

# Métodos Avanzados de Programación Científica y Computación

Jesús Cigales Canga

## Actividad grupal

# Interpolación polinómica

$$f(x) = \sin(x)$$
$$f(x) = \frac{1}{1 + 25x^2}$$
$$f(x) = e^{-20x^2}$$

1. Número de nodos: 11 y 21
2. Obtener los nodos:
  - a. Equiespaciados
    - \* valores generados  
-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 (11), para representar la función real
    - \*valores equiespaciados para el polinomio interpolador
3. Calcular la función en los 11 puntos

Bibliotecas y funciones

`numpy.arange`

`numpy.linspace`

# Interpolación polinómica

$$f(x) = \sin(x) \quad f(x) = \frac{1}{1 + 25x^2} \quad f(x) = e^{-20x^2}$$

1. Número de nodos: 11 y 21

2. Obtener los nodos:

a. Raíces del polinomio de Chebyshev (elemento 11 ó 21 de la serie `coeficientes_chheb=[0]*11+[1]`)

\*Cálculo de las raíces del polinomio para la función real

`T11 = chebyshev.Chebyshev(coeffs_chheb, [-5, 5])`

`xp_ch = T11.roots()`

\*valores equiespaciados de x para el polinomio interpolador (para las funciones que los necesiten)

3. Calcular la función en las 11 ó 21 raíces

Bibliotecas y funciones

`numpy.polynomial → import chebyshev`

Usar Chebyshev

Usar `roots()`

# Interpolación polinómica

$$f(x) = \sin(x)$$
$$f(x) = \frac{1}{1 + 25x^2}$$
$$f(x) = e^{-20x^2}$$

## 1. Interpolaciones baricéntrica

```
scipy.barycentric_interpolate (nodos,funcion_real, puntos_x)  
y = barycentric_interpolate(xp, fp, x)
```

## 2. Interpolación de Lagrange

```
scipy.interpolate → lagrange()
```

```
from scipy.interpolate import lagrange
```

```
x = np.array([0, 1, 2]) >>> y = x**3 >>> poly = lagrange(x, y)
```

## 3. Diferencias divididas.

```
Implementar en Python (ver tabla)
```

# Splines

$$f(x) = \sin(x)$$
$$f(x) = \frac{1}{1 + 25x^2}$$
$$f(x) = e^{-20x^2}$$

1. Puntos de interpolación (xp,fp): 11 y 21
2. Calcular x para la interpolación (linspace)
3. Interpolante lineal a trozos unidimensional a una función con puntos de datos discretos dados (xp, fp), evaluada en x.

`np.interp(x, xp, yp)`

`InterpolatedUnivariateSpline(xp, yp)(x)`



```
from scipy.interpolate import InterpolatedUnivariateSpline
```





[www.unir.net](http://www.unir.net)