**Лабораторная работа №2**

**«Интерполирование кубическим сплайном»**

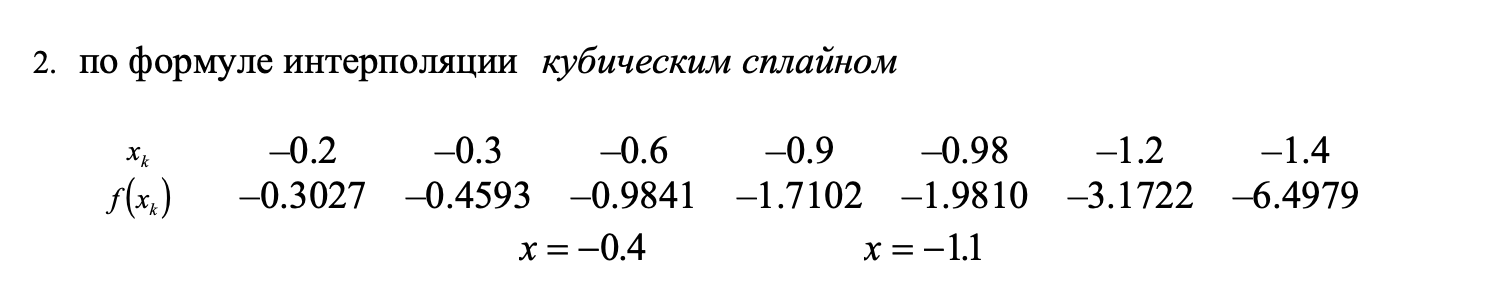
Выполнил:

Студент 2 курса, 5 группы,

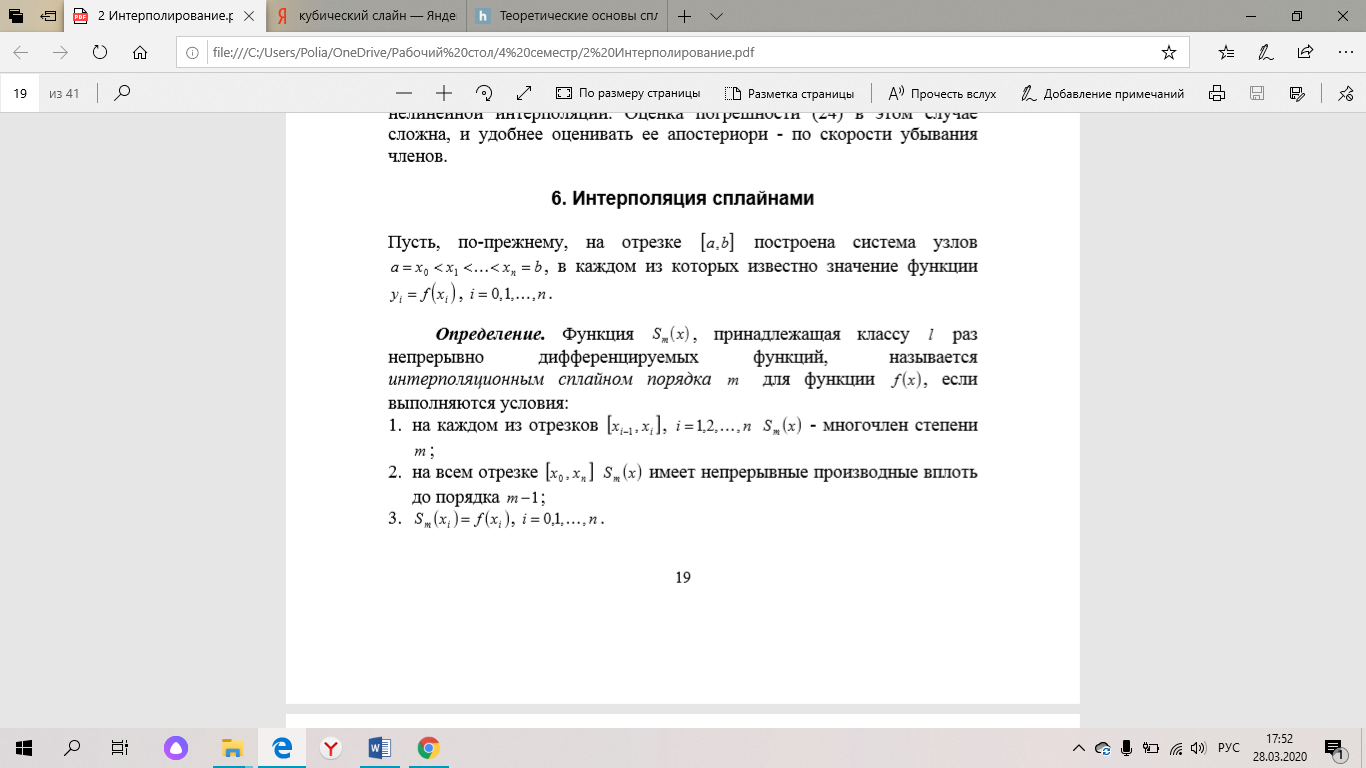
Смоляков Адам Алексеевич

Вариант 38

Условие:



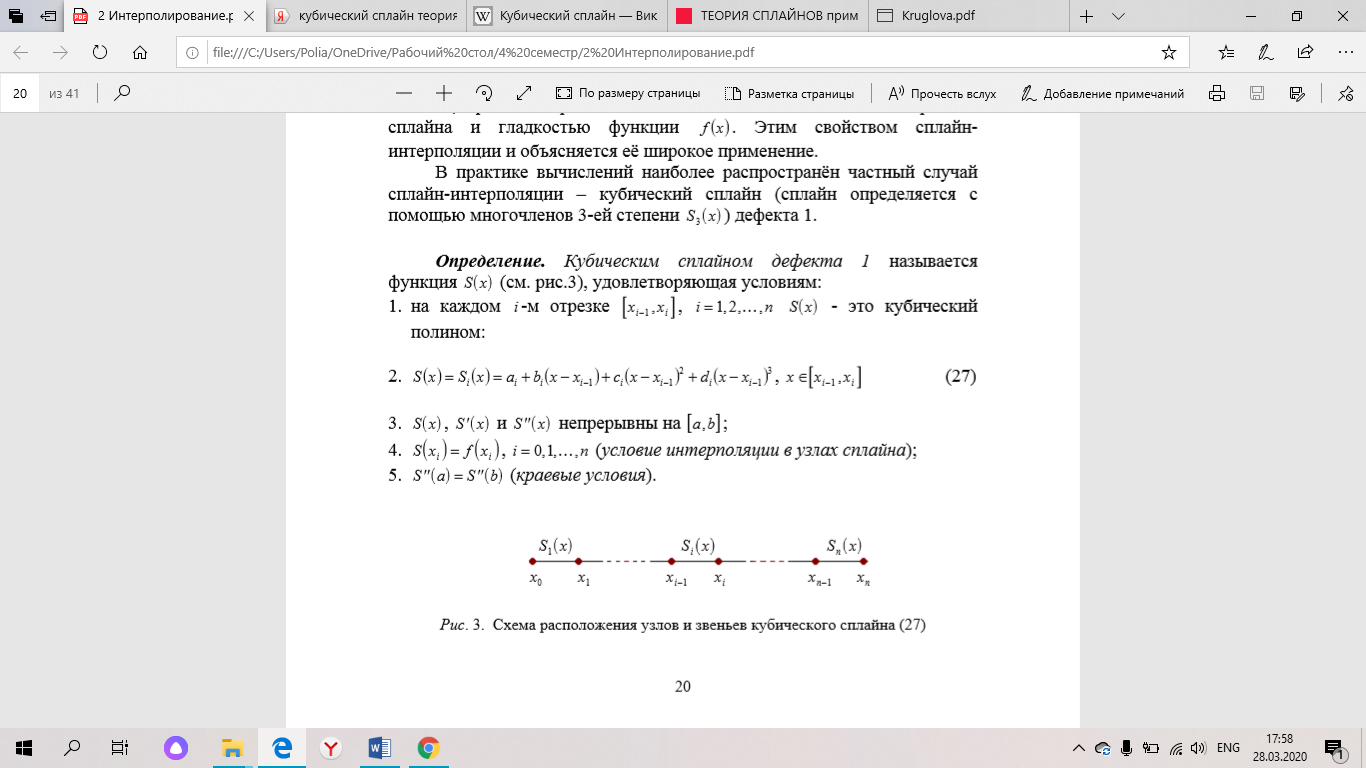
**Теория метода:**

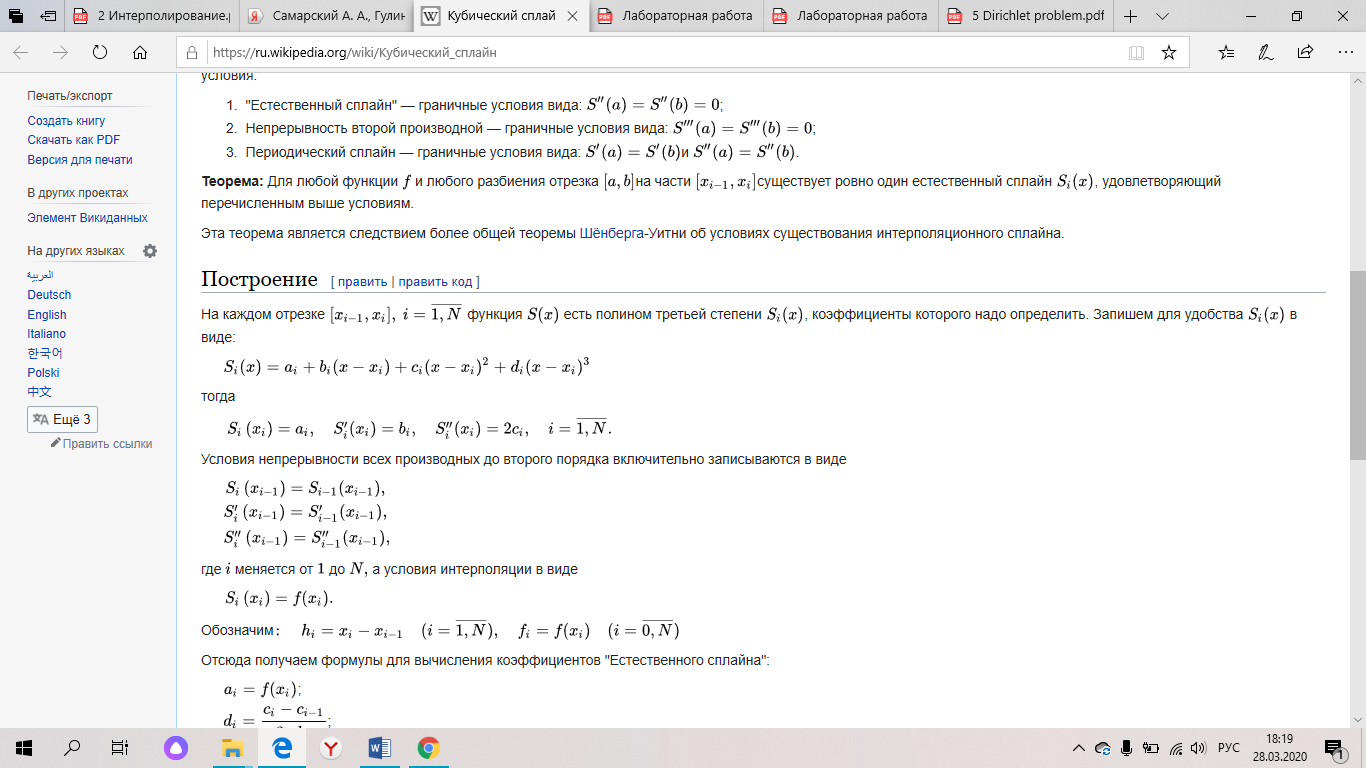


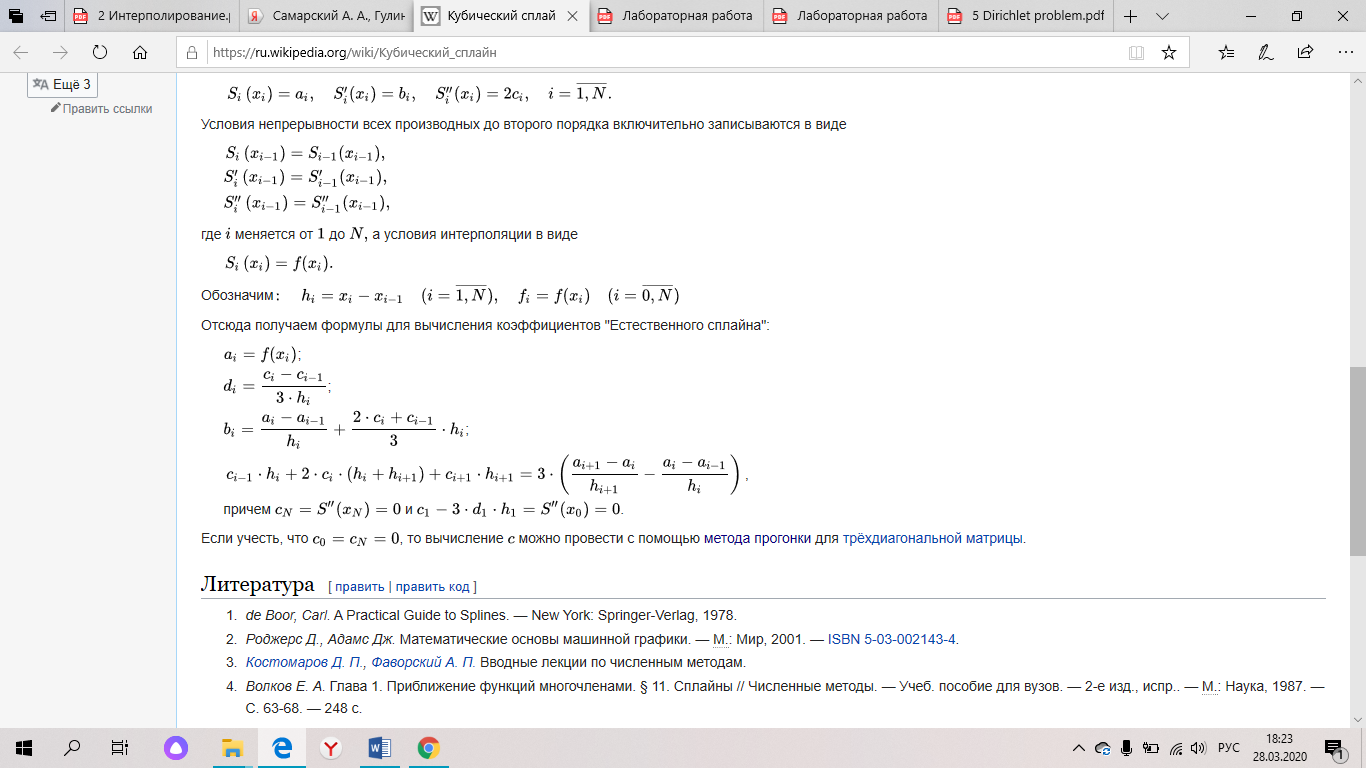
**Кубический сплайн** — гладкая функция, область определения которой разбита на конечное число отрезков, на каждом из которых она совпадает с некоторым кубическим многочленом (полиномом).

Разность d=m-l − называется дефектом сплайна.

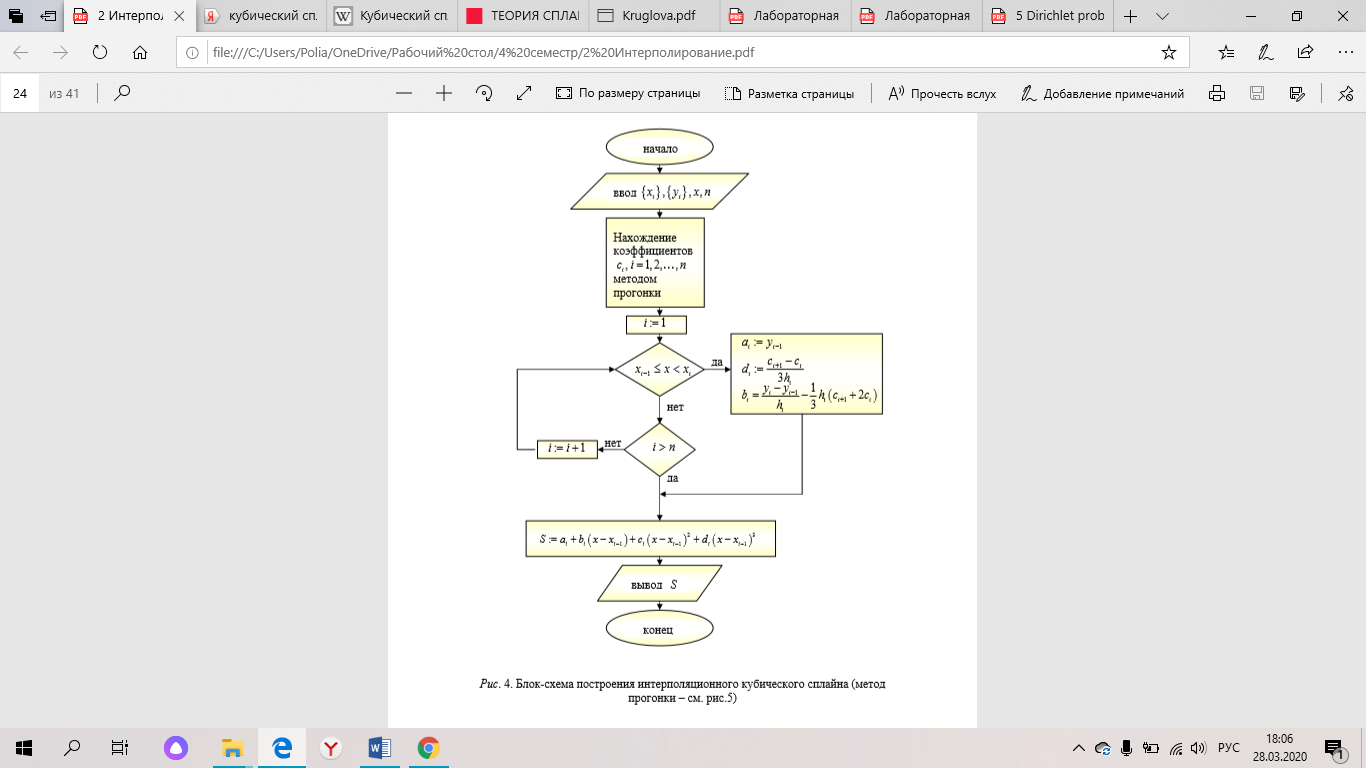
**Кубическим сплайном дефекта l наз. Функция S(x), удовлетворяющая условиям:**





Интерполяция применяется во многих задачах, связанных с вычислениями. Укажем некоторые из этих задач. Обработка физического эксперимента — построение приближенных формул для характерных величин по табличным данным, полученным экспериментально. Построение приближенных формул по данным вычислительного эксперимента. Здесь возникают нестандартные задачи интерполяции, так как обычно пишутся формулы возможно более простой структуры. Интерполяционные формулы используются также ври вычислении интегралов, при написании разностных аппроксимации для дифференциальных уравнений на основе интегральных тождеств. Математическое обеспечение любой ЭВМ содержит стандартные программы интерполирования.

**Блок-схема метода :**



**Код метода:**

**const** n=**6**;

x:**array**[**0**..**6**] **of** **real** = (**0.2** ,**0.3**,**0.6**,**0.9**,**0.98**,**1.2**,**1.4**);

y:**array**[**0**..**6**] **of** **real** = (**0.3027**,**0.4593**,**0.9841**,**1.7102**,**1.9810**,**3.1722**,**6.4979**);

**function** **Spline**(x1:**real**):**real**; //Построение интерполяционного кубического сплайна

**var** i:**integer**;

h,l,k:**array**[**1**..n] **of** **real**;

w,u,v:**array**[**2**..n] **of** **real**;

a,d,b:**array**[**1**..n] **of** **real**;

c:**array**[**1**..n+**1**] **of** **real**;

s:**real**;

**begin**

**for** i := **1** **to** n **do** //Прямой ход метода прогонки

h[i]:=x[i]-x[i-**1**];

**for** i := **2** **to** n **do**

**begin**

w[i]:=h[i-**1**];

u[i]:=**2**\*(h[i-**1**]+h[i]);

v[i]:=**3**\*((y[i]-y[i-**1**])/h[i]-(y[i-**1**]-y[i-**2**])/h[i-**1**])

**end**;

k[**1**]:=**0**;

l[**1**]:=**0**;

**for** i := **2** **to** n **do**

**begin**

k[i]:=(v[i]-w[i]\*k[i-**1**])/(u[i]-w[i]\*l[i-**1**]);

l[i]:=h[i]/(u[i]-w[i]\*l[i-**1**]);

**end**;

c[n+**1**]:=**0**; //Обратный ход метода прогонки

**for** i := n **downto** **1** **do**

c[i]:=k[i]-l[i]\*c[i+**1**];

**for** i := **1** **to** n **do** //Построение сплайна

**if** (x1>=x[i-**1**]) **and** (x1<x[i]) **then**

**begin**

a[i]:=y[i-**1**];

d[i]:=(c[i+**1**]-c[i])/(**3**\*h[i]);

b[i]:=(y[i]-y[i-**1**])/h[i]-h[i]\*(c[i+**1**]+**2**\*c[i])/**3**;

Spline:=a[i]+b[i]\*(x1-x[i-**1**])+c[i]\*sqr((x1-x[i-**1**]))+d[i]\*Power(x1-x[i-**1**],**3**);

**end**;

**end**;

**procedure** **TForm1**.**Button1Click**(Sender: **TObject**); //Исходная функция

**var** i:**integer**;

**begin**

**for** i := **0** **to** n **do**

Chart1.Series[**0**].AddXY(x[i],y[i]);

**end**;

**procedure** **TForm1**.**Button2Click**(Sender: **TObject**); //Вычисление значений функции в точках x=1.13 и x=1.17

**begin**

Chart1.Series[**2**].AddXY(**0.4**,Spline(**0.4**));

Label2.Caption:=Label2.Caption+FloatToStrF(Spline(**0.4**),ffFixed,**4**,**4**);

Chart1.Series[**2**].AddXY(**1.1**,Spline(**1.1**));

Label4.Caption:=Label4.Caption+FloatToStrF(Spline(**1.1**),ffFixed,**4**,**4**);

**end**;

**procedure** **TForm1**.**Button3Click**(Sender: **TObject**);

**var** x1,dx,f:**real**;

**begin**

x1:=**0.2**;

dx:=**0.0005**;

**repeat**

f:=Spline(x1);

Chart1.Series[**1**].AddXY(x1,f);

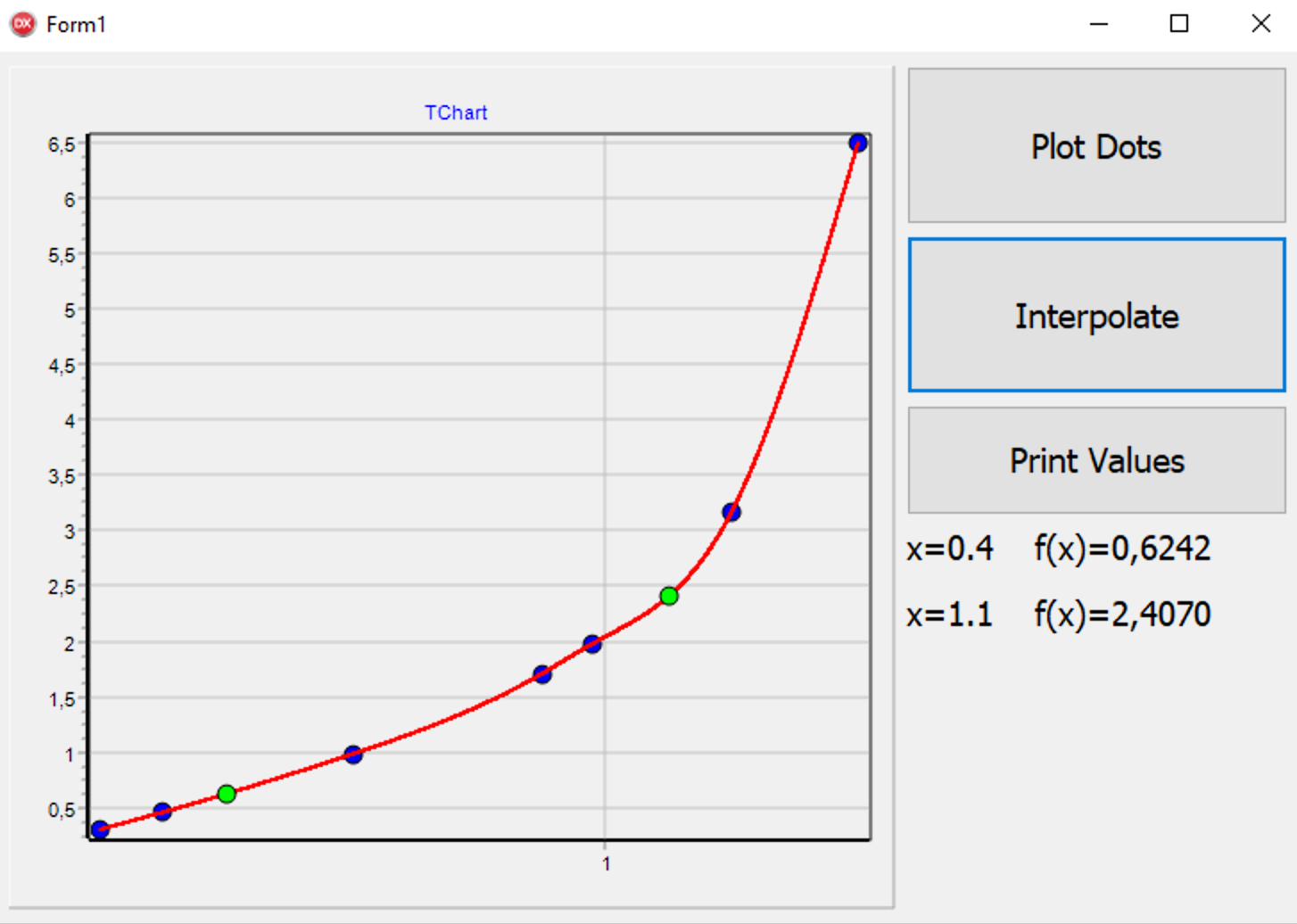
x1:=x1+dx;

**until** x1>**1.4**;

**end**;

**end**.

**Ответ:**

****

**Вывод.**

В ходе выполнения лабораторной работы был изучен кубический сплайн для интерполирования функций. Из графика видно, что метод хорошо интерполирует данную зависимость (на графике нет скачков).