БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Лабораторная работа №2

Вариант 4

**Краевая задача**

**Выполнила:**

Старостина Ангелина

3 курс 7 группа

**Преподаватель:**

Будник Анатолий Михайлович

Минск, 2023

**Содержание**

Постановка задачи…………………………….………………………………………....3

Алгоритм решения……………………….………………………..……………………..4

**Постановка задачи**

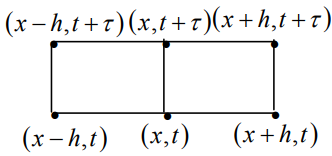
Для решения краевой задачи вида

на сетке построить разностную схему с весами при с погрешностью аппроксимации не ниже .

Необходимо:

1. Показать, что построенная разностная схема имеет заданный порядок аппроксимации.
2. Исследовать устойчивость полученной разностной схемы по начальным данным, используя принцип максимума.
3. Реализовать данную разностную схему при h = 0.05 и , выбранным из условия устойчивости.
4. Оценить приближенное решение, анализируя погрешность аппроксимации при разных шагах.

**Алгоритм решения**



Аппроксимируем на шаблоне

Разностная схема с весами:

Для получения погрешности аппроксимации с параметром возьмем . Тогда уравнение примет вид:

Аппроксимация условий:

Коэффициент выбирается из погрешности аппроксимации:

Таким образом схема имеет вид:

1. **Показать, что построенная разностная схема имеет заданный порядок аппроксимации**

Погрешность аппроксимации уравнения:

Первое и второе условия аппроксимируются точно.

Аппроксимация третьего условия:

1. **Устойчивость**

Схема в индексной форме:

Перепишем схему в виде:

Таким образом

Тогда для устойчивости необходимо:

1. **Реализация**

Для устойчивости метода возьмем

Приведем уравнение к виду удобному для определения коэффициентов метода разностной прогонки:

Следовательно:

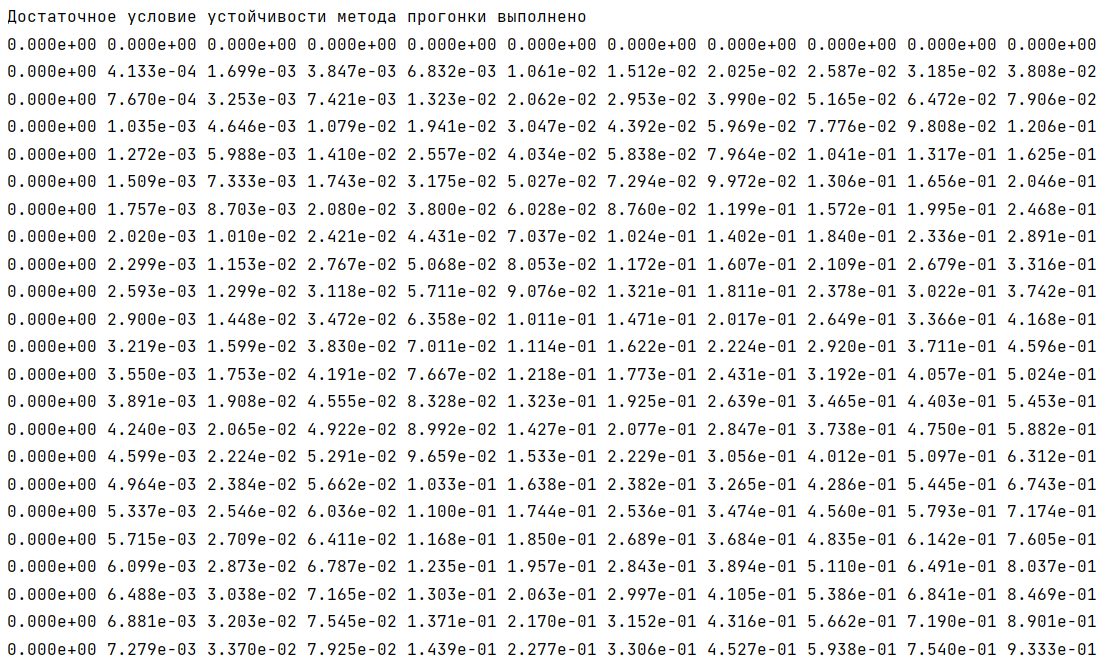
Из граничных условий получаем коэффициенты:

Решение системы найдем методом прогонки. Вычисления проводим по временным слоям.

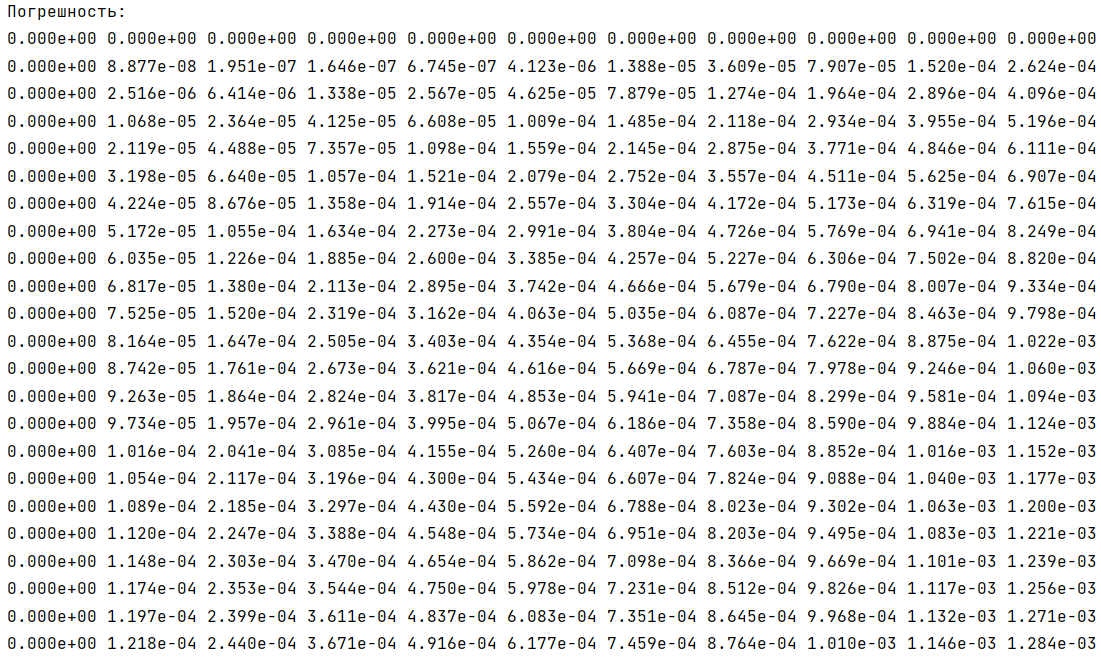
**Листинг программы**

def mu\_2(t, h, tau):  
 return (2-h) \* t + h / 2 + tau - 2 \* h \*\* 2 / 5  
  
def phi(x, t, tau):  
 return x\*\*2 - 2\*t - tau  
  
def solve(h, tau):  
 stab = []  
 N1 = int(1 / h)  
 N2 = int(1 / tau)  
 sigma = 1 / 2 - h \*\* 2 / (5 \* tau)  
 A\_value = -tau \* sigma / (h \*\* 2)  
 C\_value = 1 + tau \* 2 \* sigma / (h \*\* 2)  
 B\_value = -tau \* sigma / (h \*\* 2)  
 if abs(C\_value) >= abs(B\_value) + abs(A\_value):  
 stab.append(True)  
 if abs(1 + h \*\* 2 / (2 \* tau)) > 1:  
 stab.append(True)  
 for i in stab:  
 if i == False:  
 print('Достаточное условие устойчивости метода прогонки выполнено')  
 return -1  
 print('Достаточное условие устойчивости метода прогонки выполнено')  
 y = []  
 y.append([0 for j in range(N1 + 1)])  
 for i in range(N2):  
 A = [A\_value for j in range(N1 + 1)]  
 B = [B\_value for j in range(N1 + 1)]  
 C = [C\_value for j in range(N1 + 1)]  
 F = [0 for j in range(N1 + 1)]  
  
 C[0] = 1  
 A[0] = 0  
 B[0] = 0  
 for j in range(1, N1):  
 F[j] = y[i][j] + ((1 - sigma) \* tau / h \*\* 2) \* (y[i][j - 1] - 2 \* y[i][j] + y[i][j + 1]) + tau \* phi(j \* h, i \* tau, tau)  
 A[N1] = 1  
 C[N1] = -1 - h\*\*2 / (2 \* tau \* sigma)  
 F[N1] = -(h \* mu\_2(tau \* (i-1), h, tau) / sigma - (1-sigma)/sigma\*(y[i-1][N1]-y[i-1][N1-1])+h\*\*2/(2\*tau\*sigma)\*y[i-1][N1])  
 y.append(sweep\_method(A, C, B, F))  
 return y  
  
def sweep\_method(A, C, B, F):  
 n = len(A)  
 x = [0] \* n  
 for i in range(1, n):  
 coef = A[i] / C[i - 1]  
 C[i] = C[i] - coef \* B[i - 1]  
 F[i] = F[i] - coef \* F[i - 1]  
 x[n-1] = F[n - 1] / C[n - 1]  
 for i in range(n - 2, -1, -1):  
 x[i] = (F[i] - B[i] \* x[i + 1]) / C[i]  
 return x  
  
y1 = solve(0.05, 0.0015)  
N1 = int(1 / 0.05) + 1  
N2 = int(1 /0.0015) + 1  
for i in reversed(range(0, N2, 19)):  
 for j in range(0, N1, 2):  
 print('{:.3e}'.format(y1[i][j]), end=" ")  
 print()  
print()  
y2 = solve(0.0005, 0.00015)  
print("\nПогрешность:")  
for i in reversed(range(0, N2, 19)):  
 for j in range(0, N1, 2):  
 print('{:.3e}'.format(abs(y1[i][j] - y2[i \* 10][j\*100])), end=" ")  
 print()

**Вывод программы**

Значения при

Разница с



Была построена схема порядка , по результатам работы программы видно, что такая точность достигается. Невязка элементов и равна 0, поскольку условия аппроксимировались точно.