PROYECTO DE METODOS DIFERENCIAS FINITAS EVALUADOS EN

DIFERENTES CASOS TEMA-1

Realizado por:

David Caleb Chaparro Orozco

Isabela Gómez Ruiz

Cristian David Ocampo Uribe

Cristian Troncoso Guerra

Alexander Villada Berrio

Asignatura:  
Métodos Numéricos

Docente:

Sandra Vicky Zapata García

Ingeniería Informática / Ingeniería de Sistemas

Facultado de Ingeniería

Institución Universitaria de Envigado

Envigado, Antioquia

Mayo 2024

**Tabla de contenido**

[Resumen 2](#_Toc167610293)

[**Objetivos** 3](#_Toc167610294)

[**Justificación** 4](#_Toc167610295)

[**Marco Conceptual** 5](#_Toc167610296)

[**Desarrollo** 6](#_Toc167610297)

[**Solución de los Casos Propuestos** 11](#_Toc167610298)

[**Conclusiones** 15](#_Toc167610299)

[**Referencias** 16](#_Toc167610300)

Resumen

En relación con el aprendizaje de las ecuaciones diferenciales cada vez es mas importante ayudarnos de métodos que nos facilita la comprensión de estas, En el contexto de este proyecto, nosotros como estudiantes de una ingeniería , quisimos hacer una investigación en relación a los diferentes análisis de temas aprendidos en la materia de métodos numéricos, en este caso en este proyecto se adentran en el método de diferencias finitas en relación a ecuaciones diferenciales de Laplace, que mediante valores dados fronteras hacemos un análisis en relación a diferentes errores y valores de diferentes graficas. Este método permite abordar elementos característicos de las ecuaciones diferenciales parciales (EDP), especialmente como ya se hablo acerca de problemas con valores en la frontera y derivadas parciales, de ante mano se quiso hacer una evaluación frente a dos respectivos códigos ya que quisimos trabajar de forma estructurada la relación de las diferentes gráficas y errores de cada uno de los códigos compartidos en este proyecto."

# **Objetivos**

**Objetivo General**: Lograr estudiar y evaluar el método de diferencias finitas para dar solucion a EDP de Laplace de forma numérica

**Objetivos Específicos:**

* Reconocer patrones necesarios para resolver una ecuación diferencial parcial utilizando el método de diferencias finitas, lo que facilita la detección de posibles errores y el calculo de las diferencias finitas en estos.
* Contrastar la solución numérica obtenida con la solución analítica, para así verificar la precisión del método de diferencias finitas a través del error puntual que se da en el código evaluado.
* Emplear el lenguaje de programación Python para implementar los conocimientos adquiridos en la resolución de la ecuación diferencial parcial de Laplace mediante el método de diferencias finitas, para asi lograr el entendimiento de estas.

# **Justificación**

En relación con este proyecto evaluado, debemos tener en cuenta un aprendizaje profundo en relación con los comportamientos de la ED, en el proyecto en cuestión evaluamos los diferentes casos con el método de diferencias finitas. El método de las diferencias finitas implica discretizar una ecuación diferencial, es decir, convertirla en una forma que puede ser resuelta mediante cálculos numéricos en una computadora. En lugar de trabajar con derivadas continuas, el método de diferencias finitas reemplaza estas derivadas por aproximaciones basadas en diferencias entre valores discretos de la función. Por lo tanto, este método es de gran importancia cuando se quiere estudiar y evaluar diferentes dimensiones, es importante entender que este método es importante trabajarlo ya que presenta una Versatilidad y aplicación bastante extensa, Es ampliamente aplicable en campos como la ingeniería, la física, la economía y la biología, entre otros. Permite la resolución de problemas complejos que no tienen soluciones analíticas simples, como la propagación del calor, la dinámica de fluidos, y la mecánica cuántica. En este sentido vemos que es importante estudiarlo y comprenderlo ya que nos da la posibilidad de evaluar problemas del mundo real, Muchas ecuaciones diferenciales que modelan fenómenos del mundo real no tienen soluciones analíticas. El método de diferencias finitas permite obtener soluciones aproximadas, lo cual es crucial para la práctica en ingeniería y las ciencias aplicadas.

Sin embargo, para un buen desarrollo de este método, se debe trabajar con un buen soporte computacional en donde cada uno de los procedimientos algebraicos sean utilizados de forma correcta, para asi poder evaluar los valores de los resultados correctos, en este proyecto propiamente se quiso desarrollar una solución analítica de las EDP para hacer una correlación entre cada una de las soluciones, Por tal motivo la finalidad de este proyecto de estudio es evaluar de forma correcta diferentes soluciones que se dan el método de diferencias finitas con valores en frontera que permita identificar un análisis propio en relación a las diferentes soluciones de los casos de las ecuaciones de Laplace.

# **Marco Conceptual**

En la evaluación y desarrollo de este proyecto se pensó que era importa trabajar con el método de diferencias finitas ya que las ecuaciones de Laplace suelen venir acompañadas de condiciones de frontera específicas, y el método de diferencias finitas puede adaptarse fácilmente para incorporar estas condiciones. Esto es crucial para obtener soluciones precisas en dominios con geometrías y límites complejos. Por ende, el método de diferencias finitas es capaz de manejar una amplia variedad de problemas que involucran ecuaciones de Laplace. Esto incluye tanto problemas estáticos como dinámicos, permitiendo la solución de casos con diversas configuraciones y parámetros.

Asi mismo con la uso de herramientas como Python, es posible no solo calcular soluciones numéricas sino también generar visualizaciones detalladas. Estas visualizaciones son fundamentales para la interpretación y comprensión de los resultados, especialmente en proyectos donde la representación gráfica de la solución es crítica para el análisis. En algunos casos, es posible tener soluciones analíticas de las ecuaciones de Laplace. Utilizar el método de diferencias finitas permite comparar las soluciones numéricas con las analíticas, validando así la precisión y fiabilidad del enfoque numérico.

# **Desarrollo**

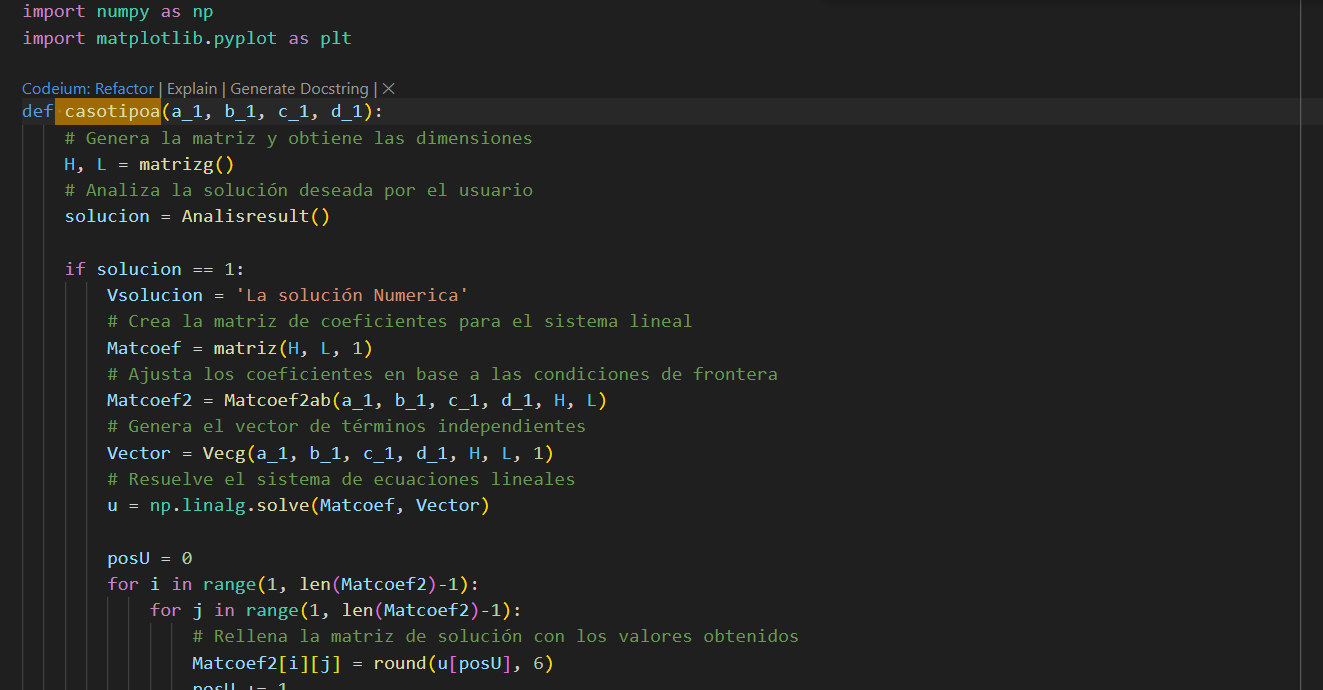
Programa1:

En el primer programa que tuvimos en cuenta para trabajar es donde manejamos los siguientes conceptos

**Explicación detallada:**

Función casotipoa:

Entrada: Recibe las condiciones de frontera a\_1, b\_1, c\_1, d\_1.



Proceso:

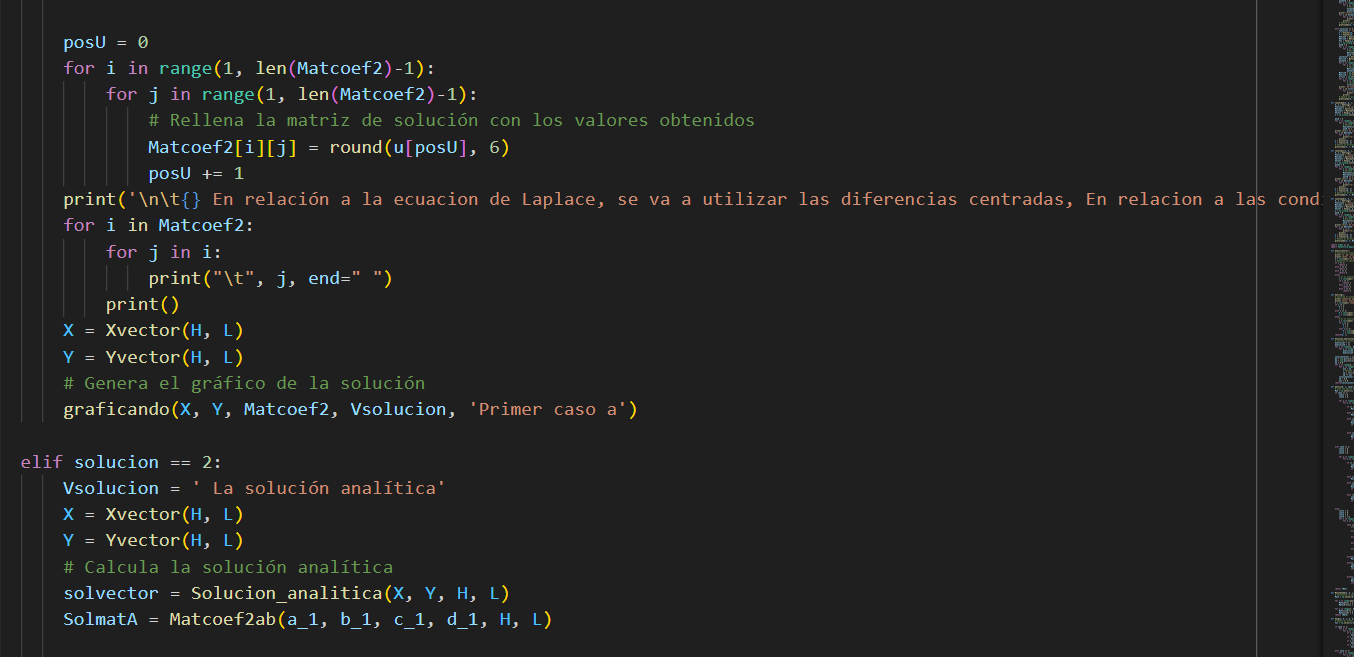
Genera la matriz y obtiene sus dimensiones.

Determina qué solución desea el usuario (numérica, analítica o error relativo).

Dependiendo de la selección, calcula la solución usando métodos numéricos o analíticos.

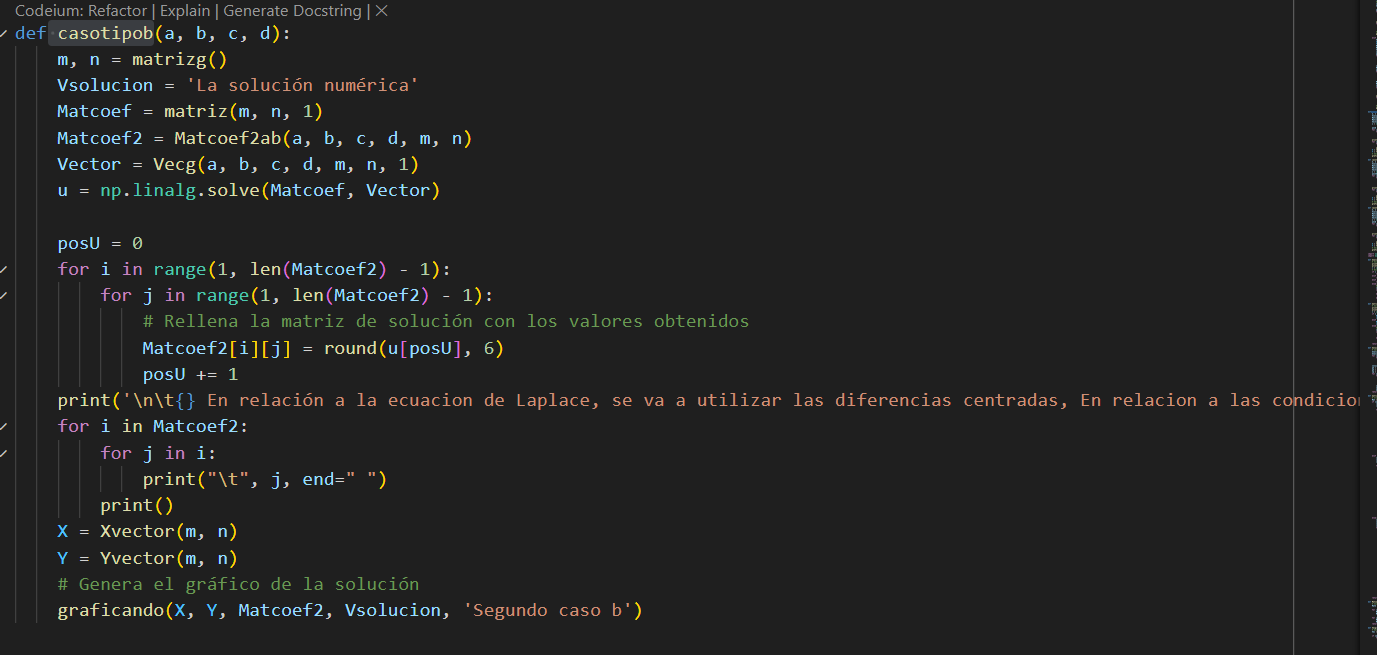
Imprime la matriz de solución.

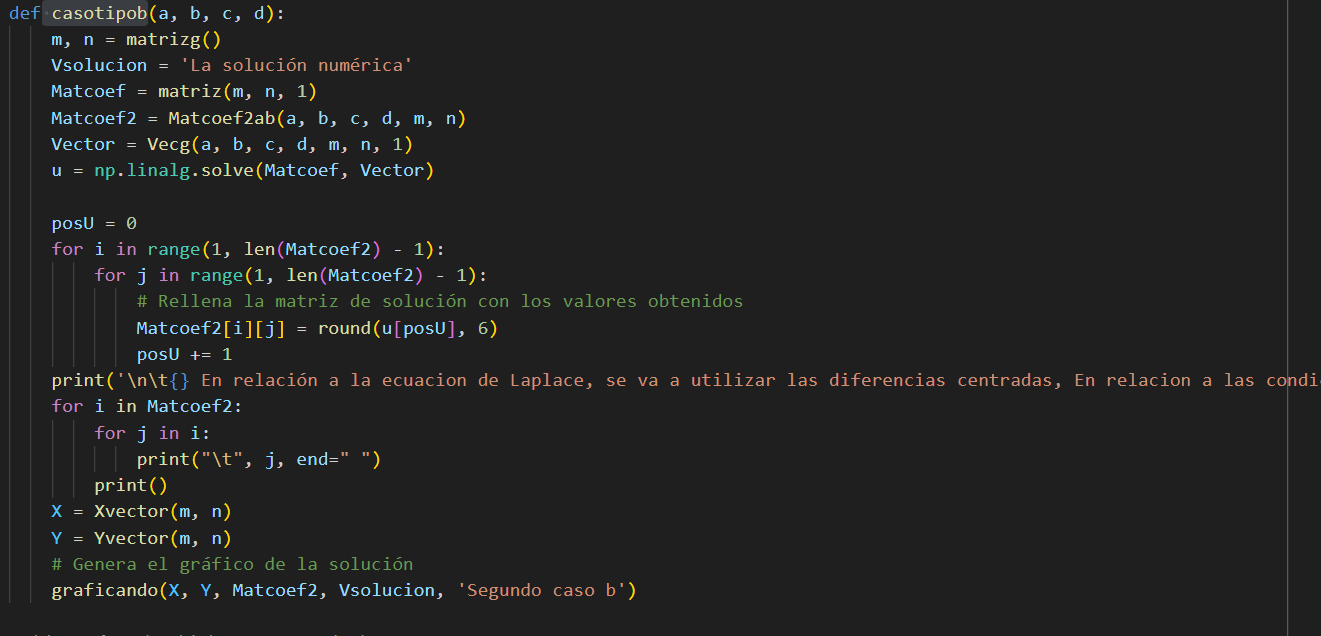
Genera y muestra un gráfico de la solución.

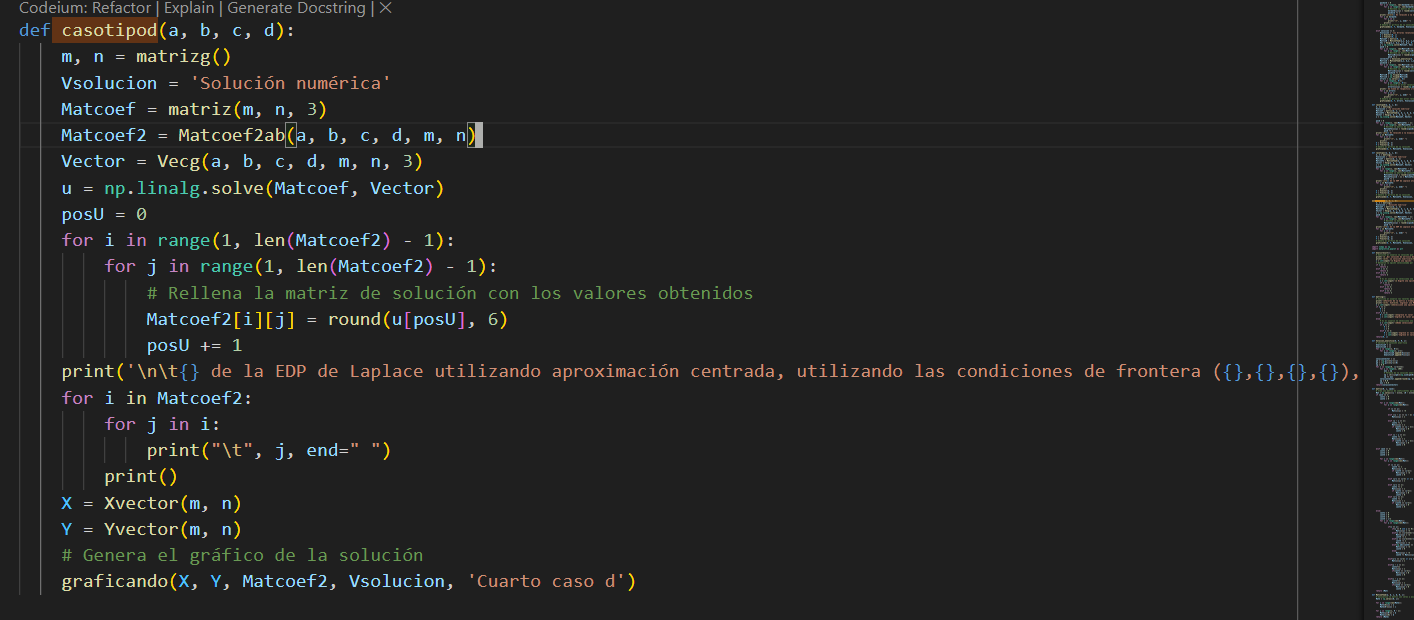


Funciones casotipob, casotipoc, casotipod:

Entrada: Reciben las condiciones de frontera a, b, c, d.







Proceso:

Generan la matriz y obtienen sus dimensiones.

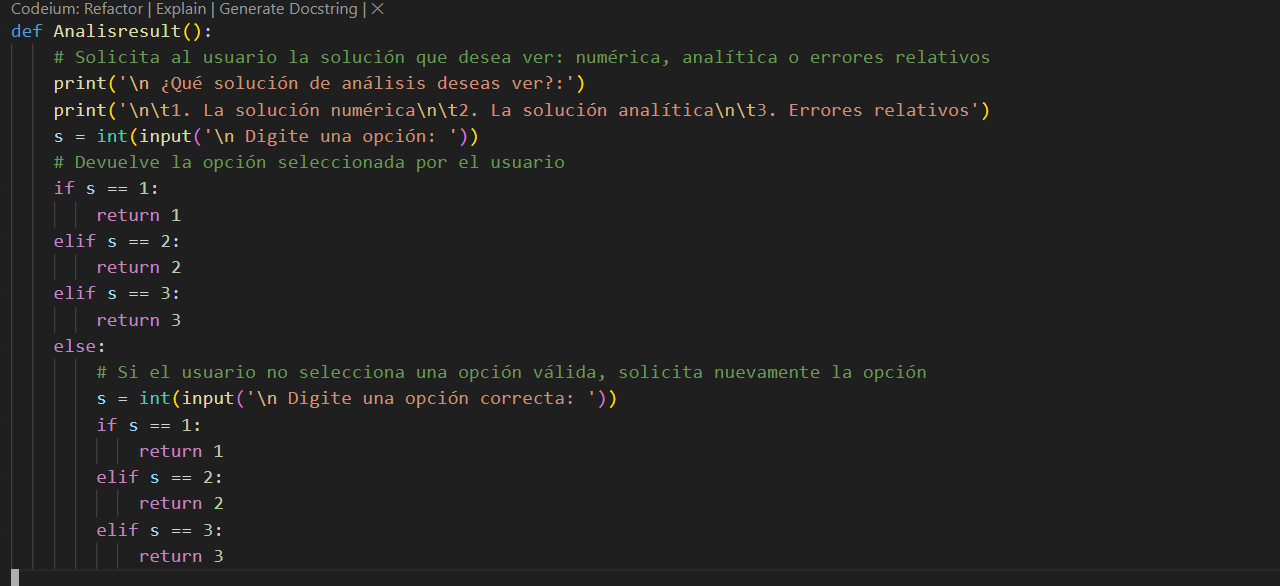
Ajustan los coeficientes en base a las condiciones de frontera.

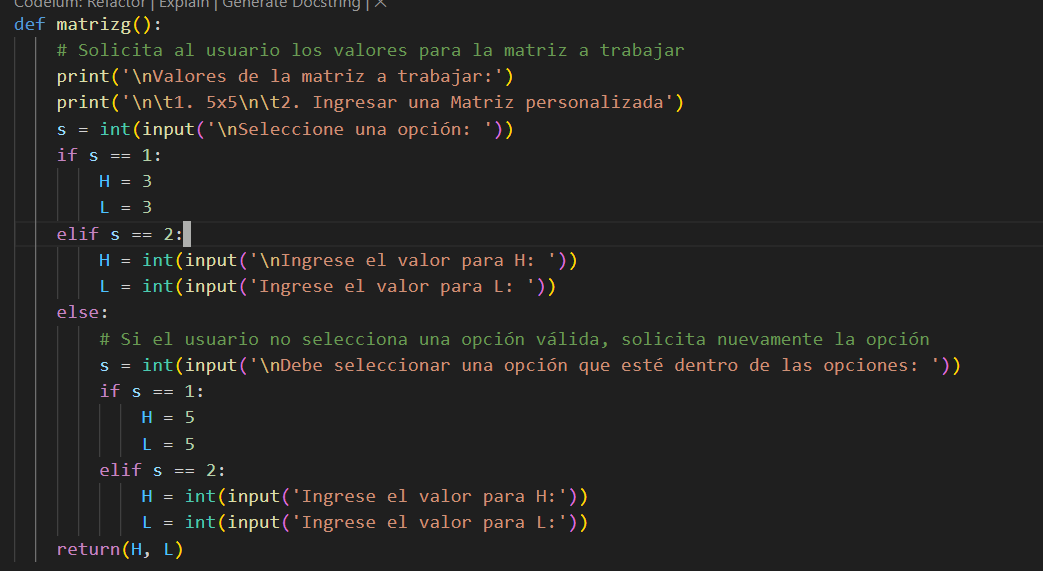
Calculan la solución numérica usando el método de diferencias centradas.

Imprimen la matriz de solución.

Generan y muestran un gráfico de la solución.

Las funciones auxiliares como matrizg, Analisresult, matriz, Matcoef2ab, Vecg, Xvector, Yvector, Solucion\_analitica, y graficando son responsables de generar matrices, analizar resultados, calcular soluciones y graficarlas.

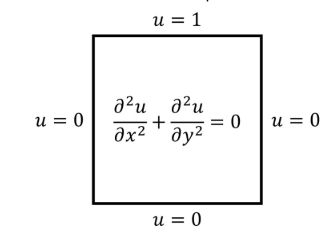


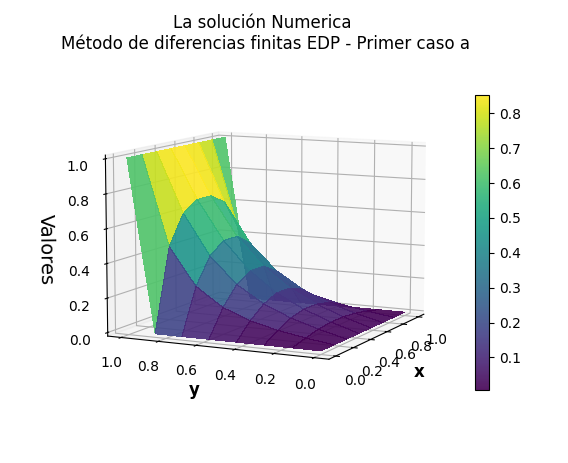


Cada sección del código está bien comentada para que puedas entender qué hace cada parte y cómo se relaciona con el cálculo y la representación gráfica de las soluciones a la ecuación de Laplace bajo diferentes condiciones de frontera.

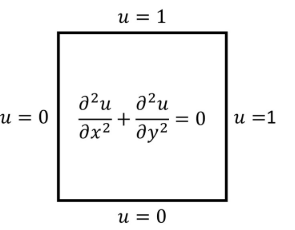
EVALUACION DE LOS DIFERENTES CASOS:

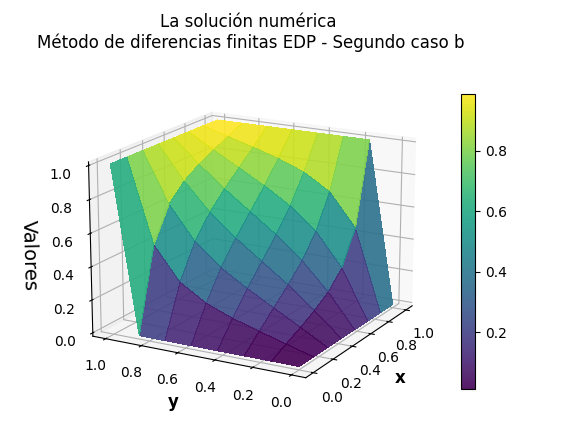
Caso Tipo a:



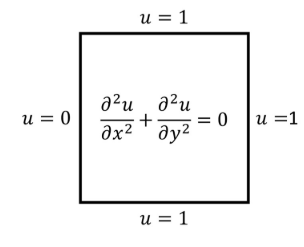


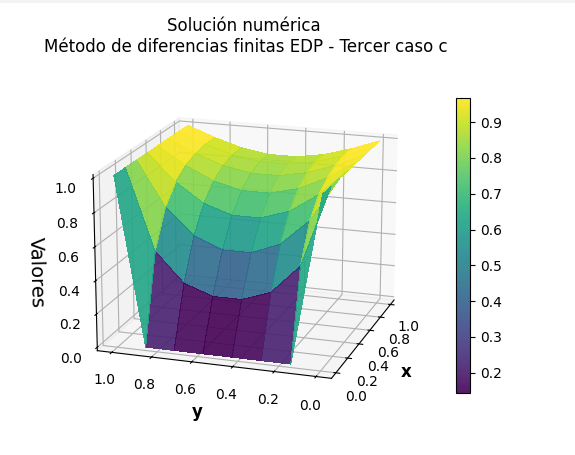
Caso tipo b:



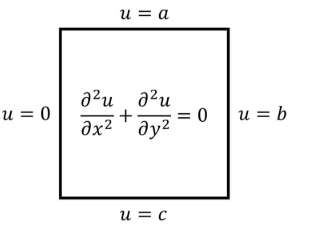


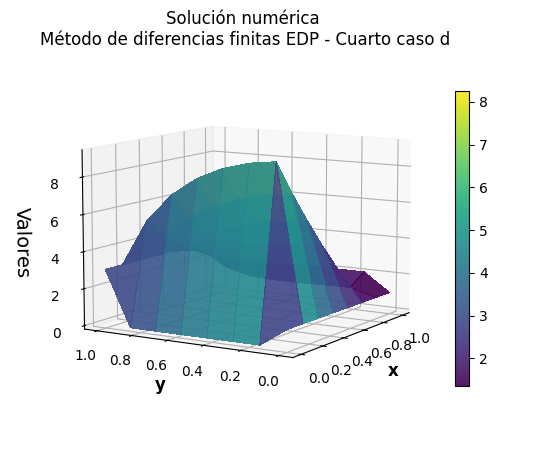
Caso Tipo c:





Caso Tipo d:





Programa 2:

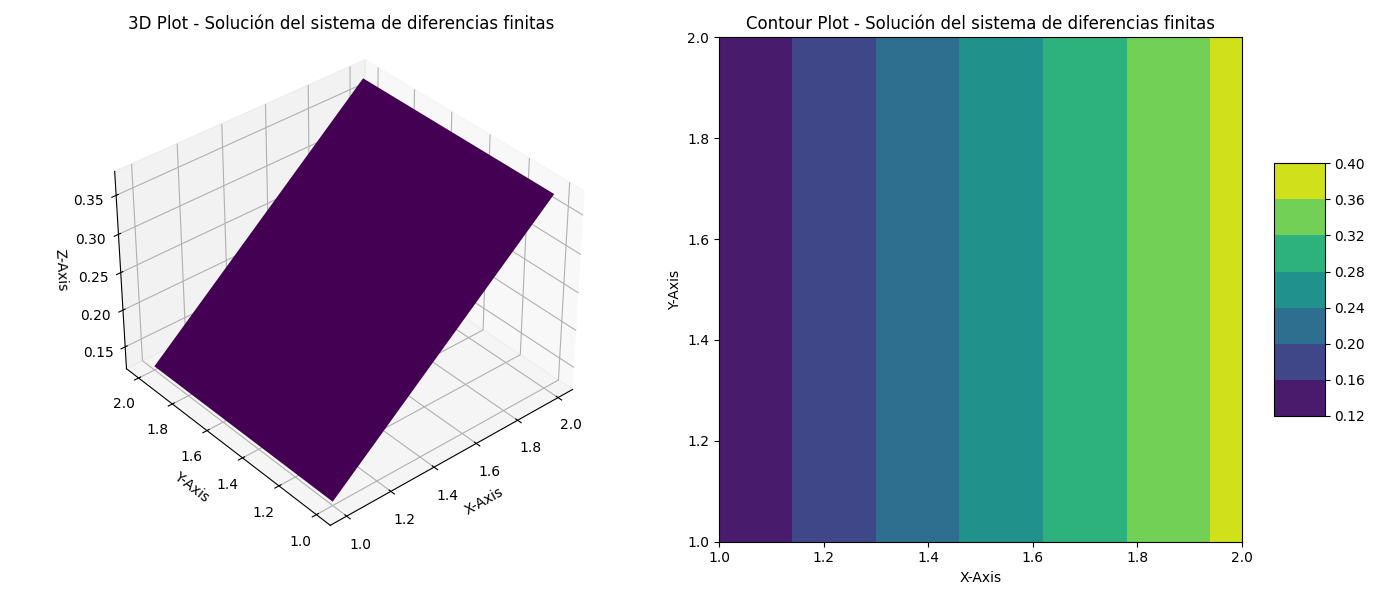
se utiliza para definir las constantes para las condiciones de frontera (Inferior, Superior, Derecha, Izquierda), también hay una función que construye la matriz de coeficientes y el vector solución. Finalmente está el archivo ‘errores\_graficas.py’ en este último modulo, lo utilizamos para graficar cada uno de los métodos desarrollados en una gráfica 3D, junto a su respectivo error local y error global.

* Código Menú Principal



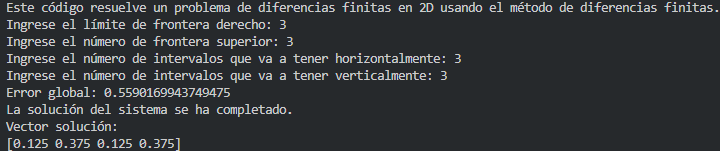
**Ilustración 1.1** - Código del Menú Principal (Python)

* Código Puntos  
    
  
* **Ilustración 1.1.1** – Código del módulo puntos (Python)
* Código Matriz Vector  
    
  
* **Ilustración 1.1.2** – Código del módulo Matriz Vector (Python)
* Código Errores Graficas  
    
  
* **Ilustración 1.1.3** – Código del módulo Errores Graficas (Python)
* Caso a
* Solución numérica



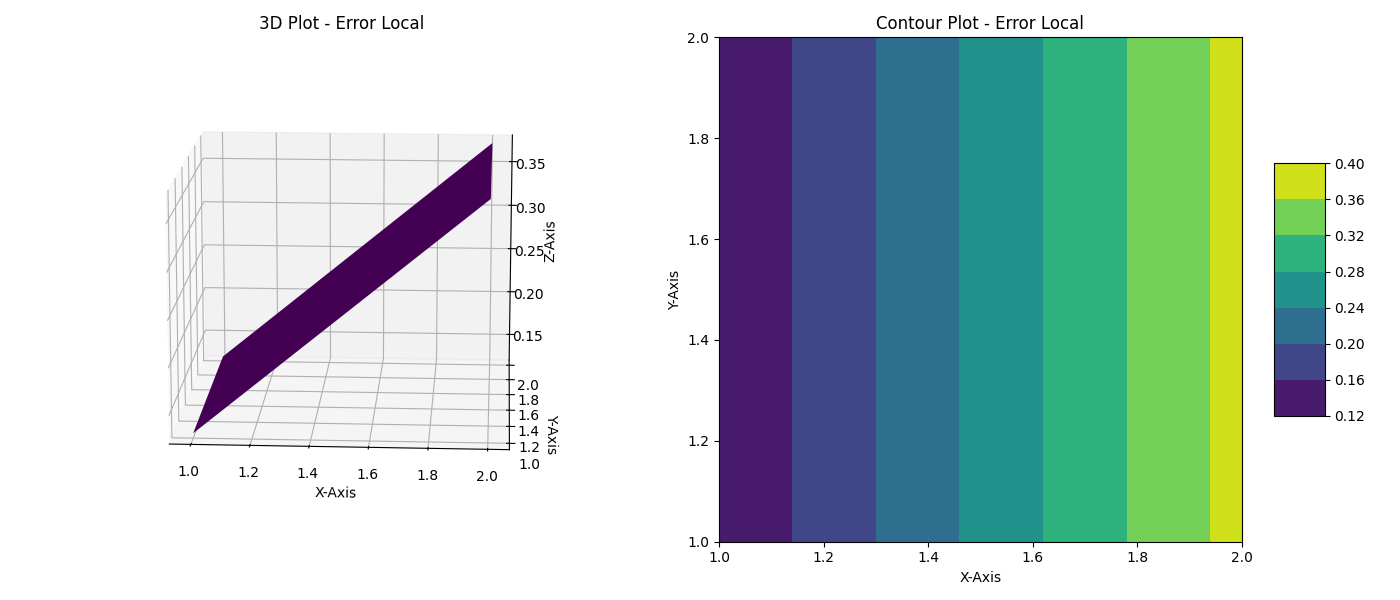
**Ilustración 1.2.1** - Caso a (Python)

* Solución analítica

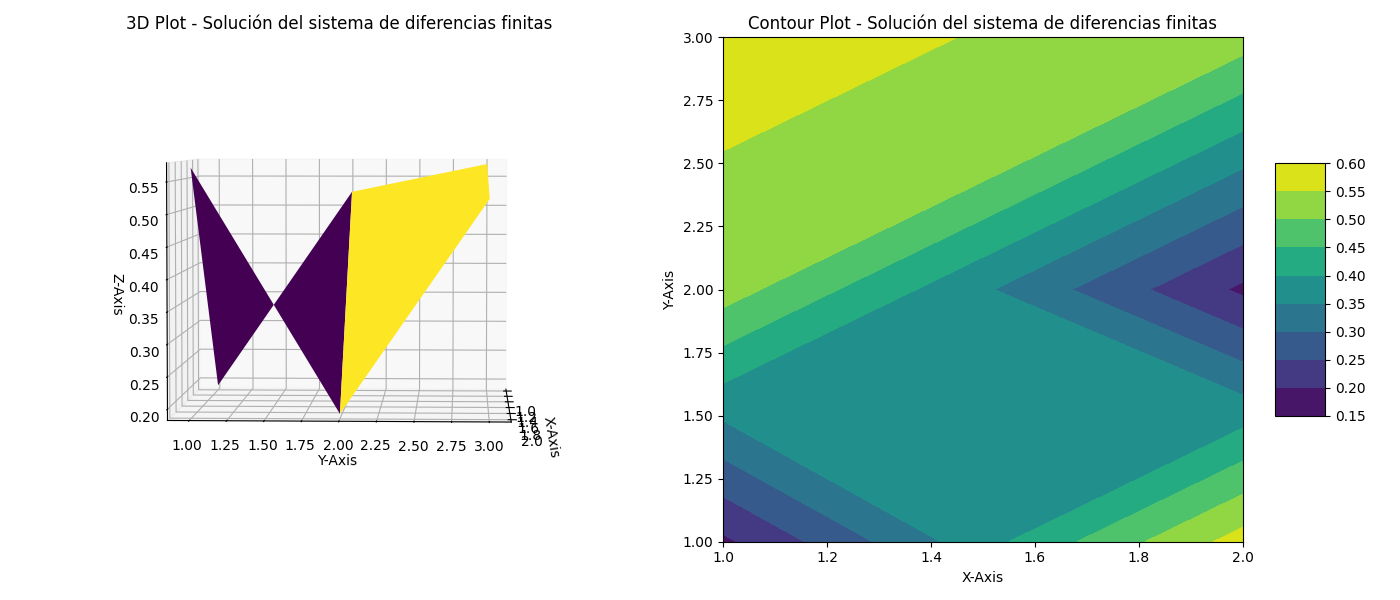
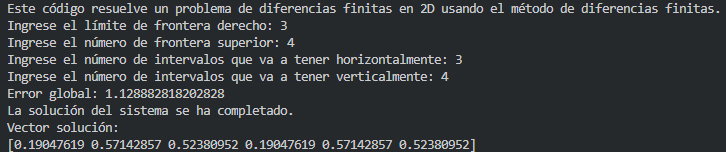


**Ilustración 1.2.2** - Solución Analítica (Python)

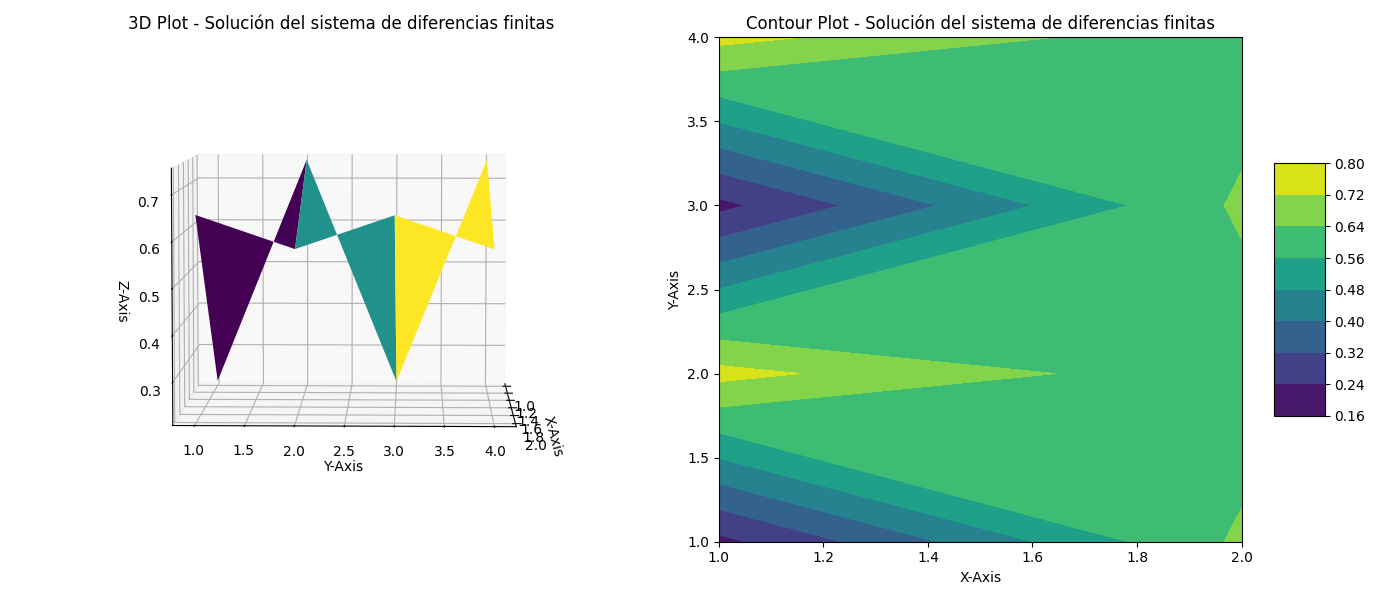
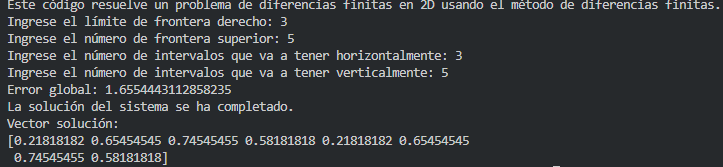
* Error puntual



**Ilustración 1.2.3** - Error Puntual (Python)

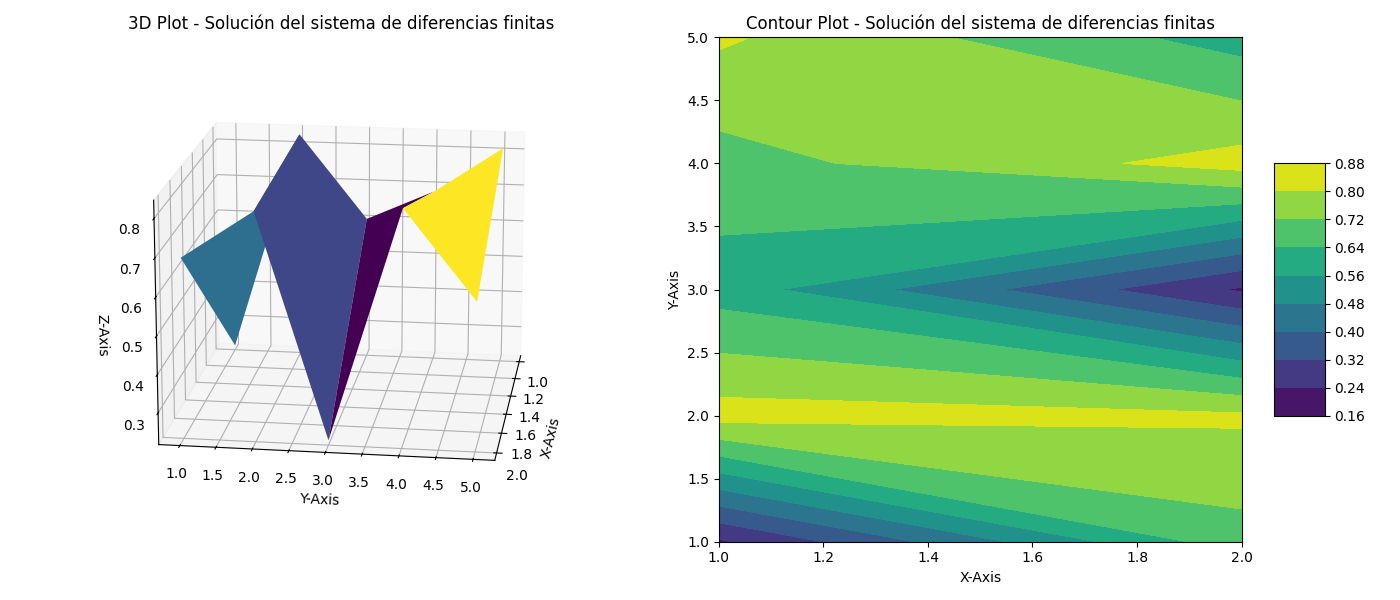
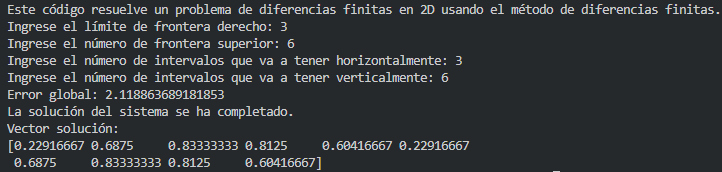
* caso b  
    
    
    
  

**Ilustración 1.3** - Caso b (Python)

* caso c  
    
    
  

**Ilustración 1.4** - Caso c (Python)

* caso d

**Ilustración 1.5** - Caso d (Python)

Análisis de Resultados:

Las siguientes respuestas las estaremos relacionado en la correlación de los diferentes datos, presentados en cada uno de los dos códigos presentados de manera que se tendrá en cuenta la relación entre el programa de proyecto de evaluación 2

**¿Cuáles son las incógnitas del PVF a solucionar?**

En el contexto del problema de valor en la frontera (PVF) para la ecuación de Laplace, las incógnitas son los valores de la función 𝑢 (𝑥, 𝑦) en el dominio dado 0 < 𝑥 < 𝐿 y 0 < y < H. La cantidad de incógnitas depende del número de puntos en los que se discretiza el dominio.

Si discretizamos el dominio en una malla con 𝑀 nodos en la dirección 𝑥 y 𝑁 nodos en la dirección 𝑦, excluyendo los nodos en la frontera, las incógnitas serán los valores de 𝑢 en los nodos internos de esta malla. el problema de valores de frontera (PVF) a solucionar son los valores de la función en los nodos interiores del dominio, es decir, los valores de 𝑢 en la malla que no están en los bordes. Estas incógnitas se calculan utilizando el método de diferencias finitas.

**¿Cuántas incógnitas contiene el problema?**

El número de incógnitas es el número de nodos internos en la malla. Si hay 𝑛𝑥, nx intervalos horizontales y 𝑛𝑦, ny intervalos verticales, entonces hay (𝑛𝑥−1)×(𝑛𝑦−1)

(nx−1)×(ny−1) incógnitas, ya que solo los nodos interiores son las incógnitas.

**¿Cuántos nodos internos define para hallar la solución del PVF?**

El número de nodos internos depende de cómo se discretiza el dominio. Si usamos una malla de 𝑀 × 𝑁 nodos, donde 𝑀 y 𝑁 incluyen los nodos de frontera, los nodos internos serán aquellos que no están en la frontera.

Supongamos que tenemos una malla de 𝑀 nodos en la dirección 𝑥 y 𝑁 nodos en la dirección 𝑦:

Los nodos en la dirección 𝑥 serían 𝑥 1 , 𝑥 2 , … , 𝑥 𝑀.

Los nodos en la dirección 𝑦 serían 𝑦 1 , 𝑦 2 , … , 𝑦 𝑁.

Excluyendo los nodos de la frontera, los nodos internos serán:

(𝑀 − 2) × (𝑁 − 2)

Este cálculo excluye los nodos en 𝑥 = 0, 𝑥 = 𝐿 y 𝑦 = 0, 𝑦 = 𝐻.

¿Cuántos nodos de frontera define para hallar la solución del PVF?

Los nodos de frontera son aquellos situados en los bordes del dominio. Su número también depende de cómo se discretiza el dominio.

Para una malla de 𝑀 × 𝑁 nodos:

En 𝑥 = 0 y 𝑥 = 𝐿 tenemos 𝑁 nodos cada uno.

En 𝑦 = 0 y 𝑦 = 𝐻 tenemos 𝑀 nodos cada uno.

Sin embargo, al contar los nodos de las esquinas dos veces, debemos ajustar el conteo:

Total, de nodos de frontera = 2 (𝑁 − 2) + 2 (𝑀 − 2) + 4

Total, de nodos de frontera = 2 𝑁 + 2 𝑀 − 4

Esto asegura que no estamos contando dos veces los nodos en las esquinas.

# **Conclusiones**

El método de diferencias finitas es una herramienta potente y flexible para resolver ecuaciones diferenciales de Laplace con diversas condiciones de frontera. Su capacidad para manejar condiciones complejas, ofrecer soluciones precisas y permitir una eficiente implementación computacional lo convierte en una opción ideal para proyectos que involucran múltiples casos y escenarios. Además, la capacidad de visualizar y validar resultados incrementa la comprensión y confianza en las soluciones obtenidas, haciendo de este método una elección crucial para el éxito del proyecto.

Los procesos iterativos en cierto punto se vuelven extensos y tediosos resolverlos paso a paso de forma manual, por lo que la programación se vuelve el mejor aliado a la hora resolver este tipo de problemas y si la herramienta esta bien desarrollada la probabilidad de fallo es muy baja.

El uso de este programa computacional es escalable ya que se puede ampliar para usarse en campos de mayor importancia como podrían ser temperaturas a gran escala y posicionamiento de placas en un espacio, siendo una gran herramienta para físicos e ingenieros de este campo.

# **Referencias**

* Cheney, W. and Kincaid, D (2013). Numerical mathematics and computing.Texas:Cengage Learning.
* Chapra, S. C., and Canale, R. P. (2011). Métodos numéricos para ingenieros. Mexico: McGraw-Hill.
* Kiusalaas, J (2014). Numerical methods in engineering with Python 3. Cambridge:Cambridge University Press.
* Burden, R. L., and Faires, J. D. (2011). Análisis numérico. Mexico: Grupo Editorial Iberoamérica.