычисления задачи на компьютере Pentium 120 *с 32 Мбайт оперативной памяти от числа базисных функций*

Размер сетки	H_2	CH_4	C_3H_6
16^3	1 м	4 м	—*
32^3	5 м	23 м	2 ч 17 м
64^3	28 м	2 ч 34 м	15 ч

Таблица 3. Зависимость точности полной энергии Хартри-Фока от числа базисных функций

Размер сетки	H_2	CH_4	C_3H_6
16^3	−1.315	−3.513	—*
32^3	−1.359	−3.585	−37.221
64^3	−1.369	−3.597	−37.240

∗ Сходимость не была достигнута.

Таблица 4. „Близость“ во фробениусовой норме предобусловливателя и исходной матрицы для различных размеров матриц при решении задачи с циклопропаном

Размер базиса	$\frac{\|T+V-P_C\|_F}{\|T+V\|_F}$	$\frac{\|T+V+V-P_C\|_F}{\|T+V+V\|_F}$
8^3	0.387	0.502
16^3	0.182	0.264
32^3	0.073	0.114

Табл. 3 демонстрирует достигаемую в этих экспериментах точность. Как видно из таблицы, приемлемая точность достигалась уже на сетках $32 \times 32 \times 32$. В качестве проверки правильности решения вычислялось минимальное собственное значение матрицы

вторых производных. Во всех задачах это была положительная величина, что говорит о достижении минимума.

Замечательным в этом подходе является тот факт, что двух- и четырехцентровые интегралы не зависят от взаимного расположения атомов, поэтому при оптимизации энергии по геометрии не было необходимости пересчитывать эти интегралы. Более того, вычисление производных энергии по координатам ядер выполнялось с использованием структур, аналогичных матрицам двухцентровых интегралов, — это потребовало порядка N арифметических операций. Алгоритм оказался хорошо масштабируемым по отношению к размеру задачи и к числу электронов.

Заметим, что при задании какого-нибудь абсурдного с точки зрения химии расположения ядер атомов (например, расстояния между двумя углеродами менее 0.5Å) алгоритм находил некоторое электронное распределение, которое соответствовало глобальному минимуму энергии Хартри-Фока, но результирующая энергия была очень высокая, что говорит в пользу такого метода.

Таблица 5. Зависимость числа итераций от числа базисных функций и использования циркулянтного предобусловливания

Размер базиса	Число итераций для нахождения начального приближения		Число итераций основной итерационной схемы		Число умножений матрицы на вектор
	с пред.	без пред.	с пред.	без пред.	с пред.
20^3	368	849	473	902	3470
30^3	366	1486	356	1354	3951
40^3	367	> 2000	342	> 2000	2873
50^3	368	> 3000	321	> 3000	2774
60^3	370	> 3000	301	> 3000	2690

Теперь продемонстрируем эффективность предложенных предобусловливателей (табл. 4). Условия теоремы 3 для всех таких матриц всегда выполнялись, поэтому можно гарантировать, что полученная матрица невырожденная. Видно, что с ростом размера задачи растет качество предобусловливания, что неизменно сказывается на сходимости итерационной схемы (табл. 5). К сожалению, предобусловливатель на основе мультициркулянта показал практически такие же результаты, как и на основе циркулянта (табл. 4). Так как сложность реализации мультициркулянтного предобусловливания очень высока для тех матриц, размер которых не есть степень двух, то было принято решение не использовать его в большинстве расчетов.

Далее рассмотрим скорость сходимости этого алгоритма. Для молекулы винилциклопропана была проведена серия экспериментов с различным числом базисных функций. Этот тест достаточно сложен, так как одновременно вычислялось 19 собственных функций. Размер области, в которую была помещена эта молекула, составлял около 6 Å. На рис. 1 показана зависимость точности от размера базиса. Как видно, на сетке $50 \times 50 \times 50$ наблюдается асимптотическая точность.

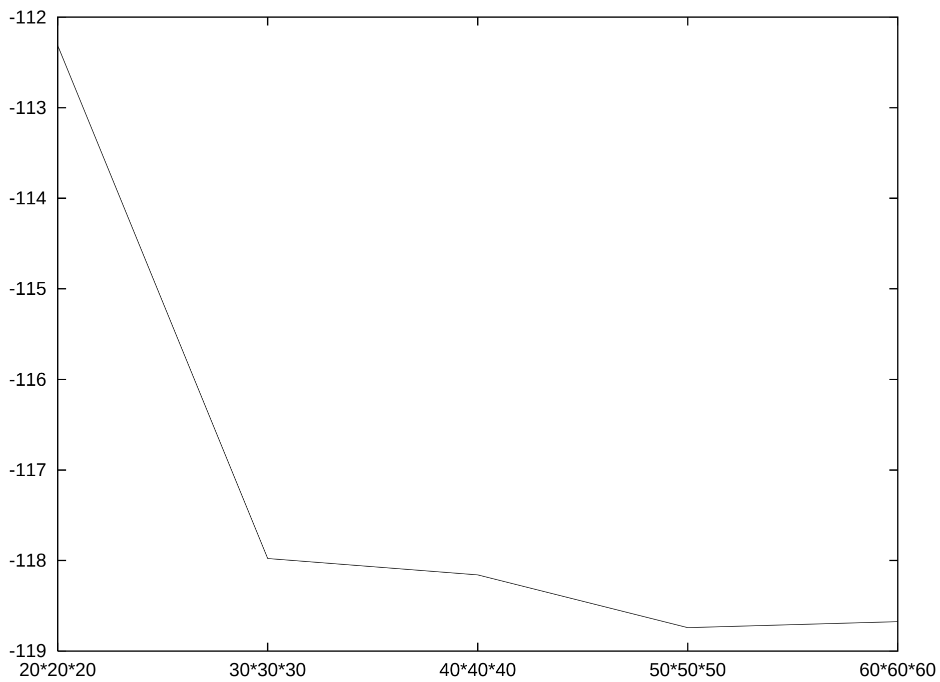


Рис. 1. Зависимость наблюдаемой точности от размера базиса

В табл. 5 показана скорость сходимости алгоритма. Эти результаты наглядно демонстрируют тот факт, что с увеличением размеров матрицы эффективность предобусловливателя увеличивается, и, как следствие, уменьшается число итераций и умножений матрицы на вектор. Заметим, что число итераций здесь дано для нахождения всех 19 собственных векторов, а в колонке „Число итераций основной итерационной схемы“ указано среднее число итераций. Видно, что для большинства всех тестов среднее число итераций с предобусловливателем для нахождения одного собственного вектора не превышает 25, в то же время, для больших размеров матриц без предобусловливателя задачу не удалось решить вообще.

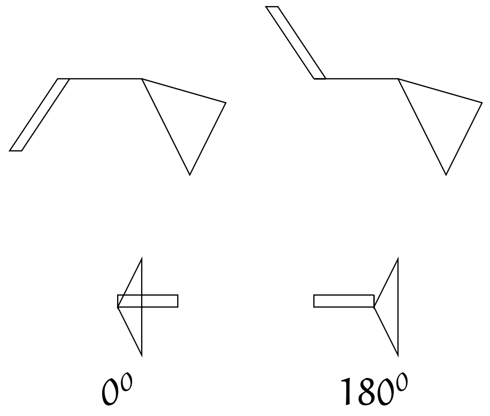


Рис. 2. Взаимное расположение двойной связи и трехчленного цикла в винилциклопропане

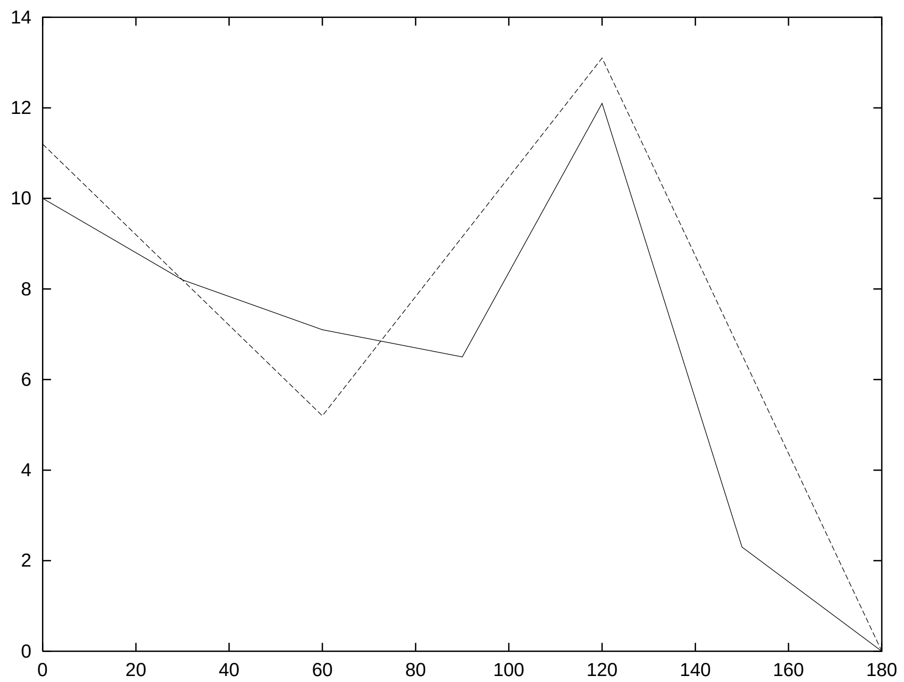


Рис. 3. Зависимость вычисленной (прямая линия) и ранее известной (пунктирная линия) энергии от угла вращения по связи, которая соединяет трехчленный цикл и двойную связь в винилциклопропане

Теперь рассмотрим одну достаточно сложную с точки зрения квантовой химии задачу — вычисление зависимости энергии образования молекулы от конформационного вращения одной части молекулы относительно другой. В качестве примера была взята молекула винилциклопропана и изучена такая зависимость при вращении по связи, которая соединяет трехчленный цикл и двойную связь (рис. 2). Было выполнено семь численных экспериментов по вычислению энергии образования винилциклопропана при повороте винильной группы относительно цикла на углы от 0 до 180^0 с шагом 30^0. Полученные результаты в базисе $40 \times 40 \times 40$ базисных функций были в хорошем согласии с известными на данный момент расчетными данными [17] (рис. 3).

Выбор именно такого размера базиса обусловлен тем, что на предварительном расчете (см. рис. 1) было получено, что начиная с этих размеров точность решения практически не улучшается.

Общее время расчета на компьютере `Pentium II 300` с 128 Мбайт оперативной памяти составило около 60 часов.

Литература

1. Herbe W. J., Radom L., Schleyer P. v. R., Pople J. A. *Ab initio Molecular Orbital Theory.* — Springer-Verlag, 1986.

2. Seitsonen A. P., Puska M. J., Neiminen R. M. Real-space electronic-structure calculations: Combination of the finite-difference and conjugate gradient methods // *Phys. Rev. B.* 1995. V. 51(20). P. 14057–14061.

3. Payne M. C., Teter M. P., Allan D. C., Arias T. A., Joannopoulos J. D. Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients // *Rev. of Mod. Phys.* 1992. V. 64(4). P. 1045–1097.

4. Umar A. S., Strayer M. R., Wu J. S., Dean D. J, Güclü M. C. Nuclear Hartree-Fock calculations with splines // *Phys. Rev. C.* 1991. V. 44(6). P. 2512–2521.

5. Calais J. L. Wavelets — Something for Quantum Chemistry // *Int. J. Quant. Chem.* 1996. V. 58. P. 541–548.

6. Hall G. G., Rees D. A Discrete Look at Localization // *Int. J. Quant. Chem.* 1995. V. 53. P. 189–205.

7. Ibraghimov I. V., Tyrtyshnikov E. E. An Application of Toeplitz Matrixes to Quantum Chamichal Problems. *Proceedings of the 9th*

International Conference "Computational Modeling and Computing in Physics". Dubna, 1997, P. 327–331.

8. Де Бор К. *Практическое руководство по сплайнам* / Пер. с англ. — М.: Радио и связь, 1985.

9. Тыртышников Е. Е. *Краткий курс численного анализа.* — М.: ВИНИТИ, 1994.

10. Хейгеман Л., Янг Д. *Прикладные итерационные методы* / Пер. с англ. — М.: Мир, 1986.

11. Демьянов В. Ф., Малоземов В. Н. *Введение в минимакс.* — М.: Наука, 1972.

12. Воеводин В. В., Тыртышников Е. Е. *Вычислительные процессы с тёплицевыми матрицами.* — М.: Наука, 1987.

13. Локшин А. А., Саакян А. С., Тарасов Ю. И. *Параметрические зависимости собственных значений.* — М.: МГУ, 1997.

14. Парлетт Б. *Симметричная проблема собственных значений* / Пер. с англ. — М.: Мир, 1983.

15. Локшин А. А., Лопатников С. Л., Тарасов Ю. И. *Метод сжатых отображений в симметричной проблеме собственных значений.* — М.: МГУ, 1995.

16. Crouzeix M., Philippe B., Sadkane M. The Davidson Method // *SIAM J. Sci. Comp.* 1994. V. 15(1). P. 62–76.

17. Верещагин А. Н., Катаев В. Е., Бредихин А. А. *Конформационный анализ углеводородов и их производных.* — М.: Наука, 1990.

Параллельные алгоритмы БПФ и скалярного произведения векторов*

И. В. Ибрагимов

Аннотация

В статье рассмотрены параллельные реализации алгоритмов быстрого преобразования Фурье и вычисления суммы вектора для компьютеров с распределенной памятью. Приводятся теоретические оценки времен выполнения таких алгоритмов. Предложены такие реализации, что (для заданной компьютерной платформы) если размер задачи пропорционален числу процессоров, то эффективность реализации этих алгоритмов не зависит от числа процессоров, вовлеченных в решение задачи, — наблюдается масштабируемость. Теоретические результаты были подтверждены численными экспериментами на компьютерах CRAY T3D *и* MBC 100.

1. ВВЕДЕНИЕ. В настоящее время для решения многих задач линейной алгебры применяют массивно-параллельные компьютеры с распределенной памятью, число процессоров в которых достигает нескольких сотен. Важнейшим условием эффективной работы алгоритмов, применяемых на таких компьютерах, является правильное распределение данных по процессорам, при котором для решения задачи пересылается минимальное количество данных от одного процессора к другому.

Нетривиальными, и вместе с тем очень распространенными, являются задачи типа вычисления БПФ, глобального суммирования и др. Сложность создания параллельных реализаций для таких

---

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155), фонда Миннауки (грант №0201.01.013) и фонда Volkswagen-Stiftung (грант № VW I/71 493).

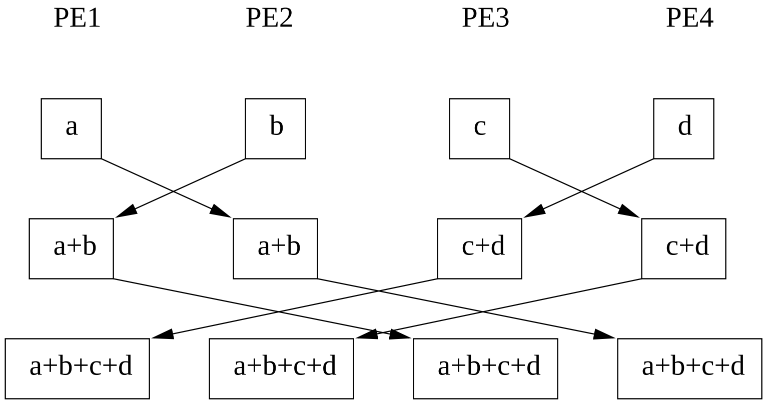


Рис. 1. Алгоритм сдваивания для вычисления суммы распределенного вектора

последовательных алгоритмов заключается в том, что каждое выходное данное зависит от всех входных, что не позволяет разбить на группы входные данные и распределить их по процессорам так, чтобы избежать обменов данными между процессорами.

В то же время эти алгоритмы объединяет свойство рекуррентности: любая задача размерности N сводится к решению двух или нескольких подобных задач в соответственное число раз меньшей размерности, так что вычисления таких алгоритмов можно выполнить за несколько рекуррентных шагов, на каждом таком шаге каждое выходное данное зависит только от некоторого числа входных данных, которое значительно меньше, чем общее число входных данных. В результате, данные можно распределить так, что вычисления на каждом таком шаге можно было выполнить полностью параллельно. Межпроцессорные обмены необходимо выполнять только между такими параллельными шагами.

Варьируя число параллельных шагов, можно изменить число пересылок и общий объем пересылаемых данных, причем при увеличении числа пересылок уменьшается объем пересылаемых данных. Покажем это на простом примере.

Пусть необходимо вычислить M сумм векторов длины N на P процессорах с распределенной памятью и результат распределить по всем процессорам. Пусть для простоты P — это степень 2 и N кратно P. Рассмотрим два параллельных алгоритма для выполнения этой работы — первый для случая, когда M очень мало, второй, когда M велико, как минимум больше, чем число процессоров. Для малых M показано, что алгоритм сдваивания [6] (рис. 1) требует минимального числа пересылок, а так как M мало, то, следовательно, объем пересылаемых данных существенной роли играть не будет и можно говорить, что будет требоваться минимальное время на пересылки. Заметим, что в этом случае надо выполнить $\log_2 P$ операций пересылки данных длины M.

Для очень больших в настоящее время известен алгоритм [4], когда за $2P-1$ шагов вычисляется сумма элементов M векторов так, что на каждом шаге пересылается только $\frac{M}{P}$ чисел, то есть суммарно объем пересланных данных $(2P-1)\frac{M}{P} \simeq 2M$ не зависит от числа процессоров, в отличие от предыдущей параллельной реализации (рис. 2).

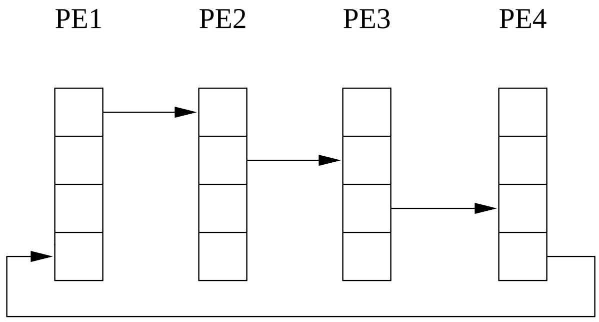


Рис. 2. Алгоритм последовательной пересылки для вычисления суммы распределенного вектора

В данной работе предлагается изучить такого рода параллельные реализации алгоритмов и построить целый класс этих параллельных реализаций для алгоритмов БПФ и вычисления суммы вектора. Тогда для заданных размеров задачи, числа процессоров и характеристик коммуникационной сети между процессорами можно найти оптимальную на таком классе параллельную реализацию.

2. ТЕОРЕТИЧЕСКИЕ ИССЛЕДОВАНИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ БПФ И ВЫЧИСЛЕНИЯ СУММЫ ВЕКТОРОВ. Прежде чем исследовать параллельные реализации алгоритмов, введем понятия, которые позволят нам характеризовать степень распараллеливания алгоритма. Пусть N — размер исходного вектора, M — число независимых одинаковых задач, которые необходимо решить, P — число процессоров в параллельном компьютере, L — число слов в памяти одного процессора. Здесь и далее наложим небольшие ограничения на N, M, P и L. Пусть P всегда является степенью 2, а N кратно P. Будем говорить, что максимально большая задача

характеризуется равенством $PL = NM$. Введем обозначения для описания характеристик скорости работы алгоритма:

$T(N)$ — число операций с плавающей точкой в последовательном алгоритме для задачи размера N;

$T_O(N, M, P)$ — время на обмены данными между процессорами при параллельной реализации алгоритма на P процессорах для задачи размера N;

$S(N, M, P) = \dfrac{MT(N)}{\frac{MT(N)}{P} + T_O(N, M, P)}$ — ускорение параллельной реализации алгоритма;

$E(N, M, P) = \dfrac{S(N, M, P)}{P} = \dfrac{MT(N)}{MT(N) + PT_O(N, M, P)}$ — эффективность параллельной реализации алгоритма.

Введем понятие масштабируемости и супермасштабируемости. Параллельную реализацию алгоритма будем называть масштабируемой на числе процессоров $[P_1, P_2]$, если для $\forall P_i < P_j \in [P_1, P_2]$ и для максимально большой задачи: $E(P_i) \leqslant E(P_j)$ [5].

Параллельную реализацию алгоритма будем называть супермасштабируемой на числе процессоров $[P_1, P_2]$, если для $\forall P_i < P_j \in [P_1, P_2]$ для одних и тех же N и M: $E(P_i) \leqslant E(P_j)$.

Будем рассматривать только компьютеры с распределенной памятью. Каждый процессор может работать только со своей локальной памятью. Пусть между процессорами есть коммуникационная сеть, которая позволяет передавать только синхронно данные от любого процессора любому другому с одними для всех вре́менными характеристиками: $\sigma + N\tau$, где σ — латентность сети — время задержки процессора для старта пересылки, измеряется в числе операций с плавающей точкой, которое можно выполнить на одном процессоре данного массивно-параллельного компьютера за это время; τ — скорость передачи данных из локальной памяти посылающего процессора в память принимающего процессора, измеряется в числе операций с плавающей точкой, которое можно выполнить на одном процессоре данного массивно-параллельного компьютера за время передачи одного числа.

В дополнение к этой модели будем рассматривать полностью асинхронную модель передачи данных. В этом случае посылающий процессор тратит на пересылку только время σ, а принимающий процессор получает данные через время $\sigma + N\tau$.

Такой способ описания архитектуры массивно-параллельного компьютера позволяет охватить многие современные модели компьютеров с распределенной памятью без значительной потери точности модели [2].

Определим класс последовательных алгоритмов. Будем рассматривать два типа алгоритмов, которые различаются числом выходных данных, тип 1 — это одно число, тип 2 — это N чисел. Пусть вычислительная сложность алгоритмов не зависит от входных данных и пусть $W(A)$ — вычислительная сложность алгоритма, то есть число арифметических операций при выполнении алгоритма A.

Назовем алгоритмом типа 1 последовательный алгоритм вычисления $c = A(\bar{x})$, $c \in \mathbb{R}$, $\bar{x} \in \mathbb{R}^N$, для которого при $N > 1$ всегда можно найти такие алгоритмы A_1 и A_2, которые также принадлежат классу алгоритмов типа 1, и если $x = (\bar{z}_1^T, \dots, \bar{z}_k^T)^T$, то $A(\bar{x}) = A_1(A_2(\bar{z}_1), \dots, A_2(\bar{z}_k))$ и $W(A) = W(A_1) + kW(A_2)$.

Алгоритмом типа 2 назовем последовательный алгоритм вычисления $\bar{y} = A(\bar{x})$, $\bar{y} \in \mathbb{R}^N$, $\bar{x} \in \mathbb{R}^N$, для которого при $N > 1$ всегда можно найти такие алгоритмы A_1 и A_2, которые тоже принадлежат классу алгоритмов типа 2, и если $x = (\bar{z}_1^T, \dots, \bar{z}_k^T)^T$ и $y = (\bar{v}_1^T, \dots, \bar{v}_k^T)^T$, то

$$v_i = A_1([A_2(\bar{z}_1)]_i, \dots, [A_2(\bar{z}_k)]_i), i = 1, \dots, k$$

где результатом операции $[a]_i$ является i-й элемент вектора $\bar{a}$, и $W(A) = W(A_1) + kW(A_2)$.

Приведем примеры подобных алгоритмов.

Вычисление нормы одного или нескольких векторов. Входные данные — M векторов размера N, выходные — M чисел — норм. Этот алгоритм является неотъемлемой частью большинства итерационных схем, а в виде скалярного произведения двух векторов — алгоритма умножения вектора на матрицу и матрицы на матрицу. Это алгоритм типа 1, так как по алгоритму необходимо выполнить $N-1$ суммирований квадратов значений элементов вектора, а для операции суммирования допускается менять порядок выполнения.

Поиск минимального или максимального значения элемента одного или нескольких векторов. Эта задача алгоритмически аналогична предыдущей, так как для поиска минимума или максимума также можно менять порядок выполнения.

Одномерное БПФ над одним или несколькими векторами: M

входных и выходных векторов размера N [1]. Это алгоритм типа 2. Пусть x_1, x_2 — векторы размеров $N/2$, N — натуральное и четное. Одномерное БПФ для $x = (x_1^T, x_2^T)^T$ $z = F(x)$ можно представить в виде $y_1 = F(x_1), y_2 = F(x_2), y(i) = y_1(i) + Wy_2(i), y(i + N/2) = y_1(i) - Wy_2(i), i = 1, \ldots, N/2$, где W — некоторое число.

БПФ любой размерности более одного. Этот алгоритм эквивалентен одномерному БПФ по каждой размерности.

Ряд алгоритмов не содержит вычислений, например: транспонирование матриц, специальные перестановки векторов и др. Так, алгоритм транспонирования матрицы размеров $N \times N$ может быть представлен в виде транспонирования блочной $K \times K$ матрицы, $\frac{N}{K}$ — целое. Для каждого блока такой матрицы транспонирование можно выполнить независимо, а далее транспонировать только блочную матрицу.

Рассмотрим алгоритм типа 1. Возьмем последовательность чисел $N_1, \ldots, N_q$, такую, что $\prod_{i=1}^{q} N_i = N$. Выполним алгоритм, используя, по возможности, все, что нам о нем известно: пусть $\hat{x}$ — q-мерный массив размерности $N_1, \ldots, N_q$. Пусть первый шаг состоит в следующем: для всех индексов $i_2, \ldots, i_q$, таких, что $i_j = 1, \ldots, N_j$, выполняется алгоритм для вектора из N_1 чисел. Результат поместим в $q-1$-мерный массив размерности $N_2, \ldots, N_q$. Аналогично выполним все остальные $q-1$ шагов.

Пусть N равно числу процессоров. Поместим каждый элемент q-мерного массива на отдельный процессор. Вначале рассмотрим только синхронные пересылки.

Каждая такая операция состоит из рассылки каждый каждому данных в группе из N_1 процессоров. Если $\frac{M}{N_1} \geqslant 1$, то можно так разослать данные, что на каждом процессоре окажется $\frac{M}{N_1}$ задач. Если $\frac{M}{N_1} < 1$, то все будет определяться временем вычисления одной задачи. Назовем $\frac{M}{N_1} \geqslant 1$ случаем A и $\frac{M}{N_1} < 1$ — случаем B. Тогда для случаев A и B суммарное время работы первого шага будет равно соответственно:

$$\frac{M}{N_1}W(A_1) + 2(N_1 - 1)\left(\sigma + \tau\frac{M(N_1-1)}{N_1}\right)$$

$$\frac{M}{N_1}W(A_1) + (N_1 - 1)(\sigma + \tau M)\,.$$

В результате на каждом процессоре появятся данные, необходимые для вычисления следующего шага. Аналогично выполняются все остальные шаги. В итоге результат работы распределен по всем процессорам.

Теперь обобщим этот алгоритм на случай $N > P$. Выполним на каждом процессоре независимо алгоритм типа 1 (пусть это алгоритм A_0) для всех данных, лежащих на этом процессоре. В результате получится одно число на каждом процессоре. Мы привели эту задачу к задаче для $N = P$.

Выпишем общее время работы такого алгоритма для случаев A и B:

$$T_{A,1} = \frac{MW(A_0)}{P} + 2\sum_{i=1}^{q}(N_i - 1)\left(\sigma + \tau\frac{M(N_i - 1)}{N_i}\right), \tag{1}$$

$$T_{B,1} = \frac{MW(A_0)}{P} + \sum_{i=1}^{q}(N_i - 1)(\sigma + \tau M) \tag{2}$$

при условии $\prod_{i=1}^{q} N_i = P$ и будем искать $\min(T_{A,1}, T_{B,1})$ по всем N_i, $i = 1, \dots, q$, причем N_i должны быть целыми положительными числами. В общем случае такая задача довольно сложна. Нас интересует алгоритм вычисления скалярного произведения векторов, поэтому упростим формулы (1) и (2), подставив туда явные значения для $W(A_i) = N_i$, а для $W(A_0) = \frac{N}{P}$. Тогда

$$T_{A,1} = \frac{MN}{P} + 2\sum_{i=1}^{q}\left[(N_i - 1)\sigma + \frac{M(N_i - 1)}{N_i}\tau\right], \tag{3}$$

$$T_{B,1} = \frac{MN}{P} + (M + \sigma + M\tau)\sum_{i=1}^{q}(N_i - 1). \tag{4}$$

Очевидно, что для $T_{B,1}$ минимум наступает при $N_i = 2$, $i = 1, \dots, q$. Легко показать, что минимум $T_{A,1}$ наступает при $N_i = C$, $i = 1, \dots, q$. Тогда выражения (3) и (4) упростятся:

$$T_{A,1} = \frac{MN}{P} + \frac{2Mq\tau(C - 1)}{C} + 2\sigma q(C - 1),$$

$$T_{B,1} = \frac{MN}{P} + (\sigma + M\tau)\log_2 P.$$

Теперь найдем минимум $T_{A,1}$ по C (заметим, что $C^q = P$) и сравним с $T_{B,1}$. Минимум $T_{A,1}$ по C наступает при

$$\frac{\delta T_{A,1}}{\delta C} = 0 \to \frac{\tau M}{\sigma} = \frac{C \ln C}{C - \ln C - 1},$$

что при грубых расчетах можно полагать как

$$C \simeq e^{\frac{M\tau}{\sigma}}. \quad (5)$$

Так как C должно быть целым, то достаточно перебрать целые значения C в окрестности (5) для нахождения наилучшего значения C. Покажем, что при этом значении C алгоритм масштабируем. Для этого вычислим

$$T_{A,1} \simeq \frac{MN}{P} + 2\ln P\sigma\left(1 + \frac{\sigma e^{\frac{M\tau}{\sigma}}}{M\tau}\right).$$

Пренебрежем ростом $\ln P$. Увеличим число процессоров (P) и размер задачи (N) в два раза, тогда $T_{A,1}$ практически не изменится, что говорит о масштабируемости алгоритма. При увеличении числа независимых задач (M) вместо размера задачи масштабируемости не будет, но рост времени пересылок будет незначительным — только как $\sqrt{M}$. Заметим, что при $M < C$ алгоритм A нельзя использовать, в этом случае следует применять алгоритм B.

Аналогично рассмотрим алгоритмы типа 2. Выражения для (1) и (2) будут идентичными, изменится только $W(A)$. Для БПФ $W(A) = N\log_2 N$, где N — размер вектора на входе алгоритма A. Тогда

$$T_{A,2} = \frac{MN\log_2 N}{P} + \frac{2MNq\tau(C-1)}{C^2} + 2\sigma q(C-1),$$

$$T_{B,2} = \frac{MN\log_2 N}{P} + (\sigma + MN\tau)\log_2 P.$$

Заметим, что после обработки данных каждым процессором нет нужды вначале посылать эти данные обратно, а затем пересылать их на следующем уровне другим — эту операцию можно сделать за одну пересылку, тогда оценка для $T_{A,2}$ улучшится:

$$T_{A,2} = \frac{MN\log_2 N}{P} + \frac{MN(q+1)\tau(C-1)}{C^2} + \sigma(q+1)(C-1).$$

Далее все аналогично случаю для $T_{A,1}$, поэтому приведем окончательные формулы. Минимум $T_{A,2}$ по C наступает при

$$\frac{\delta T_{A,2}}{\delta C} = 0 \to C \simeq \sqrt{\frac{\tau MN}{\sigma}},$$

$$T_{A,2} \simeq \frac{MN\log_2 P}{P} + \frac{2\ln P}{\ln\left(\frac{\tau MN}{\sigma}\right)}\sqrt{2\sigma\tau}\sqrt{MN}.$$

Усложним модель. Пусть данные от процессора к процессору могут передаваться асинхронно. Тогда выражения для $T_{A,1}$ и $T_{A,2}$ преобразуются в

$$T_{A,1} = \frac{MN}{P} + \frac{2Mq\tau(C-1)}{C^2} + \frac{2Mq\tau}{C^2} + 2\sigma q(C-1),$$

$$\begin{aligned} T_{A,2} = \frac{MN\log_2 N}{P} &+ \frac{MN\tau(q+1)(C+1)}{C^2} \\ &+ \frac{MN\tau(q+1)}{C^2} + \sigma(q+1)(C-1), \end{aligned}$$

так как все пересылки, кроме первой, на каждом шаге можно выполнить на фоне вычислений. Для $T_{B,1}$ и $T_{B,2}$ ничего асинхронно передавать не удается. Минимумы $T_{A,1}$ и $T_{A,2}$ по C наступают при

$$\frac{M\tau}{2\sigma} = \frac{\ln C(C\ln C - C + 1)}{\ln C + 1}, \tag{6}$$

$$\frac{MN\tau}{\sigma} = \frac{C^2(C\ln C - C + 1)}{\ln C + 1}. \tag{7}$$

Это показывает, что область, при которой $C \in [2, P]$, значительно отличается от области при синхронных передачах данных. К сожалению, при $C \in [2, P]$ ситуация с масштабируемостью такая же, как и прежде, то есть если пренебречь ростом $\ln P$, то алгоритм можно назвать масштабируемым.

Отдельно рассмотрим случай, когда необходимо выполнить БПФ для очень больших векторов, а именно таких, что значение C из (7) превышает число процессоров. Тогда единственно возможным параллельным алгоритмом является алгоритм, у которого $C = P$ и $q = 1$, тогда

$$T_{A,2} = \frac{MN\log_2 N}{P} + \frac{2MN\tau}{P^2} + 2\sigma(P-1).$$

Изучим поведение функции при увеличении числа процессоров в два раза при постоянных размере задачи (N) и числе задач (M). Найдем, когда $\dfrac{T_{A,2}(P)}{T_{A,2}(2P)} > 2$, что означает супермасштабируемость:

$$\frac{MN\log_2 N}{P} + \frac{2MN\tau}{P^2} + 2\sigma(P-1) >$$
$$2\left(\frac{MN\log_2 N}{2P} + \frac{2MN\tau}{4P^2} + 2\sigma(2P-1)\right),$$

то есть при $MN > \dfrac{2P^2(3P-1)\sigma}{\tau}$. Это означает, что существуют такие M и N, при которых будет наблюдаться супермасштабируемость.

Подытожим полученные результаты. Для алгоритмов типов 1 и 2 предлагается использовать набор параллельных реализаций алгоритмов, которые зависят от параметра C. В свою очередь, параметр зависит от типа модели обменов (синхронная или асинхронная) и ее характеристик, от числа процессоров и от размера задачи. Предложены формулы для вычисления оптимального значения параметра на этом классе алгоритмов (5), (6), (7). Оказалось, что для алгоритмов типов 1 и 2, если оптимальное значение параметра C больше единицы, наблюдается масштабируемость этого класса параллельных алгоритмов (табл. 1), а при C больше числа процессоров используется алгоритм с $C = P$, для которого наблюдается супермасштабируемость (табл. 2).

Таблица 1. Синхронная модель передачи данных

Тип алгоритма	Размер задачи	Время выполнения	Свойства
$T_{B,1}$	$M < \frac{8\sigma}{\tau}$	$\frac{MN}{P} + (\sigma + M\tau)\log_2 P$	П
$T_{A,1}$	$\frac{8\sigma}{\tau} < M < \frac{\ln P\sigma}{\tau}$	$\frac{MN}{P} + 2\ln P\sigma\left(1 + \frac{\sigma e^{\frac{M\tau}{\sigma}}}{M\tau}\right)$	П
$T_{A,1}$	$M > \frac{\ln P\sigma}{\tau}$	$\frac{MN}{P} + \frac{2M\tau(P-1)}{P} + 2\sigma(P-1)$	М
$T_{B,2}$	$MN < \frac{4\sigma}{\tau(3\log_2 P-1)}$	$\frac{MN\log_2 N}{P} + (\sigma + MN\tau)\log_2 P$	П
$T_{A,2}$	$\frac{4\sigma}{\tau(3\log_2 P-1)} < MN < \frac{P^2\sigma}{\tau}$	$\frac{MN\log_2 N}{P} + \frac{2\ln P}{\ln\left(\frac{\tau MN}{\sigma}\right)}\sqrt{2\sigma\tau}\sqrt{MN}$	П
$T_{A,2}$	$MN > \frac{P^2\sigma}{\tau}$	$\frac{MN\log_2 N}{P} + \frac{2M\tau(P-1)}{P} + 2\sigma(P-1)$	М

П — почти масштабируемый алгоритм, с точностью до $\ln P$
М — масштабируемость

Таблица 2. Асинхронная модель передачи данных

Тип алгоритма	Размер задачи	Время выполнения	Свойства
$T_{B,1}$	$M < \frac{2\sigma}{\tau}$	$\frac{MN}{P} + (\sigma + M\tau)\log_2 P$	П
$T_{A,1}$	$\frac{2\sigma}{\tau} < M < \frac{\ln P\sigma}{\tau}$	$\frac{MN}{P} + \frac{\ln P(M^2\tau^2+\sigma^2)}{M\tau\ln\left(\frac{M\tau}{2\sigma}\right)}$	П
$T_{A,1}$	$\frac{\ln P\sigma}{\tau} < M < \frac{2\sigma P^2(3P-1)}{M\tau}$	$\frac{MN}{P} + \frac{2M\tau(P-1)}{P^2} + 2\sigma(P-1)$	М
$T_{A,1}$	$M > \frac{2\sigma P^2(3P-1)}{\tau}$	$\frac{MN}{P} + \frac{2M\tau(P-1)}{P} + 2\sigma(P-1)$	С
$T_{B,2}$	$MN < \frac{4\sigma}{\tau(3\log_2 P-1)}$	$\frac{MN\log_2 N}{P} + (\sigma + MN\tau)\log_2 P$	П
$T_{A,2}$	$\frac{4\sigma}{\tau(3\log_2 P-1)} < MN < \frac{P^2\sigma}{\tau}$	$\frac{MN\log_2 N}{P} + 2\ln P(MN\tau\sigma^2)^{\frac{1}{3}}$	П
$T_{A,2}$	$\frac{P^2\sigma}{\tau} < MN < \frac{2\sigma P^2(3P-1)}{M\tau}$	$\frac{MN\log_2 N}{P} + \frac{2MN\tau(P-1)}{P^2} + 2\sigma(P-1)$	М
$T_{A,2}$	$M > \frac{2\sigma P^2(3P-1)}{\tau}$	$\frac{MN\log_2 N}{P} + \frac{2MN\tau(P-1)}{P^2} + 2\sigma(P-1)$	С

П — почти масштабируемый алгоритм, с точностью до $\ln P$
М — масштабируемость
С — супермасштабируемость

3. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ. Теперь продемонстрируем эти результаты численными экспериментами. В качестве компьютерных платформ были взяты `CRAY T3D` и `MBC 100`.

`CRAY T3D` состоял из 256 процессоров, с оперативной памятью 16 Мбайт на процессор, передача данных между процессорами осуществлялась частично асинхронно со скоростью около 20 Мслов/с, время задержки — порядка времени передачи 10 слов по сети. На задачах типа БПФ наблюдалось порядка 20–30 миллионов операций с плавающей точкой в секунду (MFlop/s). Тогда в относительных единицах синхронные пересылки характеризуются параметрами $\sigma = 20$ и $\tau = 2$, а асинхронные — $\sigma = 20$, $\tau = 1$ и $\tau = 1$ [3].

`MBC 100` состоял из 64 процессоров, с оперативной памятью 16 Мбайт на процессор, передача данных между процессорами осуществляется полностью асинхронно со скоростью 200 Кслов/с, время задержки — порядка времени передачи 30 слов по сети. На задачах типа БПФ наблюдалось порядка 2 MFlop/s. Тогда в относительных единицах наблюдаются величины $\sigma = 300$, $\tau = 0$ и $\tau = 10$.

Для алгоритмов БПФ и вычисления скалярного произведения были проведены следующие эксперименты. Варьировались:

- длина вектора от $N = 2, \dots, 2^{18}$,
- число процессоров $P = 2, \dots, 2^8$ для компьютера `CRAY T3D` и $P = 2, \dots, 2^6$ для компьютера `MBC 100`,
- тип алгоритма A или B; для алгоритмов типа A будем рассматривать не все алгоритмы, а только те, для которых C точно равны степени двух.

Было проведено более 3000 экспериментов; рассмотрим наиболее интересные из них.

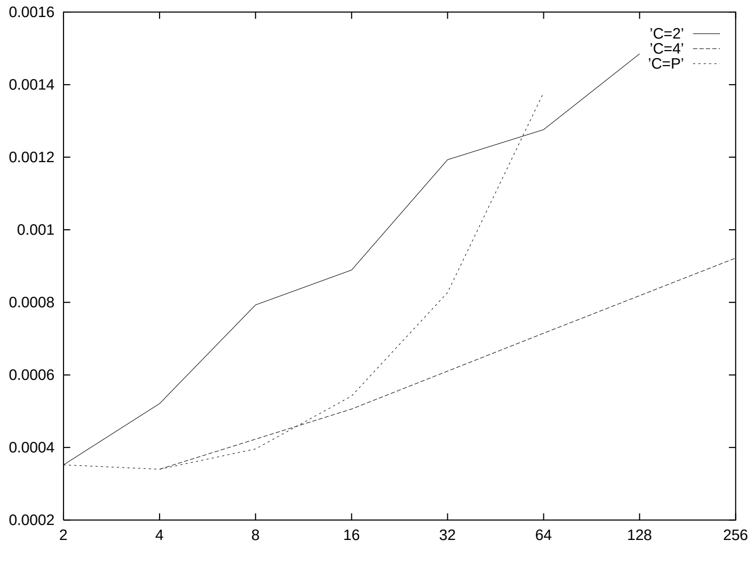


Рис. 3. Времена работы (в секундах) трех типов параллельных реализаций алгоритмов БПФ на компьютере `CRAY T3D` *для*

минимального размера задачи $\mathrm{N} = 2 \times \mathrm{P}$ *в зависимости от числа процессоров, вовлеченных в решение задачи*

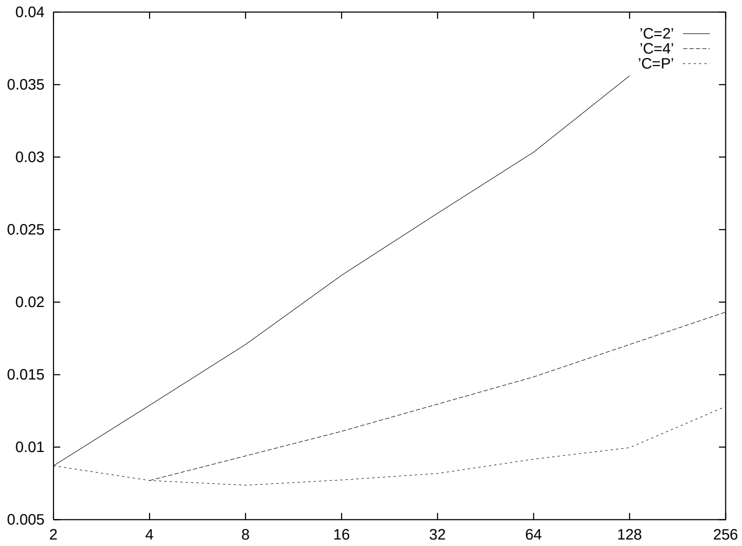


Рис. 4. Времена работы (в секундах) трех типов параллельных реализаций алгоритмов БПФ на компьютере CRAY T3D *для* $\mathrm{N} = 8192 \times \mathrm{P}$ *в зависимости от числа процессоров, вовлеченных в решение задачи*

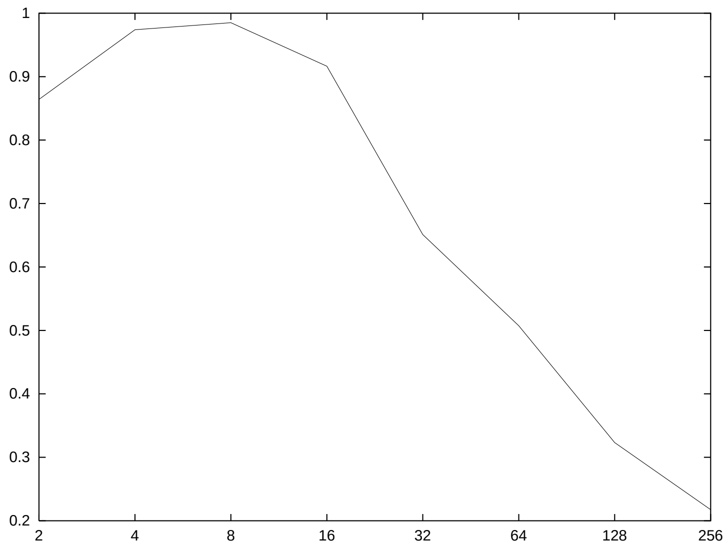


Рис. 5. Эффективность параллельной реализации алгоритма БПФ на компьютере CRAY T3D *для* $\mathrm{N} = 8192$ *в зависимости от числа процессоров, вовлеченных в решение задачи*

Для алгоритма БПФ на компьютере CRAY T3D с использованием асинхронных передач данных были получены времена работы алгоритмов при $C = 2$, $C = 4$ и $C = P$, для минимально возможной задачи ($N = 2 \times P$) (рис. 3) и средней ($N = 8192 \times P$) (рис. 4). Видно, что для минимальной задачи самым быстрым алгоритмом для небольшого числа процессоров является алгоритм с $C = 2$, в то время как для средней задачи этот алгоритм значительно проигрывает по времени другим алгоритмам. А уже для задачи $N = 8192 \times P$ наблюдается супермасштабируемость для алгоритма $C = P$ — эффективность алгоритма растет с ростом числа процессоров для одного и того же размера задачи (рис. 5). На компьютере `MBC 100` наблюдались аналогичные результаты с той только разницей, что не удалось подобрать такой размер задачи, при котором наблюдается супермасштабируемость. Это можно объяснить тем, что характеристики сети `MBC 100` значительно уступают характеристикам `CRAY T3D`.

Аналогичный эксперимент был проведен для алгоритма суммирования на компьютере `MBC 100`, в котором вместо размера задачи варьировалось число одновременно решаемых задач: рис. 6 демонстрирует время работы алгоритма для $M = 1$, а рис. 7 — для $M = 512 \times 1024$. Размер задачи был пропорционален числу вовлеченных в расчет задач. Эксперимент показал, что в зависимости от числа одновременно решаемых задач номер оптимального на этом классе алгоритма изменяется — для маленьких задач более выгодно использовать алгоритм с большим числом шагов, а для больших — с минимальным числом шагов.

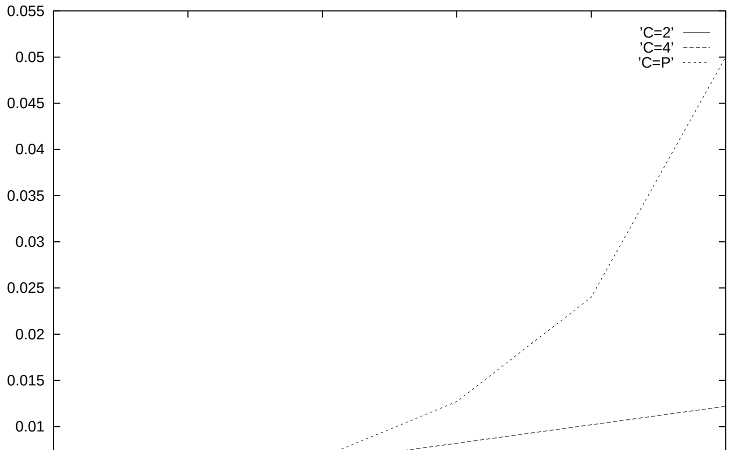


Рис. 6. Времена работы (в секундах) трех типов параллельных

реализаций алгоритмов суммирования на компьютере МВС 100 *для* $M = 1$ *в зависимости от числа процессоров, вовлеченных в решение задачи*

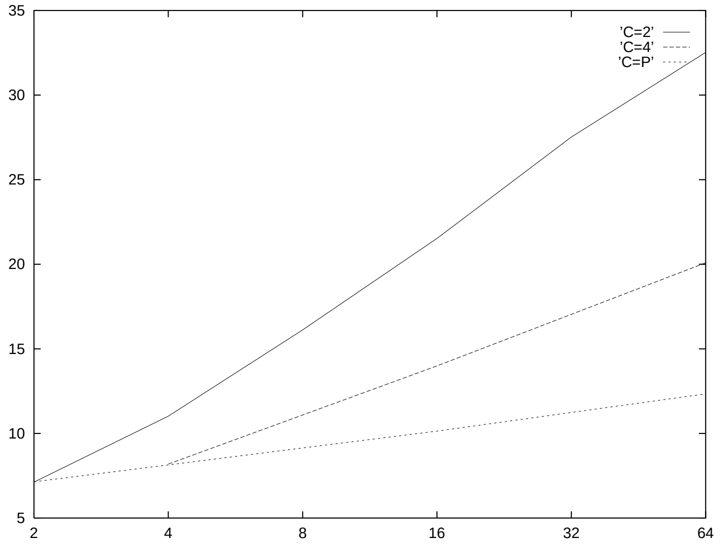


Рис. 7. Времена работы (в секундах) трех типов параллельных реализаций алгоритмов суммирования на компьютере МВС 100 *для максимально возможного числа* $M = 512 \times 1024$ *в зависимости от числа процессоров, вовлеченных в решение задачи*

Теперь покажем масштабируемость параллельных реализаций алгоритмов БПФ на CRAY T3D (рис. 8) и алгоритма суммирования на МВС 100 (рис. 9). Для каждого алгоритма был выбран минимально возможный размер задачи, на котором наблюдается масштабируемость этих алгоритмов. Так, алгоритм БПФ на CRAY T3D масштабируем уже на размере задачи $N = 1024$, в то же время алгоритм суммирования на МВС 100 масштабируем только начиная с $N = 32 \times 1024$. Это связано с различными характеристиками межпроцессорной сети.

В заключение хотелось бы дать еще одно объяснение супермасштабируемости. Заметим, что это явление наблюдается только при асинхронных передачах данных между процессорами.

Пусть имеется P процессоров и у каждого из них по D связей с другими. Если алгоритм полностью использует все D связей для передачи данных, то время обменов будет обратно пропорционально величине D. Увеличим число процессоров в два раза. Если D при этом увеличилось, то время обменов данными между процессорами уменьшится более чем в два раза, а так как та часть алгоритма, которая не содержит обменов, ускорится в два раза, то суммарно алгоритм ускорится больше чем в два раза. Этот же вывод можно сделать на основе внимательного рассмотрения формул в табл. 1 и 2.

Итак, эти результаты показывают, что экспериментальные данные хорошо согласуются с теоретически предсказанными.

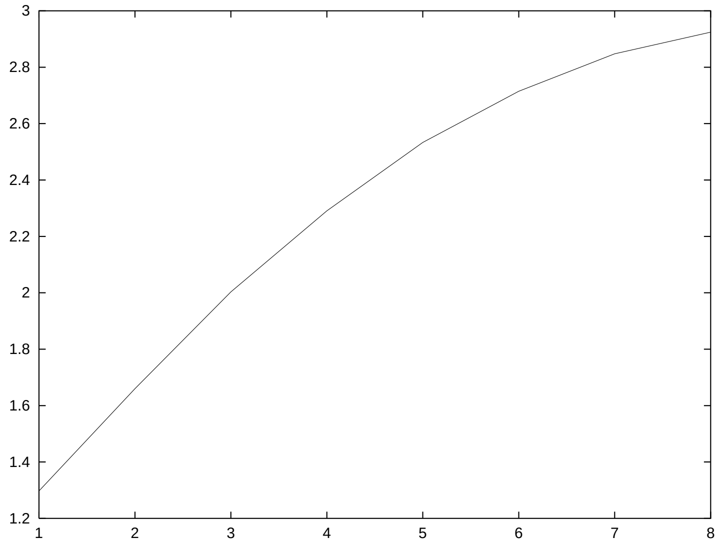


Рис. 8. Зависимость наблюдаемого числа Mflop/s от числа процессоров, вовлеченных в решение задачи для параллельной реализации БПФ на компьютере CRAY T3D *для* $N = 1024$ *(Mflop/s даны в логарифмической шкале по основанию 10)*

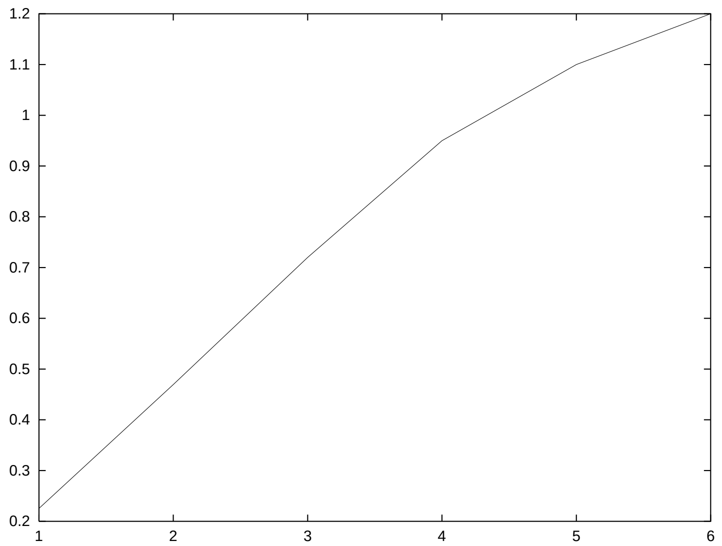


Рис. 9. Зависимость наблюдаемого числа Mflop/s от числа процессоров, вовлеченных в решение задачи для параллельной реализации суммирования на компьютере МВС 100 *для* $M = 32 \times 1024$ *(Mflop/s даны в логарифмической шкале по основанию 10)*

Литература

1. Воеводин В. В., Тыртышников Е. Е. *Вычислительные процессы с тёплицевыми матрицами.* — М.: Наука, 1987.

2. Родрига Г. *Параллельные вычисления* / Пер. с англ. — М.: Наука, 1986.

3. Ибрагимов И. В. Исследование алгоритмов выполнения глобальных операций на массивно-параллельных компьютерах // *Научная апробация слушателей ВКШ / Под ред. В. М. Репина, Вл. В. Воеводина.* — М.: 1995, МГУ, с. 39–54.

4. Agarval R. C., Gustavson F. G., Zubair M. A high-performance matrix-multiplication algorithm on a distributed-memory parallel computer, using overlapped communication //*IBM J. Res. Dev.* 1994. V. 38(6). P. 673–681.

5. Zang X., Yan Y., He K. Latency Metric: An Experimental Method for Measuring and Evaluation Parallel Program and Architecture Scalability //*J. Par. Distr. Comp.* 1994. V. 22. P. 392–410.

6. Horiguchi S., Mirankler W. L. A parallel algorithm for finding the maximum value //*J. Par. Distr. Comp.* 1989. V. 10. P. 101–108.

Новый подход к решению проблемы обобщенного сингулярного разложения*

И. В. Ибрагимов

Аннотация

В работе изучается проблема нахождения обобщенного сингулярного разложения: для матриц $A_1, \ldots, A_k \in \mathbb{R}^{m\times n}$ *необходимо найти такие* $B \in \mathbb{R}^{m\times r}$, $C \in \mathbb{R}^{r\times n}$ *и диагональные* $D_1, \ldots, D_k \in \mathbb{R}^{r\times r}$, *что* $\min\limits_{B,C,D_1,\ldots,D_k} \sum\limits_{i=1}^{k} \|A_i - BD_iC\|_F^2$.

Показано, что такая задача редуцируется с сохранением нормы к нелинейной минимизационной проблеме с $2r^2$ *неизвестными, что для большинства приложений значительно меньше, чем число неизвестных в исходной задаче. Для химических проблем, содержащих такую задачу, обсуждаются методы сглаживания результатов на основе свойств постановки задачи.*

Пусть для $k > 1$ необходимо найти

$$\min_{B,C,D_1,\ldots,D_k} \sum_{i=1}^{k} \|A_i - BD_iC\|_F^2, \tag{1}$$

где $B, C, D_1, \ldots, D_k$ — неизвестные, $B \in \mathbb{R}^{m\times r}$ содержат нормированные столбцы, $C \in \mathbb{R}^{r\times n}$ содержат нормированные строки, $D_1, \ldots, D_k \in \mathbb{R}^{r\times r}$ являются диагональными матрицами.

Эта задача для $k = 2$ рассматривалась в работах [1-3] для решения статистических задач. Проблема $k = 2$ была поставлена [4] при решении класса задач, которые возникают в люминесцентной спектроскопии. Каждая исходная матрица A_i соответствует спектру люминесценции i-й смеси неизвестных веществ. Каждый

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155) и фонда Volkswagen-Stiftung (грант № VW I/71 493).

столбец матрицы A_i соответствует определенной длине волны испускания, при котором записан этот спектр, а каждая строка матрицы A_i соответствует определенной длине волны возбуждения, то есть элементы матрицы — это величины интенсивности люминесценции. Тогда матрица B содержит столбцы, которые соответствуют спектрам испускания чистых веществ, C содержит строки, которые соответствуют спектрам возбуждения чистых веществ, а диагональные элементы матриц D_i содержат относительные концентрации чистых веществ в i-й смеси. На основании такой постановки:

1) спектры флуоресценции должны быть достаточно "гладкими", следовательно, столбцы B и строки C должны быть достаточно "гладкими";

2) значения элементов $B, C, D_1, \ldots, D_k$ должны быть неотрицательными;

3) $r > 0$ — число независимых компонент в смеси — в общем случае неизвестно;

4) r обычно не больше 10, в то же время n и/или m обычно больше 1000.

Для того чтобы учесть условие 1, в этой работе предлагается использовать регуляризатор на основе оператора Лапласа [5]. Пусть $B = PF$, $P \in \mathbb{R}^{m\times r}$, $F \in \mathbb{R}^{r\times r}$ и $C = GQ^*$, $Q \in \mathbb{R}^{n\times r}$, $G \in \mathbb{R}^{r\times r}$, где P и Q состоят из ортонормированных столбцов. Потребуем "гладкости" столбцов матриц P и Q. Так как r не более 10, то это означает, что столбцы B и строки C будут тоже "гладкими". Тогда проблема (1) преобразуется в

$$\begin{aligned} \min_{B,C,D_1,\ldots,D_k} \sum_{i=1}^{k} \|A_i - PFD_iGQ^*\|_F^2 + \\ \alpha \operatorname{tr}(P^*T_BP) + \beta \operatorname{tr}(Q^*T_CQ). \end{aligned} \tag{2}$$

Здесь α и β — сглаживающие параметры, которые необходимо задать до решения проблемы (2), а $T_B \in \mathbb{R}^{n\times n}$ и $T_C \in \mathbb{R}^{m\times m}$ —

матричные аналоги оператора Лапласа вида

$$\begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & 0 & \\ & -1 & 2 & & & \\ & & & & & \\ & & & 2 & -1 & \\ & 0 & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix}.$$

Для того чтобы удовлетворить условиям 2 и 3 постановки задачи, предлагается решать задачу (2) последовательно для $r = 1, 2, \ldots$ до тех пор, пока значения всех элементов матриц решения B, C, $D_1, \ldots, D_k$ будут неотрицательными. Это чисто эмпирический критерий, но он почти всегда предсказывает точные значения r (см. раздел "Численные эксперименты").

Теперь рассмотрим условие 4 постановки задачи. Проблема (1) — это нелинейная оптимизационная проблема с $r(n+m+k)$ неизвестными, и в большинстве случаев число неизвестных достигает тысячи. В настоящее время эту проблему решали квазиньютоновскими методами со специальным выбором начального приближения [6]. Отмечалось, что сходимость достигалась далеко не всегда, а время вычисления достигало нескольких десятков часов на рабочих станциях типа UltraSPARC.

В этой работе предлагается принципиально иной подход к решению этой задачи, а также и новой задачи (2). Предлагается преобразовать задачу (2) в новую задачу с $2r^2$ неизвестными, полностью эквивалентную исходной. Вначале вычислим ортогональные матрицы P и Q, а потом матрицы F, G и $D_1, \ldots, D_k$.

Теорема. Если $A_1, \ldots, A_k \in \mathbb{R}^{m\times n}$, $P \in \mathbb{R}^{m\times r}$, $\tilde{A}_1, \ldots, \tilde{A}_k \in \mathbb{R}^{r\times n}$ при $r \leqslant n$ и $r \leqslant m$, то

$$\min_{P, \tilde{A}_1, \ldots, \tilde{A}_k} \sum_{i=1}^{k} \|A_i - P\tilde{A}_i\|_F^2 + \alpha \operatorname{tr}(P^* T_B P) \tag{3}$$

достигается при условии, что P — матрица, которая состоит из r собственных векторов, соответствующих максимальным собственным значениям матрицы

$$\sum_{i=1}^{k} A_i A_i^* - \alpha T_B, \tag{4}$$

и $\tilde{A}_i = P^* A_i$, $i = 1, \dots, k$.

Доказательство. Пусть P вычислено, тогда

$$\min_{\tilde{A}_1,\dots,\tilde{A}_k} \sum_{i=1}^{k} \|A_i - P\tilde{A}_i\|_F^2 =$$

$$\sum_{i=1}^{k} \left(\|A_i\|_F^2 - \|P^* A_i\|_F^2\right) + \min_{\tilde{A}_1,\dots,\tilde{A}_k} \sum_{i=1}^{k} \|P^* A_i - \tilde{A}_i\|_F^2$$

из-за того, что P ортогональная и $m \leqslant n$. Тогда $\tilde{A}_i = P^* A_i$, $i = 1, \dots, k$. Подставим в (3)

$$\min_P \sum_{i=1}^{k} \|A_i - PP^* A_i\|_F^2 + \alpha \operatorname{tr}(P^* T_B P) =$$

$$\min_P \sum_{i=1}^{k} \left(\|A_i\|_F^2 - \|P^* A_i\|_F^2\right) + \alpha \operatorname{tr}(P^* T_B P) =$$

$$\sum_{i=1}^{k} \|A_i\|_F^2 + \max_P \operatorname{tr}\left(P^* \left\{\sum_{i=1}^{k} A_i A_i^* - \alpha T_B\right\} P\right),$$

следовательно, P — матрица, которая состоит из r собственных векторов, соответствующих максимальным собственным значениям матрицы (4). ∎

На основе этой теоремы вычисляются матрицы P и Q, а исходная задача (2) преобразуется в

$$\min_{F,G,D_1,\dots,D_k} \sum_{i=1}^{k} \|H_i - FD_i G\|_F^2 \tag{5}$$

с квадратными матрицами размера $r \times r$ (здесь $H_i = P^* A_i Q^*$, $i = 1, \dots, k$). Число неизвестных в задаче (5) невелико $r(2r + k)$, и в большинстве случаев эта величина не превосходит 100, поэтому предлагается использовать какой-нибудь итерационный минимизационный метод.

Для большинства итерационных алгоритмов необходимо, чтобы порядки значений аргументов функции в точке минимума были сбалансированы. Если для задачи (5) положить, что все столбцы матрицы F и строки матрицы G нормированы, то значения на диагоналях матриц $D_1, \dots, D_k$ могут на несколько порядков отличаться от элементов матриц F и G.

Пусть F, G известны, тогда $D_1, \dots, D_k$ вычисляются непосредственно из формулы

$$d_i = W^{-1}\nu_i, \tag{6}$$

где $d_i = \operatorname{diag}(D_i)$, — вектор, составленный из диагональных элементов матрицы D_i, $\nu_i = \operatorname{diag}(F^*H_iG^*)$, $W = (F^*F) \odot (GG^*)$, где $\odot$ — операция поэлементного произведения двух матриц.

Итак, если использовать выражение (6) в задаче (5), то среди аргументов по которым выполняется минимизация, не будет значений диагональных элементов матриц D_i, которые по порядку значительно отличаются от значений элементов матриц F и G. Это, в большинстве случаев, улучшает сходимость итерационных алгоритмов минимизации [8]. Тогда

$$E = \sum_{i=1}^{k} \|H_i - FD_iG\|_F^2 = \sum_{i=1}^{k} (\nu_i, W^{-1}\nu_i). \tag{7}$$

W и $\nu_1, \dots, \nu_k$ зависят только от значений элементов матриц F, G. Всего таких неизвестных немного меньше — только $2r^2$.

Формула (7) неудобна для численного решения, так как во время итераций может быть выбрано такое приближение, при котором W будет близка к вырожденной, тогда возможна значительная потеря точности. Во избежание этого предлагается следующий подход.

Пусть $Y = (\hat{g}_1F^*, \dots, \hat{g}_rF^*) \in \mathbb{R}^{r\times r^2}$, где $\hat{g}_1, \dots, \hat{g}_r$ — диагональные матрицы, составленные из столбцов матрицы G,
$\tau_i = (t_{i1}^*, \dots, t_{ir}^*)^*$, где t_{ij} — j-й столбец матрицы H_i. Пусть для Y^* выполнено разложение Грама–Шмидта [7]: $Y^* = SR$, $S \in \mathbb{R}^{r^2\times r}$ — ортогональная, $R \in \mathbb{R}^{r\times r}$ — верхняя треугольная. Тогда $W = YY^* = R^*S^*SR = R^*R$, $\nu_i = Y\tau_i = R^*S^*\tau_i$ и

$$E = \sum_{i=1}^{k} (R^*S^*\tau_i, (R^*R)^{-1}R^*S^*\tau_i) = \sum_{i=1}^{k} \|S^*\tau_i\|_2^2. \tag{8}$$

То есть для вычисления E для заданных F и G необходимо построить Y^*, найти ортонормированное пространство векторов S из Y^* и вычислить E по формуле (8).

Арифметические затраты на вычисление E для заданных F и G составляют порядка r^4 операций (так как необходимо выполнить алгоритм Грама–Шмидта для матрицы из $\mathbb{R}^{r^2 \times r}$).

Если использовать формулу (8), то в явном виде выражение для градиента выписать не удается, поэтому для вычисления градиента предлагается использовать формулу Бауэра–Штрассена [7], позволяющую считать аналитически градиент за тот же порядок арифметических операций. Минимизацию проводили методом Бройдена, так как вычислять гессиан на каждом шаге не представлялось возможным.

Вычисление начального приближения. Важной проблемой минимизации является выбор начального приближения. Если $k = 2$ и F, D_1, H_1 невырождены, то можно решить исходную проблему (1) точно [1-3]:

$$\begin{cases} H_1 = FD_1G, \\ H_2 = FD_2G, \end{cases} \Leftrightarrow \begin{cases} G = D_1^{-1}F^{-1}H_1, \\ H_2H_1^{-1}F = FD_2D_1^{-1}. \end{cases} \tag{9}$$

Используем этот результат и для $k > 2$. Заметим, что (8) переписывается в виде

$$E = \sum_{i=1}^{k} ||S^*\tau_i\tau_i^*S||_F^2 = ||S^*\Psi S||_F^2, \qquad \Psi = \sum_{i=1}^{k} \tau_i\tau_i^*.$$

Приблизим матрицу Ψ матрицей ранга 2 вида $\alpha\psi\psi^* + \beta\varphi\varphi^*$ ($||\psi||_2 = 1$ и $||\varphi||_2 = 1$). Тогда для пары ψ и φ можно применить формулу (9), и в точной арифметике для такого начального приближения значение E будет

$$E_0 = ||S^*(\Psi - \alpha\psi\psi^* - \beta\varphi\varphi^*)S||_F^2. \tag{10}$$

Пусть UZV — сингулярное разложение матрицы $(\tau_1, \ldots, \tau_k)$. Тогда ψ и φ — это первые два вектора матрицы U. Это следует из того, что S — ортогональная. Тогда $E_0 = \sum_{i=3}^{k} z_i$. Следовательно, если Ψ хорошо приближается матрицей ранга 2, то такое начальное приближение обеспечит хорошую начальную невязку. С точки зрения химии это означает, что исходные данные $A_1, \ldots, A_k$ хорошо коррелируют и величина ошибок эксперимента незначительна.

Если фактический ранг матриц $D_1, \ldots, D_k$ оказался меньше r, то использование формулы (9) может привести к неверным результатам. В этом случае предлагается использовать другой алгоритм выбора начального приближения.

Пусть задача (1) решена для $r-1$ и получены матрицы B_{r-1} и C_{r-1}. Вычислим начальное приближение для матриц F и G в виде

$$F = P^*(B_{r-1}, 0), \qquad G = \begin{pmatrix} C_{r-1} \\ 0 \end{pmatrix} Q.$$

Тогда начальная невязка будет не выше невязки, полученной при решении задачи с рангом $r-1$. Такой алгоритм выбора начального приближения применим при значительно больших r, когда невязка исходной задачи (1) при переходе от $r-1$ к r изменяется незначительно.

Итак, арифметические затраты алгоритма по порядку равны $nmkr$ операциям не считая решение нелинейной задачи (5). Арифметические затраты одной итерации метода Бройдена составляют порядка r^4 операций. Затраты алгоритма по памяти — $nmk + (n + m)r + r^4$ чисел. Далее продемонстрируем работу этих алгоритмов численными экспериментами.

Численные эксперименты. Для того чтобы показать эффективность этой теории, был разработан программный комплекс по обработке спектроскопических данных. Алгоритм для большинства задач требует не более 4 Мбайт оперативной памяти и на компьютерах класса Pentium 166 выполняется не более 20 секунд.

В качестве исходных были взяты данные химических экспериментов, которые проводились в лаборатории группы из Польши [6] и доступны в электронном виде для общего пользования. Эти данные содержат 7 матриц размера 301×33 и соответствуют спектрам антрацена и метилантрацена. Спектры чистых веществ также доступны для сравнения.

Первый эксперимент был проведен только с этими данными. Были предсказаны правильно два компонента в смесях и получены чистые спектры этих компонент (рис. 1). Относительная погрешность в предсказании чистых спектров не превышала 1%. Время вычисления составляло 14 секунд.

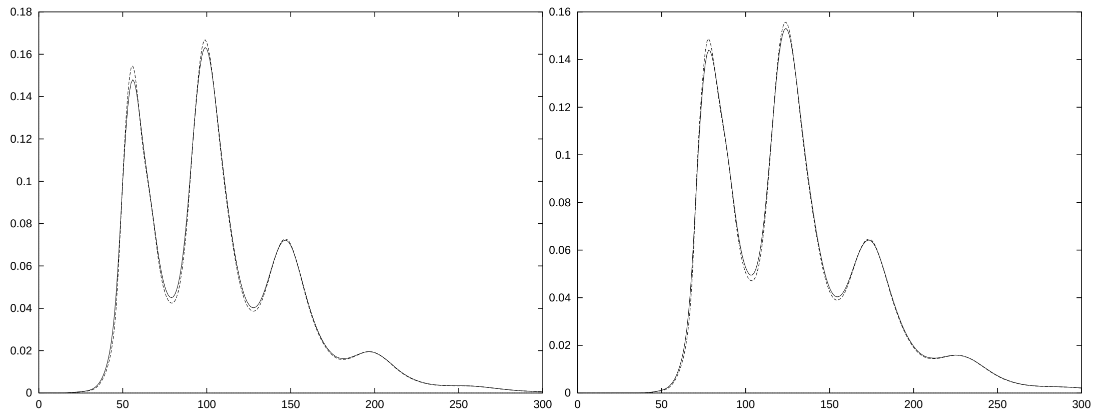


Рис. 1. Сравнение спектров люминесценции антрацена (слева) и метилантрацена (справа), полученных на основе обработки экспериментальных данных (сплошная линия) с известными (пунктирная линия) [6]

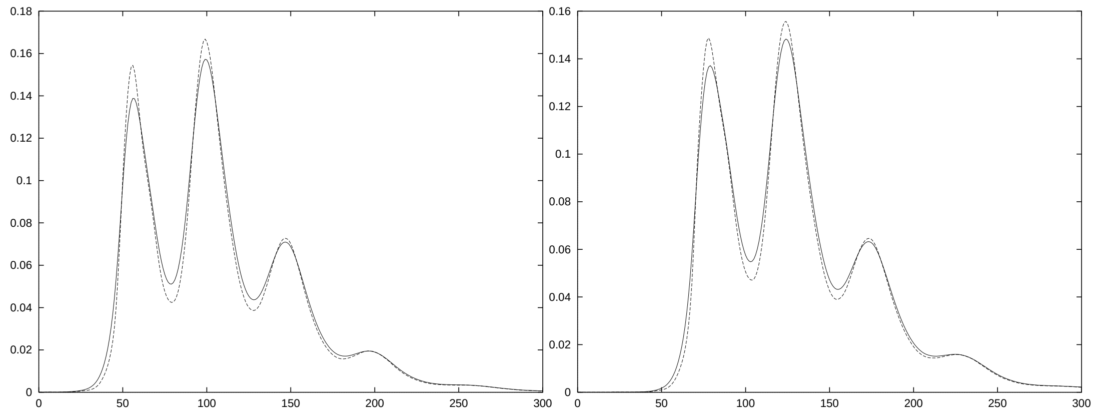


Рис. 2. Сравнение спектров люминесценции антрацена (слева) и метилантрацена (справа), полученных на основе обработки симулированных данных (сплошная линия), с известными (пунктирная линия) [6]

Был проведен еще один численный эксперимент. Были взяты те же исходные данные, и в них был добавлен случайный шум порядка 1–20 % от значений данных. Результат был подан на вход этого алгоритма. Получено, что число компонент предсказывается во всех случаях верно, а относительная погрешность в предсказании чистых спектров незначительно увеличилась и составила 6 % для случая, когда шум достигал 20 % (рис. 2).

Итак, предложенный алгоритм позволяет точно и эффективно обрабатывать данные химического эксперимента.

Литература

1. Bai Z., Zha H. A new preprocessing algorithm for the computation of the generalized singular value decomposition // *SIAM J. Sci. Comp.* 1993. V. 14. P. 1007–1012.

2. Bai Z., Demmel J. W. Computing the generalized singular value decomposition // *SIAM J. Sci. Comp.* 1993. V. 14. P. 1464–1486.

3. Paige C. Computing the generalized singular value decomposition // *SIAM J. Sci. Stat. Comput.* 1986. V. 7. P. 1126–1146.

4. Kubista M. A New Method for the Analysis of Correlated Data Using Procrustes Rotation which is Suitable for Spectral Analysis // *Chemom. Int. Lab. Sys.* 1990. V. 7. P. 273–279.

5. De Boor K. *A practical guide to splines* / Appl. Math. Sci. — Springer–Verlag. 1978. V. 27.

6. Liwo A., Skurski P., Oldziej S., Lankiewicz L., Malicka J., Wiczk W. A new approach to the resolution of the excitation–emission spectra of multicomponent systems // *Comput. Chem.* 1997. V. 21(2). P. 89–96.

7. Tyrtyshnikov E. *A Brief Introduction to Numerical Analysis.* — Birkhäuser, 1997.

8. Dennis J., Schnabel R. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* — Prentice-Hall, 1983.

9. Baur W., Strassen V. The complexity of partial derivatives // *Theor. Comput. Sci.* 1983. V. 22. P. 317–330.

Спектральная проблема для семейства ленточных теплицевых матриц*

А. А. Налват

Аннотация

Обсуждается алгоритм вычисления собственных значений ленточных теплицевых матриц, позволяющий вычислять одно собственное значение за $O(k \log_2 k \log_2^2 n)$ *арифметических действий, где* n *— порядок матрицы, а* k *— ширина ленты. Алгоритм был реализован на языке Matlab, и численные эксперименты были проведены на тестовых матрицах с шириной ленты* 3 *вплоть до размера* $2^{18} \times 2^{18}$.

В статье [1] авторы Bini и Pan предложили три алгоритма вычисления собственных значений ленточных теплицевых матриц. Каждый из этих алгоритмов представляет собой итерационный метод Ньютона, с помощью которого находится корень характеристического полинома $p(\lambda) = \det(A - \lambda I)$, который является собственным значением матрицы A. Авторы предлагают использовать этот алгоритм для вычисления действительных собственных значений матрицы A, элементы которой также являются действительными числами.

Если размер матрицы равен $n \times n$ и полуширина ленты равна k, то данный алгоритм позволяет вычислить $p(\lambda^{(i)})$ и $p'(\lambda^{(i)})$ за $O(k \log_2 k \log_2 n)$ действий. Точнее, такого числа математических действий требует один шаг Ньютона, так как $\lambda^{(i+1)}$ вычисляется по формуле $\lambda^{(i+1)} = \lambda^{(i)} - \frac{p(\lambda^{(i)})}{p'(\lambda^{(i)})}$.

Исследуем сходимость этого метода, обратив особое внимание на вид функции $p(\lambda)$. Рассмотрим простейший пример, для которого, заметим, собственные значения вычислены аналитически.

*Работа выполнена при поддержке РФФИ (грант № 97-01-00155).

На рис. 1 приведен график функции $p(\lambda)$ для трехдиагональной матрицы размера 32×32, у которой $a_{-1} = a_1 = 1$. Видно, что $P(\lambda)$ осциллирует на отрезке $[-2, 2]$, обращаясь в ноль 32 раза. При решении практических задач нас, конечно же, интересуют матрицы бóльших размеров n. В данном случае $p(\lambda)$ будет иметь n нулей на отрезке $[-2, 2]$, как следует из теоремы о кругах Гершгорина (см. ниже). Причем при $|\lambda| \gg 2$ $p(\lambda)$ ведет себя, как λ^n. Заметим, что уже $p(2.5) = 10^9$. То есть если в качестве начального приближения выбрано $|\lambda^{(0)}| \gg 2$ или на некотором шаге i мы получим значение $|\lambda^{(i)}| \gg 2$, то метод Ньютона сходится медленно. Действительно, если $p(\lambda) = \lambda^n$, то $\lambda^{(i+1)} = (1 - \frac{1}{n})\lambda^{(i)}$.

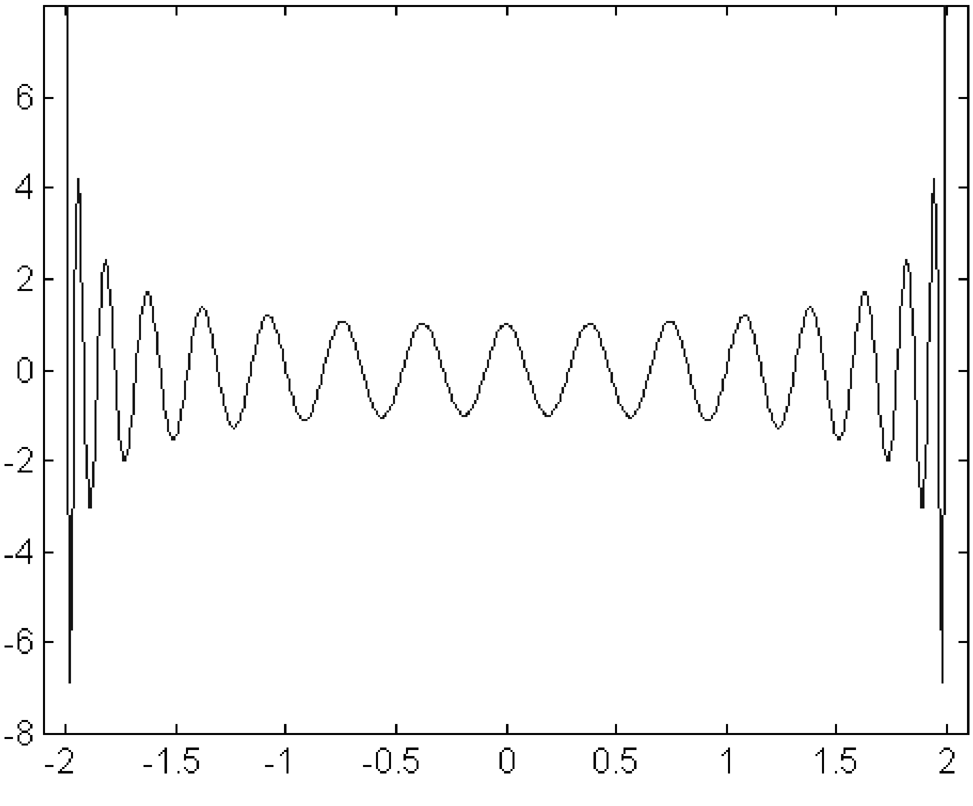


Рис. 1

Из графика на рис. 1 видно, что если на некотором шаге $\lambda^{(i)}$ будет находиться между двумя собственными значениями на примерно одинаковом расстоянии от них, то есть достаточно близко к собственному значению, то уже на следующем шаге $|\lambda^{(i+1)}|$ будет много больше 2.

Наблюдения распределений собственных значений различных ленточных теплицевых матриц (см. также [3]) показали, что даже при увеличении размера матрицы ее собственные значения располагаются не хаотично внутри объединения кругов Гершгорина, которое, заметим, остается постоянным при увеличении размера матрицы, а вдоль некоторых кривых.

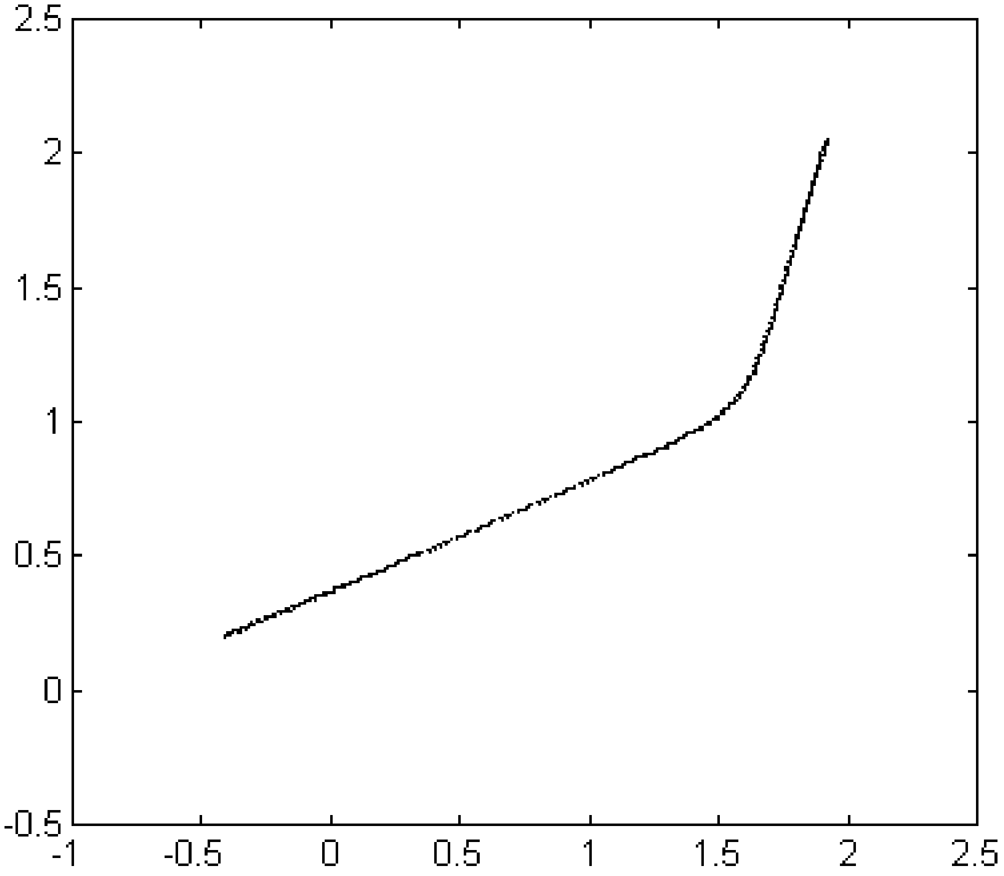


Рис. 2

На рис. 2 и 3 изображены собственные значения четырехдиагональных теплицевых матриц для $a_0 = 1 + i$, $a_1 = 0.8$, $a_{-1} = 0.2 + 0.6i$, $a_2 = 0.4i$ размера 200×200 и 7×7. Здесь мы имеем дело с „кратными“ корнями характеристического многочлена. Поэтому следует использовать аналог метода Ньютона, который бы учитывал наличие „кратных“ корней. Можно предложить следующий алгоритм вычисления $\lambda^{(i+1)}$.

Для начала заметим, что вычислять $\lambda^{(i+1)}$ на удалении от корней $p(\lambda)$ можно также по формуле $\lambda^{(i+1)} = \lambda^{(i)} - n\frac{p(\lambda^{(i)})}{p'(\lambda^{(i)})}$. Для полученного $\lambda^{(i)}$ вычислим пробное приближение $\lambda^{(i+1)}_{\text{пр}}$ по формуле $\lambda^{(i+1)}_{\text{пр}} = \lambda^{(i)} - step\,\frac{p(\lambda^{(i)})}{p'(\lambda^{(i)})}$. На первом шаге полагаем $step = n$, то есть используем метод Ньютона для кратных корней. А далее учтем необходимость уменьшения нормы невязки, то есть $|p(\lambda)|$. И если невязка уменьшается, то есть $|p(\lambda_{\text{пр}}^{(i+1)})| < |p(\lambda^{(i)})|$, то можно положить $\lambda^{(i+1)} = \lambda_{\text{пр}}^{(i+1)}$ и перейти к следующему шагу. Если же невязка для пробного приближения больше, то можно положить $step = 1$, то есть начать использовать простой метод Ньютона.

Можно также предложить изменять шаг от n до 1 постепенно, например уменьшать в два раза всякий раз при увеличении невязки, то есть когда $|p(\lambda_{\text{пр}}^{(i+1)})| > |p(\lambda^{(i)})|$.

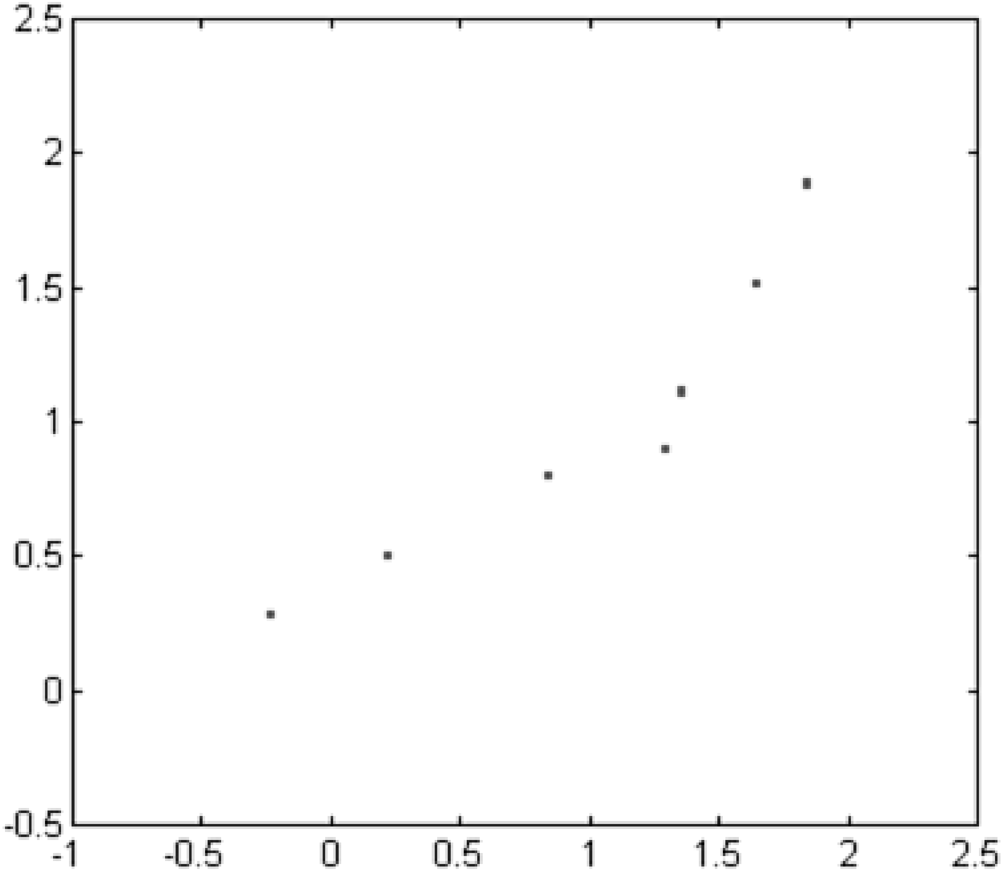


Рис. 3

Аналогичные модифицированные методы Ньютона, учитывающие особенности функции, например такие, как наличие кратных корней, приведены в [2] и [4].

Для нас представляет интерес находить не только действительные, но и комплексные собственные числа, а также собственные значения матриц, элементы которых являются комплексными числами. Для этого используем приведенный алгоритм без изменений.

n	*flops/iters*	*iters*, простой метод	*iters*, модифицированный
8	2 400	17	12
16	2 800	21	16
32	3 200	34	18
64	3 600	64	20
128	3 900	121	24
256	4 300	239	28
2^9	4 700	—	32
2^{10}	5 100	—	35
2^{11}	5 600	—	39
2^{12}	6 200	—	37
2^{13}	6 700	—	40
2^{14}	7 200	—	47
2^{15}	7 800	—	49
2^{16}	8 300	—	54
2^{17}	8 900	—	62
2^{18}	9 400	—	70

В таблице для трехдиагональных матриц ($a_{-1} = 0.4,\quad a_1 = 1,$ $a_2 = 0.6i$) разных размерностей n, для простого метода Ньютона и модифицированного приведены количество итераций (*iters*) и число арифметических действий с плавающей точкой, необходимых для выполнения одной итерации (*flops/iters*). Причем начальное приближение выбрано „очень плохое“, находящееся на границе круга Гершгорина, $\lambda^{(0)} = 1.2 + 1.2i$. Заметим, что под *iters* для модифицированного метода понимается число пробных итераций, то есть вычислений $\lambda_{\text{пр}}$.

Как видно из таблицы, отношение *flops/iters* увеличивается с ростом n, как $\log n$, что соответствует оценке, полученной Bini и Pan. Число итераций растет с ростом n тоже как $\log n$.

Литература

1. Bini D., Pan V. On the evaluation of the eigenvalues of a banded toeplitz block matrix // *Journal of complexity.* 1991. V. 7. P. 408–424.

2. Федоренко Р. П. *Введение в вычислительную физику.* — М.: Изд-во МФТИ, 1994.

3. Замарашкин Н. Л. *Матричный признак равнораспределенности и вычисление спектральных распределений структурированных матриц.* Дис. на соиск. ученой степени к.ф.-м.н., 1997. — М.: ИВМ РАН.

4. Воеводин В. В. *Численные методы алгебры (теория и алгорифмы).* — М.: Наука, 1966.

СОДЕРЖАНИЕ

Научное издание

МАТРИЧНЫЕ МЕТОДЫ И ВЫЧИСЛЕНИЯ

Сборник научных трудов
под редакцией Е. Е. Тыртышникова

Утверждено к печати Ученым советом ИВМ РАН

ЛР №021026 от 04.01.96.
Институт вычислительной математики
Российской академии наук
117951 Москва, ул. Губкина, д. 8.
Оригинал-макет изготовлен в ИВМ РАН

Подписано в печать 31 мая 1999 г.
Формат $60 \times 90^1/_{16}$. Печать офсетная.
Уч.-изд. л. 13. Тираж 100 экз. Заказ
Отпечатано в Производственно-издательском комбинате ВИНИТИ.
140010 г. Люберцы Московской обл., Октябрьский пр-т, 403.
Тел. 554-21-86