

Análise da eficiência do método SOR na solução da equação de Poisson 2D e 3D

Rudimar Luiz Nós¹

DAMAT/UTFPR, Curitiba, PR

João Pedro Santos Brito Micheletti²

UTFPR, Curitiba, PR

Resumo. Avaliamos neste trabalho a eficiência da combinação dos métodos SOR e diferenças finitas na solução numérica em malhas estruturadas da equação de Poisson, uma equação diferencial parcial elíptica de segunda ordem, com condições de contorno de Dirichlet ou de Neumann. Empregamos os códigos computacionais construídos em linguagem C para testar problemas manufaturados 2D e 3D, assim como problemas singulares 2D. Na visualização das soluções exata e numérica, utilizamos o Matlab e o Tecplot 360. Concluímos que a combinação dos dois métodos, diferenças finitas e SOR, não é eficiente para solucionar as equações de Poisson 2D e 3D com condições de contorno de Neumann, bem como problemas 2D com singularidades fortes.

Palavras-chave. Soluções Manufaturadas, Problemas Singulares, Método de Diferenças Finitas, Malhas Estruturadas, Equações Diferenciais Parciais.

1 Introdução

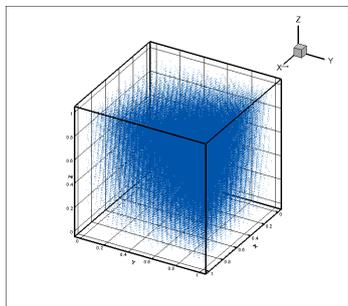


Figura 1: Nuvem de pontos em malha estruturada 3D.

Muitos fenômenos físicos são modelados por equações diferenciais parciais (EDPs) [4]. Esses fenômenos podem ser classificados em dois tipos elementares: os transientes, que evoluem com o tempo; os estacionários, que estão em um estado de equilíbrio e não evoluem com o passar do tempo [3]. Alguns problemas de equilíbrio podem ser modelados por EDPs elípticas de segunda ordem, da forma

$$\nabla^2 u = f, \quad (1)$$

onde $u = u(x_1, \dots, x_n)$, $f = f(x_1, \dots, x_n)$ e ∇^2 denota o operador Laplaciano. Se $f = 0$, a equação (1) é denominada equação de Laplace; caso contrário, é denominada equação de Poisson.

Neste trabalho, investigamos a eficiência da combinação de dois métodos, diferenças finitas [3] e sobre-relaxações sucessivas (SOR: Successive Over-Relaxation) [1, 16], na solução numérica da equação (1) em malhas estruturadas bidimensionais e tridimensionais, esta última ilustrada na Figura 1. A implantação desses dois métodos numéricos é relativamente simples, sendo adequada a iniciantes no estudo de análise numérica. Ao implementá-los, construímos códigos computacionais em linguagem C [15], compilados e executados no Dev-C++ [5], assim como soluções manufaturadas 2D e 3D para testar esses códigos com valores ótimos para o parâmetro ω de sobre-relaxação no método SOR. Empregamos ainda o esquema numérico para solucionar problemas de equilíbrio 2D com singularidades [8], e visualizamos as soluções 2D e 3D utilizando, respectivamente, o Matlab [6] e Tecplot 360 [17].

¹rudimarnos@utfpr.edu.br

²joaomicheletti@alunos.utfpr.edu.br

2 O esquema numérico

As equações iterativas do método SOR aplicado à equação (1) para $n = 2$ e $n = 3$, discretizada com diferenças centradas de segunda ordem [3], são dadas, respectivamente, por:

$$u_{i,j}^{(k+1)} = (1 - \omega)u_{i,j}^{(k)} + \frac{\omega}{2(1 + \epsilon^2)} \left(u_{i+1,j}^{(k)} + u_{i-1,j}^{(k+1)} + \epsilon^2 u_{i,j+1}^{(k)} + \epsilon^2 u_{i,j-1}^{(k+1)} - (\Delta x)^2 f_{i,j} \right); \quad (2)$$

$$\begin{aligned} u_{i,j,k}^{(k+1)} = & (1 - \omega)u_{i,j,k}^{(k)} + \\ & + \frac{\omega}{2((\Delta x)^2 + (\Delta y)^2 + \epsilon^2)} \left((\Delta y)^2 \left(u_{i+1,j,k}^{(k)} + u_{i-1,j,k}^{(k+1)} \right) + (\Delta x)^2 \left(u_{i,j+1,k}^{(k)} + u_{i,j-1,k}^{(k+1)} \right) \right) + \\ & + \frac{\omega}{2((\Delta x)^2 + (\Delta y)^2 + \epsilon^2)} \left(\epsilon^2 u_{i,j,k+1}^{(k)} + \epsilon^2 u_{i,j,k-1}^{(k+1)} - (\Delta x)^2 (\Delta y)^2 f_{i,j,k} \right), \end{aligned} \quad (3)$$

onde: Δx , Δy e Δz são os passos espaciais nas direções x , y e z , respectivamente; $\epsilon = \frac{\Delta x}{\Delta y}$; $\epsilon = \frac{\Delta x \Delta y}{\Delta z}$; $\omega > 1$ é o parâmetro de sobre-relaxação no SOR; $k + 1$, $k = 0, 1, 2, \dots$, indica a iteração em curso.

3 Simulações numéricas

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 com condições de contorno de Neumann, aplicamos o operador Laplaciano discreto em todos os pontos da malha. Neste caso, precisamos calcular valores para pontos fantasmas. Para atribuir valores para esses pontos, empregamos a discretização centrada de segunda ordem para a derivada primeira. Para o método SOR, adotamos uma variação relativa de 10^{-6} e um número máximo de iterações de 10^5 .

3.1 Soluções manufaturadas 2D e 3D

Uma solução manufaturada para a equação (1) 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 em relação às variáveis independentes [9, 12].

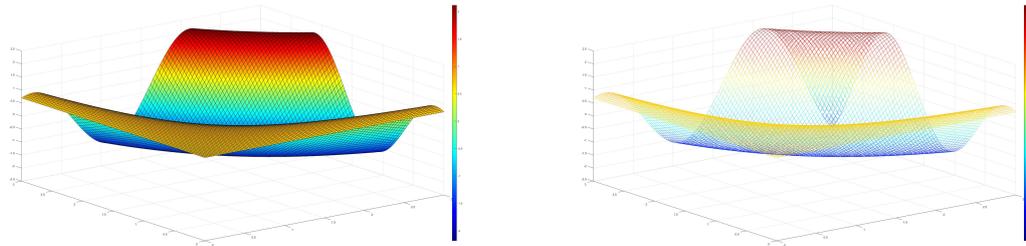
Problema 3.1. *Poisson 2D com condições de contorno de Dirichlet.*

$$\begin{aligned} u(x, y) = \ln(xy + 2)\cos(xy), \quad (x, y) \in [0, 3] \times [0, 3] \quad (4) \\ \frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) = -(x^2 + y^2)\cos(xy)\ln(xy + 2) - \frac{2(x^2 + y^2)\sen(xy)}{xy + 2} - \frac{(x^2 + y^2)\cos(xy)}{(xy + 2)^2} \end{aligned}$$

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

Tabela 1: Solução numérica do Problema 3.1 pelo método SOR, com $\omega = 1,9$.

Número de iterações	$\ erro\ absoluto \ _{\infty}$	Tempo de CPU (s)
$\Delta x = 0,006$	$\Delta y = 0,01$	
16958	$6,49659e - 005$	5785 s
$\Delta x = 0,003$	$\Delta y = 0,003$	
79519	$7,90026e - 006$	$2,621e + 004$ s



(a)

(b)

Figura 2: Solução manufaturada do Problema 3.1: (a) solução exata; (b) solução numérica usando SOR, com $\Delta x = \Delta y = 3 \times 10^{-2}$, $w = 1,9$, 838 iterações, $\| erro\ absoluto \|_{\infty} = 7,86052e - 004$ e tempo de CPU de 3,11 s.

Problema 3.2. *Poisson 3D com condições de contorno de Dirichlet.*

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

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

Este teste foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Dirichlet, dadas por: $u(x, y, 0) = \text{sen}(x)y + x$; $u(x, y, 1) = \text{sen}(x)y + \text{cos}(y) + ex$; $u(0, y, z) = \text{cos}(y)z$; $u(1, y, z) = \text{sen}(1)y + \text{cos}(y)z + e^z$; $u(x, 0, z) = z + e^z x$; $u(x, 1, z) = \text{sen}(x) + \text{cos}(1)z + e^z x$. A Tabela 2 relaciona os resultados numéricos nas malhas $50 \times 10 \times 10$ e $50 \times 50 \times 50$, enquanto a Figura 3(a) ilustra a solução numérica em uma malha $100 \times 100 \times 100$.

Tabela 2: Solução numérica do Problema 3.2 pelo método SOR, com $\omega = 1,9$.

Número de iterações	$\ erro\ absoluto \ _{\infty}$	Tempo de CPU (s)
$\Delta x = 0,02$	$\Delta y = 0,1$	$\Delta z = 0,1$
175	$6,53754e - 005$	0,7885 s
$\Delta x = 0,02$	$\Delta y = 0,02$	$\Delta z = 0,02$
179	$3,16834e - 006$	7,263 s

Problema 3.3. *Poisson 3D com condições de contorno de Neumann.*

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

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

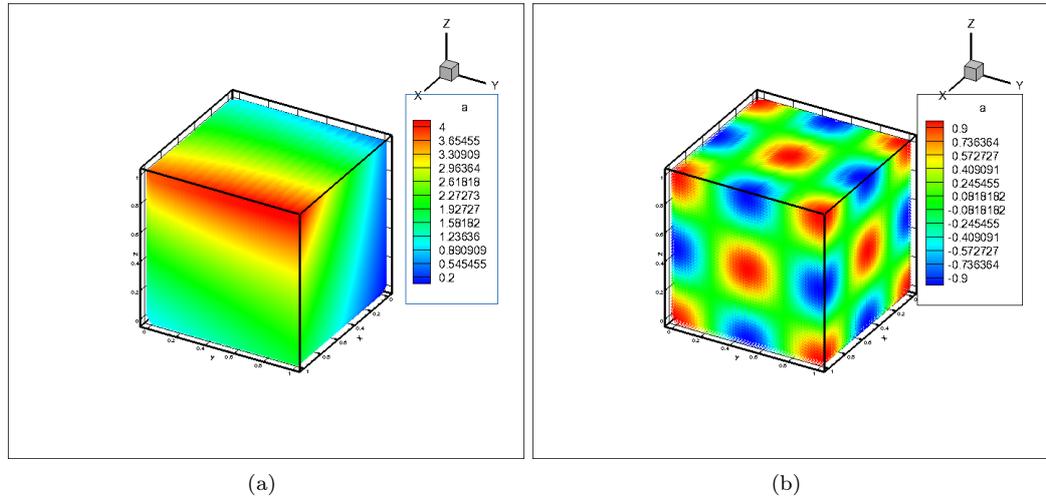


Figura 3: (a) Solução numérica do Problema 3.2 usando SOR, com $\Delta x = \Delta y = \Delta z = 10^{-2}$, $w = 1,9$, 576 iterações, $\| erro absoluto \|_{\infty} = 3,94529e - 005$ e tempo de CPU de 163,4 s; (b) solução numérica do Problema 3.3 usando SOR, com $\Delta x = \Delta y = \Delta z = 2 \times 10^{-2}$, $w = 1,9$, 620 iterações, $\| erro absoluto \|_{\infty} = 1,31701e - 003$, tempo de CPU de 34,43 s e variação relativa máxima no SOR igual a $9,82617e - 007$.

Este teste foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Neumann, dadas por: $-\frac{\partial}{\partial z}u(x, y, 0) = 0$; $\frac{\partial}{\partial z}u(x, y, 1) = 0$; $-\frac{\partial}{\partial x}u(0, y, z) = 0$; $\frac{\partial}{\partial x}u(1, y, z) = 0$; $-\frac{\partial}{\partial y}u(x, 0, z) = 0$; $\frac{\partial}{\partial y}u(x, 1, z) = 0$. A Tabela 3 relaciona os resultados numéricos nas malhas $50 \times 10 \times 10$ e $80 \times 80 \times 80$, enquanto a Figura 3(b) ilustra a solução numérica em uma malha $50 \times 50 \times 50$.

Tabela 3: Solução numérica do Problema 3.3 pelo método SOR, com $\omega = 1,9$.

Número de iterações	$\ erro absoluto \ _{\infty}$	Tempo de CPU (s)	Variação relativa máxima no SOR
$\Delta x = 0,02$	$\Delta y = 0,1$	$\Delta z = 0,1$	
295	$2,2583e - 002$	0,9348 s	$9,85619e - 007$
$\Delta x = 0,0125$	$\Delta y = 0,0125$	$\Delta z = 0,0125$	
100001	$5,142e - 004$	$1,575e + 004$ s	24013,4

3.2 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. Aplicamos o esquema numérico em problemas singulares 2D propostos por [8] para testar singularidades em malhas adaptativas.

Problema 3.4. *Singularidade na fronteira esquerda.*

$$u(x, y) = x^{0,1}, \quad (x, y) \in [0, 1] \times [0, 1] \tag{5}$$

$$\frac{\partial^2}{\partial x^2}u(x, y) + \frac{\partial^2}{\partial y^2}u(x, y) = -0,09x^{-1,9}$$

Este teste foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Dirichlet, dadas por: $u(x, 0) = x^{0,1}$; $u(x, 1) = x^{0,1}$; $u(0, y) = 0$; $u(1, y) = 1$. A Tabela 4 relaciona os resultados numéricos nas malhas 500×500 e 1000×1000 , enquanto a Figura 4 ilustra a solução manufaturada (5) e a solução numérica, esta em uma malha 500×500 .

Tabela 4: Solução numérica do Problema 3.4 pelo método SOR, com $\omega = 1,9$.

Número de iterações	$\ erro\ absoluto \ _{\infty}$	Tempo de CPU (s)	Varição relativa máxima no SOR
$\Delta x = 0,002$	$\Delta y = 0,002$		
9536	$4,51217e - 001$	360 s	$9,99538e - 007$
$\Delta x = 0,001$	$\Delta y = 0,001$		
30685	$4,22729e - 001$	4800 s	$9,99882e - 007$

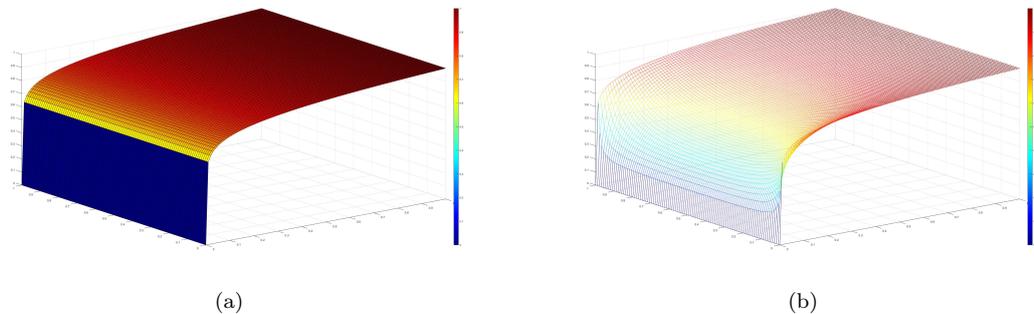


Figura 4: Solução manufaturada do Problema 3.4: (a) solução exata; (b) solução numérica na malha 500×500 .

Problema 3.5. *Singularidade interior.*

$$u(x, y) = \begin{cases} \cos(\pi y/2) & \text{se } -1 \leq x \leq 0 \\ \cos(\pi y/2) + x^{0,1} & \text{se } 0 < x \leq 1 \end{cases} \tag{6}$$

$$\frac{\partial^2}{\partial x^2}u(x, y) + \frac{\partial^2}{\partial y^2}u(x, y) = \begin{cases} -\frac{\pi^2}{4}\cos\left(\frac{\pi y}{2}\right) & \text{se } -1 \leq x \leq 0 \\ -0,09x^{-1,9} - \frac{\pi^2}{4}\cos\left(\frac{\pi y}{2}\right) & \text{se } 0 < x \leq 1 \end{cases}$$

Este teste foi construído com condições iniciais nulas no interior do domínio e com condições de contorno de Dirichlet, dadas por:

$$u(x, -1) = \begin{cases} 0 & \text{se } -1 \leq x \leq 0 \\ x^{0,1} & \text{se } 0 < x \leq 1 \end{cases}; u(x, 1) = \begin{cases} 0 & \text{se } -1 \leq x \leq 0 \\ x^{0,1} & \text{se } 0 < x \leq 1 \end{cases}; u(-1, y) = \cos(\pi y/2);$$

$u(1, y) = \cos(\pi y/2) + 1$. A Tabela 5 relaciona os resultados numéricos nas malhas 500×500 e 1000×1000 , enquanto a Figura 5 ilustra a solução manufaturada (6) e a solução numérica, esta em uma malha 500×500 .

Tabela 5: Solução numérica do Problema 3.5 pelo método SOR, com $\omega = 1,9$.

Número de iterações	$\ erro\ absoluto \ _{\infty}$	Tempo de CPU (s)	Varição relativa máxima no SOR
$\Delta x = 0,004$	$\Delta y = 0,004$		
9289	7,88971	180 s	$9,99656e - 007$
$\Delta x = 0,002$	$\Delta y = 0,002$		
29780	14,4606	4980 s	$9,99981e - 007$

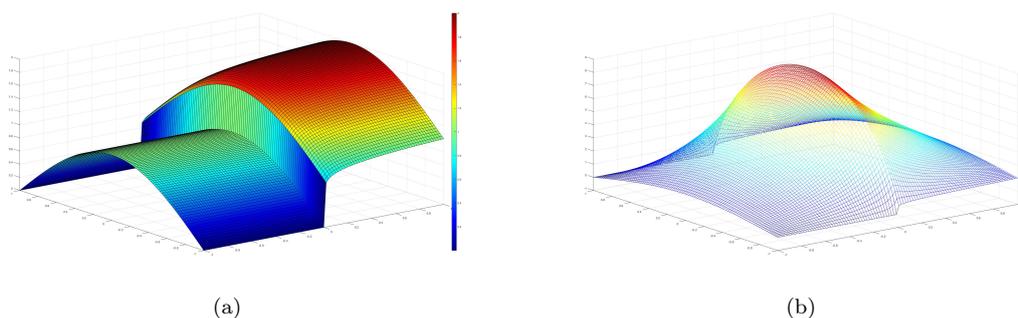


Figura 5: Solução manufaturada do Problema 3.5: (a) solução exata; (b) solução numérica na malha 500×500 .

4 Considerações Finais

A análise dos testes comprova que o refinamento da malha não necessariamente provoca erros absolutos menores. Nos testes para verificar o parâmetro de sobre-relaxação ω ótimo, constatamos que, para $\omega = 1,6$, o método SOR exigiu menor número de iterações para se atingir a variação relativa em malhas grossas, como 10×10 ou $10 \times 10 \times 10$. Em malhas mais finas, o valor ótimo observado foi $\omega = 1,9$ [7]. Esses valores para o parâmetro ω são compatíveis com dados disponíveis na literatura [14].

Na solução numérica da equação (1) com condições de contorno de Dirichlet, o método SOR mostrou-se eficiente (número de iterações \times variação relativa \times erro absoluto \times tempo de CPU) em malhas isotrópicas ($\Delta x = \Delta y = \Delta z$) e anisotrópicas ($\Delta x \neq \Delta y \neq \Delta z$). No teste com condições de contorno de Neumann, construído respeitando-se a condição de compatibilidade $\int_{\Omega} f = \int_{\partial\Omega} g = 0$ [12, 16], onde Ω representa o domínio, $\partial\Omega$ a fronteira do domínio e g as condições de contorno, o método SOR convergiu lentamente, mesmo em malhas mais finas, fato evidenciado pelo número máximo de iterações 10^5 ser empregado sem que a variação relativa 10^{-6} imposta fosse alcançada [7, 11]. Quanto aos testes com singularidades fortes, observamos grandes erros absolutos apesar da variação relativa máxima do método SOR ser inferior ao limite predeterminado (10^{-6}).

Desta forma, concluímos que a associação dos métodos SOR e diferenças finitas em malhas estruturadas não é eficiente para solucionar a equação (1) com condições de contorno de Neumann ou com singularidades fortes. Problemas como esses podem ser eficientemente solucionados com outras técnicas, como métodos multinível-multigrid em malhas adaptativas com refinamento local [2, 10, 12, 13].

Referências

- [1] R. L. Burden, D. J. Faires e A. M. Burden. **Análise numérica**. 3a. ed. São Paulo: Cengage Learning, 2016. ISBN: 978-85-221-2340-7.
- [2] H. D. Cenicerós, R. L. Nós e A. M. Roma. “Three-dimensional fully adaptive simulations of phase-field fluid models”. Em: **Journal of Computational Physics** 229 (2010), pp. 6135–6155. DOI: 10.1016/j.jcp.2010.04.045.
- [3] A. de O. Fortuna. **Técnicas computacionais para dinâmica dos fluidos**. 1a. ed. São Paulo: Edusp, 2000. ISBN: 85-314-0526-2.
- [4] V. Iorio. **EDP: um curso de graduação**. Coleção Matemática Universitária. Rio de Janeiro: IMPA, 1989. ISBN: 85-244-0065-X.
- [5] C. Laplace. **Dev-C++ oficial website**. Online. Acessado em 17/02/2023, <https://www.bloodshed.net>.
- [6] Matlab. **Math Works**. Online. Acessado em 17/02/2023, <https://www.mathworks.com/products/matlab.html>.
- [7] J. P. S. B. Micheletti. “Solução numérica da equação de Poisson em malhas estruturadas bidimensionais e tridimensionais”. UTFPR, Campus Curitiba, 2021.
- [8] W. F. Mitchell. “A collection of 2D elliptic problems for testing adaptive grid refinement algorithms”. Em: **Preprint submitted to Elsevier** (2017), pp. 1–19.
- [9] R. L. Nós. “Simulações de escoamentos tridimensionais bifásicos empregando métodos adaptativos e modelos de campo de fase”. Tese de doutorado. IME/USP, 2007.
- [10] R. L. Nós, H. D. Cenicerós e A. M. Roma. “Simulação tridimensional adaptativa da separação das fases de uma mistura bifásica usando a equação de Cahn-Hilliard”. Em: **Trends in Computational and Applied Mathematics** 13.1 (2012), pp. 37–50. DOI: 10.5540/tema.2012.013.01.0037.
- [11] R. L. Nós e J. P. S. B. Micheletti. “Solução numérica da equação de Poisson 2D e 3D em malhas estruturadas”. Em: **ForScience** 9.2 (2021), pp. 1–21. DOI: 10.29069/forscience.2021v9n2.e1091.
- [12] R. L. Nós, A. M. Roma e H. D. Cenicerós. “Solução de equações diferenciais parciais elípticas por técnicas multinível-multigríd em malhas tridimensionais bloco-estruturadas com refinamento localizado.” Em: **Anais do XXVIII Congresso Nacional de Matemática Aplicada e Computacional**. 2005, pp. 1–7.
- [13] R. L. Nós et al. “Three-dimensional coarsening dynamics of a conserved, nematic liquid crystal-isotropic fluid mixture”. Em: **Journal of Non-Newtonian Fluid Mechanics** 248 (2017), pp. 62–73. DOI: 10.1016/j.jnnfm.2017.08.009.
- [14] M. A. Onabid. “Solving three-dimensional (3D) Laplace equations by successive over-relaxation method”. Em: **African Journal of Mathematics and Computer Science Research** 5.13 (2012), pp. 204–208. DOI: 10.5897/AJMCSR12.042.
- [15] H. Schildt. **C. Completo e total**. 3a. ed. São Paulo: Makron Books, 1997. ISBN: 85-346-0595-5.
- [16] J. C. Strickwerda. **Finite difference schemes and partial differential equations**. 1st. ed. New York: Chapman & Hall, 1989. ISBN: 0-412-07221-1.
- [17] Tecplot. **Tecplot 360**. Online. Acessado em 17/02/2023, <https://www.tecplot.com/products/tecplot-360>.