



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



Matemáticas Avanzadas para la
ingeniería

PROYECTO

Calculadora de residuos y
contornos

PROFESOR:

Correa Coyac David

REALIZADO POR:

- Hernández Cuéllar Mauricio
- Ortega Marcial Marco Antonio

Introducción

El análisis complejo es una rama de suma importancia de las matemáticas que nos permite resolver integrales definidas que desafían y nos obligan a afrontar los métodos elementales con aplicaciones en campos como la física y la ingeniería.

Ante ello, se busca realizar una herramienta diseñada para facilitar el cálculo de integrales complejas mediante el teorema del residuo a través de una calculadora de residuos y contornos. Este recurso incluye la implementación de funciones como la visualización de los contornos de integración, promoviendo una comprensión mucho más intuitiva de las funciones en el dominio complejo.

Diseñada para estudiantes y profesores, la calculadora permite evaluar integrales de funciones racionales y trascendentales, además, se basa en fundamentos teóricos como el Teorema de Cauchy, series de Taylor, series de Laurent y el cálculo de residuos.

Descripción del problema

En el estudio de funciones complejas, el cálculo de integrales curvilíneas y aplicación del teorema del residuo representan un gran desafío en el ámbito estudiantil y profesional. Las dificultades incluyen aspectos como la identificación y clasificación de polos, seleccionar el contorno de integración, desarrollar series de Laurent para residuos y, así, identificar gráficamente el dominio y las singularidades. Dichos procesos al ser realizados de forma manual tienen un mayor porcentaje de error además de, consumir una mucho mayor cantidad de tiempo y llegan a dificultar el aprendizaje y verificación.

Una situación como esta, resalta la necesidad de una herramienta digital interactiva que permita automatizar el cálculo de residuos e integrales. Además de ayudar de esta manera, dicha herramienta se ofrece como un recurso pedagógico superando así la falta de retroalimentación inmediata.

Fundamentación matemática

El desarrollo de esta herramienta se sustenta primeramente del Teorema del Residuo de Cauchy, una de las herramientas más útiles para el análisis complejo. Este teorema permite evaluar integrales cerradas de funciones con polos aislados, además, dice, que para una función $f(z)$ analítica en una región conexa, salvo en polos a_k dentro de un contorno cerrado “C”, se tiene:

$$\oint_C f(z)dz = 2\pi i \sum_k \text{Res}(f, a_k)$$

En donde $\text{Res}(f, a_k)$ es el residuo en el polo a_k , transformando el cálculo en una suma de residuos.

Cálculo de residuos

Para esto, se hizo uso de $\text{Res}(f, a) = \lim_{z \rightarrow a} f(z)$, mientras que para polos de orden mayor se emplean derivadas de series de Laurent. En el desarrollo del proyecto, se hace uso de la librería **SymPy** en Python para realizar cálculos simbólicos, en el cual, podemos manejar funciones como e^z y sus derivadas así como de funciones trascendentales como seno, coseno, etc.

Identificación de polos y contornos

Los polos se determinan como raíces del denominador de $f(z)$. Se verifica si están dentro de un contorno circular definido por el usuario (centro y radio), comparando distancias. Los residuos de los polos relevantes se calculan de manera simbólica y, el teorema evalúa la integral.

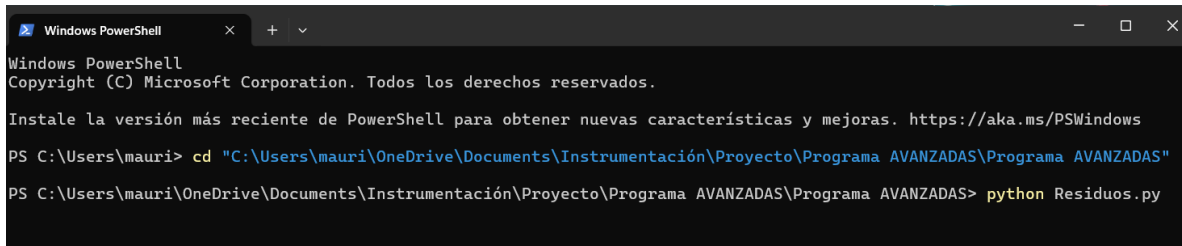
La visualización grafica que está implementada con **matplotlib**, muestra contornos y polos de manera interactiva, reforzando la interpretación.

Esta base teórica combinada con herramientas computacionales asegura una mejor precisión y apoya el aprendizaje de manera visual.

Capturas del sistema

Como primer paso abrimos nuestra terminal, o cualquier aplicación para compilar y ejecutar nuestros programas en Python.

En este caso usaremos PowerShell y nos colocamos en la dirección del programa para ejecutarlo (Importante tener ambos scripts adjuntos en una sola carpeta).

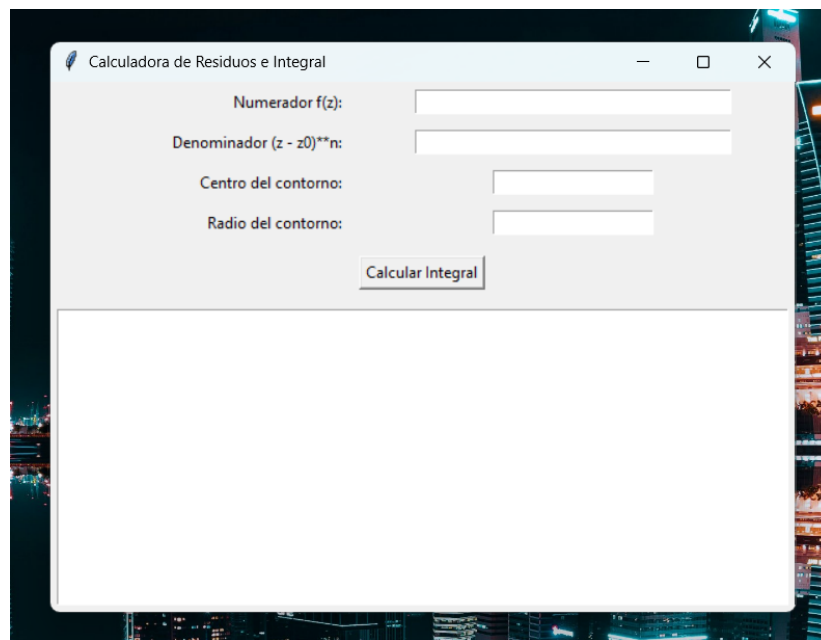


```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

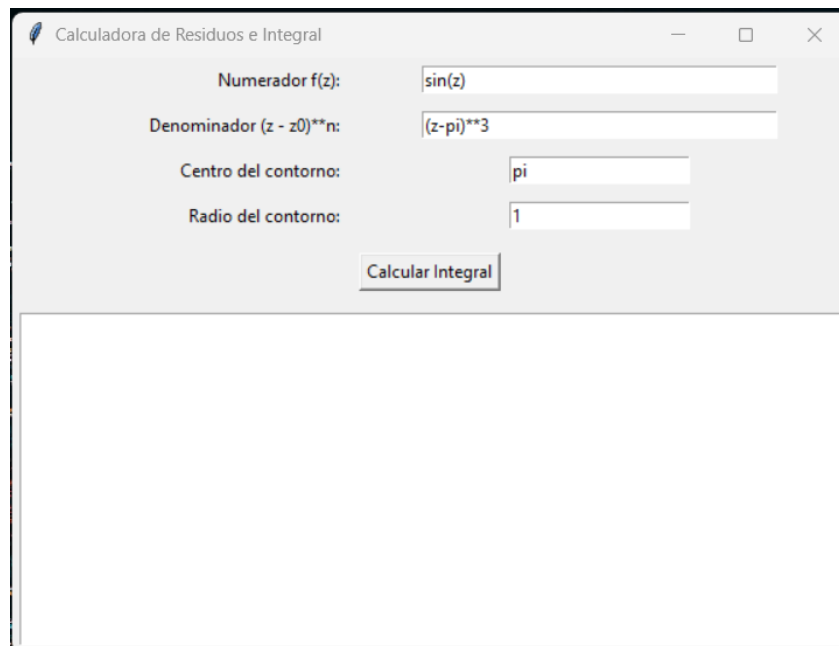
PS C:\Users\mauri> cd "C:\Users\mauri\OneDrive\Documents\Instrumentación\Proyecto\Programa AVANZADAS\Programa AVANZADAS"
PS C:\Users\mauri\OneDrive\Documents\Instrumentación\Proyecto\Programa AVANZADAS\Programa AVANZADAS> python Residuos.py
```

Ejecutamos nuestro código y se abrirá nuestra ventana de interfaz.



Como siguiente paso, el usuario deberá ingresar su función en términos de z , colocando el numerador y denominador como lo indica la interfaz.

Ejemplo:



Calculadora de Residuos e Integral

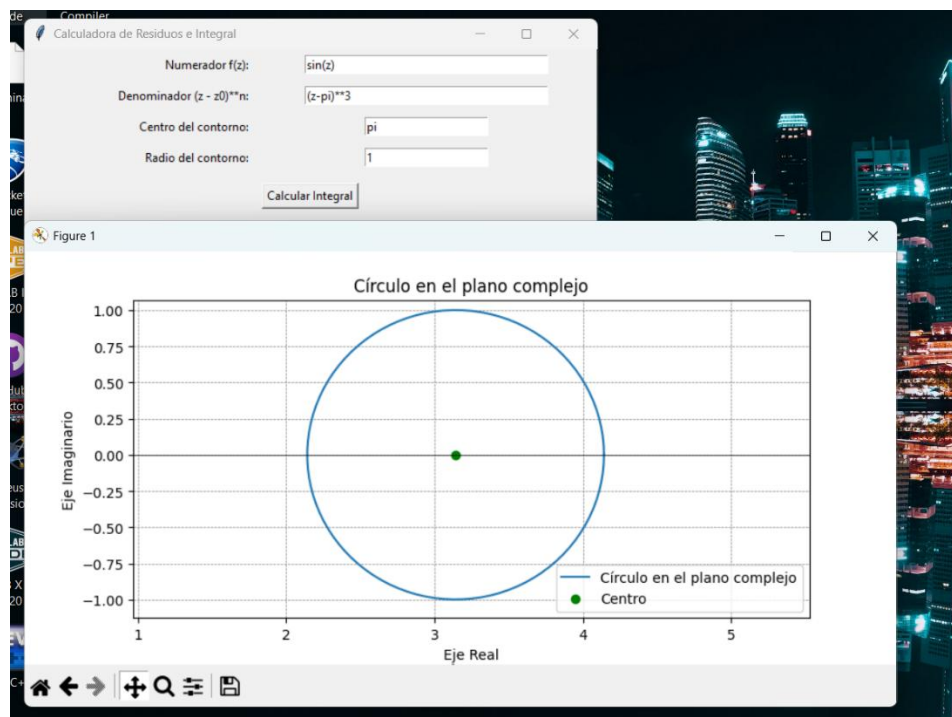
Numerador $f(z)$:

Denominador $(z - z_0)^{**}n$:

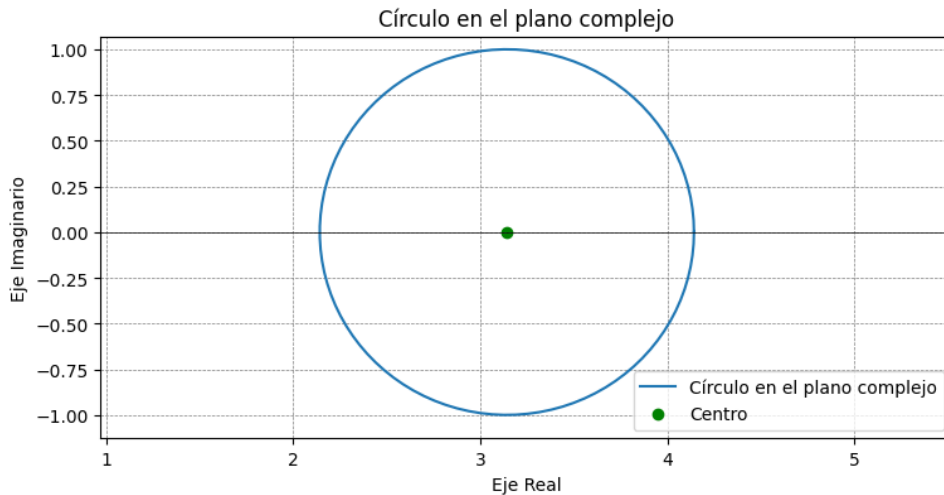
Centro del contorno:

Radio del contorno:

A continuación, pulsamos el botón “Calcular integral” y observamos la ventana de la gráfica



Una vez hecho esto podemos observar el contorno y el centro introducidos previamente por el usuario e incluso se puede explorar la gráfica, así como guardarla en una imagen PNG, solo presionamos el botón inferior derecho, seleccionamos la ruta de guardado y el resultado se verá de la siguiente manera.



Una vez visualizada la función, cerramos la pestaña de la gráfica y obtendremos los resultados del residuo y la aproximación integral.

La interfaz de usuario de la "Calculadora de Residuos e Integral" muestra los siguientes campos de entrada y resultados:

- Numerador $f(z)$: $\sin(z)$
- Denominador $(z - z_0)^n$: $(z - \pi)^3$
- Centro del contorno: π
- Radio del contorno: 1
- Botón: "Calcular Integral"
- Polos encontrados: $[\pi]$
 - Dentro del contorno: $[\pi]$
 - Fuera del contorno: $[\]$
- Residuos en cada polo:
 - Residuo en $z = \pi$: 0
- ✓ Valor de la integral ≈ 0

Como podemos observar, se cumple la condición de que, si el residuo es igual con 0, entonces la integral es igual con 0

Ejemplo 2:

Calculadora de Residuos e Integral

Numerador $f(z)$:

Denominador $(z - z_0)^{**}n$:

Centro del contorno:

Radio del contorno:

→ Polos encontrados: [1]
• Dentro del contorno: [1]
• Fuera del contorno: []

→ Residuos en cada polo:
Residuo en $z=1$: 1

✓ Valor de la integral $\approx 2 \cdot I \cdot \pi$

Ejemplo 3:

Calculadora de Residuos e Integral

Numerador $f(z)$:

Denominador $(z - z_0)^{**}n$:

Centro del contorno:

Radio del contorno:

→ Polos encontrados: [$\pi - \text{asin}(2)$, $\text{asin}(2)$]
• Dentro del contorno: [$\text{asin}(2)$]
• Fuera del contorno: [$\pi - \text{asin}(2)$]

→ Residuos en cada polo:
Residuo en $z=\pi - \text{asin}(2)$: $\sqrt{3} \cdot I / 9$
Residuo en $z=\text{asin}(2)$: $-\sqrt{3} \cdot I / 9$

✓ Valor de la integral $\approx 2 \cdot \sqrt{3} \cdot \pi / 9$

Como podemos observar en los ejemplos anteriores, hemos comprobado los casos en los que hay 0,1 o 2 polos dentro del contorno de forma exitosa, por lo que la interfaz funciona de manera óptima.

Resultados

La herramienta desarrollada gracias al módulo de Python que actúa como una interfaz para la biblioteca gráfica Tk, permitió implementar de manera exitosa una gráfica interfaz interactiva que permite y facilita al usuario el análisis de integrales complejas mediante el teorema del residuo. A través de múltiples pruebas y validaciones realizadas con diferentes tipos de funciones y configuraciones del contorno, se obtuvieron los siguientes resultados:

- Ingresar funciones racionales o trascendentales.
- Establecer el centro y radio del contorno circular.
- Visualizar gráficamente el contorno y los polos en el plano complejo.
- Calcular y mostrar los residuos en cada polo.
- Aplicar el Teorema del Residuo para obtener el valor de la integral de contorno.

Validación mediante casos de prueba:

Durante las pruebas, se consideraron diversos escenarios que nos permitieron confirmar el correcto funcionamiento del proyecto

- Cuando los polos se encontraban fuera del círculo definido, la aplicación arrojó como resultado una integral igual a cero, de acuerdo con el teorema de Cauchy.
- Se verificó que el residuo se calcula correctamente y que la integral resultante es $2\pi i \times \text{Res}(f, a)$, como establece la teoría.
- En funciones con más de un polo dentro del círculo, la herramienta sumó correctamente todos los residuos y devolvió el valor de la integral como $2\pi i \times \sum \text{Res}(f, a_k)$, con resultados constantes y precisos.

Por lo cual, se confirma que la herramienta cumple con los objetivos planteados, no solo llegando a automatizar el cálculo simbólico sino también brindando una representación gráfica intuitiva y fácil de comprender.

Conclusiones

El desarrollo de este trabajo (así como su finalización y validación de pruebas), demostró logró automatizar el calculo de integrales complejas y ofreció una visualización clara de polo, residuos y contornos, gracias a herramientas como SymPy y Tkinter. Esto, logró permitir a los usuarios ingresar funciones, definir contornos y explorar el plano complejo de una manera interactiva, validando la aplicabilidad practica de los conceptos aprendidos.

Así mismo, las pruebas confirmaron su eficiencia en diversos casos como, integrales con cero polos o varios residuos, reforzando mucho mejor nuestro entendimiento. Más allá del objetivo técnico, el proyecto se convirtió en una experiencia más visual e interactiva.

En un futuro, podría expandirse con nuevos contornos o funciones, inspirando más herramientas que fusionen teoría y tecnología para resolver y comprender problemas complejos.

Referencias

- Brown, J. W., & Churchill, R. V. (2009). *Variable compleja y sus aplicaciones* (8ª ed.). McGraw-Hill.
- Arfken, G. B., & Weber, H. J. (2013). *Mathematical Methods for Physicists* (7th ed.). Academic Press.
- Meiss, J. D. (2007). *Differential Dynamical Systems*. Society for Industrial and Applied Mathematics (SIAM).
- Python Software Foundation. (2024). *Python (versión 3.11)* [Software]. <https://www.python.org/>
- SymPy Development Team. (2024). *SymPy: Python library for symbolic mathematics* [Software]. <https://www.sympy.org/>

Link al video de explicación del software:

https://drive.google.com/drive/folders/14_ZB4ZCRQtSFA2BfclSp8GwnrRCykyI?usp=sharing