

Metodos Numéricos

Adolfo Hernández Ramírez

TAREA #. Programa para determinar $f(x)$ usando Series de Taylor.

1. Declarar las variables doubles: $x, x_i, \text{factorial}, h, \text{error}, \text{residuo}, \text{función}.$
2. Declarar variables enteras: $n, f, \text{opción}, \text{repetir}.$
3. Definir las 5 funciones a evaluar.

Double $f_1(\text{double } x) \{ \text{return } f(x) \}$

Donde $f(x)$ sera cualquier función que decidamos en este caso sera:

$\text{Sen}(x), \text{Cos}(x), \ln(1+x)$ para $x \neq 1$
 $(1/(1-x)), e^x;$

1. Definir una función donde se almacene y calculen los factoriales para cada n .

double factorial(int n)

Si $n == 0$ entonces factorial = 1

Sino

double $f = 1$.

for(int $i = 1; i \leq n; i++ \{$

$f = f * i$

return f .

- Si Definir la función para la serie de Taylor

double Taylor(double $df[n]$ / (double)(int orden, x, a)

Definir la variable double $\text{sumatoria} = 0$

y con un for(int $n = 0; n \leq \text{orden}; n++$)

Calcular la suma de la serie de Taylor

$\text{suma} += df[n] * \text{pow}(h, n) / \text{factorial}(n)$

Donde $h = x - x_i$ y $df[n]$ es un arreglo para cada derivada.

6. Definir como funciones las derivadas de $f(x)$

double fj - di (double x) { Return (f'(x)) }

Para este caso serán 5 funciones, 5 derivadas donde las derivadas nosotros las determinamos y las ponemos en la función.

7. Dentro del int main () {

8. Hacer un do-while, para que el programa se ejecuta cuantas veces quiera el usuario.

9. DO {

10. Mostrar en la pantalla las 5 funciones asignando un número para cada una.

11. Con scanf registrar la opción que desee el usuario.

12. Pedir que también ingrese el valor de x y x_i .

13. Definir un arreglo df que reciba para cada caso las derivadas de las funciones.

14. Definir otro arreglo para almacenar las funciones para cada caso F.

15. Colocar un switch para que de acuerdo a la opción que el usuario registró se hagan las asignaciones de las derivadas.

$$df[\theta] = f_1 \cdot d\theta \dots \dots df[S] = f_5 \cdot dS$$

$F = F_1$

Para cada caso colocar un break

16. Si el usuario coloca o ingresa un número que no sea 1-5 entonces imprimir que la opción es inválida.

17. Imprimir una tabla de resultados, que contenga n , $f(x)$, R_n y Error

18. En un ciclo for ($n=0$, $n \leq 4$, $n++$) {

Calculamos el error, definiendo una variable llamada aproximación

$\text{double aproximación} = \text{taylor}$ Ya que aquí hicimos la suma de términos para cada n

Establecer que el error sea igual a

$$\text{error} = \text{fabs}(fx - \text{aproximación})$$

19. Dentro del mismo for calculamos el residuo.

$$\text{Residuo} = \frac{df(n+1)(x) * \text{pow}(h, n+1)}{\text{factorial}}$$

Imprimir los valores de n , fx , aproximación y residuo y error con un `printf`.

20. Preguntar al usuario si desea realizar otro cálculo (si = 1 y no = 0)

21. Registrar con scanf su respuesta (`scanf("%d", &repetir)`)

22. `while (Repetir == 1);`