

UNIVERSIDAD DE COSTA RICA

MA-0323 MÉTODOS NUMÉRICOS

PROYECTO

TRAZADOR CÚBICO NATURAL

PROFESOR

FIGUEROA MATA GEOVANNY

ESTUDIANTES

HERNÁNDEZ SANABRIA NOEL FABIÁN B93803

MEZA MENESES MELANY DANIELA B84928

ROBLES PÉREZ DAVID JOSUÉ B96552

I SEMESTRE 2021

Pseudocódigo del Algoritmo

%TRAZADOR CÚBICO NATURAL

function Trazador_Cubico_Natural(x,y)

 %tamaño del número (n) con valor en x.

 n=length(x);

 %tamaño del número (n1) con valor en y.

 □ n1=length(y);

 %ciclo que recibe el valor del número (n)

 %el cual va restando hasta que ingresen a la función

 %para encontrar el valor h, el cual se logra restando

 %un valor mayor de x con su actual valor.

for i=1:n-1

 h(i)=x(i+1)-x(i);

endfor

 %ciclo que recibe el valor del número (n)

 %para establecerlo a la variable a, el cual

 %es la primera variable a tomar para realizar el algoritmo

 %del trazador cúbico natural. en este caso,se toma

 %el valor de an y se iguala al tamaño y correspondido por n1.

for i=1:n

 an(i)=y(i);

endfor

 %fórmula del trazador cúbico natural para encontrar

 %el primer valor de A, el cual recibe el tamaño de x (n)

 %para posteriormente encontrar los valores correspondidos

```

%según el algoritmo. En este caso, se toma por defecto
%al valor A en la posición (1,1) para luego asignarle
%otras posiciones de acuerdo al valor n.
%Luego, entra en un ciclo el cual irá restando el valor n
%hasta que termine y entren sus valores en la fórmula
%establecida por el trazador cúbico natural para el valor A.
A=zeros(length(x),length(x));
A(1,1)=1;
A(n,n)=1;
for i=2:n-1
    A(i,i)=2*(h(i-1)+h(i));
    A(i,i-1)=h(i-1);
    A(i,i+1)=h(i);
endfor
%por consiguiente, aquí asignamos el valor de B,
%el cual toma el mismo n y lo interpreta en la función
%establecida para encontrar los valores necesarios del
%trazador cúbico natural para el valor B.
b1(1)=0;
b1(n)=0;
for i=2:n-1
    b1(i)=(3/(h(i))*(an(i+1)-an(i)))-(3/(h(i-1))*(an(i)-an(i-1))));
endfor
%además, para el penúltimo valor C, se toman por defecto
%la inversa del valor A y se multiplica por la transpuesta del
%valor B, esto para poder obtener un valor c e igualarlo a su valor transpuesto.
%Con estos datos, logramos encontrar tanto valores para B (bn) y C (cn).

```

```

cn=inv(A)*(b1');
cn=cn';
for i=1:n-1
    bn(i)=(an(i+1)-an(i))/h(i)-h(i)*(cn(i+1)+2*cn(i))/3;
endfor

%por otra parte, para encontrar el valor D sencillamente se aplica la
%fórmula del trazador cúbico natural para calcular su valor con los datos
%encontrados previamente.
for i=1:n-1
    dn(i)=(cn(i+1)-cn(i))/(3*h(i));
endfor

%una vez realizado el cálculo de los valores A,B,C y D,
%procedemos a graficar dichos valores según los datos ingresados
%en los demás archivos, los cuales corresponden a las gráficas de
%las funciones en x y y.
sw=0;
for i=1:n-1
    xp=x(i):0.01:x(i+1);
    yi=an(i)+bn(i)*(xp-x(i))+cn(i)*(xp-x(i)).^2+dn(i)*(xp-x(i)).^3;
    hold on;
    set(findobj(gca,'Type','line','Color',[0 0 1]),...
        'Color','black',...
        'LineWidth',3)
    plot(xp,yi);
endfor

%impresión de los datos.
disp('TRAZADOR CÚBICO NATURAL');

```

%impresión de los datos del valor x según

%el algoritmo del trazador cúbico natural.

```
fprintf('x ');
```

```
for i=1: n
```

```
    fprintf('%10.4f',x(i));
```

```
endfor
```

%impresión de los datos del valor y según

%el algoritmo del trazador cúbico natural.

```
fprintf('\n');
```

```
fprintf('y ');
```

```
for i=1: n
```

```
    fprintf('%10.4f',y(i));
```

```
endfor
```

%impresión de los datos de la variable a según

%el algoritmo del trazador cúbico natural.

```
fprintf('\n');
```

```
fprintf('a ');
```

```
for i=1: n
```

```
    fprintf('%10.4f',an(i));
```

```
endfor
```

%impresión de los datos de la variable b según

%el algoritmo del trazador cúbico natural.

```
fprintf('\n');
```

```
fprintf('b ');
```

```
for i=1: n-1
```

```
    fprintf('%10.4f',bn(i));
```

```
endfor
```

```

%impresión de los datos de la variable c según
%el algoritmo del trazador cúbico natural.

fprintf('\n');
fprintf('c ');
for i=1: n
    fprintf('%10.4f',cn(i));
endfor

%impresión de los datos de la variable d según
%el algoritmo del trazador cúbico natural.

fprintf('\n');
fprintf('d ');
for i=1: n-1
    fprintf('%10.4f',dn(i));
endfor

%impresión de los datos de los valores x y y
%además de las variables A,B,C y D según
%el algoritmo del trazador cúbico natural.
%Además, se imprimen los valores resultantes
%comparados a la fórmula, el cual va incrementando
%su exponente en un valor.
%primero toma los valores an (y), luego los valores bn
%para después mostrar los valores x y cn, mostrando
%por último los valores d y x.

fprintf('\n');
for i=1:n-1
    fprintf('S(x)=(%0.4f)',an(i));
    fprintf('+(%0.4f)',bn(i));

```

```

fprintf('x-%0.4f',x(i));
fprintf('+(%0.4f)',cn(i));
fprintf('x-%0.4f)^2',x(i));
fprintf('+(%0.4f)',dn(i));
fprintf('x-%0.4f)^3',x(i));
fprintf('; Para x en');
fprintf('%0.2f',x(i));
fprintf('%0.2f',x(i+1));
fprintf(' \n');
endfor
fprintf('_____ \n');
end

```

Historia del Algoritmo.

La teoría fundacional de los splines fue desarrollada en la década de 1940 por el estadounidense de origen rumano Isaac Jacob Schoenberg.

Isaac Jacob fue un matemático reconocido por haber creado en la década de 1940 los splines, una familia de curvas definida por puntos o condiciones de paso, el cual se ha generalizado posteriormente con la difusión de las aplicaciones informáticas.

Un trazador (spline) es una banda de hule delgada y flexible que se usa para dibujar curvas suaves a través de un conjunto de puntos. Los trazadores cúbicos (cubic splines) naturales se utilizan para crear una función que interpola un conjunto de puntos de datos. Esta función consiste en una unión de polinomios cúbicos, uno para cada intervalo, y está construido para ser una función con derivada primera y segunda continuas. El 'spline' cúbico natural también tiene su segunda derivada igual a cero en la coordenada x del primer punto y el último punto de la tabla de datos.

Funciones

```
]function Trazador_Cubico_Natural(x,y)
n=length(x);
n1=length(y);
```

Función principal que recibe el nombre de Trazador Cúbico Natural, en esta función se localiza el algoritmo del trazador cúbico natural. Esta función recibe dos parámetros, los cuales se les aplica el método length a cada uno de los parámetros y se guarda el resultado en una variable para poder usar el tamaño de los parámetros en los ciclos que se necesitan para llevar a cabo el algoritmo correctamente.

```
for i=1:n-1
    h(i)=x(i+1)-x(i);
endfor
```

Se utiliza este ciclo for que recibe el valor del número n y avanza en decremento para restarlo hasta que se ingrese a la función con el objetivo de encontrar el valor de h. Esto se logra restando un valor mayor de x con su actual valor.

```
]for i=1:n
    an(i)=y(i);
endfor
```

En este ciclo se recibe el valor del número n, el cual se va restando hasta que se pueda ingresar a la función, para así poder encontrar el valor de h, el cual se lleva a cabo restando un valor mayor a x con su actual valor

```
A=zeros(length(x),length(x));
A(1,1)=1;
A(n,n)=1;
for i=2:n-1
    A(i,i)=2*(h(i-1)+h(i));
    A(i,i-1)=h(i-1);
    A(i,i+1)=h(i);
endfor
```

En este segmento del pseudocódigo se ingresa la fórmula del trazador cúbico natural con el objetivo de encontrar el primer valor de A, el cual recibe el tamaño de x (n) para encontrar los valores correspondientes según el algoritmo. Para esto se toma por defecto el valor A en

la posición (1,1), cabe recalcar que estas posiciones irán cambiando con respecto al valor de n. Se utiliza un ciclo for con el objetivo de decrementar la n hasta que termine e ingresen sus valores en la fórmula para el valor de A.

```
b1(1)=0;
b1(n)=0;
for i=2:n-1
    b1(i)=(3/(h(i))*(an(i+1)-an(i)))-(3/(h(i-1))*(an(i)-an(i-1))));
endfor
```

En esta parte, se asigna el valor de B, este valor toma el mismo n y lo interpreta en la función establecida para así poder encontrar los valores necesarios del algoritmo del trazador cúbico natural para el valor B.

```
cn=inv(A)*(b1');
cn=cn';
for i=1:n-1
    bn(i)=(an(i+1)-an(i))/h(i)-h(i)*(cn(i+1)+2*cn(i))/3;
endfor
```

En el valor de Cn se toman por defecto la inversa del valor de A y se multiplica por la transpuesta del valor de B, esto se hace para poder obtener un valor C e igualarlo a su valor transpuesto, ya con estos valores definidos, se logra encontrar tanto valores para B(bn) y C(cn)

```
]for i=1:n-1
    dn(i)=(cn(i+1)-cn(i))/(3*h(i));
endfor
```

En este ciclo se pretende encontrar el valor D, por lo que se aplica la fórmula del trazador cúbico natural para calcular su valor junto con los datos encontrados previamente.

```
sw=0;
for i=1:n-1
    xp=x(i):0.01:x(i+1);
    yi=an(i)+bn(i)*(xp-x(i))+cn(i)*(xp-x(i)).^2+dn(i)*(xp-x(i)).^3;
    hold on;
    set(findobj(gca,'Type','line','Color',[0 0 1]),...
        'Color','black',...
        'LineWidth',3)
    plot(xp,yi);
endfor
```

Una vez realizado ya el cálculo de los valores de A,B,C y D, se procede a graficar los valores según los parámetros recibidos en la función, lo cual corresponde a las gráficas de las funciones en X,Y.

```
disp('TRAZADOR CÚBICO NATURAL');
```

Con esta sentencia se procede a la impresión de los datos.

```
fprintf('x ');  
for i=1: n  
    fprintf('%10.4f',x(i));  
endfor
```

Se imprimen los datos del valor de x según el algoritmo del trazador cúbico natural.

```
fprintf('\n');  
fprintf('y ');  
for i=1: n  
    fprintf('%10.4f',y(i));  
endfor  
-----
```

Impresión de los datos del valor y según el algoritmo del trazador cúbico natural.

```
fprintf('\n');  
fprintf('a ');  
for i=1: n  
    fprintf('%10.4f',an(i));  
endfor
```

Impresión de los datos de la variable A, según el algoritmo aplicado.

```
fprintf('\n');  
fprintf('b ');  
for i=1: n-1  
    fprintf('%10.4f',bn(i));  
endfor
```

Impresión de los datos de la variable B, según el algoritmo del trazador cúbico natural

```
fprintf('\n');  
fprintf('c ');  
for i=1: n  
    fprintf('%10.4f',cn(i));  
endfor
```

Impresión de los datos de la variable C, según el algoritmo del trazador cúbico natural

```
fprintf('\n');
fprintf('d ');
for i=1: n-1
    fprintf('%10.4f', dn(i));
endfor
```

Impresión de los datos de la variable D, según el algoritmo del trazador cúbico natural.

```
fprintf('\n');
for i=1:n-1
    fprintf('S(x)=(%0.4f)', an(i));
    fprintf('+(%0.4f)', bn(i));
    fprintf('(x-%0.4f)', x(i));
    fprintf('+(%0.4f)', cn(i));
    fprintf('(x-%0.4f)^2', x(i));
    fprintf('+(%0.4f)', dn(i));
    fprintf('(x-%0.4f)^3', x(i));
    fprintf('; Para x en');
    fprintf(' (%0.2f', x(i));
    fprintf(' (%0.2f', x(i+1));
    fprintf(' \n');
endfor
fprintf('_____ \n');
end
```

Se imprimen los datos de los valores x, y, además de las variables A, B, C y D, según el algoritmo aplicado. Se imprimen los valores resultantes comparados en la fórmula, los cuales van incrementando su exponente en un valor. Para esto, se toma primeramente los valores an (y), luego se toman los valores x y cn, por último se muestran los valores d y x.

```
function Grafica_Curvas()
x=[1 2 5 6 7 8 10 13 17];
y=[3 3.7 3.9 4.2 5.7 6.6 7.1 6.7 4.5];
Trazador_Cubico_Natural(x,y);
```

Función llamada Gráfica Curvas, esta función se encarga de crear la gráfica con los datos de x,y y luego se llama la función principal, la gráfica es de cola y espalda(dos curvas).

```
x=[17 20 23 24 25 27 27.7];
y=[4.5 7 6.1 5.6 5.8 5.2 4.1];
Trazador_Cubico_Natural(x,y);
end
```

Gráfica cabeza(una curva).

```
function Grafica_Completa()
```

Función llamada Gráfica Completa, esta función se encarga de mostrar la gráfica utilizando los datos x, y y hace un llamado a la función principal, tiene el propósito de mostrar todas las gráficas establecidas para formar la figura final, la cual es un perro acostado boca abajo. A continuación se mostrarán las sentencias de todas las gráficas establecidas.

```
%gráfica cola y espalda
x=[1  2  5  6  7  8  10  13  17];
y=[3  3.7  3.9  4.2  5.7  6.6  7.1  6.7  4.5];
Trazador_Cubico_Natural(x,y);

%gráfica pierna tracera interna
x=[6  7  8  10  11  12  12.5  12.55];
y=[4.2  5  5.5  6  6  5.5  5  4.92];
Trazador_Cubico_Natural(x,y);

%gráfica pierna tracera adyacente codo
x=[11.2  11.7  12  12.2  12.4  12.55];
y=[3  3.5  3.7  3.9  4.3  4.92];
Trazador_Cubico_Natural(x,y);

%gráfica pierna tracera pata
x=[11.2  11.25  11.5  11.7  12  13  14  14.4  14.5];
y=[3  2.9  2.8  2.8  2.8  2.75  2.68  2  1.4];
Trazador_Cubico_Natural(x,y);

%gráfica cabeza
x=[17  20  23  24  25  27  27.7];
y=[4.5  7  6.1  5.6  5.8  5.2  4.1];
Trazador_Cubico_Natural(x,y);
```

```
%gráfica nariz
x=[27.7  27.9  28.55  28.8];
y=[4.1  4  3.5  2.5];
Trazador_Cubico_Natural(x,y);

%gráfica bigote 1
x=[27  29];
y=[4  6];
Trazador_Cubico_Natural(x,y);

%gráfica bigote 2
x=[27.2  29];
y=[3.8  5];
Trazador_Cubico_Natural(x,y);

%gráfica bigote 3
x=[27.2  28];
y=[3.5  3.45];
Trazador_Cubico_Natural(x,y);

%gráfica bigote 4
x=[27.1  28.5];
y=[3  2.7];
```

```

%gráfica ojeras 3
x=[22 22.7 24.1];
y=[3.1 2.9 3.7];
Trazador_Cubico_Natural(x,y);

%gráfica cola base
x=[1 2 5 6 7 8 8.5 8.6];
y=[3 2.9 2.8 2.8 2.8 2.9 3.1 3];
Trazador_Cubico_Natural(x,y);

%gráfica codo pata tracera
x=[8 8.1 8.2 8.6];
y=[1.8 2.3 2.5 3];
Trazador_Cubico_Natural(x,y);

%gráfica base pie y dedo 1
x=[8 8.2 9 10 10.8 11 12 12.6 12.7 12.8];
y=[1.8 1.6 1.3 1.3 1.5 1.5 1.2 1.2 1.2 1.3];
Trazador_Cubico_Natural(x,y);

%gráfica dedo completo 1
x=[12.4 12.5 12.6 12.64 12.8];
y=[2.1 2 1.9 1.8 1.3];
Trazador_Cubico_Natural(x,y);
-----\,1,

```

```

%gráfica dedo completo 2
x=[13.1 13.5 13.63 13.7];
y=[2.1 2 1.7 1.35];
Trazador_Cubico_Natural(x,y);

%gráfica dedo 3
x=[13.7 13.8 14 14.2 14.5];
y=[1.35 1.26 1.19 1.19 1.4];
Trazador_Cubico_Natural(x,y);

%gráfica estomago y pata delantera hasta inicio de la oreja
x=[14.5 15.4 17 17.5];
y=[1.4 1.3 1.2 1.2];
Trazador_Cubico_Natural(x,y);

%gráfica oreja parte 1
x=[17 17.02 17.1 17.5];
y=[4.5 4 3 1.2];
Trazador_Cubico_Natural(x,y);

%gráfica oreja parte 2
x=[18.7 20.5 21];
y=[1.2 3.5 5];
Trazador_Cubico_Natural(x,y);

%gráfica oreja parte 3
x=[20.5 20.7 21];
y=[6.5 6 5];
Trazador_Cubico_Natural(x,y);

%gráfica pata delantera e inicio de la oreja interna
x=[15.4 15.7 17];
y=[1.3 1.9 2.6];
Trazador_Cubico_Natural(x,y);

%gráfica pata delantera interno
x=[20 22 23 24 25];
y=[2.2 2.2 2.3 2.5 2.2];
Trazador_Cubico_Natural(x,y);

%gráfica parte oreja
x=[17.5 17.6 18 18.1 18.7];
y=[1.2 1.1 1 1 1.2];
Trazador_Cubico_Natural(x,y);

%gráfica pata delantera hasta inicio de dedo 1
x=[18.7 21 22 23 23.7 23.8 23.85];
y=[1.2 1 0.9 0.8 0.9 0.99 1.15];
Trazador_Cubico_Natural(x,y);

```

```

%gráfica pata delantera dedo 2
x=[23.8    24    24.1    24.3    24.4    24.46];
y=[0.98    0.9    0.89    0.92    0.99    1.2];
Trazador_Cubico_Natural(x,y);

%gráfica pata delantera dedo 3
x=[24.4    24.7    24.8    25    25.1    25.3];
y=[0.99    0.95    0.96    0.98    1    1.3];
Trazador_Cubico_Natural(x,y);

%gráfica pata delantera dedo 3 prolongación
x=[25    25.2    25.3];
y=[2.2    1.8    1.3];
Trazador_Cubico_Natural(x,y);

%gráfica inicio osico
x=[25.1    25.3    26    27    28];
y=[1    0.95    1    1.4    2.4];
Trazador_Cubico_Natural(x,y);

%gráfica inicio nariz base
x=[28    28.2    28.3    28.75    28.8];
y=[2.4    2.3    2.3    2.45    2.5];
Trazador_Cubico_Natural(x,y);

%gráfica dedo 1 pata delantera adyacentes a la nariz
x=[28.8    29    29.2    29.3    29.5];
y=[2.5    2.4    2.45    2.48    2.6];
Trazador_Cubico_Natural(x,y);

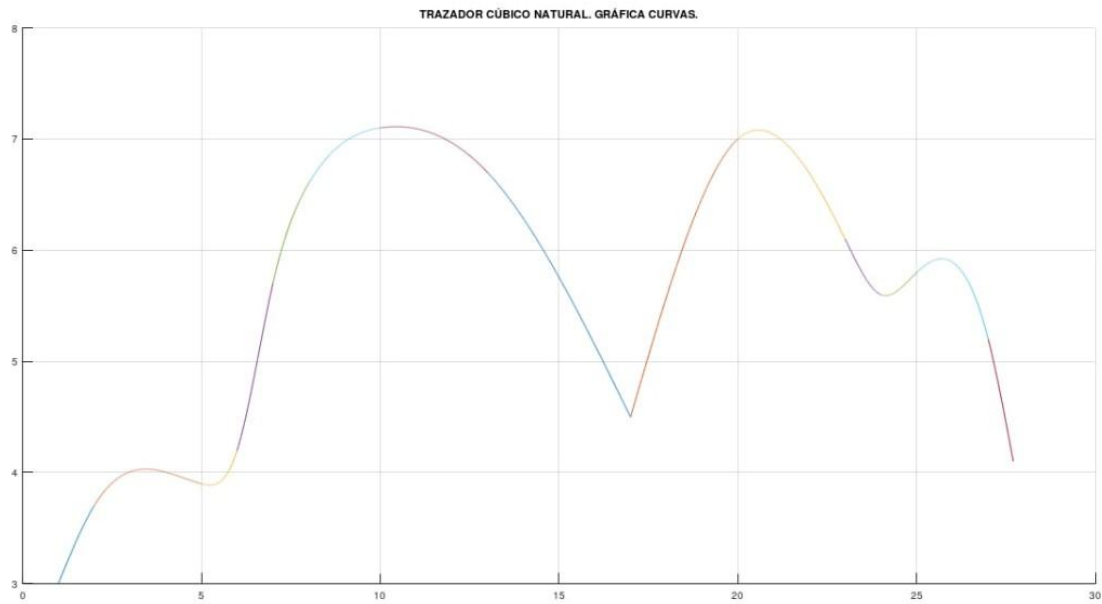
%gráfica dedo 2 pata delantera adyacentes a la nariz
x=[29.5    29.8    29.9    29.98    30];
y=[2.6    2.4    2.5    2.8    3];
Trazador_Cubico_Natural(x,y);
end

```

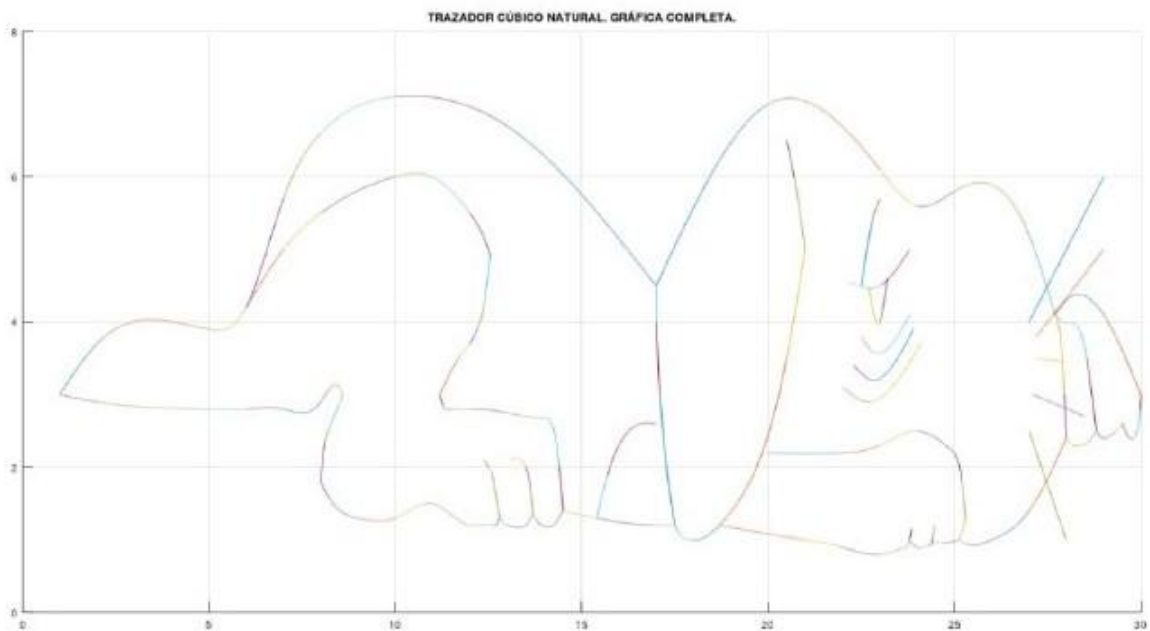
EJECUCIÓN DEL ALGORITMO

Una vez ya realizadas las funciones anteriores, se procede a la ejecución del algoritmo. Para poder llevar a cabo el algoritmo se debe llenar los dos parámetros que recibe, en este caso se llena los parámetros con los valores en la función Gráfica Curvas, con el ejemplo de la página 156 del libro Análisis Numérico, una vez llenado los datos correctamente, se llama la función principal se le pasa los parámetros creados y se ejecuta.

Con la función Gráfica Curvas se tendría un resultado igual a este.



Con la función Gráfica completa, se graficaron todas las curvas. Para ello se deben ir colocando cada uno de los puntos con sus respectivos valores, llamar al método principal hasta terminar. Una vez ingresado todos los parámetros necesarios se procede a ejecutar la función y se tendrá un resultado como el siguiente:



Bibliografía

G, A. (2014, June 24). Trazadores Cúbicos. Proyecto Métodos Numéricos MA-0320.

<https://arturoguillen90.wordpress.com/interpolacion/trazadores-cubicos/>

Burden, R., & Faires, J. Análisis numérico [PDF] (7th ed.). Math Learning.