

Problema 1: Ecuación de Poisson

En este problema se pide resolver numéricamente la ecuación de Poisson en dos dimensiones

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = f(x, y)$$

con $f(x, y) = \cos(3x + 4y) - \cos(5x - 2y)$, en el intervalo $0 \leq x, y \leq 2\pi$ junto con las condiciones a la frontera

$$\begin{aligned}\phi(x, 0) &= \phi(x, 2\pi) \\ \phi(0, y) &= \phi(2\pi, y)\end{aligned}$$

Para resolver esta ecuación, es decir, aproximar numéricamente la solución $\phi(x, y)$ en el intervalo en cuestión, considérese el desarrollo de Taylor para $\phi(x, y)$ para pequeños cambios en x y y :

$$\phi(x + \Delta x, y) = \phi(x, y) + \frac{\partial \phi}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} (\Delta x)^2 + \dots$$

$$\phi(x - \Delta x, y) = \phi(x, y) - \frac{\partial \phi}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} (\Delta x)^2 + \dots$$

$$\phi(x, y + \Delta y) = \phi(x, y) + \frac{\partial \phi}{\partial y} \Delta y + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} (\Delta y)^2 + \dots$$

$$\phi(x, y - \Delta y) = \phi(x, y) - \frac{\partial \phi}{\partial y} \Delta y + \frac{1}{2} \frac{\partial^2 \phi}{\partial y^2} (\Delta y)^2 + \dots$$

De estas ecuaciones podemos aproximar las segundas derivadas parciales de ϕ sumando a pares sus respectivas ecuaciones, truncando la expansión en serie y despejando, obteniendo:

$$\frac{\partial^2 \phi}{\partial x^2} \approx \frac{1}{(\Delta x)^2} (\phi(x + \Delta x, y) + \phi(x - \Delta x, y) - 2\phi(x, y))$$

$$\frac{\partial^2 \phi}{\partial y^2} \approx \frac{1}{(\Delta y)^2} (\phi(x, y + \Delta y) + \phi(x, y - \Delta y) - 2\phi(x, y))$$

En este punto es importante mencionar que, a diferencia de otras formas de resolución de ecuaciones diferenciales parciales donde se puede ocupar el método de diferencias finitas, en este caso la solución $\phi(x, y)$ es independiente del tiempo, por lo que no es posible expresar las derivadas parciales a través de *forward difference* u otras aproximaciones.

El dominio de esta ecuación es totalmente dependiente del espacio, en este caso con $(x, y) \in [0, 2\pi]^2$. Para resolver la ecuación de Poisson aquí, se debe crear una *mall*a para $[0, 2\pi]^2$.

Sea N_x el número de puntos en los que se quiere dividir el intervalo $0 \leq x \leq 2\pi$, entonces se define $x_{min} = 0$, $x_{max} = 2\pi$ y $\Delta x = \frac{x_{max} - x_{min}}{N_x}$. Entonces los puntos discretizados x_i (separados entre sí por Δx) están dados por

$$x_i = x_{min} + i\Delta x, \quad i = 0, 1, \dots, N_x$$

Análogamente, para discretizar $0 \leq y \leq 2\pi$ se define el respectivo N_y , Δy para obtener los puntos

$$y_i = y_{min} + j\Delta y, \quad j = 0, 1, \dots, N_y$$

De esta forma, se puede adaptar la siguiente notación

$$\begin{aligned} \phi(x_i, y_i) &= \phi_{i,j}, & f(x_i, y_i) &= f_{i,j} \\ \phi(x_i + \Delta x, y_i) &= \phi_{i+1,j}, & \phi(x_i - \Delta x, y_i) &= \phi_{i-1,j} \\ \phi(x_i, y_i + \Delta y) &= \phi_{i,j+1}, & \phi(x_i, y_i - \Delta y) &= \phi_{i,j-1} \end{aligned}$$

Así, juntando las aproximaciones de las derivadas parciales y la discretización del espacio, se puede expresar la ecuación de Poisson a resolver como

$$\begin{aligned} \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} &= f(x, y) \\ \frac{1}{(\Delta x)^2} (\phi(x + \Delta x, y) + \phi(x - \Delta x, y) - 2\phi(x, y)) \\ &+ \frac{1}{(\Delta y)^2} (\phi(x, y + \Delta y) + \phi(x, y - \Delta y) - 2\phi(x, y)) = f(x, y) \\ \frac{1}{(\Delta x)^2} (\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_{i,j}) + \frac{1}{(\Delta y)^2} (\phi_{i,j+1} + \phi_{i,j-1} - 2\phi_{i,j}) &= f_{i,j} \end{aligned}$$

es posible elegir en este caso que $N_x = N_y$ para hacer que $\Delta x = \Delta y = \Delta$, con la finalidad de simplificar aún más la expresión, así:

$$\begin{aligned} \frac{1}{\Delta^2} (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}) &= f_{i,j} \\ \phi_{i,j} &= \frac{1}{4} (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - f_{i,j}\Delta^2) \end{aligned}$$

que es una relación de la solución en (x_i, y_i) en función (de hecho, el promedio) de otros valores solución vecinos y la función f en dicho punto (ver Figura 1).

Los métodos de solución que utilizan el resultado obtenido son aproximados y se basan en la convergencia (o no) de los valores iterados a partir de una propuesta arbitraria inicial a la solución de la ecuación de Poisson. Se distinguen dos métodos distintos:

- **Método de Jacobi**

Este método se basa en la aplicación sucesiva de la expresión de $\phi_{i,j}$ para aproximar el valor de la solución en dicho punto de la siguiente manera:

- Se parte de una serie de valores $\phi_{i,j}$ arbitrarios junto con las condiciones de frontera elegidas, es decir, se parte de una matriz de soluciones inicial.
- Se calculan nuevos valores $\phi_{i,j}$ (que constituyen una nueva aproximación) **a partir de exclusivamente valores de $\phi_{i,j}$ de la aproximación anterior**, creando una nueva matriz de soluciones.

- Se puede medir la diferencia entre las dos aproximaciones y decidir si seguir calculando; si se procede, la última aproximación pasa a ser la “anterior” y se repite el proceso.
- **Método de Gauss-Seidel**
 Este método también se basa en la aplicación sucesiva de la expresión para $\phi_{i,j}$ pero se distingue del método de Jacobi por utilizar los nuevos valores calculados durante la ejecución, mezclando aproximaciones realizadas anteriormente. Esto es:
 - Se parte de una serie de valores $\phi_{i,j}$ arbitrarios junto con las condiciones de frontera elegidas, es decir, se parte de una matriz de soluciones inicial.
 - Se calculan nuevos valores $\phi_{i,j}$ (que constituyen una nueva aproximación) **a partir de valores de $\phi_{i,j}$ disponibles, es decir, que pueden ser de la aproximación anterior o bien haber sido calculados durante esta aproximación**, creando una nueva matriz de soluciones.
 - Se puede medir la diferencia entre las dos aproximaciones y decidir si seguir calculando; si se procede, la última aproximación pasa a ser la “anterior” y se repite el proceso.

En términos computacionales, la implementación del método de Jacobi requiere al menos de dos matrices de soluciones en todo momento, mientras que el método de Gauss-Seidel solo requiere una, lo cual representa una diferencia en el uso de memoria. Además, Gauss-Seidel puede incrementar la velocidad de convergencia de las aproximaciones calculadas a la solución, lo cual ayuda en evitar errores por redondeo, producto de muchas iteraciones (Jacobi).

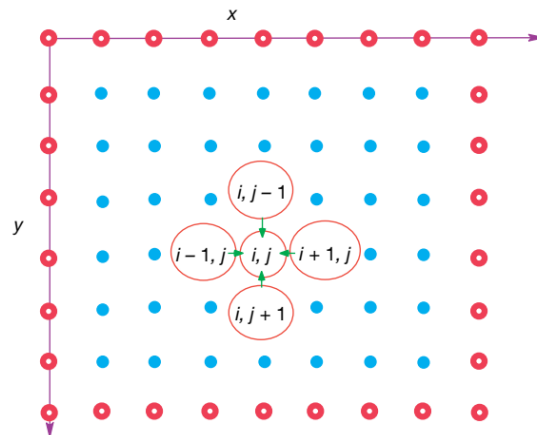


Figura 1. El dominio $[0, 2\pi]^2$ se discretiza en una malla de N_x puntos en x y N_y puntos en y . En la imagen se muestra esquemáticamente la relación de la aproximación $\phi_{i,j}$ con sus vecinos. Los puntos azules son aquellos en los que aplica la relación e iteración de aproximaciones, mientras que los puntos rojos indican los valores de condiciones a la frontera.

Para el problema requerido, se implementarán ambos algoritmos de solución para comparar su desempeño, para esto, en el programa `poisson.py` se implementan considerando:

- Distintas condiciones iniciales para los bordes.
- La matriz inicial de soluciones se puede elegir arbitrariamente, por lo que se opta por utilizar números aleatorios uniformemente distribuidos en $[0,1]$.
- La implementación de la ejecución de los algoritmos hasta un número máximo de iteraciones dado o la verificación de que las matrices de soluciones aproximadas difieran en menos de una tolerancia impuesta. Para esto último se utilizan funciones de Python para el cálculo de normas matriciales (como la norma infinita o de Frobenius) como criterio de comparación entre aproximaciones.
- Representar de forma gráfica y animada la evolución de las soluciones en función del número de iteración.

Algunos resultados obtenidos se muestran en las Figuras anexas.

Una de las observaciones más interesantes es que, en la mayoría de los casos, el método de Jacobi tiene un mejor desempeño que el de Gauss-Seidel en términos de la diferencia entre aproximaciones, esto es, que converge más rápido (en menos iteraciones). Los resultados obtenidos con ambos métodos son muy similares.

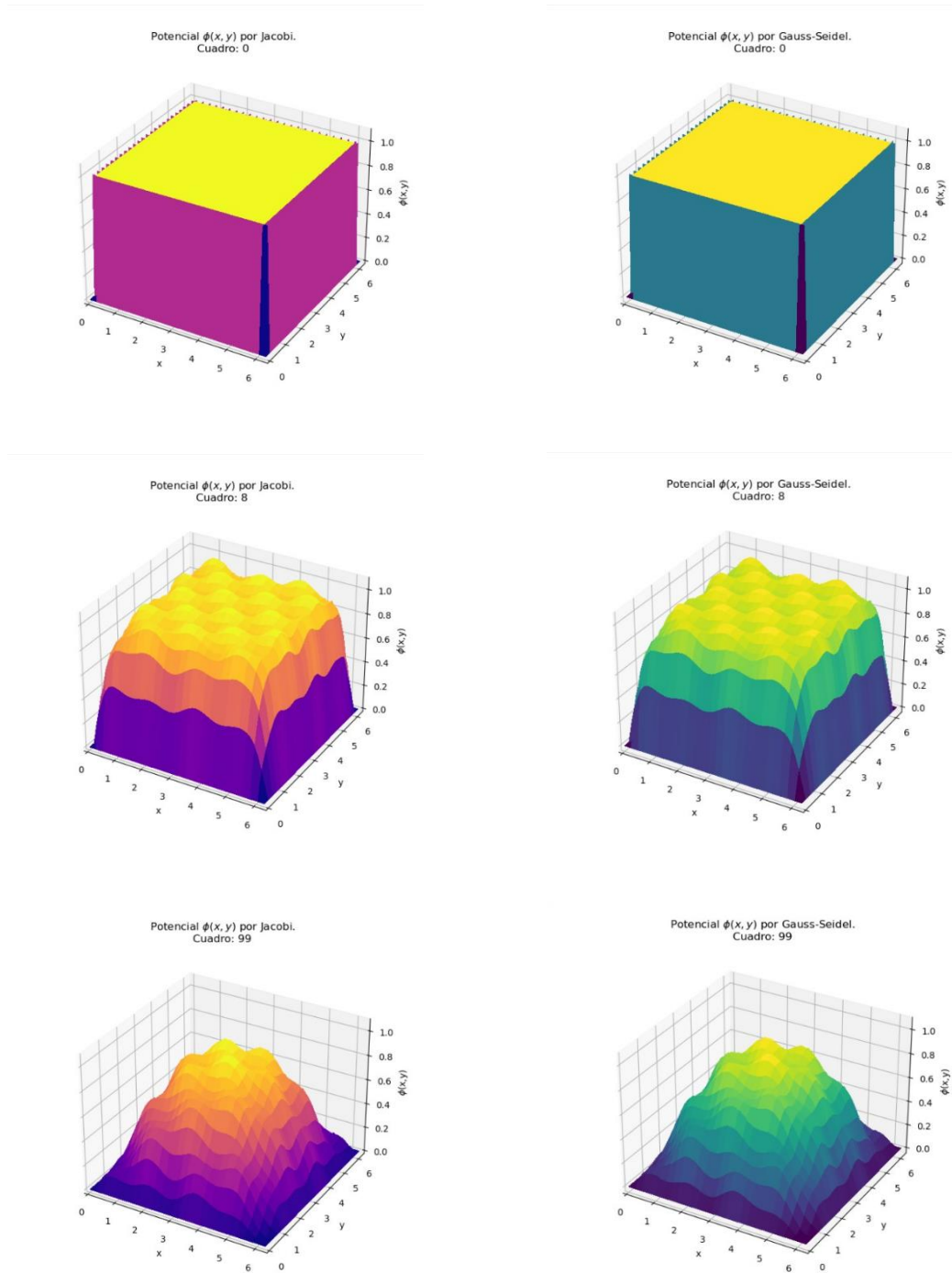


Figura 2. Evolución de la solución numérica de la ecuación de Poisson calculada con los métodos de Jacobi y Gauss-Seidel. En este caso $N_x = N_y = 50$ y se fijan 100 iteraciones como máximas permitidas. Las condiciones de frontera se fijan en cero mientras que la matriz de soluciones inicial contiene solo valores 1.0.

```
Iteración: 95 , diff: 0.05950731879730853
Iteración: 96 , diff: 0.05928459996784596
Iteración: 97 , diff: 0.05906914171345079
Iteración: 98 , diff: 0.058860555548697846
Iteración: 99 , diff: 0.058658910249675955
Iteración: 100 , diff: 0.058463805568329974
```

```
Iteración: 95 , diff: 0.10753016614111033
Iteración: 96 , diff: 0.1075117585079326
Iteración: 97 , diff: 0.10749142087708918
Iteración: 98 , diff: 0.10746888208879568
Iteración: 99 , diff: 0.1074438929058843
Iteración: 100 , diff: 0.10741622489321401
```

Figura 3. Consola durante la ejecución de la solución descrita en la Figura 2.

Se puede observar que ambos algoritmos tuvieron que terminar en el límite de iteraciones permitido, no cumpliendo la tolerancia impuesta de 3×10^{-2} , lo cual dice propiedades de la velocidad de convergencia a la solución. A pesar de la hipótesis señalada anteriormente, la implementación del método de Gauss-Seidel converge más lento a la solución que el método de Jacobi.

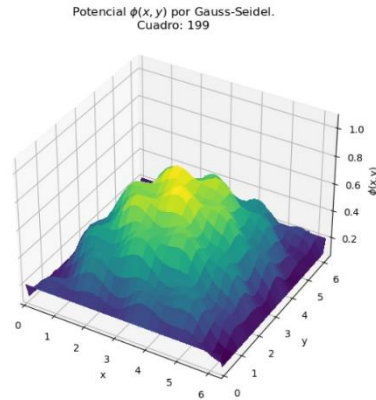
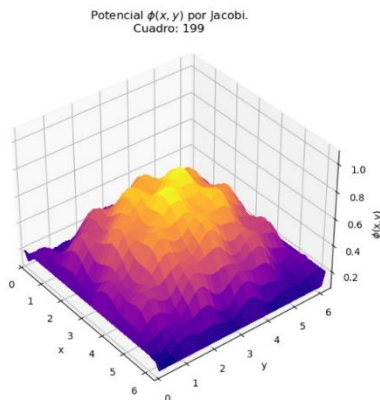
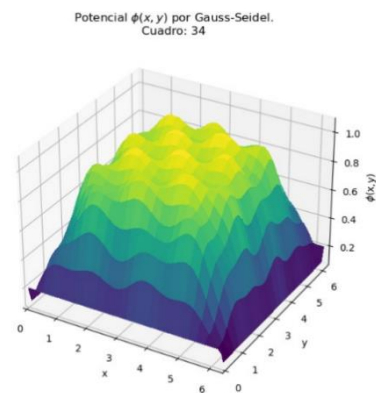
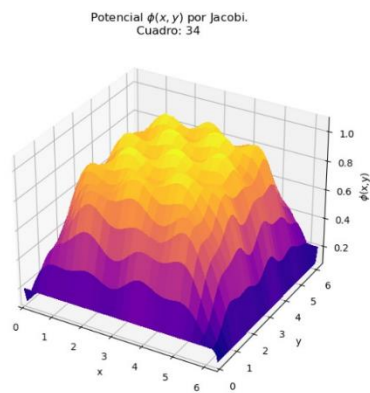


Figura 4. Evolución de la solución numérica de la ecuación de Poisson calculada con los métodos de Jacobi y Gauss-Seidel. En este caso $N_x = N_y = 50$ y se fijan 200 iteraciones como máximas permitidas. Las condiciones de frontera son $\phi(x, 0) = \phi(x, 2\pi) = 0.1$ y $\phi(0, y) = \phi(2\pi, y) = 0.2$. La matriz de soluciones inicial solo contiene valores 1.0.

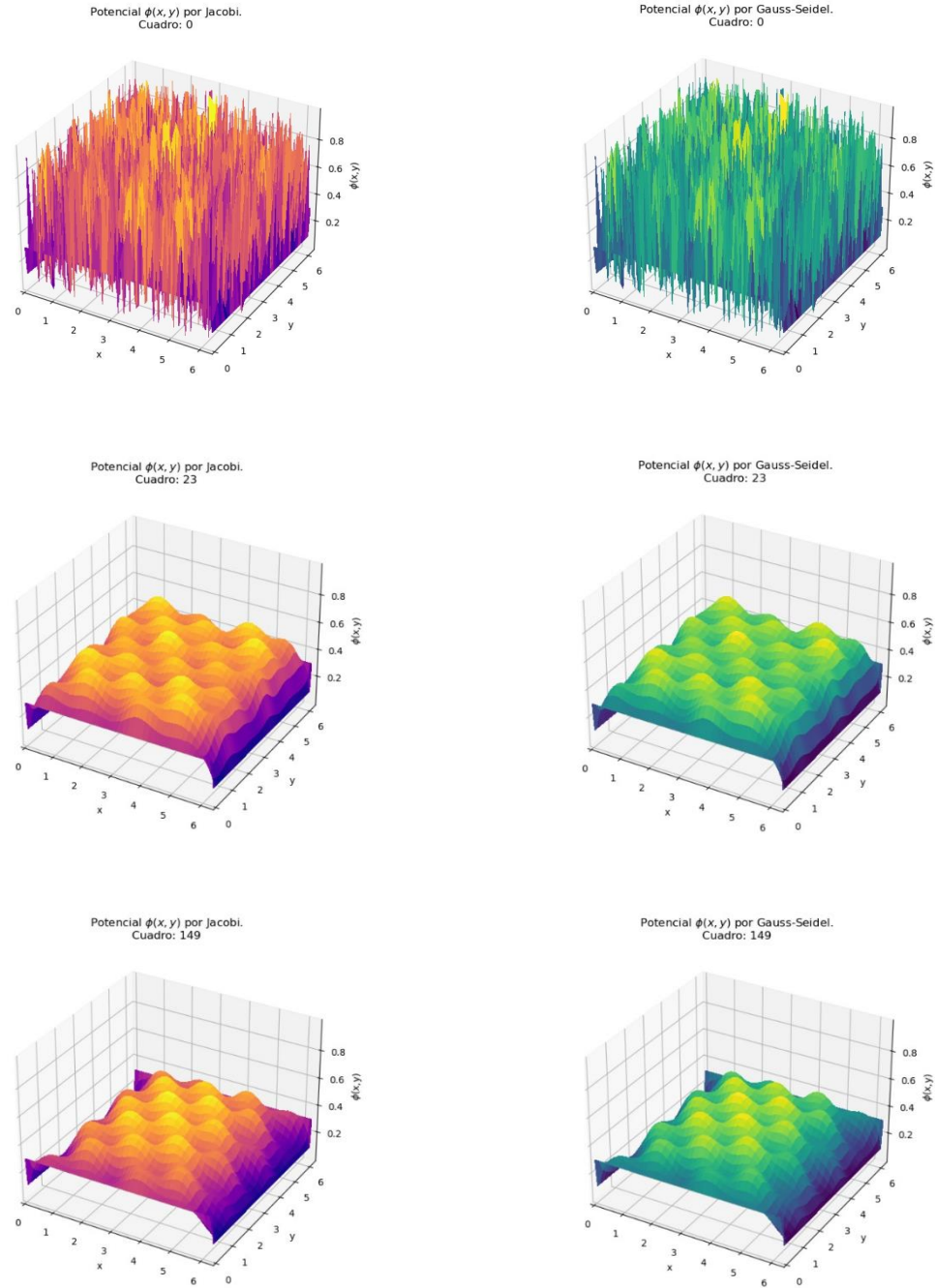


Figura 5. Evolución de la solución numérica de la ecuación de Poisson calculada con los métodos de Jacobi y Gauss-Seidel. En este caso $N_x = N_y = 75$ y se fijan 150 iteraciones como máximas permitidas. Las condiciones de frontera son $\phi(x, 0) = \phi(x, 2\pi) = 0.1$ y $\phi(0, y) = \phi(2\pi, y) = 0.3$ mientras que la matriz de soluciones inicial contiene números aleatorios uniformemente distribuidos en $[0, 1]$.

Lo interesante de este caso es que, a pesar de tener valores de matriz inicial aleatorios, la convergencia de ambos algoritmos es muy rápida y se da durante las primeras 10 iteraciones de los métodos.

Iteración: 145 , diff: 0.022480784707155976	Iteración: 145 , diff: 0.033010352622479644
Iteración: 146 , diff: 0.0215906585155521	Iteración: 146 , diff: 0.03293517494587306
Iteración: 147 , diff: 0.022401126266586857	Iteración: 147 , diff: 0.03286004819114652
Iteración: 148 , diff: 0.021500315439517692	Iteración: 148 , diff: 0.03278496938630937
Iteración: 149 , diff: 0.022323150396211133	Iteración: 149 , diff: 0.032709936049613475
Iteración: 150 , diff: 0.021412579936992415	Iteración: 150 , diff: 0.032634946156928635

Figura 6. Consola durante la ejecución de la solución descrita en la Figura 5.

Se puede observar que ambos algoritmos tuvieron que terminar en el límite de iteraciones permitido, no cumpliendo la tolerancia impuesta de 1×10^{-2} . Se repite el mismo comportamiento de convergencia acelerada del método de Jacobi sobre el de Gauss-Seidel.

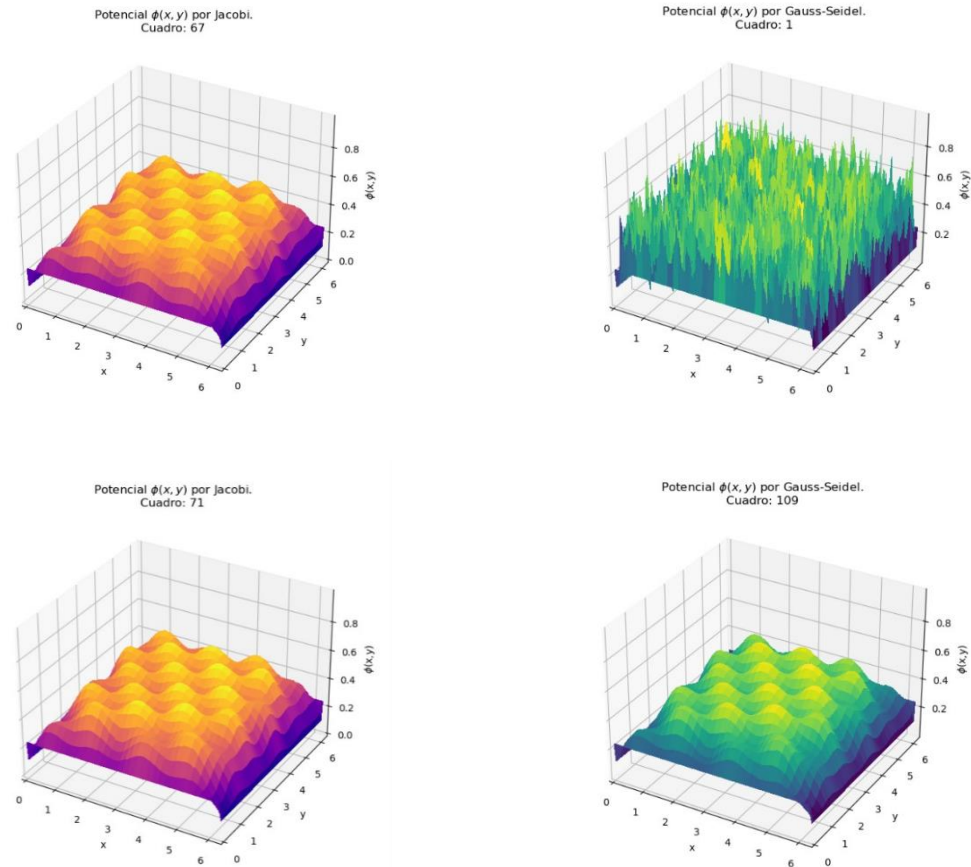


Figura 7. Evolución de la solución numérica de la ecuación de Poisson calculada con los métodos de Jacobi y Gauss-Seidel. En este caso $N_x = N_y = 100$ y se fijan 150 iteraciones como máximas permitidas. Las condiciones de frontera son $\phi(x, 0) = \phi(x, 2\pi) = 0.11$ y $\phi(0, y) = \phi(2\pi, y) = 0.24$. La matriz de soluciones inicial contiene números aleatorios uniformes en $[0,1]$.

En este caso, el método de Jacobi consigue cumplir la tolerancia impuesta de 3×10^{-2} en menos de 85 iteraciones. En comparación, el método de Gauss-Seidel no lo consigue, siendo frenado por el número de iteraciones máximas permitidas.


```
Iteración: 80 , diff: 0.030662374629812916
Iteración: 81 , diff: 0.030475657756817043
Iteración: 82 , diff: 0.030293964366947534
Iteración: 83 , diff: 0.030114387858937164
Iteración: 84 , diff: 0.02993953413131578

Iteración: 146 , diff: 0.032372656630521135
Iteración: 147 , diff: 0.03225885854043753
Iteración: 148 , diff: 0.03214615460182557
Iteración: 149 , diff: 0.032034527057227535
Iteración: 150 , diff: 0.03192395854660646
```

Figura 8. Consola durante la ejecución de la solución descrita en la Figura 7.

A la izquierda se muestran los resultados del método de Jacobi y al lado contrario los de Gauss-Seidel. Es claro que en este caso se alcanzó a cumplir el requerimiento de tolerancia de 3×10^{-2} para el primer método, mientras que el segundo terminó en el número máximo de iteraciones. Estos resultados en conjunto con los presentados en las Figuras 3 y 6 muestran un mejor desempeño del método de Jacobi, contrario a la hipótesis planteada inicialmente, lo cual también podría indicar que la implementación debe ser revisada para corroborar que se traten correctamente las ideas de cada método.