

1.6.1 Programas y funciones

- Ejemplo 3: Cálculo del polinomio de Jacobi de grado p para el vector $[x_1 \cdots x_n]$ dados los coeficientes α y β
- Se define $J_p(x, \alpha, \beta, p)$ con una fórmula de recurrencia:

$$J_{-1}=1$$

$$J_0=1$$

$$J_i=(x-g_i)J_{i-1}-h_i \cdot J_{i-2}$$



1.5.1 Programas y funciones

- Los parámetros g_i y h_i dependen del orden del polinomio:

$$i=1 \quad g_i = \frac{\beta + 1}{\alpha + \beta + 2}$$

$$h_i = 0$$

$$i=2 \quad g_i = 0,5 \left(1 - \frac{\alpha^2 - \beta^2}{(2i + \alpha + \beta - 1)^2 - 1} \right)$$

$$h_i = \frac{(\alpha + 1)(\beta + 1)}{(\alpha + \beta + 2)^2 (\alpha + \beta + 3)}$$

1.5.1 Programas y funciones

$$i \geq 3 \quad \left\{ \begin{array}{l} g_i = 0,5 \left(1 - \frac{\alpha^2 - \beta^2}{(2i + \alpha + \beta - 1)^2 - 1} \right) \\ h_i = \frac{(i-1)(i+\alpha-1)(i+\beta-1)(i+\alpha+\beta-1)}{(2i+\alpha+\beta-1)(2i+\alpha+\beta-2)^2(2i+\alpha+\beta-3)} \end{array} \right.$$

- Hay que sumar 2 a todos los subíndices de J para evitar subíndices ≤ 0

`jacobi.m`

```
function y=jacobi(x,alfa,beta,p)
```

```
y=ones(p+2,length(x));
```

1.5.1 Programas y funciones

```
y(1,:)=1; %El polinomio de grado -1
```

```
y(2,:)=1; %El polinomio de grado 0
```

```
for i=3:p+2
```

```
    k=i-2;
```

```
    if k==1      i=1       $g_i = \frac{\beta + 1}{\alpha + \beta + 2}$ 
```

```
        g=(beta+1)/(alfa+beta+2);
```

```
        h=0;
```

```
    elseif k==2
```

```
        g=0.5*(1-(alfa^2-beta^2)/((2*k+alfa+beta-1)^2-1));
```

```
        h=(alfa+1)*(beta+1)/(alfa+beta+2)^2/(alfa+beta+3);
```

$$i=2 \quad g_i = 0,5 \left(1 - \frac{\alpha^2 - \beta^2}{(2i + \alpha + \beta - 1)^2 - 1} \right)$$

$$h_i = \frac{(\alpha + 1)(\beta + 1)}{(\alpha + \beta + 2)^2 (\alpha + \beta + 3)}$$

1.5.1 Programas y funciones

else

$g = 0.5 * (1 - (\alpha^2 - \beta^2) / ((2 * k + \alpha + \beta - 1)^2 - 1));$

$h = (k - 1) * (k + \alpha - 1) * (k + \beta - 1) * (k + \alpha + \beta - 1) \dots$

$/(2 * k + \alpha + \beta - 1) / (2 * k + \alpha + \beta - 2)^2 \dots$

$/(2 * k + \alpha + \beta - 3);$

end

$y(i,:) = (x - g) .* y(i-1,:) - h * y(i-2,:);$

end

$y = y(n+2,:) \% \text{sólo queremos el polinomio de orden } p$

$$i \geq 3 \quad \left\{ \begin{array}{l} g_i = 0,5 \left(1 - \frac{\alpha^2 - \beta^2}{(2i + \alpha + \beta - 1)^2 - 1} \right) \\ h_i = \frac{(i-1)(i+\alpha-1)(i+\beta-1)(i+\alpha+\beta-1)}{(2i+\alpha+\beta-1)(2i+\alpha+\beta-2)^2(2i+\alpha+\beta-3)} \end{array} \right.$$



PROGRAMA: jacobi.m

```
x=linspace(0,1,100);  
m=10*randn;  
'el grado del polinomio a calcular es'  
p=round(abs(m))  
alfa=1;  
beta=1;  
y=fjacobi(x,alfa, beta,p)  
plot(x,y)
```


FUNCION: fjacobi.m

```
function y=fjacobi(x,alfa,beta,p);
y=ones(p+2,length(x));
%Inicialización de y de acuerdo con el tamaño de x. Como el polinomio
%comienza en grado -1 hasta n, hace falta una matriz de p+2 elementos%

y(1,:)=1; %Polinomio de grado -1%
y(2,:)=1; %Polinomio de grado 0%

for i=3:p+2
    k=i-2;

    if k==1
        g=(beta+1)/(alfa+beta+2);
        h=0;
    elseif k==2
        g=0.5*(1-(alfa^2-beta^2)/((2*k+alfa+beta-1)^2-1));
        h=(alfa+1)*(beta+1)/(alfa+beta+2)^2/(alfa+beta+3);
    else
        g=0.5*(1-(alfa^2-beta^2)/((2*k+alfa+beta-1)^2-1));
        h=((k-1)*(k+alfa-1)*(k+beta-1)*(k+alfa+beta-1))/((2*k+alfa+beta-1)*(2*k+alfa+beta-2)^2*(2*k+alfa+beta-3));

    end
    y(i,:)=(x-g).*y(i-1,:)-h*y(i-2,:);
end
y=y(p+2,:); %Sólo necesitamos el polinomio de grado p%
```