Санкт-Петербургский государственный университет

Прикладная математика и информатика

Отчет по лабораторной работе №3.2 по вычислительному практикуму

**Нахождение производных таблично-заданной функции по формулам численного дифференцирования**

Выполнил:

Чернов Павел Олегович

группа 221

Санкт-Петербург

2022

1. **Введение**

Нахождение производных таблично-заданной функции по формулам численного дифференцирования

**Подготовительный этап:**

ВЫВЕСТИ НА ПЕЧАТЬ таблицу из (m+1) значения функции f в равноотстоящих с шагом h точках xi=a+i·h, где i=0,1,...,m.

Рассматривать функцию f(x)=e1,5\*k\*x, где k=((номер Вашего варианта по mod5) +1).

Здесь число значений в таблице m+1, a, h > 0—параметры задачи.

**Решение задачи численного дифференцирования:**

Для таблично-заданной функции f (смотри таблицу, созданную на подготовительном этапе), найти значение ее первой и второй производной в узлах xi таблицы. Для этого воспользоваться известными простейшими формулами численного дифференцирования, имеющими погрешность порядка 0(h2).

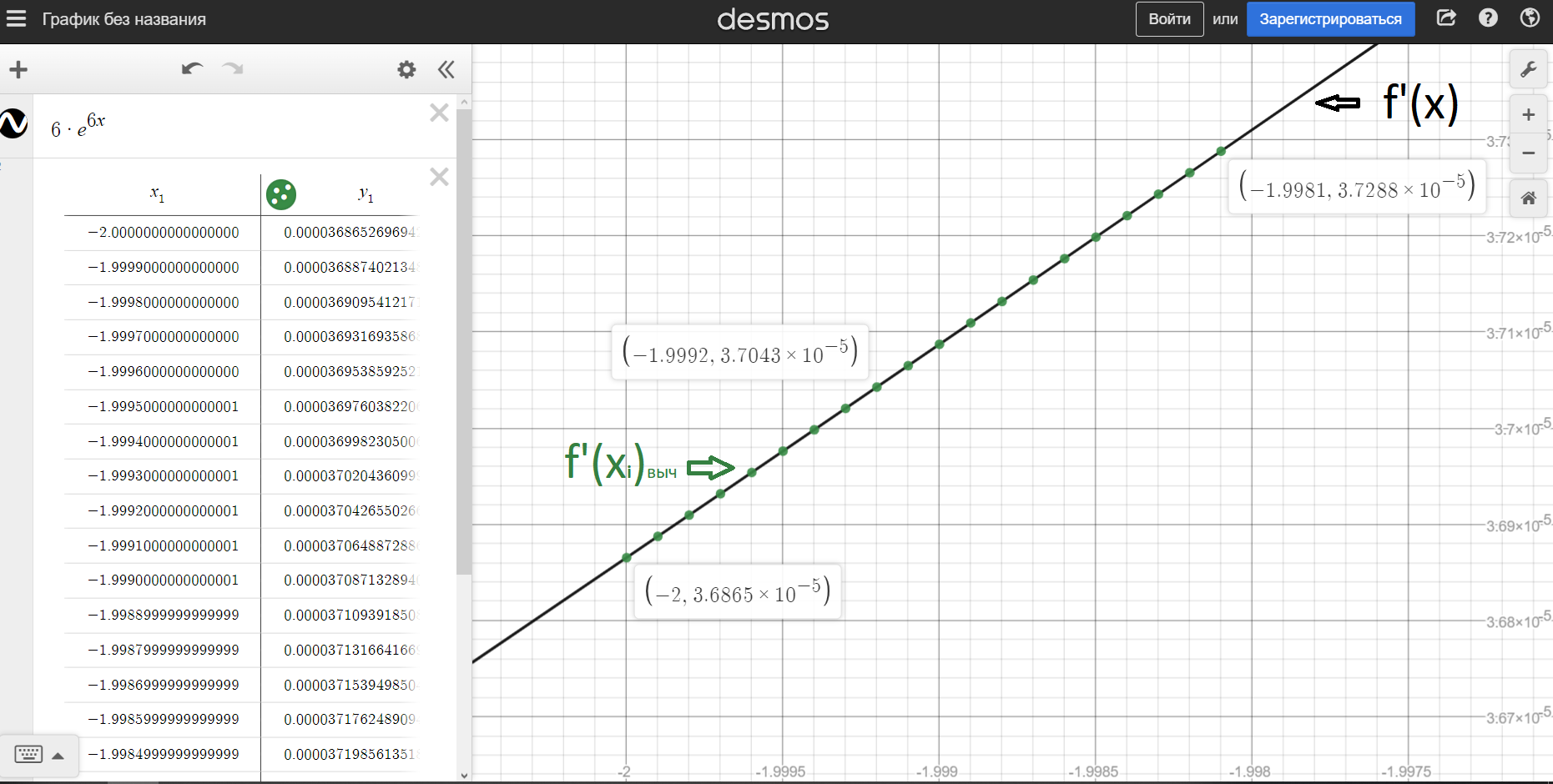
Вывести на печать таблицу вида:

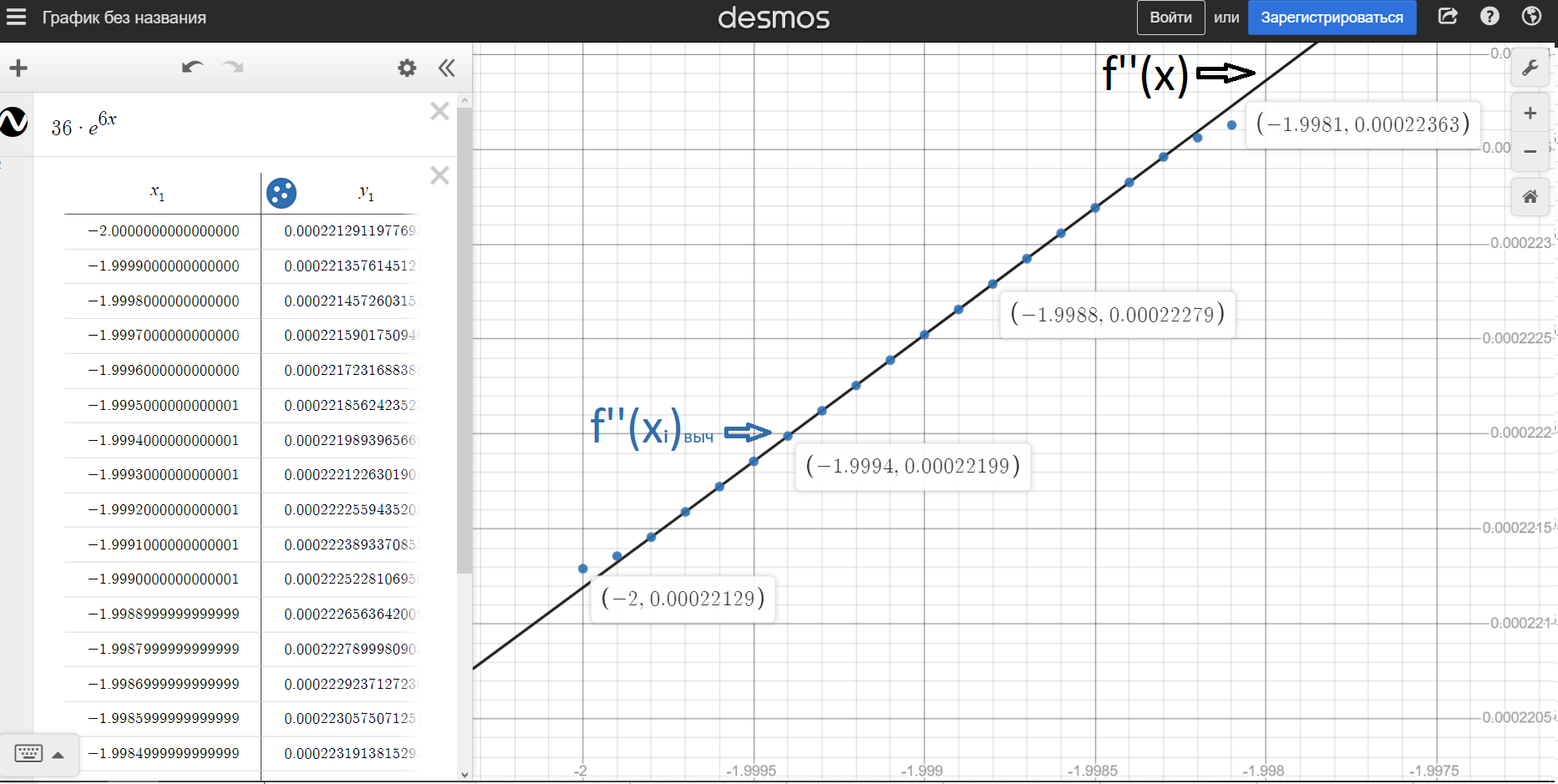
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| xi | f(xi) | f’(xi) | | f’(xi)T - f’(xi)ЧД| | f’’(xi) | | f’’(xi)T - f’’(xi)ЧД| |
|  |  |  |  |  |  |

Предусмотреть возможность ввода новых значений параметров, создание новой таблицы значений функции и расчета ее производных.

1. **Ход работы**
2. Представленная ниже программа позволяет аппроксимировано вычислять значения первой и второй производной в табличных точках таблично заданной функции.
3. Для приближенного вычисления производных используются формулы численного дифференцирования, имеющие погрешность порядка 0(h2).
4. Исходным параметром задачи является функция . В программе можно задать а) число табличных значений m, по которым будет проводиться приближение, б) начальную точку a, от которой с заданным шагом будут вычислены m табличных значений, в) h – шаг аргумента в таблице. Ввод осуществляется при помощи клавиатуры.
5. Программа написана на языке С++, в расчетах используются переменные типа double.
6. **Приложения**

А) Графики





Б) Программа

#include <iostream>

#include <Windows.h>

#include <vector>

#include <iomanip>

double f(double x) {

return exp(6 \* x);

}

double df(const std::vector<std::pair<double, double>> &table, int i, double h) {

if (!i) {

return (-3 \* table[i].second + 4 \* table[i + 1].second - table[i + 2].second) / (2 \* h);

}

else if (i == table.size() - 1) {

return (3 \* table[i].second - 4 \* table[i - 1].second + table[i - 2].second) / (2 \* h);

}

else {

return (table[i + 1].second - table[i - 1].second) / (2 \* h);

}

}

double actual\_df(const std::vector<std::pair<double, double>>& table, int i) {

return 6 \* table[i].second;

}

double actual\_d2f(const std::vector<std::pair<double, double>>& table, int i) {

return 36 \* table[i].second;

}

int main() {

SetConsoleTitleA("Algebraic reverse interpolation problem");

std::cout << std::fixed << std::setprecision(16);

bool flag;

do {

unsigned int m; //Кол-во точек в таблице (m = 10)

std::cout << "Input number of table values: ";

std::cin >> m;

std::cout << "Input starting point: ";

double a;

std::cin >> a;

std::cout << "Input length of step: ";

double h;

std::cin >> h;

std::vector <std::pair<double, double>> table\_values(m);

std::cout << "Table of values: \n" << std::setw(15) << "X\_i" << std::setw(17) << "f(X\_i)\n";

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

table\_values[i].first = a + i \* h;

table\_values[i].second = f(table\_values[i].first);

std::cout << table\_values[i].first << "|" << table\_values[i].second << "\n";

}

std::cout << "\n";

/\*std::cout << "\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\n";\*/

std::vector <std::pair<double, double>> df\_table(m);

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

df\_table[i].first = table\_values[i].first;

df\_table[i].second = df(table\_values, i, h);

/\*std::cout << df\_table[i].first << "|" << df\_table[i].second << "\n";\*/

}

/\*std::cout << "\n\n";\*/

std::vector <double> d2f\_table(m);

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

d2f\_table[i] = df(df\_table, i, h);

/\*std::cout << df\_table[i].first << "|" << d2f\_table[i] << "\n";\*/

}

/\*std::cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";\*/

std::cout << std::setw(15) << "X\_i" << std::setw(19) << "f(X\_i)" << std::setw(24) << "df(X\_i)" <<

std::setw(42) << "|df(X\_i) - act\_df(X\_i)|" << std::setw(16) << "d2f(X\_i)" << std::setw(30) << "|d2f(X\_i) - act\_d2f(X\_i)|" << "\n";

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

double actual\_dif\_f = actual\_df(table\_values, i), actual\_dif2\_f = actual\_d2f(table\_values, i);

std::cout << table\_values[i].first << "|" << std::setw(26) << table\_values[i].second << "|" << std::setw(26) << df\_table[i].second << "|" << std::setw(23) <<

abs(df\_table[i].second - actual\_dif\_f) << "|" << std::setw(20) << d2f\_table[i] << "|" << std::setw(20) << abs(d2f\_table[i] - actual\_dif2\_f) << "\n";

}

std::cout << "If you want to try another input data, enter 1, otherwise 0: ";

std::cin >> flag;

} while (flag);

return 0;

}

https://github.com/Chernovuk/Chislaki/tree/master/Third\_task/3.2