UNIVERSIDADE DE SÃO PAULO
Instituto de Física de São Carlos

Projeto 1:ANÁLISE ESPECTRAL POR TRANSFORMADAS DE FOURIERS

Anderson Araujo de Oliveira 11371311

2020

Conteúdo

1 Transformada de Fourier

A transformada de Fourier decompõe uma função temporal em frequência, tal como sinal pode ser composta por diversas ondas de amplitudes diferentes quando aplicada transformada desassociando todas ondas mostrando quais as a compõe no sinal.A transformada de Fourier de uma função temporal é uma função de valor complexo da frequência, cujo valor absoluto representa a soma das frequências presente na função original e cujo argumento complexo é a fase de deslocamento da função trigonométrica naquela frequência.

A Transformada de Fourier é mais vista em sua forma trigonométrica

$$F(\omega) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} F(t)e^{i\omega t}dt \tag{1}$$

Mas transformada de Fourier que utilizamos que receber uma normalização

$$f(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(t)e^{i\omega t}dt \tag{2}$$

Podemos achar a inversa

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{-i\omega t}d\omega \tag{3}$$

1.1 Transformada discreta de Fourier

Transformada discreta de Fourier é utilizado para uso computacional.

O N representa a quantidade de coeficientes de fourier gerados da função que será aplicada a transformada de fourier.

$$F(j) = \frac{1}{N}\sum_{k=0}^{N-1} f(k)e^{\frac{-2\pi ijk}{N}} \tag{4}$$

$$F(k) = \sum_{j=0}^{N-1} f(j)e^{\frac{2\pi ijk}{N}} \tag{5}$$

1.2 Código

```
program FT
  implicit real*8 (a-h,o-z)
  real*8, dimension(100000) :: coef
  real*8,dimension(6)::dt,a1,a2,w1,w2
  real*8::imag1
  character::nome(18)*23
  real*4:: t1,delta!para aceitar o secnds
```

```
data nome/"saida1-a-11371311.dat","saida2-a-11371311.dat","saida3-a-11371311.dat","saida4-a-11371311.dat",&!os primeiros 6 vetores est o guardando os coeficiente de fourier
"saida5-a-11371311.dat","saida6-a-11371311.dat","saida7-a-11371311.dat","saida8-a-11371311.dat",&!os 6 seguintes vetores estao armazenando os valres reais da transformada de fourier
"saida9-a-11371311.dat","saida10-a-11371311.dat","saida11-a-11371311.dat","saida12-a-11371311.dat",&!os ultimos 6 arquivos estao armazenando os valores imaginario da transformada de fourier
"saida13-a-11371311.dat","saida14-a-11371311.dat","saida15-a-11371311.dat","saida16-a-11371311.dat",&
"saida17-a-11371311.dat","saida18-a-11371311.dat"/
!valor de pi
t1 = secnds(0.0)
pi=dacos(-1d0)
N=200
!coeficientes de fourier
do l=1,6

  open((l+10),file=nome(l))
  open((l+16),file=nome(l+6))
  open((l+22),file=nome(l+12))
  dt = (/0.04d0,0.04d0,0.4d0,0.4d0,0.04d0,0.04d0/)
  a1 = (/2d0,3d0,2d0,3d0,2d0,2d0/)
  a2 = (/4d0,2d0,4d0,2d0,4d0,4d0/)
  w1 = (/4d0*pi,4d0*pi,4d0*pi,4d0*pi,4d0*pi,4.2d0*pi/)
  w2 = (/2.5d0*pi,2.5d0*pi,0.2d0*pi,0.2d0*pi,1.4d0*pi,1.4d0*pi/)

  do i=1,N

    t=dt(l)*i
    y=a1(l)*dcos(w1(l)*t)+a2(l)*dsin(w2(l)*t)
    coef(i)=y!salvando os coeficientes de fourier
    write((l+10),*)t,y

  enddo
  !transformada de fourier
  do k=1,N

      real1=0
      imag1=0
      !soma de fourier

    do j=1,N

        real1=real1+coef(j)*dcos(2d0*pi*j/N*k)
        imag1=imag1+coef(j)*dsin(2d0*pi*j/N*k)

    enddo

      !saida de dados
      f=k/(N*dt(l))
      write((l+16),*)f,real1
      write((l+22),*)f,imag1
```

```
    enddo

  enddo

  delta = secnds(t1)
  write(*,*)(delta)
end program
```

1.3 Parte 2

Nessa parte do projeto vamos achar os coeficiente de fourier de uma função.

$$f(t) = a\cos\omega_1 t + b\sin\omega_2 t \tag{6}$$

Vamos usar a função (6) e obter os valores em determinados pontos do tempo pegando 200 pontos em diferentes tempos de variação, assim conseguimos reescrever $t = i\Delta t$, $i \in \mathbb{Z}$:[0,200].

Assim conseguimos reproduzir o seguintes gráficos para utilizando os dados da tabela abaixo como parâmetros da função (6).

i	a	b	ω_1	ω_2	Δt
(a)	2	4	4π	2.5π	0.04
(b)	3	2	4π	2.5π	0.04
(c)	2	4	4π	2.5π	0.4
(d)	3	2	4π	2.5π	0.4

Tabela 1: Valores usando na diferentes ondas da função (6)

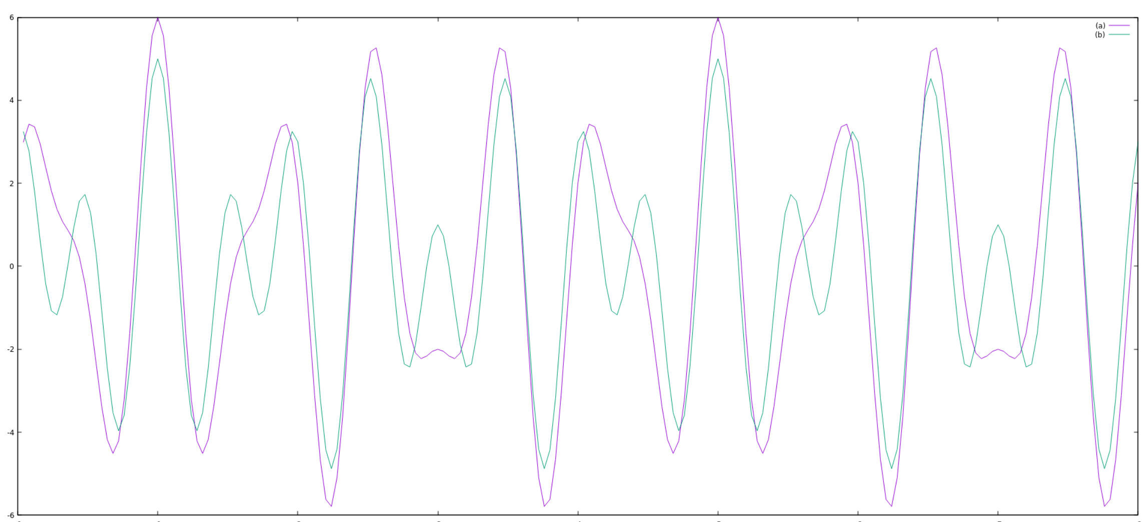

Figura 1: Nesse gráfico temos (a) e (b)

Fazendo mudanças nas amplitudes das ondas acaba ocorrendo mais ondulações, por causa que as ondas ficam quase a mesma amplitude.

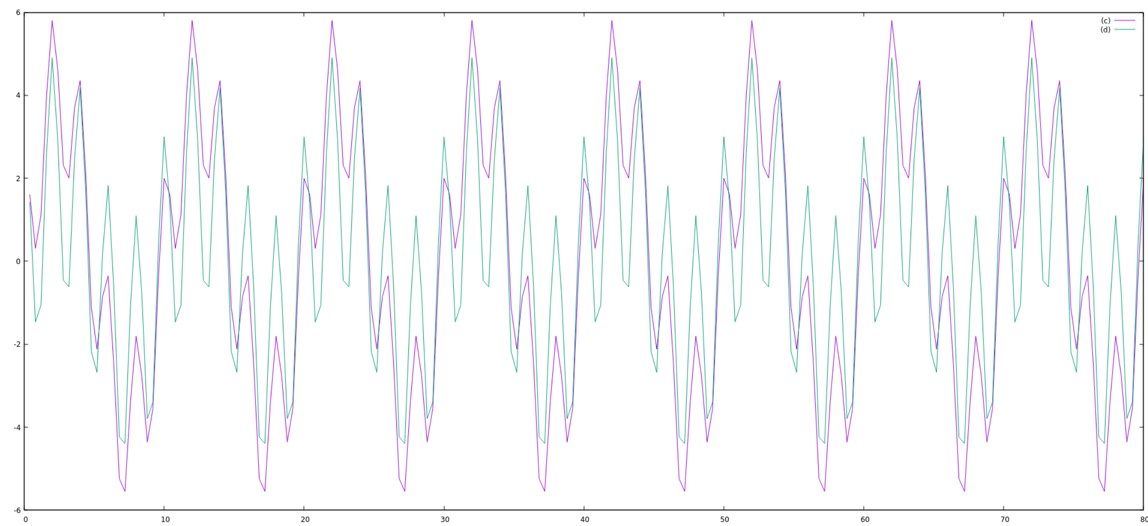

Figura 2: Nesse gráfico temos (c) e (d)

No gráfico vemos que a onda não está tão suave por causa que os pontos que obtemos ficam muito longe uma da outra, assim causando um gráfico não tão suave.

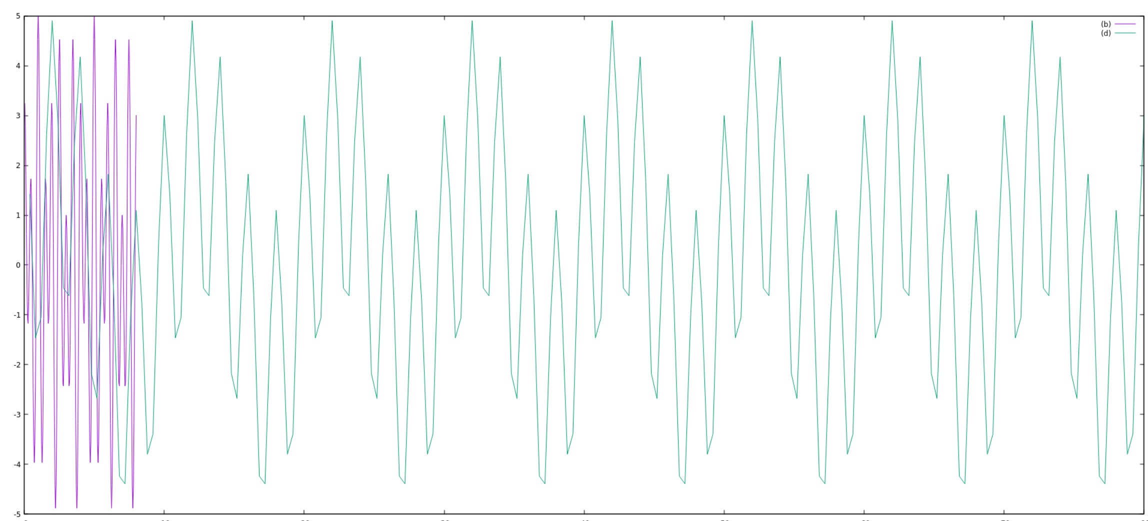

Figura 3: Nesse gráfico temos (b) e (d)

Vemos que ordem de tempo entre os ondas foram muito diferente onde podemos ver que uma percorreu um tempo 10 vezes maior que outra.

1.4 Parte 3

Nessa parte vamos pegar os dados que obtivemos fazendo o gráfico (a) e (c), colocando como coeficientes de Fourier na função (5), mas teremos que abrir a identidade de Euler para pegar a parte real e imaginaria separadamente.

$$Re(F(k)) = \sum_{j=0}^{N-1} f(j) \cos \frac{2\pi ijk}{N} \tag{7}$$

$$Im(F(k)) = \sum_{j=0}^{N-1} f(j) \sin \frac{2\pi ijk}{N} \tag{8}$$

Vamos analisar usando a transformada de Fourier mais duas ondas (e) e (f) que estão na tabela abaixo.Para verificamos que a transformada está funcionando vamos verificar se as frequências que aparecerem no gráfico são idênticas as das ondas originais, para obter esse dados temos que pegar a parte real da transformada.

i	a	b	ω_1	ω_2	Δt
(e)	2	4	4π	1.4π	0.04
(f)	2	2	4.2π	1.4π	0.04

Tabela 2: Valores usados na diferentes ondas da (6) para analise da transformada de Fourier na parte 3

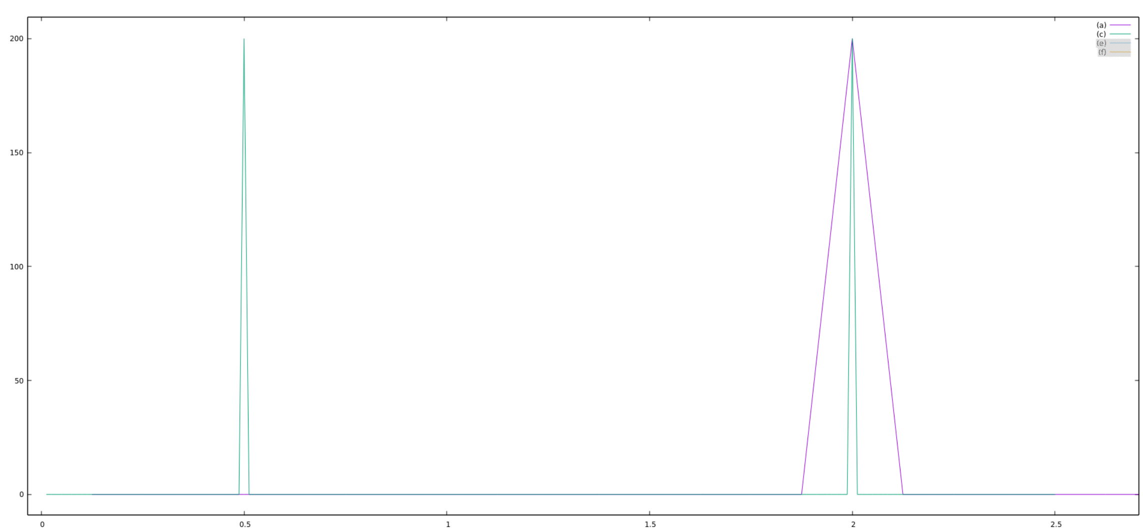

Figura 4: Nesse gráfico temos (a) e (c) da parte anterior

$$f = \frac{\omega}{2\pi} \tag{9}$$

Vamos encontra a frequência da onda, usando (9), chegamos no seguintes resultados para ω_1.

$$f = \frac{4\pi}{2\pi} = 2Hz \tag{10}$$

A frequência de ω_1 está de acordo o gráfico 4 no ponto $2Hz$.

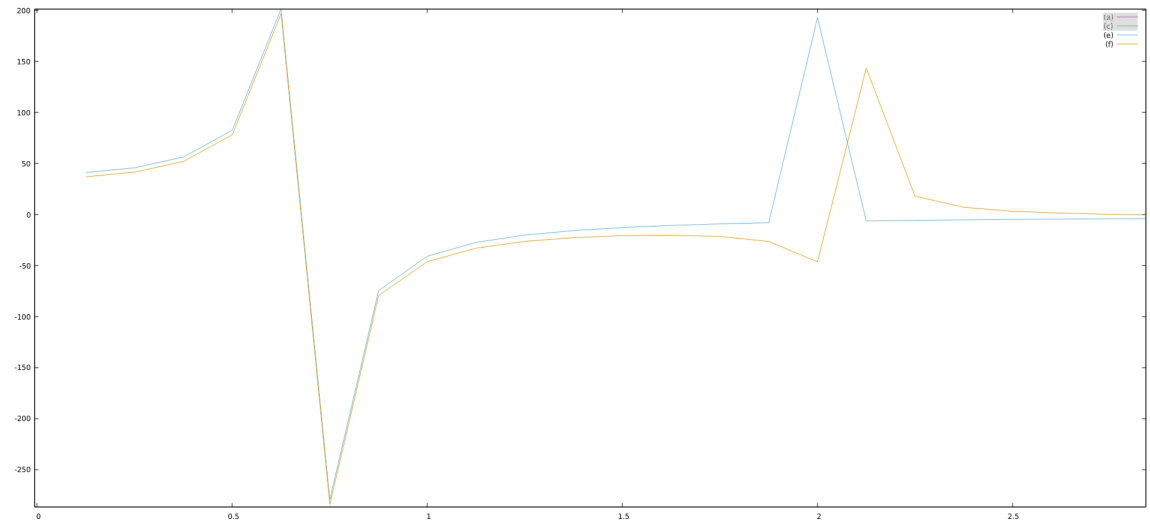

Figura 5: Nesse gráfico temos (e) e (f)

Usando o mesma mesma formula para achar frequência vamos utilizar nessa parte para encontrar $\omega_1 e \omega_2$ de (e) e (f).

Para (e).

$$f = \frac{\omega_1}{2\pi} = \frac{4\pi}{2\pi} = 2Hz \tag{11}$$

$$f = \frac{\omega_2}{2\pi} = \frac{1.4\pi}{2\pi} = 0.7Hz \tag{12}$$

Para (f).

$$f = \frac{\omega_1}{2\pi} = \frac{4.2\pi}{2\pi} = 2.1Hz \tag{13}$$

Assim podemos confirmar que os números obtivemos teoricamente se coincidem com a transformada de fourier do gráfico 5.

2 Transformada-inversa de Fourier

Para utilizar transformada inversa de Fourier temos que utilizar os coeficiente gerados pelas transformadas anteriores substituir na (4) fazendo a distributiva dos valores complexos que são definidas assim $z = a + bi$, chegamos em:

$$F(j) = \frac{1}{N}\sum_{k=0}^{N-1}(a(k) + b(k)i)e^{\frac{-2\pi ijk}{N}} \tag{14}$$

Abrindo a identidade de Euler para definição trigonométrica.

$$F(j) = \frac{1}{N}\sum_{k=0}^{N-1}(a(k) + b(k)i)(\cos\frac{-2\pi jk}{N} + i\sin\frac{-2\pi jk}{N}) \tag{15}$$

Fazendo a distributiva.

$$F(j) = \frac{1}{N}\sum_{k=0}^{N-1}(a(k)\cos\frac{-2\pi jk}{N} + ib(k)\cos\frac{-2\pi jk}{N} + ia(k)\sin\frac{-2\pi jk}{N} - b(k)\sin\frac{-2\pi jk}{N}) \tag{16}$$

Conseguimos nossa somatória em duas diferente uma para real e outra para o imaginário.

$$Re(F(j)) = \frac{1}{N}\sum_{k=0}^{N-1}(a(k)\cos\frac{-2\pi jk}{N} - b(k)\sin\frac{-2\pi jk}{N}) \tag{17}$$

$$Im(F(j)) = \frac{1}{N}\sum_{k=0}^{N-1}(b(k)\cos\frac{-2\pi jk}{N} + a(k)\sin\frac{-2\pi jk}{N}) \tag{18}$$

2.1 Código

```
program FIT
  implicit real*8 (a-h,o-z)
   real*8, dimension(200) :: valorR,valorI
   real*4:: t1,delta!para aceitar o secnds

  !coeficientes de fourier
  !valor de pi
  open(1,file='saida5-a-11371311.dat',status='old')
  open(2,file='saida9-a-11371311.dat',status='old')
  open(3,file='saida13-a-11371311.dat',status='replace')
  open(4,file='saida14-a-11371311.dat',status='replace')


  t1=0
  t1 = secnds(0.0)
  pi=dacos(-1d0)
  N=200
  do i=1,N
   read(1,*)p,valorR(i)
   read(2,*)h,valorI(i)
  enddo
  close(1)
```

```
  close(2)


  !transformada inversa de fourier
    do k=1,N


          real1=0
          imag1=0


          !soma de fourier
        do j=1,N


            real1=real1+(valorR(j)*dcos(2*pi*j/N*k)-valorI(j)*dsin(-2*pi*j/N*k))
            imag1=imag1+(valorR(j)*dsin(-2*pi*j/N*k)+valorI(j)*dcos(2*pi*j/N*k))


        enddo

          !saida de dados
          t=k*0.04
          write(3,*)t,real1/N
          write(4,*)t,imag1/N

    enddo

  close(1)
  close(2)
  delta = secnds(t1)
  write(*,*)(delta)

end program
```

2.2 Parte 4

Na parte 4 desse de projeto vamos utilizar a função a (17) para voltamos ao sinal original onde fizemos a transformada. No gráfico abaixo vemos os resultados.

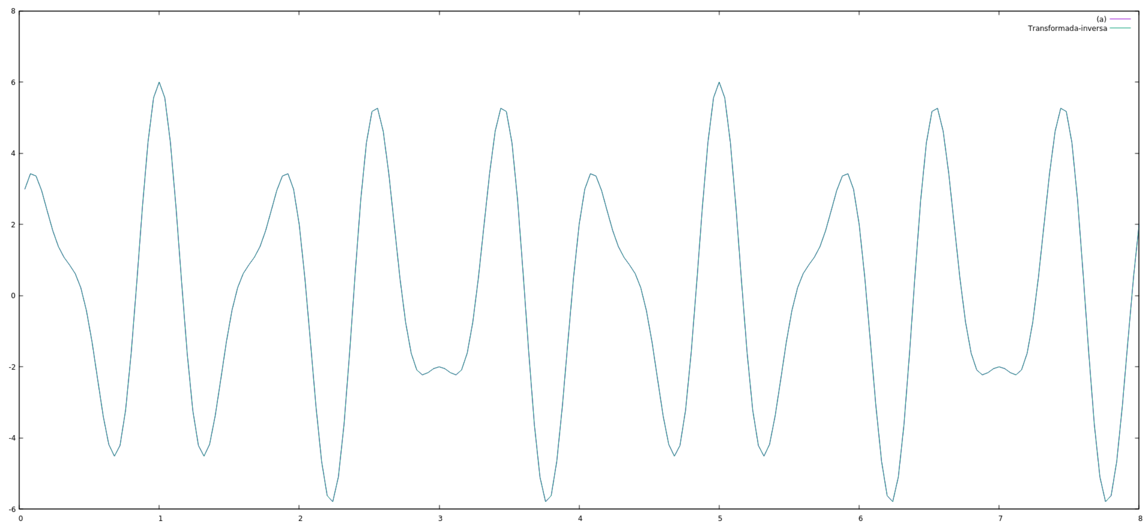

Figura 6: Transformada-inversa e função de onda original

Como é visto no gráfico acima que as duas onda estão encima uma da outra no gráfico mostrando que transformada inversa conseguiu voltar a função de onda original.

3 Parte 5

No gráfico abaixo é mostrado como é o aumento do tempo como conforme mais dados são processados no programa, seguindo um aumento de complexidade N^2 fazendo o tempo aumentar exponencialmente em função da quantidade dados rodando no programa.

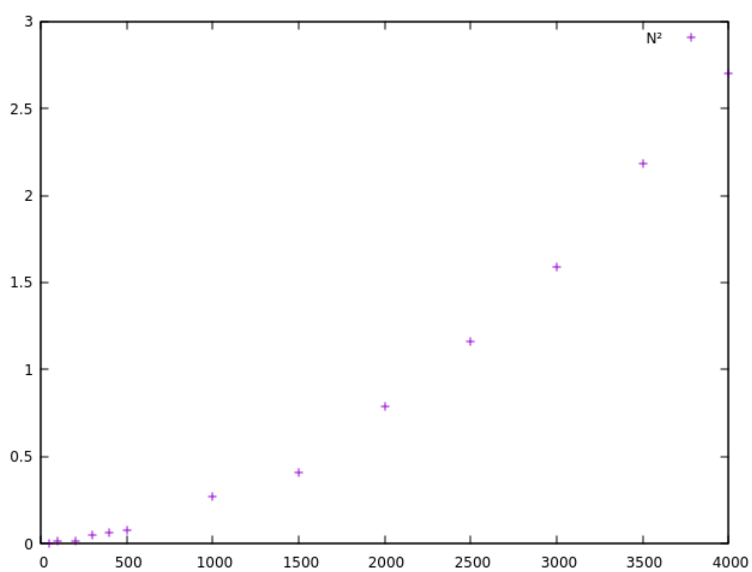

Figura 7: Gráfico Nxtempo

Os pontos que obtivemos no gráfico estão listado na tabela abaixo,

N	Tempo
50	3.906E-3
100	1.562E-2
200	1.953E-2
300	5.078E-2
400	6.250E-2
500	8.203E-2
1000	0.269
1500	0.41
2000	0.79
2500	1.16
3000	1.59
3500	2.18
4000	2.7

Tabela 3: Caption

Como vemos no gráfico 7 a curva cresce exponencialmente vendo que segue um padrão N^2 além da tabela informar o mesmo.