Geração de Grid - Funções de Controle

Rafael Umino Nakanishi

1 Objetivo

O objetivo desse trabalho foi desenvolver as técnicas de geração da grid baseados em métodos diferenciais elípticos. Nessas técnicas, a malha é gerada através das soluções de Equações Diferenciais Parciais elípticas. Os geradores de malhas mais utilizados são os pares de equação de Laplace:

$$\nabla^2 \xi = 0 \text{ e } \nabla^2 \eta = 0 \tag{1}$$

Tal sistema possui uma características desejáveis, incluindo o princípio do máximo, que garante que nem o máximo nem o mínimo de ξ, η ocorram no inerior do domínio. Além disso, é garantido que se as fronteiras do domínio variam de forma suave e monótona, então o interior da malha também terá essa característica.

1.1 Método de TTM

Em alguns casos, é desejado ainda que certas regiões da malha sejam mais refinados que as demais partes. Para isso faz-se uso das equações de Poisson:

$$\nabla^2 \xi = P(\xi, \eta) \text{ e } \nabla^2 \eta = Q(\xi, \eta) \tag{2}$$

onde $P(\xi, \eta)$ e $Q(\xi, \eta)$ são chamdas funções de controle. É possível demonstrar que as equações do problema inverso são dadas por:

$$\begin{aligned} g_{22}\frac{\partial x}{\partial \xi^2} - 2g_{12}\frac{\partial^2 x}{\partial \xi \partial \eta} + g_{11}\frac{\partial x}{\partial \eta^2} + g\Big(P\frac{\partial x}{\partial \xi} + Q\frac{\partial x}{\partial \eta}\Big) \\ g_{22}\frac{\partial y}{\partial \xi^2} - 2g_{12}\frac{\partial^2 y}{\partial \xi \partial \eta} + g_{11}\frac{\partial y}{\partial \eta^2} + g\Big(P\frac{\partial y}{\partial \xi} + Q\frac{\partial y}{\partial \eta}\Big) \end{aligned} \tag{3}$$

De fato, tomando as relações a seguir:

$$\begin{cases} \nabla \cdot u(x^1, x^2) & = \frac{\partial u}{\partial x^i} g^i \\ \nabla f(x^1, x^2) & = g^i \frac{\partial f}{\partial x^i} \end{cases}$$

é possível mostrar que:

$$\Delta x^i = \nabla \cdot \nabla x^i = g^k \frac{\partial}{\partial x^k}\Big(g^j \frac{\partial x^i}{\partial x^j}\Big)$$

Observe que $\frac{\partial x^i}{\partial x^j} = \delta^i_j$. Aplicando contração de índice, onde $g^j \delta^i_j = g^i$, e a definição do símbolo de Christoffel, temos:

$$\begin{aligned} \Delta x^i &= g^k \frac{\partial g^i}{\partial x^k} \\ &= g^k(-\Gamma^i_{kj} g^j) \\ &= -g^{kj}\Gamma^i_{kj} \end{aligned}$$

Aplicando uma das propriedades do símbolo de Christoffel, chegamos a:

$$\Delta x^i = -g^{kj}\Big(\frac{\partial^2 y_k}{\partial x^j \partial x^k}\frac{\partial x^i}{\partial y_l}\Big) \tag{4}$$

Multiplicando ambos os termos da expressão acima por $\frac{\partial y_l}{\partial x^i}$, obtemos finalmente:

$$\Delta x^i \frac{\partial y_l}{\partial x^i} = -g^{kj}\frac{\partial^2 y_k}{\partial x^j \partial x^k} \tag{5}$$

E aplicando as coordenadas $(x^1, x^2) = (\xi, \eta)$ e $(y_1, y_2) = (x, y)$ obtemos a equação (segue análogo para y):

$$\Delta\xi\frac{\partial x}{\partial \xi} + \Delta\eta\frac{\partial x}{\partial \eta} = -\Big(g^{11}\frac{\partial x}{\partial \xi^2} + 2g^{12}\frac{\partial^2 x}{\partial\xi\partial\eta} + g^{22}\frac{\partial x}{\partial \eta^2}\Big) \tag{6}$$

Note que g^{ij} mapeia das coordenadas curvilpíneas para as coordenadas cartesianas. Como queremoso efeito contrário, encontremos g_{ij}:

$$\begin{bmatrix} g^{11} & g^{12} \\ g^{12} & g^{22} \end{bmatrix} = \frac{1}{g}\begin{bmatrix} g_{22} & -g_{12} \\ -g_{12} & g_{11} \end{bmatrix} \tag{7}$$

E substituindo as Equações 2 e 7 na Equação 6 obtemos as equações do método TTM (Thomson, Thames, Mastin) (Eq. 3).

2 Técnicas Numéricas

Apresentamos nas subseções seguintes as discretizações que foram utilizadas para resolução numérica do problema e como foi feita a escolha dos valores iniciais utilizados.

2.1 Discretização

As principais discretizações realizadas foram feitas por diferenças finitas. Listamos a seguir as discretizações dos tensores métricos e das derivadas parciais de x e y.

$$\begin{aligned}
g_{11} = \Big[\Big(\frac{\partial x}{\partial \xi}^2\Big) + \Big(\frac{\partial y}{\partial \xi}\Big)^2\Big] &= \Big(\frac{x_{i+1,j} - x_{i-1,j}}{2\Delta\xi}\Big)^2 + \Big(\frac{y_{i+1,j} - y_{i-1,j}}{2\Delta\xi}\Big)^2 \\
g_{22} = \Big[\Big(\frac{\partial x}{\partial \eta}^2\Big) + \Big(\frac{\partial y}{\partial \eta}\Big)^2\Big] &= \Big(\frac{x_{i,j+1} - x_{i,j-1}}{2\Delta\eta}\Big)^2 + \Big(\frac{y_{i,j+1} - y_{i,j-1}}{2\Delta\eta}\Big)^2
\end{aligned}$$

$$\begin{aligned}
g_{12} = \Big[\Big(\frac{\partial x}{\partial \xi}\Big)\Big(\frac{\partial x}{\partial \eta}\Big) + \Big(\frac{\partial y}{\partial \xi}\Big)\Big(\frac{\partial y}{\partial \eta}\Big)\Big] = \\
= \Big(\frac{x_{i,j+1} - x_{i,j-1}}{2\Delta\eta}\Big)\Big(\frac{x_{i,j+1} - x_{i,j-1}}{2\Delta\eta}\Big) + \Big(\frac{y_{i,j+1} - y_{i,j-1}}{2\Delta\eta}\Big)\Big(\frac{y_{i+1,j} - y_{i-1,j}}{2\Delta\xi}\Big)
\end{aligned}$$

Destacamos, também as diferenças finitas utilizadas para as segunda derivadas de x (seguindo as equações a seguir análogas para y):

$$\begin{aligned}
\left(\frac{\partial x}{\partial \xi^2}\right) &= \frac{x_{i+1,j} - 2x_{i,j} + x_{i-1,j}}{(\Delta\xi)^2} \\
\left(\frac{\partial x}{\partial \eta^2}\right) &= \frac{x_{i,j+1} - 2x_{i,j} + x_{i,j-1}}{(\Delta\eta)^2} \\
\left(\frac{\partial^2 x}{\partial \xi \partial \eta}\right) &= \frac{x_{i+1,j+1} + x_{i-1,j-1} - x_{i-1,j+1} - x_{i+1,j-1}}{4(\Delta\xi)(\Delta\eta)}
\end{aligned}$$

2.2 Interpolação Transfinita

Para geração dos pontos iniciais na malha, foi utilizado o método de interpolação transfinita. Tal método permite manter a curvatura do domínio físico e gera uma malha de modo rápido e simples. Observe na figura COLOQUE UMA FIGURA que o mapeamento é feito de forma a manter as fronteiras do domínio original.

Para a interpolação fazemos uso de projetores $\mathbf{P}_\xi(\xi,\eta) = (1-\xi)\mathbf{r}(0,\eta) + \xi\mathbf{r}(1,\eta)$ e $\mathbf{P}_\eta(\xi,\eta) = (1-\eta)\mathbf{r}(\xi,0) + \eta\mathbf{r}(\xi,1)$. Utilizado esse projetores é possível preservar os vértices, mas não a fronteira. Então, compondo os projetores mencionados, em uma soma booleana, temos:

$$\begin{aligned}
\mathbf{P}_\xi \oplus \mathbf{P}_\eta(\xi,\eta) &= \mathbf{P}_\xi + \mathbf{P}_\eta - \mathbf{P}_\xi\mathbf{P}_\eta \\
&= (1-\xi)\mathbf{r}(0,\eta) + \xi\mathbf{r}(1,\eta) + (1-\eta)\mathbf{r}(\xi,0) + \eta\mathbf{r}(\xi,1) \\
&\quad -(1-\xi)(1-\eta)\mathbf{r}(0,0) - (1-\xi)\eta\mathbf{r}(0,1) \\
&\quad -(1-\eta)\xi\mathbf{r}(1,0) - \xi\eta\mathbf{r}(1,1)
\end{aligned}$$

Melhorando a escrita da equação acima, escrevemos:

$$\begin{aligned}
\mathbf{r}(\xi,\eta) &= (1-\xi)\mathbf{r}_l(\eta) + \xi\mathbf{r}_r(\eta) + (1-\eta)\mathbf{r}_b(\xi) + \eta\mathbf{r}_l(\xi) - (1-\xi)(1-\eta)\mathbf{r}_b(0) - \\
&\quad (1-\xi)\eta\mathbf{r}_l(0) - (1-\eta)\xi\mathbf{r}_b(1) - \xi\eta\mathbf{r}_l(1)
\end{aligned}$$

onde t, b, l, r são abreviações americanas para *top, bottom, left, right* respectivamente.

2.3 Funções de Controle

Um conjunto de funções de controle foi proposto por Thomson, Thames e Martin em 1974 e pode ser encontrado em [2] e transcrito a seguir:

$$P(\xi,\eta) = -\sum_{n=1}^{N} a_n \frac{(\xi-\xi_n)}{|\xi-\xi_n|} e^{-c_n|\xi-\xi_n|} - \sum_{l=1}^{L} b_l \frac{(\xi-\xi_l)}{|\xi-\xi_l|} e^{-d_l[(\xi-\xi_l)^2+(\eta-\eta_l)^2]^{1/2}} \tag{8}$$

$$Q(\xi,\eta) = -\sum_{n=1}^{N} a_n \frac{(\eta-\eta_n)}{|\eta-\eta_n|} e^{-c_n|\eta-\eta_n|} - \sum_{l=1}^{L} b_l \frac{(\eta-\eta_l)}{|\eta-\eta_l|} e^{-d_l[(\xi-\xi_l)^2+(\eta-\eta_l)^2]^{1/2}} \tag{9}$$

Note que N é o número de linhas e L é o número de pontos para onde a malha será atraída, ou seja, onde haverá maior refinamento. Além disso, os coeficientes a_n, c_n, b_l, d_l são positivos. O primeiro termo da expressão $P(\xi,\eta)$ tem o efeito, com amplitude a_n, de atrair linhas onde ξ é constante para as curvas $\xi = \xi_n$ no domínio físico. Enquanto o segundo termo da expressão, com amplitude b_l, tem o efeito de atrair as linhas ξ para os pontos (ξ_l, η_l).

3 Detalhes de Implementação

Em soluções numéricas a serem utilizadas para resolução de sistemas diferenciais, diferenças finitas frequentemente envolvem equações que geram um sistema de matriz tridiagonal. Com isso, podemos apresentar os métodos de elminação de Gauss: Jacobi, Gauss-Seidel, SOR(Successive Over Relaxation) e Thomas.

Vale ressaltar que os valores iniciais utilizados foram gerados utilizando-se interpolação transfinita, apresentada anteriormente.

Discutimos a seguir os detalhes de implementação pertinentes às técnicas implementadas.

3.1 Método de Jacobi, Gauss Seidel e SOR

Os métodos de Jacobi, Gauss Seidel e SOR são bem semelhantes e optamos por explicá-los em conjunto.

3.1.1 Método Jacobi

Para o método de Jacobi foi necessário criar uma matriz auxiliar para armazenamento dos valores da iteração corrente. Essa abordagem é própria do método, já que faz uso dos valores da iteração anterior para cálculo dos novos valores. Em termos simples, temos que o método de Jacobi pode ser escrito como:

$$x_i^{(k)} = \sum_{j=1}^{N} a_{ij} x_j^{(k-1)}$$

onde k é o valor da iteração corrente.

3.1.2 Método de Gauss Seidel

A criação de uma nova matriz não se faz necessária no método de Gauss Seidel, uma vez que os valores já calculados podem ser utilizados na iteração corrente. Além disso é possível notar uma certa aceleração do método. Podemos descrever o método de Gauss Seidel, em termos simples, como sendo:

$$x_i^{(k)} = \sum_{j=1}^{i-1} a_{ij} x_j^(k) + \sum_{j=i+1}^{N} a_{ij} x_j^{(k-1)}$$

3.1.3 Método SOR

O método de SOR é baseado no método de Gauss Seidel e são utilizados para acelerar a convergência de sistemas que convergem com o método de Gauss. Podemos, então reescrever o método de Gauss da seguinte forma:

$$x_i^{(k)} = (1-\omega) x_i^{(k)} + \omega \left[\sum_{j=1}^{i-1} a_{ij} x_j^(k) + \sum_{j=i+1}^{N} a_{ij} x_j^{(k-1)} \right] \tag{10}$$

onde $0 \leq \omega \leq 2$.

3.2 Método de Thomas

Um sistema tipico a ser resolvido pelo método de Thomas, que é baseado no método de Gauss Seidel, é apresentado da forma

$$-a_i u_{i-1} + b_i u_i - c_i u_{i+1} = d_i \tag{11}$$

com $i = 1...(N-1)$ e os valores de u_0 e u_N são supostamente conhecidos, dados as condições de contorno. Observe que os valores a, b e c são correspondentes à diagonal da matriz tridiagonal

$$\begin{bmatrix} b_1 & -c_1 & 0 & \dots & 0 \\ -a_2 & b_2 & -c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \dots & \vdots \\ 0 & \dots & -an_{N-2} & -b_{N-2} & c_{N-2} \\ 0 & \dots & \dots & -a_{N-1} & b_{N-1} \end{bmatrix}$$

Na implementação, os valores de a, b, c, d podem ser derivados e obtemos os seguintes valores:

$$\begin{aligned}
a_{i,j} &= c_{i,j} = \frac{(g_{22})_{i,j}}{(\Delta\xi)^2} - \frac{gP(i,j)}{(\Delta\xi)^2} \\
b_{i,j} &= \left[\frac{2(g_{22})_{i,j}}{(\Delta\xi)^2} + \frac{2(g_{11})_{i,j}}{(\Delta\eta)^2}\right] \\
d_{i,j} &= \frac{(g_{11})_{i,j}}{(\Delta\eta)^2}(x_{i+1,j+1} + x_{i,j-1}) - 2(g_{12})_{i,j} \times \frac{x_{i+1,j+1} + x_{i-1,j-1} - x_{i-1,j+1} - x_{i+1,j-1}}{4(\Delta\xi)(\Delta\eta)} \\
&\quad -g * Q(i,j) * (x_{i,j+1} - x_{i,j-1}) \\
e_{i,j} &= \frac{(g_{11})_{i,j}}{(\Delta\eta)^2}(y_{i+1,j+1} + y_{i,j-1}) - 2(g_{12})_{i,j} \times \frac{y_{i+1,j+1} + y_{i-1,j-1} - y_{i-1,j+1} - y_{i+1,j-1}}{4(\Delta\xi)(\Delta\eta)} \\
&\quad -g * Q(i,j) * (y_{i,j+1} - y_{i,j-1})
\end{aligned}$$

conseguindo, assim as equações da forma (11), sendo $e_{i,j}$ para a variável y.

4 Resultados

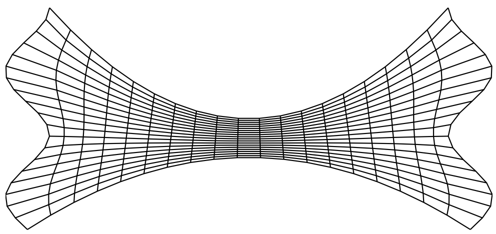
Apresentamos nessa seção os principais resultados obtidos utilizando-se as técnicas descritas anteriormente para alguns domínios físicos criados artificialmente.

As Figuras (1-6) apresentam os domínios utilizados, bem como o erro cometido a cada iteração para seus respectivos domínios. Ainda é possível observar nas figuras a quantidade de iterações que foi necessária para que os métodos convergissem. O erro tolerado nesses casos foi de 10^{-4}.

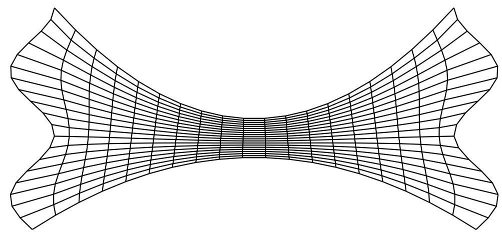
As Figuras (7-12) apresentam os domínios utilizados em conjunto com funções de controle para adaptividade. Em cada figura é possível observar que os pontos da malha foram atraídos para os respectivas localizações impostas. Também é possível verificar o número de iterações necessárias para o método convergir. Observe que é necessário muito mais iterações nesses casos. O erro tolerado também foi de 10^{-4}.

5 Considerações Finais

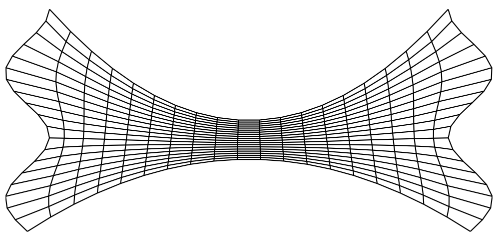
É possível notar que a adaptividade da malha pode ajudar em vários casos onde é necessário mais detalhes em regiões específicas. Com a técnica TTM é possível obter boas malhas com segurança de respeito ao domínio imposto.



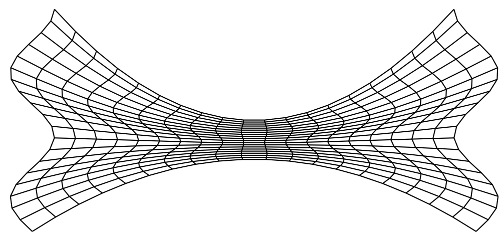
(a) Jacobi - 264 iterações



(b) Gauss-Seidel - 152 iterações



(c) SOR - 59 iterações



(d) Thomas - 163 iterações

Figura 1: Domínio ondulado

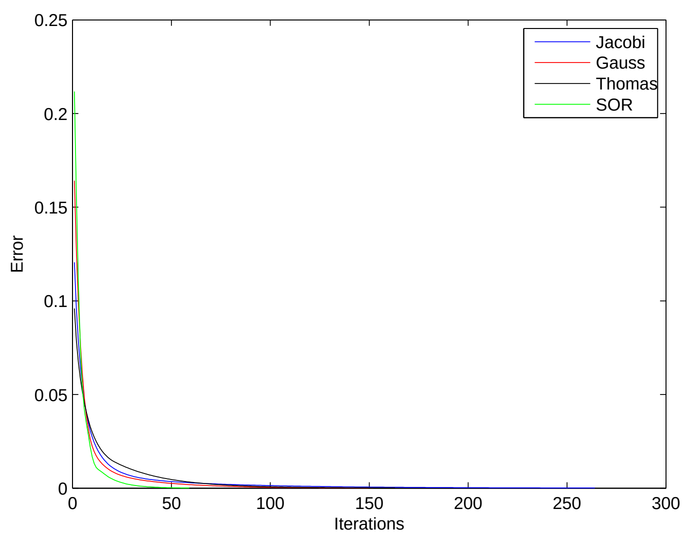
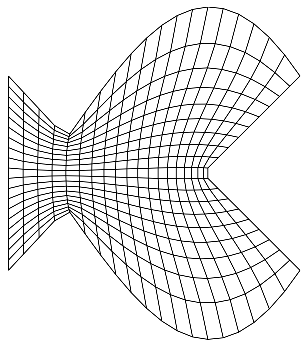
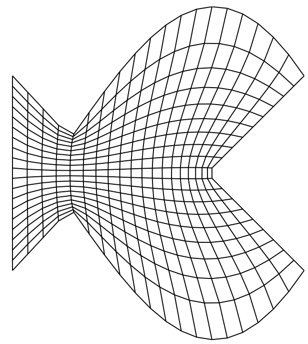


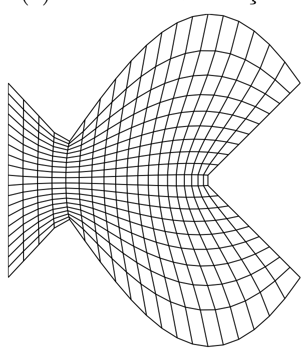
Figura 2: Gráfico do erro para o domínio ondulado



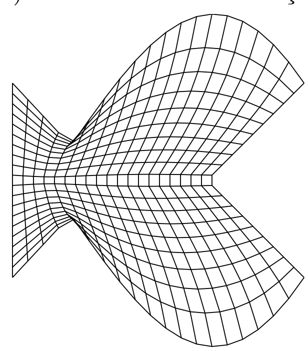
(a) Jacobi - 378 iterações



(b) Gauss-Seidel - 207 iterações



(c) SOR - 74 iterações



(d) Thomas - 120 iterações

Figura 3: Domínio peixe

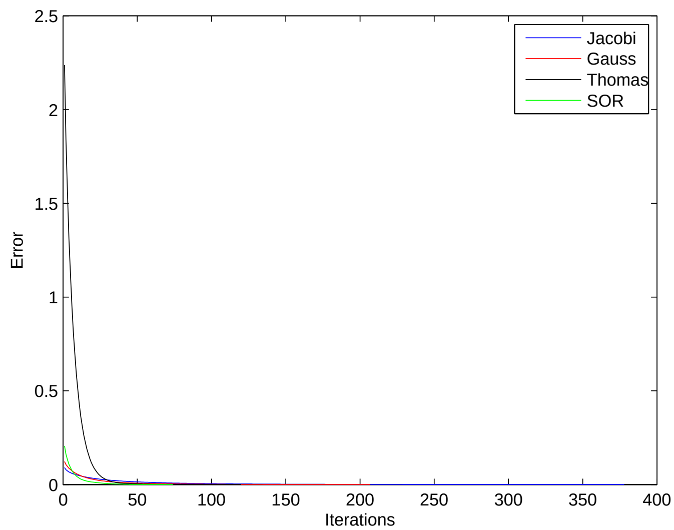
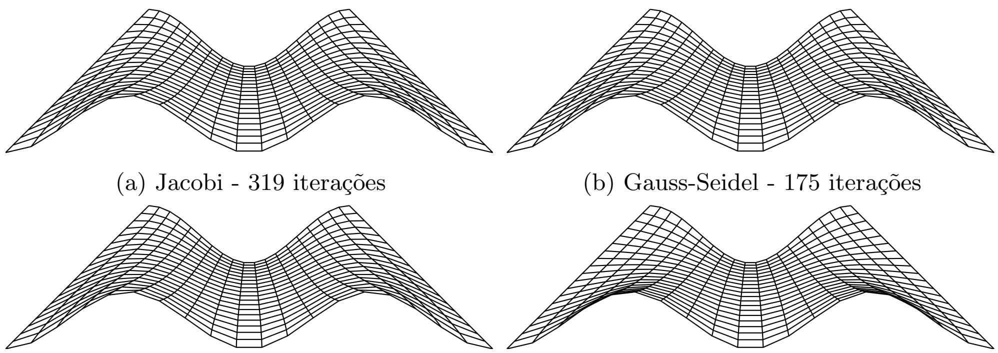


Figura 4: Gráfico do erro para o domínio peixe



(a) Jacobi - 319 iterações

(b) Gauss-Seidel - 175 iterações

(c) SOR - 66 iterações

(d) Thomas - 199 iterações

Figura 5: Domínio M

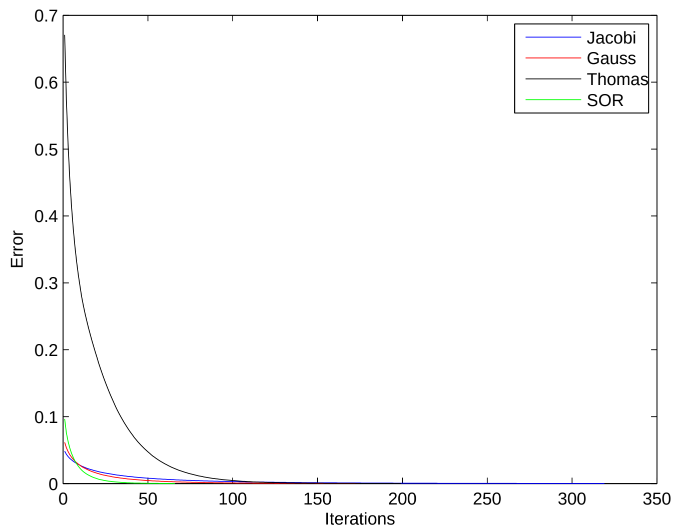
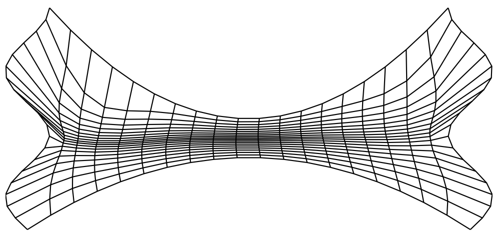
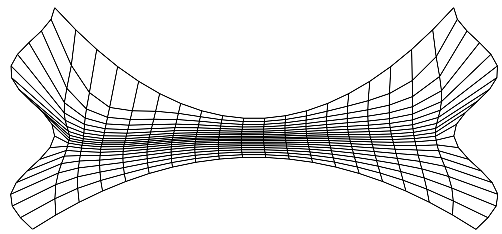


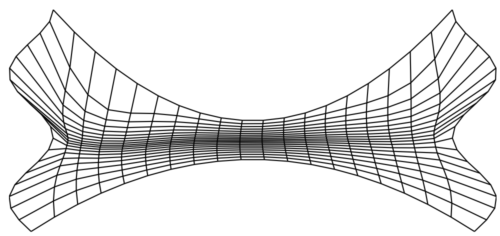
Figura 6: Gráfico do erro para o domínio M



(a) Jacobi - 915 iterações



(b) Gauss-Seidel - 574 iterações



(c) SOR - 238 iterações

Figura 7: Domínio Ondas, aplicando o método TTM para aproximação para os pontos (0.1,0.5) e (0.9,0.5)

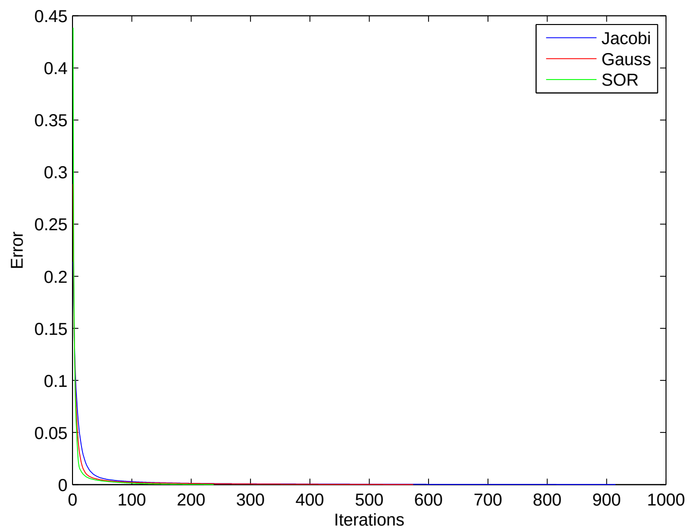
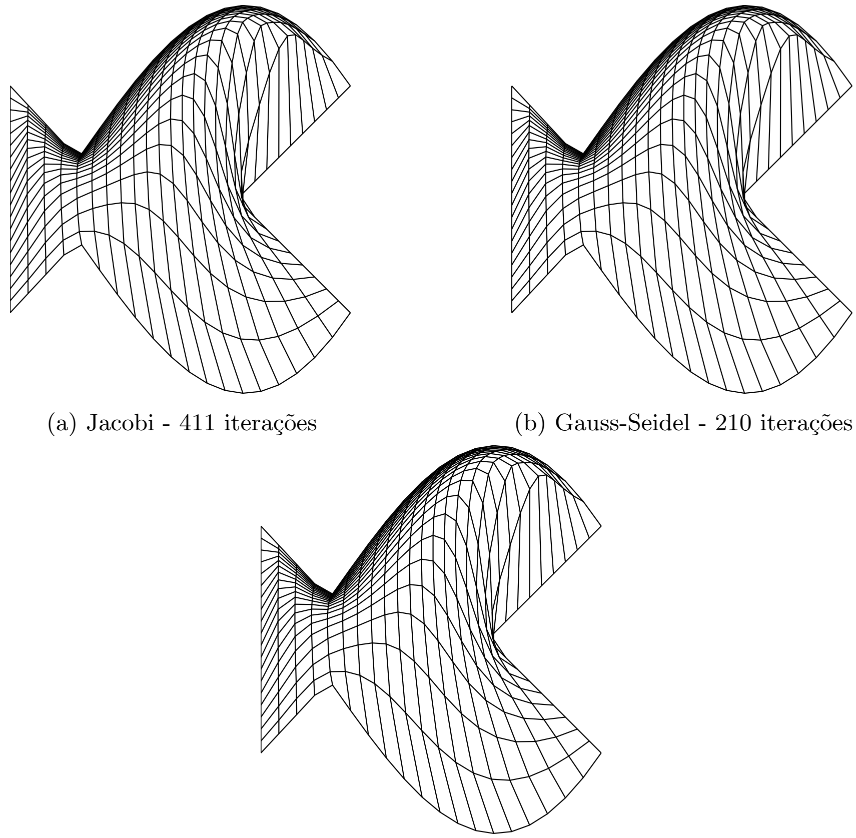


Figura 8: Gráfico do erro para o domínio ondas com aproximação pontual



(a) Jacobi - 411 iterações

(b) Gauss-Seidel - 210 iterações

(c) SOR - 71 iterações

Figura 9: Domínio Peixe, aplicando o método TTM para aproximação para a linha $\xi = 1$

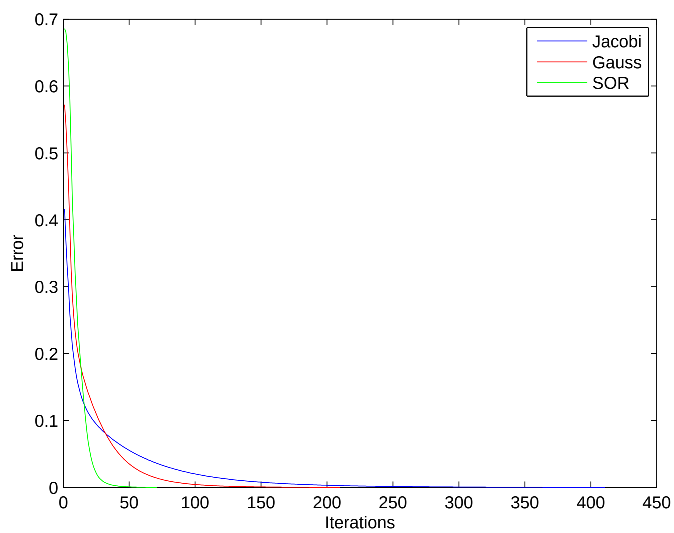
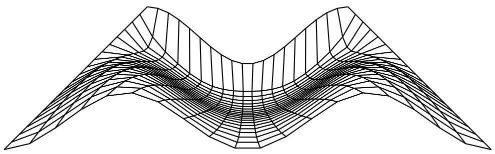
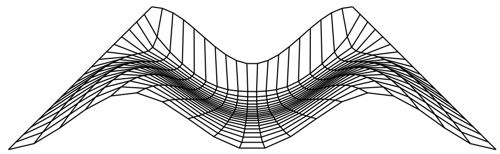


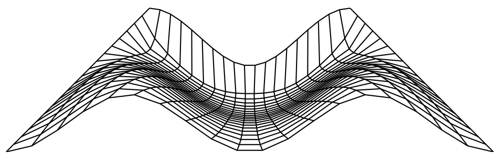
Figura 10: Gráfico do erro para o domínio peixe com aproximação em linha



(a) Jacobi - 1875 iterações



(b) Gauss-Seidel - 119 iterações



(c) SOR - 447 iterações

Figura 11: Domínio M, aplicando o método TTM para aproximação para o ponto (0.5,0.5)

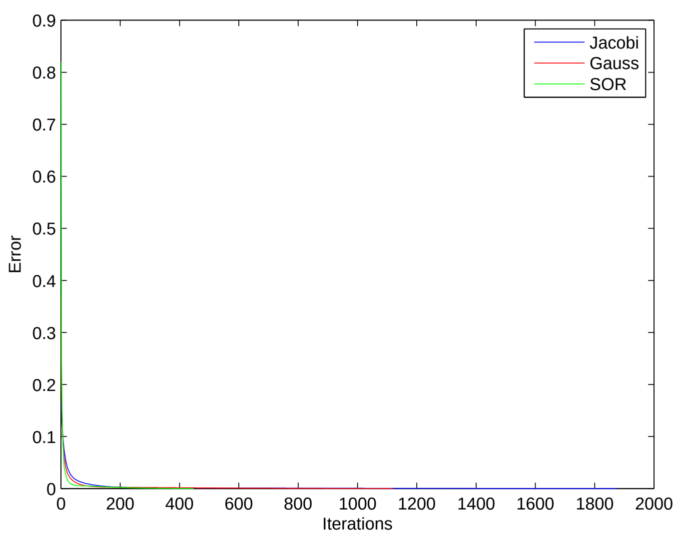


Figura 12: Gráfico do erro para o domínio M com aproximação pontual

Vale ressaltar que a amplitude escolhida nos casos relatados foi exaustivamente testada. Em casos extremos, é possivel degenerar a malha, levando a resultados muito ruins. O bom senso na escolha da amplitude se faz necessário, evitando-se assim valores muito extremos.

Referências

[1] Burden, R. L., Faires, J. D., and Tasks, A. *Análise numérica.* Cengage Learning, 2008.

[2] Farrashkhalvat, M., and Miles, J. *Basic Structured Grid Generation: With an introduction to unstructured grid generation.* Butterworth-Heinemann, 2003.