# Лабораторная работа 1

# Методы Вычислений

* В данной лабораторной работе было 2 способа выбора узлов:

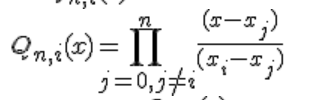
1. Равномерные узлы.

mult = a + i \* h;

1. Узлы Чебышева.

mult = (a + b) / 2.0 + (b - a) / 2.0 \* cos((2.0 \* i + 1.0) / (2.0\* (n + 1))\* acos(-1.0));

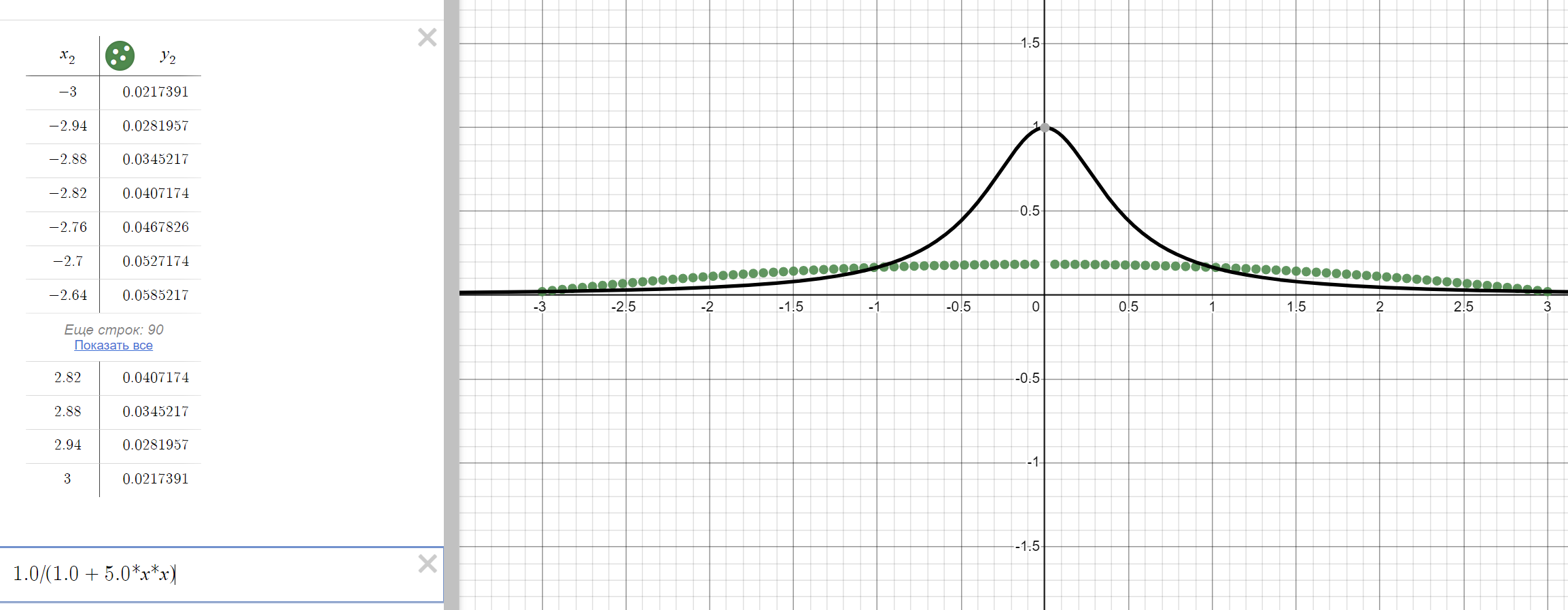
* В данной программе для построения интерполяционных многочленов использовался многочлен Ньютона. Многочлен Ньютона представляет собой сумму членов, каждый из которых представляет произведение конечных разностей функции.



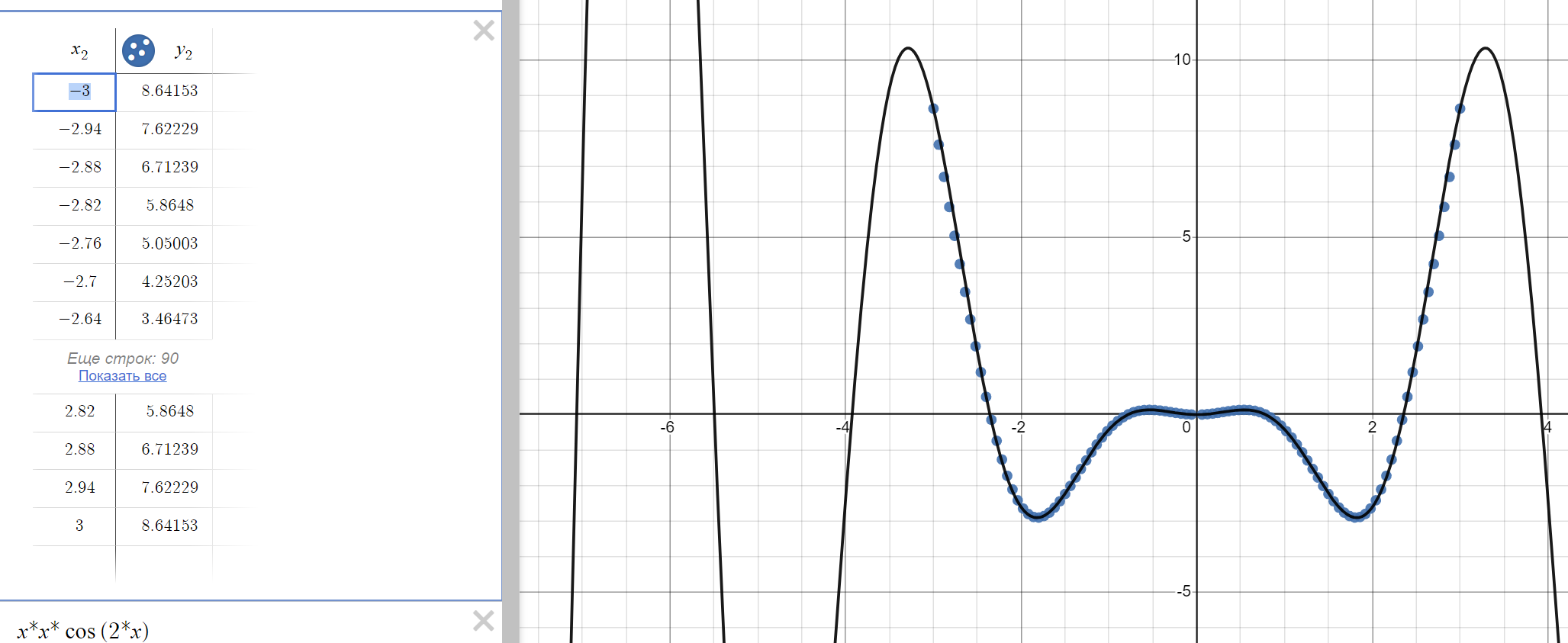
* Графики функций:  
  При n = 3;

Первой функции.

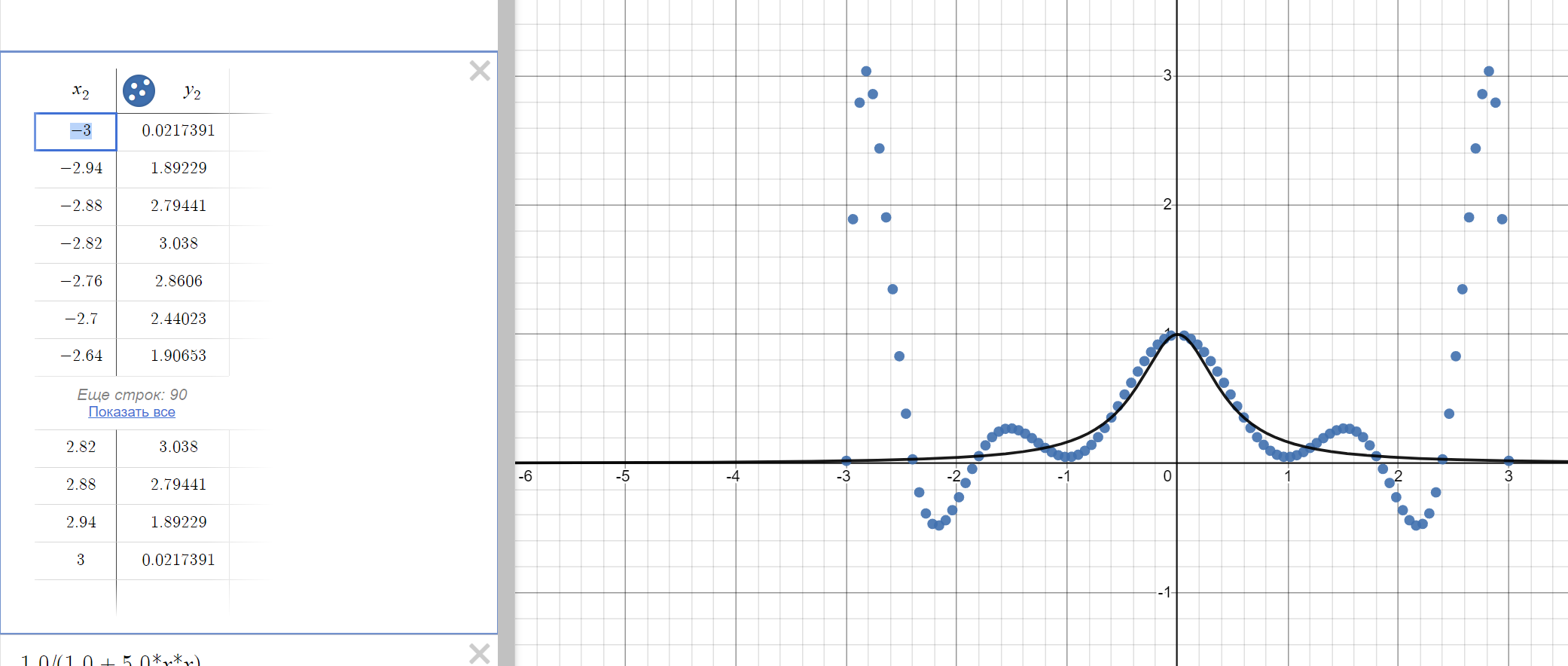


Второй функции:  


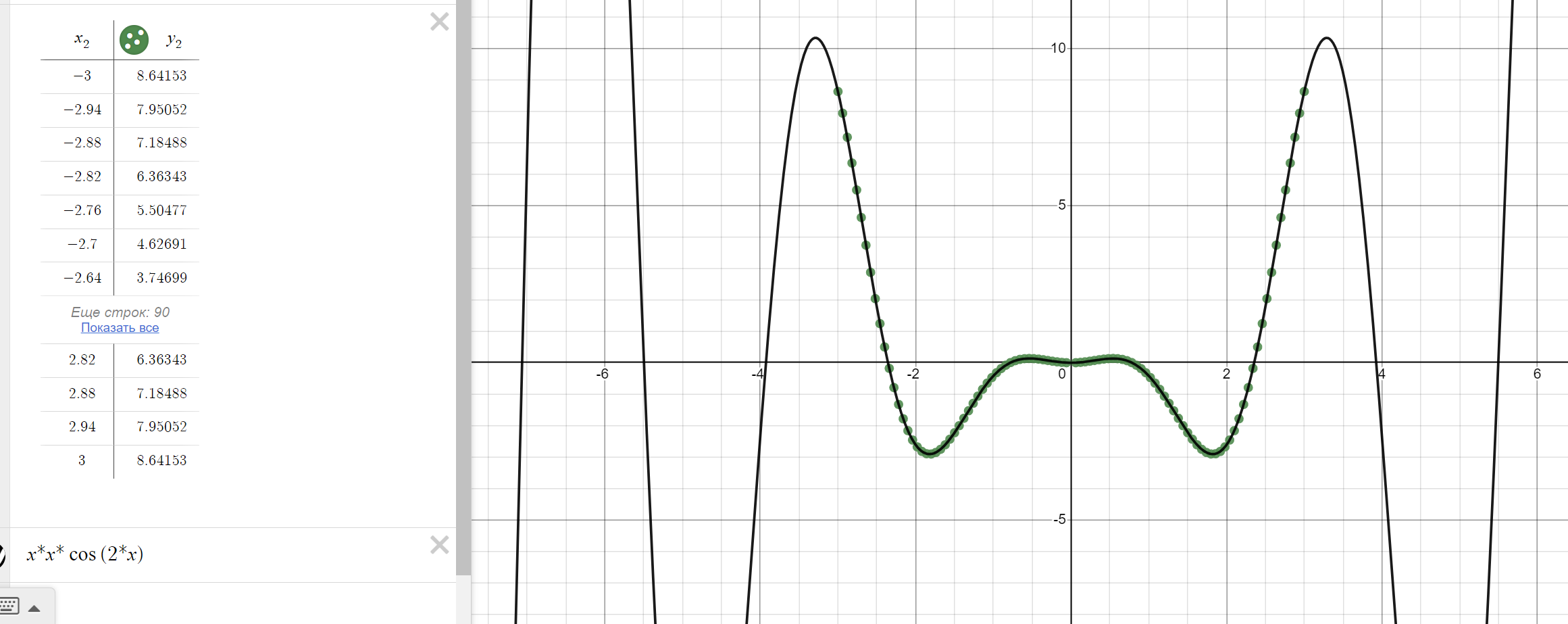
При n = 10;  
Первый график:



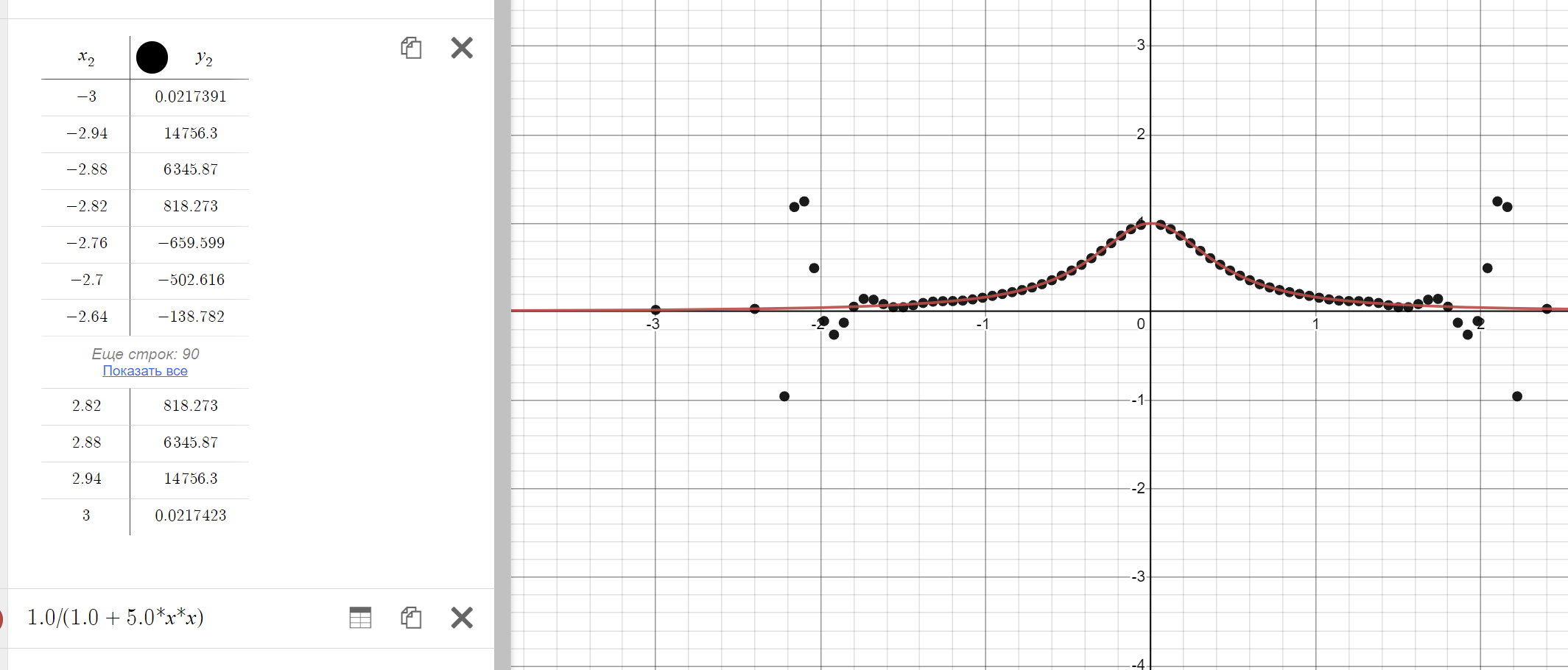
Второй:



При n = 30;



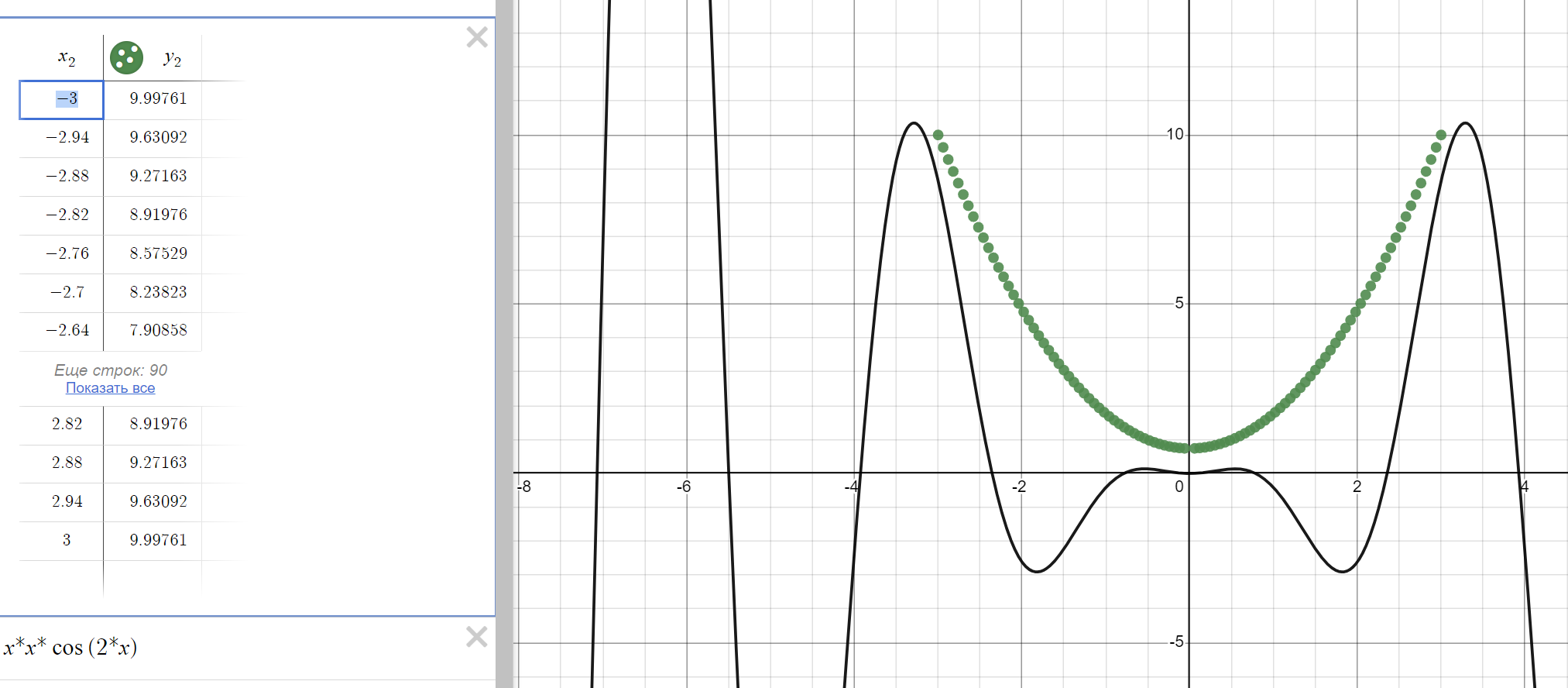
Второй:



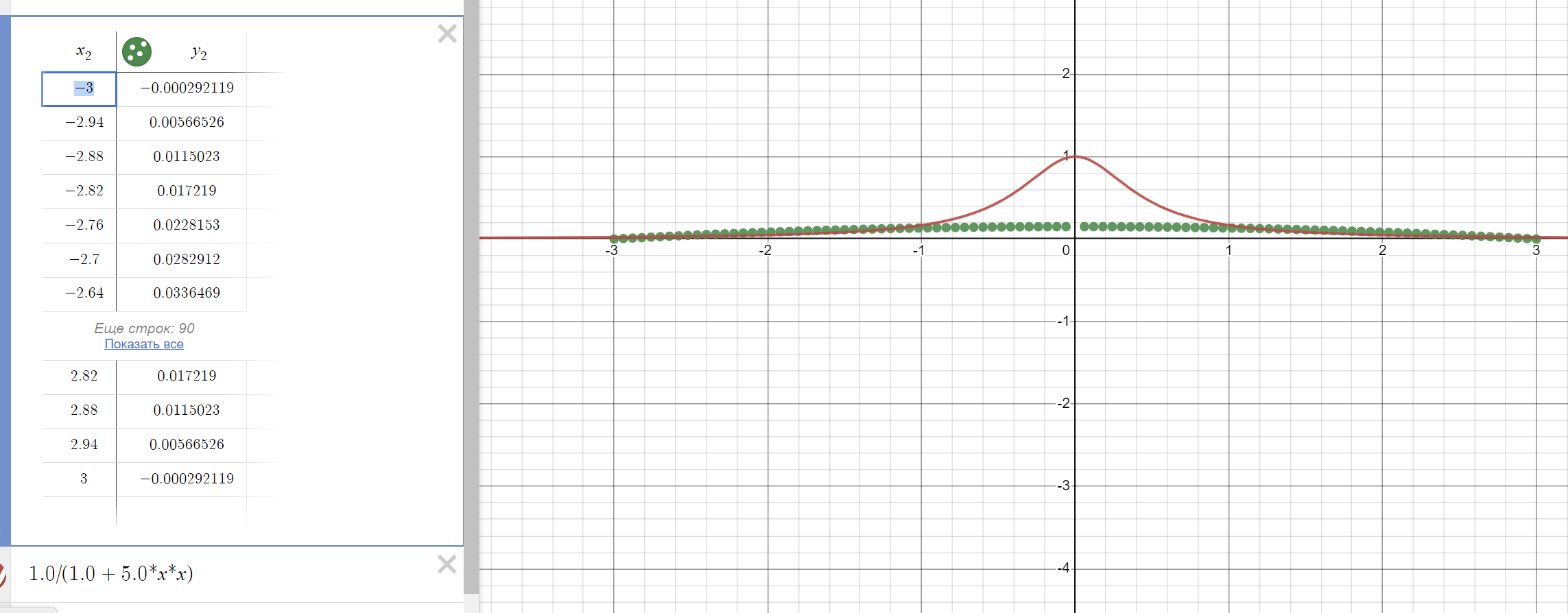
Перейдём к Узлам Чебышева:

При n = 3

Первая формула:

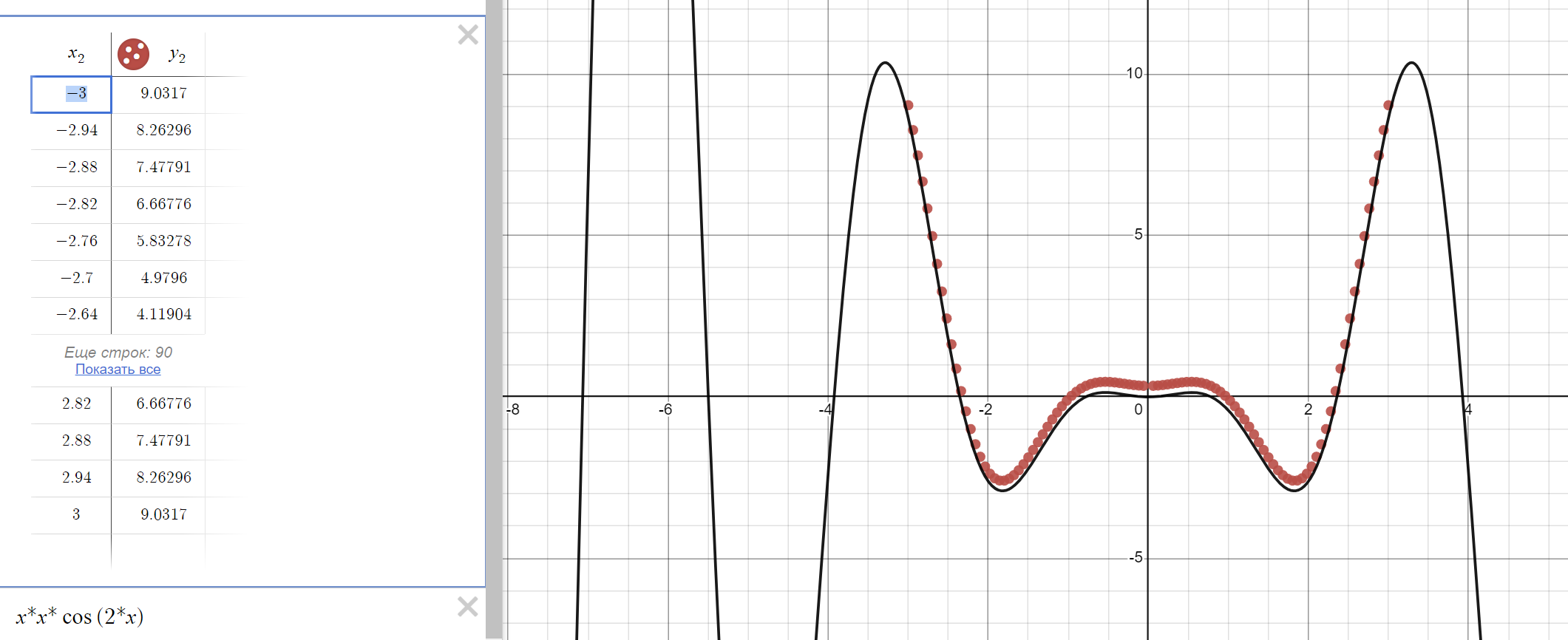
ъ

Вторая формула:

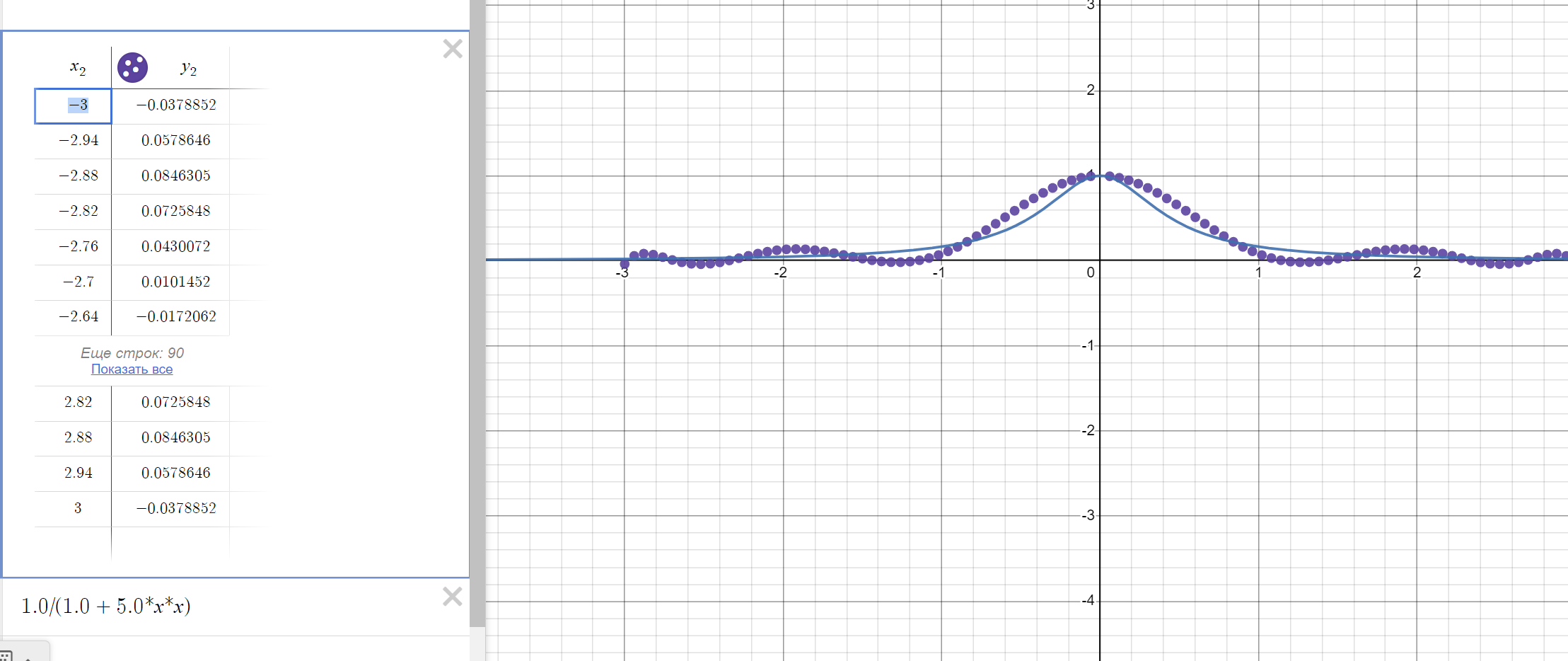


При n = 10

Первая формула:

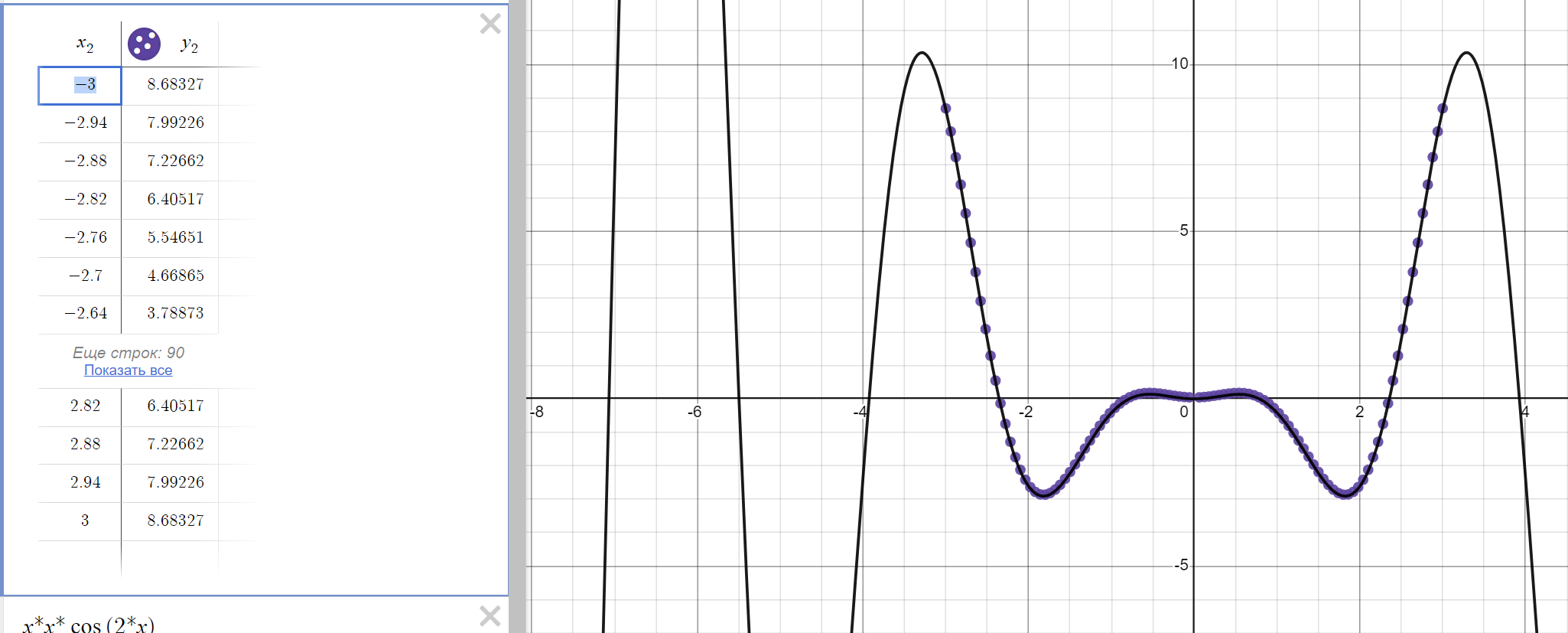


Вторая формула:

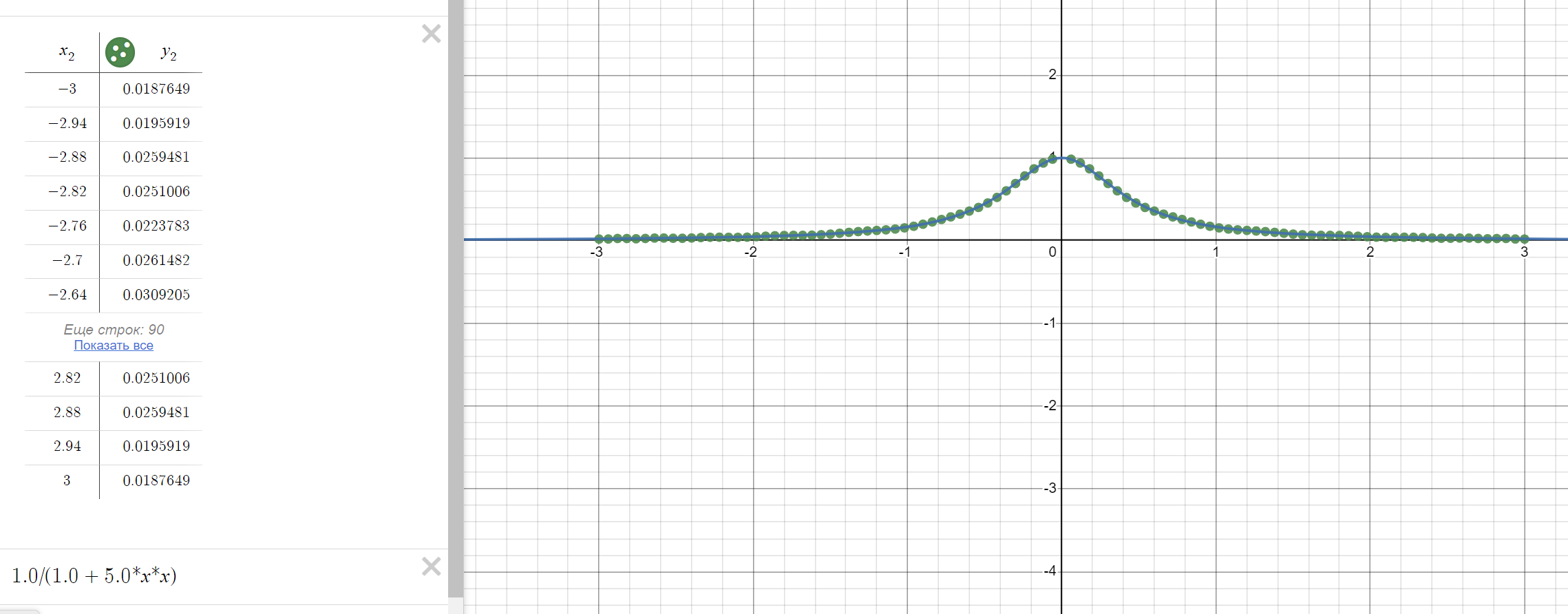


При n = 30

Первая формула:



Вторая формула:



Максимальная погрешность вычислений равномерными и Чебышевскими узлами:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Значение n | Равномерный узел 1 формулы | Равномерный узел 2 формулы | Чебышевский узел 1 формулы | Чебышевский узел 2 формулы |
| 1 | 11.547 | 0.978261 | 0.978261 | 0.978261 |
| 2 | 6.45566 | 0.726847 | 10.0399 | 0.682041 |
| 3 | 5.62326 | 0.815217 | 7.47932 | 0.849853 |
| 4 | 1.23299 | 0.487612 | 3.43375 | 0.514058 |
| 5 | 1.86525 | 0.592319 | 2.57813 | 0.703745 |
| 6 | 3.73371 | 0.754604 | 1.99506 | 0.381109 |
| 7 | 2.14233 | 0.400334 | 1.73728 | 0.558576 |
| 8 | 1.98989 | 1.44107 | 0.855154 | 0.278633 |
| 9 | 1.29775 | 0.320074 | 0.667457 | 0.431336 |
| 10 | 0.498631 | 3.01346 | 0.390166 | 0.202018 |
| 11 | 0.337336 | 0.689681 | 0.316447 | 0.327608 |
| 12 | 0.0772895 | 6.50667 | 0.245599 | 0.14441 |
| 13 | 0.0541476 | 1.52808 | 0.210512 | 0.246447 |
| 14 | 0.0083876 | 14.9399 | 0.18101 | 0.103381 |
| 15 | 0.00587351 | 3.48266 | 0.158675 | 0.184387 |
| 16 | 0.000652326 | 33.6325 | 0.140195 | 0.0737279 |
| 17 | 0.00045559 | 7.77805 | 0.124858 | 0.137537 |
| 18 | 4.02942e-05 | 75.5895 | 0.111912 | 0.0533538 |
| 19 | 2.91169e-05 | 18.0148 | 0.100889 | 0.102419 |
| 20 | 2.07767e-06 | 185.248 | 0.0914215 | 0.0411912 |
| 21 | 1.49985e-06 | 43.9713 | 0.0832303 | 0.0761984 |
| 22 | 8.65151e-08 | 450.322 | 0.0760949 | 0.0315022 |
| 23 | 6.23741e-08 | 106.508 | 0.0698412 | 0.0566633 |
| 24 | 2.9703e-09 | 1086.97 | 0.0643293 | 0.0236511 |
| 25 | 2.13804e-09 | 256.263 | 0.0594462 | 0.0421262 |
| 26 | 8.46567e-11 | 2607.28 | 0.0550997 | 0.0176211 |
| 27 | 6.10028e-11 | 612.918 | 0.0512137 | 0.0313156 |
| 28 | 1.00169e-11 | 6218.74 | 0.0477254 | 0.0134839 |
| 29 | 1.28306e-11 | 1458.06 | 0.0445823 | 0.0232788 |
| 30 | 4.53726e-11 | 14756.3 | 0.0417401 | 0.00968065 |

* Обычно узлы Чебышева приводят к более быстрой сходимости интерполяционного процесса по сравнению с равномерно распределенными узлами. Это связано с тем, что узлы Чебышева минимизируют максимальное значение абсолютной величины многочлена Чебышева, который может использоваться для оценки ошибки интерполяции.

но в моём случае получилось как-то не так(

* Листенинг программы:

Программа проводит интерполяцию для функций f1 и f2 с использованием равноотстоящих и узлов Чебышёва, а затем выводит результаты.

Определение функций:

double f1(double x) {

return x \* x \* cos(2 \* x);

}

double f2(double x) {

return 1.0 / (1.0 + 5.0 \* x \* x);

}

Вычисление конечных разностей:

double calculateFiniteDifference(double x0, double x1, double f\_x0, double f\_x1) {

return (f\_x1 - f\_x0) / (x1 - x0);

}

Заполнение вектора конечных разностей:

void vecFiller(vector<vector<double>>& vec, int n, double a, double h, double (\*f)(double)) {

/for (int i = 0; i < n + 1; i++)

{

double mult = a + i \* h;

vec[i][0] = mult;

vec[i][1] = f(mult);

}

for (int i = 1; i < n + 2; i++)

{

for (int j = 0; j < n - i + 1; j++)

{

double finite\_difference = calculateFiniteDifference(vec[j][0], vec[i + j][0], vec[j][i], vec[j + 1][i]);

vec[j][i + 1] = finite\_difference;

}

}

}

Такая же функция есть для узлов Чебышева .

Интерполяционные полиномы для равноотстоящих узлов:

double calculateInterpolationPolynomial(double x, double x0, double b, double h, double f\_x0, int n, double (\*f)(double), vector<vector<double>> vec) {

double result = f(x0);

double term = 1.0;

double mult = 0;

double test = 0;

for (int i = 1; i <= n; ++i) {

mult = x - (((x0 + b) / 2.0 + (b - x0) / 2.0 \* cos(acos(-1.0) \* (2.0 \* (i - 1) + 1.0) / (2.0 \* (n + 1)))));

term \*= mult;

test = vec[0][i + 1];

result += test \* term;

}

return result;

}

Интерполяционные полиномы для узлов Чебышёва:

double calculateInterpolationPolynomial2(double x, double x0, double b, double h, double f\_x0, int n, double (\*f)(double), vector<vector<double>> vec){

double result = f(x0);

double term = 1.0;

double mult = 0;

double test = 0;

for (int i = 1; i <= n; ++i) {

mult = x - (((x0 + b) / 2.0 + (b - x0) / 2.0 \* cos(acos(-1.0) \* (2.0 \* (i - 1) + 1.0) / (2.0 \* (n + 1)))));

term \*= mult;

test = vec[0][i + 1];

result += test \* term;

}

return result;

}

Основная функция main:

int main() {

// Инициализация параметров

// ...

// Заполнение и вычисление для f1

// ...

// Заполнение и вычисление для f2

// ...

// Вывод результатов

// ...

return 0;

}