

13
24.



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLAN".

AJUSTE PARAMETRICO DE EQUIPOS
REPRESENTADOS POR ECUACIONES
ALGEBRAICO DIFERENCIALES

T E S I S

QUE PARA OBTENER EL TITULO DE:
LICENCIADO EN MATEMATICAS
APLICADAS Y COMPUTACION
P R E S E N T A :
OSCAR CUEVAS GARCIA



MEXICO, D. F.

NOVIEMBRE 1997

TESIS CON
CALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres, Ramón y Margarita,
por darme la vida y por el tesoro tan grande
que uno puede recibir de ellos que es el estudio.

A mis hermanos, Mauricio y Javier,
como un estímulo para que siempre
luchen por sus ideales.

A mis familiares,
como una meta cabalmente cumplida.

**A1 Doctor David Juárez Romero,
por su apoyo y consejos en el presente trabajo.**

**A1 Físico Manuel Valadez Rodríguez,
por su valiosa colaboración en este trabajo.**

**A mis profesores y compañeros,
con estimación y cariño.**

ÍNDICE

1. Introducción.....	1
1.1 Características de los modelos representados en centrales generadoras.....	1
Tipos de Simulación.....	1
Técnicas de Modelación Matemática.....	2
Clasificación de los Modelos Matemáticos.....	4
Características de los Modelos Matemáticos.....	4
2. Métodos Para Ajuste de Parámetros.....	6
2.1 Mínimos Cuadrados Lineales.....	6
Descripción del método.....	7
Forma de Solución.....	10
Proyección y Mínimos Cuadrados.....	14
Algoritmo para el ajuste de datos de mínimos cuadrados.....	18
Mínimos Cuadrados como una combinación lineal de funciones.....	20
Algoritmo para el ajuste de datos de mínimos cuadrados como una combinación lineal de funciones.....	23
Mínimos Cuadrados Pesados.....	25
2.2 Métodos para Ajustes por Intervalos.....	27
Interpolación Segmentaria Lineal.....	28
Condiciones de Continuidad.....	29
Forma de Solución.....	29
Ejemplo.....	30
Interpolación Segmentaria Cuadrática.....	32
Condiciones de Continuidad.....	33
Forma de Solución.....	33
Ejemplo.....	34
Interpolación Segmentaria Cúbica.....	39
Condiciones de Continuidad.....	40
Forma de Solución.....	41
Ejemplo.....	42
Conclusiones.....	48
2.3 Mínimos Cuadrados no Lineales.....	81
Método de Levenberg-Marquardt.....	83
Algoritmo de Levenberg-Marquardt para la solución de Mínimos Cuadrados no lineales.....	84
Mínimos Cuadrados no Lineales con Restricciones.....	88
3. Diseño de un Código para Ajustar Parámetros de Equipos.....	91
Introducción.....	91
3.1 Definición de los Datos Experimentales Disponibles.....	92
Características de la Gráfica de los Datos Experimentales.....	92
3.2 Definición de los Parámetros del Método.....	93
Parámetros del Método de Mínimos Cuadrados Lineales.....	93

Parámetros del Método de Interpolación Cúbica Segmentaria.....	94
Parámetros del Método de Mínimos Cuadrados no Lineales.....	95
Requerimientos de los Métodos de Ajuste.....	97
3.3 Definición de las Características del Ajuste.....	98
4. Aplicación.....	100
4.1 Aplicación de las Técnicas de Ajuste de Parámetros.....	100
Ajuste Lineal.....	100
Ajuste por Segmentos.....	101
Ajuste no Lineal.....	101
5. Discusión de Resultados.....	102
5.1 Discusión de Resultados para Mínimos Cuadrados Lineales.....	102
5.2 Discusión de Resultados para la Interpolación Cúbica Segmentaria.....	103
5.3 Discusión de Resultados para Mínimos Cuadrados no Lineales.....	104
6. Conclusiones y Limitaciones.....	105
6.1 Conclusiones y Limitaciones para el método de Mínimos Cuadrados Lineales.....	105
6.2 Conclusiones y Limitaciones para la Interpolación Cúbica Segmentaria.....	106
6.3 Conclusiones y Limitaciones para el método de Mínimos Cuadrados no Lineales.....	107
6.4 Conclusiones Generales.....	109
Apéndice B-1	
Técnicas de Apoyo en la Resolución de Mínimos Cuadrados.....	110
Apéndice B-2	
Ajuste de Datos por Mínimos Cuadrados Utilizando el Paquete Mathematica.....	115
Bibliografía.....	119

1. INTRODUCCIÓN

1. INTRODUCCIÓN.

1.1 CARACTERÍSTICAS DE LOS MODELOS REPRESENTADOS EN CENTRALES GENERADORAS.

Normalmente los procesos industriales resultan ser muy sofisticados por lo que no es fácil operar con ellos de una forma inmediata segura y eficiente, si no se tiene la habilidad requerida para manejarlos.

Generalmente los problemas se presentan durante un encendido ó arranque ó paro de los procesos, causados por cambios en las condiciones de alimentación de un suministro. Durante estos cambios pueden surgir accidentes provocados por modificaciones para aumentar la carga, por fallas imprevistas de componentes, etc.; La reacción del proceso ante estos disturbios puede ser simulado para prever situaciones riesgosas.

La simulación nos lleva a tener un mejor entendimiento de cómo funciona un proceso ya que ésta trabaja de una manera similar al equipo real. La simulación nos permite observar las fallas ó comportamiento anormal que pudieran resultar de maniobras erróneas en el proceso.

Las ventajas de la simulación de los procesos industriales son las siguientes:

- Capacitar a los operadores para tomar acciones correctivas cuando surjan fallas en el proceso.
- Adiestrar a los operadores para realizar las maniobras adecuadas con el fin de obtener el comportamiento deseado del proceso.
- Evaluar resultados de maniobras erróneas del operador.
- Reducir el número de interrupciones del proceso.
- Mejorar la operación del proceso.
- Disminuir costos de operación.
- Prever accidentes.

El costo para desarrollar un simulador es alto, pero es mayor el costo por fallas de operación del proceso. Por lo que es preferible darle entrenamiento a los operadores con los simuladores que con el proceso real.

Tipos de Simulación.

La simulación de estados estables de procesos complejos sirve para predecir el funcionamiento de dichos procesos cuando las condiciones se han estabilizado. La simulación de condiciones estables puede facilitar el estudio de sensibilidad del proceso ante cambios en parámetros y variables.

Para llevar a cabo la representación de un proceso, para análisis de estados transitorios, análisis de seguridad, análisis de protecciones ó cambios de diseño, es necesario un Simulador. Generalmente la simulación es dinámica ya que los estados arriba mencionados cambian constantemente.

Para que el simulador cumpla plenamente con el objetivo de adiestrar a los operadores en forma práctica y eficaz, debe satisfacer una serie de requerimientos que darán la pauta al instructor para definir las funciones que efectuará el simulador.

A continuación describiremos las partes de un simulador: (Ramos, et al, 1981).

-Una computadora de alta capacidad de cómputo (la consola) donde se define el modo de operación del simulador y se lleva a cabo el monitoreo de las acciones que realiza el operador en el tablero de control.

-Varios monitores gráficos a color de alta resolución que a su vez tienen herramientas adaptables para el manejo de información en pantallas tales como:

pantalla de diálogo: Presenta menús que guían e indican al instructor cuáles

acciones debe seleccionar para llevar a cabo una función,

pantalla de despliegues: Presenta tablas y listados del estado del simulador y las claves y nombres de las posibles acciones,

pantalla de gráficas: Permite visualizar la dinámica de variables que se consideren relevantes.

-Estación de botones: Permite activar instantáneamente la función que se quiere realizar e indica al instructor las funciones que se efectúan.

-Teclado: Permite introducir en la computadora las instrucciones que eligen las acciones a realizar.

-Impresora de Pantalla: Permite obtener una copia del despliegue que aparece en cualesquiera de las pantallas.

-Teléfono: Permite al instructor comunicarse con los operadores y viceversa.

Para poder desarrollar un simulador es preciso representar mediante modelos los fenómenos que uno desea incluir en el simulador.

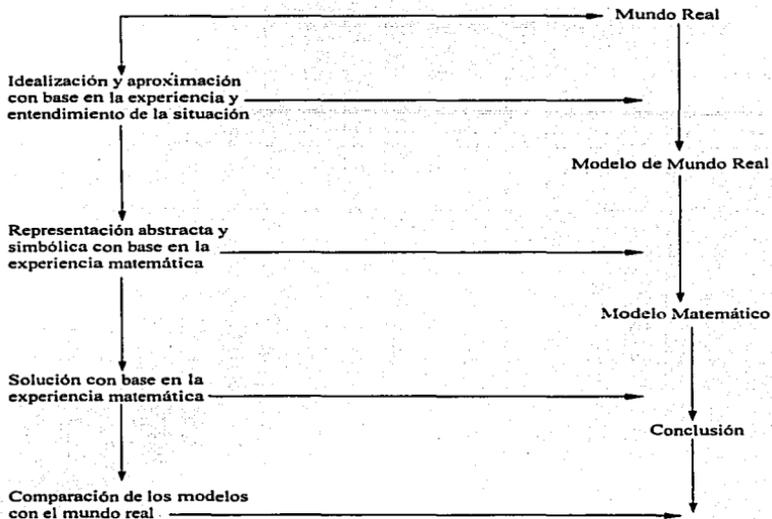
Técnicas de Modelación Matemática.

El simulador contiene modelos que representan el funcionamiento del proceso. Debido a las características de flexibilidad de tales modelos, estos son generalmente matemáticos. Estos se desarrollan de acuerdo a las características, circunstancias, objetivos, restricciones que requieren los procesos que deseamos simular.

Las técnicas de modelación matemática consisten en traducir los problemas del mundo real en problemas matemáticos, resolviendo los problemas matemáticos e interpretando su solución en el lenguaje del mundo real. Para resolver un problema del mundo real es conveniente simplificarlo o aproximararlo a otro problema que sea muy cercano al original y que aún pueda ser traducido y resuelto matemáticamente.

A continuación se mostrará un esquema de la modelación matemática.
(González, et al, 1989).

ESQUEMA DE LA MODELACIÓN MATEMÁTICA



Si las comparaciones del modelo con el mundo real no son satisfactorias, entonces modificamos la idealización supuesta o investigamos otra estructura para el modelo matemático. Esto conduce al siguiente procedimiento para resolver problemas a través de la modelación matemática.

1.- Observar el fenómeno del mundo real que se desea modelar en forma objetiva a fin de encontrar todas sus características relevantes al fenómeno y encontrar los aspectos que son irrelevantes.

2.- Formular las características relevantes y los aspectos irrelevantes del problema en términos matemáticos.

3.- Resolver el modelo de acuerdo con el planteamiento matemático hecho anteriormente.

4.- Desarrollar programas de computación de los métodos numéricos que se puedan utilizar.

5.- Verificar si los resultados obtenidos son los deseables; si no, regresar a la formulación o a la resolución del modelo.

6.- Simular el modelo para comparar los resultados observados contra los resultados que arroja la simulación para así tener la validación del modelo.

El criterio para modelar con éxito incluye una buena armonía entre las predicciones y observaciones.

Clasificación de los Modelos Matemáticos.

Los modelos matemáticos pueden ser clasificados de acuerdo a los siguientes puntos:

Su Objetivo. trazar, discernir, predecir, accionar, optimizar ó controlar.

Su Naturaleza. lineal, no lineal, continuos, discretos, dinámicos, estáticos, determinísticos, estocásticos.

Sus Técnicas de resolución. Por medio de álgebra, ecuaciones diferenciales ordinarias o parciales, optimización, cálculo de variaciones, gráficas, programación matemática, probabilidad, estadística, etc.

Características de los Modelos Matemáticos.

-Robustez. Los modelos matemáticos pueden ser ó pueden no ser robustos. Esto depende de observar si pequeños cambios en los parámetros conducen a pequeños cambios en su comportamiento. Si esto ocurre, se dice que el modelo es robusto. El análisis de sensibilidad permite conocer si el modelo es robusto o no.

-Consistentes. Los modelos que incluyen inecuaciones y/o desigualdades, deben ser consistentes en las diferentes zonas donde éstas se verifican.

-Flexibles. Un modelo debe incluir la posibilidad de mejorarse a la luz de los experimentos o datos observables. (Mediante ajustes en los parámetros).

En los modelos es muy importante su validación ya que esto nos lleva a que el modelo pueda o no ser aceptado.

La validación del modelo requiere de ajuste de parámetros para cumplir con las tendencias esperadas en él. Algunos parámetros son estimados con la ayuda de datos que pueden ser usados también para validarlo. Para hacer que dicha validación sea más confiable se deben usar también datos independientes; es decir, que no estén relacionados entre sí.

En los modelos de ingeniería química es muy importante el ajuste de parámetros, ya que en dichos modelos tenemos curvas, gráficas, y no sabemos como representarlos. Para el análisis de estos, recopilamos datos de presión, temperatura y flujo de la instrumentación instalada en el campo, así como de los valores de consumo de corriente y de voltaje en los motores de bombas. Posteriormente, se calcula la capacidad y la eficiencia de las bombas en operación real y se comparan con los valores de diseño, para obtener el comportamiento real del equipo con ayuda de las curvas características (Flujo contra Eficiencia, Flujo contra Presión Diferencial). Las técnicas para el ajuste de parámetros a utilizar en nuestro trabajo son:

Mínimos cuadrados lineales, no lineales, con peso, con restricciones; ajuste por intervalos (Spline).

2. MÉTODOS PARA AJUSTE DE PARÁMETROS

2 MÉTODOS PARA AJUSTE DE PARÁMETROS.

2.1 MÍNIMOS CUADRADOS LINEALES.

Introducción.

Los datos que estudiaremos a continuación requieren de ajustes de curvas para ver que comportamiento tienen estos a lo largo de su rango, así como para tener una curva que permita analizarlos con mayor facilidad, ya que el ajuste de curvas permite tener una mejor idea de su comportamiento. Para esto empezaremos estudiando el ajuste de datos por mínimos cuadrados.

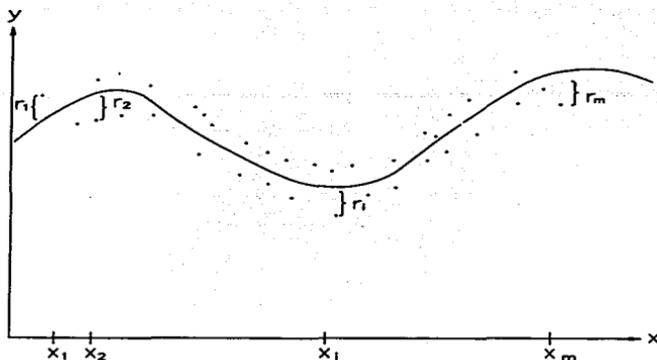
Los mínimos cuadrados se definen como la mejor curva que nos aproxima a un conjunto dado de datos. Esta aproximación se obtiene cuando el error involucrado r es la suma de los cuadrados de las diferencias entre los valores observados y los valores de la curva aproximante. Representando esta aproximación por $f(x_i) = c_0 + c_1x_i + c_2x_i^2 + \dots + c_nx_i^n$ como el i -ésimo valor de la curva aproximante y por y_i al i -ésimo valor observado, es necesario encontrar las constantes c_j (de acuerdo al grado del polinomio que deseamos) tal que minimicen el error r de mínimos cuadrados. (Fig. 2.1.3)

El error r se define como el valor de norma $r = \left| \sum_{i=1}^n r_i \right|^2$ donde

$$r_i = y_i - \sum_{j=0}^n c_j x_i^j \quad (2.1.1)$$

De manera general, el problema consiste en

$$\text{Minimizar } \left\| \sum_{i=1}^n \left(y_i - \sum_{j=0}^n c_j x_i^j \right) \right\|_2^2 \quad (2.1.2)$$



(Fig. 2.1.3)

La solución de los problemas mediante el uso de mínimos cuadrados da el mejor conjunto de coeficientes de la combinación lineal y proporciona una medida del error total en el mejor modelo. La solución mediante mínimos cuadrados nos ayuda a decidir el conjunto de parámetros, sin embargo, un error grande en el modelo, aún con coeficientes óptimos, puede indicar que el modelo es inadecuado.

Descripción del método.

Supongamos que empezamos con un sistema lineal $y = Xc$; Sea X una matriz de $m \times n$, suponemos que el número m de observaciones es mayor que el número n de incógnitas, suponemos que el sistema $y = Xc$ es inconsistente (no tiene solución; ó probablemente no exista una elección del parámetro c que ajuste perfectamente el dato y ; en otras palabras el vector y no será una combinación lineal de las columnas de X).

El problema de mínimos cuadrados es escoger el vector \bar{c} que minimice el error. En términos matemáticos, la norma utilizada más extensamente en los problemas de mínimos cuadrados es la norma Euclídiana ó norma Dos. Cuando $\|y - X\bar{c}\|$ es tan pequeña como sea

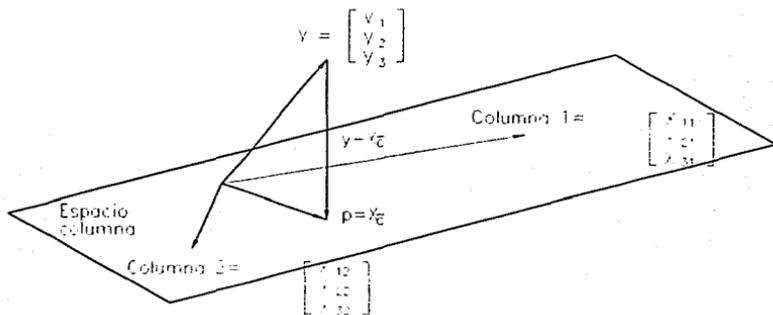
posible, nosotros decimos que \bar{c} es una solución óptima del problema lineal de mínimos cuadrados; esto es,

$$\text{Minimizar } \|y - X\bar{c}\|_2^2 \quad (2.1.4)$$

El error lo definimos como $r = \|y - X\bar{c}\|$. Este error representa la distancia de y al punto $X\bar{c}$ en el espacio columna de X ($X\bar{c}$ es la combinación lineal de las columnas de la matriz X usando los coeficientes c_0, \dots, c_n). Por lo tanto, encontrar \bar{c} para la solución de mínimos cuadrados, (lo cual minimizará el error r), es lo mismo que situar el punto $p = X\bar{c}$ lo más cercano posible a y que sólo es otro punto en el espacio columna. La interpretación geométrica se describe a continuación.

Interpretación geométrica.

p es la "proyección de y sobre el espacio columna" y el vector $y - X\bar{c}$ es perpendicular para ese espacio (Fig. 2.1.5)



(Fig. 2.1.5) Proyección sobre el espacio columna de una matriz de 3×2 .

La perpendicular $y - X\bar{c}$ al espacio es expresada como sigue. Cada vector en el espacio columna de X es una combinación lineal de columnas, con algunos coeficientes z_0, \dots, z_n . En otras palabras, éste es un vector de la forma Xz . Para toda la elección de z , estos vectores en el plano tienen que ser perpendiculares al vector error $y - X\bar{c}$:

$$(Xz)^T \cdot (X\bar{z} - y) = 0 \quad \text{ó} \quad z^T [X^T X \bar{z} - X^T y] = 0 \quad (2.1.6)$$

Esto es verdadero para toda z , y hay sólo un camino en el cual esto pueda ocurrir: el vector en paréntesis cuadrados tiene que ser el vector cero, $X^T X \bar{z} - X^T y = 0$; esta ecuación nos lleva a las ecuaciones fundamentales de mínimos cuadrados.

La solución de mínimos cuadrados para un sistema inconsistente $y = X\bar{z}$ de m ecuaciones con n incógnitas satisface

$$X^T X \bar{z} = X^T y \quad (2.1.7)$$

Estas ecuaciones son conocidas como las "ecuaciones normales". Si las columnas de X son linealmente independientes, entonces por la siguiente regla si " X tiene columnas linealmente independientes, entonces $X^T X$ es una matriz cuadrada simétrica e invertible," la matriz $X^T X$ es invertible y la solución única de mínimos cuadrados es

$$\bar{z} = (X^T X)^{-1} X^T y \quad (2.1.8)$$

La proyección de y sobre el espacio columna es por consiguiente el punto aproximado p

$$p = X\bar{z} = X(X^T X)^{-1} X^T y$$

Esta fórmula es expresada en términos matriciales como una Matriz Proyección P que describe la construcción geométrica de una línea perpendicular de y al espacio columna de X ; es decir:

$$P = X(X^T X)^{-1} X^T$$

Esta matriz proyección $P = X(X^T X)^{-1} X^T$ tiene 2 propiedades básicas:

- i) Es idempotente: $P^2 = P$
- ii) Es simétrica: $P = P^T$

Inversamente, una matriz con esas 2 propiedades representa una proyección sobre el espacio columna de P .

Forma de Solución.

Dadas las observaciones y_1, y_2, \dots, y_m a puntos distintos x_1, x_2, \dots, x_m , la curva que minimiza la norma de error $r = \|y - Xc\|$ es

$$\sum_{i=1}^m \left[y_i - \sum_{j=0}^2 c_j x_i^j \right]^2 \quad (2.1.9)$$

Para obtener la minimización derivamos la suma anterior para cada c_j del polinomio.

Ejemplo:

Para una aproximación con un polinomio de grado dos con m observaciones, descomponemos la suma (2.1.9)

$$f(x_i) = c_0 + c_1 x_i + c_2 x_i^2$$

$$\text{Minimizar } \sum_{i=1}^m \left[y_i - (c_0 + c_1 x_i + c_2 x_i^2) \right]^2$$

Derivamos parcialmente para c_0, c_1, c_2 , e igualamos a cero cada derivada parcial.

$$\begin{aligned} \frac{\partial \sum_{i=1}^m \left[y_i - (c_0 + c_1 x_i + c_2 x_i^2) \right]^2}{\partial c_0} &= 2 \sum_{i=1}^m \left[y_i - (c_0 + c_1 x_i + c_2 x_i^2) \right] \\ &= 2 \left[\sum_{i=1}^m y_i - \sum_{i=1}^m c_0 - \sum_{i=1}^m c_1 x_i - \sum_{i=1}^m c_2 x_i^2 \right] = 0 \\ \sum_{i=1}^m y_i - \sum_{i=1}^m c_0 - \sum_{i=1}^m c_1 x_i - \sum_{i=1}^m c_2 x_i^2 &= 0 \\ c_0 m + c_1 \sum_{i=1}^m x_i + c_2 \sum_{i=1}^m x_i^2 &= \sum_{i=1}^m y_i \end{aligned}$$

$$\begin{aligned} \frac{\partial \sum_{i=1}^n [y_i - (c_0 + c_1 x_i + c_2 x_i^2)]^2}{\partial c_1} &= 2x_i \sum_{i=1}^n [y_i - (c_0 + c_1 x_i + c_2 x_i^2)] \\ &= 2 \left[\sum_{i=1}^n x_i y_i - c_0 \sum_{i=1}^n x_i - c_1 \sum_{i=1}^n x_i^2 - c_2 \sum_{i=1}^n x_i^3 \right] = 0 \\ \sum_{i=1}^n x_i y_i - c_0 \sum_{i=1}^n x_i - c_1 \sum_{i=1}^n x_i^2 - c_2 \sum_{i=1}^n x_i^3 &= 0 \\ c_0 \sum_{i=1}^n x_i + c_1 \sum_{i=1}^n x_i^2 + c_2 \sum_{i=1}^n x_i^3 &= \sum_{i=1}^n x_i y_i \end{aligned}$$

$$\begin{aligned} \frac{\partial \sum_{i=1}^n [y_i - (c_0 + c_1 x_i + c_2 x_i^2)]^2}{\partial c_2} &= 2x_i^2 \sum_{i=1}^n [y_i - (c_0 + c_1 x_i + c_2 x_i^2)] \\ &= 2 \left[\sum_{i=1}^n x_i^2 y_i - c_0 \sum_{i=1}^n x_i^2 - c_1 \sum_{i=1}^n x_i^3 - c_2 \sum_{i=1}^n x_i^4 \right] = 0 \\ \sum_{i=1}^n x_i^2 y_i - c_0 \sum_{i=1}^n x_i^2 - c_1 \sum_{i=1}^n x_i^3 - c_2 \sum_{i=1}^n x_i^4 &= 0 \\ c_0 \sum_{i=1}^n x_i^2 + c_1 \sum_{i=1}^n x_i^3 + c_2 \sum_{i=1}^n x_i^4 &= \sum_{i=1}^n x_i^2 y_i \end{aligned}$$

El sistema de ecuaciones queda de la siguiente forma:

$$\begin{aligned} c_0 n + c_1 \sum_{i=1}^n x_i + c_2 \sum_{i=1}^n x_i^2 &= \sum_{i=1}^n y_i \\ c_0 \sum_{i=1}^n x_i + c_1 \sum_{i=1}^n x_i^2 + c_2 \sum_{i=1}^n x_i^3 &= \sum_{i=1}^n x_i y_i \\ c_0 \sum_{i=1}^n x_i^2 + c_1 \sum_{i=1}^n x_i^3 + c_2 \sum_{i=1}^n x_i^4 &= \sum_{i=1}^n x_i^2 y_i \end{aligned}$$

Los coeficientes a encontrar son c_0, c_1, c_2 ; éstos son los coeficientes del polinomio de mínimos cuadrados de grado dos a calcular.

En forma matricial para un polinomio de grado n se tiene:

$$\begin{bmatrix} n & \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \dots & \sum_{i=1}^n x_i^n \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 & \dots & \sum_{i=1}^n x_i^{n+1} \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 & \sum_{i=1}^n x_i^5 & \dots & \sum_{i=1}^n x_i^{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^n x_i^n & \sum_{i=1}^n x_i^{n+1} & \sum_{i=1}^n x_i^{n+2} & \sum_{i=1}^n x_i^{n+3} & \dots & \sum_{i=1}^n x_i^{2n} \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^2 y_i \\ \vdots \\ \sum_{i=1}^n x_i^n y_i \end{bmatrix}$$

$$X^T X \bar{c} = X^T y$$

Estas son las ecuaciones normales para la solución de mínimos cuadrados.

A continuación observaremos un ejemplo para que quede más claro lo visto anteriormente.

Ejemplo: Supongamos que damos 4 mediciones para aproximar una curva cuadrática:

x	y
0	0
1	1
3	8
4	15

El sistema sobredeterminado $y = X\bar{c}$ para este conjunto de medidas es:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1^0 & 1^1 & 1^2 \\ 3^0 & 3^1 & 3^2 \\ 4^0 & 4^1 & 4^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}$$

Necesitamos la forma $X^T X$ y su inversa así como $X^T y$:

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \\ 0 & 1 & 9 & 16 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 26 \\ 8 & 26 & 92 \\ 26 & 92 & 338 \end{bmatrix} = \begin{bmatrix} m & \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 \end{bmatrix}$$

$$X^T y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \\ 0 & 1 & 9 & 16 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 8 \\ 15 \end{bmatrix} = \begin{bmatrix} 24 \\ 85 \\ 313 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^2 y_i \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} 9 & -13 & 1 \\ 10 & 15 & 16 \\ -13 & 169 & -4 \\ 15 & 90 & 9 \\ 1 & 4 & 1 \\ 6 & 9 & 9 \end{bmatrix}$$

La solución de mínimos cuadrados $\vec{z} = (X^T X)^{-1} X^T y$ es la siguiente:

$$\vec{z} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 9 & -13 & 1 \\ 10 & 15 & 16 \\ -13 & 169 & -4 \\ 15 & 90 & 9 \\ 1 & 4 & 1 \\ 6 & 9 & 9 \end{bmatrix} \begin{bmatrix} 24 \\ 85 \\ 313 \end{bmatrix} = \begin{bmatrix} 1 \\ 10 \\ 3 \\ 10 \\ 1 \\ 1 \end{bmatrix}$$

La mejor curva es: $f(x) = \frac{1}{10} - \frac{3}{10}x + x^2$

Como vimos anteriormente el problema de mínimos cuadrados resulta ser un problema de álgebra lineal, pero que pasa cuando la matriz X de las ecuaciones normales $X^T X \vec{z} = X^T y$ tiene columnas ortonormales; para esto veremos lo que es una base ortonormal así como su proyección.

Proyecciones y mínimos cuadrados: el caso ortonormal. (Strang 1980).

Una base v_1, \dots, v_n es llamada ortonormal si

$$v_i^T v_j = \delta_{ij} = \begin{cases} \delta_{ij} = 1 & \text{si } i = j \\ \delta_{ij} = 0 & \text{si } i \neq j \end{cases} \quad \text{donde } \delta_{ij} \text{ es la función delta de Kronecker.}$$

Supongamos que D es una matriz de $m \times n$ con columnas ortonormales. Entonces estas columnas son independientes, y la multiplicación de $D^T D$ es la identidad por lo descrito anteriormente para bases ortonormales.

$$D^T D = I \quad (2.1.10)$$

Volviendo a la matriz proyección sobre el espacio columna y la solución de mínimos cuadrados \bar{c} se tiene:

$$P = D(D^T D)^{-1} D^T \quad y \quad \bar{c} = (D^T D)^{-1} D^T y$$

Pero como $D^T D = I$ se obtiene

$$P = D D^T \quad y \quad \bar{c} = D^T y$$

Cuando los ejes coordenados son perpendiculares, la proyección sobre el espacio está simplificada dentro de la proyección sobre cada eje (Fig. 2.1.11). Donde la matriz proyección queda de la siguiente forma $P = d_1 d_1^T y + \dots + d_n d_n^T y$.

$$P = D D^T y = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ d_1 & \vdots & \vdots & d_n \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} d_1^T y \\ \vdots \\ d_n^T y \end{bmatrix} = d_1 d_1^T y + \dots + d_n d_n^T y$$

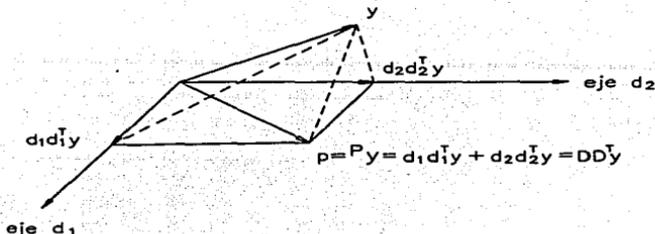


Fig. (2.1.11) La proyección sobre un plano es igual a la suma de proyecciones sobre los vectores d_1 y d_2 ortonormales.

como $D^T D$ es la matriz identidad, p es la suma de las n proyecciones separadas sobre d_1 y d_2 ortonormales.

Tenemos 5 ecuaciones las cuales son básicas.

- 1.- $y = Xc$ la ecuación dada probablemente es inconsistente.
- 2.- $X^T X \bar{c} = X^T y$ las ecuaciones normales para \bar{c} .
- 3.- $p = X \bar{c}$ la proyección de y sobre el espacio columna de X .
- 4.- $P = X(X^T X)^{-1} X^T$ la matriz proyección dado $p = Py$.
- 5.- $\bar{c} = X^T y$ y $P = X X^T = x_1 x_1^T + x_2 x_2^T + \dots + x_n x_n^T$, el caso especial en el cual X tiene columnas ortonormales.

Con los resultados de las ecuaciones $\bar{c} = X^T y$ y $P = X X^T = x_1 x_1^T + x_2 x_2^T + \dots + x_n x_n^T$ donde X tiene columnas ortonormales, describiremos lo que es una matriz ortogonal ya que ésta nos ayudará a introducirnos a la factorización QR para la solución de las ecuaciones normales $X^T X \bar{c} = X^T y$ de mínimos cuadrados.

Matriz Ortogonal.

Una matriz ortogonal es una matriz cuadrada Q con columnas ortonormales, q_1, \dots, q_n , es decir:

$$Q^T Q = I \quad (2.1.12)$$

La ecuación (2.1.12) es un caso particular de la ecuación (2.1.10). Si Q^T es una inversa izquierda, entonces es la inversa. En otras palabras Q^T es también una inversa derecha, y $Q Q^T = I$. Así para una matriz ortogonal se tiene que $Q Q^T = Q^T Q = I$; esto es $Q^T = Q^{-1}$. Además si Q es una matriz Ortogonal, entonces Q^T también lo es.

La multiplicación por una matriz Ortogonal Q preserva la longitud.

$$\|Qx\| = \|x\| \text{ para todo vector } x.$$

y también preserva el producto interno:

$$(Qx)^T (Qy) = x^T y \text{ para todos los vectores } x \text{ y } y.$$

La prueba es inmediata donde $(Qx)^T (Qy) = x^T Q^T Q y = x^T I y = x^T y$. Si $y = x$; Esta ecuación queda como $\|Qx\|^2 = \|x\|^2$ y por consiguiente es preservada la longitud siempre que el producto interno exista.

El proceso de ortogonalización de Gram-Schmidt (Ver apéndice B-1) nos ayuda a la solución del problema de mínimos cuadrados $\bar{c} = (X^T X)^{-1} X^T y$ descomponiendo la matriz X en QR donde Q es una matriz ortogonal y R es una Matriz Triangular Superior; El trabajo preliminar de ortogonalización ahorra la tarea de formar $X^T X$ y resuelve las ecuaciones normales de $X^T X \bar{c} = X^T X y$ de mínimos cuadrados.

Si tenemos el problema $\bar{c} = (X^T X)^{-1} X^T y$ y la descomposición $X = QR$ entonces,

$$\bar{c} = (R^T Q^T Q R)^{-1} R^T Q^T y \quad (2.1.13),$$

pero $Q^T Q = I$, porque las columnas de Q son ortonormales. Por lo tanto,

$$\bar{c} = (R^T R)^{-1} R^T Q^T y = R^{-1} Q^T y$$

$$\bar{c} = R^{-1} Q^T y \quad (2.1.14)$$

El cálculo de \bar{x} sólo requiere la multiplicación de la matriz vector $Q^T y$, seguido por la sustitución hacia atrás en el sistema triangular.

$$R\bar{x} = Q^T y.$$

Volviendo al tema de mínimos cuadrados, tenemos que para calcular el error del polinomio calculado se utiliza la siguiente fórmula

$$r = \sum_{i=1}^n |y_i - f(x_i)|^2 \quad (2.1.15)$$

donde las y_i son las componentes del vector de datos observados y $f(x_i)$ son las componentes del vector de la evaluación de los puntos del vector x en el polinomio.

Con los elementos descritos anteriormente se tiene el siguiente algoritmo:

Algoritmo para el ajuste de datos de mínimos cuadrados.

Paso 1.- Capturar los datos del problema dentro de un archivo de datos de la forma:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_m & y_m \end{bmatrix}$$

Paso 2.- Introducir el orden del polinomio a ser ajustado (n es el orden) de la forma:

$$f(x_i) = c_0 + c_1 x_i + c_2 x_i^2 + \dots + c_n x_i^n$$

Paso 3.- Introducir una r para el error de tolerancia dado.

Paso 4.- Si $m < n$ imprimir un mensaje de error de que el polinomio de regresión es imposible y termina el proceso. Si $m \geq n$ continuar.

Paso 5.- Calcular la multiplicación $X^T X$ y $X^T y$.

Paso 6.- Una vez calculadas estas multiplicaciones tenemos el siguiente sistema.

$$\begin{aligned} c_0 m + c_1 \sum_{i=1}^m x_i + c_2 \sum_{i=1}^m x_i^2 + \dots + c_n \sum_{i=1}^m x_i^n &= \sum_{i=1}^m y_i \\ c_0 \sum_{i=1}^m x_i + c_1 \sum_{i=1}^m x_i^2 + c_2 \sum_{i=1}^m x_i^3 + \dots + c_n \sum_{i=1}^m x_i^{n+1} &= \sum_{i=1}^m x_i y_i \\ c_0 \sum_{i=1}^m x_i^2 + c_1 \sum_{i=1}^m x_i^3 + c_2 \sum_{i=1}^m x_i^4 + \dots + c_n \sum_{i=1}^m x_i^{n+2} &= \sum_{i=1}^m x_i^2 y_i \\ \vdots & \vdots \\ c_0 \sum_{i=1}^m x_i^n + c_1 \sum_{i=1}^m x_i^{n+1} + c_2 \sum_{i=1}^m x_i^{n+2} + \dots + c_n \sum_{i=1}^m x_i^{2n} &= \sum_{i=1}^m x_i^n y_i \end{aligned}$$

Paso 7.- Resolver el sistema de ecuaciones lineales.

Paso 8.- Imprimir el vector de coeficientes o vector de parámetros del polinomio a calcular

$$\vec{c} = [c_0, c_1, \dots, c_n].$$

Paso 9.- Evaluar los datos introducidos (Vector x) del problema del polinomio calculado e imprimir $f(x_i)$.

Paso 10.- Si las componentes y , y $(f(x,))$ cumplen con el error de tolerancia impuesto $r \leq \tau$ donde $r = \sum_{i=1}^m |y_i - f(x_i)|^2$ imprimir resultados y salir del programa.

Paso 11.- Regresar al paso 2 y aumentar el orden del polinomio.

Mínimos cuadrados como una combinación lineal de funciones.

Los problemas de mínimos cuadrados lineales surgen cuando una variable dependiente es modelada como una combinación lineal de funciones involucrando parámetros específicos. En este problema lo que desconocemos son los coeficientes de la combinación lineal. Esta combinación lineal de la variable se hace de acuerdo a la transformación que deseamos darle a la variable que puede ser x a \sqrt{x} ó, x a $1/x$, etc. Estas transformaciones ayudan a tener un mejor ajuste de la curva, a suavizar (linearizar la curva) en alguna parte del dominio.

Se tienen un conjunto de m observaciones; para cada observación x_i , $i = 1, 2, \dots, m$ el modelo lineal tiene la forma:

$$y(x_i) = c_0\Phi_0(x_i) + c_1\Phi_1(x_i) + \dots + c_n\Phi_n(x_i) = \sum_{j=0}^n c_j\Phi_j(x_i)$$

donde $y(x_i)$ (el valor de la variable dependiente) es aproximada por una suma pesada de $c_0\Phi_0(x_i) + c_1\Phi_1(x_i) + \dots + c_n\Phi_n(x_i)$ (la n especifica funciones evaluadas a la observación i).

La solución \vec{c} del problema de mínimos cuadrados lineales es un n -vector que produce la suma más pequeña del cuadrado de los errores en el modelo dado sobre el conjunto de observaciones y_i , es decir, que resuelve

$$\text{Minimizar } \sum_{i=1}^m \left[y_i - \sum_{j=0}^n c_j\Phi_j(x_i) \right]^2 \quad (2.1.16)$$

Para obtener la minimización se deriva parcialmente esta ecuación para cada c_j del polinomio.

Ejemplo:

Para una aproximación con un polinomio de grado dos con m observaciones, descomponemos la ecuación (2.1.16)

$$f(x_i) = c_0\Phi_0(x_i) + c_1\Phi_1(x_i) + c_2\Phi_2(x_i) \quad i = 1, 2, \dots, m$$

$$\text{Minimizar } \sum_{i=1}^m \left[y_i - (c_0\Phi_0(x_i) + c_1\Phi_1(x_i) + c_2\Phi_2(x_i)) \right]^2$$

Derivamos parcialmente para c_0, c_1, c_2 , e igualamos a cero cada derivada parcial

$$\frac{\partial \sum_{i=1}^n [y_i - (c_0 \Phi_0(x_i) + c_1 \Phi_1(x_i) + c_2 \Phi_2(x_i))]^2}{\partial c_0} = 2 \sum_{i=1}^n \Phi_0(x_i) [y_i - (c_0 \Phi_0(x_i) + c_1 \Phi_1(x_i) + c_2 \Phi_2(x_i))]$$

$$\sum_{i=1}^n \Phi_0(x_i) y_i - c_0 \sum_{i=1}^n \Phi_0^2(x_i) - c_1 \sum_{i=1}^n \Phi_0(x_i) \Phi_1(x_i) - c_2 \sum_{i=1}^n \Phi_0(x_i) \Phi_2(x_i) = 0$$

$$\frac{\partial \sum_{i=1}^n [y_i - (c_0 \Phi_0(x_i) + c_1 \Phi_1(x_i) + c_2 \Phi_2(x_i))]^2}{\partial c_1} = 2 \sum_{i=1}^n \Phi_1(x_i) [y_i - (c_0 \Phi_0(x_i) + c_1 \Phi_1(x_i) + c_2 \Phi_2(x_i))]$$

$$\sum_{i=1}^n \Phi_1(x_i) y_i - c_0 \sum_{i=1}^n \Phi_0(x_i) \Phi_1(x_i) - c_1 \sum_{i=1}^n \Phi_1^2(x_i) - c_2 \sum_{i=1}^n \Phi_1(x_i) \Phi_2(x_i) = 0$$

$$\frac{\partial \sum_{i=1}^n [y_i - (c_0 \Phi_0(x_i) + c_1 \Phi_1(x_i) + c_2 \Phi_2(x_i))]^2}{\partial c_2} = 2 \sum_{i=1}^n \Phi_2(x_i) [y_i - (c_0 \Phi_0(x_i) + c_1 \Phi_1(x_i) + c_2 \Phi_2(x_i))]$$

$$\sum_{i=1}^n \Phi_2(x_i) y_i - c_0 \sum_{i=1}^n \Phi_0(x_i) \Phi_2(x_i) - c_1 \sum_{i=1}^n \Phi_1(x_i) \Phi_2(x_i) - c_2 \sum_{i=1}^n \Phi_2^2(x_i) = 0$$

El sistema de ecuaciones queda de la siguiente forma:

$$c_0 \sum_{i=1}^n \Phi_0^2(x_i) + c_1 \sum_{i=1}^n \Phi_0(x_i) \Phi_1(x_i) + c_2 \sum_{i=1}^n \Phi_0(x_i) \Phi_2(x_i) = \sum_{i=1}^n y_i \Phi_0(x_i)$$

$$c_0 \sum_{i=1}^n \Phi_0(x_i) \Phi_1(x_i) + c_1 \sum_{i=1}^n \Phi_1^2(x_i) + c_2 \sum_{i=1}^n \Phi_1(x_i) \Phi_2(x_i) = \sum_{i=1}^n y_i \Phi_1(x_i)$$

$$c_0 \sum_{i=1}^n \Phi_0(x_i) \Phi_2(x_i) + c_1 \sum_{i=1}^n \Phi_1(x_i) \Phi_2(x_i) + c_2 \sum_{i=1}^n \Phi_2^2(x_i) = \sum_{i=1}^n y_i \Phi_2(x_i)$$

La representación en forma matricial es semejante a la anterior ($X^T X$ y $X^T y$) solo que los coeficientes de la matriz son funciones, y en la matriz anterior los coeficientes eran variables.

$$\Phi \cdot \Phi^T = \begin{bmatrix} \sum_{i=1}^n \Phi_0^2(x_i) & \sum_{i=1}^n \Phi_0(x_i)\Phi_1(x_i) & \sum_{i=1}^n \Phi_0(x_i)\Phi_2(x_i) & \cdots & \sum_{i=1}^n \Phi_0(x_i)\Phi_n(x_i) \\ \sum_{i=1}^n \Phi_1(x_i)\Phi_0(x_i) & \sum_{i=1}^n \Phi_1^2(x_i) & \sum_{i=1}^n \Phi_1(x_i)\Phi_2(x_i) & \cdots & \sum_{i=1}^n \Phi_1(x_i)\Phi_n(x_i) \\ \sum_{i=1}^n \Phi_2(x_i)\Phi_0(x_i) & \sum_{i=1}^n \Phi_2(x_i)\Phi_1(x_i) & \sum_{i=1}^n \Phi_2^2(x_i) & \cdots & \sum_{i=1}^n \Phi_2(x_i)\Phi_n(x_i) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n \Phi_n(x_i)\Phi_0(x_i) & \sum_{i=1}^n \Phi_n(x_i)\Phi_1(x_i) & \sum_{i=1}^n \Phi_n(x_i)\Phi_2(x_i) & \cdots & \sum_{i=1}^n \Phi_n^2(x_i) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix}$$

$$\Phi^T y = \begin{bmatrix} \sum_{i=1}^n \Phi_0(x_i)y_i \\ \sum_{i=1}^n \Phi_1(x_i)y_i \\ \sum_{i=1}^n \Phi_2(x_i)y_i \\ \vdots \\ \sum_{i=1}^n \Phi_n(x_i)y_i \end{bmatrix}$$

Con lo anterior tenemos el siguiente algoritmo para encontrar los coeficientes de mínimos cuadrados.

Algoritmo para el ajuste de datos de mínimos cuadrados como una combinación lineal de funciones.

Paso 1.- Capturar los datos del problema dentro de un archivo de datos de la forma:

$$\begin{array}{|c|c|} \hline [x] & [y] \\ \hline x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_m & y_m \\ \hline \end{array}$$

paso 2.- Introducir el orden del polinomio a ser ajustado (n es el número de coeficientes).

Paso 3.- Introducir una r para el error de tolerancia dado.

Paso 4.- Si $m < n$ imprimir un mensaje de error de que el polinomio de regresión es imposible y termina el proceso. Si $m \geq n$ continuar.

Paso 5.- Seleccionar las funciones $\Phi_i(x)$ que nosotros deseamos ajustar.

Paso 6.- Calcular la multiplicación $\Phi^T \Phi$ y $\Phi^T y$ para que calcule las sumas y productos de potencia como se da en el sistema de ecuaciones siguiente

$$\begin{array}{l} c_0 \sum_{i=1}^m \Phi_0^2(x_i) + c_1 \sum_{i=1}^m \Phi_0(x_i)\Phi_1(x_i) + c_2 \sum_{i=1}^m \Phi_0(x_i)\Phi_2(x_i) + \dots + c_n \sum_{i=1}^m \Phi_0(x_i)\Phi_n(x_i) = \sum_{i=1}^m \Phi_0(x_i)y_i \\ c_0 \sum_{i=1}^m \Phi_1(x_i)\Phi_0(x_i) + c_1 \sum_{i=1}^m \Phi_1^2(x_i) + c_2 \sum_{i=1}^m \Phi_1(x_i)\Phi_2(x_i) + \dots + c_n \sum_{i=1}^m \Phi_1(x_i)\Phi_n(x_i) = \sum_{i=1}^m \Phi_1(x_i)y_i \\ c_0 \sum_{i=1}^m \Phi_2(x_i)\Phi_0(x_i) + c_1 \sum_{i=1}^m \Phi_2(x_i)\Phi_1(x_i) + c_2 \sum_{i=1}^m \Phi_2^2(x_i) + \dots + c_n \sum_{i=1}^m \Phi_2(x_i)\Phi_n(x_i) = \sum_{i=1}^m \Phi_2(x_i)y_i \\ \vdots \\ c_0 \sum_{i=1}^m \Phi_n(x_i)\Phi_0(x_i) + c_1 \sum_{i=1}^m \Phi_n(x_i)\Phi_1(x_i) + c_2 \sum_{i=1}^m \Phi_n(x_i)\Phi_2(x_i) + \dots + c_n \sum_{i=1}^m \Phi_n^2(x_i) = \sum_{i=1}^m \Phi_n(x_i)y_i \end{array}$$

Paso 7.- Resolver el sistema lineal.

Paso 8.- Imprimir el vector de coeficientes o vector de parámetros del polinomio a calcular.

Paso 9.- Evaluar los datos introducidos (Vector x) del problema del polinomio calculado e imprimir $f(x)$.

Paso 10.- Si las componentes y_i y $f(x_i)$ cumplen con el error de tolerancia impuesto

$r \leq \epsilon$ donde $r = \sum_{i=1}^m |y_i - f(x_i)|^2$ imprimir resultados y salir del programa.

Paso 11.- Regresar al paso 2 y aumentar el orden del polinomio o cambiar la función a ajustar.

Otra forma de resolver el problema de mínimos cuadrados es utilizando el paquete de Software **Mathematica** (Ver *apéndice B-2*).

Mínimos Cuadrados Pesados.

Al realizar ajustes por mínimos cuadrados, algunas veces deseamos que ciertos datos tengan más significancia que otros. Normalmente esto ocurre cuando los datos no son igualmente confiables.

Regresando al sistema lineal $y = Xc$ de mínimos cuadrados; si las m observaciones y_1, y_2, \dots, y_m no son igualmente confiables, se pueden aplicar diferentes pesos w_1, w_2, \dots, w_m a las observaciones. Esto generaliza nuestro problema original, pero hay todavía otra generalización. Las observaciones en su desigual confiabilidad pueden no ser independientes. En este caso también se pueden introducir coeficientes w_{ij} para especificar el acoplamiento entre dos observaciones; digamos la i y la j . Entonces la igualdad $Xc = y$ es cambiada por $WXc = Wy$. Los números w_{ij} permanecen a lo largo de la diagonal principal de la matriz W ; es decir, son de la forma $w_{11}, w_{22}, \dots, w_{mm}$.

Para resolver este nuevo problema $WXc = Wy$ tenemos solamente que regresar a las ecuaciones normales $X^T X \bar{c} = X^T y$ para el problema original y hacer los cambios apropiados - X es reemplazado por WX y y es reemplazada por Wy .

Si X tiene columnas independientes (es de rango n) y W es invertible, entonces la solución de mínimos cuadrados $WXc = Wy$ esta determinada por las ecuaciones normales pesadas

$$(X^T W^T W X) \bar{c}_w = X^T W^T W y \quad (2.1.17)$$

Si escribimos $W^T W = H$, entonces $\bar{c}_w = (X^T H X)^{-1} X^T H y$

Esto completa la primera aproximación de mínimos cuadrados pesados, la segunda aproximación consiste en introducir el cambio en la definición de longitud (*producto interno*). Por lo tanto realizamos exactamente el mismo procedimiento (*ver producto interno de matriz ortogonal*) que el empleado para la matriz pesada W .

Si W es una matriz invertible, podemos definir una nueva longitud (*producto interno*) por las siguientes reglas

$$\|x\|_w = \|Wx\| \quad \text{y} \quad (x, y)_w = (Wx)^T (Wy) = x^T W^T W y$$

El producto interno tiene la siguiente propiedad

$$\|x\|_w^2 = (x, x)_w > 0 \quad \text{para todos los vectores } x \text{ no ceros.}$$

Regresando al problema de mínimos cuadrados $y = Xc$ con matriz W , podemos recobrar la figura geométrica usando la nueva longitud y producto interno. La elección óptima de Xc , minimiza el error $y - Xc$, es a su vez, el punto p en el espacio columna de X próximo a y . Pero la palabra *próximo* tiene que ser ahora interpretada en términos de la nueva longitud: minimizando $\|y - Xc\|_w = \|Wy - WXc\|$. Este punto p (o más propiamente p_w , que depende de W) es así mismo la proyección de y dentro del espacio columna y el vector $y - p_w$ es perpendicular a este espacio. Pero una vez más hay un cambio, *perpendicularidad* significa ahora que el nuevo producto es cero. Así que el punto $p_w = X\bar{c}_w$ está definido por la propiedad de que $c - p_w$ es perpendicular a todo vector Xz en el espacio columna

$$(Xz)^T W^T W (y - X\bar{c}_w) = 0 \quad \text{para toda } z.$$

El vector que multiplica a z^T tiene que ser cero.

$$X^T W^T W (y - X\bar{c}_w) = 0 \quad \text{ó} \quad X^T W^T W X \bar{c}_w = X^T W^T W y;$$

lo cual es el caso análogo a las ecuaciones normales.

2.2 MÉTODOS PARA AJUSTES POR INTERVALOS.

Introducción.

Los ajustes de parámetros por mínimos cuadrados aproximan curvas continuas en todo un intervalo, sin embargo, existen aplicaciones en las que discontinuidades ó picos en algún lugar del rango restringen su uso.

Un enfoque alternativo para ajustar las curvas de comportamiento de equipos consiste en dividir el rango en una colección de subintervalos y construir un polinomio aproximante en cada subintervalo. La aproximación con funciones de este tipo se llama aproximación polinomial segmentaria.

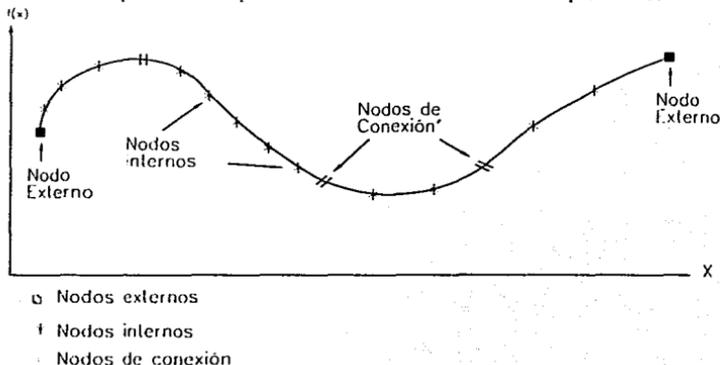
Las características de esta aproximación son:

- 1) continuidad.
- 2) forma en que se fijan las condiciones de aproximación.

A continuación veremos diferentes tipos de aproximación segmentaria.

Tipos de la Aproximación Segmentaria.

Antes de estudiar los tipos de aproximación, observaremos la siguiente gráfica donde se describen los tipos de nodos que se utilizan en la construcción de la aproximación.



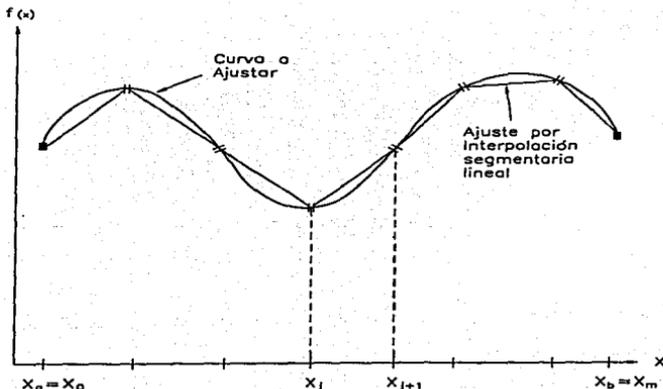
(Fig. 2.2.1)

De acuerdo al orden del polinomio, la aproximación segmentaria se clasifica en lineal, cuadrática, cúbica, etc.

INTERPOLACIÓN SEGMENTARIA LINEAL.

Descripción: Consiste en unir un conjunto de datos de nodos con una serie de líneas rectas $S(x) = a_j + b_j(x - x_j)$ donde a_j es la evaluación de la función en el punto y b_j es la pendiente de la línea recta que une a cualquier nodo entre el subintervalo $[x_j, x_{j+1}]$.

Continuidad: En general no produce una aproximación diferenciable, ya que en cada uno de los subintervalos la primera derivada de la función puede ser discontinua. En los puntos de conexión, los nodos de conexión donde coinciden los polinomios, la pendiente puede cambiar abruptamente.



(Fig. 2.2.2)

Condiciones de Continuidad.

Condiciones: Se define la interpolación segmentaria lineal como sigue: Supóngase que $f(x)$ es una función definida en $[X_a, X_b]$ y que se da un conjunto de números llamados nodos $X_a = x_0 < x_1 < \dots < x_{m-1} = X_b$. Un polinomio lineal $S_j(x) = a_j + b_j(x - x_j)$, en el subintervalo $[x_j, x_{j+1}]$ (donde $j=0, 1, \dots, m-1$), que satisface las siguientes condiciones:

1) Los valores de la aproximación deben ser iguales al valor de la función en los nodos interiores; es decir.

$$S_j(x_j) = f(x_j) \quad \text{para cada } j = 0, 1, \dots, m-1$$

2) Los valores del segmento de aproximación j deben ser iguales a los valores del siguiente segmento de aproximación en los nodos de conexión, en símbolos

$$S_{j+1}(x_{j+1}) = S_j(x_{j+1}) \quad \text{para cada } j = 0, 1, \dots, m-1$$

Forma de Solución:

Tenemos que el polinomio a obtener es $S_j(x) = a_j + b_j(x - x_j)$ en el subintervalo $[x_j, x_{j+1}]$.

Aplicamos la condición 1) a x_j , $S_j(x_j) = f(x_j)$, con $j = 0, 1, \dots, m-1$. Se tiene

$$S_j(x_j) = a_j = f(x_j)$$

Aplicamos la condición 2) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$, tenemos

$$\begin{aligned} a_{j+1} &= S_{j+1}(x_{j+1}) = S_j(x_{j+1}) \\ a_{j+1} &= a_j + b_j(x_{j+1} - x_j) \quad \text{para cada } j = 0, 1, \dots, m-1. \end{aligned}$$

$$\text{Definimos: } \Delta x_j = x_{j+1} - x_j$$

Entonces $a_{j+1} = a_j + b_j \Delta x_j$ debe satisfacerse para $j=0, 1, \dots, m-1$.

Como $a_j = f(x_j)$ y $a_{j+1} = f(x_{j+1})$ nuestro sistema de ecuaciones queda como sigue:

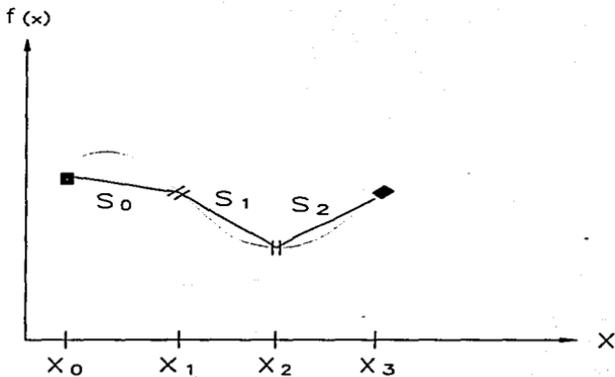
$$\begin{aligned} a_j &= f(x_j) \\ a_j + b_j \Delta x_j &= f(x_{j+1}) \end{aligned}$$

donde las incógnitas son a_j, b_j .

Ejemplo:

Se tiene la siguiente gráfica

x	$f(x)$
x_0	$f(x_0)$
x_1	$f(x_1)$
x_2	$f(x_2)$
x_3	$f(x_3)$



El polinomio a obtener es $S_j(x) = a_j + b_j(x - x_j)$ en el subintervalo $[x_j, x_{j+1}]$.

Evaluamos la aproximación en los puntos x_0, x_1

$$S_0(x_0) = a_0 + b_0(x_0 - x_0) = f(x_0)$$

$$S_0(x_1) = a_0 + b_0(x_1 - x_0) = f(x_1)$$

Evaluamos la aproximación en los puntos x_1, x_2

$$S_1(x_1) = a_1 + b_1(x_1 - x_1) = f(x_1)$$

$$S_1(x_2) = a_1 + b_1(x_2 - x_1) = f(x_2)$$

Evaluamos la aproximación en los puntos x_2, x_3

$$S_2(x_2) = a_2 + b_2(x_2 - x_2) = f(x_2)$$

$$S_2(x_3) = a_2 + b_2(x_3 - x_2) = f(x_3)$$

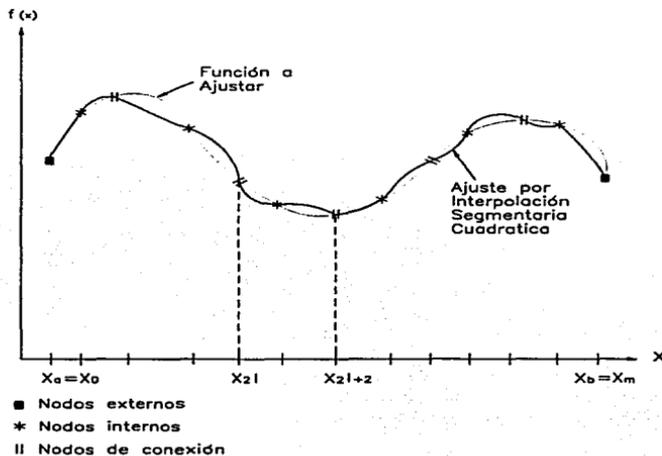
Resolviendo el sistema para cada intervalo S_0, S_1, S_2 se obtienen los coeficientes $a_0, b_0, a_1, b_1, a_2, b_2$.

INTERPOLACIÓN SEGMENTARIA CUADRÁTICA.

Descripción: Esta interpolación ajusta un polinomio cuadrático entre parejas de subintervalos. $[x_{2l}, x_{2l+2}]$. Tiene 3 constantes arbitrarias.

$S(x) = a_l + b_l(x - x_{2l}) + c_l(x - x_{2l})^2$ donde $l = 0, 1, \dots, n-1$ se refiere al número de segmento. Sólo se requieren tres condiciones para ajustar los datos en los extremos de cada intervalo, las cuales son: a) Los valores de la función deben ser iguales en los nodos interiores; b) La primera y última función deben pasar a través de los nodos extremos; c) Los valores de la primera derivada en los nodos de conexión deben ser iguales.

Continuidad: Existe la flexibilidad suficiente como para permitir que las ecuaciones cuadráticas se escojan de tal manera que la interpolante tenga derivada continua. (Chapra, 1996). Esta derivada coincide con la función en los extremos del intervalo $[x_{2l}, x_{2l+2}]$.



(Fig. 2.2.3)

Condiciones de Continuidad.

Condiciones: Se define la interpolación segmentaria cuadrática como sigue: Supóngase que $f(x)$ es una función definida en $[X_a, X_b]$ y que se da un conjunto de números llamados nodos $X_0 = x_0 < x_1 < \dots < x_{m-1} = X_b$. Un polinomio cuadrático $S_l(x) = a_l + b_l(x - x_{2l}) + c_l(x - x_{2l})^2$, en el subintervalo $[x_{2l}, x_{2l+2}]$ (donde $l = 0, 1, \dots, n-1$), que satisfice las siguientes condiciones:

1) Los valores de la aproximación deben ser iguales al valor de la función en los nodos interiores; es decir,

$$S_j(x_j) = f(x_j) \quad \text{para cada } j = 0, 1, \dots, m-1$$

2) Los valores del segmento de aproximación l deben ser iguales a los valores del siguiente segmento de aproximación en los nodos de conexión, en símbolos

$$S_{l+1}(x_{2l+2}) = S_l(x_{2l+2}) \quad \text{para cada segmento } l = 0, 1, \dots, n-1$$

3) Los valores de la derivadas del segmento de aproximación l deben ser iguales a los valores de la derivada del siguiente segmento de aproximación en los nodos de conexión; esto es

$$S'_{l+1}(x_{2l+2}) = S'_l(x_{2l+2}) \quad \text{para cada segmento } l = 0, 1, \dots, n-1$$

Forma de Solución:

Tenemos que el polinomio a obtener es $S_l(x) = a_l + b_l(x - x_{2l}) + c_l(x - x_{2l})^2$ en el subintervalo $[x_{2l}, x_{2l+2}]$.

Aplicamos la condición 1) a x_j , $S_l(x_j) = f(x_j)$, con $j = 0, 1, \dots, m-1$. Se tiene

$$S_l(x_j) = a_l = f(x_j)$$

Aplicamos la condición 2) $S_{l+1}(x_{2l+2}) = S_l(x_{2l+2})$. Para cada $l = 0, 1, \dots, n-1$, tenemos

$$\begin{aligned} a_{l+1} &= S_{l+1}(x_{2l+2}) = S_l(x_{2l+2}) \\ a_{l+1} &= a_l + b_l(x_{2l+2} - x_{2l}) + c_l(x_{2l+2} - x_{2l})^2 \quad \text{con } l = 0, 1, \dots, n-1. \end{aligned}$$

De manera similar $x = x_{2l}$, $b_l = S'(x_{2l})$ y observamos que

$$S'_l(x_{2l}) = b_l + 2c_l(x_{2l+2} - x_{2l})^2$$

implica que $S'_l(x_{2l}) = b_l$ cuando $l = 0, 1, \dots, n-1$.

Aplicamos la condición 3) $S'_{l+1}(x_{2l+2}) = S'_l(x_{2l+2})$. para cada $l = 0, 1, \dots, n-1$ se tiene

$$S'_l(x_{2l+2}) = b_l + 2c_l(x_{2l+2} - x_{2l})^2$$

Si definimos $\Delta x_l = x_{2l+2} - x_{2l}$ para cada $l = 0, 1, \dots, n-1$.

Con lo anterior nuestro sistema de ecuaciones queda como

$$\begin{aligned} a_l &= f(x_{2l}) \\ a_l + b_l \Delta x_l + c_l \Delta x_l^2 - a_{l+1} &= 0 \\ b_l + 2c_l \Delta x_l - b_{l+1} &= 0 \end{aligned}$$

donde las incógnitas son a_l , b_l , c_l .

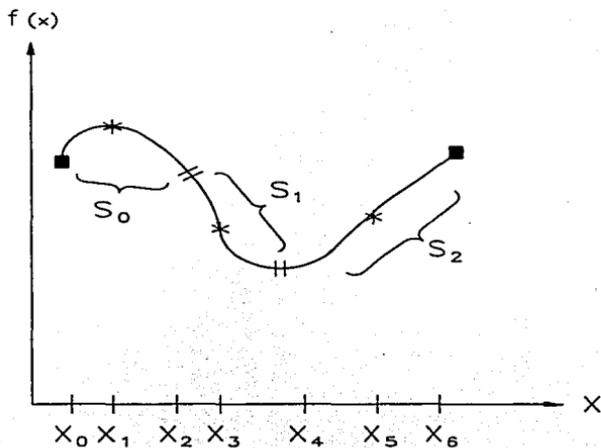
De la siguiente gráfica podemos obtener varios casos:

- a) El polinomio de aproximación es continuo solamente en la función.
- b) El polinomio de aproximación es continuo y su primera derivada es continua.
- c) La primera derivada del polinomio de aproximación es igual a la primera derivada de la función de aproximación.

Ejemplo:

Se tiene la siguiente gráfica

x	$f(x)$
x_0	$f(x_0)$
x_1	$f(x_1)$
x_2	$f(x_2)$
x_3	$f(x_3)$
x_4	$f(x_4)$
x_5	$f(x_5)$
x_6	$f(x_6)$



El polinomio a obtener es $S_l(x) = a_l + b_l(x - x_{2l}) + c_l(x - x_{2l})^2$ en el subintervalo $[x_{2l}, x_{2l+2}]$.

a) El polinomio de aproximación es continuo solamente en la función.

Evaluamos la aproximación en los puntos x_0, x_1, x_2

$$S_0(x_0) = a_0 + b_0(x_0 - x_0) + c_0(x_0 - x_0)^2 = f(x_0)$$

$$S_0(x_1) = a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 = f(x_1)$$

$$S_0(x_2) = a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 = f(x_2)$$

Evaluamos la aproximación en los puntos x_2, x_3, x_4

$$S_1(x_2) = a_1 + b_1(x_2 - x_2) + c_1(x_2 - x_2)^2 = f(x_2)$$

$$S_1(x_3) = a_1 + b_1(x_3 - x_2) + c_1(x_3 - x_2)^2 = f(x_3)$$

$$S_1(x_4) = a_1 + b_1(x_4 - x_2) + c_1(x_4 - x_2)^2 = f(x_4)$$

Evaluamos la aproximación en los puntos x_4, x_5, x_6

$$S_2(x_4) = a_2 + b_2(x_4 - x_4) + c_2(x_4 - x_4)^2 = f(x_4)$$

$$S_2(x_5) = a_2 + b_2(x_5 - x_4) + c_2(x_5 - x_4)^2 = f(x_5)$$

$$S_2(x_6) = a_2 + b_2(x_6 - x_4) + c_2(x_6 - x_4)^2 = f(x_6)$$

Resolviendo el sistema para cada sistema de ecuaciones, obtenemos los coeficientes del polinomio cuadrático aproximado S_0, S_1, S_2 .

b) El polinomio de aproximación es continuo y su primera derivada es continua.

Tenemos que el polinomio cuadrático es $S_1(x) = a_1 + b_1(x - x_{21}) + c_1(x - x_{21})^2$ donde su derivada es

$$S_1'(x) = b_1 + 2c_1(x - x_{21})$$

Evaluamos la aproximación en los puntos x_0, x_1, x_2

$$S_0(x_0) = a_0 + b_0(x_0 - x_0) + c_0(x_0 - x_0)^2 = f(x_0)$$

$$S_0(x_1) = a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 = f(x_1)$$

$$\frac{S_0(x_2)}{dx} = b_0 + 2c_0(x_2 - x_0) = \frac{S_1(x_2)}{dx}$$

Evaluamos la aproximación en los puntos x_2, x_3, x_4

$$S_1(x_2) = a_1 + b_1(x_2 - x_2) + c_1(x_2 - x_2)^2 = f(x_2)$$

$$S_1(x_3) = a_1 + b_1(x_3 - x_2) + c_1(x_3 - x_2)^2 = f(x_3)$$

$$\frac{S_1(x_4)}{dx} = b_1 + 2c_1(x_4 - x_2) = \frac{S_2(x_4)}{dx}$$

Evaluamos la aproximación en los puntos x_4, x_5, x_6

$$S_2(x_4) = a_2 + b_2(x_4 - x_4) + c_2(x_4 - x_4)^2 = f(x_4)$$

$$S_2(x_5) = a_2 + b_2(x_5 - x_4) + c_2(x_5 - x_4)^2 = f(x_5)$$

$$S_2(x_6) = a_2 + b_2(x_6 - x_4) + c_2(x_6 - x_4)^2 = f(x_6)$$

Donde el sistema de ecuaciones queda de la siguiente forma

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 = f(x_1)$$

$$b_0 + 2c_0(x_2 - x_0) - b_1 = 0$$

$$a_1 = f(x_2)$$

$$a_1 + b_1(x_3 - x_2) + c_1(x_3 - x_2)^2 = f(x_3)$$

$$b_1 + 2c_1(x_4 - x_2) - b_2 = 0$$

$$a_2 = f(x_4)$$

$$a_2 + b_2(x_5 - x_4) + c_2(x_5 - x_4)^2 = f(x_5)$$

$$a_2 + b_2(x_6 - x_4) + c_2(x_6 - x_4)^2 = f(x_6)$$

Donde las incógnitas son: $a_0, b_0, c_0, a_1, b_1, c_1, a_2, b_2, c_2$.

e) La primera derivada del polinomio de aproximación es igual a la primera derivada de la función de aproximación.

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 = f(x_1)$$

$$b_0 + 2c_0(x_2 - x_0) - b_1 = 0$$

$$b_1 + 2c_1(x_2 - x_2) = f'(x_2) \implies b_1 = f'(x_2)$$

$$a_1 + b_1(x_3 - x_2) + c_1(x_3 - x_2)^2 = f(x_3)$$

$$b_1 + 2c_1(x_4 - x_2) - b_2 = 0$$

$$b_2 + 2c_2(x_4 - x_4) = f'(x_4) \implies b_2 = f'(x_4)$$

$$a_2 + b_2(x_5 - x_4) + c_2(x_5 - x_4)^2 = f(x_5)$$

$$a_2 + b_2(x_6 - x_4) + c_2(x_6 - x_4)^2 = f(x_6)$$

El sistema de ecuaciones queda de la siguiente forma

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 = f(x_1)$$

$$b_0 + 2c_0(x_2 - x_0) = f'(x_2)$$

$$b_1 = f'(x_2)$$

$$a_1 + b_1(x_3 - x_2) + c_1(x_3 - x_2)^2 = f(x_3)$$

$$b_1 + 2c_1(x_4 - x_2) = f'(x_4)$$

$$b_2 = f'(x_4)$$

$$a_2 + b_2(x_5 - x_4) + c_2(x_5 - x_4)^2 = f(x_5)$$

$$a_2 + b_2(x_6 - x_4) + c_2(x_6 - x_4)^2 = f(x_6)$$

Estos sistemas de ecuaciones como son tridiagonales se resuelven eficientemente por medio de factorización de matrices (Crout), (Burden, 1987).

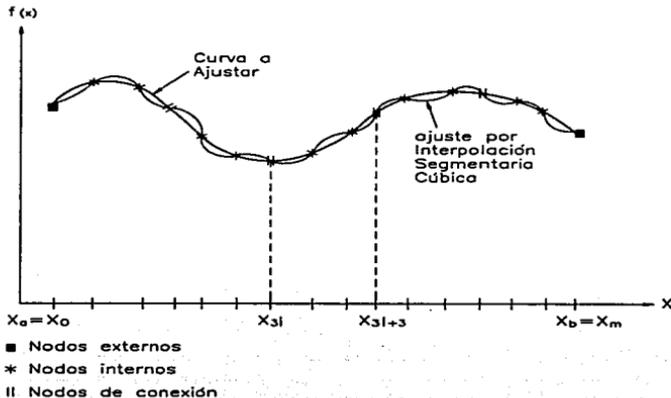
INTERPOLACIÓN SEGMENTARIA CÚBICA.

Descripción: Esta interpolación utiliza polinomios cúbicos entre parejas de subintervalos. $[x_{3j}, x_{3j+3}]$. Involucra 4 constantes.

$$S(x) = a_j + b_j(x - x_{3j}) + c_j(x - x_{3j})^2 + d_j(x - x_{3j})^3$$

por lo que requiere de 4 condiciones. Dos de ellas provienen de la restricción de que los valores de la función deben ser iguales en los nodos interiores. La primera y última función deben de pasar a través de los nodos extremos. Las otras 2 condiciones consisten en que las primeras y segundas derivadas deben ser continuas en los nodos de conexión. Estas aproximaciones también se conocen como splines.

Continuidad: Permite que el polinomio interpolante sea continuo en el intervalo y que exista primera y segunda derivada.



(Fig. 2.2.4)

Condiciones de Continuidad.

Condiciones: Se define la interpolación segmentaria cúbica como sigue: Supóngase que $f(x)$ es una función definida en $[X_a, X_b]$ y que se da un conjunto de números llamados nodos $X_a = x_0 < x_1 < \dots < x_{m-1} = X_b$. Un polinomio cúbico $S_l(x) = a_l + b_l(x - x_{1l}) + c_l(x - x_{1l})^2 + d_l(x - x_{1l})^3$, en el subintervalo $[x_{1l}, x_{1l+1}]$ (donde $l = 0, 1, \dots, m-1$), que satisfice las siguientes condiciones:

1) Los valores de la aproximación deben ser iguales al valor de la función en los nodos interiores; es decir.

$$S(x_j) = f(x_j) \quad \text{para cada } j = 0, 1, \dots, m-1$$

2) Los valores del segmento de aproximación l deben ser iguales a los valores del siguiente segmento de aproximación en los nodos de conexión, en símbolos

$$S_{l+1}(x_{1l+1}) = S_l(x_{1l+1}) \quad \text{para cada segmento } l = 0, 1, \dots, m-1$$

3) Los valores de la primera derivada del segmento de aproximación l deben ser iguales a los valores de la primera derivada del siguiente segmento de aproximación en los nodos de conexión; esto es

$$S'_{l+1}(x_{1l+1}) = S'_l(x_{1l+1}) \quad \text{para cada segmento } l = 0, 1, \dots, m-1$$

4) Los valores de la segunda derivada del segmento de aproximación l deben ser iguales a los valores de la segunda derivada del siguiente segmento de aproximación en los nodos de conexión; es decir

$$S''_{l+1}(x_{1l+1}) = S''_l(x_{1l+1}) \quad \text{para cada segmento } l = 0, 1, \dots, m-1$$

Se satisface una del siguiente conjunto de restricciones de frontera:

a) Las primeras derivadas en los nodos extremos son diferentes de cero y las segundas derivadas pueden ser cero.

$$S''(x_0) = 0 \quad (2.2.5)$$

b) Las primeras derivadas en los nodos extremos son diferentes de cero

$$S'(x_0) = f'(x_0) \text{ y } S'(x_{m-1}) = f'(x_{m-1}) \quad (2.2.6)$$

Si las segundas derivadas en los nodos extremos son cero (interpolación segmentaria natural con frontera libre), la función será una línea recta en los nodos extremos.

Si el valor de la primera derivada en los extremos fuese diferente de cero (es decir, existe alguna curvatura), entonces se tiene la restricción de frontera sujeta. La opción de frontera sujeta nos da mejores resultados que la opción de frontera libre, ya que la segunda derivada al ser diferente de cero nos garantiza la existencia de la curvatura correspondiente a la función que se desea aproximar.

Forma de Solución:

Para construir el Spline cúbico partimos de la aplicación de las condiciones de la definición de los polinomios cúbicos (*Burden 1987*).

$$S_l(x) = a_l + b_l(x - x_{3l}) + c_l(x - x_{3l})^2 + d_l(x - x_{3l})^3 \text{ para cada } l = 0, 1, \dots, n-1.$$

Donde el polinomio está definido en el intervalo $[x_{3l}, x_{3l+3}]$.

Aplicamos la condición 1) a x_j , $S_l(x_j) = f(x_j)$, con $j = 0, 1, \dots, m-1$. Se tiene

$$S_l(x_j) = a_j = f(x_j)$$

Aplicamos la condición 2) $S'_{l+1}(x_{3l+3}) = S'_l(x_{3l+3})$. Para cada $l = 0, 1, \dots, n-1$, tenemos

$$a_{l+1} = S'_{l+1}(x_{3l+3}) = S'_l(x_{3l+3})$$

$$a_{l+1} = a_l + b_l(x_{3l+3} - x_{3l}) + 2c_l(x_{3l+3} - x_{3l}) + 3d_l(x_{3l+3} - x_{3l})^2 \text{ con } l = 0, 1, \dots, n-2.$$

Definimos $\Delta x_l = x_{3l+3} - x_{3l}$ para cada $l = 0, 1, \dots, n-1$.

A $x = x_{3l}$, $a_l = f(x_{3l})$ se puede ver que esto implica que la ecuación

$$a_{l+1} = a_l + b_l \Delta x_l + c_l \Delta x_l^2 + d_l \Delta x_l^3 \text{ se satisface para toda } l = 0, 1, \dots, n-1. \text{ De manera similar } x = x_{3l}, b_l = S'(x_{3l}) \text{ y observamos que}$$

$$S'_l(x) = b_l + 2c_l(x_{3l+3} - x_{3l}) + 3d_l(x_{3l+3} - x_{3l})^2$$

implica que $S'_l(x_{3l}) = b_l$ cuando $l = 0, 1, \dots, n-1$.

Aplicamos la condición 3) $S'_{l+1}(x_{3l+3}) = S'_l(x_{3l+3})$. Para cada $l = 0, 1, \dots, n-1$ se tiene

$$b_{l+1} = b_l + 2c_l \Delta x_l + 3d_l \Delta x_l^2, \text{ para } l = 0, 1, \dots, n-1$$

De manera similar, $x = x_{3l}$. $S''(x_{3l}) = 2c_l$,

Aplicamos la condición 4) $S'_{l+1}(x_{3l+3}) = S'_l(x_{3l+3})$. Para $l = 0, 1, \dots, n-1$.

En este caso

$$2c_{l+1} = 2c_l + 6d_l \Delta x_l. \text{ entonces}$$

$$c_{l+1} = c_l + 3d_l \Delta x_l \text{ para cada } l = 0, 1, \dots, n-1.$$

Así el sistema de ecuaciones queda de la siguiente manera

$$\begin{array}{rcccccc} a_l & & & & & & = & f(x_{3l}) \\ a_l + b_l \Delta x_l + c_l \Delta x_l^2 + d_l \Delta x_l^3 - a_{l+1} & = & 0 \\ & b_l + 2c_l \Delta x_l + 3d_l \Delta x_l^2 - b_{l+1} & = & 0 \\ & & c_l + 3d_l \Delta x_l - c_{l+1} & = & 0 \end{array}$$

donde las incógnitas son a_l , b_l , c_l , d_l .

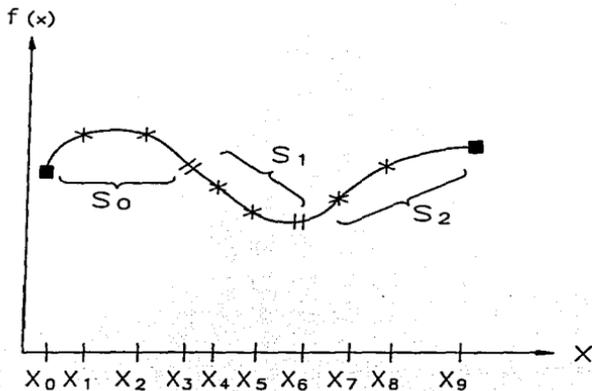
Así se pueden hacer diferentes combinaciones de casos donde deseamos las condiciones de continuidad:

- a) Las condiciones de continuidad están sobre la función solamente.
- b) Las condiciones de continuidad están sobre la función y en la primera derivada.
- c) Las condiciones de continuidad están sobre la primera derivada.
- d) Las condiciones de continuidad están sobre la primera y segunda derivada de la función de aproximación.
- e) Las condiciones de continuidad están sobre la función y segunda derivada.
- f) Las condiciones de continuidad están sobre la segunda derivada.

Ejemplo:

Se tiene la siguiente gráfica

x	$f(x)$
x_0	$f(x_0)$
x_1	$f(x_1)$
x_2	$f(x_2)$
x_3	$f(x_3)$
x_4	$f(x_4)$
x_5	$f(x_5)$
x_6	$f(x_6)$
x_7	$f(x_7)$
x_8	$f(x_8)$
x_9	$f(x_9)$



El polinomio a obtener es $S_f(x) = a_f + b_f(x - x_M) + c_f(x - x_M)^2 + d_f(x - x_M)^3$ en el subintervalo $[x_M, x_{M+1}]$ donde

a) Las condiciones de continuidad están sobre la función solamente.

Evaluamos la aproximación en los puntos x_0, x_1, x_2, x_3

$$S_0(x_0) = a_0 + b_0(x_0 - x_0) + c_0(x_0 - x_0)^2 + d_0(x_0 - x_0)^3 = f(x_0)$$

$$S_0(x_1) = a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$S_0(x_2) = a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 + d_0(x_2 - x_0)^3 = f(x_2)$$

$$S_0(x_3) = a_0 + b_0(x_3 - x_0) + c_0(x_3 - x_0)^2 + d_0(x_3 - x_0)^3 = f(x_3)$$

Evaluamos la aproximación en los puntos x_3, x_4, x_5, x_6

$$S_1(x_3) = a_1 + b_1(x_3 - x_3) + c_1(x_3 - x_3)^2 + d_1(x_3 - x_3)^3 = f(x_3)$$

$$S_1(x_4) = a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$S_1(x_5) = a_1 + b_1(x_5 - x_3) + c_1(x_5 - x_3)^2 + d_1(x_5 - x_3)^3 = f(x_5)$$

$$S_1(x_6) = a_1 + b_1(x_6 - x_3) + c_1(x_6 - x_3)^2 + d_1(x_6 - x_3)^3 = f(x_6)$$

Evaluamos la aproximación en los puntos x_6, x_7, x_8, x_9

$$S_2(x_6) = a_2 + b_2(x_6 - x_6) + c_2(x_6 - x_6)^2 + d_2(x_6 - x_6)^3 = f(x_6)$$

$$S_2(x_7) = a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$S_2(x_8) = a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$S_2(x_9) = a_2 + b_2(x_9 - x_6) + c_2(x_9 - x_6)^2 + d_2(x_9 - x_6)^3 = f(x_9)$$

Donde el sistema de ecuaciones queda de la siguiente forma

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 + d_0(x_2 - x_0)^3 = f(x_2)$$

$$a_0 + b_0(x_3 - x_0) + c_0(x_3 - x_0)^2 + d_0(x_3 - x_0)^3 = f(x_3)$$

$$a_1 = f(x_3)$$

$$a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$a_1 + b_1(x_5 - x_3) + c_1(x_5 - x_3)^2 + d_1(x_5 - x_3)^3 = f(x_5)$$

$$a_1 + b_1(x_6 - x_3) + c_1(x_6 - x_3)^2 + d_1(x_6 - x_3)^3 = f(x_6)$$

$$a_2 = f(x_6)$$

$$a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$a_2 + b_2(x_9 - x_0) + c_2(x_9 - x_0)^2 + d_2(x_9 - x_0)^3 = f(x_9)$$

Resolviendo cada sistema de ecuaciones obtenemos los coeficientes de los polinomios cúbicos S_0 , S_1 , S_2

b) Las condiciones de continuidad están sobre la función y en la primera derivada.

Se tiene el polinomio cúbico es $S_i(x) = a_i + b_i(x - x_{3i}) + c_i(x - x_{3i})^2 + d_i(x - x_{3i})^3$ y primera derivada.

$$S_i'(x) = b_i + 2c_i(x - x_{3i}) + 3d_i(x - x_{3i})^2$$

Evaluamos la aproximación en los puntos x_0 , x_1 , x_2 , x_3

$$S_0(x_0) = a_0 + b_0(x_0 - x_0) + c_0(x_0 - x_0)^2 + d_0(x_0 - x_0)^3 = f(x_0)$$

$$S_0(x_1) = a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$S_0(x_2) = a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 + d_0(x_2 - x_0)^3 = f(x_2)$$

$$\frac{dS_0(x_2)}{dx} = b_0 + 2c_0(x_2 - x_0) + 3d_0(x_2 - x_0)^2 = \frac{dS_1(x_2)}{dx}$$

Evaluamos la aproximación en los puntos x_3 , x_4 , x_5 , x_6

$$S_1(x_3) = a_1 + b_1(x_3 - x_3) + c_1(x_3 - x_3)^2 + d_1(x_3 - x_3)^3 = f(x_3)$$

$$S_1(x_4) = a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$S_1(x_5) = a_1 + b_1(x_5 - x_3) + c_1(x_5 - x_3)^2 + d_1(x_5 - x_3)^3 = f(x_5)$$

$$\frac{dS_1(x_6)}{dx} = b_1 + 2c_1(x_6 - x_3) + 3d_1(x_6 - x_3)^2 = \frac{dS_2(x_6)}{dx}$$

Evaluamos la aproximación en los puntos x_6 , x_7 , x_8 , x_9

$$S_2(x_6) = a_2 + b_2(x_6 - x_6) + c_2(x_6 - x_6)^2 + d_2(x_6 - x_6)^3 = f(x_6)$$

$$S_2(x_7) = a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$S_2(x_8) = a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$S_2(x_9) = a_2 + b_2(x_9 - x_6) + c_2(x_9 - x_6)^2 + d_2(x_9 - x_6)^3 = f(x_9)$$

Donde el sistema de ecuaciones queda de la siguiente forma

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 + d_0(x_2 - x_0)^3 = f(x_2)$$

$$b_0 + 2c_0(x_2 - x_0) + 3d_0(x_2 - x_0)^2 - b_1 = 0$$

$$a_1 = f(x_1)$$

$$a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$a_1 + b_1(x_5 - x_3) + c_1(x_5 - x_3)^2 + d_1(x_5 - x_3)^3 = f(x_5)$$

$$b_1 + 2c_1(x_6 - x_3) + 3d_1(x_6 - x_3)^2 - b_2 = 0$$

$$a_2 = f(x_6)$$

$$a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$a_2 + b_2(x_9 - x_6) + c_2(x_9 - x_6)^2 + d_2(x_9 - x_6)^3 = f(x_9)$$

e) Las condiciones de continuidad están sobre la primera derivada.

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 + d_0(x_2 - x_0)^3 = f(x_2)$$

$$b_0 + 2c_0(x_3 - x_0) + 3d_0(x_3 - x_0)^2 = f'(x_3)$$

$$b_1 = f'(x_3)$$

$$a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$a_1 + b_1(x_5 - x_3) + c_1(x_5 - x_3)^2 + d_1(x_5 - x_3)^3 = f(x_5)$$

$$b_1 + 2c_1(x_6 - x_3) + 3d_1(x_6 - x_3)^2 = f'(x_6)$$

$$b_2 = f'(x_6)$$

$$a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$a_2 + b_2(x_9 - x_6) + c_2(x_9 - x_6)^2 + d_2(x_9 - x_6)^3 = f(x_9)$$

d) Las condiciones de continuidad están sobre la primera y segunda derivada de la función de aproximación.

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$b_0 + 2c_0(x_3 - x_0) + 3d_0(x_3 - x_0)^2 - b_1 = 0$$

$$2c_0 + 6d_0(x_3 - x_0) - 2c_1 = 0$$

$$a_1 = f(x_3)$$

$$a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$b_1 + 2c_1(x_6 - x_3) + 3d_1(x_6 - x_3)^2 - b_2 = 0$$

$$2c_1 + 6d_1(x_6 - x_3) - 2c_2 = 0$$

$$a_2 = f(x_6)$$

$$a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$a_2 + b_2(x_9 - x_6) + c_2(x_9 - x_6)^2 + d_2(x_9 - x_6)^3 = f(x_9)$$

e) Las condiciones de continuidad están sobre la función y segunda derivada.

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 + d_0(x_2 - x_0)^3 = f(x_2)$$

$$2c_0 + 6d_0(x_1 - x_0) - 2c_1 = 0$$

$$a_1 = f(x_3)$$

$$a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$a_1 + b_1(x_5 - x_3) + c_1(x_5 - x_3)^2 + d_1(x_5 - x_3)^3 = f(x_5)$$

$$2c_1 + 6d_1(x_4 - x_3) - 2c_2 = 0$$

$$a_2 = f(x_6)$$

$$a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$a_2 + b_2(x_9 - x_6) + c_2(x_9 - x_6)^2 + d_2(x_9 - x_6)^3 = f(x_9)$$

f) Las condiciones de continuidad están sobre la segunda derivada.

$$a_0 = f(x_0)$$

$$a_0 + b_0(x_1 - x_0) + c_0(x_1 - x_0)^2 + d_0(x_1 - x_0)^3 = f(x_1)$$

$$a_0 + b_0(x_2 - x_0) + c_0(x_2 - x_0)^2 + d_0(x_2 - x_0)^3 = f(x_2)$$

$$2c_0 + 6d_0(x_1 - x_0) = f''(x_1)$$

$$2c_1 = f''(x_3)$$

$$a_1 + b_1(x_4 - x_3) + c_1(x_4 - x_3)^2 + d_1(x_4 - x_3)^3 = f(x_4)$$

$$a_1 + b_1(x_5 - x_3) + c_1(x_5 - x_3)^2 + d_1(x_5 - x_3)^3 = f(x_5)$$

$$2c_1 + 6d_1(x_4 - x_3) = f''(x_6)$$

$$2c_2 = f''(x_6)$$

$$a_2 + b_2(x_7 - x_6) + c_2(x_7 - x_6)^2 + d_2(x_7 - x_6)^3 = f(x_7)$$

$$a_2 + b_2(x_8 - x_6) + c_2(x_8 - x_6)^2 + d_2(x_8 - x_6)^3 = f(x_8)$$

$$a_2 + b_2(x_9 - x_6) + c_2(x_9 - x_6)^2 + d_2(x_9 - x_6)^3 = f(x_9)$$

Como hemos visto, el método de spline es muy flexible para casos de continuidad en los nodos de conexión. Nos permite seleccionar las características de continuidad del polinomio interpolante con base a las características de continuidad de la función y puede aumentar su precisión aumentando el número de datos muestrales.

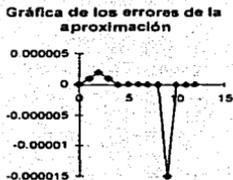
Conclusiones.

Cuando realizamos la interpolación cúbica segmentaria, observamos que, de acuerdo a las condiciones de continuidad que deseamos, la interpolación nos da un cierto error. El error no lo podemos evitar ya que estamos haciendo una aproximación a la función original, este error puede variar dependiendo de las condiciones de continuidad que nosotros hayamos pedido.

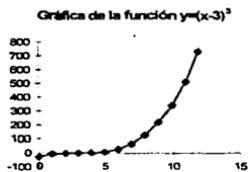
observemos los siguientes ajustes:

Ajuste de la función $y=(x-3)^3$

a) Las condiciones de continuidad están sobre la función solamente.

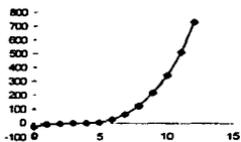


b) Las condiciones de continuidad están sobre la función y primera derivada.

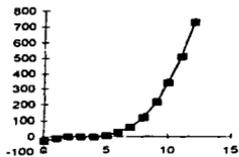


c) Las condiciones de continuidad están sobre la primera derivada.

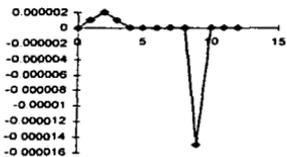
Gráfica de la función $y=(x-3)^2$



Gráfica del ajuste de la función $y=(x-3)^2$

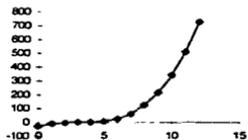


Gráfica de los errores del ajuste

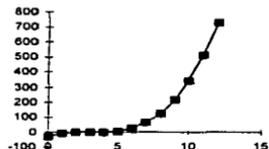


d) Las condiciones de continuidad están sobre la primera y segunda derivada.

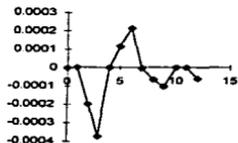
Gráfica de la función $y=(x-3)^2$



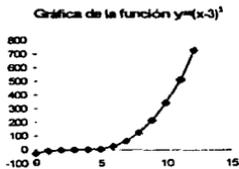
Gráfica del ajuste de la función $y=(x-3)^2$



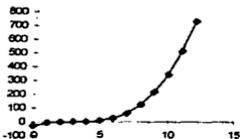
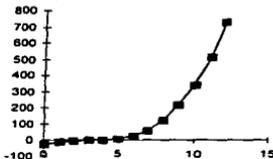
Gráfica del error del ajuste de la función



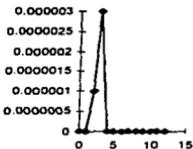
e) Las condiciones de continuidad están sobre la función y segunda derivada.



D) Las condiciones de continuidad están sobre la segunda derivada.

Gráfica de la función $y=(x-3)^3$ Gráfica del ajuste de la función $y=(x-3)^3$ 

Gráfica de los errores del ajuste



Hemos observado que en cualquier condición de continuidad que nosotros escojamos, el ajuste de la función es perfecto, esto se debe a que el polinomio de aproximación que realiza el ajuste es un polinomio cúbico, y como la función a ajustar es una función cúbica no representa mayor problema al realizar el ajuste.

Observemos el siguiente ajuste de la función $y=(x-3)^3$ con los diferentes casos de continuidad.

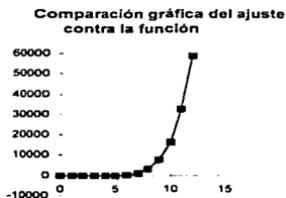
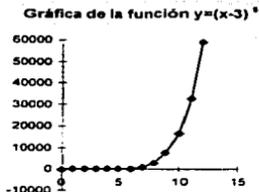
Ajuste de la función $y=(x-3)^5$.

a) Las condiciones de continuidad están sobre la función solamente.



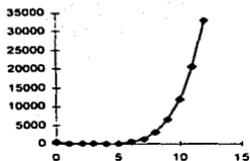
Observando las gráficas podemos concluir que el ajuste es perfecto debido a que la condición de continuidad está sobre la función, entonces el polinomio de aproximación está obligando a que en los nodos de conexión se cumpla esta condición.

b) Las condiciones de continuidad están sobre la función y la primera derivada.

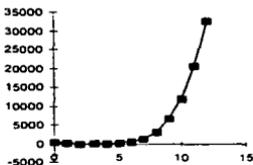


Continuación del caso b)

**Gráfica de la derivada
y = 5(x-3)² de la función**



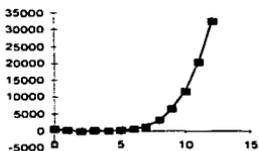
**Gráfica de la derivada del
ajuste**



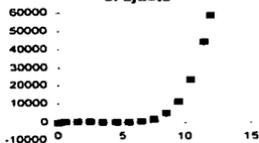
**Gráfica de los errores de la
derivada del ajuste**



**Comparación gráfica de la
derivada de la función contra
la derivada del ajuste**



**Comparación Gráfica de
puntos intermedios
evaluados en la función y en
el ajuste**



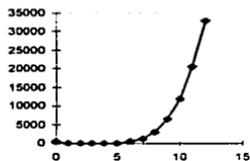
Observando las gráficas del ajuste podemos concluir que hay un error relativamente grande en éste, pero siguiendo siempre la tendencia de la función. También podemos observar que conforme aumenta el valor de x el error se va haciendo más grande. Todo esto se puede apreciar también en el ajuste de las derivadas.

c) Las condiciones de continuidad están sobre la primera derivada.

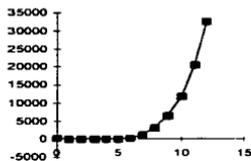


Continuación del caso c)

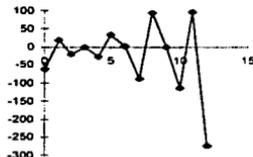
**Gráfica de la derivada
 $y=5(x-3)^2$ de la función**



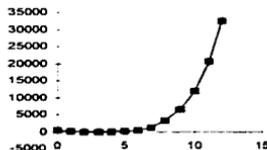
**Gráfica de la derivada del
ajuste**



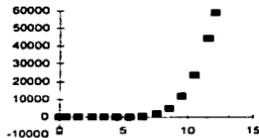
**Gráfica de errores de la
derivada del ajuste**



**Comparación gráfica de la
derivada de la función contra
la derivada del ajuste**



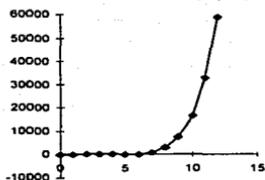
**Comparación gráfica de
puntos intermedios
evaluados en la función y en
el ajuste**



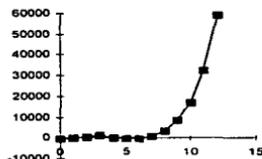
Al igual que en la condición pasada el error es relativamente grande y va creciendo conforme avanza la gráfica pero debemos observar en la gráfica de errores de la derivada del ajuste que no hay error en los nodos de conexión, debido a que se está cumpliendo la condición de continuidad en la primera derivada.

d) Las condiciones de continuidad están sobre la primera y segunda derivada.

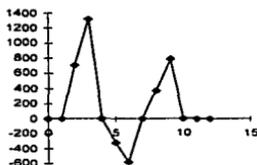
Gráfica de la función $y=(x-3)^4$



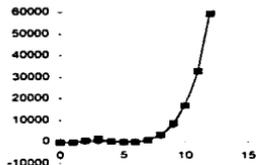
Gráfica del ajuste de la función $y=(x-3)^4$



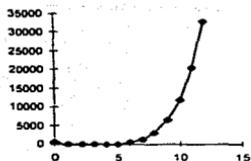
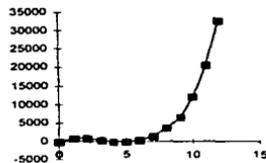
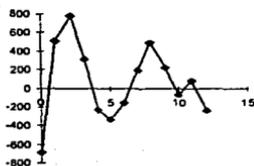
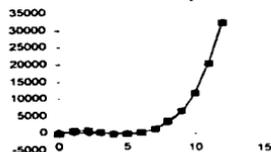
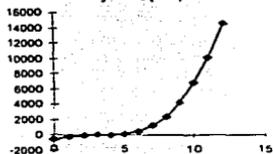
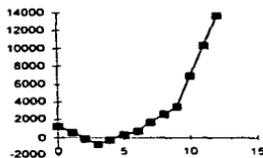
Gráfica de los errores del ajuste



Comparación gráfica del ajuste contra la función

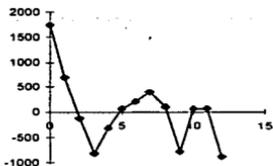


Continuación del caso d)

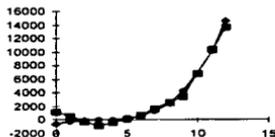
Gráfica de la derivada
 $y' = 5(x-3)^2$ de la funciónGráfica de la derivada del
ajusteGráfica de errores del ajuste
de la primera derivadaComparación Gráfica de la
derivada de la función contra
la derivada del ajusteGráfica de la segunda
derivada de la función
 $y'' = 20(x-3)$ Gráfica de la segunda
derivada del ajuste

Continuación del caso d)

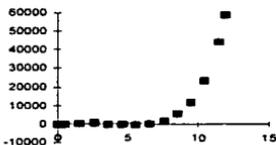
Gráfica de los errores de la segunda derivada del ajuste



Comparación gráfica de la segunda derivada de la función contra la segunda derivada del ajuste



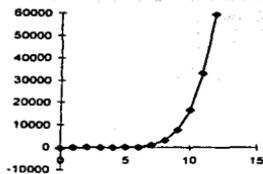
Gráfica de puntos intermedios evaluados en la función y en el ajuste



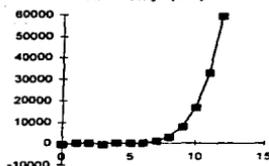
En este ajuste podemos observar que el error es grande en puntos de conexión. Esto es debido a que al polinomio de aproximación se le están pidiendo que las condiciones de continuidad se cumplan tanto en primera como en segunda derivada. Al observar las gráficas de las primeras derivadas podemos también notar que el error empieza a ser grande en valores pequeños de x , pero luego se estabiliza. También ocurre lo mismo en las gráficas de la segunda derivada, pero no tiene tendencia monótonica.

e) Las condiciones de continuidad están sobre la función y segunda derivada.

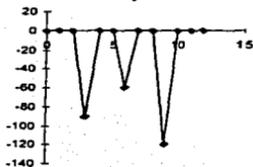
Gráfica de la función $y=(x-3)^4$



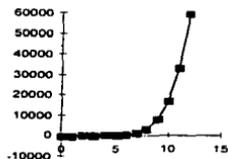
Gráfica del ajuste de la función $y=(x-3)^4$



Gráfica de los errores del ajuste

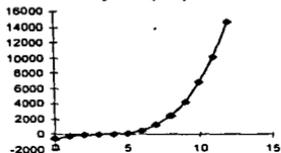


Comparación gráfica de la función contra el ajuste

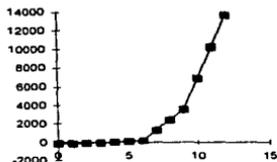


Continuación del caso c)

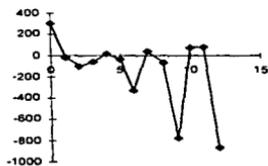
Gráfica de la segunda derivada de la función $y''=20(x-3)^2$



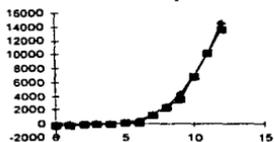
Gráfica de la segunda derivada del ajuste



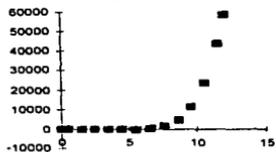
Gráfica de errores de la segunda derivada del ajuste



Comparación gráfica de la segunda derivada de la función contra la segunda derivada del ajuste



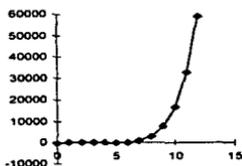
Gráfica de puntos intermedios evaluados en la función y en el ajuste



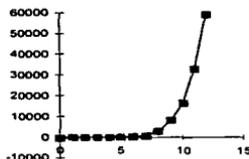
De lo anterior, podemos concluir que el error del ajuste va creciendo conforme va aumentando la gráfica de éste. Es muy importante notar que el error solo se presenta en los nodos de conexión debido a que es allí donde se están pidiendo las condiciones de continuidad. Las gráficas del ajuste de la segunda derivada también presentan el mismo caso pero el error es menos que en los nodos de conexión.

f) Las condiciones de continuidad están sobre la segunda derivada.

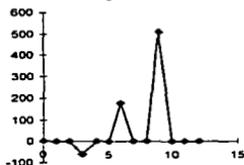
Gráfica de la función $y=(x-3)^3$



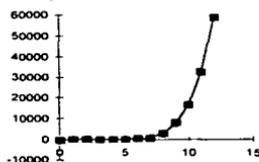
Gráfica del ajuste de la función $y=(x-3)^3$



Gráfica de los errores del ajuste

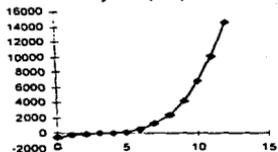


Comparación gráfica del ajuste contra la función

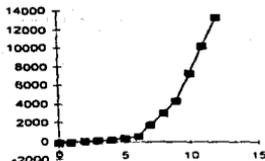


Continuación del caso f)

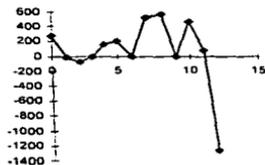
Gráfica de la segunda derivada de la función
 $y''=20(x-3)^2$



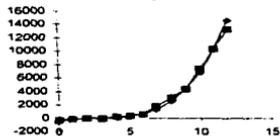
Gráfica de la segunda derivada del ajuste



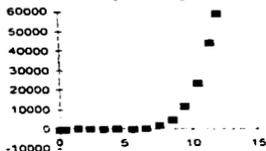
Gráfica de los errores de la segunda derivada del ajuste



Comparación gráfica de la segunda derivada de la función contra la segunda derivada del ajuste



Gráfica de puntos intermedios evaluados en la función y en el ajuste



De las gráficas del ajuste de la función observamos que el error aumenta en los nodos de conexión del caso f), esto se debe a que la condición de continuidad está sobre la segunda derivada. En las gráficas de la segunda derivada se observa que no hay error en los nodos de conexión, pero el error es muy irregular en los demás nodos.

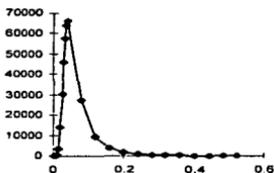
Se puede concluir de manera general para el ajuste de la función $y=(x-3)^5$ con sus diferentes casos de continuidad que; si deseamos aproximar mas puntos de la función, el error del ajuste se va a ir incrementando; esto se debe a que la función como es de orden 5 crece muy rápidamente, entonces dado que la interpolación es cúbica no alcanza a aproximar valores muy grandes entre uno y otro.

Analicemos el siguiente ajuste de la función $y = \frac{1}{x^3(e^{1/3^x} - 1)}$ con sus diferentes casos de continuidad.

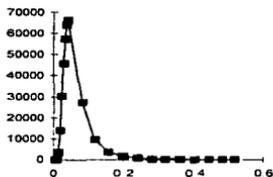
Ajuste de la función $y = \frac{1}{x^3(e^{1/x^2} - 1)}$.

a) Las condiciones de continuidad están sobre la función.

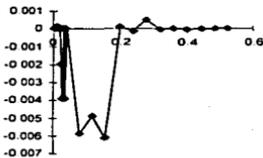
Gráfica de la función



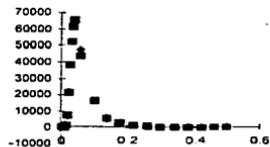
Gráfica del ajuste de la función



Gráfica de los errores del ajuste



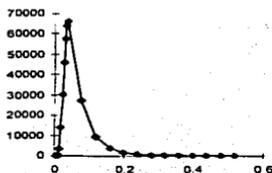
Comparación gráfica de puntos intermedios evaluados en la función y en el ajuste



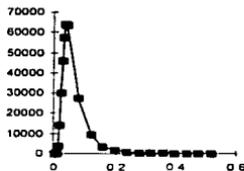
El ajuste no tiene errores ya que las condiciones de continuidad están sobre la función, pero observando la gráfica de puntos intermedios podemos ver que ya hay algunas irregularidades en éstos.

b) Las condiciones de continuidad están sobre la función y la primera derivada.

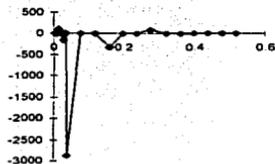
Gráfica de la función



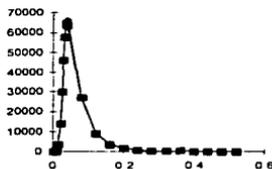
Gráfica del ajuste de la función



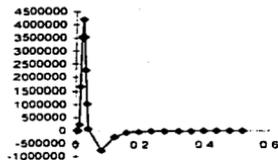
Gráfica de los errores del ajuste



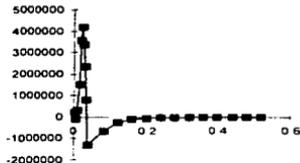
Comparación gráfica del ajuste contra la función



Gráfica de la derivada de la función

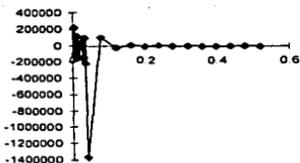


Gráfica de la derivada del ajuste

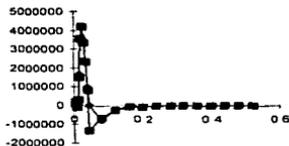


Continuación del caso b)

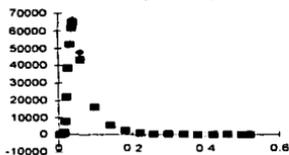
Gráfica de los errores de la derivada del ajuste



Comparación gráfica de la derivada de la función contra la derivada del ajuste



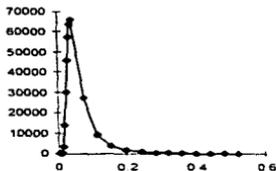
Gráfica de comparación de puntos intermedios evaluados en la función y en el ajuste



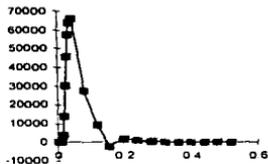
Observando este ajuste podemos ver que el error es relativamente grande en los primeros nodos de conexión, pero después se empieza a normalizar. Por otra parte observando la gráfica del ajuste en la derivada podemos ver el mismo comportamiento del error. También se puede observar que los ajustes tanto en la función como en la derivada tienen un comportamiento casi igual que a las funciones. Los puntos intermedios evaluados en el ajuste siguen presentando inestabilidad.

e) Las condiciones de continuidad están sobre la primera derivada.

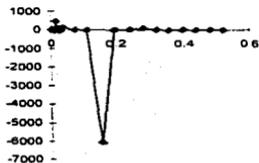
Gráfica de la función



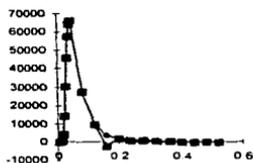
Gráfica del ajuste de la función



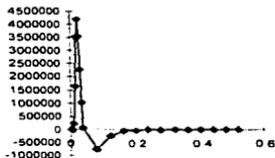
Gráfica de los errores del ajuste



Comparación gráfica del ajuste contra la función



Gráfica de la derivada de la función

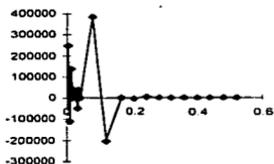


Gráfica de la derivada del ajuste

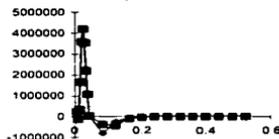


Continuación del caso c)

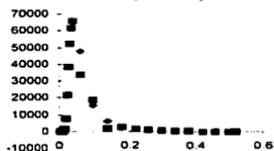
Gráfica de los errores de la derivada del ajuste



Comparación gráfica de la derivada de la función contra la derivada del ajuste



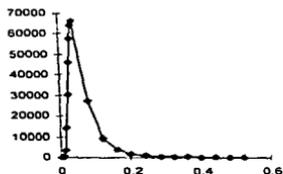
Comparación gráfica de puntos intermedios evaluados en la función y en el ajuste



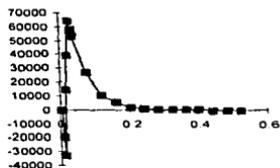
Observando vemos que el error en el ajuste es relativamente grande al principio pero después decrece. En las gráficas del ajuste en la primera derivada se puede observar que en los nodos de conexión no hay error debido a la condición de continuidad en la primera derivada. Los puntos intermedios evaluados en el ajuste siguen presentando inestabilidad.

d) Las condiciones de continuidad están sobre la primera y segunda derivada.

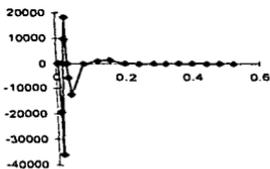
Gráfica de la función



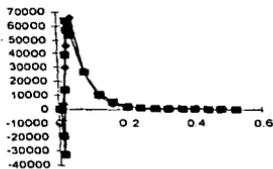
Gráfica del ajuste de la función



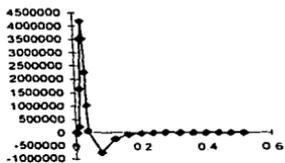
Gráfica de los errores del ajuste



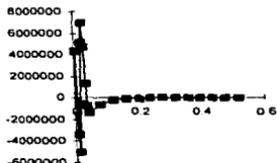
Comparación gráfica del ajuste contra la función



Gráfica de la derivada de la función

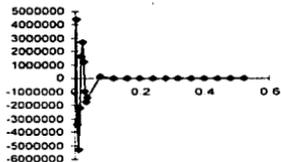


Gráfica de la derivada del ajuste

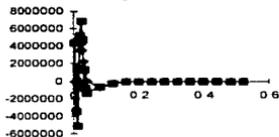


Continuación del caso d)

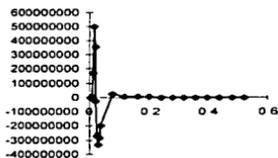
Gráfica de los errores de la derivada del ajuste



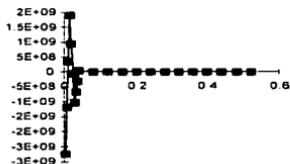
Comparación gráfica de la derivada de la función contra la derivada del ajuste



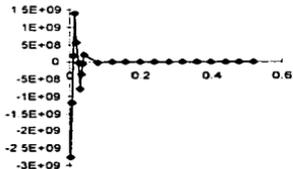
Gráfica de la segunda derivada de la función



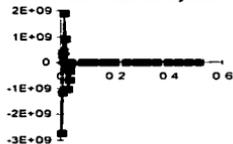
Gráfica de la segunda derivada del ajuste



Gráfica de los errores de la segunda derivada del ajuste



Comparación gráfica, de la segunda derivada de la función contra la segunda derivada del ajuste



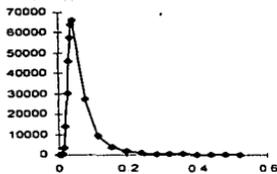
Continuación del caso d)



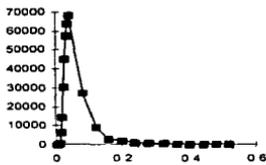
En este caso de aproximación, donde las condiciones de continuidad están sobre la primera y segunda derivadas el ajuste tiene un error muy grande tanto en la función como en dichas derivadas. Esto se debe a que el polinomio de aproximación fuerza a que se cumplan estas dos condiciones, y porque las derivadas del polinomio se vuelven más pequeñas, mientras que los términos que aparecen en las derivadas de la función van aumentando conforme se incrementa el orden de la derivada.

e) Las condiciones de continuidad están sobre la función y segunda derivada.

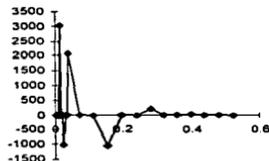
Gráfica de la función



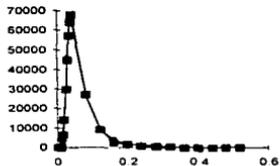
Gráfica del ajuste de la función



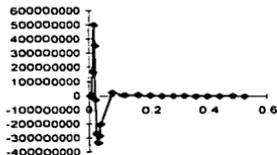
Gráfica de los errores del ajuste



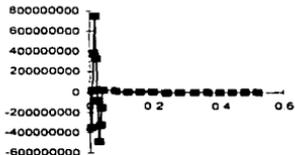
Comparación gráfica del ajuste contra la función



Gráfica de la segunda derivada de la función

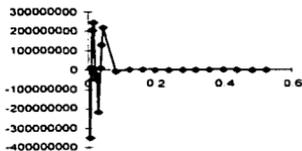


Gráfica de la segunda derivada del ajuste

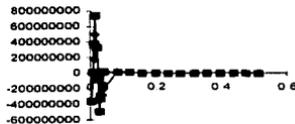


Continuación del caso e)

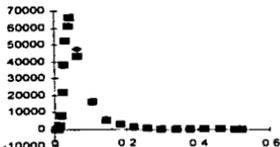
Gráfica de los errores de la segunda derivada del ajuste



Comparación gráfica de la segunda derivada de la función contra la segunda derivada del ajuste



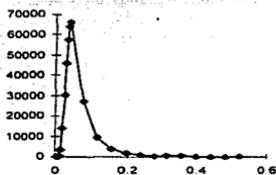
Comparación gráfica de los puntos intermedios evaluados en la función y en el ajuste



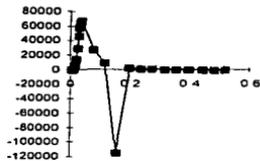
Analizando, tenemos que el error del ajuste es grande pero después se comienza a estabilizar. Esto es debido a la condición de continuidad que está sobre la función y sobre la segunda derivada. El error se debe también a que la segunda derivada del polinomio de aproximación es una línea recta mientras que la segunda derivada de la función crece con respecto a ésta. Esto se puede ver en el ajuste de la segunda derivada, en el cual, al principio hay un error grande en la aproximación. Al observar la gráfica de puntos intermedios nos damos cuenta que hay inestabilidad en algunos puntos.

f) Las condiciones de continuidad están sobre la segunda derivada.

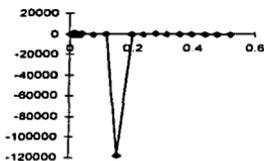
Gráfica de la función



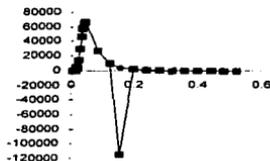
Gráfica del ajuste de la función



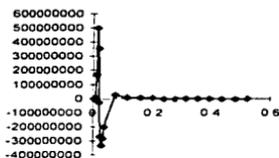
Gráfica de los errores del ajuste



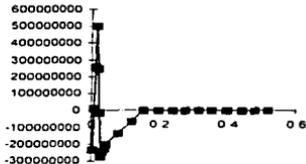
Comparación gráfica de la función contra el ajuste



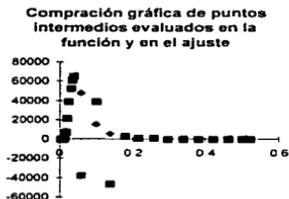
Gráfica de la segunda derivada de la función



Gráfica de la segunda derivada del ajuste



Continuación del caso f)



Observando este ajuste se aprecia que hay un error muy grande donde se impone la condición de la segunda derivada, pero al ver el error del ajuste en ésta observamos que en los nodos de conexión no hay error. También se puede apreciar que conforme se va dando el ajuste, los errores se empiezan a estabilizar. En la gráfica de puntos intermedios se hace mas notoria la inestabilidad de los puntos evaluados en el ajuste.

Para tener una mejor idea del ajuste de la función $y = \frac{1}{x^2(x^{1/2} - 1)}$ observemos las siguientes tablas donde se muestran los diferentes casos de continuidad, la tendencia de la y ajustada y los errores de los diferentes ajustes.

Tabla Comparativa de la Tendencia de las y Ajustadas contra las y Reales

TIPO DE CONTINUIDAD	TENDENCIA DE LAS y AJUSTADAS CONTRA LAS y REALES
Continuidad en la función	Tendencia Igual en el ajuste de la función.
Continuidad en la función y en la primera derivada	Tendencias parecidas tanto en el ajuste de la función como en el de la primera derivada.
Continuidad en la primera derivada	Tendencias parecidas tanto en el ajuste de la función como en el de la primera derivada.
Continuidad en la primera y segunda derivada	Tendencias inestables al comienzo del ajuste; tanto en el ajuste de la función como en el de la primera y segunda derivadas.
Continuidad en la función y en la segunda derivada	Tendencia parecida en el ajuste de la función. Tendencia inestable al comienzo del ajuste de la segunda derivada.
Continuidad en la segunda derivada	Tendencia parecida en el ajuste de la función. (solo hay un punto inestable). Tendencia inestable al comienzo del ajuste de la segunda derivada.

TABLA DE COMPARACIÓN DE LOS ERRORES DEL AJUSTE

TIPO DE CONTINUIDAD	Error de $f(x)$	Error de $f'(x)$	Error de $f''(x)$
Continuidad en la Función	Error insignificante		
Continuidad en la función y en la primera derivada	Pequeño error	Mayor error	
Continuidad en primera derivada	Pequeño error	Mayor error	
Continuidad en primera y segunda derivada	Mayor error	Mayor error	Mayor error
Continuidad en función y segunda derivada	Pequeño error		Mayor error
Continuidad en segunda derivada	Pequeño error		Mayor error

Hemos observado en los resultados obtenidos que para funciones trascendentes* no es recomendable el uso de spline para las condiciones de continuidad en las derivadas, puesto que al derivar estas funciones, en lugar de ayudar a tener una mejor aproximación (liga, conexión) en los nodos de conexión nos da un error muy grande entre el valor observado y el valor de la curva aproximante. Esto se debe a que dichas funciones cambian su comportamiento abruptamente de un segmento a otro. Al derivar estas funciones, en vez de que disminuya el número de términos algebraicos o disminuya su orden, estos pueden aumentar, provocando que los valores en los nodos de conexión sean inexactos.

Se concluye que, para funciones trascendentes se puede utilizar la interpolación cúbica segmentaria (spline) con la condición de continuidad solamente en la función; Además observando la gráfica de la función y las gráficas de sus derivadas, se puede tener una mejor idea de cuales condiciones de continuidad son las que nos convienen.

* Las funciones trascendentes son aquellas en las cuales el argumento y la función no pueden ser relacionadas por una dependencia algebraica del tipo

$$\sum_{i=1}^n a_i x^i y^m = 0$$

Todas las combinaciones posibles de las funciones trascendentes y algebraicas enumeradas, cuando una función puede ser argumento de la otra, dan las funciones trascendentes compuestas o funciones de funciones, por ejemplo $y = \ln(x)$, $y = \sin(x) + 3x + \cos(2x)$, etc. Manual de Matemáticas para Ingenieros y estudiantes; Y. Bronshtein, K. Semendiaev, (2 Edición); Editorial Mir, Moscú.

2.3 MÍNIMOS CUADRADOS NO LINEALES.

El problema de mínimos cuadrados no lineales es encontrar una minimización global de la suma de cuadrados de m observaciones usando n parámetros no lineales, es decir

$$\text{Minimizar } r^2 = \sum_{i=1}^m \left[y_i - \sum_{j=1}^n y(x_i; c_j) \right]^2, \quad (2.3.1)$$

donde solo las y_i están sujetas a los errores y los valores x_i de la variable independiente son exactos.

Los métodos de solución para problemas de mínimos cuadrados no lineales trabajan con la matriz jacobiana y la matriz hessiana de r^2 con respecto al parámetro c_j , ($j = 1, 2, \dots, n$). Suponemos que r^2 es 2 veces continuamente diferenciable, entonces la j -ésima componente del gradiente de (2.3.1) es

$$[\nabla r^2(c)]_j = \frac{\partial r^2}{\partial c_j} = -2 \sum_{i=1}^m \left(y_i - \sum_{j=1}^n y(x_i; c_j) \right) \frac{\partial y(x_i; c_j)}{\partial c_j}, \quad j = 1, 2, \dots, n \quad (2.3.2)$$

Reescribiendo la ecuación (2.3.2) en términos de la matriz jacobiana tenemos que

$$\nabla r^2(c) = -2 \sum_{i=1}^m J^T(c) r_i(c) \quad (2.3.3)$$

y el hessiano de (2.3.1) es

$$H(c) = \frac{\partial^2 r^2}{\partial c_j \partial c_k} = 2 \sum_{i=1}^m \left[\frac{\partial y(x_i; c)}{\partial c_j} \frac{\partial y(x_i; c)}{\partial c_k} - \left(y_i - \sum_{j=1}^n y(x_i; c) \right) \frac{\partial^2 y(x_i; c)}{\partial c_j \partial c_k} \right] \quad (2.3.4)$$

Reescribiendo esta ecuación en términos de la matriz hessiana tenemos lo siguiente

$$H(c) = 2 \sum_{i=1}^m [J^T(c) J(c) - r_i(c) \nabla^2 r_i^2(c)] \quad (2.3.5)$$

El problema de mínimos cuadrados no lineales se tiene que resolver mediante métodos iterativos, hasta encontrar los parámetros óptimos. Para la obtención de estos parámetros existen métodos tales como el del Gradiente (ó Descenso mas rápido ó Descenso Acelerado), el de Newton, el de Gauss-Newton, y los Métodos Cuasi-Newton (Cálculo del jacobino por diferencias finitas; por una secante). Estos métodos son utilizados

generalmente para resolver sistemas de ecuaciones no lineales y, por su puesto, no son tan exactos en su solución como en mínimos cuadrados lineales.

Una condición para que r^2 tenga un mínimo es que $\nabla r^2(c) = 0$ para todo parámetro c_j , $j = 1, 2, \dots, n$. de lo anterior tenemos que el método de Descenso acelerado es:

$$c_{k+1} = c_k - \alpha \nabla r(c_k), \quad (2.3.6)$$

donde el incremento ó tamaño de paso es $\delta c = c_{k+1} - c_k$ y el factor de amortiguamiento es α ; de esto el problema de mínimos cuadrados no lineales (2.3.1) se puede resolver mediante el método de Gauss-Newton

$$[J^T(c)J(c)]\delta c = J^T(c)r(c). \quad (2.3.7)$$

Observemos que este método tiene relación con la matriz hessiana (2.3.5) solo que el método de Gauss-Newton ignora el segundo término de dicha matriz; que es $-r(c)\nabla^2 r^2(c)$. Nosotros haremos algunos comentarios al respecto.

Las segundas derivadas ocurren porque el gradiente (2.3.3) tiene una dependencia en $\frac{\partial y}{\partial x_j}$, que es variable para el caso no lineal. El producto $-r(c)\nabla^2 r^2(c)$ puede ser despreciado cuando los residuales son bastante pequeños, ó insignificantes cuando son comparados al término que involucra la primera derivada. Para el modelo a minimizar, dicho producto representa una medida de error aleatorio en cada punto. Este error puede tener uno u otro signo, por lo tanto, el producto de la segunda derivada tiende a cancelarse cuando es sumado sobre i , o la inclusión del producto de la segunda derivada puede en realidad ser inestable si el modelo se ajusta mal. De lo anterior, en el contexto de mínimos cuadrados no lineales, los primeros términos de la matriz hessiana $H(c)$, $(J^T(c)J(c))$ son generalmente llamados, la matriz *curvatura*.

Analizaremos a continuación un algoritmo llamado Levenberg-Marquardt (Combinación del método de Gauss-Newton con Descenso Acelerado) para tener un mejor resultado de los parámetros a minimizar.

Método de Levenberg-Marquardt.

Marquardt ha propuesto un método elegante para variar suavemente entre los extremos del método de Gauss-Newton $[J^T(c)J(c)]\delta c = J^T(c)r(c)$ y el método de Descenso Acelerado $\delta c = -\alpha \nabla r(c)$. Este método empieza con Gauss-Newton, si empieza a fallar pasa al método de Descenso Acelerado y hasta que esta cerca de la aproximación lineal entonces cambia a Gauss-Newton otra vez.

Este método esta basado en 2 elementales pero importantes observaciones. La primera observación esta relacionada con la escala del problema; consideremos la constante α de la fórmula de Descenso Acelerado $c_{n+1} = c_n - \alpha \nabla r(c_n)$ ¿de qué orden de magnitud será?, ¿qué conjunto de escala tiene?. La primera observación de Marquardt es que los componentes de la matriz del método de Gauss-Newton $(J^T(c)J(c))$, dan alguna información acerca del orden de magnitud de la escala del problema. La cantidad r^2 es adimensional, es decir, es solamente un número; esto es evidente de la definición (2.3.1). Por otro lado, los términos de $J^T(c)r(c)$ pueden tener dimensiones tales como cm, kilowatts por hora, etc. (En realidad, cada componente de $J^T(c)r(c)$ puede tener diferentes dimensionalidades). La constante de proporcionalidad ó factor de amortiguamiento α entre $J^T(c)r(c)$ y δc_j debe tener dimensiones de c_j . Una forma de asignar la escala de la matriz $J^T(c)J(c)$ es multiplicar los elementos de la diagonal principal por su recíproco es decir por $\frac{1}{\sqrt{(J^T(c)J(c))_{jj}}}$. Pero la escala misma puede ser también grande. Dividimos la constante por algún factor adimensional λ , con la posibilidad de poner $\lambda \gg 1$ acortando el paso. En otras palabras reemplazando la ecuación de Descenso Acelerado (2.3.6) por

$$\lambda (J^T(c)J(c))_{jj} \delta c_j = J^T(c)r(c) \quad (2.3.8)$$

La segunda observación de Marquardt se refiere a la combinación de las ecuaciones (2.3.7) y (2.3.8), es decir, a tener de una forma mas clara y precisa la escala del problema y el factor adimensional λ . La escala del problema la representamos por una matriz D , teniendo la siguiente ecuación

$$[J^T(c)J(c) + \lambda D]\delta c = J^T(c)r(c) \quad (2.3.9)$$

Esta matriz D es una matriz diagonal definida positiva y se puede ver como una analogía a la matriz W de pesos en mínimos cuadrados lineales. Si no deseamos escala, entonces la matriz D es puesta como la matriz identidad. De esta forma, en la ecuación (2.3.9) quedan resumidas tanto la escala del problema como el factor adimensional λ .

Cuando λ es muy grande, la matriz compuesta $[J'(c)J(c) + \lambda D]$ esta forzada a ser diagonalmente dominante, por otro lado, cuando λ tiende a cero, la ecuación (2.3.9) será idéntica a (2.3.7).

La constante λ nos ayuda a utilizar el método Gauss-Newton de una mejor forma, ya que ayuda a retener la inclinación de la pendiente cerca de la dirección de descenso acelerado para prevenir la divergencia. De esta forma en la ecuación (2.3.9) quedan resumidas las 2 observaciones hechas por Marquardt, ya que la matriz D es manejada para la escala del problema y al mismo tiempo el factor λ es manejado para la elección del método de Gauss-Newton o el método de Descenso Acelerado.

Con lo anterior el método de Levenberg-Marquardt queda de la siguiente forma

$$[J^T(c)J(c) + \lambda D]\delta c = J^T(c)r(c) \quad (2.3.10)$$

El siguiente algoritmo resuelve el problema de mínimos cuadrados no lineales por el método de Levenberg-Marquardt.

Algoritmo de Levenberg-Marquardt para la solución de Mínimos Cuadrados no Lineales.

Dado un punto inicial para un conjunto de parámetros c_j , $j = 1, 2, \dots, n$, el algoritmo Levenberg-Marquardt es como sigue:

- 1.- Calcular $r^2(c)$
- 2.- Escoja un valor pequeño para λ , por ejemplo digamos $\lambda = 0.001$
- 3.- Resuelva la ecuación lineal (2.3.10) para δc y calcule $r^2(c + \delta c)$
- 4.- Si $r^2(c + \delta c) \geq r^2(c)$, incremente λ por un factor de 10 (ó por otro factor substancial) y regrese al paso 3.-. En este paso del algoritmo si se cumple la desigualdad incrementamos λ ya que, como vimos anteriormente, para valores grandes de λ el algoritmo no deja que el método diverja, en otras palabras, es como un freno para el método; estamos eligiendo el método del Gradiente.
- 5.- Si $r^2(c + \delta c) < r^2(c)$, decremente λ por un factor de 10, actualizamos la solución experimental (tentativa) $c + \delta c \rightarrow c$ y regresamos al paso 3.-. Al cumplirse esta desigualdad decrementamos λ ya que nos vamos acercando al mínimo, entonces estamos eligiendo el método Gauss-Newton y actualizamos el valor del parámetro c para hacer la próxima comparación de la siguiente iteración.

Es necesario una condición de paro para el método, por ejemplo, el número de iteraciones, la exactitud de la máquina o el límite de redondeo.

Es común que los valores de los parámetros se encuentren alrededor del mínimo pero que nunca lleguen a este, lo cual, en general ocurre, cuando se encuentran en un valle plano de complicada topología. Una falla completa por un pivote pequeño generará una corrección mayor por lo cual es rechazado, el valor de λ es entonces incrementado. Para valores suficientemente grandes de λ , la matriz de la ecuación (2.3.10) es definida positiva y no puede tener pivotes pequeños. De esta manera el método tiende a estar lejano de pivotes nulos, pero el costo es alto cuando los parámetros están cercanos al mínimo en un valle plano realizando descenso acelerado.

Estas consideraciones son importantes en las primeras iteraciones, ya que sugieren que, en la práctica, uno pueda detener la iteración en la primera o segunda ocasión en que r^2 disminuye por una cantidad insignificante (estas pueden ser 0.001 ó 0.0001). Esto solo muestra que no se eligió apropiadamente el valor de λ o que la matriz de escala D no es la apropiada, de otra forma, si después de un paso r^2 se incrementa, dejamos seguir la iteración.

Existen paquetes computacionales robustos que realizan diversos tipos de convergencia para el paro del método, calculan el segundo termino de la matriz hessiana (2.3.4), calculan automáticamente la matriz D , si hay algún valle plano ayudan a salirse de éste, calculan el jacobiano mediante diferencias finitas y chequean si la matriz hessiana es singular. Se pueden utilizar mínimos cuadrados no lineales con restricciones: y otras cosas más.

Utilizaremos el algoritmo **NLSOL**, diseñado especialmente para mínimos cuadrados no lineales y la función (*leastsq*) del ambiente computacional **MATLAB** que realiza el ajuste por medio de mínimos cuadrados no lineales para mejorarlo; ya que todos estos paquetes de Software utilizan el Método de Levenberg-Marquardt.

En seguida realizaremos algunas pruebas de ajuste para tener una idea más general de como funcionan estos algoritmos.

Realizamos el ajuste de la función $y = \frac{1}{x^3(e^{2x} - 1)}$ donde c es el parámetro a encontrar (el parámetro óptimo es 0.2) utilizando **MATLAB** y **NLSOL** donde el jacobiano se calcula mediante diferencias finitas, teniendo los siguientes resultados.

Resultados obtenidos del ajuste utilizando la función (*leastsq*) de MATLAB.

Condición Inicial	Número de Iteraciones	Número de evaluaciones en la Función	Parámetro Óptimo	Notas
0.1		24	0.2	
0.0001		46	0.2	
0.23		22	0.2	
1.0		37	0.2	
0.5		34	0.2	
-0.1		2	-0.1	No converge al óptimo esperado
5.0		2	5.0	No converge al óptimo esperado

Resultados obtenidos del ajuste utilizando NL2SOL.

Condición Inicial	Número de Iteraciones	Número de evaluaciones en la Función	Parámetro Óptimo	Notas
0.1	10	18	0.2	
0.0001	1	2	0.0001	No converge al óptimo esperado
0.23	6	12	0.2	
1.0	13	12	0.2	
0.5	10	19	0.2	
-0.1	1	2	-0.1	No converge al óptimo esperado
5.0	15	23	0.2	
9.0	15	33	0.2	

Es importante observar, antes de pasar al análisis de las tablas, que la función (*leastsq*) de MATLAB no proporciona el número de iteraciones y NL2SOL si las proporciona.

Analizando las 2 tablas de resultados podemos observar algunas características de MATLAB y NL2SOL

De la columna que contiene el número de evaluaciones en la función en ambas tablas, podemos decir que el algoritmo NL2SOL necesita calcular menos evaluaciones en la función que la función (*leastsq*) de MATLAB. Así mismo, si observamos la parte cuando la condición inicial es 0.0001, en la función (*leastsq*) de MATLAB se calcula el óptimo en 46 evaluaciones en la función, mientras que en NL2SOL no converge al óptimo esperado; en las 2 evaluaciones que hace la función, queda el mismo punto de inicio.

Otro punto importante es cuando la condición inicial es -0.1, tanto en la función (*leastsq*) de MATLAB como NL2SOL no convergen los algoritmos, ya que esta condición está fuera de la zona de convergencia; pero si observamos cuando la condición inicial es 5.0 en la función (*leastsq*) de MATLAB dicho algoritmo no converge al óptimo, mientras que en NL2SOL si converge haciendo 12 iteraciones con 23 evaluaciones en la función.

Del análisis anterior podemos concluir que:

-La función (*leastsq*) es menos susceptible a fallas ya que está dentro del ambiente computacional de MATLAB y no puede permitir suspensiones de ejecución tan fácilmente; es decir, esta mas protegido contra tales fallas; Esto se puede apreciar cuando la condición inicial es 0.0001, la función (*leastsq*) de MATLAB calculó bien el óptimo mientras que en NL2SOL no convergió a éste.

-NL2SOL es mas robusto, pero es mas susceptible a suspensiones de ejecución.

-Debido a que NL2SOL calcula el segundo término de la matriz hessiana, éste amplía mejor la zona de convergencia para el método y tiene mas posibilidades para calcular los parámetros, ya que si la condición inicial esta muy alejada del óptimo, es factible que llegue a encontrarlo. Esto lo podemos ver cuando la condición inicial es 5.0 debido a que en esta zona la gráfica se comporta como si fuera una línea recta. Podemos decir que se está dentro de un valle, la matriz hessiana es singular, entonces el algoritmo ayuda a la matriz hessiana a salir del valle.

-Es de suma importancia dar la condición inicial dentro del rango de aproximación para poder calcular el parámetro óptimo, ya que si lo hacemos fuera, como lo vimos en el análisis de las tablas donde la condición inicial es -0.1, no calcula el óptimo.

Debemos recalcar que en la función (*leastsq*) del paquete MATLAB y el algoritmo NL2SOL no introducimos el número de iteraciones, ni el valor de λ , ni el limite de error, ni los valores de la matriz D , etc; esto debido a que son muy robustos y no es necesario darles estos valores; pero si deseamos mejorar la convergencia de nuestro ajuste podemos también introducir el jacobiano de la función a ajustar.

El algoritmo NL2SOL es un algoritmo realizado en Fortran. Es factible obtenerlo de internet si no se tiene disponible.

A continuación describiremos mínimos cuadrados no lineales con restricciones.

Mínimos Cuadrados no Lineales con Restricciones.

Los problemas de mínimos cuadrados no lineales a menudo imponen restricciones en la solución. Por ejemplo, en las aplicaciones químicas que describiremos en el capítulo 4, algunas veces es necesario que los parámetros óptimos sean positivos.

Describiremos a continuación 2 estrategias para restringir la solución de aproximación a la región D del dominio de los parámetros c , donde $j = 1, 2, \dots, n$.

Recordemos el problema de mínimos cuadrados no lineales.

$$\text{Minimizar } r^2 = \sum_{i=1}^m \left[y_i - \sum_{j=1}^n y(x_i; c_j) \right]^2 \quad (2.3.1)$$

con solución iterativa a partir de

$$\left[J^T(c)J(c) + \lambda D \right] \delta c = J^T(c)r(c) \quad (2.3.10)$$

Al tener control sobre nuestra condición inicial $c_0 = (c_1, c_2, \dots, c_n)$ el método calcula el tamaño de paso δc pero puede suceder que $(c + \delta c)$ no esté en D . Si esta corrección es permitida, el algoritmo podría nunca restablecerse; es decir, la aproximación podría converger a una solución fuera de D .

La primera estrategia para restringir la corrección $(c + \delta c)$ es la de trasladar el problema restringido de mínimos cuadrados no lineales a un problema no restringido, esto es, imponer una penalidad en la función a minimizar si el algoritmo intenta dar un paso fuera de D . Por ejemplo:

Sea μ un número tal que

$$|r_i(c_0)| \leq \mu \quad i = 1, 2, \dots, m \quad (2.3.11)$$

donde

$$r(c) = \sum_{i=1}^m \left[y_i - \sum_{j=1}^n y(x_i; c_j) \right] = \mu \quad (2.3.12)$$

Si la función $r(c)$ es definida como en (2.3.12), se permitirá que la penalización impuesta, fuerce a toda aproximación c a estar en D .

Es importante recalcar que esta estrategia restringe todas las correcciones, y como consecuencia puede llevarnos a una convergencia muy lenta si la solución esta cerca de la frontera de \mathbf{D} . Esto generalmente se satisface restringiendo solo la corrección inicial.

Para profundizar más sobre las funciones de penalidad puede verse los textos de *Pravda 1991*, y *Ake Bjorek 1996*.

La otra estrategia de restricción nos dice lo siguiente: Ya que el algoritmo de MINPACK resuelve el problema de mínimos cuadrados no lineales iterativamente como sigue:

$$\text{Minimizar } \left\{ \left\| \sum_{i=1}^n \left[y_i - \sum_{j=1}^m y(x_i; c_j) \right] \right\| + J(c) \delta c \right\} : \|D \delta c\| \leq \Delta \quad (2.3.13)$$

Entonces modificamos Δ para que los nuevos valores de $(c + \delta c)$ no salgan del dominio de \mathbf{D} .

En esta restricción nosotros deseamos que se cumpla

$$\|D_0 \delta c_0\| \leq \Delta_0, \quad (2.3.14)$$

donde D_0 es la matriz de escala diagonal y Δ_0 es el límite de paso. La matriz D_0 es especificada por el usuario, mientras que el límite de paso Δ_0 está determinado por un número llamado *FACTOR*, este número lo proporciona el usuario

$$\Delta_0 = \text{FACTOR} \cdot \|D_0 c_0\| \quad (2.3.15)$$

si la condición inicial c_0 es el vector cero tenemos que

$$\Delta_0 = \text{FACTOR} \quad (2.3.16)$$

Es claro que pequeños valores de *FACTOR* corresponden a pequeños valores de paso. Para un valor de *FACTOR* suficientemente pequeño (generalmente se satisface 0.01) un punto mejorado $c_0 + \delta c_0$ pertenecerá a \mathbf{D} .

Se debe tener presente que la restricción del paso esta en $D_0 \delta c_0$ y no en δc_0 directamente. Un elemento pequeño de D_0 puede llevarnos a un componente grande en la corrección δc_0 . En muchos casos no es necesario controlar δc_0 directamente.

Como los componentes de D_0 son especificados por el usuario, esto nos lleva directamente al control de δc_0 . Si, por ejemplo, deseamos restringir la componente de δc_0 a

correcciones relativas pequeñas de los correspondientes componentes de c_0 (asumimos que los componentes son no ceros), entonces la matriz D_0 la escribimos como sigue

$$D_0 = \begin{bmatrix} \sqrt{|c_1|} & 0 & \dots & 0 \\ 0 & \sqrt{|c_2|} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{|c_n|} \end{bmatrix} \quad (2.3.17)$$

y escogemos apropiadamente el valor de *FACTOR*.

Se debe tener cuidado que la matriz D_0 tenga una escala apropiada que se relacione con nuestro fenómeno. En términos es conveniente tomar como base las características del fenómeno a ajustar.

Para justificar la elección de D_0 de la ecuación (2.3.17) tenemos que δc_0 satisface

$$\|D_0 \delta c_0\| \leq \Delta_0 = \text{FACTOR} \cdot \|D_0 c_0\|$$

y que la elección de D_0 garantiza que

$$\|D_0 c_0\| = \sqrt{n}$$

donde n es el número de parámetros a ajustar. De aquí, si δc_j es el j -ésimo componente de δc_0 , entonces

$$|\delta c_j| \leq \sqrt{n} \cdot \text{FACTOR} \cdot |c_j|$$

lo cual justifica la elección de D_0 .

Para la solución de mínimos cuadrados no lineales con restricciones encontramos un algoritmo computacional llamado MINPACK. Este algoritmo al igual que NL2SOL y la función (*teastsq*) del ambiente computacional MATLAB es también muy robusto por lo que solamente hay que codificar la función a minimizar con la penalización como lo vimos en la ecuación (2.3.12). Para la segunda restricción la matriz D_0 puede ser dada por el usuario al igual que Δ_0 ó pueden ser dados internamente por el algoritmo.

Es factible obtener el algoritmo MINPACK en Internet.

3. DISEÑO DE UN CÓDIGO PARA AJUSTAR PARÁMETROS DE EQUIPOS

3. DISEÑO DE UN CÓDIGO PARA AJUSTAR PARÁMETROS DE EQUIPOS.

INTRODUCCIÓN.

Como hemos visto, en el estudio de los métodos para ajustes de parámetros necesitamos hacer muchos cálculos para encontrar los que corresponden a nuestros ajustes. Estos cálculos son muy laboriosos y son susceptibles a errores, por lo que es importante la ayuda de programas de computadora para realizarlos.

Analizaremos las propiedades que necesitan los ajustes, los datos experimentales disponibles y los parámetros que necesitan los métodos; todo esto nos llevará a conocer los requerimientos de los programas computacionales para tener buenos ajustes de parámetros.

Utilizaremos como base ambientes estándares de matemáticas (**MATLAB**, **Mathematica**), así como algoritmos específicos para algún método en particular (**MINPACK**, **NLSOL**).

3.1 DEFINICIÓN DE LOS DATOS EXPERIMENTALES DISPONIBLES.

Al obtener los datos experimentales, es muy importante conocer que características tienen estos, tales como la forma de distribución de los puntos, si son discretos o continuos, etc; estas características se pueden obtener en muchas ocasiones a través de métodos gráficos.

Características de la gráfica de los datos experimentales.

- Tendencia de los datos. Si los datos tienen tendencia lineal, cuadrática, cúbica, exponencial, senoidal, etc.
- Distribución de los datos. Si estos están concentrados en una parte del dominio, ó uniformemente distribuidos.
- Discontinuidad. Si hay alguna discontinuidad en algún punto de la gráfica de f ; es decir; si la igualdad $\lim_{x \rightarrow x_0} f(x) = f(x_0)$ no se cumple.
- Puntos relevantes o importantes. Cuantos máximos, mínimos o puntos de inflexión tiene la gráfica.
- Aumento de Datos Experimentales. Verificar si el aumentar datos experimentales disponibles nos da una mejor idea de la gráfica.
- Cambio de Escala en los Datos Experimentales. Observar si es factible cambiar la forma de distribución de puntos para tener un mejor manejo de estos.

Observando las características de la gráfica de los datos experimentales podemos escoger que método se pega más al ajuste deseado. En seguida definiremos los parámetros de los métodos para saber cual de ellos cumple con las características de aproximación que nosotros deseamos.

3.2 DEFINICIÓN DE LOS PARÁMETROS DEL MÉTODO.

Para obtener un buen funcionamiento de los métodos de aproximación, es necesario saber manejar adecuadamente los parámetros que intervienen en él; ya que estos son una caracterización numérica de dicho método, de manera que describen parcial o completamente las propiedades de su interés del comportamiento.

En seguida mencionaremos los parámetros de cada método de aproximación.

Parámetros del Método de Mínimos Cuadrados Lineales.

Orden del Polinomio. Es el grado del polinomio a calcular como una combinación lineal de variables ó el número de funciones del polinomio a calcular en una combinación lineal de funciones. El grado del polinomio o el número de funciones lo representamos con la letra n ; donde n tiene que ser menor que el número de observaciones (m); $n < m$.

Tolerancia del error. Es el límite de error para comparar si el polinomio ajustado cumple con lo deseado, si no, aumentamos o disminuimos el orden del polinomio ó el número de funciones a aproximar.

Esta tolerancia de error está representada por la letra griega τ y se compara con el error del polinomio ajustado $r = \sum_{i=1}^m |y_i - f(x_i)|^2$ donde $r \leq \tau$.

Matriz de Pesos. Esta es una matriz de pesos para el método de Mínimos Cuadrados Pesados, tiene la propiedad de que es una matriz diagonal, donde los elementos de la diagonal son los pesos, estos pesos son positivos y son representados como

$$W = \begin{bmatrix} w_{00} & 0 & 0 & \dots & 0 \\ 0 & w_{11} & 0 & \dots & 0 \\ 0 & 0 & w_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & w_{mm} \end{bmatrix}$$

Forma del Polinomio. El polinomio ajustado debe tener la siguiente forma

$f(x_i) = c_0 + c_1 x_i + c_2 x_i^2 + \dots + c_n x_i^n$ cuando es combinación lineal de variables.

$f(x_i) = c_0 \Phi_0(x_i) + c_1 \Phi_1(x_i) + \dots + c_n \Phi_n(x_i)$ cuando es combinación lineal de funciones, donde las $\Phi_j(x_i)$ son las funciones que deseamos ajustar.

Parámetros del Método de Interpolación Cúbica Segmentaria.

Orden del Polinomio. En este método, el orden del polinomio a ajustar es de 3 ya que se trata de una interpolación cúbica y por lo tanto no varía el orden, donde el dicho orden debe de ser menor que el número de observaciones m .

Tolerancia de Error. Igual que en mínimos cuadrados lineales; excepto que puede variar mucho este error r por las condiciones de continuidad que hayamos elegido.

Características de Continuidad. Las funciones que deseamos ajustar pueden ser continuas o no en el rango seleccionado, según sea el caso de continuidad que nosotros hayamos elegido.

Estas características de continuidad pueden ser: continuidad en la función, continuidad en primera derivada, continuidad en función y primera derivada, etc.

Número de Segmentos. Es el número de observaciones dividido entre el orden del polinomio, es representado por $n = \frac{m}{3}$; donde $n \geq 2$.

Forma del polinomio. El polinomio ajustado debe tener la siguiente forma.
 $S_l(x) = a_l + b_l(x - x_{3l}) + c_l(x - x_{3l})^2 + d_l(x - x_{3l})^3$ donde l es el número de segmento a ajustar $l = 0, 1, \dots, n$

Parámetros del Método de Mínimos Cuadrados no Lineales.

Número de Parámetros. Es el número de parámetros que tiene la función o las funciones a minimizar y es representado por n . Al igual que en mínimos cuadrados lineales el número de parámetros tiene que ser menor que el número de observaciones.

Condición Inicial. Es el punto de inicio donde va a comenzar la primera iteración de la función a minimizar. Esta representado por $c^0 = (c_1, c_2, \dots, c_n)$.

Número de Iteraciones. Es el número de iteraciones para saber hasta donde deseamos parar el método. Esta representado por *NumIter*.

Tolerancia de Error. Es el límite de error para saber si los parámetros obtenidos son confiables o no. Esta representado por τ .

Matriz D . Esta es una matriz real positiva definida, puede ser la matriz Identidad ó matriz de Escalamiento; esta matriz se puede ver como una analogía de la matriz de pesos W de mínimos cuadrados lineales.

Tamaño de paso. Este parámetro sirve para indicarnos que tanto variamos el paso en el método de Levenberg-Marquardt; ya que nos ayuda a seleccionar si utilizamos Descenso Acelerado ó Gauss-Newton. Esta representado por λ .

Vector de Residuales. Es donde se almacena el cálculo de la función a minimizar a partir de la condición inicial y de los valores obtenidos de los parámetros c_j ($j = 1, 2, \dots, n$) óptimos en cada iteración, Representado por $r(x; c_j)$. Tiene analogía con la forma del polinomio de los otros métodos, solo que en éste se calculan los parámetros en cada iteración hasta encontrar el óptimo y en los otros métodos no hay iteración.

Como pudimos observar, los métodos a utilizar en el ajuste tiene parámetros en común que se deben cumplir en todos los casos, tales como tolerancia de error, el número de parámetros, etc; así como parámetros específicos como, las condiciones de continuidad, número de segmentos, etc. Representaremos en una tabla estos parámetros para tener una mejor idea de ellos.

Tabla de Parámetros de los Métodos de ajuste

Métodos de Ajuste.	Mínimos Cuadrados Lineales	Interpolación Cubica Segmentaria	Mínimos Cuadrados no Lineales.
Parámetros.			
Número de parámetros de polinomio.	n	3	n
Tolerancia de Error.	τ	τ	τ
Matriz de Pesos.	W		D
Características de Continuidad.	Continuidad en todo el rango de aproximación	Continuidad o Discontinuidad en los nodos de conexión	Continuidad en todo el rango de aproximación
Número de Segmentos.		n	
Condición Inicial.			$c_0 = (c_1, c_2, \dots, c_n)$
Número de Iteraciones.			NumIter
Tamaño de paso.			λ
Forma del Polinomio.	$f(x_i) = c_0 + c_1 x_i + \dots + c_n x_i^n$	Forma de polinomio cúbico vista anteriormente	$r(x_i; c_j)$

Requerimientos de los Métodos de Ajuste.

Para utilizar los métodos de ajuste es necesario disponer de ciertos recursos; estos representan los elementos necesarios para que funcionen los métodos mencionados, la mayoría son de características generales y se cumplen para todos los métodos de ajuste aquí estudiados. En seguida se describen tales requerimientos.

Número de Observaciones. Es el número de datos experimentales de la curva a aproximar, (número de valores de la variable independiente x y los valores de la variable dependiente y). El número de observaciones los representamos con la letra m .

En el método de Interpolación Cúbica Segmentaria el número de observaciones tiene que ser preferentemente múltiplo de 3.

Rango de Aproximación. Se debe disponer de un rango de aproximación para realizar el ajuste. Este queda representado por $[X_a, X_b]$ donde $X_a = x_1$ y $X_b = x_m$, y va ligado con el número de observaciones y con el intervalo de los números reales donde está localizado.

Continuidad. La función a ajustar debe ser continua en todo el rango de aproximación. En el método de Interpolación Cúbica Segmentaria dicha función puede ser no continua.

3.3 DEFINICIÓN DE LAS CARACTERÍSTICAS DEL AJUSTE.

Contando con las características de los datos experimentales y con los parámetros de los métodos podemos encontrar los requerimientos necesarios para los programas computacionales para el ajuste paramétrico; en seguida se mencionan las características del ajuste.

Rango de Aproximación. Intervalo donde se va a realizar el ajuste.

Tolerancia de Error. Es la tolerancia de error permitido en el ajuste.

Continuidad en el Ajuste. Procurar que la aproximación de los datos experimentales sea continua en todo el rango de aproximación.

Tipos de Continuidad en el Ajuste. Condiciones de continuidad vistas en el método de interpolación cúbica segmentaria para los nodos de conexión tales como continuidad en la función, continuidad en la primera derivada, etc.

Restricciones para el ajuste. Características especiales que se asignan a los parámetros a ajustar; por ejemplo, que los parámetros óptimos sean positivos.

Datos Experimentales más Importantes o más Significativos. Cuando se requiere que algunos datos experimentales tengan mas significancia que otros es necesario: introducir la matriz de pesos, donde éstos deben de ser positivos a lo largo de la diagonal principal.

Número de Segmentos. Es el número de subintervalos en que se reparte el rango de aproximación para la Interpolación Cúbica Segmentaria.

Número de Iteraciones. Seleccionar el número de iteraciones a realizar en el método de Mínimos Cuadrados no Lineales en caso de que no se llegue al parámetro óptimo. Punto de paro para el método.

Condición Inicial. Estimar de una forma adecuada la condición inicial para que sea factible llegar al parámetro óptimo.

De lo visto anteriormente, para el ajuste mediante mínimos cuadrados Lineales utilizamos el paquete **Mathematica**, ya que éste maneja en forma adecuada los métodos que aquí se manejan.

Para el ajuste mediante Interpolación Segmentaria Cúbica utilizamos los programas de computación específicos desarrollados para cada caso de continuidad.

Para el ajuste mediante Mínimos Cuadrados no Lineales utilizamos el algoritmo **NLSOL**, el ambiente computacional **MATLAB** con la función (*leastsq*) que realiza el ajuste mediante mínimos cuadrados no lineales y el algoritmo **MINPACK** para mínimos cuadrados no lineales con restricciones.

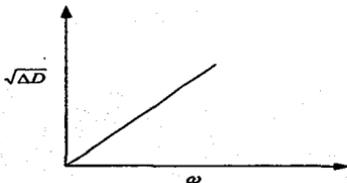
4. APLICACIÓN

4. APLICACIÓN.

4.1 APLICACIÓN DE LAS TÉCNICAS DE AJUSTE DE PARÁMETROS.

Ajuste Lineal.

Las técnicas de ajuste lineal $y(x) = \sum_{j=0}^n c_j x^j$. Tienen un amplio uso en el ajuste de propiedades físicas, por ejemplo Capacidad Calorífica vs Temperatura, Densidad vs Temperatura, Presión vs Temperatura, etc. Realizando algunas transformaciones también se pueden utilizar para describir curvas de funcionamiento de equipos como Caída de Presión vs Flujo los cuales tienen el siguiente comportamiento experimental:



Donde ΔD es la caída de presión y ω es el flujo.

Estas se pueden tratar como lineales en términos de sus parámetros, si hacemos la siguiente transformación:

$$y(x) = \sum_{j=0}^n c_j \Phi(x)$$

El ajuste se puede obtener fácilmente mediante bibliotecas disponibles en el lenguaje C (eg LAPACK, 1991), y en paquetes para manipulación de datos como MATLAB, y para manipulación simbólica como Mathematica.

Ajuste por Segmentos.

Quando se desea aproximar perfiles con salientes muy pronunciadas, los polinomios de aproximación pueden requerir un grado alto para lograr la exactitud requerida, esto puede llevar a oscilaciones que no corresponden al fenómeno real debido a esto es conveniente dividir el dominio en varios subintervalos, cada subintervalo puede entonces ser aproximados por polinomios de orden bajo (cuadrático, cúbico). La aplicación que se ha dado es el ajuste de propiedades físicas en dominios grandes, como las condiciones de ebullición del agua (temperatura, densidad, capacidad calorífica) desde una presión subatmosférica hasta presiones de varias decenas superior a la presión atmosférica (*Secker & Juárez, 1982*).

Al emplear métodos iterativos que utilizan las propiedades antes mencionadas, como parte de la solución de balances de materia y energía en los equipos, se detectó que estos no convergían debido que las derivadas utilizadas durante la iteración no eran continuas (*Hernandez, 1992*). Debido a esto se requiere tener una aproximación que además de ser continua en la *función* sea continua en su primera *derivada*. Los esquemas de aproximación propuestos nos permitirán seleccionar el nivel de continuidad deseado.

Ajuste no lineal.

Las propiedades de transporte como son coeficientes de fricción y coeficiente de transferencia de calor tienen un comportamiento no lineal con respecto al flujo, este comportamiento se debe a que las relaciones experimentales que se han encontrado tienen la forma

$$Y(x) = c_0 + c_1 x^{c_2} + c_3 x^{c_4}$$

Donde los coeficientes c_2 , c_4 son no enteros. En este tipo de aproximación se requiere que los métodos puedan converger a pesar de que se tengan valores estimados poco cercanos a los valores óptimos, y que no se interrumpa su ejecución porque la superficie a optimizar es poco sensible a los parámetros.

5. DISCUSIÓN DE RESULTADOS

5. DISCUSIÓN DE RESULTADOS.

Como hemos observado, en el ajuste de parámetros obtuvimos varios resultados de acuerdo a las características del problema y del método seleccionado; estos ajustes se llevaron a cabo mediante programas y ambientes de software matemático tomando en cuenta los criterios seleccionados para cada ajuste: continuidad, precisión, eficiencia, rango de validez.

En seguida analizaremos el ajuste de parámetros obtenido, así como el funcionamiento del software matemático seleccionado.

5.1 DISCUSIÓN DE RESULTADOS PARA MÍNIMOS CUADRADOS LINEALES.

Para mínimos cuadrados lineales ajustamos 3 funciones matemáticas que fueron $y = (x - 3)^2$, $y = (x - 3)^3$, $y = \frac{1}{x^3(e^{\sqrt{2}x} - 1)}$ utilizando el paquete **Mathematica**.

El rango de aproximación para las 2 primeras funciones fue de [0,12]. Las 2 primeras funciones no tuvieron ningún problema para hacer el ajuste por mínimos cuadrados lineales ya que se ajustaron mediante un polinomio cúbico y un polinomio de quinto orden. Al realizar el ajuste de la función $y = \frac{1}{x^3(e^{\sqrt{2}x} - 1)}$ el rango de aproximación

fue de [0,004,0,52] con incrementos de 0.004. Este ajuste no fue satisfactorio cuando se hizo mediante un polinomio de grado n es decir $f(x) = c_0 + c_1x + c_2x^2 + \dots + c_nx^n$. Al realizar el ajuste mediante el concepto de combinación lineal de funciones el polinomio de ajuste $f(x) = c_0\Phi_0(x) + c_1\Phi_1(x) + \dots + c_n\Phi_n(x)$ dio mejores resultados.

De una manera general podemos concluir que para el ajuste de parámetros mediante mínimos cuadrados lineales es recomendable utilizar el ambiente **Mathematica** ya que este ambiente maneja el concepto de mínimos cuadrados lineales como una combinación lineal de funciones, por lo que al hacer el ajuste como una combinación lineal de variables el paquete toma esto como una combinación lineal de funciones. Por otro lado es práctico, ya que permite ver la gráfica del ajuste así como la gráfica de los datos experimentales al mismo tiempo y podemos estar cambiando tanto el número de funciones a ajustar como el orden del polinomio calculado.

Otro punto importante es cuando deseamos que algunos puntos sean más significantes que otros ó que sean mas representativos que otros, en este caso utilizamos Mínimos Cuadrados Pesados; introducimos nuestra matriz de Pesos en el paquete **Mathematica** y realizamos el ajuste como se puede ver en el *apéndice B-2*.

5.2 DISCUSIÓN DE RESULTADOS PARA LA INTERPOLACIÓN CÚBICA SEGMENTARIA.

Al realizar el ajuste de estas mismas 3 funciones por medio del método de Interpolación Cúbica Segmentaria, obtuvimos los siguientes resultados.

Quando ajustamos la función $y = (x - 3)^3$ no hubo error aplicando todos los casos de continuidad vistos en el método; esto se debe a que, como es una función cúbica y el ajuste esta hecho mediante una aproximación cúbica, entonces el ajuste es perfecto.

En el ajuste de la función $y = (x - 3)^5$ el ajuste es perfecto cuando se pide la condición de continuidad en la función pero cuando escogemos las otras condiciones, el ajuste empieza a generar error, esto debido a que las condiciones de continuidad se basan en las primeras y segundas derivadas tanto en la función como en el polinomio de aproximación.

Al hacer el ajuste de la función $y = \frac{1}{x^3(e^{1/2x} - 1)}$ éste es perfecto cuando se pide la condición de continuidad en la función, cuando el ajuste se hace por medio de las otras condiciones de continuidad el error es relativamente grande. Quando hacemos el ajuste proponiendo que la condición de continuidad se cumpla tanto en primera como en segunda derivada a la vez, este error es grande debido a que estamos pidiendo que la condición se cumpla tanto en primera como en segunda derivada, el método permite esta continuidad a la expensa de error en la aproximación del valor de la función.

Con lo escrito anteriormente podemos concluir de una manera general lo siguiente.

Quando descamos ajustar nuestros datos experimentales mediante alguna condición de continuidad, o si nuestra función a ajustar es discontinua en alguna lugar del rango de aproximación o si esta función tiene algún pico, es recomendable usar la interpolación cúbica segmentaria ya que ésta satisface varias condiciones de continuidad y permite escoger la que más nos convenga. De igual forma, si nuestros datos experimentales tienen comportamiento irregular; es decir; si no tienen forma de alguna curva conocida, podemos escoger la condición de continuidad en la función para que este ajuste pase por todos los datos experimentales considerados. Es importante tener en cuenta que este método nos permite tener varias opciones de continuidad para realizar el ajuste; por tanto, esta interpolación es posible para casos donde haya discontinuidad. Por otro lado, para lograr una mejor aproximación se tiene que cumplir de preferencia solo un caso de continuidad y no varios, esto lo pudimos ver cuando las condiciones de continuidad están sobre la primera y segunda derivadas de la función de aproximación; en el ajuste de las funciones realizadas ($y = (x - 3)^5$, $y = \frac{1}{x^3(e^{1/2x} - 1)}$) al tratar de hacer el ajuste con esta condición hubo un error grande ya que el algoritmo fuerza a que se cumplan las 2 condiciones.

5.3 DISCUSIÓN DE RESULTADOS PARA MÍNIMOS CUADRADOS NO LINEALES.

Al realizar el ajuste de las funciones $y = (x - 3)^3$, $y = (x - 3)^4$ por el método de mínimos cuadrados no lineales, sacamos los datos experimentales de estas funciones y los ajustamos con las funciones $y = (x - c)^3$ y $y = (x - c)^4$ donde el parámetro a ajustar es c , realizando ajustes mediante la función (*leastsq*) del paquete matemático de MATLAB y el algoritmo hecho en Fortran NL2SOL los resultados resultaron satisfactorios ya que no presentaron problemas para calcular el parámetro óptimo.

Realizando el ajuste de la función $y = \frac{1}{x^2(e^{2/x} - 1)}$ donde c es el parámetro a ajustar, los resultados tanto en la función (*leastsq*) de MATLAB como en NL2SOL varían un poco, pero dan el parámetro óptimo, NL2SOL resultó ser más robusto que MATLAB en el ejemplo tratado.

De una manera general podemos concluir que si al analizar nuestra función a ajustar ésta tiene funciones que dependen de una manera no lineal de sus parámetros, se deben elegir los paquetes MATLAB y NL2SOL ya que estos son los más robustos que encontramos en nuestra investigación para hacer este ajuste. Es importante resaltar que es lo que deseamos en este ajuste. Cuando analizamos los paquetes computacionales mencionados para el ajuste mediante mínimos cuadrados no lineales encontramos que:

NL2SOL tiene las siguientes características.

- Es mas robusto que la función (*leastsq*) de MATLAB.
- Calcula el segundo término de la matriz hessiana.
- Verifica la región de confianza continuamente.

La función (*leastsq*) de MATLAB tiene las siguientes características.

- Es poco susceptible a interrupciones en su ejecución.
- Es mas fácil de usar ya que esta función realiza el ajuste bajo un paquete computacional.
- Es mas práctica de utilizar ya que es mas fácil que se tenga el paquete MATLAB que el compilador Fortran

Para mínimos cuadrados no lineales con restricciones hay un algoritmo también hecho en Fortran llamado MINPACK, este algoritmo al igual que NL2SOL también trabaja con mínimos cuadrados no lineales con ó sin restricciones. Este algoritmo también es muy robusto por lo que si se desea introducir alguna restricción, el algoritmo lo hace automáticamente.

6. CONCLUSIONES Y LIMITACIONES

6. CONCLUSIONES Y LIMITACIONES.

En los métodos para el ajuste de parámetros, realizamos un estudio muy cuidadoso para lograr aumentar la precisión y rango de validez de los modelos, reducir el tiempo de estimación de parámetros y mantener consistencia con las técnicas utilizadas para ajustar parámetros. En seguida mencionaremos las conclusiones y limitaciones de cada método.

6.1 CONCLUSIONES Y LIMITACIONES PARA EL MÉTODO DE MÍNIMOS CUADRADOS LINEALES.

El ajuste de parámetros por mínimos cuadrados lineales es restringido, solo es recomendable para ajuste de parámetros lineales. Al investigar este método encontramos el concepto de mínimos cuadrados lineales como una combinación lineal de funciones; dicho concepto amplía de manera sustancial el anterior y permite, no solo usar polinomios de grado n de la forma $f(x_i) = c_0 + c_1x + c_2x_i^2 + \dots + c_nx_i^n$ para ajustar parámetros, sino también utilizar funciones. Estos parámetros a ajustar tienen que estar en forma lineal.

Al realizar la investigación del software para el ajuste, nos encontramos con un ambiente standard de matemáticas llamado **Mathematica**; este ambiente permite realizar el ajuste de mínimos cuadrados lineales como una combinación lineal de funciones. En la realización de nuestros ajustes nos dio resultados satisfactorios ya que como tiene ambiente gráfico nos permite ver las curvas de los datos experimentales. Cuando realizamos el ajuste podemos comparar la gráfica de este y la de los datos experimentales al mismo tiempo, ayudándonos así a observar si el ajuste es satisfactorio. De no ser así, se puede cambiar la función o el grado del polinomio a ajustar. Permite también evaluar los datos de la variable dependiente x en el polinomio de ajuste permitiendo con esto calcular el error cometido en la elección del polinomio.

Si deseamos ajustar mínimos cuadrados pesados, introducimos nuestra matriz de pesos en **Mathematica**, realizamos las operaciones pertinentes y ajustamos los parámetros.

Entre las limitaciones del método así como del ambiente standard **Mathematica** tenemos lo siguiente:

- Las funciones deben de ser continuas en todo el rango de aproximación.
- El ajuste por mínimos cuadrados lineales es solo para ajustar parámetros lineales.
- Es necesario proveer un mecanismo de *entrada/salida* de vectores más ágil para el ambiente de **Mathematica** (Ver Apéndice B-2).

6.2 CONCLUSIONES Y LIMITACIONES PARA LA INTERPOLACIÓN CÚBICA SEGMENTARIA.

Al realizar el ajuste de parámetros por medio de la interpolación cúbica segmentaria nos encontramos que este método lo podemos utilizar especialmente cuando las gráficas que deseamos ajustar tienen algún pico dentro del rango de aproximación ó si existe alguna discontinuidad en la función. El método es muy flexible para los casos donde existe discontinuidad permitiendo seleccionar el caso de continuidad que nosotros deseamos. Se recomienda que el ajuste pase por todos los datos experimentales disponibles cuando pedimos la condición de continuidad en la función.

En la investigación para el ajuste por el método de la interpolación cúbica segmentaria nos encontramos que las referencias consultadas solo tenían algunos casos de continuidad y no varios como lo expusimos en este trabajo por lo que el desarrollo del método es muy completo; de igual forma no mencionaban algoritmos fáciles para su desarrollo. Debido a esto realizamos nuestros propios programas para realizar el ajuste, así como adecuaciones a ellos para cada condición de continuidad. Estos programas fueron hechos en lenguaje *Ansi-C*, auxiliándonos para la solución de las ecuaciones que da el desarrollo del método por medio de librerías hechas en lenguaje *C*. (**LAPACK**), estas librerías son de dominio publico y pueden ser encontradas a través de Internet.

Entre las limitaciones del método tenemos las siguientes:

-Al realizar el ajuste de funciones por medio del método señalado tenemos que al pedir las condiciones de continuidad se debe sacrificar alguna condición de continuidad para que se cumpla otra, es decir, si por ejemplo, deseamos condición de continuidad en primera derivada, los nodos de conexión tendrán un error en el polinomio de ajuste mientras que al checarlos con la derivada del polinomio de aproximación no habrá error.

-En este ajuste por medio del método antes señalado, el error variará dependiendo de cual sea la condición de continuidad que hayamos seleccionado, y de acuerdo a los datos experimentales de la función ó del fenómeno que deseamos ajustar (*Ver gráficas de ajustes. Subcapítulo 2.2*) por lo que no podemos predecir tan fácilmente el error que nos dará el ajuste.

-Nuestros programas elaborados para este ajuste son muy limitados, solo nos dan los parámetros del ajuste, no tienen graficación y trabajan bajo *MSDOS*, no tienen algún ambiente computacional comparándolo con *Mathematica* ó *MATLAB*.

6.3 CONCLUSIONES Y LIMITACIONES PARA EL MÉTODO DE MÍNIMOS CUADRADOS NO LINEALES.

Cuando realizamos nuestros ajustes de parámetros nos encontramos algunos no lineales; los métodos vistos anteriormente (mínimos cuadrados lineales, interpolación cúbica segmentaria) no pueden ajustar parámetros no lineales ó si lo hacen, dan errores muy grandes; por lo que nos decidimos a investigar otros métodos de ajuste. En la investigación nos encontramos que los métodos de ajuste con mínimos cuadrados no lineales generalmente se orientan a la solución de ecuaciones no lineales o a minimizar o maximizar funciones. Siguiendo la investigación nos encontramos con el método de Levenberg-Marquardt que realiza el ajuste de parámetros no lineales por medio de mínimos cuadrados no lineales; en la investigación del software nos encontramos con la función (*leastsq*) del ambiente computacional MATLAB y los algoritmos realizados en Fortran NL2SOL y MINPACK, estos algoritmos y la función (*leastsq*) manejan ya el método de Levenberg-Marquardt, en el análisis de estos paquetes encontramos varias características importantes; a continuación describiremos algunas de ellas.

Características importantes de los paquetes computacionales MATLAB (*leastsq*), NL2SOL y MINPACK.

-MATLAB es un ambiente computacional que se maneja bajo el ambiente Windows y es de fácil manejo. Este ambiente es menos susceptible a interrupciones en la ejecución, es amigable y permite graficación.

-El algoritmo NL2SOL es el mas robusto de los 3 paquetes investigados ya que calcula el segundo término de la matriz hessiana y esta diseñado básicamente para mínimos cuadrados no lineales.

-MINPACK nos ayuda a resolver mínimos cuadrados no lineales con restricciones.

Como hemos observado el método de Levenberg-Marquardt es muy robusto y da excelentes resultados pero es importante observar que también tiene limitaciones. A continuación describiremos algunas de las más importantes del método y de los paquetes computacionales descritos anteriormente.

-Si se trata de un ajuste de parámetros no lineales, no existen métodos que resuelvan el ajuste exactamente, como lo hace mínimos cuadrados lineales. Debido a que se trata de métodos iterativos siempre estaremos expuestos a que la convergencia este lejos del óptimo real.

-La convergencia será más rápida cuanto más cerca del óptimo real se proponga el valor de inicio.

-El ajuste de parámetros no lineales con restricciones solo toma en cuenta restricciones para la región del dominio de los parámetros c , por lo que es muy limitado. No toma en cuenta restricciones tales como funciones o desigualdades matemáticas.

Por lo que se refiere a las limitaciones de los paquetes computacionales tenemos lo siguiente.

-**NLSOL** al ser un algoritmo realizado en Fortran es más susceptible fallas en la ejecución del método, ya que no tiene mucha protección contra éstas comparándolo con **MATLAB**. Lo mismo sucede con **MINPACK**.

-**MINPACK** es limitado para la solución de mínimos cuadrados no lineales con restricciones ya que solo trabaja restringiendo la solución de aproximación a la región del dominio de los parámetros c_i .

-El único inconveniente de la función (*leastsq*) de **MATLAB** es que no calcula el segundo término de la matriz hessiana.

6.4 CONCLUSIONES GENERALES

De lo estudiado en todo nuestro trabajo y con las experiencias obtenidas de éste podemos concluir de manera general lo siguiente:

Como en todos los procesos que requieren de herramientas numéricas de aproximación, en el ajuste de parámetros se introducen errores. Con el fin de evitar que estos sean considerables, se recomienda lo siguiente:

a) Analizar muy detenidamente la gráfica de los datos experimentales así como tener información acerca del fenómeno o de la función de donde se obtuvieron. A partir de ello, elegir el método de ajuste que más convenga.

b) Si nuestros datos experimentales lo permiten, hacer nuestro ajuste por los diferentes métodos estudiados y comparar los resultados obtenidos de cada uno de ellos. Logicamente, esto brinda más alternativas para seleccionar algún método particular.

APÉNDICE B-1

APÉNDICE B-1.

TÉCNICAS DE APOYO EN LA RESOLUCIÓN DE MÍNIMOS CUADRADOS.

Ortogonalización Gram-Schmidt.

Sea $\{u_1, u_2, \dots, u_n\}$ un conjunto de vectores no nulos, linealmente independientes, deseamos construir un conjunto de vectores $\{v_1, v_2, \dots, v_n\}$ tales que para cada $k = 1, 2, \dots, n$, el conjunto $\{v_1, v_2, \dots, v_k\}$ sea un conjunto ortogonal.

Como el primer vector del nuevo conjunto elegimos el primer vector del conjunto original; esto es, definimos $v_1 = u_1$.

Ahora deseamos encontrar el segundo vector v_2 que sea ortogonal a v_1 . Como sabemos, para que dos vectores v_i y v_j sean ortogonales se debe cumplir que $\langle v_i, v_j \rangle = 0$, proponemos entonces $\langle v_2, v_1 \rangle = 0$. Para determinar el vector v_2 tenemos lo siguiente

$$v_2 = u_2 + c_{21}v_1 \quad (B.1.1)$$

donde c_{21} es un escalar a determinar. Partiendo del hecho de que v_1 y v_2 son ortogonales multiplicamos la ecuación (B.1.1) escalarmente por v_1

$$\begin{aligned} \langle v_2, v_1 \rangle &= \langle u_2 + c_{21}v_1, v_1 \rangle \\ &= \langle u_2, v_1 \rangle + c_{21} \langle v_1, v_1 \rangle = 0 \end{aligned}$$

Despejando la incógnita c_{21} resulta

$$c_{21} = - \frac{\langle u_2, v_1 \rangle}{\|v_1\|^2}$$

donde $\|v_1\|^2 = \langle v_1, v_1 \rangle$. Así sustituyendo c_{21} en la ecuación (B.1.1) se obtiene

$$v_2 = u_2 - \frac{\langle u_2, v_1 \rangle}{\|v_1\|^2} v_1 \quad (B.1.2)$$

Para obtener el tercer vector v_3 del nuevo conjunto, proponemos

$$v_3 = u_3 + c_{31}v_1 + c_{32}v_2 \quad (B.1.3)$$

donde nuevamente. c_{32} y c_{31} , son escalares que se deben calcular. Dado que se pretende que v_3 sea ortogonal tanto a v_2 como a v_1 , proponemos

$$\langle v_3, v_2 \rangle = 0, \quad \langle v_3, v_1 \rangle = 0$$

Multiplicamos la expresión (B.1.3) escalarmente por v_1 , resulta

$$\begin{aligned} \langle v_3, v_1 \rangle &= \langle u_3 + c_{31}v_1 + c_{32}v_2, v_1 \rangle \\ &= \langle u_3, v_1 \rangle + c_{31}\langle v_1, v_1 \rangle + c_{32}\langle v_2, v_1 \rangle = 0 \\ &= \langle u_3, v_1 \rangle + c_{31}\langle v_1, v_1 \rangle = 0 \end{aligned}$$

de aquí tenemos que

$$c_{31} = -\frac{\langle u_3, v_1 \rangle}{\|v_1\|^2}$$

De manera semejante se determina

$$c_{32} = -\frac{\langle u_3, v_2 \rangle}{\|v_2\|^2}$$

Sustituyendo c_{32} , y c_{31} , en la ecuación (B.1.3) quedando entonces

$$v_3 = u_3 - \frac{\langle u_3, v_1 \rangle}{\|v_1\|^2} v_1 - \frac{\langle u_3, v_2 \rangle}{\|v_2\|^2} v_2 \quad (\text{B.1.4})$$

En general, repitiendo el procedimiento k veces ($k = 1, 2, \dots, n$), se obtiene

$$v_k = u_k - \sum_{j=1}^{k-1} \frac{\langle u_k, v_j \rangle}{\|v_j\|^2} v_j, \quad (\text{B.1.5})$$

donde $v_1 = u_1$.

Si lo que se desea es obtener un conjunto ortonormal, se puede proponer

$$q_j = \frac{v_j}{\|v_j\|} \quad \text{donde } j = 1, 2, \dots, n. \quad (\text{B.1.6})$$

Ejemplo de Proceso de Ortonormalización de Gram-Schmidt

Supongamos que tenemos una matriz de 3×3 llamada U .

$$U = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

descomponemos la matriz en vectores

$$u_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad u_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad u_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Entonces $v_1 = u_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ y v_2 es calculado como en la ecuación (B.1.2)

$$v_2 = u_2 - \frac{\langle u_2, v_1 \rangle}{\|v_1\|^2} v_1$$

$$\frac{\langle u_2, v_1 \rangle}{\|v_1\|^2} = \frac{1}{2} \Rightarrow v_2 = u_2 - \frac{1}{2} v_1 = \begin{bmatrix} 1/2 \\ -1/2 \\ 1 \end{bmatrix}$$

El tercer vector v_3 es calculado como en la ecuación (B.1.4)

$$v_3 = u_3 - \frac{\langle u_3, v_1 \rangle}{\|v_1\|^2} v_1 - \frac{\langle u_3, v_2 \rangle}{\|v_2\|^2} v_2$$

$$\frac{\langle u_3, v_1 \rangle}{\|v_1\|^2} = \frac{1}{2}, \quad \frac{\langle u_3, v_2 \rangle}{\|v_2\|^2} = \frac{1}{3} \Rightarrow v_3 = u_3 - \frac{1}{2} v_1 - \frac{1}{3} v_2 = \begin{bmatrix} -2/3 \\ 2/3 \\ 2/3 \end{bmatrix}$$

Los vectores finales ortonormales son los siguientes

$$q_1 = \frac{v_1}{\|v_1\|} = \sqrt{\frac{1}{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad q_2 = \frac{v_2}{\|v_2\|} = \sqrt{\frac{2}{3}} \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \\ 1 \end{bmatrix}, \quad q_3 = \frac{v_3}{\|v_3\|} = \sqrt{\frac{3}{4}} \begin{bmatrix} -\frac{2}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} \end{bmatrix}$$

$$q_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}, \quad q_2 = \begin{bmatrix} \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \end{bmatrix}, \quad q_3 = \begin{bmatrix} -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix}$$

$$q_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad q_2 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, \quad q_3 = \frac{1}{\sqrt{3}} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

Nuestro problema ahora es preguntar cómo las columnas originales u_i pueden ser recuperadas de los vectores finales q_i . Si descomponemos las ecuaciones para el v_i , en este ejemplo encontramos

$$\begin{array}{ll} u_1 = v_1 & \text{ó} \quad u_1 = \sqrt{2}q_1 \\ (B.1.7) \quad u_2 = \frac{1}{2}v_1 + v_2 & \text{ó} \quad u_2 = \sqrt{\frac{1}{2}}q_1 + \sqrt{\frac{3}{2}}q_2 \\ u_3 = \frac{1}{2}v_1 + \frac{1}{3}v_2 + v_3 & \text{ó} \quad u_3 = \sqrt{\frac{1}{2}}q_1 + \sqrt{\frac{1}{6}}q_2 + \sqrt{\frac{4}{3}}q_3 \end{array}$$

El conjunto de ecuaciones (B.1.7) puede ser escrito en notación matricial

$$U = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 0 & \sqrt{\frac{3}{2}} & \sqrt{\frac{1}{6}} \\ 0 & 0 & \sqrt{\frac{3}{3}} \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 3\sqrt{2} & \sqrt{6} & -2\sqrt{3} \\ 3\sqrt{2} & -\sqrt{6} & 2\sqrt{3} \\ 0 & 2\sqrt{6} & 2\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 0 & \sqrt{\frac{3}{2}} & \sqrt{\frac{1}{6}} \\ 0 & 0 & \sqrt{\frac{4}{3}} \end{bmatrix}$$

Donde la matriz $U = QR$.

La matriz U es factorizada dentro de una matriz Ortogonal Q y una matriz Triangular Superior R . Las columnas de Q son los vectores ortonormales, los cuales necesitamos. Las ecuaciones (B.1.7) expresan cada vector u_i como una combinación lineal de los primeros i vectores v_i , donde tenemos que reemplazar solamente a v_1 por $\|v_1\|q_1$, v_2 por $\|v_2\|q_2$, y así sucesivamente, hasta el vector v_n por $\|v_n\|q_n$. Podemos observar que los coeficientes $\|v_i\|$ donde $i = 1, 2, \dots, n$ están en la diagonal principal de la matriz R . Esto produce el resultado de la factorización QR .

Una matriz U con columnas linealmente independientes puede ser factorizada dentro de $U = QR$. Las columnas de Q son ortonormales, y R es triangular superior e invertible. Si la matriz U es cuadrada, entonces también lo serán estos 2 factores Q y R , y Q se convierte en una matriz ortogonal. Por lo tanto el algoritmo de Gram-Schmidt ayuda a la solución del problema de mínimos cuadrados descomponiendo la matriz U en QR .

APÉNDICE B-2

APÉNDICE B-2

AJUSTE DE DATOS POR MÍNIMOS CUADRADOS UTILIZANDO EL PAQUETE MATHEMATICA.

1.- Introducir en un editor de textos los vectores X , Y , X_{prueba} , Y_{prueba} de la siguiente forma

$$\begin{bmatrix} x \\ x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y \\ y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \begin{bmatrix} x_p \\ x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y_p \\ y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Los archivos se pueden llamar *datos.m* para los vectores X , Y ; y para los vectores X_{prueba} , Y_{prueba} el archivo se puede llamar *datospru.m*

Nota: los archivos y el nombre de las variables de las operaciones realizadas se pueden llamar de cualquier otra forma; estos nombres puestos anteriormente solo son para tener una idea de como se maneja el paquete mathematica para el ajuste por mínimos cuadrados.

Es recomendable meter en el archivo de *datospru.m* puntos intermedios de los datos observados para ver que tanto cambia el error del polinomio.

Es muy importante seguir la sintaxis correcta de la función para el correcto funcionamiento del paquete.

2.- Abrir los archivos para los resultados obtenidos en el ajuste con la función siguiente:

```
OpenWrite ["file"]
```

```
OpenWrite ["result"],
```

```
OpenWrite ["resultpr"]
```

3.- Llamar al archivo *datos.m* y *datospru.m* con la función siguiente:

```
ReadList["file", {Number, Number}]
```

```
Matriz = ReadList["datos.m", {m, n}]
```

```
MatrizPr = ReadList["datospru.m", {m, n}]
```

donde m son el numero de observaciones y n es igual a 2.

4.- Descomponer las matrices *Matriz* y *MatrizPr* en vectores con la función siguiente:

```
Transpose [nombre] [[i]] ó Map [#[[i]]&, m]
```

```
xv=Transpose [Matriz] [[1]] ó Map [#[[1]]&, Matriz]
```

```
yv=Transpose [Matriz] [[2]] ó Map [#[[2]]&, Matriz]
```

```
xvpru=Transpose [MatrizPr] [[1]] ó Map [#[[1]]&, MatrizPr]
```

```
yvpru=Transpose [MatrizPr] [[2]] ó Map [#[[2]]&, MatrizPr]
```

5.- Dibujar los puntos de los vectores *xv*, *yv* con la función *ListPlot* para tener una idea de como se comportan los puntos.

```
gp=ListPlot[ {{xv[[1]],yv[[1]]}, {xv[[2]],yv[[2]]},.....{xv[[m]], yv[[m]]} } ]
```

6.- Ajustar los datos con la función *Fit*.

```
Fit[ {{xv[[1]],yv[[1]]}, {xv[[2]],yv[[2]]},.....{xv[[m]],yv[[m]]}, {f, f,...}, x ]
```

funciones donde *Fit* ajusta los puntos $\{xv[[m]], yv[[m]]\}$ a una combinación lineal de

```
f=1,2,..
```

6a) Igualar el polinomio obtenido del ajuste a la variable (polajust)

```
polajust=In[num]
```

donde *In[num]* se refiere al número de línea en donde aparece el polinomio ajustado.

Ejemplo:

```
[22]= 2-3x  
[23]= In[22]
```

6b) Guardar en el archivo *result.m* el polinomio ajustado de la forma:

```
polajust >> result.m
```

7.- Dibujar el polinomio ajustado con la función *Plot[f, {x, x_{min}, x_{max}}]* donde *f* es la función del ajuste:

```
grafun = Plot[polajust, {x, xmin, xmax}]
```

7a) Guardar en el archivo *grafic* la gráfica del polinomio ajustado con la función **Display** de la siguiente forma:

```
Display["grafic", grafun]
```

8.- Dibujar el ajuste hecho junto con los puntos dados al inicio, con la función **Show[plot]**

```
grafajpo = Show[%num, gp]
```

donde num es el número de línea en donde se realizó la gráfica del polinomio ajustado

8a) Guardar en el archivo *grafajs* la gráfica del polinomio ajustado junto con los puntos dados

```
Display["grafajs", grafajpo]
```

9.- Si los resultados no son satisfactorios, volver hacer el ajuste, aumentando el orden del polinomio, o cambiando la función a ajustar (paso 6).

10.- Evaluar los datos del vector *xv* del problema en el polinomio calculado de la siguiente forma:

```
polajust /.x-> xv[[i]]          i=1,2,...,m
```

10a) Guardar los resultados en el vector *Fx* de la siguiente forma:

```
Fx = {num[1], num[2], num[3], ..., num[i], ..., num[m] }
```

donde num[i] es la evaluación i-ésima del vector *xv*[i]

10b) Guardar el vector *Fx* en el archivo de resultados (*result.m*) de la siguiente forma:

```
Fx >>> result.m
```

11.- Evaluar el error del polinomio de la forma $\sum_{i=1}^m |y_i - f(x_i)|^2$ compararlo con el error de tolerancia asignado (τ), si los resultados son satisfactorios, se obtuvo el mejor ajuste, si no

regresar a la función Fit (paso 6) y aumentar el orden del polinomio ó cambiar la combinación lineal de funciones.

El error del polinomio lo calculamos de la siguiente forma:

$$ErrorPoli = \text{Sum}[(y_v[[i]] - f_x[[i]])^2, \{i, i_{\min}, i_{\max}\}]$$

11a) Guardar en el archivo *result.m* el error del Polinomio de la siguiente forma:

```
ErrorPoli >>> result.m
```

12.- Evaluar el vector de pruebas *xvpru* en el polinomio calculado de la forma

$$polajust /x \rightarrow xvpru[[i]] \quad i=1,2,\dots,m$$

12a) Guardar los resultados en el vector *Fxpru* de la siguiente forma:

$$Fxpru = \{num[1], num[2], num[3], \dots, num[i], \dots, num[m] \}$$

donde *num[i]* es la evaluación *i*-ésima del vector *xvpru[i]*

12b) Guardar el vector *Fxpru* en el archivo de resultados (*result.m*) de la siguiente forma:

```
Fxpru >>> result.m
```

13.- Calcular el error de la siguiente forma:

$$ErrPolpru = \text{Sum}[(y_vpru[[i]] - Fxpru[[i]])^2, \{i, i_{\min}, i_{\max}\}]$$

13a) Guardar en el archivo *result.m* el error del Polinomio de los datos de prueba de la siguiente forma:

```
ErrPolpru >>> result.m
```


BIBLIOGRAFÍA

- Åke Björck. **Numerical Methods for Least Squares Problems**. SIAM Publication (1996).
- R. L. Burden, J. D. Faires **Análisis Numérico** Grupo Editorial Iberoamérica (1987).
- C. Colín Castellanos **Diagnóstico de bombas de agua de alimentación, condensado y circulación en centrales termoeléctricas**, Boletín I.I.E. pp 126-128 (1991).
- S. C. Chapra, Raymond P. Canale, **Métodos Numéricos para Ingenieros**, McGrawHill, (1996).
- C. Van-Loan **On the Method of Weighting for Equality-Constrained Least-Squares Problems** SIAM J. Numer. Anal. V. 22 No 5 Oct. (1985) pp 851-863.
- D. L. Marquardt **An algorithm for Least Squares Estimation of Nonlinear Parameters**, SIAM 2, 431-441, (1963).
- E. Anderson, Z. Bai, Bischof J. Demmel, J. Dongarra, J. Du Cros, A. Greenbaum, S Hammarling, A. McKenny, S. Ostrouchov, and D. Sorensen, **"LAPACK, User's Guide"**, SIAM Publication, Philadelphia (1992).
- G. Ramos N., C. Gongora M., L. Barrero P., C. E. Aguirre I., D. A. Toral G., J. I. Icaza A., **Control y monitoreo del simulador por el instructor**, Boletín I.I.E., pp 133-143 (1981).
- G. Strang, **Linear Algebra and its Applications**, Hbj. College and School Div., (1988).
- J. E. Dennis, D. M. Gay, and R. E. Welsh, **An adaptive nonlinear least squares algorithm**, ACM Trans. Math. Software, 7 (1981), pp 348-368.
- J. E. Dennis, D. M. Gay, and R. E. Welsh, **Algorithm 573 NL2SOL: An adaptive nonlinear least squares algorithm**, ACM Trans. Math. Software, 7 (1981), pp 369-383.
- J. E. Fuentes, D. Juárez, Y. Mendoza, J. L. Morales, **Recomendaciones para Desarrollo de Modelos**, Boletín I.I.E. Dic (1993).
- J.J. Moré B. S. Garbow, and K. E. Hillstrom, **Users' guide for MINPACK-1**, Tech. Report ANL-80-74, Applied Math. Div., Argonne National Laboratory, Argonne Illinois (1980).
- J.J. Moré B. S. Garbow, and K. E. Hillstrom, **"Implementation Guide for MINPACK-1"**, Argonne National Laboratory, Argonne Illinois (1980).
- The Mathworks MATLAB **High Performance Numerical Computation and Visualization Software**. Guía de Referencia, (1993).

K. Ogata, Dinámica de Sistemas, Prentice Hall (1993).

L. V. Atkinson, P J Harley, Introducción a los Métodos Numéricos con Pascal, Addison-Wesley Iberoamericana, (1988).

Ma. A. E. Sandoval, *Simuladores para entretenimiento de operadores de sistemas de potencia*, Boletín I.I.E., pp 129-131 (1981).

M. Gonzales Hernandez, Memorias del II Congreso Nacional de Informatica Aplicada a la Ingenieria, Modelación Matemática y Simulación en Ingeniería, pp 1-20, Instituto Politecnico Nacional (1992).

Prawda J., Métodos y Modelos de Investigación de Operaciones. Vol I, Noriega Editores (1990).

P. A. Secker (Univ. de Arizona) y D. Juárez, *Paquete para la evaluación rápida de las propiedades físicas del agua*, Depto de Simulación, IIE. (1982).

Philip E. Gill, Walter Murray, Margaret M. Wright, Numerical Linear Algebra and Optimization. Volumen I, Addison-Wesley, (1990).

M. J. D. Powell, Approximation Theory and Methods, Cambridge Univ. Pr., (1981).

R. B. Deshpande, B. K. Shankramurty and C. P. Revankar, Comparasion of Curve Fitting Techniques, Int J Num Meth Engng v 10, pp 457-463 (1976).

R. I. López *Comportamiento de las propiedades Termodinámicas en las zonas de transición; Localización de discontinuidades en las derivadas de las Propiedades Termodinámicas del Agua* SIMEX Dic (1992).

Ralston A., and P. Rabinowitz, A First Course in Numerical Analysis, McGraw-Hill, Second Edition, (1978).

S. González, A. Villavicencio, *Herramientas para el adiestramiento y la operación en centrales termoeléctricas. Tecnologías en evolución*, Boletín I.I.E. pp 245-249, (1989).

S. W. Morgan, S. P. Sendelbach, W. B. Stewart, *Improve process training with dynamic simulation*, pp 51-60, (1994).

S Vasantharajan, J Logson, and L T Biegler, Simultaneous Strategies for Optimization of Differential-Algebraic Systems, paper 173 Annual AIChE Meeting Washington D C Nov (1988).

Wolfgang Research Institute Mathematica a program to make Mathematics by Computer (1996).

Y. Mendoza A., J. M. Molina, D. Juárez R., *Formato Computacional de los Modelos de Equipos y Recomendaciones Asociadas para su Formulación y Codificación*. V1.0. (1995).

W. H. H. Press, B. P. Flannery, S. A. Teukolsky, W.P. Vetterlin, **Numerical Recipes In C: The Art of Scientific Computing**, Second Edition, Cambridge University Press (1993).