

# MÉTODOS NUMÉRICOS

## GUÍA DE LABORATORIO NRO. 7

### INTERPOLACIÓN EN MATLAB (Parte II)

---

#### OBJETIVOS:

Utilizar comandos y funciones para interpolación.  
Resolver problemas mediante el uso del MATLAB.

#### MEDIOS Y MATERIALES EDUCATIVOS:

Guía de laboratorio, computadora, software de Matlab, tutoriales y manuales de Matlab, apuntes, Internet y flash memory.

#### INFORME:

Realizar un informe del laboratorio realizado, puede ser individual o de un máximo de dos estudiantes.

#### TAREA 1. INTERPOLACIÓN POR SPLINES

a) Analizar el siguiente código y explicar cómo funciona, y cómo se aplica la interpolación:

Calcular el spline cúbico que interpola los datos. Dibujar en la misma ventana el spline calculado junto con el polinomio de interpolación lineal a trozos.

1. Crea dos variables  $x$  e  $y$  con las abscisas y las ordenadas que vas a interpolar:

```
x=[0, 2, 3, 5, 6, 8, 9, 11, 12, 14, 15];  
y=[10, 20, 30, -10, 10, 10, 10.5, 15, 50, 60, 85];
```

2. Calcula el spline cúbico

```
z=linspace(0,15);  
s=spline(x,y,z);
```

3. Dibuja el spline cúbico y la interpolante lineal a trozos junto con los datos:

```
plot(x,y,'or',z,s,x,y,'LineWidth',2)
```

- b) Cuando se calcula un spline cúbico con la función spline es posible cambiar la forma en que éste se comporta en los extremos. Para ello hay que añadir al vector y dos valores extra, uno al principio y otro al final. Estos valores sirven para imponer el valor de la pendiente del spline en el primer punto y en el último. El spline así construido se denomina sujeto. Naturalmente, todos los procedimientos de interpolación antes explicados permiten aproximar funciones dadas: basta con interpolar un soporte de puntos construido con los valores exactos de una función. En este ejercicio se trata de calcular y dibujar una aproximación de la función  $\sin(x)$  en el intervalo  $[0; 10]$  mediante la interpolación con dos tipos distintos de spline cúbico y comparar estos resultados con la propia función. Hay por lo tanto que dibujar tres curvas en  $[0; 10]$ :

1. La curva  $y = \sin(x)$ .
2. El spline que calcula MATLAB por defecto (denominado not-a-knot).
3. El spline sujeto con pendiente = -1 en  $x = 0$  y pendiente = 5 en  $x = 10$ .

**Observación:** Este ejercicio no es imprescindible.

1. Construimos un conjunto de nodos para la interpolación en  $[0; 10]$  y calculamos los valores en estos nodos de la función  $\sin(x)$ . Por ejemplo:

```
x = 0:10;  
y = sin(x);
```

Estos vectores x e y van a ser los puntos soporte para la construcción de los splines.

2. Lo que queremos es dibujar los dos splines y la función. Para ello construimos un vector de puntos en el intervalo  $[0; 10]$  que utilizaremos para las gráficas, calculando en ellos los valores de las tres funciones:

```
z=linspace(0,10);
```

3. Calculamos el valor en estos puntos z del spline not-a-knot:

```
s1=spline(x,y,z);
```

4. Calculamos ahora el valor del spline sujeto. Para ello añadimos al vector y los valores -1 y 5 al principio y al final respectivamente:

```
ys=[-1, y, 5]; % o tambien simplemente  
s2=spline(x,ys,z); % s2 = spline(x,[-1,y,5],z);
```

5. Ahora dibujamos las tres curvas y los puntos del soporte

```
axis([-1,11,-1.4,2]);  
hold on
```

```
plot(x,y,'mo')
h1 = plot(z,sin(z),'Color',[1,0,0]);
h2 = plot(z,s1, 'Color',[0,1,0]);
h3 = plot(z,s2, 'Color',[0,0,1]);
legend([h1,h2,h3],'sen(x)','spline not-a-knot','spline sujeto')
hold off
```