Física estatística computacional

Análise espectral por transformadas de Fourier

Aluno
Leonardo Vaz Ferreira n° 13862330

São Carlos
2025

1 Tarefa 1

Nesta primeira tarefa, o objetivo é criar um programa em Fortran que faça o cálculo da transformada discreta de Fourier, para a seguinte equação de onde:

$$a_1 cos(\omega_1 t_i) + a_2 sen(\omega_2 t_i), t_i = i\Delta t, i = 1, ..., N \tag{1}$$

```
program TF
implicit real*8 (a-h, o-z)
complex*16:: X

open(unit = 10, file = "data.in")
open(unit = 20, file = "real.out")
open(unit = 30, file = "img.out")
open(unit = 40, file = "value_serie.out")

read(10,*)t, a1, a2, w1, w2

n = 200

PI = acos(-1.0d0)

do i = 0,n-1
    X = (0.0d0, 0.0d0)

    do j = 0,n-1
        y = cmplx(a1*cos(w1*PI*j*t) + a2*sin(w2*PI*j*t), 0.0d0)

        X = X + y * exp(-2*PI*(0.0d0, 1.0d0)*dble(i)*dble(j) / dble(n))
    end do

    write(20,*)t*i, real(X)
    write(30,*)t*i, aimag(X)
    write(40,*)t*i, a1*cos(w1*PI*i*t) + a2*sin(w2*PI*i*t)
end do

close(10)
close(20)

end program
```

Figura 1: Código da primeira tarefa

O código da Fig.1 calcula a transformada discreta de Fourier; para isso, são iterados os coeficientes das frequências de tal maneira que:

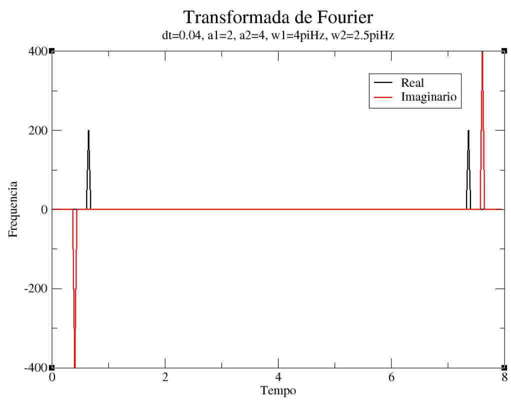
$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-i\pi kn}{N}} \tag{2}$$

Onde n representa a iteração temporal e k a iteração das frequências da transformada.

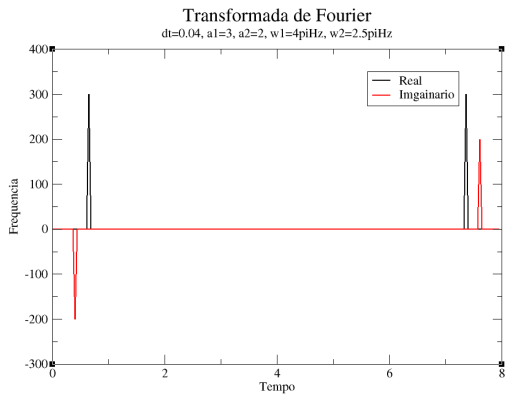
2 Tarefa 2

Com o código da Fig.1 podemos testar para diferentes valores para as variáveis, como:

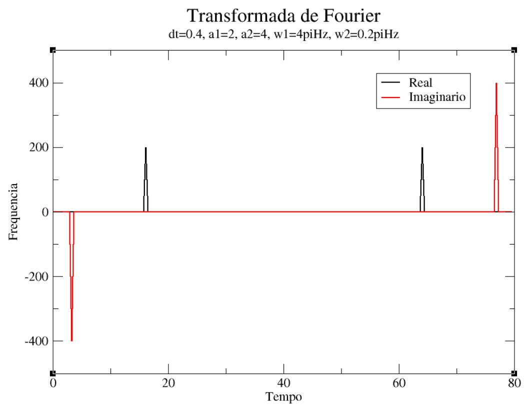
$$
\begin{gathered}
a)N = 200, \Delta t = 0.04, a_1 = 2, a_2 = 4, \omega_1 = 4\pi HZ, w_2 = 2.5\pi HZ \\
b)N = 200, \Delta t = 0.04, a_1 = 3, a_2 = 2, \omega_1 = 4\pi HZ, w_2 = 2.5\pi HZ \\
c)N = 200, \Delta t = 0.4, a_1 = 2, a_2 = 4, \omega_1 = 4\pi HZ, w_2 = 0.2\pi HZ \\
d)N = 200, \Delta t = 0.4, a_1 = 3, a_2 = 2, \omega_1 = 4\pi HZ, w_2 = 2.5\pi HZ
\end{gathered}
$$



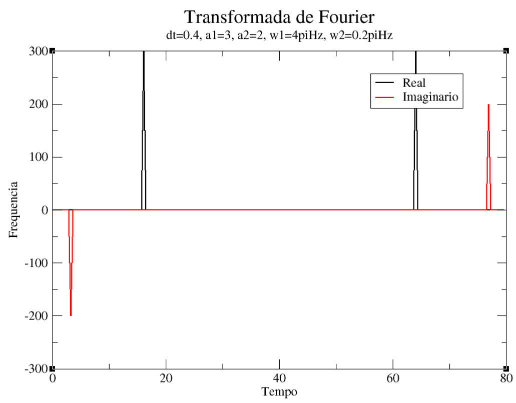
(a) $\Delta t = 0.04, a_1 = 2, a_2 = 4, \omega_1 = 4\pi Hz, \omega_2 = 2.5\pi Hz$



(b) $\Delta t = 0.04, a_1 = 3, a_2 = 2, \omega_1 = 4\pi Hz, \omega_2 = 2.5\pi Hz$



(c) $\Delta t = 0.4, a_1 = 2, a_2 = 4, \omega_1 = 4\pi Hz, \omega_2 = 0.2\pi Hz$



(d) $\Delta t = 0.4, a_1 = 2, a_2 = 4, \omega_1 = 4\pi Hz, \omega_2 = 0.2\pi Hz$

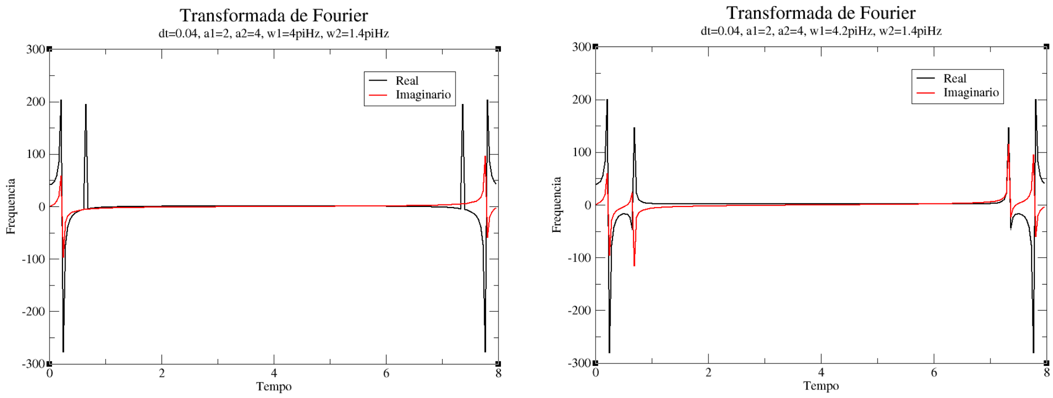
Figura 2: Frequências ao longo do tempo da transformada de Fourier

Podemos notar alguns pontos na Fig.2, primeiramente a simetria da parte real da transformada, assim como esperado, juntamente com a parte imaginária apresentando uma antissimetria, onde o primeiro pulso se apresenta negativo. Podemos notar também que com a variação de alguns parâmetretro, como por exemplo, nos itens (a) e (b), com a variação da amplitude, podemos notar o aumento no ponto de máximo na parte real da transformada e uma diminuição na parte imaginária, semelhante quando comparamos (c) com (d). Já quando variamos o parâmetro temporal Δt, podemos notar um atraso no sinal, como apresentado, de maneira que nas imagens (a) e (b) o sinal aparece próximo da origem, já para (c) e (d) aparece com certo atraso.

3 Tarefa 3

Nesta seção iremos observar algumas variações no comportamento nos sinais das transformadas, utilizando o mesmo código da Fig.1 trabalharemos com as seguintes variáveis na função de onda:

$$e)N = 200, \Delta t = 0.04, a_1 = 2, a_2 = 4, \omega_1 = 4\pi HZ, w_2 = 1.4\pi HZ$$
$$f)N = 200, \Delta t = 0.04, a_1 = 4, a_2 = 4, \omega_1 = 4.2\pi HZ, w_2 = 1.4\pi HZ$$



(a) $\Delta t = 0.04, a_1 = 2, a_2 = 4, \omega_1 = 4\pi Hz, \omega_2 = 1.4\pi Hz$ (b) $\Delta t = 0.04, a_1 = 2, a_2 = 4, \omega_1 = 4.2\pi Hz, \omega_2 = 1.4\pi Hz$

Figura 3: Frequências ao longo do tempo da transformada de Fourier

Podemos observar que com a variação para um ω específico, conseguimos não só um pico no aspectro das frequências. Podemos observar, um comportamento muito diferente das anteriores, onde o aspectro real apresenta um pico no começo e inverte para a parte negativa rapidamente, assim como também a parte imaginária do aspectro.

4 Tarefa 4

Agora, para verificar a consistência dos resultados obtidos, podemos realizar a transformada inversa de Fourier.

$$x_k = \frac{1}{N} \sum_{n=0}^{N-1} X_n e^{\frac{i\pi kn}{N}} \tag{3}$$

```
program TIF
implicit real*8 (a-h, o-z)
complex*16:: X(0:199), x_n
dimension t(0:199), f_real(0:199), f_img(0:199)

open(unit = 10, file = "real.in")
open(unit = 20, file = "img.in")
open(unit = 30, file = "serie.out")

PI = acos(-1.0d0)
n = 200

do k = 0, n-1
    read(10,*) t(k), f_real(k)
    read(20,*) t(k), f_img(k)
    X(k) = cmplx(f_real(k), f_img(k))
end do

do j = 0, n-1
    x_n = cmplx(0.0d0, 0.0d0)
    do k = 0, n-1
        x_n = x_n + X(k) * exp(2.0d0 * PI * (0.0d0, 1.0d0) * dble(k) * dble(j) / dble(n))
    end do
    x_n = x_n / dble(n)
    write(30,*) t(j), real(x_n)
end do

close(10)
close(20)
close(30)

end program
```

Figura 4: Código da quarta tarefa

Com o código da Fig.4 e utilizando os dados obtidos nos passos anteriores, podemos realizar o processo inverso e calcular a função de onda, a partir das transformadas.

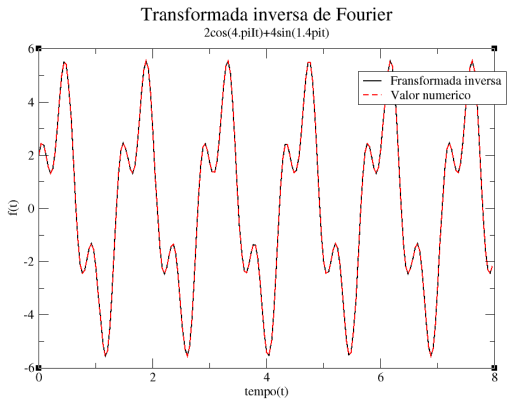


Figura 5: Transformada inversa de Fourier

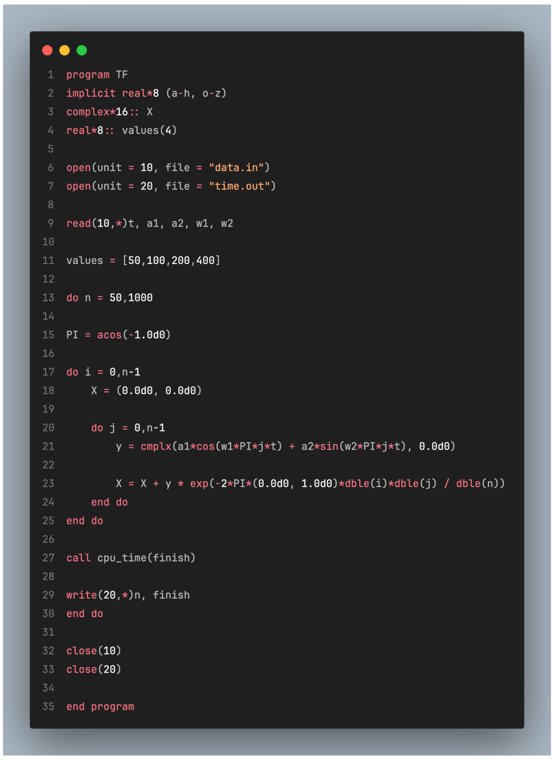
A partir dos dados obtidos pela transformada da função:

$$2cos(4\pi t) + 4sin(1.4\pi t) \tag{4}$$

Obtivemos a função da Fig.5 e podemos notar que o valor obtido a partir da transforma condiz muito bem com o valor original da onda.

5 Tarefa 5

Nessa parte iremos observar o tempo de processamento das transformadas e notar o comportamento com o aumento do parâmetro N.



```fortran
program TF
implicit real*8 (a-h, o-z)
complex*16:: X
real*8:: values(4)

open(unit = 10, file = "data.in")
open(unit = 20, file = "time.out")

read(10,*)t, a1, a2, w1, w2

values = [50,100,200,400]

do n = 50,1000

PI = acos(-1.0d0)

do i = 0,n-1
    X = (0.0d0, 0.0d0)

    do j = 0,n-1
        y = cmplx(a1*cos(w1*PI*j*t) + a2*sin(w2*PI*j*t), 0.0d0)

        X = X + y * exp(-2*PI*(0.0d0, 1.0d0)*dble(i)*dble(j) / dble(n))
    end do
end do

call cpu_time(finish)

write(20,*)n, finish
end do

close(10)
close(20)

end program
```

Figura 6: Código da quinta tarefa

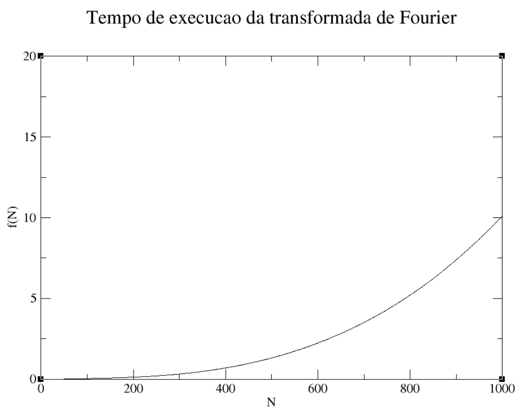


Figura 7: Função do tempo de execução com a variação de N

Com a utilização do código da Fig.6, foi possível observar o comportamento do tempo de execução com a variação de N. Dessa maneira, segundo a Fig.7, podemos notar que, com a variação de N o tempo de execução aumenta cresce proporcionalmente com N^2, assim como o esperado.