Informe laboratorio N°2 - Implementación de funciones trigonométricas y raiz cuadrada en MIPS

Benjamín Zúñiga Jofré
Departamento de Ingeniería Informática
Universidad de Santiago de Chile, Santiago, Chile
benjamin.zuniga.j@usach.cl

***Resumen*—En este informe se verá cómo es posible implementar funciones y/o operaciones matemáticas complejas como funciones trigonométricas, mediante series de Taylor , y la raíz cuadrada, en un contexto donde se desea calcular la distancia euclidiana entre dos puntos, mediante el método numérico Newton-Raphson, en el lenguaje ensamblador MIPS por medio de otras operaciones más básicas como sumas, restas, multiplicación y división.**

***Palabras claves*—Serie de Taylor, Método numérico, Método de Newton-Raphson y recursión.**

I. Introducción

Tanto las series de Taylor como los métodos numéricos son recursos muy útiles a la hora de querer aproximar valores de operaciones y/o funciones complejas, como el de una función trigonométrica o el de una raíz cuadrada, es por esto que este laboratorio N°2, de la asignatura Arquitectura de Computadores, se desafió a implementar en el lenguaje ensamblador MIPS, la función seno, coseno y raíz cuadrada, las dos primeras mediante series de Taylor parciales, puesto que estas son infinitas y la última mediante el método numérico Newton-Raphson, cada implementación con requisitos y restricciones particulares.

II. Antecedentes

II-A. Palabras claves

1 - Serie de Taylor: "Serie de potencias de infinitos términos que sirve para aproximar el valor de una función *f* que sea continua y derivable varias veces o infinitas."[1]

2 - Método Numérico: "Sucesión más o menos larga de operaciones numéricas (normalmente mediante la ayuda de un ordenador), al cabo de las cuales encontramos un valor numérico que, si bien no es la solución exacta del problema, se le parece mucho, es decir, aproxima la solución buscada con una precisión razonablemente buena."[2]

3 - Método de Newton-Raphson: "Método numérico que permite aproximar la raíz de un función f continua y diferenciable, es decir el valor donde la función se hace 0, mediante un proceso recursivo, que implica el valor de un punto inicial, la función evaluada en el punto y la derivada de la función evaluada en el punto, hasta alcanzar una precisión considerablemente aceptable."[3][4]

4 - Recursión: "Proceso de definir (o solucionar) un problema en terminos de si mismo."[5]

II-B. Formulas

Las fórmulas utilizadas para resolver lo planteado, fueron las siguientes:
1 - Serie de Taylor para aproximar la función seno:

$$sen(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \tag{1}$$

2 - Serie de Taylor para aproximar la función coseno:

$$cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} \tag{2}$$

3- Distancia euclidiana entre dos puntos:

Sea A = (A_x, A_y, A_z) y B = (B_x, B_y, B_z), dos puntos en un espacio R^3, la disnacia ecludiana entre ellos es:

$$d = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2} \tag{3}$$

4- Metodo de Newton-Raphson general

$$x_{n+1} = x_n + \frac{f(x_n)}{f'(x_n)} \tag{4}$$

5 - Método de Newton-Raphson para aproximar la raíz cuadrada:
Sea $f(x) = x^2 - k$, *f(x)* será una función que tiene como raiz la raiz cuadrada de k, implica que *f'(x)* sera $f'(x) = 2x$, al reemplazar en la fórmula (4) se obtiene:

$$\sqrt{k} \approx \frac{x_n}{2} + \frac{k}{2x_n} \tag{5}$$

III. Materiales y Métodos

III-A. Materiales

Para desarrollar la actividad fue utilizado el lenguaje ensamblador MIPS, por medio del IDE MARS en su versión 4.5. Por otra parte utilizado fue un ordenador portátil que cuenta con los siguientes componentes: procesador Intel Core i5-10300H, una tarjeta gráfica Nvidia Geforce GTX 1650, memoría RAM de 8GB a 2933 MHz, una unidad de estado sólido Intel de 512 GB y cuenta con el sistema operativo Windows 11 de 64 Bits.

III-B. Métodos

Para entender mejor los métodos utilizados estos serán separados y explicados uno a uno en el orden que fueron realizados.

III-B1. Menú para escoger la función trigonométrica: Lo primero realizado fue el procedimiento de crear una especie de menú en el cual primero se le pide al usuario del programa que ingrese el angulo que quiere operar, valor el cual es almacenado en el registro t0, seguido de la función trigonométrica deseada (seno o coseno) mediante el ingreso de un número, según se muestra en el menú y finalmente dependiendo de cual es este último valor ingresado se salta al cálculo de la función deseada evaluada en el ángulo deseado.

Cómo parte de los requisitos de estas dos primeras implementaciones, la función seno y coseno, era el no uso de cualquier instrucción de multiplicación y división, a continuación se detalla cómo se implementan estas dos operaciones, las cuales van a ser necesarias para la implementación de la función seno y coseno como tal.

III-B2. Multiplicación: La manera en que se implementó la multiplicación, frente a la prohibición de usar la operación mul, fue realizar un ciclo de sumas , tantas veces sea el valor requerido y la manera en la que se implementó fue copiar el valor requerido en el registro t1 y t2, y luego actualizar el contenido de a0 con el mismo contenido de a0 más el contenido de t1, luego restar 1 al contenido de t2, y así vuelve a empezar el ciclo, que se repite hasta que en t2 haya un 0, todo esto porque como se indica en la fórmula (1) y (2) las multiplicaciones serán usadas sólo para calcular potencias de un número, por otra parte con respecto al signo del ángulo ingresado, este no se tomó en consideración, ya que existe otra instancia que verifica el valor del ángulo ingresado y hace las operaciones correspondientes, aparte de dejar positivo el ángulo en caso de que originalmente fuese negativo.

III-B3. División: Para lograr implementar la división, frente a la prohibición del uso de div, se diseñó un algoritmo el cual consiste en un ciclo de restas y funciona para calcular tanto la parte entera de la división como los cinco primeros decimales, está está implementada en 2 partes, una de control que sirve para controlar cuántas veces se divide y con qué términos, y otra que hace la división como tal y funciona restando al registro s5, que contiene la potencia calculada, el contenido de s6 que contiene el factorial correspondiente a cada término, el resultado se almacena en s4 y si es menor que 0 se vuelve al control, debido a que el factorial no .ªlcanza.en la potencia, de lo contrario se suma un 1.0 o el decimal correspondiente en el registro f0, se actualiza el valor de s5 realizando la misma resta del contenido de s5 menos el de s6 y si el resultado es 0 se salta directamente al final de la división, sino se vuelve al inicio de la operación. Por otra parte, en el proceso de control se cargan los decimales correspondientes para ir calculando el resultado final y luego entre una división y otra, se multiplica por 10 el valor del resto que quedó de la división anterior y finalmente cuando se calcularon todos los decimales se vuelve a la función trigonométrica correspondiente mediante la instrucción jr.

III-B4. Función seno: Para calcular el seno del ángulo ingresado, primero se verifica si el ángulo es positivo o negativo, ya que para la multiplicación y división se trabaja solo con números enteros positivos, en caso de ser positivo no se hace nada, de lo contrario se hace positivo el ángulo, restando 0 menos el ángulo y se almacena un 1 en el registro a3, que luego al final va a determinar el signo del resultado, pues aprovechándose de que la función seno es una función impar, es decir, $sen(-x) = -sen(x)$, por lo tanto solo basta con trabajar con el ángulo positivo y cambiar el signo del resultado al final, lo cual se hace dependiendo el valor de en el registro a3. Luego de lo anterior se almacena el valor del angulo como punto flotante de doble precisión en el registro f4, sumando un 1.0 una cantidad de veces igual al valor del ángulo, y restando 1 a una copia del ángulo hasta que esta sea 0, luego se calcula el cuadrado del ángulo, pues siempre va a ser necesario multiplicar por este, debido a que la potencia de cada termino es la del termino anterior multiplicada por x^2, después de calcular el cuadrado del angulo, este se almacena para no perderse y se multiplica el valor del angulo por su cuadrado en caso del segundo termino o el valor de la potencia del angulo del segundo termino por el cuadrado del angulo para el tercer termino y asi sucesivamente, inmediatamente despues de obtener la potencia del angulo correspondiente al termino que se quiere calcular, se carga el factorial correspondiente al termino en el registro s6 para asi saltar a la division a calcular el termino como tal, luego de eso el resultado que queda en el registro f0 se mueve al registro correspondiente para no perderlo, en caso del segundo a f6, del tercero a f8 y del cuarto a f10, y así ese proceso de multiplicar, cargar el factorial, dividir y mover el resultado se hace 3 veces, ya que se usaron 3 terminos y por ultimo se calcula el seno, restando el valor de f4 (primer termino) menos f6 (segundo termino), luego al resultado de la operacion anterior sumarle el contenido de f8 (tercer termino), despues al resultado restarle el contenido de f10 (cuarto termino), tal y como indica la ecuación (1) y finalmente guardar el resultado de todas las operaciones anteriores en el registro f12 para ser impreso y ser entregado al usuario.

III-B5. Función coseno: La función coseno en la práctica fue implementada en su mayoría igual que la función seno, por lo menos estructuralmente hablando y en términos de los procesos y métodos utilizados, solo que tiene algunas variaciones para respetar la fórmula (2), variaciones tales como que al comprobar el signo del ángulo ingresado este en caso de ser negativo simplemente se hace positivo y se trabaja así, esto aprovechándose de que la función coseno es par, es decir, $cos(-x) = cos(x)$, también primero se calcula desde el segundo término hasta el quintos, realizando el mismo proceso que se usó para la función seno de multiplicación, cargar el factorial, dividir y mover el resultado, ya que en términos aritméticos la forma de calcular estos términos es la misma que los términos del seno, y finalmente se realizan las operaciones de suma y resta según indica la fórmula (2), para el primer término se carga un 1.0 directamente en f2 justo antes de realizar el cálculo del resultado final y por último el resultado se guarda en el registro f12 para ser impreso y ser entregado al usuario.

III-B6. Obtener los puntos para calcular la distancia euclidiana: Para esta segunda parte del laboratorio en la que se debe calcular la distancia euclidiana entre dos puntos, lo primero implementado fue un proceso en el cual se obtienen los puntos, este consta de varias repeticiones de un par de instrucciones que varían en contenido, pero esencialmente son lo mismo, que es, imprimir un mensaje indicando cual es la coordenada del y el punto que se debe ingresar, luego el usuario debe ingresar el valor de la coordenada y este va a ser movido a un registro distinto para ser usado y calcular el radicando de la raíz cuadrada de la fórmula (3).

III-B7. Cacular el radicando: En este proceso simplemente se realizan las operaciones que se muestran en la fórmula (3) primero calculando el $(A_x - B_x)^2$, luego el $(A_y - B_y)^2$, despues el $(A_z - B_z)^2$, con las operaciones sub.d y mul.d, todo con los valores ingresados por el usuario, luego se suman los 3 resultados anteriores y se guarda el resultado en el registro f2 y finalmente, en el caso de esta implementación se carga un 20 en el registro s0, que indica la cantidad de veces que se realizara el método de Newton-Raphson, para calcular el valor de la raíz cuadrada del radicando y un 1 en f0 que respresenta el punto inicial desde el cual se parte la aproximación.

III-B8. Newton-Raphson: Para realizar el método numérico de Newton-Raphson se utilizará la fórmula (5), ya que sale de un despeje matemático explicado en 5 de II-B. Este proceso consiste en primero revisar si el número de iteraciones restantes es 0, sino guarda el primer valor de la pila de MIPS en f0, realizar la división de k entre x, donde k es el radicando calculado, que se encuentra en f2 y x el punto inicial en la primera iteración y luego el resultado del método, que se encuentra en f0, luego sumarle al resultado de la división x, dividir el resultado entre 2, cargar el resultado de la iteración anterior que se encuentra en el primer valor de la pila de MIPS, comparar si es igual al de la iteración reciente y si son iguales saltar al final del procedimiento, sino restarle 1 al número de iteraciones restantes y finalmente volver a empezar.

IV. Resultados

Los resultados obtenidos luego de poner a prueba los procedimientos anteriormente mencionados en III-B, fueron satisfactorios y buenos en su mayoría, arrojando los resultados que debería, calculados con calculadora, por lo menos para los casos probados, notar que se mostrarán casos en los que se pide calcular seno y coseno de ángulos menores a 3, para que sean más reales", puesto que al usarse una serie que en su concepción es infinita con pocos términos finitos, a medida nos alejamos del 0 los valores si bien son los que corresponden por calculadora, son peores aproximaciones debido a la cantidad limitada de términos, incluso escapando de las cotas -1 y 1 de ambas funciones. A continuación se adjuntan imágenes de algunos procedimientos:

```
Ingresar el ángulo: 0
1)Seno
2)Coseno
Escoger operación: 1

Elresultado es: 0.0
-- program is finished running --

Ingresar el ángulo: 0
1)Seno
2)Coseno
Escoger operación: 2

Elresultado es: 1.0
-- program is finished running --
```

Figura 1: Imagen 1.

En la imagen anterior se pide al programa calcular el seno y coseno de 0, y este entrega por resultados 0 y 1 respectivamente, lo cual está bien y son casos básicos.

```
Ingresar el ángulo: 3
1)Seno
2)Coseno
Escoger operación: 1

Elresultado es: 0.09108000000000083
-- program is finished running --


Reset: reset completed.

Ingresar el ángulo: 3
1)Seno
2)Coseno
Escoger operación: 2

Elresultado es: -0.9747799999999998
-- program is finished running --
```

Figura 2: Imagen 2.

En la imagen anterior el procedimiento realizado fue calcular el seno y coseno de 3, el número entero más cercano a pi, y el programa entrega como resultados 0.09108000000000083 en el caso del seno y -0.9747799999999998 en el caso del coseno, valores muy cercanos a 0 y -1, que serían los resultados de seno y coseno de pi respectivamente.

```
Ingresar el ángulo: -2
1)Seno
2)Coseno
Escoger operación: 1

Elresultado es: -0.90794
-- program is finished running --


Reset: reset completed.

Ingresar el ángulo: 2
1)Seno
2)Coseno
Escoger operación: 1

Elresultado es: 0.90794
-- program is finished running --
```

Figura 3: Imagen 3.

En la imagen anterior el procedimiento realizado fue calcular el seno de -2 y 2, y se puede observar como el programa entrega el mismo valor con el signo contrario como corresponde.

```
Ingresar el ángulo: 1
1)Seno
2)Coseno
Escoger operación: 2

Elresultado es: 0.5403
-- program is finished running --


Reset: reset completed.

Ingresar el ángulo: -1
1)Seno
2)Coseno
Escoger operación: 2

Elresultado es: 0.5403
-- program is finished running --
```

Figura 4: Imagen 4.

En la imagen anterior el procedimiento realizado fue calcular el coseno de 1 y -1, y se puede observar como el programa entrega el mismo valor con el mismo signo como corresponde.

```
Coordenada X del primer vector: 10

Coordenada Y del primer vector: 8

Coordenada Z del primer vector: 2

Coordenada X del segundo vector: 2

Coordenada Y del segundo vector: 2

Coordenada Z del segundo vector: 2

La distancia entre los dos vectores es: 10.0
-- program is finished running --
```

Figura 5: Imagen 5.

En la imagen anterior el procedimiento realizado fue calcular la distancia entre el punto (10,8,2) y el punto (2,2,2), resultando básicamente en calcular la raíz cuadrada de 100, luego de calcular el radicando, y se puede observar como el programa entrega por resultado 10, resultado que corresponde a la operación pedida.

```
Coordenada X del primer vector: 9

Coordenada Y del primer vector: 7

Coordenada Z del primer vector: 5

Coordenada X del segundo vector: 4

Coordenada Y del segundo vector: 2

Coordenada Z del segundo vector: 5

La distancia entre los dos vectores es: 7.0710678118654755
-- program is finished running --
```

Figura 6: Imagen 6.

En la imagen anterior el procedimiento realizado fue calcular la distancia entre el punto (9,7,5) y el punto (4,2,5), resultando básicamente en calcular la raíz cuadrada de 50, luego de calcular el radicando, y se puede observar como el programa entrega por resultado 7.0710678118651755, el cual es el resultado efectivamente de la raíz cuadrada de 50.

V. CONCLUSIONES

Finalmente se puede concluir que a pesar que programar en un lenguaje de bajo nivel como lo es el lenguaje ensamblador MIPS es una tarea difícil en la mayoría de casos, más si hay restricciones que obligan a utilizar los elementos más primarios y básicos del lenguaje, esto no fue un impedimento para poder realizar la actividad propuesta como laboratorio N°2, a pesar de en un inicio parecer muy difícil, debido a que los conceptos que había que implementar ya son de una dificultad considerable en cualquier otro lenguaje de programación, incluso de alto nivel. Por otra parte, la mayoría de los procedimientos que se requirieron para completar los objetivos principales propuestos en este laboratorio están implementados, como la multiplicación división, el uso de números flotantes de doble precisión y el uso de la la pila de MIPS, y funcionan tal y como se planeaba desde un inicio, cumpliendo su objetivo y realizando lo que le corresponde. Concluyendo así que el laboratorio fue realizado de buena manera, debido a que desafíos y problemas propuestos fueron superados y ayudaron a poner en práctica los conocimientos adquiridos en las clases de teoría de la asignatura Arquitectura de Computadores.

REFERENCIAS

[1] A. T. Martínez. (2015) Series de taylor y series de fourier: Un estudio comparativo. [Online]. Available: https://www.ugr.es/~acanada/docencia/matematicas/definitivoAlejandraTorresMartinezTFG.pdf

[2] U. de Sevilla. (2019) Métodos numéricos. [Online]. Available: http://departamento.us.es/edan/php/asig/GRABIO/GBM/Tema4.pdf

[3] U. of Texas at Austin. (s.f) The idea of newton's method. [Online]. Available: https://web.ma.utexas.edu/users/m408n/CurrentWeb/LM4-8-2.php

[4] G. Strang and E. Herman. (s.f) Newton's method. [Online]. Available: https://math.libretexts.org/Bookshelves/Calculus/Calculus_(OpenStax)/04%3A_Applications_of_Derivatives/4.09%3A_Newtons_Method

[5] U. of Utah. (s.f) Recursion. [Online]. Available: https://users.cs.utah.edu/~germain/PPS/Topics/recursion.html