МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Приближение функций

Лабораторная работа №4

по дисциплине «Вычислительная математика»

Выполнил студент Чекулаев В. Ю.

Факультет, группа ФКФН, ПО(аб)-81

Проверил Резак Е.В.

Хабаровск – 2020г.

Исходная информация: наборы значений (x, y) .

Наборы точек записаны в файлы.

Требуется: Построить графики:

- Прямой, полученной методом наименьших квадратов
- Интерполяционных полиномов Лагранжа и Ньютона

Метод наименьших квадратов

Исходная информация: набор значений (x_i, y_i) $(i=1,2,3..,N)$.

Требуется построить функцию F(x), такую, что:

1. $F(x)=kx+b$
2. для F(x) выполняется условие

$$\sum_{i=1}^{N}(F(x_i)-y_i)^2=\sum_{i=1}^{N}((kx_i+b)-y_i)^2-min$$

Обозначим

$$S(k,b)=\frac{1}{2}\sum_{i=1}^{N}((kx_i+b)-y_i)^2$$

Для нахождения точки минимума функции двух переменных S(k,b) запишем необходимые условия минимума

$$\frac{\partial S}{\partial k}=0 \quad , \quad \frac{\partial S}{\partial b}=0$$

Используя правила дифференцирования, получаем

$$\frac{\partial S}{\partial k}=k\sum_{i=1}^{N}x_i^2+b\sum_{i=1}^{N}x_i-\sum_{i=1}^{N}x_i y_i \quad ; \quad \frac{\partial S}{\partial b}=k\sum_{i=1}^{N}x_i+b\sum_{i=1}^{N}1-\sum_{i=1}^{N}y_i$$

Таким образом приходим к системе двух уравнений для двух неизвестных k и b :

$$k\sum_{i=1}^{N}x_i^2+b\sum_{i=1}^{N}x_i=\sum_{i=1}^{N}x_i y_i$$

$$k\sum_{i=1}^{N}x_i+b\sum_{i=1}^{N}1=\sum_{i=1}^{N}y_i$$

Выражая из второго уравнения b и подставляя в первое уравнение, получим следующие формулы для вычисления k и b:

$$k=(\sum_{i=1}^{N} x_i y_i-\frac{1}{N}\sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i)/(\sum_{i=1}^{N} x_i^2-\frac{1}{N}\sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i)$$

$$b=\frac{1}{N}(\sum_{i=1}^{N} y_i-k\sum_{i=1}^{N} x_i)$$

Интерполяционный полином Лагранжа

Исходная информация: таблица значений функции y=f(x).

Требуется построить функцию L(x), удовлетворяющую двум условиям:

1. $F(x_i)=y_i, i=0,..n;$
2. $L(x)=P_n(x).$

Итоговая формула:

$$L_n(x)=\sum_{i=1}^{n} y_i \frac{(x-x_0)*...*(x-x_{i-1})(x-x_{i+1})*...*(x-x_n)}{(x_i-x_0)*...*(x_i-x_{i-1})(x_i-x_{i+1})*...*(x_i-x_n)}$$

Интерполяционный полином Ньютона

Конечные разности

Пусть функция задана таблицей с постоянным шагом. Разности между значениями функции в соседних узлах интерполяции называются конечными разностями первого порядка:

$$\delta y_i=y_{i+1}-y_i (i=0,1,2,...).$$

Из конечных разностей первого порядка образуются конечные разности второго порядка:

$$\delta^2 y_i=\delta y_{i+1}-\delta y_i (i=0,1,2,...).$$

Аналогично для разностей третьего порядка:

$$\delta^3 y_i=\delta^2 y_{i+1}-\delta^2 y_i (i=0,1,2,...).$$

Первая интерполяционная формула Ньютона

$$P_n(x)=P_n(x_0+th)=y_0+t\,\delta y+\frac{t(t-1)}{2!}\delta^2 y_0+...+\frac{t(t-1)*...*(t-n+1)}{n!}\delta^n y_0$$ *, где*

$$t=\frac{x-x_0}{h}$$

Эта формула применяется для интерполирования в начале отрезка инерполяции, когда t мало оп абсолютной величине. Первую интерполяционную форму Ньютона называют по этой причине формулой для интерполирования вперед. За начальное приближение x_0 можнл принимать любое табличное значение аргумента x.

Вторая интерполяционная формула Ньютона

Когда значение аргумента находится ближе к концу отрезка интерполяции, применять первую интерполяционную формулу становится невыгодно. В этом случае применяется формула для интерполирования назад — вторая интерполяционная формула Ньютона.

$$P_n(x)=P_n(x_n+th)=y_n+t\,\delta y_{n-1}+\frac{t(t+1)}{2!}\delta^2 y_{n-2}+\ldots+\frac{t(t+1)*\ldots*(t+n-1)}{n!}\delta^n y_0$$

Основные функции программы

Метод Наименьших Квадратов

```
int N = 10;  // кол-во исходных точек
int n = 0;  // кло-во точек на графике
double h = 0.01; // шаг
double ax = 0, bx = 10;
QVector<double> x(N);//double *x = new double[N];
QVector<double> y(N);//double *y = new double[N];

double *summs = new double[4];
for(int i = 0; i < 4; i++){
    summs[i] = 0;
}

x[0] = 4.7; y[0] = 2.8; // точки
x[1] = 4.8; y[1] = 3.5;
x[2] = 5.3; y[2] = 4.3;
x[3] = 5.6; y[3] = 4.1;
x[4] = 5.9; y[4] = 5.4;
x[5] = 6.7; y[5] = 6.3;
x[6] = 7.2; y[6] = 6.9;
x[7] = 7.7; y[7] = 7.9;
x[8] = 8.0; y[8] = 8.1;
x[9] = 9.0; y[9] = 8.3;
```

```
    for(int i = 0; i < N; i++){
        summs[0] += x[i];
        summs[1] += y[i];
        summs[2] += x[i]*y[i];
        summs[3] += x[i]*x[i];
    }

    double k = (summs[2]-summs[0]*summs[1]/N)/(summs[3]-summs[0]*summs[0]/N);
    double b = (summs[1]-k*summs[0])/N;
```

Полином Лагранжа

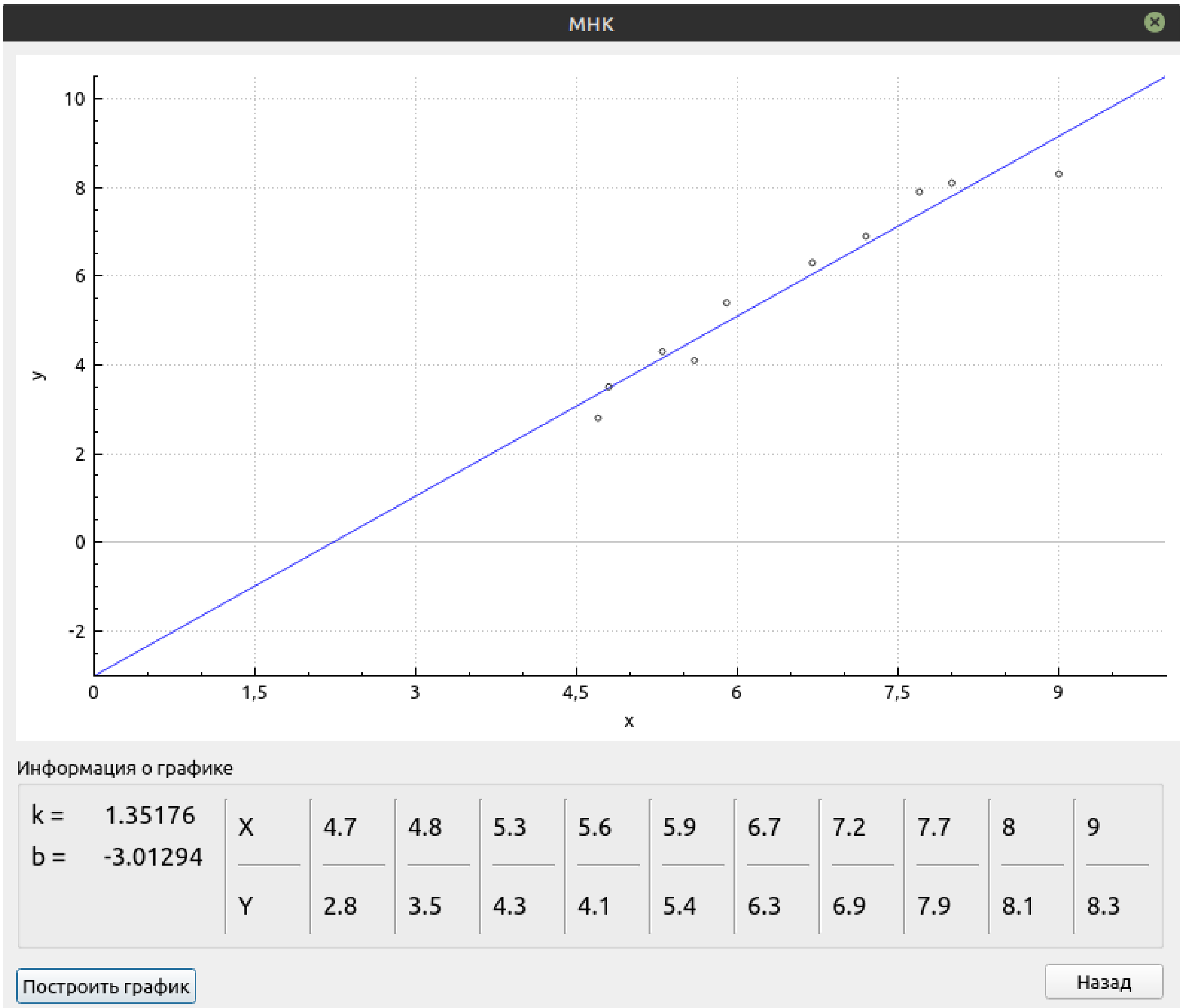
```
double lagrange(QVector<double> x, QVector<double> y, double X){
    double result = 0;
    double p;
    for(int i = 0; i < x.size(); i++){
        p = 1;
        for(int j = 0; j < x.size(); j++){
            if(j!=i){
                p *= (X-x[j])/(x[i]-x[j]);
            }
        }
        result += y[i]*p;
    }

    return result;
}
```

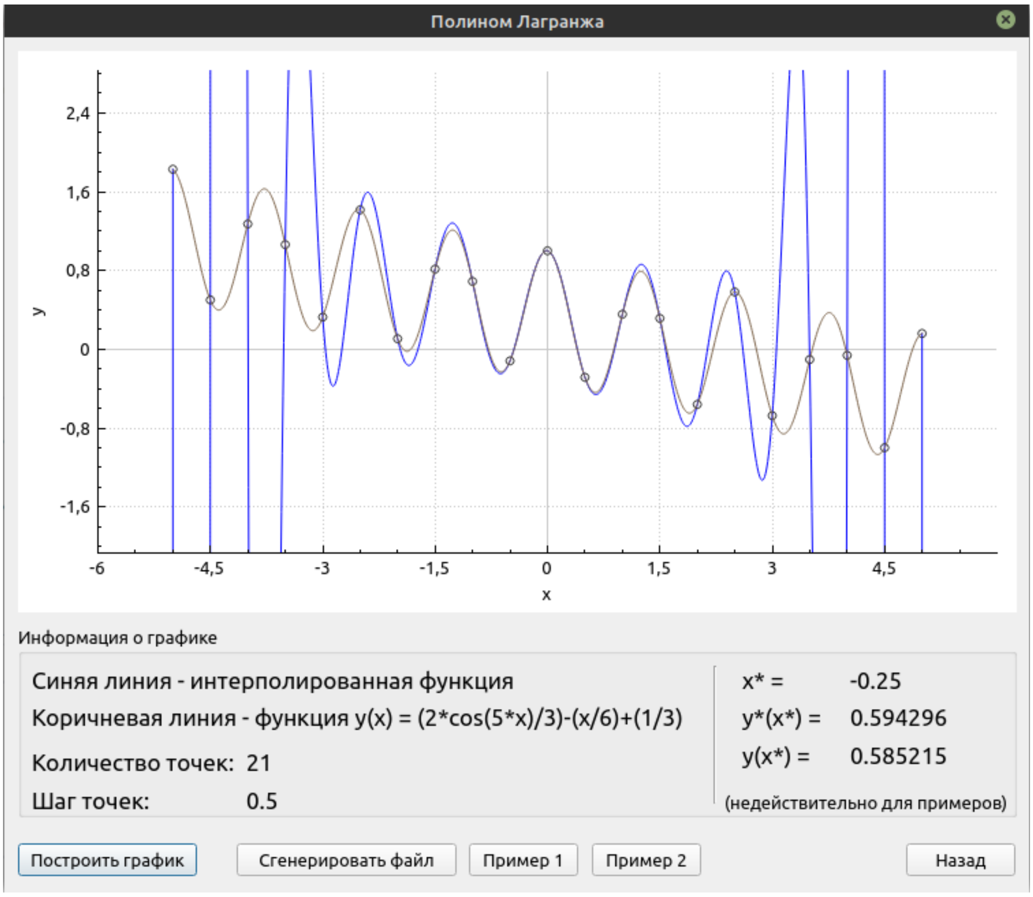
Полином Ньютона

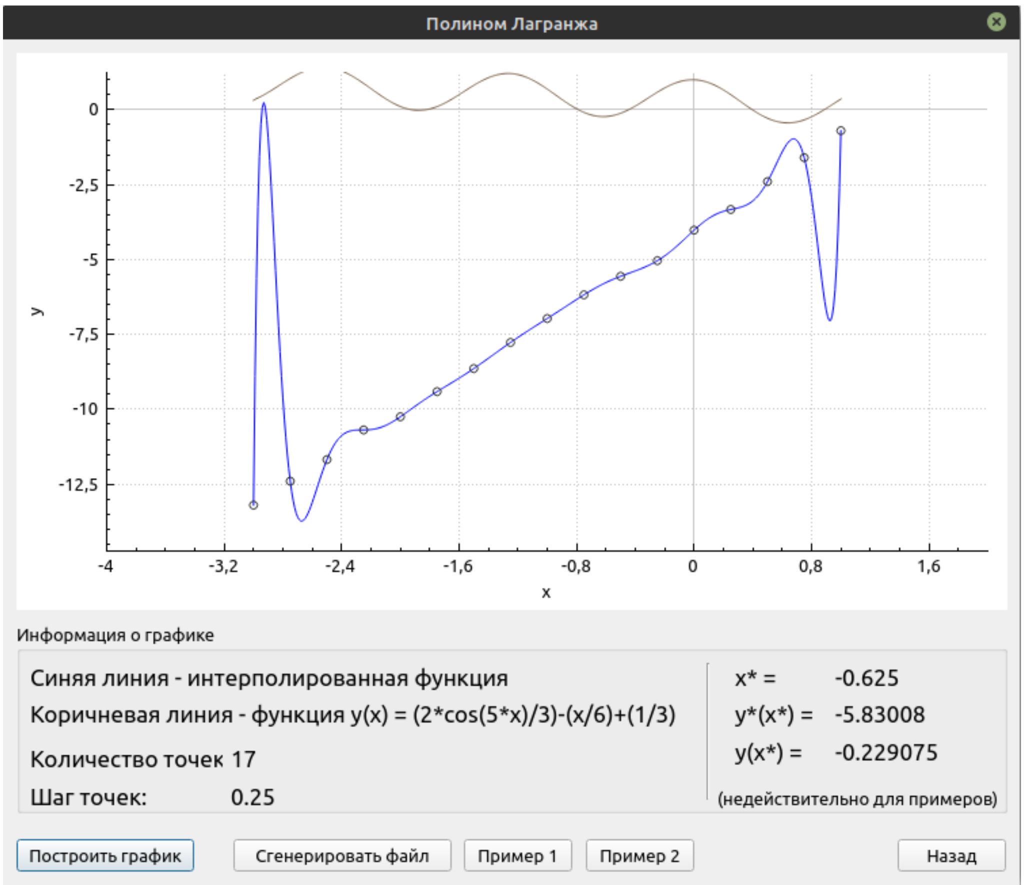
```
double newton(QVector<double> x, QVector<double> y, double X){
    QVector<double> temp(y.size());
    QVector<double> temp1(y.size());
    for(int i = 0; i < temp1. size(); i++){
        temp1[i] = 1;
    }
    temp = y;
    double result = temp[0];
    int inter = 0;
    int c = x.size()-1;
    int count = 1;

    for(int i = 1; i < y.size(); i++){
```

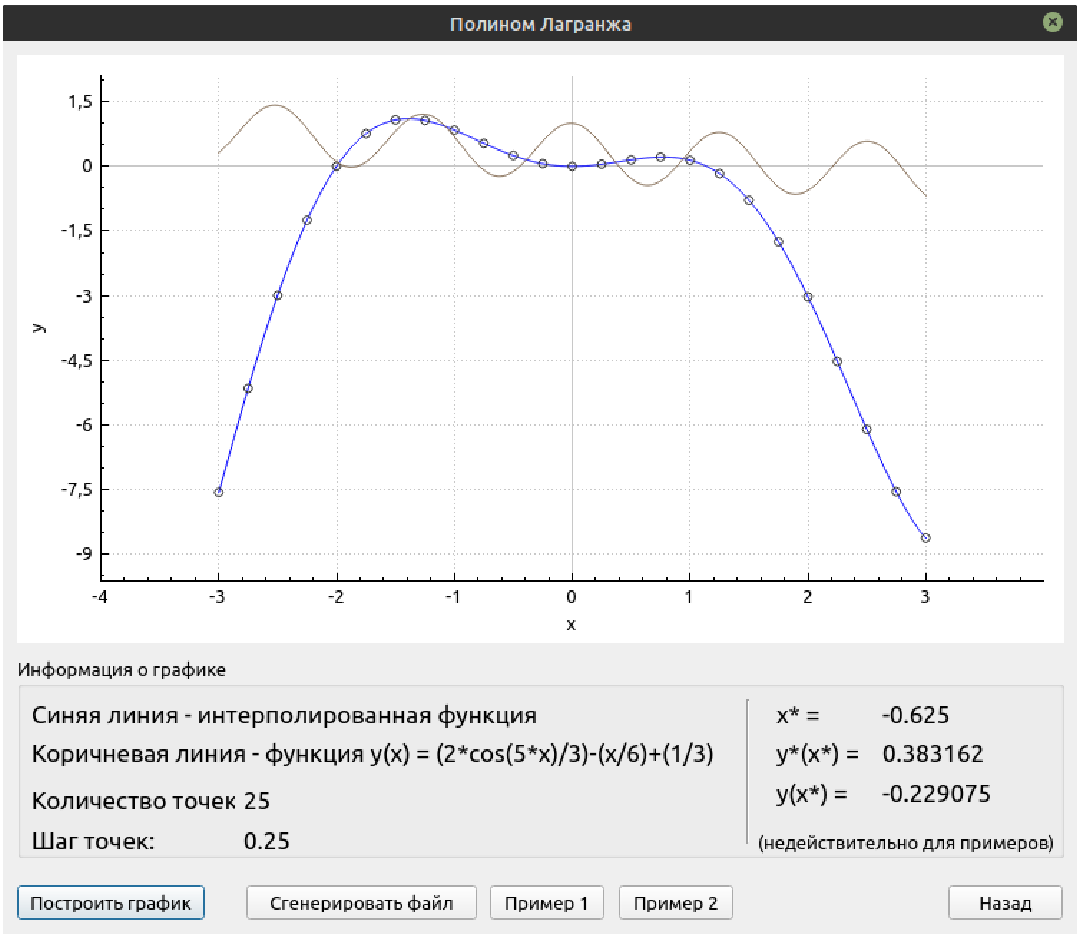
```
        for(int j = 0; j < c; j++){
            temp[j] = (temp[j+1]-temp[j])/(x[j+1+inter] - x[j]);
        }
        inter++;
        c--;
        for(int j = 0; j < count; j++){
            temp1[i] *= (X-x[j]);
        }
        count++;
        temp1[i] *= temp[0];
        result += temp1[i];

    }
    return result;
}
```

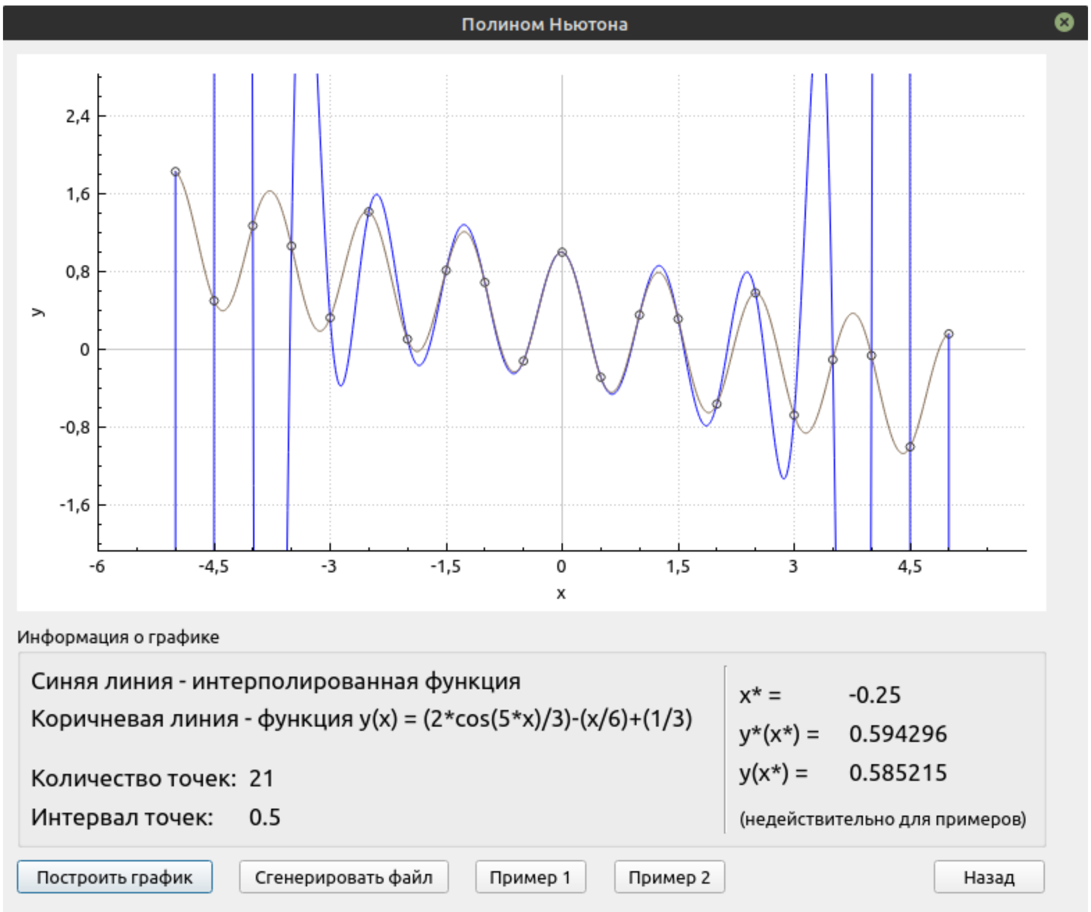
Результаты работы программы

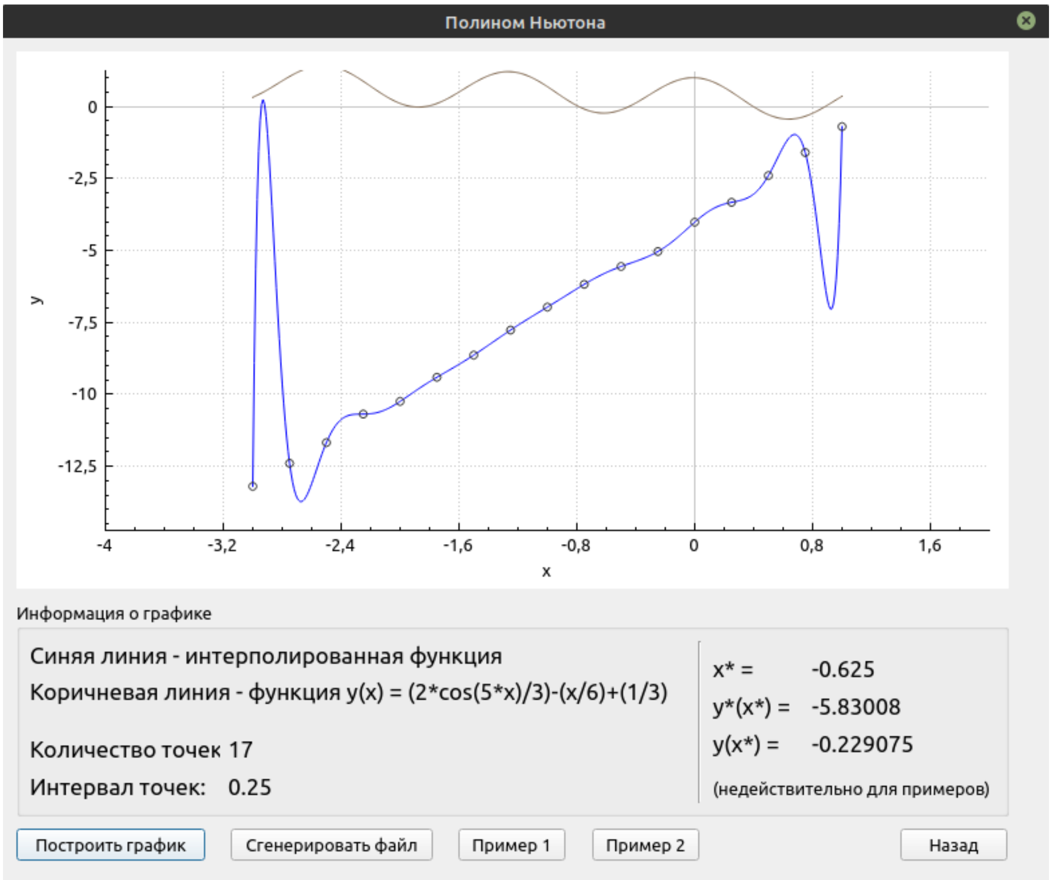


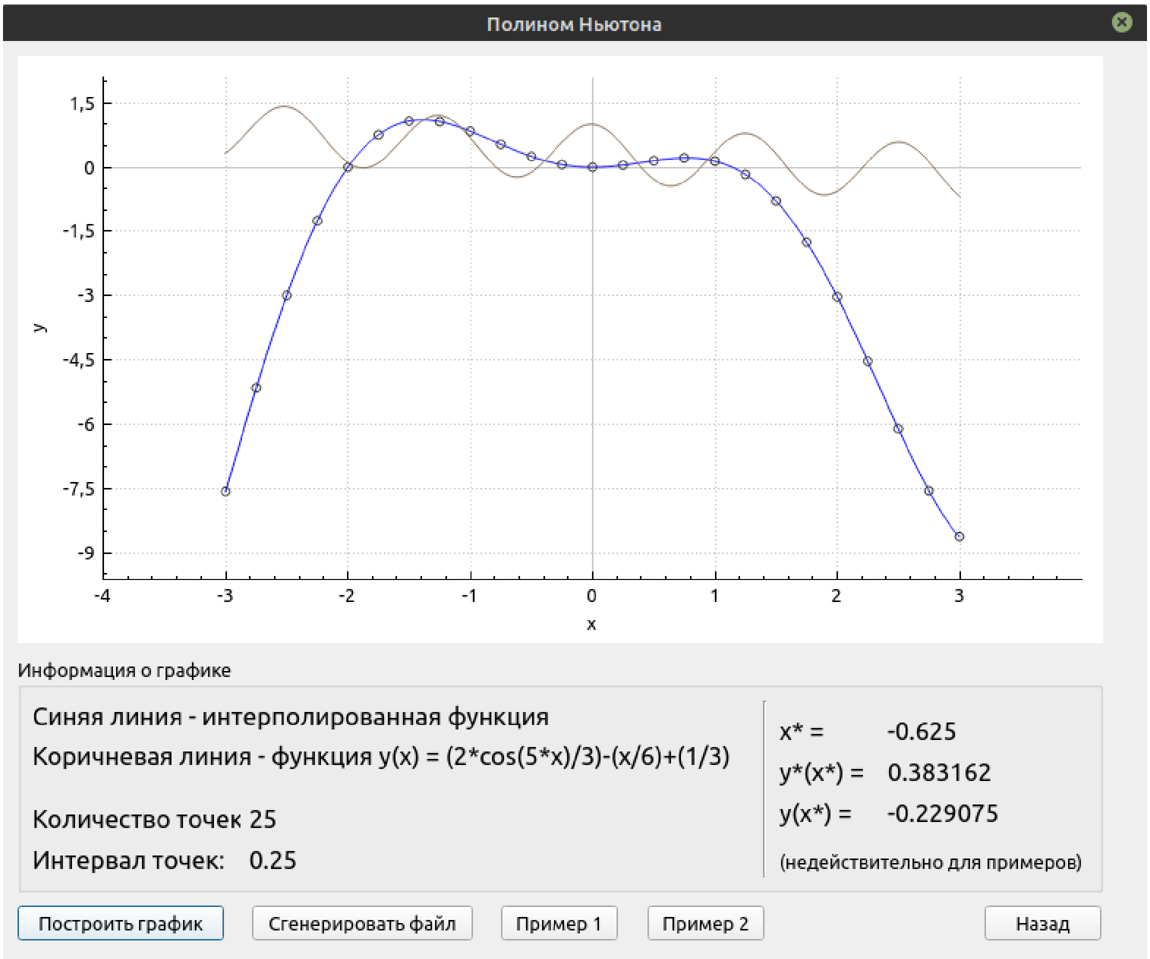

Полином Лагранжа
2,4
1,6
0,8
0
-0,8
-1,6
y
-6
-4,5
-3
-1,5
0
1,5
3
4,5
x
Информация о графике
Синяя линия - интерполированная функция
Коричневая линия - функция y(x) = (2*cos(5*x)/3)-(x/6)+(1/3)
Количество точек: 21
Шаг точек: 0.5
x* = -0.25
y*(x*) = 0.594296
y(x*) = 0.585215
(недействительно для примеров)
Построить график
Сгенерировать файл
Пример 1
Пример 2
Назад


Полином Лагранжа
0
-2,5
-5
-7,5
-10
-12,5
y
-4
-3,2
-2,4
-1,6
-0,8
0
0,8
1,6
x
Информация о графике
Синяя линия - интерполированная функция
Коричневая линия - функция y(x) = (2*cos(5*x)/3)-(x/6)+(1/3)
Количество точек 17
Шаг точек: 0.25
x* = -0.625
y*(x*) = -5.83008
y(x*) = -0.229075
(недействительно для примеров)
Построить график
Сгенерировать файл
Пример 1
Пример 2
Назад


Полином Лагранжа
1,5
0
-1,5
-3
-4,5
-6
-7,5
-9
y
-4
-3
-2
-1
0
1
2
3
x
Информация о графике
Синяя линия - интерполированная функция
Коричневая линия - функция y(x) = (2*cos(5*x)/3)-(x/6)+(1/3)
Количество точек 25
Шаг точек: 0.25
x* = -0.625
y*(x*) = 0.383162
y(x*) = -0.229075
(недействительно для примеров)
Построить график
Сгенерировать файл
Пример 1
Пример 2
Назад


Полином Ньютона
2,4
1,6
0,8
0
-0,8
-1,6
y
-6
-4,5
-3
-1,5
0
1,5
3
4,5
x
Информация о графике
Синяя линия - интерполированная функция
Коричневая линия - функция y(x) = (2*cos(5*x)/3)-(x/6)+(1/3)
Количество точек: 21
Интервал точек: 0.5
x* = -0.25
y*(x*) = 0.594296
y(x*) = 0.585215
(недействительно для примеров)
Построить график
Сгенерировать файл
Пример 1
Пример 2
Назад


Полином Ньютона
y
x
Информация о графике
Синяя линия - интерполированная функция
Коричневая линия - функция y(x) = (2*cos(5*x)/3)-(x/6)+(1/3)
Количество точек 17
Интервал точек: 0.25
x* = -0.625
y*(x*) = -5.83008
y(x*) = -0.229075
(недействительно для примеров)
Построить график
Сгенерировать файл
Пример 1
Пример 2
Назад


Полином Ньютона
y
x
Информация о графике
Синяя линия - интерполированная функция
Коричневая линия - функция y(x) = (2*cos(5*x)/3)-(x/6)+(1/3)
Количество точек 25
Интервал точек: 0.25
x* = -0.625
y*(x*) = 0.383162
y(x*) = -0.229075
(недействительно для примеров)
Построить график
Сгенерировать файл
Пример 1
Пример 2
Назад

Вывод

В ходе данной лабораторной работы были изучены два метода приближения функций: метод наименьших квадратов для приближения линейных функций и метод интерполирования полиномами Ньютона и Лагранжа. На основе теоретических данных была написана программа по результатам работы которой были построены графики приближенных функций.