UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA-LEÓN
FACULTAD DE CIENCIAS
DEPARTAMENTO DE ESTADÍSTICA-MATEMÁTICA

APROXIMACIÓN DISCRETA POR MÍNIMOS CUADRADOS

MONOGRAFÍA
PARA OPTAR AL TÍTULO DE LICENCIADA EN MATEMÁTICA

PRESENTADA POR
BRA. NORA DEL CARMEN GARCÍA MARTÍNEZ

TUTORA
MSC. ANGELA ALTAMIRANO SLINGER

León, Enero 2006

AGRADECIMIENTO

Por sobre todo a ***DIOS*** *quien con su bendición y amor hizo posible en cada instante de mi vida terminar esta tesis.*

A MI TUTORA *MSC. Ángela Altamirano Slinger, que guió con paciencia y sabiduría este trabajo investigativo y además por haberme brindado siempre su apoyo y amistad.*

A la Dra. *Ana Cristina Rostrán, quien me brindo en todo momento su apoyo.*

A todas *las personas que de una u otra forma, se vieron involucrados en la culminación de esta tesis en especial a:*

Julio Rostrán y Maria Ramona Fonseca.

DEDICATORIA

A MIS PADRES: *Pedro García y Paula Martínez, por darme la vida, brindarme siempre su apoyo, amor y confianza y así lograr mi meta.*

A MIS HIJAS: *María Amanda y María Angélica Rostrán, por ser la razón de mi vida, y empeño para culminar mi carrera.*

A MIS HERMANOS: *Sonia, Carlos y Pedro García, que de una u otra forma contribuyeron en la realización de esta tesis.*

INDICE

Página

INTRODUCCIÓN

A menudo se presenta la necesidad de aproximar una función a un conjunto de datos representados por puntos. Para este fin se han desarrollado diversos métodos. Estos métodos pueden dividirse en dos categorías generales: Regresión e Interpolación.

La Regresión se emplea cuando hay un grado significativo de error asociado a los datos, frecuentemente los resultados experimentales son de esta clase. Para estas situaciones se trata de encontrar una función que represente la tendencia general de los datos sin necesidad de que la gráfica de la función pase por los puntos individuales.

Por otra parte, la Interpolación se maneja cuando hay que determinar valores intermedios entre datos que estén relativamente libres de error. Tal es el caso de la información tabulada. Para estas situaciones se trata de encontrar una función cuya gráfica pase por los puntos y usar esta función para predecir valores intermedios.

Por considerarlo de mayor interés, en este trabajo se aborda el problema de aproximar una función a un conjunto de datos mediante algunos métodos de regresión por mínimos cuadrados en contraste con la aproximación por interpolación.

El presente trabajo consta de tres capítulos. En el primer capítulo se trata la Interpolación Polinomial, obteniendo el polinomio aproximante correspondiente mediante la fórmula explícita de la función o bien una tabla de valores de la función, para luego realizar las interpolaciones que se necesiten.

En el segundo capítulo se analizan algunos métodos de regresión que ajustan funciones que minimizan la suma de los cuadrados de las diferencias entre los datos y la función. En el tercer capítulo se presentan aplicaciones, con algunos casos, donde se utilizan los métodos de Regresión por mínimos cuadrados en contraste con métodos de interpolación.

Finalmente en los anexos se incluyen los programas elaborados para los diversos métodos estudiados, así como su corrida correspondiente.

OBJETIVOS

OBJETIVO GENERAL

- Mostrar algunos métodos de aproximación de funciones por mínimos cuadrados en contraste con algunos métodos de interpolación.

OBJETIVOS ESPECIFICOS

- Establecer algunos métodos para aproximar una función por mínimos cuadrados.
- Caracterizar el método que mejor aproxime una función por mínimos cuadrados.
- Comparar la aproximación por mínimos cuadrados con la interpolación, mediante el análisis de algunos casos.

CAPÍTULO 1
INTERPOLACIÓN POLINOMIAL

1.1 Introducción

El método más común para aproximar una función es el de aproximación por polinomios. Entre los varios tipos de aproximación por polinomios que existen, el más flexible es el de aproximación por el polinomio de interpolación. Interpolar significa estimar el valor desconocido de una función en un punto, tomando una media ponderada de sus valores conocidos en puntos cercanos al dado.

La interpolación polinomial es un método para aproximar el valor de una función en un punto, por medio de un polinomio que se satisface para valores conocidos de la función.

En este capítulo trataremos la aproximación polinomial, utilizando una fórmula explícita que defina la función, o bien una tabla de valores de la función para obtener el polinomio aproximante correspondiente, para luego realizar las interpolaciones que se necesiten.

1.2 Interpolación y Aproximación Polinomial

La forma general de un polinomio de grado n es: $P_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$ donde n es un entero no negativo y $a_0, a_1, ..., a_n$ son constantes reales y $a_n \neq 0$.

Una razón de la importancia de los polinomios es que aproximan uniformemente funciones continuas.

Teorema 1.1 (Teorema de Aproximación de Weierstrass)
Si f es definida y continua en $[a,b]$, y una ε > 0 está dada, supongamos que existe un polinomio P definido en $[a,b]$, con la propiedad de que $|f(x) - P(x)| \pi \in$, para toda $x \in [a,b]$.

El teorema anterior expresa que dada cualquier función definida y continua en un intervalo cerrado, existe un polinomio que se halla tan próximo de la función como se desee.

Algo que se necesita con cierta frecuencia, es estimar valores intermedios entre valores conocidos de una función. El método más común empleado para este propósito es la interpolación polinomial.

Si se tienen n+1 puntos dados de una función, es posible determinar un único polinomio de grado n o menor que pasa a través de todos los puntos. Este polinomio proporciona una fórmula para calcular los valores intermedios requeridos de la función. Este polinomio es conocido como polinomio de interpolación.

Existe una variedad de maneras diferentes de expresar un polinomio de interpolación, entre ellas tenemos: El polinomio de interpolación con diferencias divididas de Newton, el polinomio de interpolación de Lagrange y el polinomio de interpolación segmentaría, que estudiaremos a continuación.

1.3 Polinomio de interpolación con diferencias divididas de Newton

Definición (Diferencias Divididas)

Las diferencias divididas de una función se definen como:

$f[x_k]$ = $f(x_k)$ diferencias divididas de orden cero.

$f\ [x_{k-1}, x_k] = \dfrac{f[x_k] - f[x_{k-1}]}{x_k - x_{k-1}}$ diferencias divididas de orden uno.

$f\ [x_{k-2}, x_{k-1}, x_k] = \dfrac{f[x_{k-1}, x_k] - f[x_{k-2}, x_{k-1}]}{x_k - x_{k-2}}$ diferencias divididas de orden dos.

$f\ [x_{k-3}, x_{k-2}, x_{k-1}, x_k] = \dfrac{f[x_{k-2}, x_{k-1}, x_k] - f[x_{k-3}, x_{k-2}, x_{k-1}]}{x_k - x_{k-3}}$ diferencias divididas de orden tres.

Las diferencias divididas de orden superior se forman de acuerdo con la siguiente regla recursiva:

$$f\ [x_{k-i}, x_{k-i+1}, \ldots, x_k] = \frac{f[x_{k-i+1}, \ldots, x_k] - f[x_{k-i}, \ldots, x_{k+1}]}{x_k - x_{k-i}}$$

Para construir la tabla de diferencias divididas para $y = f(x)$ se considera la siguiente estructura:

x_k	$f[x_0]$	$dif.div.$ $de\ orden 1$	$dif.div.$ $de\ orden 2$	$dif.div.$ $de\ orden 3$	$dif.div.$ $de\ orden 4$
x_0	$f[x_0]$				
x_1	$f[x_1]$	$f[x_0,x_1]$			
x_2	$f[x_2]$	$f[x_1,x_2]$	$f[x_0,x_1,x_2]$		
x_3	$f[x_3]$	$f[x_2,x_3]$	$f[x_1,x_2,x_3]$	$f[x_0,x_1,x_2,x_3]$	
x_4	$f[x_4]$	$f[x_3,x_4]$	$f[x_2,x_3,x_4]$	$f[x_1,x_2,x_3,x_4]$	$f[x_0,x_1,x_2,x_3,x_4]$

La primera columna de la tabla anterior contiene los nodos o puntos a considerar. La segunda columna contiene las diferencias divididas de orden cero de la función f con respecto a x_k. El resto de columnas contiene las diferencias divididas de orden 1 en adelante.

Los coeficientes a_k de los polinomios $P_n(x)$ dependen de los valores de $f(x_i)$ $(con\ i=0,1,...,k)$, el siguiente teorema establece que los a_k pueden calcularse usando diferencias divididas de $f(x)$, esto es. $a_k = f[x_0, x_1,...x_k]$.

Teorema 1.2 (Polinomio Interpolante de Newton)

Supongamos que $x_0, x_1,..., x_k$ son $n+1$ números distintos en $[a,b]$. Entonces existe un único polinomio $P_n(x)$ de grado menor o igual que n tal que

$$f(x_i)=P_n(x_i) \text{ para } i=0,1,...,n.$$

La forma de Newton de este polinomio interpolador es:

$$P_n(x)=a_0+a_1(x-x_0)+\cdots+a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

siendo $a_k = f[x_0, x_i, ..., x_k]$ para $k = 0,1,...,n$.

Observación

Si $\{(x_i, y_i)/\ 0 \le i \le n\}$ es un conjunto de puntos cuyas abscisas son todas distintas, entonces los valores $f(x_i) = y_i$ pueden usarse para construir el único polinomio de grado menor o igual que n que pasa por los $n+1$ puntos dados.

Corolario (Aproximación de Newton)

Supongamos que $P_n(x)$ es el polinomio Interpolador de Newton dado en el teorema 1.2 y que usamos para aproximar la función $f(x)$, esto es,

$$f(x) = P_n(x) + E_n(x)$$

Si $f \in C^{n+1}[a,b]$, entonces para cada $x \in [a,b]$ existe un número ξ(x) en (a,b), tal que el término del error puede escribirse como:

$$E_n(x) = \frac{(x-x_0)-(x-x_1)...(x-x_n)f^{(n+1)}(\xi(x))}{(n+1)!}$$

Observación

Para usar la fórmula de $E_n(x)$ la función f debe ser conocida y diferenciable n+1 veces. Usualmente, este no es el caso. Existe una fórmula alternativa que no requiere conocimiento previo de la función. Sin embargo debe disponerse de un dato adicional $f(x_{n+1})$.

La ecuación siguiente da una aproximación del error E_n usando Diferencias Divididas:

$$E_n \cong f[x_0, x_1, ..., x_n, x_{n+1}](x - x_0)(x - x_1)...(x - x_n)$$

Ejemplo 1.1

Calcular los polinomios interpoladores de Newton de grados 1, 2, 3 y 4 considerando los puntos siguientes:

i	x_i	$f(x_i)$
0	4.0	2.00000
1	5.0	2.23607
2	6.0	2.44949
3	7.0	2.64575
4	8.0	2.82843

Solución:

Calculamos primero la tabla de diferencias divididas para la función tabulada, la cual es:

i	x_i	$f(x_i)$	$dif.\,div.$ $orden\,1$	$dif.\,div.$ $orden\,2$	$dif.\,div.$ $orden\,3$	$dif.\,div.$ $orden\,4$
0	4.0	2.00000				
1	5.0	2.23607	0.23607			
2	6.0	2.44949	0.21342	-0.01133		
3	7.0	2.64575	0.19626	-0.00858	0.000092	
4	8.0	2.82843	0.18268	-0.00679	0.00060	-0.00008

La tabla anterior se obtuvo siguiendo la regla recursiva, por ejemplo:

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{2.23067 - 2.00000}{5.0 - 4.0} = 0.23607$$

$$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{2.44949 - 2.23607}{6.0 - 5.0} = 0.21342$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_1 - x_0} = \frac{0.21342 - 0.23607}{6.0 - 4.0} = -0.01133$$

Usando los nodos x_0, x_1, x_2, x_3, x_4 y los elementos diagonales a_0, a_1, a_2, a_3 y a_4 de la tabla podemos escribir los cuatro polinomios interpoladores de Newton.

$$P_1(x) = 2.00000 + 0.23607(x - 4.0)$$

$$P_2(x) = 2.00000 + 0.23607\,(x-4.0) + 0.01133\,(x-4.0)\,(x-0.5)$$

$$P_3(x) = 2.00000 + 0.23607\,(x-4.0) + 0.01133\,(x-4.0)\,(x-0.5) \\ + \ 0.00092 \quad (x-4.0)\,(x-5.0)(x-6.0)$$

$$P_4(x) = 2.00000 + 0.23607\,(x-4.0) + 0.01133\,(x-4.0)\,(x-0.5) \\ + 0.00092(x-4.0)\,(x-5.0)(x-6.0) - 0.00008(x-4.0)(x-5.0)(x-6.0)(x-7.0)$$

Ejemplo 1.2

Los valores tabulados en el ejemplo 1.1 corresponden a la función $f(x) = \sqrt{x}$. Calcule los valores de los polinomios hallados en el ejemplo 1.1 en los puntos x =4.5 y x = 7.5. Compare los valores obtenidos con los valores de la función $f(x)$.

Solución:

	$P_1(x)$	$P_2(x)$	$P_3(x)$	$P_4(x)$	$f(x)$
$x = 4.5$	2.11804	2.12087	2.12121	2.12129	2.12132
$x = 7.5$	2.82625	2.72711	2.73919	2.73867	2.73861

Observe que para x =4.5 y x =7.5 la mejor aproximación se obtiene para el polinomio de cuarto grado.

Ejemplo 1.3

Calcule el error del polinomio de interpolación de segundo grado $P_2(x)$ del ejemplo 1.1 en la aproximación para x = 4.5, usando los datos x =4.0, x =5.0, x =6.0 y el adicional x = 7.0.

Solución:

$E_n \cong f[x_0, x_1, ..., x_n, x_{n+1}](x-x_0)(x-x_1)...(x-x_n)$

$E_3 \cong (0.00092)(4.5-4.0)(4.5-5.0)(4.5-6.0) = 3.45\times10^{-4}$ que comparado con el error real $2.12132-2.12087 = 4.5\times10^{-4}$ resulta ser del mismo orden.

Para realizar los cálculos con una computadora se necesita disponer del algoritmo para calcular las diferencias divididas y escribir el polinomio $P(x)$ en la forma habitual.

Las diferencias divididas se pueden almacenar en una matriz con elementos $D_{k,i}$ tales que $D_{k,i} = f\,[x_{k-1}, x_{k-i+1}, ...x_k]$ para i ≤k. Los elementos de la matriz se calculan recursivamente usando la fórmula

$$D_{k,i} = \frac{D_{k,i-1} - D_{k-1,i-1}}{x_k - x_{k-i}}$$

ALGORITMO DE DIFERENCIAS DIVIDIDAS DE NEWTON

Para obtener los coeficientes de las diferencias divididas del polinomio de interpolación P en los puntos distintos $x_0, x_1 ... x_n$ para una función f .

Entrada: Los puntos $(x_k, y_k) = (x_k, f(x_k))$ con k =0,1,…n donde $D_{k,0} = y_k$

Salida: Los valores $D_{k,k} = a_k$ con k =0,1,…n., son los coeficientes del polinomio de interpolación de Newton.

Paso 1 Para k =0,1,…n

Para i = 1,2,…k

Sea $D_{k,i} = \dfrac{D_{k,i-1} - D_{k-1,i-1}}{x_k - x_{k-i}}$

Paso 2 Salida $(a_0, a_1, ..., a_n)$ $D_{k,k} = a_k$

Parar

El programa, en el lenguaje C, correspondiente a este algoritmo puede verse en el Anexo No 1.

El polinomio de interpolación de diferencias divididas de Newton se adapta idealmente en aquellos casos en que el grado del polinomio a utilizar en la aproximación se desconoce. El polinomio de Newton es apropiado para tales situaciones ya que se programa fácilmente en un formato que compara los resultados con grados diferentes. Además, se puede incorporar con facilidad una aproximación del error en el método. De esta forma, se puede comparar y escoger a partir de los resultados usando varios polinomios de grados diferentes.

1.4 POLINOMIO DE INTERPOLACIÓN DE LAGRANGE

El polinomio de interpolación de Lagrange, simplemente, es una reformulación del polinomio de Newton que evita los cálculos de las diferencias divididas. Para mostrarlo consideremos el caso del polinomio interpolador de Newton de grado 1.

$$P_1(x) = a_0 + a_1 (x - x_0)$$

$$= f[x_0] + f[x_0, x_1](x - x_0)$$

$$= f[x_0] + \frac{f[x_1] - f[x_0]}{x_1 - x_0}(x - x_0)$$

$$= f[x_0] + \left[\frac{f[x_1]}{x_1 - x_0} - \frac{f[x_0]}{x_1 - x_0} \right](x - x_0)$$

$$= f[x_0] + \frac{x - x_0}{x_1 - x_0} f[x_1] - \frac{x - x_0}{x_1 - x_0} f[x_0]$$

$$= f[x_0] - \frac{x - x_0}{x_1 - x_0} f[x_0] + \frac{x - x_0}{x_1 - x_0} f[x_1]$$

$$= (1 - \frac{x - x_0}{x_1 - x_0}) f[x_0] + \frac{x - x_0}{x_1 - x_0} f[x_1]$$

$$P(x) = \frac{x - x_1}{x_0 - x_1} f[x_0] + \frac{x - x_0}{x_1 - x_0} f[x_1]$$

Esta última expresión será el polinomio interpolador de Lagrange de grado 1

Teorema 1.3 (Polinomio de Interpolación de Lagrange)

Supongamos que $f \in C^{n+1}[a,b]$ y que $x_0, x_i, \ldots, x_n \in [a,b]$ son $n+1$ nodos de interpolación si $x \in [a,b]$ entonces

$$f(x) = P_n(x) + E_n(x)$$

donde $P_n(x) = \sum_{k=0}^{n} L_k(x) f(x_k)$ con $L_k(x) = \prod_{i=1}^{n} \frac{x - x_i}{x_k - x_i}$ es llamado "polinomio interpolador de Lagrange" de f para los nodos dados, y el error será

$$E_n(x) = \frac{(x - x_0)(x - x_1)\ldots(x - x_n) f^{(n+1)}(\xi(x))}{(n+1)!}$$

para alguna $\xi(x)$ en (a,b).

El Polinomio Interpolador de Lagrange de grado uno se representa como:

$$P_1(x) = L_0(x)\ f(x_0) + L_1(x)\ f(x_1) = \frac{x-x_1}{x_0-x_1} f(x_0) + \frac{x-x_0}{x_1-x_0} f(x_1)$$

El Polinomio Interpolador de Lagrange de grado dos se representa como:

$$P_2(x) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2)$$

$$= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2)$$

Observe que $L_k(x)=1$ con $x = x_k$ y $L_k(x)=0$ en todos los demás puntos. Por lo tanto, cada producto $L_i(x) f(x_i)$ toma un valor de $f(x_i)$ en el punto x_i. Por consiguiente la sumatoria de todos los productos dada en la ecuación $P(x)=\sum_{k=0}^{n} L_k(x) f(x_k)$ es el único polinomio de n-esimo grado que pasa exactamente por $n+1$ puntos. Al igual que el método de Newton, la ecuación de Lagrange tiene un error aproximado dado por

$$E_n \cong f[x_0, x_1, ..., x_n, x_{n+1}](x-x_0)(x-x_1)...(x-x_2).$$

Ejemplo 1.4

Dado los puntos

x	0	0.5	1.0	1.5	2.0	2.5
$f(x)$	1	2.119	2.910	3.945	5.720	8.695

Calcule f (1.6) usando polinomios de interpolación de Lagrange de grado 1 hasta el 3. Escójase la secuencia de puntos de las aproximaciones para lograr exactitud.

Solución:

Polinomio de Interpolación de Lagrange de primer grado

$$P_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

x_0 =1.5 x_1 =2.0

P_1(1.6)= $L_0(1.6) f(1.5) + L_1(1.6) f(2.0) = \frac{1.6 - 2.0}{1.5 - 2.0}(3.945) + \frac{1.6 - 1.5}{2.0 - 1.5}(5.720) = 4.3$

Polinomio de Interpolación de Lagrange de segundo grado:

$x_0 = 1.0 \quad x_1 = 1.5 \quad x_2 = 2.0$

$$P_2(1.6) = L_0(1.6) f(x_0) + L_1(1.6) f(x_1) + L_2(1.6) f(x_2) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1)$$

$$+ \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

$$P_2(1.6) = \frac{(1.6 - 1.5)(1.6 - 2)}{(1 - 1.5)(1 - 2)}(2.910) + \frac{(1.6 - 1)(1.6 - 2)}{(1.5 - 1)(1.5 - 2)}(3.945) + \frac{(1.6 - 1)(1.6 - 1.5)}{(2 - 1)(2 - 1.5)}(5.720) = 4.2408$$

Polinomio de Interpolación de Lagrange de tercer grado

$x_0 = 1.0 \quad x_1 = 1.5 \quad x_2 = 2.0 \quad x_3 = 2.5$

$$P_3(1.6) = L_0(1.6) f(x_0) + L_1(1.6) f(x_1) + L_2(1.6) f(x_2) + L_3(1.6) f(x_3)$$

$$P_3(1.6) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} f(x_0) + \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} f(x_1)$$

$$+ \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} f(x_2) + \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} f(x_3)$$

$$P_3(1.6) = \frac{(1.6-1.5)(1.6-2)(1.6-2.5)}{(1-1.5)(1-2)(1-2.5)}(2.910) + \frac{(1.6-1)(1.6-2)(1.6-2.5)}{(1.5-1)(1.5-2)(1.5-2.5)}(3.945)$$

$$+\frac{(1.6-1)(1.6-1.5)(1.6-2.5)}{(2-1)(2-1.5)(2-2.5)}(5.720) + \frac{(1.6-1)(1.6-1.5)(1.6-2)}{(2.5-1)(2.5-1.5)(2.5-2)}(8.695) = 4.22608$$

El polinomio de interpolación de Lagrange es apropiado cuando el grado del polinomio de aproximación se conoce a priori.

Para realizar los cálculos con una computadora se necesita disponer del algoritmo para generar aproximaciones con polinomios de Lagrange.

El procedimiento para generar recurrentemente los polinomios de Lagrange se llama Método de Neville, que tiene la ventaja de utilizar los cálculos previos. El polinomio de Lagrange que concuerda con f en los $k+1$ puntos $x_0, x_1, x_2, ..., x_k$ se representa por $P_{0,1,...,k}$.

Teorema 1.4

Sea f que esté definida en $x_0, x_1, ..., x_k$ y sean x_j y x_i dos puntos de este conjunto. Entonces

$$P(x) = \frac{(x-x_j)P_{0,1,...,j-1,j+1,...,k} - (x-x_i)P_{0,1,...,i-1,i+1,...,k}}{x_i - x_j}$$

describe el k-ésimo polinomio de Lagrange que interpola a f en los $k+1$ puntos $x_0, x_1, ..., x_k$.

La aproximación de los polinomios se puede almacenar en una matriz con elementos $Q_{i,j}$. Talque $Q_{i,j}$ para $0 \le i \le j$ representa el polinomio de interpolación de grado j en los $(j+1)$ números $x_{i-j}, x_{i-j+1}..., x_{i-1}, x_i$ esto es, $Q_{i,j} = P_{i-j,i-j+1,\ldots,i-1,i}$.

ALGORITMO DE NEVILLE

Para evaluar en el punto al polinomio interpolador P en los $(n+1)$ puntos distintos $x_0,...,x_n$, para la función f.

Entrada: Los puntos $(x_k, y_k) = (x_k, f(x_k))$ con k = 0,1,…n donde $Q_{k,0} = y_k$

Salida: La tabla Q con $P(x) = Q_{n,n}$

Paso 1. Para i =1,2,…n

Para j =1,2…,i

Tomar $Q_{i,j} = \dfrac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}$

Paso 2. **SALIDA** (Q);

PARAR

El programa correspondiente a este algoritmo puede verse en el Anexo No 2, así como también las salidas para los polinomios considerados en el ejemplo 1.4.

1.5 INTERPOLACIÓN SEGMENTARÍA (SPLINE)

Frecuentemente la interpolación polinomial para un conjunto numeroso de n +1 datos $\{(x_k, y_k)\}_{k=0}^{n}$ resulta ser muy poco satisfactoria, ya que el polinomio de grado n obtenido puede llevar a resultados erróneos.

Una alternativa es la de aplicar polinomios de grado inferior a subconjuntos de datos. Estos polinomios conectados se llaman funciones de interpolación segmentaría (en inglés, Spline Functions).

1.5.1 INTERPOLACIÓN SEGMENTARIA LINEAL

El polinomio más simple que podemos usar, es un polinomio de grado 1, produce una línea quebrada que consta de los segmentos rectilíneos que unen los puntos consecutivamente.

La expresión que define cada trozo de esta línea quebrada es $y = y_k + m_k(x - x_k)$ para $x_k \leq x \leq x_{k+1}$ siendo $m_k = \dfrac{y_{k+1} - y_k}{y_{k+1} - x_k}$ la pendiente de recta que une los puntos (x_k, y_k) y (x_{k+1}, y_{k+1}), esta ecuación se usa en la evaluación de funciones de cualquier punto entre x_0 y x_n localizando primero el intervalo dentro del cual se encuentra el punto. Después se usa la ecuación apropiada y se determina el valor funcional dentro del intervalo.

Ejemplo1. 5

Dado los datos de la tabla siguiente, con interpolación segmentaría lineal, evalúe la función en x = 5.

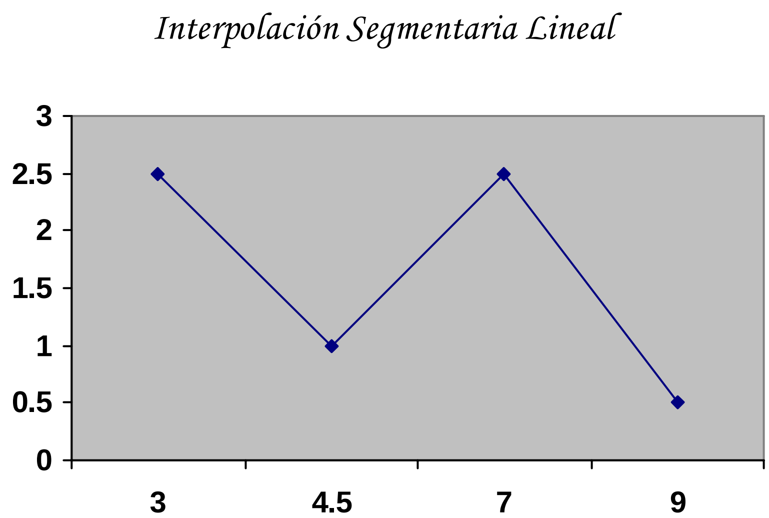
x	3.0	4.5	7.0	9.0
y	2.5	1.0	2.5	0.5

Solución:

Ubicamos primero el intervalo donde se encuentra x = 5. Los intervalos considerados son $[3.0, 4.5]$, $[4.5, 7.0]$ y $[7.0, 9.0]$. Luego $x \in [4.5, 7.0]$. Calcularemos la pendiente del segmento rectilíneo

$$m = \frac{2.5 - 1.0}{7.0 - 4.5} = 0.60$$

El valor de y para x = 5 es $y = 1.0 + 0.60(5 - 4.5) = 1.3$. La gráfica de los polinomios de primer grado para estos datos se presenta en la figura siguiente



Observe que en los puntos donde coincidan los polinomios (llamados nodos) la pendiente cambia abruptamente. En términos formales la primera derivada de la función es discontinua en estos puntos. Esta deficiencia se supera con el uso de polinomios de grado superior que aseguran uniformidad en los nodos.

1.5.2 INTERPOLACION SEGMENTARIA CUADRÁTICA

Para asegurar que las m-ésimas derivadas sean continuas en los nodos, se debe usar un polinomio de al menos (m +1)-ésimo grado.

Los polinomios cuadráticos tienen la primera derivada continua en los nodos. Aunque los polinomios cuadráticos no garantizan segundas derivadas iguales en los nodos, sirven muy bien para mostrar el procedimiento general en el desarrollo de polinomios Interpolantes segmentarios de orden superior.

El objetivo de los polinomios cuadráticos es el de obtener un polinomio de segundo grado para cada uno de los intervalos entre los puntos. El polinomio para cada uno de los intervalos se representa generalmente como:

$$y_i = a_i x^2 + b_i x + c_i\,.$$

Para los $n+1$ puntos (i = 0, 1,2…n), existen n intervalos, y por lo tanto, 3n Incógnitas constantes para evaluar (las a, las b, y las c). Por lo tanto, se requieren 3n ecuaciones o condiciones para evaluar las incógnitas. Además:

1. Los valores de los polinomios deben ser iguales en los nodos interiores. Esta condición se representa mediante las ecuaciones:

$$a_{i+1} x_{i-1}^2 + b_{i-1} x_{i-1} + c_{i-1} = y(x_{i-1})$$

$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = y(x_{i-1})$$

para i =2 hasta n. Como se usan solo los nodos interiores, las ecuaciones anteriores proporcionan cada una n-1 condiciones, con un total de 2n-2.

2. El primer y el último polinomio deben pasar a través de los puntos finales. Esto agrega dos ecuaciones adicionales:

$$a_1 x_0^2 + b_1 x_0 + c_1 = y(x_0)$$

$$a_n x_n^2 + b_n x_n + c_n = y(x_n)$$

con un total de 2n-2+2 =2n condiciones.

3. Las primeras derivadas en los nodos interiores deben ser iguales. La primera derivada en la ecuación $y = ax^2 + bx + c$ es $y' = 2ax + b$. Por lo tanto la condición se representa generalmente como $2a_i x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i$ para i =2 hasta n.

Esto proporciona otras n-1 condiciones con un total de 2n+ n -1= 3n -1. Debido a que hay 3n incógnitas, se tiene una condición menos. A menos que exista una información adicional en relación a los polinomios o sus derivadas, se debe escoger arbitrariamente una condición para calcular eficientemente las constantes. Aunque hay algunas alternativas diferentes que se pueden hacer, aquí se escoge la siguiente:

4. Se supone que la segunda derivada es cero en el primer punto. Ya que la segunda derivada de la ecuación $y_i = a_i x^2 + b_i x + c_i$ es $2a_i$ esta condición se expresa matemáticamente como $a_1 = 0$.

La interpretación visual de esta condición es que los primeros dos puntos se conectarán mediante una línea recta.

Ejemplo 1.6

Ajuste polinomios cuadráticos por segmentos a los datos usados en el ejemplo 1.5. Use los resultados para calcular el valor de la función para x = 5.

Solución:

Teniendo cuatro datos y n =3 intervalos. Con ayuda del ejemplo 1.5, sabemos que el intervalo donde se encuentra x =5 es $[4.5, 7.0]$. Por lo tanto, se debe determinar

3(3) =9 incógnitas. Las ecuaciones $a_{i+1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = y(x_{i-1})$ y $a_{i+1}x_{i-1}^2 + b_i x_{i-1} + c_i = y(x_{i-1})$ nos llevan a las 2(3)-2 = 4 condiciones siguientes:

$$
\begin{array}{ccccccc}
20.25a_1 & + & 4.5b_1 & + & c_1 & = & 1.0 \\
20.25a_2 & + & 4.5b_2 & + & c_2 & = & 1.0 \\
49a_2 & + & 7b_2 & + & c_2 & = & 2.5 \\
49a_3 & + & 7b_3 & + & c_3 & = & 2.5
\end{array}
$$

Evaluando el primer y último polinomio, $a_1x_0^2 + b_1x_0 + c_1 = y(x_0)$ y $a_nx_n^2 + b_n, x_n + c_n = y(x_n)$, por los valores inicial y final se agregan dos ecuaciones más

$$9a_1 + 3b_1 + c_1 = 2.5$$

$$81a_3 + 9b_3 + c_3 = 0.5$$

La igualdad de las derivadas $2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i$ crea adicionalmente 3 -1 = 2 ecuaciones.

$$9a_1 + b_1 = 9a_2 + b_2$$

$$14a_2 + b_2 = 14a_3 + b_3$$

Finalmente la ecuación $a_1 = 0$, el problema se reduce a resolver nueve ecuaciones simultáneamente.

$$
\begin{cases}
20.25a_1 + 4.5b_1 + c_1 = 1.0 \\
20.25a_2 + 4.5b_2 + c_2 = 1.0 \\
49a_2 + 7b_2 + c_2 = 2.5 \\
49a_3 + 7b_3 + c_3 = 2.5 \\
9a_1 + 3b_1 + c_1 = 2.5 \\
81a_3 + 9b_3 + c_3 = 0.5 \\
9a_1 + b_1 - 9a_2 - b_2 = 0 \\
14a_2 + b_2 - 14a_3 - b_3 = 0 \\
a_0 = 0
\end{cases}
$$

Con la ayuda del programa de Eliminación Gaussiana con Sustitución hacia atrás que está en el Anexo No. 3. Se obtiene como solución del sistema

a_1 =1.24802e-006

b_1 =-1.00001

c_1 =5.50002

a_2 =0.639999

b_2 =-6.75999

c_2 =18.46

a_3 =-1.6

b_3 =24.6

c_3 =-91.3

Estos coeficientes se sustituyen en las ecuaciones cuadráticas originales, desarrollando la relación siguiente para cada intervalo:

$$y_1(x) = 1.24802 \times 10^{-6} x^2 - 1.00001x + 5.50002 \qquad 3.0 \le x \le 4.5$$

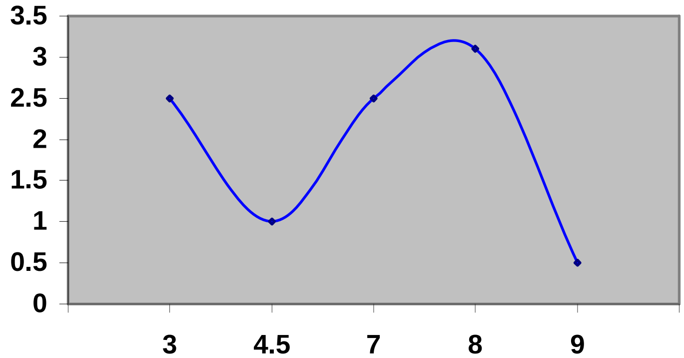
$$y_2(x) = 0.639999x^2 - 6.75999x + 18.46 \qquad 4.5 \le x \le 7.0$$

$$y_3(x) = -1.6x^2 + 24.6x - 91.3 \qquad 7.0 \le x \le 9.0$$

La predicción para $x = 5$ es, por la tanto

$$y_2(5) = 0.639999(5)^2 - 6.75999\,(5) + 18.46 = 0.660025$$

Interpolación Segmentaria Cuadrática



El ajuste polinomial total se muestra en la figura anterior. Nótese que hay dos inconvenientes en el ajuste: 1) La línea casi recta que une los primeros dos puntos y 2) el polinomio del último intervalo parece serpentear demasiado alto. Hay polinomios que no muestran estos inconvenientes por ejemplo los polinomios segmentarios cúbicos, que en general son mejores métodos de interpolación segmentaría.

CAPÍTULO 2

APROXIMACIÓN POR MINIMOS CUADRADOS

2.1 Introducción

El estudio de aproximación involucra dos tipos generales de problemas, uno de ellos la aproximación polinomial, ya fue abordado en el capítulo 1. El otro problema es el concerniente a ajustar funciones a un conjunto de datos y encontrar la “mejor” función de cierta clase que pueda usarse para representarlos.

Una estrategia para obtener esta función aproximada es que se ajuste el comportamiento de los datos sin coincidir necesariamente con cada punto. Además se debe tener un criterio que cuantifique la suficiencia del ajuste, esto es, un criterio que minimice la diferencia entre los datos y la función.

En este capítulo se analiza un método para llevar a cabo este objetivo, al que se le llama regresión por mínimos cuadrados.

2.2 Regresión por mínimos cuadrados

La regresión se emplea para desarrollar la “mejor” curva que ajuste todas las tendencias de los datos sin pasar necesariamente a través de algún punto. Todos los métodos de regresión se diseñan de manera que ajusten funciones que minimicen la suma de los cuadrados de las diferencias entre los datos y la función. A estos métodos se les conoce como regresión por mínimos cuadrados.

Estudiaremos la regresión por mínimos cuadrados lineal, la regresión polinomial, la regresión lineal múltiple y la regresión no lineal.

2.2.1 Regresión Lineal

El ejemplo más simple de una aproximación por mínimos cuadrados es el ajuste de una línea recta a un conjunto de parejas de datos observados $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. La expresión matemática de una línea recta es $y = a_0 + a_1 x$, en donde a_0 y a_1, son coeficientes que representan la intersección con el eje de las abscisas y la pendiente, respectivamente.

Si E es el error entre el modelo usado y las observaciones, entonces el error E es la diferencia entre el valor real de y y el valor aproximado $a_0 + a_1 x$ predicho por la ecuación lineal, esto es, una estrategia para obtener la “mejor” línea a través de los puntos es el de minimizar la suma de los cuadrados de los errores S_r de la siguiente manera $S_r = \sum_{i=1}^{n} E_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$ este criterio tiene muchas ventajas, entre ellas, el que ajusta una línea única a un conjunto dado de datos.

A continuación, se muestra un método que determina los valores de a_0 y a_1, que minimizan la ecuación $S_r = \sum_{i=1}^{n} E_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$.

Para determinar los valores de las constantes a_0 y a_1, se deriva la ecuación $S_r = \sum_{i=1}^{n} E_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$ con respecto a cada uno de los coeficientes:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^{n} [(y_i - a_0 - a_1 x_i)\, x_i]$$

Igualando estas derivadas a cero, se genera un mínimo S_r. Es decir,

$$0 = \sum y_i - \sum a_0 - \sum a_1 x_i$$
$$0 = \sum y_i x_i - \sum a_0 x_i - \sum a_1 x_i^2$$

Ahora, considerando que $\sum a_0 = na_0$, las ecuaciones se pueden expresar como un conjunto de dos ecuaciones lineales simultaneas con dos incógnitas (a_0, a_1)

$$na_0 + \sum x_i a_1 = \sum y_i$$
$$\sum x_i a_0 + \sum x_i^2 a_1 = \sum y_i x_i$$

En donde todas las sumatorias van desde $i = 1$hasta n. A estas ecuaciones se les conoce como ecuaciones normales de Gauss, se pueden resolver simultáneamente y obtener

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

Sustituimos este resultado en la ecuación $na_0 + \sum x_i a_1 = \sum y_i$ obtenemos $a_0 = \bar{y} - a_1 \bar{x}$ en donde $\bar{y}$ y $\bar{x}$ son la media de y y x respectivamente.

Para calcular el error en la regresión lineal tenemos la ecuación $S_{y/x} = \sqrt{\frac{S_r}{n-2}}$ en donde $S_{y/x}$ se llama error estándar de la aproximación. La notación con subíndice "y/x" indica que el error es para un valor predicho de y correspondiente a un valor particular de x. También notemos que ahora la división es por $n-2$ ya que se usan dos aproximaciones obtenidas de los datos a_0 y a_1 para calcular S_r.

El error estándar de la aproximación $S_{y/x}$ cuantifica la dispersión alrededor de la línea de regresión, contrario a la desviación estándar original, s, que cuantifica la dispersión alrededor de la media.

Para obtener la eficiencia del ajuste determinamos la suma de los cuadrados alrededor de la media para la variable dependiente (en este caso y). Se le puede llamar a ésta S_t. Después de llevar a cabo la regresión lineal, se puede calcular S_r, que es la suma de los cuadrados de los residuos alrededor de la línea de regresión. Éste presenta la dispersión que existe después de la regresión. La diferencia $S_t - S_r$ cuantifica la mejora en la reducción del error debido al modelo de la línea recta. Esta diferencia se puede normalizar al error total y obtener $r^2 = \frac{S_t - S_r}{S_t}$ en donde r es el coeficiente de correlación y r^2 es el coeficiente de determinación.

Ejemplo 2.1

Ajústese una línea recta a los valores x y y de la siguiente tabla.

x_i	1	2	3	4	5	6	7
y_i	0.5	2.5	2.0	4.0	3.5	6.0	5.5

Solución:

Resumiendo los cálculos, tenemos

x_i	y_i	$x_i y_i$	x_i^2
1	0.5	0.5	1
2	2.5	5	4
3	2	6	9
4	4	16	16
5	3.5	17.5	25
6	6	36	36
7	5.5	38.5	49
28	24	119.5	140

Luego

$$n = 7$$
$$\bar{x} = \frac{28}{7} = 4$$
$$\bar{y} = \frac{24}{7} = 3.428571429$$

$$\sum x_i = 28$$
$$\sum y_i = 24$$
$$\sum x_i^2 = 140$$
$$\sum x_i y_i = 119.5$$

Ahora usando las ecuaciones:

$a_1 = \dfrac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$ y $a_0 = \bar{y} - a_1\bar{x}$ para encontrar a_0 y a_1 tenemos que

$$a_1 = \frac{7(119.5) - 28(24)}{7(140) - (28)^2} = \frac{164.5}{196} = 0.839285714$$

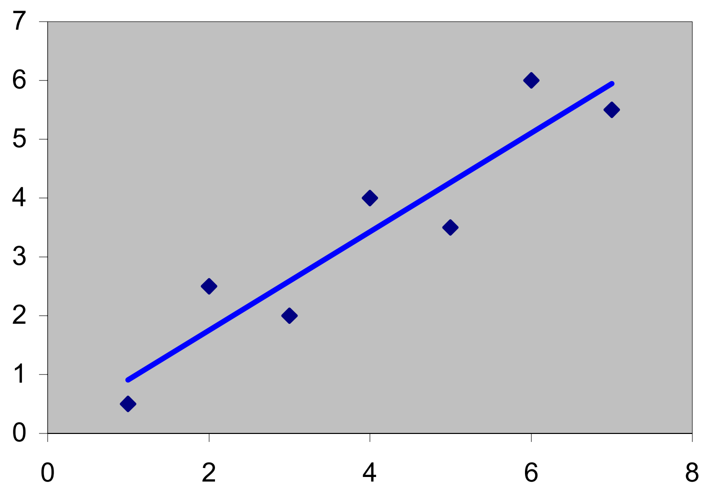
$$a_0 = 3.428571429 - 0.839285714(4) = 0.071428573$$

Por lo tanto el ajuste con mínimos cuadrados lineal es:

$$y = 0.071428573 + 0.839285714x$$

En la figura siguiente se presenta la recta de regresión junto con los datos considerados.

Regresión Lineal



Ejemplo 2.2

Calcular la desviación estándar total, el error estándar de la aproximación y el coeficiente de correlación de los datos del ejemplo 2.1.

Solución:

Resumimos los siguientes cálculos:

x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1 x_i)^2$
1	0.5	8.5765	0.1687
2	2.5	0.8622	0.5625
3	2.0	2.0408	0.3473
4	4.0	0.3265	0.3265
5	3.5	0.0051	0.5896
6	6.0	6.6122	0.7972
7	5.5	4.2908	0.1993
$\sum$	24	22.7413	2.9911

Tenemos que:

S_t =22.7143

S_r =2.9911

Por lo tanto la desviación estándar esta dada por

$$S_y = \sqrt{\frac{S_t}{n-1}} = \sqrt{\frac{22.7143}{7-1}} = 1.9457$$

Y el error estándar es

$$S_{y/x} = \sqrt{\frac{S_r}{n-2}} = \sqrt{\frac{2.9911}{7-2}} = 0.7735$$

Por lo tanto ya que $S_{y/x} \pi S_y$, el modelo de regresión lineal es aceptable. El alcance de la mejoría se cuantifica mediante:

$$r^2 = \frac{S_t - S_r}{S_t} = \frac{22.7146 - 2.9911}{22.7146} = 0.868$$

$$r = \sqrt{0.868} = 0.932$$

Este resultado indica que el 86.8% de la incertidumbre original se ha explicado mediante el modelo lineal.

El programa, en el lenguaje C, correspondiente a este modelo puede verse en el Anexo No 4, así como la salida para los ejemplos 2.1 y 2.2.

2.2.2 Regresión Polinomial

Para ajustar mejor una curva a los datos, otra alternativa es ajustar un polinomio a los datos usando regresión polinomial. El procedimiento de mínimos cuadrados se puede extender fácilmente y ajustar datos a un polinomio de m-esimo grado:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$$

En este caso, la suma de los cuadrados de los errores es: $S_r = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m \right)^2$

Siguiendo el mismo procedimiento empleado para la regresión lineal se toma la derivada de la ecuación $S_r = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m \right)^2$ con respecto a cada uno de los coeficientes del polinomio, para obtener:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 \ldots - a_m x_i^m \right)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^{n} x_i \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m \right)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum_{i=1}^{n} x_i^2 \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_m x_i^m \right)$$

.

.

.

$$\frac{\partial S_r}{\partial a_m} = -2 \sum_{i=1}^{n} x_i^m \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 \ldots - a_m x_i^m \right)$$

Ahora, considerando $\sum a_0 = n a_0$, estas ecuaciones se pueden igualar a cero y reordenando de tal forma que se obtenga el siguiente conjunto de ecuaciones normales:

$$a_0 n + a_1 \sum x_i + a_2 \sum x_i^2 + ... + a_m \sum x_i^m = \sum y_i$$

$$a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + ... + a_m \sum x_i^{m+1} = \sum x_i y_i$$

$$a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + ... + a_m \sum x_i^{m+2} = \sum x_i^2 y_i$$

$$\begin{array}{ccccc} . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{array}$$

$$a_0 \sum x_i^m + a_1 \sum x_i^{m+1} + a_2 \sum x_i^{m+2} + ... + a_m \sum x_i^{2m} = \sum x_i^m y_i$$

En donde todas las sumatorias van desde $i = 1$ hasta n. Nótese que las $m+1$ ecuaciones anteriores son lineales y tienen $m+1$ incógnitas: $a_0, a_1, ..., a_m$. Los coeficientes de las incógnitas se pueden calcular directamente de los datos observados. Por lo tanto, el problema de determinar polinomios de grado m con mínimos cuadráticos es equivalente a resolver un sistema de $m+1$ ecuaciones lineales simultáneas.

Así como en la regresión lineal, el error en la regresión polinomial se puede cuantificar mediante el error estándar de la aproximación que es: $S_{y/x} = \sqrt{\dfrac{S_r}{n-(m+1)}}$ en donde m es el orden del polinomio. Esta cantidad se divide por $n-(m+1)$ ya que se usarón $m+1$ coeficientes $a_0, a_1, ... a_m$ derivados de los datos; por lo tanto se han perdido $m+1$ grados de libertad. Además del error estándar, se puede calcular también el coeficiente de correlación en la regresión polinomial de la misma manera que para el caso lineal que es: $r^2 = \dfrac{S_t - S_r}{S_t}$

Ejemplo 2.3

Ajústese un polinomio de segundo orden a los datos de la siguiente tabla

x	0	1	2	3	4	5
y	2.1	7.7	13.6	27.2	40.9	61.1

Solución:

Resumiendo los cálculos tenemos:

	x_i	y_i	$\sum x_i^2$	$\sum x_i^3$	$\sum x_i^4$	$\sum x_i y_i$	$\sum x_i^2 y_i$
	0	2.1	0	0	0	0	0
	1	7.7	1	1	1	7.7	7.7
	2	13.6	4	8	16	27.2	54.4
	13	27.2	9	27	81	81.6	244.8
	4	40.9	16	64	256	163.6	654.4
	5	61.1	25	125	625	305.5	1527.5
$\sum$	15	152.6	55	225	979	585.6	2488.8

Luego,

$$m = 2 \qquad n = 6 \qquad \bar{x} = 2.5 \qquad \bar{y} = 25.433$$

$$\sum x_i = 15 \qquad \sum y_i = 152.6 \qquad \sum x_i^2 = 55 \qquad \sum x_i^3 = 255$$

$$\sum x_i^4 = 979 \qquad \sum x_i y_i = 585.6 \qquad \sum x_i^2 y_i = 2488.8$$

Por lo tanto las ecuaciones lineales simultáneas son:

$$6a_0 + 15a_1 + 55a_2 = 152.6$$
$$15a_0 + 55a_1 + 225a_2 = 585.6$$
$$55a_0 + 225a_1 + 979a_2 = 2488.8$$

Resolviendo este sistema con la ayuda del programa de Eliminación Gaussiana con Sustitución hacia atrás que está en el Anexo No. 3. Se obtiene como solución del sistema

a_0 =2.47859

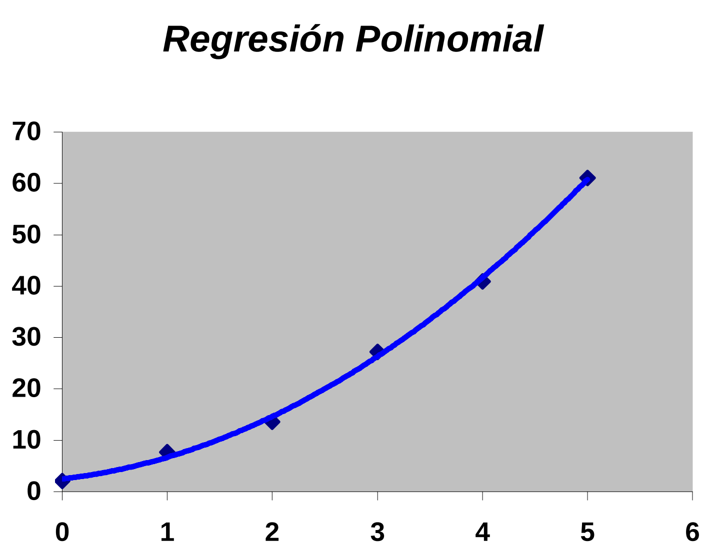
a_1=2.35926

a_2=1.86072

Y por lo tanto, la ecuación cuadrática con mínimos cuadrados en este caso es:

$$y = 2.47859 + 2.35926x + 1.86072x^2$$

En la figura siguiente se presenta el ajuste de un polinomio de segundo orden y los datos considerados.



Con

$S_t = \sum_{i=1}^{6} \left(y_i - \bar{y}\right)^2 = 2513.39$

$S_r = \sum_{i=1}^{6} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2 - a_3 x_i^3\right)^2 = 3.74657$

$$S_{y/x} = \sqrt{\frac{S_r}{n-(m+1)}} = \sqrt{\frac{3.74657}{6-3}} = 1.12$$

El coeficiente de correlación es:

$$r^2 = \frac{2513.39 - 3.74657}{2513.39} = 0.99851$$

Y el coeficiente de determinación es: $r = 0.99925$

Estos resultados indican que el 99.851% de la incertidumbre se ha explicado mediante el modelo; este resultado apoya que la ecuación cuadrática representa un ajuste perfecto.

Para realizar los cálculos con una computadora se necesita disponer del algoritmo de regresión polinomial.

ALGORITMO PARA LA REGRESION POLINOMIAL

Paso 1: Introducir el orden del polinomio ajustado, m.

Paso 2: Introducir el número de puntos, n

Paso 3: Si n ≤ m, imprimir un mensaje de error de que el polinomio de regresión es imposible y termina el proceso. Si n > m, continuar

Paso 4: Calcular las sumas y productos de potencia.

Paso 5: Arreglar estas sumas y productos en forma de matriz aumentada.

Paso 6: Resolver la matriz aumentada en los coeficientes $a_0, \ldots, a_m$, usando un método de eliminación.

Paso 7: Imprimir los resultados.

El programa, en el lenguaje C, correspondiente a este algoritmo puede verse en el Anexo No 5.

2.2.3 Regresión Lineal Múltiple

Una extensión útil en la regresión lineal es el caso en que y es una función lineal de dos o más variables. Por ejemplo, y pudiera ser una función lineal de x_1 y x_2 de la forma:

$$y = a_0 + a_1 x_1 + a_2 x_2 \,.$$

Tal ecuación es útil particularmente cuando se ajusta datos experimentales en donde la variable que se está analizando, a menudo es función de otras dos variables. En este caso bidimensional, la "línea" de regresión viene a ser un "plano".

Como con los casos anteriores, los "mejores" valores de los coeficientes se determinan agrupando la suma de los cuadrados de las diferencias entre los datos y la función, $S_r = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i} \right)^2$ y derivando con respecto a cada uno de los coeficientes tenemos:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i} \right)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^{n} x_{1i} \left(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i} \right)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum_{i=1}^{n} x_{2i} \left(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i} \right)$$

Los coeficientes que generan la suma mínima de los cuadrados de las diferencias se obtiene igualando cada una de las derivadas parciales a cero y expresando las ecuaciones anteriores como un conjunto de ecuaciones lineales simultáneas, de la forma:

$$na_0 + \sum x_{1i}a_1 + \sum x_{2i}a_2 = \sum y_i$$

$$\sum x_{1i}a_0 + \sum x_{1i}^2 a_1 + \sum x_{1i}, x_{2i}a_2 = \sum x_{1i}y_i$$

$$\sum x_{2i}a_0 + \sum x_{1i}x_{2i}a_1 + \sum x_{2i}^2 a_2 = \sum x_{2i}y_i$$

O en forma matricial

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 a_1 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}, x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \end{bmatrix}$$

en donde todas las sumatorias van desde $i = 1$ hasta n.

La regresión lineal múltiple se puede formular en el caso más general como:

$$y = a_0 + a_1x_1 + a_2x_2 + ... + a_mx_m$$

en donde los coeficientes que minimizan la suma de los cuadrados de las diferencias se determina resolviendo el sistema:

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} & . & . & . & \sum x_{mi} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{2i}x_{1i} & . & . & . & \sum x_{1i}x_{mi} \\ \sum x_{2i} & \sum x_{2i}x_{1i} & \sum x_{2i}^2 & . & . & . & \sum x_{2i}x_{mi} \\ . & . & . & & & & . \\ . & . & . & & & & . \\ . & . & . & & & & . \\ \sum x_{mi} & \sum x_{mi}x_{1i} & \sum x_{mi}x_{2i} & . & . & . & \sum x_{mi}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ . \\ . \\ . \\ a_m \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \\ . \\ . \\ . \\ \sum x_{mi}y_i \end{bmatrix}$$

El error estándar de la aproximación para regresión lineal múltiple se fórmula de la siguiente manera

$$S_{y/x_1,x_2,,,x_m} = \sqrt{\frac{S_r}{n-(m+1)}}$$

El coeficiente de correlación se calcula mediante la ecuación $r^2 = \dfrac{S_t - S_r}{S_t}$

Ejemplo 2.4

Use regresión lineal múltiple para ajustar:

x_1	0	1	2	0	1	2
x_2	2	2	4	4	6	6
y	19	12	11	24	22	15

calcule los coeficientes, el error estándar de la aproximación y el coeficiente de correlación.

Solución:

Resumiendo los cálculos tenemos:

	y	x_1	x_2	x_1^2	x_2^2	$x_i x_2$	$x_1 y$	$x_2 y$
	19	0	2	0	4	0	0	38
	12	1	2	1	4	2	12	24
	11	2	4	4	16	8	22	44
	24	0	4	0	16	0	0	96
	22	1	6	1	36	6	22	132
	15	2	6	4	36	12	30	90
$\sum$	103	6	24	10	112	28	86	424

Luego,

$n = 2$

$\sum y = 103$

$\sum x_1 = 6$

$\sum x_2 = 24$

$\sum x_1^2 = 10$

$\sum x_2^2 = 112$

$\sum x_1 x_2 = 28$

$\sum x_1 y = 86$

$\sum x_2 y = 424$

Por lo tanto las ecuaciones lineales simultáneas son:

$$6a_0 + 6a_1 + 24a_2 = 103$$
$$6a_0 + 10a_1 + 28a_2 = 86$$
$$24a_0 + 28a_1 + 112a_2 = 424$$

Resolviendo este sistema con la ayuda del programa de Eliminación Gaussiana con Sustitución hacia atrás que está en el Anexo No. 3. Se obtiene como solución del sistema

a_0 = 14.1667

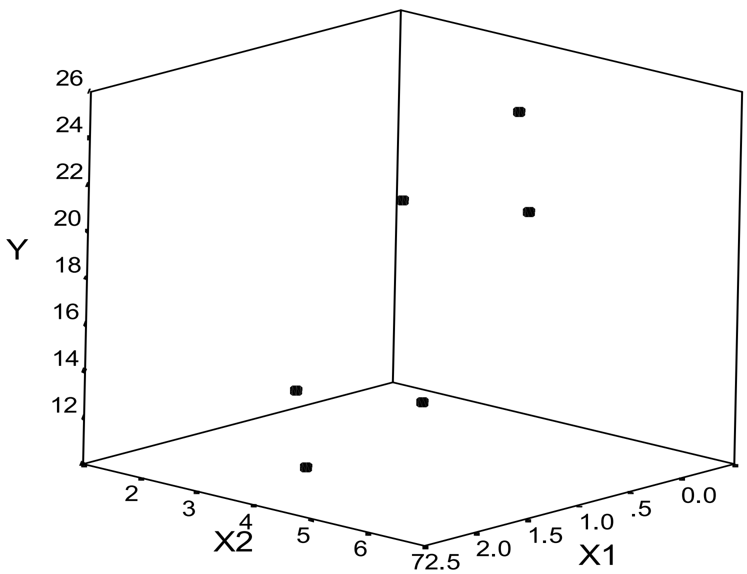
a_1=-6.66667

a_2= 2.41667

Y por lo tanto, la ecuación multilineal con mínimos cuadrados en este caso es:

$$y = 14.1667 - 6.66667x_1 + 2.41667x_2$$

En la figura siguiente se presenta el ajuste multilineal y los datos considerados.



Además

$$S_t = \sum_{i=1}^{6}\left(y_i - \overline{y}\right)^2 = 142.833$$

$$S_r = \sum_{i=1}^{6}\left(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i}\right)^2 = 0.5$$

El error estándar de la aproximación es:

$$S_{y/x_1,x_2,,,x_m} = \sqrt{\frac{S_r}{n-(m+1)}} = \sqrt{\frac{0.5}{6-(2+1)}} = 0.4082248$$

El coeficiente de correlación es:

$$r^2 = \frac{142.833 - 0.5}{142.833} = 0.996499$$

Estos resultados indican que el 99.6499% de la incertidumbre se ha explicado mediante el modelo; este resultado apoya que la ecuación multilineal representa un ajuste perfecto.

Para realizar los cálculos con una computadora se necesita disponer del algoritmo de regresión lineal múltiple.

ALGORITMO PARA LA REGRESIÓN LINEAL MÚLTIPLE

Paso 1: Introducir el número de variables a usar, m.

Paso 2: Introducir el número de puntos, n

Paso 3: Si n ≤ m, imprimir un mensaje de error de que la ecuación de regresión es imposible y termina el proceso. Si n > m, continuar

Paso 4: Calcular las sumas y productos de potencia.

Paso 5: Arreglar estas sumas y productos en forma de matriz aumentada.

Paso 6: Resolver la matriz aumentada en los coeficientes $a_0, \ldots, a_m$, usando un método de eliminación.

Paso 7: Imprimir los resultados.

El programa, en el lenguaje C, correspondiente a este algoritmo puede verse en el Anexo No 6.

2.2.4 Regresión no lineal

La regresión lineal proporciona una técnica muy poderosa para ajustar datos a una "mejor" línea. Sin embargo, se ha predicho que la relación entre variables dependientes e independientes es lineal. Esto no es siempre el caso, en otros casos se pueden hacer transformaciones que expresen los datos de manera que sean compatibles con la regresión lineal.

Un ejemplo de un modelo no lineal es la ecuación de potencia: $y = C\ x^{A}$ en donde C y A son constantes, un segundo ejemplo seria un modelo exponencial $y = Ce^{Ax}$ en donde C y A son constantes.

Supongamos que queremos ajustar una curva exponencial de la forma: $y = Ce^{Ax}$ a un conjunto de datos $(x_1, y_1),(x_2, y_2),\dots,(x_n, y_n)$ dado de antemano. El método no lineal de los mínimos cuadrados consiste en hallar el mínimo de la función.

$$S_r = \sum_{i=1}^{n}\left(y_i - Ce^{Ax_i}\right)^2$$

para ello hallamos las derivadas parciales de S_r respecto de C y A,

$$\frac{\partial S_r}{\partial A} = 2\sum_{i=1}^{n}\left(y_i - Ce^{Ax_i}\right)\left(-Cx_i e^{Ax_i}\right)$$

$$\frac{\partial S_r}{\partial C} = 2\sum_{i=1}^{n}\left(y_r - Ce^{Ax_i}\right)\left(-e^{Ax_i}\right)$$

Al igualar a cero las derivadas parciales anteriores obtenemos, después de simplificar, las correspondientes ecuaciones normales

$$C\sum_{i=1}^{n} x_i e^{2Ax_i} - \sum_{i=1}^{n} x_i y_i e^{Ax_i} = 0,$$
$$C\sum_{i=1}^{n} e^{2Ax_i} - \sum_{i=1}^{n} y_i e^{Ax_i} = 0$$

Como vemos se nos presenta un inconveniente con el método, ya que las ecuaciones anteriores no son lineales para las incógnitas A y C. Sin embargo, una alternativa más simple es la de usar manipulaciones matemáticas y transformar las ecuaciones a la forma lineal. A continuación trataremos la alternativa de linealización para el ajuste exponencial.

Ajuste exponencial

Supongamos que queremos ajustar una curva exponencial de la forma $y = Ce^{Ax}$ a un conjunto de puntos $(x_1, y_1),(x_2, y_2),...,(x_n, y_n)$. Para afrontar este problema, empezaremos tomando logaritmo de la ecuación $y = Ce^{Ax}$ que nos da $\ln(y) = Ax + \ln C$, luego hacemos un cambio de variable y uno de constante así

$Y = \ln y$ $\quad X = x$ y $B = \ln C$

Lo que obtenemos es una relación lineal entre las nuevas variables X e Y

$$Y = AX + B$$

Los datos originales (x_i, y_i) se han transformado, con el cambio de variable, en $(X_i, Y_i) = (x_i, \ln(y_i))$. A este proceso se le llama método de linealización de los datos. El problema ahora es calcular la recta de regresión $Y = AX + B$, para los puntos $\{(X_i, Y_i)\}$, para lo que planteamos las correspondientes ecuaciones de Gauss (ya estudiadas en 2.2.1).

$$\left(\sum_{i=1}^{n} X_i^2\right)A + \left(\sum_{i=1}^{n} X_i\right)B = \sum_{i=1}^{n} X_i Y_i$$

$$\left(\sum_{i=1}^{n} X_i\right)A + nB = \sum_{i=1}^{n} Y_i$$

De aquí que $A = \dfrac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum X_i^2 - \left(\sum X_i\right)^2}$ y $B = \overline{Y} + A\overline{X}$, en donde $\overline{Y}$ y $\overline{X}$ son la media de Y y X respectivamente. Para encontrar C se calcula e^B.

Ejemplo 2.5

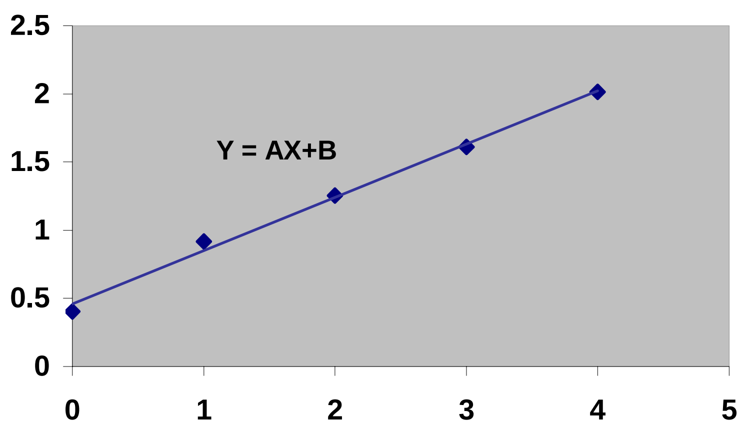
Usaremos el método de linealización de los datos para un ajuste exponencial $y = Ce^{ax}$ a los cinco datos $(0,1.5)$, $(1,2.5)$, $(2,3.5)$, $(3,5)$ y $(4,7.5)$.

Solución:

Los Cálculos de los coeficientes de las ecuaciones de Gauss para los datos linealizados $\{(X_i, Y_i)\}$ son:

x_i	y_i	X_i	$Y_i = \ln(y_i)$	X_i^2	$X_i Y_i$
0	1.5	0	0.405465	0	0
1	2.5	1	0.916291	1	0.916291
2	3.5	2	1.252763	4	2.505526
3	5	3	1.609438	9	4.828314
4	7.5	4	2.014903	16	8.059612
10		10	6.198860	30	16.309743

Gráfica de los datos linealizados $\{(X_i, Y_i)\}$



Los puntos transformados aparentan estar alineados, como se ve en la figura anterior. Ahora se encuentra la recta de regresión $Y = AX + B$ para los puntos calculados.

$$A = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum X_i^2 - \left(\sum X_i\right)^2}$$

$$A = \frac{5(16.309743) - (10)(6.198860)}{5(30) - (10)^2} = \frac{19.560115}{50} = 0.3912023$$

Ahora si, $\overline{X} = \frac{10}{5} = 2$ y $\overline{Y} = \frac{6.198860}{5} = 1.239772$

Encontremos $B = \overline{Y} + A\overline{X}$

$$B = 1.239772 - 2(0.3912023)$$
$$B = 0.4573674$$

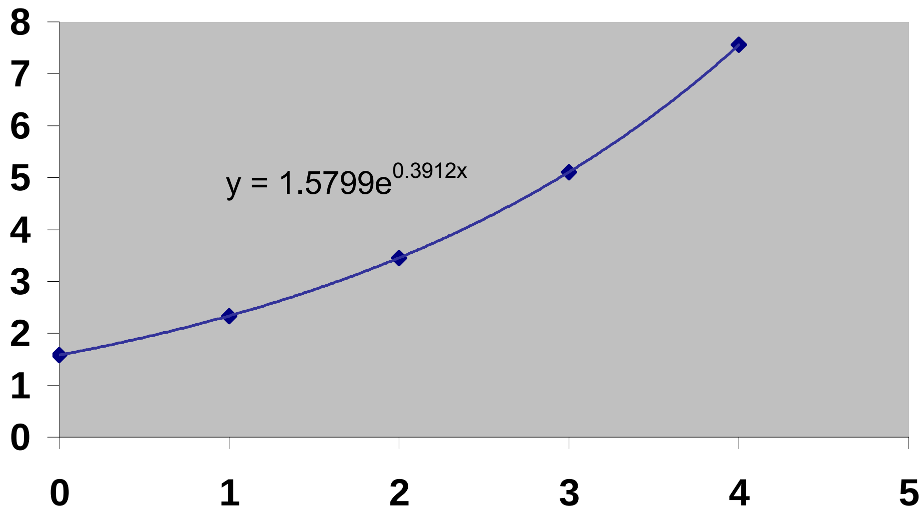
La recta de regresión lineal es:

$$Y = 0.391202X + 0.457367$$

Finalmente obtenemos el valor de C, que es $C = e^{0.457367} = 1.579910$ sustituyendo el valor de A y C en la ecuación $y = Ce^{Ax}$ obtenemos el ajuste exponencial.

$$y = 1.579910e^{0.3912023x}$$

Ajuste Exponencial obtenido con el Método de Linealización de los Datos



CAPÍTULO 3
APLICACIONES

3.1 Introducción

El propósito de este capitulo es el hacer uso de los métodos de Regresión estudiados, en el capítulo dos, en la solución de problemas aplicados a la Física, Ingeniería y otras ciencias, en contraste con los métodos de Interpolación abordados en el capítulo uno. Se tratarán a continuación, tres casos.

3.2 CASO1: MODELO PARA DERIVAR UNA ECUACIÓN QUE PREDIGA LA RELACIÓN ENTRE FUERZA DE FRICCIÓN Y VELOCIDAD

En física se define como rozamiento o fricción a la resistencia que se opone al deslizamiento de un cuerpo sobre otro, o también la fuerza que aparece en la superficie de contacto de dos cuerpos cuando se intenta deslizar uno sobre el otro.

La fuerza de fricción es siempre de sentido contrario a la fuerza que empuja al cuerpo, su valor es menor o igual al de la fuerza que empuja al otro cuerpo.

Se está llevando a cabo un estudio para determinar la relación entre la fuerza de fricción que actúa hacia arriba y la velocidad de caída de un paracaidista. Para ello se realizan algunos experimentos. Si en uno de estos experimentos se obtiene la siguiente información sobre la velocidad (v medida en centímetros por segundos) y la fuerza de fricción (f_v medida en 10^6 dinas).

v	1000	2000	3000	4000	5000
f_v	5	15.3	29.3	46.4	66.3

Usaremos Regresión Lineal y Polinomial en contraste con interpolación para determinar la relación entre la fuerza de fricción y la velocidad. Además se calculará la fuerza de fricción para una velocidad de 2500 cm/seg.

REGRESIÓN LINEAL

Ajustando una línea recta a los datos de la tabla dada:

v	1000	2000	3000	4000	5000
f_v	5	15.3	29.3	46.4	66.3

Para ello haremos los cálculos correspondientes usando el programa en C ubicado en el Anexo 4, obteniendo los resultados siguientes:

La recta de regresión es f_v = -13.65 + 0.01537 v

La desviación estándar total es 24.4845

El error estándar de la aproximación es 3.44495

El coeficiente de correlación es 0.985153

El coeficiente de determinación es 0.992549

Por lo tanto, para una velocidad de 2500 cm/seg tenemos una fuerza de fricción f_v = -13.65 + 0.01537(2500)= 24.775.

REGRESION POLINOMIAL

Con la ayuda del programa en C del Anexo 5, se obtiene los resultados siguientes:

La matriz ampliada del sistema formado es

5	15000	5.5e+007	162.3
15000	5.5e+007	2.25e+011	640600
5.5e+007	2.25e+011	9.79e+014	2.7298e+009

Los coeficientes del polinomio son

a[0]=-2.49989
a[1]=0.00581277
a[2]=1.59287e-006

Por lo tanto, el ajuste con mínimos cuadrados es:

$$f_v = -2.49989 + 0.00581277v + 1.59287 \times 10^{-6} v^2$$

Además:
El error estándar de la aproximación es 0.202836
El coeficiente de correlación es 0.999966
El coeficiente de determinación es 0.999983

Por lo tanto, para una velocidad de 2500 cm/seg tenemos una fuerza de fricción $f_v = -2.49989 + 0.00581277(2500) + 1.59287 \times 10^{-6}(2500)^2 =$ 21.9874725.

INTERPOLACIÓN

Tabla de Aproximaciones

x	dif.div0	dif.div1	dif.div2	dif.div3	dif.div4
1000	5.000e+000				
2000	1.530e+001	1.030e-002			
3000	2.930e+001	1.400e-002	1.850e-006		
4000	4.640e+001	1.710e-002	1.550e-006	-1.000e-010	
5000	6.630e+001	1.990e-002	1.400e-006	-5.000e-011	1.250e-014

Los polinomios de interpolación de Newton de grado 0 al 4 evaluados en 2500

son:

$$P_0(2500) = 5$$

$$P_1(2500) = 5 + 1.030 \times 10^{-2}(2500 - 1000) = 20.45$$

$$P_2(2500) = P_1 + 1.850 \times 10^{-6}(2500 - 1000)(2500 - 2000) = 21.8375$$

$$P_3(2500) = P_2 - 1 \times 10^{-10}(2500 - 1000)(2500 - 2000)(2500 - 3000) = 21.875$$

$$P_4(2500) = P_3 + 1.250 \times 10^{-14}(2500 - 1000)(2500 - 2000)(2500 - 3000)(2500 - 4000) = 21.88203125$$

Resumiendo los resultados:

Grado del Polinomio n	*Fuerza de fricción con velocidad de* 2500
0	5
1	20.45
2	21.8375
3	21.875
4	21.88203125

En base al análisis, se toma la aproximación dada por el polinomio de grado 4, 21.88203125; que es una aproximación razonable para una velocidad de 2500 cm/seg.

De esta manera, los dos tipos de Regresión y la Interpolación llevan a resultados diferentes de la fuerza de fricción. Debido al realismo físico y al comportamiento más satisfactorio a través del rango completo de los datos, se optará por la Regresión Polinomial ya que proporciona mejores predicciones.

3.3 CASO 2: AJUSTE DE CURVAS EN EL DISEÑO DE UN MÁSTIL PARA BARCO (INGENIERÍA CIVIL)

El mástil de un barco tiene un área transversal de 0.876 pulg^2 y se construye de una aleación de aluminio experimental. Se llevan a cabo pruebas para definir la relación entre esfuerzo (fuerza por área) aplicado al material y deformación (deflexión por unidad de longitud) los resultados de estas pruebas se muestran en el cuadro siguiente:

$Esfuerzo\ lib/pu\lg^2$	$Deformación\ pies/pie$
1800	0.0005
5200	0.0013
7200	0.0020
7500	0.0045
8000	0.0060
10000	0.0085

También es necesario calcular el cambio de longitud del mástil debido a la deformación causada por la fuerza del viento. La compresión causada por el aire se puede calcular usando la relación:

$$Esfuerzo = \frac{fuerza\ \ del\ \ mástil}{área\ de\ la\ \sec ción\ transversal\ \ del\ \ mástil}$$

En este caso, se tiene una fuerza transversal del viento de 6745.2 libras, y el esfuerzo se calcula mediante:

$$Esfuerzo = \frac{6745.2}{0.876} = 7700\ lb/pu\lg^2$$

Este esfuerzo se puede usar para calcular la deformación, el cual, a su vez, se puede sustituir en la ley de Hooke y calcular el cambio en la longitud del mástil:

$$\Delta L = (deformación)(longitud)$$

en donde la longitud se refiere a la altura del mástil. Por lo tanto, el problema se reduce a la determinación de valores de la deformación de los datos de la tabla anterior. Ya que no se dispone de ningún punto para un valor de esfuerzo de 7700, el problema necesitará algún ajuste de curvas. En este caso se usarán dos planteamientos: el de Interpolación y el de Regresión con mínimos cuadrados lineal, no lineal y polinomial.

Solución

En el primer planteamiento se aplicará la interpolación polinomial de Newton haciendo uso del programa en C del Anexo 1, para obtener los polinomios de grado 0 al 5, y calcular la deformación a un esfuerzo de $7700\ lb/pu\lg^2$.

Tabla de Aproximaciones

x	dif.div0	dif.div1	dif.div2	dif.div3	dif.div4	dif.div5
1800	5.000e-004					
5200	1.300e-003	2.353e-007				
7200	2.000e-003	3.500e-007	2.124e-011			
7500	4.500e-003	8.333e-006	3.471e-009	6.052e-013		
8000	6.000e-003	3.000e-006	-6.667e-009	-3.621e-012	-6.816e-016	
10000	8.500e-003	1.250e-006	-7.000e-010	2.131e-012	1.198e-015	2.292e-019

Los polinomios de interpolación de Newton de grado 0 al 5 evaluados en 7700 son:

$$P_0(7700) = 5\times10^{-4}$$

$$P_1(7700) = 5\times10^{-4} + 2.353\times10^{-7}(7700-1800) = 0.00188827$$

$$P_2(7700) = P_1 + 2.124\times10^{-11}(7700-1800)(7700-5200) = 0.00220156$$

$$P_3(7700) = P_2 + 6.052\times10^{-13}(7700-1800)(7700-5200)(7700-7200) = 0.00666491$$

$$\begin{aligned} P_4(7700) &= P_3 - 6.816\times10^{-16}(7700-1800)(7700-5200)(7700-7200)(7700-7500) \\ &= 0.00565955 \end{aligned}$$

$$\begin{aligned} P_5(7700) = P_4 &+ 2.292\times10^{-19}(7700-1800)(7700-5200)(7700-7200)(7700-7500) \\ &(7700-8000) = 0.005558129 \end{aligned}$$

Resumiendo los resultados:

Grado del Polinomio n	*Deformación con esfuerzo de* 7700
0	0.0005
1	0.00188827
2	0.00220156
3	0.00666491
4	0.00565955
5	0.005558129

En base al análisis, se toma la aproximación dada por el polinomio de grado 5, 0.005558129 pies/ pie; que es una aproximación razonable de la deformación para un esfuerzo de $7700\ lb/pulg^2$.

La Regresión proporciona una alternativa, usaremos a continuación Regresión Lineal, no Lineal y Polinomial para llevar acabo los mismos cálculos.

REGRESIÓN LINEAL

Haciendo uso del programa en C del Anexo 4 obtenemos los siguientes resultados:

La recta de regresión es D = -0.00252686 + 9.56201e-007E

La desviación estándar total es 0.00309193

El error estándar de la aproximación es 0.00169648

El coeficiente de correlación es 0.759159

El coeficiente de determinación es 0.871297

Por lo tanto, para un Esfuerzo de $7700\, lb/pu\lg^2$ tenemos una Deformación D = -0.00252686 + 9.56201e-007(7700)= 0.0048358877 pies/pie.

REGRESIÓN NO LINEAL

Haciendo uso del programa en C del Anexo 7 obtenemos los siguientes resultados:

A = 0.000363308 B =-8.36193

La recta de regresión es D = 0.000363308 E - 8.36193

El valor de C = 0.000233593 y A = 0.000363308

Se tiene que D = 0.000233593 exp (0.000363308E)

Por lo tanto, para un Esfuerzo de $7700\, lb/pu\lg^2$ tenemos una Deformación D=0.000233593 exp (0.000363308(7700))=0.003831654161 pies/pie.

REGRESIÓN POLINOMIAL

Haciendo uso del programa en C del Anexo 5 obtenemos los siguientes resultados:

La matriz ampliada del sistema formado es

6	39700	3.0237e+008	0.0228
39700	3.0237e+008	2.45356e+012	188.81
3.0237e+008	2.45356e+012	2.06891e+016	1.62758e+006

Los coeficientes del polinomio son

a[0]=0.00150697

a[1]=-8.77109e-007

a[2]=1.60662e-010

Y por lo tanto, la ecuación cuadrática con mínimos cuadrados en este caso es:

$$D = 0.00150697 - 8.77109 \times 10^{-7} E + 1.60662 \times 10^{-10} E^2$$

El error estándar de la aproximación es 0.00118294

El coeficiente de correlación es 0.912175

El coeficiente de determinación es 0.955079

Por lo tanto para un Esfuerzo de $7700\, lb/pu\lg^2$ tenemos una Deformación

$$D = 0.00150697 - 8.77109 \times 10^{-7}(7700) + 1.60662 \times 10^{-10}(7700)^2 = 0.0042788806\,8\; pies/pie$$

De esta manera, la interpolación y los tres tipos de Regresión llevan a resultados diferentes de la Deformación. Debido a como están distribuidos los datos dados, se optará por la regresión lineal ya que proporciona una mejor predicción para el caso particular de un esfuerzo de $7700\, lb/pu\lg^2$, aunque el coeficiente de correlación sea menor que el de la regresión polinomial .

Usando el valor de la longitud =30 pies y con la ecuación $\Delta L = (deformación)(longitud)$ se obtienen el siguiente resultado del cambio en la longitud del mástil:

$$\Delta L = (0.0048358877\ pies/pie)(30\,pies) = 0.1451\,pies$$

3.4 CASO 3: MODELO PARA DERIVAR UNA ECUACION PARA PREDECIR LA CAPACIDAD CALORIFICA (k) EN FUNCION DE LA TEMPERATURA (T).

En Química la capacidad calorífica es la cantidad de energía necesaria para aumentar 1grado kelvin la temperatura de un metal, ésta varía según el metal.

Se llevaron cabo varios experimentos y se determinaron los siguientes valores de capacidad calorífica (k) a varias temperaturas (T) para un metal dado. Se quiere obtener un modelo para predecir k para una temperatura de T = 85.

T	-50	-20	10	70	100	120
k	0.125	0.128	0.134	0.144	0.150	0.155

Ya que no se dispone de ningún punto para un valor de temperatura dado de 85, el problema necesitará algún ajuste de curvas. En este caso se usarán dos planteamientos: el de Regresión con mínimos cuadrados e Interpolación.

REGRESIÓN LINEAL

Haciendo uso del programa en C del Anexo 4 obtenemos los siguientes resultados:

La recta de regresión es k = 0.132537 + 0.000177289T

La desviación estándar total es 0.0121929

El error estándar de la aproximación es 0.00114244

El coeficiente de correlación es 0.992977

El coeficiente de determinación es 0.996482

Por lo tanto, para una temperatura de 85 tenemos una capacidad calorífica
k = 0.132537 + 0.000177289(85)=0.147606565.

REGRESIÓN NO LINEAL

Haciendo uso del programa en C del Anexo 8 obtenemos los siguientes resultados:

A = 0.00127486 B =-2.02295

La recta de regresión es C = 0.00127486 T -2.02295

El valor de C = 0.132265 y A = 0.00127486

Se tiene que k = 0.132265 exp(0.00127486T)

Por lo tanto, para una temperatura de 85 tenemos una capacidad calorífica
k = 0.132265 exp(0.00127486(85))= 0.147403037

REGRESIÓN POLINOMIAL

Haciendo uso del programa en C del Anexo 5 obtenemos los siguientes resultados:

La matriz ampliada del sistema formado es

6 230 32300 0.836

230 32300 2.939e+006 36.21

32300 2.939e+006 3.3779e+008 4814.7

Los coeficientes del polinomio son

a[0]=0.131725

a[1]=0.00015471

a[2]=3.11743e-007

Y por lo tanto, la ecuación cuadrática con mínimos cuadrados en este caso es:

$$k = 0.131725 + 0.00015471\,T + 3.11743 \times 10^{-7} T^2$$

El error estándar de la aproximación es 0.00064893

El coeficiente de correlación es 0.9983

El coeficiente de determinación es 0.99915

Por lo tanto, para una temperatura de 85 tenemos una capacidad calorífica $k = 0.131725 + 0.00015471(85) + 3.11743 \times 10^{-7}(85)^2 =$ 0.147131093

INTERPOLACIÓN

Haciendo uso del programa en C del Anexo 2, Interpolación de Neville, obtenemos los siguientes resultados:

Tabla de Aproximaciones

x	pol.gr0	pol.gr1	pol.gr2	pol.gr3	pol.gr4	pol.gr5
-50.000000	0.125000					
-20.000000	0.128000	0.138500				
10.000000	0.134000	0.149000	0.162125			
70.000000	0.144000	0.146500	0.146083	0.144078		
100.000000	0.150000	0.147000	0.146917	0.146812	0.146539	
120.000000	0.155000	0.146250	0.146775	0.146820	0.146818	0.146761

La aproximación dada por el polinomio de grado 5, 0.146761, es una aproximación razonable para una temperatura de 85.

De esta manera, los tres tipos de regresión con la interpolación llevan a resultados diferentes de la Capacidad Calorífica. Debido al comportamiento más satisfactorio a través del rango completo de los datos, se optará por la Regresión Polinomial ya que proporciona una mejor predicción.

3.5 CASO 4: MODELO PARA DERIVAR UNA ECUACION QUE PREDIGA EL PESO FINAL DE UN ANIMAL EN FUNCIÓN DE SU PESO INICIAL Y DE LA CANTIDAD DE ALIMENTO QUE RECIBE.

Un Instituto de Investigación desea determinar si es posible pronosticar el peso de un animal después de un período de tiempo determinado sobre la base de su peso inicial y de la cantidad de alimento que recibe. Se llevaran a cabo experimentos y se registraron los siguientes datos en kilogramos:

Peso final	*Peso Inicial*	*Alimento Consumido*
y	x_1	x_2
95	42	272
77	33	226
80	33	259
100	45	292
97	39	311
70	36	183
50	32	173
80	41	236
92	40	230
84	38	235

Se quiere derivar una ecuación que prediga el peso final de un animal en función del peso inicial y de la cantidad de alimento recibido. En este caso, la ecuación se adapta a la regresión múltiple.

Haciendo uso del programa en C del Anexo 6, obtenemos los siguientes resultados:

La matriz ampliada del sistema formado es

10 379 2417 825

379 14533 92628 31726

2417 92628 601365 204569

Los coeficientes del polinomio son

a[0]=-22.9932

a[1]=1.39568

a[2]=0.217613

Y por lo tanto, la ecuación multilineal con mínimos cuadrados en este caso es:

$$y = -22.9932 + 1.39568x_1 + 0.217613x_2$$

El peso final de un animal que tiene un peso inicial de 35 kilogramos y que recibe 250 kilogramos de alimento es de

$$y = -22.9932 + 1.39568(35) + 0.217613(250) = 80.37 \text{ kilogramos}$$

El empleo de la Interpolación de Neville, en este caso, para los datos de las varibles x_1 y y no es posible porque hay datos iguales para x_1. La Interpolación de Neville para los datos de las variables x_2 y y da como resultado un peso final de 38.44 kilogramos para una cantidad de 250 kilogramos de alimentos consumido, que dista mucho del resultado obtenido en la regresión múltiple, 80.37 kilogramos que es muy satisfactorio.

CONCLUSIONES

- Existe una variedad de maneras diferentes de expresar un polinomio de interpolación: El polinomio de interpolación con diferencias divididas de Newton, el polinomio de interpolación de Lagrange y el polinomio de interpolación segmentaria.

- El polinomio de interpolación de diferencias divididas de Newton se adapta idealmente en aquellos casos en que el grado del polinomio a utilizar en la aproximación se desconoce El polinomio de interpolación de Lagrange es apropiado cuando el grado del polinomio de aproximación se conoce a priori. El polinomio de interpolación segmentaria resulta satisfactorio para un conjunto numeroso de n +1 datos $\{(x_k, y_k)\}_{k=0}^{n}$.

- La regresión por mínimos cuadrados se emplea para desarrollar la “mejor” curva que ajuste todas las tendencias de los datos sin pasar necesariamente a través de algún punto. El ajuste se hace minimizando la suma de los cuadrados de las diferencias entre los datos y la función.

- La regresión por mínimos cuadrados puede ser lineal, polinomial, lineal múltiple y no lineal. No se puede hablar de un método particular que sea el óptimo, porque la selección del mejor método para realizar la aproximación depende del comportamiento de los datos involucrados en el análisis.

- La aproximación por mínimos cuadrados resulta ser mejor que la aproximación usando interpolación.

ANEXOS

ANEXO No. 1

PROGAMA POLINOMIO INTEPOLADOR DE NEWTON

```
/* Programa para obtener los coeficientes */
/* del polinomio de interpolación  de Newton dados n+1 puntos*/
#include<stdio.h>
#include<stdlib.h>
main()
{       int n,i,j,k;
        float x[10],y[10],D[10][10];
        system("cls");
        printf("\n Introduzca el valor de n: ");
        scanf("%d",&n);
        printf("\n Introduzca los puntos (x,y) ");
        for(k=0;k<=n;k++)
             {
               printf("\n x[%d]:",k);
               scanf("%f",&x[k]);
               printf("\n y[%d]:",k);
               scanf("%f",&y[k]);
               D[k][0]= y[k];
               }
        for(k=1;k<=n;k++)
             for(i=1;i<=k;i++)
               D[k][i]= (D[k][i-1]-D[k-1][i-1]) /( x[k]-x[k-i]);
        printf("\n Tabla de Aproximaciones ");
        printf("\n");
        printf("\n  x    ");
        for(i=0;i<=n;i++)
           printf("  dif.div%d",i);
           printf("\n");
           for(i=0;i<=n;i++)
                {
                 printf("%g ",x[i]);
                 for(j=0;j<=i;j++)
                    printf(" %.3e",D[i][j]);
                 printf("\n");
                }
        printf("\n Los coeficientes del polinomio interpolador de Newton");
        printf("\n de grado n=%d son:",n);
        for(k=0;k<=n;k++)
             printf("\n a[%d]=%.3e",k,D[k][k]);
}
```

SALIDA DEL EJEMPLO 1.1

Tabla de Aproximaciones

x	dif.div0	dif.div1	dif.div2	dif.div3	dif.div4
4	2.000e+000				
5	2.236e+000	2.361e-001			
6	2.449e+000	2.134e-001	-1.132e-002		
7	2.646e+000	1.963e-001	-8.580e-003	9.149e-004	
8	2.828e+000	1.827e-001	-6.790e-003	5.967e-004	-7.956e-005

Los coeficientes del polinomio interpolador de Newton de grado n=4 son:

a[0]=2.000e+000
a[1]=2.361e-001
a[2]=-1.132e-002
a[3]=9.149e-004
a[4]=-7.956e-005

Elapsed time = 00:00:51.30. Program returned (4). Press any key.

ANEXO No. 2

PROGAMA DE NEVILLE

```
/*Programa  para evaluar  en el punto a, el polinomio  interpolador P */
 /* Usando el método de Neville*/
 # include <stdio.h>
# include <stdlib.h>
# include <math.h>
 main ( )
 {
 int  i,j,n,m,k;
 float  a, x[10], y[10], Q[10][10];
 system ("cls");
 printf("\n Introduzca  el número de puntos a usar: ");
 scanf ("%d",&m);
 printf("\n Introduzca  el valor a donde  se va a evaluar  el polinomio: ");
 scanf ("%f",&a);
 printf("\n Introduzca  los puntos (x,y) a usar: ");
  n=m-1;
 for (k=0;k<=n;k++)
   {
   printf("\n x[%d]=",k);
   scanf ("%f",&x[k]);
   printf("\n y[%d] =",k);
   scanf ("%f",&y[k]);
   Q[k][0]=y[k];
   }
for (i=1;i<=n;i++)
   {
  for (j=1;j<=i;j++)
    Q[i][j]= ((a- x[i-j])*Q[i][j-1]-(a -x[i])*Q[i-1][j-1])/(x[i]-x[i-j]);
   }
  printf("\n Tabla de Aproximaciones ");
  printf("\n");
  printf("\n x     ");
  for(i=0;i<=n;i++)
             printf(" pol.gr %d      ",i);
  printf("\n");
  for(i=0;i<=n;i++)
         {
          printf("%f ",x[i]);
          for(j=0;j<=i;j++)
             printf("   %f",Q[i][j]);
          printf("\n");
         }
}
```

SALIDA DEL EJEMPLO 1.4

Tabla de Aproximaciones

X	pol.gr0	pol.gr1	pol.gr2	pol.gr3	pol.gr4	pol.gr5
0.000000	1.000000					
0.500000	2.119000	4.580800				
1.000000	2.910000	3.859200	3.426240			
1.500000	3.945000	4.152000	4.181280	4.231616		
2.000000	5.720000	4.300000	4.240800	4.224928	4.226265	
2.500000	8.695000	3.340000	4.204000	4.226080	4.225562	4.225815

Elapsed time = 00:00:44.93. Program returned (5). Press any key.

ANEXO No. 3

PROGRAMA DE ELIMINACIÓN GAUSSIANA

```
/*Programa de Eliminación Gaussiana con sustitución hacia atrás*/
/*para resolver un sistema de n ecuaciones con n variables*/
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
main()
{
int n,k,i,j,p;
float a[11][11],x[11],m[11][11],y,sum;
  system("cls");
  printf("\n introduzca el numero de ecuaciones y de variables n:");
  scanf("%d",&n);
  printf("\n Introduzca la matriz ampliada fila a fila");
  for(i=1;i<=n;++i)
     for(j=1;j<=n+1;++j)
           {
           printf("\n a[%d][%d]=",i,j);
           scanf("%f",&a[i][j]);
           }
     for(i=1;i<=n-1;++i)
            {
               if(a[i][i]==0)
                    {
               for(p =i+1;p<= n;++p)
                    {
                      if(a[p][i]!=0)
                       {
```

```
            for(k=1;k<=n+1;++k)
                  {
                    y=a[i][k];
                    a[i][k]=a[p][k];
                    a[p][k]=y;
                    }
            goto seguir;
          }
        }
         printf("\n No existe solucion unica para el sistema" );
         exit(0);
    }
 seguir: for(j=i+1;j<=n;++j)
          {
          m[j][i]=a[j][i]/a[i][i];
         for(k=i;k<=n+1;++k)
                 a[j][k]=a[j][k]- m[j][i]*a[i][k];
             }
}
 if(a[n][n]==0)
             {
                printf("\n No existe solucion unica para el sistema" );
                exit(0);
                }
 x[n]=a[n][n+1]/a[n][n];
 for (i=n-1;i>=1;--i)
   {
       sum=0;
       for(j=i+1;j<=n;++j)
            sum=sum+a[i][j]*x[j];
       x[i]=(a[i][n+1]-sum)/a[i][i];
```

```
        }
        printf("\n La solución del sistema es ");
        for(i=1;i<=n;++i)
            printf("\n x[%d]=%g",i,x[i] );
}
```

SALIDA PARA EL EJEMPLO 1.6

```
La solución del sistema es
 x[1]=1.24802e-006
 x[2]=-1.00001
 x[3]=5.50002
 x[4]=0.639999
 x[5]=-6.75999
 x[6]=18.46
 x[7]=-1.6
 x[8]=24.6
 x[9]=-91.3
Elapsed time = 00:02:59.55.  Program returned (9).  Press any key.
```

ANEXO No. 4

PROGRAMA DE REGRESIÓN LINEAL

```
/*Programa que calcula la recta de regresión  lineal por mínimos cuadrados*/
/*Calcula además una estimación de los errores */
# include <stdio.h>
# include <stdlib.h>
# include <math.h>
main()
{
int n,i;
float x[10],y[10],sx=0,sy=0,sxy=0,sx2=0,mx,my,A0,A1,dy,dyx,r,rr,St=0,Sr=0;
system("cls");
printf("Introduzca el numero de puntos a usar:");
scanf("%d",&n);
for (i=1;i<=n;i++)
        { printf("\n x[%d]=",i);
          scanf("%f",&x[i]);
          printf("y[%d] =", i);
          scanf("%f",&y[i]);
        }
for(i=1;i<=n;i++)

{
  sx=sx+x[i];
  sy=sy+y[i];
  sx2=sx2+x[i]*x[i];
  sxy=sxy+x[i]*y[i];

}
```

```
  mx=sx/n;
  my=sy/n;
  A1=(n*sxy-sx*sy)/(n*sx2-(sx*sx));
  A0=my-A1*mx;

for(i=1;i<=n;i++)

{
   St=St + (y[i]-my)*(y[i]-my);
   Sr=Sr + (y[i]-A0-A1*x[i])*(y[i]-A0-A1*x[i]);

}
dy=sqrt(St/(n-1));
dyx=sqrt(Sr/(n-2));
rr=(St-Sr)/St;
r=sqrt(rr);
printf("\n La recta de regresión es y = %g + % gx",A0,A1);
printf("\n La desviación estándar total es %g",dy);
printf("\n El error estándar de la aproximación es %g",dyx);
printf("\n El coeficiente de correlación es %g",rr);
printf("\n El coeficiente de determinación es %g",r);

}
```

SALIDA PARA LOS EJEMPLOS 2.1 Y 2.2

La recta de regresión es y= 0.0714285 + 0.839286x

La desviación estándar total es 1.94569

El error estándar de la aproximación es 0.773443

El coeficiente de correlación es 0.868318

El coeficiente de determinación es 0.931836

Elapsed time = 00:00:23.18. Program returned (45). Press any key.

ANEXO No. 5

PROGRAMA DE REGRESIÓN POLINOMIAL

```
/*Programa que calcula los coeficientes de una regresión polinomial */
/*Calcula además una estimación de los errores*/
 #include <stdio.h>
 #include<math.h>
 #include<stdlib.h>
main()
{
int m,n,k,i,j,p,t;
float a[10][10],co[10],mu[10][10],aux,sum;
float  x[10],y[10],sx[10],sxy[10],sy=0,dyx=0,Sr=0,St=0,rr,r,pol,my,z;
system("cls");
printf("Introduzca el numero de puntos a usar:");
scanf("%d",&n);
for (i=1;i<=n;i++)
               { printf("\n x[%d]=",i);
                 scanf("%f",&x[i]);
                 printf("y[%d] =", i);
                 scanf("%f",&y[i]);
               }
  printf("\n introduzca el grado del polinomio a usar m:");
  scanf("%d",&m);

  for(j=1;j<=2*m;j++)
  {sx[j]=0;
   for(i=1;i<=n;i++)
   sx[j]=sx[j]+pow(x[i],j);
 }
  for(i=1;i<=n;i++)
   sy=sy+y[i];
for(j=1;j<=m;j++)
   {
   sxy[j]=0;
   for(i=1;i<=n;i++)
   sxy[j]=sxy[j]+pow(x[i],j)*y[i];
   }

  a[1][1]=n;
  for(j=2;j<=m+1;j++)
     a[1][j]=sx[j-1];
  a[1][m+2]=sy;
```

```
t=0;
for(i=2;i<=m+1;++i)
  {
    for(j=1;j<=m+1;++j)
       a[i][j]=sx[j+t];
       t=t+1;
       a[i][m+2]=sxy[i-1];

       }
           printf("\n La matriz ampliada del sistema formado es \n ");

           for(i=1;i<=m+1;++i)
                    {
                      for(j=1;j<=m+2;j++)
                          printf("%g  ",a[i][j]);
                          printf("\n ");

                          }
   m=m+1;
  for(i=1;i<=m-1;++i)

          {
            if(a[i][i]==0)

                {

            for(p=i+1;p<=m;++p)

                {
                  if(a[p][i]!=0)
                   {
                     for(k=1;k<=m+1;++k)
                            {
                              aux=a[i][k];
                              a[i][k]=a[p][k];
                              a[p][k]=aux;
                              }

                     goto seguir;
                   }
                }

                 printf("\n No existe solución única para el sistema" );
                 exit(0);

            }
```

```
seguir: for(j=i+1;j<=m;++j)

         {
         mu[j][i]=a[j][i]/a[i][i];

            for(k=i;k<=m+1;++k)

                 a[j][k]=a[j][k]- mu[j][i]*a[i][k];
             }
        }


if(a[m][m]==0)

             {
                printf("\n No existe solución única para el sistema" );
                exit(0);
                }


co[m]=a[m][m+1]/a[m][m];

for (i=m-1;i>=1;--i)
  {

       sum=0;
       for(j=i+1;j<=m;++j)
            sum=sum+a[i][j]*co[j];
       co[i]=(a[i][m+1]-sum)/a[i][i];
  }
 printf("\n Los coeficientes del polinomio son ");

 for(i=0;i<=m-1;++i)
        printf("\n a[%d]=%g",i,co[i+1] );
 my=sy/n;
 for(i=1;i<=n;i++)

  {
       St=St + (y[i]-my)*(y[i]-my);
       pol= co[1];
       for(k=2;k<=m+1;k++)
          pol=pol + co[k]* pow(x[i],k-1);

       Sr=Sr + (y[i]-pol)*(y[i]-pol);

  }
m=m-1;
```

```
dyx=sqrt(Sr/(n-(m+1)));
rr=(St-Sr)/St;
r=sqrt(rr);
printf("\n El error estándar de la aproximación es %g",dyx);
printf("\n El coeficiente de correlación es %g",rr);
printf("\n El coeficiente de determinación es %g",r);



}
```

SALIDA PARA EL EJEMPLO 2.3

```
La matriz ampliada del sistema formado es
 6  15  55  152.6
 15  55  225  585.6
 55  225  979  2488.8

 Los coeficientes del polinomio son
 a[0]=2.47859
 a[1]=2.35926
 a[2]=1.86072

 El error estándar de la aproximación es 1.11752
 El coeficiente de correlación es 0.998509
 El coeficiente de determinación es 0.999254
Elapsed time = 00:00:25.98.  Program returned (45).  Press any key.
```

ANEXO No. 6

PROGRAMA DE REGRESIÓN LINEAL MÚLTIPLE

```
/*programa que evalúa una regresión lineal múltiple */
# include <stdio.h>
# include <math.h>
# include <stdlib.h>
main()
{
int n,i,j,k,m,p;
float x1[10],x2[10],y[10],sx1=0,sx2=0,sy=0,sx1y=0;
float sx2y=0,sx1x1=0,sx2x2=0,sx1x2=0,a[10][10],mu[10][10],co[10];
float a0,a1,a2,aux,sum,dyx=0,Sr=0,St=0,rr,r,pol,my,z; ;
system("cls");
printf("Introduzca el numero de puntos a usar:");
scanf("%d",&n);
for (i=1;i<=n;i++)
            { printf("\n x1[%d]=",i);
              scanf("%f",&x1[i]);
              printf("\n x2[%d]=",i);
              scanf("%f",&x2[i]);
              printf("y[%d] =", i);
              scanf("%f",&y[i]);
            }
for(i=1;i<=n;i++)

{
  sx1=sx1+x1[i];
  sx2=sx2+x2[i];
  sy=sy+y[i];
  sx1x1=sx1x1+x1[i]*x1[i];
  sx2x2=sx2x2+x2[i]*x2[i];
  sx1x2=sx1x2+x1[i]*x2[i];
  sx1y=sx1y+x1[i]*y[i];
  sx2y=sx2y+x2[i]*y[i];

 }
  a[1][1]=n;
  a[1][2]=sx1;
  a[1][3]=sx2;
  a[1][4]=sy;
  a[2][1]=sx1;
  a[2][2]=sx1x1;
```

```
a[2][3]=sx1x2;
a[2][4]=sx1y;
a[3][1]=sx2;
a[3][2]=sx1x2;
a[3][3]=sx2x2;
a[3][4]=sx2y;

        printf("\n La matriz ampliada del sistema formado es \n ");

          for(i=1;i<=3;++i)
                  {
                   for(j=1;j<=4;j++)
                       printf("%g  ",a[i][j]);
                       printf("\n ");

                       }
   m=3;
  for(i=1;i<=m-1;++i)

         {
           if(a[i][i]==0)

               {

           for(p=i+1;p<=m;++p)

               {
                 if(a[p][i]!=0)
                  {
                    for(k=1;k<=m+1;++k)
                          {
                           aux=a[i][k];
                           a[i][k]=a[p][k];
                           a[p][k]=aux;
                           }

                    goto seguir;
                  }
               }

                printf("\n No existe solución única para el sistema" );
                exit(0);

           }

         seguir: for(j=i+1;j<=m;++j)
```

```
        {
        mu[j][i]=a[j][i]/a[i][i];

          for(k=i;k<=m+1;++k)

               a[j][k]=a[j][k]- mu[j][i]*a[i][k];
           }
       }


if(a[m][m]==0)

           {
              printf("\n No existe solución única para el sistema" );
              exit(0);
              }


co[m]=a[m][m+1]/a[m][m];

for (i=m-1;i>=1;--i)
  {

      sum=0;
      for(j=i+1;j<=m;++j)
           sum=sum+a[i][j]*co[j];
      co[i]=(a[i][m+1]-sum)/a[i][i];
  }
 printf("\n Los coeficientes del polinomio son ");

 for(i=0;i<=m-1;++i)
        printf("\n a[%d]=%g",i,co[i+1] );
 my=sy/n;
 m=m-1;
 for(i=1;i<=n;i++)

  {
      St=St + (y[i]-my)*(y[i]-my);
      pol = co[1] + co[2]* x1[i] + co[3]* x2[i];

      Sr=Sr + (y[i]-pol)*(y[i]-pol);

      }


dyx=sqrt(Sr/(n-(m+1)));
rr=(St-Sr)/St;
```

```
        printf("\n El error estándar de la aproximación es %g",dyx);
        printf("\n El coeficiente de correlación es %g",rr);


}
```

SALIDA PARA EL EJEMPLO 2.4

```
La matriz ampliada del sistema formado es
 6  6  24  103
 6  10  28  86
 24  28  112  424

 Los coeficientes del polinomio son
 a[0]=14.1667
 a[1]=-6.66667
 a[2]=2.41667
 El error estándar de la aproximación es 0.408248
 El coeficiente de correlación es 0.996499
Elapsed time = 00:01:01.62.  Program returned (43).  Press any key.
```

ANEXO No. 7

PROGRAMA DE REGRESIÓN NO LINEAL (ECUACIÓN DE POTENCIAS DE BASE e)

```
/*programa que ajusta una ecuación de potencias de base e */
/* mediante linealización de los datos */
# include <stdio.h>
 # include <stdlib.h>
# include <math.h>
main()
{
int n,i;
float aux,x[10],y[10],sx=0,sy=0,sxx=0,sxy=0,mx,my,A,B,C;
system("cls");
printf("Introduzca el numero de puntos a usar:");
scanf("%d",&n);
for (i=1;i<=n;i++)
              { printf("\n x[%d]=",i);
                scanf("%f",&x[i]);
                printf("y[%d] =", i);
                scanf("%f",&y[i]);
              }
for (i=1;i<=n;i++)
{
      aux= log(y[i]);
      y[i]= aux;
 }

for(i=1;i<=n;i++)
{
```

```
   sx=sx+x[i];
   sy=sy+y[i];
   sxx=sxx+x[i]*x[i];
   sxy=sxy+x[i]*y[i];
}
  mx=sx/n;
  my=sy/n;
  A=(n*sxy-sx*sy)/(n*sxx-(sx*sx));
  B=my-mx*A;
printf("\n A= %g  B=%g",A,B);
printf("\n La recta de regresi¢n es y = %g x+ %g",A,B);

C= exp(B);
printf("\n El valor de C= %g y A = %g",C,A);
printf("\n Se tiene que  y = %g exp(%gx) ",C,A);
 }
```

SALIDA PARA EL EJEMPLO 2.5

A= 0.391202 B=0.457367

La recta de regresión es y = 0.391202 x+ 0.457367

El valor de C= 1.57991 y A = 0.391202

Se tiene que y = 1.57991 exp(0.391202x)

Elapsed time = 00:00:42.73. Program returned (43). Press any key.

BIBLIOGRAFIA

- Burden , Richard L. y Faires J Douglas, Análisis Numérico, 2da Edición, Grupo Editorial Iberoamericano, S.A de C.V, México, 1996.

- Chapra Steven C. y Canales Raymomd P., Métodos Numéricos para Ingenieros, Editorial McGraw-Hill, México, 1998.

- Henrici Meter, Elementos de Análisis Numérico, Editorial Trillas, México, 1980.

- Mathews John H., Fink kurtis D., Métodos Numéricos con MATLAB, Tercera Edición, Editorial Pretice Hill, Madrid, 2000.

- Ralston Anthony, Introducción al Análisis Numérico, Editorial Limusa, México, 1978.