

## Практическое ускорение численного решения трехмерной скалярной задачи дифракции методом дискретных особенностей

А. Ю. Анфиногенов

*ВВИА им. проф. Н.Е.Жуковского*

This paper covers practical results in the area of numerical evaluation of some two-dimensional diffraction theory regular integrals and integrals with weak singularity produced by discrete singularity method. Analytical formula for integration over triangle is offered. Some cubature formulae for regular integrand function over uniform triangular and quadrangular meshes are offered. Several non-uniform Gaussian-like cubature formulae over triangles are surveyed. Numerical efficiency of above mentioned cubature formulae was estimated and compared by series of numerical experiments.

### 1. Введение

Многие пространственные задачи дифракции акустических, электромагнитных волн и теории антенн сводятся к сингулярным или гиперсингулярным интегральным уравнениям. Даже если исходная задача дифракции (например,  $E$ -поляризованной электромагнитной волны) сводится к интегральному уравнению первого рода с логарифмической особенностью, то все равно удобнее от него перейти к сингулярному интегральному уравнению первого рода. Решение этих сингулярных или гиперсингулярных интегральных уравнений выполняется методом дискретных особенностей (МДО), неотъемлемым элементом которого является формирование системы линейных алгебраических уравнений (СЛАУ) с последующим решением этой системы [1].

Если обозначить отношение диаметра объекта к длине волны, как  $A = D/\lambda$ , то для получения адекватных результатов решения задачи дифракции следует взять не менее  $5A$  точек по каждой координате. Это означает, что число операций по формированию матрицы СЛАУ будет не менее  $625A^4N_I$ , где  $N_I$  – число операций для вычисления каждого коэффициентов. Как следствие, количество операций  $N_I$  и погрешность  $A_I$  вычисления коэффициентов СЛАУ оказывают сильное влияние на границы применимости МДО в дифракционных задачах, а их минимизация позволит при тех же вычислительных ресурсах решать более широкий круг практических задач.

Методы снижения вычислительных затрат можно разделить на две группы – алгоритмические и оптимизационные, или машинно-зависимые. На примере численного решения трехмерной скалярной задачи дифракции рассмотрим практическое применение этих методов для ускорения МДО.

### 2. Численное решение трехмерной скалярной задачи дифракции с помощью МДО

В качестве примера рассмотрим решение скалярной трехмерной задачи

дифракции на абсолютно жестком теле с кусочно-гладкой границей  $\sigma$ . Эта задача сводится к внешней задаче Неймана для уравнения Гельмгольца:

$$\nabla^2 p(M) + k^2 p(M) = 0, \quad M \notin \sigma, \quad (1)$$

$$\frac{\partial p(M_0)}{\partial n_{M_0}} = -\frac{\partial p^I(M_0)}{\partial n_{M_0}}, \quad M_0 \in \sigma, \quad (2)$$

где  $p(M)$  – звуковое давление в точке наблюдения,  $p^I(M_0)$  – поле падающей волны,  $k$  – волновое число. Функция  $p(M)$  удовлетворяет также условию излучения на бесконечности.

При помощи потенциала двойного слоя эта задача сводится к гиперсингулярному интегральному уравнению [1]:

$$\frac{1}{4\pi} \frac{\partial}{\partial \vec{n}_{M_0}} \int_{\sigma} \frac{\partial}{\partial \vec{n}_M} \left( \frac{e^{ikr_{MM_0}}}{r_{MM_0}} \right) \cdot g(M) ds_M = f(M_0), \quad M, M_0 \in \sigma, \quad (3)$$

где  $\vec{n}_M^o$  и  $\vec{n}_{M_0}^o$  – орты нормалей в соответствующих точках.

После замены поверхности тела  $\sigma$  набором четырехугольных рамок  $\Omega_i$  с границами  $\Gamma_i$ ; замены в пределах каждой рамки плотности потенциала  $g(M)$  на неизвестные константы  $g_i$  и применения квадратурных формул метода дискретных особенностей (МДО) исходная задача сводится к СЛАУ с коэффициентами

$$C_{ij} = \frac{1}{4\pi} \left[ k^2 \int_{\Omega_i} (\vec{n}_{M_{0j}}^o \cdot \vec{n}_M^o) \frac{e^{-ikr_{M_{0j}M}}}{r_{M_{0j}M}} dS_M + \int_{\Gamma_i} \left( \nabla_M \left( \frac{e^{-ikr_{M_{0j}M}}}{r_{M_{0j}M}} \right) \times \vec{dl}_M \right) \cdot \vec{n}_{M_{0j}}^o \right] = \frac{1}{4\pi} [k^2 I_{1ij} + I_{2ij}] \quad (4)$$

где  $M \in \Omega_i$ , точки  $M_{0j}$  расположены в центре рамок [2].

### 3. Основные источники вычислительных затрат и методы их снижения

Основными источниками вычислительных затрат являются заполнение матрицы СЛАУ и решение СЛАУ.

Если поверхность  $\sigma$  не обладает симметрией, то число операций по формированию матрицы СЛАУ будет не менее  $625L^4 N_I$ , где  $N_I$  – число операций для вычисления каждого коэффициента. Если для решения СЛАУ используется метод Гаусса (или эквивалентный ему метод  $LU$ -разложения), то число операций составит  $\frac{2}{3}N^3$ , что применительно к  $L$  это означает порядка

$\frac{2}{3}(5L)^6$  операций.

Два основных метода снижения времени вычислений – алгоритмический (совершенствование численных методов и алгоритмов вычислений) и оптимизационный (оптимизация кода для используемой архитектуры ЭВМ).

К алгоритмическим методам можно отнести использование более эффективное вычисление коэффициентов  $C_{ij}$ .

К оптимизационным методам относятся использование потоковых (SIMD) расширений набора инструкций современных процессоров (например, SSE-SSE4 для x86-совместимых процессоров), конвейеризации, использования предварительной выборки из памяти и эффективное использование многоядерности современных процессоров. Блестящим примером применения оптимизации являются существующие реализации пакетов линейной алгебры BLAS и LAPACK (например, ATLAS или Intel MKL), демонстрирующие производительность, близкую к пиковой производительности процессора и рекомендуемые для решения СЛАУ в МДО.

#### 4. Эффективное вычисление коэффициентов $C_{ij}$ при кубатурных формул

Наибольшие вычислительные затраты в выражении (4) приходятся на интеграл  $I_{1ij}$ , который при  $i = j$  имеет слабую особенность.

Для плоской рамки  $\Omega_j$  орт нормали к рамке  $\vec{n}_M^o$  является постоянным, тогда:

$$I_{1ij} = (\vec{n}_{M_{0j}}^o \cdot \vec{n}_M^o) \left[ \int_{\Omega_i} \frac{1}{r_{M_{0j}M}} dS_M - \int_{\Omega_i} \frac{1 - e^{-ikr_{M_{0j}M}}}{r_{M_{0j}M}} dS_M \right] = (\vec{n}_{M_{0j}}^o \cdot \vec{n}_M^o) [I_{11ij} - I_{12ij}], \quad (5)$$

$$\text{где } I_{11ij} = \int_{\Omega_i} \frac{1}{r_{M_{0j}M}} dS_M, \quad I_{12ij} = \int_{\Omega_i} \frac{1 - e^{-ikr_{M_{0j}M}}}{r_{M_{0j}M}} dS_M. \quad (6)$$

Поскольку интеграл  $I_{11ij}$  вычисляется аналитически [3], остается оптимизировать по скорости вычисление интеграла  $I_{12ij}$  от гладкой функции по плоской трех- или четырехугольной рамке.

Традиционно для вычисления интегралов на четырехугольниках общего вида используются последовательные кубатурные формулы (КФ), являющиеся декартовым произведением одномерных квадратурных формул Ньютона-Котеса

$$I \approx S_{\Omega} \sum_{i=1}^N c_i f(M_i), \quad (7)$$

где  $c_i$  – коэффициенты кубатурной формулы, по определению не зависящие от значений  $f(M_i)$ , но могущие зависеть от области интегрирования  $\Omega$ .

Последовательные кубатурные формулы могут быть также применены и на треугольнике, если отобразить его в квадрат при помощи преобразования Даффи [4]:

$$\int_0^1 \int_0^x f(x, y) dy dx = \int_0^1 \int_0^1 f(x, xt) dt dx, \quad (8)$$



квадратур наивысшей степени точности Гаусса-Кристоффеля на многомерный случай. Всеобъемлющее обобщение квадратур Гаусса-Кристоффеля на многомерный случай отсутствует до сих пор, не смотря на то, что в последние три десятилетия велись активные исследования в этом направлении. Особо следует отметить работы И. П. Мысовских [5] и Р. Кулса [6,7]. Рассмотрим несколько наиболее эффективных кубатурных формул типа Гаусса, предложенных в этих работах.

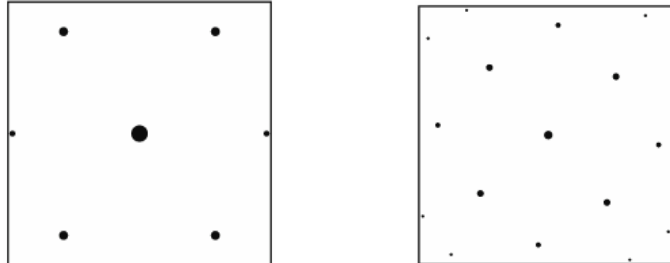


Рис.2. Разбиения для симметричных КФ 7-го и 9-го порядков.

### 5. Результаты вычислительного эксперимента

Для сравнения эффективности КФ была проведена практическая проверка и сравнение точности и вычислительных затрат для различных кубатурных формул, рассмотренных выше.

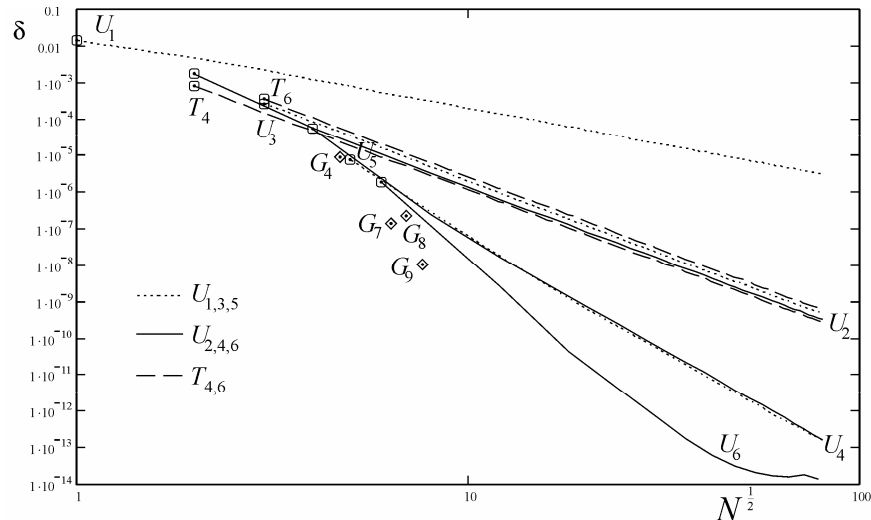


Рис.3. Относительная погрешность вычисления интеграла  $I_{12ij}$  для разных КФ.

На рис. 3 в приведены зависимости относительной погрешности  $\delta$  вычисления интеграла  $I_{12ij}$  для выпуклого плоского неправильного четырехугольника  $M_1M_2M_3M_4$  от величины  $\sqrt{N}$ . Она равна числу отрезков  $n$ , на которые разбивается каждая сторона для выпуклого плоского неправильного четырехугольника для равномерных формул. Точка  $M_0 \notin M_1M_2M_3M_4$  лежит в плоскости четырехугольника. На рис. 3 формулы на равномерной сетке для  $C_{Si}$  обозначены как  $U_i, i = \overline{1,6}$ , а последовательные

формулы на равномерной обозначены, как  $T_4$  и  $T_6$ . Точки  $G_j$  соответствуют кубатурным формулам типа Гаусса. Поскольку сетка для них нерегулярна, то в качестве  $\sqrt{N}$  взят корень квадратный из суммы количества точек внутри области и числа точек на границе, необходимых для вычисления контурного интеграла в (4).

Графики на рис. 3 содержат несколько закономерностей. Во-первых, формулы вида (10) для нечетных порядков имеют почти такую же точность и порядок сходимости, как формулы меньших на единицу четных порядков, поэтому использовать формулы нечетных порядков не имеет смысла. Во-вторых, точность и порядок сходимости последовательных формул приблизительно соответствуют поведению формул (10). В-третьих, формулы типа Гаусса действительно обеспечивают очень хорошую точность и скорость сходимости, но с учетом указанных выше особенностей МДО, уже не являются безоговорочными фаворитами, как в методе конечных элементов. В зависимости от требуемой погрешности конкуренцию в рамках МДО им могут составить формулы (10) 4-го или 6-го порядков.

#### 6. Применение машинно-зависимых методов оптимизации

Суперскалярная архитектура всех массово выпускаемых процессоров и поддержка ими потоковых наборов команд (MMX, SSE1-4, AltiVec и др.) позволяет подвергнуть программную реализацию МДО эффективной оптимизации. Рассмотрим методы оптимизации на примере x86-совместимых процессоров Intel, AMD, VIA и Transmeta. Использование векторных операций и векторно-ориентированных библиотек (ATLAS, VSIPL, Intel MKL и IPP) на стандартных ПК способно вплотную приблизить среднюю производительность к пиковой. Например, на Intel Pentium 4 (P4) использование предвыборки и последовательного доступа к данным позволяет достичь пиковой скорости обмена с DDR-памятью до 3.2 Гбайт/с против 360 Мбайт/с средней скорости при произвольном доступе, а производительность при решении СЛАУ методом Гаусса достигает 12,81 с или 95% от пиковой (против 174,12 с или 7% при использовании того же метода, написанного на Си++ без учета особенностей подсистем процессора и памяти).

Таблица 1. Производительность при решении СЛАУ МДО

Процессор (ГГц)	Память (МГц)	Время, с	Операций за такт
Pentium 4HT (2.8)	2xDDR (800)	2.153 / 12.813	2.99 / 2.97
Pentium 4M (1.7)	1xDDR (533)	3.575 / 23.824	2.95 / 2.63
Celeron 4 (2.0)	1xDDR (400)	3.422 / 27.453	2.58 / 1.94
Pentium 3 (1.13)	1xSDR (133)	10.596 / 67.667	1.46 / 1.39
Pentium M (1.4)	1xDDR (400)	5.498 / 41.429	2.43 / 1.83
Crusoe (1.0)	1xDDR (400)	14.902 / 116.828	1.21 / 0.91

В таблице 1 приведены результаты сравнения производительности различных архитектур при решении действительной СЛАУ размерности 2048 и 4096. Матрицы занимали 64 и 256 Мбайт оперативной памяти, соответственно. Для решения использовалась библиотека ATLAS. Из таблицы следует, что с использованием потоковых библиотек вопрос эффективного решения СЛАУ можно считать решенным.

Остается открытым вопрос о том, в каком виде эффективней решать СЛАУ – в вещественном или комплексном? Для этого была проведена еще серия экспериментов на Р4М с использованием библиотеки ATLAS. Рассмотрим СЛАУ МДО вида

$$Cg = b, \quad (11)$$

где  $C$  – матрица с комплексными коэффициентами  $C_{ij}$  (4),  $g$  и  $b$  – векторы с комплексными коэффициентами. СЛАУ (11) можно решать как в комплексном виде, так и тождественно преобразовав к СЛАУ вдвое большей размерности с вещественными коэффициентами.

$$\begin{bmatrix} C_{re} & -C_{im} \\ C_{im} & C_{re} \end{bmatrix} \begin{bmatrix} g_{re} \\ g_{im} \end{bmatrix} = \begin{bmatrix} b_{re} \\ b_{im} \end{bmatrix}, \quad (12)$$

где индексами  $re, im$  обозначены действительные и комплексные части соответствующих векторов и матриц.

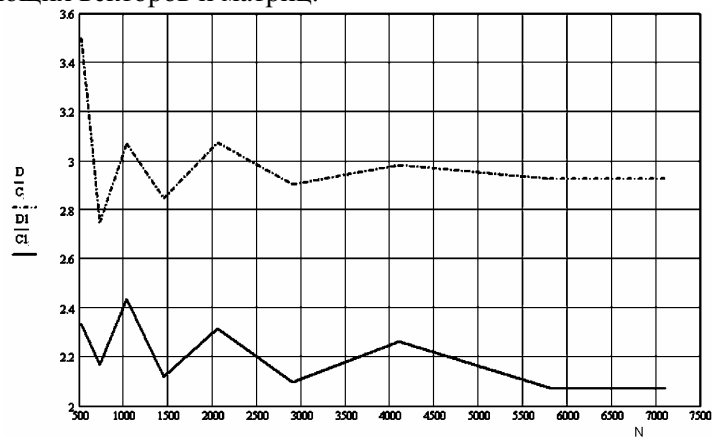


Рис.4. Ускорение решения комплексной СЛАУ для SSE1 и SSE2.

Численный эксперимент показывает парадоксальный, на первый взгляд, результат. На рис. 4 приведена зависимость отношения времен решения вещественной и комплексной СЛАУ от порядка, причем пунктиру соответствует реализация для набора команд SSE1, а сплошной линии – для SSE2. Из графиков видно, что решение СЛАУ в комплексном виде выполняется от 3.5 до 2.1 раз быстрее, чем в вещественном. Это объясняется большей эффективностью работы с компактно размещенными в памяти данными, не смотря на то, что работа с комплексными числами требует больших, по сравнению с вещественными, накладных расходов.

Еще одной машинной зависимостью является зависимость от схемы хранения матрицы и способа формирования системы. Существует два способа хранения матриц – в стиле Си, когда рост адреса  $j$  элемента  $M[i][j]$  ведет к росту номера элемента в строке (т.е.  $M[i]$  – это указатель на  $i$ -ю строку); и в стиле Фортрана, когда рост адреса  $j$  элемента  $M[i][j]$  ведет к росту номера элемента в столбце (т.е.  $M[i]$  – это указатель на  $i$ -й столбец).

Поскольку современные компьютерные архитектуры используют медленную, по отношению к ЦП, оперативную память, то наиболее эффективным является

последовательное обращение к памяти. Поэтому для схемы хранения Си заполнение матрицы СЛАУ целесообразно выполнять по строкам, т.е. фиксировать точку наблюдения  $M_{oj}$ , а дифракционные интегралы вычисляются для рамок  $\Omega_i$ , что, как показано выше, требует многократных вычислений внутренних точек каждой рамки, причем операция плохо векторизуется.

Для схемы хранения стиле Фортрана заполнение матрицы СЛАУ следует выполнять по столбцам, фиксируя  $i$  и увеличивая  $j$ . Это означает, что фиксируется рамка  $\Omega_i$ , а точки наблюдения  $M_{oj}$  изменяются, поэтому все кубатурные суммы и подынтегральные функции вычисляются относительно вектора  $\langle M_{oj} \rangle$ , что позволяет эффективно использовать потоковые операции и преимущества последовательного доступа к памяти. При этом необходимые для интегрирования внутренние точки рамки вычисляются только один раз. Кроме того, такой способ организации вычислений гораздо легче распараллеливается.

В заключение можно отметить, что совместное применение алгоритмических и оптимизационных методов ускорения численного решения задач дифракции методом дискретных особенностей позволило существенно расширить в высокочастотную область диапазон применимости метода, а также дало возможность решать более масштабные и интересные задачи на прежней аппаратной базе.

#### ЛИТЕРАТУРА

1. Лифанов И.К. Метод сингулярных интегральных уравнений и численный эксперимент. – М.:ТОО “Янус”, - 1995. – 520с.
2. Anfinogenov A.Yu., Lifanov I.I. On Numerical Solution Of Integral Equations Of Planar And Spatial Diffraction Problems. – Russ.J.Numer. Anal. Math. Modelling. – 1992. – Vol. 7, №5, pp. 387-404.
3. Анфиногенов А.Ю. К вычислению некоторых интегралов теории дифракции. – Электромагнитные волны и электронные системы.– 2004, №11, с.11-20.
4. Lynes J. N., Cools R. A. Survey of Numerical Cubature over Triangles. – AMS Proc. Symp. Appl. Math. (PSAPOM) of ‘Mathematics of Computation 1943-1993’, Vancouver, 1994.
5. Cools R., Mysovskikh I.P., Schmidt H.J. Cubature Formulae and Orthogonal Polynomials. – Elsevier Preprint, 13 January 2000.
6. Cools R., Rabinowitz P. Monomial cubature rules since ‘Stroud’: A compilation. – J. Comp. Appl. Math., 48 (1993), pp. 309-326.
7. Cools R. Monomial cubature rules since ‘Stroud’: A compilation – part 2. – J. Comp. Appl. Math., 112 (1999), pp. 21-27.