

Е.М. КАРЧЕВСКИЙ

**Численные методы  
решения интегральных уравнений  
и комплекс программ  
на языке Matlab**

Учебное пособие

Казанский университет

2017

---

---

## Оглавление

<b>Предисловие</b> . . . . .	2
<b>ГЛАВА 1. Методы решения уравнений Вольтерра II рода</b> . . . . .	3
§ 1. Метод квадратур . . . . .	3
§ 2. Метод квадратур для нелинейного уравнения . . . . .	11
§ 3. Метод простой итерации . . . . .	17
<b>ГЛАВА 2. Методы решения уравнений Фредгольма II рода</b> . . . . .	22
§ 1. Метод квадратур . . . . .	22
§ 2. Метод вырожденных ядер . . . . .	27
§ 3. Метод наименьших квадратов . . . . .	34
§ 4. Метод Галеркина — Петрова . . . . .	40
§ 5. Метод коллокации . . . . .	45
§ 6. Метод простой итерации . . . . .	52
§ 7. Метод моментов . . . . .	52
<b>ГЛАВА 3. Однородное уравнение Фредгольма II рода</b> . . . . .	53
§ 1. Вырожденные ядра . . . . .	53
§ 2. Метод Бубнова — Галеркина . . . . .	57
§ 3. Метод Ритца . . . . .	58
§ 4. Метод следов . . . . .	58
§ 5. Метод Келлога . . . . .	58
<b>Литература</b> . . . . .	59

---

---

## Предисловие

Книга предназначена для проведения практических занятий по численным методам решения интегральных уравнений со студентами, специализирующихся в области прикладной математики, информатики и математического моделирования.

Автор приносит свою искреннюю признательность студентам кафедры прикладной математики Казанского университета за предоставленные расчетные материалы.

---

---

## ГЛАВА 1

# Методы решения уравнений Вольтерра II рода

### § 1. Метод квадратур

1. **Линейное уравнение Вольтерра II рода** имеет следующий вид:<sup>1)</sup>

$$y(x) - \int_a^x K(x, s)y(s)ds = f(x), \quad x \in [a, b]. \quad (1)$$

Здесь  $y(x)$  — неизвестная функция,  $K(x, s)$  — ядро интегрального уравнения,  $f(x)$  — свободный член (правая часть) интегрального уравнения. Однородное уравнение (при  $f \equiv 0$ ) имеет только тривиальное решение, а условия существования решения неоднородного уравнения (1) связаны с различными ограничениями на ядро  $K(x, s)$  и правую часть  $f(x)$  (см., напр., [3], с. 19, [9], с. 45). В частности (см., напр., [10], с. 8), решение существует и единственно в классе непрерывных на отрезке  $[a, b]$  функций, если ядро непрерывно внутри и на сторонах треугольника, ограниченного прямыми  $s = a$ ,  $x = b$ ,  $x = s$ , а функция  $f(x)$  непрерывна на  $[a, b]$ .

Уравнение (1) содержит интегральный оператор

$$A\varphi(x) = \int_a^x K(x, s)\varphi(s)ds. \quad (2)$$

Ясно, что значения функции  $\psi(x) = A\varphi(x)$  при любом  $x$  определяются значениями функции  $\varphi(s)$  только при  $s < x$ . Интегральные операторы, характеризующиеся этим свойством, называются операторами Вольтерра и широко применяются при описании процессов с последствием и обратной связью (см., напр., [3], с. 22.).

---

<sup>1)</sup>Вито Вольтерра (итал. Vito Volterra; 1860–1940) — итальянский математик и физик.

**2. Метод квадратур.** При численном решении интегральных уравнений входящие в них интегралы обычно заменяют конечными суммами. Согласно методу квадратур интегральные операторы заменяют суммами, полученными с помощью различных квадратурных формул (см., напр., [1], [4], [8]):

$$\int_a^b g(x)dx = \sum_{i=1}^n A_i g(x_i) + R. \quad (3)$$

Здесь  $a \leq x_1 < x_2 < \dots < x_n \leq b$  — узлы,  $A_i$ ,  $i = 1, 2, \dots, n$  — веса, а  $R$  — ошибка аппроксимации квадратурной формулы.

Чтобы применить метод квадратур к решению уравнения (1), необходимо использовать следующие равенства:

$$y(x_i) - \int_a^{x_i} K(x_i, s)y(s)ds = f(x_i), \quad i = 1, 2, \dots, n. \quad (4)$$

Они получаются из исходного уравнения при фиксированных значениях  $x_i$  независимой переменной  $x$ . Узлы сетки  $x_i$  могут быть выбраны специальным образом или заданы заранее, если, например, правая часть  $f$  задана таблицей.

Примем значения  $x_i$  в качестве узлов квадратурной формулы и заменим с ее помощью интеграл в (4) конечной суммой. Получим систему

$$y(x_i) - \sum_{j=1}^i A_j K(x_i, x_j)y(x_j) = f(x_i) + R_i, \quad i = 1, 2, \dots, n, \quad (5)$$

где  $A_j$  — веса квадратурной формулы,  $R_i$  — ошибки аппроксимации. Положим ошибки  $R_i$  малыми и отбросим их. Получим систему линейных алгебраических уравнений:

$$y_i - \sum_{j=1}^i A_j K_{ij} y_j = f_i, \quad i = 1, 2, \dots, n. \quad (6)$$

Здесь  $y_i = \tilde{y}(x_i)$ ,  $f_i = f(x_i)$ ,  $K_{ij} = K(x_i, x_j)$ ,  $\tilde{y}$  — приближение к искомой функции  $y$ . Решение системы уравнений (6) дает приближенные значения искомой функции в узлах  $x_i$ .

Приведем систему (6) к следующему виду:

$$-\sum_{j=1}^{i-1} A_j K_{ij} y_j + (1 - A_i K_{ii}) y_i = f_i, \quad i = 1, 2, \dots, n, \quad (7)$$

подробнее,

$$\begin{pmatrix} 1 - A_1 K_{11} & & & \\ -A_1 K_{21} & 1 - A_2 K_{22} & & \\ \vdots & \vdots & \ddots & \\ -A_1 K_{n1} & -A_2 K_{n2} & \cdots & 1 - A_n K_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}.$$

Хорошо видно, что матрица коэффициентов этой системы — треугольная. Если все диагональные элементы матрицы отличны от нуля, то значения  $y_1, y_2, \dots, y_n$  можно последовательно найти по рекуррентной формуле

$$y_i = (1 - A_i K_{ii})^{-1} \left( f_i + \sum_{j=1}^{i-1} A_j K_{ij} y_j \right), \quad i = 1, 2, \dots, n. \quad (8)$$

Выполнения условий  $(1 - A_i K_{ii}) \neq 0, i = 1, 2, \dots, n$ , можно добиться путем выбора узлов квадратурной формулы и обеспечения достаточной малости коэффициентов  $A_i$ .

Отметим особенность выражения (8), состоящую в росте количества вычислений вместе с номером шага дискретизации из-за увеличения членов суммы, причем значения коэффициентов  $A_j K_{ij}$  при  $y_j$  меняются для каждого  $i$ , что в общем случае не позволяет воспользоваться результатами вычислений на предыдущих шагах. Кроме того, имеются особенности в применении различных квадратурных формул. Например, применение формулы Симпсона должно чередоваться для нечетных узлов с каким-либо другим правилом, например с формулой прямоугольников или формулой трапеций. Возникают сложности при применении формул Гаусса и Чебышева (см., напр., [3], с. 37).

Достаточно простым и во многих случаях эффективным является применение формулы трапеций (см., напр., [5], с. 341). Для равномерной сетки с шагом  $h$  имеем

$$A_1 = A_n = \frac{h}{2}, \quad A_2 = A_3 = \dots = A_{n-1} = h.$$

Тогда формула (8) примет следующий вид:

$$y_i = \left(1 - \frac{h}{2}K_{ii}\right)^{-1} \left(f_i + \frac{h}{2}K_{i1}y_1 + h \sum_{j=2}^{i-1} K_{ij}y_j\right), \quad i = 1, 2, \dots, n. \quad (9)$$

**3. Компьютерная программа.** Напишем на языке Matlab функцию `Volt_II_Rect.m`, реализующую вычисления по этой формуле.

```
% Функция для решения уравнения Вольтерра второго
% рода методом квадратур. Используется формула
% трапеций с равноотстоящими узлами
% Входные данные: K - ядро уравнения, f - правая
% часть (задаются аналитически), a - начало
% отрезка интегрирования, b - конец отрезка, h - шаг
% сетки. Результат - вектор y приближений к
% решению в узлах сетки
% Автор: Валиева М.
function [ y ] = Volt_II_Rect(K,f,a,b,h)
format long
x=[a:h:b]
y(1)=f(x(1));
for i=2:1:fix((b-a)/h+1)
    s=0;
    if i>2
        for j=2:1:(i-1)
            s=K(x(i),x(j))*y(j)+s;
        end
    end
    y(i)=((1-h/2*K(x(i),x(i)))^(-1))*((f(x(i)))+...
        (h/2)*K(x(i),x(1))*y(1)+h*s));
end
```

**4. Пример.** Решим с помощью функции `Volt_II_Rect.m` упражнение 1.9, с. 39, из книги [3]. Дано уравнение

$$y(x) - \int_0^x e^{-(x-s)}y(s)ds = e^{-x}, \quad x \in [0, 1]. \quad (10)$$

Таблица 1. Результаты приближенного решения уравнения (10) методом квадратур, основанным на применении квадратурной формулы трапеций на равномерных сетках с двумя разными шагами  $h$ .

$x$	$y$ при $h = 0.25$	$y$ при $h = 0.05$
0.00	1.000000000000000	1.000000000000000
0.25	1.00131529252038	1.00005210423069
0.50	1.00263231503517	1.00010421117622
0.75	1.00395106981982	1.00015632083675
1.00	1.00527155915278	1.00020843321241

Точное решение этого уравнения  $y \equiv 1$ . Надо найти приближенное решение этого уравнения методом квадратур, основанным на использовании формулы трапеций с равномерной сеткой и сравнить с точным. На языке Matlab сценарий решения этого упражнения выглядит следующим образом.

```
% Сценарий решения упражнения 1.9, с. 39, из книги
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>
% Автор: Валиева М.
close all
clear all
clc
format short;
a=0; % начало отрезка интегрирования
b=1; % конец отрезка интегрирования
K=@(x,s)exp(-(x-s)); % ядро уравнения
f=@(x)exp(-x); % правая часть уравнения
y_exact=@(x) 1; % точное решение
h=0.25; % шаг сетки
x = a:h:b; % вектор значений переменной в узлах сетки
n=size(x,2);
y1 = zeros(1,n); % вектор значений точного решения
for j=1:n
    y1(j)=y_exact(x(j));
end;
plot(x,y1);
hold on;
y_approx=Volt_II_Rect(K,f,a,b,h)
```



```

y2=transp(y_approx);
plot(x,y2,'or');
axis([a,b,0.99,1.01]);
h=0.05;
x = a:h:b;
y_approx=Volt_II_Rect(K,f,a,b,h)
y3=transp(y_approx);
plot(x,y3,'sr');
xlabel('x');
ylabel('y');
hold off;

```

С помощью этого сценария проведем два численных эксперимента: найдем приближенные решения уравнения на сетках с шагом  $h = 0.05$  и  $h = 0.25$ . Результаты счета представлены в таблице 1 и на рис. 1.

### Упражнения

- 1) Объясните, почему точность вычислений падает при приближении к правому концу отрезка интегрирования.
- 2) Нарисуйте график зависимости от  $h$  нормы относительной ошибки решения в более широком диапазоне изменений шага.
- 3) С помощью функции `Volt_II_Rect.m` найдите приближенное решение уравнения

$$y(x) = e^{x^2} + \int_0^x e^{x^2-s^2} y(s) ds, \quad x \in [0, 1]. \quad (11)$$

Его точное решение  $y(x) = e^{x^2+x}$  (см. пример 1.2., с. 26, [3]). Исследуйте зависимость точности решения от  $h$ .

- 4) Модифицируйте функцию `Volt_II_Rect.m` для работы с неравномерной сеткой. Найдите приближенные решения уравнения

$$y(x) = (1 - xe^{2x}) \cos 1 - e^{2x} \sin 1 + \int_0^x (1 - (x-s)e^{2x}) y(s) ds,$$

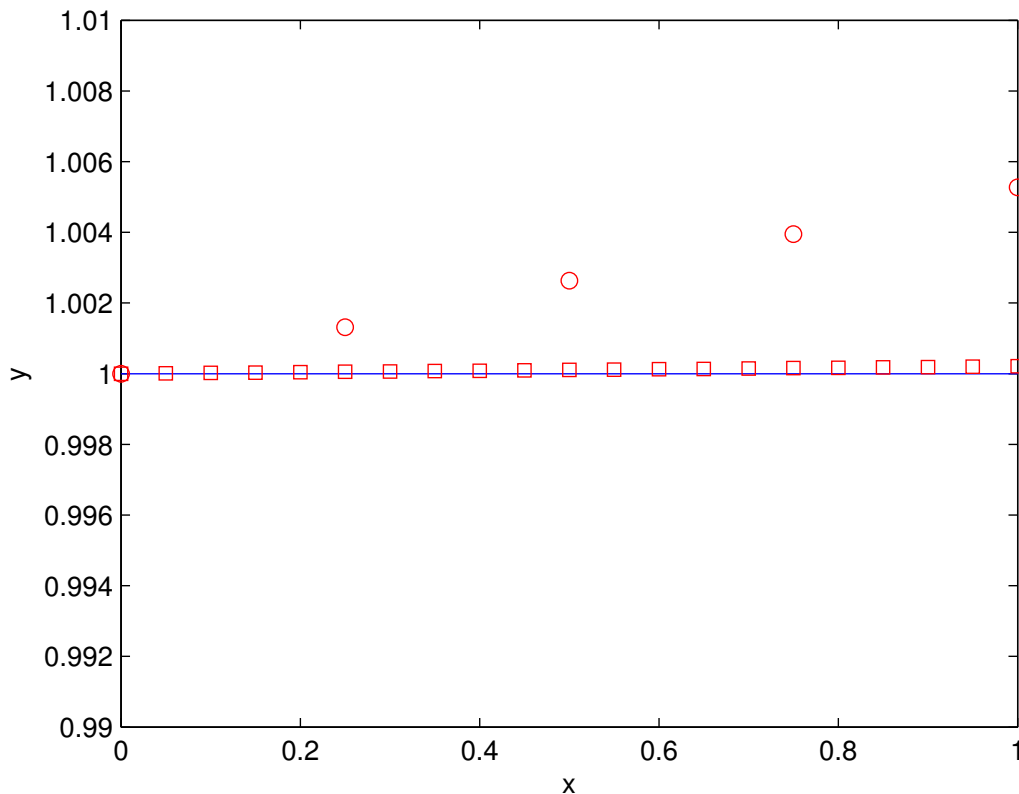


Рис. 1. Результаты приближенного решения уравнения (10) методом квадратур, основанным на применении квадратурной формулы трапеций с равномерной сеткой. Непрерывной линией обозначено точное решение, кружочками — приближенное решение при шаге сетки  $h = 0.25$ , квадратиками — при  $h = 0.005$ .

где  $x \in [0, 2.5]$ . Его точное решение  $y(x) = e^x(\cos(e^x) - e^x \sin(e^x))$  (см. пример 1.11., с. 43, [3]). Исследуйте зависимость точности решения и скорости работы функции от распределения узлов сетки.

- 5) Напишите функцию, реализующую метод квадратур на основе квадратурной формулы Симпсона (см., напр., [5], с. 342). В этом случае в формуле (3) полагают:

$$\begin{aligned}
 A_1 &= A_{2m+1} = \frac{h}{3}, \\
 A_2 &= A_4 = \dots = A_{2m} = \frac{4h}{3}, \\
 A_3 &= A_5 = \dots = A_{2m-1} = \frac{2h}{3},
 \end{aligned} \tag{12}$$

$$x_i = a + h(i - 1), \quad i = 1, 2, \dots, n, \quad h = \frac{b - a}{n - 1},$$

где  $n = 2m + 1$ ,  $m = 1, 2, \dots$ . Решите с помощью новой функции уравнение (11) и сравните эффективность использования формул Симпсона и трапеций.

## § 2. Метод квадратур для нелинейного уравнения

**1. Описание метода.** Нелинейное уравнение Вольтерра II рода имеет следующий вид:

$$y(x) - \int_a^x K(x, s, y(s)) ds = f(x), \quad x \in [a, b]. \quad (1)$$

Отметим, что ядро  $K(x, s, y(s))$  интегрального уравнения зависит от искомой функции  $y(x)$ . Предположим, что ядро и правая часть этого уравнения таковы, что его решение существует и единственно в классе непрерывных на отрезке  $[a, b]$  функций. Построим метод приближенного решения уравнения (1), основанный на аппроксимации интеграла с помощью квадратурной формулы.

Пусть на отрезке  $[a, b]$  задана сетка с узлами  $x_i$ ,  $i = 1, 2, \dots, n$ . Зафиксируем в уравнении (1) значения переменной  $x$  в узлах сетки. Получим следующие равенства:

$$y(x_i) - \int_a^{x_i} K(x_i, s, y(s)) ds = f(x_i), \quad i = 1, 2, \dots, n. \quad (2)$$

Заменяем интегралы в этих равенствах конечными суммами. Получим систему нелинейных рекуррентных соотношений

$$y_i - \sum_{j=1}^i A_j K_{ij}(y_j) = f_i, \quad i = 1, 2, \dots, n. \quad (3)$$

Здесь  $A_j$  — веса квадратурной формулы,  $f_i = f(x_i)$ ,  $y_i = \tilde{y}(x_i)$ ,  $\tilde{y}$  — приближение к искомой функции  $y$ ,  $K_{ij}(y_j) = K(x_i, x_j, \tilde{y}(x_j))$ . Соотношения (3) позволяют находить значения  $y_1, y_2, \dots, y_n$  приближенного решения в узлах сетки путем последовательного решения  $n$  нелинейных уравнений:

$$y_i - A_i K_{ii}(y_i) = f_i + \sum_{j=1}^{i-1} A_j K_{ij}(y_j), \quad i = 1, 2, \dots, n. \quad (4)$$

**2. Пример.** Решим предложенным выше методом нелинейное уравнение

$$y(x) - \int_0^x e^{-(x-s)} y^2(s) ds = e^{-x}, \quad x \in [0, 0.1]. \quad (5)$$

Точное решение этого уравнения  $y \equiv 1$  (см. пример 1.10, с. 40, [3]). Это уравнение относится к более узкому классу нелинейных уравнений с подынтегральными выражениями вида  $K(x, s)y^2(s)$ . Равенства (2) принимают следующий вид:

$$y(x_i) - \int_0^{x_i} e^{-(x_i-s)} y^2(s) ds = e^{-x_i}, \quad i = 1, 2, \dots, n. \quad (6)$$

Пусть узлы  $x_i$  образуют на отрезке  $[a, b]$  равномерную сетку с шагом  $h$ . Заменяем интегралы в (6) конечными суммами с помощью формулы трапеций. Согласно (4) получим следующую систему расчетных соотношений:

$$y_1 = f_1, \quad y_i - \frac{h}{2} K_{ii} y_i^2 = f_i + \sum_{j=1}^{i-1} h K_{ij} y_j^2, \quad i = 2, 3, \dots, n. \quad (7)$$

Заметим, что в данном случае  $K_{ii} = 1$ . Обозначим

$$c_i = f_i + \sum_{j=1}^{i-1} h K_{ij} y_j^2, \quad i = 2, 3, \dots, n.$$

Для вычисления значений  $y_i$ , начиная со второго, получим квадратные уравнения:

$$\frac{h}{2} y_i^2 - y_i + c_i = 0, \quad i = 2, 3, \dots, n.$$

Тогда

$$y_1 = f_1, \quad y_i = \frac{1 \pm \sqrt{1 - 4 \frac{h}{2} c_i}}{h}, \quad i = 2, 3, \dots, n. \quad (8)$$

Оставим знак « $-$ » перед корнем для того, чтобы выбрать одно из решений.

**3. Компьютерные программы.** Напишем на языке Matlab функцию `Nonlin_Volt_II_Rect.m`, реализующую вычисления по формулам (8).

```
% Функция для решения нелинейного уравнения
% Вольтерра второго рода методом квадратур. Используется
```

Таблица 2. Результаты приближенного решения уравнения (5) методом квадратур, основанным на применении квадратурной формулы трапеций с равномерной сеткой. Шаг сетки  $h = 0.02$ .

$x$	точное решение	приближенное решение
0.00	1.00	1.000000000
0.02	1.00	1.010003722
0.04	1.00	1.010208603
0.06	1.00	1.010417709
0.08	1.00	1.010631130
0.10	1.00	1.010848956

```
% формула трапеций с равноотстоящими узлами
% Входные данные: K - ядро уравнения, f - правая
% часть (задаются аналитически), a - начало
% отрезка интегрирования, b - конец отрезка, h - шаг
% сетки. Результат - вектор y приближений к
% решению в узлах сетки
% Автор: Самирханов Р.
function [y] = Nonlin_Volt_II_Rect(a,b,h,K,f)
%задаем x(i)
x=[a:h:b];
n=size(x,2);
%вычисляем y(i)
y(1)=f(x(1)); for i=2:n
    s=0;
    for j=1:1:(i-1)
        s=s+h*K(x(i),x(j))*(y(j)*y(j));
    end
    y(i)=(1-sqrt(1-4*(h/2)*(f(x(i))+s)))/h;
end
end
```

Напишем функцию `plots.m` для сравнения приближенного решения с точным. Эта функция результаты вычислений записывает в файл и представляет графически.

```
% Функция для сравнения точного и приближенного
% решений уравнения. Входные данные: y_exact -
% аналитически заданное точное решение,
```

```
% y_аpprox - вектор приближенного решения,  
% a - начало отрезка интегрирования, b - конец  
% отрезка, h - шаг сетки.  
% Автор: Самирханов Р.  
function [] = plots(a,b,h,y_еxact,y_аpprox)  
x = a:h:b;  
n=size(x,2);  
%точное решение  
for i=1:n;  
    y(i)=y_еxact(x(i));  
end  
%матрица для записи результатов в файл  
A(:,1)=x;  
A(:,2)=y;  
A(:,3)=y_аpprox;  
dlmwrite('C:matrica.txt',...  
    A,'delimiter',' ','newline','pc',...  
    'precision', 10);  
plot(x,y_аpprox,'or');  
hold on;  
plot(x,y,'b');  
axis([ a,b,1-0.1,1+0.1]);  
xlabel('x');  
ylabel('y');  
hold off;  
end
```

Теперь напишем на языке Matlab сценарий решения уравнения (5).

```
% Сценарий решения упражнения 1.10, с. 40, из книги  
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>  
% Автор: Самирханов Р.  
close all  
clear all  
clc  
a=0;  
b=0.1;  
h=0.02;
```

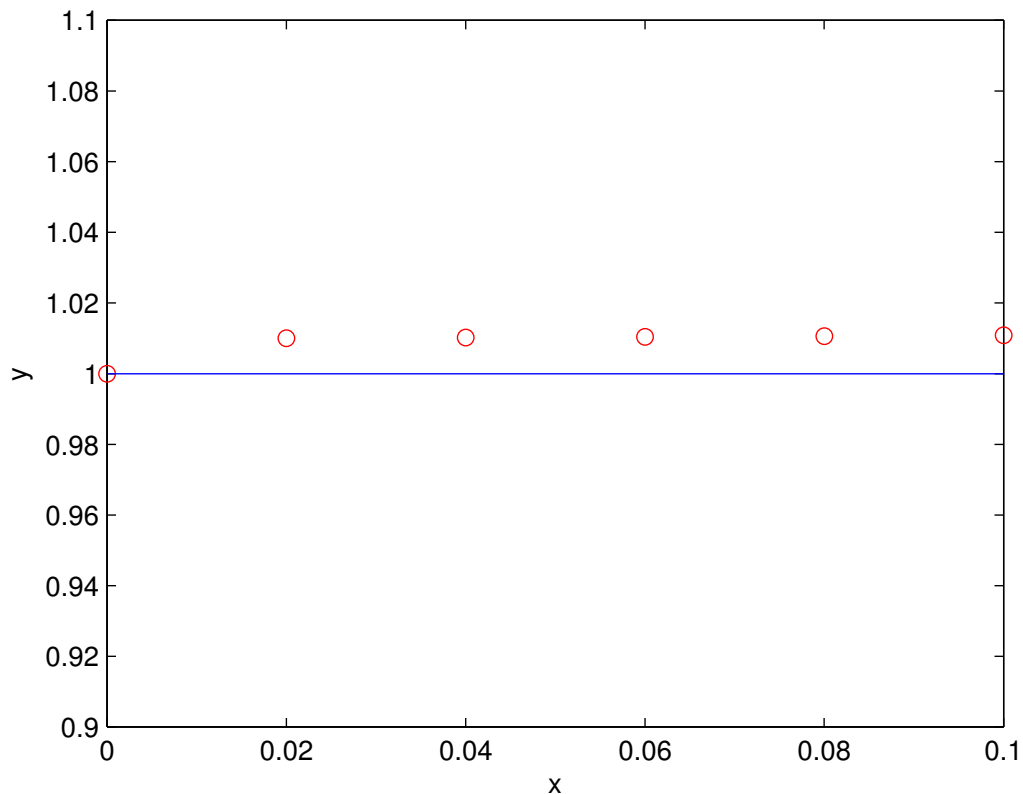


Рис. 2. Результаты приближенного решения уравнения (5) методом квадратур, основанным на применении квадратурной формулы трапеций с равномерной сеткой. Непрерывной линией обозначено точное решение, кружочками — приближенное решение при шаге сетки  $h = 0.02$ .

```
format long;  
K = @(x,s)exp(-(x-s));  
f = @(x)exp(-x);  
y_approx=Nonlin_Volt_II_Rect(a,b,h,K,f);  
y_exact=@(x)1;  
plots(a,b,h,y_exact,y_approx);
```

С помощью этого сценария найдем приближенное решение уравнения на сетке с шагом  $h = 0.02$ . Результаты счета представлены в таблице 2 и на рис. 2.

### Упражнения

- 1) Объясните, зависимость точности вычислений от номера узла сетки.



- 2) Нарисуйте график зависимости от  $h$  нормы относительной ошибки решения в более широком диапазоне изменений шага.
- 3) Как изменится решение, если в формуле (8) перед корнем оставить знак «+»?
- 4) Напишите функцию реализующую вычисления по общим формулам (4). Решите с помощью этой функции уравнение (5). Проанализируйте точность вычислений.

### § 3. Метод простой итерации

**1. Описание метода.** Запишем линейное уравнение Вольтерра II рода в удобном для применения метода простой итерации виде:

$$y(x) = f(x) + \int_a^x K(x, s)y(s)ds, \quad x \in [a, b]. \quad (1)$$

Построим последовательность функций  $y_k(x)$ ,  $k = 0, 1, 2, \dots$ , с помощью рекуррентного соотношения

$$y_k(x) = f(x) + \int_a^x K(x, s)y_{k-1}(s)ds, \quad k = 1, 2, \dots \quad (2)$$

Если правая часть  $f(x)$  непрерывна на отрезке  $[a, b]$ , а ядро  $K(x, s)$  непрерывно в замкнутом треугольнике  $a \leq s \leq x \leq b$ , эта последовательность сходится при любом начальном приближении  $y_0(x)$  (см., напр., [2], с. 613). Скорость сходимости зависит от свойств ядра и правой части уравнения. Ясно, что число итерационных шагов для получения аппроксимации необходимой точности зависит от степени близости начального приближения к искомому решению. В качестве начального приближения часто выбирают  $f(x)$ , если нет дополнительной информации о решении.

При численной реализации итерационных методов интеграл вычисляется посредством квадратурных формул. Воспользуемся квадратурной формулой трапеций с равномерной сеткой и шагом  $h$ . Узлы сетки обозначим  $x_i$ ,  $i = 0, 1, \dots, n$ , пусть  $K_{ij} = K(x_i, x_j)$ ,  $y_{ki} = y_k(x_i)$ . Получим расчетное выражение

$$y_{k+1}(x_i) = f(x_i) + \int_0^{x_i} K(x_i, s)y_k(s)ds \approx \quad (3)$$

$$\approx f(x_i) + \frac{h}{2} [K_{i0}y_{k0} + 2(K_{i1}y_{k1} + K_{i2}y_{k2} + \dots + K_{i,i-1}y_{k,i-1}) + K_{ii}y_{ki}],$$

где  $i = 0, 1, \dots, n$ . Для окончания итерационного процесса, как обычно, будем использовать условие

$$\frac{\|y_k - y_{k-1}\|}{\|y_k\|} \leq \varepsilon, \quad (4)$$

где  $\|y\| = \max_{a \leq x \leq b} |y(x)|$ ,  $\varepsilon$  — заданная относительная ошибка. Данное условие означает, что в процессе решения необходимо сравнивать результаты, полученные для двух смежных итерационных шагов; близость полученных при этом приближений свидетельствует о достигнутой точности. Таким образом, количество итерационных шагов зависит также от требований к точности результата.

**2. Компьютерные программы.** Напишем на языке Matlab функцию `inK.m`, реализующую вычисления по формуле (3).

```
% Функция для вычисления очередного приближения
% к решению уравнения Вольтерра второго рода
% в ходе метода простой итерации. Используется формула
% трапеций с равноотстоящими узлами.
% Входные данные: K - аналитически заданное ядро уравнения
% x - сетка, на которой вычисляется интеграл,
% h - шаг сетки, n - число узлов сетки.
% y - вектор значений в узлах сетки приближения к решению,
% вычисленный на предыдущем шаге итерационного процесса.
% Результат - вектор новых значений приближения в узлах сетки.
% Автор: Сергеев П.
function [yk] = CalcInt(y,h,x,n,K,f)
    yk = y;
    for i = 1 : n
        yk(i) = 0;
        for j = 1 : i
            yk(i) = yk(i) + 2*K(x(i),x(j))*y(j);
        end
        yk(i) = yk(i) - K(x(i),x(1))*y(1) - K(x(i),x(i))*y(i);
        yk(i) = f(x(i)) + yk(i)*h/2;
    end
end
end
```

Напишем на языке Matlab функцию `IterVolt.m`, предназначенную для приближенного решения уравнения (1) методом простой итерации.

```
% Функция для решения уравнения Вольтерра второго рода
% методом простой итерации. Используется формула
```

```

% трапеций с равноотстоящими узлами.
% Входные данные: K - ядро уравнения, f - правая
% часть (задаются аналитически), x - сетка, на которой
% строится решение, h - шаг сетки, eps - заданная точность.
% Результат - вектор yk приближений к
% решению в узлах сетки. Iter - количество итераций, за
% которое была достигнута требуемая точность
% Автор: Сергеев П.
function [yk,iter] = IterVolt(x,h,eps,f,K)
    n = numel(x);
    y = f(x);
    yk = CalcInt(y,h,x,n,K,f);
    iter = 0;
    while norm(yk-y,inf)/norm(yk,inf) > eps
        y = yk;
        yk = CalcInt(y,h,x,n,K,f);
        iter = iter + 1;
    end
end
end

```

**3. Пример.** Решим с помощью функции `Iter_Volt.m` упражнение 1.19, с. 73, из книги [3]. Дано уравнение

$$y(x) = 1 + \int_0^x y(s) ds, \quad x \in [0, 7]. \quad (5)$$

Точное решение этого уравнения  $y(x) = e^x$ . Надо найти приближенное решение этого уравнения методом последовательных приближений, основанным на использовании формулы трапеций с равномерной сеткой. Шаг сетки  $h = 0.07$ , относительная погрешность решения  $\varepsilon = 10^{-3}$ . На языке Matlab сценарий решения этого упражнения выглядит следующим образом.

```

% Сценарий решения упражнения 1.19, с. 73, из книги
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>
% Автор: Сергеев П.
clear all
close all

```

```

clc

f = @(x) x*0 + 1;
K = @(x,s) x*0 + s*0 + 1;
a = 0;
b = 7;
h = 0.07;
eps = 1e-03;

y_exact = @(x) exp(x);
x = a : h : b;
[y_approx,iter] = IterVolt(x,h,eps,f,K);
y=y_exact(x);
plot(x,y,'o',x,y_approx,'r');
er = norm(y-y_approx,inf)/norm(y,inf);
xlabel('x');
ylabel('y');

```

Результаты счета представлены на рис. 3.

### Упражнения

- 1) На каком шаге итерационного процесса была достигнута требуемая точность?
- 2) Исследуйте, как зависит от шага сетки  $h$  точность вычислений и скорость сходимости итерационного процесса.
- 3) С помощью функции `Iter_Volt.m` найдите приближенное решение уравнения

$$y(x) = x - \int_0^x (x-s)y(s)ds, \quad x \in [0, 2\pi].$$

Его точное решение  $y(x) = \sin x$  (см. пример 1.18., с. 72, [3]).

- 4) Напишите функцию, реализующую метод последовательных приближений, на основе квадратурной формулы Симпсона (12), с. 9. Решите с помощью этой функции уравнения из этого параграфа и сравните эффективность применения метода Симпсона с использованием метода трапеций.

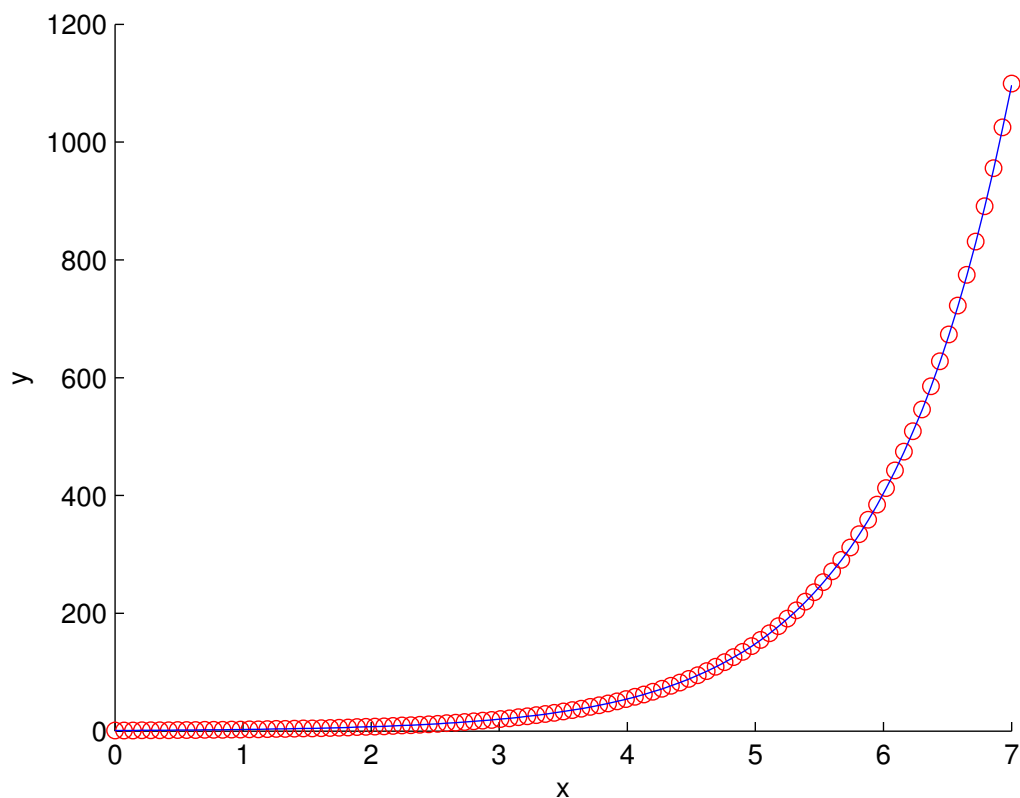


Рис. 3. Результаты приближенного решения уравнения (5) методом простой итерации, основанным на применении квадратурной формулы трапеций с равномерной сеткой. Непрерывной линией обозначено точное решение, кружочками — приближенное решение при шаге сетки  $h = 0.07$ .

- 5) Напишите функцию, предназначенную для решения методом простой итерации нелинейных уравнений Вольтерра второго рода. Найдите приближенное решение уравнения

$$y(x) = \int_0^x \frac{1 + y^2(s)}{1 + s^2} ds, \quad x \in [0, 10].$$

Точное решение  $y(x) = x$  (см. пример 1.20., с. 74, [3]).

---

---

## ГЛАВА 2

# Методы решения уравнений Фредгольма II рода

### § 1. Метод квадратур

1. **Линейное уравнение Фредгольма II рода** имеет следующий вид:<sup>1)</sup>

$$y(x) - \lambda \int_a^b K(x, s)y(s)ds = f(x), \quad x \in [a, b]. \quad (1)$$

Здесь  $y(x)$  — неизвестная функция,  $K(x, s)$  — ядро интегрального уравнения,  $f(x)$  — свободный член (правая часть) интегрального уравнения. Для удобства анализа в интегральном уравнении (1) по традиции принято выделять числовой параметр  $\lambda$ , который называют параметром интегрального уравнения.

На вопросы существования решения уравнения (1) отвечает классическая теория Фредгольма (см., напр., [10], с. 77. [14], с. 48). Она применима, в частности, для непрерывных в прямоугольнике  $[a, b] \times [a, b]$  ядер. Будем считать, что правая часть уравнения (1) непрерывна на отрезке  $[a, b]$ , а его решение будем разыскивать в классе непрерывных на  $[a, b]$  функций. Если однородное уравнение ( $f(x) \equiv 0$ ) имеет только тривиальное решение, то значение параметра  $\lambda$  называется правильным или регулярным. Тогда у неоднородного уравнения при любой правой части  $f(x)$  существует единственное решение. Всюду далее в этой главе будем считать это условие выполненным.

Приложения интегральных уравнений Фредгольма второго рода весьма разнообразны (см., напр., [10], с. 167. [3], с. 143): граничные задачи теории потенциала, граничные задачи для обыкновенных дифференциальных уравнений, граничные задачи теории упругости и т.д.

---

<sup>1)</sup>Эрик Ивар Фредгольм (Erik Ivar Fredholm; 1866–1927) — Шведский математик.

**2. Описание метода.** Найдем приближенное решение уравнения (1) методом квадратур. Построим на отрезке  $[a, b]$  сетку с узлами  $x_1, x_2, \dots, x_n$ . Запишем уравнение (1) в узлах сетки:

$$y(x_i) - \lambda \int_a^b K(x_i, s)y(s)ds = f(x_i), \quad i = 1, 2, \dots, n. \quad (2)$$

Аппроксимируем интегралы в равенствах (2) конечными суммами с помощью одной из квадратурных формул:

$$y_i - \lambda \sum_{j=1}^n A_j K_{ij} y_j = f_i, \quad i = 1, 2, \dots, n. \quad (3)$$

Здесь  $y_i = \tilde{y}(x_i)$ ,  $f_i = f(x_i)$ ,  $K_{ij} = K(x_i, x_j)$ ,  $\tilde{y}$  — приближение к искомой функции  $y$ ,  $A_j$  — веса квадратурной формулы.

Решение системы уравнений (3) дает приближенные значения искомой функции в узлах  $x_i$ . По ним с помощью интерполяции можно построить приближенное решение интегрального уравнения (1) на всем отрезке  $[a, b]$ .

Пусть  $\lambda = 1$ , а сетка  $x_1, x_2, \dots, x_n$  — равномерная с шагом  $h$ . Используем квадратурную формулу трапеций. Тогда система линейных алгебраических уравнений (3) примет следующий вид:

$$y_i - h \sum_{j=1}^n w_j K_{ij} y_j = f_i, \quad i = 1, 2, \dots, n, \quad (4)$$

где  $w_1 = w_n = 1/2$ ,  $w_j = 1$  при  $j = 2, 3, \dots, n-1$ .

**3. Компьютерная программа.** Напишем на языке Matlab функцию `Fred_II_Rect.m`, реализующую вычисления по формуле (4).

```
% Функция для решения уравнения Фредгольма второго
% рода методом квадратур. Используется формула
% трапеций с равноотстоящими узлами.
% Входные данные: K - ядро уравнения, f - правая
% часть (задаются аналитически), a - начало
% отрезка интегрирования, b - конец отрезка, h - шаг
% сетки. Результат - вектор y приближений к
```



```

% решению в узлах сетки
% Автор: Файрушин Р.
function [y] = Fred_II_Rect(K,f,a,b,h)
x = a:h:b;
n=size(x,2);
wt=1/2;
wj=1;
A = zeros(n);
for i=1:n
    A(i,1)= -h*wt*K(x(i),x(1));
    for j=2:n-1
        A(i,j)= -h*wj*K(x(i),x(j));
    end;
    A(i,n)= -h*wt*K(x(i),x(n));
    A(i,i)= A(i,i)+ 1;
end;
B = zeros(n,1);
for j=1:n
    B(j,1) = f(x(j));
end;
y = A\B;

```

**4. Пример.** Выполним с помощью функции `Fred_II_Rect.m` упражнение 3.9, с. 162, из книги [3]. Дано уравнение (1) с границами отрезка интегрирования  $a = -\pi$  и  $b = \pi$ , параметром  $\lambda = 3/(10\pi)$ , ядром

$$K(x, s) = \frac{1}{0.64 \cos^2 \left( \frac{x+s}{2} \right) - 1}$$

и правой частью  $f(x) = 25 - 16 \sin^2(x)$ . Точное решение этого уравнения  $y(x) = 17/2 + (128/17) \cos(2x)$ . Надо найти приближенное решение этого уравнения методом квадратур, основанным на использовании формулы трапеций с равномерной сеткой с шагом  $h = \pi/18$ , и сравнить с точным.

На языке Matlab сценарий решения этой задачи выглядит следующим образом. Для сравнения приближенного решения с точным используется функция `plots.m` (см. с. 13).

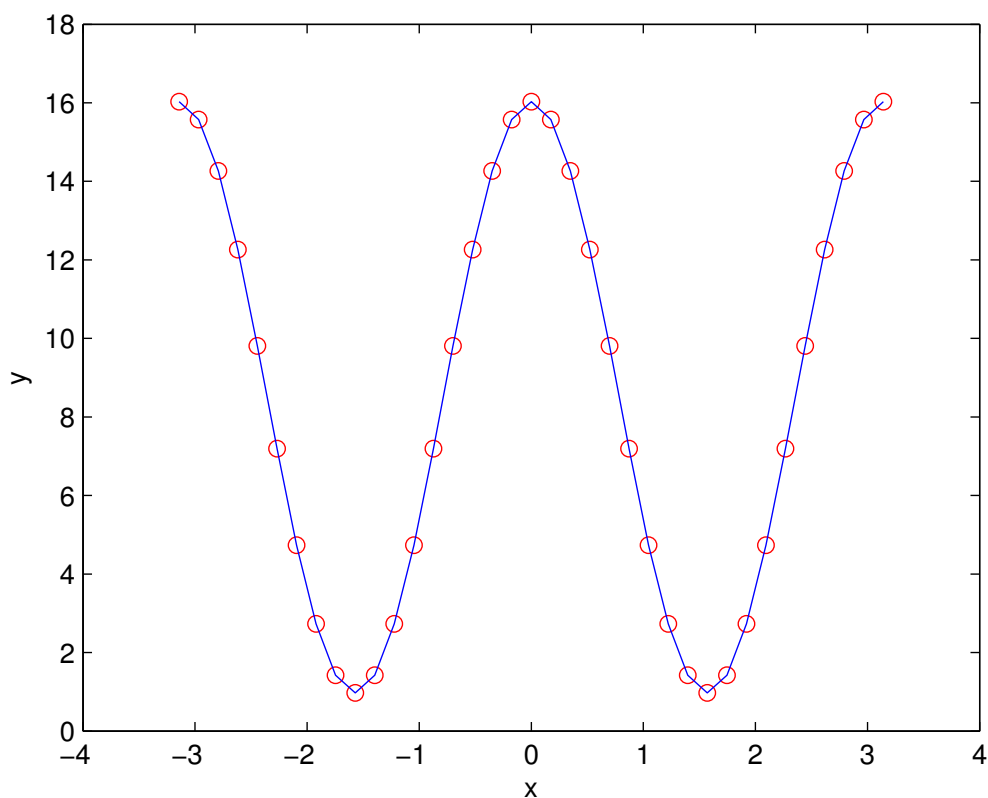


Рис. 1. Результаты решения примера 4, с. 24. Непрерывной линией обозначено точное решение, кружочками — приближенное решение.

```
% Сценарий решения задачи 3.9, с. 162, из книги
% Верлань А.Ф., Сизиков В.С. <<Интегральные уравнения...>>
% Автор: Файрушин Р.
close
all clear
all clc
format long;
h=pi/18;
a=-pi;
b =pi;
lambda = 3/(10*pi);
K =@(x1,s)1/(0.64*(cos((x1+s)/2))^2-1)*lambda;
f = @(x1)25-16*(sin(x1))^2;
y_exact= @(x1)17/2+128/17*cos(2*x1);
y_approx=Fred_II_Rect(K,f,a,b,h);
```

Таблица 1. Результаты решения примера 4, с. 24.

$x$	точное решение	приближенное решение
0.000000000000000	16.02941176470588	16.02941176470589
0.17453292519943	15.57533267415272	15.57533267415272
1.57079632679490	00.97058823529412	00.97058823529412
2.96705972839036	15.57533267415272	15.57533267415273
3.14159265358979	16.02941176470588	16.02941176470588

```
plots(a,b,h,y_exact,y_approx);
```

Результаты счета представлены в таблице 1 и на рис. 1.

### Упражнения

- 1) Исследуйте, как зависит от шага сетки  $h$  точность вычислений.
- 2) С помощью функции `Fred_II_Rect.m` найдите приближенное решение уравнения

$$y(x) = \frac{5}{6}x + \frac{1}{2} \int_0^1 xsy(s)ds, \quad x \in [0, 1]. \quad (5)$$

Его точное решение  $y(x) = x$  (см. пример 3.8, с. 158, [3]).

- 3) Напишите функцию, реализующую метод квадратур, на основе квадратурной формулы Симпсона (12), с. 9. Решите с помощью этой функции уравнение

$$y(x) + \int_0^1 xe^{xs}y(s)ds = e^x, \quad x \in [0, 1].$$

Точное решение этого уравнения есть  $y \equiv 1$  (см., пример 1, с. 344, [5]). Исследуйте, как зависит от шага сетки  $h$  точность вычислений. Сравните эффективность применения формулы Симпсона с использованием формулы трапеций.

## §2. Метод вырожденных ядер

**1. Решение уравнения с вырожденным ядром.** Ядро интегрального уравнения называется вырожденным, если оно имеет следующий вид:

$$K(x, s) = \sum_{i=1}^m \alpha_i(x) \beta_i(s). \quad (1)$$

Будем считать, что  $\{\alpha(x)\}_{i=1}^m$ ,  $\{\beta(x)\}_{i=1}^m$  — линейно независимые системы непрерывных на отрезке  $[a, b]$  функций.

Подставим ядро (1) в уравнение Фредгольма второго рода (см. с. 22). Получим уравнение с вырожденным ядром:

$$y(x) - \lambda \int_a^b \left[ \sum_{i=1}^m \alpha_i(x) \beta_i(s) \right] y(s) ds = f(x), \quad x \in [a, b]. \quad (2)$$

Поменяем в (2) местами операции интегрирования и суммирования:

$$y(x) - \lambda \sum_{i=1}^m \alpha_i(x) \int_a^b \beta_i(s) y(s) ds = f(x), \quad x \in [a, b]. \quad (3)$$

Введем обозначения для интегралов в (3):

$$c_i = \int_a^b \beta_i(s) y(s) ds, \quad i = 1, 2, \dots, m. \quad (4)$$

Из равенства (3) получим следующее представление решения уравнения с вырожденным ядром (2):

$$y(x) = f(x) + \lambda \sum_{i=1}^m c_i \alpha_i(x), \quad x \in [a, b]. \quad (5)$$

Отметим, что константы  $c_i$  неизвестны, они являются интегралами от искомой функции. Для их вычисления построим систему линейных алгебраических уравнений. Подставим (5) в (2). Получим равенство

$$\sum_{i=1}^m \alpha_i(x) \left\{ c_i - \int_a^b \beta_i(s) \left[ f(s) + \lambda \sum_{j=1}^m c_j \alpha_j(s) \right] ds \right\} = 0, \quad x \in [a, b]. \quad (6)$$

Система функций  $\{\alpha(x)\}_{i=1}^m$  линейно независимая, следовательно, все коэффициенты в этой линейной комбинации равны нулю:

$$c_i - \int_a^b \beta_i(s) \left[ f(s) + \lambda \sum_{j=1}^m c_j \alpha_j(s) \right] ds = 0, \quad i = 1, 2, \dots, m, \quad (7)$$

или

$$c_i - \lambda \sum_{j=1}^m c_j \int_a^b \alpha_j(s) \beta_i(s) ds = \int_a^b \beta_i(s) f(s) ds, \quad i = 1, 2, \dots, m. \quad (8)$$

Обозначим

$$a_{ij} = \int_a^b \alpha_j(s) \beta_i(s) ds, \quad i, j = 1, 2, \dots, m. \quad (9)$$

$$f_i = \int_a^b \beta_i(s) f(s) ds, \quad i = 1, 2, \dots, m. \quad (10)$$

Запишем равенства (8) в виде системы линейных алгебраических уравнений относительно неизвестных  $c_i$ :

$$c_i - \lambda \sum_{j=1}^m a_{ij} c_j = f_i, \quad i = 1, 2, \dots, m, \quad (11)$$

подробнее,

$$\begin{pmatrix} 1 - \lambda a_{11} & -\lambda a_{12} & \cdots & -\lambda a_{1m} \\ -\lambda a_{21} & 1 - \lambda a_{22} & \cdots & -\lambda a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ -\lambda a_{m1} & -\lambda a_{m2} & \cdots & 1 - \lambda a_{mm} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \cdots \\ c_m \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \cdots \\ f_m \end{pmatrix}. \quad (12)$$

Если число  $\lambda$  регулярное, то определитель матрицы этой системы отличен от нуля, и она имеет единственное решение.

Таким образом, алгоритм решения интегрального уравнения с вырожденным ядром (2) состоит в вычислении интегралов (9), (10) и решении системы линейных алгебраических уравнений (12). Решение уравнения (2) затем записывается аналитически в виде (5).

**2. Компьютерная программа.** Реализуем описанный алгоритм в виде функции `Degenerate_Fred_II.m` на языке Matlab.

```
% Функция для решения уравнения Фредгольма второго
% рода с вырожденным ядром.
% Входные данные: lambda - параметр уравнения,
% alpha, beta - ядро уравнения, f - правая
% часть (задаются аналитически), a - начало
% отрезка интегрирования, b - конец отрезка.
% Результат: y - символьное выражения решения.
% Автор: Хайруллин Р.
function [ y ] = Degenerate_Fred_II(a,b,lambda,alpha,beta,f)
syms x
format long
n = length(alpha)
r = zeros(n,1); % правая часть СЛАУ
M = zeros(n); % матрица СЛАУ
for i=1:n
    r(i) = int(beta(i)*f, x, a, b);
    for j=1:n
        M(i,j) = -lambda*int(beta(i)*alpha(j), x, a, b);
    end
end for i=1:n
    M(i,i) = 1 + M(i,i);
end
c = M\r % решение СЛАУ
y_x = 0;
for i=1:n
    y_x = y_x + c(i)*alpha(i);
end
y = y_x*lambda + f;
end
```

**3. Пример.** Решим с помощью функции `Degenerate_Fred_II.m` уравнение с вырожденным ядром

$$y(x) - \int_0^1 (1 + 2xs)y(s)ds = -\frac{1}{6}x - \frac{1}{2}, \quad x \in [0, 1].$$

Точное решение этого уравнения (см. пример 3.12, с. 171, [3]):

$$y(x) = x + \frac{1}{2}.$$

Сценарий решения этого уравнения на языке Matlab выглядит следующим образом.

```
% Сценарий решения задачи 3.12, с. 171, из книги
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>
% Автор: Хайруллин Р.
close all
clear all
clc
syms x y
a = 0;
b = 1;
alpha = [1 2*x];
beta = [1 x];
lambda = 1;
f = (-x/6) - 1/2;
y=Degenerate_Fred_II(a,b,lambda,alpha,beta,f)
```

Результатом работы этого сценария является символьное выражение точного решения исходного уравнения:  $y = 1/2+x$ .

**4. Аппроксимация ядра вырожденным.** Пусть ядро интегрального уравнения невырожденное, но достаточно гладкое. Тогда его можно аппроксимировать вырожденным, например, разложив в ряд Тейлора, а полученное уравнение решить описанным выше способом. В этом состоит метод вырожденных ядер.

Решим этим методом уравнение

$$y(x) + \int_0^1 x(e^{xs} - 1)y(s)ds = e^x - x, \quad x \in [0, 1]. \quad (13)$$

Точное решение уравнения  $y \equiv 1$  (см. пример 3.12, с. 171, [3]). Разложим ядро этого уравнения в ряд Тейлора, ограничившись тремя первыми членами. Напишем соответствующий сценарий.

```

% Сценарий решения задачи 3.16, с. 181, из книги
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>
% Часть 1 - аппроксимация ядра вырожденным
% Автор: Гаджиева З.
close all
clear all
clc
syms x s
taylor(x*(exp(x*s)-1),5,x)

```

Результатом работы сценария является следующее символьное выражение.

```

ans =

s*x^2+1/2*s^2*x^3+1/6*s^3*x^4

```

Напишем сценарий решения уравнения с таким ядром. Используем в нем функцию `Degenerate_Fred_II.m`.

```

% Сценарий решения задачи 3.16, с. 181, из книги
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>
% Часть 2 - решение уравнения с вырожденным ядром
% Автор: Гаджиева З.
close all
clear all
clc
syms x y
a = 0;
b = 1;
alpha = [x^2 x^3 x^4];
beta = [x 1/2*x^2 1/6*x^3];
lambda = -1;
f = exp(x)-x;
y=Degenerate_Fred_II(a,b,lambda,alpha,beta,f);
pretty(taylor(y,5,x)) % решение в удобном виде

```

Результатом работы сценария является следующее символьное выражение.



$$\begin{array}{r}
 4589889984855 \quad 2 \quad 24810467431825 \quad 3 \\
 1 - \text{-----} x - \text{-----} x - \\
 4503599627370496 \quad 54043195528445952 \\
 \\
 14994606363791 \quad 4 \\
 - \text{-----} x \\
 108086391056891904
 \end{array}$$

Эта функция на отрезке  $[0, 1]$  достаточно хорошо аппроксимирует точное решение  $y \equiv 1$ .

### Упражнения

- 1) Решите уравнение с вырожденным ядром (см. пример 1, с. 347, [5]):

$$y(x) = x^2 + \lambda \int_{-1}^1 (x+s)y(s)ds, \quad x \in [-1, 1].$$

Проверьте, что если  $\lambda^2 \neq 3/4$ , то решение этого уравнения есть

$$y(x) = x^2 + \frac{2\lambda x + 4\lambda^2/3}{3 - 4\lambda^2}.$$

- 2) Приблизительно решите уравнение (см. пример 3, с. 352, [5])

$$y(x) - \int_0^{1/2} \exp(-x^2 s^2) y(s) ds = 1, \quad x \in [0, 1/2]. \quad (14)$$

Ядро уравнения аппроксимируйте вырожденным ядром, пользуясь известным разложением:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

Проверьте, что при  $n = 2$ , т.е. если

$$K(x, y) = 1 - x^2 s^2 + \frac{x^4 s^4}{2},$$

приближенное решение уравнения (14) имеет вид:

$$\tilde{y}(x) = 1.9930 - 0.0833x^2 + 0.0007x^4.$$

Исследуйте, как приближенное решение зависит от  $n$ .

- 3) Напишите функцию, реализующую метод вырожденных ядер, основанный на аппроксимации ядра отрезком ряда Тейлора. Решите с помощью этой функции уравнения (13) и (14).
- 4) Исследуйте численно, как зависит точность решения уравнений (13) и (14) от числа членов в ряде Тейлора, аппроксимирующем ядро уравнения.
- 5) Решите уравнение (см. пример. 1, с. 357, [5])

$$y(x) = x^2 + \int_{-1}^1 \operatorname{sh}(x+s)y(s)ds, \quad x \in [-1, 1]. \quad (15)$$

Проверьте, что ядро этого уравнения является вырожденным,

$$\operatorname{sh}(x+s) = \operatorname{sh} x \operatorname{ch} s + \operatorname{ch} x \operatorname{sh} s,$$

а его точное решение имеет вид:

$$y(x) = x^2 + \alpha \operatorname{sh} x + \beta \operatorname{ch} x,$$

где

$$\alpha = \frac{6 \operatorname{sh} 1 - 4 \operatorname{ch} 1}{2 - \left(\frac{\operatorname{sh} 2}{2}\right)^2} = -0.6821, \quad \beta = \alpha \left(\frac{\operatorname{sh} 2}{2} - 1\right) = -0.5548.$$

### § 3. Метод наименьших квадратов

**1. Описание метода.** Выражение для невязки уравнения Фредгольма II рода (1), с. 22, имеет следующий вид:

$$Ry(x) = y(x) - \lambda \int_a^b K(x, s)y(s)ds - f(x), \quad x \in [a, b]. \quad (1)$$

Если  $y(x)$  — точное решение уравнения (1), с. 22, невязка равна нулю. Его приближенное решение  $\tilde{y}(x)$  будем искать в виде

$$\tilde{y}(x) = \sum_{i=1}^n c_i \varphi_i(x). \quad (2)$$

Здесь  $c_i$  — неизвестные константы, подлежащие определению (свободные параметры),  $\varphi_i(x)$  — заданные линейно независимые (координатные) функции. Обозначим буквой  $c$  вектор неизвестных коэффициентов  $c_i$ ,  $i = 1, 2, \dots, n$ . Подставим (2) в (1) и положим

$$\varepsilon(x, c) = R\tilde{y}(x).$$

Тогда для ненулевой невязки имеем

$$\varepsilon(x, c) = \sum_{i=1}^n c_i \left[ \varphi_i(x) - \lambda \int_a^b K(x, s)\varphi_i(s)ds \right] - f(x), \quad x \in [a, b]. \quad (3)$$

Постоянные  $c_i$  найдем из условий минимума функционала

$$J(c) = \int_a^b \varepsilon^2(x, c)dx, \quad (4)$$

т.е. из условий

$$\frac{\partial J}{\partial c_i} = 0, \quad i = 1, 2, \dots, n. \quad (5)$$

Подставим (3) в (4):

$$J(c) = \int_a^b \left\{ \sum_{i=1}^n c_i \left[ \varphi_i(x) - \lambda \int_a^b K(x, s)\varphi_i(s)ds \right] - f(x) \right\}^2 dx.$$

Вычислим частные производные:

$$\frac{\partial J}{\partial c_i} = \int_a^b 2 \left\{ \sum_{j=1}^n c_j \left[ \varphi_j(x) - \lambda \int_a^b K(x, s) \varphi_j(s) ds \right] - f(x) \right\} \times \\ \times \left[ \varphi_i(x) - \lambda \int_a^b K(x, s) \varphi_i(s) ds \right] dx, \quad i = 1, 2, \dots, n.$$

Приравняем их нулю. Получим систему линейных алгебраических уравнений с симметричной матрицей относительно неизвестных  $c$ :

$$\sum_{j=1}^n a_{ij} c_j = b_i, \quad i = 1, 2, \dots, n, \quad (6)$$

где

$$a_{ij} = \int_a^b \left[ \varphi_j(x) - \lambda \int_a^b K(x, s) \varphi_j(s) ds \right] \times \\ \times \left[ \varphi_i(x) - \lambda \int_a^b K(x, s) \varphi_i(s) ds \right] dx, \quad i, j = 1, 2, \dots, n. \\ b_i = \int_a^b f(x) \left[ \varphi_i(x) - \lambda \int_a^b K(x, s) \varphi_i(s) ds \right] dx, \quad i = 1, 2, \dots, n.$$

Решим эту систему и запишем приближенное решение исходного интегрального уравнения в виде (2). Теоретически этот метод обоснован, например, в [12], с. 453.

**2. Компьютерные программы.** Напишем на языке Matlab функцию `CoefMНК.m`, реализующую этот метод.

```
% Функция для решения линейного уравнения
% Фредгольма второго рода методом наименьших квадратов.
% Входные данные: K - ядро уравнения, f - правая часть
% (задаются аналитически), a - начало отрезка,
% b - конец отрезка интегрирования, L - параметр уравнения,
% n - число координатных функций, m - число узлов сетки.
```

```

% Результат: y - вектор значений решения на равномерной
% сетке из m узлов на отрезке [a,b],
% c - вектор коэффициентов разложения решения по
% координатным функциям fii
% Авторы: Гибадуллина Э., Касимова Э.
function [y, c] = CoefMNM(a,b,K,L,f,n,m)
V=zeros(n,1); % правая часть СЛАУ
A=zeros(n); % матрица слау
syms x s
format long
for i=1:n
    for j=1:n
        A(i,j)=int((fii(x,j)-...
            L.*int(K(x,s).*fii(s,j),s,a,b)).*(fii(x,i)-...
            L.*int( K(x,s).*fii(s,i),s,a,b)), x, a, b);
    end
    V(i)=int(f(x).*(fii(x,i)-...
        L.* int( K(x,s).*fii(s,i),s,a,b)), x, a, b);
end
c=A\V; % коэффициенты C
x1 = linspace(a,b,m);
y=0;
for i = 1 : n y = y+c(i)*fii(x1,i);
end
end

```

Координатные функции определим в `fii.m`.

```

% Функция, задающая координатные функции.
% Авторы: Гибадуллина Э., Касимова Э.

```

```

function P = fii( x,n )
P = x.^(n-1);
end

```

**3. Пример.** На языке Matlab решим уравнение 3.24, с. 206, [3]:

$$y(x) = x + \int_{-1}^1 xsy(s)ds, \quad x \in [-1, 1].$$

Точное решение этого уравнения  $y = 3x$ . Приближенное решение будем искать в виде

$$\tilde{y}(x) = c_1 + c_2x,$$

т.е. положим  $\varphi_1(x) = 1$ ,  $\varphi_2(x) = x$ . Сценарий решения этого уравнения MNK.m на языке Matlab выглядит следующим образом.

```
% Сценарий решения задачи 3.24, с. 206, из книги
% Верлань А.Ф., Сизиков В.С. <<Интегральные уравнения...>>
% Авторы: Гибадуллина Э., Касимова Э.
```

```
clear all
close all
clc
syms x s;
a=-1;
b=1;
K=@(x,s) x*s;      % ядро
f=@(x) x;          % правая часть
y_exact=@(x) 3*x; %точное решение
lambda=1;
n=2; % число координатных функций
m=20; % число узлов сетки
[y,c]=CoefMNK( a,b,K,lambda,f,n,m );
c %коэф-ты
x = linspace(a,b,m); % сетка для построения графиков
figure(1)
plot(x,y_exact(x),'b') % график точного решения
hold on
plot(x,y,'ro') % график приближенного решения
xlabel('x');
ylabel('y');
```

Результатами работы этого сценария являются графики точного и приближенного решений (см. рис. 2), а также вектор искомых коэффициентов:

```
c =
     0
     3
```

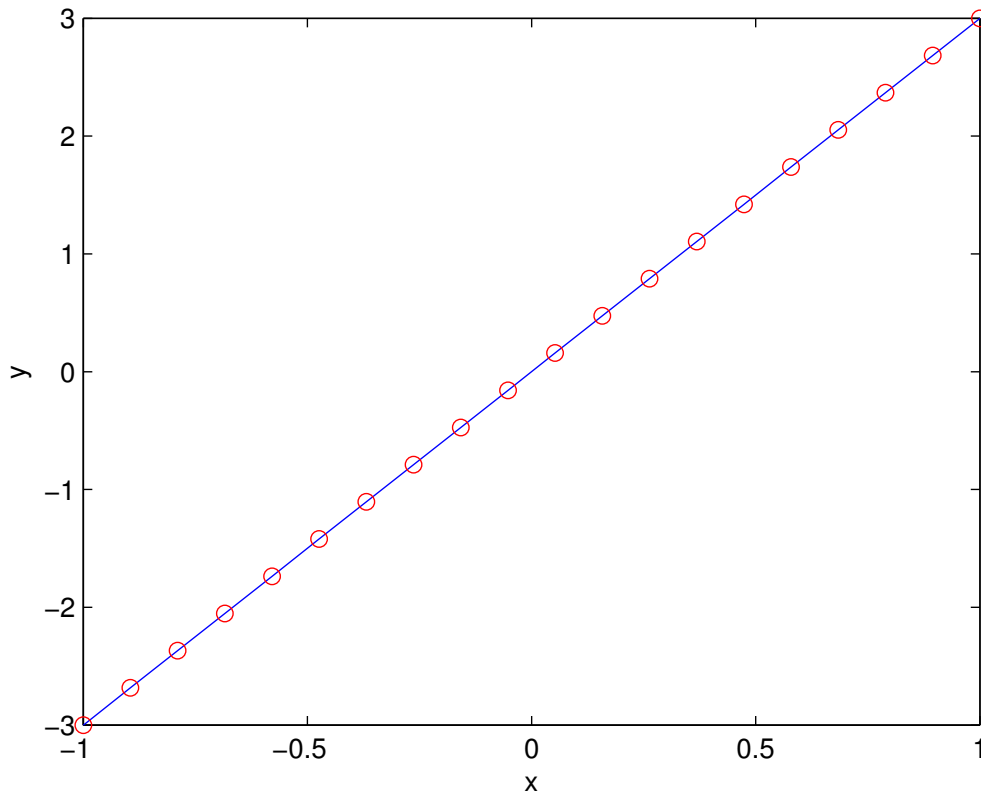


Рис. 2. Результаты решения задачи 3.24, с. 206, из книги [3]. Непрерывной линией обозначено точное решение, кружочками — приближенное решение.

### Упражнения

- 1) Можно ли сказать, что приближенное решение совпало с точным, если да, то почему это произошло?
- 2) Что будет, если увеличить число координатных функций  $n$ ? Постройте график зависимости от  $n$  погрешности решения рассмотренной в этом параграфе задачи.
- 3) С помощью функции `CoefMNC.m` найдите приближенное решение уравнения (5), с. 26. Исследуйте зависимость точности решения от параметра  $n$ .
- 4) С помощью функции `CoefMNC.m` найдите приближенное решение уравнения (13), с. 30. Исследуйте зависимость точности решения от параметра  $n$ .

- 5) Методом наименьших квадратов найдите приближенное решение уравнения (15), с. 33. Разыскивайте его в виде

$$y(x) = c_1 + c_2x + c_3x^2.$$

Сравните приближенное решение с точным (см. с. 33) при малых значениях  $x$  и на концах отрезка  $[-1, 1]$ . Что будет, если увеличить число координатных функций?

- 6) Методом наименьших квадратов найдите приближенное решение уравнения (15), с. 33, разыскивая его в виде

$$y(x) = c_1x^2 + c_2 \operatorname{sh} x + c_3 \operatorname{ch} x.$$

Сравните приближенное решение с точным (см. с. 33).



## § 4. Метод Галеркина — Петрова

**1. Описание метода.** Пусть решение уравнения (1), с. 22, разыскивается в линейном пространстве со скалярным произведением. Выберем в нем две системы линейно-независимых функций:  $\varphi_i(x)$  и  $\psi_i(x)$ ,  $i = 1, 2, \dots, n$ . Приближенное решение уравнения запишем в виде

$$\tilde{y}(x) = f(x) + \sum_{j=1}^n c_j \varphi_j(x), \quad (1)$$

где  $c_j$  — подлежащие определению коэффициенты. Ясно, что если невязка (1), с. 34, для функции  $\tilde{y}(x)$  равна нулю (т. е.  $\tilde{y}(x)$  — точное решение), то она ортогональна каждой из функций  $\psi_i(x)$ ,  $i = 1, 2, \dots, n$ . Используем эти же условия ортогональности для обеспечения малости невязки уравнения с приближенным решением  $\tilde{y}(x)$ :

$$(R\tilde{y}, \psi_i) = 0, \quad i = 1, 2, \dots, n. \quad (2)$$

Пусть скалярное произведение определено формулой

$$(f, g) = \int_a^b f(x)g(x)dx.$$

Тогда равенства (2) представляют собой систему линейных алгебраических уравнений относительно неизвестных коэффициентов  $c_j$ :

$$\sum_{j=1}^n a_{ij}c_j = b_i, \quad i = 1, 2, \dots, n. \quad (3)$$

Здесь

$$a_{ij} = \int_a^b \varphi_j(x)\psi_i(x)dx - \lambda \int_a^b \psi_i(x) \int_a^b K(x, s)\varphi_j(s)dsdx,$$

$$b_i = \lambda \int_a^b \psi_i(x) \int_a^b K(x, s)f(s)dsdx.$$

Решив систему (3), найдем коэффициенты в представлении приближенного решения (1). В наиболее общем виде этот метод обоснован, например, в [7], с. 190.

**2. Компьютерная программа.** Напишем на языке Matlab функцию `kem_Fred.m`, реализующую метод Галеркина — Петрова.

```
% Функция для решения уравнения Фредгольма II рода
% методом Галеркина - Петрова.
% Входные данные: K - ядро уравнения,
% f - правая часть (задаются аналитически),
% a - начало отрезка интегрирования,
% b - конец отрезка интегрирования,
% lambda - параметр уравнения,
% n - число координатных функций,
% Результат: y - вектор значений приближенного решения
% Авторы : Матвеев Д.И., Липатов В.С.
function [y]= kem_Fred(K,f,a,b,lambda,n)
syms x;
syms s;
format long
B=zeros(n,1); % правая часть СЛАУ
A=zeros(n); % матрица СЛАУ
for j=1:n
    B(j)=lambda*int(int(psi(x,j).*f(s).*...
        K(x,s),s,a,b),x,a,b);
    for i=1:n
        A(i,j)= quadl(@(x) fi(x,i).*psi(x,j),a,b)- ...
            lambda*int(int( K(x,s).*fi(s,i).*...
                psi(x,j),s,a,b),x,a,b);
    end
end
c=A\B; % решение СЛАУ - коэффициенты вектора c
x1 = linspace(a,b);
y = f(x1);
for i = 1 : n
    y = y + c(i)*fi(x1,i);
end
end
```

**3. Пример.** Решим методом Галеркина — Петрова уравнение

$$y(x) = 1 + \int_{-1}^1 (xs + x^2)y(s)ds, \quad x \in [-1, 1]. \quad (4)$$

Точное решение этого уравнения  $y(x) = 1 + 6x^2$  (см. упражнение 3.25, с. 207, [3]). Приближенное решение ищем в виде

$$\tilde{y}(x) = 1 + c_1\varphi_1(x) + c_2\varphi_2(x),$$

где  $\varphi_1(x) = x$ ,  $\varphi_2(x) = x^2$ , из условия ортогональности невязки функциям  $\psi_1(x) = 1$ ,  $\psi_2(x) = x$ . Напишем на языке Matlab функции, определяющие  $\varphi_m(x)$  и  $\psi_m(x)$ .

```
% Входные данные: x - аргумент функции,
% m - порядковый номер функции
% Результат: f - значение функции
function [ f ] = fi( x,m)
f=x.^m;
end
```

```
% Входные данные: x - аргумент функции,
% m - порядковый номер функции
% Результат: f - значение функции
function [ f ] = psi(x,m)
f=x.^(m-1);
end
```

Теперь напишем сценарий выполнения упражнения.

```
% Сценарий выполнения упражнения 3.25, с. 207, из книги
% Верлань А.Ф Сизиков В.С. "Интегральные уравнения..."
% Авторы : Матвеев Д.И., Липатов В.С.
close all
clear all
clc
syms x s;
K= @(x,s)x.*s+x.^2; % ядро уравнения
a=-1; % начало отрезка интегрирования
```

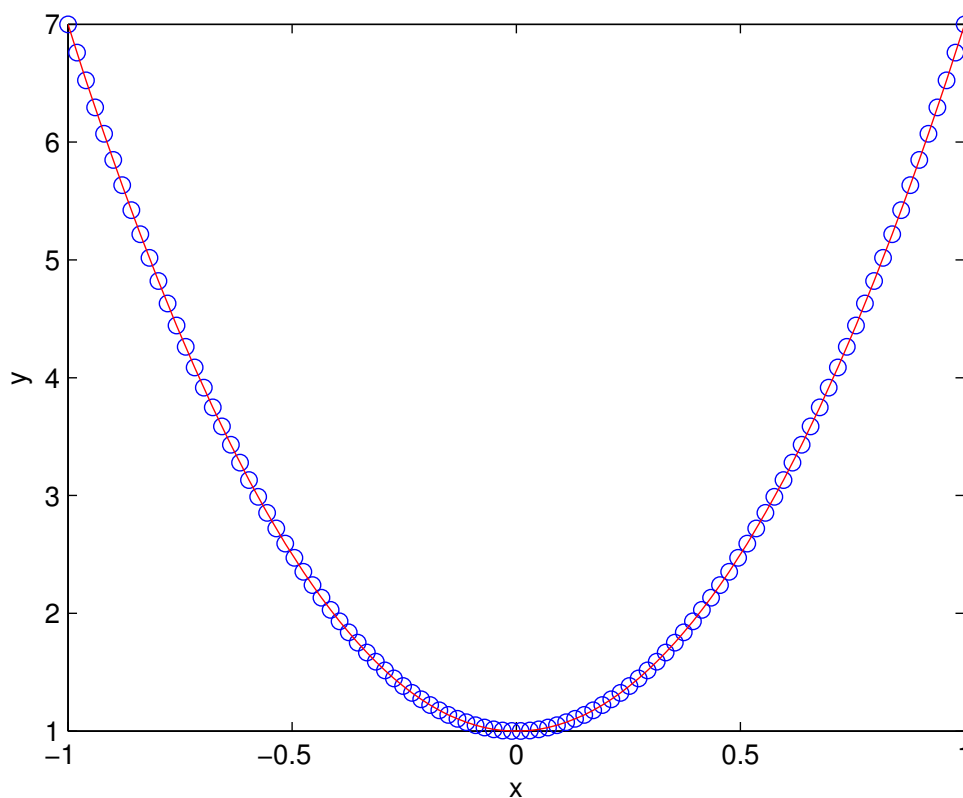


Рис. 3. Результаты решения уравнения (4) методом Галеркина — Петрова. Непрерывной линией обозначено точное решение, кружочками — приближенное решение.

```
b=1; % конец отрезка интегрирования
lambda = 1; % параметр уравнения
f=@(x)0*x+1; % правая часть уравнения
n=2; % число координатных функций
y=kem_Fred (K,f,a,b,lambda,n);
x=linspace(a,b); % сетка для построения графика
y_exact=@(x)x.^2*6+1;
figure(1)
plot(x,y_exact,'o')
hold on
plot(x,y,'r')
xlabel('x')
ylabel('y')
```

Результатом работы этого сценария являются графики точного и приближенного решений (см. рис. 3).

### Упражнения

- 1) Можно ли сказать, что приближенное решение совпало с точным, если да, то почему это произошло?
- 2) Что будет, если увеличить число координатных функций  $n$ ? Постройте график зависимости от  $n$  погрешности решения задачи и числа обусловленности матрицы  $A$ .
- 3) С помощью функции `kem_Fred.m` найдите приближенное решение уравнения с невырожденным ядром (пример постройте самостоятельно). Исследуйте зависимость точности решения от  $n$ .
- 4) Частным случаем метода Галеркина — Петрова является метод Бубнова — Галеркина. В этом методе полагают  $\varphi_i(x) = \psi_i(x)$  для всех  $i = 1, 2, \dots, n$ . Постройте расчетные формулы и напишите компьютерные программы. Решите этим методом уравнение (4). Приближенное решение ищите в виде

$$\tilde{y}(x) = 1 + c_1x + c_2x^2,$$

из условия ортогональности невязки функциям  $\varphi_1(x) = x$  и  $\varphi_2(x) = x^2$  (см. пример 3.26, с. 209, [3]).

- 5) Методом Бубнова — Галеркина найдите приближенное решение уравнения с невырожденным ядром (пример постройте самостоятельно). Исследуйте зависимость точности решения от  $n$ .

## § 5. Метод коллокации

**1. Описание метода коллокации** дадим, следуя [5]. Запишем интегральное уравнение Фредгольма второго рода в виде равенства нулю его невязки:

$$Ry(x) = y(x) - \lambda \int_a^b K(x, s)y(s)ds - f(x) = 0, \quad x \in [a, b]. \quad (1)$$

Будем искать приближенное решение уравнения (1) в виде

$$\tilde{y}(x) = \sum_{j=1}^n c_j \varphi_j(x). \quad (2)$$

Здесь  $c_j$ ,  $j = 1, 2, \dots, n$ , — неизвестные константы, подлежащие определению,  $\varphi_j(x)$ ,  $j = 1, 2, \dots, n$ , — заданные линейно независимые (координатные) функции. Подставим выражение (2) в левую часть (1), получим невязку для приближенного решения:

$$\begin{aligned} R\tilde{y}(x) &= \sum_{j=1}^n c_j \varphi_j(x) - \lambda \int_a^b K(x, s) \sum_{j=1}^n c_j \varphi_j(s) ds - f(x) = \\ &= \sum_{j=1}^n c_j \left[ \varphi_j(x) - \lambda \int_a^b K(x, s) \varphi_j(s) ds \right] - f(x). \end{aligned}$$

Потребуем чтобы невязка  $R\tilde{y}(x)$  обращалась в нуль в заданной системе точек коллокации из отрезка  $[a, b]$ :

$$a \leq x_1 < x_2 < \dots < x_n \leq b.$$

Получим систему линейных алгебраических уравнений относительно неизвестных  $c_j$ ,  $j = 1, 2, \dots, n$ ,

$$\sum_{j=1}^n c_j \left[ \varphi_j(x_i) - \lambda \int_a^b K(x_i, s) \varphi_j(s) ds \right] = f(x_i), \quad i = 1, 2, \dots, n. \quad (3)$$

Обозначим элементы матрицы этой системы следующим образом:

$$\psi_j(x_i, \lambda) = \varphi_j(x_i) - \lambda \int_a^b K(x_i, s) \varphi_j(s) ds, \quad i, j = 1, 2, \dots, n. \quad (4)$$

Тогда система (3) записывается в виде

$$\sum_{j=1}^n \psi_j(x_i, \lambda) c_j = f(x_i), \quad i = 1, 2, \dots, n. \quad (5)$$

Если определитель системы (5) не равен нулю, то из нее можно однозначно определить значения  $c_j$ ,  $j = 1, 2, \dots, n$ , и, следовательно, найти приближенное решение  $\tilde{y}(x)$  по формуле (2).

Таким образом, алгоритм решения интегрального уравнения Фредгольма второго рода методом коллокации состоит в вычислении элементов матрицы по формуле (4) и решении системы линейных алгебраических уравнений (5). Приближенное решение интегрального уравнения записывается аналитически в виде (2). Доказательство сходимости этого метода для общего случая многомерных слабо сингулярных уравнений можно найти в книге [15].

**2. Компьютерная программа.** Реализуем описанный алгоритм в функции `Fredgolm_II_Colloc.m` на языке Matlab.

```
% Функция для решения уравнения Фредгольма второго
% методом коллокации
% Входные данные: lambda - параметр уравнения,
% K - ядро интегрального уравнения,
% f - правая часть (задаются аналитически),
% a - начало отрезка интегрирования,
% b - конец отрезка,
% n - число базисных функций;
% m - число узлов в сетке:
% Fi - функция, реализующая построение базисных функций
% Результат: y - символьное решение,
% y_f - значение функции y в узлах сетки,
% CondNumb - число обусловленности матрицы СЛАУ
% Автор: Ганиева Э.
```

```
function [ y,y_f,CondNumb] = ...
    Fredgolm_II_Colloc(a,b,lambda,K,f,n,m)
xx=linspace(a,b,n+1); % точки коллокации
ff=f(xx); % правая часть СЛАУ
M=zeros(n+1,n+1); % матрица СЛАУ
```

```

syms x

for i=1:n+1
    for j=1:n+1
        % вычисляем интеграл
        integr=quadr(@(s) lambda*K(xx(i),s).*Fi(s, j-1),a,b);
        % вычисляем значение функции Psi_j в точке x(i)
        M(i,j)=Fi(xx(i), j-1)-integr;
    end
end
end
CondNumb=cond(M,inf); % число обусловленности
C=M\ff'; % решение СЛАУ
y=0;
% Решение уравнения в символьном виде
for i=1:n+1
    y=y+C(i)*Fi(x, i-1);
end
% Решение уравнения в узлах сетки
y_f=0;
x=linspace(a,b,m);
for i=1:n+1
    y_f=y_f+C(i)*Fi(x, i-1);
end
end
end

```

**3. Координатные функции.** В качестве координатных функций будем использовать полиномы Лежандра. Вычислять их будем по рекуррентной формуле (см., напр., [13], с. 166)

$$P_n(x) = \frac{2n-1}{n}xP_{n-1} - \frac{n-1}{n}P_{n-2}, \quad n \geq 2,$$

где  $P_0 = 1$ ,  $P_1(x) = x$ . Создадим соответствующую функцию `Fi.m`.

```

% Полином Лежандра
% Входные данные: x - независимая переменная,
% n - степень полинома,
% Результат: P - значение полинома

```



```
% Автор: Ганиева Э.
```

```
function P = Fi(x,n)
```

```
if n==0
```

```
    P=1;
```

```
else
```

```
    if n==1
```

```
        P=x;
```

```
    else
```

```
        P=(2*n-1)./n.*x.*Fi(x,n-1) - (n-1)./n.*Fi(x,n-2);
```

```
    end
```

```
end
```

```
end
```

**4. Пример.** Решим уравнение

$$y(x) = 1 + \frac{4}{3}x + \int_{-1}^1 (xs^2 - x)y(s)ds, \quad x \in [-1, 1]. \quad (6)$$

Точное решение этого уравнения  $y \equiv 1$  (см. пример 3.30, с. 213, [3]). Сценарий решения этого уравнения на языке Matlab выглядит следующим образом.

```
clc
```

```
clear all
```

```
close all
```

```
K= @(x,s) x.*s.^2-x; % ядро уравнения
```

```
f= @(x) 1+4*x/3; % правая часть
```

```
lambda=1;
```

```
a=-1;
```

```
b=1;
```

```
n=2;
```

```
m=10;
```

```
[y,y_f,CondNumb]=Fredholm_II_Colloc(a,b,lambda,K,f,n,m);
```

```
pretty(y) % решение в удобном виде
```

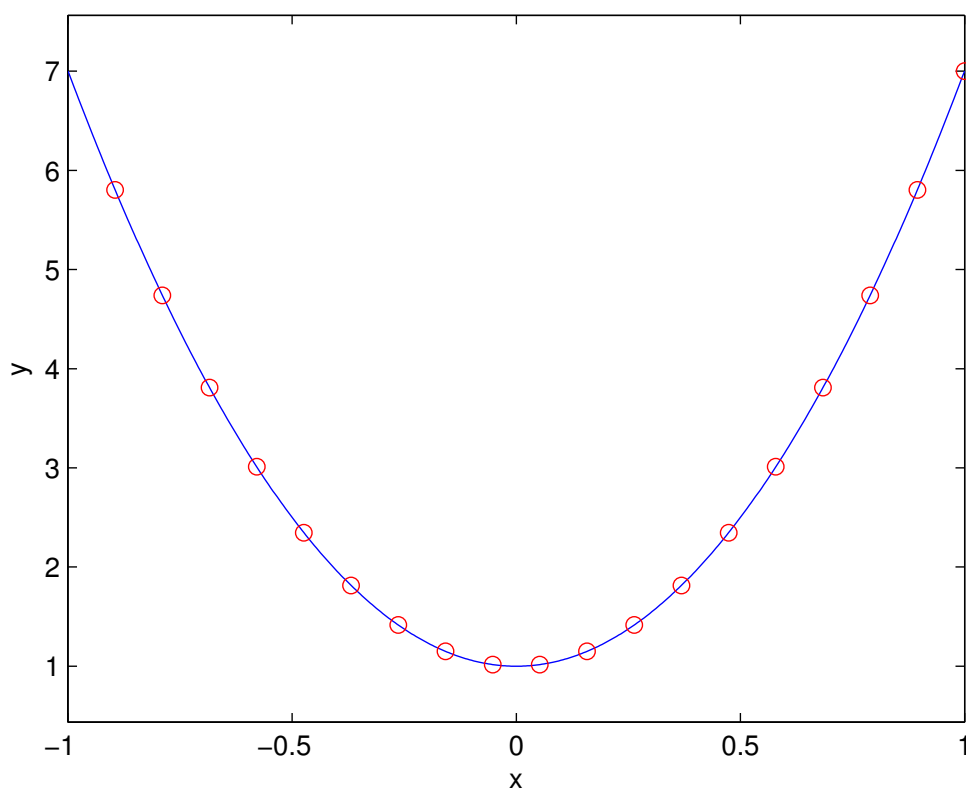


Рис. 4. Результаты решения уравнения (4), с. 42, методом коллокации. Непрерывной линией обозначено приближенное решение, кружочками — точное решение.

Результатом работы этого сценария является символьное выражение решения исходного уравнения: 1.

**5. Пример.** Решим методом коллокации уравнение (4), с. 42. Сценарий решения этого уравнения на языке Matlab выглядит следующим образом.

```
clc
clear all
close all

n=3;
m=20;
K= @(x,s) x*s+x^2; % ядро уравнения
f= @(x) 1 + 0*x; % правая часть
lambda=1;
a=-1;
```

```

b=1;

[y, y_f, condNumb]=Fredholm_II_Colloc(a,b,lambda,K,f,n,m);
x=linspace(a,b,m);
y_exact=@(x) 6*x.^2+1; % точное решение
pretty(y) % решение в удобном виде

figure
ezplot(y,a,b) % приближенное решение
title(' ')
hold on
plot(x, y_exact(x), 'or')
xlabel('x');
ylabel('y');
hold off

% относительная ошибка решения
error=norm(y_f-y_exact(x), inf)/norm(y_exact(x), inf)

```

Результатом работы этого сценария является символьное выражение решения исходного уравнения:

$$\begin{array}{c}
 \begin{array}{c}
 / \quad 2 \quad \backslash \\
 | \quad 3 \quad x \quad 1 \quad | \\
 30399297484750865 \quad x \quad | \quad \text{----} \quad - \quad - \quad | \\
 \backslash \quad 2 \quad 2 \quad / \\
 \text{-----} \\
 121694457621910022543683507716096 \\
 \\
 94237822202727589 \quad x \quad \quad \quad 2 \\
 - \quad \text{-----} \quad + \quad 6 \quad x \quad + \quad 1 \\
 60847228810955011271841753858048
 \end{array}
 \end{array}$$

Напомним, что точное решение уравнения (4), с. 42, имеет вид

$$y(x) = 6x^2 + 1.$$

Ясно, что приближенное решение на отрезке  $[-1, 1]$  хорошо аппроксимирует точное. В сценарии вычисляется относительная ошибка решения:

error =

3.806478941571965e-016

На рис. 4 показаны графики точного решения этого уравнения и приближенного, полученного методом коллокации.

### Упражнения

- 1) Модифицируйте программы так, чтобы все координатные функции вычислялись заранее, а не по мере необходимости. Какого увеличения скорости удалось достичь?
- 2) Решите методом коллокации уравнение

$$y(x) = e^x - \frac{1 + e^\pi}{2} + (1 - e^\pi) \cos(x) + \int_0^\pi (\sin(s) + \cos(x)) y(s) ds,$$

где  $x \in [0, \pi]$ . Убедитесь в том, что точное решение этого уравнения  $y(x) = e^x$ . Используйте в качестве координатных функций полиномы Лежандра. Увеличивайте число координатных функций до тех пор, пока число обусловленности матрицы системы метода коллокации не станет больше  $10^{16}$ . Постройте графики зависимости числа обусловленности и относительной ошибки решения от числа координатных функций.

- 3) Напишите функции, реализующие построение координатных функций на основе ортогональных полиномов Чебышева (см., напр., [6], с. 141),

$$T_0(x) \equiv 1, \quad T_1(x) = x,$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1,$$

и естественного базиса в пространстве полиномов с вещественными коэффициентами (см., напр., [6], с. 126):  $1, x, x^2 \dots$

- 4) Решите уравнение

$$y(x) = x + 4 \int_0^1 x^2 s^2 y(s) ds, \quad x \in [0, 1].$$

Поверьте, что точное решение этого уравнения  $y(x) = x + 5x^2$ . В качестве координатных функций используйте полиномы Лежандра, Чебышева и элементы естественного базиса. Убедитесь в том, что во всех трех случаях приближенное решение совпадает с точным.

5) Решите уравнение

$$y(x) = \left(1 - \frac{3\pi}{2}\right) \cos(x) + \int_0^{3\pi} \cos(x+s)y(s)ds, \quad x \in [0, 3\pi].$$

Проверьте, что точное решение этого уравнения  $y(x) = \cos(x)$ . Используйте полиномы Чебышева в качестве координатных функций. Увеличивайте число координатных функций до тех пор, пока число обусловленности матрицы системы метода коллокации не станет больше  $10^{16}$ . Постройте графики зависимости числа обусловленности и относительной ошибки решения от числа координатных функций.

6) Решите методом коллокации уравнение (см. пример 1, с. 355, [5])

$$y(x) - \int_0^1 \frac{s^2 y(s)}{x^2 + s^2} ds = x \operatorname{arctg} \frac{1}{x}, \quad x \in [0, 1].$$

Точное решение этого уравнения есть  $y \equiv 1$ . Разыскивайте решение в виде

$$y(x) = c_1 + c_2 x,$$

выбирая точки коллокации  $x_1 = 0$ ,  $x_2 = 1$ .

## § 6. Метод простой итерации

На языке Matlab решите пример 3.17, с. 185, [3].

## § 7. Метод моментов

На языке Matlab решите пример 3.29, с. 212, [3].

---

---

ГЛАВА 3

Однородное уравнение Фредгольма II рода

§ 1. Вырожденные ядра

1. Однородное линейное уравнение Фредгольма II рода имеет следующий вид:

$$y(x) = \lambda \int_a^b K(x, s)y(s)ds, \quad x \in [a, b]. \quad (1)$$

Число  $\lambda$  называется характеристическим числом уравнения (1), если у этого уравнения существует нетривиальное решение. Это решение называется собственной функцией интегрального уравнения, соответствующей характеристическому числу  $\lambda$ .

Если ядро  $K(x, y)$  непрерывно по совокупности аргументов, то интегральный оператор, определяемый правой частью уравнения (1) является компактным (см., напр., [14], с. 23) на пространстве непрерывных на  $[a, b]$  функций, и для теоретического анализа спектральной задачи (1) можно применить результаты теории компактных операторов (см., напр., [14], с. 34). В этом случае конечных характеристических чисел может и не существовать. Если ядро симметрично ( $K(x, s) = K(s, x)$ ), то согласно спектральной теории самосопряженных компактных операторов (см., напр., [14], с. 271) существует счетное множество вещественных характеристических чисел.

**2. Решение задачи с вырожденным ядром.** Если ядро уравнения вырождено (см. с. 27), то спектральная задача (1) эквивалентна конечномерной алгебраической спектральной задаче. Действительно, запишем уравнение с вырожденным ядром следующим в виде (сравните с (3), с. 27):

$$y(x) - \lambda \sum_{i=1}^m \alpha_i(x) \int_a^b \beta_i(s)y(s)ds = 0, \quad x \in [a, b]. \quad (2)$$

Пусть  $\lambda$  — характеристическое число. Тогда каждая собственная функция имеет вид

$$y(x) = \lambda \sum_{i=1}^m c_i \alpha_i(x), \quad x \in [a, b], \quad (3)$$

где  $c_i$  — неизвестные константы, определенные равенствами (4), с. 27.

Эти константы и характеристические числа являются решениями следующей алгебраической спектральной задачи:

$$\begin{pmatrix} 1 - \lambda a_{11} & -\lambda a_{12} & \cdots & -\lambda a_{1m} \\ -\lambda a_{21} & 1 - \lambda a_{22} & \cdots & -\lambda a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ -\lambda a_{m1} & -\lambda a_{m2} & \cdots & 1 - \lambda a_{mm} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \cdots \\ c_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}, \quad (4)$$

где элементы матрицы вычисляются по формулам (9), с. 28. Для того чтобы сформулировать эту задачу надо положить правую часть в (12), с. 28, равной нулю.

Таким образом, для решения однородного уравнения (2) с вырожденным ядром достаточно вычислить интегралы (9), с. 28, составить определитель

$$D(\lambda) = \begin{vmatrix} 1 - \lambda a_{11} & -\lambda a_{12} & \cdots & -\lambda a_{1m} \\ -\lambda a_{21} & 1 - \lambda a_{22} & \cdots & -\lambda a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ -\lambda a_{m1} & -\lambda a_{m2} & \cdots & 1 - \lambda a_{mm} \end{vmatrix}, \quad (5)$$

развернуть его в уравнение  $D(\lambda) = 0$ , решение которого дает искомые характеристические числа, каждое из которых, будучи поставлено в линейную систему (4), позволяет после решения последней найти набор постоянных  $c_i$ , определяющий одну из собственных функций согласно выражению (3).

**3. Компьютерная программа.** Напишем на языке Matlab сценарий решения однородного уравнения с ядром

$$K(x, s) = \cos^2(x) \cos(2s) + \cos(3x) \cos^3(s)$$

и пределами интегрирования  $a = 0$ ,  $b = \pi$ . Характеристические числа этого уравнения и отвечающие им собственные функции известны (см. пример 3.13, с. 172, [3]):

$$\lambda_1 = \frac{4}{\pi}, \quad y_1 = c_1 \cos^2(x),$$

$$\lambda_2 = \frac{8}{\pi}, \quad y_2 = c_2 \cos(3x).$$

```

% Сценарий решения упражнения 3.13, с. 172, из книги
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>
% Автор: Журавлева А.
close all
clear all
clc
%начальные данные---
syms x s;
a = 0;
b = pi;
alpha=[(cos(x))^2 cos(3*x)];
beta=[cos(2*s) (cos(s))^3];
%ход решения---
syms lambda;
m=length(alpha);
%заменяем в alpha переменную x на s ф-ей subs
for i=1:m
    alpha(i) = subs(alpha(i),x,s);
end;
%считаем коэф-ты a
for i=1:m
    for j=1:m
        A(i,j)=int(alpha(j)*beta(i),a,b);
    end;
end;
%выполним обратную замену в alpha переменной s на x
for i=1:m
    alpha(i)=subs(alpha(i),s,x);
end;
%создаем столбец постоянных C
C=sym([1:m]);
for i=1:m
    C(i)=strcat('c',int2str(i));
end;
%составляем матрицу D(lambda) алгебраической

```



```

% спектральной задачи  $D(\lambda)C=0$ 
for i=1:m
    for j=1:m
        if i~=j
             $D(i,j)=-1*\lambda*A(i,j)$ ;
        else
             $D(i,j)=1-1*\lambda*A(i,j)$ ;
        end;
    end;
end;
%составляем определитель матрицы D,
% разворачиваем его в уравнение  $d(\lambda)=0$ 
% и находим искомые характеристический числа  $\lambda$ 
d = det(D);  $\lambda_k=solve(d,\lambda)$ ;
%подставляем найденные значения в линейную систему
%для определения C
for i=1:m
    %матрица однородной системы при конкретном значении  $\lambda$ 
    B1=subs(D, 'lambda',  $\lambda_k(i)$ );
    syms x;
    %фундаментальная система решений однородной системы
    x=sym(null(double(B1), 'r'));
    anss=C(i)*x;
    %сохраняем результат в матрицу
    for j=1:m
         $O(j,i)=anss(j)$ ;
    end;
end;
%формируем собственные функции
for i=1:m
    for j=1:m
        if  $O(i,j) \neq 0$ 
             $y(i)=C(i)*\alpha(i)$ ;
        end;
    end;
end;
end;
%печатаем ответ

```

```
for i=1:m
    lambda=lambdak(i)
    Y=y(i)
    disp(' ')
end;
```

Результат работы сценария совпадает с точным решением задачи:

lambda =

4/pi

Y =

c1\*cos(x)^2

lambda =

8/pi

Y =

c2\*cos(3\*x)

### Упражнения

- 1) Модифицируйте сценарий так, чтобы решить уравнение из примера 3.27, с. 210, [3].
- 2) Решите с помощью предложенного сценария уравнение из примера 3.33, с. 220, [3].
- 3) Решите с помощью предложенного сценария уравнение из примера 3.34, с. 221, [3].
- 4) Решите с помощью предложенного сценария уравнение из примера 3.35, с. 225, [3].

## § 2. Метод Бубнова — Галеркина

На языке Matlab решите пример 3.27, с. 210, [3].

### **§ 3. Метод Рунца**

На языке Matlab решите пример 3.33, с. 220, [3].

### **§ 4. Метод следов**

На языке Matlab решите пример 3.34, с. 221, [3].

### **§ 5. Метод Келлога**

На языке Matlab решите пример 3.35, с. 225, [3].

---

---

## Литература

1. Бахвалов Н.С. Численные методы. — М.: Изд-во «Наука», 1975. — 632 с.
2. Березин И.С., Жидков Н.П. Методы вычислений. Том 2. — М.: Государственное издательство физико-математической литературы, 1959. — 620 с.
3. Верлань А.Ф., Сизиков В.С. Интегральные уравнения: методы, алгоритмы, программы. — Киев.: Наукова думка, 1986. — 544 с.
4. Глазырина Л.Л., Карчевский М.М. Введение в численные методы: учебное пособие. — Казань.: Казан. ун-т, 2012. — 122 с.
5. Демидович Б.П., Марон И.А., Шувалова Э.З. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения. — М.: Изд-во «Наука», 1967. — 368 с.
6. Карчевский Е.М., Карчевский М.М. Лекции по линейной алгебре и аналитической геометрии: учебное пособие. — Казань: Изд-во Казан. ун-та, 2014. — 352 с.
7. Красносельский М.А., Вайникко Г.М., Забрейко П.П., Рунтцкий Я.Б., Стеценко В.Я. Приближенное решение операторных уравнений. — М.: Изд-во «Наука», 1969. — 455 с.
8. Крылов В.И. Приближенное вычисление интегралов. — М.: Изд-во «Наука», 1967. — 500 с.
9. Манжиров А.В., Полянин А.Д. Справочник по интегральным уравнениям: методы решения. — М.: Изд-во «Факториал Пресс», 2000. — 384 с.
10. Михлин С.Г. Лекции по линейным интегральным уравнениям. — М.: Государственное издательство физико-математической литературы, 1959. — 232 с.

11. Михлин С.Г., Смолицкий Х.Л. Приближенные методы решения дифференциальных и интегральных уравнений. — М.: Изд-во «Наука», 1965. — 384 с.
12. Михлин С.Г. Вариационные методы в математической физике. — М.: Изд-во «Наука», 1970. — 512 с.
13. Янке Е., Эмде Ф., Леш Ф. Специальные функции. — М.: Изд-во «Наука», 1968. — 344 с.
14. Kress R. Linear Integral Equations. — New York: Springer-Verlag, 1999. — 365 p.
15. Vainikko G. Multidimensional Weakly Singular Integral Equations. — Berlin, Heidelberg: Springer-Verlag, 1993. — 159 p.
16. Brian R. Hunt [и др.] Matlab R2007 с нуля [пер. с англ.] — М.: Лучшие книги, 2008. — 352 с.