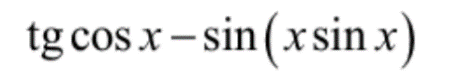
Практическая работа №4

по дисциплине «Вычислительная математика»

**«Интерполяция функций»**

студента группы M3207 Бойцова Виталия

f(x) =

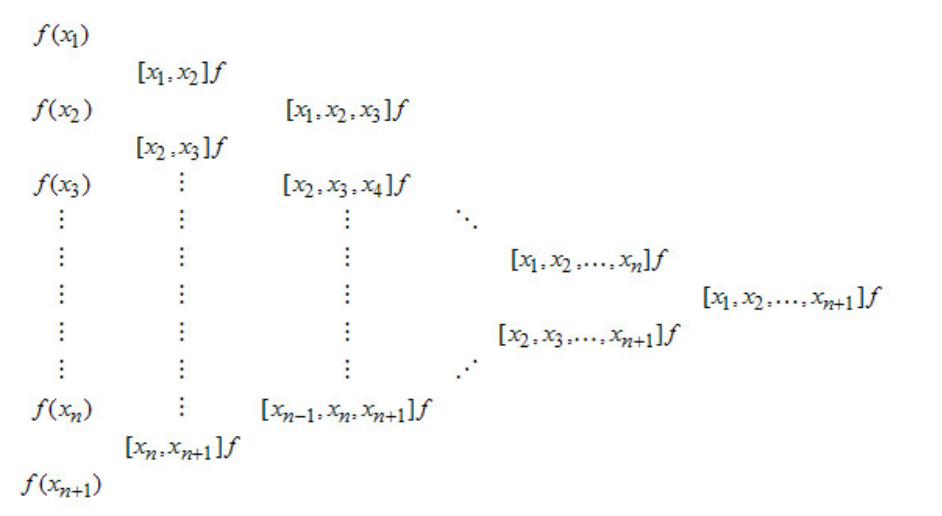
Равноотстоящие узлы:

Узлы Чебышева:



http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image155.pnghttp://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image154.png**Разделенные разности** нулевого порядка совпадают со значениями функции в узлах. Разделенные разности первого порядка определяются через разделенные разности нулевого порядка.

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image157.pngРазделенные разности второго порядка определяются через разделенные разности первого порядка.

Разделенные разности   
*k*-го порядка определяются через разделенные разности порядка k – 1.

http://aco.ifmo.ru/el_books/numerical_methods/lectures/images/image159.pngИспользуя понятие разделенной разности интерполяционный **многочлен Ньютона** можно записать в следующем виде:

//Everything is possible...

//...with object-oriented code

#include <iostream>

#include <vector>

#include <string>

#include <cmath>

#include <functional>

#include <algorithm>

using namespace *std*;

const double EPS = 1e-8;

const double PI = *acos*(-1);

class Polynom {

public:

Polynom(function<double(double)> f, *vector*<double> nodes) : nodes(nodes) {

coeffs.*resize*(nodes.*size*() + 1);

*vector*<double> diffs(nodes.*size*());

for (int i = 0; i < diffs.*size*(); i++)

diffs[i] = f(nodes[i]);

coeffs[0] = diffs[0];

for (int k = 1; k < diffs.*size*(); k++) {

for (int i = 0; i <= diffs.*size*() - k - 1; i++)

diffs[i] = (diffs[i + 1] - diffs[i]) / (nodes[i + k] - nodes[i]);

coeffs[k] = diffs[0];

}

}

//возвращает значение функции в точке x

double operator () (double x) {

double ans = coeffs.*back*();

for (int i = coeffs.*size*() - 2; i >= 0; i--)

ans = coeffs[i] + ans \* (x - nodes[i]);

return ans;

}

private:

*vector*<double> nodes, coeffs;

};

int main() {

int n;

double a, b, h;

*cin* >> a >> b >> n >> h;

//a = -5; b = 5; n = 10; h = 0.1;

//находим узлы

*vector*<double> nodes1(n), nodes2(n);

for (int i = 0; i < n; i++) {

nodes1[i] = a + i \* (b - a) / (n - 1);

nodes2[i] = (a + b) / 2 + (b - a) / 2 \* *cos*(PI \* (2 \* i + 1) / (2 \* n));

}

//интерполируем функцию f

function<double(double)> f = [](double x) { return *tan*(*cos*(x)) - *sin*(x \* *sin*(x)); };

Polynom p1(f, nodes1);

Polynom p2(f, nodes2);

//выводим и находим погрешности

double d1 = 0, d2 = 0;

for (double x = a; x <= b + EPS; x += h) {

double fx = f(x), p1x = p1(x), p2x = p2(x);

*printf*("%lf %lf %lf\n", fx, p1x, p2x);

d1 = *max*(d1, *abs*(p1x - fx));

d2 = *max*(d2, *abs*(p2x - fx));

}

*cout* << d1 << ' ' << d2 << '\n';

}