



Informe Laboratorio: Análisis Numérico

Práctica No. 0

Estudiante: Hendrik López

Código: 2170129

Grupo: B2

Escuela de ingeniería de sistemas e informática

Universidad Industrial de Santander

20 de abril de 2021

1 Introducción

El presente laboratorio ha sido realizado con el propósito de entender y aplicar la aproximación aritmética y el formato de coma flotante, por medio de herramientas de cómputo, como Matlab; para ello, se ha organizado el desarrollo de la actividad en tres fases principales: una serie de preguntas conceptuales y teóricas, que cumplen el papel de introducir al estudiante en el tema en cuestión; problemas manuales que ponen a prueba la capacidad de análisis, aplicando lo explicado previamente y ejercicios aplicados haciendo uso de Matlab.

2 Desarrollo

Con el fin de realizar la actividad, es menester dar respuesta a las siguientes preguntas:

- a. ¿Qué es el error absoluto y el error relativo?

El error absoluto es la diferencia numérica que se tienen en las medidas obtenidas respecto al valor tomado como valor exacto. En síntesis, este puede ser calculado por

medio de la siguiente ecuación:

$$E_p = |p - \hat{p}|$$

donde E_p es el error absoluto y $|p - \hat{p}|$ la diferencia entre las medidas teóricas y las medidas experimentales; de esta manera, se deduce que $E_p \geq 0$.

Por ende, el error relativo se interpreta como el cociente entre el error absoluto y el valor teórico p , usando la siguiente ecuación:

$$R_p = \frac{E_p}{p}$$

donde es el error relativo. Debido a esto, se entiende que $p \neq 0$

b. ¿Cómo calcular los dígitos significativos de un número?

De manera conceptual, los dígitos de un número poseen un grado de importancia al momento de realizar cálculos; debido a esto, existen una serie de consideraciones a la hora de definir dicha relevancia:

- Todos los dígitos diferentes de cero son importantes
- Todos los ceros entre dígitos importantes, son importantes.
- El número significativo de cifras está determinado por el número diferente de cero más cerca de la izquierda.
- El número ubicado más hacia la derecha, es el menos significativo
- Si no hay punto decimal, el número diferente de cero más hacia la derecha, es el menos significativo.

Lo cual, puede interpretarse con la siguiente ecuación:

$$R_p < \frac{10^{1-d}}{2}$$

donde d es considerado el entero positivo más grande dentro de d cifras significativas.

c. ¿Qué propiedades posee el orden de aproximación?

El orden de aproximación se refiere a expresiones formales o informales de cuán precisa es una aproximación. Sus propiedades son:

- $O(h^p) + O(h^p) = O(h^p)$ (Sumar dos órdenes de aproximación de la misma complejidad polinómica, da como resultado el mismo).
- $O(h^p) + O(h^q) = O(h^r) \forall r = \min(p, q)$ (Sumar dos órdenes de aproximación de distinta

complejidad polinómica, da como resultado el orden de aproximación menor).

- $O(h^p)O(h^q)=O(h^{p+q})$ (Multiplicar dos órdenes de aproximación da como resultado un nuevo orden de aproximación, donde su exponente es la suma de los exponentes anteriores).

Acto seguido, se resuelven los siguientes ejercicios, aplicando los conceptos explicados por medio del cuestionario anterior:

f. Use aproximación aritmética de tres dígitos para calcular las siguientes sumatorias:

- $\sum_{k=1}^6 \frac{1}{3^k}$

El procedimiento es:

$$\sum_{k=1}^6 \frac{1}{3^k} = \frac{1}{3^1} + \frac{1}{3^2} + \frac{1}{3^3} + \frac{1}{3^4} + \frac{1}{3^5} + \frac{1}{3^6}$$

$$\sum_{k=1}^6 \frac{1}{3^{7-k}} \approx 0.333 + 0.111 + 0.037 + 0.012 + 0.004 + 0.001$$

$$\sum_{k=1}^6 \frac{1}{3^k} \approx 0.498$$

- $\sum_{k=1}^6 \frac{1}{3^{7-k}}$

El procedimiento es:

$$\sum_{k=1}^6 \frac{1}{3^{7-k}} = \frac{1}{3^{7-1}} + \frac{1}{3^{7-2}} + \frac{1}{3^{7-3}} + \frac{1}{3^{7-4}} + \frac{1}{3^{7-5}} + \frac{1}{3^{7-6}}$$

$$\sum_{k=1}^6 \frac{1}{3^{7-k}} \approx 0.001 + 0.004 + 0.012 + 0.037 + 0.111 + 0.333$$

$$\sum_{k=1}^6 \frac{1}{3^{7-k}} \approx 0.498$$

b. Mejorar la ecuación cuadrática; se debe asumir que $a \neq 0$ y $b^2 - 4ac > 0$, teniendo en cuenta la ecuación $ax^2 + bx + c = 0$. Las raíces pueden ser calculadas usando:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Demuestre que las raíces pueden ser calculadas por medio de las siguientes ecuaciones:

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}} \quad x_2 = \frac{-2c}{b - \sqrt{b^2 - 4ac}}$$

En este caso, la manera de llegar a estas expresiones, es por medio de la racionalización.

Para el caso de x_1 , el proceso es:

$$x_1 = \left(\frac{-b + \sqrt{b^2 - 4ac}}{2a} \right) \left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} \right)$$

$$x_1 = \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-2a(b + \sqrt{b^2 - 4ac})}$$

$$x_1 = \frac{b^2 - b^2 + 4ac}{-2a(b + \sqrt{b^2 - 4ac})}$$

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

El proceso es el mismo en el caso de x_2 :

$$x_2 = \left(\frac{-b - \sqrt{b^2 - 4ac}}{2a} \right) \left(\frac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}} \right)$$

$$x_2 = \frac{(-b)^2 - (\sqrt{b^2 - 4ac})^2}{-2a(b - \sqrt{b^2 - 4ac})}$$

$$x_1 = \frac{b^2 - b^2 + 4ac}{-2a(b - \sqrt{b^2 - 4ac})}$$

$$x_1 = \frac{-2c}{b - \sqrt{b^2 - 4ac}}$$

3 Anexos

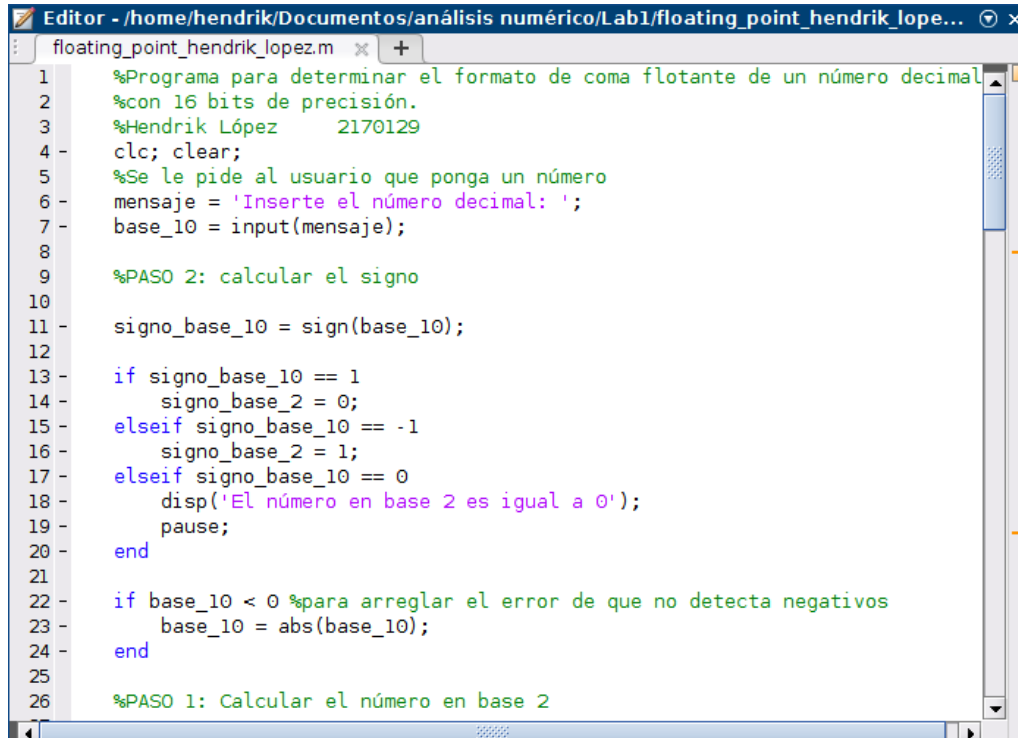
En la presente sección, se presentan los ejercicios aplicados por medio de Matlab, además de capturas del código explicado por medio de comentarios:

- a. Crear una función en Matlab llamada *floating_point_function_hendrik_lopez()* que determine el formato de coma flotante para números almacenados en un computador, con 16 bits de precisión, como se muestra en la siguiente figura:

Sign	Exponent						Mantissa							

Figura 1. Representación del formato de coma flotante

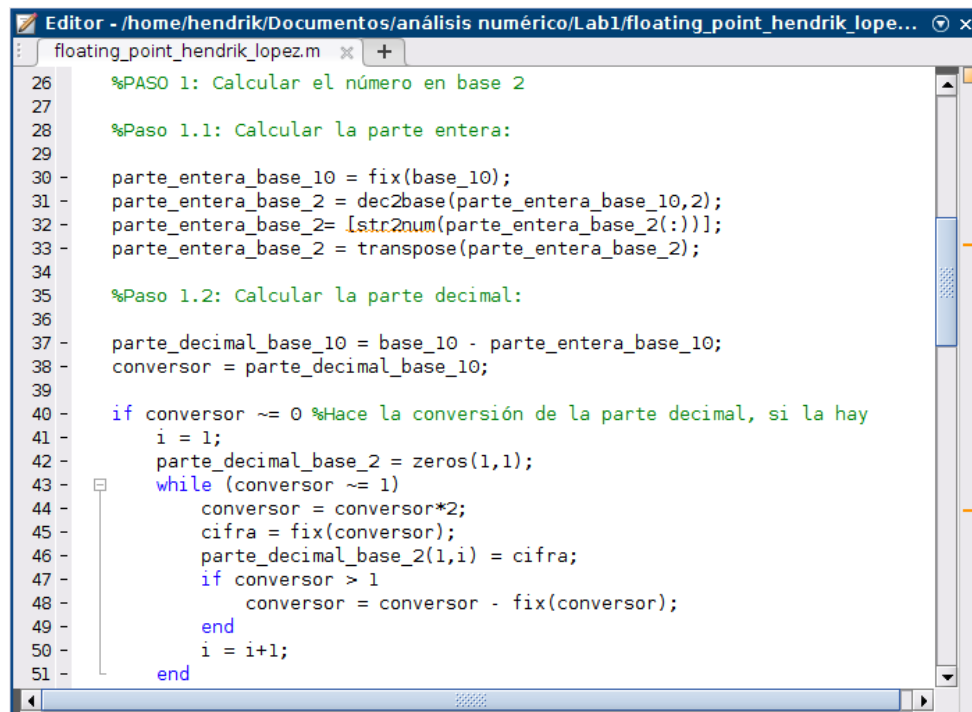
El código que resuelve este problema es el siguiente; primero, se le solicita al usuario que ingrese un número en base 10 con números decimales. Acto seguido, calcula el signo del formato de coma flotante, o define si el número ingresado es 0. Una vez hallado el signo, se calcula el valor absoluto de dicho número, con el fin de evitar errores al manejar números negativos, tal como se muestra en la figura 2:



```
Editor - /home/hendrik/Documents/análisis numérico/Lab1/floating_point_hendrik_lopez...
floating_point_hendrik_lopez.m
1 %Programa para determinar el formato de coma flotante de un número decimal
2 %con 16 bits de precisión.
3 %Hendrik López 2170129
4 clc; clear;
5 %Se le pide al usuario que ponga un número
6 mensaje = 'Inserte el número decimal: ';
7 base_10 = input(mensaje);
8
9 %PASO 2: calcular el signo
10
11 signo_base_10 = sign(base_10);
12
13 if signo_base_10 == 1
14     signo_base_2 = 0;
15 elseif signo_base_10 == -1
16     signo_base_2 = 1;
17 elseif signo_base_10 == 0
18     disp('El número en base 2 es igual a 0');
19     pause;
20 end
21
22 if base_10 < 0 %para arreglar el error de que no detecta negativos
23     base_10 = abs(base_10);
24 end
25
26 %PASO 1: Calcular el número en base 2
```

Figura 2. Entrada del número base 10 y cálculo del signo.

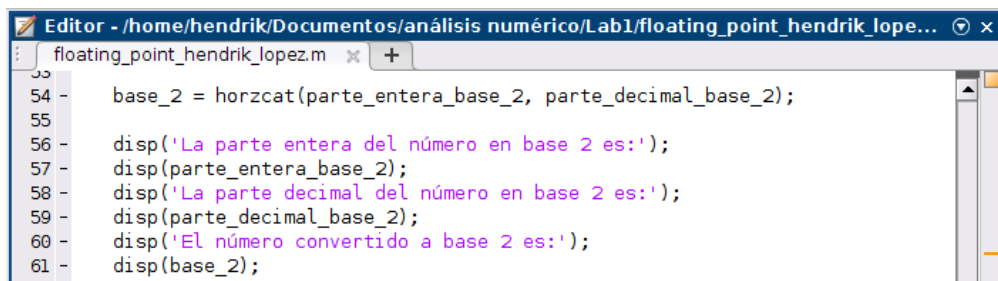
Una vez teniendo esta información, el programa procede a hacer la conversión del número en base 10 a base 2, como se puede observar en la figura 3:



```
26 %PASO 1: Calcular el número en base 2
27
28 %Paso 1.1: Calcular la parte entera:
29
30 parte_entera_base_10 = fix(base_10);
31 parte_entera_base_2 = dec2base(parte_entera_base_10,2);
32 parte_entera_base_2 = [str2num(parte_entera_base_2(:))];
33 parte_entera_base_2 = transpose(parte_entera_base_2);
34
35 %Paso 1.2: Calcular la parte decimal:
36
37 parte_decimal_base_10 = base_10 - parte_entera_base_10;
38 conversor = parte_decimal_base_10;
39
40 if conversor ~= 0 %Hace la conversión de la parte decimal, si la hay
41     i = 1;
42     parte_decimal_base_2 = zeros(1,1);
43     while (conversor ~= 1)
44         conversor = conversor*2;
45         cifra = fix(conversor);
46         parte_decimal_base_2(1,i) = cifra;
47         if conversor > 1
48             conversor = conversor - fix(conversor);
49         end
50         i = i+1;
51     end
```

Figura 3. Conversión del número en base 10 a base 2.

Una vez se tienen las partes enteras y decimales del número en base 2, estas se concatenan en un único vector, con el propósito de visualizar el número completo; este proceso se ve reflejado en la figura 4:



```
54 base_2 = horzcat(parte_entera_base_2, parte_decimal_base_2);
55
56 disp('La parte entera del número en base 2 es:');
57 disp(parte_entera_base_2);
58 disp('La parte decimal del número en base 2 es:');
59 disp(parte_decimal_base_2);
60 disp('El número convertido a base 2 es:');
61 disp(base_2);
```

Figura 4. Visualización del número en base 2

Con el número convertido en base 2, es posible realizar el formato de coma flotante; previamente se ha calculado el signo, ahora es menester calcular el exponente, como se puede observar en la figura 5:

```

Editor - /home/hendrik/Documentos/análisis numérico/Lab1/floating_point_hendrik_lope...
floating_point_hendrik_lopez.m
63 - disp('FORMATO DE PUNTO FLOTANTE:');
64
65 - disp('El signo es:');
66 - disp(signo_base_2);
67
68 - %PASO 3: Calcular el exponente
69
70 - %Paso 3.1: Calcular el bias
71
72 - bits_exponente = 7;
73 - bias = 2^(bits_exponente-1)-1;
74
75 - %Hacer la suma y convertir el resultado a base 2
76
77 - i = length(parte_entera_base_2);
78 - exponente = 0;
79
80 - while i > 1
81 -     i = i - 1;
82 -     exponente = exponente + 1;
83 - end
84
85 - exponente_base_10 = exponente + bias;
86 - exponente_base_2 = dec2base(exponente_base_10,2);
87 - exponente_base_2 = [str2num(exponente_base_2(:))];
88 - exponente_base_2 = transpose(exponente_base_2);

```

Figura 5. Cálculo del exponente

Finalmente, se calcula la mantisa, como se puede apreciar en la figura 6:

```

Editor - /home/hendrik/Documentos/análisis numérico/Lab1/floating_point_hendrik_lope...
floating_point_hendrik_lopez.m
89
90 - disp('El exponente es:');
91 - disp(exponente_base_2);
92
93 - %PASO 4: Calcular la mantisa en formato de coma flotante
94
95 - mantisa = zeros(1,8);
96
97 - %Paso 3.2: Hallar la mantisa
98
99 - i = 1;
100 - while i < 9
101 -     mantisa(1,i) = base_2(1,i+1);
102 -     i=i+1;
103 - end
104
105 - disp('La mantisa es:');
106 - disp(mantisa);
107

```

Figura 6. Cálculo de la mantisa

b. Usar los siguientes números decimales, para probar la función creada:

- 1612.078125_{10}

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 7:

```

Command Window
Inserte el número decimal: 1612.078125
La parte entera del número en base 2 es:
  1  1  0  0  1  0  0  1  1  0  0

La parte decimal del número en base 2 es:
  0  0  0  1  0  1

El número convertido a base 2 es:
Columns 1 through 13
  1  1  0  0  1  0  0  1  1  0  0  0  0

Columns 14 through 17
  0  1  0  1

FORMATO DE PUNTO FLOTANTE:
El signo es:
  0

El exponente es:
  1  0  0  1  0  0  1

La mantisa es:
  1  0  0  1  0  0  1  1

fx >> |

```

Figura 7. Formato de coma flotante de 1612.078125_{10}

- 6317.9136_{10}

El resultado que muestra el algoritmo desarrollado se puede observar en las figuras 8 y 9:

```

Command Window
Inserte el número decimal: 6317.9136
La parte entera del número en base 2 es:
  1  1  0  0  0  1  0  1  0  1  1  0  1

La parte decimal del número en base 2 es:
Columns 1 through 13
  1  1  1  0  1  0  0  1  1  1  1  0  0

Columns 14 through 26
  0  0  1  1  0  1  1  0  0  0  0  1  0

Columns 27 through 35
  0  0  1  0  0  1  1  0  1

El número convertido a base 2 es:
Columns 1 through 13
  1  1  0  0  0  1  0  1  0  1  1  0  1

Columns 14 through 26
  1  1  1  0  1  0  0  1  1  1  1  0  0

Columns 27 through 39

```

Figura 8. Formato de coma flotante de 6317.9136_{10} (parte 1).

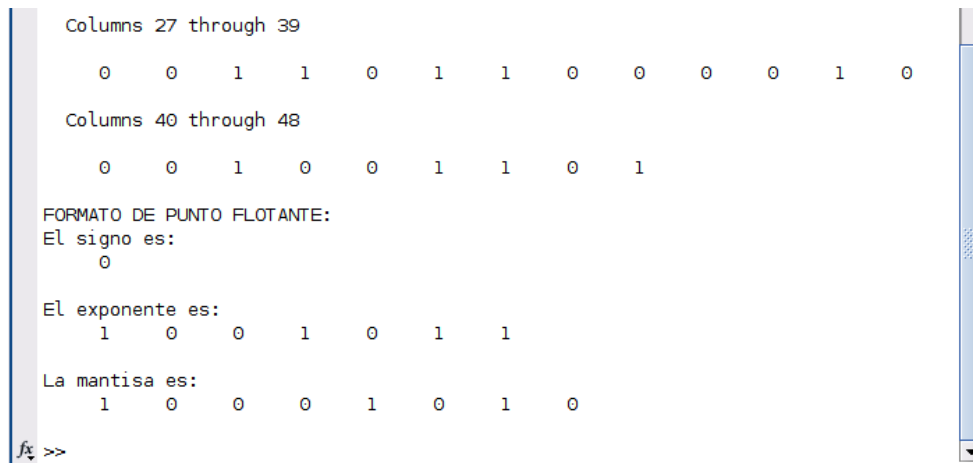


Figura 9. Formato de coma flotante de 6317.9136_{10} (parte 2).

- -962.0153_{10}

El resultado que muestra el algoritmo desarrollado se puede observar en las figuras 10 y 11:

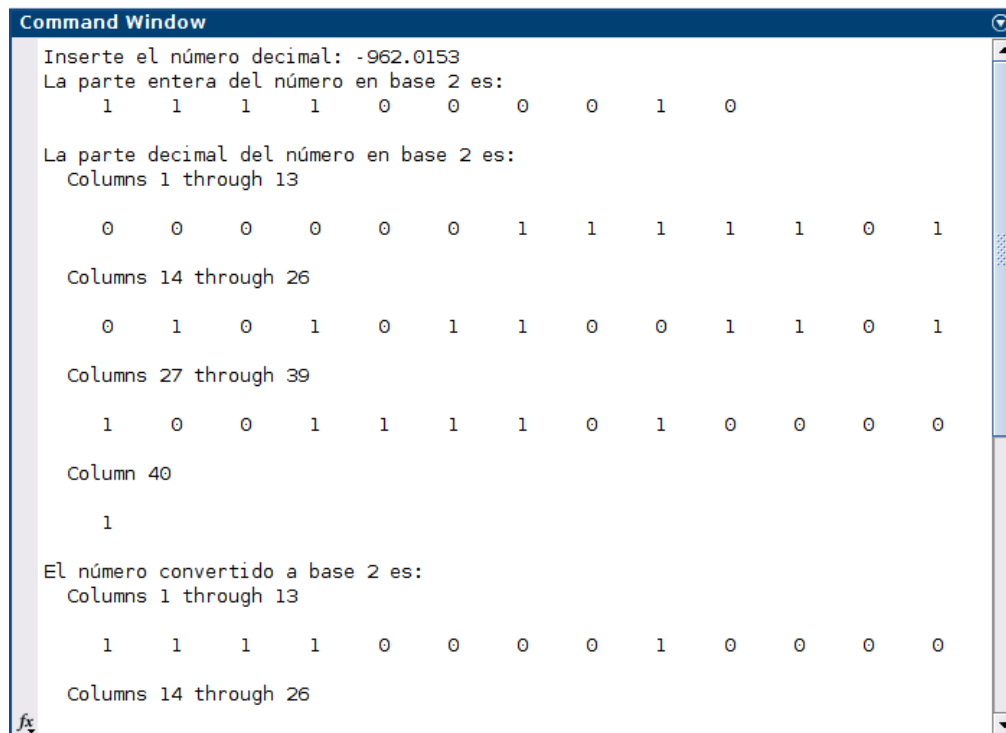


Figura 10. Formato de coma flotante de -962.0153_{10} (parte 1).

```

Columns 14 through 26
    0    0    0    1    1    1    1    1    0    1    0    1    0

Columns 27 through 39
    1    0    1    1    0    0    1    1    0    1    1    0    0

Columns 40 through 50
    1    1    1    1    0    1    0    0    0    0    1

FORMATO DE PUNTO FLOTANTE:
El signo es:
    1

El exponente es:
    1    0    0    1    0    0    0

La mantisa es:
    1    1    1    0    0    0    0    1
fx >> |

```

Figura 11. Formato de coma flotante de -962.0153_{10} (parte 2).

- c. Calcule el error absoluto y el error relativo entre los números decimales dados y los números almacenados en el computador con 16 bits de precisión.

Para calcular el error absoluto y error relativo, es necesario convertir el valor base 2 almacenado por el computador, a base 10, tal como se puede observar en la figura 12:

```

Editor - /home/hendrik/Documents/análisis numérico/Lab1/floating_point_hendrik_lope...
floating_point_hendrik_lopez.m  x  +
108 %PASO 5: Calcular error absoluto y relativo
109
110 %Paso 5.1: Hallar el valor real almacenado
111
112 parte_entera_base_2_conversion = num2str(parte_entera_base_2);
113 parte_decimal_base_2_conversion = num2str(parte_decimal_base_2);
114
115 parte_entera_base_10_conversion = bin2dec(parte_entera_base_2_conversion)
116 parte_decimal_base_10_conversion = bin2dec(parte_decimal_base_2_conversion)
117
118 while parte_decimal_base_10_conversion > 1
119     parte_decimal_base_10_conversion = parte_decimal_base_10_conversion/10
120 end
121
122 valor_real = parte_entera_base_10_conversion + parte_decimal_base_10_conv
123
124 disp('El valor real almacenado (en valor absoluto) es:');
125 disp(valor_real);
126 disp('El signo del valor real almacenado es:');
127 disp(signo_base_10);
128

```

Figura 12. Obtención del valor real.

Una vez se ha obtenido el valor real almacenado del número decimal en base 10, es posible calcular el error absoluto y error relativo; esto se puede ver en la figura 13:

```

129 %Paso 5.2: Calcular el error absoluto
130
131 error_absoluto = abs(base_10-valor_real);
132 disp('El error absoluto es:');
133 disp(error_absoluto);
134
135 %Paso 5.3: Calcular el error relativo
136
137 error_relativo = error_absoluto/abs(base_10);
138 disp('El error relativo es:');
139 disp(error_relativo);
140
141

```

Figura 13. Cálculo del error absoluto y error relativo.

Para probar la eficacia de este algoritmo, las figuras 14, 15 y 16 muestran los resultados, usando los números decimales convertidos en el inciso b; es menester aclarar que, en el caso de los números en base 10, el signo es representado con el número 1; para los negativos, el número correspondiente es -1.

```

El valor real almacenado (en valor absoluto) es:
1.6125e+03

El signo del valor real almacenado es:
1

El error absoluto es:
0.4219

El error relativo es:
2.6170e-04

fx >>

```

Figura 14. Valor real almacenado de 1612.078125_{10}

```

El valor real almacenado (en valor absoluto) es:
6.3173e+03

El signo del valor real almacenado es:
1

El error absoluto es:
0.5997

El error relativo es:
9.4919e-05

fx >>

```

Figura 15. Valor real almacenado de 6317.9136_{10} .

```

El valor real almacenado (en valor absoluto) es:
962.1682

El signo del valor real almacenado es:
-1

El error absoluto es:
0.1529

El error relativo es:
1.5896e-04

fx >>

```

Figura 16. Valor real almacenado de -962.0153_{10} .

- d. Use los resultados obtenidos en el ejercicio *mejorar la ecuación cuadrática* para construir un programa en Matlab que calcule las raíces de la ecuación cuadrática en todas las situaciones, incluyendo el caso donde $|b| \approx \sqrt{b^2 - 4ac}$.

El proceso llevado a cabo consiste en pedirle al usuario que digite los valores de a, b y c; estos son interpretados por el algoritmo y revelan la naturaleza de dicha ecuación. Para ello, detecta si la ecuación no es cuadrática; en ese caso, el programa alerta de ello y se pausa; en caso contrario, analiza el valor del discriminante; si es igual a 0, ambas raíces son iguales; si es mayor a 0, muestra el valor de x_1 y x_2 ; si es menor a 0, avisa que la solución de dicha ecuación no se encuentra en los reales. Las figuras 17 y 18 muestran el código implementado en Matlab:

```

Editor - /home/hendrik/Documents/análisis numérico/Lab1/quadratic_formula_hendrik_l...
: floating_point_hendrik_lopez.m x quadratic_formula_hendrik_lopez.m x +
1 %Programa para calcular las raices cuadradas de una ecuación de la forma
2 %ax^2+bx+c=0, donde a != 0 y b^2-4ac > 0
3 %Hendrik López 2170129
4 clc; clear;
5 %Se le pide al usuario que ingrese los valores de a,b y c
6 mensaje_a = 'Ingrese el valor de a: ';
7 a = input(mensaje_a);
8 mensaje_b = 'Ingrese el valor de b: ';
9 b = input(mensaje_b);
10 mensaje_c = 'Ingrese el valor de c: ';
11 c = input(mensaje_c);
12
13 excluyente = b^2-4*a*c;
14

```

Figura 17. Cálculo de raíces de una ecuación cuadrática (parte 1).

```

15 %El algoritmo se detendrá si a = 0 o si b^2-4ac <= 0
16
17 if a == 0
18     disp('No es una ecuación cuadrática.');
```

Figura 18. Cálculo de raíces de una ecuación cuadrática (parte 2).

e. Encuentre las raíces de las siguientes ecuaciones cuadráticas, usando la función anterior:

- $x^2 - 1,000.001x + 1 = 0$

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 19:

```

Command Window
Ingrese el valor de a:1
Ingrese el valor de b:-1000.001
Ingrese el valor de c:1
El valor de x1 es:
    1.0000e+03

El valor de x2 es:
    1.0000e-03

fx >> |
```

Figura 19. Raíces de la ecuación cuadrática $x^2 - 1,000.001x + 1 = 0$.

- $x^2 - 10,000.0001x + 1 = 0$

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 20:

```

Command Window
Ingrese el valor de a:1
Ingrese el valor de b:-10000.0001
Ingrese el valor de c:1
El valor de x1 es:
    1.0000e+04

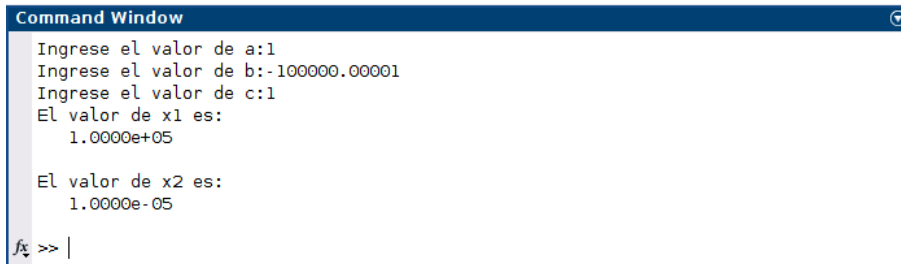
El valor de x2 es:
    1.0000e-04

fx >>
```

Figura 20. Raíces de la ecuación cuadrática $x^2 - 10,000.00001x + 1 = 0$

- $x^2 - 100,000.00001x + 1 = 0$

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 21:



```
Command Window
Ingrese el valor de a:1
Ingrese el valor de b:-100000.00001
Ingrese el valor de c:1
El valor de x1 es:
    1.0000e+05

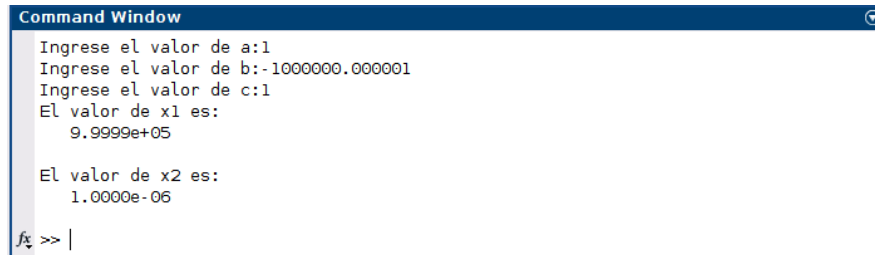
El valor de x2 es:
    1.0000e-05

fx >> |
```

Figura 21. Raíces de la ecuación cuadrática $x^2 - 100,000.00001x + 1 = 0$

- $x^2 - 1,000,000.000001x + 1 = 0$

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 22:



```
Command Window
Ingrese el valor de a:1
Ingrese el valor de b:-1000000.000001
Ingrese el valor de c:1
El valor de x1 es:
    9.9999e+05

El valor de x2 es:
    1.0000e-06

fx >> |
```

Figura 22. Raíces de la ecuación cuadrática $x^2 - 1,000,000.000001x + 1 = 0$

Referencias

- Argüello, H. (2021). *Numerical Methods Preliminaries* [Diapositivas]. Universidad Industrial de Santander. <https://es.scribd.com/document/423857218/Chapter-1>
- Helmenstine, A. (2020, 1 julio). *Determining significant figures*. ThoughtCo. <https://www.thoughtco.com/how-to-determine-significant-figures-608326>
- The MathWorks, Inc. (2021). *Centro de ayuda*. MathWorks. <https://la.mathworks.com/help/index.html>