

Escola Politécnica da Universidade de São Paulo
MAP3121
Métodos Numéricos e Aplicações

Exercício Programa 2 - Análise Harmônica e Sinais Sonoros

Lucas Sorensen Paes - 9836919

Renato de Oliveira Freitas - 9837708

São Paulo
2018

1 Introdução

A análise de Fourier consiste:

- Análise da Função

Decompor diversas funções de variáveis reais em uma soma convergente com número finito ou infinito de componentes senoidais

- Síntese da Função

Construir uma função genérica $s(t)$ a partir da soma de componentes senoidais

Na engenharia elétrica, as funções $s(t)$ serão interpretadas como um sinal utilizado em grande parte das vezes para transmitir informação. A análise de Fourier em tempo contínuo pode ser feita através da Série de Fourier ou das integrais de Fourier. A Série se aplica sobretudo à sinais periódicos enquanto as integrais são para sinais quaisquer.

Esses sinais periódicos são decompostos na soma de componentes senoidais de frequências múltiplas de uma frequência fundamental. Essas componentes são chamadas de espectros do sinal. A transformada de Fourier aparece na matemática como um meio para facilitar a análise e síntese desses espectros, inclusive os não periódicos.

2 Objetivos

O exercício programa consiste de diferentes partes. Em um primeira parte o objetivo é implementar as seguintes transformadas, calcular os testes iniciais e comparar o tempo de execução de cada uma:

- a) Transformada Direta 'Lenta'

$$F(x_j) = \sum_{k=0}^{2N-1} F(x_j) e^{ikx}, j = 0, \dots, 2N - 1$$

- b) Transformada Inversa 'Lenta'

$$c_k = \frac{1}{2N} \sum_{j=0}^{2N-1} F(x_j) e^{-ikx}, j = 0, \dots, 2N - 1$$

- c) Transformada Direta/Inversa 'Rápida' (FFT)

É a forma recursiva da transformada, sendo que o algoritmo genérico foi nos dado. Bastava implementar em C.

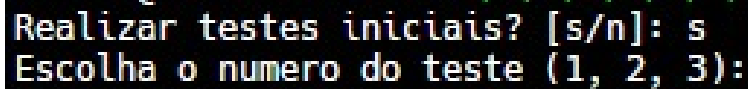
- d) Transformada de um pacote computacional pronto (FTTPACK4)

É uma biblioteca que possui a função pronta da Transformada direta e indireta, foi adicionada ao programa para que pudéssemos comparar o tempo de execução.

A segunda parte do exercício programa consiste em analisar um sinal sonoro qualquer e, a partir dos códigos das transformadas implementadas, conseguir os espectros do sinal estudado.

3 Execução do Programa

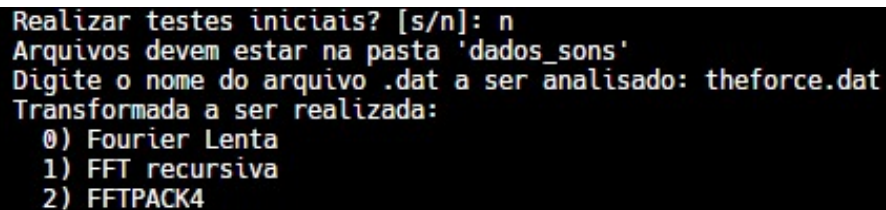
Ao executar o programa, o usuário será perguntado se quer realizar os testes iniciais propostos no enunciado do problema. Caso selecione a opção 's', deverá escolher o número do teste a ser realizado (1, 2 ou 3), de acordo com a ordem apresentada no enunciado.



```
Realizar testes iniciais? [s/n]: s
Escolha o numero do teste (1, 2, 3):
```

Figura 1: Decisão de qual teste será iniciado

Caso contrário, será pedido o arquivo no formato .dat a ser analisado, que deve estar na pasta "dados_sons", e depois escolher a opção da transformada a ser realizada. O usuário será informado se o arquivo for aberto com sucesso e será mostrado o número total de dados presentes nesse arquivo. Será realizada a transformada direta de acordo com a opção selecionada anteriormente, mostrando o tempo de execução.



```
Realizar testes iniciais? [s/n]: n
Arquivos devem estar na pasta 'dados_sons'
Digite o nome do arquivo .dat a ser analisado: theforce.dat
Transformada a ser realizada:
0) Fourier Lenta
1) FFT recursiva
2) FFTPACK4
```

Figura 2: Decisão de qual tipo de Transformada será realizada

4 Testes Iniciais e Verificação

- a) Considere a função com os seguintes valores em $0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$: $(5, -1, 3, 1)$.
O resultado da transformada direta deve ser: $c = (2, \frac{1}{2} + \frac{i}{2}, 2, \frac{1}{2} - \frac{i}{2})$.
Aplicando a transformada inversa deve-se recuperar os valores da função original.

```
Realizar testes iniciais? [s/n]: s
Escolha o numero do teste (1, 2, 3): 1

Verificacao dos algoritmos implementados
F = (5, -1, 3, 1)

Transformada direta Lenta:
vetor:
  2.00    0.00i
  0.50    0.50i
  2.00    0.00i
  0.50   -0.50i

Transformada direta Rápida:
vetor:
  2.00    0.00i
  0.50    0.50i
  2.00    0.00i
  0.50   -0.50i

Transformada Inversa Lenta:
vetor:
  5.00    0.00i
 -1.00    0.00i
  3.00   -0.00i
  1.00    0.00i

Transformada inversa Rápida:
vetor:
  5.00    0.00i
 -1.00    0.00i
  3.00    0.00i
  1.00   -0.00i
```

Figura 3: Resposta da questão A

- b) Considere a função com os seguintes valores $f = (6, 2, 5, 2, 11, 2, 8, 8)$ e compare seus códigos com as rotinas do FFTPACK4.

```
Realizar testes iniciais? [s/n]: s
Escolha o numero do teste (1, 2, 3): 2

Verificacao dos algoritmos implementados
F = (6, 2, 5, 2, 11, 2, 8, 8)
FFT direta:
Tempo = 0.000000
vetor:
    1.38      0.00i
   -0.02      0.23i
    0.12      0.19i
   -0.29      0.04i
    0.50      0.00i
   -0.29     -0.04i
    0.12     -0.19i
   -0.02     -0.23i

FFT inversa nos valores anteriores normalizados:
vetor:
    1.50      0.00i
    0.50     -0.00i
    1.25      0.00i
    0.50     -0.00i
    2.75      0.00i
    0.50      0.00i
    2.00     -0.00i
    2.00      0.00i
```

Figura 4: Resposta da questão B

5 Análise dos dados fornecidos

A segunda parte do exercício consistia na análise de diferentes sons. Tais sons eram transformados do formato .wav em arquivos de texto (.txt) a partir do software SOX, explicitado na apresentação do EP. Após feita a análise de acordo com a opção escolhida, dois arquivos de saída serão gerados: um com os coeficientes da transformada direta calculado e outro com o resultado da transformada inversa desses coeficientes para comparação com os valores originais.

```
Realizar testes iniciais? [s/n]: n
Arquivos devem estar na pasta 'dados_sons'
Digite o nome do arquivo .dat a ser analisado: dog.dat
Transformada a ser realizada:
  0) Fourier Lenta
  1) FFT recursiva
  2) FFTPACK4
0
Total de dados no arquivo: 28930.0
Arquivo ./dados_sons/dog.dat aberto, fazer a transformada lenta
Tempo de execucao: 76.328125 segundos
Tempo de execucao: 75.796875 segundos
Arquivos de saida gerados.
```

Figura 5: Análise arquivo dog.dat utilizando a transformada lenta

```
Realizar testes iniciais? [s/n]: n
Arquivos devem estar na pasta 'dados_sons'
Digite o nome do arquivo .dat a ser analisado: dog.dat
Transformada a ser realizada:
  0) Fourier Lenta
  1) FFT recursiva
  2) FFTPACK4
1
Total de dados no arquivo: 28930.0
Arquivo ./dados_sons/dog.dat aberto, fazer FFT recursiva
Tempo de execucao: 0.062500 segundos
Tempo de execucao: 0.062500 segundos
Arquivos de saida gerados.
```

Figura 6: Análise arquivo dog.dat utilizando FFT recursiva

Analisando os tempos de execução, podemos ver o quão ineficiente é a implementação do primeiro algoritmo ("Fourier Lenta"), comparado com o algoritmo recursivo.

1	1.29985621e+00	-6.61529271e-03
2	7.68682524e-05	4.13630727e-01
3	1.31681475e-04	2.07116455e-01
4	-1.02839209e-04	1.38323120e-01
5	-6.19550600e-05	1.03126554e-01
6	-5.81997844e-06	8.31251155e-02
7	-6.47686448e-05	6.90141405e-02
8	1.97845444e-04	5.92151739e-02
9	1.05106299e-04	5.17542786e-02
10	-9.23401034e-05	4.58611227e-02
11	-3.43425975e-05	4.13322025e-02
12	-1.24021197e-04	3.75537286e-02
13	-9.35151185e-05	3.44689568e-02
14	-4.91807552e-05	3.17237561e-02
15	-5.02762752e-05	2.94067155e-02
16	-1.75345218e-04	2.75934146e-02
17	-2.56179678e-04	2.58271487e-02
18	-1.46058385e-04	2.43597046e-02
19	-1.34656856e-04	2.30425844e-02
20	-1.99590929e-04	2.18592260e-02
21	-8.96396724e-05	2.06845501e-02
22	-1.87491727e-04	1.98666469e-02
23	-1.41462228e-04	1.89349304e-02
24	-2.44531751e-05	1.81266825e-02
25	4.03409472e-05	1.74213049e-02

Figura 7: Primeiros 25 coeficientes calculados do arquivo dog.dat (arquivo saida_coef.txt)

1	3.38152115e-11	1.33870692e-12
2	8.98715154e-05	-6.40143494e-12
3	1.79743001e-04	-4.11387591e-12
4	2.69614472e-04	-1.19567134e-11
5	3.59485963e-04	-7.75857156e-12
6	4.49357436e-04	-1.05209175e-11
7	5.39228911e-04	-3.31457084e-13
8	6.29100409e-04	2.33457698e-12
9	7.18971888e-04	-3.05794279e-12
10	8.08843372e-04	3.47194495e-12
11	8.98714861e-04	-2.33368880e-13
12	9.88586348e-04	-7.92821364e-12
13	1.07845781e-03	-1.15483179e-11
14	1.16832931e-03	-2.16204832e-12
15	1.25820082e-03	-1.51290092e-12
16	1.34807232e-03	-5.23087129e-12
17	1.43794372e-03	-4.93849406e-12
18	1.52781522e-03	-6.64612809e-12
19	1.61768672e-03	-7.34257100e-12
20	1.70755821e-03	-8.59762261e-12
21	1.79742970e-03	-2.04425366e-12
22	1.88730121e-03	4.06497058e-12
23	1.97717262e-03	-8.83682016e-13
24	2.06704412e-03	3.32572858e-12
25	2.15691562e-03	-5.78431747e-12

Figura 8: Transformada inversa dos coeficientes calculados

3							0	0
4							8.9871484e-005	0
5							0.00017974297	0
6							0.00026961445	0
7							0.00035948594	0
8							0.00044935742	0
9							0.0005392289	0
10							0.00062910039	0
11							0.00071897187	0
12							0.00080884335	0
13							0.00089871484	0
14							0.00098858632	0
15							0.0010784578	0
16							0.0011683293	0
17							0.0012582008	0
18							0.0013480723	0
19							0.0014379437	0
20							0.0015278152	0
21							0.0016176867	0
22							0.0017075582	0
23							0.0017974297	0
24							0.0018873012	0
25							0.0019771726	0
26							0.0020670441	0
27							0.0021569156	0

Figura 9: Arquivo original dog.dat

Comparando o resultado da transformada inversa com o arquivo original e aproximando os números que são $\leq 10^{-12}$ para zero, verificamos o funcionamento correto dos algoritmos implementados.