Instituto Tecnológico de Costa Rica
Área Académica Ingeniería en Computadores
CE-3201: Análisis Numérico para Ingeniería

Manual Funtras

Integrantes

Josué Araya García
Jonathan Guzmán Araya
Mariano Muñoz Masís
Daniel Prieto Sibaja

Cartago, Costa Rica
27 Marzo, 2021

Índice

1. Introducción

En las matemáticas existen diversos tipos de funciones como lo pueden ser:

- Algebraicas
- Trascendentes

Para este desarrollo nos enfocaremos en las funciones trascendentes, estas son las funciones que no satisfacen una ecuación polinomial cuyos coeficientes sean a su vez polinomios; esto contrasta con las funciones algebraicas, las cuales satisfacen dicha ecuación.

1.1. ¿Qué es Funtras?

Funtras es una biblioteca de funciones trascendentes desarrolladas en el lenguaje C^{++} con el objetivo de aproximar dichas funciones mediante el uso de métodos iterativos utilizando únicamente operaciones de suma, resta, multiplicación y potencia con una cantidad de iteraciones máximas de 2500 y una tolerancia de 10^{-8}.

2. Requisitos e Instalación

En esta sección se abarcarán los requisitos mínimos para su ejecución así como una breve guía de instalación de la misma.

2.1. Requisitos

El desarrollo y las pruebas de esta biblioteca se realizaron en el SO Windows 10, por lo tanto como requisitos se tiene:

- Sistema operativo Windows 10
- MinGW
- CLion

2.2. Instalación

3. Funciones implementadas en Funtras

A continuación se detallan las funciones implementadas en la biblioteca funtras.

3.1. Inverso multiplicativo a^{-1}

Esta función calcula el inverso multiplicativo, recíproco o inverso de un número x real positivo, el uso de la misma se realiza de la siguiente manera:

$$\boxed{\text{varM1(a)}}$$

3.1.1. Formulación matemática

El cálculo se realiza mediante el método iterativo que se describe a continuación.

$$x_{k+1} = x_k(2 - a \cdot x_k) \tag{1}$$

3.1.2. Valores iniciales

El valor de x_0 esta dado por:

$$x_0 = \begin{cases} eps^{15} \;\; si \;\; 80! < a \leq 100! \\ eps^{11} \;\; si \;\; 60! < a \leq 80! \\ eps^{8} \;\; si \;\; 40! < a \leq 60! \\ eps^{4} \;\; si \;\; 20! < a \leq 40! \\ eps^{2} \;\; si \;\; 0! < a \leq 20! \end{cases}$$

donde eps es una constante ya definida con valor de:

$$eps = 2{,}2204x10^{-16} \tag{2}$$

3.1.3. Condición de parada

La condición de parada de la iteración está dada por:

$$\left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| \tag{3}$$

Cuando la tolerancia dada sea mayor que esta, entonces devuelve el resultado obtenido.

3.1.4. Ejemplos numérico

3.2. Exponencial de Euler e^x

Esta función calcula el exponencial de e elevado a un número natural x, el uso de la misma se realiza de la siguiente manera:

exp_t(x)

3.2.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \sum_{n=0}^{k} \frac{a^n}{n!} \tag{4}$$

3.2.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{5}$$

3.2.3. Ejemplo numérico

3.3. Seno $\sin(x)$

Esta función calcula el *seno* de un número x, el uso de la misma se realiza de la siguiente manera:

sin_t(x)

3.3.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \sum_{n=0}^{k} (-1)^n \frac{a^{2n+1}}{(2n+1)!} \tag{6}$$

3.3.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{7}$$

3.3.3. Ejemplo numérico

3.4. Coseno $\cos()x)$

Esta función calcula el *coseno* de un número x, el uso de la misma se realiza de la siguiente manera:

cos_t(x)

3.4.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \sum_{n=0}^{k} (-1)^n \frac{a^{2n}}{(2n)!} \tag{8}$$

3.4.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{9}$$

3.4.3. Ejemplo numérico

3.5. Tangente $\tan(x)$

Esta función calcula la *tangente* de un número x, el uso de la misma se realiza de la siguiente manera:

tan_t(x)

3.5.1. Formulación matemática

La función tangente se puede componer a partir de otras como lo son *seno* y *coseno*, es por ello que el calculo de la misma se realiza mediante la siguiente ecuación:

$$tan(x) = sen(x) \cdot cos(x)^{-1} \tag{10}$$

3.5.2. Ejemplo numérico

3.6. Logaritmo natural $\ln(x)$

Esta función calcula el *logaritmonatural* de un número x, el uso de la misma se realiza de la siguiente manera:

ln_t(x)

3.6.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \frac{2(a-1)}{a+1} \sum_{n=0}^{k} \frac{1}{2n+1} \left(\frac{a-1}{a+1} \right)^{2n} \tag{11}$$

3.6.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{12}$$

3.6.3. Ejemplo numérico

3.7. Logaritmo $\log_a(x)$

Esta función calcula el *logaritmo* de base a a un número x, el uso de la misma se realiza de la siguiente manera:

log_t(x)

3.7.1. Formulación matemática

La función logaritmo se puede componer a partir de otra como $logaritmonatural$, es por ello que el calculo de la misma se realiza mediante la siguiente ecuación:

$$\log_a(x) = \ln(x) \cdot (\ln(a))^{-1} \tag{13}$$

3.7.2. Ejemplo numérico

3.8. Exponencial a^x

Esta función calcula el exponencial de un número a elevado a un número x, el uso de la misma se realiza de la siguiente manera:

power_t(x)

3.8.1. Formulación matemática

3.8.2. Valores iniciales

3.8.3. Condición de parada

3.8.4. Ejemplo numérico

3.9. Seno hiperbólico $\sinh(x)$

Esta función calcula el *senohiperblico* de un número x, el uso de la misma se realiza de la siguiente manera:

sinh_t(x)

3.9.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \sum_{n=0}^{k} \frac{a^{2n+1}}{(2n+1)!} \tag{14}$$

3.9.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{15}$$

3.9.3. Ejemplo numérico

3.10. Coseno hiperbólico $\cosh(x)$

Esta función calcula el *cosenohiperblico* de un número x, el uso de la misma se realiza de la siguiente manera:

```
cosh_t(x)
```

3.10.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \sum_{n=0}^{k} \frac{a^{2n}}{(2n)!} \tag{16}$$

3.10.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{17}$$

3.10.3. Ejemplo numérico

3.11. Tangente hiperbólico $\tanh(x)$

Esta función calcula la *tangentehiperblica* de un número x, el uso de la misma se realiza de la siguiente manera:

```
tanh_t(x)
```

3.11.1. Formulación matemática

La función tangente hiperbólico se puede componer a partir de otras como lo son *senohiperblico* y *cosenohiperblico*, es por ello que el calculo de la misma se realiza mediante la siguiente ecuación:

$$tanh(x) = senh(x) \cdot cosh(x)^{-1} \tag{18}$$

3.11.2. Ejemplo numérico

3.12. Raíz cuadrada $\sqrt{x}$

Esta función calcula la $razcuadrada$ de un número x mediante el método de Newton-Raphson, el uso de la misma se realiza de la siguiente manera:

sqrt_t(x)

3.12.1. Formulación matemática

El método de Newton-Raphson se define como:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},\ x_0 = \alpha \tag{19}$$

Donde para encontrar la p-ésima raíz de a:

$$\sqrt[p]{a} \tag{20}$$

El cero de la función está dado por:

$$g(x) = x^p - a,\ x_0 = \frac{a}{2} \tag{21}$$

Por lo que resulta en la siguiente iteración:

$$\begin{cases} x_0 = \frac{a}{2} \\ x_{k+1} = x_k - \frac{x_k^2 - a}{2 \cdot x_k} \end{cases}$$

3.12.2. Valore inicial

El valor inicial está dado por: $x_{0=\frac{a}{2}}$.

3.12.3. Condición de parada

3.12.4. Ejemplo numérico

3.13. Raíz $\sqrt[a]{x}$

Esta función calcula la $raza - sima$ de un número x mediante el método de Newton-Raphson, el uso de la misma se realiza de la siguiente manera:

root_t(x)

3.13.1. Formulación matemática

3.13.2. Valores iniciales

3.13.3. Condición de parada

3.13.4. Ejemplo numérico

3.14. Arcoseno $\sin^{-1}(x)$

Esta función calcula el *arcoseno* de un número x, el uso de la misma se realiza de la siguiente manera:

asin_t(x)

3.14.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \sum_{n=0}^{k} \frac{2n!}{4^n (n!)^2 (2n+1)} a^{2n+1} \tag{22}$$

3.14.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{23}$$

3.14.3. Ejemplo numérico

3.15. Arcotangente $\tan^{-1}(x)$

Esta función calcula el *arcotangente* de un número x, el uso de la misma se realiza de la siguiente manera:

atan_t(x)

3.15.1. Formulación matemática

El cálculo se realiza mediante la sumatoria que se describe a continuación.

$$S_k(a) = \sum_{n=0}^{k} (-1)^n \frac{a^{2n+1}}{2n+1} \tag{24}$$

3.15.2. Condición de parada

La condición de parada de la iteración está dada por:

$$|S_{k+1}(a) - S_k(a)| < tol \tag{25}$$

3.15.3. Ejemplo numérico