Ampliación de interpolación con Splines

Miguel Anguita Ruiz Pablo Baeyens Fernández Pablo David Medina Sánchez
Ruben Morales Pérez Francisco Javier Morales Piqueras

Índice

1. Splines cuadráticos

El espacio de splines de clase 2 se nota $S_2(x_1, x_2, ..., x_n)$. Una base sería $\{1, x, x^2, (x - x_1)_+, (x - x_2)_+, ..., (x - x_n)_+\}$

1.1. Descripción del espacio de splines cuadráticos

El espacio de splines de clase 2 con n nodos se denota $S_2(x_1, x_2, ..., x_n)$. Los splines de clase 2 están constituidos por parábolas, de forma que además de tener una función continua, su derivada también lo es. Por lo tanto, para $i = 1, ..., n-1$ tenemos la siguiente condición:

$$\boxed{\begin{array}{l} s_i(x_i) = s_{i+1}(x_{i+1}) \\ s_i'(x_i) = s_{i+1}'(x_{i+1}) \end{array}}$$

Proposición 1 *El conjunto $S_2(x_1, x_2, ..., x_n)$ satisface las propiedades siguienes:*

1. *Es un espacio vectorial con* $\dim(S_2(x_1, x_2, ..., x_n))$

1.2. Ejemplos

2. Splines cúbicos

2.1. Construcción a partir de los valores de $s''(x)$ en los nodos $\{x_i\}$

2.2. Propiedades de minimización

2.2.1. Cota de error en los splines cúbicos

Teorema 1 *Sea* $f \in C^4([a,b])$, $n \in \mathbb{N}$, $P = \{x_i\}_{i=0\ldots n} \in \mathscr{P}([a,b])$ *y* $s \in S_3^1(P)$ *spline para* f*. Además, sean* $h = max\{x_i - x_{i-1}\}_{i=1\ldots n}$, $M > 0$ *cota superior de* $|f^{iv)}|$ *en* $[a,b]$, $E = f - s$, $x \in [a,b]$.

Se verifica:

$$|E(x)| \leq \frac{5M}{384} h^4 \tag{1}$$

La prueba, así como cotas para las derivadas, puede consultarse en *Optimal Error Bounds for Cubic Spline Interpolation*, Charles Hall y Weston Meyer, (1976).

2.3. Ejemplos

3. Implementación en ordenador: Octave

Hemos implementado las siguientes funciones en Octave:

1. `SplineLineal`: Calcula spline **lineal**. *(Usado en los splines cúbicos)*
2. `Spline31` : Calcula spline de **clase 1**.
3. `SplineNat`: Calcula spline **natural**.
4. `SplinePer`: Calcula spline **periódico**.
5. `SplineSuj`: Calcula spline **sujeto**.
6. `SplineCuad`: Calcula spline **cuadrático** de clase 1.

3.1. Spline Lineal

La función que nos permite calcular un spline lineal es muy

```
function s = SplineLineal(x,y)
  p = diff(y)./diff(x);
  A = [p' y(1:end-1)'];
  s = mkpp(x,A);
end
```

3.2. Splines cuadráticos

Utilizando el sistema que vimos anteriormente, podemos definir fácilmente una función que calcule los coeficientes de un spline cuadrático de clase 1:

```
function s = coefsSplineCuad(x, y, d_k, k)
  # Número de intervalos
  n = length(x) - 1;

  # 1, x, x²
  A(:,1) = [ones(n+1,1); 0];
  A(:,2) = [x'         ; 1];
  A(:,3) = [x'.^2      ; 2.*x(k+1)];

  # Potencias truncadas
  for j = 4 : n + 2
    pot    =  @(t) (t > x(j-2)) .* (t - x(j-2));
    A(:,j) = [pot(x').^2; 2.*pot(x(k+1))];
  end

  # Resolución del sistema
  s = A \ [y' ; d_k];

end
```

Definimos una función `pot` correspondiente a la potencia truncada en el valor correspondiente: `pot = @(t) (t > x(j-2)) .* (t - x(j-2))`. Como *Octave* tiene tipos dinámicos convertirá `(t > x(j-2))` a 1 o 0. De esta forma, $pot(x) = (x - x_{j-1})_+$.