INTERFAZ GRÁFICA PARA LA INTERPOLACIÓN DE DATOS A TRAVÉS DE SPLINES CÚBICOS

GUI for Data Interpolation by Cubic Spline

RESUMEN
En este documento se presentan los resultados obtenidos al desarrollar una interfaz gráfica en la utilidad *GUIDE (Graphical User Interface Development Environment)* de *Matlab©*, la cual permite llevar a cabo la interpolación de los datos ingresados por el usuario a través de *Splines Cúbicos*. El programa también posibilita la realización de la interpolación lineal, con el objetivo de comparar los resultados obtenidos con las funciones cúbicas arrojadas al realizar la reconstrucción con polinomios de tercer orden, cuyos coeficientes pueden ser recuperados en un archivo de texto una vez es ejecutado el programa.

PALABRAS CLAVES: Interfaz Gráfica, interpolación, reconstrucción de funciones, splines.

ABSTRACT
This paper presents the results obtained by developing a GUI utility in the GUIDE (Graphical User Interface Development Environment) in Matlab©, which allows interpolation of the data entered by the user via Cubic Splines. The program also enables the realization of the linear interpolation, in order to compare the results with cubic functions thrown when performing the interpolation with polynomials of third order, whose coefficients can be retrieved in a text file after running the program.

KEYWORDS: *GUI, interpolation, reconstruction of functions, splines.*

EDWIN ANDRÉS QUINTERO S.
Ingeniero Electrónico
Candidato a Magíster en Instrumentación Física
Grupo de Investigación en Astronomía Alfa Orión
Profesor Auxiliar
Universidad Tecnológica de Pereira
equintero@utp.edu.co

WILLIAM ARDILA URUEÑA
Licenciado en Física
MSc. Física
Profesor Titular
Universidad Tecnológica de Pereira
williamar@utp.edu.co

HUGO ARMANDO GALLEGO
Licenciado en Física
MSc. Física
Profesor Asistente
Universidad Tecnológica de Pereira
ugo@utp.edu.co

1. INTRODUCCIÓN

En el desarrollo de una práctica de laboratorio de física universitaria, generalmente los estudiantes realizan lecturas de los diferentes valores que toma cierta variable física en particular, que podría ser, entre otras, distancia, temperatura, presión, tiempo, etc. Naturalmente, estos datos recopilados corresponden a un conjunto finito de muestras que ofrecen una idea acerca de la tendencia existente en el comportamiento del parámetro medido. Dado que el experimentador no puede tomar una cantidad infinita de valores (de tal manera que la representación corresponda a una curva continua), la gráfica final construida con todas las muestras recopiladas corresponderá a una función discreta, es decir, una función que toma algún valor de un conjunto finito de posibles valores dentro de un intervalo de tiempo dado. Debido a que todas las variables físicas presentan un comportamiento continuo, es posible concluir que la gráfica construida al unir con rectas todos los puntos tomados, corresponderá a una visión un poco alejada de la realidad. Si en vez de trazar rectas para unir los puntos recolectados, se plantean funciones de orden superior que unan cada pareja de puntos sin afectar la continuidad de la curva final, entonces se estará realizando una interpolación polinomial a trozos [1], la cual entregará como resultado final una curva mucho más aproximada al comportamiento real de la variable medida. Así, la interpolación por medio de *Splines Cúbicos* permite crear funciones de orden superior que aproximan los datos tomados a una curva continua que conserva la suavidad inherente a toda variable física. Además de la utilidad que presentan los *Splines* en la interpolación de datos, también se encuentran múltiples aplicaciones de este tipo de funciones en diferentes áreas de la ingeniería, especialmente en el campo del procesamiento digital de imágenes, pues son ideales en la reconstrucción, alisado, filtrado, ampliación y reducción de fotografías [2]; todo esto gracias, entre otras cosas, a que las funciones reconstruidas a partir de estos polinomios conservan su continuidad, lo cual permite reducir los efectos de los valores extremos de una manera muy eficaz [3].

El objetivo fundamental del programa desarrollado consiste en brindar una herramienta informática a los estudiantes de los cursos de laboratorio de física para ingenierías, la cual permitirá la construcción de las gráficas solicitadas en las guías, a partir de la interpolación con *Splines Cúbicos*, facilitando así la realización del análisis de cada gráfica necesario para la comprensión del fenómeno estudiado. Además, gracias a que el programa permite obtener los coeficientes de los polinomios utilizados para la interpolación, el estudiante también podrá familiarizarse con esta importante aplicación de los métodos numéricos.

Fecha de Recepción: Enero 26 de 2010
Fecha de Aceptación: Marzo 25 de 2010

2. INTERPOLACIÓN POLINOMIAL

Cuando se desea interpolar un conjunto de datos por medio de polinomios, es posible determinar la función que identifica cada pareja de puntos teniendo en cuenta únicamente la información suministrada por el par de valores. A este tipo de interpolación polinomial se le conoce como *local*. Si por el contrario, en el momento de obtener la ecuación que unirá un par de puntos, no solamente se tiene en cuenta la información suministrada por la pareja, sino que además se toma todo el conjunto de muestras, entonces se estará realizando una interpolación polinomial *global* [4].

2.1 INTERPOLACIÓN POR SPLINES CÚBICOS

Un ejemplo claro de la aproximación *global* de funciones descrita anteriormente consiste en la interpolación por *Splines Cúbicos*. Supóngase que se desea interpolar cierta función $f(t_i)$ definida por un conjunto finito de valores y_i (ver figura 1), mediante una serie de polinomios $p_i(t)$. Los datos $t_0,t_1,\ldots,t_{i-1},t_i,t_{i+1},\ldots,t_{n-1},t_n$; no necesariamente tienen que estar igualmente espaciados, pero si deben ser ordenados [5], es decir, se debe cumplir que:

$$t_0 < t_1 < .... < t_{i-1} < t_i < t_{i+1} < ... < t_{n-1} < t_n \qquad (1)$$

El objetivo es entonces encontrar los polinomios $p_i(t)$ que interpolen la función $y_i = f(t_i)$ en los intervalos $[t_i,t_{i+1}]$, con $i=1,2,3,\ldots,n-1$.

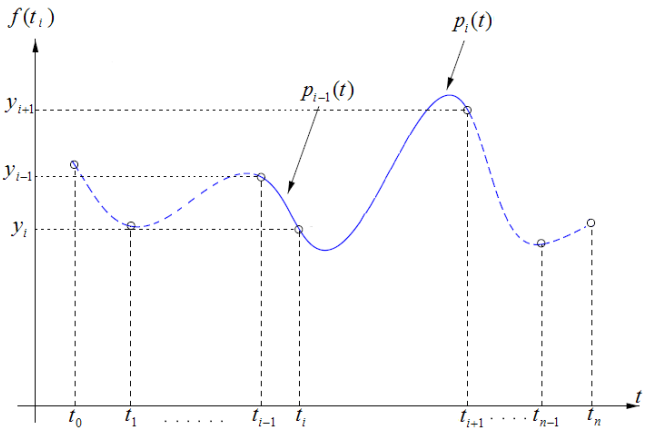

Figura 1. Interpolación de una función $f(t_i)$ mediante polinomios individuales para cada par de datos.

Para que los polinomios $p_i(t)$ interpolen la función se debe cumplir que:

$$p_i(t_i) = y_i \qquad (2)$$

$$p_i(t_{i+1}) = y_{i+1} \qquad (3)$$

Ahora, para que la función interpolada no contenga cambios bruscos, las derivadas en los extremos de los polinomios que aproximan cada tramo deben coincidir, al igual que su segunda derivada; esto con el fin de mantener la continuidad de la curva final obtenida:

$$p'_{i-1}(t_i) = p'_i(t_i) \qquad (4)$$

$$p''_{i-1}(t_i) = p''_i(t_i) \qquad (5)$$

Si $p_i(t)$ es un polinomio de grado tres, su segunda derivada corresponderá a una línea recta, tal como se observa en la figura 2.

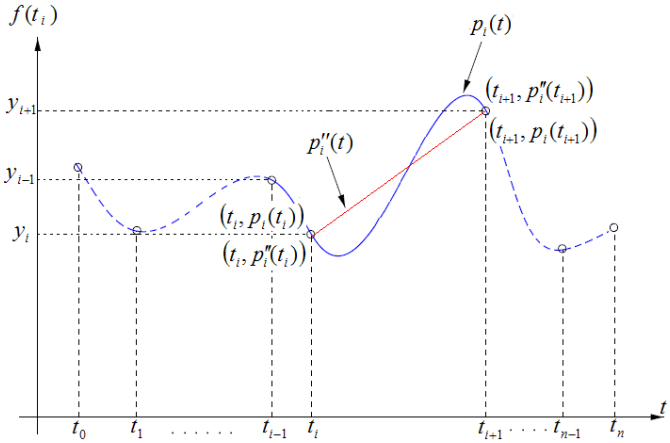

Figura 2. Polinomios de primer grado $p''_i(t)$, y tercer grado $p_i(t)$ que interpolan la función $y_i = f(t_i)$ en el intervalo definido por $[t_i,t_{i+1}]$.

Bajo estas condiciones, primero se obtienen polinomios de grado uno que interpolen cada pareja de datos, para lo cual se aplica la ecuación de la recta:

$$y - y_0 = m(x - x_0) \qquad (6)$$

$$p''_i(t) - p''_i(t_i) = \frac{p''_i(t_{i+1}) - p''_i(t_i)}{t_{i+1} - t_i}(t - t_i) \qquad (7)$$

$$p''_i(t) = p''_i(t_{1+1})\frac{t - t_i}{t_{i+1} - t_i} - p''_i(t_i)\frac{t - t_i}{t_{i+1} - t_i} + p''_i(t_i) \qquad (8)$$

$$p''_i(t) = p''_i(t_i)\left[1 - \frac{t - t_i}{t_{i+1} - t_i}\right] + p''_i(t_{i+1})\frac{t - t_i}{t_{i+1} - t_i} \qquad (9)$$

$$p''_i(t) = p''_i(t_i)\left[\frac{t_{i+1} - t_i - (t - t_i)}{t_{i+1} - t_i}\right] + p''_i(t_{i+1})\frac{t - t_i}{t_{i+1} - t_i}$$

$$p''_i(t) = p''_i(t_i)\frac{t - t_{i+1}}{t_i - t_{i+1}} + p''_i(t_{i+1})\frac{t - t_i}{t_{i+1} - t_i} \qquad (10)$$

$$p''_i(t) = p''_i(t_i)\frac{t_{i+1} - t}{t_{i+1} - t_i} + p''_i(t_{i+1})\frac{t - t_i}{t_{i+1} - t_i} \qquad (11)$$

Sea: $t_{i+1} - t_i = h_i$ (12)

Entonces (11) se puede expresar como:

$$p_i''(t) = p_i''(t_i)\frac{t_{i+1}-t}{h_i} + p_i''(t_{i+1})\frac{t-t_i}{h_i} \quad (13)$$

Por otra parte, si:

$$p_i''(t_i) = \sigma_i \quad (14)$$
$$p_i''(t_{i+1}) = \sigma_{i+1} \quad (15)$$

La ecuación 12 queda:

$$p_i''(t) = \frac{\sigma_i}{h_i}(t_{i+1}-t) + \frac{\sigma_{i+1}}{h_i}(t-t_i) \quad (16)$$

La expresión (16) representa la ecuación de la recta que interpola la pareja de puntos pertenecientes al intervalo $[t_i,t_{i+1}]$. Así, para obtener la ecuación del *Spline Cúbico* se debe integrar dos veces la ecuación (16):

$$p_i(t) = \frac{\sigma_i}{h_i}\frac{(t_{i+1}-t)^3}{6} + \frac{\sigma_{i+1}}{h_i}\frac{(t-t_i)^3}{6} + C_i t + D_i \quad (17)$$

Donde los términos C_i y D_i representan las constantes originadas durante el proceso de integración, las cuales se pueden expresar de la siguiente forma:

$$C_i t + D_i = A_i(t-t_i) + B_i(t_{i+1}-t) \quad (18)$$

Así, la ecuación (17), se puede escribir como:

$$p_i(t) = \frac{\sigma_i}{h_i}\frac{(t_{i+1}-t)^3}{6} + \frac{\sigma_{i+1}}{h_i}\frac{(t-t_i)^3}{6} + A_i(t-t_i) + B_i(t_{i+1}-t) \quad (19)$$

Para determinar los valores de las constantes A_i y B_i, aplicamos las condiciones (2) y (3) a la ecuación (19):

$$p_i(t_i) = y_i$$

$$y_i = \frac{\sigma_i}{h_i}\frac{(t_{i+1}-t_i)^3}{6} + \frac{\sigma_{i+1}}{h_i}\frac{(t_i-t_i)^3}{6} + A_i(t_i-t_i) + B_i(t_{i+1}-t_i)$$

$$y_i = \frac{\sigma_i h_i^2}{6} + B_i h_i$$

$$B_i = \frac{y_i}{h_i} - \frac{\sigma_i h_i}{6} \quad (20)$$

$$p_i(t_{i+1}) = y_{i+1}$$

$$y_{i+1} = \frac{\sigma_i}{h_i}\frac{(t_{i+1}-t_{i+1})^3}{6} + \frac{\sigma_{i+1}}{h_i}\frac{(t_{i+1}-t_i)^3}{6} + A_i(t_{i+1}-t_i) + B_i(t_{i+1}-t_{i+1})$$

$$y_{i+1} = \frac{\sigma_{i+1}h_i^2}{6} + A_i h_i$$

$$A_i = \frac{y_{i+1}}{h_i} - \frac{\sigma_{i+1}h_i}{6} \quad (21)$$

Reemplazando A_i y B_i en (19):

$$p_i(t) = \frac{\sigma_i}{6}\left[\frac{(t_{i+1}-t)^3}{h_i} - h_i(t_{i+1}-t)\right] + \frac{\sigma_{i+1}}{6}\left[\frac{(t-t_i)^3}{h_i} - h_i(t-t_i)\right] + y_i\frac{(t_{i+1}-t)}{h_i} + y_{i+1}\frac{(t-t_i)}{h_i} \quad (22)$$

Solo resta calcular los términos σ_i y σ_{i+1}, para lo cual se aplica sobre (22) las condiciones (4) y (5). Primero se determina $p_i'(t)$:

$$p_i'(t) = \frac{\sigma_i}{6}\left[\frac{-3(t_{i+1}-t)^2}{h_i} + h_i\right] + \frac{\sigma_{i+1}}{6}\left[\frac{3(t-t_i)^2}{h_i} - h_i\right] + \frac{y_{i+1}-y_i}{h_i} \quad (23)$$

$$p_i'(t_i) = \frac{\sigma_i}{6}\left[\frac{-3(t_{i+1}-t_i)^2}{h_i} + h_i\right] + \frac{\sigma_{i+1}}{6}\left[\frac{3(t_i-t_i)^2}{h_i} - h_i\right] + \frac{y_{i+1}-y_i}{h_i} \quad (24)$$

$$p_i'(t_i) = \frac{\sigma_i}{6}(-3h_i + h_i) + \frac{\sigma_{i+1}}{6}(-h_i) + \frac{y_{i+1}-y_i}{h_i}$$

$$p_i'(t_i) = -\frac{\sigma_i h_i}{3} - \frac{\sigma_{i+1}h_i}{6} + \frac{y_{i+1}-y_i}{h_i} \quad (25)$$

Ahora se debe hallar el término $p'_{i-1}(t_i)$. Para esto se determina primero $p'_i(t_{i+1})$:

$$p'_i(t_{i+1}) = \frac{\sigma_i}{6}\left[\frac{-3(t_{i+1}-t_{i+1})^2}{h_i}+h_i\right]+\frac{\sigma_{i+1}}{6}\left[\frac{3(t_{i+1}-t_i)^2}{h_i}-h_i\right] + \frac{y_{i+1}-y_i}{h_i} \qquad (26)$$

$$p'_i(t_{i+1}) = \frac{\sigma_i}{6}[h_i]+\frac{\sigma_{i+1}}{6}\left[\frac{3(h_i)^2}{h_i}-h_i\right]+\frac{y_{i+1}-y_i}{h_i}$$

$$p'_i(t_{i+1}) = \frac{\sigma_i h_i}{6}+\frac{\sigma_{i+1}h_i}{3}+\frac{y_{i+1}-y_i}{h_i} \qquad (27)$$

Reemplazando i por $i-1$ en (27) se obtiene:

$$p'_{i-1}(t_i) = \frac{\sigma_{i-1}h_{i-1}}{6}+\frac{\sigma_i h_{i-1}}{3}+\frac{y_i-y_{i-1}}{h_{i-1}} \qquad (28)$$

Igualando (25) y (28), con el fin de cumplir con lo establecido en (4), se obtiene:

$$\frac{\sigma_{i-1}h_{i-1}}{6}+\frac{\sigma_i h_{i-1}}{3}+\frac{y_i-y_{i-1}}{h_{i-1}} = -\frac{\sigma_i h_i}{3}-\frac{\sigma_{i+1}h_i}{6} + \frac{y_{i+1}-y_i}{h_i} \qquad (29)$$

$$\frac{\sigma_{i-1}h_{i-1}}{6}+\frac{\sigma_i h_{i-1}}{3}+\frac{\sigma_i h_i}{3}+\frac{\sigma_{i+1}h_i}{6} = \frac{y_{i+1}-y_i}{h_i} - \frac{y_i-y_{i-1}}{h_{i-1}} \qquad (30)$$

$$\sigma_{i-1}h_{i-1}+2\sigma_i(h_{i-1}+h_i)+\sigma_{i+1}h_i = 6\left(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i-y_{i-1}}{h_{i-1}}\right) \qquad (31)$$

Donde *i=1,2,3,…,n-1*; siendo *n* el número de muestras que constituyen la función a interpolar. Al evaluar la ecuación (31) en cada uno de estos valores, se obtendrá un sistema de *n-2* ecuaciones y *n* incógnitas. Los dos datos restantes necesarios para resolver el sistema, resultan al asignar valores a la segunda derivada de los *Splines Cúbicos* en los extremos de la función. En el programa desarrollado estos valores pueden ser ingresados por el usuario.

3. INTERFAZ GRÁFICA

La figura 3 contiene uno de los tramos más relevantes del código desarrollado, en el cual se realiza el cálculo de los *Splines*. Es importante aclarar que para esta operación no se utilizó la función propietaria de *Matlab*.

```
function chkCubico_Callback(hObject, eventdata, handles)
checkBoxStatus=get(handles.chkCubico,'Value');
ssigma0=get(handles.edtInicial,'String');
sigma0=str2num(ssigma0);
ssigmaf=get(handles.edtFinal,'String');
sigmaf=str2num(ssigmaf);
x=handles.x;
y=handles.y;
n=handles.n
if(checkBoxStatus)
    k=n-1
    hk=zeros(1,k);
    for i=1:k
        hk(1,i)=x(1,(i+1))-x(1,i)
        i=i+1
    end
    coef=zeros(n,n);
    indep=zeros(n,1);
    coef(1,1)=1;
    indep(1,1)=sigma0;
    coef(n,n)=1;
    indep(n,1)=sigmaf;
    for i=2:k
        coef(i,(i-1))=hk(1,(i-1));
        coef(i,i)=2*(hk(1,(i-1))+hk(i));
        coef(i,(i+1))=hk(1,i);
        indep(i,1)=6*(((y(1,(i+1))-y(1,i))/hk(1,i))-((y(1,i)-y(1,(i-
      1)))/hk(1,(i-1))));
        i=i+1;
    end
    sigma=zeros(n,1);
    sigma=inv(coef)*indep
    menor=min(hk);
    con=zeros(k,4);
    for i=1:k
        con(i,1)=(sigma((i+1),1)/(6*hk(1,i)))-(sigma(i,1)/(6*hk(1,i)));
        con(i,2)=((sigma(i,1)*x(1,(i+1)))/(2*hk(1,i)))-
((sigma((i+1),1)*x(1,i))/(2*hk(1,i)));
        con(i,3)=((sigma(i,1)*hk(1,i))/6)-
((sigma(i,1)*x(1,(i+1)))*x(1,(i+1)))/(2*hk(1,i)))+((sigma((i+1),1)*x(1,i)
*x(1,i))/(2*hk(1,i)))-((sigma((i+1),1)*hk(1,i))/6)-
(y(1,i)/hk(1,i))+(y(1,(i+1))/hk(1,i));
con(i,4)=((sigma(i,1)*x(1,(i+1)))*x(1,(i+1)))*x(1,(i+1)))/(6*hk(1,i)))-
((sigma(i,1)*hk(1,i)*x(1,(i+1)))/6)-
((sigma((i+1),1)*x(1,i)*x(1,i)*x(1,i))/(6*hk(1,i)))+((sigma((i+1),1)*hk(
1,i)*x(1,i))/6)+((y(1,i)*x(1,(i+1)))/(hk(1,i)))-
((y(1,(i+1))*x(1,i))/(hk(1,i)));
        xpol=x(1,i):(menor/1000):x(1,(i+1));
        splines=(con(i,1)*xpol.*xpol.*xpol)+(con(i,2).*xpol.*xpol)+(con(
        i,3)*xpol)+con(i,4);
        axes(handles.axes1);
        plot(xpol,splines,'r',x,y,'o');
        hold on;
        i=i+1;
        g=con;
        handles.g=g;
        guidata(hObject,handles);
    end
    set(handles.btnGuardar,'Enable','on');
else
   hold off;
   axes(handles.axes1);
   plot(x,y,'o');
   xlabel('Muestras');
   ylabel('Funcion');
   title('Interpolacion');
end
```

Figura 3. Porción más representativa del código desarrollado.

El programa construido en la utilidad *GUIDE* de *Matlab*©, permite al usuario ingresar una cantidad determinada de datos, con el objetivo de obtener el conjunto de polinomios de grado tres que realiza la interpolación de los mismos. La figura 4 contiene la apariencia de la aplicación una vez es ejecutada. Para lograr la reconstrucción de la función, el algoritmo calcula el término h_i determinado por la ecuación (12), los polinomios que interpolan cada tramo establecidos en la expresión (22), y el conjunto de ecuaciones dadas por (31) que permiten obtener los diferentes valores que toma σ_i.

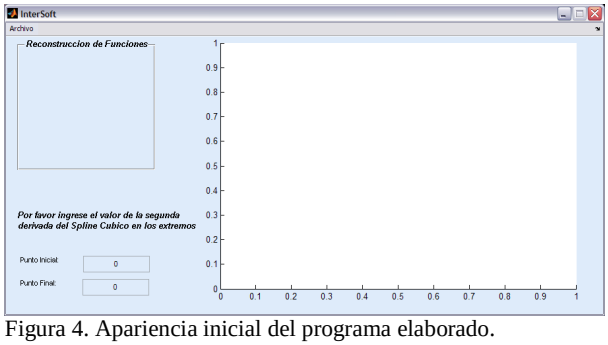

Figura 4. Apariencia inicial del programa elaborado.

El conjunto de valores a interpolar debe ser ingresado en un archivo de texto, en forma de matriz de *n* filas y 2 columnas, siendo *n* el número total de datos. En la fila uno es necesario arreglar los números correspondientes a las abscisas, mientras que la fila 2 debe contener los correspondientes valores de las ordenadas. No es necesario que los datos de la columna uno se encuentren igualmente espaciados, pero si deben aparecer en orden ascendente, ya que de no ser así, no corresponderían a una función sino a una relación.

Para observar el funcionamiento del programa es posible utilizar el siguiente ejemplo: supóngase que se desea interpolar las diez parejas de datos mostrados en la figura 5, los cuales corresponden a las lecturas tomadas de cierta variable física durante un experimento de laboratorio.

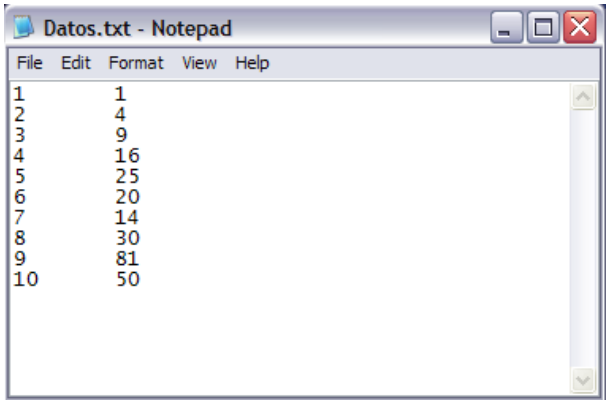

Figura 5. Conjunto de datos a interpolar por *Splines Cúbicos.*

El archivo de texto debe ser cargado accionando el botón *Abrir*, ubicado en el menú *Archivo*, tal como se muestra en la figura 6.

Una vez ingresados los datos, dentro del grupo lógico *"Reconstrucción de Funciones"*, se habilitan los cuadros de chequeo *"Interpolación Lineal"* y *"Spline Cúbico"*, mientras que en el campo de gráficos se muestran todos los datos cargados del archivo de texto. La apariencia de la aplicación en este estado es mostrada en la figura 7.

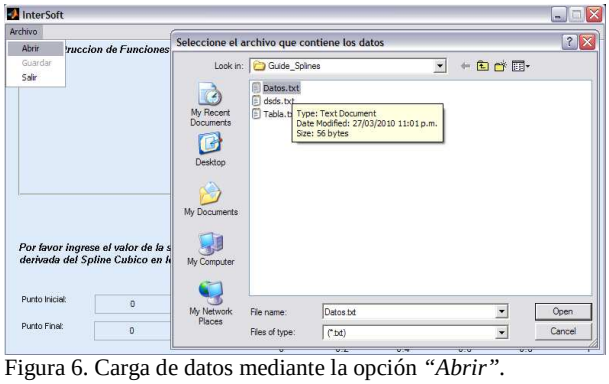

Figura 6. Carga de datos mediante la opción *"Abrir"*.

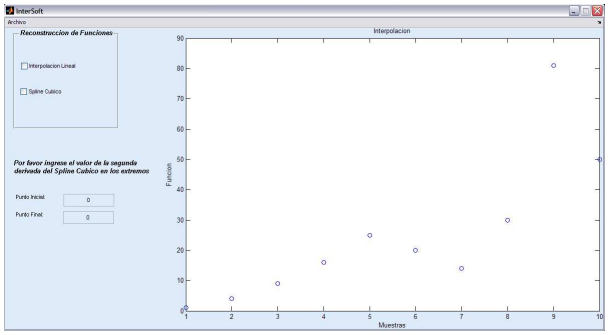

Figura 7. Presentación de los datos ingresados en la interfaz gráfica.

Al seleccionar los cuadros de chequeo *"Interpolación Lineal"* y *"Spline Cúbico"* ubicados el la parte superior izquierda, el programa presenta las curvas obtenidas al conectar los puntos a través de líneas rectas (en color verde), y polinomios de grado tres (color rojo). Esta operación permite comparar las diferencias existentes entre las reconstrucciones realizadas, notándose la enorme ventaja que ofrece la interpolación por *Splines Cúbicos*, la cual consiste en la preservación del carácter continuo de la función interpolada. Esta propiedad es vital a la hora de reconstruir funciones cuyas muestras proceden de variables físicas, (como las tomadas por los estudiantes de ingeniería durante una práctica de laboratorio típica) pues su naturaleza es de curva suave. En la figura 8 se presenta la curva construida por el programa usando las dos estrategias anteriormente mencionadas.

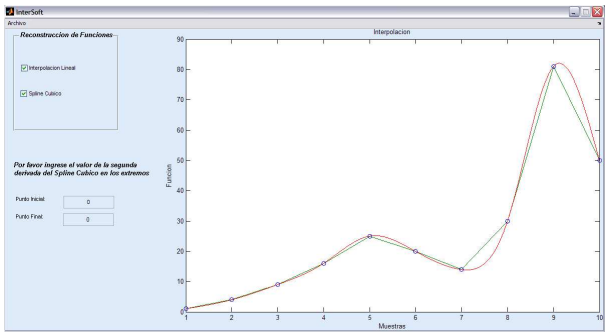

Figura 8. Curva final construida por el programa al aplicar interpolación lineal (color verde) y por *Splines* (color rojo).

En la figura 8 se observa que para construir la función en color rojo, el programa debe generar nueve polinomios individuales de tercer grado, los cuales se pueden recuperar del algoritmo mediante la opción *Guardar*. Esta operación se muestra en la figura 9.

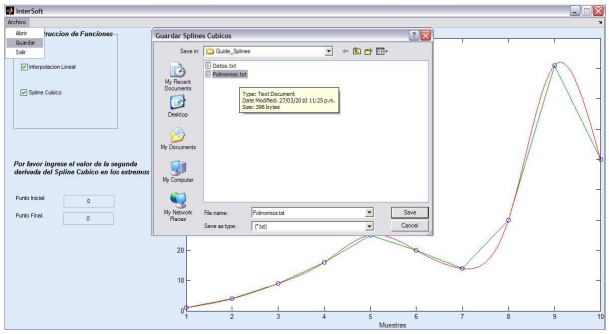

Figura 9. Recuperación de los coeficientes de los polinomios de tercer grado generados por el programa.

Los coeficientes de estos polinomios son almacenados dentro de un archivo de texto de *n* filas y cuatro columnas, donde *n* es el número de tramos reconstruidos. Así, la primera columna contendrá el número que acompaña a t^3 en los *Splines* generados, la segunda corresponderá a los diferentes coeficientes de t^2, la tercera a *t*, mientras que en la última columna se encuentran los números independientes. El archivo de texto que contiene los coeficientes generados para el ejemplo propuesto se muestra en la figura 10.

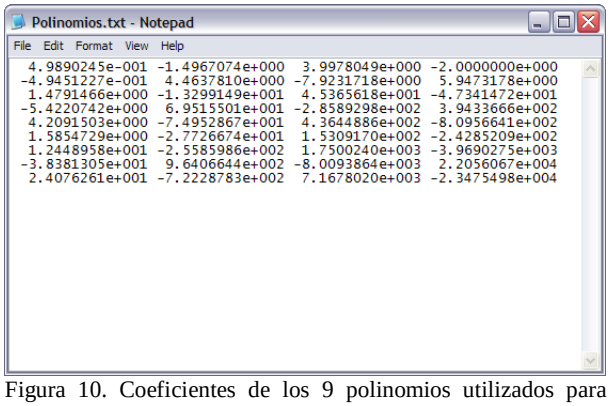

Figura 10. Coeficientes de los 9 polinomios utilizados para interpolar los datos del ejemplo propuesto.

4. CONCLUSIONES Y RECOMENDACIONES

La reconstrucción de funciones por *Splines Cúbicos* presenta una gran ventaja frente a la interpolación lineal, ya que permite mantener el carácter de curva suave inherente a toda variable física, lo cual hace que el programa desarrollado sea bastante útil a la hora de obtener las gráficas solicitadas en las guía de laboratorio de los cursos prácticos de física para ingenierías. Actualmente, los estudiantes de uno de los cursos de Laboratorio de Física III de la Universidad Tecnológica de Pereira, se encuentran utilizando el aplicativo para la interpolación de los datos tomados durante la práctica, arrojando resultados satisfactorios.

Gracias a que el programa permite recuperar los coeficientes de los diferentes polinomios de tercer grado generados al ejecutar el algoritmo, esta interfaz gráfica también puede ser utilizada en los cursos de métodos numéricos con el fin de que los estudiantes puedan verificar la veracidad de los resultados obtenidos durante el desarrollo de la clase teórica.

5. BIBLIOGRAFÍA

[1] Julio G. Benedito, *Métodos numéricos en ingeniería: prácticas con Matlab*, II Edición, Servicio de Publicaciones de la Universidad de Oviedo, 2006.

[2] Hsieh S. Hou, Harry C. Andrews, *Cubic Splines for Image Interpolation and Filtering*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol 26, Nº 6, Diciembre de 1978.

[3] Zhang X. Guang, *A New Kind of Super-Reconstruction Algorithm Based on the ICM and the Constrained Cubic Spline Interpolation*, Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference. IEEE, Agosto de 2008.

[4] Alberto Sierra, *Interpolación Polinomial Cúbica Segmentaria*. Centro de Publicaciones de la Universidad Tecnológica de Pereira. 1981.

[5] Jesús García Quesada, *Tutorial de Análisis Numérico - Interpolación: Splines Cúbicos*, III Edición, Universidad de las Palmas de Gran Canaria, 2 de octubre de 2000, Recuperado el 26 de marzo de 2010 de: http://pcm.dis.ulpgc.es/an/tutor/splines.pdf