



Área Académica de Ingeniería en Computadores
Licenciatura en Ingeniería en Computadores
CE-3102: Análisis Numérico para Ingeniería

Manual de Librería Funtras

Tarea 1 - Parte 1 : Pregunta 3

Brayan Alfaro González
alfabrayan12@gmail.com
2019380074

Sebastián Alva Vives
sebasjosue84@gmail.com
2017097108

Kevin Zeledón Salazar
kev_sala@estudiantec.cr
2018076244

Daniel Camacho González
ddcamachog1501@estudiantec.cr
2017114058

Cartago, Costa Rica
Marzo 2021

Índice general

1. Introducción	3
1.1. ¿Qué es Funtras?	3
1.2. Resumen de funciones implementadas	3
2. Instalación y Requisitos	4
2.1. Requisitos	4
2.2. Instalación y Uso	4
3. Funciones Implementadas	5
3.1. Inverso multiplicativo: x^{-1}	5
3.1.1. Descripción	5
3.1.2. Formulación Matemática	5
3.1.3. Ejemplos	5
3.2. Seno: $\sin(x)$	6
3.2.1. Descripción	6
3.2.2. Formulación Matemática	6
3.2.3. Ejemplos	6
3.3. Coseno: $\cos(x)$	6
3.3.1. Descripción	6
3.3.2. Formulación Matemática	6
3.3.3. Ejemplos	7
3.4. Tangente: $\tan(x)$	7
3.4.1. Descripción	7
3.4.2. Formulación Matemática	7
3.4.3. Ejemplos	7
3.5. Logaritmo natural: $\ln(x)$	7
3.5.1. Descripción	7
3.5.2. Formulación Matemática	7
3.5.3. Ejemplos	8
3.6. Logaritmo: $\log_a(x)$	8
3.6.1. Descripción	8
3.6.2. Formulación Matemática	8
3.6.3. Ejemplos	8
3.7. Exponencial natural: e^x	8
3.7.1. Descripción	8
3.7.2. Formulación Matemática	8
3.7.3. Ejemplos	9
3.8. Exponencial: a^x	9
3.8.1. Descripción	9
3.8.2. Formulación Matemática	9
3.8.3. Ejemplos	9
3.9. Seno hiperbólico: $\sinh(x)$	9
3.9.1. Descripción	9
3.9.2. Formulación Matemática	9

3.9.3. Ejemplos	10
3.10. Coseno hiperbólico: $\cosh(x)$	10
3.10.1. Descripción	10
3.10.2. Formulación Matemática	10
3.10.3. Ejemplos	10
3.11. Tangente hiperbólica: $\tanh(x)$	10
3.11.1. Descripción	10
3.11.2. Formulación Matemática	10
3.11.3. Ejemplos	11
3.12. Raíz: $\sqrt[n]{x}$	11
3.12.1. Descripción	11
3.12.2. Formulación Matemática	11
3.12.3. Ejemplos	11
3.13. Raíz cuadrada: \sqrt{x}	11
3.13.1. Descripción	11
3.13.2. Formulación Matemática	12
3.13.3. Ejemplos	12
3.14. Arcoseno: $\sin^{-1}(x)$	12
3.14.1. Descripción	12
3.14.2. Formulación Matemática	12
3.14.3. Ejemplos	12
3.15. Arcotangente: $\tan^{-1}(x)$	13
3.15.1. Descripción	13
3.15.2. Formulación Matemática	13
3.15.3. Ejemplos	13

Capítulo 1

Introducción

1.1. ¿Qué es Funtras?

La librería Funtras consiste en una colección de métodos de naturaleza matemática escritos en el lenguaje de programación C++ con el objetivo de aproximar múltiples funciones no polinomiales con una alta precisión. La librería se compone de quince métodos matemáticos, los cuales se aproximan con una tolerancia de 10^{-8} y con una cantidad de iteraciones máximas de 2500. Además, las funciones implementadas incluyen un chequeo del dominio para evitar intentar cálculos erróneos, como la raíz de un número negativo.

1.2. Resumen de funciones implementadas

A continuación se presenta una tabla con el resumen de las funciones implementadas y el nombre del correspondiente método en la librería:

Cuadro 1.1: Funciones implementados y el nombre de su método

Función $f(x)$	Función en la librería
x^{-1}	div_lt(x)
$\sin(x)$	sin_t(x)
$\cos(x)$	cos_t(x)
$\tan(x)$	tan_t(x)
$\ln(x)$	ln_t(x)
$\log_a(x)$	log_t(x,a)
e^x	exp_t(x)
a^x	power_t(x,a)
$\sinh(x)$	sinh_t(x)
$\cosh(x)$	cosh_t(x)
$\tanh(x)$	tanh_t(x)
\sqrt{x}	sqrt_t(x)
$\sqrt[a]{x}$	root_t(x,a)
$\sin^{-1}(x)$	asin_t(x)
$\tan^{-1}(x)$	atan_t(x)

Capítulo 2

Instalación y Requisitos

2.1. Requisitos

1. Sistema operativo Windows o Linux.
2. Compilador GCC.
3. Editor de texto.

2.2. Instalación y Uso

1. Si no se tiene, instalar el compilador GCC para C y C++:
 - En Linux se instala directamente el compilador.
 - En Windows es necesario usar un instalador como MinGW para instalar GCC.
2. Crear una carpeta que será el espacio de trabajo para el proyecto.
3. Crear un archivo de tipo .cpp (Ej: ejemplo.cpp) en el cual se usarán las funciones de la librería.
4. Copiar los archivos funtras.h y funtras.a a la carpeta de trabajo.
5. Incluir el header de la librería en el archivo ejemplo.cpp, mediante la instrucción:

```
#include "funtras.h"
```

6. Programar el script deseado en el archivo ejemplo.cpp.
7. Abrir la terminal en la carpeta de trabajo.
8. Compilar el script de la siguiente manera:

```
$ g++ -c ejemplo.cpp -o ejemplo.o
```
9. Generar un ejecutable conectado con la librería de la siguiente manera:

```
$ g++ ejemplo.o funtras.a -o ejecutable.out
```
10. Correr el ejecutable:

```
$ ejecutable.out
```

Capítulo 3

Funciones Implementadas

3.1. Inverso multiplicativo: x^{-1}

3.1.1. Descripción

Esta función calcula el inverso multiplicativo de un número x , el cual debe ser un número real positivo. La función se llama de la siguiente manera:

```
div_1t(x)
```

3.1.2. Formulación Matemática

Método Iterativo

$$eps = 2,2204 \times 10^{16} \quad (3.1)$$

$$y_0 = \begin{cases} eps^{15}, & 80! < a \leq 100! \\ eps^{11}, & 60! < a \leq 80! \\ eps^8, & 40! < a \leq 60! \\ eps^4, & 20! < a \leq 40! \\ eps^2, & 0! < a \leq 20! \end{cases} \quad (3.2)$$

$$y_{k+1} = y_k \cdot (2 - x \cdot y_k) \quad (3.3)$$

Criterio de Parada

$$|(x_{k+1} - x_k)/x_{k+1}| < 10^{-8} \quad (3.4)$$

3.1.3. Ejemplos

```
cout << div_1t(10) << endl; // output = 0.1
cout << div_1t(5) << endl; // output = 0.2
cout << div_1t(20) << endl; // output = 0.05
```

3.2. Seno: $\sin(x)$

3.2.1. Descripción

Esta función calcula el seno de un número x , el cual debe ser un número real, mediante una sumatoria. La función se llama de la siguiente manera:

```
sin_t(x)
```

3.2.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad (3.5)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.6)$$

3.2.3. Ejemplos

```
cout << sin_t(0) << endl;      // output = 0
cout << sin_t(3.14) << endl;    // output = 0.00159265
cout << sin_t(3.14/2) << endl;  // output = 1
cout << sin_t(-3.14/2) << endl; // output = -1
```

3.3. Coseno: $\cos(x)$

3.3.1. Descripción

Esta función calcula el coseno de un número x , el cual debe ser un número real, mediante una sumatoria. La función se llama de la siguiente manera:

```
cos_t(x);
```

3.3.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k (-1)^n \frac{x^{2n}}{(2n)!} \quad (3.7)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.8)$$

3.3.3. Ejemplos

```
cout << cos_t(0) << endl;      // output = 1
cout << cos_t(3.14) << endl;    // output = -0.999999
cout << cos_t(3.14/2) << endl;  // output = 0.000796327
cout << cos_t(-3.14/2) << endl; // output = 0.000796327
```

3.4. Tangente: $\tan(x)$

3.4.1. Descripción

Esta función calcula la tangente de un número x , el cual debe ser un número real, mediante el uso de otros métodos definidos. La función se llama de la siguiente manera:

```
tan_t(x)
```

3.4.2. Formulación Matemática

$$\tan(x) = \frac{\sin(x)}{\cos(x)} \quad (3.9)$$

3.4.3. Ejemplos

```
cout << tan_t(0) << endl;      // output = 1
cout << tan_t(1) << endl;      // output = 1.55741
cout << tan_t(2) << endl;      // output = -2.18504
cout << tan_t(3) << endl;      // output = -0.142547
```

3.5. Logaritmo natural: $\ln(x)$

3.5.1. Descripción

Esta función calcula el logaritmo natural de un número x , el cual debe ser un número real mayor a cero, mediante una sumatoria. La función se llama de la siguiente manera:

```
ln_t(x)
```

3.5.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k \frac{(x-1)^n}{nx^n} \quad (3.10)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.11)$$

3.5.3. Ejemplos

```
cout << ln_t(1) << endl;      // output = 0
cout << ln_t(2.7183) << endl;  // output = 1.00001
cout << ln_t(7.3890) << endl;  // output = 1.99999
```

3.6. Logaritmo: $\log_a(x)$

3.6.1. Descripción

Esta función calcula el logaritmo en base a de un número x , los cuales deben ser números reales mayores a cero, mediante el método de logaritmo natural implementado. La función se llama de la siguiente manera:

```
log_t(x, a)
```

3.6.2. Formulación Matemática

$$\log_a(x) = \frac{\ln(x)}{\ln(a)} \quad (3.12)$$

3.6.3. Ejemplos

```
cout << log_t(9, 3) << endl;    // output = 2
cout << log_t(16, 4) << endl;    // output = 2
cout << log_t(10, 2) << endl;    // output = 3.32193
```

3.7. Exponencial natural: e^x

3.7.1. Descripción

Esta función calcula el exponencial natural de un número x , el cual debe ser un número real, mediante una sumatoria. La función se llama de la siguiente manera:

```
exp_t(x)
```

3.7.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k \frac{x^n}{n!} \quad (3.13)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.14)$$

3.7.3. Ejemplos

```
cout << exp_t(-1) << endl;    // output = 0.367879
cout << exp_t(0) << endl;     // output = 1
cout << exp_t(1) << endl;     // output = 2.71828
cout << exp_t(2) << endl;     // output = 7.38906
```

3.8. Exponencial: a^x

3.8.1. Descripción

Esta función calcula el exponencial base a de un número x , los cuales deben ser números reales positivos, mediante otros métodos definidos. La función se llama de la siguiente manera:

```
power_t(x,a)
```

3.8.2. Formulación Matemática

$$a^x = e^{x \cdot \ln(a)} \quad (3.15)$$

3.8.3. Ejemplos

```
cout << power_t(8, 2) << endl;    // output = 256
cout << power_t(2, 10) << endl;   // output = 100
cout << power_t(4, 3) << endl;    // output = 81
cout << power_t(4, 1.5) << endl;  // output = 5.0625
```

3.9. Seno hiperbólico: $\sinh(x)$

3.9.1. Descripción

Esta función calcula el seno hiperbólico de un número x , el cual debe ser un número real, mediante una sumatoria. La función se llama de la siguiente manera:

```
sinh_t(x)
```

3.9.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k \frac{x^{2n+1}}{(2n+1)!} \quad (3.16)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.17)$$

3.9.3. Ejemplos

```
cout << sinh_t(-10) << endl;    // output = -11013.2
cout << sinh_t(0) << endl;      // output = 0
cout << sinh_t(10) << endl;     // output = 11013.2
```

3.10. Coseno hiperbólico: $\cosh(x)$

3.10.1. Descripción

Esta función calcula el coseno hiperbólico de un número x , el cual debe ser un número real, mediante una sumatoria. La función se llama de la siguiente manera:

```
cosh_t(x)
```

3.10.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k \frac{x^{2n}}{(2n)!} \quad (3.18)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.19)$$

3.10.3. Ejemplos

```
cout << cosh_t(-10) << endl;    // output = 11013.2
cout << cosh_t(0) << endl;      // output = 0
cout << cosh_t(10) << endl;     // output = 11013.2
```

3.11. Tangente hiperbólica: $\tanh(x)$

3.11.1. Descripción

Esta función calcula la tangente de un número x , el cual debe ser un número real, mediante otros métodos ya definidos. La función se llama de la siguiente manera:

```
tanh_t(x)
```

3.11.2. Formulación Matemática

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} \quad (3.20)$$

3.11.3. Ejemplos

```
cout << tanh_t(-10) << endl;    // output = -1
cout << tanh_t(0) << endl;      // output = 0
cout << tanh_t(10) << endl;     // output = 1
```

3.12. Raíz: $\sqrt[a]{x}$

3.12.1. Descripción

Esta función calcula la raíz a-ésima de un número x , el cual debe ser un número real, mediante la aplicación del método de Newton-Raphson a la función $g(x) = x^p - a$, evaluando posteriormente a como x y p como a . La función implementada se llama de la siguiente manera:

```
root_t(x, a)
```

3.12.2. Formulación Matemática

Método Iterativo

$$\begin{cases} y_0 = x/2 \\ y_{n+1} = y_n - \frac{y_n^a - x}{a \cdot y_n^{(a-1)}} \end{cases} \quad (3.21)$$

Criterio de Parada

$$\left| \frac{y_n - y_{n-1}}{y_n} \right| < 10^{-8} \quad (3.22)$$

3.12.3. Ejemplos

```
cout << root_t(100,2) << endl;    // output = 10
cout << root_t(512, 3) << endl;    // output = 8
cout << root_t(100, 3) << endl;    // output = 4.64159
```

3.13. Raíz cuadrada: \sqrt{x}

3.13.1. Descripción

Esta función calcula la raíz cuadrada de un número x , el cual debe ser un número real mayor a cero, mediante la aplicación del método de Newton-Raphson a la función $g(x) = x^p - a$, evaluando posteriormente a como x y p como 2. La función implementada se llama de la siguiente manera:

```
sqrt_t(x)
```

3.13.2. Formulación Matemática

Método Iterativo

$$\begin{cases} y_0 = x/2 \\ y_{n+1} = y_n - \frac{y_n^2 - x}{2 \cdot y_n} \end{cases} \quad (3.23)$$

Criterio de Parada

$$\left| \frac{y_n - y_{n-1}}{y_n} \right| < 10^{-8} \quad (3.24)$$

3.13.3. Ejemplos

```
cout << sqrt_t(512) << endl; // output = 22.6274
cout << sqrt_t(81) << endl;  // output = 9
cout << sqrt_t(144) << endl; // output = 12
cout << sqrt_t(2) << endl;   // output = 1.41421
```

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

3.14.1. Descripción

Esta función calcula el arcoseno de un número x , el cual debe ser un número real entre -1 y 1, mediante una sumatoria. La función se llama de la siguiente manera:

```
asin_t(x)
```

3.14.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1} \quad (3.25)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.26)$$

3.14.3. Ejemplos

```
cout << asin_t(1) << endl; // output = 1.50957
cout << asin_t(-1) << endl; // output = -1.50957
cout << asin_t(0) << endl; // output = 0
```

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

3.15.1. Descripción

Esta función calcula el arcotangente de un número x , el cual debe ser un número real, mediante una sumatoria. La función se llama de la siguiente manera:

```
atan_t(x)
```

3.15.2. Formulación Matemática

Método Iterativo

$$S_k(x) = \sum_{n=0}^k (-1)^n \frac{x^{2n+1}}{2n+1} \quad (3.27)$$

Criterio de Parada

$$|S_{k+1}(x) - S_k(x)| < 10^{-8} \quad (3.28)$$

3.15.3. Ejemplos

```
cout << atan_t(1) << endl;    // output = 0.785298
cout << atan_t(-1) << endl;   // output = -0.785298
cout << atan_t(0) << endl;    // output = 0
```