**Теория**

Другой подход к решению исходной задачи состоит в использовании закона сохранения энергии. Действительно, формально мы можем разбить стержень на бесконечно большое число бесконечно малых частей и для каждой части стержня записать уравнения баланса тепла (по закону сохранения энергии количество теплоты, полученное участком стержня = количеству теплоты, отданное участком стержня).

Для каждого уравнения мы записываем уравнение баланса, заменяем интегралы квадратурными формулами и, таким образом, приходим к системе линейных алгебраических уравнений. На примере третьего уравнения исходной задачи, рассмотрим, как работает метод.

Записываем для данного уравнения уравнение баланса:

Здесь – количество теплоты, принятое левым концом участка стержня, а – количество теплоты отданное правым концом участка стержня, – количество теплоты, отданное окружающей среде, – количество теплоты, принятое от внешних источников. – тепловой поток.

.

Заменим интегралы в уравнении баланса и интеграл в коэффициенте квадратурными формулами правых прямоугольников и средних прямоугольников соответственно:

Приходим к следующему уравнению:

Расписав разностную производную и приведя подобные, получим следующее алгебраическое уравнение (последние уравнение итоговой СЛАУ):

Аналогичным образом выводятся уравнения для левого края стержня и внутренних узлов стержня.

Итоговая СЛАУ выглядит следующим образом:

Можно заметить, что матрица системы имеет трехдиагональную структуру, поэтому наиболее оптимальным методом решения СЛАУ будет метод разностной прогонки.

**Листинг**

**def form\_matrix\_2(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] = - ( KAPPA\_0 + k(x[0] + 0.5 \* h) / h + 0.5 \* h \* q(x[0]) ) # нашли B[0]**

**C[0] = k(x[0] + 0.5 \* h) / h # нашли C[0]**

**F[0] = - ( 0.5 \* h \* f(x[0]) + G\_0 ) # нашли F[0]**

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

**A[i] = k(x[i] - 0.5 \* h) / h**

**B[i] = -( k(x[i] + 0.5 \* h) / h + k(x[i] - 0.5 \* h) / h + h \* q(x[i]) )**

**C[i] = k(x[i] + 0.5 \* h) / h**

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

**A[N] = k(x[N] - 0.5 \* h) / h # нашли A[N]**

**B[N] = - ( KAPPA\_1 + k(x[N] - 0.5 \* h) / h + 0.5 \* h \* q(x[N]) ) # нашли B[N]**

**F[N] = - ( 0.5 \* h \* f(x[N]) + G\_1 ) # нашли F[N]**

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