

PEP2 Álgebra II

Programa de aproximación de rectas a través de los Mínimos Cuadrados

Primer Semestre 2015

Licenciatura en Ciencias de la Computación

Alumno : Danilo Abellá

Ayudante: Luis Hernan Herrera

Profesora: Mónica



Glosario

1. Introducción
2. ¿Que es el método de los mínimos cuadrados?
3. ¿Como usar el programa? (Mini Guía)
4. ¿Como trabaja internamente cada parte del programa?

1. Introducción

En este informe se pretende explicar cual es la idea del programa y del método utilizado en éste, dando a saber que beneficios y ayudas puede ofrecer , explicando claramente las operaciones matemáticas implicadas en el proyecto.

2. ¿Que es el método de los mínimos cuadrados?

El método de los mínimos cuadrados nos permite encontrar la ecuación de una recta a partir de los datos experimentales (puntos).

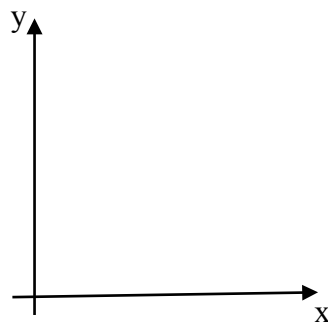
Es decir, utilizando solamente las mediciones experimentales se obtendrá la pendiente y la ordenada al origen de la recta que mejor se ajuste a tales mediciones, en otra palabras, teniendo los puntos necesarios se podrá saber que recta , parábola u cualquier otra gráfica de polinomio, pasa mas cerca de todos los puntos a la vez.

Ejemplo:

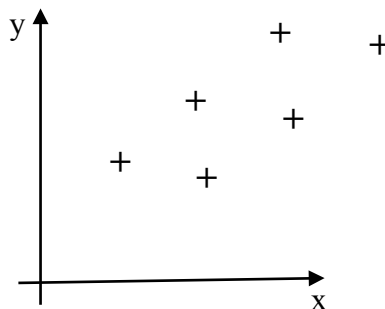
El método de los mínimos cuadrados se calcula en base al siguiente criterio:

La distancia del punto experimental a la “mejor recta” es mínima.

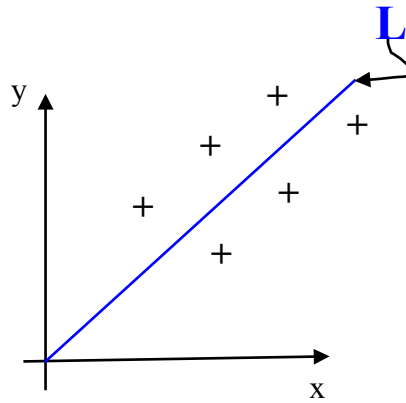
Se grafican unos
ejes de coordenadas



Graficamos todos los ptos
experimentales

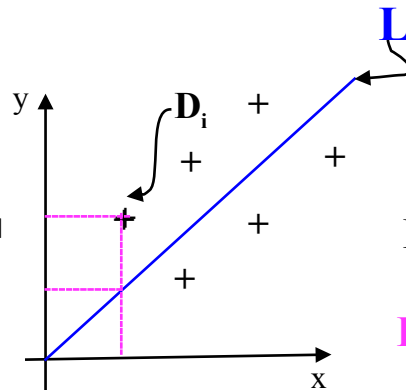


Trazamos la mejor
recta de tal manera que:



Criterio:

La distancia , D_i , del punto experimental
a la “mejor recta”, L , es mínima.



$$D_i = y_i - y(x_i)$$

$$D_i = y_i - (a_1 x_i + a_0)$$

3. ¿Como usar el programa? (Mini Guía)

El programa se hizo con la intención de ser lo mas sencillo y amigable posible para el usuario...

¿Que puedo hacer con el programa?

Al abrirlo nos muestra un menú con 5 opciones:

(1) Ingreso de puntos

(2) Evaluar en polinomios

(3) Trabajar con sistemas (x , y ,z)

(4) Cerrar gráfico

(Cualquier otra tecla) Salir

(1) Ingreso de puntos: Se ingresan los puntos con las coordenadas X , Y para luego ser trabajadas en la opción **2** (Evaluar en polinomios), con un límite de hasta 16 puntos y con valores de x y y del -25 al 25.

(2) Evaluar en polinomios: Se evalúan los puntos ingresados en la opción **1** (Ingreso de puntos) para ver qué ecuación de la recta o parábola pasa más cerca de todos los puntos más cercanos a la vez.

Se admiten polinomios de 1° a 3° grado.

Al terminar la operación nos va a mostrar todos los cálculos realizados con matrices en x y y la ecuación obtenida. Además se abrirá una gráfica mostrándonos claramente cómo se expresan los resultados dependiendo de los puntos ingresados y los polinomios seleccionados. Al abrirse el gráfico se va a activar la manipulación gráfica la cual le permite alargarlo y maximizarlo a gusto.

NOTA: Para volver al programa debe apretar una tecla con el gráfico seleccionado y volver a seleccionar el programa para volver a trabajar. UNA VEZ APRETADA LA TECLA DE SALIDA DEL GRÁFICO LA MANIPULACIÓN GRÁFICA SERÁ DESACTIVADA Y NO SE PODRÁ MANIPULAR EL GRÁFICO HASTA VOLVER A ACTIVARLO NUEVAMENTE.

(3) Trabaja con sistemas (x , y , z)

Se ingresan sistemas con las variables X , Y , Z y nos resuelven el sistema de ecuación, dándonos el valor aproximado de las tres variables. Solo admite de 3 a 6 sistemas.

(4) Cerrar gráfico

En caso de que el gráfico esté abierto lo cierra.

(Cualquier otra tecla) Salir

Cierra el programa. En caso de tener el gráfico abierto no es necesario cerrarlo antes.

4. ¿Como trabaja internamente cada parte del programa?

Primero que nada el programa fue hecho total y completamente en el lenguaje C++.

El cual opera desde una función principal (**int main()**) con un menú qu permite trabajar de forma cómoda y practica con el resto de funciones y operaciones del programa.

Para el menú se utilizaron los código **Switch()** junto a un **while()** los cuales permitieron formar de una forma mucho mas cómoda y sencilla lo que sería el trabajar en algo y volver al menú.

Para el ingreso de puntos se usó el arreglo **p[MAX]** el cual guarda los valores de **X** en las posiciones pares y los de **Y** en las impares.

En todas las funciones disponibles que trabajamos con polinomios y con sistemas se usaron matrices, las cuales fueron creados con arreglos de la estructura: **A[][]**.

Cabe decir que prácticamente TODO el programa jira entorno a los arreglos los cuales están tanto en la función principal como en prácticamente todo lo que forman las otras funciones.

Generalmente las matrices utilizadas fueron las siguientes:

A[MAX][MAX]	-Matriz A (La que recibe los datos ingresados del valor X).
B[MAX][MAX]	-Matriz B (La que recibe los datos ingresados del valor Y).
A_t[MAX][MAX]	-Matriz A^t (A transpuesta).
A_t_A[MAX][MAX]	-Matriz $(A^t)*A$ (La transpuesta de A , multiplicada luego por A).
A_t_A_inv[MAX][MAX]	-Matriz $((A^t)*A)^{-1}$ (La inversa de la matriz anteriormente mencionada).
A_t_B[MAX][MAX]	-Matriz $A*B$ (Multiplicación de A con B).
X[MAX][MAX]	-Matriz X (Matriz Final).

Para las operaciones con matrices como estas se utilizó el ciclo **For(; ;)** el cual fue perfecto para realizar operaciones tanto básicas (sacar la transpuesta o multiplicar) como complejas (sacar la inversa).También ayudó a poder mostrar os datos de las matrices y desenvolverse en un programa creado prácticamente solo con matrices.

Para la gráfica se utilizó la librería **#include<winbgim.h>** junto a la **#include<graphics.h>** las cuales permitieron la creación de una ventana y mostrar una simulación de como se comportan los polinomios gráficamente en sus distintos casos.

NOTA: Para poder entrar y compilar el código fuente del programa en code.blocks es necesario también instalarle las librerías **winbgim.h** y **graphics.h>** las cuales están disponibles en el archivo comprimido junto a un link de un video que explica clara y detalladamente como instalar las librerías.