|  |
| --- |
| Московский авиационный институт |
| (Национальный исследовательский университет) |
| Институт №8 «Компьютерные науки и прикладная математика» |
| Кафедра вычислительной математики и программирования |
| **Курсовая работа** |
| по курсу «Численные методы» |
| Вариант 15 |
| Выполнил: Ларшин Т.А. |
| Группа: М8О-402Б-20 |
| Проверил: Пивоваров Д.Е. |
| Дата: |
| Оценка: |
| Москва, 2023 |

**Численное решение интегральных уравнений Фредгольма**

**2-го рода.**

# Теоретические сведения

Интегральным уравнением называется всякое уравнение, содержащее неизвестную функцию под знаком интеграла.

Линейным интегральным уравнением Фредгольма 2–го рода называется уравнение вида

𝑏

𝑓(𝑥) − 𝜆 ∫ 𝐾(𝑥, 𝑡)𝑓(𝑡)𝑑𝑡

𝑎

= 𝑔(𝑥),

где 𝑓(𝑥) – искомая функция; 𝐾(𝑥, 𝑡) и 𝑔(𝑥) – известные функции, заданные на основном квадрате [𝑎, 𝑏] × [𝑎, 𝑏] и отрезке [𝑎, 𝑏] соответственно; 𝜆 – числовой параметр. Функция 𝐾(𝑥, 𝑡) называется ядром интегрального уравнения, а 𝑓(𝑥) – свободным членом этого уравнения. Если 𝑔(𝑥) ≠ 0, то уравнение называется неоднородным, если же 𝑔(𝑥) = 0, то данное уравнение называется однородным.

Пределы интегрирования a и b могут быть как конечными, так и бесконечными. Решением называется любая функция 𝑓(𝑥), при подстановке, которой в уравнения последние обращаются в тождества относительно 𝑥 ∈ [𝑎,𝑏]. Эти уравнения могут помочь в вычислении распределения физических величин в пространстве и времени. Таким образом, уравнения Фредгольма второго рода используются для моделирования различных физических и математических процессов, а также для численного решения подобных задач в различных областях науки и техники.

# Методы

Существует несколько методов решения таких уравнений. Вот несколько из них:

1. Метод простых итераций

Этот метод основан на представлении уравнения Фредгольма второго рода в интегральной форме и последующем его итеративном решении. Путем применения итераций уравнение может быть сведено к последовательности линейных уравнений, что позволяет получить приближенное решение.

1. Метод Монте-Карло

Метод Монте-Карло позволяет численно решать уравнения Фредгольма второго рода путем использования статистических методов..

1. Метод импульсного отклика (метод Вольтерра)

Этот метод основан на представлении уравнения Фредгольма второго рода через импульсную функцию, что позволяет рассмотреть его как уравнение нахождения отклика системы на внешнее воздействие.

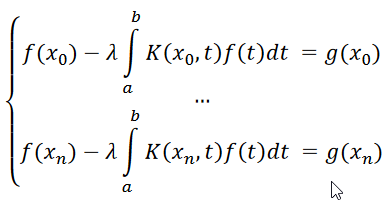
1. Метод квадратур

Одним из наиболее распространенных методов, применяемых для решения уравнения Фредгольма, является метод квадратур, состоящий в замене входящего в левую часть уравнения интеграла какой-либо формулой численного интегрирования.

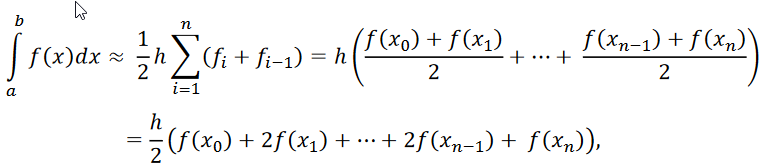
В данной задаче для простоты будет использован метод квадратур.

# Решение

Данный метод будем применять к уравнению 2 рода. Для начала построим на отрезке [a,b] сетку с узлами Для каждого узла сетки запишем уравнен, соответственно получим систему линейных уравнений:

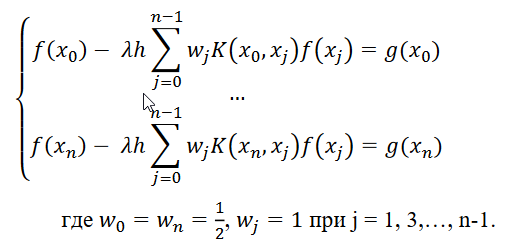


Аппроксимируем интегралы, используя формулу трапеций:

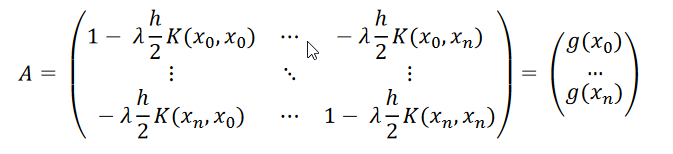


где h – постоянный шаг сетки.

После преобразования система будет иметь вид:

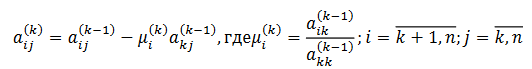


Запишем матрицу данной системы



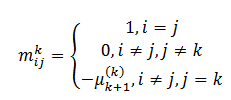
Решив систему методом Гаусса, получим функцию f(x), заданную таблично в узлах сетки.

Рассмотрим k-й шаг метода Гаусса, на котором осуществляется обнуление элементов под диагональю k-го столбца матрицы 𝐴(𝑘−1). С этой целью используется операция:



В терминах матричных операций такая операция эквивалентна умножению 𝐴(𝑘) =

=𝑀𝑘𝐴(𝑘−1), где элементы матрицы 𝑀 определяются следующим образом



В результате прямого хода метода Гаусса получим



Где - верхняя треугольная матрица, а L

В дальнейшем 𝐿𝑈-разложение будет эффективно использовано при решении систем линейных алгебраических уравнений вида 𝐴𝑥 = 𝑏 .

**Реализация программы на Python**

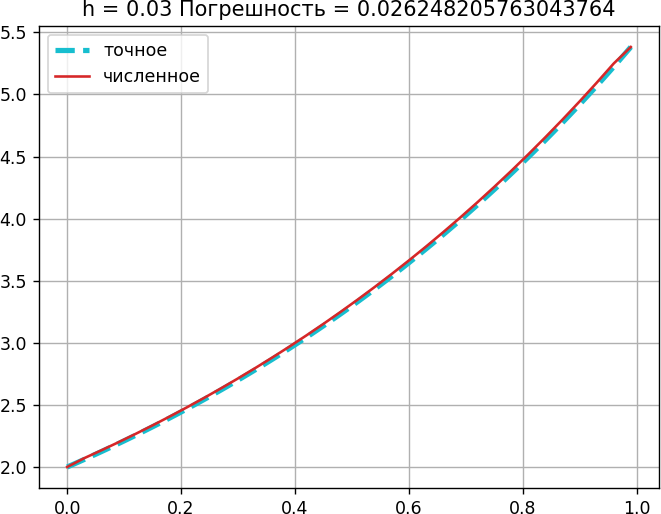
Возьмём простейший пример. Дано уравнение с:

границами [0,1]

параметром

ядром 𝐾(𝑥, 𝑠) = 𝑒𝑥−𝑡 и правой частью 𝑔(𝑥) = 𝑒𝑥.

Точное решение этого уравнения: 𝑦(𝑥) = 2𝑒𝑥.

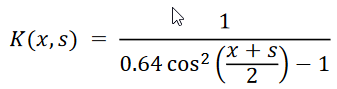


# Пример 2

Дано уравнение с

границами отрезка интегрирования [−𝜋, 𝜋],

параметром

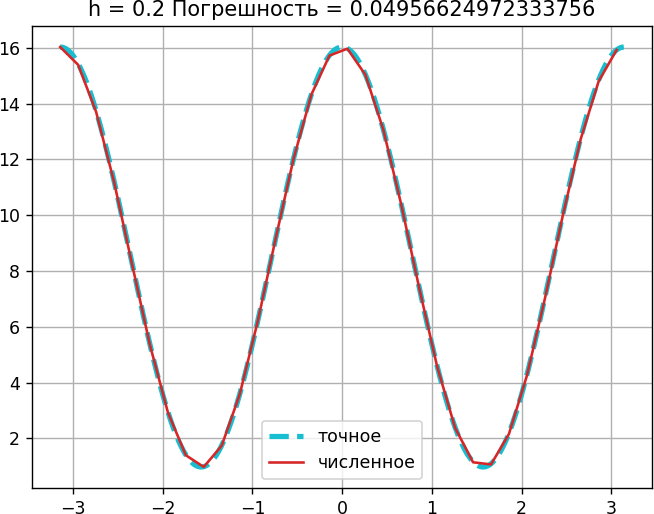
ядром 

Точное решение этого уравнения

𝑦(𝑥) =

17 128

2 + 17 𝑐𝑜𝑠(2𝑥)



# Код программы

import numpy as np  
import matplotlib.pyplot as plt  
  
  
def Fredgolm\_II(K, g, x, h, lamb):  
 n = x.size  
 A = np.zeros((n, n))  
 for i in range(n):  
 if i == 0:  
 A[i, 0] = 1 - 0.5 \* lamb \* h \* K(x[i], x[0])  
 else:  
 A[i, 0] = - 0.5 \* lamb \* h \* K(x[i], x[0])  
 for i in range(n):  
 for j in range(n):  
 if j == i:  
 A[i, j] = 1 - lamb \* h \* K(x[i], x[j])  
 else:  
 A[i, j] = - lamb \* h \* K(x[i], x[j])  
  
 for i in range(n):  
 if i == n - 1:  
 A[i, n - 1] = 1 - lamb \* 0.5 \* h \* K(x[i], x[n - 1])  
 else:  
 A[i, n - 1] = - lamb \* 0.5 \* h \* K(x[i], x[n - 1])  
 return solve(A, g(x))  
  
def error(x, y, f):  
 max = 0  
 for i in range(x.size):  
 e = np.abs(f(x[i]) - y[i])  
 if (e > max):  
 max = e  
 return max  
  
def solve(A, b):  
 n = len(A)  
 for k in range(n - 1):  
 for i in range(k + 1, n):  
 factor = A[i, k] / A[k, k]  
 for j in range(k + 1, n):  
 A[i, j] -= factor \* A[k, j]  
 b[i] -= factor \* b[k]  
 x = [0] \* n  
 x[n - 1] = b[n - 1] / A[n - 1, n - 1]  
 for i in range(n - 2, -1, -1):  
 sum\_ax = sum(A[i, j] \* x[j] for j in range(i + 1, n))  
 x[i] = (b[i] - sum\_ax) / A[i, i]  
 return x  
  
def K(x, t):  
 #return 1 / (0.64 \* numpy.cos((x + t) / 2) \*\* 2 - 1)  
 return np.exp(x-t)  
def g(x):  
 # return 25 - 16 \* numpy.sin(x) \*\* 2 return numpy.exp(x)  
 return np.exp(x)  
def f(x):  
 # return 17 / 2 + (128 / 17) \* numpy.cos(2 \* x) return 2\*numpy.exp(x)  
 return 2\*np.exp(x)  
  
a = 0  
b = 1  
lamb = 1 / 2  
#a = -numpy.pi # b = numpy.pi  
# lamb = 3 / (10 \* numpy.pi)  
  
h\_exact = 0.01  
x\_exact = np.arange(a, b, h\_exact)  
y\_exact = f(x\_exact)  
plt.plot(x\_exact, y\_exact, 'C9', label='точное', linewidth=3, linestyle='--')  
  
h\_approx = 0.03  
x\_approx = np.arange(a, b, h\_approx)  
y\_approx = Fredgolm\_II(K, g, x\_approx, h\_approx, lamb)  
y\_approx[0] = f(x\_approx[0])  
y\_approx[len(y\_approx) - 1] = f(x\_approx[x\_approx.size - 1])  
plt.plot(x\_approx, y\_approx, 'C3', label='численное')  
plt.title('h = ' + str(h\_approx) + ' Погрешность = ' + str(error(x\_approx, y\_approx, f)))  
plt.grid(True)  
plt.legend()  
plt.show()

# Заключение

Как можно увидеть на графиках метод даёт приемлемую точность на достаточно малых отрезках при достаточно малом шаге.

# Список литературы

1. Сумин Е. В., Шерстюков В. Б., Шерстюкова О. В. Интегральные уравнения Фредгольма и Вольтерра, краевые задачи и методы их решения: Учебно- методическое пособие. б.м. : НИЯУ МИФИ, 2016.
2. Калашников А. Л. Методы приближённого решения интегральных уравнений второго рода. Нижний Новгород : Нижегородский университет, 2017.
3. Карачевский Е. М. Численные методы решения уравнений и комплекс программ на языке Matlab. Казань : Казанский университет, 2019.