Міністерство освіти і науки України  
НТУУ «КПІ ім. Ігоря Сікорського»  
Кафедра диференціальних рівнянь

Лабораторна робота №7  
Тема: “Апроксимація сплайнами”  
Варіант № 6

Виконав студент 2-го курсу  
ТЕФ, групи ТР-71  
Зуєв Михайло Олександрович  
  
Перевірив: Сігайов А.О.

Київ – 2018

""" Лабораторна работа номер 7  
 з курсу Чисельні методи, варіант 6  
 Завдання: Побудувати кубічний сплайн з глобальним заданням нахилів і  
 розвязуванням СЛАР розмірністю n-2 віносно других похідних Si''(xi)  
 для функції f(x), яку задано таблично.  
 Для оцінювання похибки значень функція f(x) задана.  
 f(x) = x^2 + sin(x)  
 Виконав студент 2 курсу: Зуєв Михайло Олександрович  
"""  
  
import math  
import numpy as np  
import matplotlib.pyplot as plt  
from numpy.core.multiarray import ndarray  
  
  
def my\_function(x):  
 return x \* x + math.sin(x)  
  
  
def spline(x\_data, y\_data, n):  
 """  
 Строит полином вида Si(x) = ai + bi(x - xi) + ci(x - xi)^2 + di(x - xi)^3,  
 для таблично заданной функции.  
 **:param** x\_data: Таблица аргументов функции  
 **:param** y\_data: Таблица значений функции  
 **:param** n: количество точек  
 **:return**: массивы коэффициентов ai, bi, ci, di  
 """  
 # Массивы коэффициентов  
 a: ndarray = np.array([y\_data[i] for i in range(n - 1)], dtype=np.float64)  
 b: ndarray = np.zeros(n - 1, dtype=np.float64)  
 c: ndarray = np.zeros(n - 1, dtype=np.float64)  
 d: ndarray = np.zeros(n - 1, dtype=np.float64)  
  
 # Значенния h: h[i] = x[i + 1] - x[i]  
 h = np.array([x\_data[i + 1] - x\_data[i] for i in range(n - 1)], dtype=np.float64)  
  
 # y[i + 2] - y[i + 1] y[i + 1] - y[i]  
 # Значения g: g[i] = ( ------------------- - --------------- )  
 # h[i + 1] h[i]  
 g = np.array([(((y\_data[i + 2] - y\_data[i + 1]) / h[i + 1]) - ((y\_data[i + 1] - y\_data[i]) / h[i])) \  
 \* 3 for i in range(n - 2)], dtype=np.float64)  
  
 # Формируем СЛАР t\*c = g  
 # ┍ -----------------------------------------------------------------┑ ┍--------┑  
 # | 2(h1 + h2) h2 | | c2 |  
 # | h2 2(h2 + h3) h3 | | c3 |  
 # | ...... | | .. |  
 # | h[n-2] 2(h[n - 2) + h[n - 1]) | | c[n-1] |  
 # ┕ -----------------------------------------------------------------┙ ┕--------┙  
 t = np.zeros((n - 2, n - 2), dtype=np.float64)  
  
 t[0, 0] = (h[0] + h[1]) \* 2  
 t[0, 1] = h[1]  
 for i in range(1, n - 3):  
 t[i, i - 1] = h[i]  
 t[i, i] = (h[i] + h[i + 1]) \* 2  
 t[i, i + 1] = h[i + 1]  
 t[n - 3, n - 4] = h[n - 3]  
 t[n - 3, n - 3] = (h[n - 3] + h[n - 2]) \* 2  
  
 # Решаем СЛАР и находим сi, i >= 2 (c1 = 0)  
 c[1:] = np.linalg.solve(t, g)  
  
 # Вычисляем d1, ..., d[n - 1] и b1, ..., b[n - 1]  
 for i in range(n - 2):  
 d[i] = (c[i + 1] - c[i]) / (3 \* h[i])  
 b[i] = ((y\_data[i + 1] - y\_data[i]) / h[i]) - (((c[i + 1] + 2 \* c[i]) \* h[i]) / 3.0)  
 d[n - 2] = (-c[n - 2]) / (3 \* h[n - 2])  
 b[n - 2] = ((y\_data[n - 1] - y\_data[n - 2]) / h[n - 2]) - (2.0 / 3.0) \* c[n - 2] \* h[n - 2]  
  
 return a, b, c, d  
  
  
def find\_index(x\_data, x):  
 '''  
 Находит индекс i такой что x є [x\_data[i]; x\_data[i+1]]  
 **:param** x\_data: таблица аргументов функции  
 **:param** x: арумент для которого ищется индекс  
 **:return**: нужный индекс  
 '''  
 # Определяем i  
 global i  
 x\_size = len(x\_data)  
 flag = True  
 for j in range(x\_size - 2):  
 if x\_data[j] <= x <= x\_data[j + 1]:  
 i = j  
 flag = False  
 break  
 if flag:  
 i = 0 if x < x\_data[0] else x\_size - 2  
 return i  
  
  
def calculate\_spline(a, b, c, d, x\_data, x):  
 """  
 Считает значение полинома вида Si(x) = ai + bi(x - xi) + ci(x - xi)^2 + di(x - xi)^3  
 в точке x  
 **:param** a: массив коэффициентов a полинома  
 **:param** b: массив коэффициентов b полинома  
 **:param** c: массив коэффициентов c полинома  
 **:param** d: массив коэффициентов d полинома  
 **:param** x\_data: таблица аргументов функции  
 **:param** x: аргумент полинома  
 **:return**: значение полинома Si(x)  
 """  
 # Определяем i  
 i = find\_index(x\_data, x)  
  
 # Считаем значение Si(x)  
 tmp = x - x\_data[i]  
 tmp\_2 = tmp \*\* 2  
 tmp\_3 = tmp\_2 \* tmp  
 return a[i] + b[i] \* tmp + c[i] \* tmp\_2 + d[i] \* tmp\_3  
  
  
def calculate\_derivative(x, n, x\_data, d=[], c=[], b=[], a=[], pos='right'):  
 """  
 Вычисляет значение производной полинома вида Si(x) = ai + bi(x - xi) + ci(x - xi)^2 + di(x - xi)^3  
 **:param** x: аргумент полинома  
 **:param** n: степень производной  
 **:param** x\_data: агрументы функции  
 **:param** pos: 'left' - что бы посчитать производную слева  
 **:param** a: массив коэффициентов a полинома  
 **:param** b: массив коэффициентов b полинома  
 **:param** c: массив коэффициентов c полинома  
 **:param** d: массив коэффициентов d полинома  
 **:return**: значение производной n-й степени Si(x)  
 """  
 if n == 0:  
 return calculate\_spline(a, b, c, d, x\_data, x)  
  
 i = find\_index(x\_data, x)  
 if pos == 'left':  
 i = i - 1 if i > 0 else 0  
 if n == 1:  
 tmp = x - x\_data[i]  
 return b[i] + 2 \* c[i] \* tmp + 3 \* d[i] \* tmp \* tmp  
 elif n == 2:  
 tmp = x - x\_data[i]  
 return 2 \* c[i] + 6 \* d[i] \* tmp  
 elif n == 3:  
 return 6 \* d[i]  
 return 0.0  
  
  
# Границы функции  
left = 1  
right = 3  
  
# Количество точек  
m = 6  
  
# Шаг  
H = (right - left) / (m - 1)  
  
# Таблица значений функций  
xData = np.zeros(m)  
yData = np.zeros(m)  
  
# Создание таблицы  
for i in range(m):  
 xData[i] = left + i \* H  
 yData[i] = my\_function(xData[i])  
  
# Получаем коэффициенты кубического сплайна  
a, b, c, d = spline(xData, yData, m)  
  
# Рисуем график и выводим таблицу  
h1 = (right - left) / 20  
H2 = 2 \* H  
start = left - H2  
end = right + H2  
xlist = []  
ylist1 = []  
ylist2 = []  
print("----------------------------------------------------------------------")  
print("| | | | | f(xj) - Si(xj) |")  
print("| xj | f(xj) | Si(xj) | f(xj) - Si(xj) | -------------- \* 100 |")  
print("| | | | | Si(xj) |")  
print("----------------------------------------------------------------------")  
while start <= end:  
 f = my\_function(start)  
 s = calculate\_spline(a, b, c, d, xData, start)  
 xlist.append(start)  
 ylist1.append(f)  
 ylist2.append(s)  
 # Выводим таблицу  
 print("|{0:5.2f} | {1:8.3f} | {2:8.3f} | {3:14.9f} | {4:21.16f}|".format(start, f, s, s - f, (s - f) \* 100 / f))  
 start += h1  
print("----------------------------------------------------------------------")  
  
plt.plot(xlist, ylist1, 'k', label='f(x)')  
plt.plot(xlist, ylist2, '--', label='S(x)')  
  
plt.legend()  
plt.show()  
  
# Дополнительное задание  
eps = 1e-10  
print("----------------------------------------------------------------------------")  
print("| | | | | S^(a)(xj+0) - S^(a)(xj-0) |")  
print("| xj | a | S^(a)(xj-0) | S^(a)(xj+0) | ------------------------- \* 100 |")  
print("| | | | | S^(a)(xj-0) |")  
print("----------------------------------------------------------------------------")  
for j in range(1, m - 1):  
 x = xData[j]  
 for t in range(4):  
 l = calculate\_derivative(x + eps, t, xData, d, c, b, a, pos='left')  
 r = calculate\_derivative(x + eps, t, xData, d, c, b, a, pos='right')  
 if l != 0:  
 tmp = (r - l) \* 100 / l  
 print("|{0:5.2f} |{1:4d} |{2:12.7f} |{3:12.7f} |{4:33.27f}|".format(x, t, l, r, tmp))  
 else:  
 print("|{0:5.2f} |{1:4d} |{2:12.7f} |{3:12.7f} | -------------------------- |".format(x, t, l, r))  
print("----------------------------------------------------------------------------")

# Результати роботи:

----------------------------------------------------------------------

| | | | | f(xj) - Si(xj) |

| xj | f(xj) | Si(xj) | f(xj) - Si(xj) | -------------- \* 100 |

| | | | | Si(xj) |

----------------------------------------------------------------------

| 0.20 | 0.239 | -0.574 | -0.813132509 | -340.6941754583515944|

| 0.30 | 0.386 | -0.216 | -0.601122993 | -155.9251583830628931|

| 0.40 | 0.549 | 0.121 | -0.428908185 | -78.0658657218382359|

| 0.50 | 0.729 | 0.437 | -0.292300252 | -40.0726648690800147|

| 0.60 | 0.925 | 0.737 | -0.187150234 | -20.2402808818134901|

| 0.70 | 1.134 | 1.025 | -0.109357037 | -9.6416268482340648|

| 0.80 | 1.357 | 1.302 | -0.054875937 | -4.0428548582485586|

| 0.90 | 1.593 | 1.574 | -0.019726522 | -1.2380712479501306|

| 1.00 | 1.841 | 1.841 | 0.000000000 | 0.0000000000000000|

| 1.10 | 2.101 | 2.109 | 0.008134222 | 0.3871213491555003|

| 1.20 | 2.372 | 2.380 | 0.008422729 | 0.3550839218264029|

| 1.30 | 2.654 | 2.658 | 0.004523134 | 0.1704554298411576|

| 1.40 | 2.945 | 2.945 | -0.000000000 | -0.0000000000000151|

| 1.50 | 3.247 | 3.245 | -0.002358890 | -0.0726372248211894|

| 1.60 | 3.560 | 3.557 | -0.002586480 | -0.0726626342316393|

| 1.70 | 3.882 | 3.880 | -0.001495882 | -0.0385371235603414|

| 1.80 | 4.214 | 4.214 | 0.000000000 | 0.0000000000000211|

| 1.90 | 4.556 | 4.557 | 0.001071177 | 0.0235097980241242|

| 2.00 | 4.909 | 4.911 | 0.001518101 | 0.0309229886678984|

| 2.10 | 5.273 | 5.274 | 0.001228670 | 0.0233002290202779|

| 2.20 | 5.648 | 5.648 | 0.000000000 | 0.0000000000000000|

| 2.30 | 6.036 | 6.034 | -0.002124770 | -0.0352033463752890|

| 2.40 | 6.435 | 6.432 | -0.003854429 | -0.0598935819438218|

| 2.50 | 6.848 | 6.845 | -0.003640015 | -0.0531507667012588|

| 2.60 | 7.276 | 7.276 | 0.000000000 | 0.0000000000000000|

| 2.70 | 7.717 | 7.725 | 0.007291571 | 0.0944824624790235|

| 2.80 | 8.175 | 8.189 | 0.013625877 | 0.1666776338494772|

| 2.90 | 8.649 | 8.662 | 0.013155605 | 0.1521011238104787|

| 3.00 | 9.141 | 9.141 | -0.000000000 | -0.0000000000000389|

| 3.10 | 9.652 | 9.620 | -0.031745581 | -0.3289158747023446|

| 3.20 | 10.182 | 10.094 | -0.087999867 | -0.8643007380588587|

| 3.30 | 10.732 | 10.558 | -0.174685741 | -1.6276705298086955|

| 3.40 | 11.304 | 11.007 | -0.297720254 | -2.6336532899268708|

| 3.50 | 11.899 | 11.436 | -0.463004712 | -3.8910519984480434|

| 3.60 | 12.517 | 11.841 | -0.676414909 | -5.4037628427232134|

| 3.70 | 13.160 | 12.216 | -0.943791619 | -7.1715795395302671|

----------------------------------------------------------------------

----------------------------------------------------------------------------

| | | | | S^(a)(xj+0) - S^(a)(xj-0) |

| xj | a | S^(a)(xj-0) | S^(a)(xj+0) | ------------------------- \* 100 |

| | | | | S^(a)(xj-0) |

----------------------------------------------------------------------------

| 1.40 | 0 | 2.9454497 | 2.9454497 | 0.000000000000000000000000000|

| 1.40 | 1 | 2.9332608 | 2.9332608 | 0.000000000000000000000000000|

| 1.40 | 2 | 1.2998542 | 1.2998542 | -0.000000031415199964532657212|

| 1.40 | 3 | 3.2496355 | -0.8338817 | -125.660775576650337370665511116|

| 1.80 | 0 | 4.2138476 | 4.2138476 | 0.000000000000000000000000000|

| 1.80 | 1 | 3.3864919 | 3.3864919 | 0.000000000000000000000000000|

| 1.80 | 2 | 0.9663015 | 0.9663015 | 0.000000011295613807857583863|

| 1.80 | 3 | -0.8338817 | 0.2576146 | -130.893423631510160021207411774|

| 2.20 | 0 | 5.6484964 | 5.6484964 | 0.000000000000000000000000000|

| 2.20 | 1 | 3.7936217 | 3.7936217 | 0.000000000000011706207071406|

| 2.20 | 2 | 1.0693474 | 1.0693474 | 0.000000018639250904972167361|

| 2.20 | 3 | 0.2576146 | 2.2507973 | 773.707208507142240705434232950|

| 2.60 | 0 | 7.2755014 | 7.2755014 | 0.000000000000000000000000000|

| 2.60 | 1 | 4.4014244 | 4.4014244 | 0.000000000000000000000000000|

| 2.60 | 2 | 1.9696663 | 1.9696663 | -0.000000036427301704333166878|

| 2.60 | 3 | 2.2507973 | -4.9241657 | -318.774285558183009925414808095|

----------------------------------------------------------------------------

