Tarea 1. Graficador de funciones

Salma Patricia Gutiérrez Rivera
Maestría en Computación
salma.gutierrez@cimat.mx

August 22, 2022

1 Introducción

Se implementó un graficador de funciones de una variable en C++ utilizando las librerías Cairo(*Cairo Graphics Documentation*, 2016) y fparser(*Function Parser for C++ Documentation*, 2011).

2 Metodología

Los pasos que sigue el programa son:

1. Leer y evaluar la función con fparser.
2. Encontrar el rango de la función para poder establecer la escala del eje y.
3. Dibujar el fondo de la gráfica.
4. Etiquetar los ejes coordenados.
5. Finalmente, graficar la función.

En el paso 1, el graficador evalúa las funciones en cincuenta puntos equidistantes ubicados entre el dominio [a, b], dado por el usuario. Cincuenta puntos es suficiente para dar una apariencia ”suave” a casi cualquier función. Los puntos se guardan en el vector **misxs** y las evaluaciones en **fdex**.

El paso 2 existe porque ambos ejes coordenados tienen la misma longitud pero distinta escala. Es decir, distancias equivalentes en pixeles entre marcas de los ejes pueden representar distancias diferentes en la recta de los reales. Para encontrar la escala del eje y, se divide la longitud del dominio $[a, b]$ entre el número de divisiones que tendrá la cuadrícula, que en este caso está fijo en 10. Para la escala del eje x, lo que se divide es el rango de la función en ese dominio. Las variables que representan estas escalas son **metrica ejex** y **metrica ejey**, utilizadas para dibujar la cuadrícula, y **avancex** y **avancey**, utilizadas para etiquetar los ejes y más adelante para graficar.

El rango de la función se fija buscando los valores máximo y mínimo del vector **fdex**.

La parte del código utilizado en esta tarea para el paso 3 (dibujar los ejes, el fondo y la cuadrícula de la gráfica) fue tomado de las notas del Curso 0 sobre Cairo impartido por el Dr. Iván Cruz el pasado 4 de agosto. Los ejes se cruzan en la esquina inferior izquierda de la superficie de pintado en lugar de cruzarse en el origen.

Para el paso 4, es necesario transformar los números contenidos en el vector **metricax** a strings. Este problema lo resolví gracias al link https://stackoverflow.com/questions/29200635/convert-float-to-string-with-precision-number-of-decimal-digits-specified. Declaro la variable stream en cada iteración del ciclo donde la utilizo porque fue la única forma que encontré de evitar un bug que repetía las etiquetas.

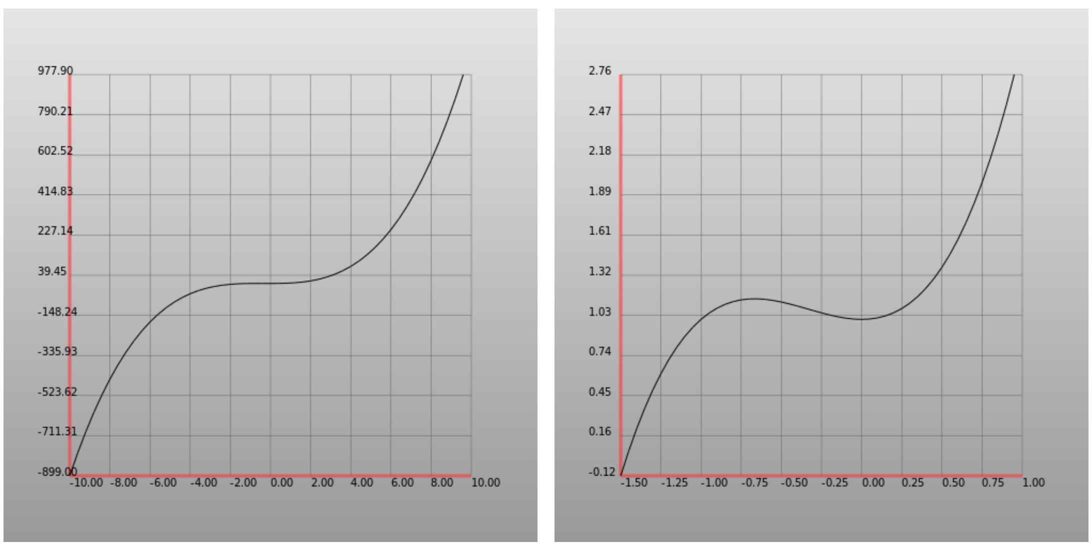
Para el último paso, primero hay que ubicar los puntos $(x, f(x))$ en la gráfica y después unirlos con líneas rectas. Para conseguirlo, a cada entrada de **misxs** se le suma la coordenada del origen del eje x mas la distancia al origen real (el punto de la gráfica que representa $x = 0$, que en este caso es **abs(a)**). Después se escala con **avancex** para tener la ubicación del punto en pixeles. El proceso es análogo para las coordenadas en y.

En otras palabras, a cada par $(x, f(x))$ se le añade un término de correción para que la gráfica se centre en el origen de coordenadas y se respete la escala de los ejes al traducir a pixeles.

3 Resultados

Función $f(x) = x^3 + x^2 + 1$

Se observa en la figura 1a que en [-10, 10] la función luce achatada a pesar de que el rango en el eje **y** es mucho mayor al dominio en el eje **x**. En la figura 1b, se aprecia un poco mejor el punto de inflexión que presenta la función en $(-1/3, 29/27)$.



(a) (-10, 10)

(b) (-1.5, 1)

Figure 1: $f(x) = x^3 + x^2 + 1$ en los dominios [-10, 10] y [-1.5, 1]

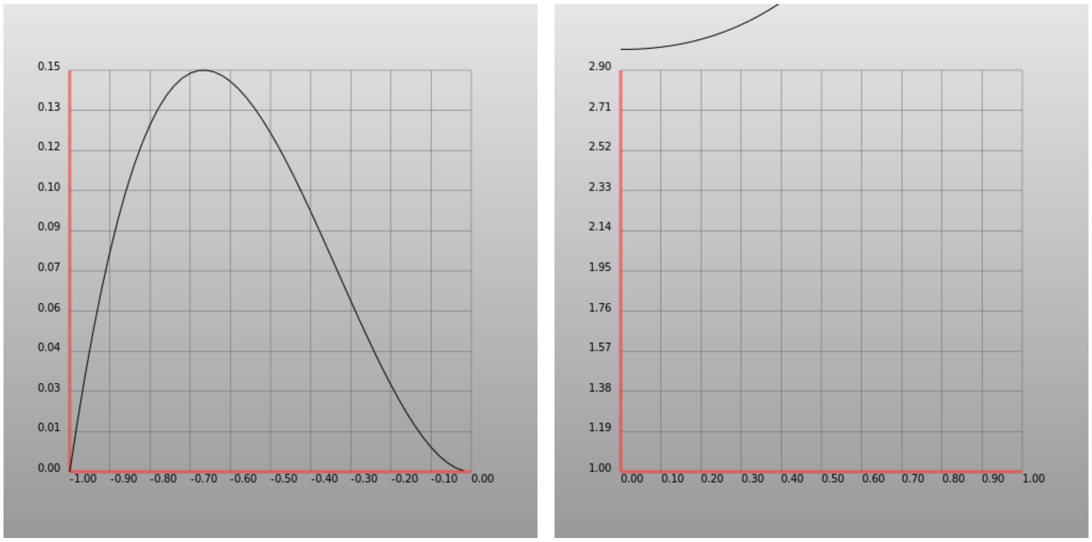
El comportamiento en $[-1, 0]$ es el esperado (fig. 2a), pero en $[0, 1]$ la gráfica se sale de la cuadrícula (fig. 2b).

Función $g(x) = \sin(10x)$

Para las funciones senoidales convendría utilizar más puntos para graficar. La función en la figura 3 no luce lo suficientemente suave. Esto mejora en las figuras 4a y 4b, donde el dominio es más pequeño.

Función $h(x) = \frac{1}{x+1}$

La función tiene una asíntota vertical en $x = -1$ y tiende a menos infinito por la izquierda y a mas infinito por la derecha de esta asíntota. El graficador no puede ver esto. Como se observa en la figura



(a) (-1, 0)

(b) (0, 1)

Figure 2: $f(x) = x^3 + x^2 + 1$ en los dominios [-1, 0] y [0, 1]

5, lo que hace es unir el punto más cercano a $x = -1$ por la izquierda con el punto más cercano a $x = -1$ por la derecha.

En las figuras 6a y 6b se observa la función en un dominio cercano a la asíntota por la izquierda y otro cercano por la derecha. Se observa que $h(x)$ se aleja mucho más del cero en estas dos figuras que en la anterior. Esto es porque la función se evaluó en valores mucho más cercanos al punto de indeterminación.

4 Conclusiones

Posibles mejoras al graficador:

1. Dar la opción al usuario de elegir la cantidad de puntos para graficar, aunque con mil la creación de la imagen png de salida toma demasiado tiempo.
2. Adaptar los ejes coordenados para que su longitud sea proporcional a los intervalos que representan, que ambos tengan la misma métrica y que se crucen en el origen.

References

Cairo graphics documentation. (2016). Retrieved 2022-08-21, from `https://www.cairographics.org/documentation/`

Function parser for c++ documentation. (2011). Retrieved 2022-08-21, from `http://warp.povusers.org/FunctionParser/fparser.html`

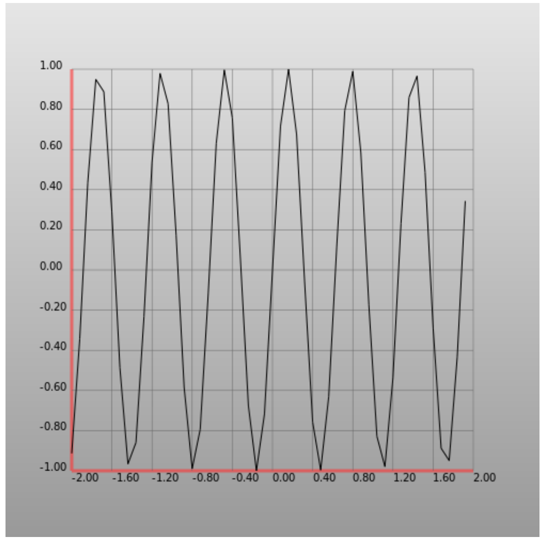
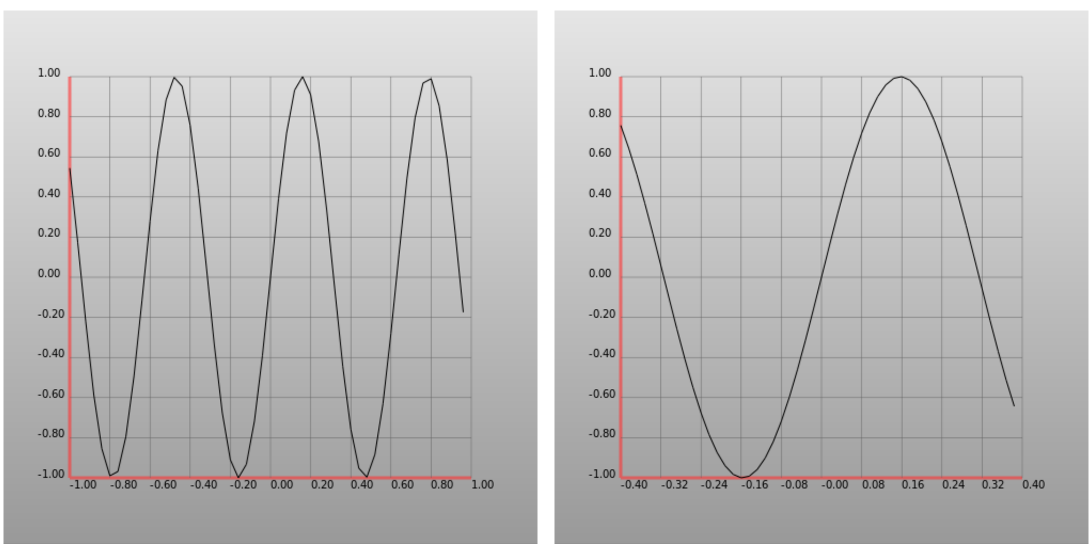


Figure 3: $g(x) = \sin(10x)$ en el dominio [-2, 2]



(a) (-1, 1)

(b) (-0.4, 0.4)

Figure 4: $g(x) = \sin(10x)$ en los dominios [-1,1] y [-0.4, 0.4]

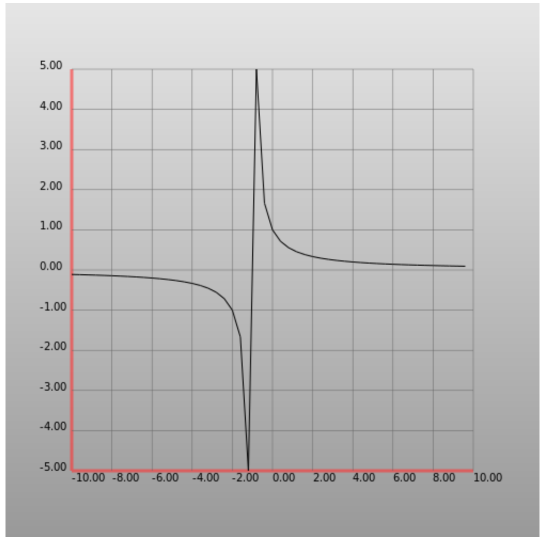
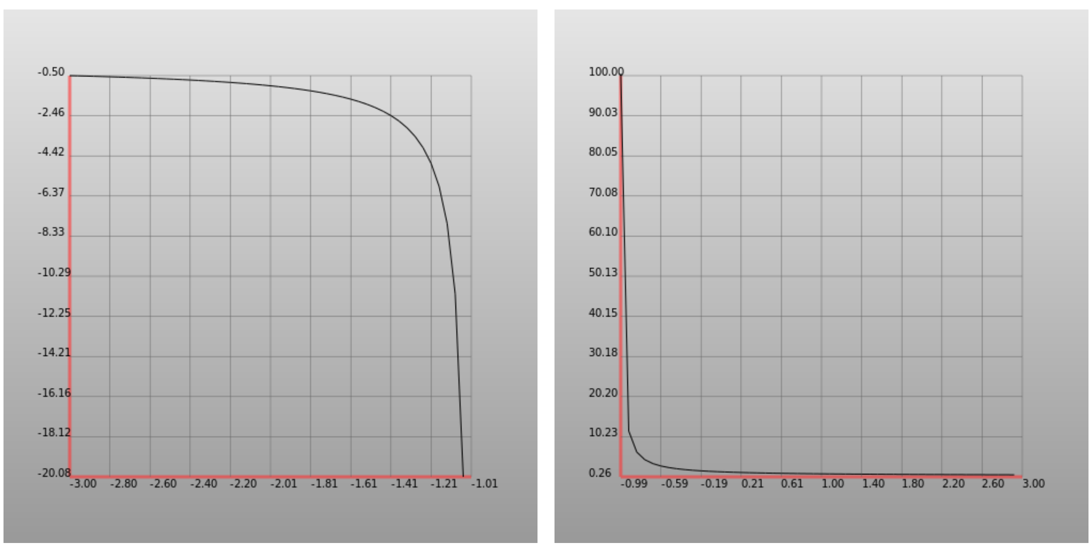


Figure 5: $h(x) = \frac{1}{x+1}$ en el dominio [-10, 10]



(a) (-3, -1.01)

(b) (-0.99, 3)

Figure 6: $h(x) = \frac{1}{x+1}$ en dominios cercanos a la asíntota.