**Факультет ПИиКТ**

**Дисциплина: вычислительная математика**

Лабораторная работа №4

**“Интерполяция функции с помощью полинома Лагранджа”**

Выполнил: Тарасов Александр

Группа: Р3212

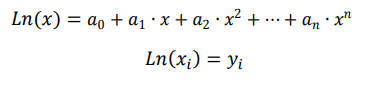
Преподаватель: Перл О.В.

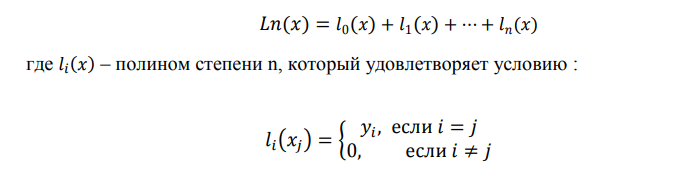
Вариант: Полином Лагранджа

Санкт-Петербург, 2020 г.

**Описание метода**

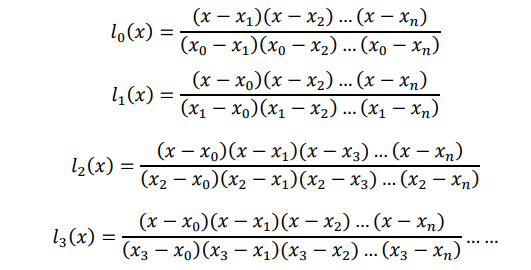
Пусть некоторая функция f(x) задана набором точек на некотором интервале [a,b].  
Задачей интерполяции является поиск функции F(x), принимающей в точках xi те же значения yi.  
Точки xi называют узлами интерполяции.

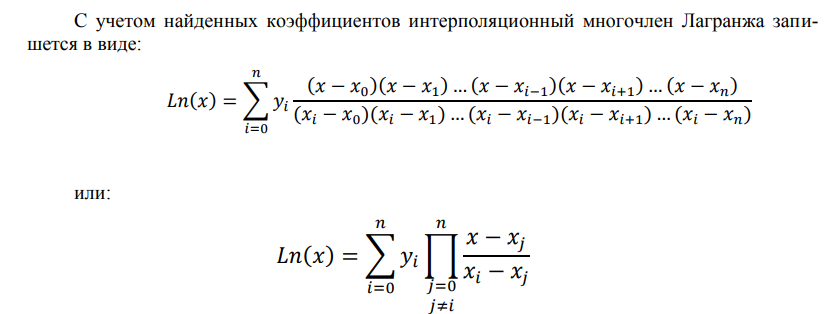
Существуют различные методы, позволяющие решить задачу интерполяции. Один из них – поиск интерполяционного многочлена Лагранжа.   
  
Суть метода заключается в построении так называемого интерполяционного многочлена Ln(x), который будет являться искомой функцией F(x).

Ln(x) имеет вид:

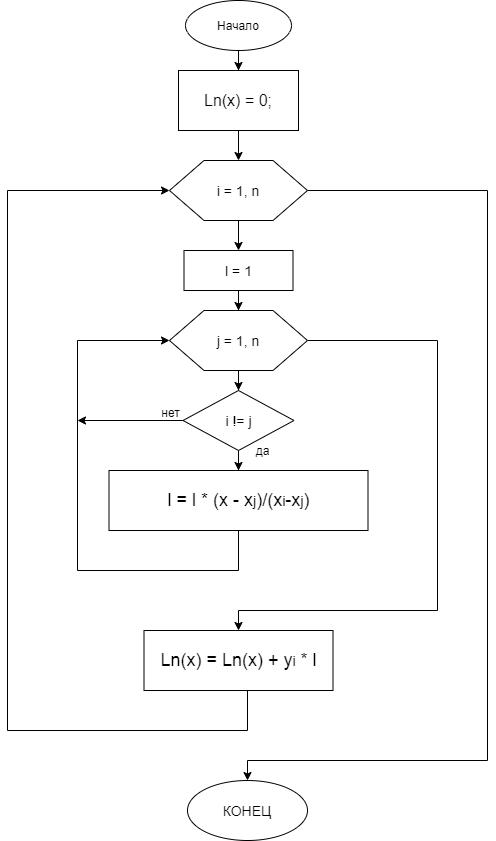
Полиномы *l(x)* составляются следующим образом:

После некоторых преобразований полиномы примут вид:



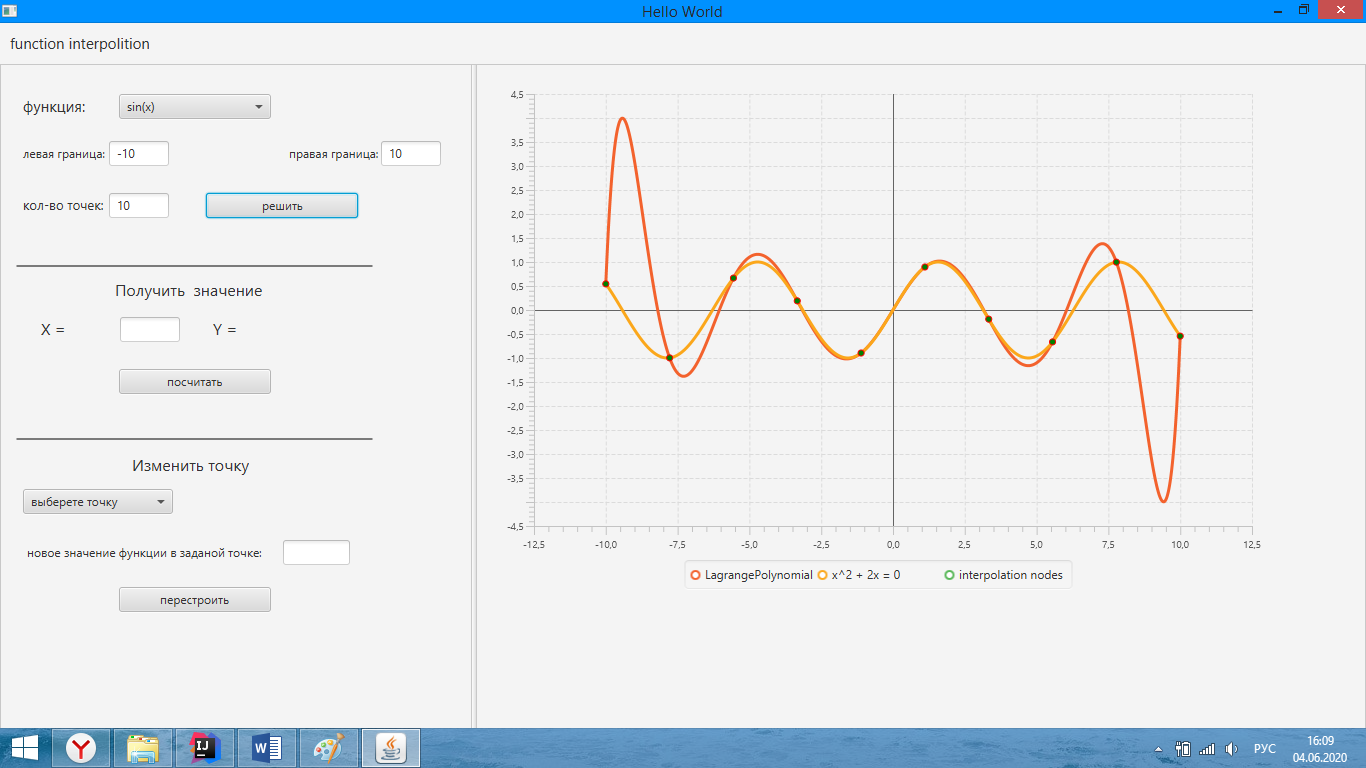
****

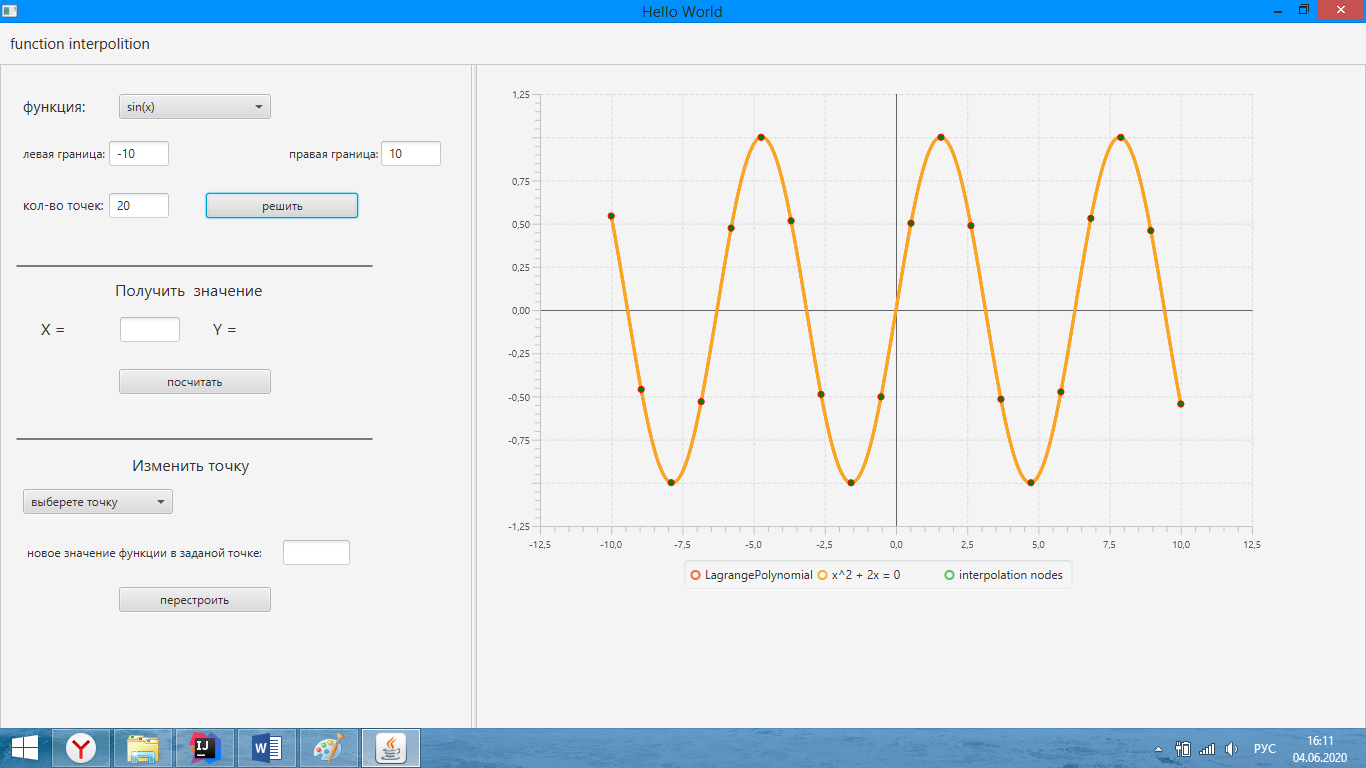
**Блок-схема численного метода**



**Примеры**

f(x) = sin(x);

Кол-во точек 10:

Кол-во точек 10:

**Листинг численного метода**

***LagrangePolynomial.java***

<…>

**public static void** initParam(Functions func, **double** leftBrdr, **double** rightBrdr, **int** dotsCnt){  
 *function* = func;  
 *leftBorder* = leftBrdr;  
 *rightBorder* = rightBrdr;  
 *dotsCount* = dotsCnt;  
 *xValues* = **new** ArrayList<>();  
 *yValues* = **new** ArrayList<>();  
 *prepareData*();  
 }  
  
  
 **public static void** prepareData(){  
 **double** step = Math.*abs*(*rightBorder* - *leftBorder*) / (*dotsCount*-1);  
 **double** j = *leftBorder*;  
  
 **for** (**int** i = 0; i < *dotsCount*; i++){  
 *xValues*.add(i, j);  
 *yValues*.add(i, *function*.solve(j, 0));  
 j += step;  
 }  
 }  
  
  
 **public static void** changeValue(**int** index, **double** yValue){  
 *yValues*.set(index, yValue);  
  
 }  
  
**public static double** interpolite (**double** x)  
 {  
 **double** lagrangePol = 0;  
 **for** (**int** i = 0; i < *dotsCount*; i++)  
 {  
 **double** basicsPol = 1;  
 **for** (**int** j = 0; j < *dotsCount*; j++)  
 {  
 **if** (j != i)  
 {  
 basicsPol \*= (x - *xValues*.get(j))/(*xValues*.get(i) - *xValues*.get(j));  
 }  
 }  
 lagrangePol += basicsPol \* *yValues*.get(i);  
 }  
  
 **return** lagrangePol;  
 }

<…>

***AppController.java***

<…>

@FXML  
**void** solveBtnPressed(ActionEvent event) {  
 **leftBorder** = Double.*parseDouble*(**leftBorderInput**.getText());  
 **rightBorder** = Double.*parseDouble*(**rightBorderInput**.getText());  
 **dotsCount** = Integer.*parseInt*(**dotsCountInput**.getText());  
  
 LagrangePolynomial.*initParam*(**currentFunction**, **leftBorder**, **rightBorder**, **dotsCount**);  
  
 initDotsBox();  
 drawGraph();  
 drawPoints();  
  
}  
  
@FXML  
**void** resolveBtnClicked(ActionEvent event) {  
 **YOut**.setText(String.*valueOf*(LagrangePolynomial.*interpolite*(Double.*parseDouble*(**XInput**.getText()))));  
}  
  
@FXML  
**void** changeValueBtnCliked(ActionEvent event) {  
 LagrangePolynomial.*changeValue*(Integer.*parseInt*(**dotNumberBox**.getValue())-1, Double.*parseDouble*(**changedYInput**.getText()));  
 drawGraph();  
 drawPoints();  
}  
  
**private void** drawPoints() {  
 XYChart.Series<Double, Double> series = **new** XYChart.Series<>();  
 **for** (**int** i = 0; i < **dotsCount**; i++){  
 XYChart.Data point = **new** XYChart.Data(LagrangePolynomial.*getxValues*().get(i), LagrangePolynomial.*getyValues*().get(i));  
 point.setNode(**new** Circle(3.0, Color.***GREEN***));  
 series.getData().add(point);  
 }  
 series.setName(**"interpolation nodes"**);  
 **chart**.getData().add(series);  
}  
  
**private void** drawGraph(){  
 **double** step = 0.02;  
 XYChart.Series<Double, Double> series = **new** XYChart.Series<>();  
 series.setName(**"LagrangePolynomial"**);  
 **for** (**double** x = **leftBorder**; x <= **rightBorder** + 0.00001; x = x + step) {  
 series.getData().add(**new** XYChart.Data<>(x, LagrangePolynomial.*interpolite*(x)));  
 }  
  
 XYChart.Series<Double, Double> series2 = **new** XYChart.Series<>();  
 series2.setName(**functionsList**.get(**currentFunctionIndx**));  
 **for** (**double** x = **leftBorder**; x <= **rightBorder** + 0.00001; x = x + step) {  
 series2.getData().add(**new** XYChart.Data<>(x, **currentFunction**.solve(x,0)));  
 }  
 **chart**.getData().setAll(series, series2);  
} <…>

**Вывод**

Основное различие аппроксимации и интерполяции заключается в условии прохождения полученной функции через заданные точки. В интерполяции в этих точках – узлах интерполяции, полученный полином должен принимать заданные значения. Для аппроксимации данное требование необязательно.   
  
Плюсы интерполяции полиномом Лагранжа:  
 + малая погрешность при n<20

+ узлы могут быть удалены друг от друга на различное расстояние

+ удобен, когда значения функций меняются, а узлы интерполяции неизменны

Минусы:

- при изменении числа узлов необходимо производить все вычисления заново.

Так же следует отметить метод Ньютона.  
 + позволяет не пересчитывать все коэффициенты заново при добавлении новой точки.

- работает только с таблицами с равноудаленными узлами.