

Pseudocódigo

Scribe

Función Factorial (n)

Fact = 1

para i desde 1 hasta n

Fact = Fact * i

Fin For

retornar Fact

Fin Función

Función F1(x) → retornar $\exp(3 \cdot x)$

Función F2(x) → retornar $\sin(z \cdot x) + \cos(x)$

Función F3(x) → retornar $\log(x+1)$

Función F4(x) → retornar $\cos(5 \cdot x)$

Función F5(x) → retornar $\sin(x)$

Función derivada_k_F1(x, k)

retornar $3^k \cdot \exp(3 \cdot x)$

Función derivada_k_F2(x, k)

mod = k mod 4

si mod = 0 → retornar $\sin(z \cdot x) + \cos(x)$

si mod = 1 → retornar $z \cdot \cos(z \cdot x) - \sin(x)$

si mod = 2 → retornar $-4 \cdot \sin(z \cdot x) - \cos(x)$

retornar $-8 \cdot \cos(z \cdot x) + \sin(x)$

Función derivada_k_F3(x, k)

signo = (k mod 2 = 0)

Fact = Factorial(k-1)

si k = 0 → retornar $\log(x+1)$

retornar signo * Fact / pow(x+1, k)

Función derivada_k_F4(x, k)

mod = k mod 4

Factor = pow(5, k)

si mod = 0 → retornar Factor * $\cos(5 \cdot x)$

si mod = 1 → retornar -Factor * $\sin(5 \cdot x)$

si mod = 2 → retornar -Factor * $\cos(5 \cdot x)$

retornar Factor * $\sin(5 \cdot x)$

Continuación

Función derivada_k_FS(x, k)

mod = k mod 4

si mod = 0 → retornar $\sin(x)$

si mod = 1 → retornar $\cos(x)$

si mod = 2 → retornar $-\sin(x)$

retornar $-\cos(x)$

Declarar $x_i, h, aprox, R_n, error, x_i, z, m, orden, opción$
do

mostrar las Funciones y preguntar al usuario cual quiere usar

1. e^{3x} 2. $\sin(zx) + \cos(x)$

3. $\log(x+1)$ 4. $\cos(5x)$ 5. $\sin(x)$

Solicitar x_i, h y el orden n

$x_i, z = x_i + h$ $m = (x_i + x_i, z) / z$ $aprox = 0$

For n = 0 hasta orden n

switch

deriv = derivada_k_Fn(x_i, n)

deriv_medio = derivada_k_Fn($m, n+1$)

valor_real = Fn($x_i + h$)

$aprox = aprox + deriv * h^n / \text{Factorial}(n)$

$R_n = deriv_medio * h^{(n+1)} / \text{Factorial}(n+1)$

error = valor_real - aprox

print("n aprox Rn error")

end For

Preguntar: ¿Quieres usar otra función?

while respuesta = 's' o 'S'

Diagrama de Flujo

INICIO

Función para el Factorial

For

Fact = 1

i = 1; i <= n; i++

Fact *= i

end For

Función F1 = $\exp(3 * x)$

Función F2 = $\sin(z * x) \cdot \cos(x)$

Función F3 = $\log(x + 1)$

Función F4 = $\cos(s * x)$

Función F5 = $\sin(x)$

Derivar las Funciones

F1 $\Rightarrow 3^k * \exp(3 * x)$

F2 $\Rightarrow \text{mod} = k \% 4$

Si mod = 0 $\Rightarrow \sin(z * x) \cdot \cos(x)$

Si mod = 1 $\Rightarrow z * \cos(z * x) - \sin(x)$

Si mod = 2 $\Rightarrow -4 * \sin(z * x) \cdot \cos(x)$

F3 $\Rightarrow \text{signe}(k \% 2 = 0)$

Factorial (k - 1)

Si k = 0 $\Rightarrow \log(x + 1)$

F4 $\Rightarrow \text{mod} = k \% 4$

Factor pow(s, k)

Si mod = 0 $\Rightarrow \text{Factor} * \cos(s * x)$

Si mod = 1 $\Rightarrow -\text{Factor} * \sin(s * x)$

Si mod = 2 $\Rightarrow \text{Factor} * \cos(s * x)$

F5 $\Rightarrow \text{mod} = k \% 4$

Si mod = 0 $\Rightarrow \sin(x)$

Si mod = 1 $\Rightarrow \cos(x)$

Si mod = 2 $\Rightarrow -\sin(x)$

Pedacar

Float xi, h, aprox, Rn,

error, xiz, m

int orden, opción

char respuesta

FIN

do While

¿Quieres usar otra Función?

respuesta = 's' o 'S'

Print("n aprox Rn error")

$R_n = \text{deriv_medio} * h^{(n+1)} / \text{Factorial}(n+1)$

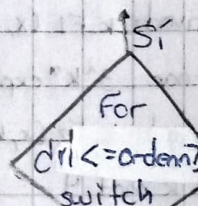
$\text{aprox} = \text{aprox} + \text{deriv} * h^n / \text{Factorial}(n)$

$\text{error} = \text{valor_real} - \text{aprox}$

$\text{deriv} = \text{derivada_k_Fn}(xi, n)$

$\text{deriv_medio} = \text{derivada_k_Fn}(m, n+1)$

$\text{valor_real} = \text{Fn}(xi + h)$



FIN

$xi = x; \text{th} = (xi + xi) / 2; \text{aprox} = 0$

Solicitar xi, h, orden n

Print("¿Cuál Función quieres utilizar?")

Mostrar al usuario las Funciones

1. $\exp(3 * x)$

2. $\sin(z * x) + \cos(x)$

3. $\log(x + 1)$

4. $\cos(s * x)$

5. $\sin(x)$

Cambie por completo mi diagrama de Flujo y por ende el pseudocódigo ya que el inicial lo hice en base al error que el usuario ingresaba, sin embargo, no lo pude realizar, porque al momento de tener un error muy pequeño, se necesitaban demasiadas derivadas, que no se podían observar en los resultados al ser números muy grandes, así que pte por realizar Funciones para cada una de ellas Fuera del int main, agregando un For y dentro de él un switch para cada caso o Función que se tiene, esto lo hace más fácil, ya que el usuario puede decidir cuantas iteraciones hacer y observar el error que tiene y puede ingresar más derivadas hasta que sea lo que espera, es por eso que cambie el diagrama y pseudocódigo