Diferencias Finitas

Comparación entra Fortran, Python y Solución Analítica

Julian Avila, Camilo Huertas, David Rodríguez, Juan Acuña

11 de mayo de 2025

Se solucionaron distintas Ecuaciones de Poisson con distintas condiciones de frontera. Se comparó el tiempo de ejecución para solucionar numéricamente por medio de diferencias finitas usando Python y Fortran. Se emplearon mallas de 100x100 con una tolerancia de 10^{-7}.

Problema 1

Ecuación diferencial:

$$\nabla^2 V = \left(x^2 + y^2\right) e^{xy} \tag{1}$$

Condiciones de frontera:

$$V(x,y) = \begin{cases} V(0,y) = 1 & V(2,y) = e^{2y} \\ V(x,0) = 1 & V(x,1) = e^{x} \end{cases} \tag{2}$$

Solución Analítica:

$$V = e^{xy} \tag{3}$$

Resultados

Lenguaje	Tiempo (s)
Fortran	0,686
Python	91,495

Cuadro 1: Tiempo de ejecución ecuación (1).

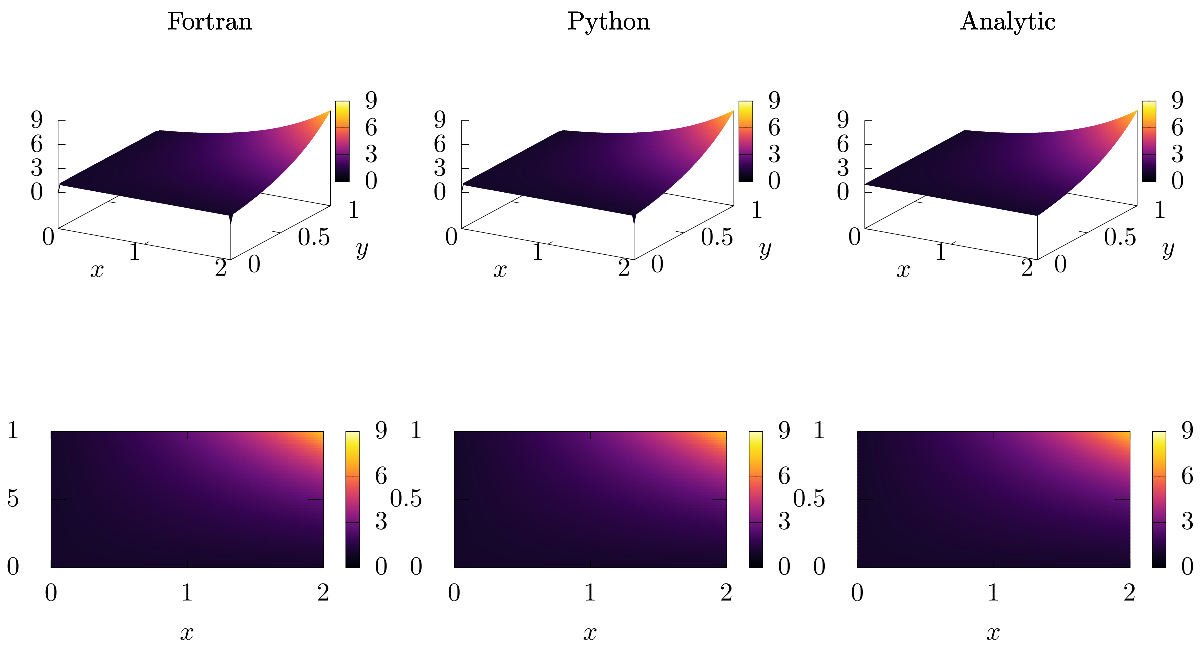

Figura 1: Solución de la ecuación (1)

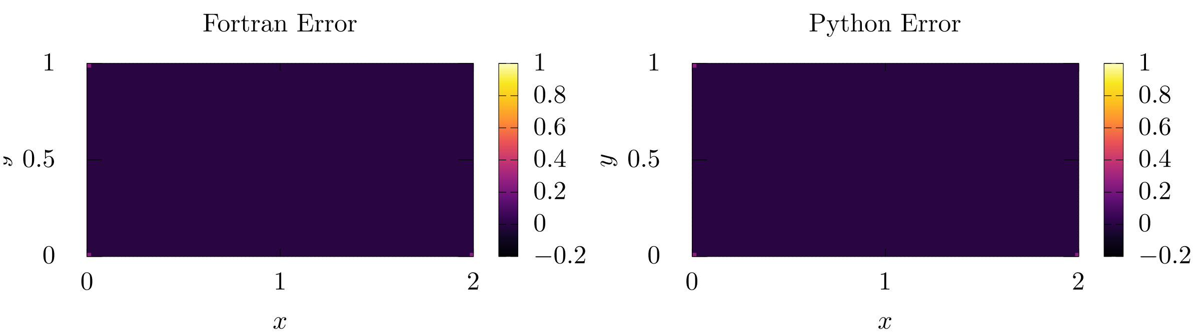

Figura 2: Error de la solución

Se observa un error más elevado en las fronteras de la malla.

Problema 2

Ecuación diferencial:

$$\nabla^2 V = 0 \tag{4}$$

Condiciones de frontera:

$$V(x,y) = \begin{cases} V(1,y) = \ln\left(y^2+1\right) & V(2,y) = \ln\left(y^2+4\right) \\ V(x,0) = 2\ln x & V(x,1) = \ln\left(x^2+1\right) \end{cases} \tag{5}$$

Solución Analítica:

$$V = \ln\left(y^2 + x^2\right) \tag{6}$$

Resultados

Lenguaje	Tiempo (s)
Fortran	1,286
Python	93,352

Cuadro 2: Tiempo de ejecución ecuación (4).

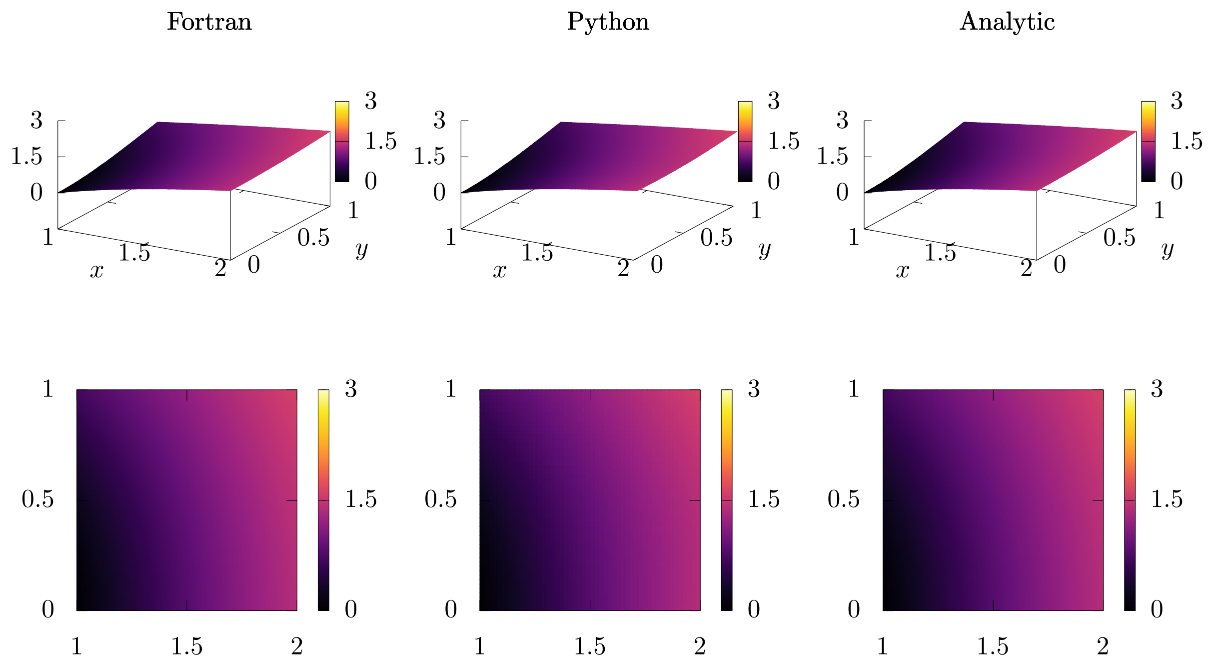

Figura 3: Solución de la ecuación (4)

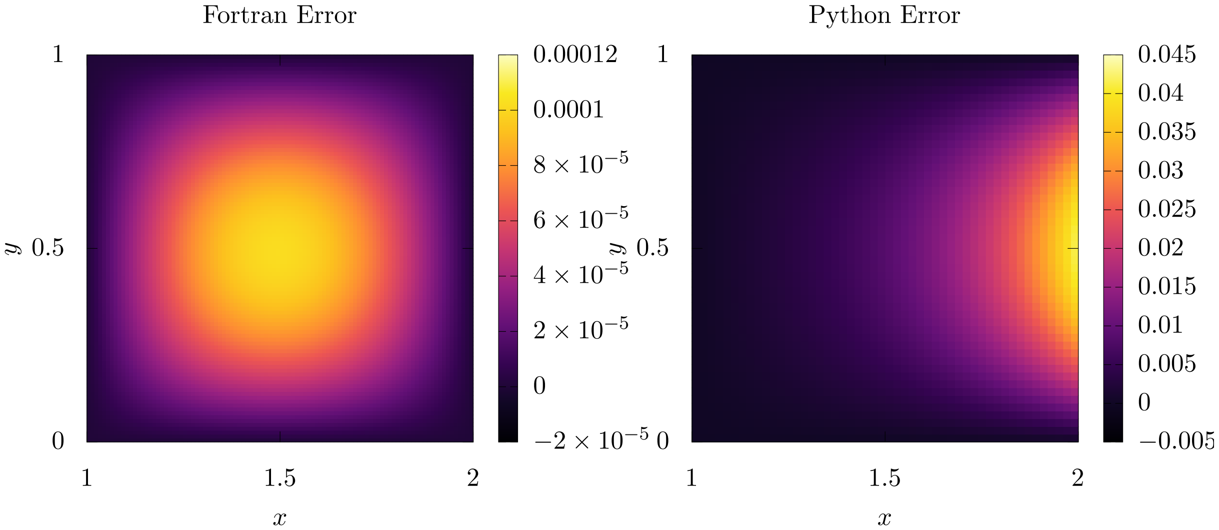

Figura 4: Error de la solución

La solución obtenida con Fortran presenta un error menor en dos órdenes de magnitud en comparación con la solución de Python. En cuanto a la distribución espacial del error, la solución de Fortran muestra los mayores errores en el centro de la malla, los cuales disminuyen hacia los bordes, mientras que la de Python presenta un error máximo en la frontera derecha, disminuyendo progresivamente hacia la izquierda.

Problema 3

Ecuación diferencial:

$$\nabla^2 V = 4 \tag{7}$$

Condiciones de frontera:

$$V(x,y) = \begin{cases} V(0,y) = y^2 & V(2,y) = (y-2)^2 \\ V(x,0) = x^2 & V(x,2) = (x-2)^2 \end{cases} \tag{8}$$

Solución Analítica:

$$V = (x-y)^2 \tag{9}$$

Resultados

Lenguaje	Tiempo (s)
Fortran	4,577
Python	79,064

Cuadro 3: Tiempo de ejecución ecuación (7).

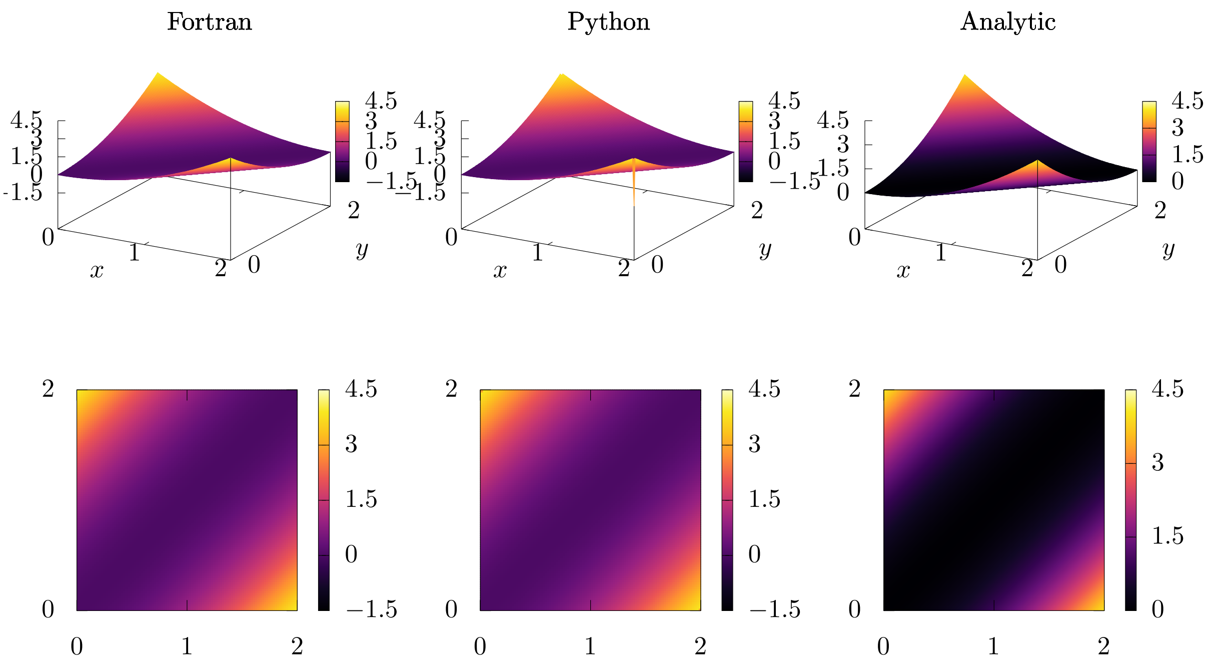

Figura 5: Solución de la ecuación (7)

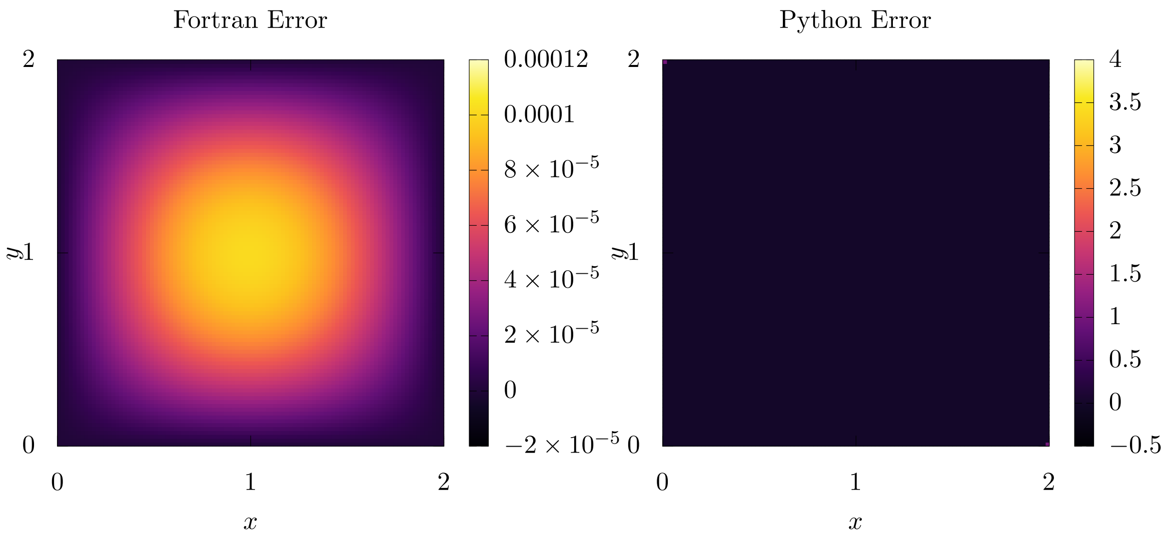

Figura 6: Error de la solución

La solución de Fortran, nuevamente con un error menor, muestra una distribución donde este disminuye radialmente a partir del centro de la malla. En contraste, la solución de Python exhibe errores significativos en las fronteras de la malla.

Problema 4

Ecuación diferencial:

$$\nabla^2 V = \frac{x}{y} + \frac{y}{x} \tag{10}$$

Condiciones de frontera:

$$V(x,y) = \begin{cases} V(1,y) = y\ln y & V(2,y) = 2y\ln 2y \\ V(x,1) = x\ln x & V(x,2) = 2x\ln 2x \end{cases} \tag{11}$$

Solución Analítica:

$$V = xy\ln xy \tag{12}$$

Resultados

Lenguaje	Tiempo (s)
Fortran	1,478
Python	84,873

Cuadro 4: Tiempo de ejecución ecuación (10).

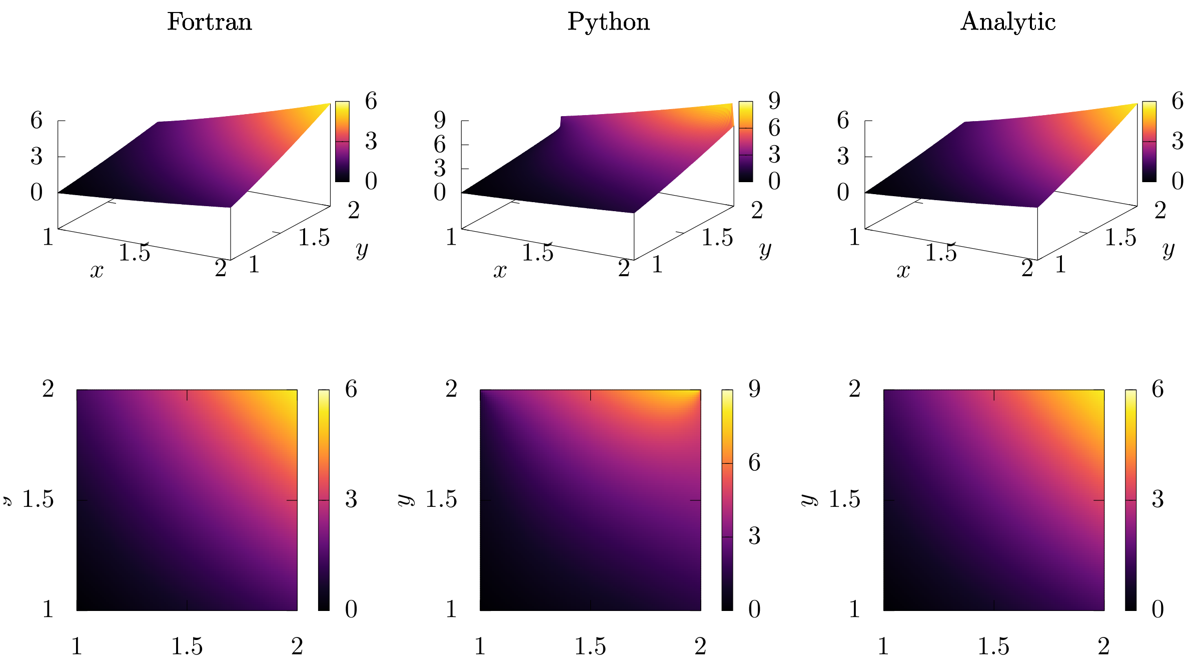

Figura 7: Solución de la ecuación (10)

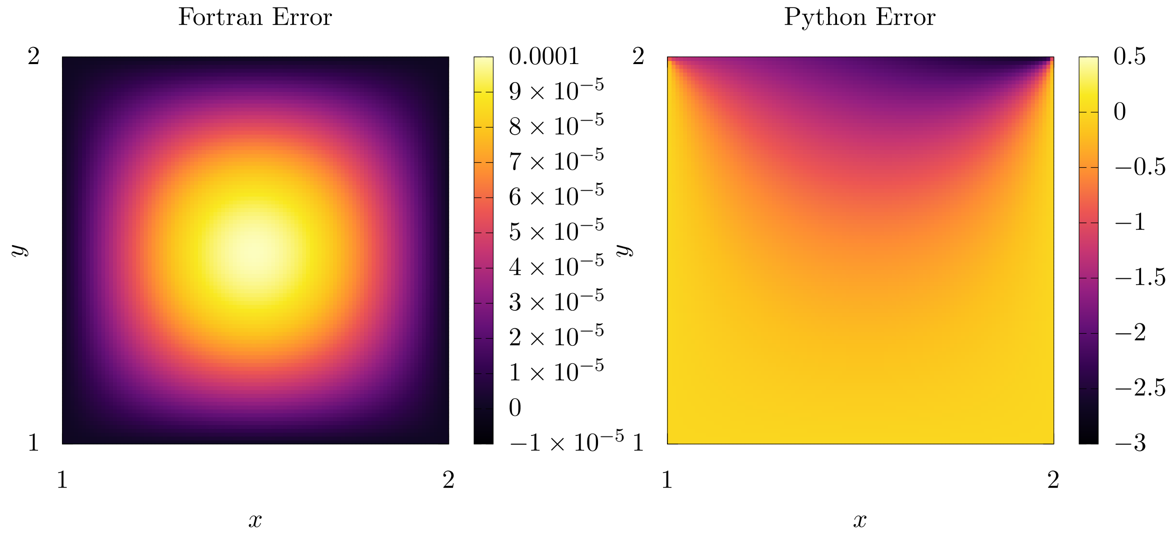

Figura 8: Error de la solución

La solución de Fortran, nuevamente con un error menor, muestra una distribución donde este disminuye radialmente a partir del centro de la malla. En contraste, la solución de Python exhibe errores significativos en la frontera superior, hay cuatro órdenes de magnitud de diferencia entre los errores de los métodos.

1. Tiempos

Para la resolución de todas las Ecuaciones se emplearon las mismas condiciones en ambos códigos. La ejecución en Fortran resultó sistemáticamente más rápida que en Python, registrando diferencias que oscilaron entre 4 y 100 veces.