



Centro de Investigación en Matemáticas, A.C.

Descripción tarea - Métodos numéricos

Erick Salvador Alvarez Valencia

15 de Octubre de 2017

1. Aproximación de una función por mínimos cuadrados

1.1. Descripción

El método a describir es la aproximación de una función, en este caso de grado dos (una parábola) por medio del método de mínimos cuadrados, el cual, dado un conjunto de puntos $(x_i, y_i)_{i=1}^m$ podemos encontrar la función que mejor se aproxime a ellos por medio de una minimización del error $\|Ax = y\|$ donde A es una matriz que contiene familias de funciones, x contiene los coeficientes desconocidos y y las ordenadas de los puntos anteriormente definidos.

La idea de este método es que una vez tengamos definido el sistema a minimizar sus error, obtengamos su gradiente y lo igualemos a cero, para lo cual llegaremos a la siguiente igualdad: $E(a, b, c) = \sum_{i=1}^m E_i^2 = \|Ax - y\|^2$ y finalmente llegando a: $\nabla E = 0 \Rightarrow A^T Ax = A^T y$. Al resolver ese sistema podremos encontrar el vector x el cual contiene los coeficientes de la mejor aproximación a la función de grado dos. Queda notar que esto se puede extender a polinomios de grado n .

1.2. Ejemplo de ejecución

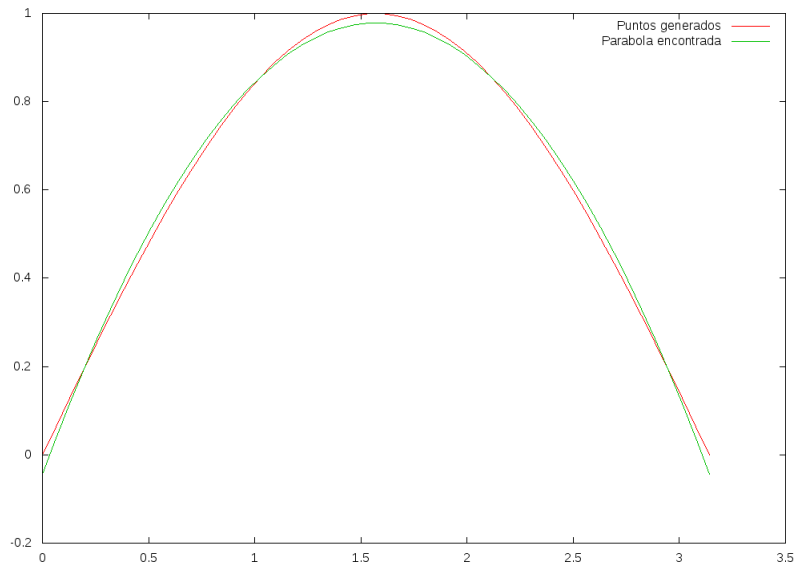
A continuación se mostrará el resultado de la ejecución del método de mínimos cuadrados donde los puntos (x_i, y_i) fueron generados usando medio ciclo de la función seno, y con una discretización de 50 puntos.

```
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 7/tarea7_ErickAV$ ./main
a: -0.414679, b: 1.302752, c: -0.044525
Error: 0.131001
ericksav22@Erick-A-UB-GL552VM:~/Documentos/Maestria/GIT/Semestre 1/Métodos numéricos/Tarea 7/tarea7_ErickAV$
```

(a) Figura 1. Resultados de la aproximación por mínimos cuadrados.

En la Figura 1. podemos observar que el programa nos arroja los coeficientes de la parábola que mejor se ajusta a los puntos generados. De la misma forma también se muestra el error entre los y_i y los $f(x_i)$ donde $f(x)$ es la función encontrada.

Para una mejor ejemplificación se mostrará una gráfica de los resultados anteriores.



(b) Figura 2. Función generada (seno) en contra de la función encontrada (parábola).

1.3. Compilación y ejecución

Para compilar: En la carpeta encontraremos los archivos `.c` y `.h` con los que se podrá compilar el ejecutable. De la misma forma, en conjunto con los archivos anteriores, también podremos encontrar un `Makefile` para, en caso de encontrarse en linux, compilar de manera sencilla.

1. **Compilar usando Makefile:** En la terminal, nos colocamos en el directorio donde se encuentre el programa, y ejecutamos el comando `make`, automáticamente se realizará la compilación y se generará el ejecutable. El `Makefile` también contiene el comando `make clean` el cual limpiará los archivos generados por la compilación, incluyendo el ejecutable.
2. **Compilar directamente:** De la misma forma, podemos compilar directamente usando los siguientes comandos (en terminal):

- `gcc -c main.c`
- `gcc -c memo.c`
- `gcc -c matriz_vector.c`
- `gcc -c met_num.c`

- `gcc -o main main.o memo.o matriz_vector.o met_num.o -lm`

Para ejecutar: Únicamente debemos de usar el comando `./main` para ejecutar el programa en consola, este no recibe argumentos.

El programa imprimirá los coeficientes a , b , y c encontrados, el error generado por las funciones, y un archivo de texto con los puntos x_i , y_i y $f(x_i)$ mediante el cual se generó el gráfico con GNUplot.