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 $x_0 = a, x_1, x_2, \ldots, x_N = b$ на этом отрезке и даны значения $y_0 = f(x_0) = a, y_1 = f(x_1), \ldots, y_N = f(x_N) = 0$ загаданной функции в этих точках.

Мы хотим приблизить нашу функцию так:
$f(x) \approx \sum_{n=1}^{N-1} \sin(\pi(n - 0.5)x) = 0 + c_1 \sin(\pi * 0.5 * x) + c_2 \sin(\pi * 1.5 * x) + \cdots + c_{N-1} \sin(\pi(N - 1 - 0.5)x) + 0.$
То есть $(f_0\,; f_1; \ldots\ f_N)^T = \ c_1(\sin(\pi * 0.5 * x_0)\,; \sin(\pi * 0.5 * x_1) \ldots\ \sin(\pi * 0.5 * x_N)\ )^T +\ c_2(\sin(\pi * 1.5 * x_0)\,; \sin(\pi * 1.5 * x_1) \ldots\ \sin(\pi * 1.5 * x_N)\ )^T + \cdots + c_{N-1}(\sin(\pi * (N - 1 - 0.5) * x_0)\,; \sin(\pi * (N - 1 - 0.5) * x_1) \ldots\ \sin(\pi * (N - 1 - 0.5) * x_N)\ )^T$ (*)

Обозначим за $\varphi^{(n)} \coloneqq\ c_1(\sin(\pi * (n - 0.5) * x_0)\,; \sin(\pi * (n - 0.5) * x_1) \ldots\ \sin(\pi * (n - 0.5) * x_N)\ )^T; n = 0 \ldots. N$
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 c 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 c, 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 $c_n = \frac{<f, \varphi^{(n)}>}{<\varphi^{(n)}, \varphi^{(n)}>}$
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

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

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

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

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*5.5*x); тогда очевидно, что все коэффициенты, кроме $c_6 = 1$, будут равны нулю.
Проверяем это для N=7:

```
(base) MacBook-Pro-Aleksandra:task2_0101 aleksandra$ ./a.out 0 1 7
ravnom myfunc2
Hello!
 a=0.000000 b=1.000000 N=7 is_ravnom=0 myfunc2
ravnom uzl
Vector x:
0.000000 0.153846 0.307692 0.461538 0.615385 0.769231 0.923077
1.076923
Vector f:
0.000000 0.464723 -0.822984 0.992709 -0.935016 0.663123 -0.239316 -
0.239316
Vector c:
0.000000 0.000000 0.000000 -0.000000 0.000000 -0.000000 1.000000
0.000000
tmp_str=7
output_name=output_7ravnommyfunc2.txt
0.000000 0.000000 0.000000 0.000000e+00
```

```
0.051282 0.774605 0.774605 5.020899e-08
0.102564 0.979791 0.979791 1.099421e-07
0.153846 0.464723 0.464723 1.720438e-07
0.205128 -0.391967 -0.391967 2.112831e-07
0.256410 -0.960518 -0.960518 2.008510e-07
0.307692 -0.822984 -0.822984 1.341063e-07
0.358974 -0.080467 -0.080467 3.344679e-08
0.410256 0.721202 0.721203 -6.261986e-08
0.461538 0.992709 0.992709 -1.259019e-07
0.512821 0.534466 0.534466 -1.582665e-07
0.564103 -0.316668 -0.316668 -1.879236e-07
0.615385 -0.935016 -0.935016 -2.426854e-07
0.666667 -0.866025 -0.866025 -3.192545e-07
0.717949 -0.160411 -0.160411 -3.732439e-07
0.769231 0.663123 0.663123 -3.417592e-07
0.820513 0.999189 0.999189 -1.870631e-07
0.871795 0.600742 0.600742 6.743587e-08
0.923077 -0.239316 -0.239316 3.357124e-07
0.974359 -0.903450 -0.903451 5.078371e-07
1.025641 -0.903450 -0.903451 5.078371e-07
err=0.000001
Goodbuy!
```

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

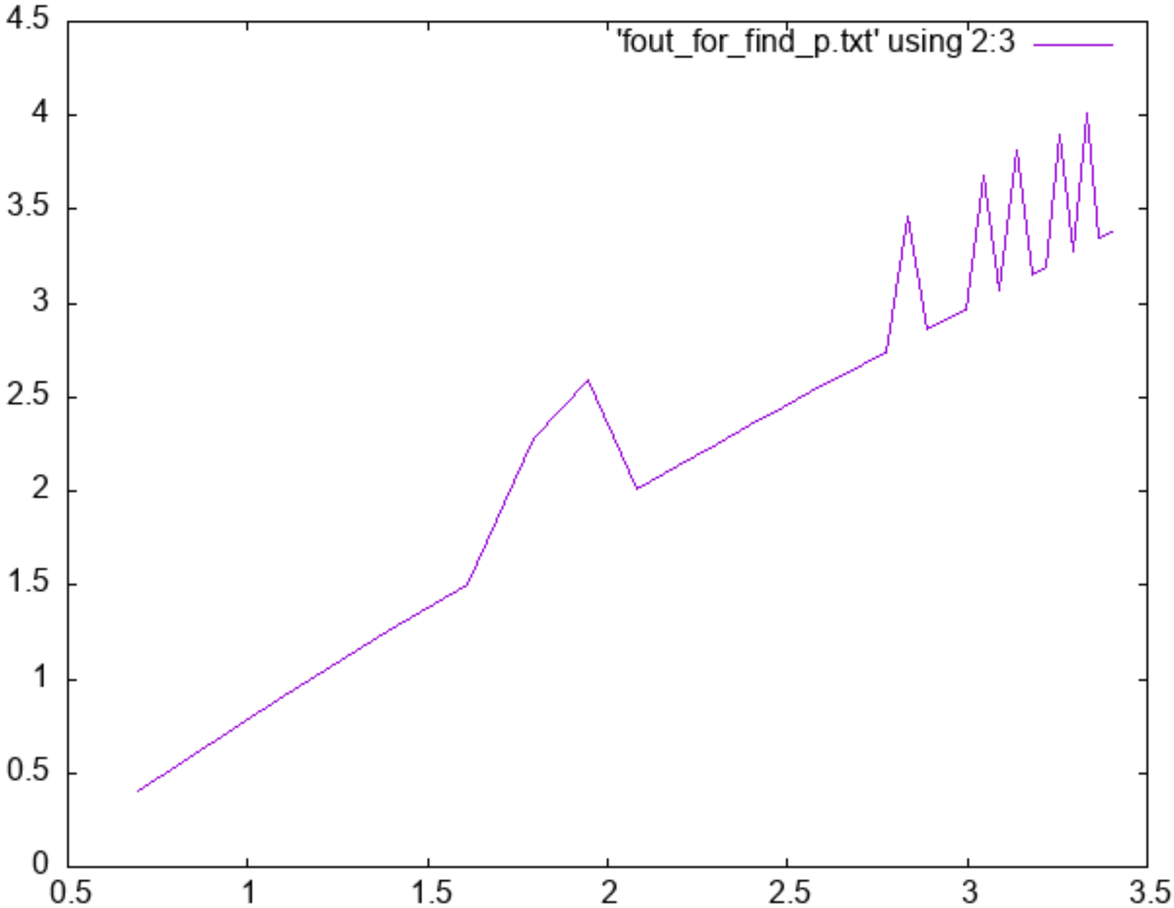
Теперь для построения графика зависимости log(1/err) от log(N) запусим программу на функции y=-0.5 + |x-0.5| для N=2,3,4…30 и запишем результаты в формате N, log(N) , log(1/err) в файл `"fout_for_find_p.txt"`

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

```
2 0.693147 0.405466
3 1.098612 0.916291
4 1.386294 1.252763
5 1.609438 1.504078
6 1.791759 2.286142
7 1.945910 2.590300
8 2.079442 2.014901
9 2.197225 2.140062
10 2.302585 2.251288
11 2.397895 2.351376
12 2.484907 2.442350
13 2.564949 2.525729
14 2.639057 2.602690
15 2.708050 2.674145
16 2.772589 2.740841
17 2.833213 3.462396
18 2.890372 2.862208
19 2.944439 2.917771
```

```
20 2.995732 2.970415
21 3.044522 3.683191
22 3.091042 3.068049
23 3.135494 3.811543
24 3.178054 3.156991
25 3.218876 3.198677
26 3.258097 3.901607
27 3.295837 3.277143
28 3.332205 4.012155
29 3.367296 3.349900
30 3.401197 3.384395
```

Получим график



Это явно почти прямая, и p=тангенс угла наклона

Посчитаем разность значений для строчек с N=5 и N=30

```
5 1.609438 1.504078
30 3.401197 3.384395
```

=>p=(3.384395-1.504078)/(3.401197/1.609438)=
1,880317/2,1132824=0,889761349

То есть p примерно равно 0.9

То есть эмпирически получилось, что err ≈C*h^(0.9)