**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Отчет по лабораторной работе №1

Численное решение стационарного уравнения теплопроводности

Преподаватель: Будник Анатолий Михайлович

Студент: Высоцкий Богдан Георгиевич

3 курс 5 группа

**Постановка задачи**

Решить третью краевую задачу для ОДУ второго порядка следующего вида:

где

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  | 2 | 2 | 0 |  |

**Теория**

Численно решить данную задачу можно заменив дифференциальные операторы разностными. Наша задача – аппроксимировать исходную задачу разностной схемой 2-го порядка на минимальном шаблоне.

При аппроксимации первого уравнения, мы получили погрешность порядка , однако нам удалось повысить порядок аппроксимации до , использовав для замены первой производной в уравнении центральную разностную производную.

Аппроксимируя второе уравнение, мы получили первого порядка точности, но мы добились повышения порядка точности до второго, уничтожая главный член погрешности аппроксимации этого уравнения путем вариации параметров и .

Аналогичным образом как и со вторым уравнением, мы будем варьировать параметры и таким образом, чтобы уничтожить главный член погрешности аппроксимации этого уравнения и получить второй порядок точности. Рассмотрим погрешность аппроксимации третьего уравнения:

Чтобы получить , занулим обе скобки и получим искомые значения параметров и :

Выбирая параметры и таким образом, мы получим второй порядок аппроксимации третьего уравнения и, соответственно, разностную схему второго порядка точности.

В конце концов мы приходим к следующей разностной схеме (в безындексной форме):

Переходя к индексной форме записи, расписывая разностные производные и приводя подобные, мы придем к следующей системе линейных алгебраических уравнений:

Нетрудно заметить, что матрица системы имеет трехдиагональную структуру, поэтому решить ее можно методом разностной прогонки.

**Листинг**

**def form\_matrix\_1(N):**

**h = 1 / N # шаг сетки**

**x = [i \* h for i in range(N + 1)] # равномерная сетка**

**A = [.0] \* (N + 1) # коэффиценты при y\_i-1**

**B = [.0] \* (N + 1) # коэффиценты при y\_i**

**C = [.0] \* (N + 1) # коэффиценты при y\_i+1**

**F = [.0] \* (N + 1) # правый столбец системы**

**B[0] = k(x[0]) / h + get\_kappa0\_wave(KAPPA\_0, h) # нашли B[0]**

**C[0] = -k(x[0]) / h # нашли C[0]**

**F[0] = get\_g0\_wave(G\_0, h) # нашли F[0]**

**for i in range(1, N): # в цикле вычисляем A\_i, B\_i ,C\_i, F\_i**

**A[i] = -k\_first\_derivative(x[i]) / (2 \* h) + k(x[i]) / h\*\*2**

**B[i] = ( -2 \* k(x[i]) ) / h\*\*2 - q(x[i])**

**C[i] = ( k\_first\_derivative(x[i]) / (2 \* h) ) + ( k(x[i]) / h\*\*2 )**

**F[i] = -f(x[i])**

**A[N] = k(x[N]) / h # нашли A[N]**

**B[N] = -( k(x[N]) / h + get\_kappa1\_wave(KAPPA\_1, h) ) # нашли B[N]**

**F[N] = -get\_g1\_wave(G\_1, h) # нашли F[N]**

**return A, B, C, F**