

# Solucion de Ecuaciones No Lineales

Rubn Cuadra A01019102

May 4, 2016

### **Abstract**

Manual de usuario: 'aproxima.m', el objetivo de este texto es documentar la implementacion 2 metodos tales como Interpolacion y metodo del minimo cuadrado para obtener coeficientes de polinomios.

## 0.1 Introduccion

El codigo consiste en un archivo llamado de la misma manera que la funcion el cual recibe 5 argumentos y nos regresa 2 respuestas y una grafica

```
function [A E] = aproxima(x, y, n, o , p)
```

**X** Es un vector ordenado que contiene los componentes x de los puntos por los que pasara el polinomio **Y** Es un vector ordenado que contiene los componentes y de los puntos por los que pasara el polinomio **N** Es el grado del polinomio que deseamos obtener. **O** Es una bandera True(1) o False(0), Donde True nos generara el polinomio por el metodo de minimo cuadrado, mientras que un falso lo obtendra con interpolacion **P** Son una serie de valores en X. Una vez que se obtiene el polinomio se evalua con estos puntos en X

## 0.2 Comprension del algoritmo

### 0.2.1 Interpolacion

La interpolacion nos genera un polinomio de grado n donde n = numero de puntos -1. El ejemplo mas claro es cuando uno tiene 2 puntos, se necesitara una recta para unirlos (Grado 1).

Paso 1

Inicializar matriz vacia

Paso 2

Iteramos por cada valor en el Vector X

Paso 3

Agregamos a la matriz nueva de nxn el vector  $x_{Actual}^{totalDeVectores-iteracionActual}$

Paso 4

En este punto tendremos una matriz b con 'Resultados' que queremos, despejamos  $Ax=b$  y nos dara que los coeficientes  $(x) = inv(A)*b$ . Realizamos esta operacion

Paso 5

Regresamos los coeficientes

### 0.2.2 Minimo Cuadrado

El minimo cuadrado consiste en realizar regresiones a traves de iteraciones para generar una funcion del grado deseado

Paso 1

Obtener la cantidad de vectores en X

Paso 2

Generamos un vector  $b$  que contenga la Suma de potencias en  $X$  por  $Y$  (Iteracion hasta grado deseado)

Paso 3

Generamos otro vector  $p$  que tendra las potencias en  $X$  (Iteracion hasta grado deseado)

Paso 4

Movemos el Vector  $p$  a una matriz denominada  $A$  (Iteracion hasta grado deseado)

Paso 5

Obtenemos la  $\text{inv}(A) * b$ , donde tendremos los coeficientes del polinomio

### 0.2.3 Aproxima - Grafica

Despues de obtener los coeficientes por alguno de los dos metodos procedemos a graficar

Paso 1

Obtenemos un vector ordenado  $P$  con los resultados de evaluar los puntos extras en el polinomio generado por los metodos anteriores

Paso 2

Evaluamos el polinomio en intervalos de cada 0.05, el origen lo define el valor minimo entre el vector  $X$  y el vector de puntos extra, asi como el final lo define el maximo de estos mismos

Paso 3

Inicializamos un plot que pinte de rojo todo el polinomio, de inicio a fin cada 0.05 lugares

Paso 4

A la recta le pintamos los vectores  $(x,y)$  que nos dieron en un inicio, de color negro con una  $X$

Paso 5

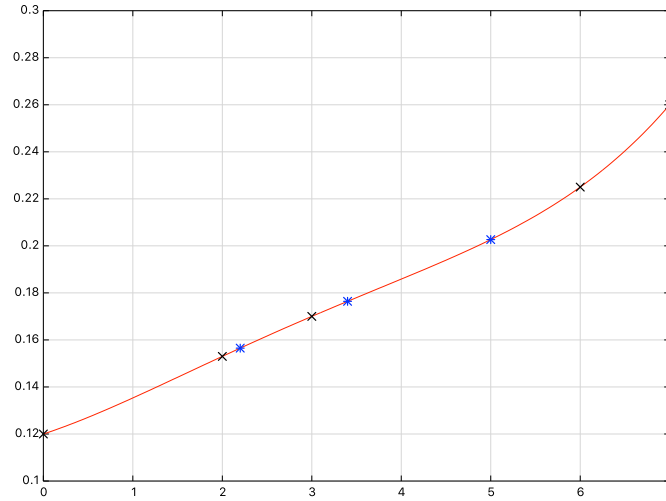
Agregamos los puntos extra que ya evaluamos, se pintan en azul con un \*

Al final se regresan 2 valores,  $\mathbf{A}$ , que es el vector de Coeficientes y  $\mathbf{E}$  que nos mandara un 1 si no se puede realizar la interpolacion con el grado de polinomio solicitado

## 0.3 Ejemplo

Mandamos a llamar la funcion desde un archivo main.m. Usando los argumentos:

Figure 1: Grafica del ejemplo.



```
X = [0, 2, 3, 6, 7];
Y = [0.120, 0.153, 0.170, 0.225, 0.260];
N = 4;
O = 0;
P = [2.2, 3.4, 5.0];
```

Llamamos la funcion:

```
[A E] = aproxima(X,Y,N,O,P)
```

y nos devolveria

```
A = [1.2000e-01
      1.2533e-02
      3.8278e-03
      -1.1333e-03
      1.0556e-04]
E = 0
```

## 0.4 Conclusion

Es un modulo portable que posee varias excepciones, esta facil de implementar y todo bien bien comentado

---

<sup>0</sup>Los acentos no se pudieron agregar por cuestion de la codificacion