ANÁLISE ESPECTRAL POR TRANSFORMADAS DE FOURIER

Edgard Macena Cabral Nº 11820833

Março 2023

Edgard Macena Cabral

Março 2023

#+latex$_{\text{header extra}}$ pdfpages

Abstract

Nessa tarefa, buscamos analisar a Transformada Discreta de Fourier sem nenhuma aceleração. Criamos um programa que realiza a transformada e outro que realiza sua inversa. Observamos que, para alguns casos, essas operações são bem comportadas, e conseguimos recuperar o sinal sem dificuldades. Observamos também que, para outras sinais obtivemos um reflexo em torno da frequência de Nyquist, provando que a maior frequência que podemos obter com a transformada é $\frac{1}{2\Delta t}$. Por fim, vimos a relação entre o número N de pontos e o tempo de 50 execuções do programa.

Introdução

A transformada de Fourier é uma transformação que nos leva do espaço das amplitude de um sinal para um espaço de frequências. Para sinais contínuos, podemos escrever:

$$Y(f) = \int_{-\infty}^{\infty} y(t) e^{2\pi f i t} \tag{1}$$

Porém, na física experimental, da análise da luminosidade de estrelas na busca de exoplanetas ao estudo do movimento de um pêndulo no laboratório de física I, trabalhamos essencialmente com dados discretos.

Para adequar a transformada aos dados laboratoriais, usamos as substituições $t_j = j\Delta t$, $f_k = \frac{k}{N\Delta t}$, que nos dão:

$$Y_k = \sum_{j=0}^{j<N/2} y_j e^{2\pi jki/N} \tag{2}$$

Conseguimos também obter a inversa através de

$$y_j = \frac{1}{k} \sum_{k=0}^{k<N/2} Y_k e^{-2\pi jki/N} \tag{3}$$

Nessas equações, a razão de irmos apenas até $k < N/2$ está ligada a frequência de Nyquist, que pode ser entendida como a frequência associada ao menor comprimento de onda que pode formar um modo fundamental entre pontos de data consecutivos.

Caso a frequência do sinal seja maior que a frequência de Nyquist, o gráfico, obtemos um pico relacionada à reflexão da verdadeira frequência do sinal com a de Nyquist.

$$f_{encontrada} = f_{verdadeira} - f_{Nyquist} \tag{4}$$

Como já dito, transformadas de Fourier são amplamente usadas na Física, então é importante que possamos tê-la eficientemente. Podemos estimar com facilidade a ordem do tempo de execução. Temos N termos, cada um dos quais é calculado com uma soma de $\frac{N}{2}$ termos. A ordem do tempo de execução é então da ordem de $\tilde{} N^2$.

Geração de dados

Para gerar os dados, usamos a equação a seguir:

$$y_i = a_1 \cos(\omega_1 t_i) + a_2 \sin(\omega_2 t_i) \tag{5}$$

$$t_i = i\Delta t, \quad i = 1, \cdots, N$$

Que foi executada no programa:

```
program gerarSinais
    implicit none
    real*8, parameter :: pi = 3.1415926537989
```

```
    real*8 :: a1, a2, w1, w2, dt
    integer :: N
    ! (a) N = 200, t = 0.04, a1 = 2, a2 = 4, 1 = 4Hz, 2 = 2.5Hz
    ! (b) N = 200, t = 0.04, a1 = 3, a2 = 2, 1 = 4Hz, 2 = 2.5Hz
    ! (c) N = 200, t = 0.4, a1 = 2, a2 = 4, 1 = 4Hz, 2 = 0.2Hz
    ! (d) N = 200, t = 0.4, a1 = 3, a2 = 2 1 = 4Hz, 2 = 0.2Hz

    call escreveSinal(200, 0.04d0, 2.d0, 4.d0, 4.d0*pi, 2.5d0, "a")
    call escreveSinal(200, 0.04d0, 3.d0, 2.d0, 4.d0*pi, 2.5d0, "b")
    call escreveSinal(200, 0.4d0, 2.d0, 4.d0, 4.d0*pi, 0.2d0, "c")
    call escreveSinal(200, 0.4d0, 3.d0, 2.d0, 4.d0*pi, 0.2d0, "d")
end program gerarSinais


subroutine escreveSinal(N, dt, a1,  a2, w1, w2, label)
    real*8, intent(in) :: a1, a2, w1, w2, dt
    integer, intent(in) :: N
    char*8, intent(in) :: label
    real*8 :: calculaSinal
    integer :: i

    open(1, file="saida-2-"//label)

    do i = 1, N
        write(1, *) i*dt, calculaSinal(i*dt, a1, a2, w1, w2)
    end do

    close(1)

end subroutine escreveSinal


function calculaSinal(t, a1, a2, w1, w2) result(retval)
    real*8, intent(in) :: a1, a2, w1, w2, t
    real*8 :: retval
    retval = a1*cos(w1*t) + a2*sin(w2*t)

end function calculaSinal
```

O programa foi executado sobre as séries de parâmetros

(a) $N = 200, \Delta t = 0,04,\ a_1 = 2,\ a_2 = 4,\ \omega_1 = 4\pi Hz,\ \omega_2 = 2,5\pi Hz$

(b) $N = 200, \Delta t = 0,04,\ a_1 = 3,\ a_2 = 2, \omega_1 = 4\pi Hz,\ \omega_2 = 2,5\pi Hz$

(c) $N = 200, \Delta t = 0,4,\ a_1 = 2,\ a_2 = 4,\ \omega_1 = 4\pi Hz,\ \omega_2 = 0,2\pi Hz$

(d) $N = 200, \Delta t = 0,4,\ a1 = 2,\ a2 = 4,1 = 4\pi Hz,\ 2 = 2,5\pi Hz$

Os resultados estão no gráfico a seguir
Sinais gerados para os testes seguintes.

Transformada de Fourier

Para a Transformada de Fourier, usamos o programa a seguir:

```
program gerarEspacoFrequencias
    implicit none
    real*8, dimension(200) :: y_t
    real*8 :: dt
    character :: label
    integer :: N
    read(*,*) label
    call leTabela(label, dt, y_t, N)
    call escreveFrequencias(y_t, dt, label, N)


end program gerarEspacoFrequencias

subroutine leTabela(label, dt, y_t, N)
    character, intent(in) :: label
    real*8, dimension(200), intent(out) :: y_t
    real*8, intent(out) :: dt
    integer, intent(out) :: N
    real*8 :: ignorada
    integer :: i

    open(1, file="data.in")

    do i = 1, 200
        read(1,*, end=10) ignorada, y_t(i)
```

```
        if ( i == 1 ) then
            dt = ignorada
        end if
10  end do
    close(1)

    N = i

end subroutine leTabela

subroutine escreveFrequencias(y_t, dt, label, N)
    real*8, dimension(200), intent(in) :: y_t
    real*8, intent(in) :: dt
    character :: label
    integer :: k, N, M
    complex*16 :: Yk, currYk
    M = floor((N-1)/2.d0)

    open(2, file="data.out")
    do k = 1, M
        currYk = Yk(k, y_t, N)
        write(1,*) k/(200*dt), real(currYk), aimag(currYk)
    end do
    close(2)
end subroutine escreveFrequencias

function Yk(k, y_t, N)
    integer, intent(in) :: k
    integer, intent(in) :: N
    real*8, dimension(200):: y_t
    complex*16 :: Yk, i = (0,1)
    real*8, parameter :: pi = 3.1415926537989
    integer :: j
    Yk = (0,0)
    somatoria : do  j = 1, N
        Yk = Yk + y_t(j)*exp(2.d0*pi*i*j*k/N)
    end do somatoria

end function Yk
```