Токаева Александра, 409 группа
Отчет по практикуму на ЭВМ

Задание 2:
загадана некоторая функция f условием $f(a) = f(b) = 0$; дан отрезок [a, b], дана N+1 точка $x_0 = a, x_1, x_2, \dots, x_N = b$ на этом отрезке и даны значения $y_0 = f(x_0) = a, y_1 = f(x_1), \dots, y_N = f(x_N) = 0$ загаданной функции в этих точках.

Мы хотим приблизить нашу функцию так:
$f(x) \approx \sum_{n=1}^{N-1} \sin(\pi n x) = 0 + c_1 \sin(\pi x) + c_2 \sin(\pi 2x) + \cdots + c_{N-1} \sin(\pi(N-1)x) + 0.$
То есть $(f_0\, ; f_1; \dots\ f_N)^T = \ c_1(\sin(\pi * 1 * x_0)\, ; \sin(\pi * 1 * x_1) \dots\ \sin(\pi * 1 * x_N)\ )^T + \ c_2(\sin(\pi * 2 * x_0)\, ; \sin(\pi * 2 * x_1) \dots\ \sin(\pi * 2 * x_N)\ )^T + \cdots + c_{N-1}(\sin(\pi * (N-1) * x_0)\, ; \sin(\pi * (N-1) * x_1) \dots\ \sin(\pi * (N-1) * x_N)\ )^T$ (*)

Обозначим за $\varphi^{(n)} :=\ c_1(\sin(\pi * n * x_0)\, ; \sin(\pi * n * x_1) \dots\ \sin(\pi * n * x_N)\ )^T; n = 0 \dots . N$
Тогда вектор φ имеет длину N+1, вектор f тоже имеет размерность N+1 (при этом 0-я и N-я координаты вектора f равны нулю), а вектор с имеет длину N-1.
Для чисто технического удобства (а именно взятия скалярного произведения, ведь считать скалярное произведение можно только у векторов одинаковой длины) мы хотим равенства размерностей векторов φ, f и с, поэтому мы введем фиктивные нулевые значения $c_0 = c_N = 0$. Теперь нам надо найти коэффициенты c_n; для этого возьмем последовательно скалярное произведения вектора f с каждым из N-1 вектора $\varphi^{(n)}$; Слева будет стоять честное скалярное произведение $<$f, $\varphi^{(n)}>$, а справа благодаря представлению (*) и ортогональности синусов с разными n будет не равно нулю только одно слагаемое $c_n<\varphi^{(n)}, \varphi^{(n)}>$.
Из этого мы сразу находим коэффициент $c_n = \frac{<\mathrm{f}, \varphi^{(n)}>}{<\varphi^{(n)}, \varphi^{(n)}>}$
Причем скалярное произведение не обычное,
а с домножением на h, чтобы в пределе при h->0 получался интеграл: $< a, b > = \ \sum_{n=0}^{N} a_n\, b_n h$

Еще нам требуется найти, какой степени h пропорционален err=максимум по всем узлам (а мы между каждыми двумя исходными узлами поместим еще по 2 дополнительных узла) модулей разностей между f_i и нашим тригонометрическим многочленом;
Пусть мы из каких-то соображений надеемся, что этот максимум пропорционален h в какой-то степени p; чтобы приблизительно оценить значение p (то есть понять, p примерно равно 1, 3/2, 5, 10 и тд), мы посчитаем err для N=2,2,4…30 и построим график зависимости log(1/err) от log(N). Если наша гипотеза про то что err=C*h^p верна, то на графике мы увидим прямую, и тангенс угла наклона этой прямой и будет искомым значением p, потому что

$err = Ch^p = C\left(\frac{1}{N}\right)^p \Rightarrow \log(err) = logC + plog(\frac{1}{N})$

$\Rightarrow -\log(err) = -logC - plog(\frac{1}{N})$

$\Rightarrow \log(1/err) = -logC + plog(N)$

Но $log(\frac{1}{N})$ будет очень близок к (-infinity) при растущих N, поэтому график зависимости $\log(err)$ от $log(\frac{1}{N})$ мы строить не будем, а построим график зависимости log(1/err) от log(N).

Для начала проверим, что наша программа вообще правильно работает. Для этого введем в качестве функции f=sin(pi*5*x); тогда очевидно, что все коэффициенты, кроме $c_5 = 1$, будут равны нулю. Проверяем это для N=7:

```
Hello!
 a=0.000000 b=1.000000 N=7 is_ravnom=0 myfunc2
ravnom uzl
Vector x:
0.000000 0.142857 0.285714 0.428571 0.571429 0.714286
0.857143 1.000000
Vector f:
0.000000 0.781831 -0.974928 0.433884 0.433884 -0.974928
0.781831 0.000000
Vector c:
0.000000 -0.000000 -0.000000 -0.000000 -0.000000
1.000000 0.000000 0.000000
tmp_str=7
output_name=output_7ravnommyfunc2.txt
0.000000 0.000000 0.000000 0.000000e+00
```

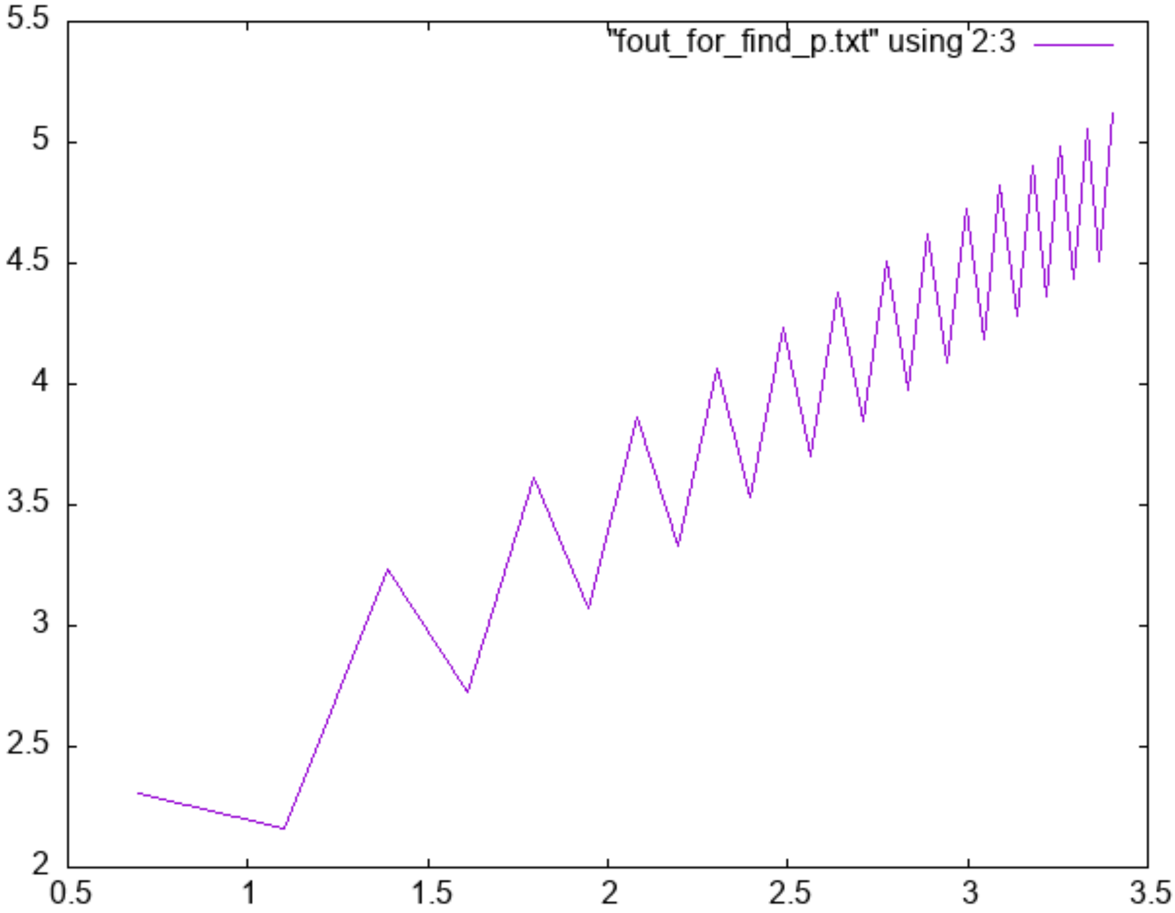
```
0.055556 0.766044 0.766044 2.839940e-07
0.111111 0.984808 0.984807 4.575832e-07
0.166667 0.500000 0.500000 4.648381e-07
0.222222 -0.342020 -0.342020 3.263879e-07
0.277778 -0.939693 -0.939693 1.181133e-07
0.333333 -0.866025 -0.866025 -7.657999e-08
0.388889 -0.173648 -0.173648 -2.085560e-07
0.444444 0.642788 0.642788 -2.737288e-07
0.500000 1.000000 1.000000 -2.916505e-07
0.555556 0.642788 0.642788 -2.737288e-07
0.611111 -0.173648 -0.173648 -2.085560e-07
0.666667 -0.866025 -0.866025 -7.657999e-08
0.722222 -0.939693 -0.939693 1.181133e-07
0.777778 -0.342020 -0.342020 3.263879e-07
0.833333 0.500000 0.500000 4.648381e-07
0.888889 0.984808 0.984807 4.575832e-07
0.944444 0.766044 0.766044 2.839940e-07
1.000000 0.000000 0.000000 2.136464e-22
err=0.000000
Goodbuy!
```

Видим, что действительно 5-й коэффициент=1, а остальные-нулевые;

Теперь для построения графика зависимости log(1/err) от log(N) запусим программу на функции y=-0.5 + |x-0.5| для N=2,3,4…30 и запишем результаты в формате N, log(N) , log(1/err) в файл `"fout_for_find_p.txt"`

А потом построим график `plot "fout_for_find_p.txt" using 2:3 with lines`

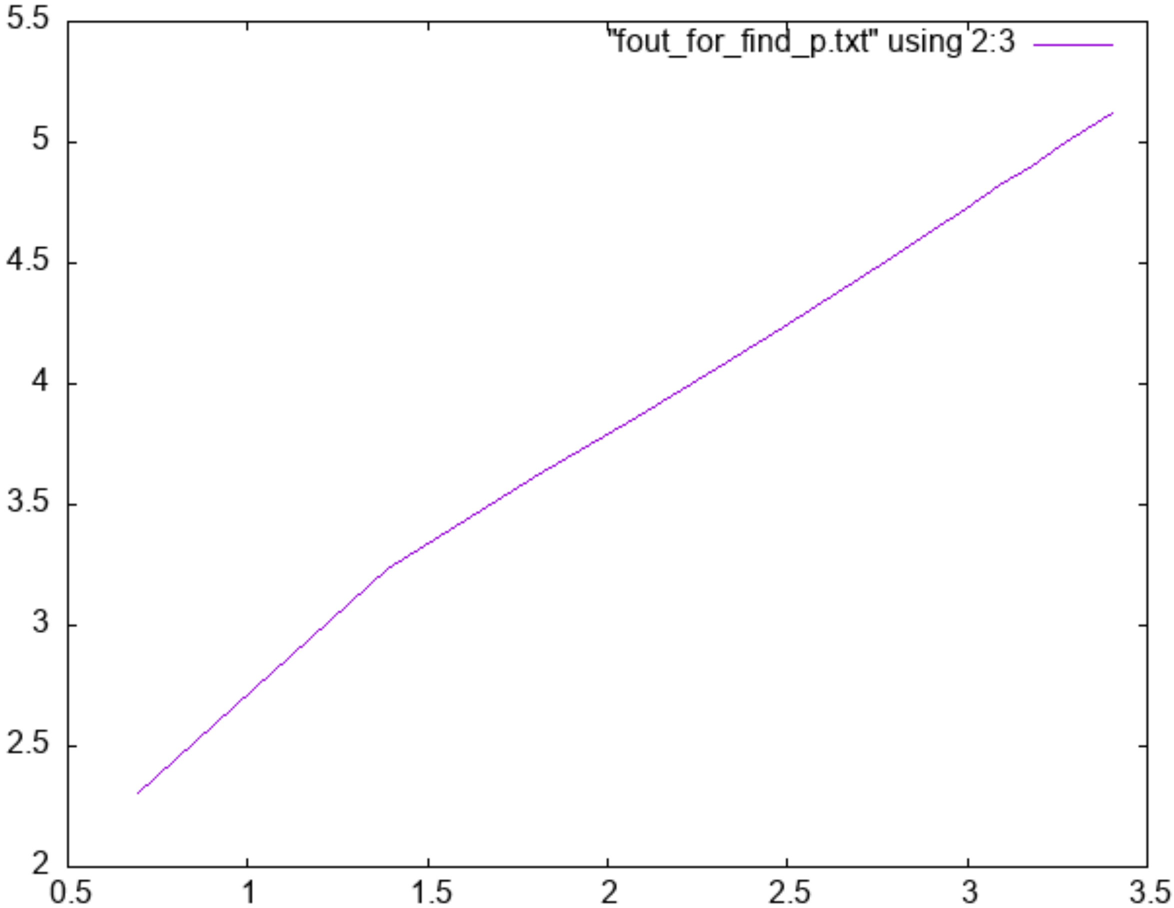
Получим график



Видим, что отдельно при четных и при нечетных N это прямая с одинаковым углом наклона, но при нечетных у нее свободный член отличается от свободного члена при четных N. Чтобы вычислить p, построим этот график только при четных N, то есть при таких значениях:

2 0.693147 2.305797
4 1.386294 3.237503
6 1.791759 3.610759
8 2.079442 3.860691
10 2.302585 4.064934
12 2.484907 4.235894
14 2.639057 4.382505
16 2.772589 4.510607
18 2.890372 4.624354
20 2.995732 4.726530
22 3.091042 4.819341
24 3.178054 4.904276
26 3.258097 4.982605
28 3.332205 5.055192
30 3.401197 5.122930

Получим график



Это явно почти прямая, и p=тангенс угла наклона
Посчитаем разность значений для строчек с N=10 и N=30
10 2.302585 4.064934
30 3.401197 5.122930

=>p=(5.122930-4.064934)/(3.401197-2.302585)=1.057996/1.098612=0.96302971 ≈ 1

То есть p примерно равно единице

То есть эмпирически получилось, что err ≈Ch