Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

Лабораторна робота № 5

З дисципліни «Спеціальні розділи математики»

На тему: «**Інтерполяційні поліноми»**

Варіант № 26

Виконав студент гр. ІС-91

Хмелiнiн А.М.

Перевірила викладач

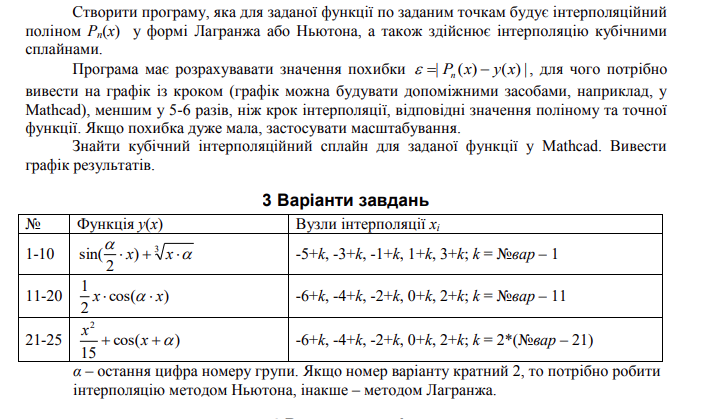
доц. Рибачук Л.В.

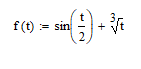
Київ 2020

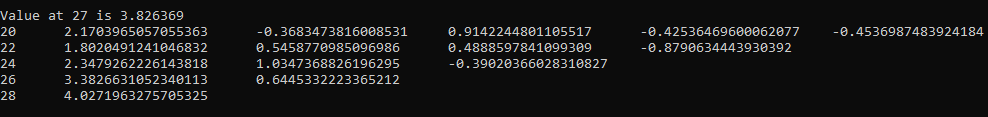
Зміст

1. Постановка задачі…………………………………………………..3
2. Вигляд поліномів ………………………………………………….4
3. Порівняльний графік функції та інтерполяційного поліному…...5
4. Сплайни (коефіцієнти інтерполяційних поліномів)……………...5
5. Розв’язок у MathCAD………………………………………………6
6. Лістинг програми…………………………………………………...7

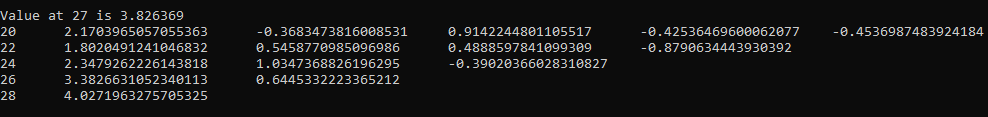
Постановка задачі



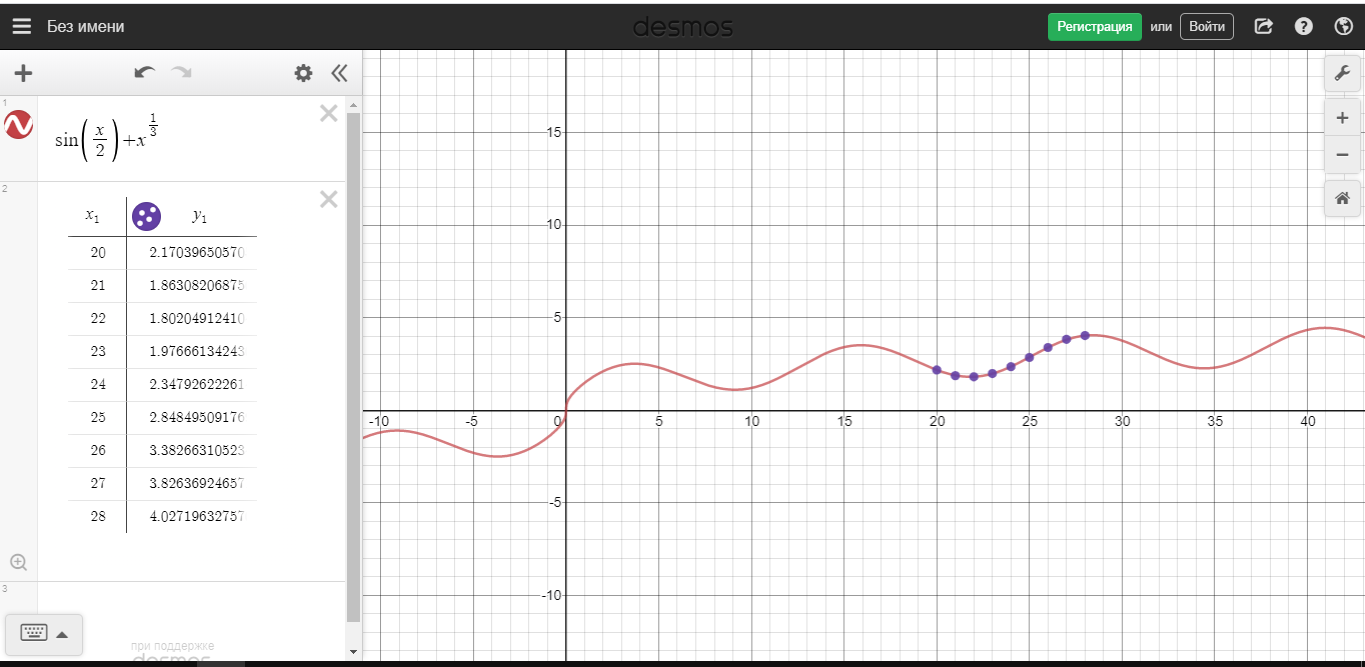


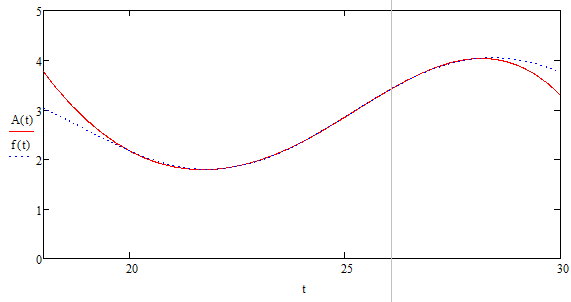


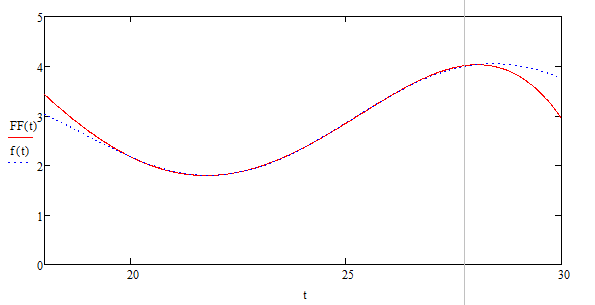
Вигляд поліномів



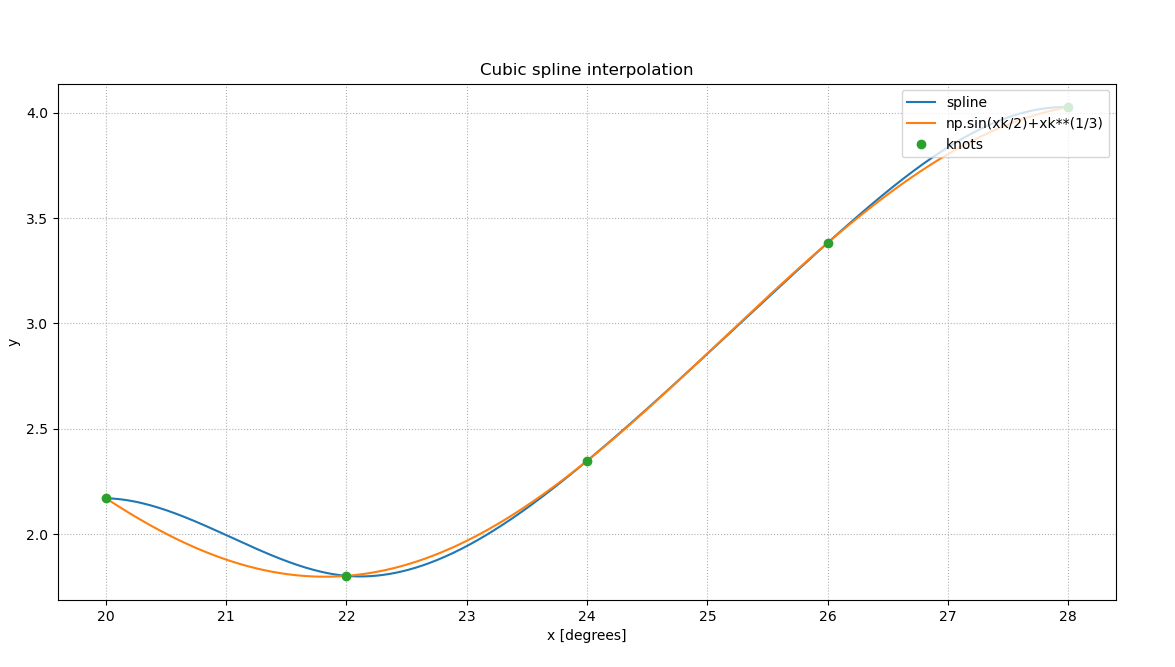
Порівняльний графік функції та інтерполяційного поліному



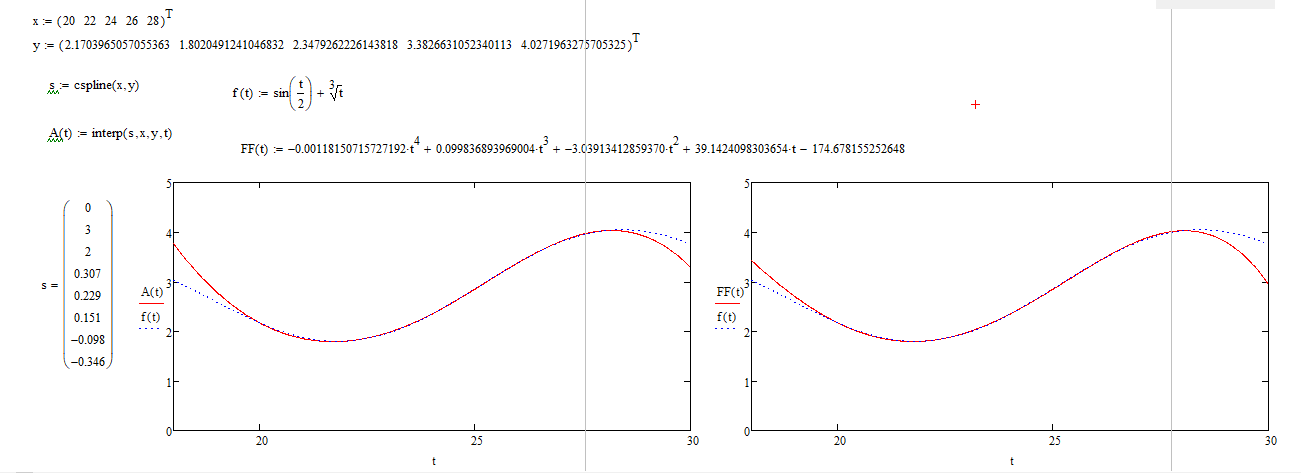




Сплайни (коефіцієнти інтерполяційних поліномів)



Розв’язок у MathCAD



Лістинг програми

import numpy as np

import matplotlib.pyplot as plt

import math

def my\_func(x):

return math.sin((1/2)\*x)+math.pow((x\*1),1/3)

def fact(n):

f = 1;

for i in range(2, n + 1):

f \*= i;

return f

def find\_q(q, n):

temp = q;

for i in range(1, n):

temp = temp \* (q - i);

return temp;

n = 5

x = x = [20, 22, 24, 26, 28];

y = [[0 for i in range(n)]

for j in range(n)];

y[0][0] = my\_func(x[0])

y[1][0] = my\_func(x[1])

y[2][0] = my\_func(x[2])

y[3][0] = my\_func(x[3])

y[4][0] = my\_func(x[4])

#ситаем разницу столбцов

for i in range(1, n):

for j in range(n - i):

y[j][i] = y[j + 1][i - 1] - y[j][i - 1];

#

value =27

#

sum = y[0][0];

q = (value - x[0]) / (x[1] - x[0]);

for i in range(1, n):

sum = sum + (find\_q(q, i) \* y[0][i]) / fact(i);

print("\nValue at", value,

"is", round(sum, 6));

for i in range(n):

print(x[i], end = "\t");

for j in range(n - i):

print(y[i][j], end = "\t");

print("");

ar\_x=x = [20, 22, 24, 26, 28];

ar\_y=[2.1703965057055363, -0.3683473816008531, 0.9142244801105517, -0.42536469600062077, -0.4536987483924184]

print("\nValue at", value,

"is", sum)

print('true value for ',value ,my\_func(value))

############################################################################################################################

#create sampling points for sin(x)

xk=np.linspace(20,28,5)

yk=np.sin(xk/2)+xk\*\*(1/3)

n=len(xk)-1

hk=np.array([xk[i+1]-xk[i] for i in range(n)])

print(xk,yk,hk)

def abcd(yk,hk,yppk):

ak=yk

bk=np.array([(yk[i+1]-yk[i])/hk[i]-(2.\*yppk[i]+yppk[i+1])\*hk[i]/6.for i in range(len(hk))])

ck=yppk\*0.5

dk=np.array([(yppk[i+1]-yppk[i])/hk[i]/6. for i in range(len(hk))])

return (ak,bk,ck,dk)

def S(x,a,b,c,d,xk):

i=0

while i<len(xk)-1 and x>=xk[i]:

i+=1

i-=1

dx=x-xk[i]

return a[i]+dx\*(b[i]+dx\*(c[i]+dx\*d[i]))

A=np.zeros((n,n))

b=np.zeros(n)

#calculate the matrix vector coefficients

A[0,0]=2\*(hk[0]+hk[n-1])

A[0,1]=hk[0]

A[0,-1]=hk[n-1]

b[0]=6\*((yk[1]-yk[0])/hk[0]-(yk[n]-yk[n-1])/hk[n-1])

for i in range(1,n):

A[i,i]=2\*(hk[i-1]+hk[i])

A[i,i-1]=hk[i-1]

if i<n-1:A[i,i+1]=hk[i]

b[i]=6\*((yk[i+1]-yk[i])/hk[i]-(yk[i]-yk[i-1])/hk[i])

A[-1,0]=hk[n-1]

print(A,"x,y'' =",str(b))

z=np.linalg.solve(A,b)

yppn=z[0]

ypp=np.append(z,yppn)

print("\nSolution for y'' \_0 ... y'' \_n:",ypp)

(a,b,c,d)=abcd(yk,hk,ypp)

print("asdsadsad",a,b,c,d)

xkplot=np.linspace(xk[0],xk[-1],600)

spline=[S(x,a,b,c,d,xk) for x in xkplot]

ax=plt.subplot(1,1,1)

ax.set\_title("Cubic spline interpolation")

ax.set\_xlabel("x [degrees]")

ax.set\_ylabel("y")

ax.grid(linestyle="dotted")

ax.plot(xkplot,spline,label="spline")

ax.plot(xkplot,np.sin(xkplot/2)+xkplot\*\*(1/3),label="np.sin(xk/2)+xk\*\*(1/3)")

ax.plot(xk,yk,"o",label="knots")

ax.legend(loc="upper right")

plt.show()

