

Algoritmo para la solución numérica de la ecuación de Poisson usando diferencias finitas

Algorithm for the numerical solution of the Poisson's equation using the finite difference

Yohan Díaz Ferrer (ydferrer@uho.edu.cu)

Miguel Cruz Ramírez (mcruzr@uho.edu.cu)

Universidad de Holguín
Provincia Holguín
Cuba

Resumen

El algoritmo está diseñado a partir de un método que resulta una adaptación del método de diferencia finita para problemas de valor de frontera. En el procedimiento se utiliza el método iterativo de Gauss-Seidel para resolver el sistema lineal producido. En la estructura se permiten diferentes tamaños de red en los ejes. Para visualizar la ejecución del algoritmo se incluye la solución numérica de un problema particular.

Palabras y frases clave: algoritmo, ecuación de Poisson, ecuaciones diferenciales en derivadas parciales, solución numérica, diferencias finitas.

Abstract

The algorithm is designed from a method that results in an adaptation of the finite difference method for boundary value problems. In the procedure, the iterative Gauss-Seidel method is used to solve the linear system produced. Different network sizes in the axes are allowed in the structure. To visualize the execution of the algorithm, the numerical solution of a particular problem is included.

Key words and phrases: algorithm, Poisson's equation, differential equations in partial derivatives, numerical solution, finite differences.

1 Introducción

Las situaciones físicas que involucran más de una variable se pueden expresar frecuentemente usando ecuaciones que contienen derivadas parciales [6]. En este trabajo presentamos un algoritmo que ejecuta una de las técnicas disponibles para aproximar la solución de ecuaciones diferenciales parciales que involucran dos variables, mostrando cómo pueden aplicarse a ciertos problemas típicos [1].

Recibido 17/11/2018. Revisado 24/03/2019. Aceptado 21/07/2019.

MSC (2010): Primary 65D15, 33F05; Secondary 35J05, 65N06.

Autor de correspondencia: Yohan Díaz Ferrer

2 Ecuaciones diferenciales parciales elípticas

La ecuación diferencial parcial elíptica que consideraremos es la ecuación de Poisson,

$$\nabla^2 u(x, y) \equiv \frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) \quad (1)$$

para $(x, y) \in \mathbf{R}$ y

$$u(x, y) = g(x, y) \text{ para } (x, y) \in \mathbf{S},$$

donde

$$\mathbf{R} = \{(x, y) \mid a < x < b, c < y < d\}$$

y \mathbf{S} denota la frontera de \mathbf{R} , véase [2]. Para esta discusión supondremos que f y g son continuas en sus dominios con lo que se asegura una solución única de la ecuación (1).

El método que usaremos es una adaptación del método de diferencia finita para problemas de valor de frontera [12]. El primer paso consiste en escoger dos enteros n y m , y en definir los tamaños de pasos h y k por $h = (b - a)/n$ y $k = (d - c)/m$.

Partiendo el intervalo $[a, b]$ en n partes iguales de longitud h y el intervalo $[c, d]$ en m partes iguales de longitud k , ver la figura (1), da una manera de asociar la red al rectángulo \mathbf{R} pasando rectas verticales y horizontales por los puntos con coordenadas (x_i, y_j) donde

$$\begin{aligned} x_i &= a + ih & \text{para cada } i = 0, 1, \dots, n, \\ y_j &= c + jk & \text{para cada } j = 0, 1, \dots, m. \end{aligned}$$

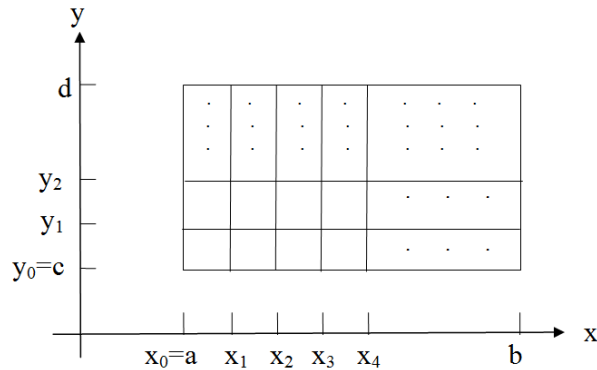


Figura 1. Líneas de red y puntos de red.

Las rectas $x = x_i$ y $y = y_j$ se llaman **líneas de red** y sus intersecciones se llaman **puntos de red** [10]. Por cada punto de red en el interior de la malla (x_i, y_j) , $i = 0, 1, \dots, n - 1$ y $j = 0, 1, \dots, m - 1$ usamos la serie de Taylor en la variable x alrededor de x_i para generar la fórmula de diferencia centrada

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} - \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi_i, y_j)$$

donde $\xi \in (x_{i-1}, x_{i+1})$ y la serie de Taylor en la variable y alrededor de y_j para generar la fórmula de diferencia centrada

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} - \frac{k^2}{12} \frac{\partial^4 u}{\partial y^4}(x_i, \eta_j),$$

donde $\eta_j \in (y_{i-1}, y_{i+1})$.

Usando estas fórmulas en la ecuación (1) podemos expresar la ecuación de Poisson en los puntos (x_i, y_j) como:

$$\begin{aligned} & \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} = \\ & = f(x_i, y_j) + \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) + \frac{k^2}{12} \frac{\partial^4 u}{\partial y^4}(x_i, \eta_j) \end{aligned}$$

para cada $i = 0, 1, \dots, (n - 1)$ y $j = 0, 1, \dots, (m - 1)$, y las condiciones de frontera como

$$\begin{aligned} u(x_0, y_j) &= g(x_0, y_j) && \text{para cada } j = 0, 1, \dots, m, \\ u(x_n, y_j) &= g(x_n, y_j) && \text{para cada } j = 0, 1, \dots, m, \\ u(x_i, y_0) &= g(x_i, y_0) && \text{para cada } i = 0, 1, \dots, n - 1, \\ u(x_i, y_m) &= g(x_i, y_m) && \text{para cada } i = 0, 1, \dots, n - 1, \end{aligned}$$

En la forma de ecuación de diferencia, esto da lugar a un método llamado **diferencia centrada** con error de truncamiento local de orden $\mathcal{O}(h^2 + k^2)$, que puede escribirse como:

$$2 \left[\left(\frac{h}{k} \right)^2 + 1 \right] w_{i,j} - (w_{i+1,j} + w_{i-1,j}) - \left(\frac{h}{k} \right)^2 (w_{i,j} + w_{i,j-1}) = -h^2 f(x_i, y_j), \quad (2)$$

para cada $i = 0, 1, \dots, (n - 1)$ y $j = 0, 1, \dots, (m - 1)$,

$$\begin{aligned} w_{0,j} &= g(x_0, y_j) && \text{para cada } j = 0, 1, \dots, m, \\ w_{n,j} &= g(x_n, y_j) && \text{para cada } j = 0, 1, \dots, m, \\ w_{i,0} &= g(x_i, y_0) && \text{para cada } i = 0, 1, \dots, n - 1, \\ w_{i,m} &= g(x_i, y_m) && \text{para cada } i = 0, 1, \dots, n - 1, \end{aligned} \quad (3)$$

donde $w_{i,j}$ aproxima a $u(x_i, y_j)$.

La ecuación típica (2) involucra aproximaciones a $u(x, y)$ en los puntos

$$(x_{i-1}, y_j)(x_i, y_j)(x_{i+1}, y_j)(x_i, y_{j-1})(x_i, y_{j+1}).$$

Reproduciendo la porción de la red donde están localizados estos puntos, figura (2), vemos que cada ecuación contiene aproximaciones en una región en forma de estrella alrededor de (x_i, y_j) . Si usamos la información de las condiciones de frontera (3) donde sea apropiado en el sistema dado por (2), esto es, en todos los puntos (x, y) que están adyacentes a un punto de red en la frontera, tendremos un sistema de $(n - 1)(m - 1)$ ecuaciones lineales con $(n - 1)(m - 1)$ incógnitas, siendo las incógnitas las aproximaciones $w_{i,j}$ de $u(x_i, y_j)$ para los puntos interiores de la red.

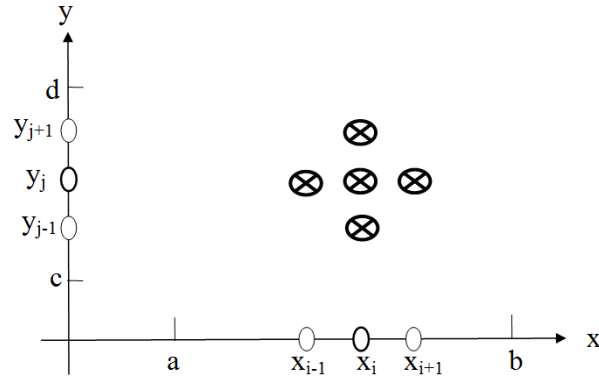


Figura 2. Porción de la red en forma de estrella.

El sistema lineal que contiene a estas incógnitas puede expresarse más eficientemente para los cálculos de la matriz si se introduce un renombramiento de los puntos interiores de la red [7]. Una manera recomendable de etiquetar estos puntos tomando $P_l = (x_i, y_j)$ y $w_l = w_{i,j}$, donde $l = i + (m - 1 - j)(n - 1)$ para cada $i = 0, 1, \dots, (n - 1)$ y $j = 0, 1, \dots, (m - 1)$ [4]. Esto, de hecho, reordena los puntos de red consecutivamente de izquierda a derecha y de arriba hacia abajo. Por ejemplo con $n = 4$ y $m = 5$ esta reordenación da por resultado una malla cuyos puntos se muestran en la figura (3). Marcando los puntos de esta manera asegura que el sistema necesario para determinar $w_{i,j}$ será una matriz banda con espesor de la banda lo más $2n - 1$, véase [11].

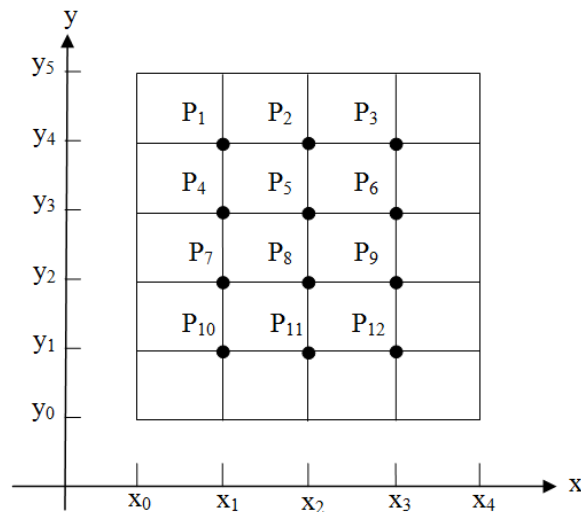


Figura 3. Puntos de una malla.

3 Problema particular

Considérese el problema de determinar la distribución estacionaria de calor en una lámina delgada de metal en forma de cuadrado con dimensiones de 0,5 metros por 0,5 metros, la cual se mantiene a 0°C en dos fronteras adyacentes mientras que el calor en las otras fronteras se va

incrementando linealmente de 0°C en una esquina a 100°C donde estos lados se encuentran Si ponemos los lados con condición de frontera cero a lo largo de los ejes x y y , el problema se expresará matemáticamente como:

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = 0,$$

para (x, y) en el conjunto $\mathbf{R} = \{(x, y) | 0 < x < 0,5, 0 < y < 0,5\}$ con las condiciones de frontera

$$u(0, y) = 0, u(x, 0) = 0, u(x, 0,5) = 200x, u(0,5, y) = 200y \tag{4}$$

Si $n = m = 4$, la malla del problema es la dada en la figura (4) y la ecuación de diferencia (2) es

$$4w_{i,j} - w_{i+1,j} - w_{i-1,j} - w_{i,j-1} - w_{i,j+1} = 0$$

para cada $i = 1, 2, 3$ y $j = 1, 2, 3$,

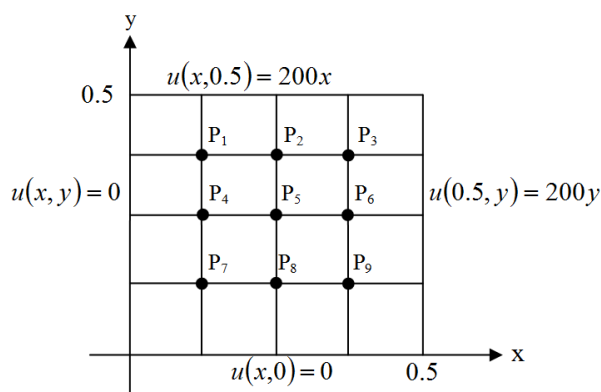


Figura 4. Malla del problema.

Expresando esto en términos de los puntos interiores de la red reetiquetados $w_i = u(P_i)$ implica que las ecuaciones son:

$$\begin{aligned} 4w_1 - w_2 - w_4 &= w_{0,3} + w_{1,4}, \\ 4w_2 - w_3 - w_1 - w_5 &= w_{2,4}, \\ 4w_3 - w_2 - w_6 &= w_{4,3} + w_{3,4}, \\ 4w_4 - w_5 - w_1 - w_7 &= w_{0,2}, \\ 4w_5 - w_6 - w_4 - w_2 - w_8 &= 0, \\ 4w_6 - w_5 - w_3 - w_9 &= w_{4,2}, \\ 4w_7 - w_8 - w_4 &= w_{0,1} + w_{1,0}, \\ 4w_8 - w_9 - w_7 - w_5 &= w_{2,0}, \\ 4w_9 - w_8 - w_6 &= w_{3,0} + w_{4,1}, \end{aligned}$$

donde los lados de la derecha de las ecuaciones se obtienen de las condiciones de frontera [3]. De hecho, las condiciones dadas en (4) implican que

$$w_{1,0} = w_{2,0} = w_{3,0} = w_{0,1} = w_{0,2} = w_{0,3} = 0$$

$$w_{1,4} = w_{4,1} = 25, \quad w_{2,4} = w_{4,2} = 50 \quad \text{y}$$

$$w_{3,4} = w_{4,3} = 75.$$

El sistema lineal asociado con este problema tiene la forma

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \end{bmatrix} = \begin{bmatrix} 25 \\ 50 \\ 150 \\ 0 \\ 0 \\ 50 \\ 0 \\ 0 \\ 25 \end{bmatrix}$$

Los valores de w_1, w_2, \dots, w_9 que se encuentran aplicando el método de Gauss-Seidel a esta matriz, están dados como sigue:

i	1	2	3	4	5	6	7	8	9
$w_{i,j}$	18.75	37.50	56.25	12.50	25.00	37.50	6.25	12.50	18.75

Las respuestas son correctas, ya que la solución $u(x, y) = 400xy$ tiene

$$\frac{\partial^4 u}{\partial x^4}(x, y) = \frac{\partial^4 u}{\partial y^4}(x, y) \equiv 0,$$

así que el error de truncamiento es cero en cada paso.

El problema que consideramos tiene un tamaño de red de 0.125 en cada eje y requiere la solución de solamente un sistema lineal de 9×9 . Esto simplifica considerablemente la situación y no introduce los problemas computacionales que se presentan cuando el sistema es mucho más grande [4]. El algoritmo que se presenta a continuación utiliza el método de Gauss-Seidel para resolver el sistema lineal producido y permite diferentes tamaños de red en los ejes.

4 Algoritmo de diferencia finita para la ecuación de Poisson

Para aproximar la solución de la ecuación de Poisson

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = f(x, y), \quad a \leq x \leq b, \quad c \leq y \leq d,$$

sujeta a las condiciones de frontera

$$\begin{aligned} u(x, y) &= g(x, y) && \text{si } x = a \text{ o } x = b \text{ y } c \leq y \leq d \\ y \quad u(x, y) &= g(x, y) && \text{si } y = c \text{ o } y = d \text{ y } a \leq x \leq b, \end{aligned}$$

se establece el siguiente algoritmo

Algoritmo

Entrada: puntos extremos a, b, c, d ;

$m, n \in \mathbb{Z}$;

tolerancia TOL ;

número máximo de iteraciones $LBOUND$.

Salida: aproximaciones $w_{i,j}$ a $u(x_i, y_j)$ para cada $i = 0, 1, \dots, n - 1$ y $j = 0, 1, \dots, m - 1$

o un mensaje de que el número de iteraciones fue excedido.

1. Calcular $h = (b - a)/n$;
 $k = (d - c)/m$.
 (Los pasos 2 y 3 construyen los puntos de red)
2. **para** $i = 1, 2, \dots, n - 1$ **hacer**
 calcular $x_i = a + ih$.
3. **para** $j = 1, 2, \dots, m - 1$ **hacer**
 calcular $y_j = c + jk$.
4. **para** $i = 1, 2, \dots, n - 1$ **hacer**
 para $j = 1, 2, \dots, m - 1$ **hacer**
 calcular $w_{i,j} = 0$.
5. Calcular $\lambda = h^2/k^2$;
 $\mu = 2(1 + \lambda)$;
 $l = 1$.
6. (Los pasos del 7 al 20 realizan iteraciones de Gauss-Seidel)
 mientras $l \leq LBOUND$ **hacer**
7. Calcular $z = (-h^2 f(x_1, y_{m-1}) + g(a, y_{m-1}) + \lambda g(x_1, d) + \lambda w_{1,m-2} + w_{2,m-1})/\mu$;
 $NORM = |z - w_{1,m-1}|$;
 $w_{1,m-1} = z$.
8. **para** $i = 2, \dots, n - 2$ **hacer**
 calcular $z = (-h^2 f(x_i, y_{m-1}) + \lambda g(x_i, d) + w_{i-1,m-1} + w_{i+1,m-1} + \lambda w_{i,m-2})/\mu$;
 si $|w_{i,m-1} - z| > NORM$ **entonces**
 Calcular $NORM = |w_{i,m-1} - z|$;
 Calcular $w_{i,m-1} = z$.
9. Calcular $z = (-h^2 f(x_{n-1}, y_{m-1}) + g(b, y_{m-1}) + \lambda g(x_{n-1}, d) + w_{n-2,m-1} + \lambda w_{n-1,m-2})/\mu$;
 si $|w_{n-1,m-1} - z| > NORM$ **entonces**
 Calcular $w_{n-1,m-1} = z$.
10. **para** $j = m - 2, \dots, 2$ **hacer** (Seguir los pasos 11, 12 y 13)
11. Calcular $z = (-h^2 f(x_1, y_j) + g(a, y_j) + \lambda w_{1,j+1} + \lambda w_{1,j-1} + w_{2,j})/\mu$;
 si $|w_{1,j} - z| > NORM$ **entonces**
 Calcular $NORM = |w_{1,j} - z|$;
 Calcular $w_{1,j} = z$.

12. **para** $i = 2, \dots, n - 2$ **hacer**
 Calcular $z = (-h^2 f(x_i, y_j) + w_{i-1,j} + \lambda w_{i,j+1} + w_{i+1,j} + \lambda w_{i,j-1})/\mu$;
si $|w_{i,j} - z| > NORM$ **entonces**
 Calcular $NORM = |w_{i,j} - z|$;
 Calcular $w_{i,j} = z$.
13. Calcular $z = (-h^2 f(x_{n-1}, y_j) + g(b, y_j) + w_{n-2,j} + \lambda w_{n-1,j+1} + \lambda w_{n-1,j-1})/\mu$;
si $|w_{n-1,j} - z| > NORM$ **entonces**
 Calcular $NORM = |w_{n-1,j} - z|$;
 Calcular $w_{n-1,j} = z$.
14. Calcular $z = (-h^2 f(x_1, y_1) + g(a, y_1) + \lambda g(x_1, c) + \lambda w_{1,2} + w_{2,1})/\mu$;
si $|w_{1,1} - z| > NORM$ **entonces**
 Calcular $NORM = |w_{1,1} - z|$;
 Calcular $w_{1,1} = z$.
15. **para** $i = 2, \dots, n - 2$ **hacer**
 Calcular $z = (-h^2 f(x_i, y_1) + \lambda g(x_i, c) + w_{i-1,1} + \lambda w_{i,2} + w_{i+1,1})/\mu$;
si $|w_{i,1} - z| > NORM$ **entonces**
 Calcular $NORM = |w_{i,1} - z|$;
 Calcular $w_{i,1} = z$.
16. Calcular $z = (-h^2 f(x_{n-1}, y_1) + g(b, y_1) + \lambda g(x_{n-1}, c) + w_{n-2,1} + \lambda w_{n-1,2})/\mu$;
si $|w_{n-1,1} - z| > NORM$ **entonces**
 Calcular $NORM = |w_{n-1,1} - z|$;
 Calcular $w_{n-1,1} = z$.
17. **si** $NORM \leq TOL$ **entonces** (Seguir los pasos 18 y 19)
18. **para** $i = 1, \dots, n - 1$ **hacer**
para $j = 1, \dots, m - 1$ **hacer**
SALIDA $(x_i, y_j, w_{i,j})$.
19. **PARAR.** (Procedimiento completado satisfactoriamente.)
20. Calcular $l = l + 1$.
21. **SALIDA** ('Número máximo de iteraciones excedido');
PARAR.

5 Acerca de la estabilidad y convergencia

Aun cuando, por simplicidad, se ha incorporado el procedimiento iterativo de Gauss-Seidel en el algoritmo, es recomendable en general usar una técnica directa como la eliminación Gaussiana cuando el sistema es pequeño, del orden de 100 o menor, ya que la simetría y la propiedad de ser positiva definida asegurarán la estabilidad con respecto a errores de redondeo. En particular, la generalización del algoritmo de reducción de Crout, ver [5], es muy eficiente para resolver este sistema, ya que la matriz está en la forma tridiagonal simétrica por bloques

$$\begin{bmatrix} A_1 & C_1 & 0 & \cdots & \cdots & 0 \\ C_1 & A_2 & C_2 & \ddots & & \vdots \\ 0 & C_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & C_{m-1} \\ 0 & \cdots & \cdots & 0 & C_{m-1} & A_{m-1} \end{bmatrix}$$

con bloques cuadrados de tamaño $(n - 1)$ por $(n - 1)$.

Para sistemas muy grandes se recomienda usar un método iterativo, específicamente, el método SOR. El valor de w que es óptimo en esta situación viene del hecho de que cuando A se descompone en sus partes diagonal D y triangulares superior e inferior U y L ,

$$A = D - L - U,$$

y B es la matriz de Jacobi,

$$B = D^{-1}(L + U),$$

entonces el radio espectral de B es (ver [9])

$$\rho(B) = \frac{1}{2} \left[\cos\left(\frac{\pi}{m}\right) + \left(\frac{\pi}{n}\right) \right].$$

El valor de w a usar es consecuentemente

$$w = \frac{2}{1 + \sqrt{1 - [\rho(B)]^2}} = \frac{4}{2 + \sqrt{4 - [\cos\left(\frac{\pi}{m}\right) + \left(\frac{\pi}{n}\right)]^2}}.$$

Para una convergencia más rápida del procedimiento SOR, se puede incorporar una técnica de bloques en el algoritmo. Para una presentación de la técnica involucrada, ver [9].

6 Ejemplo

Considere la ecuación de Poisson

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = xe^y, \quad 0 < x < 2, \quad 0 < y < 1,$$

con las condiciones de frontera

$$\begin{aligned} u(0, y) &= 0, & u(2, y) &= 2e^y, & 0 \leq y \leq 1 \\ u(x, 0) &= x, & u(x, 1) &= ex, & 0 \leq x \leq 2. \end{aligned}$$

Usaremos el algoritmo para aproximar la solución exacta $u(x, y) = xe^y$ con $n = 6$ y $m = 5$. El criterio de paro requerido en el paso 17 fue

$$|w_{i,j}^{(l)} - w_{i,j}^{(l-1)}| \leq 10^{-10},$$

para cada $i = 1, \dots, 5$, y $j = 1, \dots, 4$; así, la solución de la ecuación de diferencia se obtuvo con precisión y el procedimiento fue parado en $l = 61$. Los resultados, junto con los valores correctos, se presentan en la siguiente tabla

i	j	x_i	y_j	$w_{i,j}^{(61)}$	$u(x_i, y_j)$	$ u(x_i, y_j) - w_{i,j}^{(61)} $
1	1	0,3333	0,2000	0,40726	0,40713	$1,30 \times 10^{-4}$
1	2	0,3333	0,4000	0,49748	0,49727	$2,08 \times 10^{-4}$
1	3	0,3333	0,6000	0,60760	0,60737	$2,23 \times 10^{-4}$
1	4	0,3333	0,8000	0,74201	0,74185	$1,60 \times 10^{-4}$
2	1	0,6667	0,2000	0,81452	0,81427	$2,55 \times 10^{-4}$
2	2	0,6667	0,4000	0,99496	0,99455	$4,08 \times 10^{-4}$
2	3	0,6667	0,6000	1,2152	1,2147	$4,37 \times 10^{-4}$
2	4	0,6667	0,8000	1,4840	1,4837	$3,15 \times 10^{-4}$
3	1	1,0000	0,2000	1,2218	1,2214	$3,64 \times 10^{-4}$
3	2	1,0000	0,4000	1,4924	1,4918	$5,80 \times 10^{-4}$
3	3	1,0000	0,6000	1,8227	1,8221	$6,24 \times 10^{-4}$
3	4	1,0000	0,8000	2,2260	2,2255	$4,51 \times 10^{-4}$
4	1	1,3333	0,2000	1,6290	1,6285	$4,27 \times 10^{-4}$
4	2	1,3333	0,4000	1,9898	1,9891	$6,79 \times 10^{-4}$
4	3	1,3333	0,6000	2,4302	2,4295	$7,35 \times 10^{-4}$
4	4	1,3333	0,8000	2,9679	2,9674	$5,40 \times 10^{-4}$
5	1	1,6667	0,2000	2,0360	2,0357	$3,71 \times 10^{-4}$
5	2	1,6667	0,4000	2,4870	2,4864	$5,84 \times 10^{-4}$
5	3	1,6667	0,6000	3,0375	3,0369	$6,41 \times 10^{-4}$
5	4	1,6667	0,8000	3,7097	3,7092	$4,89 \times 10^{-4}$

Referencias

- [1] Antonietti, P. , Verani, M. & Zikatanov, L. *A two level method for mimetic finite difference discretizations of elliptic problems*, Computer and Matematics with applications, (2015), doi: 10.1016/j.camwa.2015.06.010.
- [2] Al-Jawary, M. & Hatif, S. *A semi-analitical iterative method for solving differential algebraic equations*, Ain Chams Engineering Journal, (1989), doi: 10.1016/j.asej.2017. 07.004.
- [3] Beiro da Veiga, L., Lopez, L. & Vacca, G. *Mimetic finite difference methods for Hamiltonian wave equations in 2D*, Computer and Matematics with applications, (2017), doi: 10.1016/j.camwa.2017.05.022.

- [4] Cacasús Acevedo, A. *Aplicación del método de diferencias finitas generalizadas a problemas de elasto-dinámica. (Tesis inédita de doctorado)*, E.T.S. de Ingenieros Industriales UNED, Madrid, España, 2011.
- [5] De la Fuente, J.L. *Técnicas de cálculo para sistemas de ecuaciones, programación lineal y programación entera: códigos en FORTRAN y C con aplicaciones de sistemas de energía eléctrica.*, Reverté, Madrid, España, 1997.
- [6] Fernández, L. A. *Introducción a las ecuaciones en derivadas parciales*, Universidad de Cantabria, Cantabria, España, 2016.
- [7] Gosse, L. *Dirichlet-to-neumann mapping and finite-differences for anisotropic diffusion*, Computers and Fluids, **156**(2017), 58–65.
- [8] Quintanilla Murillo, J. *Métodos numéricos en diferencias finitas para la resolución de ecuaciones difusivas fraccionarias. (Tesis inédita de doctorado)*, Universidad de Extremadura, Badajoz, España, 2016.
- [9] Varga, R.S. *Matrix iterative analysis.*, Prentice Hall, Englewood Cliffs, New Jersey, United State of America, 1962.
- [10] Wang, H., Liang, H., & Chai, Z. *Finite-difference lattice Boltzmann model for nonlinear convection-diffusion equations*, Applied Mathematics and Computation, **309**(2017), 334–349, doi: 10.1016/j.amc.2017.04.015.
- [11] Xue, G. & Zhang, L. *A new finite difference schemes for generalized Rosenau-Burgers equation*, Applied Mathematics and Computation, **222**(2013), 490–496, doi: 10.1016/j.amc.2013.07.052.
- [12] Zapata, M. U. & Itzá, R. *High-order implicit finite difference schemes for the two dimensional Poisson equation*, Applied Mathematics and Computation, **309**(2017), 222–244, doi: 10.1016/j.amc.2017.04.006.