INTERPOLACIÓN CON PYTHON Y TRANSFORMA DE FOURIER

CONTENIDO

- Interpolación
- Obtener datos
- Extraer muestra
- Mapear
- Interpolación con Python
- Cálculos de Interpolación
- Transformada de Fourier
- Interpolación basada en Transformada de Fourier

Interpolación

- Se denomina en un conjunto de número se determine encontrar un polinomio de grado n.

In [1]:
```
from scipy.interpolate import barycentric_interpolate
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
def runge(x):
    """Función de Runge."""
    return 1 / (1 + x ** 2)
N = 11  # Nodos de interpolación
xp = np.linspace(-5,5,N)
fp = runge(xp)
x = np.linspace(-5, 5,200)
y = barycentric_interpolate(xp, fp, x)
```

Obtener datos

Se obtienen los datos de la siguiente Imagen:

```
int main(){
    //Reader Image
    vtkSmartPointer<vtkPNGReader> reader =
    vtkSmartPointer<vtkPNGReader>::New();
    reader->SetFileName("FFT.png");

    //Filtro FTT
    //implements a fast Fourier transform
    vtkSmartPointer<vtkImageFFT> fftFilter =
    vtkSmartPointer<vtkImageFFT>::New();
    fftFilter->SetInputConnection(reader->GetOutputPort());
    fftFilter->Update();

    //Filter casts the input type to match the output
    //type in the image processing pipeline.
    vtkSmartPointer<vtkImageCast> fftCastFilter =
    vtkSmartPointer<vtkImageCast>::New();
    fftCastFilter->SetInputConnection(fftFilter->GetOutputPort());
    fftCastFilter->SetOutputScalarTypeToUnsignedChar();
    fftCastFilter->Update();
```

Mapear

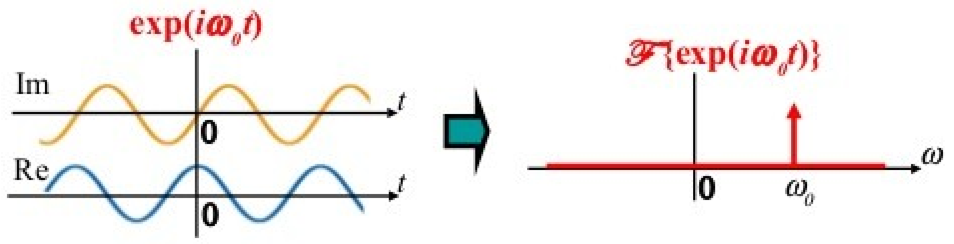
```
//Filter casts the input type to match the output
//type in the image processing pipeline.
vtkSmartPointer<vtkImageCast> fftCastFilter =
vtkSmartPointer<vtkImageCast>::New();
fftCastFilter->SetInputConnection(fftFilter->GetOutputPort());
fftCastFilter->SetOutputScalarTypeToUnsignedChar();
fftCastFilter->Update();

//Implements the reverse fast Fourier transform.
vtkSmartPointer<vtkImageRFFT> rfftFilter =
vtkSmartPointer<vtkImageRFFT>::New();
rfftFilter->SetInputConnection(fftFilter->GetOutputPort());
rfftFilter->Update();

vtkSmartPointer<vtkImageExtractComponents> extractRealFilter =
  vtkSmartPointer<vtkImageExtractComponents>::New();
extractRealFilter->SetInputConnection(rfftFilter->GetOutputPort());
extractRealFilter->SetComponents(0);
extractRealFilter->Update();

vtkSmartPointer<vtkImageCast> rfftCastFilter =
  vtkSmartPointer<vtkImageCast>::New();
rfftCastFilter->SetInputConnection(extractRealFilter->GetOutputPort());
rfftCastFilter->SetOutputScalarTypeToUnsignedChar();
rfftCastFilter->Update();

// Create actors
vtkSmartPointer<vtkImageActor> originalActor =
  vtkSmartPointer<vtkImageActor>::New();
originalActor->GetMapper()->SetInputConnection(reader->GetOutputPort());

vtkSmartPointer<vtkImageActor> fftActor =
  vtkSmartPointer<vtkImageActor>::New();
fftActor->GetMapper()->SetInputConnection(fftCastFilter->GetOutputPort());
```

Cálculos de la Transforma de Fourier

La transformada de Fourier de la onda plana $exp(i\omega_0 t)$

$$F\{e^{i\omega_0 t}\} = \int_{-\infty}^{\infty} e^{i\omega_0 t}\, e^{-i\omega t}\, dt =$$

$$\int_{-\infty}^{\infty} e^{-i(\omega-\omega_0)t}\, dt = 2\pi\,\delta(\omega-\omega_0)$$



La TF de $\exp(i\omega_0 t)$ es una frecuencia pura.

Cálculos de la Transforma de Fourier

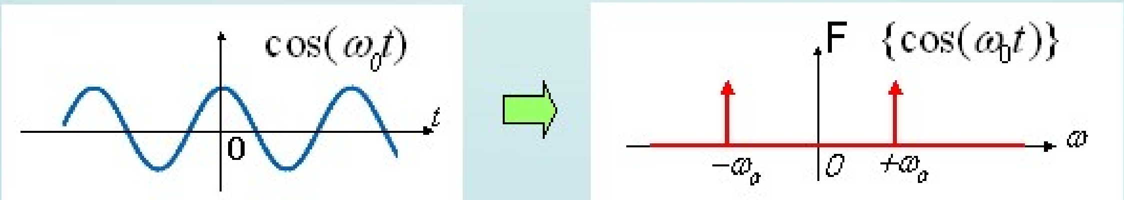
Transformada de Fourier de la función coseno

$$f(t)=\cos(\omega_0 t) \qquad \hat{f}(\omega)=\int_{-\infty}^{\infty}\cos(\omega_0 t)\,e^{-i\omega t}\,dt$$

$$=\int_{-\infty}^{\infty}\left(\frac{e^{i\omega_0 t}+e^{-i\omega_0 t}}{2}\right)e^{-i\omega t}\,dt=\frac{1}{2}\int_{-\infty}^{\infty}\left(e^{-i(\omega-\omega_0)t}+e^{-i(\omega+\omega_0)t}\right)dt=$$

$$\hat{f}(\omega)=\frac{2\pi}{2}\left[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right]$$

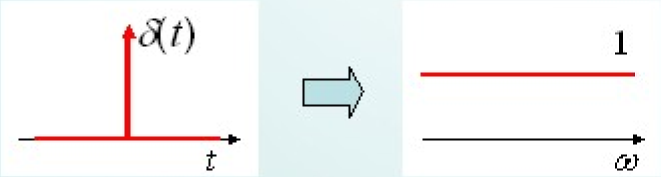
$$\hat{f}(\omega)=\pi\left[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right]$$



Cálculos de la Transforma de Fourier

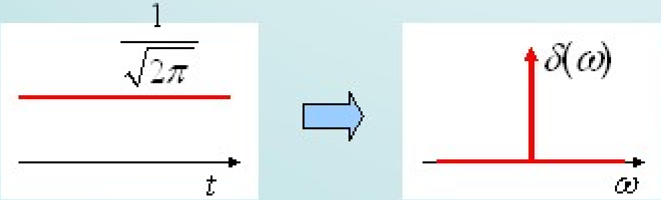
Transformada de Fourier de la δ(t):

$$f(t) = \delta(t) \quad \rightarrow \hat{f}(\omega) = \int_{-\infty}^{\infty} \delta(t)\, e^{-i\omega t}\, dt = 1$$



Observa que la transformada de Fourier de f(t) = 1 es:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} 1\, e^{-i\omega t}\, dt = 2\pi\, \delta(\omega)$$



Recordemos →

Transformada de Fourier