

## Гибкая модель приближенных вычислений ядер двумерных гиперсингулярных операторов и архитектура программной реализации

В. О. Мищенко

*Харьковский национальный университет им. В.Н. Каразина, Украина*

We have developed the concept and a model of dependable software tools for computing cores of pseudo-differential operators, which arise in the actual task of spatial diffraction at the plane-parallel structures. To implement the model we have developed a method of calculation based on the representation by orthogonal polynomials and another method that uses fast Fourier transform. The difference in the properties of these algorithms provides an opportunity to save time of calculations and monitor the accuracy.

### 1. О проблеме

В определённых упрощающих предположениях относительно рассматриваемых областей 3-мерные задачи дифракции электромагнитных волн на экранах произвольной формы могут сводиться к граничным системам псевдодифференциальных операторов методом параметрических представлений псевдодифференциальных и гиперсингулярных интегральных операторов [1]. Это позволяет исследовать модельные [1] и новые [2] задачи дифракции на плоскопараллельных структурах, на цилиндрических структурах [4], а также в скалярных задачах пространственной дифракции на плоскопараллельных структурах [3] и других задачах. Построенные при этом граничные уравнения, которые можно интерпретировать как гиперсингулярные интегральные уравнения (ГСИУ), выгодны для вычислительных экспериментов и прикладных расчётов тем, что допускают прозрачные схемы методов дискретных особенностей [5]. Для 3-мерных задач эти схемы получают обоснование с привлечением вычислительных экспериментов, тогда как 2-мерные задачи опираются на теорию одномерных псевдодифференциальных операторов [6].

Далее, простота математических моделей и прозрачность вычислительных схем не могут сколь угодно сильно снизить вычислительную трудоёмкость объективно сложных задач. Для вышеуказанных классов пространственных задач эта сложность проявляется, в частности, в форме высокой вычислительной трудоёмкости формирования элементов матрицы дискретной модели (порядка  $N^4$  вычислений взаимодействия пар элементов дискретной модели по сравнению с  $N^2$  в 2-мерных задачах). При этом требуется порядка  $N^2$  различных значений функций - ядер псевдодифференциальных операторов - которые являются обобщёнными преобразованиями Фурье определённых функций, зависящих от геометрических особенностей задачи и её волновых чисел.

Статья посвящена разработке подхода к формированию адаптируемых и адаптивных алгоритмов вычислений ядер псевдодифференциальных операторов в форме программных реализаций с требуемыми свойствами и средствами поддержки формальной оценки их качества.

В численных реализациях МДО применялись следующие подходы к борьбе с вычислительной сложностью. Первое - выяснение существенных для упрощения алгоритмов аналитических особенностей искомым ядер (примеры имеются в [1-4]). Второе - ставка на непосредственное ускорение машинных вычислений. То есть, выбор удачных вспомогательных алгоритмов и поиск готовых высокопроизводительных программных средств или компонент, в том числе ориентированных на конкретную вычислительную систему: примеры в [7, 8].

Однако для 3-мерных задач дифракции в кусочно-неоднородных средах особо актуален поиск новых методов преодоления вычислительной трудоёмкости. Отправляясь от необходимости исследования возможностей численного построения ядер гиперсингулярной системы граничных уравнений [2], мы развиваем новые в данном контексте подходы на основе многоверсионности и встраивания тестовых функций.

## 2. Конкретизация прикладной задачи

В [2] была рассмотрена краевая задача для электромагнитного поля

$$\exp(-i\omega t) \cdot (E(x), H(x)) \quad (x = (x', x_3), x' \in R^2, x_3 \in R), \quad (2.1)$$

рассеянного в результате падения плоской волны

$$\exp(-i\omega t) \cdot (E^0(x), H^0(x)), \quad (2.2)$$

на идеально проводящий экран

$$\Sigma = (\sigma, 0) = \langle x \in R^3 \mid x_3 = 0, \sigma \in R^2 \rangle, \quad (2.3)$$

который расположен на поверхности

$$\Pi = (R^2, 0) \quad (2.4)$$

раздела диэлектрических полупространств

$$X_+ = \langle x \in R^3 \mid x_3 > 0 \rangle \quad \text{и} \quad X_- = \langle x \in R^3 \mid x_3 < 0 \rangle. \quad (2.5)$$

Эта задача включает систему дифференциальных уравнений Максвелла, условия на разделе сред, на экране, на бесконечности и на ребре диска:

$$\operatorname{rot} H = -i \varepsilon_{\pm} \varepsilon_0 \omega \cdot E \quad (x \in X_{\pm}, \text{ i.e. } x_3 > 0 \text{ or } x_3 < 0), \quad (2.6)$$

$$\operatorname{rot} E = i \varepsilon_{\pm} \varepsilon_0 \omega \cdot H \quad (x \in X_{\pm}), \quad (2.7)$$

$$\operatorname{div} E = 0, \quad \operatorname{div} H = 0 \quad (x \notin \Pi, \text{ i.e. } x_3 \neq 0) \quad (2.8)$$

$$i = 1, 2 \Rightarrow [E_i]_{-0}^{+0} = E_i(x', +0) - E_i(x', -0) = 0 \quad (x' \in \Pi \setminus \Sigma), \quad (2.9)$$

$$i = 1, 2 \Rightarrow [H_i]_{-0}^{+0} = 0 \quad (x' \in \Pi \setminus \Sigma) \quad (2.10)$$

$$i = 1, 2 \Rightarrow E_i(x', +0) = -E_i^0(x', 0) = E_i(x', -0) \quad (x' \in \sigma) \quad (2.11)$$

$$E, H = o\left(\frac{1}{R}\right) \quad (R \rightarrow \infty) \quad (2.12)$$

$$\frac{\partial}{\partial R} E - ik_{\pm} E = o\left(\frac{1}{R}\right), \quad \frac{\partial}{\partial R} H - ik_{\pm} H = o\left(\frac{1}{R}\right) \quad (R \rightarrow \infty, x \in X_{\pm}) \quad (2.13)$$

$$E, H \in L_2^{loc}(R^3) \quad (\text{эквивалент условия Майкснера на ребре экрана}). \quad (2.14)$$

Далее используется аппарат преобразования Фурье (двумерный случай):

$$F[u] = F[u](\lambda') = F[u(x')](\lambda') = \frac{1}{2\pi} \int_{R^2} u(x') \cdot \exp(-i\lambda'x') dx', \quad (2.15)$$

причём часто оно будет пониматься в смысле преобразования обобщенных функций медленного роста  $\mathfrak{S}'$  [12].

В силу единственности решения краевой задачи (2.6)-(2.14) на основании проведенных в [2] выкладок решение этой задачи можно искать в такой форме:

$$E_i = F^{-1} [ C_i(\lambda') \exp(\mp x_3 \gamma_{\pm}(\lambda)) ] \quad (i = 1, 2), \quad (2.16)$$

$$E_3 = -i \cdot F^{-1} [ (\lambda_1 C_1(\lambda') + \lambda_2 C_2(\lambda')) \cdot \gamma_{\pm}^{-1}(\lambda) \exp(\mp x_3 \gamma_{\pm}(\lambda)) ], \quad (2.17)$$

где

$$\lambda' = (\lambda_1, \lambda_2), \quad \lambda = \sqrt{\lambda_1^2 + \lambda_2^2} \quad (2.18)$$

$$\gamma_{\pm} = \gamma_{\pm}(\lambda) = \sqrt{\lambda^2 - k_{\pm}^2}, \quad \gamma_{\pm}^{-1} = 1/\gamma_{\pm} \quad (\sqrt{1} = 1, \sqrt{-i} = ((-1+i)/\sqrt{2})) \quad (2.19)$$

$$R = \sqrt{|x'|^2 + x_3^2}, \quad k_{\pm} = \sqrt{\varepsilon_{\pm} \varepsilon_0 \mu_{\pm} \mu_0} \quad (2.20)$$

а  $C_1(\lambda')$ ,  $C_2(\lambda')$  - искомые функции, которые, очевидно, нужно предполагать суммируемыми на вещественной оси.

Далее для компактности записи используем специальное обозначение [2]:

$$(a, b)_{\mu} = \frac{\mu_-}{\mu_- + \mu_+} \cdot a + \frac{\mu_+}{\mu_- + \mu_+} \cdot b = \alpha \cdot a + (1 - \alpha) \cdot b \quad (2.21)$$

Вместо  $C_1(\lambda')$  и  $C_2(\lambda')$  в [2] вводятся две другие искомые функции

$$\begin{aligned} u_1 &= F^{-1} [ ( (\gamma_+, \gamma_-)_{\mu} - \lambda_1^2 (\gamma_+^{-1}, \gamma_-^{-1})_{\mu} ) \cdot C_1 - \lambda_1 \lambda_2 (\gamma_+^{-1}, \gamma_-^{-1})_{\mu} C_2 ] \\ u_2 &= F^{-1} [ -\lambda_1 \lambda_2 (\gamma_+^{-1}, \gamma_-^{-1})_{\mu} \cdot C_1 - ( -(\gamma_+, \gamma_-)_{\mu} + \lambda_2^2 (\gamma_+^{-1}, \gamma_-^{-1})_{\mu} ) C_2 ] \end{aligned} \quad (2.22)$$

которые имеют сосредоточенные на области  $\sigma$  носители и удовлетворяют системе уравнений в свёртках

$$\begin{aligned} G_{1,1} * u_1 + G_{1,2} * u_2 &= -E^0_1(x', 0), \\ G_{2,1} * u_1 + G_{2,2} * u_2 &= -E^0_2(x', 0) \quad (x' \in \sigma), \end{aligned} \quad (2.23)$$

где ядра  $g_{i,j}(\lambda')$  операторов свёртки  $G_{i,j}$  являются, как показано в [2], обобщенными преобразованиями Фурье функций следующего вида:

$$\begin{aligned} g_{i,i} &= \lambda_i^2 \frac{(\gamma_+^{-1}, \gamma_-^{-1})_{\mu}}{D} - \frac{(k_+^2 \gamma_-^{-1}, k_-^2 \gamma_+^{-1})_{\mu}}{D} \quad (i = 1, 2), \\ g_{i,j} &= \lambda_i \lambda_j \frac{(\gamma_+^{-1}, \gamma_-^{-1})_{\mu}}{D} \quad (i \neq j), \end{aligned} \quad (2.24)$$

где согласно формуле (3.20) работы [2],

$$D = -q^2 - \alpha(1 - \alpha) \cdot (\lambda^2 (k_+^2 + k_-^2) - 2k_+^2 k_-^2) \cdot \gamma_+^{-1} \gamma_-^{-1} \cdot \left( q^2 = -(k_+^2, k_-^2)_{\mu} \right). \quad (2.25)$$

### 3. Цель исследования

Цель работы – это

1. Разработка концепции и модели поддержки (на алгоритмическом, проектном и программном уровнях) гибких средств вычисления ядер псевдодифференциальных операторов, возникающих в задачах пространственной дифракции электромагнитных волн в форме обобщенных интегральных преобразований определённых функций.
2. Обоснование практической корректности разработанного подхода на основе его реализации, нацеленной на вычисление ядер задачи [2].

### 4. Постановка вычислительной задачи

Подход [1] к моделированию рассеянного поля, который использован в [2] и на котором базируется наша работа, ориентирован на решение псевдодифференциальной системы (2.23) методом дискретных замкнутых рамок. Этот метод (вычислительная успешность которого для одного уравнения показана в [9]) представляет одно из направлений развития метода [5,10] дискретных замкнутых вихревых рамок. Используется (обоснованная в [2] применительно к нашему случаю) возможность интегрирования по частям в «гиперсингулярном интеграле», что позволяет в данном случае ограничиться численным построением функций двух переменных следующего вида:

$$h(x') = h_1(|x'|; k) = F^{-1} \left[ \frac{\gamma^{-1}}{D} \right] (x'), \quad h_l(x') = i \cdot F^{-1} \left[ \lambda_l \frac{\gamma^{-1}}{D} \right] (x') = \frac{\partial}{\partial x_l} h(x') \\ (0 < r \leq |x'| \leq R < \infty, \quad l = 1, 2), \quad (4.1)$$

где использовано обозначение, аналогичное (2.19),

$$\gamma^{-1} = \gamma^{-1}(\lambda; k) = \frac{1}{\sqrt{\lambda^2 - k^2}} \quad (k \in \{k_+, k_-\}), \quad (4.2)$$

а дифференцирование априори понимается в смысле обобщенных функций [12], но, как увидим, допускает комбинированную реализацию аналитическим (по отношению к «главной» части) и численным (для «остаточной» части) методом.

Реализация алгоритмической схемы [2] на компьютере нуждается в численной реализации обобщенных преобразований вида (4.1). Можно применить различные математические алгоритмы, но их исследование с оценками всех видов погрешностей является процессом, зависящим от творческой работы математиков. Между тем, организацию попеременного выполнения этих алгоритмов на машине с целью получения достоверных результатов за приемлемое время счёта, можно осуществить значительно быстрее. При этом подходе, однако, наиболее проблемным становится вопрос доверия к созданной по такому принципу компьютерной программе.

Тогда вычислительная задача дополняется требованием контроля качества программного средства многократных вычислений (4.1) на уровне строгости, принятом для программного обеспечения критических применений [11].

Концепция гибкости требует подбора альтернативных алгоритмов вычислений (как говорят, алгоритмической диверсности, то есть, разнообразия). Из требований к программному средству необходимо вывести модель для

проектных решений. А для конкретного интегрального преобразования нужно дать модульную схему реализации и написать код всех программных модулей.

### 5. Аналитическая часть вычислений

Высокие требования к скорости вычислений накладывают

**Требование 5.1.** Отдать предпочтение методам, которые используют одномерные (а не двумерные) интегральные преобразования.

В нашем случае это означает, что необходимо использовать полную поворотную симметрию функций, выражая двумерное преобразование Фурье через одномерное преобразование Ганкеля порядка  $\nu = 0$ :

$$f(|x|) = \int_0^{\infty} \tilde{f}(\lambda) \cdot \lambda \cdot J_0(\lambda|x|) d\lambda = G[\tilde{f}](|x|). \quad (5.1)$$

Далее потребуем:

**Требование 5.2.** Численный метод расчёта  $f(x) = f(|x|)$  (4.1) должен обеспечивать получение результата в форме непрерывно дифференцируемого приближения  $f_n(x)$  (где  $\rho$  - параметр управления точностью), так что

$$\frac{\partial}{\partial x_l} f_n(x) = \left[ \frac{d}{d\rho} f_n(\rho) \right]_{\rho=|x|} \cdot \frac{x_l}{|x|}, \quad (5.2)$$

предполагая практическую близость  $\partial f_n(x)/\partial x_l$  к  $\partial f(x)/\partial x_l$  по норме  $L_1$ , что может подтверждаться (или подвергаться сомнению) в результате численного эксперимента.

Рассмотрим при  $0 < a < 1$  функцию

$$\gamma_a^{1-}(\lambda) = \begin{cases} \frac{-i}{\sqrt{1-\lambda^2}} \Leftrightarrow \lambda < 1-a \\ \frac{-i \cdot (1+a-\lambda)}{2a\sqrt{a(2-a)}} + \frac{\lambda-1+a}{2a\sqrt{a(2+a)}} \Leftrightarrow |\lambda-1| \leq a \\ \frac{1}{\sqrt{\lambda^2-1}} \Leftrightarrow \lambda > 1+a \end{cases} \quad (5.3)$$

Положим

$$\gamma_a^{1-}(\lambda; k) = \frac{1}{k} \gamma_a^{1-}\left(\frac{\lambda}{k}\right). \quad (5.4)$$

**Лемма 5.1.** Функции от  $\lambda$  семейства (5.4) обладают следующими свойствами

1<sup>0</sup>.  $\gamma_a^{1-} \in C[0, \infty) \cap L_{1+\varepsilon}$  (при любом  $\varepsilon > 0$ ).

2<sup>0</sup>. Преобразование Ганкеля можно выразить через одну функцию (5.3):

$$G[\gamma_a^{1-}(\cdot; k)] = k \cdot G[\gamma_a^{1-}(\cdot)] \quad (k > 0). \quad (5.5)$$

3<sup>0</sup>. В сравнении с функцией, введенной в (4.2),

$$\gamma_a^{1-}(\lambda; k_{\pm}) = (\gamma(\lambda))^{-1} \Leftrightarrow \lambda \in (0; k-ak) \cup (k+ak; \infty). \quad (5.6)$$

4<sup>0</sup>. Структура преобразования Ганкеля модельной функции (5.3) такова

$$G[\gamma_a^{-1}(\cdot)](\rho) = \frac{\exp(i\rho)}{\rho} + g_a(\rho) \quad (g_a \in C^\infty[0, \infty), g_a(\rho) \xrightarrow{\rho \rightarrow \infty} 0). \quad (5.7)$$

Доказательство первых трёх утверждений леммы – прямая проверка. Последнее утверждение основано на известной формуле Зоммерфельда, которую можно рассматривать как утверждение о виде преобразования Ганкеля функций вида  $\gamma^{-1}$  (см. (2.19)). Действительно, в силу  $3^0$  разность между этой функцией и (5.3) представляет собой финитную интегрируемую функцию. Её преобразование Ганкеля можно, очевидно, дифференцировать, внося операцию дифференцирования под интеграл, сколько угодно раз.

Для конструктивного использования соображений работы [2], относящихся к аналитическим свойствам ядер (2.24), выясним структуру функций вида (4.1).

**Лемма 5.2.** Составляющие ядер (2.26) - функции вида  $h$  (4.1) можно представить следующим образом

$$h_1(\rho; k) = G[\gamma_a^{-1}(\cdot; k)](\rho) + g(\rho; k) \quad (\forall k: g(\cdot; k) \in C^1[0, \infty)), \quad (5.8)$$

причём функция  $g$  является преобразованием Ганкеля (5.1) в классическом смысле, и, следовательно, при заданном  $k$  её значения могут быть найдены приближенно с помощью различных численных методов.

Доказательство получим, записав

$$\frac{\gamma^{-1}}{D(\lambda)} = \frac{\gamma_a^{-1}(\lambda; k)}{D(\infty)} + \left( \frac{\gamma^{-1}}{D(\lambda)} - \frac{\gamma_a^{-1}(\lambda; k)}{D(\infty)} \right) = \frac{\gamma_a^{-1}(\lambda; k)}{D(\infty)} + \tilde{g}(\lambda; k), \quad (5.9)$$

где оба слагаемые непрерывны, а второе, принимая во внимание (2.25) и (5.6), имеет следующее поведение на бесконечности:

$$\begin{aligned} \tilde{g}(\lambda; k) &= \gamma^{-1}(\lambda) \frac{D(\infty) - D(\lambda)}{D(\infty)D(\lambda)} = \\ &= \gamma^{-1}(\lambda) (D(\infty)D(\lambda))^{-1} \left( \alpha(1-\alpha) \left( (k_+^2 + k_-^2) \left( \frac{\lambda^2}{\gamma_+ \gamma_-} - 1 \right) + \frac{2k_+^2 k_-^2}{\gamma_+ \gamma_-} \right) \right) \quad (5.10) \\ &\sim \lambda^{-1} D(\infty)^{-2} \alpha(1-\alpha) \left( \frac{(k_+^2 + k_-^2)}{2} + 2k_+^2 k_-^2 \right) \lambda^{-2} + O(\lambda^{-5}) \quad (\lambda \rightarrow \infty) \end{aligned}$$

так что преобразования Ганкеля функций  $\tilde{g}$ ,  $\lambda \cdot \tilde{g}$  непрерывны.

## 6. Алгоритмическая диверсность численного метода

Из различных численных методов, возможных для расчёта преобразования (5.1), выбираем по требованию 5.2 такие, которые позволяли бы строить приближение к результирующей функции из некоторого конечномерного подпространства непрерывно дифференцируемых функций.

**Версия 1.** Разложение по собственным функциям оператора Ганкеля.

Известно (см., напр., [13]) разложение по собственным функциям самосопряженного унитарного оператора

$$H_\nu[f](x) = \int_0^\infty f(y)\sqrt{xy}J_\nu(xy)dy \quad \left( G[\tilde{g}](x) = x^{-1/2}H_\nu[y^{1/2}\tilde{g}](x) \right) \quad (6.1)$$

(параметр  $\nu = \operatorname{Re} \nu > -1$ , «наш» случай  $\nu = 0$ ). Это разложение имеет вид

$$f(y) = y^{\nu+1/2} \exp\left(-y^2/2\right) \operatorname{li.m.} \sum_0^\infty f_l L_l^\nu(y^2) \quad (f \in L_2(0, \infty)) \quad , \quad (6.2),$$

где  $L_l^\nu(t)$  ( $l = 0, 1, \dots$ ) - классические ортогональные полиномы Лагерра; причём собственной функции с номером  $l$  соответствует собственное число  $(-1)^l$ , а

$$f_l = \frac{1}{\|E_l\|_{L_2}} (f, E_l)_{L_2} \quad \left( E_l(y) = y^{\nu+1/2} \exp\left(-y^2/2\right) L_l^\nu(y^2) \right) . \quad (6.3)$$

Методы точного вычисления  $L_l^\nu(t)$  и их нормировки, хорошо известны (см., напр., [14]). В частности,

$$\|E_l\|_{L_2}^2 = \int_0^\infty (E_l(y))^2 dy = \int_0^\infty (E_l(\sqrt{t}))^2 \frac{dt}{2\sqrt{t}} = \|L_l^\nu\|_{L_2(\rho)}^2 = \frac{\Gamma(l+\nu+1)}{l!} \quad (6.4)$$

$$(\rho(t) = t^\nu e^{-t}) .$$

Разложение (6.1) позволяет получить приближение к искомому преобразованию в форме аналитической функции с конечным числом параметров следующего образом:

$$G[\tilde{g}](x) = x^{-1/2}H_0[y^{1/2}\tilde{g}](x) = x^{-1/2}H_0[f](x) = x^{-1/2} \sum_{l=0}^\infty (-1)^l f_l E_l(x) =$$

$$\approx x^{-1/2} \sum_{l=0}^N (-1)^l (y^{1/2}\tilde{g})_l E_l(x) \quad , (6.5)$$

где коэффициенты  $f_l$  будем вычислять на основе квадратурных формул Гаусса по нулям классических ортогональных полиномов на полуоси, то есть [14], тех же полиномов Лагерра:

$$(y^{1/2}\tilde{g})_l = \int_0^\infty y^{1/2}\tilde{g}(y)E_l(y)dy = 2^{1/2} \int_0^\infty t^{-1/2}\tilde{g}(\sqrt{2t})L_l^\nu(2t)\rho_\nu(t)dt =$$

$$\approx 2^{1/2} \sum_{i=0}^m \lambda_i^m x_i^{-1/2}\tilde{g}(\sqrt{2x_i})L_l^\nu(2x_i) \quad (x_0 < x_1 \dots < x_m, L_{m+1}^\nu(x_i) = 0) \quad . (6.6)$$

**Версия 2.** Использование быстрого косинус-преобразования.

Согласно интегральному представлению Пуассона (при  $\nu = 0$ ) имеем:

$$J_0(z) = \frac{1}{\pi} \int_{-1}^1 (1-t^2)^{-1/2} \cos(zt) dt$$

и в силу теоремы Фубини (достаточно непрерывности  $\tilde{g}(\lambda)$  и её убывания не хуже  $\lambda^{-2-\varepsilon}$  при  $\lambda \rightarrow \infty$ ) и квадратуры Гаусса-Лежандра (см., напр., [14]) получим

$$G[\tilde{g}](x) = \frac{1}{\pi} \int_{-1}^1 (1-t^2)^{-\frac{1}{2}} dt \int_0^{\infty} \lambda \tilde{g}(\lambda) \cos(\lambda xt) d\lambda = \frac{1}{\pi} \int_{-1}^1 (1-t^2)^{-\frac{1}{2}} f(xt) dt = \\ \frac{1}{\pi} \sum_{i=1}^{2M} \lambda_i f(xt_i) \quad ,$$

где  $x_i$  - нули полинома Лежандра степени  $2M$ , расположенные симметрично относительно нуля,

$\lambda_i = \pi/2M$  - соответствующие числа Кристоффеля (см. [14]).

Таким образом,

$$G[\tilde{g}](x) \approx \frac{1}{M} \sum_{i=1}^M f(z_i) \quad (z_i = xt_i > 0), \quad (6.7)$$

где

$$f(z) = C[\tilde{g}_1](z) = \int_0^{\infty} \tilde{g}_1(\lambda) \cos(z\lambda) d\lambda \quad (\tilde{g}_1(\lambda) = \lambda g(\lambda)) \quad . \quad (6.8)$$

Имея в виду последующую сплайн-интерполяцию, введём узлы:

$$z^k = \frac{k}{N} R \quad (k = 0, \dots, N, \quad N = N_1 \cdot N_2) \quad , \quad (6.9)$$

где множители  $N_i$  выберем позже.

Далее мы следуем известным идеям (см., напр., [15]), позволяющим использовать алгоритм быстрого дискретного преобразования Фурье.

Интегральное косинус-преобразование в узлах (6.9) сведём к вычислению коэффициентов в разложении по косинусам вспомогательных функций:

$$f(z^k) = \int_0^{\infty} \tilde{g}_1(\lambda) \cos\left(\frac{kR\lambda}{N}\right) d\lambda = \frac{2\pi N}{R} \int_0^{\infty} \tilde{g}_1\left(\frac{2\pi N}{R}\mu\right) \cos(2\pi k\mu) d\mu \\ = c \sum_{l=0}^{\infty} \left\{ \int_l^{l+\frac{1}{2}} + \int_{l+\frac{1}{2}}^{l+1} \right\} \tilde{g}_1(c\mu) \cos(2\pi k\mu) d\mu \\ = c \sum_{l=0}^{\infty} \int_0^{\frac{1}{2}} \left[ \tilde{g}_1(c(v+l)) + (-1)^k \tilde{g}_1\left(c\left(v+l+\frac{1}{2}\right)\right) \right] \cos(2\pi kv) dv \quad \left(c = \frac{2\pi N}{R}\right).$$

Меняя здесь местами порядок суммирования и интегрирования на основании теоремы Лебега, в итоге получим

$$f(z^k) = \int_0^{\frac{1}{2}} (\varphi(v) + (-1)^k \psi(v)) \cos(2\pi kv) dv = \Phi_k + (-1)^k \Psi_k, \quad (6.10)$$

где



$$\varphi(v) = c \sum_{l=0}^{\infty} \tilde{g}_1(c(v+l)) , \quad \psi(v) = c \sum_{l=0}^{\infty} \tilde{g}_1(c(v+l + \frac{1}{2})) . \quad (6.11)$$

Для приближенного вычисления коэффициентов вида

$$\Phi_k = \int_0^{\frac{1}{2}} \varphi(v) \cos(2\pi k v) dv \quad (6.12)$$

в соответствии с идеями быстрого преобразования Фурье мы должны использовать квадратуру по значениям функции в равноотстоящих узлах. Разобьём интервал интегрирования на  $N_2$  одинаковых отрезка и применим на них одну и ту же квадратурную формулу Ньютона-Котесса с  $N_1 + 1$  узлами:

$$\Phi_k \approx \sum_{i=0}^{N_2-1} \frac{1}{2N_2} \sum_{j=0}^{N_1} Q_j \varphi(\mu_{ij}) \cos(2\pi k \mu_{ij}) , \quad (6.13)$$

где

$$\mu_{ij} = \frac{N_1 i + j}{2N_1 N_2} = \frac{N_1 i + j}{2N} , \quad (6.14)$$

причём  $Q_j = Q_j^{N_1}$  при малых  $N_1$  хорошо известны, а при больших – известным образом вычисляются (см., напр., [15]).

Для преобразования суммы вида (6.13) к однократной сумме введём функцию соответствия индексов:

$$l = l(i, j) : \quad 0 = l(0,0), \quad 1 = l(0,1), \dots, N_1 = l(0, N_1) = l(1, 0), \quad N_1 + 1 = l(1, 1), \dots \\ N - N_1 = l(N_2 - 2, N_1) = l(N_2 - 1, 0), \dots, N = l(N_2 - 1, N_1) .$$

Тогда формула (6.13) примет следующий вид:

$$\Phi_k \approx \frac{\phi_0 + \phi_N}{2} + \sum_{l=1}^{N-1} \phi_l \cos(\pi k l) \quad (k = 0 \dots N) , \quad (6.15)$$

где

$$\phi_l = \begin{cases} l = 0 : & 2Q_0 \varphi(0)/2N_2 \\ l = N : & 2Q_{N_1} \varphi(1/2)/2N_2 \\ l = iN_1 : & 2Q_{N_1} \varphi(i/2N_2)/2N_2 \\ \text{others} : & Q_{N_1} \varphi(l/2N)/2N_2 \end{cases} . \quad (6.16)$$

В качестве значений  $N_i$  выгодно брать степени двойки: тогда для серии вычислений (6.15) пригоден в своей известнейшей версии алгоритм быстрого дискретного косинус-преобразования Фурье, программные реализации которого широко доступны. Обсуждаемый алгоритм, как известно, имеет сложность порядка  $N \log(N)$  (по числу операций). Это даёт очевидные преимущества в сравнении с прямым алгоритмом вычисления совокупности значений (6.15), порядок сложности которого есть  $N^2$ .

Та же схема вычислений применима к коэффициентам  $\Psi_k$ , определяемым по функции  $\psi$  (см. (6.11)), которые необходимы для вычисления значений (6.10).

Остаётся интерполировать значения функции (6.8) на интервале  $[0; R]$  по её значениям в узлах (6.9). Поскольку степени классических полиномиальных сплайнов нечётны, а наряду со значениями самой функции необходимо интерполировать значения производной, то естественное требование использовать сплайны минимального порядка диктует применение кубических сплайнов с предварительной оценкой значений производных на концах интервала. Алгоритм построения таких сплайнов хорошо известен [15].

**Версия 3.** Специальные методы для специальных функций.

Для нас интерес представляет функция

$$g_a(\rho) = A \int_{1-a}^{1+a} \lambda J_0(\lambda \rho) d\lambda + B \int_{1-a}^{1+a} \lambda^2 J_0(\lambda \rho) d\lambda - \int_{1-a}^{1+a} \frac{\lambda J_0(\lambda \rho)}{\sqrt{\lambda^2 - 1}} d\lambda \quad (6.17)$$

$$= A \cdot j_0(\rho) + B \cdot j_1(\rho) - j_2(\rho),$$

где

$$A = \frac{(a-1)\sqrt{2-a} - i \cdot (a+1)\sqrt{2+a}}{2a\sqrt{a(4-a^2)}}, \quad B = \frac{\sqrt{2-a} + i \cdot \sqrt{2+a}}{2a\sqrt{a(4-a^2)}}. \quad (6.18)$$

Ясно, что

$$j_0(\rho) = \rho^{-1}((1+a)J_1((1+a)\rho) - (1-a)J_1((1-a)\rho)),$$

$$j_2(\rho) = \int_{\sqrt{a(a-2)}}^{\sqrt{a(a+2)}} J_0(\rho\sqrt{t^2+1}) dt, \quad (6.19)$$

а интеграл  $j_1(\rho)$  можно взять, выразив, как известно, ответ с помощью функций Бесселя и Струве. Но последнее не выглядит привлекательным с вычислительной точки зрения, поскольку требует 8 вычислений функций типа Бесселя. Решающим для быстрого вычисления значений (6.7) является то, что можно потребовать малости параметра  $a$ , например,

$$aR \leq 3 \quad (0 < \rho \leq R). \quad (6.20)$$

На основании этого под интегралом в  $j_1$  функцию Бесселя можно заменить интерполирующим полиномом фиксированной степени, выбирая её по допустимой погрешности интерполяции. Для интеграла (6.9) такой прием не приведёт к явному аналитическому выражению. Но, наметив степень интерполирующего многочлена для функции Бесселя, можно оценить необходимый порядок квадратуры Гаусса-Лежандра. Коэффициенты обсуждаемых интерполирующих многочленов так же, как и соответствующие погрешности нетрудно получить известными методами [15].

Аналогичным образом вычисляются и производные функции (6.7), если по заданным требованиям точности её численное дифференцирование по приближенно вычисленным значениям самой функции не желательно.

**Замечание 6.1.** Не только для вычисления ядер псевдодифференциальных уравнений работы [2], но и для вычисления полей, и при решении других задач дифракции электромагнитных и акустических волн на плоскопараллельных структурах желательно иметь гибкие программные средства для расчёта

преобразований Фурье функций точки на плоскости, в особенности таких функций, которые зависят только от расстояния точки до начала координат.

**Замечание 6.2.** Первоочередного внимания заслуживают версии 1 и 2, поскольку обе применимы к широким классам функций. Поскольку оценки максимальной погрешности всех использованных при синтезе этих версий методов известны (на классах функций), то ясно, как составить теорему об оценке максимальной погрешности для каждой из версий 1 и 2. Но для вычислений представляет интерес иное: какая из версий приводит к минимальной (не максимальной!) погрешности на данной функции (а не в классе функций!). То, что на разных функциях минимум погрешности достигается разными версиями, можно проверять так. Очевиден случай, когда этот минимум доставляется первой версией и равен нулю. А именно, если преобразуемая функция есть произведение экспоненты (с определённым показателем) на произвольный многочлен. Напротив, лучшее приближение (при прочих равных условиях) может дать вторая версия, если преобразуемая функция финитна (тогда ряды в (6.11) превращаются в конечные суммы) и бесконечно дифференцируема. Поэтому выбор версии был бы уместен в процессе самих вычислений в порядке адаптации алгоритма по итогам оценки скорости (или точности) в уже произведенных вычислениях.

### 7. Архитектура реализации алгоритмической диверсности

Вопрос заключается в том, чтобы вывести из требований к библиотеке реализации (5.1), модель её архитектуры, то есть, правила определения состава основных модулей с базовыми типами и вспомогательных модулей.

Начнём с требований, имея в виду применения на базе МДО.

Если для реализации (5.1) используется алгоритм, основанный на версиях 6.1-6.3, то требования 5.1 и 5.2 выполнены. Далее уместны такие требования.

**Требование 7.1.** Программная реализация некоего интегрального преобразования функций

$$G : f \mapsto G[f] \quad (f = f(\lambda), G[f] = G[f](x)) \quad (7.1)$$

должна предусматривать возможность статического и динамического определения в программе объектов, символизирующих преобразование  $G$  вида (7.1). Эти объекты должны быть параметризованы функцией-преобразованием  $f$  и допускать, по крайней мере, следующие операции. Инициализация (создание функции-образа  $G[f]$ ) и вычисление значения  $G[f](x)$  по заданному  $x$ .

Нам требуется также операция вычисления производной функции-образа (требование 5.2), но в изложении она для краткости не упоминается.

**Требование 7.2.** Пусть для некоторого преобразования (7.1) математическая реализация диверсна, то есть предусматривает альтернативные варианты необходимых структур данных и различные версии алгоритмов для реализующих операций. Тогда программная реализация преобразования (7.1) должна быть представлена в исполняемой программе объектом, при (вообще говоря, динамическом) определении которого и должна выбираться та единственная версия реализации, которая всегда будет использоваться для данного объекта. Требуется, чтобы при этом выбор версии мог производиться

независимо от выбора параметра  $f$ . Далее, требуется, чтобы набор предусмотренных для реализации объекта операций не зависел от версии реализации, и при том выполнение всех операций для всех версий было корректным по смыслу. Какие-либо возможности взаимодействия с объектом помимо этих общих операций, должны быть пользователю закрыты.

Из этих требований вытекает, что структуры данных и операции, используемые для реализации математического объекта вида (7.1), должны для каждой версии реализации программного объекта инкапсулироваться в отдельной конструкции используемого языка программирования. При этом пользовательская программа взаимодействует с такой конструкцией только средствами, которые на уровне исходного текста выглядят одинаково для всех объектов, какую бы версию алгоритма они не использовали (скрытие реализации). Связь между версиями программной реализации должен обеспечить механизм наследования, который позволял бы объединять все эти версии внутри иерархии, где все члены кроме одного, корневого, создаются по образцу корневого, но имеют добавления по данным и собственные реализации операций (расширение состава публичных операций в принципе, возможно, но у нас оно запрещается требованием 7.2). Наконец, требуемая корректность операций означает автоматический выбор версии алгоритма её выполнения по версии реализации, фиксированной в определении данного объекта. Это позволяет работать с полиморфными объектами.

Итак, из «рамочных» требований 7.1, 7.2 выводим модель реализации диверсности, которая соответствует принципам объектно-ориентированного программирования (ООП), предполагая использование таких языков, как Ада, С++ или Дельфи. Не очевидно, что это согласуется (а не входит в конфликт) с конкретным содержанием задачи программирования и с требованием поддержки контроля качества (о чём подробнее в следующем разделе). Обоснование этой согласованности не может быть формальным, но может следовать логике доказательства непротиворечивости формальных систем. А именно, следует интерпретировать модель в полнофункциональной программной реализации.

Для целей проведения вычислительных экспериментов, связанных с математической моделью [2], нами была разработана на языке Ада библиотека `Hankel_Transform`, реализующая версии 1-3 преобразования Ганкеля из раздела 6. Эта библиотека сразу разрабатывалась как интерпретация вышеописанной объектно-ориентированной схемы. Поэтому она может служить искомым неформальным доказательством её корректности. Поясним интерпретацию.

В корневом модуле `H` (см. рис.1) определяется абстрактный тип и абстрактные операции для преобразования (7.1). Далее для версий реализации данного преобразования разработаны библиотечные пакеты `H.1`, `H.2` `H.3`, описания которых содержат определения типов, которые наследует абстрактному типу из `H`, а тела содержат реализации абстрактных операций. Проблеме необходимой параметризации объектов таких типов мы дали прямое решение средствами языка Ада: в определении абстрактного типа предусмотрен дискриминант ссылочного типа, который ссылается на преобразуемые комплексные функции одной вещественной переменной. Вот описания `H` и `H.1`.

```

with Base_Arrays_Types; use Base_Arrays_Types;
package Hankel_Transform is
  use Complex_Double;
  type Source_Function is access function (X: Double) return Complex;
  subtype Positive_Double is Double range 0.0..0.1*2.0**Double'Machine_EMax;
  type Segment is record Left, Right: Positive_Double; end record;
  type Transform_Of (F: Source_Function) is abstract tagged private;
  procedure Initialize (H: in out Transform_Of;
    On_The: Segment;
    F_Delta: Positive_Double) is abstract;
  function Value (H: Transform_Of; X: Positive_Double) return Complex is abstract;
private
  -- Опустим подробности реализации
end Hankel_Transform;

package Hankel_Transform.By_Laguerre is
  type H0_Transform_Of --(F: Source_Function)
    is new Transform_Of with private;
  procedure Initialize (H: in out H0_Transform_Of;
    On_The: Segment;
    F_Delta: Positive_Double);
  function Value (H: H0_Transform_Of; X: Positive_Double) return Complex;
private
  -- Опустим подробности реализации
  function Get_Inner_Representation (H: H0_Transform_Of) return Complex_Vector;
  --Это дополнительная, но приватная операция (требование 7.1 не нарушается)
end Hankel_Transform.By_Laguerre;

```

Другие модули - потомки Н – это пакеты интегрирования (Н.0), построения сплайнов (Н.4), реализации БПФ (Н.5 – подпрограмма Fast\_Cosine). Аппарат многочленов Лагерра обеспечивает вспомогательная библиотека с корнем L .

### 8. Контроль качества, «встроенный в разработку»

Модель качества (в смысле [11]), которая предложена в [17] для исследовательских программных реализаций методов дискретных особенностей в задачах дифракции, предусматривает обязательную оценку 6 общепринятых характеристик внутреннего и, по возможности, - таких же характеристик внешнего качества. Наборы соответствующих мер (метрик) для указанных видов качества в данной модели приведены во взаимнооднозначное соответствие.

Для обеспечения поддержки вычисления этих метрик объектно-ориентированную модель предыдущего подраздела мы дополняем дочерними модулями тестирования. В рассматриваемой интерпретации нашей модели (рис.1) это модули, помеченные как Н.0.0, Н.1.0, Н.2.0, Н.3.0, Н.4.0, L.0.0, L.1.0 (это подпрограммы с идентификатором Test), Н.6 (Fast\_Cosine\_Test) и L.0 (Test\_Tools) – модуль-заготовка, реализованная в L.0.0. Отметим, что среди тестовых функций в пакете Funcs\_For\_Tests есть примеры, предполагающие достижение более высокой точности по алгоритму версии 1 или по версии 2.

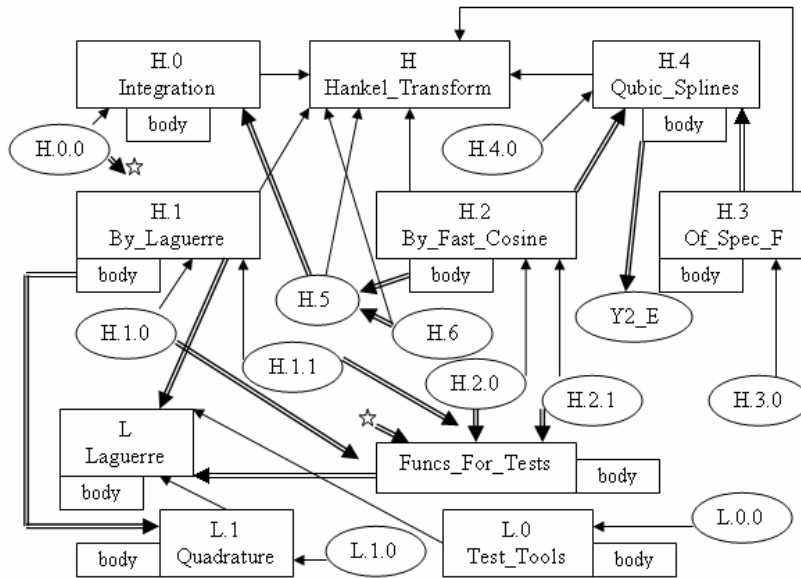


Рис.1. Схема дивергентной реализации (5.1). Иерархии потомственных модулей, помечены: H – подсистема преобразования Ганкеля, L – подсистема полиномов Лагерра. При Ада-кодировании прямоугольники суть пакеты, овалы – подпрограммы, стрелки указывают: простая – модуль-родитель, двойная – модуль контекста.

### 9. Автономное тестирование и метрические оценки

Модель качества [17] для исследовательского ПО рассматриваемого здесь класса следует стандартам (см. [11]) и предусматривает оценки качества: внутреннего (основанного на анализе разработки ПО) и внешнего (основанного на анализе его пробного функционирования) путём оценивания свойств в рамках системы метрик. На практике создаваемые исследователями программы либо вовсе лишены нужных свойств, либо требуют их проявления путём уникальных исследований. Система встроенных тестовых функций предыдущего раздела индуцирует появление или (и) проявление ряда нужных свойств. Это воздействие на метрический контроль затрагивает 4 из 6 главных характеристик качества. При этом реализуются метрики, которые в половине случаев относятся к наиболее весомым подхарактеристикам (в смысле весов, присвоенных в [17]).

Табл.1. Воздействие разработанной архитектурной модели на возможность оценки [17] характеристик качества программной реализации преобразования Ганкеля

Характеристика	«Функциональность»	«Надёжность»	«Эффективность»	«Сопровождаемость»		
				«Изменяемость»	«Устойчив.»	«Тестируем.»
<b>Подхарактеристика (вес)</b>	«Точность» (0.2)	«Завершён-ть» (0.5)	«Временные хар.» (0.5)	«Изменяемость» (0.4)	«Устойчив.» (0.2)	«Тестируем.» (0.2)
<b>Метрики</b>	2 из 2	1 из 2	1 из 1	1 из 2	1 из 1	3 из 3
<b>Оценка воздейств.</b>	$0.2 \times 1 = 0.2$	$0.5 \times 0.5 = 0.25$	$0.5 \times 1 = 0.5$	$0.4 \times 0.5 + 0.2 \times 1 + 0.2 \times 1 = 0.6$		

Переходя от правильности реализации к вопросам полезности алгоритмов в наших задачах, отметим наличие двух основных параметров. По версии 1 это  $N$  и  $m$  (6.5), а по версии 2 -  $M$  (6.7) и  $N$  (6.9). Точность и скорость вычислений зависят от них, а также от ряда решений внутри реализаций. Исследование версий в зависимости от настройки параметров и обоснование рекомендаций по их практическому использованию для целей вычислительных экспериментов в задачах 3-мерной дифракции на плоско-параллельных структурах - отдельная тема. Ограничимся предварительными выводами по итогам численных экспериментов со встроенными тестами. Версия 1 может иметь преимущества при отладке программного обеспечения вычислительных экспериментов в целом, когда к выбору волновых чисел и точности расчётов серьёзных требований нет. С ужесточением требований эту версию в её роли альтернативного подхода, похоже, придётся заменять счётом параметрических интегралов (5.1) (как в [8]) в сочетании со сплайн-интерполяцией.

#### 10. Заключительные выводы

Предлагаемая в работе концепция гибкости численных методов для построения гиперсингулярных ядер, описанных в форме обобщенных интегральных преобразований, основана на одновременном использовании существенно различных математических алгоритмов вычисления данного преобразования для регулярных функций. В конкретных задачах это требует аналитической работы по выделению частей ядер, имеющих неинтегрируемые особенности. Эта работа выполнена в статье для ядер псевдодифференциальных уравнений, недавно построенных в одной актуальной задаче дифракции [2].

Обоснована возможность гибкой программной реализации за счёт переноса на неё алгоритмической диверсности. Соответствующие требования теоретически ведут к модели программной библиотеки на основе объектно-ориентированного подхода. Практическая корректность этой модели неформально обоснована её интерпретацией в схеме созданной автором диверсной программной библиотеки преобразования Ганкеля.

Показано, что многие весомые, но сложные для оценки метрики стандартного контроля качества библиотек, реализующих преобразования вида (7.1) по нашей модели, можно поддержать простым дополнением их архитектуры.

Анализ результатов содержательных численных экспериментов, в которых используется наша библиотека реализации (5.1), будет опубликован отдельно.

#### ЛИТЕРАТУРА

1. Гандель Ю.В. Парные и гиперсингулярные интегральные уравнения задач дифракции электромагнитных волн на плоских решетках и экранах // Труды XI Международного симпозиума МДОЗМФ-2003.-Харьков-Херсон,2003.- С.53-58.
2. Гандель Ю.В., Мищенко В.О. Псевдодифференциальные уравнения электромагнитной дифракции на плоскопараллельной структуре и их дискретная модель // Вестник Харк. нац. ун-та., – 2006. – № 733. Сер. «Математическое моделирование. Информационные технологии. Автоматизированные системы управления», вып. 6. – С. 58-75:

3. Гандель Ю.В., Духопельников С.В. Краевые задачи для уравнений Гельмгольца и Максвелла на многощелевых цилиндрических структурах и граничные интегральные уравнения на системе отрезков. //Крайові задачі для диференціальних рівнянь: Зб. наук. пр. – Чернівці: Прут, 2008. – Вип. 16. – с.264-293.
4. Гахов А. В., Мищенко В.О. Трехмерная модель метода дискретных особенностей рассеяния скалярных волн экраном на границе раздела сред // Вестник ХНТУ. – 2006. – №2(25). – С.135-140.
5. Вайникко Г.М., Лифанов И.К., Полтавский Л.Н. Численные методы в гиперсингулярных интегральных уравнениях и их приложения. – М.: Янус-К.– 2001.– 508 с.
6. Гандель Ю.В. Введение в методы вычисления сингулярных и гиперсингулярных интегралов. – Харьков: Харківський національний університет імені В.Н. Каразіна, 2004. – 96 с.
7. Анфиногенов А. Ю. Практическое ускорение численного решения трехмерной скалярной задачи дифракции методом дискретных особенностей // Вестник Харк. нац. ун-та. – 2007. – № 775. Сер. "Математическое моделирование. Информационные технологии. Автоматизированные системы управления", вып. 7. – С. 3-10.
8. Гахов А. В. Численное исследование рассеяния скалярных волн плоским экраном на границе слоя в полупространстве над жесткой плоскостью // Вестник Харк. нац. ун-та. – 2007. – № 780. Сер. "Математическое моделирование. Информационные технологии. Автоматизированные системы управления", вып. 8. – С. 79-93.
9. Антоненко А. В., Гандель Ю. В. Численный анализ гиперсингулярного уравнения задач дифракции на плоском экране // Вестник Харк. нац. ун-та. – 2003. – № 590. Сер. "Математическое моделирование. Информационные технологии. Автоматизированные системы управления", вып. 1. – С. 9-14.
10. Лифанов И.К. Метод сингулярных интегральных уравнений и численный эксперимент. – М.: ТОО «Янус», 1995. – 520 с.
11. Харченко В.С, Склад В.В., Тарасюк О.М. Методы моделирования и оценки качества и надёжности программного обеспечения. – учеб. пособие. – Харьков: Нац. аэрокосм. ун-т «Харьк.авиацин-т», 2004. – 159 с.
12. Владимиров В.С. Обобщенные функции в математической физике. – М.: Наука. –1979, 320 с.
13. Ахиезер Н.И. Лекции об интегральных преобразованиях. – Харьков: Вища школа. Изд-во при Харьк. Ун-те, 1984. – 120 с.
14. Никифоров А.Ф., Уваров В.Б. Основы теории специальных функций. – М.: Глав. ред. физ-мат. лит. из-ва «Наука», 1974. – 303 с.
15. Бахвалов Н.С., Жидков Н.П., Кобельков. Численные методы. – М.: Наука. Гл. ред. Физ.-мат. лит., 1987. – 600 с.
16. Многоверсионные системы, технологии, проекты / В.С. Харченко, В.Я. Жихарев, В.М. Ильющко, Н.В. Ничипорук / Под ред. В.С. Харченко. – Х.: НАКУ «ХАИ», 2003. – 486 с.
17. Gahov A.V., Mishchenko V.O. The Validation of the Software that Was Developed for Calculations Related to the Design of Antennas // Радіоелектронні і комп'ютерні системи. – 2007. – N 6. – P. 180-185.926.