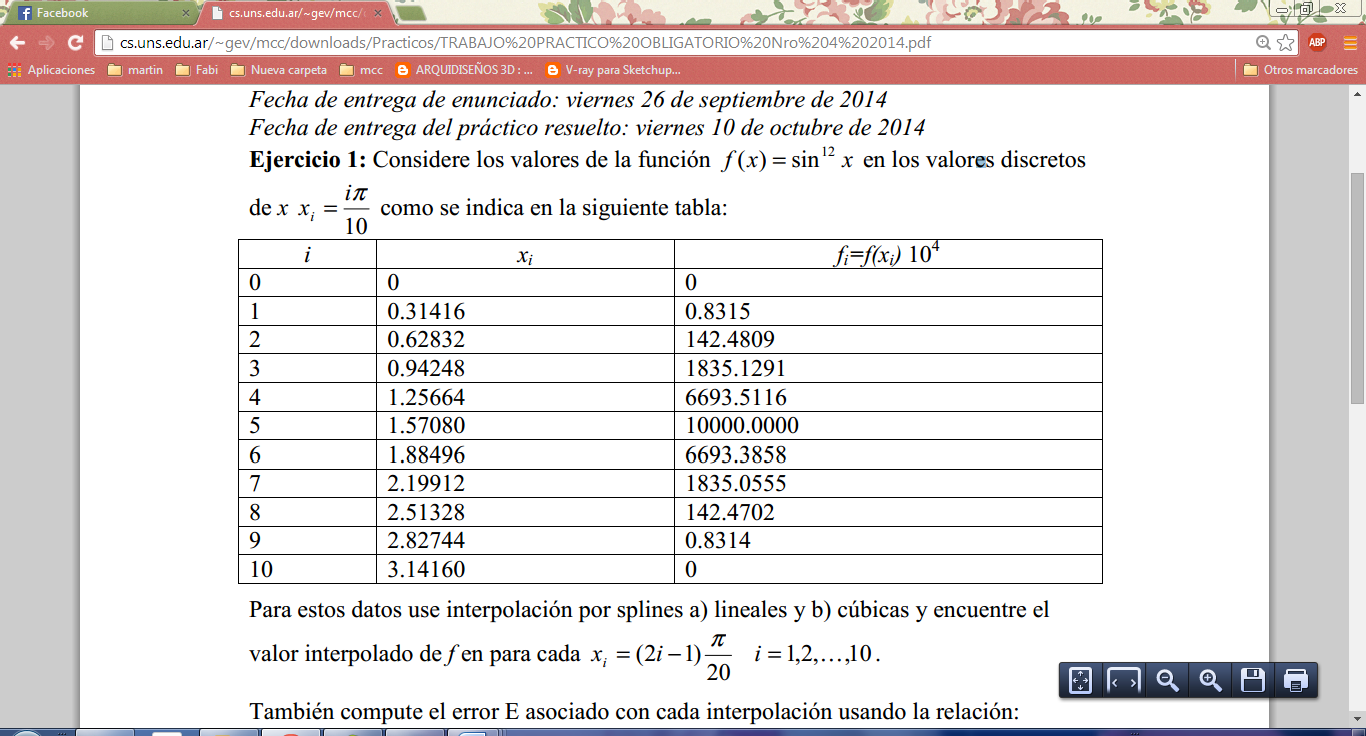
|  |
| --- |
| Métodos de Computación Científica - 2014 - |
| **Trabajo Práctico N° 4: “Aproximación de Funciones”**  **GARAT, Fabiana Yamel - LU 89108 -** |
|  |
|  |



**Ejercicio 1 –** Considere los valores de la función en los valores discretos de x como se indica en la siguiente tabla



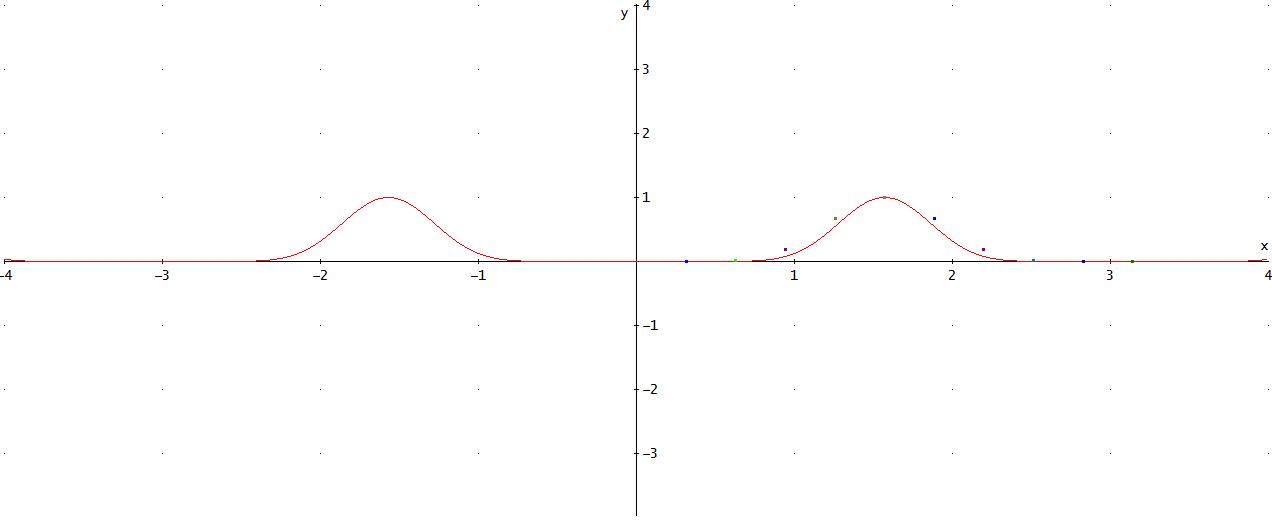
Para estos datos use interpolación por splines

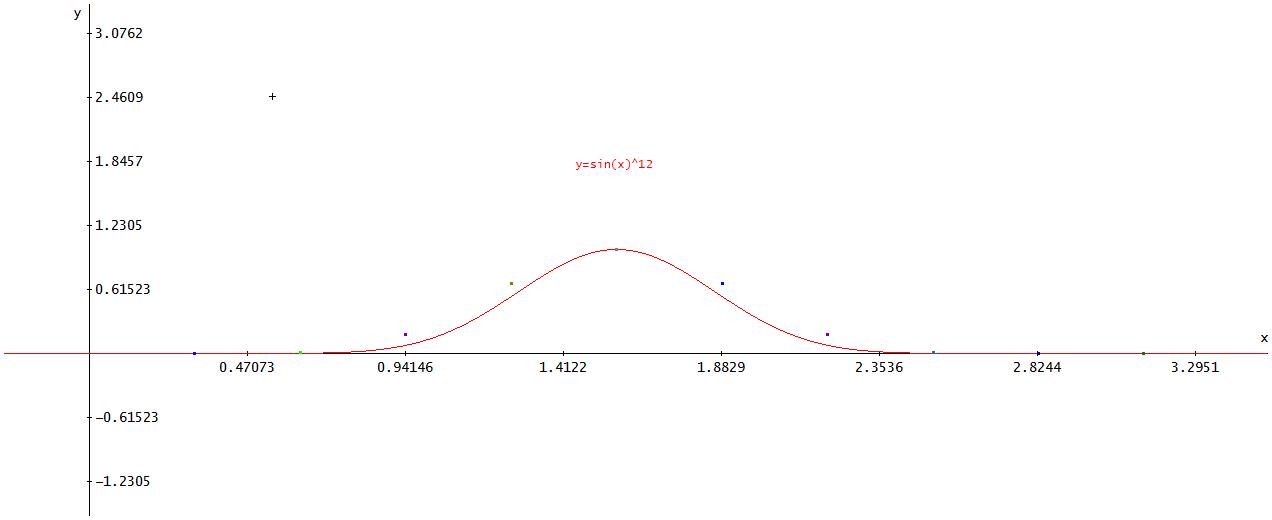
1. Lineales,
2. Cubicas

Y encuentre el calor interpolado de para cada

También compute el error E asociado con cada interpolación usando la relación:

**Solución –** Veamos la curva  acompañadas de los datos enunciado en la tabla dada





Se ingresan a Matlab los vectores:

>> x = 0: (pi/10) : pi;

>> y = sin(x).^12;

>> tabla = [x' y']

tabla =

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

Luego, hallamos el valor interpolado de *f* para cada  en cada uno de los splines solicitados por enunciado.

*Usando interpolación por splines lineales*  
>> 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

* 1. 0.0001 0.0009

2.9845 0.0000 0.0000

Invoque la función interp1 provista por el entorno Matlab, con los argumentos:

* x: valores ** entre 0 y 10,
* y: valores de *fi=f(xi)* 104,
* : valores de ,
* Inter1: tabla con los en la primer columna, los valores del seno en cada en la segunda columna y, por último, en la tercer columna los valores obtenidos de la interpolación.

Se grafica la función a modo de verificación de los datos obtenidos

>> plot(xi,yi,'--',xi,yi,'o',xi,yy)



Referencias de la gráfica:

* En rojo los valores reales de reales de la función seno en el cada punto , y
* En azul y punteado la función en los valores interpolados por medio de una spline lineal. (Pasa por los puntos marcados con círculos).

Calculamos ahora la medida del error cometido en la interpolación usando la relación:



>> suma = 0;

>> for i =1:10

dif = inter1(i,2) - inter1(i,3);

suma = suma + dif^2;

end

>> E = suma

E =

0.0246

*Usando interpolación por splines cúbicas*

Utilizo los datos de intervalos de hallados con anterioridad e interpolamos utilizando splines cubicas

>> yc = interp1(x,y,xi,'cubic');

>> interc = [xi' yy' yc']

interc =

0.1571 0.0000 0.0000

0.4712 0.0001 0.0004

0.7854 0.0156 0.0241

1.0996 0.2504 0.2721

1.4137 0.8619 0.8314

1.7279 0.8619 0.8314

2.0420 0.2504 0.2721

2.3562 0.0156 0.0241

2.6704 0.0001 0.0004

2.9845 0.0000 0.0000

Nuevamente graficamos para verificar los datos obtenidos

>> plot(xi,yc,'--',xi,yc,'o',xi,yy)

>> xlabel('x'); ylabel('y'); title('Interpolación spline cúbica');

>> gtext('y=(sin^1^2(x))');



Teniendo en cuenta que las referencias gráficas son las mismas que las utilizadas en los splines lineales.

Puede observase, comparando ambas gráficas, que la spline cubica interpola los datos de una manera más óptima.

Para comprobar lo dicho calculo la medida de error E, para los datos nuevos:

>> suma = 0;

>> for i =1:10

dif = interc(i,2) - interc(i,3);

suma = suma + dif^2;

end

>> E = suma

E =

0.0029

Puedo concluir, de esta manera, que la medida del error cometido es mucho menor para splines cúbicas que para splines lineales, o lo que es equivalente, las splines cúbicas brindan una mejor aproximación que las splines lineales.