|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 3**

|  |  |
| --- | --- |
| **Тема** Программно-алгоритмическая реализация моделей на основе ОДУ второго порядка с краевыми условиями II и III рода.  **Студент** Сушина А.Д.  **Группа** ИУ7-61б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Градов В.М. |  |

Москва.

2020 г

**Цель работы**. Получение навыков разработки алгоритмов решения краевой задачи при реализации моделей, построенных на ОДУ второго порядка.

# Исходные данные

1. Задана математическая модель.

Уравнение для функции

(1)

Краевые условия

(2)

2. Функции заданы своими константами

(3, 4)

Константы следует найти из условий , а константы из условий .

Величины задает пользователь.

Получаем:

(5)

(6)

(7)

(8)

3. Разностная схема с разностным краевым условием при .

(9)

Система (9) совместно с краевыми условиями решается методом прогонки.

Для величин численно вычисляя можно получить различные приближенные выражения, интеграл методом трапеций или методом средних. Выражения для приближений:

(10)

Разностный аналог краевого условия при х=0:

(11)

Простая аппроксимация:

(12)

(13)

Учесть, что поток , а . (14, 15)

4. Значения параметров для отладки (все размерности согласованы)

0.4 Вт/см К,

0.1 Вт/см К,

0.05 Вт/см2 К,

0.01 Вт/см2 К,

10 см,

300К,

0.5 см,

50 Вт/см2.

**Физическое содержание задачи.**

Сформулированная математическая модель описывает температурное поле вдоль цилиндрического стержня радиуса и длиной , причем и температуру можно принять постоянной по радиусу цилиндра. Ось направлена вдоль оси цилиндра и начало координат совпадает с левым торцем стержня. Слева при цилиндр нагружается тепловым потоком . Стержень обдувается воздухом, температура которого равна . В результате происходит съем тепла с цилиндрической поверхности и поверхности правого торца при . Функции являются, соответственно, коэффициентами теплопроводности материала стержня и теплоотдачи при обдуве.

# Код программы

Код программы представлен на листингах 1-2.

|  |
| --- |
| Листинг 1. functions.py  from data import data, graph1  import numpy as np  b = data['kN'] \* data['l'] / (data['kN'] - data['k0'])  a = - b \* data['k0']  d = data['AlphaN'] \* data['l'] / (data['AlphaN'] - data['Alpha0'])  c = - b \* data['Alpha0']  h = data['h']  N = data['l']  def k(x):  return a / (x - b)  def Alpha(x):  return c / (x - d)  def P(x):  return 2/data['R'] \* Alpha(x)  def f(x):  return 2\*data['t0']/data['R'] \*Alpha(x)  def KHIminus12(x):  return 2\*k(x)\*k(x-h)/(k(x) + k(x-h))  def KHIplus12(x):  return 2\*k(x)\*k(x+h)/(k(x) + k(x+h))  def A(n):  return KHIplus12(n)/h  def C(n):  return KHIminus12(n)/h  def B(n):  return A(n) + C(n) + P(n)\*h  def D(n):  return f(n) \* h  def calculate():  xstart = 0  xend = data['l']  #левые  p0 = P(xstart)  p1 = P(xstart + h)  p12 = (p0+p1)/2  f0 = f(xstart)  f1 = f(xstart+h)  f12 = (f0+f1)/2  M0 = KHIplus12(xstart) - h\*h/8\*p12  K0 = KHIplus12(xstart) + h\*h/8\*p12 + h\*h/4\*p0  P0 = h\*data['F0'] + h\*h/4\*(f12 + f0)  #правые  pn = P(xend)  pn1 = P(xend - h)  pn12 = (pn+pn1)/2  fn = f(xend)  fn1 = f(xend-h)  fn12 = (fn+fn1)/2  Mn = -KHIminus12(xend) + h \* h \* pn12 / 8  Kn = KHIminus12(xend) + data['AlphaN'] \* h + h \* h \* pn12 / 8 - h \* h \* pn / 4;  Pn = data['AlphaN'] \* data['t0'] \* h - h \* h \* (fn12 + fn) / 4;  eps = [0, M0/K0 ]  eta = [0, P0/K0]  x = h  n = 1  while x + h < N:  eps.append(C(x) / (B(x) - A(x) \* eps[n]))  eta.append((A(x) \* eta[n] + D(x)) / (B(x) - A(x) \* eps[n]))  n += 1  x += h  t = [0] \* (n + 1)  t[n] = (Pn - Mn \* eta[n]) / (Kn + Mn \* eps[n])  for i in range(n - 1, -1, -1):  t[i] = eps[i + 1] \* t[i + 1] + eta[i + 1]  graph1[0] = [i for i in np.arange(0, N, h)]  graph1[1] = t[:-1] |
| Листинг 2. main.py  import matplotlib.pyplot as plt  import data as const  from functions import calculate  from tkinter import \*  root = Tk()  varList = {  "k0": StringVar(),  "kN": StringVar(),  "Alpha0": StringVar(),  "AlphaN": StringVar(),  "l": StringVar(),  "t0": StringVar(),  "R": StringVar(),  "F0": StringVar(),  "h": StringVar()  }  def create\_grid(root):  i = 0  for var in varList.keys():  label = Label(root, text=var)  label.grid(row=i, column=0, sticky="e")  entry = Entry(root,width=10,textvariable=varList[var])  entry.grid(row=i, column=1)  entry.insert(0, str(const.resetData[var]))  i+=1  def check\_is\_num():  for var in varList.values():  try:  float(var.get())  except ValueError:  return False  return True  def clear\_graphs():  const.graph1[0].clear()  const.graph1[1].clear()  def start\_work(Event):  clear\_graphs()  if not check\_is\_num():  print("WARNING NOT DIGIT")  return  for var in varList.keys():  const.data[var] = float(varList[var].get())  calculate()  plt.title('T(x)')  plt.plot(const.graph1[0], const.graph1[1])  plt.xlable("x, sm")  plt.ylabel("T, K")  plt.show()  if \_\_name\_\_ == '\_\_main\_\_':  btn = Button(root, text="START")  create\_grid(root)  btn.bind("<Button-1>", start\_work)  btn.grid(column=1, padx=10, pady=10)  root.mainloop() |

# Результаты работы

1. Представить разностный аналог краевого условия при и его краткий вывод интегро -интерполяционным методом.

Обозначим .

Запишем (1) в виде:

(16)

где

Проитегрируем (15) на отрезке [xN-1/2, xN]

(17)

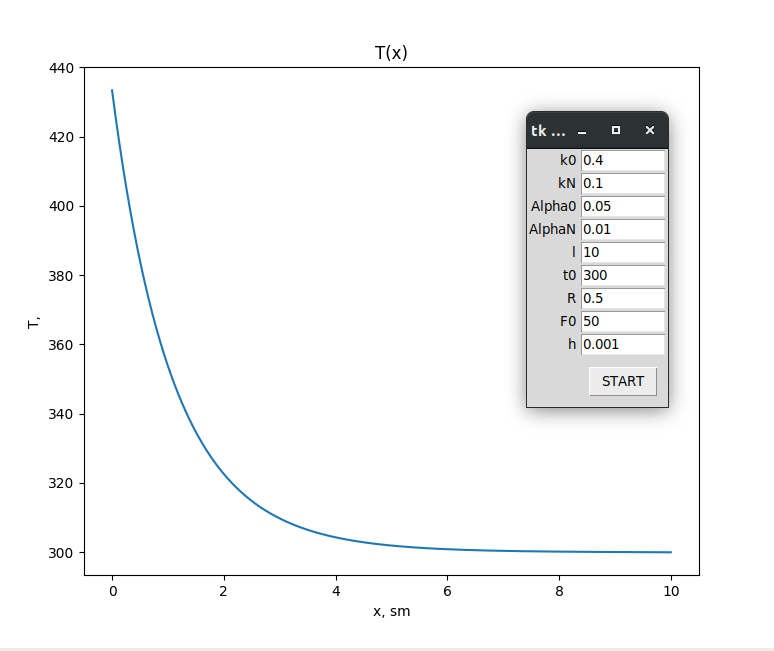
Применим метод трапеций. Получим:

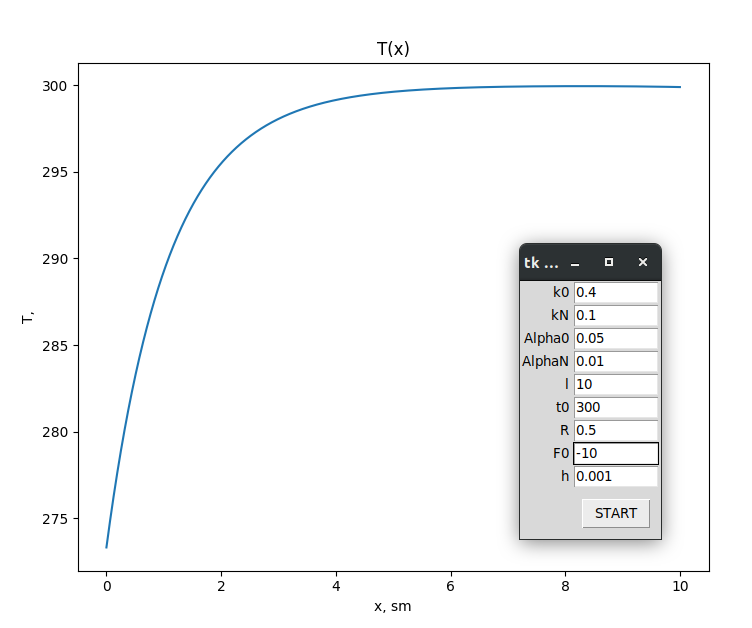
(18)

Применив (14, 15) к (18), получим

(19)

2. График зависимости температуры от координаты при заданных выше параметрах.



3. График зависимости при -10 Вт/см2.

*Справка*. При отрицательном тепловом потоке слева идет съем тепла, поэтому производная должна быть положительной.

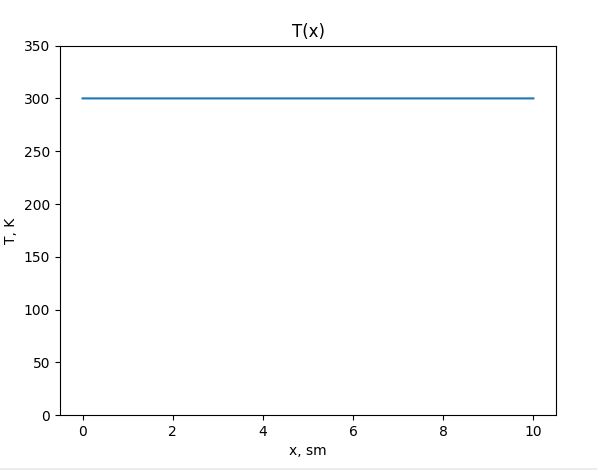
4. График зависимости при увеличенных значениях (например, в 3 раза). Сравнить с п.2.

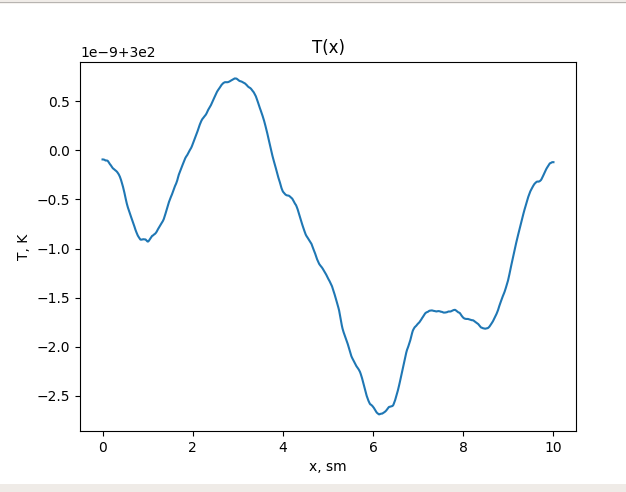
*Справка*. При увеличении теплосъема и неизменном потоке уровень температур должен снижаться, а градиент увеличиваться.

|  |  |
| --- | --- |
| увеличено в 3 раза | Пункт 2 |

5. График зависимости при 0.

*Справка*. В данных условиях тепловое нагружение отсутствует, причин для нагрева нет, температура стержня должна быть равна температуре окружающей среды (разумеется с некоторой погрешностью, определяемой приближенным характером вычислений).

Погрешность:



# Вопросы при защите лабораторной работы

Ответы на вопросы дать письменно в Отчете о лабораторной работе.

1. Какие способы тестирования программы можно предложить?

Программа должна выводить результаты, соответствующие законам физики. Это можно определить по виду графиков при определенных значениях входных параметров. Отталкиваясь от физического смысла задачи, можно представить как должен вести себя результат. Например, при F0=0 температура должна быть равна температуре окружающей среды Т0. При отрицательном тепловом потоке слева идет съем тепла, поэтому производная должна быть положительной.

2. Получите простейший разностный аналог нелинейного краевого условия при

,

где - заданная функция.

Производную аппроксимируйте односторонней разностью.

(20)

3. Опишите алгоритм применения метода прогонки, если при краевое условие линейное (как в настоящей работе), а при , как в п.2.

Будет использоваться правая прогонка.

(21)

Из краевого условия при x = 0 получим:

(22)

отсюда начальные прогоночные коэффиценты:

(23)

(24)

Прогоночные коэффициенты:

(25)

(26)

Исходя из того, что

(27)

найдем :

В (20) подставим (27):

(28)

Таким образом, два этапа:

Прямой ход: по 23, 24 вычисляем начальные значения коэфицентов, по 25, 26 вычисляем массивы прогоночных коэфицентов.

Обратный ход: по (28) определяем . Далее по 27 вычисяем все значения y.

4. Опишите алгоритм определения **единственного** значения сеточной функции в **одной** заданной точке . Использовать встречную прогонку, т.е. комбинацию правой и левой прогонок (лекция №8). Краевые условия линейные.

Левая прогонка:

(29)

Прогоночные коэффициенты:

(30)

(31)

Объединив левую и правую (21), (25), (26) прогонки, получим систему:

(32)

Подставив второе уравнение в первое:

(33)