**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)»**

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

**Лабораторная работа № 5**

“Решение неоднородного дифференциального уравнения второго порядка в постановке краевой задачи”

по дисциплине «Численные методы»

Вариант 17

Работу выполнил

студент группы ИУ9-62Б

Сербин Денис

Москва, 2022

# **1. Цель работы**

Целью работы является аналитическое решение задачи Коши, численное решение краевой задачи, а так же вычисление погрешности численного решения.

**2. Постановка задачи**

Дано:

Тестовый вариант:

Найти:

1. Аналитическое решение задачи
2. Численное решение задачи
3. Погрешность численного решения

**3. Индивидуальный вариант**

**4. Теоретические сведения**

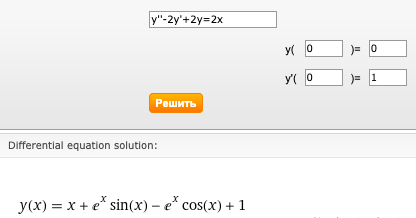
h= , где n –разбиение отрезка [a, b]

C помощью аппроксимации производных получим приближенную систему уравнений:

После преобразования получим:

c краевыми условиями y(0)=a, y(n)=b

**5. Практическая реализация**

Решим аналитически задачу с помощью Wolfram alpha:

Подставив в решение точки a и b получим:

y(0) = 0

y(1) = 2+e\*sin(1)-e\*cos(1)

**import** math  
  
  
**def** exp\_func(x):  
 **return** math.exp(x)  
  
  
**def** exp\_q\_func(x):  
 **return** -1  
  
  
**def** exp\_p\_func(x):  
 **return** 1  
  
  
x\_exp\_A = 0  
x\_exp\_B = 1  
  
y\_exp\_A = exp\_func(x\_exp\_A)  
y\_exp\_B = exp\_func(x\_exp\_B)  
  
n = 10  
  
h\_exp = (x\_exp\_B - x\_exp\_A) / n  
  
f\_exp = [0] \* n  
p\_exp = [0] \* n  
q\_exp = [0] \* n  
  
  
**def** my\_func(x):  
 **return** 2 \* x  
  
  
**def** my\_func\_d(x):  
 **return** 2  
  
  
**def** my\_func\_dd(x):  
 **return** 0  
  
  
**def** new\_func(x):  
 **return** x + math.exp(x) \* (math.sin(x) - math.cos(x)) + 1  
  
  
**def** my\_p\_func(x):  
 **return** -2  
  
  
**def** my\_q\_func(x):  
 **return** 2  
  
  
x\_my\_func\_A = 0  
x\_my\_func\_B = 1  
  
# y\_my\_func\_A = my\_func(x\_my\_func\_A)  
# y\_my\_func\_B = my\_func(x\_my\_func\_B)  
  
y\_my\_func\_A = 0  
y\_my\_func\_B = 2 + math.exp(1) \* (math.sin(1) - math.cos(1))  
  
h\_my\_func = (x\_my\_func\_B - x\_my\_func\_A) / n  
  
f\_my\_func = [0] \* n  
p\_my\_func = [0] \* n  
q\_my\_func = [0] \* n  
  
  
**def** solve\_matrix(a, b, c, d):  
 size = len(d)  
 alpha = [- c[0] / b[0]]  
 beta = [d[0] / b[0]]  
  
 **for** i **in** range(1, size):  
 **if** i != size - 1:  
 alpha.append(- c[i] / (a[i - 1] \* alpha[i - 1] + b[i]))  
 beta.append((d[i] - a[i - 1] \* beta[i - 1]) / (a[i - 1] \* alpha[i - 1] + b[i]))  
  
 res = [0.0 **for** i **in** range(size)]  
 res[-1] = beta[-1]  
  
 i = size - 2  
 **while** i > -1:  
 res[i] = alpha[i] \* res[i + 1] + beta[i]  
 i -= 1  
  
 **return** res  
  
  
**def** count(n, x\_A, h, func, p\_func, q\_func, f, p, q):  
 **for** i **in** range(0, n):  
 x = x\_A + i \* h  
 # f[i] = func\_dd(x) + p\_func(x) \* func\_d(x) + q\_func(x) \* func(x)  
 f[i] = func(x)  
 p[i] = p\_func(x)  
 q[i] = q\_func(x)  
  
  
**def** solve(n, x\_A, y\_A, y\_B, h, f, p, q, func):  
 a = []  
 b = [q[1] \* h \* h - 2]  
 c = [1 + p[1] \* (h / 2)]  
 d = [f[1] \* h \* h - y\_A \* (1 - p[1] \* (h / 2))]  
  
 **for** i **in** range(2, n - 1):  
 a.append(1 - p[i] \* (h / 2))  
 b.append(q[i] \* h \* h - 2)  
 c.append(1 + p[i] \* (h / 2))  
 d.append(f[i] \* h \* h)  
  
 a.append(1 - p[n - 1] \* (h / 2))  
 b.append(q[n - 1] \* h \* h - 2)  
 d.append(f[n - 1] \* h \* h - y\_B \* (1 + p[n - 1] \* (h / 2)))  
  
 res = [y\_A]  
 **for** x **in** solve\_matrix(a, b, c, d):  
 res.append(x)  
  
 res.append(y\_B)  
  
 text = ""  
  
 **for** i **in** range(0, n):  
 text += "{}: x: {}\n y: {}\n solve\_result: {}\n ".format(i, x\_A + i \* h, func(x\_A + i \* h), res[i])  
 text += "diff: {0:0.20f}\n\n".format(abs(func(x\_A + i \* h) - res[i]))  
  
 print(text)  
  
  
**if** \_\_name\_\_ == '\_\_main\_\_':  
 # count(n, x\_exp\_A, h\_exp, exp\_func, exp\_func, exp\_func,  
 # exp\_p\_func, exp\_q\_func, f\_exp, p\_exp, q\_exp)  
 # solve(n, x\_exp\_A, y\_exp\_A, y\_exp\_B, h\_exp,  
 # f\_exp, p\_exp, q\_exp, exp\_func)  
  
 count(n, x\_my\_func\_A, h\_my\_func, new\_func,  
 my\_p\_func, my\_q\_func, f\_my\_func, p\_my\_func, q\_my\_func)  
 solve(n, x\_my\_func\_A, y\_my\_func\_A, y\_my\_func\_B, h\_my\_func,  
 f\_my\_func, p\_my\_func, q\_my\_func, new\_func)

**6. Результаты**

Для тестирования был взят индивидуальный вариант:

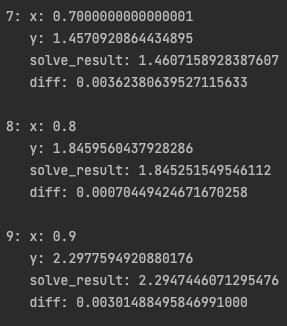
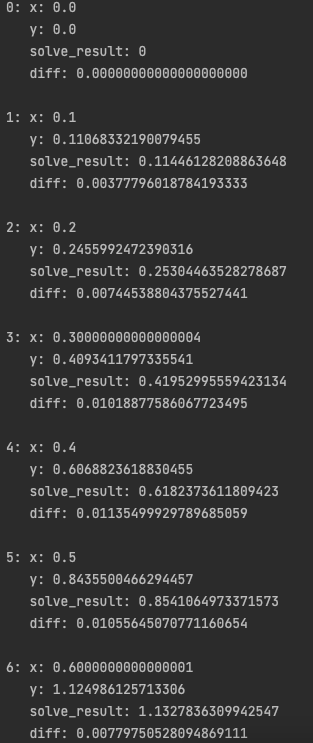


Рис.1 Выходные результаты программы

Таблица 1 Результаты программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | Значение x | Значение у | Численное значение | Абсолютная погрешность |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0.1 | 0.11068332190079455 | 0.11446128208863648 | 0.00377796018784193333 |
| 2 | 0.2 | 0.2455992472390316 | 0.25304463528278687 | 0.00744538804375527441 |
| 3 | 0.3 | 0.4093411797335541 | 0.41952995559423134 | 0.01018877586067723495 |
| 4 | 0.4 | 0.6068823618830455 | 0.6182373611809423 | 0.01135499929789685059 |
| 5 | 0.5 | 0.8435500466294457 | 0.8541064973371573 | 0.01055645070771160654 |
| 6 | 0.6 | 1.124986125713306 | 1.1327836309942547 | 0.00779750528094869111 |
| 7 | 0.7 | 1.4570920864434895 | 1.4607158928387607 | 0.00362380639527115633 |
| 8 | 0.8 | 1.8459560437928286 | 1.845251549546112 | 0.00070449424671670258 |
| 9 | 0.9 | 2.2977594920880176 | 2.2947446071295476 | 0.00301488495846991000 |

**7. Вывод**

В ходе выполнения лабораторной работы были выполнено аналитическое решение задачи Коши, численное решение краевой задачи, а так же вычисление погрешности численного решения, и написана реализация на языке программирования Python.

Данный метод вычисляет результат с высокой точностью. Погрешность обусловлена малым количеством разбиений отрезка.