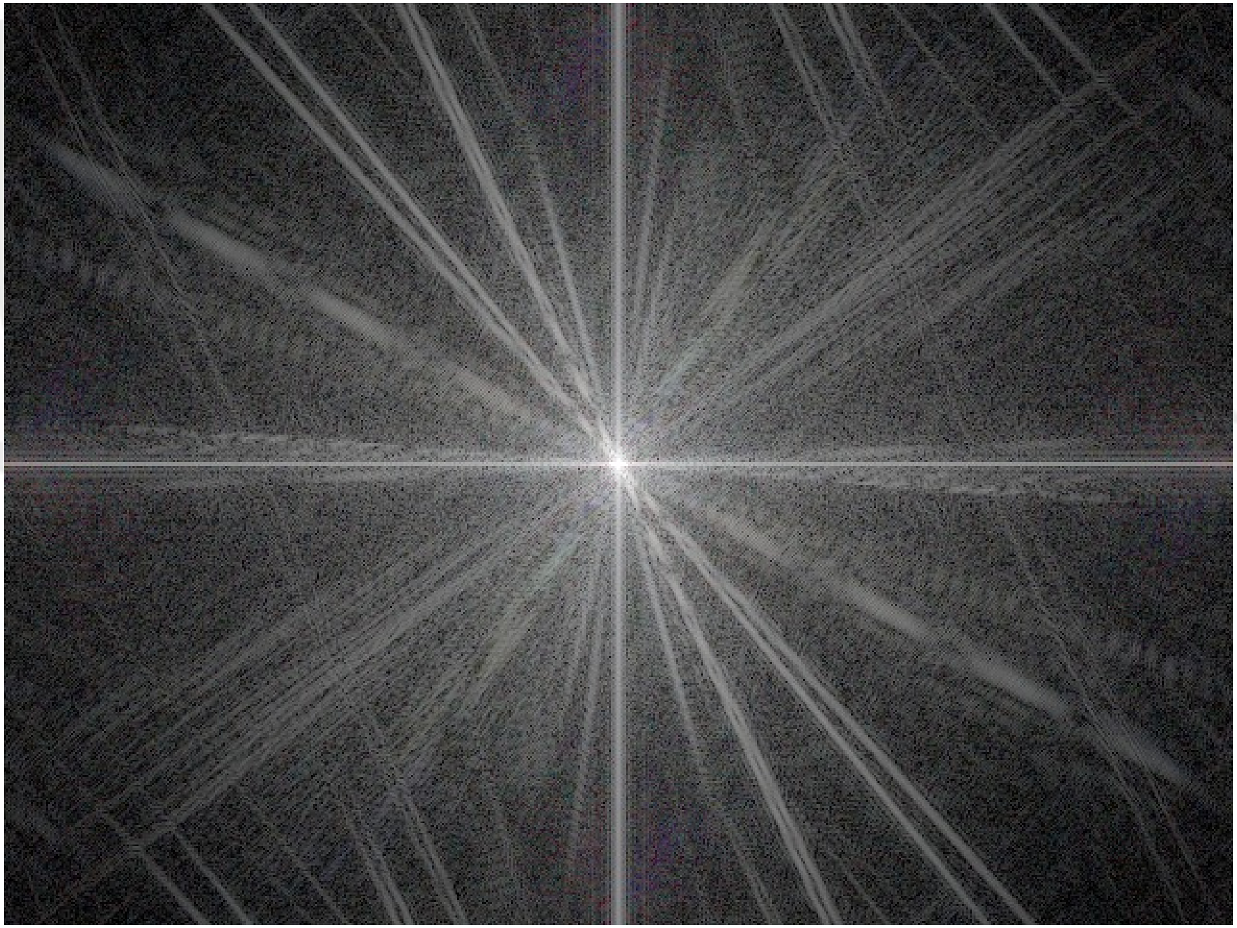
Processamento Digital de Imagens

Prof. Bogdan Tomoyuki Nassu



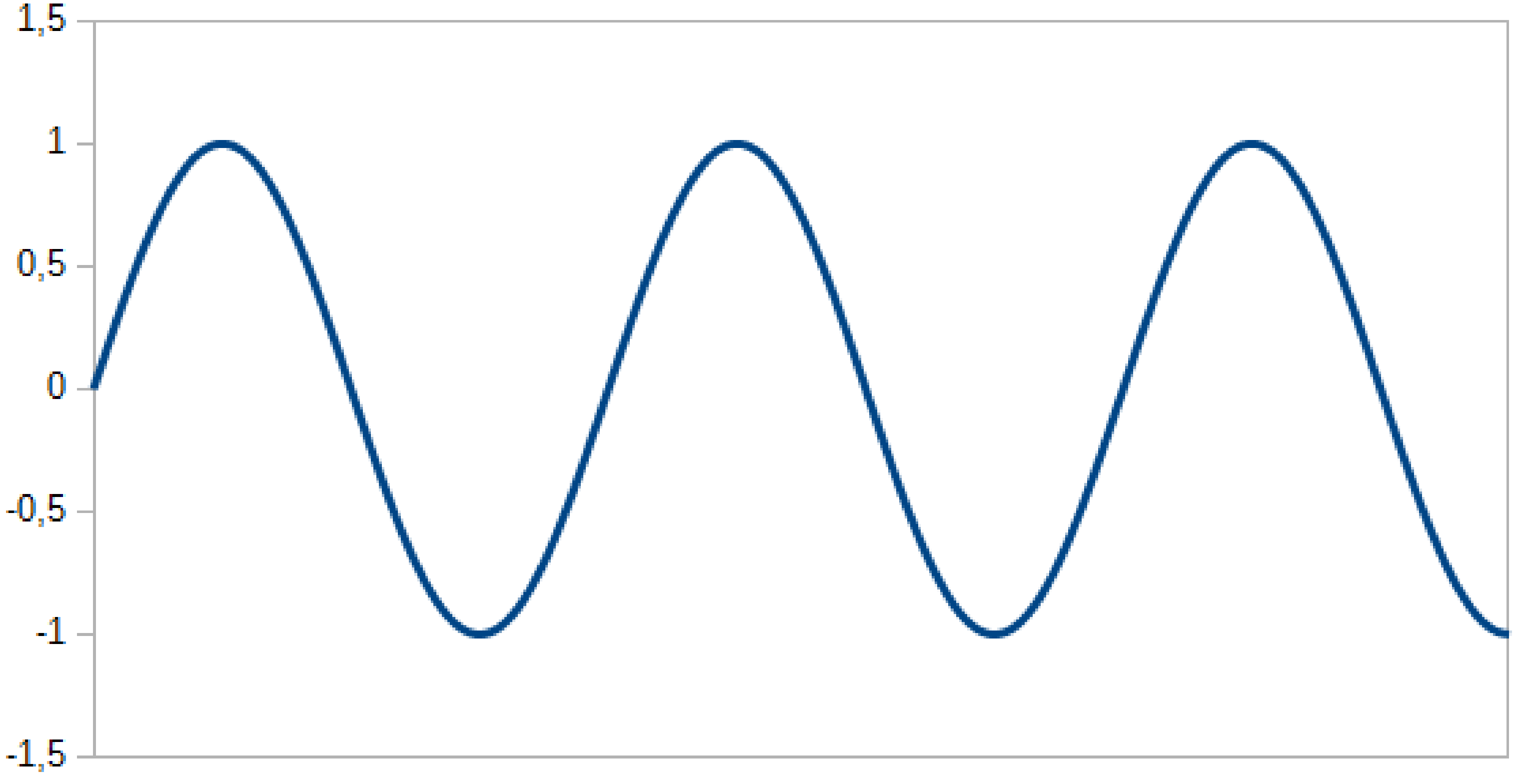
UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

A transformada de Fourier

- Ideia: todo sinal pode ser decomposto em uma soma de senoides.
  - 1822.
  - Aplicações diversas em matemática, física e processamento de sinais.
- Antes de pensar em 2D, vamos pensar em 1D.

Exemplo

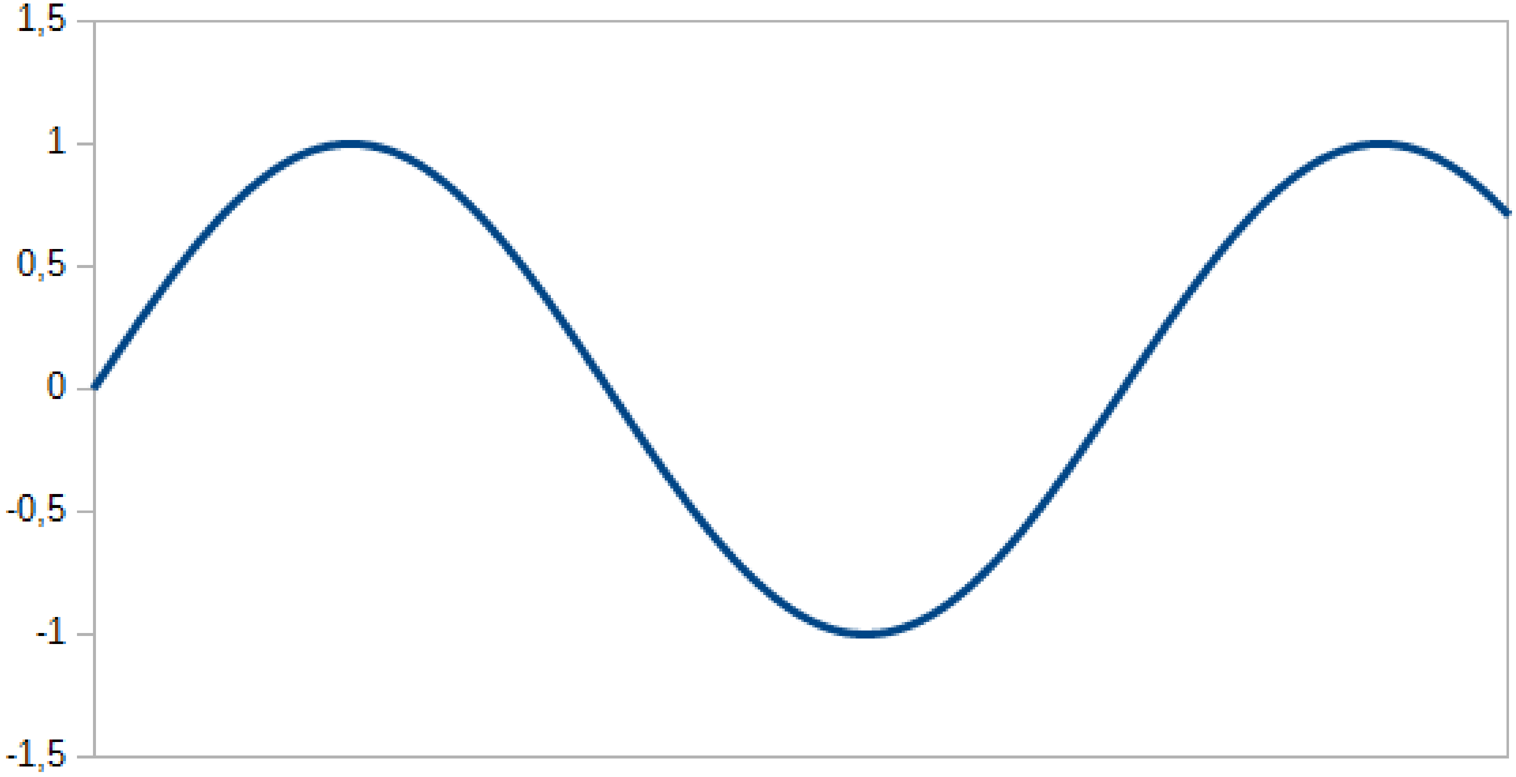
• *sen*(*x*), com *x* começando em 0º e variando em passos de 10º.



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Exemplo

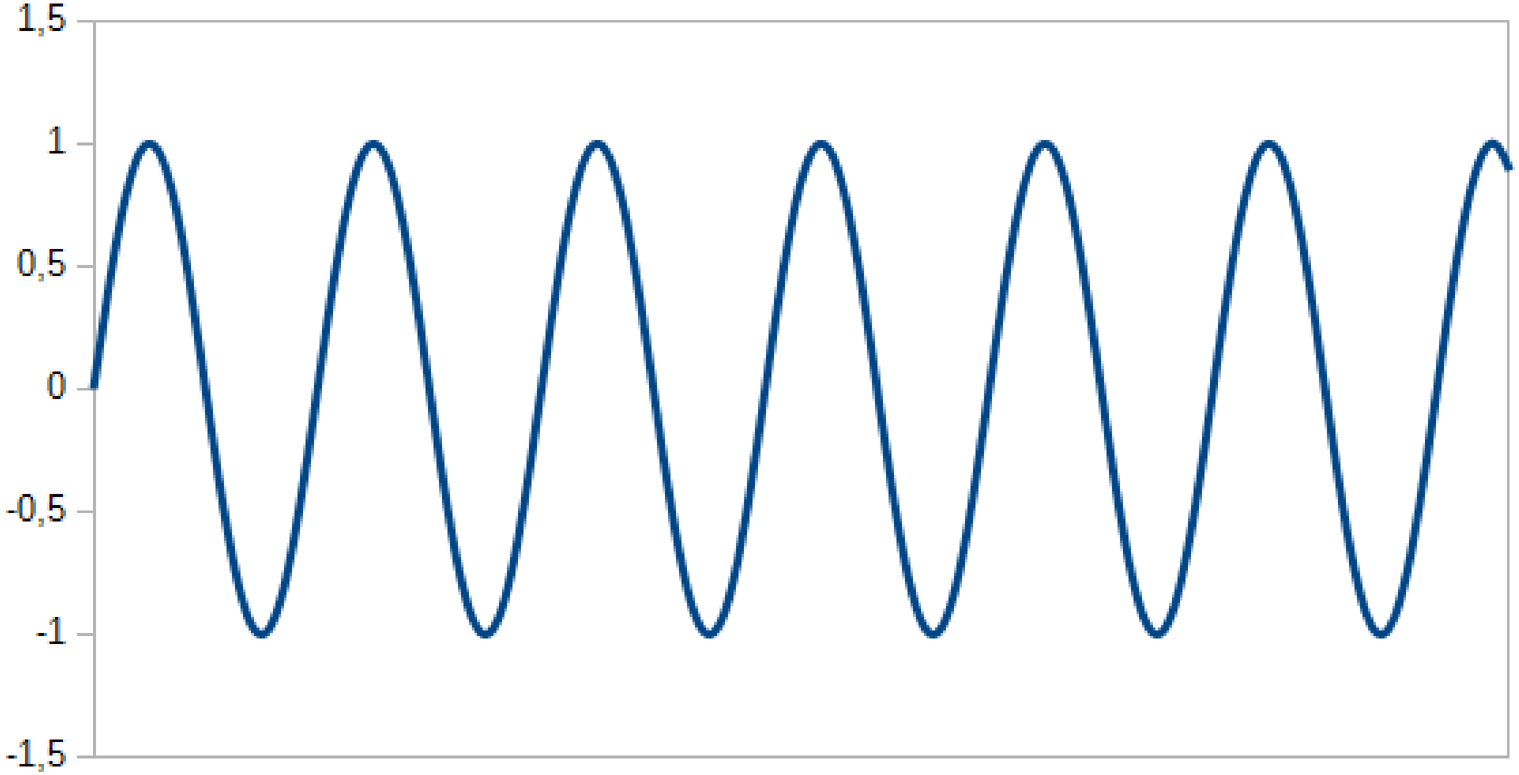
• *sen*(*x*), com *x* começando em 0º e variando em passos de 5º.



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

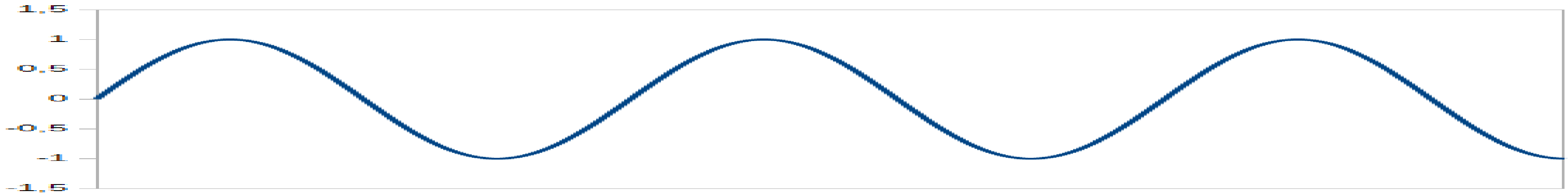
Exemplo

• $sen(x)$, com x começando em 0^o e variando em passos de 23^o.

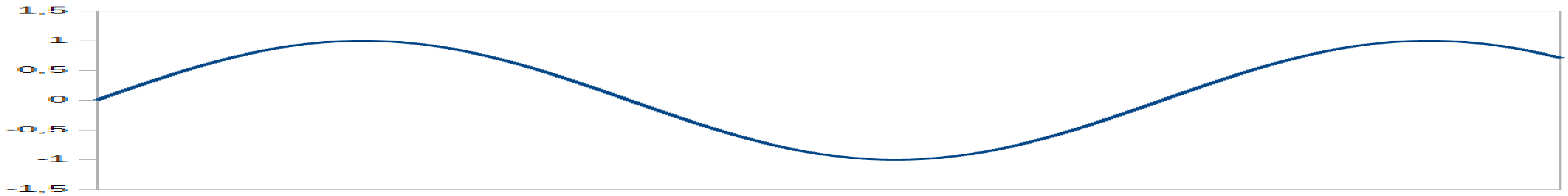


UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

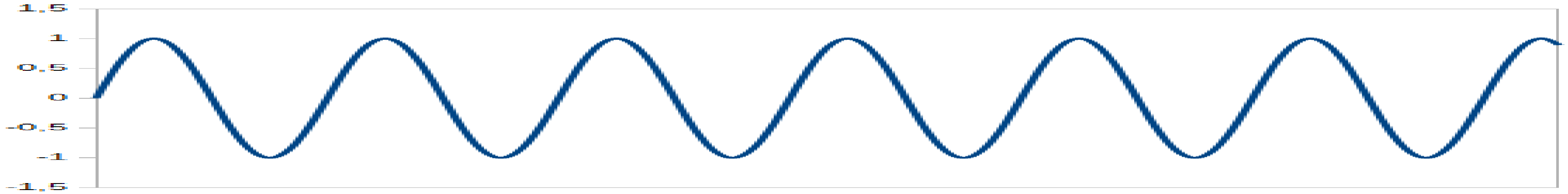
Exemplo



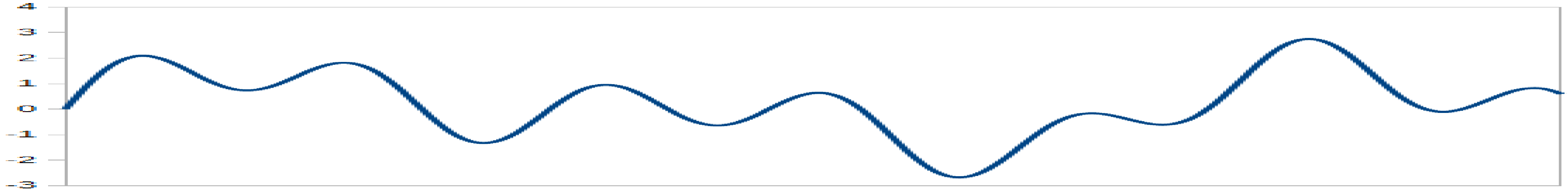
+



+



=



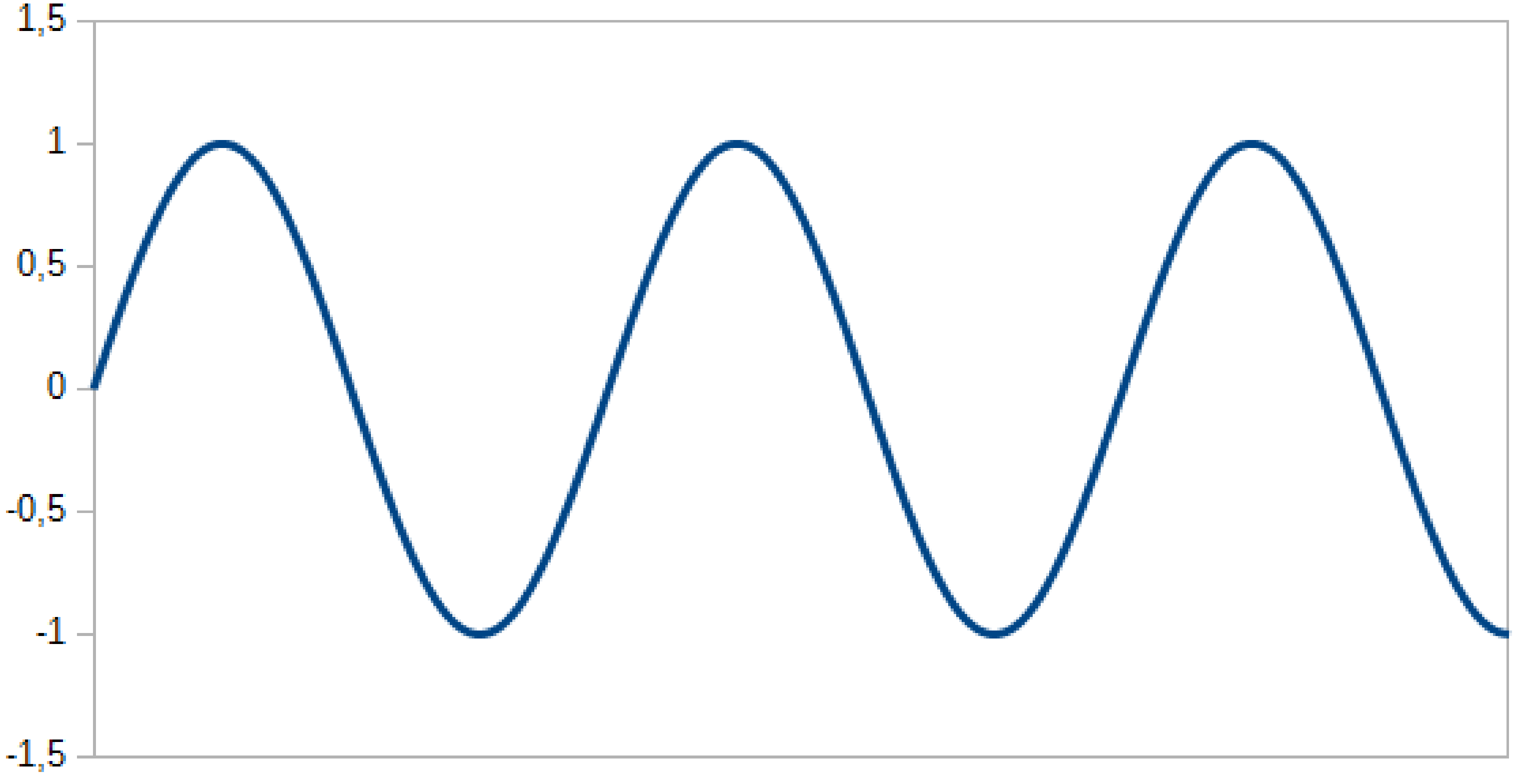
UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Exemplo: áudio

- Podemos decompor um sinal de áudio em senoides.
  - Descritas em termos de “ciclos por segundo” (Hz).
    - = frequência.
- Exemplo:
  - A 5ª corda de um violão, tocada solta, produz a nota Lá.
    - Sinal com frequência fundamental de 110 Hz.
    - Somado a outras frequências, com volumes menores, forma o timbre do violão.
    - Cada uma das frequências é um *componente* do sinal.
- Precisamos de mais 2 conceitos:
  - Magnitude: o “volume” de cada componente.
  - Fase: a “posição inicial” de cada componente.

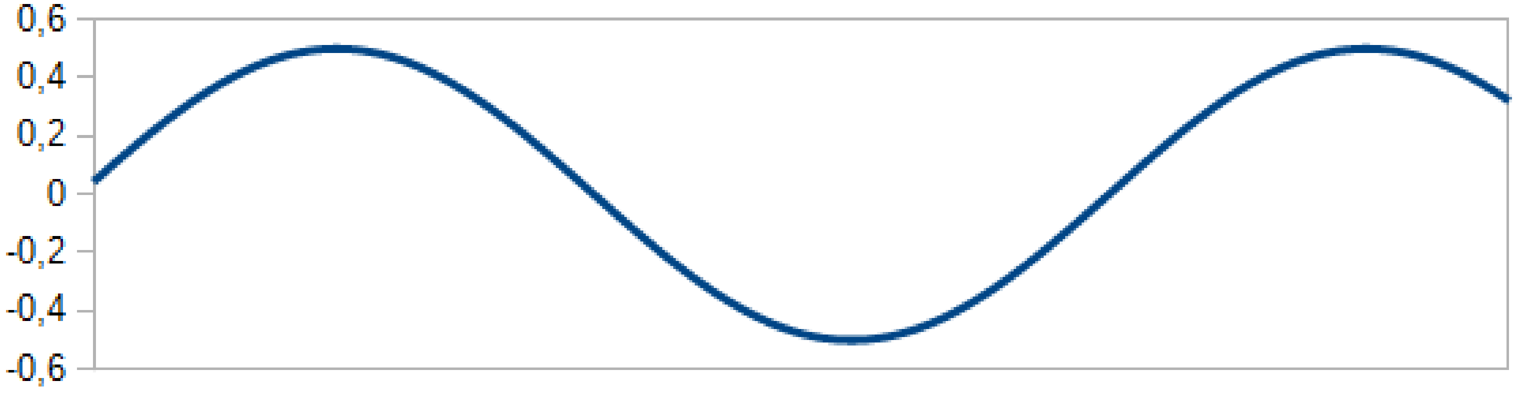
Exemplo

•*sen*(*x*), com *x* começando em 0º e variando em passos de 10º, com magnitude = 1 (não mudou!).



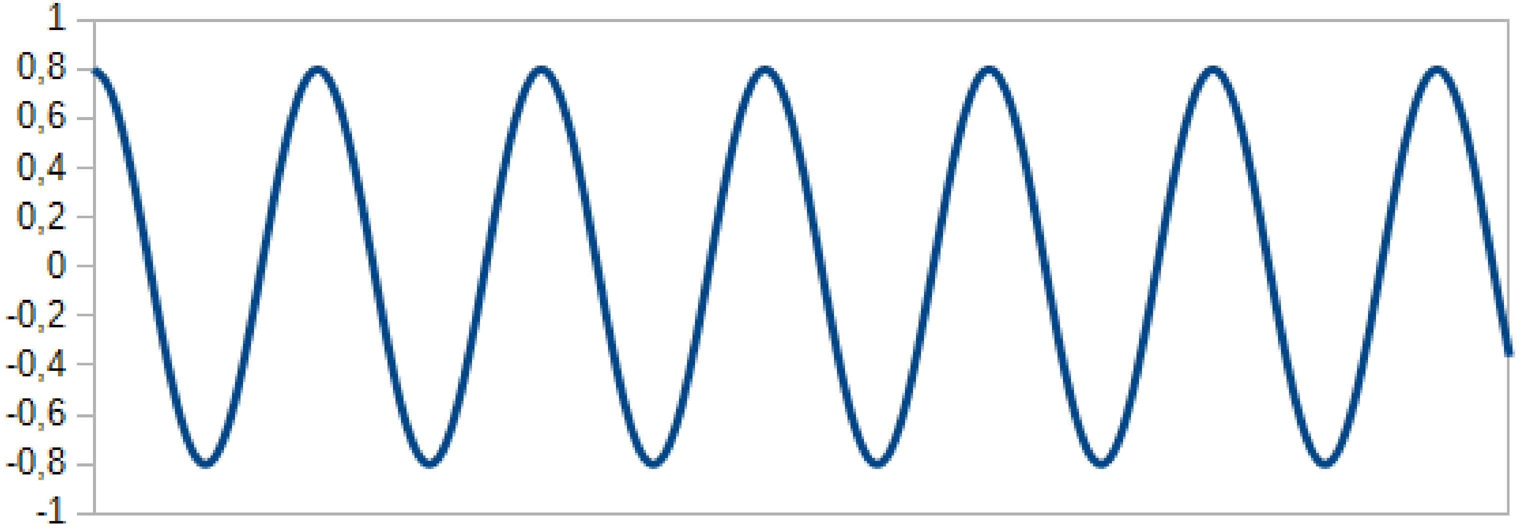
Exemplo

- *sen*(*x*), com *x* começando em 15º e variando em passos de 5º, com magnitude 0.5.

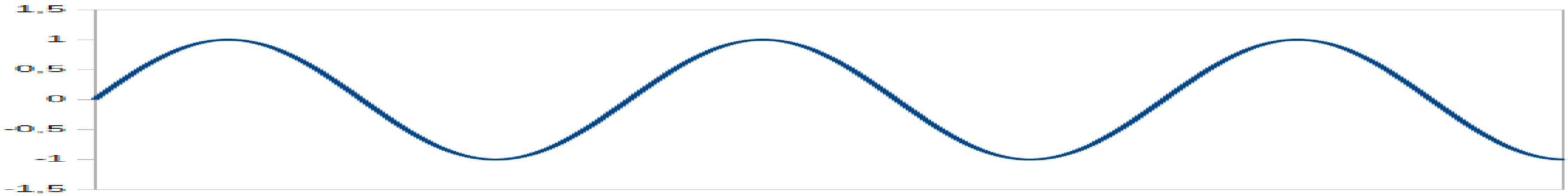


Exemplo

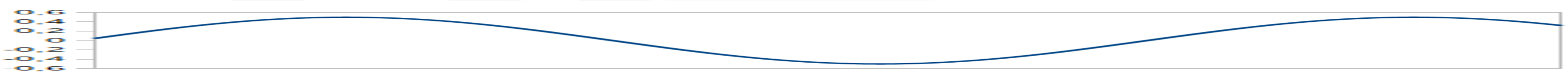
- *sen*(*x*), com *x* começando em 90º e variando em passos de 23º, com magnitude 0.8.



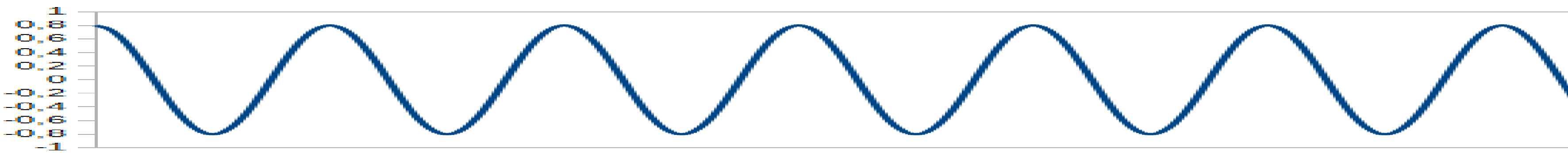
Exemplo


1,5
1
0,5
0
-0,5
-1
-1,5

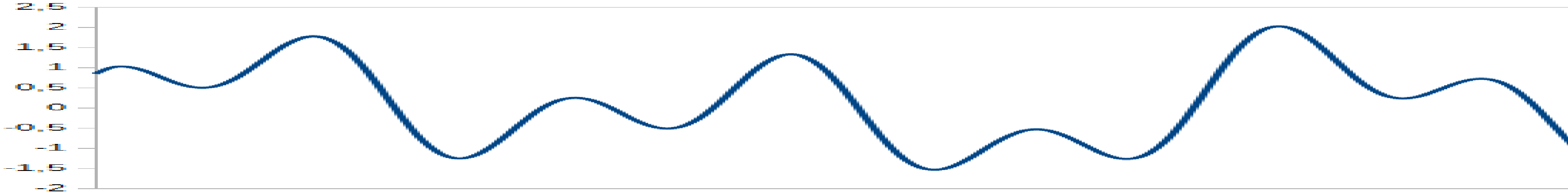
+


0,6
0,4
0,2
0
-0,2
-0,4
-0,6

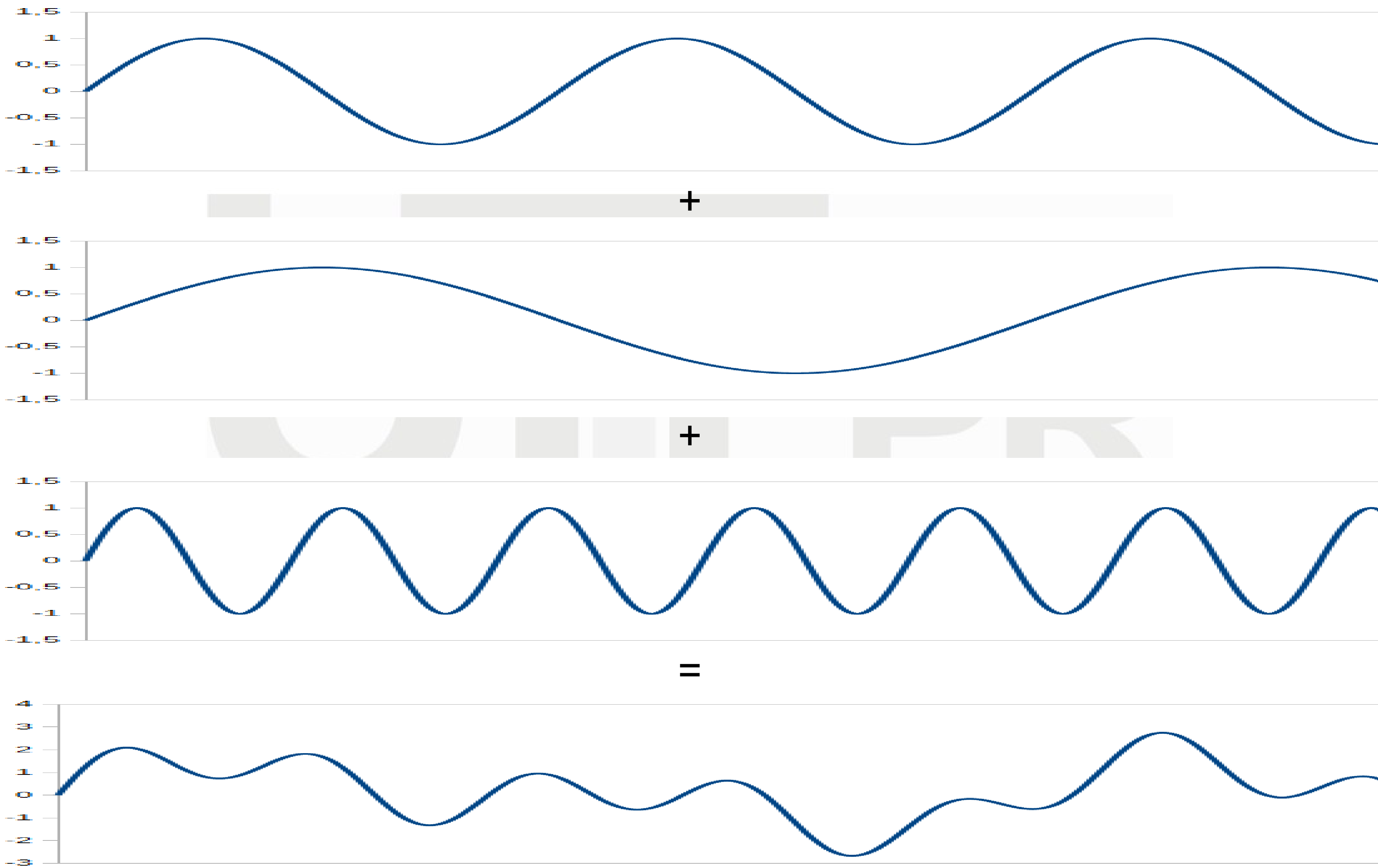
+


1
0,8
0,6
0,4
0,2
0
-0,2
-0,4
-0,6
-0,8
-1

=


2,5
2
1,5
1
0,5
0
-0,5
-1
-1,5
-2

Anterior (mag 1, fase 0)



Animações!

- Vejamos algumas animações!

www.youtube.com/watch?v=LznjC4Lo7IE

www.youtube.com/watch?v=-GYB7khbIA0

www.youtube.com/watch?v=cZFaepZL7wc

Objetivo

- A transformada de Fourier decompõe um sinal em seus componentes de frequência.
  - = análise ou processamento no domínio da frequência.
  - Entrada: um sinal.
    - É dado como um vetor de amostras.
  - Saída: para cada frequência possível, 2 valores:
    - A magnitude (o quanto aquela frequência está presente no sinal).
    - A fase.
- Nota 1: vamos considerar apenas a versão discreta da transformada.
  - DFT: *discrete Fourier transform*.
- Nota 2: pressupõe-se que o sinal é periódico e infinito.
  - Existem vários jeitos de tratar isso, mas para os nossos fins, vamos supor que o sinal simplesmente se repete.

Utilidade

- Considerando ainda sinais de áudio, qual a utilidade de se decompor o sinal desta forma?
  - Importante: a transformada pode ser invertida – ou seja, podemos decompor e recompor um sinal.

Utilidade

- Considerando ainda sinais de áudio, qual a utilidade de se decompor o sinal desta forma?
  - Compressão: remover frequências inaudíveis.
  - Análise e modificação de timbres.
  - Encontrar a frequência fundamental (afinadores).
  - Equalização.
  - Simulação de equipamentos e ambientes.

A transformada

- A transformada de Fourier 1D é definida por:

$$F(k)=\int_{-\infty}^{+\infty} f(t)\, e^{-2\pi i t k/n}\, dt$$

A transformada

•A DFT 1D é definida por:

$$F(k)=\sum_{t=0}^{n-1} f(t)e^{-2\pi itk/n}$$

?!?!?!?!?!?!?!?!?

A transformada

•A DFT é definida por:

$$F(k)=\sum_{t=0}^{n-1} f(t)e^{-2\pi itk/n}$$

Para cada frequência k...

A transformada

•A DFT é definida por:

$$F(k)=\sum_{t=0}^{n-1} f(t)e^{-2\pi itk/n}$$

Para cada frequência k...

Nota: se a entrada tem *n* amostras, temos *k* frequências também.

A transformada

- A DFT é definida por:

$$F(k)=\sum_{t=0}^{n-1} f(t)e^{-2\pi itk/n}$$

Para cada frequência k...

... observa todas as posições do vetor.

A transformada

- A DFT é definida por:

= vetor [t]

$$F(k)=\sum_{t=0}^{n-1} f(t)e^{-2\pi itk/n}$$

Para cada frequência k...

... observa todas as posições do vetor.

A transformada

- A DFT é definida por:

= vetor [t]

$$F(k)=\sum_{t=0}^{n-1} f(t)\, e^{-2\pi i t k/n}$$

Para cada frequência k...

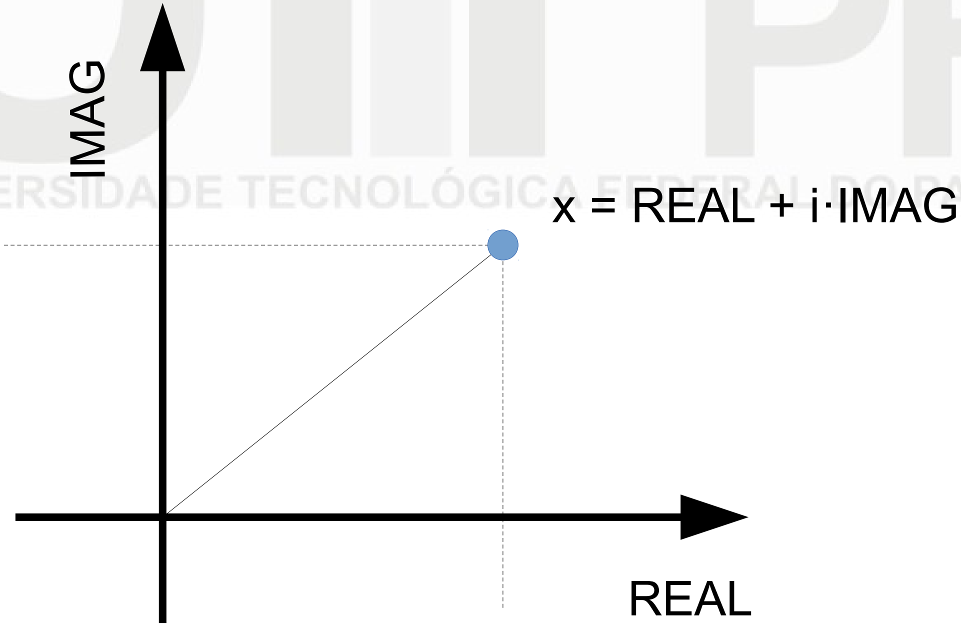
Número imaginário!

... observa todas as posições do vetor.

Sobre números complexos

- Não podemos usar números complexos diretamente.
- Normalmente, usa-se a forma cartesiana:

$$x = \boxed{REAL} + i \cdot \boxed{IMAG}$$



Fórmula de Euler

$$e^{i\alpha} = \cos(\alpha) + i \cdot sen(\alpha)$$

$$F(k) = \sum_{t=0}^{n-1} f(t) \boxed{e^{-2\pi itk/n}}$$

Aplicando a fórmula de Euler a esta parte...

DFT no formato cartesiano

$$\alpha = 2\pi \frac{tk}{n}$$

Um ângulo, que será usado várias vezes.

Parte REAL.

$$F(k) = \sum_{t=0}^{n-1} \boxed{f(t)_{real} \cdot \cos(\alpha) + f(t)_{imag} \cdot \sin(\alpha)}$$

$$+ i \cdot \boxed{[-f(t)_{real} \cdot \sin(\alpha) + f(t)_{imag} \cdot \cos(\alpha)]}$$

Parte IMAGINÁRIA.

DFT no formato cartesiano

- O nosso sinal original só tem valores reais!
  - = para todo t, $f(t)_{imag} = 0$.
  - Poderíamos simplificar a equação para a forma abaixo.
    - Isso não é feito porque vamos reaproveitar a equação para a transformada inversa...

$$\alpha = 2\pi \frac{tk}{n}$$

$$F(k) = \sum_{t=0}^{n-1} \underbrace{\boxed{f(t) \cdot \cos(\alpha)}}_{\text{Parte REAL.}} + i \cdot \underbrace{\boxed{-f(t) \cdot \sin(\alpha)}}_{\text{Parte IMAGINÁRIA.}}$$

DFT: algoritmo

```
Vetores de entrada (ambos com n posições):
f: parte real do sinal (= sinal original)
f': parte imaginária do sinal (= um vetor de 0s)

Vetores de saída (ambos com n posições):
F: parte real da DFT
F': parte imaginária da DFT

for (cada posição k entre 0 e n-1)
   soma_real = 0
   soma_imag = 0
   for (cada posição t entre 0 e n-1)
      alfa = 2*PI*t*k/n
      soma_real += f[t] * cos(alfa) + f'[t] * sin(alfa)
      soma_imag += -f[t] * sin(alfa) + f'[t] * cos(alfa)

   F[k] = soma_real
   F'[k] = soma_imag
```

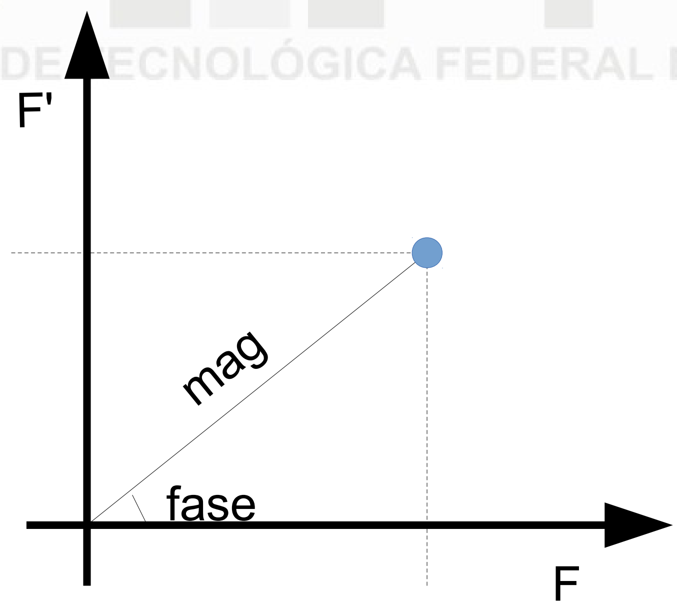
O algoritmo é surpreendentemente simples!

DFT: interpretando

- A DFT produz 2 vetores, F e F'.
  - = partes reais e imaginárias dos componentes.
- Para converter F e F' em fase e magnitude (espectro):

$$mag(k) = \sqrt{F(k)^2 + F'(k)^2}$$

$$fase(k) = atan2(F'(k)/F(k))$$



A DFT invertida

- A transformada de Fourier é invertível!
  - = podemos decompor um sinal em frequências... ou reconstruir um sinal dadas as suas frequências!
  - = podemos converter um sinal para o domínio da frequência, alterá-lo, e reconvertê-lo para o domínio original.
  - Ex: equalização.

A DFT invertida

$$\alpha = 2\pi \frac{tk}{n}$$

Parte REAL.

$$f(t) = \frac{1}{n} \cdot \sum_{k=0}^{n-1} \Big(\boxed{F(k)_{imag} \cdot \cos(\alpha) + F(k)_{real} \cdot \sin(\alpha)}$$

$$+ i \cdot \boxed{\left[-F(k)_{imag} \cdot \sin(\alpha) + F(k)_{real} \cdot \cos(\alpha) \right]} \Big)$$

Parte IMAGINÁRIA.

$$\alpha = 2\pi \frac{tk}{n}$$

DFT

$$F(k) = \sum_{t=0}^{n-1} f(t)_{real} \cdot \cos(\alpha) + f(t)_{imag} \cdot \sin(\alpha)$$
$$+ i \cdot [-f(t)_{real} \cdot \sin(\alpha) + f(t)_{imag} \cdot \cos(\alpha)]$$

$$f(t) = \frac{1}{n} \cdot \sum_{k=0}^{n-1} \Big(F(k)_{imag} \cdot \cos(\alpha) + F(k)_{real} \cdot \sin(\alpha)$$
$$+ i \cdot [-F(k)_{imag} \cdot \sin(\alpha) + F(k)_{real} \cdot \cos(\alpha)] \Big)$$

DFT invertida

A DFT invertida

- A DFT invertida é *muito* parecida com a DFT, exceto que:
  - Entradas e saídas são trocadas.
  - As partes reais e imaginárias das entradas são trocadas.
  - Existe uma divisão a mais.

Vetores de entrada (ambos com n posições):
F: parte real da DFT
F': parte imaginária da DFT

Vetores de saída (ambos com n posições):
f: parte real do sinal
f': parte imaginária do sinal

Se você não fizer nada de estranho, *f'* costuma ser um vetor com valores muito próximos de 0, indicando o erro residual causado por arredondamentos.

Dada a função DFT (f, f', F, F'), chama DFT (F', F, f', f). Ao final, divide todas as posições de f e f' por n.

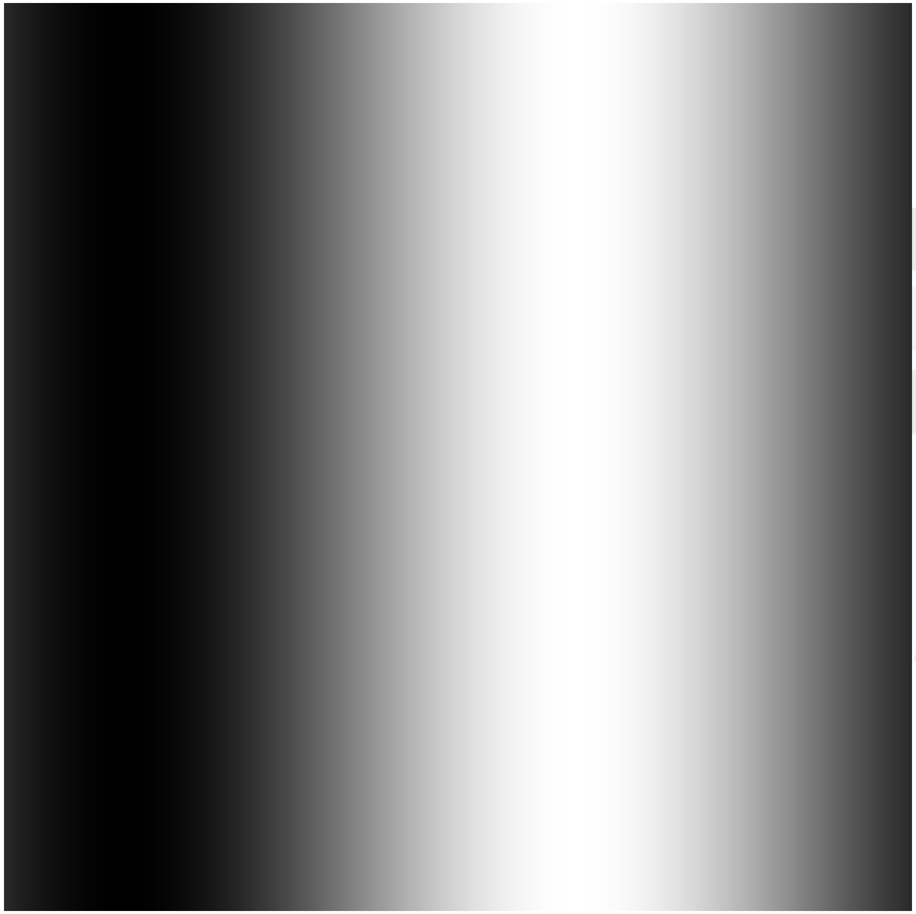
FFT

- Implementações "reais" da DFT normalmente são mais rápidas do que o algoritmo mostrado.
  - FFT = *Fast Fourier Transform.*
- Existem vários algoritmos para FFT.
  - Existem também várias implementações disponíveis.
- Entre outras técnicas, as FFT usam:
  - Eliminação de operações redundantes.
  - Divisão e conquista.
  - Aproximações.
- Algumas implementações exigem imagens quadradas, ou com altura e largura potências de 2.

Indo para 2D

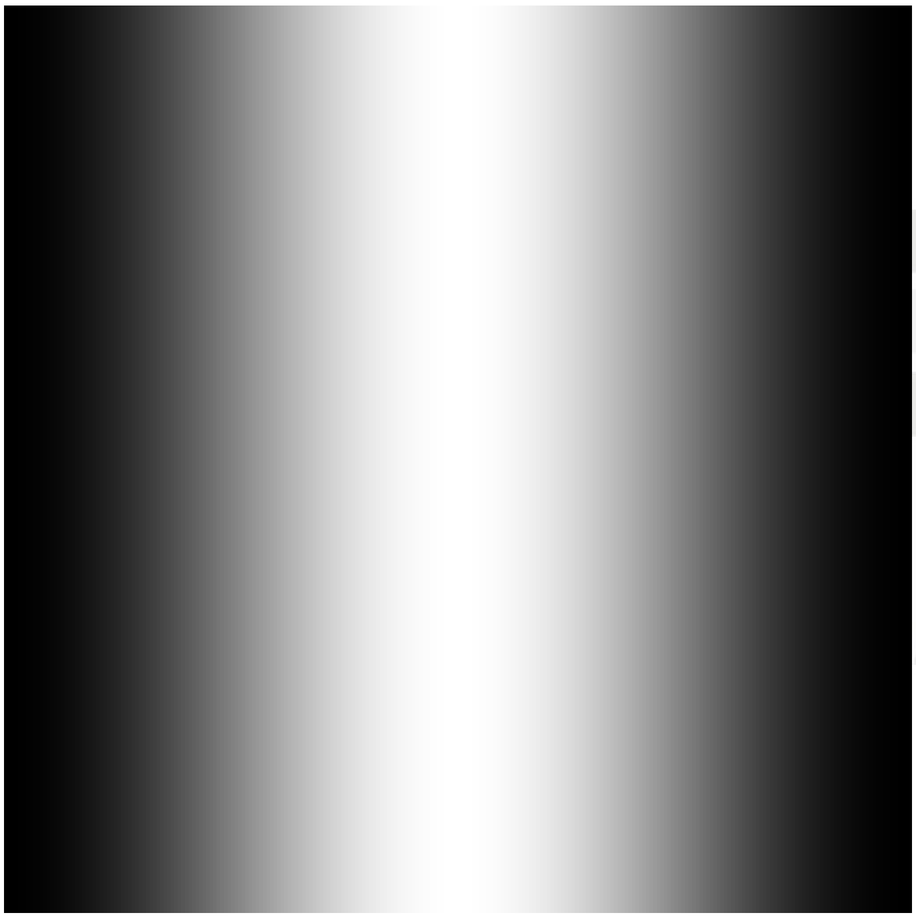
•Senoides 2D?

Indo para 2D



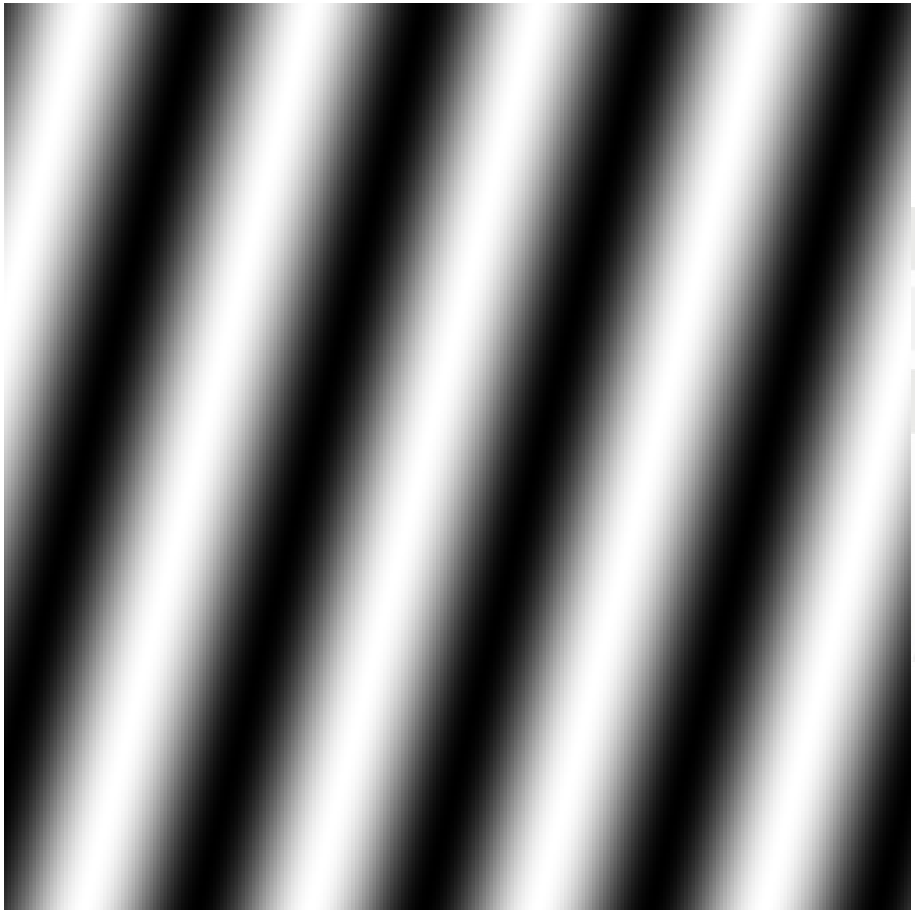
Senoide 2D com 1 ciclo
na horizontal e 0 na vertical
(frequência (1,0)).

Indo para 2D



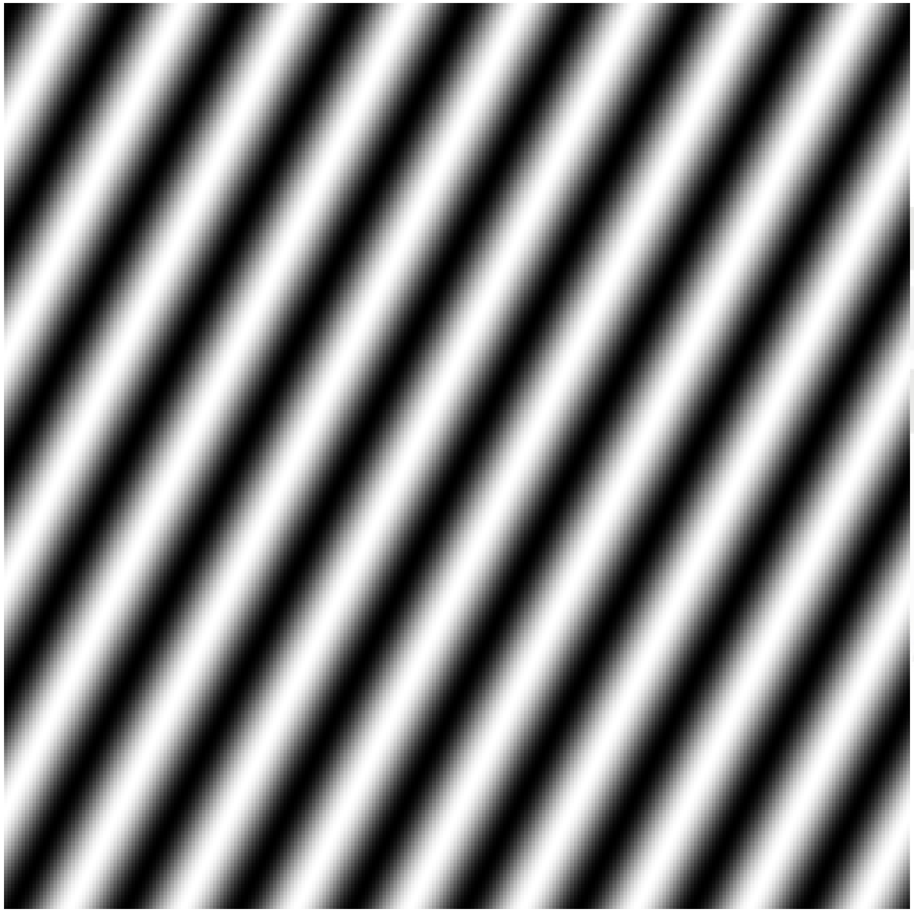
Senoide 2D com 1 ciclo
na horizontal e 0 na vertical
(frequência (1,0)), mas
outra fase.

Indo para 2D



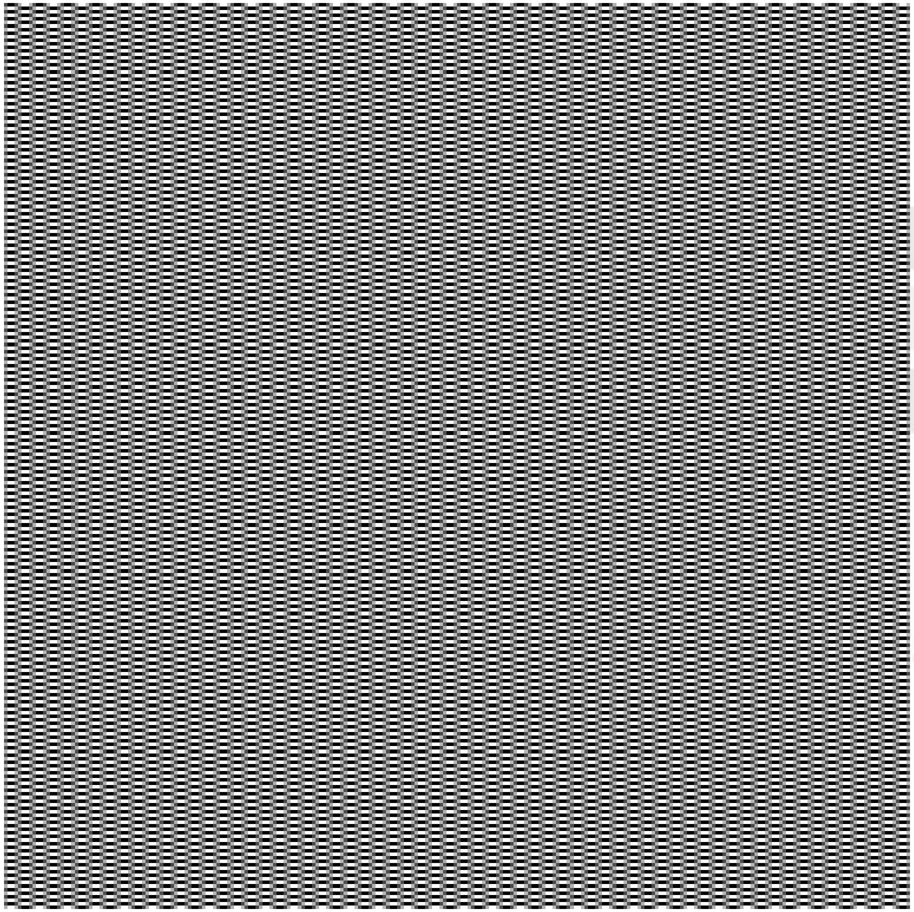
Senoide 2D com 4 ciclos na horizontal e 1 na vertical (frequência (4,1)).

Indo para 2D



Senoide 2D com 8 ciclos na horizontal e 4 na vertical (frequência (8,4)).

Indo para 2D



Senoide 2D com (muitos) ciclos na horizontal e (muitos)na vertical (frequência (?,?)).

Indo para 2D

- A transformada de Fourier pode ser aplicada em funções 2D.
  - A equação é para imagens em escala de cinza.
  - Para imagens coloridas, fazemos a transformação para cada canal separadamente.

1D

$$F(k)=\sum_{t=0}^{n-1} f(t)e^{-2\pi itk/n}$$

2D

$$F(k,l)=\sum_{y=0}^{M-1}\sum_{x=0}^{N-1} f(x,y)e^{-2\pi i\left(\frac{ky}{M}+\frac{lx}{N}\right)}$$

Indo para 2D

$$F(k,l)=\sum_{y=0}^{M-1}\sum_{x=0}^{N-1} f(x,y)e^{-2\pi i\left(\frac{ky}{M}+\frac{lx}{N}\right)}$$

```
for (cada frequência k na vertical)
   for (cada frequência l na horizontal)
      soma_real = 0
      soma_imag = 0
      for (cada linha y da imagem)
         for (cada coluna x da imagem)
            soma_real += (um monte de coisas)
            soma_imag += (um monte de coisas)

      F[k] = soma_real
      F'[k] = soma_imag
```

IMPORTANTE!!!

Aqui, "frequência" não tem relação com cores, e sim com variações de intensidade na imagem.

DFT 2D

- A DFT 2D “força bruta” tem alta complexidade computacional.
  - Para uma imagem com MxN pixels, são $(MN)^2$ iterações!
- Felizmente, a DFT é separável!
  - (o que significa isso mesmo)?

DFT 2D a partir da DFT 1D

```
Imagem de entrada (com MxN pixels): f

Imagens de saída (ambas com MxN pixels):
F: parte real da DFT
F': parte imaginária da DFT

cria duas imagens H e H'

for (cada linha y da imagem)
   h = vetor de N posições contendo os valores na linha y da imagem
   h' = vetor de N posições contendo 0s
   DFT (h, h', H[y], H'[y])

for (cada coluna x da imagem)
   v = vetor de M posições contendo os valores H[?][x]
   v' = vetor de M posições contendo os valores H'[?][x]
   DFT (v, v', F[?][x], F'[?][x])
```

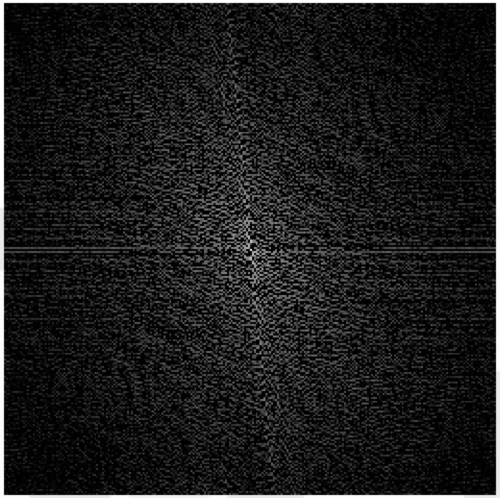
Em vez de $(MN)^2$, são $MN^2 + M^2N$ iterações (ou ainda menos, se usarmos uma FFT).

DFT 2D: resultados

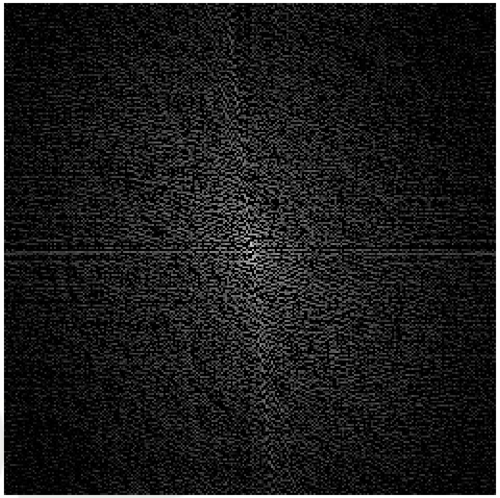
- Os resultados da DFT 2D normalmente são rearranjados para que a frequência (0,0) fique no centro das imagens de saída.

Exemplo

Imagem original

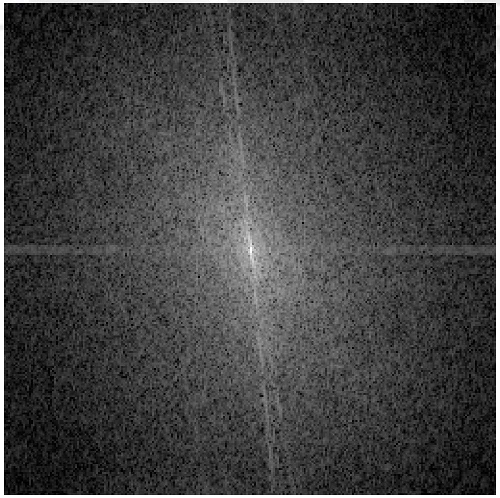


DFT (real)

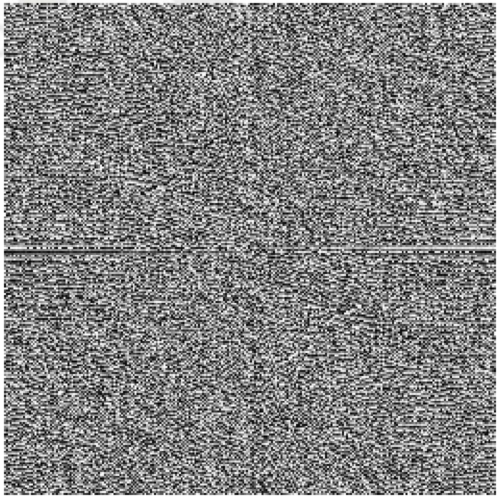


DFT (imag)

Nota: as imagens da DFT e da magnitude são muito difíceis de ver. Aqui, modificamos cada pixel, obtendo 0.25*log10 (pixel).



Magnitude



Fase

Exemplo

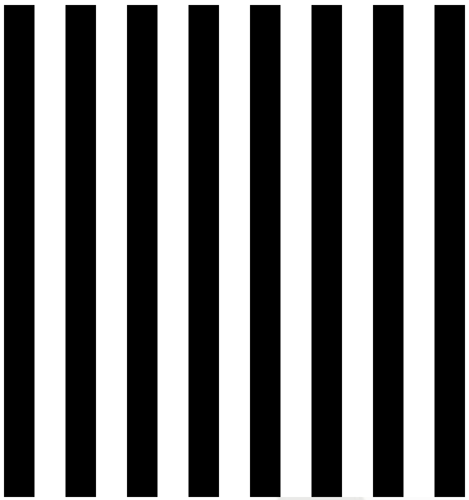
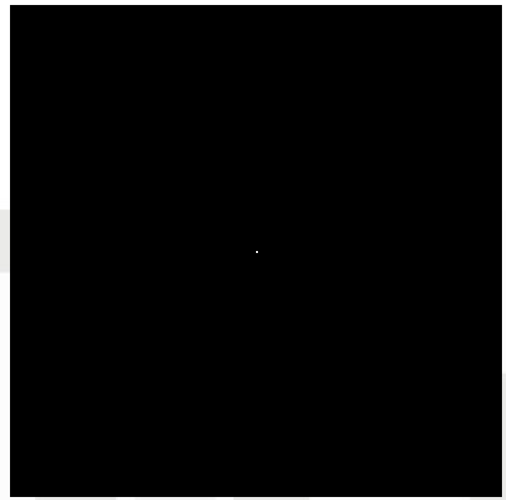
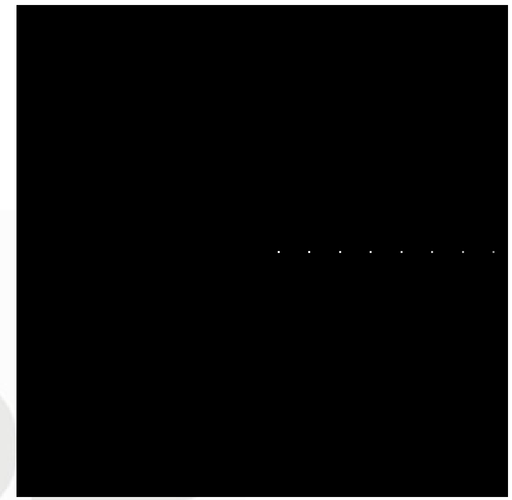


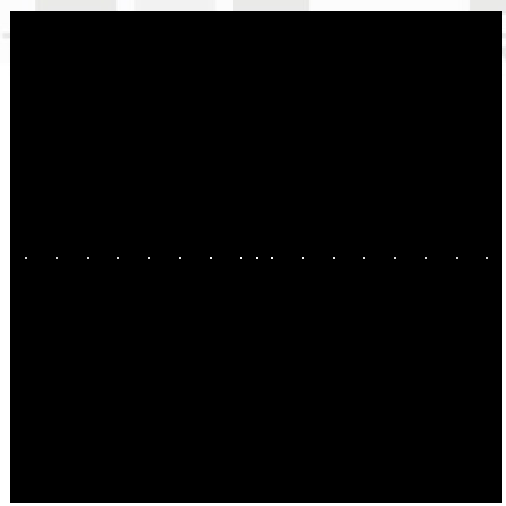
Imagem original



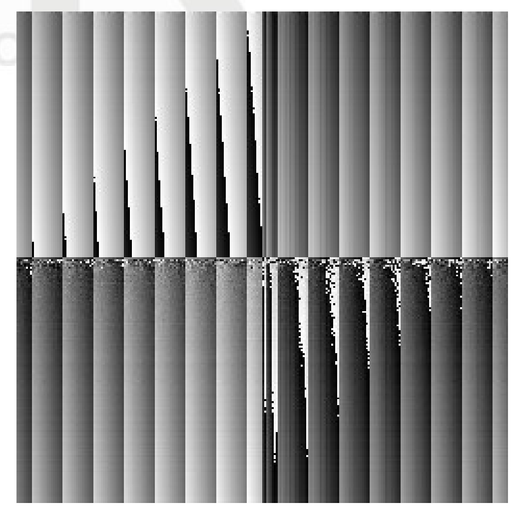
DFT (real)



DFT (imag)



Magnitude



Fase

Exemplo

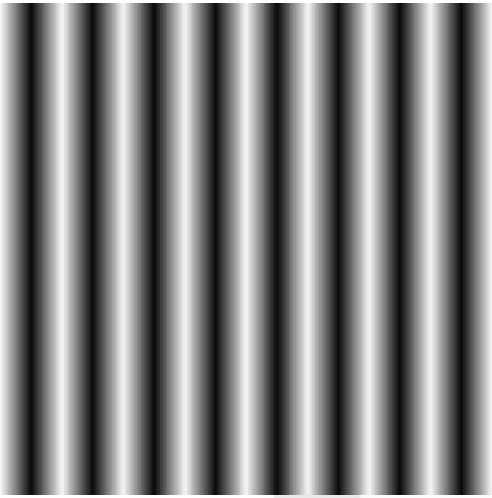
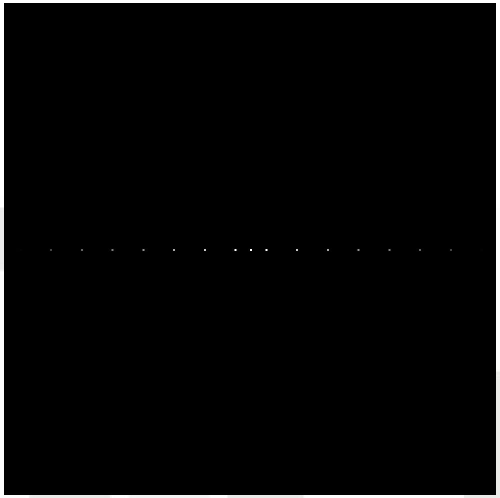
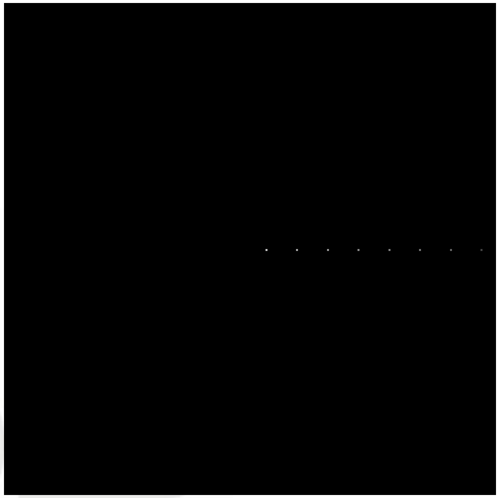


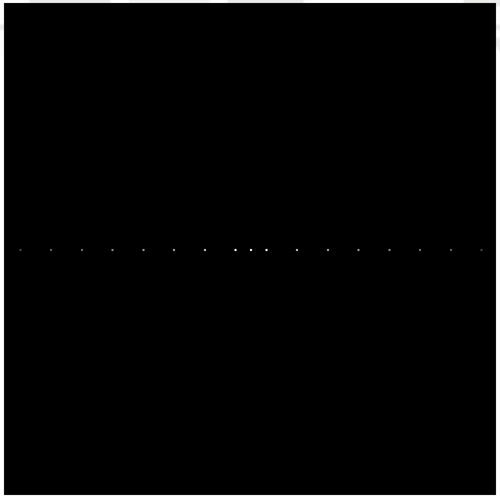
Imagem original



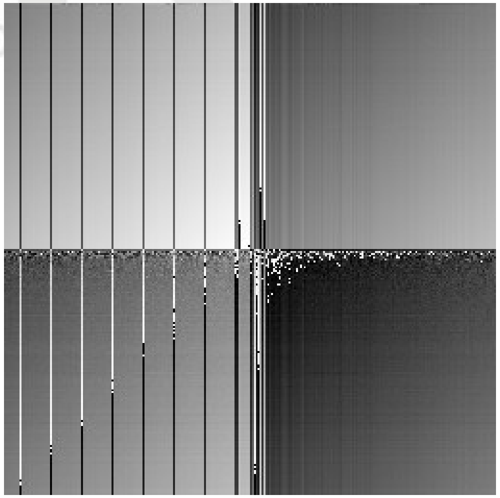
DFT (real)



DFT (imag)



Magnitude



Fase

Exemplo

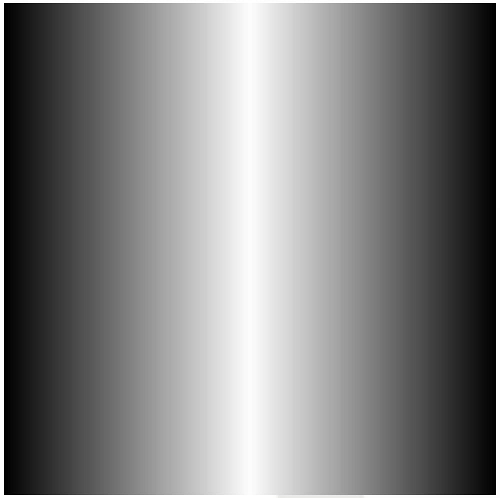
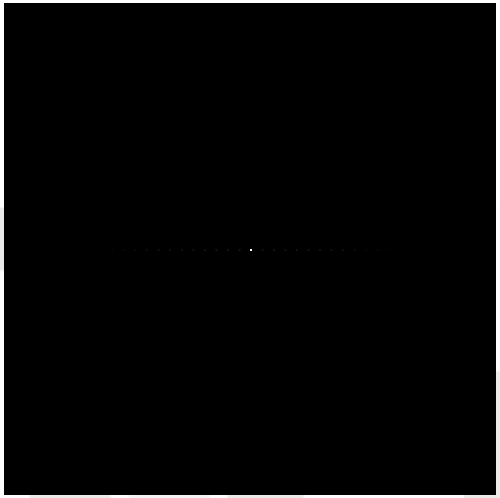
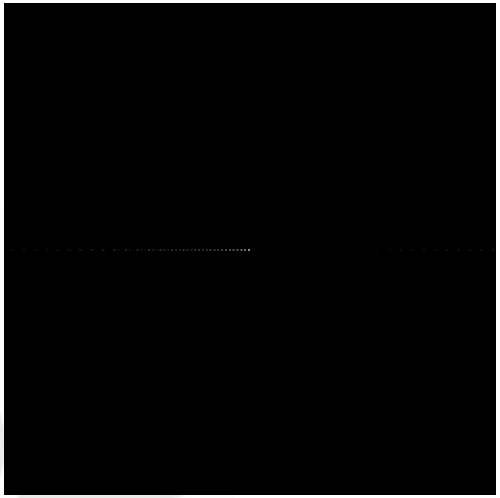


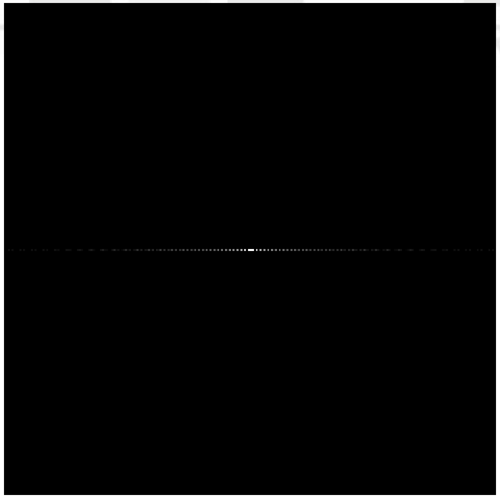
Imagem original



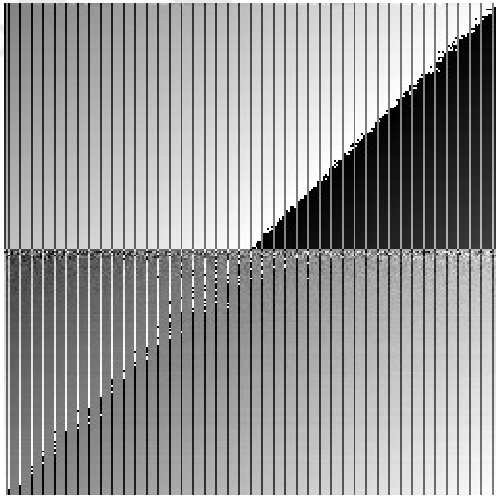
DFT (real)



DFT (imag)



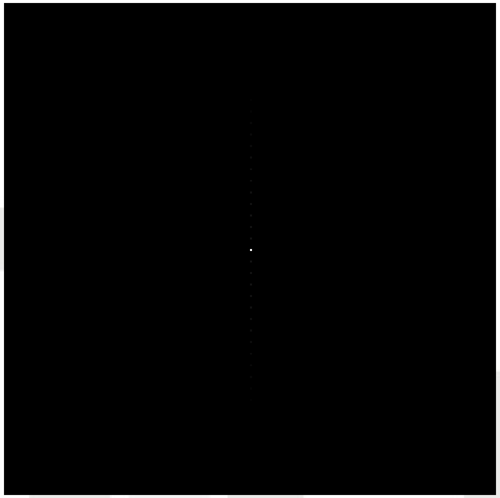
Magnitude



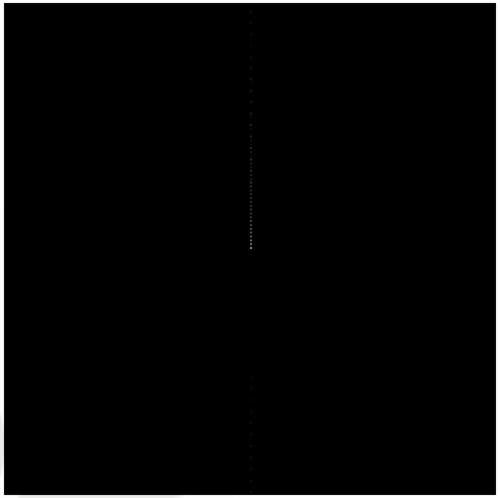
Fase

Exemplo

Imagem original

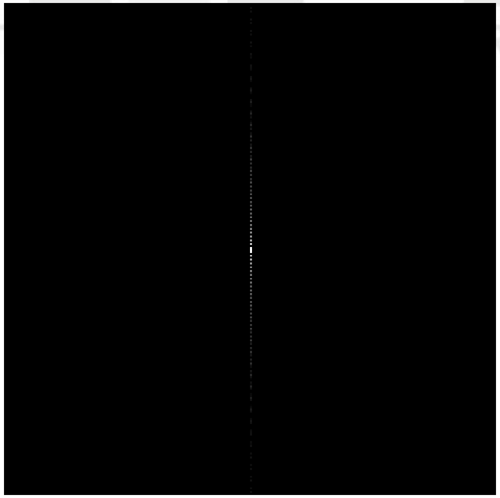


DFT (real)

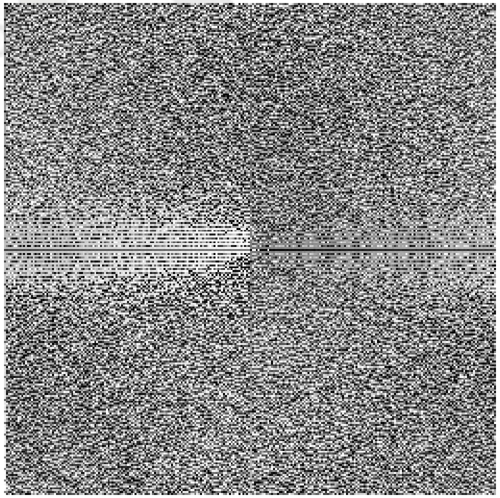


DFT (imag)

!



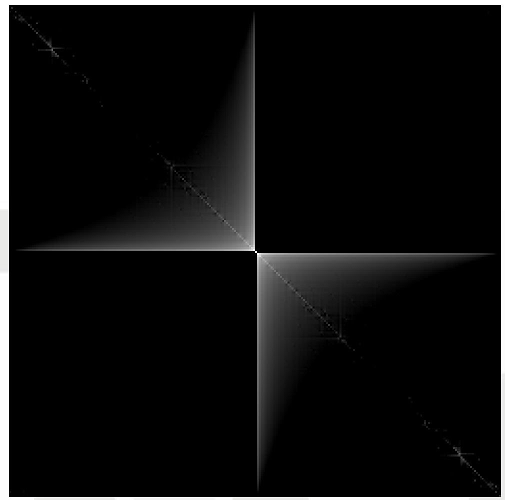
Magnitude



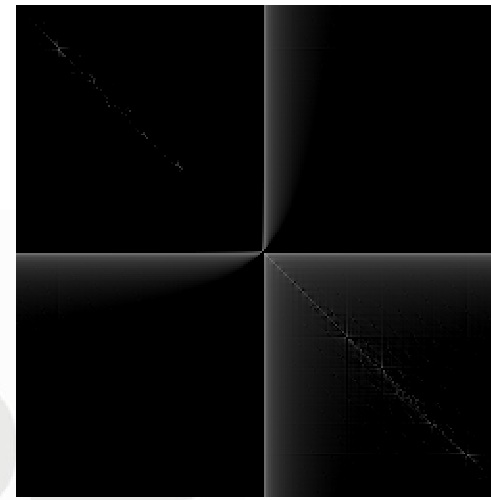
Fase

Exemplo

Imagem original

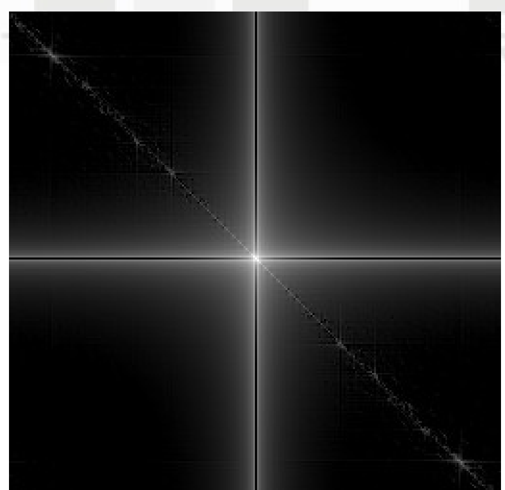


DFT (real)

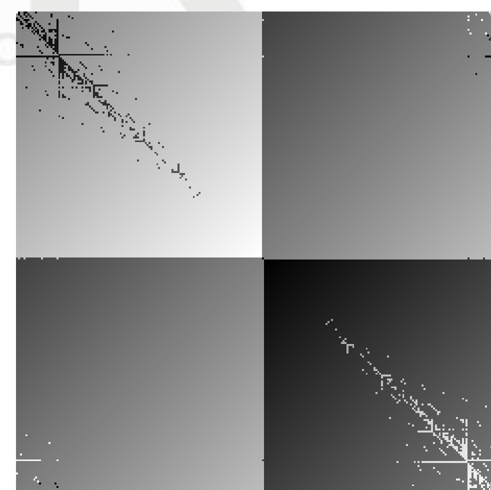


DFT (imag)

!



Magnitude



Fase

Exemplo

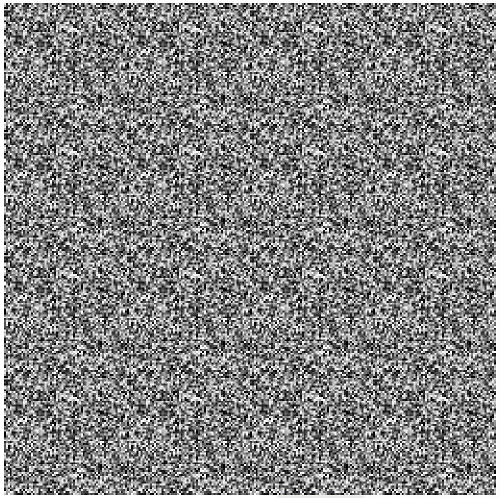
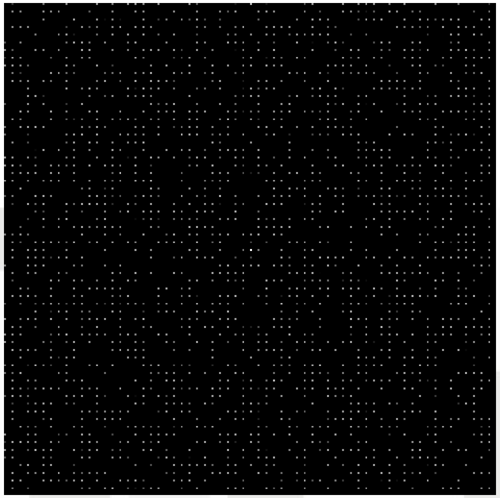
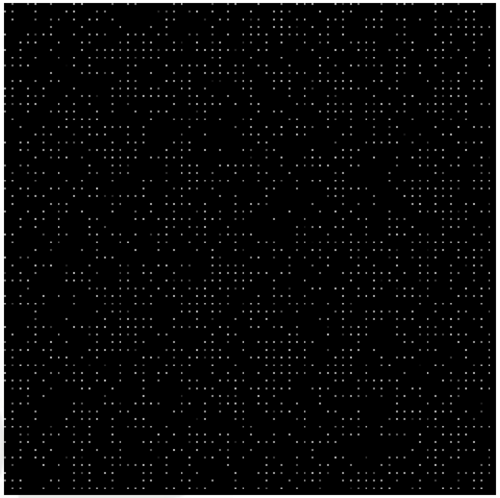


Imagem original

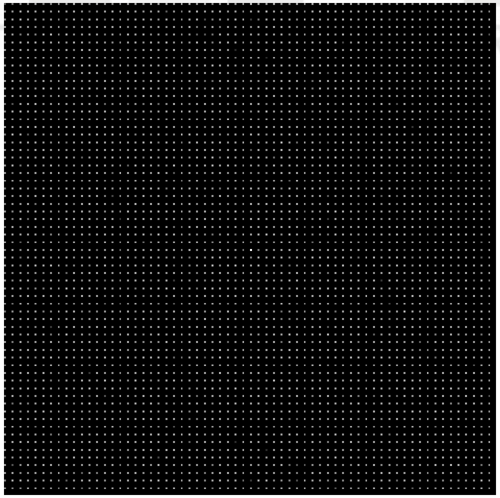


DFT (real)

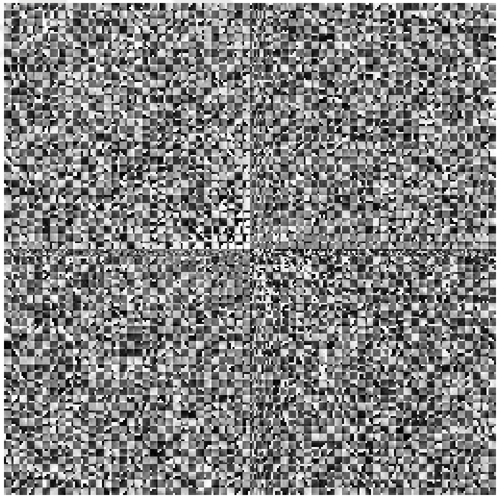


DFT (imag)

O ruído desta imagem é periódico (o padrão se repete 4 vezes).



Magnitude



Fase

Exemplo

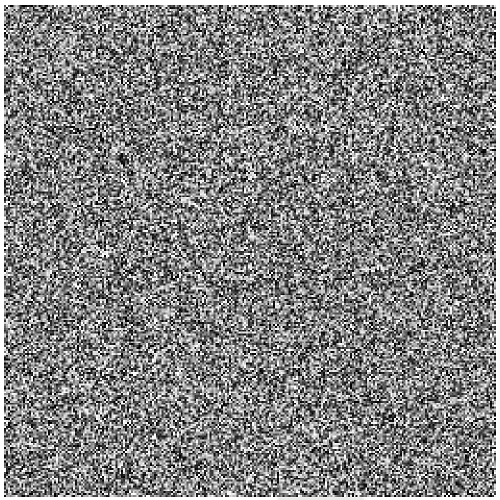
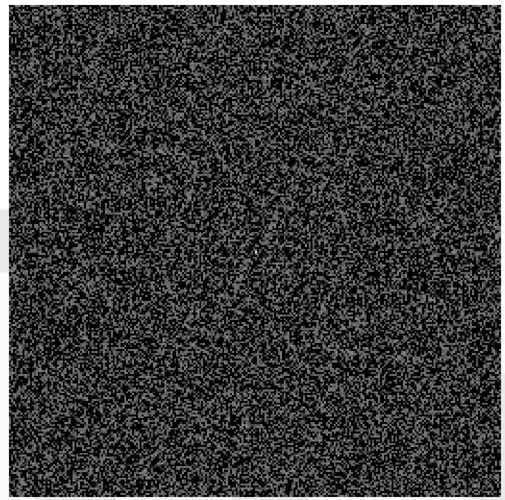
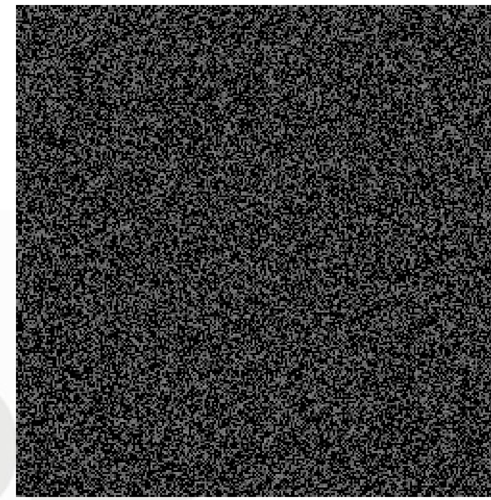
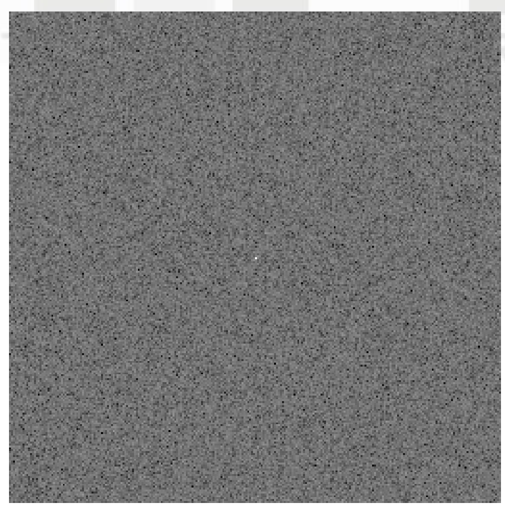


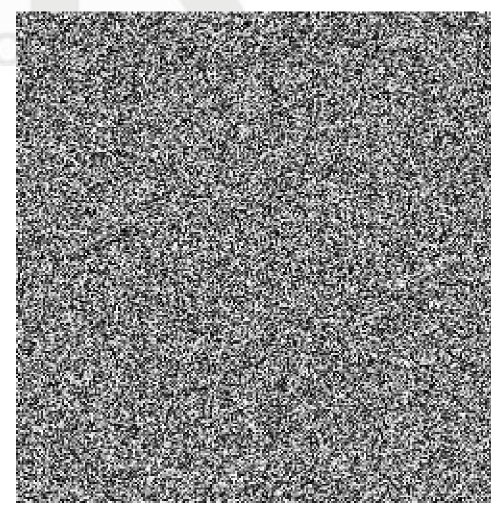
Imagem original


DFT (real)


DFT (imag)

O ruído desta imagem
NÃO é periódico.


Magnitude


Fase

Exemplo

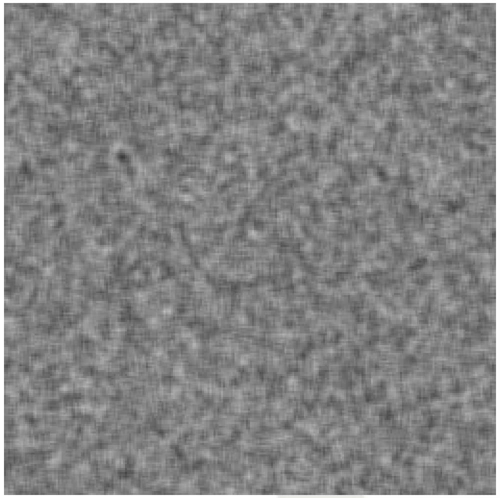
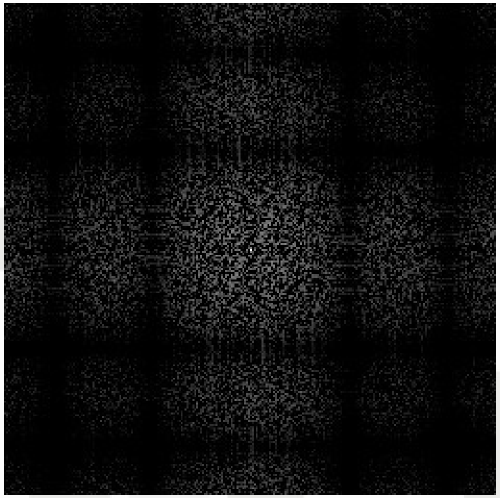
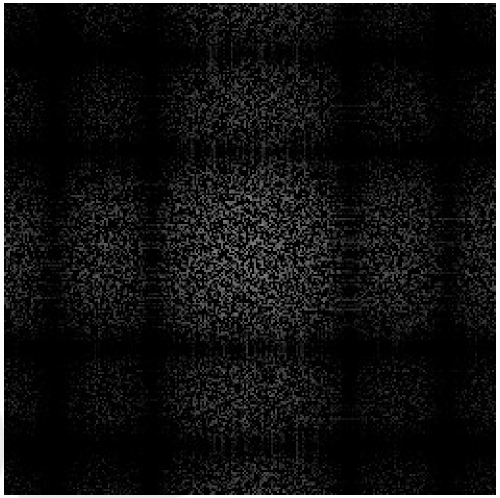


Imagem original

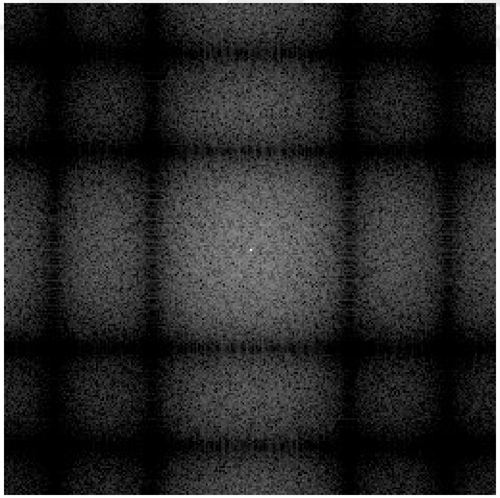


DFT (real)

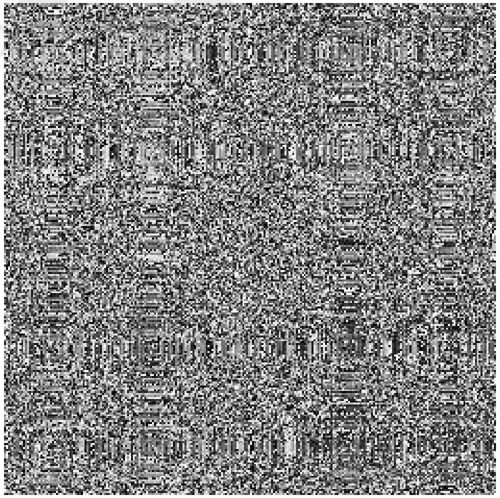


DFT (imag)

Borrando a imagem anterior com um filtro da média 5x5.



Magnitude



Fase

Exemplo

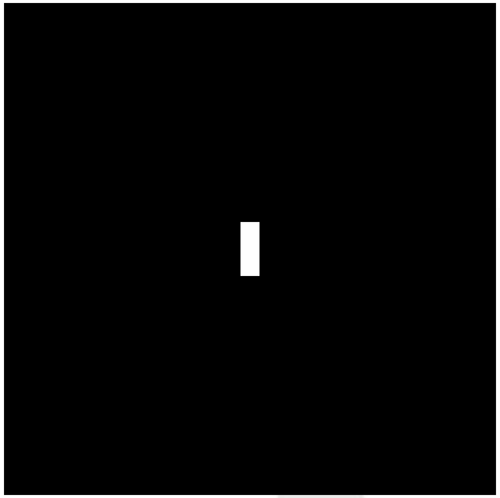
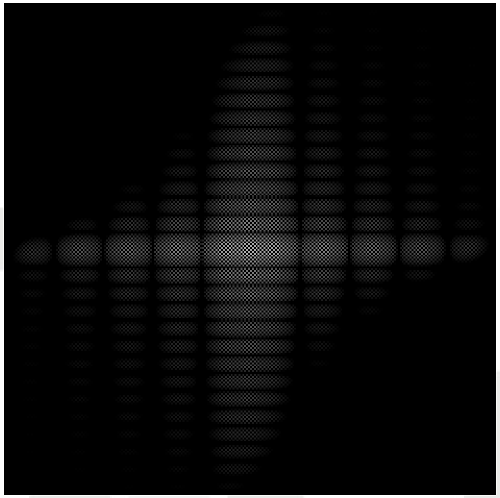
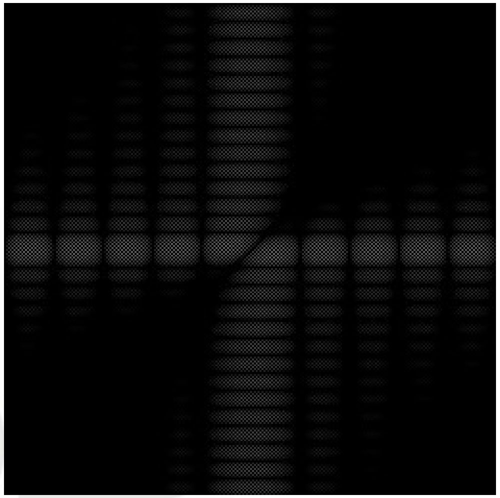
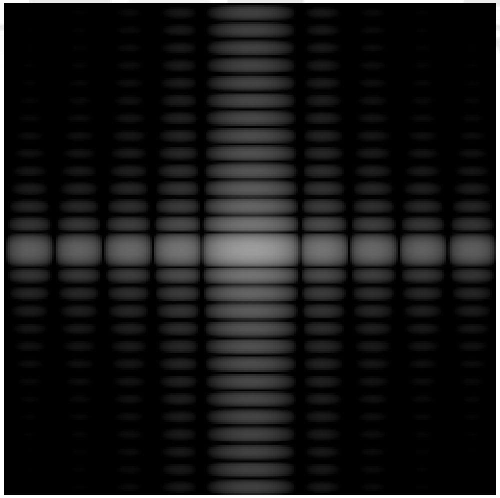
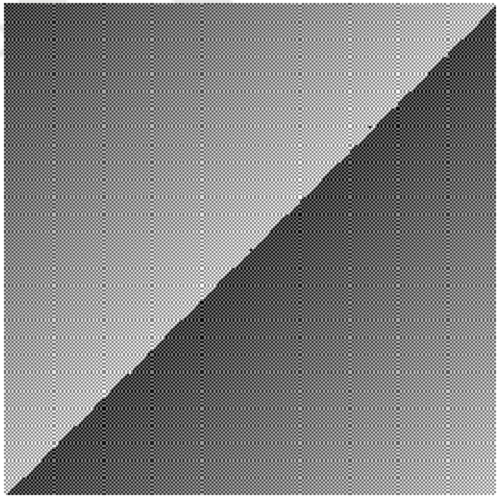


Imagem original


DFT (real)


DFT (imag)


Magnitude


Fase

Exemplo

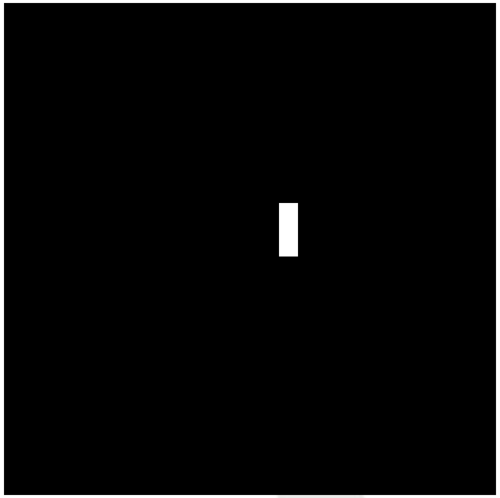
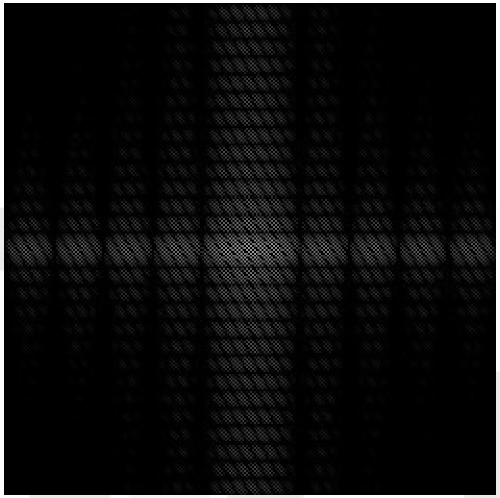
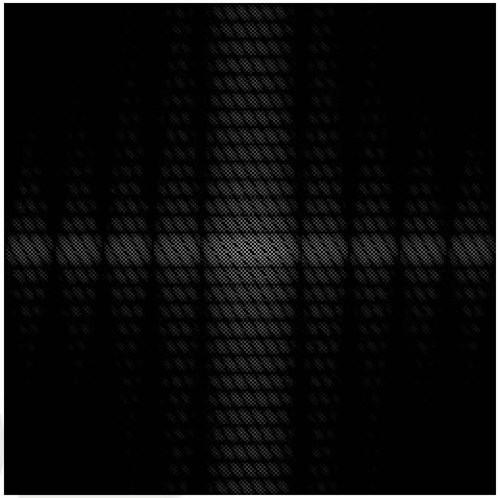


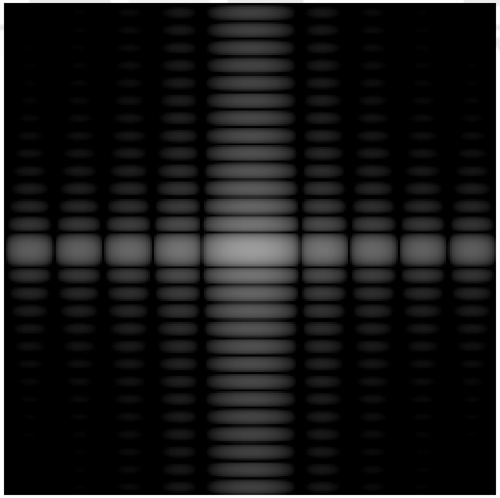
Imagem original



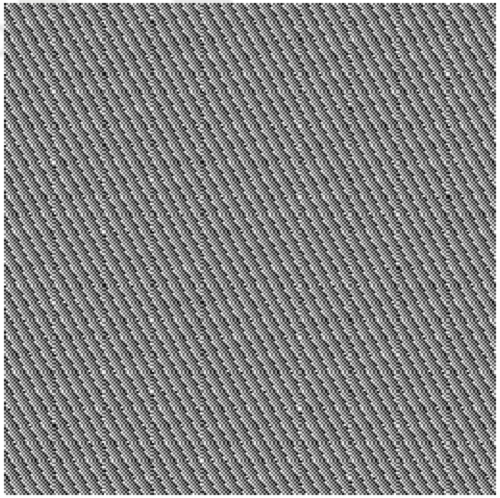
DFT (real)



DFT (imag)



Magnitude



Fase

Exemplo

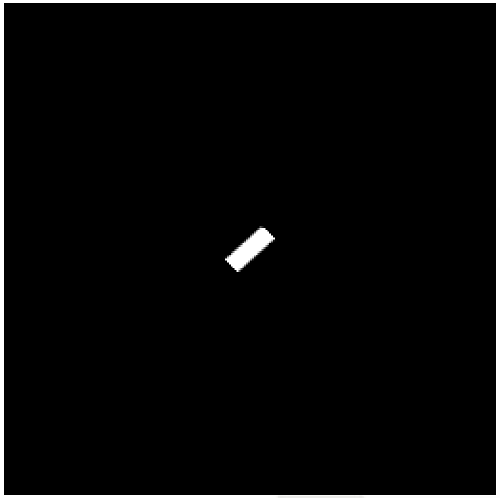
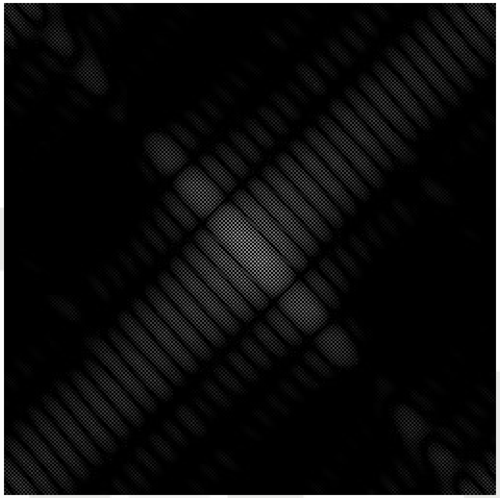
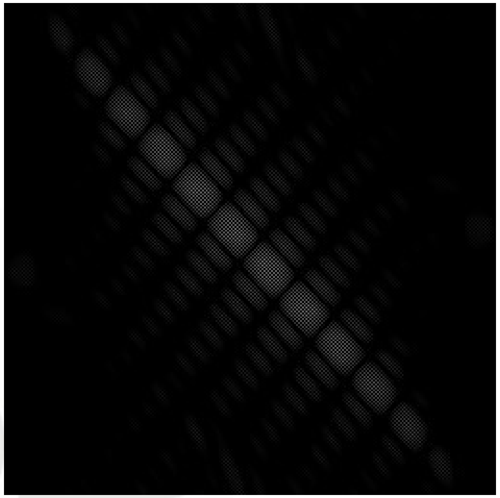


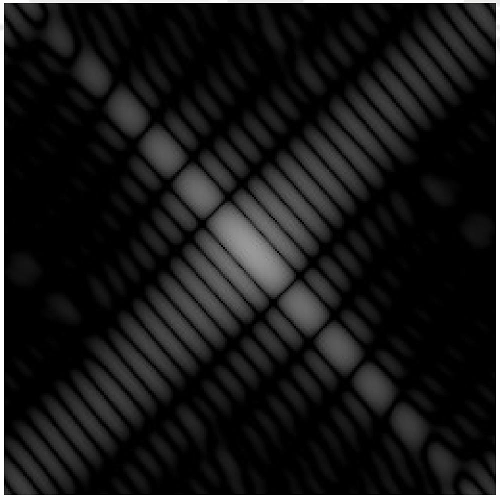
Imagem original



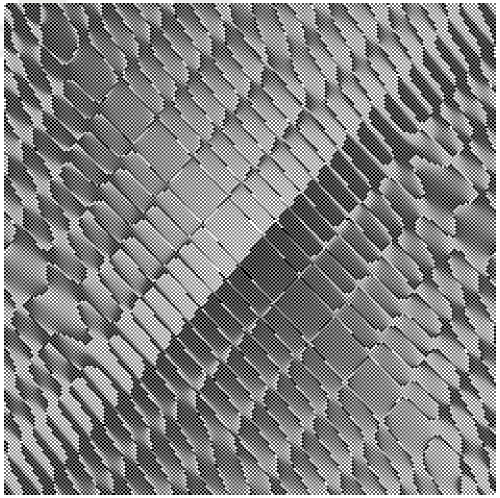
DFT (real)



DFT (imag)



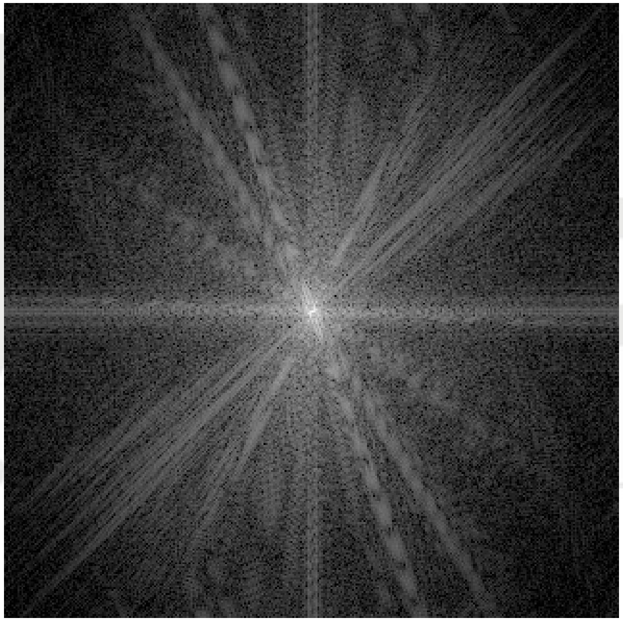
Magnitude



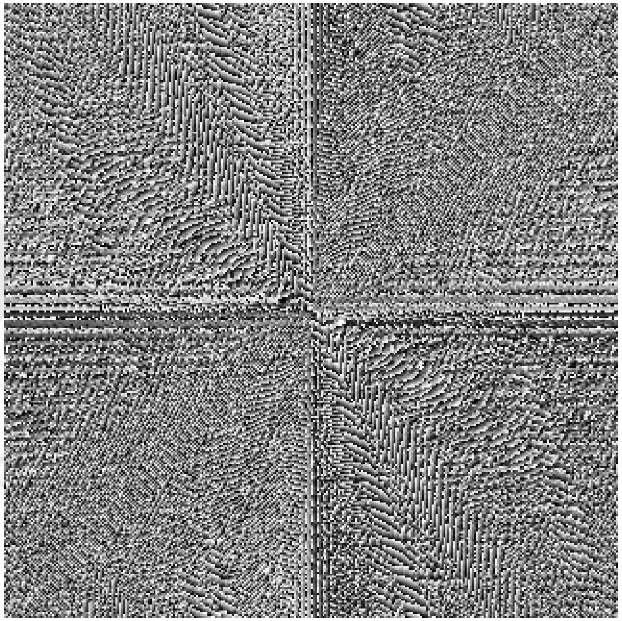
Fase

Exemplo

Imagem original



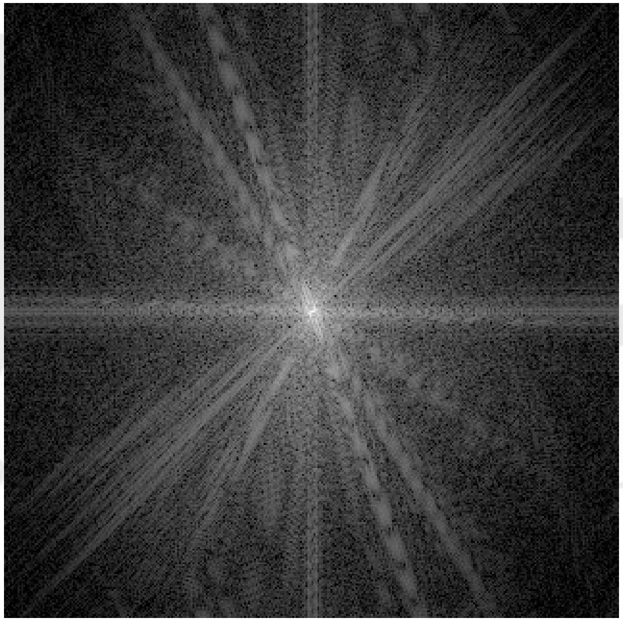
Magnitude



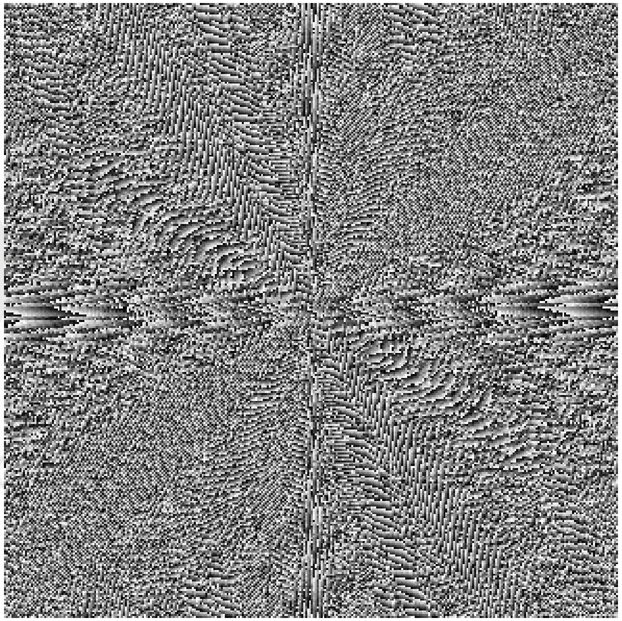
Fase

Exemplo

Imagem original



Magnitude

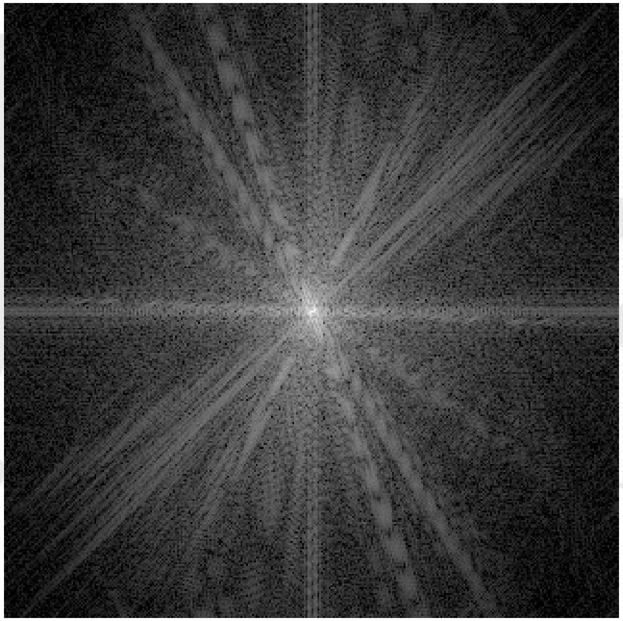


Fase

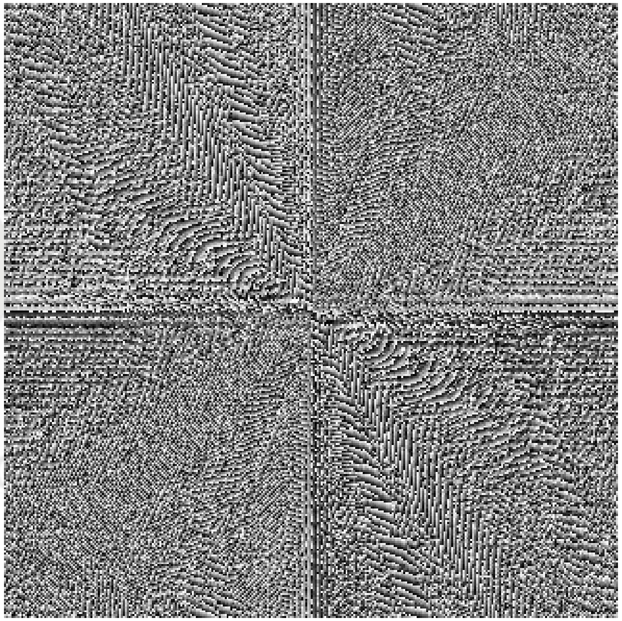
A mesma imagem, deslocada com repetições.

Exemplo

Imagem original



Magnitude

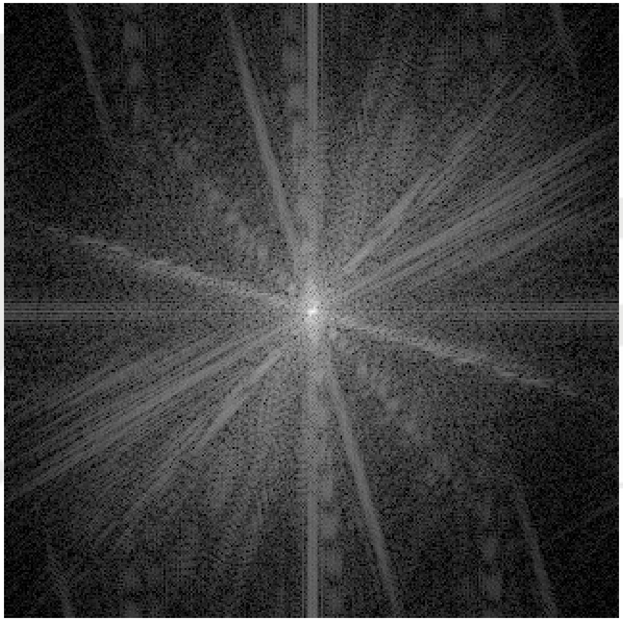


Fase

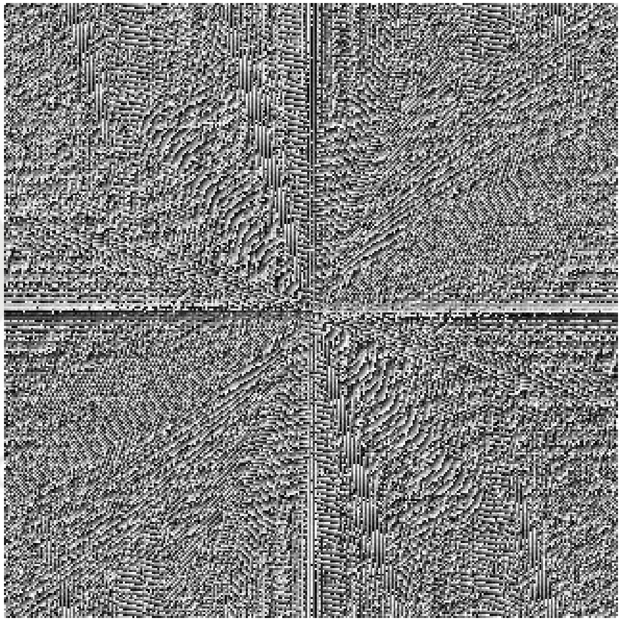
A mesma imagem, deslocada mas com novos dados.

Exemplo

Imagem original



Magnitude

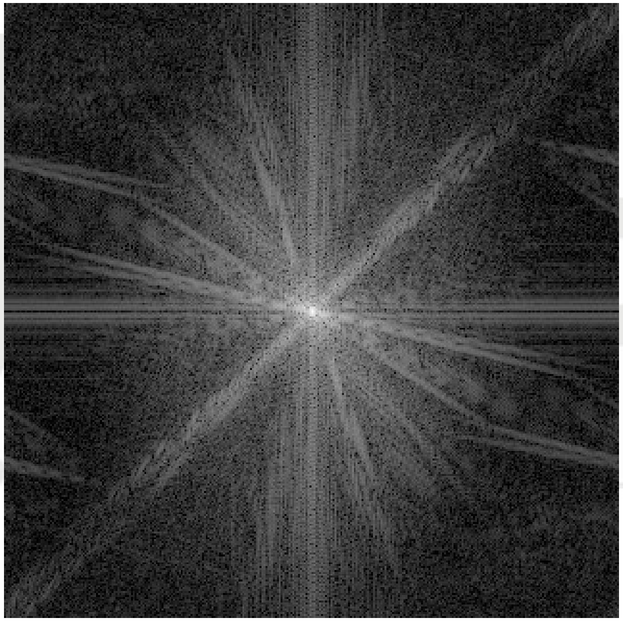


Fase

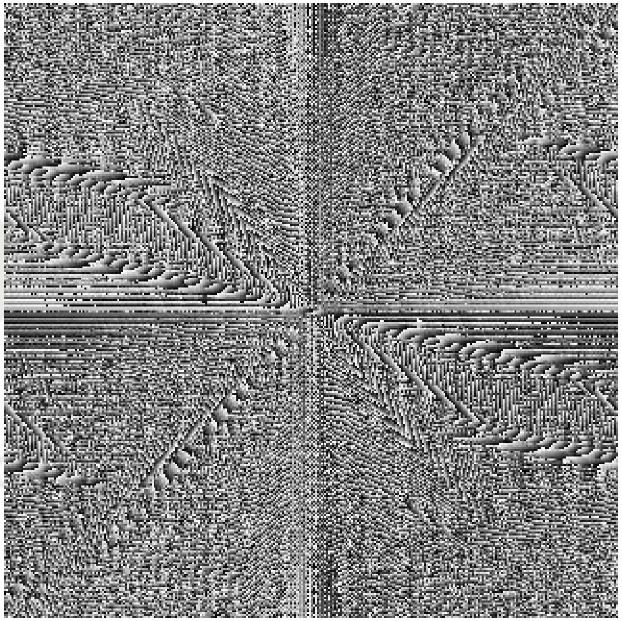
A mesma imagem, rotacionada 15º.

Exemplo

Imagem original



Magnitude



Fase

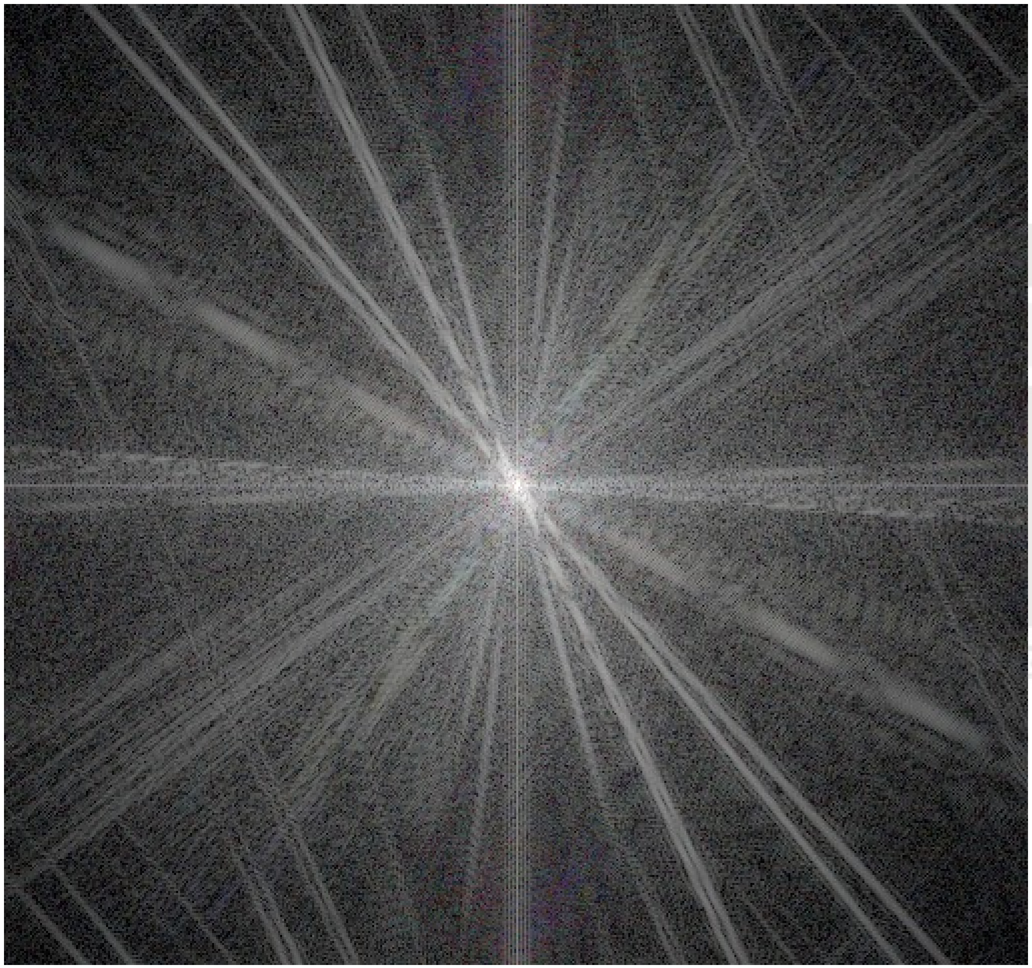
A mesma imagem, rotacionada -45º.

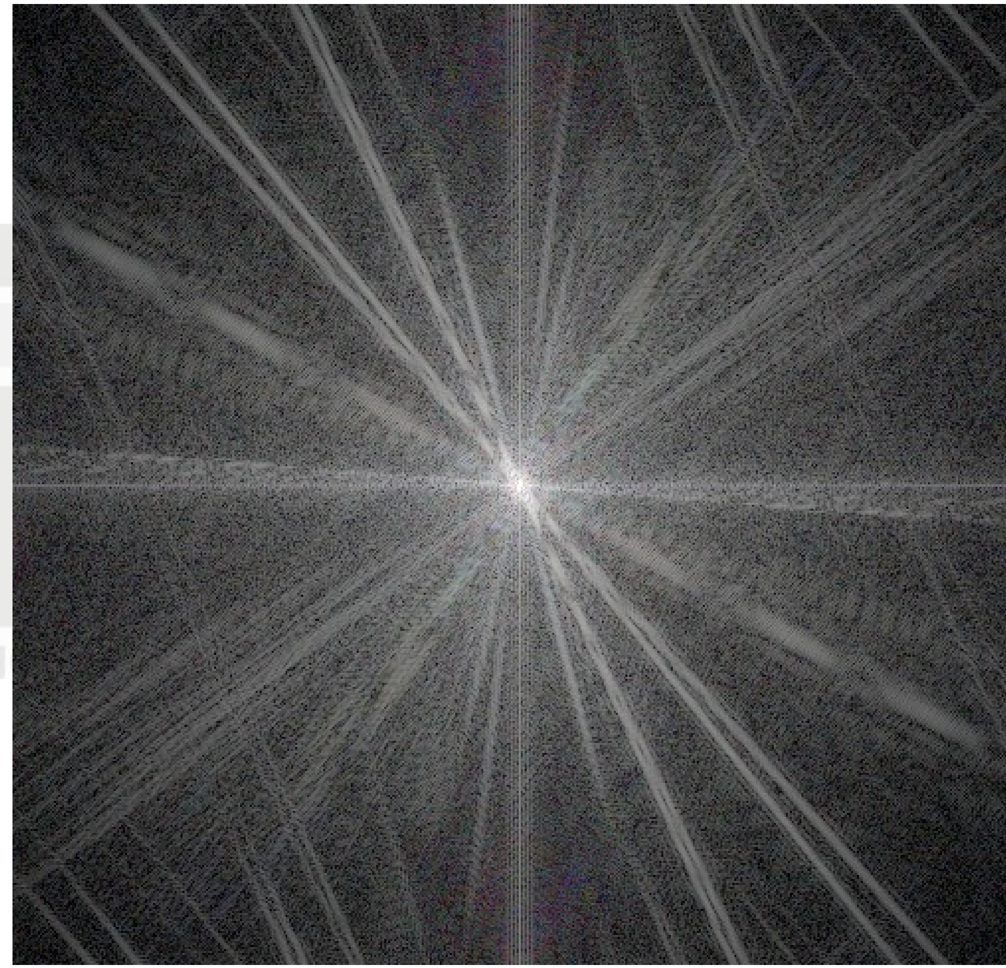
UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Exemplo

Imagem original

Exemplo

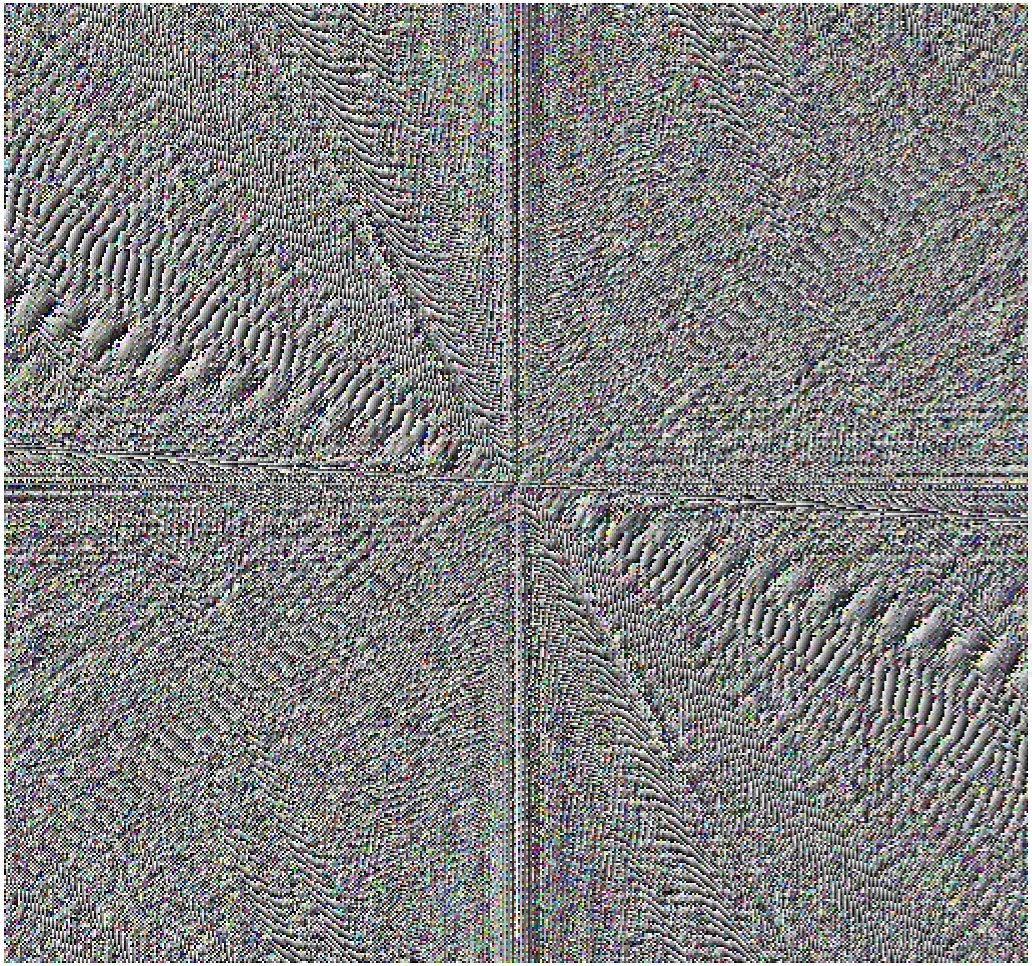


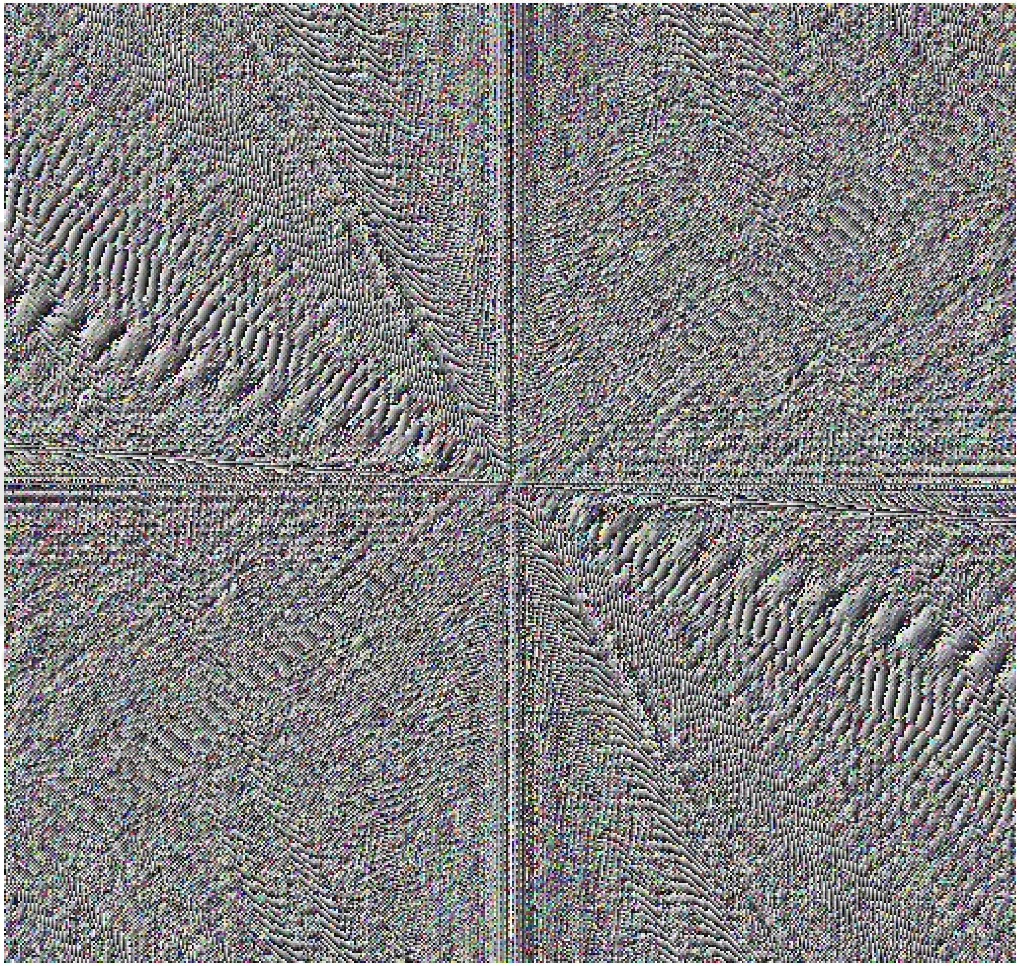


Magnitude

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Exemplo





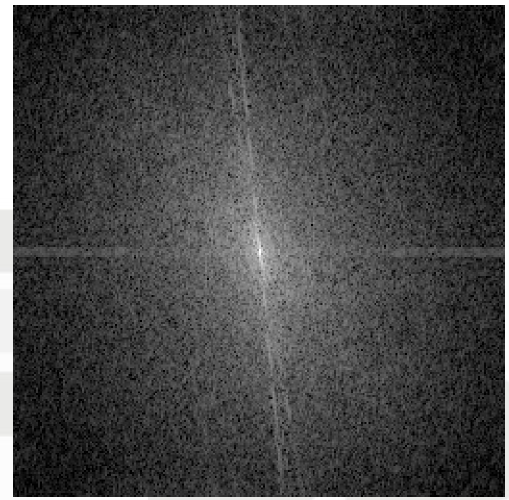
Fase

DFT 2D: analisando

- Como analisar os resultados da DFT 2D?
  - Não é possível associar diretamente a DFT com regiões da imagem.
  - Padrões repetitivos geram valores altos na DFT.
  - Translações não modificam a magnitude, mas afetam a fase.
  - Rotações na imagem fazem a magnitude ficar rotacionada também.
  - Valores altos próximos ao centro.
  - Valores mais próximos do centro = frequências baixas.
  - Valores mais afastados do centro = frequências altas.
- O que significam frequências altas e baixas?

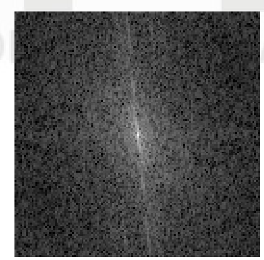
Escala

Imagem original



Magnitude

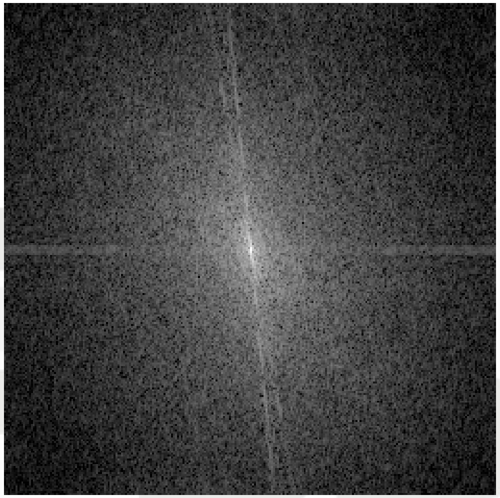
Redimensionada
(NN)



Magnitude

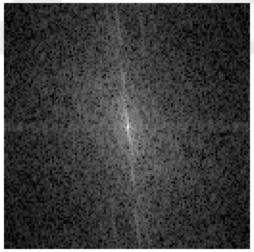
Escala

Imagem original



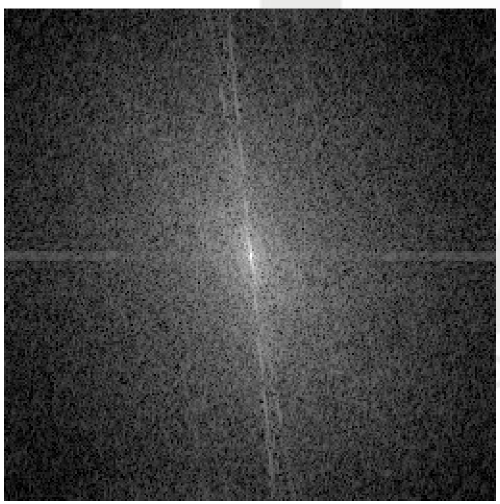
Magnitude

Redimensionada
(bilinear)

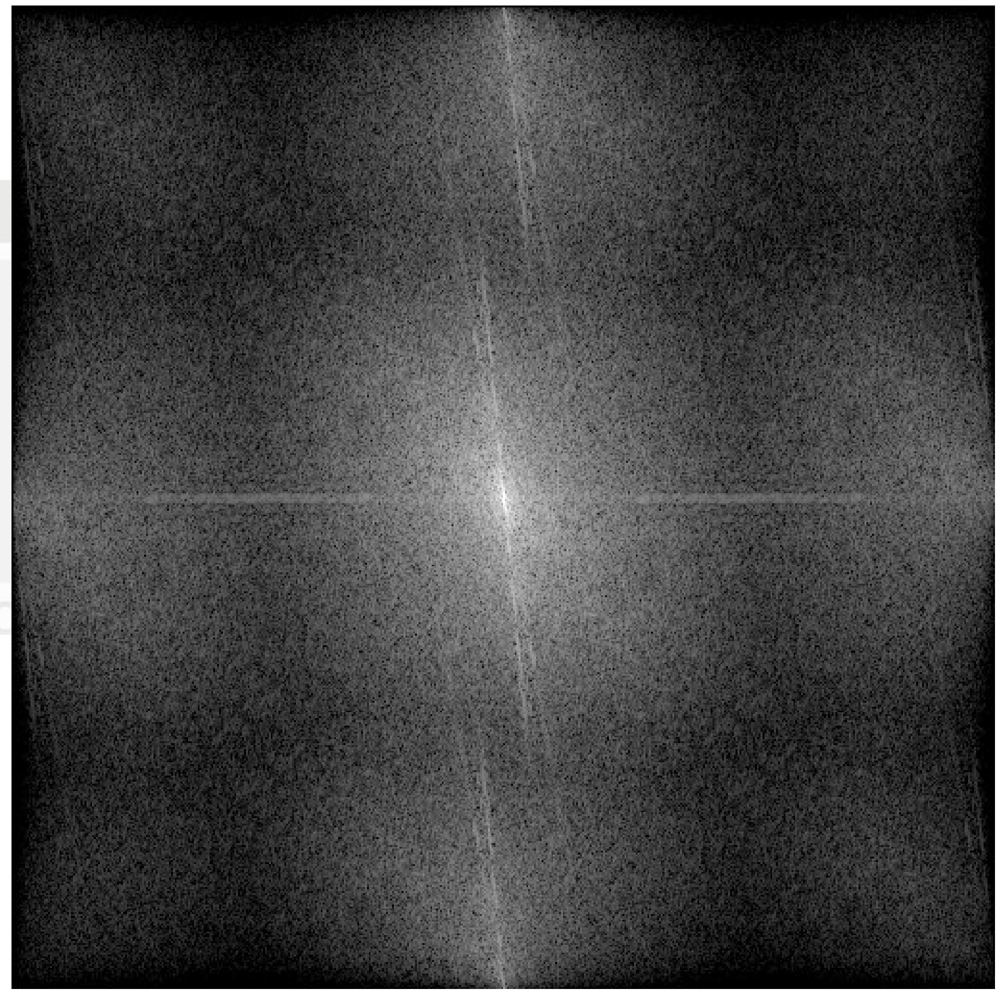


Magnitude

Escala

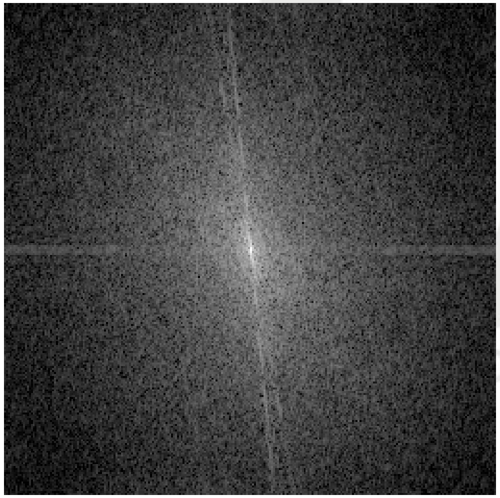


Magnitude (original)

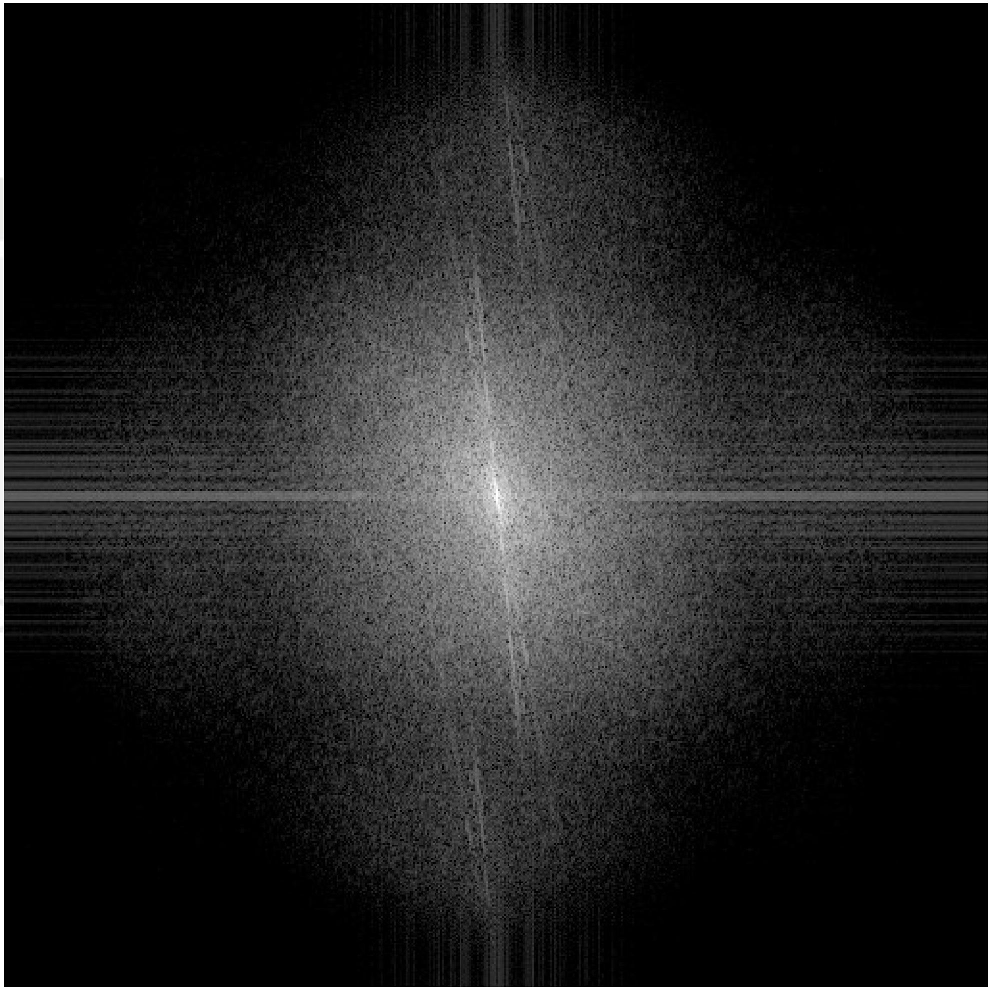


Magnitude (redimensionada, NN)

Escala



Magnitude (original)



Magnitude (redimensionada, bilinear)

Efeito da escala

- Como as mudanças de escala afetam a DFT?

Efeito da escala

- Como as mudanças de escala afetam a DFT?
  - Redução: frequências altas são perdidas.
    - = detalhes são perdidos.
    - Em princípio, metade do tamanho → metade da informação.
  - Ampliação: frequências altas são criadas.
    - = detalhes são criados.
    - Esses detalhes **NÃO** existem na imagem original.
  - Métodos de interpolação reduzem as frequências mais altas.
    - = produzem menos serrilhados.
    - = criam menos detalhes que não existiam na imagem.
- Estas alterações nas frequências são instâncias de *aliasing*.