Métodos Numéricos
Graficación de Funciones y Ceros de Funciones
Tarea 1

August 26, 2020

Erika Rivadeneira Pérez
Matemáticas Aplicadas - CIMAT
erika.rivadeneira@cimat.mx

1 Resumen

En el presente reporte se exponen métodos para graficar cualesquir función real, el primer método transforma coordenadas reales de la función a coordenadas en pixeles de la pantalla. El segundo método utiliza librerías de *python* como mathplotlib y numpy. Además, se indican ejemplos de acercamientos o zoom en las gráficas aplicando un cambio de intervalo. Por otro lado, se presentan el método de Bisección y método de Newton-Raphson para encontrar ceros en funciones. Posteriormente, se usan cuatro ejemplos de funciones, las cuales son graficadas y evaluadas en ambos métodos para encontrar sus respectivas raíces. Por último, se comparan los resultados obtenidos al utilizar los métodos indicados.

2 Metodología

Se sabe que una función f es la relación o mapeo entre dos conjuntos, dominio y codominio, i.e.,

$$\begin{aligned} f :& X \mapsto Y \\ & x \mapsto f(x) = y, \end{aligned}$$

de donde $f(x)$ puede ser de la siguiente forma:

$$f(x) = \sum_{i=0}^{n} a_i x^i + \sum_{i=0}^{n} b_i sin(p_i x_i) + \sum_{i=0}^{n} c_i cos(p_i x_i) + d_i e^{x_i} + ..., \quad i = 0, 1, ..., n, \quad n \in R,$$

con $a, b, c, d \in R$ son constantes arbitrarias. Una presentación gráfica nos permite conocer intuitivamente el comportamiento de dicha función. Se tomará en cuenta funciones de una sola variable, con un sistema de coordenadas cartesianas, donde cada abscisa representa un valor de la variable del dominio y cada ordenada representa el valor correspondiente del conjunto imagen.
En la primera parte se realiza una graficación genérica usando un cambio de coordenadas reales a coordenadas en pixeles de la computadora, asumiento un tamaño de $640 * 480$ pixeles. Para esto es necesario conocer ciertos datos como:

- $X = (x_1, ..., x_n), n \in R$ con $a \leq x \leq b, \quad a, b \in x,$
- $f(X) = Y, \quad Y = (y_1, ..., y_n),$
- Límites superior e inferior de Y y
- $\Delta x = \frac{b-a}{n}$, el incremento de X.

En la segunda parte se muestra un método de graficación sin cambio de coordenadas. Para ambas partes se utilizan librerías del lenguaje *Python* como *numpy, pylab,* y *matplotlib.*
Por otro lado, se utiliza la gráfica de la función para visualizar la localización de los ceros de la función. Para encontrar estos ceros se consideran el método de bisección y el método de Newton Raphson.

2.1 Método de Bisección y de Newton-Raphson para ceros de funciones

El *método de bisección* consiste en obtener una mejor aproximación de la raíz a partir de un intervalo inicial y está basado en la siguiente propiedad:
Propiedad 1. (teorema de ceros para funciones continuas)
Dada una función continua $f : [a, b] \mapsto R$, tal que $f(a)f(b) < 0$, entonces $\exists \alpha \in (a, b)$ tal que $f(\alpha) = 0$.

Empezando desde $I_0 = [a, b]$, el método de bisección genera una secuencia de subintervalos $I_k = [a_k, b_k], k \geq 0$, con $I_k \subset I_{k1}, k \geq 1$, cuya propiedad es $f(a_k)f(b_k) < 0$. Precisamente, hacemos $a_0 = a$, $b_0 = b$ y $x(0) = \frac{(a_0+b_0)}{2}$ [1] .

Por otro lado, *el método de Newton-Raphson* es una aplicación del cálculo diferencial que se utiliza para hallar los ceros de una función derivable de enésimo grado con la precisión deseada. Los procedimientos para hallar las raíces o ceros de funciones lineales o cuadráticas a partir de los coeficientes de la ecuación son sencillos y exactos. Este método está basado en los polinomios de Taylor, esta derivación particular produce no solo el método, sino también un límite para el error de la aproximación.
Asuma que $f \in C^2[a, b]$. Sea $p_0 \in [a, b]$ una aproximación a p tal que $f'(p_0) \neq 0$ y $|p - p_0|$ es 'pequeño'. Considere el primer polinomio de Taylor para $f(x)$ expandir alrededor de p_0 y evaluar en $x = p$:

$$f(p) = f(p_0) + f(p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p))$$

donde $\xi(p)$ se encuentra entre p y p_0. Como $f(p) = 0$, esta ecuación se convierte en

$$0 = f(p_0) + f(p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p))$$

El método de Newton se deriva al asumir que desde $|p - p_0|$ es pequeño, el término que involucra $(p - p_0)^2$ es mucho más pequeño, así

$$0 \approx f(p_0) + f(p - p_0)f'(p_0)$$

Resolviendo para p nos da

$$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} = p_1$$

Esto prepara el escenario para el método de Newton, que comienza con una aproximación inicial p_0 y genera la secuencia $\{p_n\}_{n=0}^{\infty}$ [2], por

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{para } n \geq 1$$

3 Desarrollo

3.1 Método de graficación

Se tienen como datos iniciales los vectores X, Y y los límites (a, b) de X, a partir de esto se puede calcular directamente el incremento de X, de la siguiente manera:

$$\Delta x = \frac{b - a}{n}$$

de este modo tendremos situados en una posición específica cada punto, posterior a esto se evalua cada punto en la función para obtener Y, es decir,

$$\begin{aligned}
x[0] &= a \qquad \mapsto y[0] = f(a) \\
x[1] &= a + \Delta x \mapsto y[1] = f(a + \Delta x) \\
x[2] &= a + 2\Delta x \mapsto y[2] = f(a + 2\Delta x) \\
&\vdots \\
x[n] &= a + n\Delta x \mapsto y[n] = f(a + n\Delta x)
\end{aligned}$$

A partir de aquí es posible determinar los límites de nuestra función $f(x) = y$. Se muestra a continuación el pseudocódigo del procedimiento para la obtención de y_{min} y y_{max}:

```
y_min = 1e20
y_max = 1e-20
for ( i = 0:n ):
    if ( y_min > y[i] ):
        y_min = y[i]
    if ( y_max < y[i] ):
        y_max = y[i]
```

En este punto se considera que se tiene una pantalla de $640x480$ pixeles, tomando como origen la esquina superior izquierda de la pantalla. Para que la imagen se centre se pone como borde 10 pixeles hacia cada lada, es decir, el centro de la imagen tendrá una dimensión de $620x440$. Ahora, teniendo los márgenes de la imagen lo que queda es conocer la posición $(x_i, y_i) \to i$ de cada pixel, es decir, tener ya el cambio de coordenadas reales a coordenadas en pixeles. Para lograr esto debemos calcular el ancho y largo de cada pixel de la siguiente manera:

$$p\Delta x = \frac{b-a}{620}$$
$$p\Delta y = \frac{y_{max} - y_{min}}{440},$$

en donde $p\Delta x$ representa el ancho de cada pixel y $p\Delta y$ representa la altura del mismo. De aquí, para tener la coordenada del pixel, p, del punto x_i y del punto y_i se hace:

$$px_i = \frac{x_i}{p\Delta x} + 10$$
$$py_i = \frac{y_{max} - y_i}{p\Delta y} + 10,$$

siendo (px_i, py_i) las coordenadas en pantalla de la función.

El algoritmo usado para la implementación del método donde se considero la función $f(x) = sin(x)$, donde x es un arreglo de 100 puntos igualmente espaciados, $\Delta x = 0.1$.

3.2 Método de bisección

Para el método de bisección se utilizó la siguiente rutina para $k > 0$:

$$\begin{aligned}
&\text{Hacemos } a_{k+1} = a_k, b_{k+1} = x_k \qquad \text{si } f(x_k)f(a_k) < 0;\\
&\text{Hacemos } a_{k+1} = x_k, b_{k+1} = b_k \qquad \text{si } f(x_k)f(b_k) < 0;\\
&\text{finalmente, hacemos } x_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}
\end{aligned}$$

Los argumentos de entrada para la función denominada "*biseccion*" son:

- Vector de puntos x
- Función f que haga $f(x) = y$
- La tolerancia y
- Número máximo de iteraciones

Por otro lado, la salida es la coordenada de la raíz aproximada de $y = f(x)$ o el aviso de que no existen raíces para la función en el intervalo dado, el número de iteraciones y por último se calcula el error relativo.

3.3 Método de Newton-Raphson

El método de Newton-Raphson sigue las siguientes instrucciones:

```
Step 1. Set i=1
Step 2. while i<= max_iteraciones do Steps 3-6
     Step 3. Set x=x0-f(x0)/f'(x0)
     Step 4. If |x-x0|<Tolerancia then
            Return x
     Step 5. Set i=i+1
     Step 6. Set x0=x
```

Los parámetros de entrada para la función que se denominó "newton-Raphson" son:

- x_0 valor inicial
- f que evalue $f(x) = y$
- f' que retorne la derivada de $f(x)$

Esta función ya establece la tolerancia (1.e-10) y el número máximo de iteraciones en las cuales se considera 1000.
El output de esta función es la raíz aproximada de la función dada, el número de iteranciones y además se calcula el error relativo del resultado.

Para probar estos métodos se consideraron las funciones:

1. $y = sin(x)$ en $(-2\pi, 2\pi)$ para los métodos de graficación.
2. Para el método de Bisección y Newton-Raphson se usan las funciones:
   - x^2 en $(-1, 1)$
   - $x^3 - 2x^2 + 8x - 1$ en $(-3, 3)$
   - $y = sin(x)$ en $(-2\pi, 2\pi)$
   - $ln(x)$ en $[0, 2)$

4 Resultados

La figura (1) muestra el plot de la función $f(x) = sin(x)$ con coordenadas de pixeles en pantalla, mientras que la figura (2) muestra la gráfica de la misma función pero usando la librería Matplotlib de *python*. Por otro lado, las figuras (3) y (4) muestran un "zoom" de la función haciendo un cambio de intervalo.

Los resultados de los métodos para encontrar ceros de funciones fueron los siguiente:

1. x^2 en $(-1, 1)$

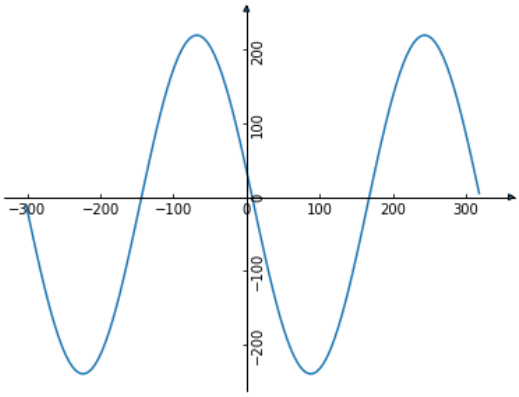

Figure 1: Gráfica con coordenadas de pixeles en pantalla para f(x)=sin(x)

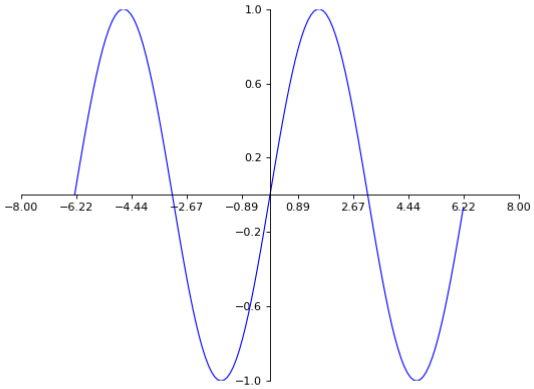

Figure 2: Gráfica con coordenadas reales usando Matplotlib para f(x)=sin(x)

- Utilizando el método de bisección se obtuvo:

```
No hay raíces en este intervalo
```

- Usando el método de Newton-Raphson se obtuvo:

```
La raíz de f(x)=x^2 es 0.0
Número de iteraciones:  30
Error relativo: 9.999999999991326e-09
```

  Proyectando la raíz en la gráfica de $sin(x)$

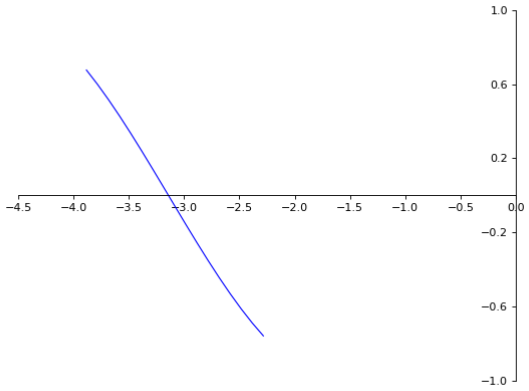

Figure 3: Zoom de $f(x) = sin(x)$ en $(-4, -2.5)$

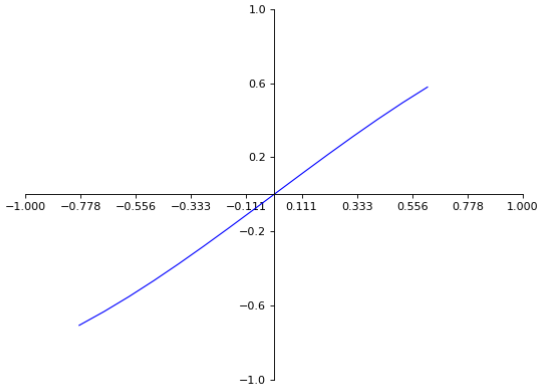

Figure 4: Zoom de $f(x) = sin(x)$ en $(-0.8, 0.6)$

2. $x^3 - 2x^2 + 8x - 1$ en $(-3, 3)$

   - Utilizando el método de bisección se obtuvo:

     ```
     La raíz de x^3-2x^2+8x-1 se encuentra en la coordenada:
                (0.1286132812500016, -0.002049061569373234)
     La raíz se encontró en la iteración número 10
     El error relativo es:  0.002049071569373234
     ```

   - Usando el método de Newton-Raphson se obtuvo:

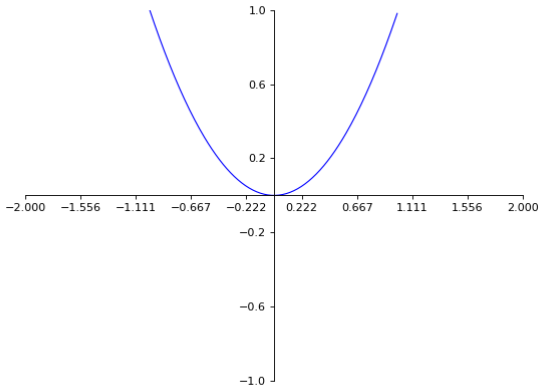

Figure 5: Gráfica de x^2 en $(-1, 1)$

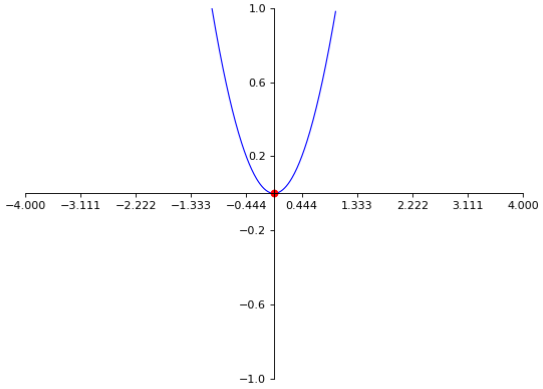

Figure 6: Gráfica de x^2 en $(-1, 1)$ con su raíz

```
La raíz de y(x)=x^3-2x^2+8x-1 es 0.1288852301
Número de iteraciones:  4
El error relativo es:  1e-08
```

3. $y = sin(x)$ en $(-2\pi, 2\pi)$

   - Utilizando el método de bisección se obtuvo:

     ```
     La primera raíz de sin(x) se encuentra en la coordenada:
     ```

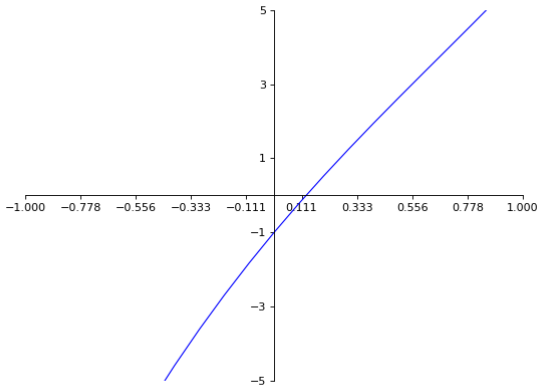

Figure 7: Gráfica de $x^3 - 2x^2 + 8x - 1$ en $(-3, 3)$

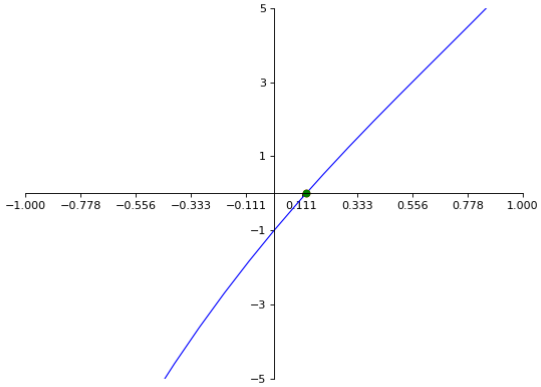

Figure 8: Gráfica de $x^3 - 2x^2 + 8x - 1$ en $(-3, 3)$ con su raíz

```
(-3.1355290571795935, -0.006063559253356268)
La raíz se encontró en la iteración número 7
El error relativo es:  0.006063569253356268

La segunda raíz de sin(x) se encuentra en la coordenada:
(-0.0042790571795915345, -0.004279044121111735)
La raíz se encontró en la iteración número 8
El error relativo es:  0.004279054121111735
```

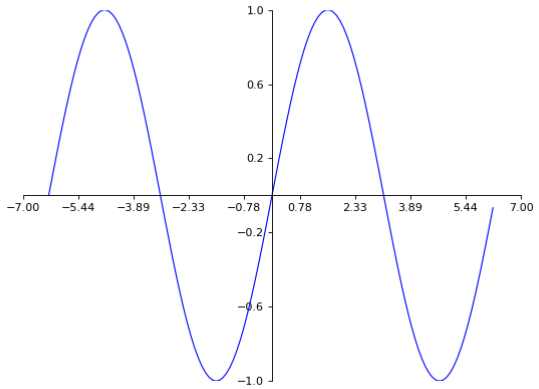

Figure 9: Gráfica de $y = sin(x)$ en $(-2\pi, 2\pi)$

```
La tercera raíz de sin(x) se encuentra en la coordenada:
(3.138689692820409, 0.0029029566920899894)
La raíz se encontró en la iteración número 8
El error relativo es:  0.0029029466920899895
```

- Usando el método de Newton-Raphson se obtuvo:

```
La primera raíz de y(x)=sin(x) es -6.2831853072
Número de iteraciones:  4
El error relativo es:  9.99999975507064e-09

La segunda raíz de y(x)=sin(x) es -3.1415926536
Número de iteraciones:  4
El error relativo es:  1.000000012246468e-08

La tercera raíz de y(x)=sin(x) es 0.0
Número de iteraciones:  3
El error relativo es:  1e-08

La cuarta raíz de y(x)=sin(x) es 3.1415926536
Número de iteraciones:  4
El error relativo es:  9.99999987753532e-09
```

En la gráfica (10) se sobreponen las raíces encontradas con ambos métodos. Los círculos de color verde son las raíces encontradas por el método de Newton-Raphson mientras que los círculos rojos son las raíces encontradas por el método de Bisección.

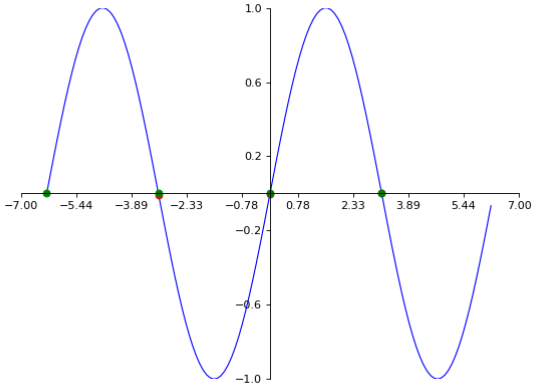

Figure 10: Gráfica de $y = sin(x)$ en $(-2\pi, 2\pi)$ con sus raíces

4. $ln(x)$ en $[0, 2)$

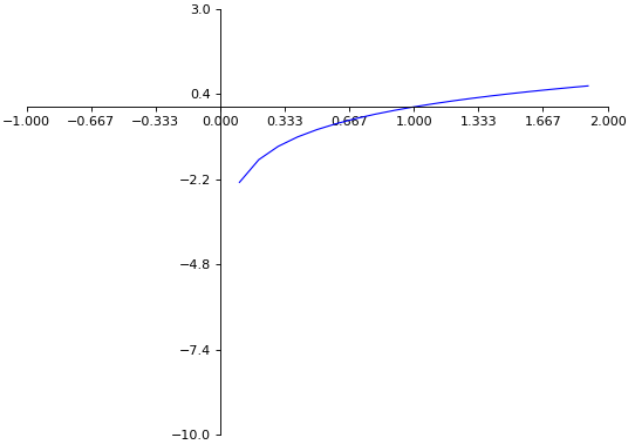

Figure 11: Gráfica de $ln(x)$ sin validar x $[0, 2)$

Se puede observar en (13) si graficamos con este intervalo, se muestra parcialmente la gráfica de $y = ln(x)$, mientras que si validamos x haciendo $f(x) = ln(x + \epsilon)$ con $\epsilon = 1e - 21$ se puede apreciar la gráfica completa como se ve en (12). Esto se debe a que $ln(x)$ no está definido en 0.

- Utilizando el método de bisección se obtuvo:

```
La raíz de f(x)=ln(x) se encuentra en la coordenada:
(1.0019531250000004, 0.0019512201312621926)
```

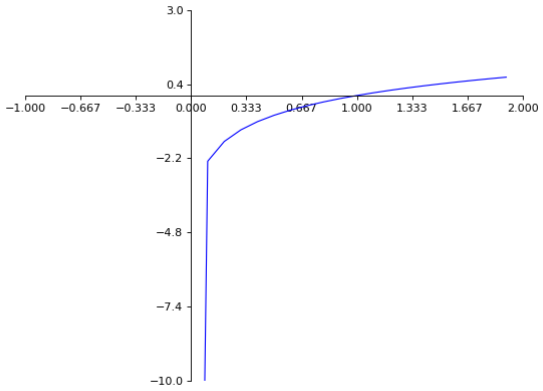

Figure 12: Gráfica de $ln(x)$ validando x validar x $[0, 2)$

```
La raíz se encontró en la iteración número 8
El error relativo es:  0.0019512101312621926
```

- Usando el método de Newton-Raphson se obtuvo:

```
La raíz de y(x)=ln(x) es 1.0
Número de iteraciones:  5
El error relativo es:  1e-08
```

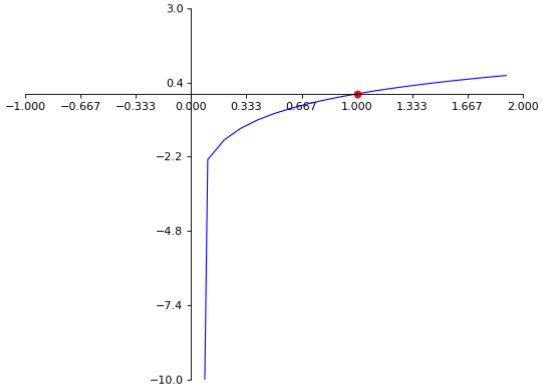

Figure 13: Gráfica de $ln(x)$ en $[0, 2)$ con su raíz

5 Discusión

Se puede observar que ambos métodos muestran plots de la función con ejes centrados en el origen $(0, 0)$. Usando el primer método con cambio de coordenadas se debe conocer la posición de cada punto para poder realizar cualquier tipo de operación con la función y mostrar el resultado gráficamente. Además se puede observar que una vez que se conocen las coordenadas en pantalla es muy sencillo hacer un *zoom*, con tan solo un cambio de intervalo. Por otro lado, se debe conocer el tamaño de la pantalla que se usa, en pixeles, para poder utilizar este método de graficación. Además, hay que tener en cuenta que si se consideran pocos puntos es probable que haya que calcular las distancias de los centroides de cada pixel a la recta que une dos puntos distantes para que se pueda visualizar bien la gráfica. Es importante mencionar que usando la librería matplotlib se puede editar o personalizar a conveniencia cada gráfica para una mejor visualización de los datos.

Por otro lado, el segundo método mostrado no requiere de un cambio de coordenadas como el visto en el anterior método. Con esta técnica también se puede acer un acercamiento cambiando de intervalos e igualmente es posible manipular los ejes de coordenadas, en este caso se lo centró en el origen. Esto puede ser ventajoso para realizar diversos estudios para conocer el comportamiento de la función, como visualizar los ceros de la función.

En cuanto a los métodos para encontrar los ceros de funciones, se puede observar que para la función x^2 en $(-1, 1)$ el método de bisección no encontró la raíz dado que $f(a) * f(b) > 0$, es decir, no existe un cambio de signo, mientras que el método de Newton-Raphson sí encontró la raíz en 0. También es notorio que el número de iteraciones del método de Newton es menos que el del método de bisección, mientras que los errores relativos no varían mucho. Además, es necesario mencionar que para el ejemplo de la función $sin(x)$ en $(-2\pi, 2\pi)$ tenemos varias raíces las cuales no son encontradas simulatáneamente con ambos métodos, ya que los métodos convergieron únicamente a una raíz, por esta razón es necesario cambiar los intervalos de estudio, cercanos a cada cero, para el caso del método de bisección o un punto que sabemos que está cercano a la raíz para el método de Newton. Con el método de Newton se encontró una raíz adicional en -6.2831853072 que con el método de bisección, esto igualmente es debido a que alrededor de este punto no hay un cambio de signo, por tanto el segundo método mencionado no considera esta raíz. Finalmente, para obtener una visualización deseada de una función es necesario conocer el comportamiento de la misma, como por ejemplo en la figura (12) no se puede observar la función completa ya que se la graficó en un intervalo donde la función no está definida pero si se validan los valores de x ya se puede obtener una gráfica adecuada.

6 Conclusiones

Ambos métodos permiten visualizar bien el comportamiento de la función pero al hacer uso del primer método puede existir un error de locación de alguno de los puntos $x_i, i = 1 : n$ lo cual conlleva a cometer errores al momento de realizar alguna operación con los datos o al momento de transformar las coordenadas de pixeles a reales o viceversa. Se concluye que el segundo método es más ventajoso que el primero y más sencillo de usar.
Por otro lado, el método más eficaz para encontrar ceros en funciones es el de Newton-Raphson debido a que este no se limita a encontrar raíces en intervalos restringidos en los que debe haber cambio de signo, $f(a)f(b) < 0$. Además, el número de iteraciones es menor que el del método de Bisección, aunque el error relativo entre ambos métodos no varia mucho.

References

[1] A. Quarteroni, P. Quarteroni, F. Riccardo Sacco, R. Sacco, and F. Saleri, Numerical Mathematics, ser. Texts in Applied Mathematics. Springer, 2007. [Online]. Available: https://books.google.com.ec/books?id=Y7grAAAAYAAJMathematician105Final Grade Project

[2] A. Quarteroni, P. Quarteroni, F. Riccardo Sacco, R. Sacco, and F. Saleri,NumericalMathematics, ser. Texts in Applied Mathematics.Springer, 2007. [Online].Available: https://books.google.com.ec/books?id=Y7grAAAAYAAJMathematician105Final Grade Project