MÍNIMOS CUADRADOS

Métodos Numéricos

Instituto Tecnológico de Monterrey Campus Guadalajara
Diego Andrés Camargo Villarreal A01224864
Manuel Becerra Marrufo A01222366

Contenido

Descripción y Requisitos

El proyecto consiste en desarrollar una herramienta que a partir de una función tabular realice un ajuste polinomial por mínimos cuadrados para determinar los coeficientes de una funcíon polinomial que se ajuste mejor a los puntos que se presentan en la función tabular.

Requisitos de Entrada/Salida

Los datos de la función tabular son proporcionados desde un archivo de texto.

Requisitos de Funcionalidad

1. El programa pide al usuario el archivo que contiene la información de la función tabular, la cual puede ser de incrementos constantes o variables.
2. El programa muestra al usuario la opción de realizar un ajuste polinomial automáticamente o un ajuste al orden polinomial indicado por el usuario.
3. El programa despliega la función polinomial de la forma:

Análisis de la Solución

Para lograr esto inicialmente es necesario leer un archivo que contenga la función tabular en un formato donde de pares de coordenadas (X y Y), donde se tiene una coordenada por línea.

Ya que se tienen las coordenadas de la función tabular, se procede a determinar de qué orden debe ser la función polinomial según la precisón que se busque para así poder realizar el cálculo de mínimos cuadrados de acuerdo al orden del polinomio seleccionado.

El algoritmo para calcular la matriz de coefientes a0, a1, a2, ----, am consta en calcular para cada posición de la matriz generada a partir del orden del polinomio. El tamaño de la matriz tiene como dimensiones:

- Columnas = orden + 2
- Filas = orden + 1

El orden “N” significa que habrá N + 1 coeficientes. Por lo que en caso de que hagamos un ajuste de orden dos nuestro cálculo sería el siguiente

$$(a_0 + a_1 x_i + a_2 x_i^2 - y_i) =$$

$$(a_0 x_i + a_1 x_i^2 + a_2 x_i^3 - y_i x_i)$$

$$(a_0 x_i^2 + a_1 x_i^3 + a_2 x_i^4 - y_i x_i^2)$$

Tomand o en cuenta i como la columna y j como el número de fila, el exponente de cada x en la ecuación es el resultado de la suma de i+j, excepto en la última columna, que simplemente es el resultado de la fila (j), y se multiplica las x^j por la yi.

Para cada uno de estos valores se calcula la sumatoria con la función tabular proporcionada por el archivo, que funciona de la siguiente manera:

$$a_0(4) + a_1 \sum x_i + a_2 \sum x_i^2 - \sum y_i = 0$$

$$a_0 \sum(x_i) + a_1 \sum x_i^2 + a_2 \sum x_i^3 - \sum x_i y_i = 0$$

$$a_0 \sum x_i^2 + a_1 x_i^3 + a_2 \sum_1^4 - \sum x_i^2 y_i = 0$$

En base a eso se genera la matriz ampluada de coeficientes

$$4a_0 + 2a_1 + 6a_2 - 14 = 0$$

$$2a_0 + 6a_1 + 8a_2 - 12 = 0$$

$$6a_0 + 8a_1 + 18a_2 - 30 = 0$$

Esto se pasa a la forma de una matriz ampliada con la que se determinan los valores de los coeficientes del polinomio a través del método de Gass-Jordan.

Una vez calculados los coeficientes, se imprime el polinomio resultante que descirbe la función que mejor se acomoda a los puntos descritos en la función tabular.

Diseño de la Solución

Clase minimosCuadrados

Atributos Privados

double * equis;
Arreglo de los valores de X
double * ye
Arreglo de los valores de Y
double Sx;
Sumatoria de los valores de X
double Sy;
Sumatoria de los valores de Y
Matrix * coeficientesNoResueltos;
Matriz donde se guarda la información
que será utilizada en Gauss-Jordan para obtener resultado
double * coeficientes;
Coeficientes del polinomio resultante
int tama;
Tamaño de la cantidad de puntos que se tiene
int tamaCoefs;
Tamaño de el arreglo que satisface el polinomio

Métodos públicos

minimosCuadrados();
Constructor por default de la clase, pide un archivo e inicializa el menú
minimosCuadrados(int tam);
Constructor que recibe un tamaño para pedir los puntos
minimosCuadrados(string archivo);
Constructor que genera las tablas desde un archivo
~minimosCuadrados();
Destructor
void generarSumatorias();
Genera las sumatorias de X & Y
void resolverMinimosCuadrados(int ord);
Resuelve por mínimos cuadrados según el orden específicado
void gaussJordan();
Resuelve por Gauss-Jordan la matriz para obtener los coeficientes
double sXOrden(int orden);

Regresa la sumatoria de X^orden

double sYOrden(int orden);

Regresa la sumatoria de Y^orden

double sXYOrden(int ordenX, int ordenY);

Regresa la sumatoria de X^ordenX * Y^ordenY

double sXYOrden(int orden);

Regresa la sumatoria de X^orden * Y^orden

double sumatoria(double * a, int n);

Regresa la sumatoria de un arreglo de tamaño n

void printCoeficientes();

Imprime el polinomio resultante

double hConstanteAll();

Checa si las diferencias son finitas o no

int ordenOptimo();

Revisa si el orden es el óptimo o no, según las diferencias finitas

void print();

Imprime los datos que se introdujeron a través del archivo

void readFile();

Permite leer un archivo

void usar();

Menú para el usuario

Clase Matrix

Atributos privados

int m;

Numero de Filas

int n;

Numero de Columnas

double **e;

Datos de la matriz

Métodos Públicos

Matrix();

Constructor por default

Matrix(int m, int n);

Constructor que define numero de filas y columnas

Matrix(int order);

Constructor que genera una matriz cuadrada de orden "order"

Matrix(int m, int n, double **e);
Constructor que genera una matriz de orden m x n con los datos e
Matrix(int order, double **e);
Constructor que genera una matriz cuadrada de orden "order" con los datos e
Matrix(Matrix *b);
Constructor que copia una matriz
Matrix(string filename);
Constructor que genera una matriz a partir de una archivo
~Matrix();
Destructor
double getElement(int i, int j);
Regresa el elemento en la posición i,j
int getRows();
Regresa la cantidad de filas
int getColumns();
Regresa la cantidad de columnas
void setElement(int i, int j, double value);
Inserta el elemento value en la posición i,j
void setIdentity();
Modifica una matriz para volverla identidad
void setNull();
Modifica la matriz para convertirla en nula
bool resolverPorGaussJordan(Matrix *&sol);
Resuelve por Gauss-Jordan una matriz regresandola a través de una referencia de memoria
double * resolverPorGaussJordan(int n, Matrix * ampliada);
Resuelve por Gauss-Jordan y regresa los coeficientes en un arreglo de doubles
Matrix * sum(Matrix *b);
Devuelve el resultado de la matriz del objeto más la mátriz b
void print();
Imprime la matriz
bool isIdentity(double r);
Regresa si la matriz es identidad
Matrix * getTranspose();
Regresa la traspuesta de una matriz

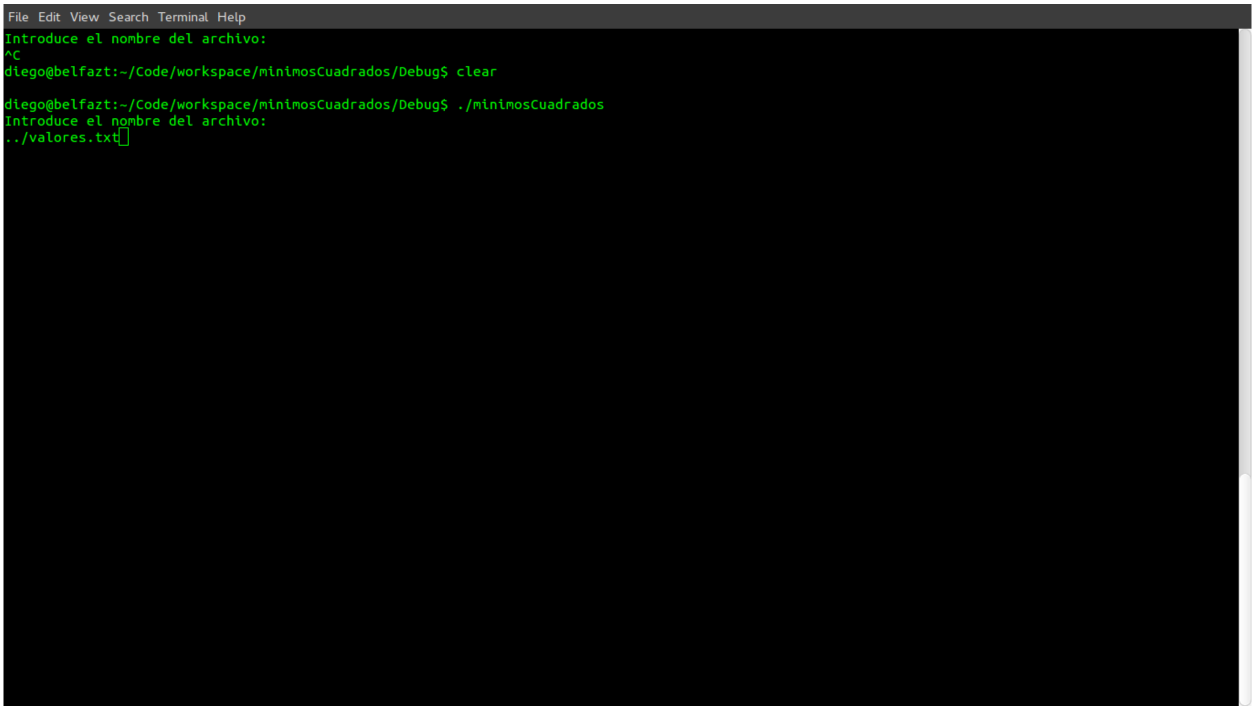
Métodos privados

double ** createMatrix(int m, int n);
Metodo que nos permite crear una matriz de tamaño m, n

Documentación del Programa

Casos de Uso

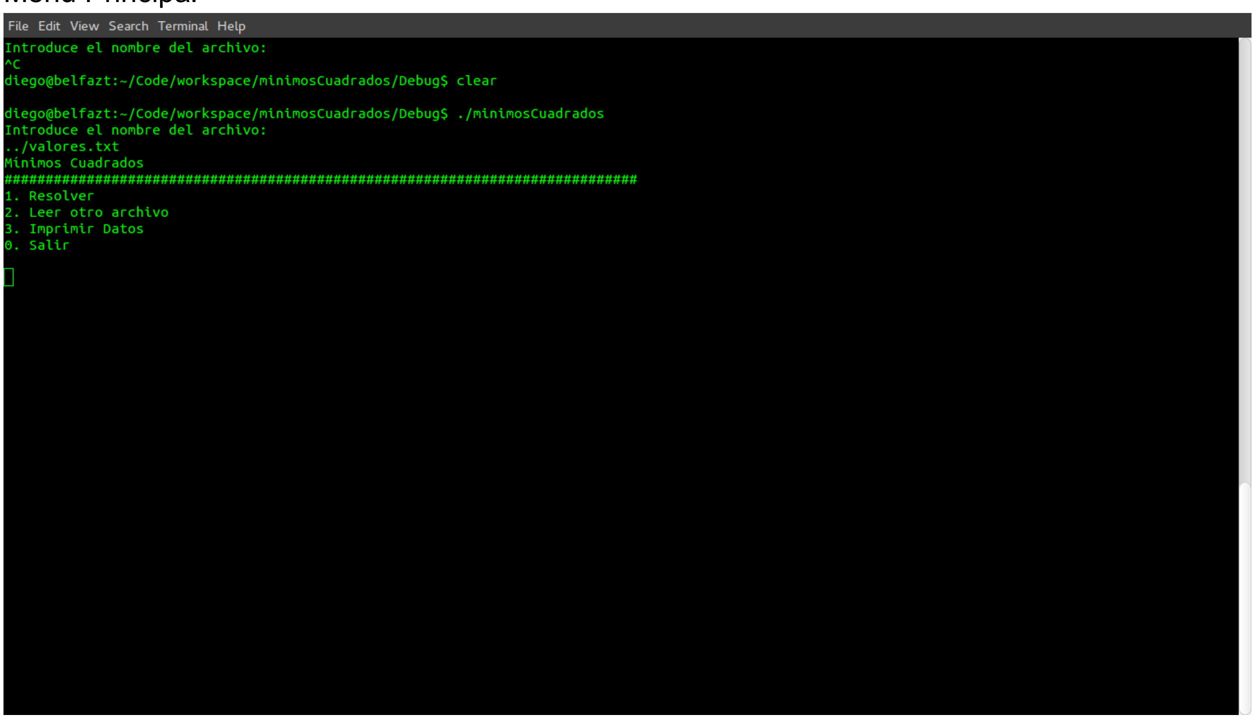
Introducción del archivo



```
File  Edit  View  Search  Terminal  Help
Introduce el nombre del archivo:
^C
diego@belfazt:~/Code/workspace/minimosCuadrados/Debug$ clear

diego@belfazt:~/Code/workspace/minimosCuadrados/Debug$ ./minimosCuadrados
Introduce el nombre del archivo:
../valores.txt
```

Menú Principal



```
File  Edit  View  Search  Terminal  Help
Introduce el nombre del archivo:
^C
diego@belfazt:~/Code/workspace/minimosCuadrados/Debug$ clear

diego@belfazt:~/Code/workspace/minimosCuadrados/Debug$ ./minimosCuadrados
Introduce el nombre del archivo:
../valores.txt
Minimos Cuadrados
#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir
```

Impresión de Datos

```
File  Edit  View  Search  Terminal  Help
Introduce el nombre del archivo:
^C
diego@belfazt:~/Code/workspace/minimosCuadrados/Debug$ clear

diego@belfazt:~/Code/workspace/minimosCuadrados/Debug$ ./minimosCuadrados
Introduce el nombre del archivo:
../valores.txt
Minimos Cuadrados
#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir

3
X | Y
-1 | 3
0 | 2
1 | 3
2 | 6
------
Sx: 2
Sy: 14

#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir
```

Resolución por Mínimos Cuadrados

```
File  Edit  View  Search  Terminal  Help
Introduce el nombre del archivo:
../valores.txt
Minimos Cuadrados
#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir

3
X | Y
-1 | 3
0 | 2
1 | 3
2 | 6
------
Sx: 2
Sy: 14

#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir

1
Resolviendo puntos, ingrese el grado que quiere resolver(-1 para automático):2
Matriz ampliada de coeficientes no resueltos:
4 2 6 14
2 6 8 12
6 8 18 30
Resultado:
2 + x^2

#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir
```

Salida del programa

```
File Edit View Search Terminal Help
#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir

3
X | Y
-1 | 3
0 | 2
1 | 3
2 | 6
------
Sx: 2
Sy: 14

#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir

1
Resolviendo puntos, ingrese el grado que quiere resolver(-1 para automático):2
Matriz ampliada de coeficientes no resueltos:
4 2 6 14
2 6 8 12
6 8 18 30
Resultado:
2 + x^2

#############################################################################
1. Resolver
2. Leer otro archivo
3. Imprimir Datos
0. Salir

0
Saliendo

diego@belfazt:~/Code/workspace/minimosCuadrados/Debug$
```

Reflexiones y Conclusiones

El desarrollo de esta herramienta fue muy interesante ya que aplicamos conocimientos y metodologías previamente utilizadas en el curso para crear una nueva solución de aproximación de polinomios más precisa. Un método que calculado a mano puede tardar más de 20 minutos, en esta aplicación se puede realizar en cuestión de milisegundos. Es por eso que los método numéricos, aunque requieran más iteraciones y no ser tan exactos con pocas iteraciones; son mucho más rápidos y se vuelven métodos más prácticos cuando se utilizan en una computadora en situaciones en que la cantidad de datos es demasiada y no se tiene un método analítico que pueda resolver el problema de manera directa.