Interpolación

Interpolación de Lagrange

Dados los datos

x_1	x_2	x_3	$\dots$	x_n
y_1	y_2	y_3	$\dots$	y_n

con $x_i \neq x_j, \quad \forall i \neq j$, sabemos que existe un único polinomio p de grado menor o igual que $n-1$ tal que

$$p(x_i) = y_i.$$

El polinomio interpolante de Lagrange se obtiene como:

$$p(z) = \sum_{i=1}^{n} y_i l_i(z) = \sum_{i=1}^{n} y_i \prod_{k=1, k\neq i}^{n} \frac{z - x_k}{x_i - x_k}.$$

Algoritmo 1. *Dados los vectores $x, y \in \mathbb{R}^n$, este algoritmo evalúa el polinomio que interpola los datos (x_i, y_i) en el punto $z \in \mathbb{R}$.*

```
   s=0;
for i=1:n
   l=1;
   for k=1:i-1
    l=l*(z-x(k))/(x(i)-x(k));
   end
   for k=i+1:n
    l=l*(z-x(k))/(x(i)-x(k));
   end
   s=s+y(i)*l;
end
```

Interpolación de Newton y diferencias divididas

Algoritmo 2. *Dados los vectores $x, y \in \mathbb{R}^n$, este algoritmo calcula los coeficientes del polinomio interpolante mediante diferencias divididas. Luego, evalúa el polinomio que interpola los datos (x_i, y_i) en el punto $z \in \mathbb{R}$.*

```
  for i=2:n
    for j=n:-1:i;
     y(j) = (y(j)-y(j-1)) / (x(j)-x(j-i+1)) ;
    end
end
```

```
  p = y(n) ;
    for i=n-1:-1:1
     p = y(i)+(z-x(i))*p;
    end
w = p ;
```