**Токаева Александра, 409 группа**

**Отчет по практикуму на ЭВМ**

**Задание**: загадана некоторая функция f; дан отрезок [a, b],

даны n точек на этом отрезке и даны значения

загаданной функции в этих точках. Нужно по этим данным построить для загаданной функции на данном отрезке интерполяционный многочлен двумя способами:

1) в форме многочлена и найти его коэффициенты как решение СЛУ используя какой-нибудь точный метод решения системы (я буду использовать метод отражений).

2) в форме где – базисные функции, которые равны 1 в своем узле и равны 0 в остальных узлах.

Требуется построить этот многочлен по равноотстоящим узлам (то есть ) и по системе чебышевских узлов (то есть ).

Построить графики функции и ее многочлена для своей функции, функции Рунге и функции модуль.

**Решение:** сначала опишем интерфейс программы на си, затем способ вычисления , и наконец, суть метода отражений для вычисления .

Для запуска программы в текущей директории должен лежать makefile и файл mnog\_lagranzh.cpp. Запускаем в командной строке make, а потом строки наподобие ./a.out -1 1 10 ravnom myfunc

Инструкция по использованию:

Usage: ./a.out a b n ravnom/chebushov myfunc/runge/modul

Здесь a,b – концы отрезка, n-число узлов ( в случае равноотстоящих узлов оба конца отрезка являются первым и последним узлом соответственно, в случае чебышевских узлов – все n узлов лежат внутри отрезка, концы отрезка не являются узлами), ravnom/chebushov отвечает за вид узлов; вид загаданной функции регулируется функцией double func(double x).

Функция int generate\_input(double a, double b, int n, char\* tip\_uzlov, const char\* filename) записывает входные данные (узлы и значения функции func) в файл filename, потом функция main открывает этот файл, считывает массив точек в mas1, массив значений в mas2, вычисляет значения обоих многочленов и в исходных узлах и в добавочных узлах (по два добавочных узла между каждыми старыми узлами) и записывает их в файл типа output\_5ravnomrunge.txt в 7 столбиков в формате

x\_i, f(x\_i), P\_n\_1(x\_i), L\_n(x\_i), f(x\_i)- P\_n\_1(x\_i), ), f(x\_i)- L\_n(x\_i), P\_n\_1(x\_i)- L\_n(x\_i). Еще дополнительно в файл commands\_for\_gnuplot.txt дописывается что-то наподобие

set terminal png size 640,480

set output 'output\_10ravnommyfunc.txt.png'

plot 'output\_10ravnommyfunc.txt' using 1:2 with lines, 'output\_10ravnommyfunc.txt' using 1:3 with lines, 'output\_10ravnommyfunc.txt' using 1:4 with lines, 'output\_10ravnommyfunc.txt' using 1:2 with points

После того, как все нужные расчеты были выполнены, следует запустить в командной строке gnuplot, после приглашения написать

>gnuplot load ‘comands\_for\_gnuplot.txt’

И все графики сами нарисуются и сохранятся в файлы типа

'output\_10ravnommyfunc.txt.png'

Значения многочлена в любой точке x вычислить легко, имея mas1 и mas2 – просто честно посчитать значение каждой из n базисных функций в этой точке, умножить на и просуммировать по всем i. Этим и занимаются функции

double F\_i(int n, int i, double x, double\* mas1)

и

double L\_n(int n, double x, double\* mas1, double\* mas2).

Для вычисления значения многочлена в любой точке x нужно сначала вычислить его коэффициенты . Для этого надо составить и решить систему

Это делает функция

int algoritm(int n, double\* A, double\* X, double\* B).

Она приводит систему методом отражений к верхнетреугольному виду, а потом применяет обратный ход метода Гаусса. То есть она на k-м шаге для вектора y:= k-го стобца матрицы ( у которого сверху обрубили n-k элементов) по формуле строит единичный вектор x ( которому соответствует n\*n-матрица отражения U(x)= E-xx\*) такой, U(x)y = ||y||e\_k, то есть после применения матрицы U(x) к k-му столбцу получится на диагонали число ||y||, а ниже диагонали – нули. После этого последовательно применяем матрицу U(x) ко всем обрубкам столбцов с номерами k+1, …, n и еще к вектору B.

Матрица U(x) хороша по трем причинам:

1)она ортогональная

2)результат ее применения к вектору y считается не за n\*3 (как если бы мы честно перемножали две матрицы), а за 2n, за счет вида самой матрицы: U(x)y = (E-xx\*)y=y - x<x,y>.

3)Физический смысл применения матрицы U(x) к вектору y состоит в том, что мы вектор y отражаем относительно плоскости с нормалью x, поэтому используя этот факт и то, что U(x)y = ||y||e\_1 получаем, что y-||y||e\_1 параллелено вектору x, то есть y-||y||e\_1=x. Из условия единичной длины вектора x находим явную формулу для него:

Теперь построим графики функций y=x\*x + cos(5\*x\*x\*x - 2), y=1/(25x\*x+1) и y=|x| .

Сначала запускам в терминале make

Теперь у нас есть файл a.out, который можно запускать так

(см. Usage: ./a.out a b n ravnom/chebushov myfunc/runge/modul)

**1)Моя функция**

Это функция x\*x + cos(5\*x\*x\*x - 2);

Пусть мы хотим построить нашу функцию по n=10 узлам

**a)Пусть сначала узлы равномерные**

Запускаем в терминале ./a.out -1 1 10 ravnom mufunc

В текущей директории появился файл output\_10ravnommyfunc.txt содержащий 7 столбиков.

Посмотрим на них

-1.000000 1.753902 1.753902 1.753902 2.543433e-07 2.543433e-07 0.000000e+00

-0.925926 1.808437 1.808553 1.808553 -1.154482e-04 -1.154482e-04 1.731948e-14

-0.851852 1.095039 1.051079 1.051079 4.395908e-02 4.395908e-02 8.881784e-15

-0.777778 0.252803 0.252801 0.252801 2.340125e-06 2.340125e-06 -1.998401e-15

-0.703704 -0.329700 -0.301780 -0.301780 -2.791930e-02 -2.791930e-02 -5.717649e-15

-0.629630 -0.597907 -0.579334 -0.579334 -1.857295e-02 -1.857295e-02 -5.884182e-15

-0.555556 -0.651229 -0.651229 -0.651229 -1.952960e-07 -1.952960e-07 -3.330669e-15

-0.481481 -0.602717 -0.610767 -0.610767 8.050062e-03 8.050062e-03 2.220446e-16

-0.407407 -0.528223 -0.533615 -0.533615 5.392470e-03 5.392470e-03 2.442491e-15

-0.333333 -0.465348 -0.465348 -0.465348 1.124412e-07 1.124412e-07 2.275957e-15

-0.259259 -0.426481 -0.423979 -0.423979 -2.502059e-03 -2.502059e-03 9.436896e-16

-0.185185 -0.410512 -0.408823 -0.408823 -1.689325e-03 -1.689325e-03 -4.440892e-16

-0.111111 -0.410028 -0.410028 -0.410028 6.362791e-08 6.362791e-08 -1.221245e-15

-0.037037 -0.415006 -0.415586 -0.415586 5.798157e-04 5.798157e-04 -7.771561e-16

0.037037 -0.414544 -0.414656 -0.414656 1.114148e-04 1.114148e-04 4.440892e-16

0.111111 -0.397555 -0.397555 -0.397555 1.468245e-07 1.468245e-07 2.053913e-15

0.185185 -0.352775 -0.353801 -0.353801 1.025791e-03 1.025791e-03 3.275158e-15

0.259259 -0.268225 -0.270130 -0.270130 1.905034e-03 1.905034e-03 4.107825e-15

0.333333 -0.130493 -0.130492 -0.130492 -6.745488e-07 -6.745488e-07 4.024558e-15

0.407407 0.075012 0.080418 0.080418 -5.406110e-03 -5.406110e-03 3.400058e-15

0.481481 0.360360 0.369445 0.369445 -9.084774e-03 -9.084774e-03 2.498002e-15

0.555556 0.723817 0.723815 0.723815 2.218912e-06 2.218912e-06 2.220446e-15

0.629630 1.126779 1.099188 1.099188 2.759084e-02 2.759084e-02 2.442491e-15

0.703704 1.462195 1.411924 1.411924 5.027100e-02 5.027100e-02 2.664535e-15

0.777778 1.543438 1.543438 1.543438 -2.278358e-07 -2.278358e-07 1.110223e-15

0.851852 1.187482 1.367886 1.367886 -1.804034e-01 -1.804034e-01 -3.774758e-15

0.925926 0.469427 0.818218 0.818218 -3.487908e-01 -3.487908e-01 -7.105427e-15

1.000000 0.010008 0.010008 0.010008 -4.966004e-07 -4.966004e-07 -4.423545e-15

Мы видим, что 3 и 4 столбики абсолютно одинаковые, то есть значения интерполяционного многочлена, вычисленные двумя способами совпадают. Это говорит о том, что используемая в первом способе СЛУ получилась хорошо обусловленная и что мы ее хорошо решили (найдя тем самым коэффициенты интерполяционного многочлена), не накопив огромную погрешность. Далее, 2 и 3(4) столбики совпадают не во всех строках, а с шагом три: 1,4,7,10… Это связано с тем, что исходная функция и интерполяционный многочлен обязаны совпадать только в узлах, а между узлами мы добавили по две точки просто чтобы посмотреть, сильно ли многочлен от исходной функции отличается в точках, отличных от узлов.

Построим график по этим точкам, используя gnuplot.

Пишем в командной строке gnuplot, дальше в гнуплоте пишем

>gnuplot load 'comands\_for\_gnuplot.txt'

>gnuplot q

Изображение выглядит как карта

Автоматически созданное описание

Смотрим на график – видим, что функция хорошо приблизилась.

**б)Теперь узлы равномерные, функция по-прежнему моя**

Запускаем в терминале ./a.out -1 1 10 chebushov myfunc

В текущей директории появился файл output\_10chebushovmyfunc.txt

Посмотрим на него

0.987688 0.027561 0.027560 0.027560 6.362131e-07 6.362131e-07 -5.967449e-15

0.955461 0.202248 0.228412 0.228412 -2.616358e-02 -2.616358e-02 -5.800915e-15

0.923234 0.496489 0.520464 0.520464 -2.397467e-02 -2.397467e-02 -9.103829e-15

0.891007 0.827860 0.827865 0.827865 -4.947077e-06 -4.947077e-06 -6.106227e-15

0.829707 1.343947 1.297512 1.297512 4.643415e-02 4.643415e-02 -9.992007e-15

0.768407 1.554612 1.512266 1.512266 4.234610e-02 4.234610e-02 -8.215650e-15

0.707107 1.473156 1.473155 1.473155 5.704527e-07 5.704527e-07 -7.993606e-15

0.622735 1.089850 1.136352 1.136352 -4.650197e-02 -4.650197e-02 -6.661338e-15

0.538362 0.633647 0.671896 0.671896 -3.824865e-02 -3.824865e-02 -5.995204e-15

0.453990 0.244746 0.244748 0.244748 -1.970864e-06 -1.970864e-06 -6.161738e-15

0.354805 -0.078541 -0.108022 -0.108022 2.948126e-02 2.948126e-02 -7.230327e-15

0.255619 -0.273506 -0.296192 -0.296192 2.268622e-02 2.268622e-02 -8.104628e-15

0.156434 -0.374195 -0.374195 -0.374195 -2.569574e-07 -2.569574e-07 -7.549517e-15

0.052145 -0.412783 -0.399821 -0.399821 -1.296210e-02 -1.296210e-02 -5.939693e-15

-0.052145 -0.414072 -0.405558 -0.405558 -8.514373e-03 -8.514373e-03 -3.164136e-15

-0.156434 -0.409003 -0.409003 -0.409003 2.380963e-07 2.380963e-07 -4.996004e-16

-0.255619 -0.425205 -0.426650 -0.426650 1.444903e-03 1.444903e-03 -5.551115e-17

-0.354805 -0.481311 -0.480483 -0.480483 -8.286548e-04 -8.286548e-04 -1.332268e-15

-0.453990 -0.575387 -0.575387 -0.575387 3.744529e-07 3.744529e-07 -3.330669e-15

-0.538362 -0.645564 -0.646364 -0.646364 8.001859e-04 8.001859e-04 -4.996004e-15

-0.622735 -0.610032 -0.607355 -0.607355 -2.677385e-03 -2.677385e-03 -3.885781e-15

-0.707107 -0.310274 -0.310275 -0.310275 7.900664e-07 7.900664e-07 3.885781e-16

-0.768407 0.161019 0.144897 0.144897 1.612198e-02 1.612198e-02 4.996004e-15

-0.829707 0.831441 0.800496 0.800496 3.094501e-02 3.094501e-02 1.132427e-14

-0.891007 1.528057 1.528053 1.528053 4.233463e-06 4.233463e-06 1.398881e-14

-0.923234 1.792233 1.832626 1.832626 -4.039310e-02 -4.039310e-02 1.665335e-14

-0.955461 1.909862 1.976825 1.976825 -6.696267e-02 -6.696267e-02 9.103829e-15

-0.987688 1.836102 1.836101 1.836101 1.401688e-06 1.401688e-06 9.992007e-15

Опять 3 и 4 столбики полностью совпадают, а 2 и 3(4) совпадают с шагом три. Строим график в gnuplot, читая данные из файла ‘output\_10chebushovmyfunc.txt’

и сохраняя полученный график в файл ‘output\_10chebushovmyfunc.txt.png’.

Изображение выглядит как карта

Автоматически созданное описание

Смотрим на график – тоже хорошо приблизилась.

**2)Функция Рунге**

Это функция 1/(25\*x\*x + 1);

Пусть мы хотим построить нашу функцию по n=12 узлам

**a)Пусть сначала узлы равномерные**

Запускаем в терминале ./a.out -1 1 12 ravnom runge

В текущей директории появился файл output\_12ravnomrunge.txt

Посмотрим на него

-1.000000 0.038462 0.038462 0.038462 -4.615384e-07 -4.615385e-07 -2.808864e-14

-0.939394 0.043362 0.597214 0.597214 -5.538512e-01 -5.538512e-01 -5.169198e-13

-0.878788 0.049245 0.335129 0.335129 -2.858837e-01 -2.858837e-01 -2.914891e-13

-0.818182 0.056384 0.056383 0.056383 5.440006e-07 5.440006e-07 -3.946149e-14

-0.757576 0.065155 -0.038870 -0.038870 1.040254e-01 1.040254e-01 7.176898e-14

-0.696970 0.076079 0.004843 0.004843 7.123659e-02 7.123659e-02 5.739072e-14

-0.636364 0.089896 0.089896 0.089896 -4.864217e-07 -4.864217e-07 1.021405e-14

-0.575758 0.107673 0.149168 0.149168 -4.149540e-02 -4.149540e-02 -2.842171e-14

-0.515152 0.130984 0.166026 0.166026 -3.504194e-02 -3.504194e-02 -3.966272e-14

-0.454545 0.162198 0.162198 0.162198 3.767443e-07 3.767442e-07 -2.586820e-14

-0.393939 0.204930 0.174554 0.174554 3.037621e-02 3.037621e-02 -4.385381e-15

-0.333333 0.264706 0.233546 0.233546 3.115994e-02 3.115994e-02 1.215694e-14

-0.272727 0.349711 0.349710 0.349710 6.227715e-07 6.227715e-07 1.765255e-14

-0.212121 0.470614 0.510041 0.510041 -3.942697e-02 -3.942697e-02 1.265654e-14

-0.151515 0.635356 0.682949 0.682949 -4.759296e-02 -4.759296e-02 2.775558e-15

-0.090909 0.828767 0.828767 0.828767 3.025643e-07 3.025643e-07 -5.995204e-15

-0.030303 0.977558 0.912039 0.912039 6.551925e-02 6.551925e-02 -1.021405e-14

0.030303 0.977558 0.912039 0.912039 6.551925e-02 6.551925e-02 -1.032507e-14

0.090909 0.828767 0.828767 0.828767 3.025643e-07 3.025643e-07 -8.437695e-15

0.151515 0.635356 0.682949 0.682949 -4.759296e-02 -4.759296e-02 -8.770762e-15

0.212121 0.470614 0.510041 0.510041 -3.942697e-02 -3.942697e-02 -1.154632e-14

0.272727 0.349711 0.349710 0.349710 6.227715e-07 6.227715e-07 -1.604272e-14

0.333333 0.264706 0.233546 0.233546 3.115994e-02 3.115994e-02 -1.867950e-14

0.393939 0.204930 0.174554 0.174554 3.037621e-02 3.037621e-02 -1.476597e-14

0.454545 0.162198 0.162198 0.162198 3.767442e-07 3.767442e-07 -3.080869e-15

0.515152 0.130984 0.166026 0.166026 -3.504194e-02 -3.504194e-02 1.137979e-14

0.575758 0.107673 0.149168 0.149168 -4.149540e-02 -4.149540e-02 2.436940e-14

0.636364 0.089896 0.089896 0.089896 -4.864217e-07 -4.864217e-07 2.262079e-14

0.696970 0.076079 0.004843 0.004843 7.123659e-02 7.123659e-02 1.462198e-14

0.757576 0.065155 -0.038870 -0.038870 1.040254e-01 1.040254e-01 1.707662e-14

0.818182 0.056384 0.056383 0.056383 5.440005e-07 5.440006e-07 6.124268e-14

0.878788 0.049245 0.335129 0.335129 -2.858837e-01 -2.858837e-01 1.789124e-13

0.939394 0.043362 0.597214 0.597214 -5.538512e-01 -5.538512e-01 3.060885e-13

1.000000 0.038462 0.038462 0.038462 -4.615385e-07 -4.615385e-07 -9.228729e-15

Опять 3 и 4 столбики полностью совпадают, а 2 и 3(4) совпадают с шагом три. Строим график в gnuplot, читая данные из файла ‘output\_12ravnomrunge.txt’ и сохраняя полученный график в файл ‘output\_12ravnomrunge.txt.png’

Изображение выглядит как карта, текст

Автоматически созданное описание

Смотрим на график – видим, что к концам отрезка приближающий многочлен разбалтывается, причем при увеличении n размах и частота колебаний будет увеличиваться. При этом, к сожалению, область колебаний (то есть где многочлен очень плохо похож на исходную функцию) не будет приближаться к концам отрезка, а так и останется примерно [-1,-0.86] и [0.86,1]. То есть увеличение n не улучшает качество приближения. Однако окажется, что если узлы не равномерные, а Чебышевские (то есть более скучены к концам отрезка), то разбалтывания не будет. Проверим это, построив график для чебышевских узлов

**б)Теперь узлы равномерные, функция по-прежнему Рунге**

Запускаем в терминале ./a.out -1 1 12 chebushov runge

В текущей директории появился файл output\_12chebushovrunge.txt

0.991445 0.039102 0.039102 0.039102 9.132475e-08 9.132474e-08 -9.707513e-15

0.968923 0.040866 0.048259 0.048259 -7.393279e-03 -7.393279e-03 9.089951e-16

0.946402 0.042750 0.048165 0.048165 -5.414687e-03 -5.414687e-03 -1.908196e-15

0.923880 0.044765 0.044765 0.044765 4.901973e-08 4.901973e-08 -6.557255e-15

0.880371 0.049077 0.040661 0.040661 8.415335e-03 8.415335e-03 -7.167877e-15

0.836862 0.054029 0.046277 0.046277 7.751919e-03 7.751919e-03 -2.310652e-15

0.793353 0.059754 0.059754 0.059754 2.026462e-07 2.026462e-07 -1.387779e-15

0.731822 0.069497 0.081057 0.081057 -1.156005e-02 -1.156005e-02 -1.720846e-15

0.670292 0.081751 0.094001 0.094001 -1.225008e-02 -1.225008e-02 -1.221245e-15

0.608761 0.097421 0.097421 0.097421 -2.046271e-07 -2.046271e-07 -1.151856e-15

0.533402 0.123260 0.103158 0.103158 2.010205e-02 2.010205e-02 -1.887379e-15

0.458042 0.160126 0.135547 0.135547 2.457935e-02 2.457935e-02 -1.082467e-15

0.382683 0.214539 0.214539 0.214539 9.972033e-09 9.972033e-09 -2.775558e-16

0.298631 0.309645 0.359375 0.359375 -4.972984e-02 -4.972984e-02 8.881784e-16

0.214578 0.464879 0.537466 0.537466 -7.258670e-02 -7.258670e-02 5.551115e-16

0.130526 0.701299 0.701298 0.701298 7.057378e-07 7.057378e-07 -8.881784e-16

0.043509 0.954813 0.803576 0.803576 1.512377e-01 1.512377e-01 -3.108624e-15

-0.043509 0.954813 0.803576 0.803576 1.512377e-01 1.512377e-01 -3.996803e-15

-0.130526 0.701299 0.701298 0.701298 7.057378e-07 7.057378e-07 -2.664535e-15

-0.214578 0.464879 0.537466 0.537466 -7.258670e-02 -7.258670e-02 1.110223e-16

-0.298631 0.309645 0.359375 0.359375 -4.972984e-02 -4.972984e-02 2.886580e-15

-0.382683 0.214539 0.214539 0.214539 9.972029e-09 9.972033e-09 3.441691e-15

-0.458042 0.160126 0.135547 0.135547 2.457935e-02 2.457935e-02 1.942890e-15

-0.533402 0.123260 0.103158 0.103158 2.010205e-02 2.010205e-02 -1.401657e-15

-0.608761 0.097421 0.097421 0.097421 -2.046271e-07 -2.046271e-07 -3.538836e-15

-0.670292 0.081751 0.094001 0.094001 -1.225008e-02 -1.225008e-02 -4.149459e-15

-0.731822 0.069497 0.081057 0.081057 -1.156005e-02 -1.156005e-02 -3.025358e-15

-0.793353 0.059754 0.059754 0.059754 2.026462e-07 2.026462e-07 -1.554312e-15

-0.836862 0.054029 0.046277 0.046277 7.751919e-03 7.751919e-03 -2.581269e-15

-0.880371 0.049077 0.040661 0.040661 8.415335e-03 8.415335e-03 -9.339751e-15

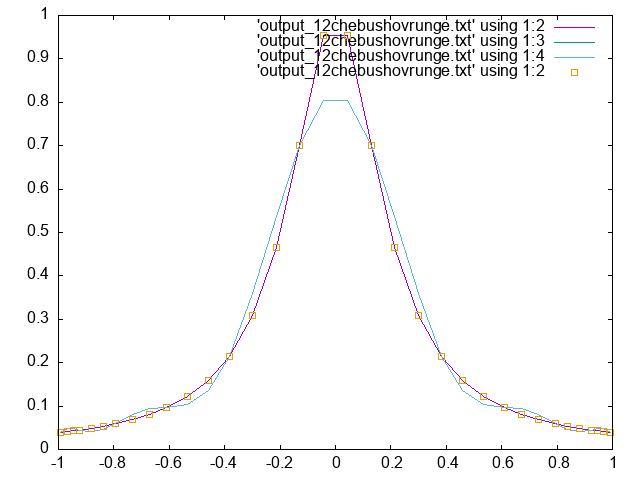
-0.923880 0.044765 0.044765 0.044765 4.901974e-08 4.901973e-08 -1.487005e-14

-0.946402 0.042750 0.048165 0.048165 -5.414687e-03 -5.414687e-03 -7.334411e-15

-0.968923 0.040866 0.048259 0.048259 -7.393279e-03 -7.393279e-03 1.616762e-15

-0.991445 0.039102 0.039102 0.039102 9.132475e-08 9.132474e-08 -8.555656e-15

Опять 3 и 4 столбики полностью совпадают, а 2 и 3(4) совпадают с шагом три. Строим график в gnuplot, читая данные из файла ‘output\_12chebushovrunge.txt txt' и сохраняя полученный график в файл ‘output\_12chebushovrunge.txt.png’.



Смотрим на график – видим, что к концам отрезка разбалтывание есть, то намного меньше, чем при интерполяции с равномерными узлами.

**3)Функция модуль**

Это функция (x>0)?x:-x;

Пусть мы хотим построить нашу функцию по n=15 узлам

**a)Пусть сначала узлы равномерные**

Запускаем в терминале ./a.out -1 1 15 ravnom modul

В текущей директории появился файл output\_15ravnommodul.txt

Посмотрим на него

-1.000000 1.000000 1.000000 1.000000 -2.728484e-12 0.000000e+00 2.728484e-12

-0.952381 0.952381 -3.006273 -3.006273 3.958654e+00 3.958654e+00 1.945688e-11

-0.904762 0.904762 -0.840552 -0.840552 1.745314e+00 1.745314e+00 7.022827e-12

-0.857143 0.857143 0.857146 0.857146 -3.091200e-06 -3.091201e-06 -1.294964e-12

-0.809524 0.809524 1.261678 1.261678 -4.521537e-01 -4.521537e-01 -2.945200e-12

-0.761905 0.761905 1.019501 1.019501 -2.575959e-01 -2.575959e-01 -1.613598e-12

-0.714286 0.714286 0.714285 0.714285 1.133719e-06 1.133719e-06 -4.203304e-13

-0.666667 0.666667 0.566332 0.566332 1.003343e-01 1.003343e-01 -2.688960e-13

-0.619048 0.619048 0.551154 0.551154 6.789378e-02 6.789378e-02 -8.324452e-13

-0.571429 0.571429 0.571429 0.571429 -5.252590e-07 -5.252605e-07 -1.501688e-12

-0.523810 0.523810 0.559774 0.559774 -3.596445e-02 -3.596445e-02 -1.856182e-12

-0.476190 0.476190 0.504238 0.504238 -2.804705e-02 -2.804705e-02 -1.836642e-12

-0.428571 0.428571 0.428572 0.428572 -2.489160e-07 -2.489176e-07 -1.614098e-12

-0.380952 0.380952 0.361450 0.361450 1.950254e-02 1.950254e-02 -1.409262e-12

-0.333333 0.333333 0.315937 0.315937 1.739649e-02 1.739649e-02 -1.365408e-12

-0.285714 0.285714 0.285714 0.285714 1.179442e-07 1.179427e-07 -1.489642e-12

-0.238095 0.238095 0.254049 0.254049 -1.595352e-02 -1.595352e-02 -1.693035e-12

-0.190476 0.190476 0.206985 0.206985 -1.650848e-02 -1.650848e-02 -1.857126e-12

-0.142857 0.142857 0.142857 0.142857 -6.575061e-08 -6.575252e-08 -1.902062e-12

-0.095238 0.095238 0.073910 0.073910 2.132820e-02 2.132820e-02 -1.816158e-12

-0.047619 0.047619 0.020272 0.020272 2.734745e-02 2.734745e-02 -1.649899e-12

0.000000 -0.000000 -0.000000 0.000000 1.480074e-12 -0.000000e+00 -1.480074e-12

0.047619 0.047619 0.020272 0.020272 2.734745e-02 2.734745e-02 -1.367725e-12

0.095238 0.095238 0.073910 0.073910 2.132820e-02 2.132820e-02 -1.330533e-12

0.142857 0.142857 0.142857 0.142857 -6.575117e-08 -6.575252e-08 -1.341705e-12

0.190476 0.190476 0.206985 0.206985 -1.650848e-02 -1.650848e-02 -1.353612e-12

0.238095 0.238095 0.254049 0.254049 -1.595352e-02 -1.595352e-02 -1.331935e-12

0.285714 0.285714 0.285714 0.285714 1.179440e-07 1.179427e-07 -1.280975e-12

0.333333 0.333333 0.315937 0.315937 1.739649e-02 1.739649e-02 -1.243450e-12

0.380952 0.380952 0.361450 0.361450 1.950254e-02 1.950254e-02 -1.270206e-12

0.428571 0.428571 0.428572 0.428572 -2.489163e-07 -2.489176e-07 -1.373568e-12

0.476190 0.476190 0.504238 0.504238 -2.804705e-02 -2.804705e-02 -1.493139e-12

0.523810 0.523810 0.559774 0.559774 -3.596445e-02 -3.596445e-02 -1.506018e-12

0.571429 0.571429 0.571429 0.571429 -5.252592e-07 -5.252605e-07 -1.321832e-12

0.619048 0.619048 0.551154 0.551154 6.789378e-02 6.789378e-02 -9.928725e-13

0.666667 0.666667 0.566332 0.566332 1.003343e-01 1.003343e-01 -7.909229e-13

0.714286 0.714286 0.714285 0.714285 1.133720e-06 1.133719e-06 -1.140754e-12

0.761905 0.761905 1.019501 1.019501 -2.575959e-01 -2.575959e-01 -2.212674e-12

0.809524 0.809524 1.261678 1.261678 -4.521537e-01 -4.521537e-01 -3.173017e-12

0.857143 0.857143 0.857146 0.857146 -3.091200e-06 -3.091201e-06 -1.555645e-12

0.904762 0.904762 -0.840552 -0.840552 1.745314e+00 1.745314e+00 5.198508e-12

0.952381 0.952381 -3.006273 -3.006273 3.958654e+00 3.958654e+00 1.353673e-11

1.000000 1.000000 1.000000 1.000000 1.818989e-12 0.000000e+00 -1.818989e-12

Опять 3 и 4 столбики полностью совпадают, а 2 и 3(4) совпадают с шагом три. Строим график в gnuplot, читая данные из файла output\_15ravnommodul.txt и сохраняя полученный график в файл output\_15ravnommodul.txt.png.

Изображение выглядит как текст, карта

Автоматически созданное описание

Смотрим на график – видим, что к концам отрезка приближающий многочлен разбалтывается.

**б)Теперь узлы равномерные, функция по-прежнему модуль**

Запускаем в терминале ./a.out -1 1 15 chebushov modul

В текущей директории появился файл output\_15chebushovmodul.txt

Посмотрим на него

0.994522 0.994522 0.994522 0.994522 1.076916e-14 0.000000e+00 -1.076916e-14

0.980034 0.980034 0.973035 0.973035 6.998918e-03 6.998918e-03 -5.229150e-14

0.965545 0.965545 0.960589 0.960589 4.956222e-03 4.956222e-03 -3.286260e-14

0.951057 0.951057 0.951057 0.951057 1.554312e-14 0.000000e+00 -1.554312e-14

0.922713 0.922713 0.929753 0.929753 -7.039780e-03 -7.039780e-03 -5.095924e-14

0.894369 0.894369 0.900397 0.900397 -6.028128e-03 -6.028128e-03 -4.696243e-14

0.866025 0.866025 0.866025 0.866025 -4.440892e-16 0.000000e+00 4.440892e-16

0.825065 0.825065 0.817456 0.817456 7.608624e-03 7.608624e-03 -1.654232e-14

0.784105 0.784105 0.776990 0.776990 7.115365e-03 7.115365e-03 -2.331468e-15

0.743145 0.743145 0.743145 0.743145 -2.575717e-14 0.000000e+00 2.575717e-14

0.691358 0.691358 0.700179 0.700179 -8.820343e-03 -8.820343e-03 3.252953e-14

0.639572 0.639572 0.648357 0.648357 -8.785349e-03 -8.785349e-03 4.629630e-14

0.587785 0.587785 0.587785 0.587785 -5.362377e-14 0.000000e+00 5.362377e-14

0.527436 0.527436 0.516304 0.516304 1.113156e-02 1.113156e-02 6.528111e-14

0.467086 0.467086 0.455292 0.455292 1.179405e-02 1.179405e-02 7.560619e-14

0.406737 0.406737 0.406737 0.406737 -8.315570e-14 0.000000e+00 8.315570e-14

0.340462 0.340462 0.356408 0.356408 -1.594557e-02 -1.594557e-02 8.648637e-14

0.274187 0.274187 0.292516 0.292516 -1.832871e-02 -1.832871e-02 8.132384e-14

0.207912 0.207912 0.207912 0.207912 -6.972201e-14 0.000000e+00 6.972201e-14

0.138608 0.138608 0.109682 0.109682 2.892571e-02 2.892571e-02 5.509482e-14

0.069304 0.069304 0.030505 0.030505 3.879907e-02 3.879907e-02 4.202194e-14

0.000000 -0.000000 0.000000 0.000000 -3.256439e-14 -0.000000e+00 3.256439e-14

-0.069304 0.069304 0.030505 0.030505 3.879907e-02 3.879907e-02 2.717618e-14

-0.138608 0.138608 0.109682 0.109682 2.892571e-02 2.892571e-02 2.593759e-14

-0.207912 0.207912 0.207912 0.207912 -2.997602e-14 0.000000e+00 2.997602e-14

-0.274187 0.274187 0.292516 0.292516 -1.832871e-02 -1.832871e-02 3.957945e-14

-0.340462 0.340462 0.356408 0.356408 -1.594557e-02 -1.594557e-02 5.101475e-14

-0.406737 0.406737 0.406737 0.406737 -5.451195e-14 0.000000e+00 5.451195e-14

-0.467086 0.467086 0.455292 0.455292 1.179405e-02 1.179405e-02 4.013456e-14

-0.527436 0.527436 0.516304 0.516304 1.113156e-02 1.113156e-02 1.443290e-15

-0.587785 0.587785 0.587785 0.587785 5.229150e-14 0.000000e+00 -5.229150e-14

-0.639572 0.639572 0.648357 0.648357 -8.785349e-03 -8.785349e-03 -9.503509e-14

-0.691358 0.691358 0.700179 0.700179 -8.820343e-03 -8.820343e-03 -9.514611e-14

-0.743145 0.743145 0.743145 0.743145 6.261658e-14 0.000000e+00 -6.261658e-14

-0.784105 0.784105 0.776990 0.776990 7.115365e-03 7.115365e-03 -1.543210e-14

-0.825065 0.825065 0.817456 0.817456 7.608624e-03 7.608624e-03 3.874678e-14

-0.866025 0.866025 0.866025 0.866025 -2.886580e-14 0.000000e+00 2.886580e-14

-0.894369 0.894369 0.900397 0.900397 -6.028128e-03 -6.028128e-03 -9.658940e-14

-0.922713 0.922713 0.929753 0.929753 -7.039780e-03 -7.039780e-03 -1.931788e-13

-0.951057 0.951057 0.951057 0.951057 2.429168e-13 0.000000e+00 -2.429168e-13

-0.965545 0.965545 0.960589 0.960589 4.956222e-03 4.956222e-03 -2.314815e-13

-0.980034 0.980034 0.973035 0.973035 6.998918e-03 6.998918e-03 -1.660894e-13

-0.994522 0.994522 0.994522 0.994522 1.076916e-14 0.000000e+00 -1.076916e-14

Опять 3 и 4 столбики полностью совпадают, а 2 и 3(4) совпадают с шагом три. Строим график в gnuplot, читая данные из файла output\_15chebushovmodul.txt и сохраняя полученный график в файл output\_15chebushovmodul.txt.png.

Изображение выглядит как карта

Автоматически созданное описание

Смотрим на график – видим, что функция хорошо приблизилась.