Práctica 0: Octave

Grupo 13:

* David Ortiz Fernández
* Andrés Ortiz Loaiza

En esta práctica se pretende calcular la integral F(x) de una función real positiva e integrable f(x) en un intervalo x ∈ [a; b] utilizando para ello el método de Montecarlo.

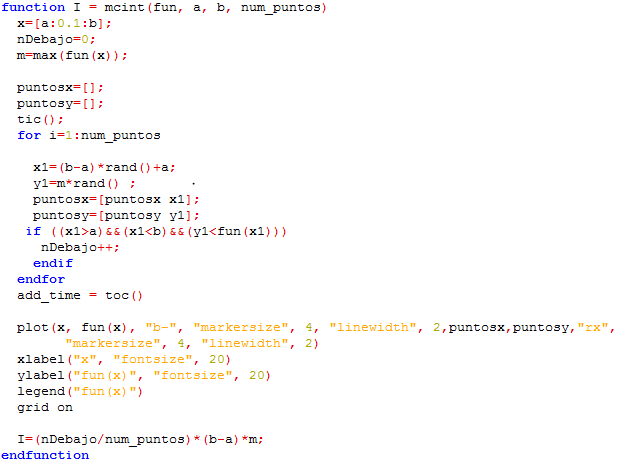
Hemos realizado dos implementaciones

1-Implementación iterativa

2-Implementación vectorial

**1-Implementación iterativa**

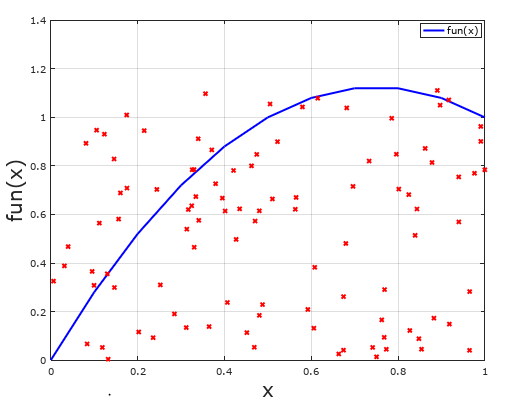
Código:



Como se puede apreciar en el código, mediante un bucle for vamos generando puntos aleatorios en el rango x ∈ [0; 1], los cuales compararemos con el valor obtenido de despejar en la función para ver si están por encima o debajo de la gráfica, y de esta manera iremos incrementando los puntos que quedan por debajo.

Posteriormente se pinta la gráfica y se muestra por pantalla

Para fun(x)= -2\*x.^2 + 3\*x con x ∈ [0; 1] y para 100 puntos generados aleatoriamente hemos obtenido la siguiente gráfica:



Resultados:



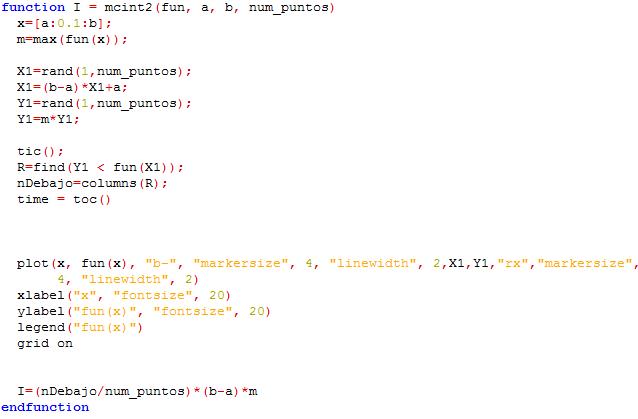
Resultados al aplicar la función quad de octave:



Como podemos apreciar el resultado al aplicar la función que hemos implementado es bastante aproximado al obtenido mediante la función quad sobre todo al ir aumentando el número de puntos generados aleatoriamente.

**2-Implementación vectorial.**

Código

****

El objetivo es el mismo que en el método iterativo , solo que esta vez lo haremos usando operaciones vectoriales.

Generamos en X1 e Y1 “num\_puntos” aleatorios.

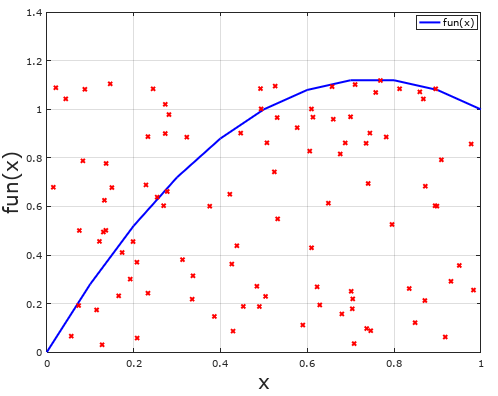
En X1 utilizamos un rango de a-b.

En Y1 utilizamos números aleatorios en un rango de 0 al máximo de la función.

Acto seguido empezamos a calcular el tiempo que van a tomar las operaciones con tic().

En R almacenamos los valores de Y1 que están por debajo de la gráfica ( fun(X1) ) y los contamos en nDebajo.

Actualizamos el tiempo que hemos tardado mediante toc() y pintamos el plot para visualizar la gráfica.

****

Resultado:

****

Resultados al aplicar la función quad de octave:



Se puede apreciar que en este caso la implementación resulta más eficiente que la anterior si nos fijamos en los tiempos de cálculo, en el caso de la implementación vectorial es necesario incrementar el orden de magnitud de los puntos generados aleatoriamente para que el tiempo de cálculo sea distinto de 0.