

PRINCIPIO DE COMPUTADORES
Curso 2022-2023
Guión Práctica 1

Descripción.

Te proponemos realizar en ensamblador un programa para calcular la aproximación de la función $\sin(x)$ en un punto comprendido en el intervalo $[-1,1]$, y con un error estimado.

La función $\sin(x)$ en el intervalo $[-1,1]$ se puede aproximar a un polinomio que tiene la siguiente forma:

$$\Rightarrow \sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

para $n \geq 0$

Vamos a suponer, que el error que se comete en cada iteración es:

$$\text{error cometido} = \left| \frac{\text{aproximación actual} - \text{aproximación anterior}}{\text{aproximación actual}} \right|$$

Ejemplo, si lo hacemos manualmente:

$\sin(0.5)$ según la calculadora científica es 0.4794255386

$n=0$ iteraciones, polinomio = x , polinomio = 0.5, error cometido no determinado

$n=1$ iteraciones,

polinomio = $x - (x^3/3!)$

polinomio = 0.47916

error cometido = $| (0.47916 - 0.5) / 0.47916 | = 0.04349$

$n=2$ iteraciones,

polinomio = $x - (x^3/3!) + (x^5/5!)$

polinomio = 0.47942

error cometido = $| (0.47942 - 0.47916) / 0.47942 | = 0.00054$

El programa deberá solicitar los siguientes datos de entrada:

- Punto en el que debe evaluarse el polinomio (x , número flotante doble precisión).
- Error máximo calculado que debe cometerse en la aproximación (error, número flotante doble precisión).

El programa deberá mostrar por consola información sobre:

- Resultado de la aproximación de $\sin(x)$ para el punto x introducido por teclado.
- error calculado en la aproximación.
- Número de iteraciones (términos calculados, n).

Deberá comprobarse siempre que el número x se encuentra dentro del intervalo $[-1,1]$, solicitando nuevamente que se introduzca si no se encuentra en dicho intervalo.

Deberá además comprobarse que el error máximo introducido es menor que 1. Si se introduce un error máximo mayor o igual que uno deberá ser solicitado nuevamente. Si se introduce un error menor o igual que cero, el programa deberá finalizar.

NOTA: Trata de optimizar tu programa reutilizando cálculos previos.

Dejaremos en el campus el código C++ que resuelve este problema, y también pondremos a tu disposición una aplicación web para pasar tests a tu práctica. Antes de ver el código en C++ que te dejaremos intenta hacer una versión propia tuya (te ayudará a comprender el algoritmo). Recuerda que si no lo entiendes puedes preguntar a tu profesor de prácticas en una tutoría.

Te daremos también un esqueleto del programa en MIPS con las cadenas de textos ya definidas, que debes usar para no tener problemas con la aplicación para hacer tests.

Ejemplo de ejecución:

```

● ● ● Console

Aproximación a sen(x) (-1 <= x <= 1) con un error máximo, usando Taylor

Introduzca el valor de x (-1 <= x <= 1): 2

Introduzca el valor de x (-1 <= x <= 1): 0.5

Introduzca el error maximo permitido en la aproximación (0 < error < 1) (error <= 0 sale del programa): 0.0001

sen(x) calculado: 0.47942553323412701
error calculado: 3.23324291025955825e-06
numero de iteraciones calculadas: 3

Introduzca el valor de x (-1 <= x <= 1): 1

Introduzca el error maximo permitido en la aproximación (0 < error < 1) (error <= 0 sale del programa): 0.000001

sen(x) calculado: 0.841470984648068021
error calculado: 2.97718030031020764e-08
numero de iteraciones calculadas: 5

Introduzca el valor de x (-1 <= x <= 1): 0

Introduzca el error maximo permitido en la aproximación (0 < error < 1) (error <= 0 sale del programa): 0

Fin del programa
|
```