Trabajo práctico 1

Materia: Métodos en Computación Científica
Alumno: Duarte Daniela LU: 88187
Fecha: 23 de octubre de 2013

Ejercicio 1

Ingresamos la función dada $f(x)=\sin^{12} x$ en MATLAB y obtenemos los valores discretos de x $x_i = \frac{i\pi}{10}$

```
>> 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
```

Se desea encontrar encuentre el valor interpolado de f en para cada

$$x_i = (2i-1)\frac{\pi}{20} \quad i = 1,2,\ldots,10$$

a) Interpolación lineal

Se utilizara el comando interp1 se emplea para interpolar una serie de datos. . El formato de este comando es:

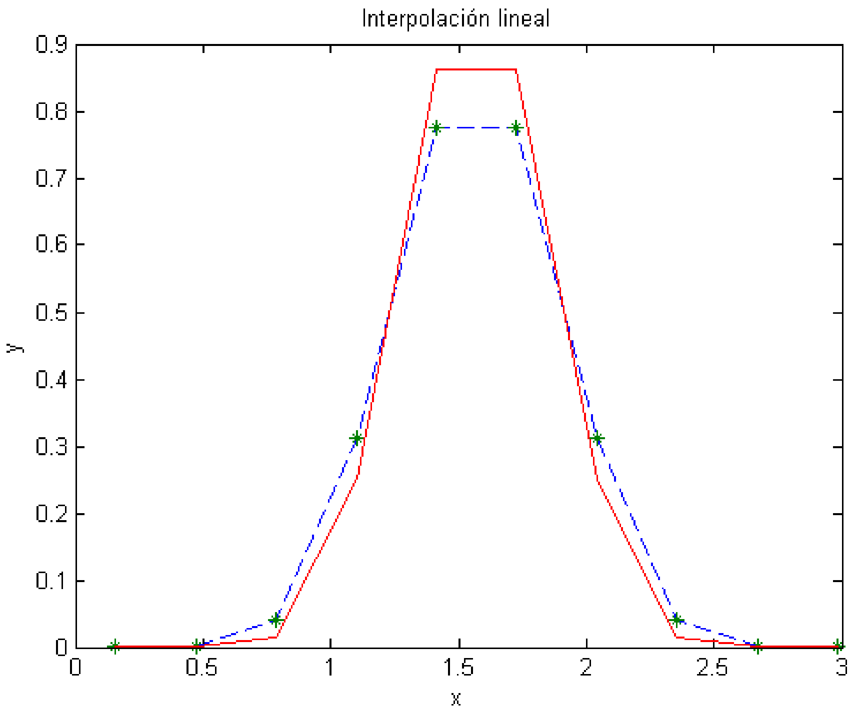
yi = interp1(x, y, xi, método)
Donde:
x : abscisa de los puntos a interpolar, expresada como vector fila.
y : ordenada de los puntos a interpolar, expresada como vector fila.
xi : abscisas para construir la función de interpolación, expresada como vector fila.
Invocamos la función con:

- x: valores $x_i = \frac{i\pi}{10}$ entre 0 y 10.
- y = valores de $f_i=f(x_i)$ 10^4
- xi = valores de $x_i = (2i-1)\frac{\pi}{20} \quad i = 1,2,\ldots,10$.
- inter1: tabla con los xi en la primera columna, los valores del seno en cada xi en la segunda columna, y en la tercera los valores obtenidos de la interpolación.

```
>> 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
```

Graficamos la función para verificar los resultados obtenidos:

```
>> plot(xi,yi,'--',xi,yi,'*',xi,yy)

>> xlabel('x')

>> ylabel('y')

>>title('Interpolación lineal')
```



En rojo los valores reales de al función seno en el cada punto $x_i = (2i-1)\dfrac{\pi}{20} \quad i = 1,2,\ldots,10$

En azul y punteado la función en los valores interpolados por medio de una spline lineal. (Pasa por los puntos marcados con ‘*´).

Se computara el error E asociado con cada interpolación usando la relación:

$$E = \sum_{i=1}^{10} (f_{i,exacto} - f_{i,\text{interpolado}})^2$$

Se implemento la siguiente función en MATLAB:

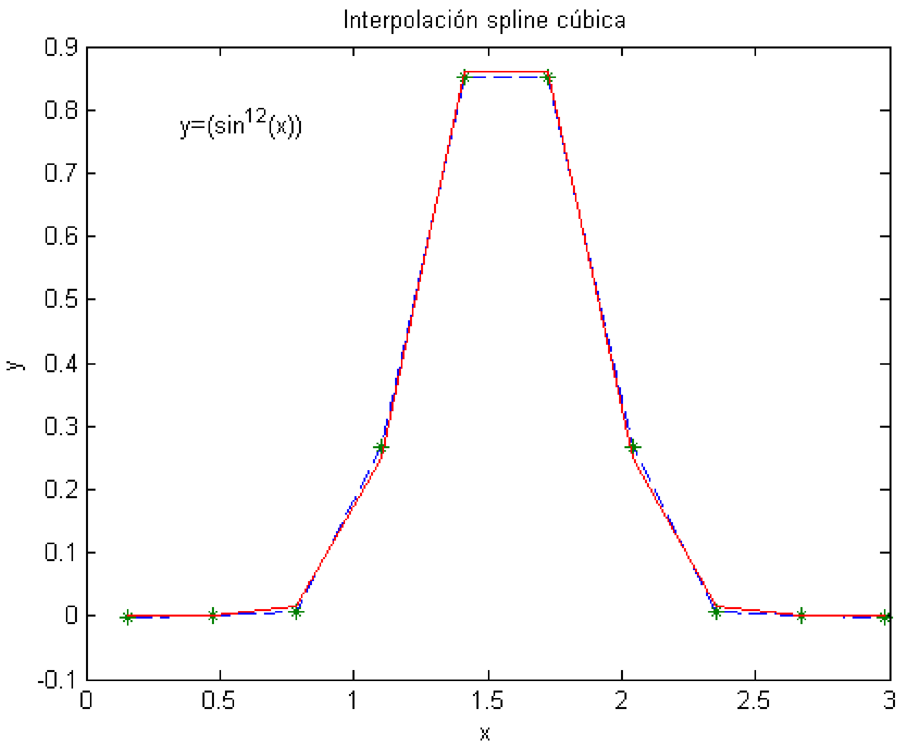
```
function E = error(y, yi)
% error calcula el error asociado a cada interpolación
% y fi exacto
% yi fi interpolado
E = 0;
for k=1:10
E = E + (y(k) - yi(k)).^2;
       End
```

Calculamos el error:

```
>> E = error(yy', yi');
E =
0.0246
```

b) Interpolación cúbica

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

intercub =

    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

Calculamos el error:

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

       8.4216e-04


Graficamos:
>> plot(xi,yc,'--',xi,yc,'*',xi,yy)
>> xlabel('x'); ylabel('y'); title('Interpolación spline cúbica');
>> gtext('y=(sin^1^2(x))');
```



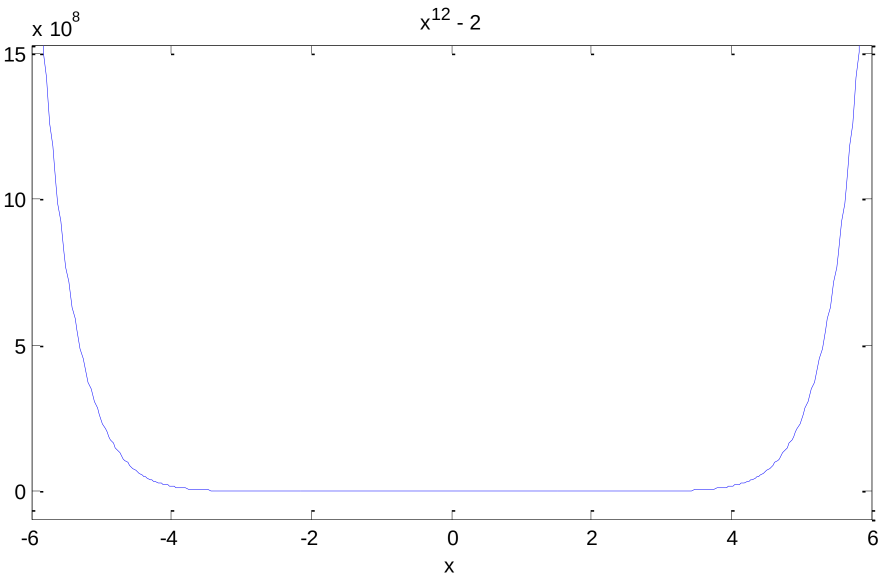
Comparando ambos resultados en los 2 gráficos, se puede concluir que la spline cúbica interpola mucho mejor los datos.

Ejercicio 2

a) $f(x) = x^{12} - 2 = 0$

A partir del método gráfico podemos observar que el cero de la función, sabiendo que la función es par y que está desplazada dos unidades hacia abajo, podemos suponer que la solución se encuentra cerca de cero, más precisamente, hay una raíz a la izquierda de cero y otra a la derecha:

```
>>ezplot('x^12 - 2')
```



Definimos la función de manera anónima:

```
>> f = @(x) x^12 - 2
f =
   @(x)x^12 - 2
```

Y luego evaluamos los ceros de la función, primero a la izquierda (digamos cerca de x=-1):

```
>> fzero(f,-1)
ans =
  -1.0595
```

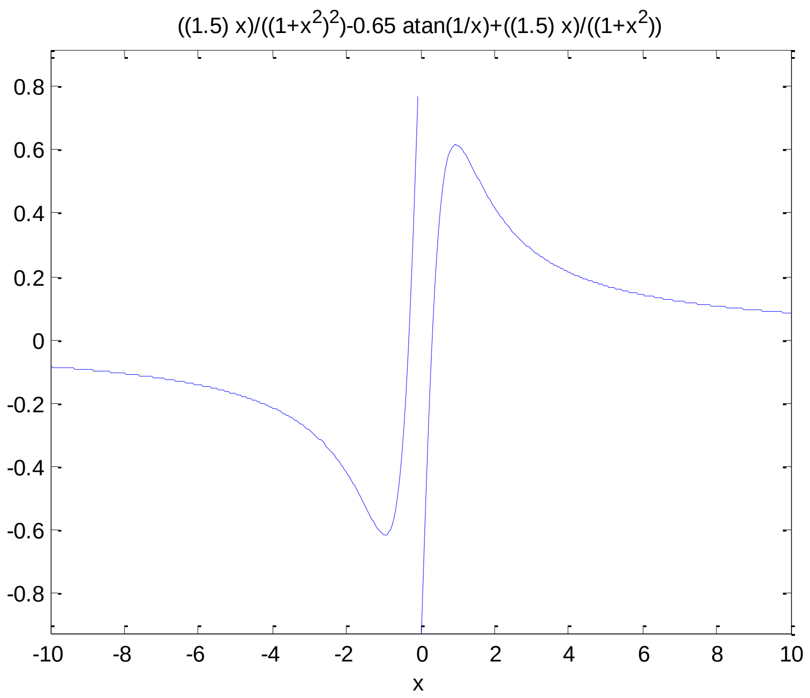
Ahora evaluamos los ceros de la función, primero a la izquierda (digamos cerca de x=1):

```
>> fzero(f,1)
ans =
  1.0595
```

d) $f(x) = \frac{1.5x}{\left(1+x^2\right)^2} - 0.65 \tan^{-1}\frac{1}{x} + \frac{0.65x}{\left(1+x^2\right)} = 0$

Comenzamos empleando el método gráfico para poder visualizar aproximadamente donde se encuentran las funciones:

```
>> syms x real
>> ezplot(f,[-10,10])
```



Del gráfico podemos observar que hacia ±∞ la función tiende a cero pero nunca se hace cero. En un entorno de x=0 la función presenta una asíntota vertical. Y se puede visualizar que la función cambia de signo primero en el intervalo (-2,0) y luego en el (0,2).

Por lo tanto buscamos las soluciones en un entorno de -1.5 y en un entorno de 1,5:

>> f =@(x)((1.5)*x)/((1+x^2)^2)-0.65*atan(1/x)+((1.5)*x)/((1+x^2))

>> x = fzero(f,-1.5)

x = -0.3157

>> x = fzero(f,1.5)

x = 0.3157

Ejercicio 3

Cargamos los datos en MATLAB

```
>> syms d
>> A = d*d;
>> P = 4*d;
>> l = 0.1;
>> k = 240;
>> hinf = 9;
>> lambda =((k*A)/(hinf*P))^(1/2)
```

lambda =

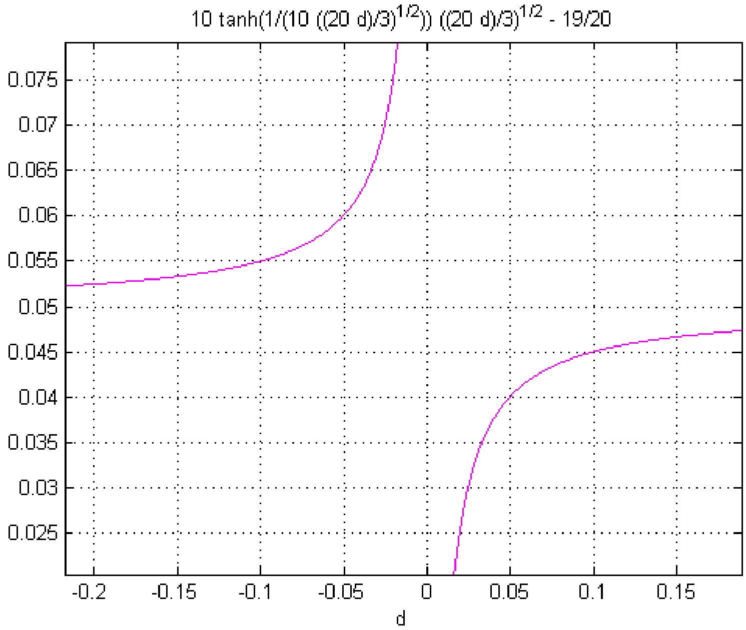
((20*d)/3)^(1/2)

Cargamos la function en MATLAB

```
>> x =((tanh(l/lambda))/(l/lambda)) - 0.95
```

a) Metodo grafico

```
>> g = ezplot(x), grid on;
>> set(g, 'Color', 'm');
```



En el grafico se puede observar que las raíces de la función se encuentran en un entorno muy cercano al 0.Mas precisamente que la raíz se encuentra en el intervalo (-0.05, 0) U (0, 0.05).

b)
biseccion.m

```
function [ r ] = biseccion( f, a, b, N, eps_step, eps_abs )
% Controla que ningun punto extremo es raiz
% y si f(a) y f(b) tienen el mismo signo, lanza una excepcion.
if ( feval(f,a) == 0 )
    r = a;
    return;
elseif ( feval(f,b) == 0 )
    r = b;
    return;
elseif ( feval(f,a) * feval(f,b) > 0 )
    error( 'f(a) y f(b) no tienen signos opuestos' );
end
% Se itera N veces y si no es posible encontrar la raiz
% luego de esas N iteraciones, se lanza una excepcion.
for k = 1:N
    c = (a + b)/2;% Encuentra el punto medio
% Controla si encontramos una raiz o no
% y si debemos seguir iterando:
% [a, c] si f(a) y f(c) tienen signos opuestos, o
% [c, b] si f(c) y f(b) no tienen signos opuestos.
if ( feval(f,c) == 0 )
    r = c;
    return;
elseif ( feval(f,c)*feval(f,a) < 0 )
    b = c;
else
    a = c;
end
% Si |b - a| < eps_step, controlar si
% |f(a)| < |f(b)| y |f(a)| < eps_abs y retornar 'a', o
% |f(b)| < eps_abs y retornar 'b'.
if ( b - a < eps_step )
if ( abs( feval(f,a) ) < abs( feval(f,b) ) && abs( feval(f,a) ) < eps_abs )
    r = a;
    return;
elseif ( abs( feval(f,b) ) < eps_abs )
    r = b;
    return;
end
end
end
    error( 'el metodo no converge' );
end
```

```
   >> format long
>> eps_step = 1e-5;
>> eps_abs = 1e-5;
>> biseccionR
   =biseccion(inline('10*tanh(1/(10*((20*d)/3)^(1/2)))*((20*d)/3)^(1/2) -
   19/20'), 0, 2.0,     20, eps_step, eps_abs)

biseccionR =

   0.009399414062500
```

c) newton_raphson.m

```
function [ r ] = newton_raphson(f, df, xi, emax)
aux = xi;
e = 100; i = 1; h = 0;
fprintf('i           xi           f(xi)            df(xi)       |ep|\n')
fprintf('---------------------------------------\n')
fx = feval(f,xi);
dfx = feval(df,xi);
fprintf('%2d %8d         %10.6f  %10.6f  %10.6f \n', i, xi, fx, dfx, e);
i = i+1;
aux = xi;
h = xi - (fx/dfx);
xi = h;
    while e > emax
        fx = feval(f,xi);
        dfx = feval(df,xi);
        e = abs((xi-aux)/xi);
        fprintf('%2d %8d    %10.6f  %10.6f  %10.6f \n', i, xi, fx, dfx,e);
        aux = xi;
        h = xi - (fx/dfx);
        xi = h;
        i = i+1;
    end
end
```

```
>> dx = diff(x,d)
dx =
   (100*tanh(1/(10*((20*d)/3)^(1/2))))/(3*((20*d)/3)^(1/2))+
    (tanh(1/(10*((20*d)/3)^(1/2)))^2 - 1)/(2*d)

>> newton_raphson(inline('10*tanh(1/(10*((20*d)/3)^(1/2)))*((20*d)/3)^(1/2) -
19/20'),inline('(100*tanh(1/(10*((20*d)/3)^(1/2))))/(3*((20*d)/3)^(1/2)) +
(tanh(1/(10*((20*d)/3)^(1/2)))^2 - 1)/(2*d)'),1.0,eps)

i          xi           f(xi)           df(xi)      |ep|
---------------------------------------
 1         1            0.049500    0.000499  100.000000
 2   -9.811943e+01       0.050005    0.000000    1.010192
 3   -9.629268e+05       0.050000    0.000000    0.999898
 4   -9.272282e+13       0.050000    0.000000    1.000000
 5   -2.880566e+27       0.050000   -0.000000    1.000000
 6   7.426329e+40        0.050000    0.000000    1.000000
 7   -1.088711e+54       0.050000    0.000000    1.000000
 8   -5.737409e+67       0.050000   -0.000000    1.000000
 9   1.103288e+81        0.050000    0.000000    1.000000
10   -1.123710e+94       0.050000    0.000000    1.000000
11   -1.173247e+107      0.050000   -0.000000    1.000000
12   4.398621e+120       0.050000   -0.000000    1.000000
13   1.918606e+133       0.050000   -0.000000    1.000000
14   3.646778e+146       0.050000    0.000000    1.000000
15   -1.996052e+159      0.050000    0.000000    1.000000
16   -1.214532e+172      0.050000   -0.000000    1.000000
17   1.477246e+185       0.050000   -0.000000    1.000000
18   4.523182e+198       0.050000   -0.000000    1.000000
19   1.691762e+211       0.050000    0.000000    1.000000
20   -1.196110e+224      0.050000   -0.000000    1.000000
21   8.947348e+236       0.050000   -0.000000    1.000000
22   3.075193e+249       0.050000   -0.000000    1.000000
23   5.687163e+261       0.050000   -0.000000    1.000000
24   3.805781e+274       0.050000   -0.000000    1.000000
25   1.004269e+287       0.050000    0.000000    1.000000
26   -4.564004e+299      0.050000   -0.000000    1.000000
27       Inf           NaN     0.000000          NaN
```