Métodos de Computación Científica

Trabajo práctico de programación #4

24-10-2012

Victoria Martínez de la Cruz - LU. 87620

1)

a) El polinomio de interpolación de Lagrange fue generado a partir de la utilización de dos funciones auxiliares: coeficiente_lagrange e interpolacion_lagrange.

coeficiente_lagrange.m

```
function li = coeficiente_lagrange(x,i)
% li = coeficiente_lagrange(x,i) calcula el coeficiente de lagrange li
% a partir de n+1 puntos base x = [ x0 x1 ... xn ]

n = length(x);
li = [ 1 ];
for j=1:n
   if i~=j
      li = conv(li,([1 -x(j)])/(x(i)-x(j)));
   end
end

%conv - convolution is the same operation of multiplying the polynomials whose
coefficients are the elements of u and v

%La función anterior devuelve el polinomio según el formato utilizado por Matlab, es
%decir, un vector fila de n+1 componentes que almacena los coeficientes del polinomio
%ordenados de mayor a menor grado.
```

interpolacion_lagrange.m

```
function p = interpolacion_lagrange(x, y, f)
% p = interpolacion_lagrange(x,y,f) calcula el polinomio de interpolacion
% de lagrange realizando la sumatoria de los coeficientes
% y el valor de la funcion en un punto dado
% x vector que representa las absisas
% y vector que representa las respectivas ordenadas
% f matriz de puntos
% p solucion de la interpolacion de lagrange
p = 0;
m = length(x);
n = length(y);
for i = 1 : m
    for j = 1 : n
        p = p + (coeficiente_lagrange(x,i) .* coeficiente_lagrange(y,j) .* f(i,j));
```

```
    end
end
```

La función interpolacion_lagrange hace uso de coeficiente_lagrange para el cálculo de los respectivos coeficientes y efectúa la sumatoria con los datos de la función en los puntos dados.

```
>> x = [0 1 2]

x =

     0     1     2

>> y  = [0 2 4]

y =

     0     2     4

>> f = [1 2.7183 7.3891; 7.3891 20.0855 54.5982; 54.5982 148.4132 403.4288]'

f =

    1.0000    2.7183    7.3891
    7.3891   20.0855   54.5982
   54.5982  148.4132  403.4288

>> p = interpolacion_lagrange(x,y,f)

p =

    7.5325   -1.6970    1.0000
```

b) Haciendo uso de la función `polyval`, estimamos el valor de f en x = 1.5 e y = 3.0.

```
>> polyval(p,[1.5 3.0])

ans =

   15.4028   63.7019
```

2) Para computar el error E asociado a cada interpolación usamos la función error.m

Métodos de Computación Científica

Trabajo práctico de programación #4

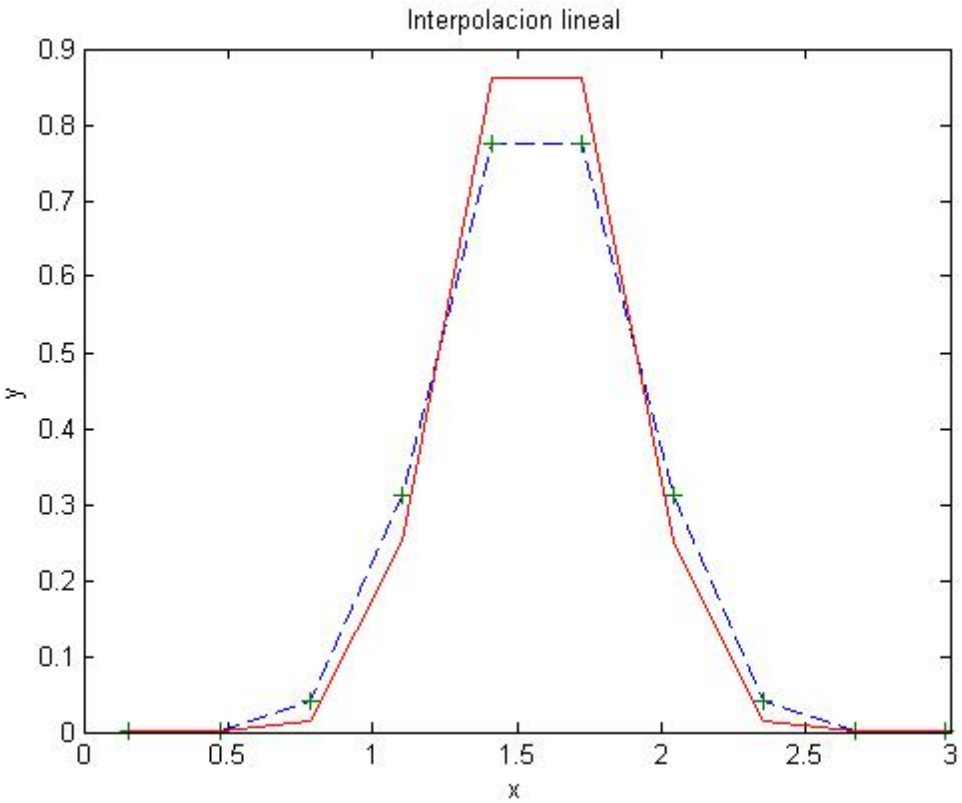
24-10-2012

```
function E = error(y, yi)
% error calcula el error asociado a cada interpolación
% y fi exacto
% yi fi interpolado
E = 0;
for k=1:10
    E = E + (y(k) - yi(k)).^2;
end
```

a) Interpolación lineal

```
>> x=0:pi/10:pi;
>> y=(sin(x)).^12;
>> datos=[x' y']

datos =

         0         0
    0.3142    0.0000
    0.6283    0.0017
    0.9425    0.0786
    1.2566    0.5476
    1.5708    1.0000
    1.8850    0.5476
    2.1991    0.0786
    2.5133    0.0017
    2.8274    0.0000
    3.1416    0.0000

>> xi = pi/20:pi/10:pi-pi/20;
>> yi=interp1(x,y,xi);
>> yy=(sin(xi)).^12;
>> inter1=[xi' yy' yi']

inter1 =

    0.1571    0.0000    0.0000
    0.4712    0.0001    0.0009
    0.7854    0.0156    0.0402
    1.0996    0.2504    0.3131
    1.4137    0.8619    0.7738
    1.7279    0.8619    0.7738
    2.0420    0.2504    0.3131
    2.3562    0.0156    0.0402
    2.6704    0.0001    0.0009
```

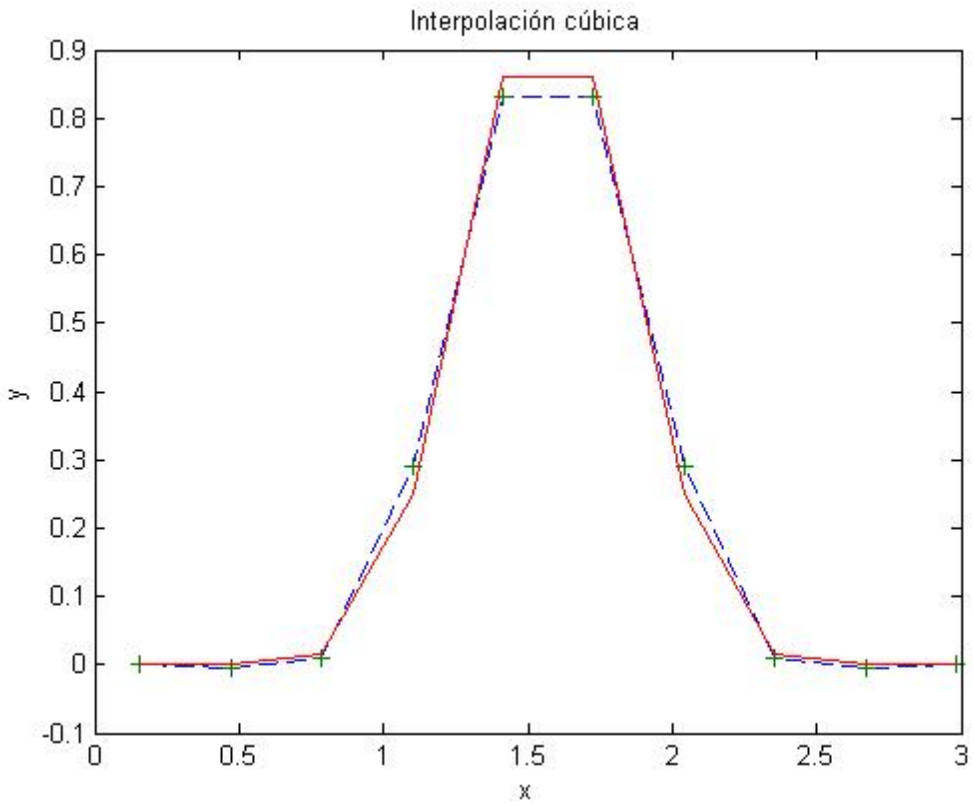
```
    2.9845    0.0000    0.0000

>> E = error(yy', yi');

E =

    0.0246

>> plot(xi,yi,'--',xi,yi,'+',xi,yy)
>> xlabel('x')
>> ylabel('y')
>> title('Interpolación lineal')
```



b) Interpolación cúbica

```
>> x=0:pi/10:pi;
>> y=(sin(x)).^12;
>> xi = pi/20:pi/10:pi-pi/20;
>> yc=interp1(x,y,xi,'v5cubic');  %a partir de matlab 5 ‘cubic’ == ‘v5cubic’
>> interc=[xi' yy' yc']

interc =
```

```
    0.1571    0.0000   -0.0002
    0.4712    0.0001   -0.0040
    0.7854    0.0156    0.0110
    1.0996    0.2504    0.2896
    1.4137    0.8619    0.8314
    1.7279    0.8619    0.8314
    2.0420    0.2504    0.2896
    2.3562    0.0156    0.0110
    2.6704    0.0001   -0.0040
    2.9845    0.0000   -0.0002
>> E = error(yy', yc');

E =

    0.0050

>> plot(xi,yc,'--',xi,yc,'+',xi,yy)
>> xlabel('x')
>> ylabel('y')
>> title('Interpolación cúbica')
```



c) Interpolación splines

Métodos de Computación Científica

Trabajo práctico de programación #4

24-10-2012

```
>> x=0:pi/10:pi;
>> y=(sin(x)).^12;
>> xi = pi/20:pi/10:pi-pi/20;
>> ys=interp1(x,y,xi,'spline');
>> inters=[xi' yy' ys'];
>> inters

inters =

    0.1571    0.0000   -0.0018
    0.4712    0.0001    0.0022
    0.7854    0.0156    0.0077
    1.0996    0.2504    0.2664
    1.4137    0.8619    0.8523
    1.7279    0.8619    0.8523
    2.0420    0.2504    0.2664
    2.3562    0.0156    0.0077
    2.6704    0.0001    0.0022
    2.9845    0.0000   -0.0018

>> E = error(yy', ys')

E =

  8.4216e-004

>> plot(xi,ys,'--',xi,ys,'+',xi,yy)
>> xlabel('x')
>> ylabel('y')
>> title('Interpolación Spline')
```

Métodos de Computación Científica

Trabajo práctico de programación #4

24-10-2012