Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

Лабораторная работа № 7

«Тригонометрическая интерполяция функций с помощью быстрого преобразования Фурье»

по курсу «Численные методы»

Выполнил:

студент группы ИУ9-61Б

Митрошкин Алексей

Проверила:

Домрачева А. Б.

Москва, 2024

1. Цель

Целью данной работы является изучение метода быстрого преобразования Фурье для построения тригонометрической интерполяции функций.

2. Постановка задачи

Дано: периодическая функция $f(x)$

Задание:

- Вычислить значения функции $f(x)$ в узлах сетки $x_j = \frac{j}{N}, N = 128$;
- Построить тригонометрическую интерполяцию функции, пользуясь БНФ для подсчета дискретных коэффициентов Фурье;
- Сравнить значения тригонометрической интерполяции в средних точках всех отрезков разбиения $y_j = 0{,}5 + \frac{j}{N}, j = 0, \dots, N-1$.

Индивидуальный вариант:

$$f(x) = \sin(\left|x - \frac{1}{2}\right|)$$

3. Основные теоретические сведения и этапы работы

Описание метода:

Пусть периодическая функция с периодом I может быть разложена в ряд Фурье:

$$f(x) = \sum_{q=-\infty}^{\infty} a_q \exp(2\pi i q x)$$

Рассмотрим значения этой функции на сетке из точек $x_j = j/N$, где j, N – целые числа, и обозначим $f(x_j) = f_j$. Тогда из-за периодичности функции $f(x)$ ряд Фурье можно записать в виде

$$f(x_j) = f_j = \sum_{q=0}^{N-1} A_q \exp(2\pi i q x_j)$$

Где

$$A_q = \sum_{S=-\infty}^{\infty} a_{q+sN}$$

Верно и обратное равенство:

$$A_q = \frac{1}{N}\sum_{j=0}^{N-1} f_j \exp(-2\pi i q x_j)$$

Аппроксимация функции с помощью приближенного равенства

$$f(x) \approx \sum_{q=0}^{N-1} A_q \exp(2\pi i q x)$$

Носит название тригонометрической интерполяции: коэффициенты A_q называются дискретными коэффициентами Фурье.

Быстрое преобразование Фурье (БНФ) применяют, если число узлов сетки $N = 2^r$, представим числа q и j, входящие в предыдущие формулы и лежащие в пределах $0 \leq q, j \ < N$, в виде двоичного разложения:

$$q = \sum_{k=1}^{r} q_k 2^{k-1},\ j = \sum_{m=1}^{r} j_{r+1-m}\, 2^{m-1},$$

Где $q_k, j_m = 0, 1$. БНФ состоит в подсчете коэффициентов A_q с помощью рекуррентных соотношений

$$A_q = A^{(r)}(q_1, \dots, q_r),$$

$$A^{(m)}(q_1, \dots, q_m, j_{m+1}, \dots, j_r) =$$

$$= \frac{1}{2}\sum_{j_m=0}^{i} \exp\left(-2\pi i q j_m 2^{-m} \sum_{k=1}^{m} q_k 2^{k-1}\right) A^{(m-1)}(q_1, \dots, q_m, j_m, \dots, j_r),$$

$$m = 1, \dots, r,$$

Где

$$A^{(0)}(j_1, \dots, j_r) = f_{j_r + j_{r-1}2 + \dots + 2^{r-1}}$$

4. Реализация

Листинг 1. Метода быстрого преобразования Фурье для построения тригонометрической интерполяции функций.

```
using System;
using System.Numerics;

public class FftFunction
{
    private readonly Complex[] coefs;

    public FftFunction(Complex[] coefs)
    {
        this.coefs = coefs;
    }

    public Complex CalcAt(double x)
    {
        Complex res = Complex.Zero;

        for (int q = 0; q < coefs.Length; q++)
        {
            res += coefs[q] * Complex.Exp(-2 * Math.PI * Complex.ImaginaryOne *
q * x);
        }

        return res;
    }
}

public class Program
{
    private  static int r = 7;
    private static int N = Convert.ToInt32(Math.Pow(2,r));

    private static double Func(double x)
    {
        return Math.Sin(Math.Abs(x - 0.5));
    }

    private static FftFunction FFT((double, double)[] points)
    {
        Complex[] Aq = new Complex[N];

        for (int q = 0; q < N; q++)
        {
            Complex[] APrev = new Complex[points.Length];

            for (int i = 0; i < points.Length; i++)
            {
                APrev[i] = points[i].Item2;
            }

            for (int m = 1; m <= r; m++)
            {
                Complex[] A = new Complex[APrev.Length / 2];

                for (int k = 0; k < APrev.Length / 2; k++)
                {
                    A[k] = 0.5 * (
```

```
                        Complex.Exp(-2 * Math.PI * Complex.ImaginaryOne * 0 *
(1.0 / Math.Pow(2, m)) * (q % (int)Math.Pow(2, m))) * APrev[k] +
                        Complex.Exp(-2 * Math.PI * Complex.ImaginaryOne * 1 *
(1.0 / Math.Pow(2, m)) * (q % (int)Math.Pow(2, m))) * APrev[k + APrev.Length /
2]);
                }

                APrev = A;
            }

            Aq[q] = APrev[0];
        }

        return new FftFunction(Aq);
    }

    public static void Main()
    {
        (double, double)[] points = new (double, double)[N];
        for (int j = 0; j < N; j++)
        {
            points[j] = (j / (double)N, Func(j / (double)N));
        }

        FftFunction fft = FFT(points);

        foreach ((double xj, _) in points)
        {
            var abs = Math.Abs(fft.CalcAt(xj).Real - Func(xj + 0.5 / N));
            Console.WriteLine($"{xj + 0.5 / N} | {fft.CalcAt(xj).Real:F15} |
{Func(xj + 0.5 / N):F15} | {abs:F15}");
        }


    }
}
```

5. Результаты

Было написано программное обеспечение для метода быстрого преобразования Фурье для построения тригонометрической интерполяции функций.

Рисунок 1 – Пример вывода программы

В первом столбце – значение точки x_j, во втором – значение тригонометрической аппроксимации, в третьем – аналитическое значение функции, в четвёртом – абсолютная погрешность измерений

0,00390625	0,479425538604203	0,475993832717880	0,003431705886323
0,01171875	0,472554863751304	0,469108684178911	0,003446179572393
0,01953125	0,465655346585160	0,462194903663735	0,003460442921425
0,02734375	0,458727408216736	0,455252913153880	0,003474495062857
0,03515625	0,451771471491684	0,448283136352666	0,003488335139018
0,04296875	0,444787960964527	0,441285998659345	0,003501962305182
0,05078125	0,437777302872755	0,434261927143138	0,003515375729617
0,05859375	0,430739925110803	0,427211350517167	0,003528574593636
0,06640625	0,423676257203938	0,420134699112290	0,003541558091648
0,07421875	0,416586730282041	0,413032404850832	0,003554325431209
0,08203125	0,409471777053295	0,405904901220230	0,003566875833065
0,08984375	0,402331831777773	0,398752623246568	0,003579208531205
0,09765625	0,395167330240934	0,391576007468031	0,003591322772903
0,10546875	0,387978709727025	0,384375491908255	0,003603217818770
0,11328125	0,380766408992390	0,377151516049597	0,003614892942793
0,12109375	0,373530868238693	0,369904520806310	0,003626347432383
0,12890625	0,366272529086047	0,362634948497631	0,003637580588416
0,13671875	0,358991834546065	0,355343242820786	0,003648591725279
0,14453125	0,351689228994815	0,348029848823905	0,003659380170909
0,15234375	0,344365158145698	0,340695212878865	0,003669945266834
0,16015625	0,337020069022254	0,333339782654038	0,003680286368216
0,16796875	0,329654409930860	0,325964007086974	0,003690402843886
0,17578125	0,322268630433388	0,318568336356999	0,003700294076389
0,18359375	0,314863181319746	0,311153221857734	0,003709959462013
0,19140625	0,307438514580380	0,303719116169549	0,003719398410830
0,19921875	0,299995083378682	0,296266473031940	0,003728610346742
0,20703125	0,292533342023326	0,288795747315831	0,003737594707495
0,21484375	0,285053745940546	0,281307394995814	0,003746350944732
0,22265625	0,277556751646335	0,273801873122318	0,003754878524017
0,23046875	0,270042816718585	0,266279639793714	0,003763176924871
0,23828125	0,262512399769153	0,258741154128352	0,003771245640801
0,24609375	0,254965960415878	0,251186876236544	0,003779084179335
0,25390625	0,247403959254523	0,243617267192475	0,003786692062048
0,26171875	0,239826857830662	0,236032789006066	0,003794068824595
0,26953125	0,232235118611511	0,228433904594775	0,003801214016737
0,27734375	0,224629204957706	0,220821077755338	0,003808127202367
0,28515625	0,217009581095011	0,213194773135470	0,003814807959541
0,29296875	0,209376712085994	0,205555456205496	0,003821255880499
0,30078125	0,201731063801640	0,197903593229946	0,003827470571693
0,30859375	0,194073102892911	0,190239651239099	0,003833451653812
0,31640625	0,186403296762271	0,182564098000472	0,003839198761800
0,32421875	0,178722113535152	0,174877401990272	0,003844711544880
0,33203125	0,171030022031396	0,167180032364807	0,003849989666590
0,33984375	0,163327491736614	0,159472458931843	0,003855032804771
0,34765625	0,155614992773558	0,151755152121938	0,003859840651619
0,35546875	0,147892995873411	0,144028582959723	0,003864412913688
0,36328125	0,140161972347065	0,136293223035154	0,003868749311911
0,37109375	0,132422394056349	0,128549544474732	0,003872849581617
0,37890625	0,124674733385225	0,120798019912684	0,003876713472541
0,38671875	0,116919463210978	0,113039122462117	0,003880340748860
0,39453125	0,109157056875320	0,105273325686140	0,003883731189180
0,40234375	0,101387988155528	0,097501103568962	0,003886884586566
0,41015625	0,093612731235511	0,089722930486961	0,003889800748550
0,41796875	0,085831760676878	0,081939281179732	0,003892479497145
0,42578125	0,078045551389966	0,074150630721111	0,003894920668855
0,43359375	0,070254578604859	0,066357454490177	0,003897124114682
0,44140625	0,062459317842379	0,058560228142240	0,003899089700140
0,44921875	0,054660244885065	0,050759427579807	0,003900817305258
0,45703125	0,046857835748133	0,042955528923539	0,003902306824595
0,46484375	0,039052566650421	0,035149008483186	0,003903558167235
0,47265625	0,031244913985325	0,027340342728520	0,003904571256805
0,48046875	0,023435354291722	0,019530008260252	0,003905346031470
0,48828125	0,015624364224883	0,011718481780940	0,003905882443943

6. Вывод

В ходе выполнения лабораторной работы был изучен и реализован метод быстрого преобразования Фурье для построения тригонометрической интерполяции функции.

В результате тестирования было проведено совпадение графика, построенного аналитически и графика по результатам вычисления узлов программой. Так же мы сравнили значения тригонометрической интерполяции в средних точках всех отрезков разбиения.