Practica No. 9

Ajuste de curvas

Nombre(s):

Objetivo:

Con la realización de esta práctica se pretende: implementar métodos aproximados e interpolados para el ajuste de curvas mediante ANSI C.

Fundamento Teórico:

Una situación frecuente en ingeniería es la de tener una función *f* dada por un conjunto de puntos $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$ donde falta la representación del modelo analítico
La función *f* puede representar los datos de un experimento o el resultado de mediciones a gran escala de una magnitud física que no se puede convertir a una forma analítica sencilla
Puede ser necesario evaluar la función *f* en algún punto *x* en el conjunto de datos $x_1,\ldots,x_n$, pero donde *x* es diferente de los valores tabulados, este proceso se denomina **ajuste de curvas**

Ajuste por mínimos cuadrados
El ajuste de curvas por aproximación mediante el método de mínimos cuadrados es un proceso en el cual, dado un conjunto de *N* pares de puntos (x_n,y_n) (siendo *x* la variable independiente y *y* la dependiente), se determina una función matemática *f(x)* tal que la suma de los cuadrados de la diferencia entre la imagen real y la correspondiente obtenida mediante la función ajustada en cada punto sea mínima
Generalmente, se escoge una función genérica *f(x)* en función de uno o más parámetros y se ajusta el valor de estos parámetros de forma que se minimice el error cuadrático (ε)
La forma más típica de esta función ajustada es la de un polinomio de grado *M*; obteniéndose para *M=1* un ajuste lineal (o regresión lineal):

$$f(x) = a_0 + a_1 x \tag{4.2}$$

Para calcular la pendiente (a_1) tenemos:

$$a_1 = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2} \tag{4.4}$$

Para calcular la intersección (a_0) tenemos:

$$a_0 = \frac{\sum y - a_1(\sum x)}{n} \tag{4.5}$$

Donde:
n es el número de puntos de la imagen real

$$\sum y = y_1 + y_2 + y_3 + ... + y_n$$
$$\sum x = x_1 + x_2 + x_3 + ... + x_n$$
$$\sum x^2 = x_1^2 + x_2^2 + x_3^2 + ... + x_n^2$$
$$\sum xy = x_1y_1 + x_2y_2 + x_3y_3 + ... + x_ny_n$$

Interpolación

Polinomio unico de Interpolación

El problema de la interpolación consiste en estimar el valor de una función en un punto a partir de valores conocidos en puntos cercanos.
En el caso de la interpolación polinómica, la función incógnita se sustituye por un polinomio que coincide con aquella en los puntos conocidos
Se eligen los polinomios porque son fáciles de evaluar y por el hecho fundamental de que dados $n+1$ puntos de abscisa distinta, (x_0, y_0), (x_1, y_1),..., (x_n, y_n), existe exactamente un polinomio $P_n(x)$ de grado no superior a n, que pasa por dichos puntos
Asumiendo un polinomio de la forma:

$$P_n(x_i) = a_0 + a_1x + a_2x^2 + ... + a_nx^n \qquad (4.7)$$

Al tener que cumplir con las restricciones (4.6), se generan $n+1$ ecuaciones en $n+1$ incógnitas; siendo éstas los coeficientes a_i's:

$$\begin{aligned} a_0 + a_1x_0 + a_2x_0^2 + a_3x_0^3 ... + a_nx_0^n &= y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 ... + a_nx_1^n &= y_1 \\ a_0 + a_1x_2 + a_2x_2^2 + a_3x_2^3 ... + a_nx_2^n &= y_2 \\ \vdots \qquad & \qquad \vdots \\ a_0 + a_1x_n + a_2x_n^2 + a_3x_n^3 ... + a_nx_n^n &= y_n \end{aligned} \qquad (4.8)$$

Y, en forma matricial:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ & & \vdots & & \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad (4.9)$$

Al resolver el sistema se encuentran los valores del vector **a = [a_0,a_1,a_2,...,a_n]**

Polinomio de Interpolación de Lagrange

Sea $L_{in}(x)$ un polinomio de grado n, que se anule en todos los puntos x_j, $j = 0, 1, ..., n$, salvo en el *i-ésimo*, donde vale 1; es decir, tal que:

$$L_i(x_j) = 0 \text{ si } j \neq i \text{ y } L_i(x_i) = 1$$

La existencia de este polinomio se deriva del resultado anterior, pero puede obtenerse directamente, sin necesidad de resolver un sistema, gracias a la siguiente fórmula debida a Lagrange:

$$L_{in}(x) = \frac{(x-x_0)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}$$

Es inmediato comprobar entonces que el polinomio:

$$P_n(x) = y_0L_0(x) + y_1L_1(x) + y_2L_2(x) + ... + y_nL_n(x)$$

cumple las condiciones de la ec. 4.6, lo que prueba directamente la existencia del polinomio de interpolación
La unicidad se puede garantizar utilizando el hecho de que un polinomio de grado *n* puede tener a lo sumo *n* raíces
Si dos polinomios de grado $\leq n$ interpolan *n+1* puntos, su diferencia se anula en dichos puntos, por lo que sólo puede ser el polinomio idénticamente nulo

Seudocodigo:

Algoritmo 10: MÉTODO DE INTERPOLACIÓN DE LAGRANGE

Entradas: numero de coordenadas: n, datos (x, y) y el valor que se desea interpolar x_{int}:
Salidas: $f(x_{int})$

INICIO
- $f(x_{int}) \leftarrow 0$;
- $i \leftarrow 0$;
- **while** $i \leq (n-1)$ **do**
  - $L \leftarrow 1$;
  - $j \leftarrow 0$;
  - **while** $j \leq (n-1)$ **do**
    - **if** $i \neq j$ **then**
      - $L \leftarrow L * \frac{x_{int}-x(j)}{x(i)-x(j)}$;
    - $j \leftarrow j+1$;
  - $f(x_{int}) \leftarrow f(x_{int}) + L * f(x(i))$;
  - $i \leftarrow i+1$;
- *Imprimir* $f(x_{int})$

FIN

Polinomio de Interpolación de Newton

Dada una familia de $n+1$ puntos $\{(x_0,f(x_0)),(x_1,f(x_1)),\ldots,(x_n,f(x_n))\}$, todos con abscisas distintas, definimos la base de Newton asociada a dicho soporte, como:

$$\{1,(x-x_0),(x-x_0),(x-x_1),(x-x_0),(x-x_1),(x-x_2),\ldots,\prod_{i=0}^{n-1}(x-x_i)\} \quad (4.14)$$

Aunque no es necesario, se trabajará con el soporte ordenado de menor a mayor.

Dada una familia de $n+1$ puntos $\{(x_0,y_0),(x_1,y_1),\ldots,(x_n,y_n)\}$, todos con abscisas distintas, definimos el polinomio interpolador de Newton asociado a dicho soporte, como:

$$P_n(x) = A_0+A_1(x-x_0)+A_2(x-x_0)+(x-x_1)+A_3(x-x_0),(x-x_1),(x-x_2)+\ldots+A_n\prod_{i=0}^{(n-1)}(x-x_i) \quad (4.15)$$

siempre que $P_n(x_i) = y_i$ para $i = 0, 1, 2, \ldots, n$.

Es importante hacer notar que el polinomio que se acaba de definir, de grado n, es una combinación lineal de la base de Newton, lo que permitirá ampliarlo con facilidad; y además es el mismo que se obtiene por Lagrange para esos mismos datos (recordar que el polinomio interpolador asociado a un soporte es único).

Sólo hay que explicitar cómo calcular los coeficientes A_i.

Dada una familia de $n + 1$ puntos $\{(x_0, y_0), (x_1, y_1),\ldots, (x_n, y_n)\}$, todos con abscisas distintas donde $f(x_i) = y_i$ definimos las DIFERENCIAS DIVIDIDAS de NEWTON asociadas a dichos datos, de la siguiente manera:

En general, diferencias divididas de orden $k < n$:

$$f[x_i, x_{(i+1)},\ldots,x_{(i+k)}] = \frac{f[x_{(i+1)},\ldots,x_{(i+k)}] - f[x_i, x_{(i+1)},\ldots,x_{(i+k-1)}]}{x_{(i+k)} - x_i} \quad (4.16)$$

para $i=0,1,2,\ldots,(n-k)$ con $1 \leq k$.

Seudocódigo

Algoritmo 11: MÉTODO DE INTERPOLACIÓN DE NEWTON

Entradas: numero de coordenadas: n, datos (x, y) y el valor que se desea interpolar x_{int}:

Salidas: $f(x_{int})$

INICIO

`/* Tabla de diferencias */`

$m \leftarrow n - 1;$

$i \leftarrow 0;$

$f(x_{int}) \leftarrow f(x_0);$

while $i \leq (m - 1)$ **do**

$T(i, 0) \leftarrow \frac{f(x(i+i)) - f(x(i))}{x(i+1) - x(i)};$

$i \leftarrow i + 1;$

$j \leftarrow 1;$

while $j \leq m - 1$ **do**

$i \leftarrow j;$

while $i \leq m - 1$ **do**

$T(i, j) \leftarrow \frac{T(i, j-1) - T(i-1, j-1)}{x(i+1) - x(i-j)};$

$i \leftarrow i + 1;$

$j \leftarrow j + 1;$

`/* Inicia función principal */`

$f(x_{int}) \leftarrow f(x(0));$

$i \leftarrow 0;$

while $i \leq n - 1$ **do**

$p \leftarrow 1;$

$j \leftarrow 0;$

while $j \leq i$ **do**

$p \leftarrow p * (x_{int} - x(j));$

$j \leftarrow j + 1;$

$f(x_{int}) \leftarrow f(x_{int}) + T(i, i) * p;$

$i \leftarrow i + 1;$

Imprimir $f(x_{int})$

FIN

Forma de trabajo:

Colaborativa en equipos de 3 personas

Material:

1. Computadora
2. Compilador de lenguaje ANSI C

Procedimiento:

1. Calcular la función de regresión lineal para el siguiente conjunto de datos:

   Cinco niños de 2, 3, 5, 7 y 8 años de edad pesan, respectivamente, 14, 20, 32, 42 y 44 kilos.

   Hallar la ecuación de la **recta de regresión** de la edad sobre el peso.

   ¿Cuál sería el peso aproximado de un niño de seis años?

2. Encontrar el polinomio de interpolación único para los valores:

   (10, 0.1763),(20, 0.3640),(30, 0.5774)

   e interpolar el valor para x=21

3. Obtener el polinomio de interpolación usando la fórmula de interpolación de Lagrange con la siguiente tabla de valores, e interpolar en el punto x = −4.
   xk = {−6, 6, −1}   yk= { 8, −16, −2}
4. Calcular el polinomio de interpolación de Newton para los siguientes datos

x_i	-2	-1	2	3
y_i	4	1	4	9

Para la creación del programa deberán realizarse los siguientes pasos:

1. En las primeras líneas elaborar comentarios con la siguiente información:
   a. Nombre de la institución
   b. Nombre del centro al que pertenece la carrera
   c. Nombre del departamento al que pertenece la carrera
   d. Nombre de la materia
   e. Nombre(s) de quien(es) realiza(n) la práctica
   f. Nombre del profesor
   g. Una descripción breve de lo que realiza el programa
2. Incluir las librerías necesarias.
3. Se debe desplegar un menú para ejecutar los métodos anteriormente señalados y una opción para salir del sistema.
4. Al seleccionar ejecutar el método el usuario debe proporcionar: las coordenadas establecidas para cada problema.
5. Una vez realizada cualquier operación debe regresar al menú principal.
6. Al salir se debe detener el programa y luego regresar el control al sistema inicial.

Resultados:

Realizar al menos dos corridas de prueba para cada operación y mostrar imágenes de las pantallas de texto generadas.

Una vez terminado el programa debe subirse a la plataforma de **aulavirtual** junto con este reporte.

Conclusiones: