

SOLUÇÃO NUMÉRICA DA EQUAÇÃO DE POISSON 2D E 3D EM MALHAS ESTRUTURADAS¹

Rudimar Luiz Nós²
João Pedro Santos Brito Micheletti

RESUMO

Apresentamos neste trabalho a solução numérica de algumas equações de Poisson, uma equação diferencial parcial elíptica de segunda ordem, em malhas estruturadas bidimensionais e tridimensionais. Na determinação da solução numérica, empregamos o método iterativo SOR para solucionar o sistema de equações lineares proveniente da discretização da equação de Poisson por intermédio do método de diferenças finitas. Além disso, construímos algumas soluções manufaturadas 2D e 3D para a equação de Poisson, testamos valores ótimos para o parâmetro de sobre-relaxação no método SOR e analisamos o comportamento dos métodos empregados na solução numérica de problemas 2D com singularidades. Na visualização das soluções manufaturadas e numéricas 2D e 3D, utilizamos, respectivamente, o Matlab e o Tecplot 360. Concluímos que a convergência do método SOR é lenta em problemas com condições de contorno de Neumann e em problemas com singularidades fortes.

Palavras-chave: Método de diferenças finitas. Método SOR. Soluções manufaturadas.

NUMERICAL SOLUTION OF 2D AND 3D POISSON EQUATION IN STRUCTURED MESHES

ABSTRACT

We present in this work the numerical solution of some Poisson equations, an elliptic partial differential equation of second order, in two-dimensional and three-dimensional structured meshes. In determining the numerical solution, we used the iterative SOR method to solve the system of linear equations arising from the discretization of the Poisson equation using the finite difference method. Furthermore, we build some 2D and 3D manufactured solutions for the Poisson equation, and test optimal values for the over-relaxation parameter in the SOR method and analyze the behavior of the methods used in the numerical solution of 2D problems with singularities. In the visualization of the 2D and 3D manufactured and numerical solutions, we used, respectively, Matlab and Tecplot 360. We concluded that the convergence of the SOR method is slow in problems with Neumann boundary conditions and in problems with strong singularities.

Keywords: Finite difference method. SOR method. Manufactured solutions.

¹ **Como citar este artigo:** NÓS, R. L.; MICHELETTI, J. P. S. B. Solução numérica da equação de poisson 2d e 3d em malhas estruturadas. **ForScience**, Formiga, v. 9, n. 2, e00945, jul./dez. 2021. [DOI: 10.29069/forscience.2021v9n2.e1091](https://doi.org/10.29069/forscience.2021v9n2.e1091).

² **Autor para correspondência:** Rudimar Luiz Nós, e-mail: rudimarnos@gmail.com.

1 INTRODUÇÃO

Uma equação diferencial parcial (EDP) é uma igualdade que relaciona uma função (ou funções) de duas ou mais variáveis independentes com suas derivadas parciais, ou seja, derivadas relativas a pelo menos duas variáveis independentes (IÓRIO, 1989).

A EDP

$$\nabla u = \frac{\partial^2 u}{\partial x_1^2} + \dots + \frac{\partial^2 u}{\partial x_n^2} = f, \quad (1)$$

onde $u = u(x_1, \dots, x_n)$, $f = f(x_1, \dots, x_n)$ e ∇ denota o operador Laplaciano é denominada Equação de Laplace³ se $f = 0$ (equação homogênea) e, neste caso, as funções u que a satisfazem são funções harmônicas; é denominada Equação de Poisson⁴ se $f \neq 0$ (equação não homogênea). A Equação (1) é linear, de segunda ordem (IÓRIO, 1989).

A unicidade de solução da Equação (1) depende de condições auxiliares iniciais e/ou de contorno. Na condição inicial é imposto o valor da solução no domínio de cálculo, como por exemplo, $u(x_1, \dots, x_n) = 0$; nas condições de contorno, são impostos o valor da solução e de suas derivadas nas fronteiras do domínio espacial. Essas condições são do tipo

$$\alpha u(x_1, \dots, x_n) + \beta \frac{\partial}{\partial n} u(x_1, \dots, x_n) = f(x_1, \dots, x_n), \quad u(x_1, \dots, x_n) \in \partial\Omega, \quad (2)$$

onde α e β são constantes dadas, $f(x_1, \dots, x_n)$ é uma função dada no bordo $\partial\Omega$ do domínio espacial Ω e $\frac{\partial}{\partial n} u(x_1, \dots, x_n)$ é a derivada de $u(x_1, \dots, x_n)$ na direção normal à $\partial\Omega$. Quando $\beta = 0$, a condição (2) é denominada condição de Dirichlet⁵; quando $\alpha = 0$, condição de Neumann⁶; quando $\alpha, \beta \neq 0$, condição de Robin⁷.

A Equação (1) modela vários problemas da Física Matemática, entre eles o estudo de campos eletrostáticos, a energia de uma partícula sobre a qual agem apenas forças gravitacionais e a distribuição de temperatura em estado estacionário (IÓRIO, 1989). A Figura 1 ilustra a distribuição de temperatura em uma placa plana, modelada pela Equação de Poisson

$$\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = -\frac{5}{10^6}, \quad (3)$$

³Pierre-Simon Laplace (1749-1827): matemático, astrônomo e físico francês.

⁴Siméon Denis Poisson (1781-1840): matemático e engenheiro francês.

⁵Peter Gustav Lejeune Dirichlet (1805-1859): matemático alemão.

⁶John von Neumann (1903-1957): matemático, físico e engenheiro húngaro-americano.

⁷Victor Gustave Robin (1855-1897): matemático aplicado francês.

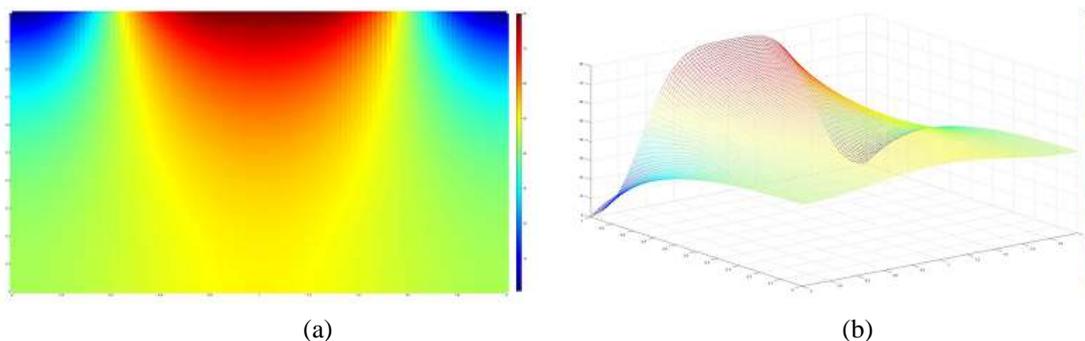


Figura 1 - Distribuição de temperatura no domínio $[0,2] \times [0,1]$ dada pela Equação (3), com condições iniciais nulas e condições de contorno de Dirichlet: (a) visualização bidimensional; (b) visualização tridimensional
Fonte: Os autores com o Matlab (2021).

com condição inicial $u(x, y) = 0, 0 < x < 2, 0 < y < 1$, e condições de contorno de Dirichlet: $u(x, 0) = 20, u(x, 1) = 80e^{-5(1-x)^4}, u(0, y) = 20, u(2, y) = 20$ (MICHELETTI, 2021, p. 83).

Dessa forma, solucionar numericamente a Equação (1) com condições de contorno de Dirichlet e de Neumann, em domínios espaciais bidimensionais ($n = 2$) e tridimensionais ($n = 3$), é o objetivo do presente trabalho. Para tanto, construímos códigos computacionais 2D e 3D, em linguagem C (LAPLACE, 2021; SCHILDT, 1997), que empregam os métodos de diferenças finitas (FORTUNA, 2000) e SOR (BURDEN; FAIRES; BURDEN, 2016), e determinamos soluções manufaturadas para testar esses códigos com valores ótimos para o parâmetro de sobre-relaxação no SOR. Além disso, utilizamos o esquema numérico na solução de problemas 2D com singularidades (MITCHELL, 2017). Para visualizar graficamente as soluções 2D e 3D usamos, respectivamente, o Matlab (MATLAB, 2021) e o Tecplot 360 (TECPLOT, 2021).

2 REFERENCIAL TEÓRICO

2.1 Discretização

A solução de uma EDP, como a Equação de Poisson, em uma região contínua R implica na obtenção de valores para a função incógnita em todos os pontos de R . Porém, numericamente, o computador pode calcular a solução em uma quantidade finita de pontos dessa região. Este conjunto de pontos onde a solução é calculada denomina-se malha (FORTUNA, 2000). A malha é dita estruturada quando todo ponto da mesma possui o mesmo número de pontos vizinhos, exceto quando o ponto pertence à fronteira (ou contorno) (MALISKA, 2004). Em malhas estruturadas 2D, as células computacionais são quadriláteros e os nós de cada célula são identificados pelos índices i e j ; em malhas estruturadas 3D, as células

computacionais são hexaedros e os nós de cada células são identificados pelos índices i, j e k . Assim, as malhas estruturadas facilitam a programação numérica devido à distribuição matricial dos pontos (MALISKA, 2004). A Figura 2 ilustra uma malha 2D 4×4 , onde 4 indica o número de partições em cada direção, com 25 nós (pontos), relativa à região contínua R .

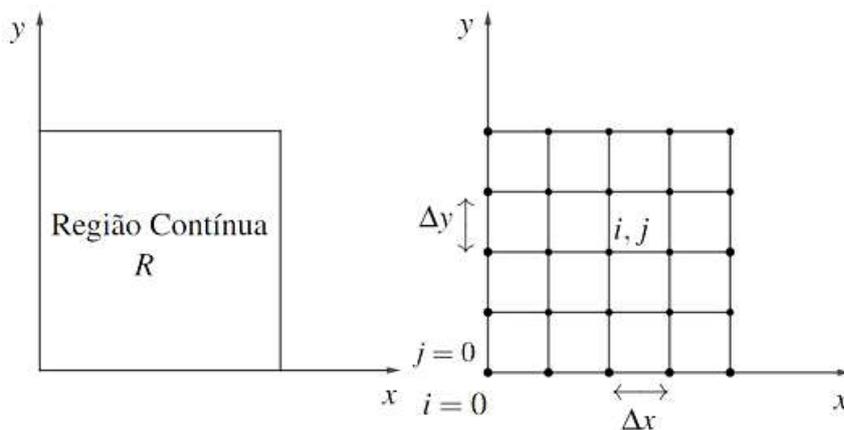


Figura 2 – Malha 2D 4×4 com 25 nós relativa à região contínua R

Fonte: Os autores baseados em Fortuna (2000, p. 74).

Os pontos da malha na Figura 2 estão localizados na intersecção das retas paralelas ao eixo x com as retas paralelas ao eixo y , separados respectivamente por distâncias Δx e Δy . Quanto $\Delta x = \Delta y$, a malha é isotrópica; caso contrário, é anisotrópica. Desta forma, um ponto (i, j) da malha tem coordenadas $(x_0 + i\Delta x, y_0 + j\Delta y)$, sendo que (x_0, y_0) representa o origem do sistema de coordenadas.

Para que possamos solucionar uma EDP numericamente, esta deve ser expressa na forma de operações aritméticas que o computador possa executar (FORTUNA, 2000).

Segundo Fortuna (2000, p. 74), “devemos representar os diferenciais da EDP por expressões algébricas, ou seja, *discretizar* a EDP”. As expressões algébricas relativas aos termos da EDP são escritas em função dos pontos da malha e constituem aproximações por diferenças finitas (LOGAN, 2015). A conclusão desse processo é uma equação algébrica, denominada equação de diferenças finitas ou EDF, que é aplicada a cada ponto da malha, originando um sistema de equações lineares. A solução desse sistema é uma aproximação da solução da EDP (FORTUNA, 2000).

Para discretizar a Equação de Poisson e as condições de contorno de Neumann, precisamos expressar as derivadas primeira e segunda em função dos pontos da malha. Uma das estratégias possíveis é truncar a expansão em série de Taylor da função incógnita (FORTUNA, 2000; MICHELETTI, 2021; STRIKWERDA, 1989). Dessa forma, temos

respectivamente para as derivadas primeira e segunda de uma função $u(x)$ as seguintes aproximações por diferenças finitas centradas de segunda ordem:

$$\frac{du}{dx} \Big|_{x=x_i} \approx \frac{u(x_i + h) - u(x_i - h)}{2h}; \quad (4)$$

$$\frac{d^2u}{dx^2} \Big|_{x=x_i} \approx \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2}, \quad (5)$$

onde $h = x_i - x_{i-1} = x_{i+1} - x_i$.

Uma vez aplicada a aproximação (5) à equação de Poisson, ao isolarmos $u(x_i)$ obtemos uma EDF que deve ser aplicada aos pontos da malha, gerando um sistema de equações lineares. Solucionamos esse sistema empregando o método SOR (*Successive Over-Relaxation*) (BURDEN; FAIRES; BURDEN, 2016; STRIKWERDA, 1989), cuja relação de recorrência é dada por:

$$u(x_i)^{(k)} = (1 - \omega)u(x_i)^{(k-1)} + \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij}u(x_j)^{(k)} - \sum_{j=i+1}^n a_{ij}u(x_j)^{(k-1)} \right], \quad (6)$$

onde k indica a iteração e a_{ii} e b_i são, respectivamente, os elementos da matriz dos coeficientes A e do vetor dos termos independentes b no sistema linear $Au(x) = b$.

Na Equação (6), que estabelece um processo iterativo que deve ser executado até que uma precisão prefixada ϵ seja alcançada, $\omega > 1$ é o parâmetro de sobre-relaxação empregado para acelerar a convergência do método de Gauss⁸-Seidel⁹. Este método é definido pela Equação (6) com $\omega = 1$. Pode-se mostrar que o método de Gauss-Seidel converge se a matriz dos coeficientes A no sistema linear $Au(x) = b$ é estritamente diagonal dominante, ou seja,

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|, \forall i = 1, 2, \dots, n. \quad (7)$$

A condição dada pela desigualdade (7), denominada critério das linhas, é suficiente à convergência do método de Gauss-Seidel (BURDEN; FAIRES; BURDEN, 2016). Há outros critérios de convergência para esse método (KALABA; SPINGARN, 1978; GARCIA; HUMES; STERN, 2003).

⁸Carl Friedrich Gauss (1777-1855): matemático alemão, um dos mais influentes da história da matemática.

⁹Philipp Ludwig von Seidel (1821-1896): matemático alemão.

2.1.1 Equação de Poisson 2D

A equação de Poisson bidimensional é dada por:

$$\nabla u = \frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = f(x, y), \quad (x, y) \in \Omega \subset \mathbb{R}^2 \text{ e } f(x, y) \neq 0; \quad (8)$$

$$u(x, y) = g(x, y), \text{ se } (x, y) \in \partial\Omega,$$

onde $\partial\Omega$ representa a fronteira do domínio Ω .

A discretização da Equação (8) com diferenças centradas de segunda ordem, definidas pela aproximação (5), resulta em:

$$\frac{\partial^2}{\partial x^2} u(x, y)|_{i,j} + \frac{\partial^2}{\partial y^2} u(x, y)|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} \approx f_{i,j}, \quad (9)$$

onde $i, j = (x_i, y_j) \in \Omega \cup \partial\Omega$.

A expressão discreta (9) define um estêncil de cinco pontos para o cálculo de $u_{i,j}$, ilustrado na Figura 3(a).

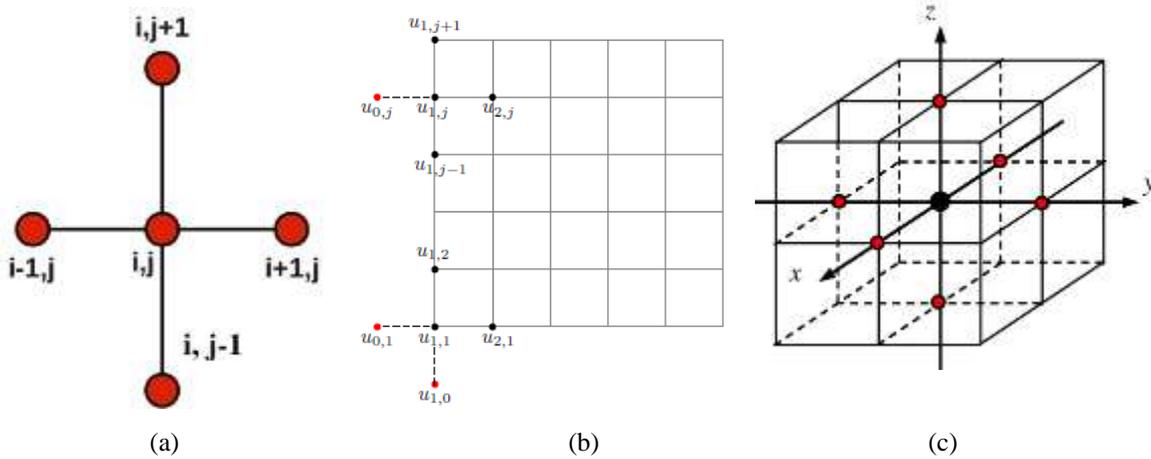


Figura 3 – Discretização centrada de segunda ordem: (a) estêncil de cinco pontos bidimensional; (b) pontos fantasmas; (c) estêncil de sete pontos tridimensional
 Fonte: (a) Williams (2011); (b) Chen (2020); (c) Um (2013).

Isolando $u_{i,j}$ na equação discreta (9) e considerando $\frac{\Delta x}{\Delta y} = \varepsilon$, obtemos que:

$$u_{i,j} = \frac{1}{2(1 + \varepsilon^2)} [u_{i+1,j} + u_{i-1,j} + \varepsilon^2 u_{i,j+1} + \varepsilon^2 u_{i,j-1} - (\Delta x)^2 f_{i,j}]. \quad (10)$$

A Equação (10) deve ser aplicada a cada ponto do domínio discreto, gerando um sistema de equações lineares. Esse sistema pode ser solucionado por intermédio do método SOR (6), cuja equação iterativa é dada por:

$$u_{i,j}^{(k+1)} = (1 - \omega)u_{i,j}^{(k)} + \frac{1}{2(1 + \varepsilon^2)} [u_{i+1,j}^{(k)} + u_{i-1,j}^{(k+1)} + \varepsilon^2 u_{i,j+1}^{(k)} + \varepsilon^2 u_{i,j-1}^{(k+1)} - (\Delta x)^2 f_{i,j}]. \quad (11)$$

Quando agregamos à Equação (11) condições de contorno de Dirichlet, a matriz A no sistema linear $Au(x) = b$ é simétrica, diagonal dominante e definida positiva (CHENG, 2020). Já quando utilizamos condições de contorno de Neumann, com a derivada primeira na fronteira aproximada por diferenças finitas centradas de segunda ordem dada por (4), a matriz A é simétrica e semidefinida positiva (CHENG, 2020). Neste caso, precisamos determinar valores para $u_{i,j}$ em pontos externos ao domínio computacional, denominados pontos fantasmas, como ilustra a Figura 3(b).

2.1.2 Equação de Poisson 3D

A equação de Poisson tridimensional é dada por:

$$\begin{aligned} \nabla u &= \frac{\partial^2}{\partial x^2} u(x, y, z) + \frac{\partial^2}{\partial y^2} u(x, y, z) + \frac{\partial^2}{\partial z^2} u(x, y, z) = f(x, y, z); \\ (x, y, z) &\in \Omega \subset \mathbb{R}^3 \text{ e } f(x, y, z) \neq 0; \\ u(x, y, z) &= g(x, y, z), \text{ se } (x, y, z) \in \partial\Omega. \end{aligned} \quad (12)$$

Aplicando à Equação (12) diferenças centradas de segunda ordem, definidas pela aproximação (5), temos que:

$$\begin{aligned} \frac{\partial^2}{\partial x^2} u(x, y, z)|_{i,j,k} + \frac{\partial^2}{\partial y^2} u(x, y, z)|_{i,j,k} + \frac{\partial^2}{\partial z^2} u(x, y, z)|_{i,j,k} \\ = \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{(\Delta x)^2} + \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{(\Delta y)^2} + \\ + \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{(\Delta z)^2} \approx f_{i,j,k}, \end{aligned} \quad (13)$$

onde $i, j, k = (x_i, y_j, z_k) \in \Omega \cup \partial\Omega$.

A expressão discreta (13) define um estêncil de sete pontos para o cálculo de $u_{i,j,k}$, ilustrado na Figura 3(c).

Isolando $u_{i,j,k}$ na equação discreta (13) e considerando $\frac{\Delta x \Delta y}{\Delta z} = \varepsilon$, obtemos que:

$$u_{i,j,k} = \frac{1}{2[(\Delta x)^2 + (\Delta y)^2 + \varepsilon^2]} [(\Delta y)^2(u_{i+1,j,k} + u_{i-1,j,k}) + (\Delta x)^2(u_{i,j+1,k} + u_{i,j-1,k}) + \varepsilon^2 u_{i,j,k+1} + \varepsilon^2 u_{i,j,k-1} - (\Delta x)^2(\Delta y)^2 f_{i,j,k}]. \quad (14)$$

A Equação (14) deve ser aplicada a cada ponto do domínio discreto, gerando um sistema de equações lineares. Esse sistema pode ser solucionado por intermédio do método SOR (6), cuja equação iterativa é dada por:

$$u_{i,j,k}^{(k+1)} = (1 - \omega)u_{i,j,k}^{(k)} + \frac{\omega}{2[(\Delta x)^2 + (\Delta y)^2 + \varepsilon^2]} \{(\Delta y)^2[u_{i+1,j,k}^{(k)} + u_{i-1,j,k}^{(k+1)}] + (\Delta x)^2[u_{i,j+1,k}^{(k)} + u_{i,j-1,k}^{(k+1)}] + \varepsilon^2 u_{i,j,k+1}^{(k)} + \varepsilon^2 u_{i,j,k-1}^{(k+1)} - (\Delta x)^2(\Delta y)^2 f_{i,j,k}\}.$$

2.2 Soluções manufaturadas 2D E 3D

Uma solução manufaturada para a equação de Poisson $\nabla u = f$ é uma solução construída por intermédio da determinação da função f a partir da atribuição de uma função para u . A partir desta atribuição, deriva-se u duas vezes em relação às variáveis independentes. Na construção da solução manufaturada devemos considerar, além da função u , as condições de contorno.

Nos testes com condições de contorno de Dirichlet, aplicamos o operador Laplaciano discreto nos pontos interiores da malha, mantendo a solução exata nos pontos de fronteira. Já nos testes com condições de contorno de Neumann, aplicamos o operador Laplaciano discreto em todos os pontos da malha. Neste caso, atribuímos valores para os pontos fantasmas isolando-os na aproximação por diferenças centradas de segunda ordem para a derivada primeira, definida em (4). Para os testes com condição de Neumann em todas as fronteiras, a condição de compatibilidade ou de integrabilidade (NÓ S *et al.*, 2005; STRIKWERDA, 1989)

$$\int_{\Omega} f = \int_{\partial\Omega} g,$$

onde $g = \frac{\partial u}{\partial \eta}$ e η é a normal à fronteira, garante que os problemas são bem postos.

Quanto aos testes que realizamos com as soluções manufaturadas, avaliamos os seguintes parâmetros:

1. hx, hy, hz ($\Delta x, \Delta y, \Delta z$): passo espacial nas direções x, y e z , respectivamente;
2. *número de iterações*: número de vezes que o computador precisou executar o código para satisfazer o critério de parada (tolerância de 10^{-6}) do método SOR;
3. *erro absoluto*: diferença entre as soluções exata (manufaturada) e numérica, na norma do máximo ($\|erro\|_{\infty}$) (BURDEN; FAIRES; BURDEN, 2016);
4. *tempo de CPU*: quantos segundos foram necessários para que o computador executasse o código;
5. ω : parâmetro de aceleração de convergência ótimo do método SOR.

2.2.1 Teste1

$$u(x, y) = x^3y^2 + x^2y^3, (x, y) \in [0,1] \times [0,1]$$

$$\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = 2x^3 + 2y^3 + 6x^2y + 6xy^2 \quad (15)$$

O Teste 1 foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Dirichlet, dadas por: $u(x, 0) = 0; u(x, 1) = x^3 + x^2; u(0, y) = 0; u(1, y) = y^3 + y^2$. A Tabela 1 relaciona os resultados numéricos nas malhas 500×300 e 1000×1000 . A Figura 4 ilustra a solução manufaturada (15) e a solução numérica, esta em uma malha 100×100 .

Tabela 1 – Solução numérica da Equação de Poisson no Teste 1 pelo método SOR

Passo espacial: $hx = 0,002$ $hy = 0,003$	
$\omega = 1,9$	
Número de iterações	22781
$\ erro\ _{\infty}$	$2,9074e - 012$
Tempo de CPU (s)	1238
Passo espacial: $hx = hy = 0,001$	
$\omega = 1,9$	
Número de iterações	138985
$\ erro\ _{\infty}$	$1,06273e - 012$
Tempo de CPU (s)	$4,747e + 004$

Fonte: Autores (2021).

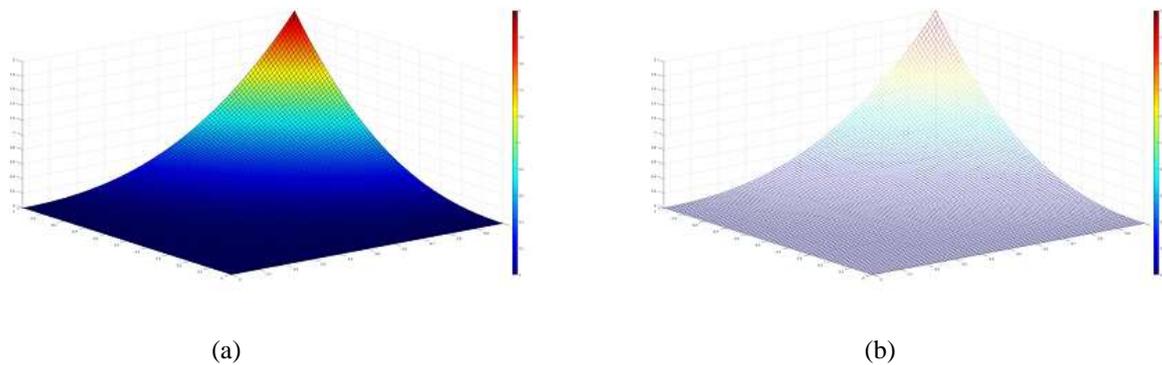


Figura 4 – Equação de Poisson do Teste 1, no domínio $[0,1] \times [0,1]$: (a) solução exata; (b) solução numérica usando SOR, com $hx = hy = 10^{-2}$ e $\omega = 1,9$, em 1244 iterações, $\|erro\ absolute\|_{\infty} = 3,82409e - 012$ e tempo de CPU de 5,638 s

Fonte: Os autores com o Matlab (2021).

2.2.2 Teste 2

$$u(x, y) = e^{\sin(x)+\cos(y)}, (x, y) \in [0,10] \times [0,6]$$

$$\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = [\sin^2(y) + \cos^2(x) - \sin(x) - \cos(y)]e^{\sin(x)+\cos(y)} \quad (16)$$

O Teste 2 foi construído com condições iniciais nulas no interior do domínio e com as seguintes condições de contorno de Dirichlet: $u(x, 0) = e^{\sin(x)+1}$; $u(x, 6) = e^{\sin(x)+\cos(6)}$; $u(0, y) = e^{\cos(y)}$; $u(10, y) = e^{\sin(10)+\cos(y)}$. A Tabela 2 relaciona os resultados numéricos nas malhas 500×300 e 1000×1000 . A Figura 5 ilustra a solução manufaturada (16) e a solução numérica, esta em uma malha 100×100 .

Tabela 2 – Solução numérica da Equação de Poisson no Teste 2 pelo método SOR

Passo espacial: $hx = hy = 0,02$	
$\omega = 1,9$	
Número de iterações	6895
$\ erro\ absolute\ _{\infty}$	$2,29626e - 004$
Tempo de CPU (s)	1402
Passo espacial: $hx = 0,01$ $hy = 0,006$	
$\omega = 1,9$	
Número de iterações	41504
$\ erro\ absolute\ _{\infty}$	$7,31003e - 004$
Tempo de CPU (s)	$1,4554e + 004$

Fonte: Autores (2021).

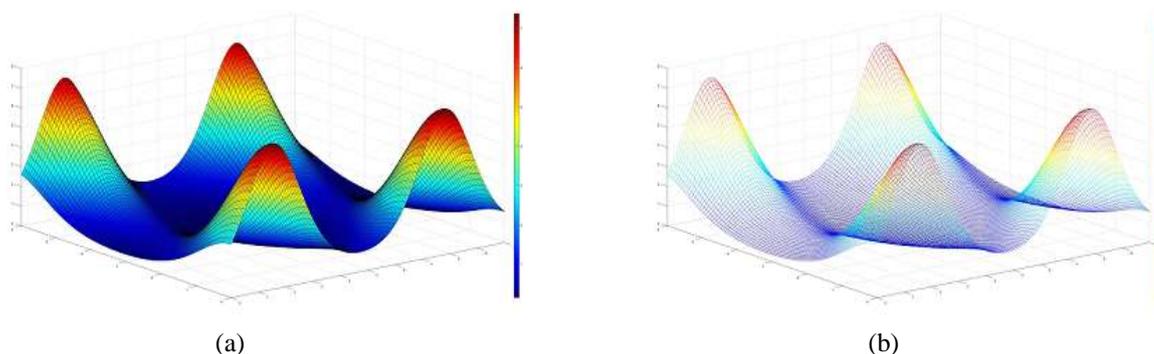


Figura 5 – Equação de Poisson do Teste 2, no domínio $[0,10] \times [0,6]$: (a) solução exata; (b) solução numérica usando SOR, com $hx = 10^{-1}$, $hy = 6 \times 10^{-2}$ e $\omega = 1,9$, em 621 iterações, $\|erro\ absolute\|_{\infty} = 2,32476e - 003$ e tempo de CPU de 2,96 s

Fonte: Os autores com o Matlab (2021).

2.2.3 Teste 3

$$u(x, y) = \cos(2\pi x)\cos(2\pi y), (x, y) \in [0,1] \times [0,1]$$

$$\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = -8\pi^2 \cos(2\pi x)\cos(2\pi y) \quad (17)$$

O Teste 3 foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Neumann, dadas por: $-\frac{\partial}{\partial y} u(x, 0) = \frac{\partial}{\partial y} u(x, 1) = -\frac{\partial}{\partial x} u(0, y) = \frac{\partial}{\partial x} u(1, y) = 0$. A Tabela 3 relaciona os resultados numéricos nas malhas 10×50 e 500×500 . A Figura 6 ilustra a solução manufaturada (17) e a solução numérica, esta em uma malha 100×100 .

Tabela 3 – Solução numérica da Equação de Poisson no Teste 3 pelo método SOR

Passo espacial: $hx = 0,1$ $hy = 0,02$	
$\omega = 1,9$	
Número de iterações	263
Varição relativa máxima no SOR	$9,87492e - 007$
$\ erro\ absolute\ _{\infty}$	$1,71823e - 002$
Tempo de CPU (s)	0,5767
Passo espacial: $hx = hy = 0,002$	
$\omega = 1,9$	
Número de iterações	100001
Varição relativa máxima no SOR	7,0558
$\ erro\ absolute\ _{\infty}$	$1,31596e - 005$
Tempo de CPU (s)	3955

Fonte: Autores (2021).

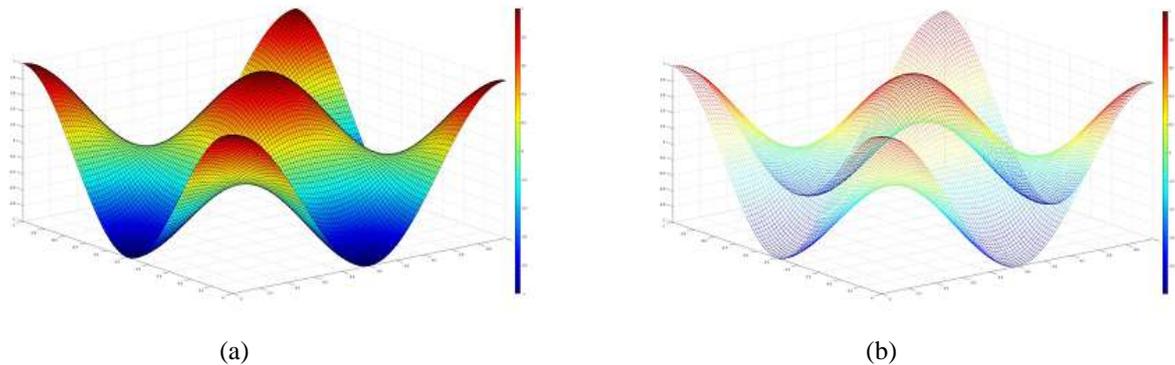


Figura 6 – Equação de Poisson do Teste 3, no domínio $[0,1] \times [0,1]$: (a) solução exata; (b) solução numérica usando SOR, com $hx = hy = 10^{-2}$ e $\omega = 1,9$, em 100001 iterações, $\|erro\ absolute\|_{\infty} = 3,29052e - 004$, variação relativa máxima no SOR de $8,98902e - 001$ e tempo de CPU de 160,6 s
Fonte: Os autores com o Matlab (2021).

2.2.4 Teste 4

$$u(x, y, z) = x^2 e^y + y^2 \text{sen}(z) + z^2 \text{cos}(x), \quad (x, y, z) \in [0,1] \times [0,1] \times [0,1]$$

$$\begin{aligned} \frac{\partial^2}{\partial x^2} u(x, y, z) + \frac{\partial^2}{\partial y^2} u(x, y, z) + \frac{\partial^2}{\partial z^2} u(x, y, z) \\ = 2[e^y + \text{sen}(z) + \text{cos}(x)] - z^2 \text{cos}(x) + x^2 e^y - y^2 \text{sen}(z) \end{aligned}$$

O Teste 4 foi construído com condições iniciais nulas no interior do domínio e com as seguintes condições de contorno de Dirichlet: $u(x, y, 0) = x^2 e^y$; $u(x, y, 1) = x^2 e^y + y^2 \text{sen}(1) + \text{cos}(x)$; $u(0, y, z) = y^2 \text{sen}(z) + z^2$; $u(1, y, z) = e^y + y^2 \text{sen}(z) + z^2 \text{cos}(1)$; $u(x, 0, z) = x^2 + z^2 \text{cos}(x)$; $u(x, 1, z) = x^2 e + \text{sen}(z) + z^2 \text{cos}(x)$. A Tabela 4 relaciona os resultados numéricos nas malhas $10 \times 50 \times 10$ e $50 \times 50 \times 50$. A Figura 7 ilustra a solução numérica em uma malha $100 \times 100 \times 100$.

Tabela 4 – Solução numérica da Equação de Poisson no Teste 4 pelo método SOR

Passo espacial: $hx = hz = 0,1$ $hy = 0,02$	
$\omega = 1,9$	
Número de iterações	179
$\ erro\ absolute\ _{\infty}$	$2,30278e - 005$
Tempo de CPU (s)	0,6102
Passo espacial: $hx = hy = hz = 0,02$	
$\omega = 1,9$	
Número de iterações	219
$\ erro\ absolute\ _{\infty}$	$1,91005e - 006$
Tempo de CPU (s)	18,55

Fonte: Autores (2021).

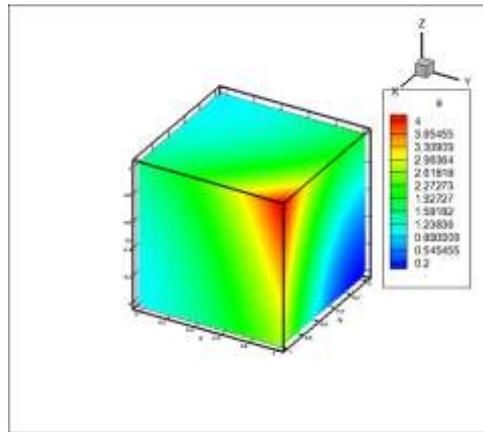


Figura 7 – Equação de Poisson do Teste 4, no domínio $[0,1] \times [0,1] \times [0,1]$: solução numérica usando SOR, com $hx = hy = hz = 10^{-2}$ e $\omega = 1,9$, em 635 iterações, $\|erro\ absolute\|_{\infty} = 7,67582e - 006$ e tempo de CPU de 258,2 s

Fonte: Os autores com o Tecplot 360 (2021).

2.2.5 Teste 5

$$u(x, y, z) = \cos(2\pi x)\cos(2\pi y)\cos(2\pi z), (x, y, z) \in [0,1] \times [0,1] \times [0,1]$$

$$\frac{\partial^2}{\partial x^2} u(x, y, z) + \frac{\partial^2}{\partial y^2} u(x, y, z) + \frac{\partial^2}{\partial z^2} u(x, y, z) = -12\pi^2 \cos(2\pi x)\cos(2\pi y)\cos(2\pi z)$$

O Teste 5 foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Neumann, dadas por: $-\frac{\partial}{\partial z} u(x, y, 0) = \frac{\partial}{\partial z} u(x, y, 1) = -\frac{\partial}{\partial x} u(0, y, z) = \frac{\partial}{\partial x} u(1, y, z) = -\frac{\partial}{\partial y} u(x, 0, z) = \frac{\partial}{\partial y} u(x, 1, z) = 0$. A Tabela 5 relaciona os resultados numéricos nas malhas $50 \times 10 \times 10$ e $80 \times 80 \times 80$. A Figura 8 ilustra a solução numérica em uma malha $50 \times 50 \times 50$.

Tabela 5 – Solução numérica da Equação de Poisson no Teste 5 pelo método SOR

Passo espacial: $hx = 0,02$ $hy = hz = 0,1$	
$\omega = 1,9$	
Número de iterações	295
Varição relativa máxima no SOR	$9,85619e - 007$
$\ erro\ absolute\ _{\infty}$	$2,2583e - 002$
Tempo de CPU (s)	0,9348
Passo espacial: $hx = hy = hz = 0,0125$	
$\omega = 1,9$	
Número de iterações	100001
Varição relativa máxima no SOR	24013,4
$\ erro\ absolute\ _{\infty}$	$5,142e - 004$
Tempo de CPU (s)	$1,575e + 004$

Fonte: Autores (2021).

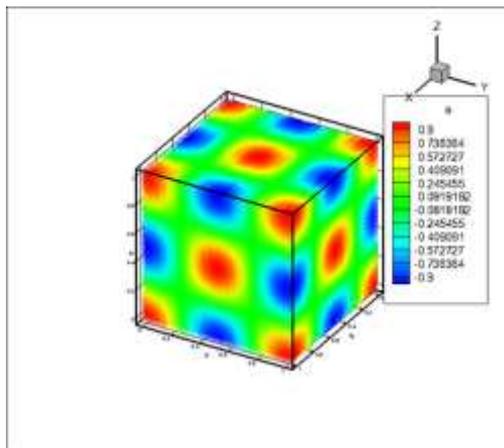


Figura 8 – Equação de Poisson do Teste 5, no domínio $[0,1] \times [0,1] \times [0,1]$: solução numérica usando SOR, com $hx = hy = hz = 2 \times 10^{-2}$ e $\omega = 1,9$, em 620 iterações, $\|erro\ absolute\|_{\infty} = 1,31701e - 003$, variação relativa máxima no SOR de $9,82617e - 007$ e tempo de CPU de 34,43 s

Fonte: Os autores com o Tecplot 360 (2021).

3 RESULTADOS E DISCUSSÃO

Testamos valores para o parâmetro de sobre relaxação ω no SOR variando de 1,1 a 1,9 (FERZIGER, 1981). Constatamos que, com $\omega = 1,6$, o método SOR exigiu menor número de iterações para se alcançar a tolerância em malhas grossas, como 10×10 ou $10 \times 10 \times 10$. Para as malhas testadas mais finas do que as anteriormente citadas, o melhor valor observado foi $\omega = 1,9$ (MICHELETTI, 2021). Esses valores para ω são compatíveis com dados disponíveis na literatura sobre o tema (ONABID, 2012).

Nos testes com condições de contorno de Dirichlet, o método SOR mostrou-se eficiente (número de iterações x variação relativa x erro absoluto x tempo de CPU) em malhas isotrópicas e anisotrópicas. Já nos testes com condições de contorno de Neumann, construídos respeitando-se a condição de compatibilidade $\int_{\Omega} f = \int_{\partial\Omega} g = 0$, o método SOR teve convergência lenta em malhas mais finas, mesmo para $\omega = 1,9$, evidenciada pelo fato de que o número máximo de iterações imposto (10^5) foi atingido sem que a variação relativa imposta (10^{-6}) fosse alcançada. A análise dos testes comprova que o refinamento da malha não necessariamente provoca erros absolutos menores, evidenciando assim que o emprego do método SOR em malhas estruturadas não é uma estratégia adequada para solucionar a Equação de Poisson com quaisquer tipo de condições de contorno.

3.1 Problemas singulares 2D

Um ponto singular, ou singularidade, é um ponto no qual uma função não é definida ou deixa de ser “bem comportada” devido, por exemplo, à falta de diferenciabilidade ou de analiticidade. Desta forma, testamos a associação dos métodos SOR e diferenças finitas para solucionar numericamente problemas singulares 2D em malhas estruturadas. Para tanto, selecionamos três problemas, com solução analítica conhecida, dos propostos por Mitchell (2017) para testar singularidades em malhas adaptativas.

3.1.1 Pico no centro do domínio

Este problema apresenta um pico exponencial no interior do domínio. O par ordenado (x_c, y_c) sinaliza a localização do pico, enquanto o parâmetro α determina a amplitude do pico.

$$u(x, y) = e^{-\alpha[(x-x_c)^2+(y-y_c)^2]}$$

$$\alpha = 1000; (x_c, y_c) = \left(\frac{1}{2}, \frac{1}{2}\right): u(x, y) = e^{-1000[(x-0,5)^2+(y-0,5)^2]}, (x, y) \in [0,1] \times [0,1]$$

$$\begin{aligned} \frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) \\ = e^{-1000(x^2-x+y^2-y+0,5)} [4000000(x^2 - x + y^2 - y) + 1996000] \end{aligned} \quad (18)$$

Este teste foi construído com condições iniciais nulas no interior do domínio e com as seguintes condições de contorno de Dirichlet: $u(x, 0) = e^{-1000[(x-0,5)^2+(-0,5)^2]}$; $u(x, 1) = e^{-1000[(x-0,5)^2+(0,5)^2]}$; $u(0, y) = e^{-1000[(-0,5)^2+(y-0,5)^2]}$; $u(1, y) = e^{-1000[(0,5)^2+(y-0,5)^2]}$. A Tabela 6 relaciona os resultados numéricos nas malhas 500×500 e 1000×1000 . A Figura 9 ilustra a solução exata (18) e a solução numérica, esta em uma malha 1000×1000 .

Tabela 6 – Solução numérica do problema *Pico no centro do domínio* pelo método SOR

Passo espacial: $hx = hy = 0,002$	
$\omega = 1,9$	
Número de iterações	36674
Varição relativa máxima no SOR	$9,99412e - 007$
$\ erro\ absolute\ _{\infty}$	$1,00121e - 003$
Tempo de CPU (s)	3434
Passo espacial: $hx = hy = 0,001$	
$\omega = 1,9$	
Número de iterações	100001
Varição relativa máxima no SOR	$3,74592e - 002$
$\ erro\ absolute\ _{\infty}$	$2,50074e - 004$
Tempo de CPU (s)	$3,68e + 004$

Fonte: Autores (2021).

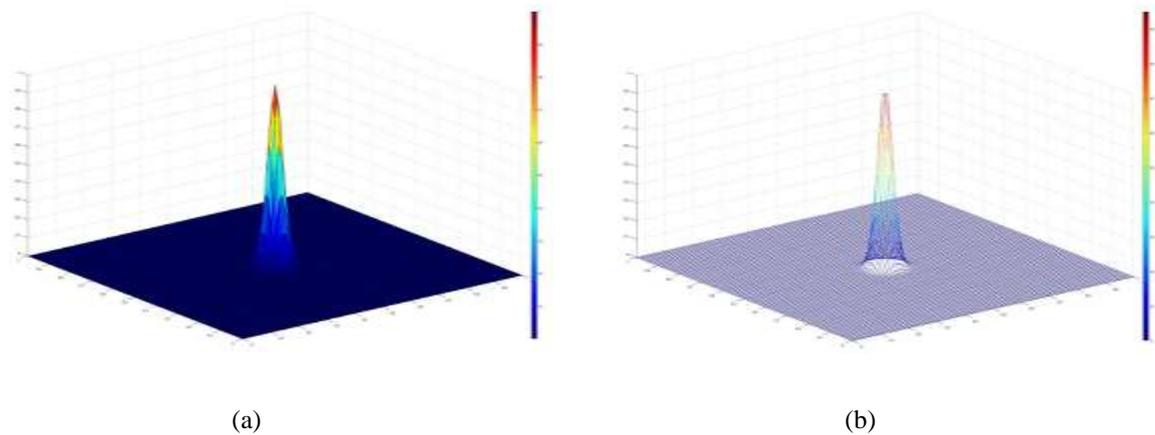


Figura 9 – Pico no centro do domínio $[0,1] \times [0,1]$: (a) solução exata; (b) solução numérica usando SOR, com $hx = hy = 10^{-3}$ e $\omega = 1,9$, em 100001 iterações, $\|erro\|_{\infty} = 2,50074e - 004$, variação relativa máxima no SOR de $3,74592e - 002$ e tempo de CPU de $3,68e + 004$ s
Fonte: Os autores com o Matlab (2021).

3.1.2 Singularidade na fronteira esquerda

De acordo com Mitchell (2017), muitos autores empregam um exemplo 1D com uma singularidade da forma x^{α} no ponto extremo à esquerda do domínio. Esse exemplo pode ser estendido para 2D considerando-se a solução constante em y . No quadrado unitário, o resultado é uma solução que é singular ao longo da fronteira esquerda. O parâmetro $\alpha \geq \frac{1}{2}$ indica a “força” da singularidade.

$$u(x, y) = x^{\alpha}$$

$$\alpha = 0,6: u(x, y) = x^{0,6}, (x, y) \in [0,1] \times [0,1]$$

$$\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = -0,24x^{-1,4} \quad (19)$$

Este teste foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Dirichlet dadas por: $u(x, 0) = x^{0,6}$; $u(x, 1) = x^{0,6}$; $u(0, y) = 0$; $u(1, y) = 1$. A Tabela 7 relaciona os resultados numéricos nas malhas 500×500 e 1000×1000 . A Figura 10 ilustra a solução exata (19) e a solução numérica, esta em uma malha 1000×1000 .

Tabela 7 – Solução numérica do problema *Singularidade na fronteira esquerda* pelo método SOR

Passo espacial: $hx = hy = 0,002$	
$\omega = 1,9$	
Número de iterações	9651
Varição relativa máxima no SOR	$9,99512e - 007$
$\ erro\ absolute\ _{\infty}$	$6,38978e - 003$
Tempo de CPU (s)	2355
Passo espacial: $hx = hy = 0,001$	
$\omega = 1,9$	
Número de iterações	31087
Varição relativa máxima no SOR	$9,99899e - 007$
$\ erro\ absolute\ _{\infty}$	$4,58187e - 003$
Tempo de CPU (s)	4763

Fonte: Autores (2021).

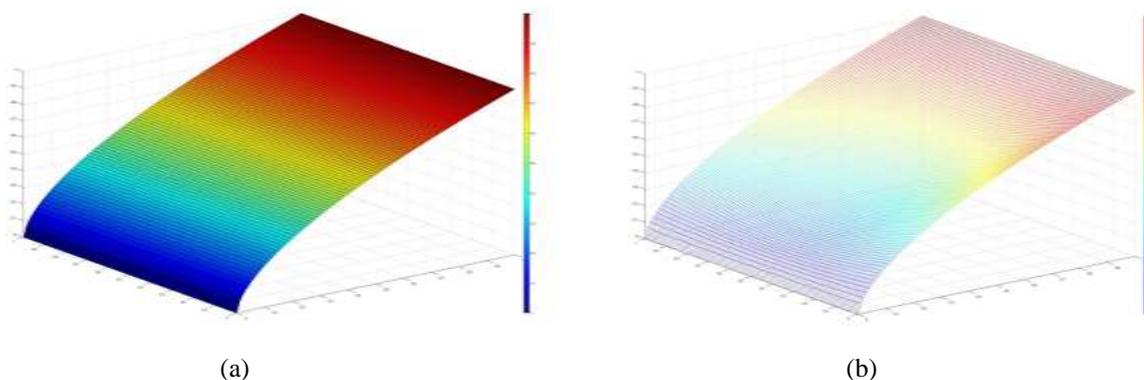


Figura 10 – Singularidade na fronteira esquerda do domínio $[0,1] \times [0,1]$: (a) solução exata; (b) solução numérica usando SOR, com $hx = hy = 10^{-3}$ e $\omega = 1,9$, em 31087 iterações, $\|erro\ absolute\|_{\infty} = 4,58187e - 003$, variação relativa máxima no SOR de $9,99899e - 007$ e tempo de CPU de 4763 s
Fonte: Os autores com o Matlab (2021).

3.1.3 Singularidade interior

Este problema é uma extensão do problema com singularidade na fronteira esquerda, idealizado para definir uma linha singular que não coincida com a fronteira. Os parâmetros α e β determinam, respectivamente, a magnitude da singularidade e a inclinação da linha singular.

$$u(x, y) = \begin{cases} \cos(0,5\pi y), & \text{se } x \leq \beta(y + 1) \\ \cos(0,5\pi y) + [x - \beta(y + 1)]^{\alpha}, & \text{se } x > \beta(y + 1) \end{cases}$$

$$\alpha = 1,1; \beta = 0:$$

$$u(x, y) = \begin{cases} \cos(0,5\pi y), & \text{se } -1 \leq x \leq 0 \\ \cos(0,5\pi y) + x^{1,1}, & \text{se } 0 < x \leq 1 \end{cases}, (x, y) \in [-1,1] \times [-1,1] \quad (20)$$

$$\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = \begin{cases} -0,25\pi^2 \cos(0,5\pi y), & \text{se } -1 \leq x \leq 0 \\ 0,11x^{-0,9} - 0,25\pi^2 \cos(0,5\pi y), & \text{se } 0 < x \leq 1 \end{cases}$$

Este teste foi construído com condições iniciais nulas no interior do domínio e com as seguintes condições de contorno de Dirichlet: $u(x, -1) = \begin{cases} 0, & \text{se } -1 \leq x \leq 0 \\ x^{1,1}, & \text{se } 0 < x \leq 1 \end{cases}$; $u(x, 1) = \begin{cases} 0, & \text{se } -1 \leq x \leq 0 \\ x^{1,1}, & \text{se } 0 < x \leq 1 \end{cases}$; $u(-1, y) = \cos(0,5\pi y)$; $u(1, y) = (0,5\pi y) + 1$. A Tabela 8 relaciona os resultados numéricos nas malhas 500×500 e 1000×1000 . A Figura 11 ilustra a solução exata (20) e a solução numérica, esta em uma malha 1000×1000 .

Tabela 8 – Solução numérica do problema *Singularidade interior* pelo método SOR

Passo espacial: $hx = hy = 0,004$	
$\omega = 1,4$	
Número de iterações	53164
Varição relativa máxima no SOR	$9,99963e - 007$
$\ erro\ absolute\ _{\infty}$	$1,88862e - 001$
Tempo de CPU (s)	2214
Passo espacial: $hx = hy = 0,002$	
$\omega = 1,4$	
Número de iterações	100001
Varição relativa máxima no SOR	$3,85374e - 006$
$\ erro\ absolute\ _{\infty}$	$4,48575e - 002$
Tempo de CPU (s)	$1,654e + 004$

Fonte: Autores (2021).

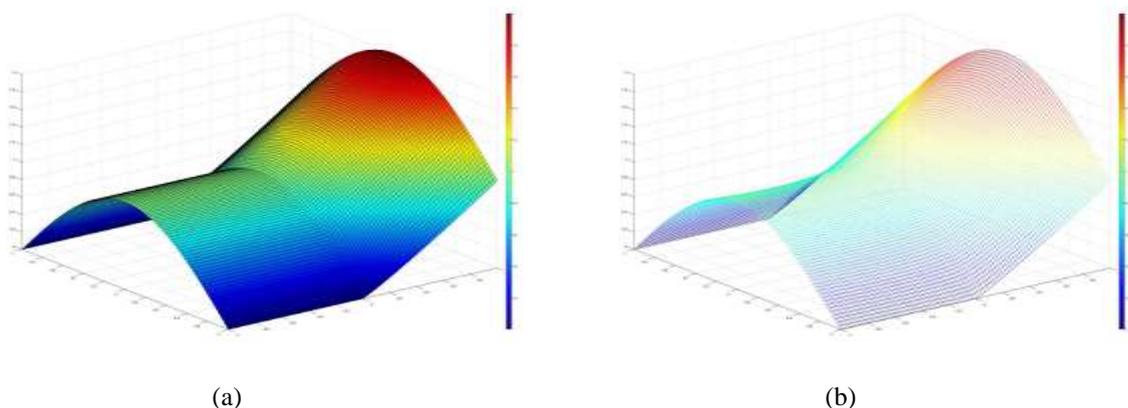


Figura 11 – Singularidade no interior do domínio $[-1,1] \times [-1,1]$: (a) solução exata; (b) solução numérica usando SOR, com $hx = hy = 2 \times 10^{-3}$ e $\omega = 1,4$, em 100001 iterações, $\|erro\ absolute\|_{\infty} = 4,48575e - 002$, variação relativa máxima no SOR de $3,85374e - 006$ e tempo de CPU de $1,654e + 004$ s
Fonte: Os autores com o Matlab (2021).

3.2 Resultados dos testes

Os métodos SOR e diferenças finitas não são eficientes para solucionar numericamente problemas com singularidades fortes em malhas estruturadas, como evidencia o problema singularidade interior. Neste teste, o parâmetro $\omega = 1,4$ gerou menor valor absoluto para o erro.

4 CONCLUSÕES

Apresentamos neste trabalho soluções manufaturadas para a equação de Poisson 2D e 3D. Aproximamos essas soluções numericamente empregando o método de diferenças finitas em malhas estruturadas. Nas aproximações, calculadas através de códigos computacionais construídos em Linguagem C, utilizamos o método iterativo SOR com ω ótimo para solucionar os sistemas lineares provenientes da discretização. Com o auxílio dos softwares Matlab e Tecplot 360, visualizamos as soluções exata e numérica. Testamos ainda o esquema numérico na solução de problemas 2D com singularidades.

Os testes computacionais presentes neste trabalho foram executados em um computador com as seguintes configurações: processador AMD Ryzen 7 3700X; placa de vídeo AMD Radeon RX 5700 XT; 16 Gb de memória RAM 3200 MHz; fonte de 600 W com PFC ativo e certificado 80 plus bronze.

Concluimos que a associação dos métodos SOR e diferenças finitas em malhas estruturadas não é eficiente para solucionar a Equação de Poisson com condições de contorno de Neumann ou com singularidades fortes. Estes problemas podem ser solucionados de forma mais eficiente com o emprego de métodos multigrid em malhas adaptativas com refinamento localizado (NÓ S *et al.*, 2005; CENICEROS; NÓ S; ROMA, 2010; NÓ S; CENICEROS; ROMA, 2012; NÓ S *et al.*, 2017).

REFERÊNCIAS

BURDEN, R. L.; FAIRES, D. J.; BURDEN, A. M. **Análise numérica**. 3. ed. São Paulo: Cengage Learning, 2016.

CENICEROS, H. D.; NÓ S, R. L.; ROMA, A. M. Three-dimensional fully adaptive simulations of phase-field fluid models. **Journal of Computational Physics**, Amsterdam, v. 229, p. 6135-6155, 2010.

CHENG, L. **Finite difference methods for Poisson equation**. 2020. Disponível em: <https://www.math.uci.edu/~chenlong/226/FDM.pdf>. Acesso em: 16 out. 2021.

FERZIGER, J. H. **Numerical methods for engineering application**. New York: John Wiley & Sons, 1981.

FORTUNA, A. de O. **Técnicas computacionais para dinâmica dos fluidos: conceitos básicos e aplicações**. São Paulo: Edusp, 2000.

GARCIA, M. V. P.; HUMES, C.; STERN, J. M. Generalized line criterion for Gauss-Seidel method. **Computational and Applied Mathematics**, New York, v. 22, n. 1, p. 91-97, 2003.

IÓRIO, V. **EDP: um curso de graduação**. Rio de Janeiro: IMPA, 1989.

KALABA, R. E.; SPINGARN, K. A criterion for the convergence of the Gauss-Seidel method. **Applied Mathematics and Computation**, Amsterdam, v. 4, n. 4, p. 359-367, 1978.

LAPLACE, C. **Dev-C++ official website**. 2020. Disponível em: <https://www.bloodshed.net/>. Acesso em: 16 out. 2021.

LOGAN, J. D. **Applied partial differential equations**. 3. ed. New York: Springer, 2015.

MATLAB. **Math Works**. 2021. Disponível em: <https://www.mathworks.com/products/matlab.html>. Acesso em: 16 out. 2021.

MALISKA, C. R. **Transferência de calor e mecânica dos fluidos computacional**. Rio de Janeiro: LTC, 2004.

MICHELETTI, J. P. S. B. **Solução numérica da equação de Poisson em malhas estruturadas bidimensionais e tridimensionais**. 2021. 111 f. Trabalho de Conclusão de Curso (Licenciatura em Matemática) – Universidade Tecnológica Federal do Paraná, Curitiba, 2021.

MITCHELL, W. F. A collection of 2D elliptic problems for testing adaptive grid refinement algorithms. **Preprint submitted to Elsevier**, Amsterdam, p. 1-19, 2017.

NÓS, R. L. *et al.* Three-dimensional coarsening dynamics of a conserved, nematic liquid crystal-isotropic fluid mixture. **Journal of Non-Newtonian Fluid Mechanics**, Amsterdam, v. 248, p. 62-73, 2017.

NÓS, R. L.; CENICEROS, H. D.; ROMA, A. M. Simulação tridimensional adaptativa da separação das fases de uma mistura bifásica usando a equação de Cahn-Hilliard. **TEMA Tendências em Matemática Aplicada e Computacional**, São Carlos, v. 13, p. 37-50, 2012.

NÓS, R. L.; ROMA, A. M.; CENICEROS, H. D. Solução de equações diferenciais parciais elípticas por técnicas multinível-multigrid em malhas tridimensionais bloco-estruturadas com refinamento localizado. *In*: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 28, 2005, Santo Amaro. **Anais[...]**. Santo Amaro: SBMAC, 2005.

ONABID, M. A. Solving three-dimensional (3D) Laplace equations by successive over-relaxation method. **African Journal of Mathematics and Computer Science Research**,

Lagos, v. 5, n. 13, p. 204-208, 2012. Disponível em:
https://academicjournals.org/article/article1379673859_Onabid.pdf. Acesso em: 22 out. 2021.

SCHILDT, H. C. **Completo e total**. 3. ed. São Paulo: Makron Books, 1997.

STRIKWERDA, J. C. **Finite difference schemes and partial differential equations**. New York: Chapman & Hall, 1989.

TEC PLOT. **Tecplot 360**. 2021. Disponível em: <https://www.tecplot.com/products/tecplot-360/>. Acesso em: 16 out. 2021.

UM, S. Y. **Dispersion and local-error analysis of compact LFE-27 formula for obtaining sixth-order accurate numerical solutions of 3D Helmholtz equation**. 2013. Disponível em: https://www.researchgate.net/publication/275876371_Dispersion_and_local-error_analysis_of_compact_LFE-27_formula_for_obtaining_sixth-order_accurate_numerical_solutions_of_3D_Helmholtz_equation/figures?lo=1. Acesso em: 16 out. 2021.

WILLIAMS, G. **Five-point stencil**. 2011. Disponível em:
<https://source.ggy.bris.ac.uk/wiki/File:Five-point-stencil.jpg>. Acesso em: 16 out. 2021.

DADOS DOS AUTORES:

Nome: Rudimar Luiz Nós

E-mail: rudimarnos@gmail.com

Curriculum Lattes: <http://lattes.cnpq.br/4377393528295346>

Pós-doutorado em Matemática Aplicada pela University of California at Santa Barbara, doutorado e mestrado em Matemática Aplicada pela Universidade de São Paulo, especialização em Matemática e Licenciatura em Matemática pela Universidade Regional do Noroeste do Estado do Rio Grande do Sul. Atualmente, é professor titular do Departamento Acadêmico de Matemática da Universidade Tecnológica Federal do Paraná - *Campus* Curitiba. Suas áreas de interesse são: Ensino de Matemática e Análise Numérica com aplicações em Dinâmica dos Fluidos Computacional. Tem experiência na formação de professores de matemática e em Matemática Aplicada, atuando principalmente nos seguintes temas: técnicas multinível-multigrid, equações diferenciais, Modelos B e H, equação de Cahn-Hilliard, malhas refinadas localmente e métodos semi-implícitos.

Nome: João Pedro Santos Brito Micheletti

E-mail: joaomicheletti@alunos.utfpr.edu.br

Curriculum Lattes: <http://lattes.cnpq.br/8333938722545121>

Licenciando em matemática pela Universidade Tecnológica Federal do Paraná, *Campus* Curitiba.