

Investigar funciones de MATLAB:

La función `interp1` en MATLAB (o GNU Octave) se usa para realizar interpolación 1D, y uno de sus usos más comunes es la interpolación lineal segmentaria. Aquí te explico cómo usarla paso a paso.

Código pequeño

```
vq = interp1(x, v, xq)
```

Explicación

x: vector de puntos conocidos (eje x).

v: valores conocidos correspondientes a x (eje y).

xq: puntos donde quieres interpolar.

vq: valores interpolados en los puntos xq.

Esto hace una interpolación lineal segmentaria entre los puntos (2, 4) y (3, 1) para estimar el valor en $xq = 2.5$.

spline

Spline cúbico natural

No explícitamente (usa derivada 0 en extremos por defecto)

Muy simple y directo

csape

Spline cúbico con condiciones personalizadas

✓ Sí

Muy flexible, parte del toolbox Curve Fitting

Código para interpolar

```
yq = interp1(x, y, xq, 'spline');
```

Código para graficar con plot

```
% Datos base
```

```
x = [0 1 2 3 4];
```

```
y = [0 1 0 1 0];
```

```
% Puntos de evaluación
```

```
xq = 0:0.1:4;
```

```
% Interpolación lineal
```

```
yq_linear = interp1(x, y, xq, 'linear');
```

```
% Interpolación spline cúbico (natural)
```

```
yq_spline = interp1(x, y, xq, 'spline');
```

```
% Gráfico
```

```
figure;
```

```
plot(x, y, 'ko', 'MarkerFaceColor', 'k', 'DisplayName', 'Datos'); hold on;
```

```
plot(xq, yq_linear, '-r', 'DisplayName', 'Lineal');
```

```
plot(xq, yq_spline, '-b', 'LineWidth', 1.5, 'DisplayName', 'Spline cúbico');
```

```
legend('Location', 'best');
```

```
xlabel('x');
```

```
ylabel('y');
```

```
title('Interpolación: Lineal vs. Spline Cúbico');
```

```
grid on;
```

Qué muestra este gráfico:

- Puntos negros: datos originales.

- Línea roja discontinua: interpolación lineal.
- Línea azul continua: spline cúbico.

Como resultado, vas a ver que el spline crea una curva suave que pasa por todos los puntos, mientras que la lineal es simplemente segmentos rectos entre ellos.

Documentar:

- Sintaxis básica de cada función.

Interp1:

`vq = interp1(x, v, xq)`

splines cúbicos:

`pp = spline(x, y)`

`pp = csape(x, y, conds)`

plot:

`plot(Y)`

`plot(X, Y)`

`plot(X, Y, LineSpec)`

`plot(X1, Y1, ..., Xn, Yn)`

`plot(X1, Y1, LineSpec1, ..., Xn, Yn, LineSpecn)`

- Parámetros importantes (como tipo de spline: 'natural', 'complete').

1. Spline estándar (por defecto)

- **Uso:** Suave, sin condiciones especiales en extremos.
- **Función:** `spline`
- `pp = spline(x, y);` % crea el spline
- `yy = ppval(pp, xx);` % lo evalúa

2. Spline natural

- **Condición:** Derivada segunda = 0 en los extremos.
- **Función:** `csape` con '`variational`'
- `pp = csape(x, y, 'variational');`
- `yy = ppval(pp, xx);`

3. Spline completo

- **Condición:** Tú eliges la pendiente inicial y final.
- **Función:** `csape` con '`complete`'
- `pp = csape(x, [0, y, 0], 'complete');` % derivadas 0 en extremos
- `yy = ppval(pp, xx);`

Interpolación en MATLAB, agregar código y evidencia en el documento o bien link de repositorio.

