

Programación Concurrente y de Tiempo Real

Grado en Ingeniería Informática

Asignación de Prácticas Número 1

Se le plantean a continuación un conjunto de ejercicios sencillos de programación, que debe resolver de forma individual como complemento a la primera sesión práctica. Para cada uno, debe desarrollar un programa independiente que lo resuelva.

1. Ejercicios

1. Escriba un programa en java para encontrar el cero de una función $f(x)$ mediante el método de *Newton-Raphson*. Este método iterativo construye una sucesión x_0, x_1, x_2, \dots de aproximaciones a la solución utilizando la siguiente ecuación:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

La aproximación inicial será introducida por teclado, junto con el número de iteraciones que permitirán obtener la aproximación a la raíz de la función $f(x)$. El programa irá imprimiendo en pantalla las sucesivas aproximaciones que va calculando. Aplique su programa a las funciones siguientes:

- $f(x) = \cos(x) - x^3$ en $[0, 1]$
- $f(x) = x^2 - 5$ en $[2, 3]$

Verifique la corrección de sus soluciones usando un sencillo graficador de funciones (por ejemplo <https://www.geogebra.org/graphing?lang=es>). Guarde su programa en un fichero llamado `NewtonRaphson.java`.

2. La integral definida en $[0-1]$ de una función real de variable real $f(x)$ puede calcularse mediante un método de Monte-Carlo (probabilístico) inscribiendo la curva de la función en un cuadrado de lado igual a la unidad. Para aproximar el valor de la integral, se generan puntos aleatorios en el marco determinado por el cuadrado, y se cuentan únicamente aquellos puntos que están situados bajo la curva. La razón entre el número de puntos bajo la curva y el número total de puntos es una aproximación al valor buscado que naturalmente, conforme

mayor es el número de puntos, mejora la aproximación. Escriba un programa en Java que permita realizar tal cálculo, leyendo desde teclado el número de puntos con el cuál generar la aproximación para las funciones siguientes:

- $f(x) = \sin$

- $f(x) = x$

Verifique ahora la corrección de su trabajo utilizando cualquier integrador on-line (por ejemplo <https://www.integral-calculator.com/>). Guarde el programa en `intDefinidaMonteCarlo.java`