

Fun Graphic

Integrantes:

Alvarado Juan Sebastián.

Fonseca Robert Daniel.

Osorio Wilberson.

Repositorio: proyecto_final.py

<https://github.com/Maicken052/->.git

Resumen:

Solemos trabajar mucho con funciones, aún más siendo de MACC puesto que estas nos ayudan para representar dependencia de dos magnitudes, ciertas veces pasa que como estudiantes no imaginamos siquiera el boceto de cómo se ve esa función, y se vuelve aún más difícil cuando necesitamos saber los puntos de corte o necesitamos alguna coordenada en específico.

La gráfica de una función nos brinda información suficiente para facilitar su análisis y para conocer su comportamiento, por esto es que nos dimos a la tarea de hacer una graficadora de funciones, de esta manera cuando quieras saber por cuales puntos pasar una función lineal o cuál es el vértice de una parábola solo debes ingresar la función y analizar la gráfica. De esta manera planeamos facilitar el trabajo con funciones para los estudiantes de la escuela de ingeniería, ciencia y tecnología, y mejorar nuestra habilidad como programadores, aprendiendo a usar nuevas librerías y métodos de más complejidad.

Funcionalidad:

El usuario no tendrá que especificar qué tipo de función usará. En la interfaz de la graficadora veremos dos recuadros que reciben información y un botón que dice “graficar”. En el recuadro que se encuentra justo arriba del botón de graficar ingresamos la función que deseamos; en el recuadro que se encuentra en la esquina inferior derecha ingresamos separados por coma dos valores para denotar el intervalo que va a tomar la graficadora respecto al eje x (justo debajo de este recuadro nos aparecerán las coordenadas de la ubicación del cursor), después de haber hecho esto oprimimos el botón de “graficar” para que nos muestre la gráfica deseada.

Además de esto tenemos una barra de herramientas las cuales nos permitirán obtener mayor información y movernos de una forma más cómoda por la gráfica, así como también nos permitirá guardar la imagen. De esta manera los usuarios de la graficadora tendrán infinidad

de información sobre la gráfica así como también después de guardar la imagen podrán trabajar sobre ella poniendo datos importantes como puntos de corte, vértices, etc.

Objetivos:

Aviso: Téngase en cuenta que en un inicio los objetivos específicos iban enfocados hacia quién iba a programar cómo se graficaba cierto tipo de función. Pero esto cambió al darnos cuenta que numpy ya contiene las funciones por lo cuál no es necesario hacerlo una por una.

Alvarado Juan Sebastián: En primera instancia fue quién dio el paso inicial al hacer un código para que python muestre la gráfica de una función en la consola.

Osorio Wilberson: Hacer con tkinter una ventana que se asemeje a una interfaz de un plano cartesiano, agregar recuadros para ingresar texto y botón para graficar más adelante.

Fonseca Daniel: Hacer que la interfaz sea funcional, asignando así una función al botón “graficar” y creando una ventana emergente cuando se introduce un rango incorrecto.

El programa no grafica:

Funciones racionales, función de tangente, función de una raíz impar.

Aclaraciones:

Cuando desee elevar un número dentro de Fun Graphic use doble signo de multiplicación (**). cuando desee sacar raíz use $\sqrt[n]{x}$ o eleve al 0.5 si es raíz cuadrada, 0.25 si es raíz cuarta, etc.

Herramienta de programación:

-Tkinter: Módulo usado para crear la interfaz gráfica que trae diferentes tipos de widgets.

-Matplotlib: Biblioteca que usamos para la generación de gráficos usando matplotlib.pyplot. Siguiendo con matplotlib, el .figure nos permitirá crear una figura y con ciertas propiedades y el style mejorar la apariencia de la gráfica. Un backend nos permitirá interactuar con la ventana, y junto con un navigation importar la barra de herramientas.

-Numpy: se especializa en el cálculo numérico y análisis de datos, además contiene una gran cantidad de funciones matemáticas.

-Math: Nos da acceso a las funciones matemáticas.

Descripción del programa:

i. y ii.

Solo hay un archivo que compone el programa, que claramente es el mismo que contiene el bloque de ejecución, el nombre del archivo es proyecto_final.py.

iii.

En el archivo proyecto_final.py definimos las funciones reemplazar, animate y represent.

iv. (En el archivo del proyecto se encuentran varias líneas de código comentadas para ayudar a entender su funcionamiento)

Primero vamos a importar las librerías que vamos a usar. Después vamos a crear una ventana, le vamos a poner dimensiones y título, procedemos a crear espacios de entrada y botón de graficar y los ubicamos; vamos a crear el contenedor que tendrá la gráfica, importamos los íconos de navegación de la gráfica.

Creamos los rangos que vamos a usar, hacemos un diccionario con las posibles funciones, creamos las funciones que abajo explicamos y asignamos el comando de la función representar al botón “graficar”

v.

Reemplazar: Usamos esta función para cambiar el texto simplificado que se ingresa en el recuadro de la graficadora, y lo convertimos en una de las funciones especiales numpy

(np.sin, np.cos, np.exp, etc), con el fin de que grafique correctamente, y no se tengan que escribir los módulos en el recuadro inicial, facilitando el uso de Fun graphic.

Represent:

Esta función transforma los datos de los recuadros y los guarda en variables para usarlas posteriormente en la función animate, y usa reemplazar por si se ingresó una función especial.

Animate:

Esta función es la encargada de usar los valores obtenidos en los recuadros, y colocarlos en los métodos de matplotlib para obtener la gráfica.

Primero nos encargamos del rango obtenido en el recuadro inferior derecho, que anteriormente dividimos con un split, para facilitar su uso en el método que nos permite establecer los rangos de la gráfica (np.arange). Además de esto, cuenta con una alerta por si se ingresaron mal los datos (orden invertido, solo un dato), y borra instantáneamente los datos erróneos ingresados. Si no se ingresa un rango, establece automáticamente uno del 1 al 10. Finalmente evalúa la función obtenida en el recuadro, borra lo que esté en la ventana (por si se había graficado algo previamente), y se procede a usar el método .plot para finalmente obtener la gráfica. Como la graficadora se abre sin datos previos, se inicializa con valores predeterminados.