

Е.М. КАРЧЕВСКИЙ

**Интегральные уравнения:
применение в математическом
моделировании, численные
методы решения и
комплекс программ
на языке MatLab**

Учебное пособие

Вестфалика

2022

УДК 519.64

ББК 22.193

К27

Рекомендовано Учебно-методической комиссией

Института вычислительной математики и информационных технологий

Рецензенты:

кандидат физ.-мат. наук, доцент, заместитель директора по научной деятельности Института вычислительной математики и информационных технологий Казанского федерального университета Тумаков Дмитрий Николаевич,

доктор физ.-мат. наук, доцент, профессор кафедры прикладной математики и искусственного интеллекта Казанского федерального университета Конохов Владимир Михайлович.

Карчевский Е.М.

К27 Интегральные уравнения: применение в математическом моделировании, численные методы решения и комплекс программ на языке MatLab: учебное пособие / Е.М. Карчевский. — Казань: Вестфалика, 2022. — 89 с.

Учебная пособие предназначено для студентов, специализирующихся в области прикладной математики, информатики и математического моделирования. Даются основные сведения о возникновении интегральных уравнений при математическом моделировании физических процессов. Подробно описываются численные методы решения уравнений Вольтера и Фредгольма второго рода, их реализация в виде комплекса программ на языке MatLab.

© Карчевский Е.М., 2022

Оглавление

Предисловие	3
Интегральные уравнения в математическом моделировании	4
ГЛАВА 1. Методы решения уравнений Вольтерра II рода	6
§ 1. Метод квадратур	6
§ 2. Метод квадратур для нелинейного уравнения	15
§ 3. Метод простой итерации	21
ГЛАВА 2. Методы решения уравнений Фредгольма II рода	27
§ 1. Метод квадратур	27
§ 2. Метод вырожденных ядер	33
§ 3. Метод наименьших квадратов	40
§ 4. Метод Галеркина — Петрова	48
§ 5. Метод моментов	53
§ 6. Метод коллокации	59
§ 7. Метод простой итерации	67
ГЛАВА 3. Однородное уравнение Фредгольма II рода	72
§ 1. Вырожденные ядра	72
§ 2. Метод Ритца	78
Литература	88

Предисловие

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

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

Интегральные уравнения в математическом моделировании

Линейное уравнение Вольтерра II рода имеет следующий вид:¹⁾

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

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, [10], с. 45). В частности (см., напр., [11], 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, характеризуемые этим свойством, называются операторами Вольтерра и широко применяются при описании процессов с последствием и обратной связью (см., напр., [3], с. 22.).

Линейное уравнение Фредгольма II рода имеет следующий вид:²⁾

$$y(x) - \lambda \int_a^b K(x, s)y(s)ds = f(x), \quad x \in [a, b]. \quad (3)$$

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

²⁾Эрик Ивар Фредгольм (Erik Ivar Fredholm; 1866–1927) — Шведский математик.

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 λ , который называют параметром интегрального уравнения.

На вопросы существования решения уравнения (3) отвечает классическая теория Фредгольма (см., напр., [11], с. 77. [16], с. 48). Она применима, в частности, для непрерывных в прямоугольнике $[a, b] \times [a, b]$ ядер. Будем считать, что правая часть уравнения (3) 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 λ называется правильным или регулярным. Тогда у неоднородного уравнения при любой правой части $f(x)$ существует единственное решение. Всюду во второй главе будем считать это условие выполненным.

Приложения интегральных уравнений Фредгольма второго рода весьма разнообразны (см., напр., [11], с. 167. [3], с. 143): граничные задачи теории потенциала, граничные задачи для обыкновенных дифференциальных уравнений, граничные задачи теории упругости и т.д.

Методы решения уравнений Вольтерра II рода

§ 1. Метод квадратур

1. Описание метода. 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], [9]):

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

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

Чтобы применить метод квадратур к решению уравнения

$$y(x) - \int_a^x K(x, s)y(s)ds = f(x), \quad x \in [a, b], \quad (2)$$

необходимо использовать следующие равенства:

$$y(x_i) - \int_a^{x_i} K(x_i, s)y(s)ds = f(x_i), \quad i = 1, 2, \dots, n. \quad (3)$$

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 (3) конечной суммой. Получим систему

$$y(x_1) = f(x_1), \quad (4)$$

$$y(x_i) - \sum_{j=1}^i A_{ij} K(x_i, x_j) y(x_j) = f(x_i) + R_i, \quad i = 2, 3, \dots, n, \quad (5)$$

где $A_{i,j}$ — веса квадратурной формулы, R_i — ошибки аппроксимации. Положим ошибки R_i малыми и отбросим их. Получим систему линейных алгебраических уравнений:

$$y_1 = f_1, \quad (6)$$

$$y_i - \sum_{j=1}^i A_{ij} K_{ij} y_j = f_i, \quad i = 2, 3, \dots, n. \quad (7)$$

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), (7) дает приближенные значения искомой функции в узлах x_i .

Приведем уравнения (7) к следующему виду:

$$-\sum_{j=1}^{i-1} A_{ij} K_{ij} y_j + (1 - A_{ii} K_{ii}) y_i = f_i, \quad i = 2, 3, \dots, n, \quad (8)$$

подробнее,

$$\begin{pmatrix} 1 & & & & \\ -A_{21}K_{21} & 1 - A_{22}K_{22} & & & \\ \vdots & \vdots & \ddots & & \\ -A_{n1}K_{n1} & -A_{n2}K_{n2} & \cdots & 1 - A_{nn}K_{nn} & \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}. \quad (9)$$

Хорошо видно, что матрица коэффициентов этой системы — треугольная. Если все диагональные элементы матрицы отличны от нуля, то $y_1 = f_1$, а значения y_2, y_3, \dots, y_n можно последовательно найти по рекуррентной формуле

$$y_i = (1 - A_{ii} K_{ii})^{-1} \left(f_i + \sum_{j=1}^{i-1} A_{ij} K_{ij} y_j \right), \quad i = 2, 3, \dots, n. \quad (10)$$

Выполнения условий $(1 - A_{ii} K_{ii}) \neq 0, i = 2, 3, \dots, n$, можно добиться путем выбора узлов квадратурной формулы и обеспечения достаточной малости коэффициентов A_{ii} .

Отметим особенность выражения (10), состоящую в росте количества вычислений вместе с номером шага дискретизации из-за увеличения членов суммы, причем значения коэффициентов $A_{ij}K_{ij}$ 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 для четных i может сочетаться в конце отрезка интегрирования с какой-либо двухточечной схемой, 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

$$A_{i1} = A_{ii} = \frac{h}{2}, \quad A_{i2} = A_{i3} = \dots = A_{ii-1} = h, \quad i = 2, 3, \dots, n. \quad (11)$$

Тогда формула (10) примет следующий вид:

$$y_i = \left(1 - \frac{h}{2}K_{ii}\right)^{-1} \left(f_i + \frac{h}{2}K_{i1}y_1 + h \sum_{j=2}^{i-1} K_{ij}y_j\right), \quad i = 2, 3, \dots, n. \quad (12)$$

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

```
%Function for solving the Volterra equation of the second
%kind by the
%quadrature method. The quadrature formula of trapezoids
%on a uniform grid is used.
%Input data: K is the kernel of the equation,
%f is the right side,
 %[a,b] is the integration segment,
 %n is the number of the grid nodes,
 %y is the vector of the approximate solution at the
 %grid nodes.
function y = Volt_II_Rect(K,f,a,b,n) %Full
x=linspace(a,b,n);
```

```

h=x(2)-x(1);
y = 1:n;
y(1)=f(a);
for i=2:n
    s = f(x(i)) + h/2*K(x(i),x(1))*y(1);
    for j=2:(i-1)
        s = s + h*K(x(i),x(j))*y(j);
    end
    y(i) = s/(1 - h/2*K(x(i),x(i)));
end
end

```

Эту же процедуру перепишем, используя векторные операции. Так будет работать быстрее.

```

function y = Volt_II_Rect(K,f,a,b,n) % Speed
x=linspace(a,b,n);
h=x(2)-x(1);
y = 1:n;
y(1)=f(a);
for i=2:n
    j=2:(i-1);
    y(i) = (f(x(i)) +
    h/2*K(x(i),x(1))*y(1)+sum(h*K(x(i),x(j)).*y(j)))/...
    (1 -h/2*K(x(i),x(i)));
end
end

```

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

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

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.

Таблица 1. Результаты приближенного решения уравнения (13) 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

```

close all;
clear all;
clc;
format long;
a=0;
b=1;
K=@(x,s)exp(-(x-s));
f=@(x)exp(-x);
u=@(x)ones(1,5);
n=5;
x = linspace(a,b,n);
plot(x,u(x));
hold on;
y=Volt_II_Rect(K,f,a,b,n)
plot(x,y,'or');
axis([a,b,0.99,1.01]);
n=21;
x = linspace(a,b,n);
y=Volt_II_Rect(K,f,a,b,n)
plot(x,y,'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.

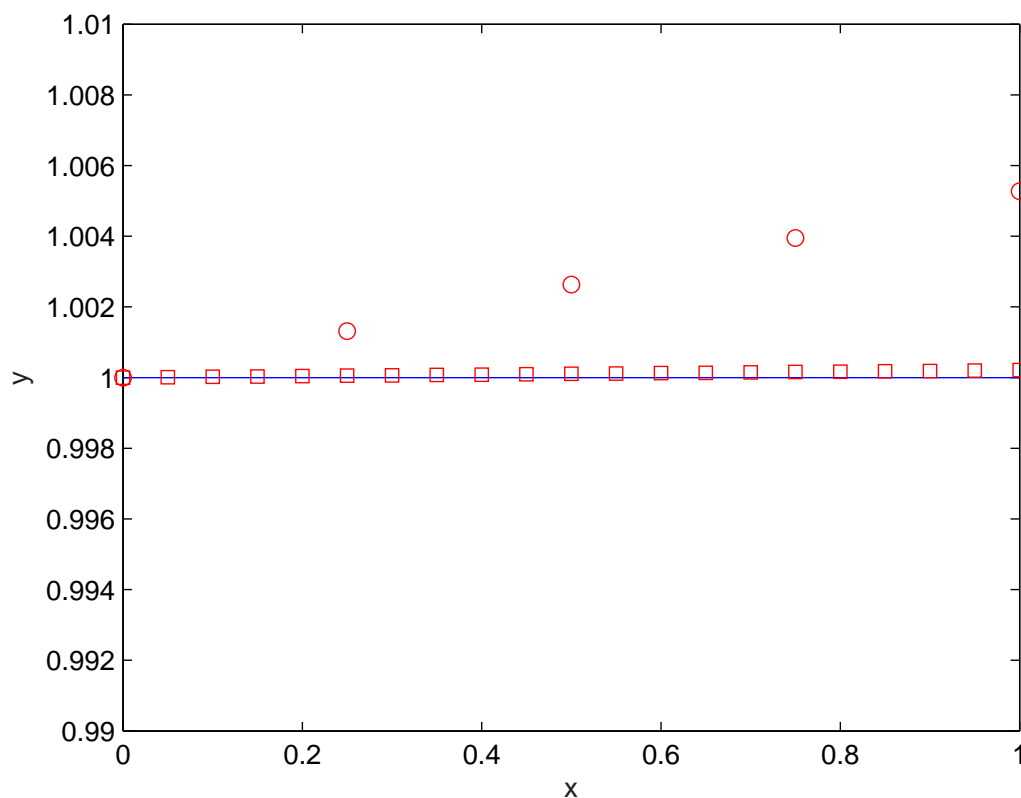


Рис. 1. Результаты приближенного решения уравнения (13) методом квадратур, основанным на применении квадратурной формулы трапеций с равномерной сеткой. Непрерывной линией обозначено точное решение, кружочками — приближенное решение при шаге сетки $h = 0.25$, квадратиками — при $h = 0.005$.

Упражнения

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

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

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

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

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

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.

- 6) Напишите функцию, реализующую метод квадратур на основе квадратурной формулы Симпсона (см., напр., [5], с. 342). Решите с помощью новой функции уравнение (14) и сравните эффективность использования формул Симпсона и трапеций.

Указания

К упражнению 3. Поведение относительной ошибки решения в зависимости от шага сетки h традиционно исследуют в связи с тем, что теоретические оценки точности методов строятся в терминах h . Конечно, в исследуемой в этом разделе программе для этого надо изменять число узлов сетки.

К упражнению 5. Неравномерную сетку можно создать, например, с помощью следующих фрагментов кода, сгущающих узлы у левой или у правой границы отрезка интегрирования.

```

Mesh = @(x) exp(x)-1;
pseudo_b = log(b+1);
pseudo_a = log(a+1);
pseudo_x = linspace(pseudo_a,pseudo_b,n);
x = Mesh(pseudo_x);

Mesh = @(x) (-exp(-x)+a+1)*(b-a+1);
pseudo_b = -log(a+1-b/(b-a+1));
pseudo_a = -log(a+1-a/(b-a+1));
pseudo_x = linspace(pseudo_a,pseudo_b,n);
x = Mesh(pseudo_x);

```

К упражнению 6. Формулы Симпсона предполагают нечетное число узлов в сетке. Поэтому можно, например, сочетать их с формулой трапеций в конце отрезка интегрирования при четном числе узлов. В этом случае в формуле (1) полагают:

$$A_{11} = A_{12} = \frac{h}{2},$$

$$A_{i1} = \frac{h}{3},$$

если $(\text{mod}(i,2)=1)$, то

$$A_{i2} = A_{i4} = \dots = A_{i,i-1} = \frac{4h}{3}, \quad (15)$$

$$A_{i3} = A_{i5} = \dots = A_{i,i-2} = \frac{2h}{3},$$

$$A_{ii} = \frac{h}{3},$$

если $\text{mod}(i,2)=0$, то

$$A_{i2} = A_{i4} = \dots = A_{i,i-2} = \frac{4h}{3},$$

$$A_{i3} = A_{i5} = \dots = A_{i,i-3} = \frac{2h}{3}, \quad A_{i,i-1} = \frac{5h}{6},$$

$$A_{ii} = \frac{h}{2},$$

где $i = 3, 4, \dots, n$. Тогда вычисления могут быть реализованы, например, с помощью такой функции.

```
function y = Volt_II_Rect_Simpson(K,f,a,b,n)
h = (b-a)/(n-1);
x=linspace(a,b,n);
y = 1:n;
y(1)=f(x(1));
y(2)=(f(x(2)) + h/2*K(x(2),x(1))*y(1))/...
(1 - h/2*K(x(2),x(2)));
for i=3:n
    if mod(i,2)==1
        A = h.*[1, 2*(1+mod((1:i-2),2)), 1] ./3;
```

```

else
    A = h.*[1/3, 2/3*(1+mod((1:i-3),2)), 5/6, 1/2];
end
j=1:(i-1);
y(i) = (f(x(i))+sum(A(j).*K(x(i),x(j)).*y(j)))/ ...
(1 - A(i)*K(x(i),x(i)));
end
end

```

Недостаток у подобной реализации следующий: проверка условия и вычисление весов квадратурной формулы происходят циклически, что замедляет работу функции. Решение этой проблемы может быть, например, таким:

```

function y = Volt_II_Rect_Simpson(K,f,a,b,n)
h = (b-a)/(n-1);
rev_h = (n-1)/(b-a);
x=linspace(a,b,n);
y = 1:n;
y(1)=f(x(1));
y(2)=(f(x(2)) + h/2*K(x(2),x(1))*y(1))/...
(1 - h/2*K(x(2),x(2)));
A = [1, 2*(1+mod(1:(n-3),2))]./3;
for i=3:n
    j=1:(i-2);
    y(i) = (f(x(i))/h+sum(A(j).*K(x(i),x(j)).*y(j))+...
(mod(i,2)*(1/2)+(5/6))*K(x(i),x(i-1))*y(i-1))/...
(rev_h - 1/(2+mod(i,2))*K(x(i),x(i))));
end
end

```

§ 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 $a = x_1 < x_2 < \dots < x_n = b$. 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

$$y_1 = f_1, \quad (3)$$

$$y_i - \sum_{j=1}^i A_{ij} K_{ij}(y_j) = f_i, \quad i = 2, 3, \dots, n. \quad (4)$$

Здесь $A_{i,j}$ — веса квадратурной формулы, $f_i = f(x_i)$, $y_i = \tilde{y}(x_i)$, \tilde{y} — приближение к искомой функции y , $K_{ij}(y_j) = K(x_i, x_j, \tilde{y}(x_j))$. Равенство (3) определяет y_1 . Соотношения (4) позволяют далее находить значения y_2, y_3, \dots, y_n приближенного решения в узлах сетки путем последовательного решения $n - 1$ нелинейного уравнения:

$$y_i - A_{ii} K_{ii}(y_i) = f_i + \sum_{j=1}^{i-1} A_{ij} K_{ij}(y_j), \quad i = 2, 3, \dots, n. \quad (5)$$

2. Пример. Решим предложенным выше методом нелинейное уравнение

$$y(x) - \int_0^x e^{-(x-s)} y^2(s) ds = e^{-x}, \quad x \in [0, 0.1]. \quad (6)$$

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:

$$y(x_i) - \int_0^{x_i} e^{-(x_i-s)} y^2(s) ds = e^{-x_i}, \quad i = 1, 2, \dots, n. \quad (7)$$

Пусть узлы x_i образуют на отрезке $[a, b]$ равномерную сетку с шагом h . Заменяем интегралы в (7) конечными суммами с помощью формулы трапеций. Согласно (3), (5) получим следующую систему расчетных соотношений:

$$y_1 = f_1, \quad y_i - \frac{h}{2} K_{ii} y_i^2 = f_i + \frac{h}{2} K_{i1} y_1^2 + \sum_{j=2}^{i-1} h K_{ij} y_j^2, \quad i = 2, 3, \dots, n. \quad (8)$$

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

$$c_i = f_i + \frac{h}{2} K_{i1} y_1^2 + \sum_{j=2}^{i-1} h K_{ij} y_j^2, \quad i = 2, 3, \dots, n.$$

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

$$y_1 = f_1, \quad y_i = \frac{1 \pm \sqrt{1 - 4 \frac{h}{2} c_i}}{h}, \quad i = 2, 3, \dots, n. \quad (9)$$

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

Таблица 2. Результаты приближенного решения уравнения (6) 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	1.000000000000000
0.02	1.00	1.00000067351003
0.04	1.00	1.00000136062954
0.06	1.00	1.00000206163358
0.08	1.00	1.00000277680270
0.10	1.00	1.00000350642317

3. Компьютерные программы. Напишем на языке Matlab функцию `Nonlin_Volt_II_Rect.m`, реализующую вычисления по формулам (9).

```
%Function for solving the nonlinear Volterra equation of
%the second kind by the
%quadrature method. The quadrature formula of trapezoids
%on a uniform grid is used.
%Input data: K is the kernel of the equation,
%f is the right side,
%[a,b] is the integration segment,
%n is the number of the grid nodes,
%y is the vector of the approximate solution at the
%grid nodes.
function [y,x] = Nonlin_Volt_II_Rect(a,b,n,K,f)
y=1:n;
x=linspace(a,b,n);
h = x(2)-x(1);
y(1)=f(x(1));
for i=2:n
    c=f(x(i))+h/2*K(x(i),x(1))*y(1)*y(1);
    for j=2:(i-1)
        c=c+h*K(x(i),x(j))*y(j)*y(j);
    end
    y(i)=(1-sqrt(1-4*(h/2)*c))/h;
end
end
```

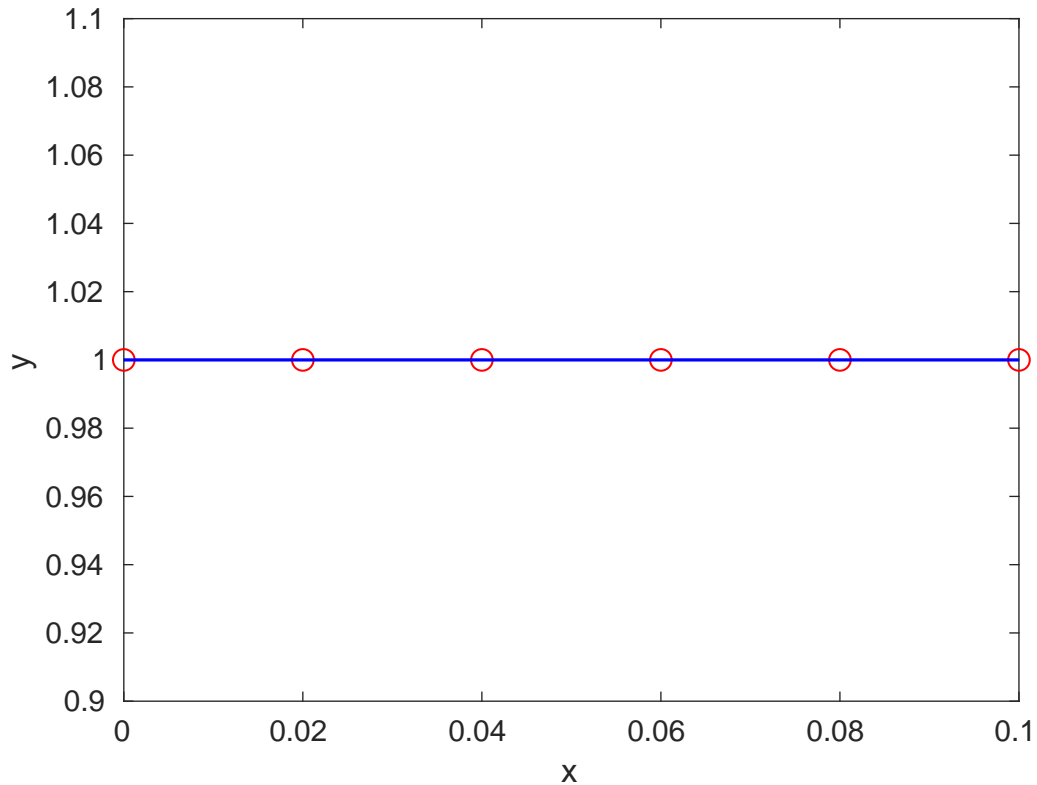


Рис. 2. Результаты приближенного решения уравнения (6) методом квадратур, основанным на применении квадратурной формулы трапеций с равномерной сеткой. Непрерывной линией обозначено точное решение, кружочками — приближенное решение при шаге сетки $h = 0.02$.

Теперь напишем на языке Matlab сценарий решения уравнения (6).

```
close all;clear all;
a=0;
b=0.1;
n=6;
x = linspace(a,b,n);
K = @(x,s)exp(-(x-s));
f = @(x)exp(-x);
y=Nonlin_Volt_II_Rect(a,b,n,K,f); %aprocsimate solution
u=@(x)ones(1,numel(x)); %exact solution
plot(x,y,'or');
hold on;
plot(x,u(x),'b');
axis([ a,b,1-0.1,1+0.1]);
```

```
xlabel('x');  
ylabel('y');  
hold off;
```

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 нормы относительной ошибки решения от h в более широком диапазоне изменений шага.
- 3) Как изменится решение, если в формуле (9) перед корнем оставить знак «+»?
- 4) Напишите функцию, реализующую вычисления по общим формулам (5). Решите с помощью этой функции уравнение (6). Проанализируйте точность вычислений.

Указания

К упражнению 3. Если в формуле (9) перед корнем поставить знак «+», то Matlab будет разыскивать решение исходной задачи, по сути, в пространстве комплекснозначных функций. Оценивают качество решения в условиях, когда точное решение не известно, обычно с помощью невязки:

$$R(x) = y(x) - \int_a^x K(x, s, y(s)) ds - f(x), \quad x \in [a, b]. \quad (10)$$

Относительная невязка может быть вычислена, например, с помощью следующей функции.

```
%Function for calculating the relative residual  
%Input data: K is the kernel of the equation,  
%f is the right side,  
%x is the grid nodes vector
```

```
%y is the vector of the approximate solution at the grid nodes.  
function res = rel_residual(y,x,K,f)  
n = numel(y);  
res = 0;  
for i =2:n  
    j = 1:i;  
    res = max(res,abs(y(i) -...  
    trapz(x(j),K(x(i),x(j)).*y(j).*y(j))-f(x(i)))));  
end  
res = res/max(abs(y));  
end
```

§ 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 , $i = 0, 1, \dots, n$, так что $x_0 = a$, а $x_n = b$. Пусть $K_{ij} = K(x_i, x_j)$, $y_{ki} = y_k(x_i)$. Получим расчетное выражение

$$\begin{aligned} y_{k+1}(x_0) &= f(x_0) \\ y_{k+1}(x_i) &= f(x_i) + \int_a^{x_i} K(x_i, s)y_k(s)ds \approx \\ &\approx f(x_i) + \frac{h}{2} [K_{i0}y_{k0} + 2(K_{i1}y_{k1} + K_{i2}y_{k2} + \dots + K_{i,i-1}y_{k,i-1}) + K_{ii}y_{ki}], \end{aligned} \quad (3)$$

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

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

где $\|y\| = \max_{a \leq x \leq b} |y(x)|$, ε — 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 `VoltII_SimpleIter.m`, предназначенную для приближенного решения уравнения (1) методом простой итерации.

```
%Function for solving the nonlinear Volterra equation
%of the second kind by the fixed-point iteration method.
%The quadrature formula of trapezoids
%on a uniform grid is used.
%Input data: K is the kernel of the equation,
%f is the right side,
%a is the beginning of the integration segment,
%b is the end of the segment, n is the number of grid
%nodes
%tol is the calculation accuracy,
%y0 is the initial approximation
%Output: x is the grid nodes vector
%y is the vector of the approximate solution at the grid
% nodes.
%iter is the the number of iterations required for
% convergence
function [y, x, iter] = ...
    VoltII_SimpleIter(a,b,n,K,f,tol,y0)
x = linspace(a,b,n);
h = x(2)-x(1);
F = f(x);
if(nargin<=6) y0=F;
end
if(nargin<=5) tol=1e-10;
end
y = ones(1,n)*inf;
yk = y0;
```

```

iter = 0;
yk(1)=F(1);
y(1) = F(1);
while(norm(yk-y,inf)/norm(yk,inf)>tol)
    y = yk;
    iter=iter+1;
    for i = 2:n
        j = 2:i-1;
        yk(i) = F(i)+h/2*K(x(i),x(1))*y(1)+...
            h/2*K(x(i),x(i))*y(i)+...
            sum(h*K(x(i),x(j)).*y(j));
    end
end
end

```

3. Пример. Решим с помощью функции `VoltII_SimpleIter.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 точность решения $\text{tol} = 10^{-3}$. 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...>>
clear all;
close all;
f = @(x) ones(1,numel(x));
K = @(x,s) ones(1,numel(x));
a = 0;
b = 7;
n=101;%h=(b-a)/(n-1)=0.07
tol = 1e-03;

```

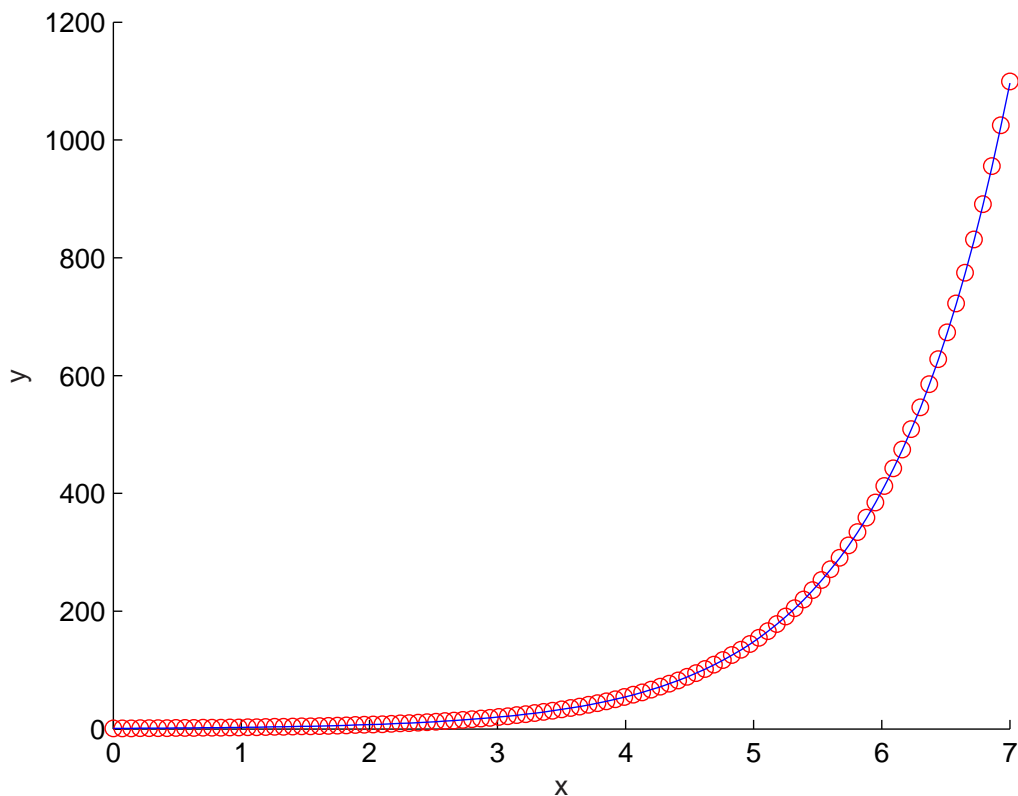



Рис. 3. Результаты приближенного решения уравнения (5) методом простой итерации, основанным на применении квадратурной формулы трапеций с равномерной сеткой. Непрерывной линией обозначено точное решение, кружочками — приближенное решение при шаге сетки $h = 0.07$.

```
u = @(x) exp(x);  
[y, x, iter] = VoltII_SimpleIter(a,b,n,K,f,tol);  
plot(x,y,'o',x,u(x),'r');  
er = norm(u(x)-y,inf)/norm(u(x),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 (15), с. 13. Решите с помощью этой функции уравнения из этого параграфа и сравните эффективность применения метода Симпсона с использованием метода трапеций.
- 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]).

- 6) Перепишите процедуру `VoltII_SimpleIter.m` так, чтобы не требовалось хранить одновременно векторы y и uk .

Указания

К упражнению 6. Решение может выглядеть, например, следующим образом.

```
function [y, x, iter] = VoltII_SimpleIter(a,b,n,K,f,tol,y0)
x = linspace(a,b,n);
h = x(2)-x(1);
F = f(x);
if(nargin<=6) y0=F; end
if(nargin<=5) tol=1e-10; end
y = y0;
iter = 0;
delta = inf;
```

```
max_y = max(abs(y));
y(1) = F(1);
while(delta/max_y>tol)
    iter=iter+1;
    delta = 0;
    max_y = abs(y(1));
    for i = 2:n
        buf = y(i);
        max_y = max(max_y,abs(buf));
        j = 2:i-1;
        y(i) = F(i)+h/2*K(x(i),x(1))*y(1)+...
            h/2*K(x(i),x(i))*y(i)+...
            sum(h*K(x(i),x(j)).*y(j));
        delta = max(delta,abs(y(i)-buf));
    end
end
end
```

ГЛАВА 2

Методы решения уравнений Фредгольма II рода

§ 1. Метод квадратур

1. Описание метода. Найдем приближенное решение уравнения

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

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 $j = 2, 3, \dots, n - 1$.

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

```
%Function for solving the Fredholm equation
%of the second kind by the quadrature method.
%The quadrature formula of trapezoids
%on a uniform grid is used.
%Input data: K is the kernel of the equation,
%f is the right side,
%lambda is the constant from equation
%a is the beginning of the integration segment,
%b is the end of the segment, n is the number of grid nodes
%Output: x is the grid nodes vector
%y is the vector of the approximate solution at
%the grid nodes.
function [y, x] = fredII_trapez(K,f,lambda,a,b,n)
x = linspace(a,b,n)';
h = x(2) - x(1);
w = [1/2, ones(1,n-2) ,1/2]*h*lambda;
A = zeros(n);
for i = 1:n
    A(:,i) = -w(i)*K(x,x(i));
    A(i,i) = A(i,i)+1;
end
F = f(x);
y = A\F;
end
```

3. Пример. Выполним с помощью функции `fredII_trapez.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 ре-

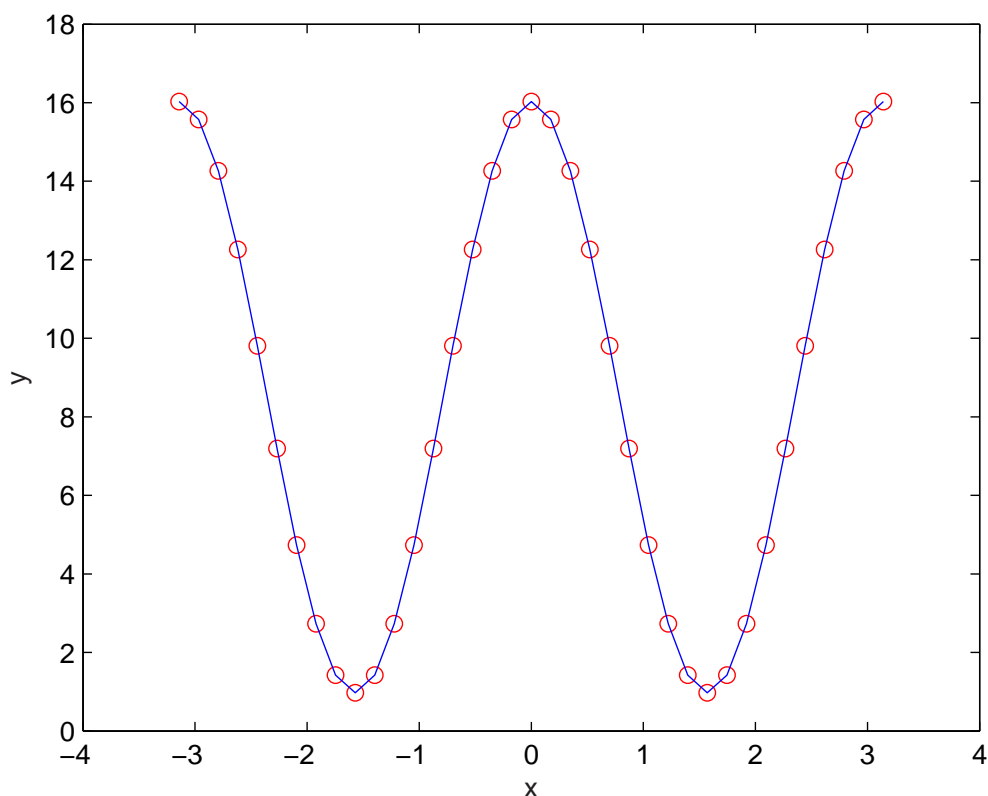


Рис. 1. Результаты решения примера 3, с. 28. Непрерывной линией обозначено точное решение, кружочками — приближенное решение.

шение 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.

```
% Сценарий решения задачи 3.9, с. 162, из книги
% Верлань А.Ф, Сизиков В.С. <<Интегральные уравнения...>>
\begin{verbatim}
clear all;close all;
K = @(x,s) 1./(0.64*cos((x+s)/2).*cos((x+s)/2)-1);
f = @(x) 25-16*sin(x).*sin(x);
lambda = 3/(10*pi);
a = -pi;
b = pi;
u = @(x) 17/2 + (128/17)*cos(2*x);
\end{verbatim}
```

Таблица 1. Результаты решения примера 3, с. 28.

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

```

n = 37;
[y, x] = fredII_trapez(K,f,lambda,a,b,n);
plot(x,y,'ro',x,u(x),'g');
xlabel('x')
ylabel('y')
err = norm(y-u(x),inf)/norm(u(x),inf)

```

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) Замените в функции `Fred_II_Rect.m` цикл `for` с векторными вычислениям матричными операциями и сравните скорость работы нового и старого вариантов функции.
- 4) Напишите функцию, реализующую метод квадратур, на основе квадратурной формулы Симпсона (15), с. 13. 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.

Указания

К упражнению 3. Возможны, например, два таких варианта решения поставленной задачи:

```
A = -K(x,x')*diag(w')+eye(n);
```

и

```
A = -K(x,x')*spdiags(w',0,n,n)+speye(n);
```

Попробуйте оба и объясните, почему один работает быстрее другого. Сравните скорость работы лучшего из матричных вариантов со скоростью работы исходного кода.

К упражнению 4. При решении этой задачи на сетке с чётным числом узлов, возникает та же проблема, что в шестом упражнении 1-го параграфа 1-ой главы (см. с. 12).

В отличие от того случая, значения интегрируемой функции во всех узлах нам известны заранее, поэтому использование формулы трапеций возможно с обеих сторон отрезка интегрирования. Код решения задачи может быть, например, таким (трапеции слева):

```
function [y, x] = fred2Simpson(K,f,lambda,a,b,n)
x = linspace(a,b,n)';
h = x(2) - x(1);
if mod(n,2)==1
    w = [1/3, 2/3*(1+mod(1:(n-2),2)),1/3]*h*lambda;
else
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    w = [1/2, 5/6, 2/3*(1+mod(1:(n-3),2)),1/3]*h*lambda;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
A = zeros(n);
for i = 1:n
    A(:,i) = -w(i)*K(x,x(i));
```



```

    A(i,i) = A(i,i)+1;
end
F = f(x);
y = A\F;
end

```

При замене выделенного участка кода на

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    w = [1/3, 2/3*(1+mod(1:(n-3),2)),5/6,1/2]*h*lambda;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

получим доопределение трапециями справа. Сравните погрешности решений при использовании формулы трапеций с разных сторон отрезка интегрирования и объясните разницу.

§ 2. Метод вырожденных ядер

1. Решение уравнения с вырожденным ядром. Ядро интегрального уравнения Фредгольма второго рода

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

называется вырожденным, если оно имеет следующий вид:

$$K(x, s) = \sum_{i=1}^m \alpha_i(x)\beta_i(s). \quad (2)$$

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

Подставим ядро (2) в уравнение (1). Получим уравнение с вырожденным ядром:

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

Поменяем в (3) местами операции интегрирования и суммирования:

$$y(x) - \lambda \sum_{i=1}^m \alpha_i(x) \int_a^b \beta_i(s)y(s)ds = f(x), \quad x \in [a, b]. \quad (4)$$

Введем обозначения для интегралов в (4):

$$c_i = \int_a^b \beta_i(s)y(s)ds, \quad i = 1, 2, \dots, m. \quad (5)$$

Из равенства (4) получим следующее представление решения уравнения с вырожденным ядром (3):

$$y(x) = f(x) + \lambda \sum_{i=1}^m c_i \alpha_i(x), \quad x \in [a, b]. \quad (6)$$

Отметим, что константы c_i неизвестны, они являются интегралами от искомой функции. Для их вычисления построим систему линейных алгебраических уравнений. Подставим (6) в (3). Получим равенство

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

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

$$c_i - \int_a^b \beta_i(s) \left[f(s) + \lambda \sum_{j=1}^m c_j \alpha_j(s) \right] ds = 0, \quad i = 1, 2, \dots, m, \quad (8)$$

или

$$c_i - \lambda \sum_{j=1}^m c_j \int_a^b \alpha_j(s) \beta_i(s) ds = \int_a^b \beta_i(s) f(s) ds, \quad i = 1, 2, \dots, m. \quad (9)$$

Обозначим

$$a_{ij} = \int_a^b \alpha_j(s) \beta_i(s) ds, \quad i, j = 1, 2, \dots, m. \quad (10)$$

$$f_i = \int_a^b \beta_i(s) f(s) ds, \quad i = 1, 2, \dots, m. \quad (11)$$

Запишем равенства (9) в виде системы линейных алгебраических уравнений относительно неизвестных c_i :

$$c_i - \lambda \sum_{j=1}^m a_{ij} c_j = f_i, \quad i = 1, 2, \dots, m, \quad (12)$$

подробнее,

$$\begin{pmatrix} 1 - \lambda a_{11} & -\lambda a_{12} & \cdots & -\lambda a_{1m} \\ -\lambda a_{21} & 1 - \lambda a_{22} & \cdots & -\lambda a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ -\lambda a_{m1} & -\lambda a_{m2} & \cdots & 1 - \lambda a_{mm} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \cdots \\ c_m \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \cdots \\ f_m \end{pmatrix}. \quad (13)$$

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

Таким образом, алгоритм решения интегрального уравнения с вырожденным ядром (3) состоит в вычислении интегралов (10), (11) и решении системы линейных алгебраических уравнений (13). Решение уравнения (3) затем записывается аналитически в виде (6).

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

```
%Function for solving the Fredholm equation
%of the second kind with a degenerate kernel.
%Input data: alpha, beta are the kernel of the equation,
%f is the right side,
%lambda is the parameter of the equation
%a is the beginning of the integration segment,
%b is the end of the segment,
%Output:
%y is the analytical solution.
function[ y ] = Degenerate_Fred_II(alpha,beta,f,a,b,lambda)
n = numel(alpha);
A = zeros(n);
for j=1:n
    A(:,j)= -lambda*int(beta*alpha(j),a, b);
    A(j,j)= 1 + A(j,j);
end
r = int(beta*f, a, b)'; %right side
c = A\r;
y = dot(c,alpha)*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.

```
close all; clear all;
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(alpha,beta,f,a,b,lambda)
```

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

$$y(x) + \int_0^1 x(e^{xs} - 1)y(s)ds = e^x - x, \quad x \in [0, 1]. \quad (14)$$

Точное решение уравнения $y \equiv 1$ (см. пример 3.12, с. 171, [3]). Разложим ядро этого уравнения в ряд Тейлора, ограничившись пятью 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.

$$1 - \frac{4589889984855}{4503599627370496} x^2 - \frac{24810467431825}{54043195528445952} x^3 - \frac{14994606363791}{108086391056891904} x^4$$

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])

$$y(x) - \int_0^{1/2} \exp(-x^2 s^2) y(s) ds = 1, \quad x \in [0, 1/2]. \quad (15)$$

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 (15) 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 (14) и (15).
- 4) Исследуйте численно, как зависит точность решения уравнений (14) и (15) от числа членов в ряде Тейлора, аппроксимирующем ядро уравнения.

5) Решите уравнение (см. пример. 1, с. 357, [5])

$$y(x) = x^2 + \int_{-1}^1 \operatorname{sh}(x+s)y(s)ds, \quad x \in [-1, 1]. \quad (16)$$

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

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

имеет следующий вид:

$$Ry(x) = y(x) - \lambda \int_a^b K(x, s)y(s)ds - f(x), \quad x \in [a, b]. \quad (2)$$

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

$$\tilde{y}(x) = \sum_{i=1}^n c_i \varphi_i(x). \quad (3)$$

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 (3) в (2) и положим

$$\varepsilon(x, c) = Ry(x).$$

Тогда для ненулевой невязки имеем

$$\varepsilon(x, c) = \sum_{i=1}^n c_i \left[\varphi_i(x) - \lambda \int_a^b K(x, s)\varphi_i(s)ds \right] - f(x), \quad x \in [a, b]. \quad (4)$$

Постоянные c_i найдем из условий минимума функционала

$$J(c) = \int_a^b \varepsilon^2(x, c)dx, \quad (5)$$

т.е. из условий

$$\frac{\partial J}{\partial c_i} = 0, \quad i = 1, 2, \dots, n. \quad (6)$$

Подставим (4) в (5):

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

Вычислим частные производные:

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

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

$$\sum_{j=1}^n a_{ij} c_j = b_i, \quad i = 1, 2, \dots, n, \quad (7)$$

где

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

Решим эту систему и запишем приближенное решение исходного интегрального уравнения в виде (3). Теоретически этот метод обоснован, например, в [13], с. 453.

2. Компьютерные программы. Напишем на языке Matlab функцию `MLS_Fred_II_sym`, реализующую этот метод.

```

%Function for solving the Fredholm equation
%of the second kind by the Least Squares method.
%Input data: Ker is the kernel of the equation,
%f is the right side,
%fii(x,i) returns the values of the i-th basis function on the
%grid x
%m is the number of basis functions
%lambda is the constant from equation
%a is the beginning of the integration segment,
%b is the end of the segment
function [c] = MLS_Fred_II_sym(K,f,fii,n,lambda,a,b)
    syms x s
    I = fii(x,1:n) - lambda*int(K(x,s)*fii(s,1:n),s,a,b);
    A = int(I'*I,x,a,b);
    B = int(f(x)*I',x,a,b);
    c=A\B;
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 на языке Matlab выглядит следующим образом.

```

clear all; close all;
a=-1;
b=1;
K=@(x,s) x*s; % kernel
f=@(x) x; % right side
fii = @(x,i) x.^(i-1); % basis functions
y_exact=@(x) 3*x; % exact solution
lambda=1; % constant from the equation
n=2; % number of busis functions

```

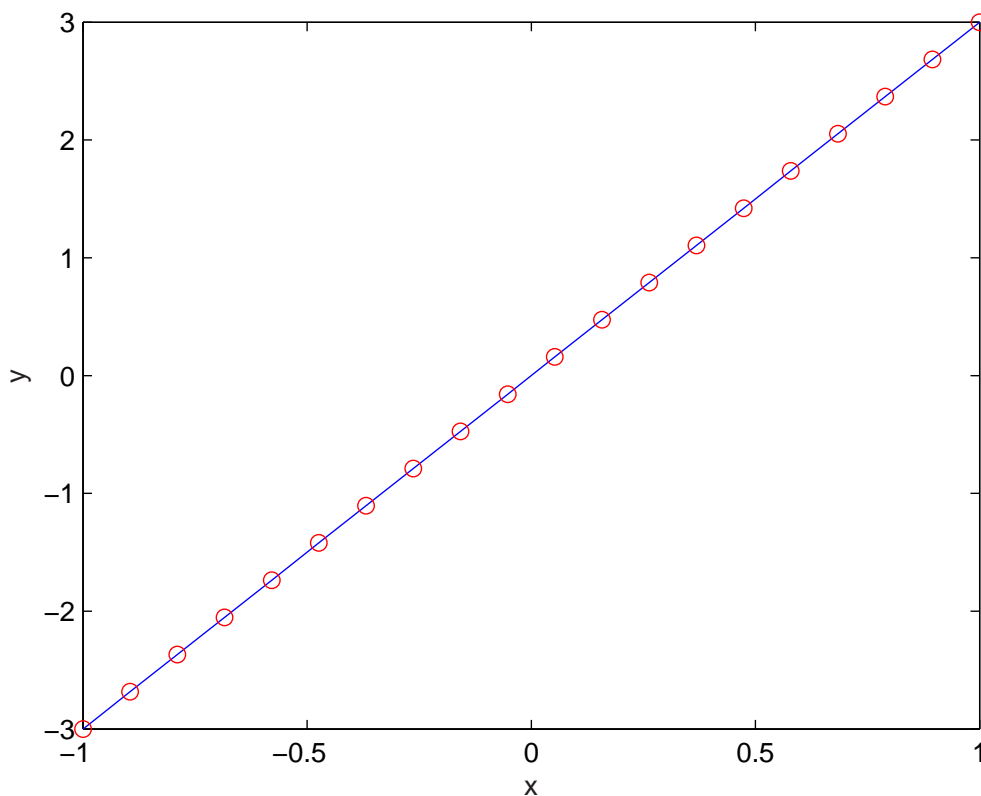


Рис. 2. Результаты решения задачи 3.24, с. 206, из книги [3]. Непрерывной линией обозначено точное решение, кружочками — приближенное решение.

```
c = MLS_Fred_II_sym(K,f,fii,n,lambda,a,b)
y = @(x) fii(x,1:n)*c;
testGrid = linspace(a,b,20)';
plot(testGrid,y_exact(testGrid),'b',testGrid,...
     y(testGrid),'ro')
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
```

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 `MLS_Fred_II_sym` найдите приближенное решение уравнения (см. упражнение 2, с. 30)

$$y(x) = \frac{5}{6}x + \frac{1}{2} \int_0^1 xsy(s)ds, \quad x \in [0, 1]. \quad (8)$$

Его точное решение $y(x) = x$ (см. пример 3.8, с. 158, [3]). Исследуйте зависимость точности решения от параметра n .

- 4) С помощью функции `MLS_Fred_II_sym` найдите приближенное решение уравнения (см., раздел 4, с. 36)

$$y(x) + \int_0^1 x(e^{xs} - 1)y(s)ds = e^x - x, \quad x \in [0, 1]. \quad (9)$$

Точное решение уравнения $y \equiv 1$ (см. пример 3.12, с. 171, [3]). Исследуйте зависимость точности решения от параметра n .

- 5) Методом наименьших квадратов найдите приближенное решение уравнения (см. упражнение 5, с. 39)

$$y(x) = x^2 + \int_{-1}^1 \operatorname{sh}(x+s)y(s)ds, \quad x \in [-1, 1]. \quad (10)$$

Напомним, что его точное решение имеет вид:

$$y(x) = x^2 + \alpha \operatorname{sh} x + \beta \operatorname{ch} x, \quad (11)$$

где

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

Разыскивайте приближенное решение этого уравнения в виде

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

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

- 6) Теперь методом наименьших квадратов найдите приближенное решение уравнения (10), разыскивая его в виде

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

Сравните приближенное решение с точным (11).

- 7) Замените в функции `MLS_Fred_II_sym` символьное вычисление интегралов матричными операциями и сравните скорость и точность работы нового и старого вариантов функции на примерах, рассмотренных выше.
- 8) В сценарии использования функции `MLS_Fred_II_sym` при решении тестового примера базисные функции `fii` задаются как неявные функции двух переменных x и i . Это эффективно, если базисные функции регулярно зависят от индекса i . Разработайте способ задания базисных функций, определяемых произвольным образом.

Указания

К упражнению 7. Возможен, например, такой вариант решения поставленной задачи:

```
%Function for solving the Fredholm integral equation
%of the second kind by the Finite Difference method.
%Input data: Ker is the kernel of the equation,
%f is the right side,
%basisFunction(x,i) returns the values of the i-th
%basis function on the
%grid x
%m is the number of basis functions
%lambda is the constant from equation
```

```

%a is the beginning of the integration segment,
%b is the end of the segment,
%n is the grid size for numerical integration
function c = finiteDifference_Fred_II(Ker,f,...
    basisFunction,m,lambda,a,b,n)
    if(nargin<=7) n = 100; end
%Nodes and weights of a quadrature method
    grid = linspace(a,b,n);
    weights = [1/2, ones(1,n-2) ,1/2]*(grid(2)-grid(1));
%Change it, try your luck.
    K = Ker(grid', grid);
    Basis = basisFunction(grid',1:m);
    W = spdiags(weights',0,n,n);
    preA = Basis-lambda*K*W*Basis;
    A = preA'*W*preA;
    F = (f(grid).*weights*preA)';
    c = A\F;
en

close all; clear all;
a = -1;
b = 1;
f = @(x) x;
Ker = @(x,s) x.*s;
u = @(x) 3*x;
lambda = 1;
basisFunction = @(x,i) x.^(i-1);
m = 2;
n = 100;
c = finiteDifference_Fred_II(Ker,f,...
    basisFunction,m,lambda,a,b,n)
y = @(t) basisFunction(t,1:m)*c;
testGrid = linspace(a,b,20)';
plot(testGrid,u(testGrid),'b')
hold on
plot(testGrid,y(testGrid),'ro')
xlabel('x');
ylabel('y')

```

К упражнению 8. Можно, например, воспользоваться оператором `switch` или функцией `piecewise`.

§ 4. Метод Галеркина — Петрова

1. Описание метода. Пусть решение уравнения

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

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 (1) запишем в виде

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

где c_j — подлежащие определению коэффициенты. Ясно, что если невязка

$$R\tilde{y}(x) = \tilde{y}(x) - \lambda \int_a^b K(x, s)\tilde{y}(s)ds - f(x), \quad x \in [a, b]. \quad (3)$$

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 $\tilde{y}(x)$:

$$(R\tilde{y}, \psi_i) = 0, \quad i = 1, 2, \dots, n. \quad (4)$$

Пусть скалярное произведение определено формулой

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

Тогда равенства (4) представляют собой систему линейных алгебраических уравнений относительно неизвестных коэффициентов c_j :

$$\sum_{j=1}^n a_{ij}c_j = b_i, \quad i = 1, 2, \dots, n. \quad (5)$$

Здесь

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

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

Решив систему (5), найдем коэффициенты в представлении приближенного решения (2). В наиболее общем виде этот метод обоснован, например, в [8], 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
V=zeros(n,1); % правая часть СЛАУ
A=zeros(n); % матрица СЛАУ
for j=1:n
    V(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

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

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
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(x),'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 .

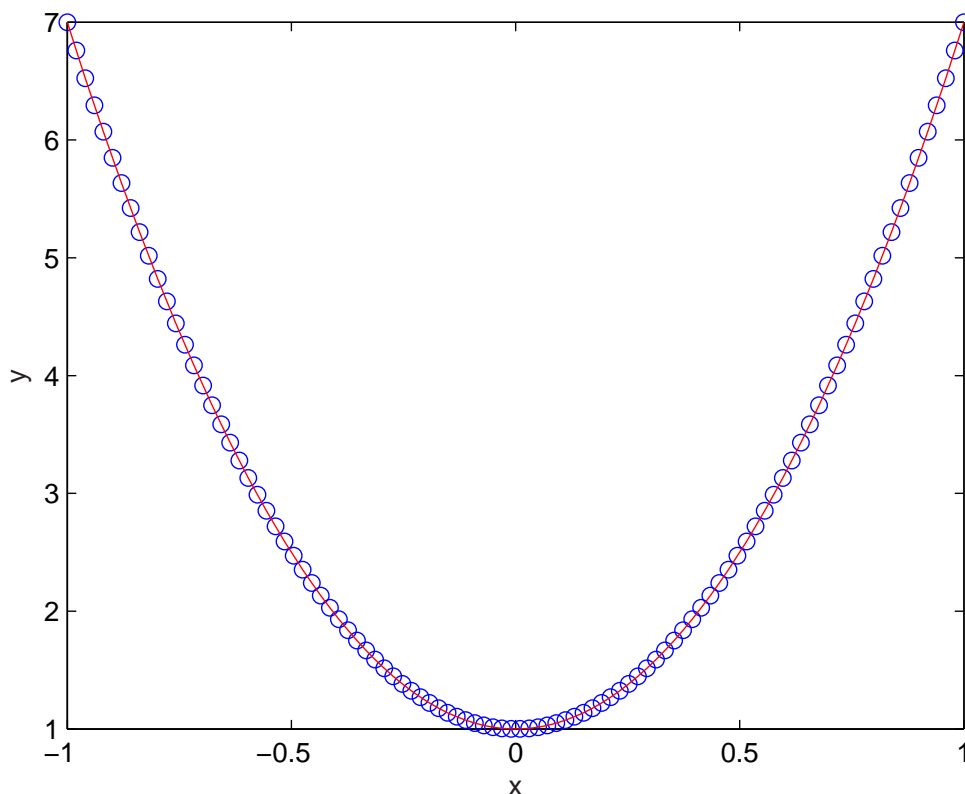


Рис. 3. Результаты решения уравнения (6) методом Галеркина — Петрова. Непрерывной линией обозначено точное решение, кружочками — приближенное решение.

- 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 (6). 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 построите самостоятельно). Исследуйте зависимость точности решения от n .

§ 5. Метод моментов

1. Описание метода и компьютерная программа. Метод моментов, по существу, является методом Бубнова — Галеркина (см. упражнение 4, с. 52), в котором система функций $\varphi_i(x)$, $i = 1, 2, \dots, n$, выбирается специальным образом. Сначала фиксируется некоторая функция $\varphi_1(x)$, а затем остальные координатные функции определяются с помощью следующих выражений (см., например, [3], с. 211):

$$\varphi_2 = \int_a^b K(s, t)\varphi_1(s)ds, \quad (1)$$

...

$$\varphi_n = \int_a^b K(s, t)\varphi_{n-1}(s)ds. \quad (2)$$

В связи с этим, напомним функцию `BubnovGalerkin_Fred_II.m`, которая реализует алгоритм решения интегрального уравнения Фредгольма второго рода по методу Бубнова — Галеркина.

```
% Функция для решения уравнения Фредгольма второго
% методом Бубнова-Галеркина
% Входные данные: lambda - параметр ИУ,
% K - ядро ИУ,
% f - правая часть (задаются аналитически),
% a - начало отрезка интегрирования,
% b - конец отрезка,
% fi - массив базисных функций
% n - число базисных функций
% Результат: y - вектор значений приближенного решения,
% C - найденные коэффициенты.
% Авторы: Сергеев П.
function [ C, y ] = BubnovGalerkin_Fred_II(K, f, a, b,...
    lambda, fi, n)

    syms x s;
    for i = 1 : n
        for j = 1 : n
```

```

        alpha(i, j) = double(int(fi{i} .* fi{j}, a, b));
        betta(i, j) = double(int(int(fi{i} .* ...
            K(x, s) .* subs(fi{j}, 'x', s), s, a, b), a, b));
    end
    gamma(i) = double(int(int(fi{i} .* K(x, s) .* ...
        subs(f, x, s), s, a, b), a, b));
end
A = alpha - lambda * betta;
C = A \ (lambda * gamma)';
x = linspace(a, b);
    %сетка из 100 узлов на отрезке [a, b]
y = f(x);
for i = 1 : n
    y = y + C(i) .* subs(fi{i}, 'x', x);
end

end

```

Для вычисления функций $\varphi_i(x)$ напомним функцию `GetFi.m`.

```

% Функция, реализующая построение базисных функций
% Входные данные:
% n - количество функций, K - ядро ИУ,
% f1 - фиксированная функция phi_1,
% a, b - отрезок интегрирования
% Результат: fi - массив функций
% Автор: Сергеев П.
function [fi] = GetFi2(K, f1, a, b, n)

    syms x s;
    fi = zeros(1, n);
    fi{1} = sym(f1);
    for j = 1 : n - 1
        fi{j + 1} = int(K(x, s) * fi{j}, s, a, b);
    end

end

```

2. Пример. Начнем с решения того же уравнения (6), с. 50, что и при изучении метода Галеркина — Петрова. Напомним, что уравнение имеет вид

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

Его точное решение $y(x) = 1 + 6x^2$ (см. упражнение 3.25, с. 207, [3], см. также пример 3.29, с. 212, [3]). Зададим $\varphi_1(x) = 1$. Количество базисных функций выберем равное трем. Приведем сценарий решения этой задачи на языке Matlab.

```
clear all
close all
clc
a = -1;
b = 1;
lambda = 1;
K = @(x,s) x.*s + x.*x; %ядро ИУ
f = @(x) 0*x + 1; %правая часть ИУ
y_ex = @(x) 1 + 6*x.*x; %точное решение
n = 3; %кол-во базисных функций
f1 = @(x) 1 + 0*x; %функция phi_1
[fi] = GetFi(K, f1, a, b, n);
[ C, y ] = BubnovGalerkin_Fred_II(K,
f, a, b, lambda, fi, n);
x = linspace(a, b); %сетка из 100 узлов на отрезке [a, b]
plot(x, y_ex(x), x, y, 'or');
er = double(norm(y_ex(x)-y,inf) / norm(y_ex(x),inf))
%относительная погрешность
legend('точное', 'приближенное');
title('График приближенного и точного решения');
```

В результате работы этого сценария строятся точно такие же графики, как и на рисунке 3, с. 52. Причем, приближенное решение методом моментов получается с относительной ошибкой $er = 1.356 \cdot 10^{-15}$.

3. Пример. Решим теперь уравнение

$$y(x) = \left(1 - \frac{3\pi}{2}\right) \cos(x) + \int_0^{3\pi} \cos(x+s)y(s)ds, \quad x \in [0, 3\pi]. \quad (4)$$

Легко проверить, что точное решение этого уравнения $y(x) = \cos(x)$. В качестве первой координатной функции выберем

$$\varphi_1 = \left(1 - \frac{3\pi}{2}\right) \cos(x).$$

Количество базисных функций оставим равным трем. Сценарий решения этого уравнения выглядит следующим образом.

```
clear all
close all
clc
a = 0;
b = 3*pi;
lambda = 1;
K = @(x,s) cos(x + s); % ядро ИУ
f = @(x) cos(x).*(1 - 3*pi/2); % правая часть ИУ
y_ex = @(x) cos(x); % точное решение ИУ
n = 3; %кол-во базисных функций
f1 = @(x) cos(x).*(1 - 3*pi/2); % функция phi_1
[fi] = GetFi(K, f1, a, b, n);
[ C, y ] = BubnovGalerkin_Fred_II(K, f, a, b, lb, fi, n);
x = linspace(a, b); %сетка из 100 узлов на отрезке [a, b]
plot(x, y_ex(x), x, y, 'or');
er = double(norm(y_ex(x)-y,inf) / norm(y_ex(x),inf))
%относительная погрешность
legend('точное', 'приближенное');
title('График приближенного и точного решения');
```

Графики точного и приближенного решения приведены на рисунке 4. Приближенное решение получено с погрешностью $er = 6.6929 \cdot 10^{-15}$.

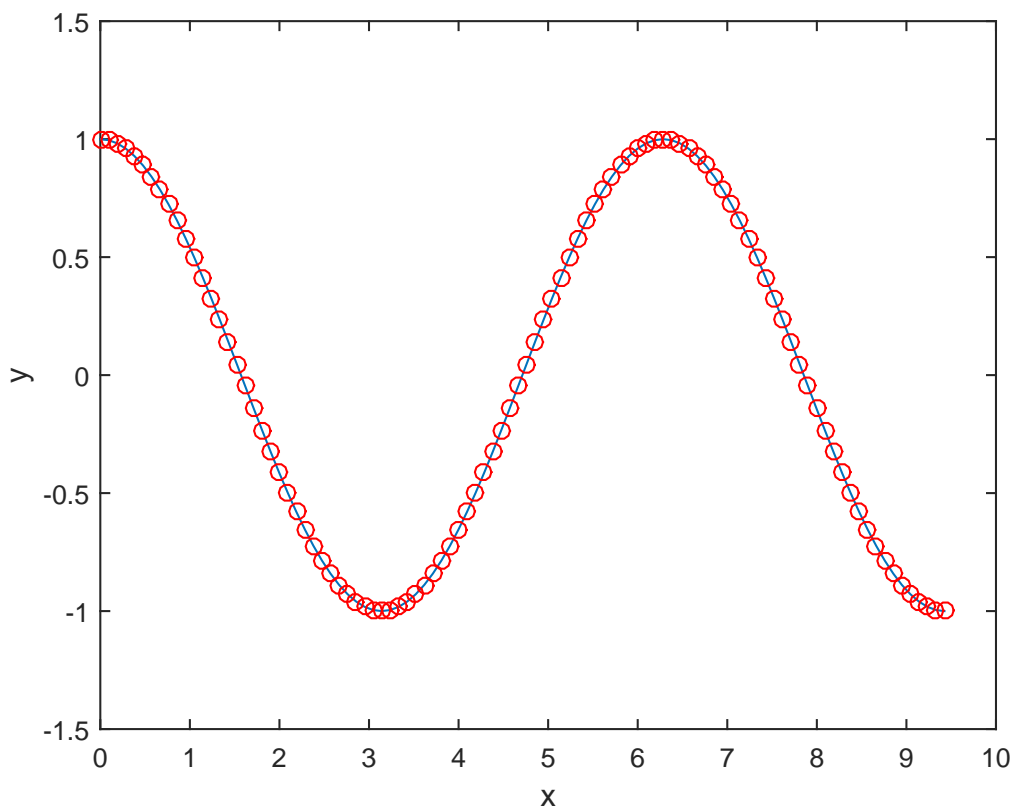


Рис. 4. Результаты решения уравнения (4) методом моментов. Непрерывной линией обозначено приближенное решение, кружочками — точное решение.

Упражнения

- 1) Методом моментов решите уравнение

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

Точное решение этого уравнения есть $y \equiv 1$ (см., пример 1, с. 344, [5]). В качестве первой базисной функции возьмите функцию, тождественно равную единице. Количество базисных функций равно трем. Определите относительную точность приближенного решения.

- 2) Решая уравнение (3), исследуйте зависимость времени работы программы и относительной точности решения от количества базисных функций. Объясните замеченные эффекты.

3) Методом моментов 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 $y(x) = x + 5x^2$. Для разных первых базисных функций $\varphi_1(x) = 1, x, x^2, x + 5x^2$ исследуйте зависимость времени работы программы и относительной точности решения от числа базисных функций. Объясните результат.

§ 6. 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 [17].

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., [14], 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]=Fredgolm_II_Colloc(a,b,lambda,K,f,n,m);
```

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

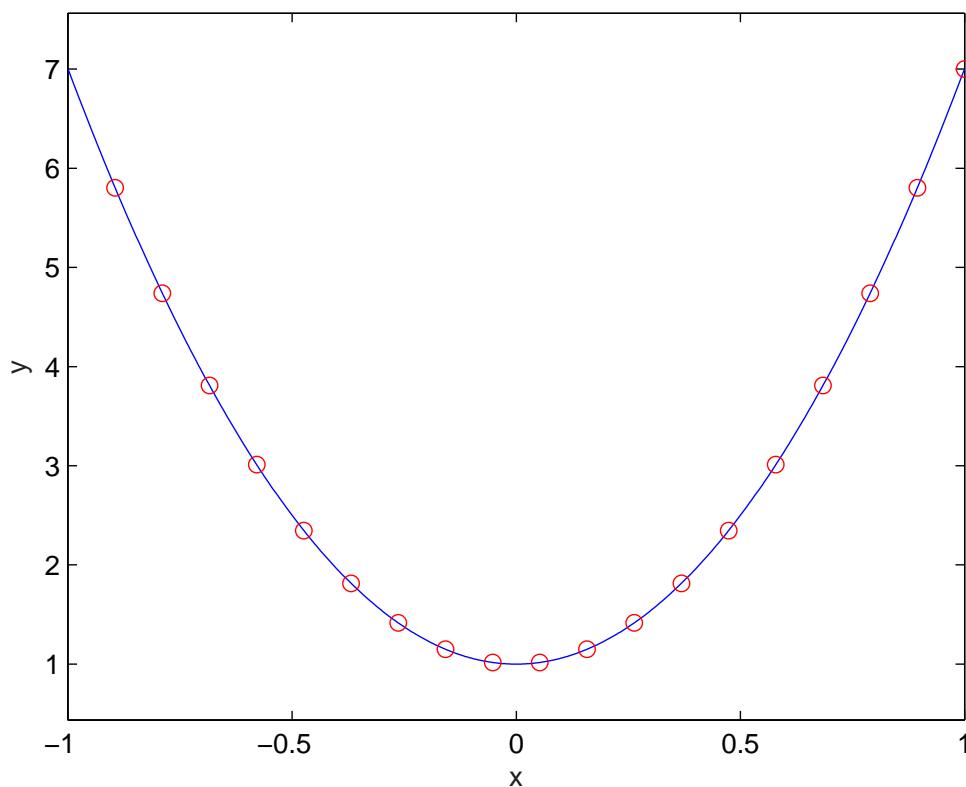


Рис. 5. Результаты решения уравнения (6), с. 50, методом коллокации. Непрерывной линией обозначено приближенное решение, кружочками — точное решение.

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

5. Пример. Решим методом коллокации уравнение (6), с. 50. 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:

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

Напомним, что точное решение уравнения (6), с. 50, 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. 5 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 $x_1 = 0$, $x_2 = 1$.

§ 7. Метод простой итерации

1. Описание метода. Запишем неоднородное интегральное уравнение Фредгольма второго рода в следующем виде:

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

Метод простой итерации реализуется с помощью формулы

$$y(x)_{k+1} = f(x) + \lambda \int_a^b K(x, s)y_k(s)ds, \quad x \in [a, b], \quad k = 0, 1, 2, \dots, \quad (2)$$

служащей для построения последовательности функций $y_k(x)$, являющихся приближениями к искомому решению уравнения. При этом начальное приближение $y_0(x)$ может быть выбрано произвольно, если нет каких-либо сведений о характере искомой функции. Часто его просто полагают равным нулю. Исходя из физической постановки задачи, такие априорные данные иногда можно определить. Это позволяет ускорить итерационный процесс удачным выбором начального приближения.

В качестве приближенного решения принимается $y_k(x)$, полученное при достаточно большом k . Признаком близости получаемых приближений к искомой функции является достижение малой величины разности двух следующих друг за другом приближений.

Метод простой итерации для уравнений Фредгольма второго рода (в отличие от уравнений Вольтерра) сходится не всегда. Удобным для проверки достаточным условием сходимости метода является (см. например, [3], с. 185, [7], с. 160) выполнение неравенства

$$|\lambda|B < 1, \quad (3)$$

где λ — числовой параметр исходного уравнения,

$$B = \left(\int_a^b \int_a^b |K(x, s)|^2 ds dx \right)^{1/2}. \quad (4)$$

Причем, чем ближе к нулю произведение $|\lambda|B$, тем более высокую скорость сходимости метода в метрике $L_2(a, b)$ показывают теоретические оценки (см. например, [3], с. 185, [7], с. 160).

2. Компьютерная программа и пример. Решим следующее уравнение (см. [7], с. 161, упражнение 332):

$$y(x) = x + 4 \int_0^1 x^2 s^2 y(s) ds, \quad x \in [0, 1]. \quad (5)$$

Точное решение этого уравнения известно (см. [7], с. 186), а именно, $y(x) = x + 5x^2$. Примем начальное приближение $y_0(x) = x$.

Реализуем алгоритм решения этого уравнения на языке Matlab. Будем вычислять приближения к искомому решению уравнения по формуле (2) с помощью функции `Fredgolm_II.m` следующего вида.

```
% Функция для решения уравнения Фредгольма второго рода
% методом простой итерации.
% Входные данные:
% K - ядро интегрального уравнения,
% f - правая часть (задаются аналитически),
% y0 - начальное приближение,
% a - начало отрезка интегрирования,
% b - конец отрезка,
% Результат:
% yk - вектор приближенных выражений собственных функций.
% Автор: Годовых К.
function [yk, k, B] = Fredgolm_II(K, f, lambda, y0, a, b, n)
% Для определения сходимости последовательных приближений
B = sqrt(quad2d(@(x,s) K(x,s).* K(x,s), a, b, a, b));
syms x s
if (abs(lambda)*B > 1)
    disp('Внимание! Достаточное условие не выполняется.');
```

```
end
k = 0;
while(k < n)
    y0 = subs(y0, 'x', s);
    yk = f(x) + lambda*int(K(x,s)*y0, s, a, b);
    k = k + 1;
    y0 = yk;
end
end
```

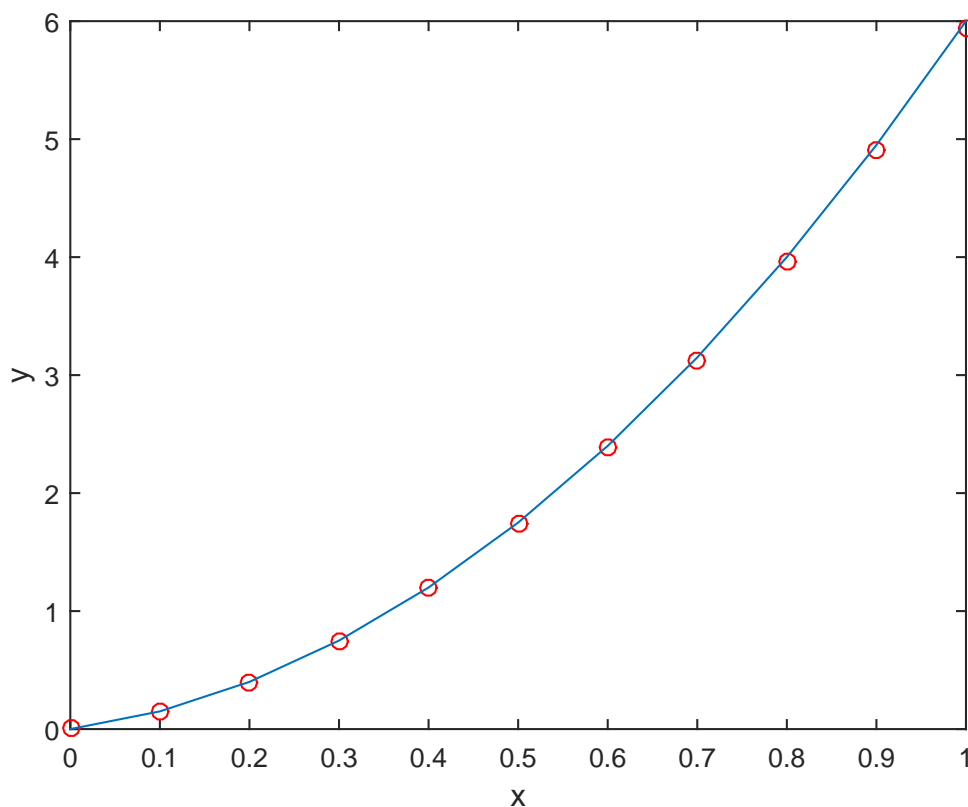


Рис. 6. Результаты решения уравнения (5), с. 68. Непрерывной линией обозначено точное решение, кружочками — приближенное решение на двадцатой итерации.

Напишем сценарий для решения уравнения с использованием приведенной выше функции.

```
clc
clear all
close all
syms x
K = @(x,s) x.^2.*s.^2; % ядро уравнения
a = 0; b = 1;
y0 = x; % начальное приближение
f = @(x) x; % правая часть
lambda = 4; % числовой параметр
y = @(x) 5.*x.^2 + x; % Точное решение
h = 0.1; % шаг
xx = a:h:b;
yt = y(xx);
```

Таблица 2. Результаты решения уравнения (5), с. 68.

n	относительная ошибка
4	3.41333e-01
6	2.18453e-01
8	1.39810e-01
10	8.94785e-02
12	5.72662e-02
14	3.66504e-02
16	2.34562e-02
18	1.50120e-02
20	9.60768e-03

```

N = numel(xx);
er = [];%погрешность
j = 1;
fid = fopen('f.txt', 'wt'); % открыли файл для записи
fprintf(fid, '%2s| %10s| \n', 'n', 'погрешность');
for n = 4:2:20 %количество итераций
    [yk, k] = Fredgalm_II(K, f, lambda, y0, a, b, n);
    for i = 1:N
        yy(i)=subs(yk, 'x', xx(i));
    end
    er(j) = double(norm(yt - yy, inf)/norm(yt, inf));
    % погрешность решения
    nn(j) = n;
    fprintf(fid, '%2.0f| %6.5e \n', [nn(j) er(j)]);
    j = j + 1;
end
plot(xx, yy, 'or')
hold on
plot(xx, yt)
fclose(fid);

```

Результаты счета представлены в таблице 2 и на рис. 6. При этом, в ходе вычислений получено $B = 0.2$, а параметр $\lambda = 4$. Следовательно, условие (3) выполняется.

Упражнения

- 1) Постройте график зависимости относительной ошибки решения уравнения (5), с. 68, от числа итераций n . Исследуйте, как зависит от начального приближения количество итераций n , которое необходимо сделать для достижения заданной точности вычислений ε . Установите, сколько для этого уравнения необходимо совершить итераций при $\varepsilon = 10^{-6}$.
- 2) С помощью функции `Fredgolm_II.m` найдите приближенное решение уравнения

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

Его точное решение $y(x) = 1 + (4/9)x$ (см. пример 3.17, с. 185, [3]). В качестве начального приближения задайте $y_0(x) = 1$. Сравните скорость сходимости решения этого уравнения и уравнения (5), с. 68. Чем объясняется разница?

- 3) С помощью функции `Fredgolm_II.m` найдите приближенное решение уравнения

$$y(x) = \frac{5}{6}x + \frac{1}{2} \int_0^1 xsy(s)ds, \quad x \in [0, 1]. \quad (7)$$

Его точное решение $y(x) = x$ (см. упражнение 333, с. 161, [7]). В качестве начального приближения задайте $y_0(x) = 0$. Сравните скорость сходимости решения этого уравнения, уравнения из предыдущего упражнения и исходного уравнения (5), с. 68. Чем объясняется разница?

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

Если ядро $K(x, y)$ непрерывно по совокупности аргументов, то интегральный оператор, определяемый правой частью уравнения (1) является компактным (см., напр., [16], с. 23) на пространстве непрерывных на $[a, b]$ функций, и для теоретического анализа спектральной задачи (1) можно применить результаты теории компактных операторов (см., напр., [16], с. 34). Если ядро симметрично ($K(x, s) = K(s, x)$), то согласно спектральной теории самосопряженных компактных операторов (см., напр., [16], с. 271) существует счетное множество вещественных характеристических чисел.

2. Решение задачи с вырожденным ядром. Если ядро уравнения вырождено (см. с. 33), то спектральная задача (1) эквивалентна конечномерной алгебраической спектральной задаче. Действительно, запишем уравнение с вырожденным ядром следующим в виде (сравните с (4), с. 33):

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

Пусть λ — 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 (5), с. 33.

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 (10), с. 34. Для того чтобы сформулировать эту задачу надо положить правую часть в (13), с. 34, равной нулю.

Таким образом, для решения однородного уравнения (2) с вырожденным ядром достаточно вычислить интегралы (10), с. 34, 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  $i \sim j$ 
             $D(i,j)=-1*\lambda*A(i,j)$ ;
        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  $\lambda$ 
 $d = \det(D)$ ;  $\lambda_{ak}=\text{solve}(d,\lambda)$ ;
%подставляем найденные значения в линейную систему
%для определения C
for i=1:m
    %матрица однородной системы при конкретном значении  $\lambda$ 
     $B1=\text{subs}(D, 'lambda', \lambda_{ak}(i))$ ;
    syms x;
    %фундаментальная система решений однородной системы
     $x=\text{sym}(\text{null}(\text{double}(B1), 'r'))$ ;
     $anss=C(i)*x$ ;
    %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  $O(i,j) \sim 0$ 
             $y(i)=C(i)*\alpha(i)$ ;
        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

$$y(x) - \lambda \int_0^1 K(x, s)y(s)ds = 0, \quad x \in [0, 1],$$

$$K(x, s) = \begin{cases} x(1-s), & 0 \leq x \leq s \leq 1, \\ s(1-x), & 0 \leq s \leq x \leq 1. \end{cases}$$

Точные значения характеристических чисел этого уравнения (см. пример 3.27, с. 210, [3])

$$\lambda_1 = \pi^2, \quad \lambda_2 = 4\pi^2,$$

и соответствующие им собственные функции

$$y_1(x) = \sqrt{2} \sin \pi x, \quad y_2(x) = \sqrt{2} \sin^2 \pi x.$$

2) Решите с помощью предложенного сценария уравнение с ядром

$$K(x, s) = x^2 s^2, \quad a = 0, \quad b = 1.$$

Наименьшее характеристическое число известно, $\lambda_1 = 108/21$.
(см. пример 3.33, с. 220, [3]).

3) Решите с помощью предложенного сценария уравнение с ядром

$$K(x, s) = xs, \quad a = 0, \quad b = 1.$$

Наименьшее характеристическое число известно, $\lambda_1 = 3$ (см.
пример 3.34, с. 221, [3]).

§ 2. Метод Ритца

Пусть ядро однородного интегрального уравнения

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

симметрично, $K(x, y) = K(y, x)$, $x, y \in [a, b]$. Напомним, что характеристическим числом ядра интегрального уравнения называется такое значение численного параметра λ , при котором уравнение (1) имеет нетривиальное решение. Заметим, что характеристические числа симметричного ядра — вещественны. В свою очередь числа, обратные к характеристическим числам ядра, называются собственными числами этого ядра.

Опишем метод Ритца поиска характеристических чисел симметричного ядра и отвечающих им собственных функций, следуя [3], с. 219 (см. также [7], с. 165). Выбирается последовательность функций $\{\psi_n(x)\}$, в которой каждая функция ψ_k , $k = 1, 2, \dots, n$, интегрируема с квадратом на $[a, b]$. Система $\{\psi_n(x)\}$ принимается полной в классе указанных функций, а все входящие в нее функции являются линейно независимыми на $[a, b]$.

Приближенные собственные функции будем искать в виде

$$y_n(x) = \sum_{k=1}^n a_k \psi_k(x). \quad (2)$$

Разделим обе части уравнения (1) на λ , получим

$$\sigma y(x) = \int_a^b K(x, s)y(s)ds, \quad x \in [a, b], \quad (3)$$

где $\sigma = 1/\lambda$ — собственное число. Подставим вместо $y(x)$ в равенство (3) выражение (2), получим

$$\sigma \sum_{k=1}^n a_k \psi_k(x) = \sum_{k=1}^n a_k \int_a^b K(x, s)\psi_k(s)ds, \quad x \in [a, b]. \quad (4)$$

Обе части уравнения (4) последовательно умножим скалярно на функции $\psi_j(x)$ для $j = 1, 2, \dots, n$. Получим n равенств:

$$\sum_{k=1}^n a_k \int_a^b \left(\int_a^b K(x, s) \psi_k(s) ds \right) \psi_j(x) dx - \sigma \sum_{k=1}^n a_k \int_a^b \psi_k(x) \psi_j(x) dx = 0, \quad j = 1, 2, \dots, n. \quad (5)$$

Равенства (5) представляют собой однородную систему линейных алгебраических уравнений относительно неизвестных $a_k, k = 1, 2, \dots, n$. Элементы матрицы $M(\sigma)$ этой системы вычисляются по формуле

$$m_{j,k}(\sigma) = \int_a^b \left(\int_a^b K(x, s) \psi_k(s) ds \right) \psi_j(x) dx - \sigma \int_a^b \psi_k(x) \psi_j(x) dx, \quad j, k = 1, 2, \dots, n. \quad (6)$$

Неоднородное решение системы (5) существует, если определитель ее матрицы $M(\sigma)$ равен нулю:

$$\det M(\sigma) = 0. \quad (7)$$

Корни уравнения (7) являются приближенными значениями собственных чисел ядра $K(x, s)$.

Найденные из (7) значения σ_l , будучи подставленными в систему (5), позволяют найти ее ненулевое решение $a_k, k = 1, 2, \dots, n$. Последующая подстановка полученных значений $a_k, k = 1, 2, \dots, n$, в (2) позволяет найти приближенные выражения собственных функций, соответствующих уже найденным приближенным собственным значениям.

1. Компьютерная программа. Напишем на языке Matlab функцию `Fredholm_II_Ritz.m`, реализующую метод Рунге.

```
% Функция для нахождения характеристических чисел
% однородного уравнения Фредгольма второго рода
```

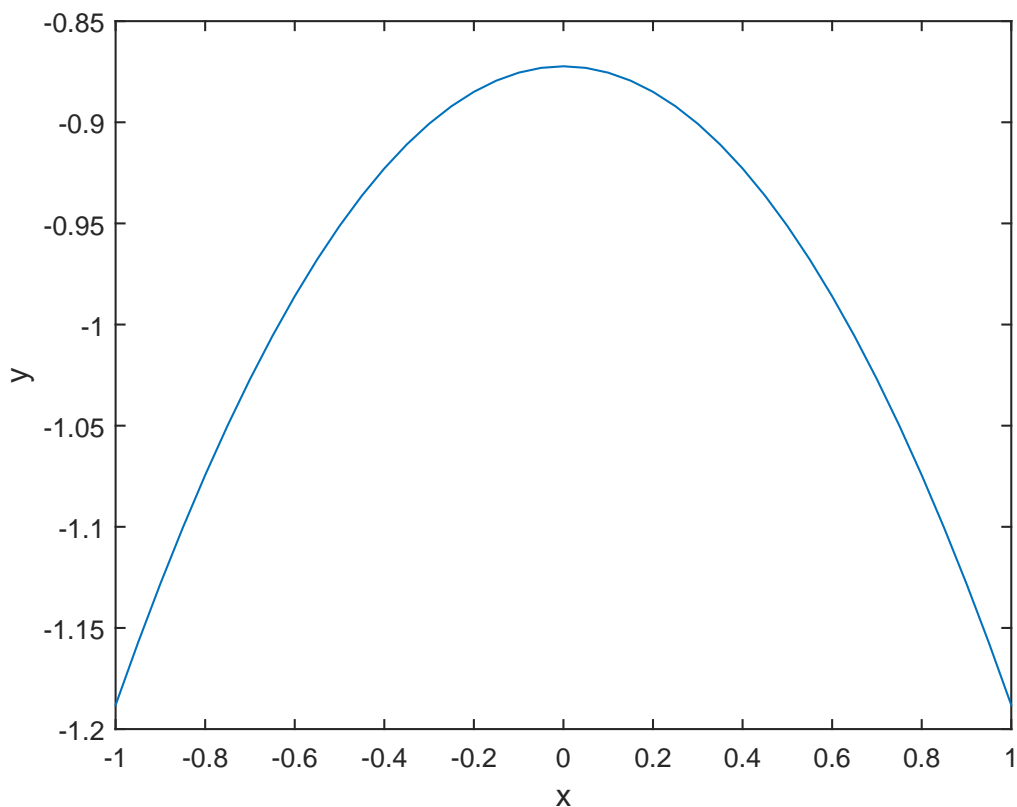



Рис. 1. Собственная функция интегрального уравнения Фредгольма второго рода с ядром (8), с. 84, отвечающая характеристическому значению $\lambda = 0.4462$ построенная методом Ритца.

```

% методом Ритца.
% Входные данные:
% K - ядро интегрального уравнения,
% a - начало отрезка интегрирования,
% b - конец отрезка,
% n - число базисных функций;
% Результат:
% lambda - вектор характеристических чисел ИУ,
% yk - вектор приближенных выражений собственных функций
% в символьном виде.

% Автор: Ганиева Э.

function [lambda, yk] = Fredgolm_II_Ritz(K, a, b, n)
syms Sigma x s

```

```
% Строим матрицу M по формуле (6).
for j= 1:n+1
    for k= 1:n+1
        M(j, k) = int(int(K(x,s).*Psi(x, j-1).* ...
            Psi(s, k-1), a, b), a, b)- ...
            Sigma*int( Psi(x, j-1).*Psi(x, k-1), a, b);
    end
end
end

% Составляем определитель матрицы M,
% разворачиваем его в уравнение d(Sigma)=0
% и находим приближенные значения собственных чисел ядра K.
Sg = sort(double(solve(det(M))), 'descend');

% Находим приближение характеристических чисел ИУ.
if (Sg(k) ~= 0)
    lambda = 1./Sg;
end

for k = 1:numel(Sg)
    if (Sg(k) ~= 0)

        % Находим приближение характеристических чисел ИУ.
        lambda(k) = 1/Sg(k);

        % Матрица однородной системы при конкретном
        % значении сигма.
        M2 = subs(M, 'Sigma', Sg(k));
        % Фундаментальная система решений однородной
        % системы.
        A = null(double(M2));

        % Приближенные выражения собственных функций,
        % соответствующих приближенным собственным
        % значениям.
        y = 0;
        for j = 1:n+1
```

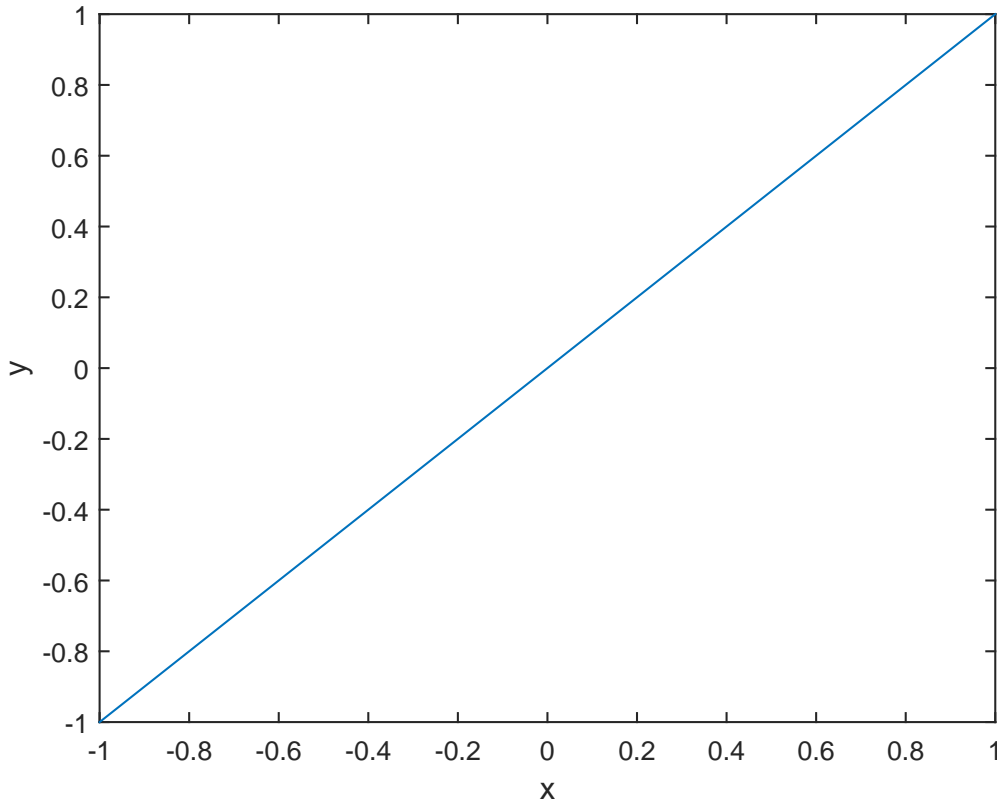


Рис. 2. Собственная функция интегрального уравнения Фредгольма второго рода с ядром (8), с. 84, отвечающая характеристическому значению $\lambda = 1.5$ построенная методом Рунге.

```

        y = y + A(j)*Psi(x, j-1);
    end
    yk(k) = y;
end
end
end

```

2. Координатные функции В качестве координатных функций будем использовать полиномы Лежандра. На отрезке $[-1, 1]$ вычислять их будем по формулам

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n, \quad n = 0, 1, 2, \dots$$

а на отрезке $[0, 1]$ по формулам

$$P_n(x) = \frac{1}{n!} \frac{d^n}{dx^n} (x^2 - x)^n, \quad n = 0, 1, 2, \dots$$

Создадим функцию `Psi.m` для построения полиномов Лежандра на отрезке $[-1, 1]$.

```
% Функция, реализующая построение полиномов
% Лежандра на отрезке [-1, 1]
% Входные данные: x - символьная переменная,
% n - степень полинома,
% Результат: P - координатная функция в символьном виде
% Автор: Ганиева Э.
```

```
function P = Psi(x, n)
if n == 0
    P = 1+0.*x;
else
    P = diff((x.^2-1).^n, n)/(2^n*factorial(n));
end
end
```

Теперь создадим соответствующую функцию `Psi.m` для построения полиномов Лежандра на отрезке $[0, 1]$.

```
% Функция, реализующая построение смещенных полиномов
% Лежандра на отрезке [0, 1]
% Входные данные: x - символьная переменная,
% n - степень полинома,
% Результат: P - координатная функция в символьном виде
% Автор: Ганиева Э.
```

```
function P = Psi(x, n)
if n == 0
    P = 1+0.*x;
else
    P = diff((x.^2-x).^n, n)/factorial(n);
end
end
```

3. Пример. Необходимо найти три первых характеристических числа и соответствующие им собственные функции однородного

интегрального уравнения Фредгольма второго рода с ядром

$$K(x, s) = 1 + xs + x^2s^2, \quad a = -1, \quad b = 1. \quad (8)$$

Решение этой задачи известно (см. упражнение 134, с. 65, [7]),

$$\lambda_1 = \frac{27 - 3\sqrt{61}}{8}, \quad y_1(x) = \left(x^2 + \frac{6 + \sqrt{61}}{5} \right) C,$$

$$\lambda_2 = \frac{3}{2}, \quad y_2(x) = Cx,$$

$$\lambda_3 = \frac{27 + 3\sqrt{61}}{8}, \quad y_3(x) = \left(x^2 + \frac{6 - \sqrt{61}}{5} \right) C.$$

Ясно, что собственные функции определяются с точностью до постоянного множителя C . Собственные числа упорядочены, как обычно, по возрастанию.

Сценарий решения этой задачи выглядит следующим образом.

```

clc
clear all
close all
K = @(x,s)1+x.*s+ x.^2.*s.^2;
a = -1;
b = 1;
n = 2;
[lambda, y] = Fredgolm_II_Ritz(K, a, b, n)
pretty(y)
% Построение графиков собственных функций.
h = 0.05;
X = a:h:b;
for i = 1:numel(y)
    for j = 1:numel(X)
        Y(j) = subs(y(i), 'x', X(j));
    end
    figure
    plot(X, Y)
    title(['\lambda = ', num2str(lambda(i))])
    xlabel('x');

```

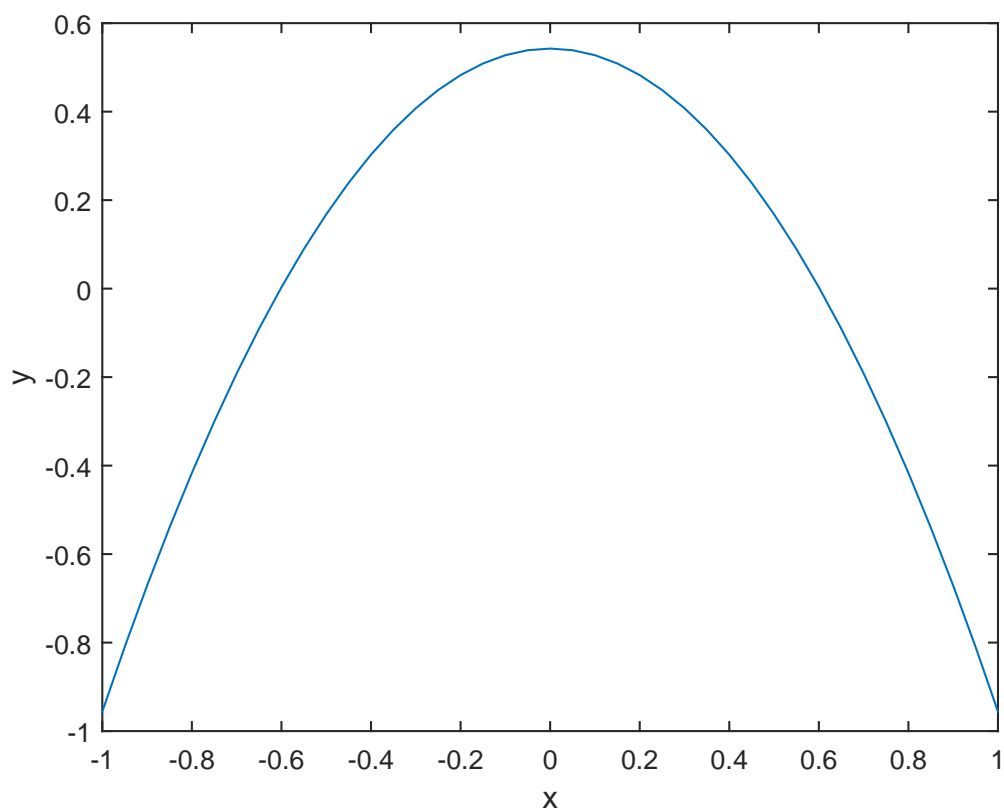


Рис. 3. Собственная функция интегрального уравнения Фредгольма второго рода с ядром (8), с. 84, отвечающая характеристическому значению $\lambda = 6.3038$ построенная методом Рунге.

```
ylabel('y');
end
```

В результате работы программы на экран выводятся характеристические значения и символьные выражения собственных функций, затем строятся их графики (см. рисунки 1–3).

Покажем, что полученные функции $y(x)$ являются собственными функциями, отвечающими характеристическим значениям λ . Для этого подставим в правую часть уравнения (1) полученные λ и $y(x)$ и сравним с левой частью.

```
syms x s
h = 0.05;
X = a:h:b;
for k = 1:numel(y)
    z = lambda(k) * int(subs(y(k), 'x', 's').*K(x, s), s, a, b);
```

```

for j = 1:numel(X)
    Y(j) = double(subs(y(k), 'x', X(j)));
    Z(j) = double(subs(z, 'x', X(j)));
end
er(k) = max(Y-Z);
end

```

Невязка оказалась на уровне машинной точности, что говорит о верности решения задачи.

Упражнения

- 1) Найдите приближенное значение наименьшего характеристического числа ядра

$$K(x, s) = x^2 s^2, \quad a = 0, \quad b = 1,$$

и отвечающую ему собственную функцию. Это число известно, а именно, $\lambda_1 = 108/21$. (см. пример 3.33, с. 220, [3]). Покажите, что полученное решение удовлетворяет исходному уравнению.

- 2) Найти три первых характеристических числа однородного интегрального уравнения

$$y(x) - \lambda \int_0^1 K(x, s)y(s)ds = 0, \quad x \in [0, 1],$$

$$K(x, s) = \begin{cases} x(1-s), & 0 \leq x \leq s \leq 1, \\ s(1-x), & 0 \leq s \leq x \leq 1, \end{cases}$$

и отвечающие им собственные функции. Точные значения характеристических чисел известны (см. пример 3.27, с. 210, [3])

$$\lambda_1 = \pi^2, \quad \lambda_2 = 2^2\pi^2, \quad \lambda_3 = 3^2\pi^2.$$

Решение ищите в виде

$$y(x) = a_1 + a_2x(1-x) + a_3x(1-x)(1-2x).$$

Покажите, что полученное решение удовлетворяет исходному уравнению.

- 3) Используя решения упражнений этого параграфа покажите, что различным характеристическим значениям отвечают ортогональные собственные функции.
- 4) Проиллюстрируйте утверждение, что если ядро $K(x, s)$ — кососимметричное, т.е. $K(x, s) = -K(s, x)$, то все характеристические числа чисто мнимые. Для этого найдите характеристические числа ядра

$$K(x, s) = s - x, \quad a = 0, \quad b = 1.$$

- 5) Используя функцию Psi.m для отрезка $[0, 1]$, найдите минимальное характеристическое число и отвечающую ему собственную функцию, ядра

$$K(x, s) = \sin(x) \sin(s), \quad a = 0, \quad b = 2\pi.$$

Решение этой задачи известно (см. пример 125, с. 60, [7])

$$\lambda = \frac{1}{\pi}, \quad y(x) = C \sin(x).$$

Для того, чтобы применить к решению этой задачи полиномы Лежандра, определенные на отрезке $[0, 1]$, сделайте линейную замену переменных.

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. Краснов М.Л., Кисилев А.И., Макаренко Г.И. Интегральные уравнения: задачи и примеры с подробными решениями: учебное пособие. Изд. 3-е, испр. — М.: Едиториал УРСС, 2003. — 192 с.
8. 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.
9. Крылов В.И. Приближенное вычисление интегралов. — М.: Изд-во «Наука», 1967. — 500 с.
10. Манжиров А.В., Полянин А.Д. Справочник по интегральным уравнениям: методы решения. — М.: Изд-во «Факториал Пресс», 2000. — 384 с.

11. Михлин С.Г. Лекции по линейным интегральным уравнениям. — М.: Государственное издательство физико-математической литературы, 1959. — 232 с.
12. Михлин С.Г., Смолицкий Х.Л. Приближенные методы решения дифференциальных и интегральных уравнений. — М.: Изд-во «Наука», 1965. — 384 с.
13. Михлин С.Г. Вариационные методы в математической физике. — М.: Изд-во «Наука», 1970. — 512 с.
14. Янке Е., Эмде Ф., Леш Ф. Специальные функции. — М.: Изд-во «Наука», 1968. — 344 с.
15. Brian R. Hunt [и др.] Matlab R2007 с нуля [пер. с англ.] — М.: Лучшие книги, 2008. — 352 с.
16. Kress R. Linear Integral Equations. — New York: Springer-Verlag, 1999. — 365 p.
17. Vainikko G. Multidimensional Weakly Singular Integral Equations. — Berlin, Heidelberg: Springer-Verlag, 1993. — 159 p.

Учебное издание

Карчевский Евгений Михайлович

**Интегральные уравнения:
применение в математическом моделировании,
численные методы решения и
комплекс программ на языке MatLab**

Учебное пособие

21.09.2022. 60 84 1/16.
. . . 5,75. 100. 2102/1.
« » (. .)
420111, . , . , 22. .: 292-98-92
e-mail: westfalika@inbox.ru
