MAP2320 - Métodos Numéricos em EDPs
Potencial eletrostático entre placas paralelas

30 de outubro de 2019

1 Objetivos

Os objetivos deste exercício computacional são:

- Apresentar a modelagem do campo eletrostático entre duas placas paralelas;
- Implementar o método das diferenças finitas para a equação de Poisson no quadrado unitário;
- Comparar os métodos de Jacobi, Gauss-Seidel e SOR para a resolução dos sistemas.

2 Modelagem do problema

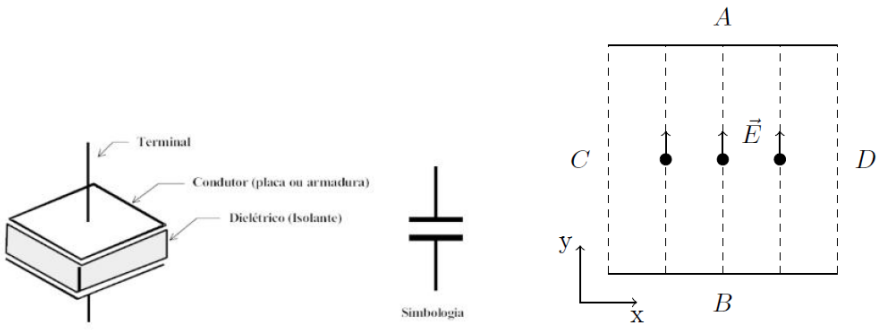
Sejam A e B duas placas paralelas planas, de área $L \times L$ e colocadas a uma distância L uma da outra. Sendo conhecidos os potenciais eletrostáticos V_A e V_B , vamos analisar como obter uma estimativa do potencial V em cada ponto no interior da região entre as duas placas. Para simplificarmos a modelagem, consideramos que os potenciais na região fora da placa são nulos, de modo que pela simetria do problema é possível analisar apenas uma secção transversal do domínio, conforme a Figura 1.

Sendo ε o coeficiente de permissividade elétrica do meio, $\rho_v = \rho_v(x, y)$ a densidade de carga total em um ponto (x, y) entre as placas e $\vec{E}$ o campo elétrico entre as placas, pela Lei de Maxwell temos que (Sadiku, 2010):

$$\nabla \cdot (\varepsilon \vec{E}) = \rho_v \tag{1}$$

Além disso, pela Lei de Coulomb, temos que o campo elétrico $\vec{E}$ é contrário ao gradiente do potencial, isto é

$$\vec{E} = -\nabla V \tag{2}$$



(a) Placas planas (b) Secção Transversal

Figura 1: Placas e seção transversal

Substituindo (2) em (1), e considerando o meio homogêneo (ε constante entre as placas) temos que:

$$\nabla \cdot (\varepsilon \vec{E}) = \nabla \cdot (-\varepsilon \nabla V) = -\varepsilon \Delta V = \rho_v$$

Considerando $f(x,y) = \frac{\rho_v}{\varepsilon}$ e $\Omega =]0, L[\times]0, L[$, temos o seguinte problema de Poisson:

$$\begin{cases} -\Delta V = f \text{ em } \Omega \\ V = v_0 \text{ em } \partial\Omega \end{cases} \tag{3}$$

onde v_0 é o potencial no contorno que, neste caso, é dado por:

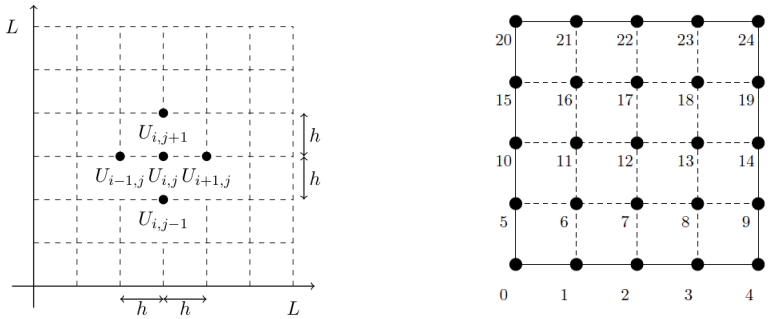
$$v_0(x,y) = \begin{cases} V_A(x,L), 0 \leq x \leq L \\ V_B(x,0), 0 \leq x \leq L \\ V_C(0,y), 0 \leq y \leq L \\ V_D(0,y), 0 \leq y \leq L \end{cases}$$

onde V_C e V_D são definidas de forma a tornar o problema bem posto.

3 Discretização

Por conveniência de notação, vamos descrever a solução do problema (3) como $u(x,y)$ ao invés de $V(x,y)$. Também vamos descrever a condição de fronteira como $g(x,y)$ ao invés de $v_0(x,y)$. Dado um número natural $N \in \mathbb{N}$, consideramos o espaçamento $h = \frac{L}{N}$ de modo que o domínio discreto da equação é descrito pelo conjunto:

$$\overline{\Omega}_h = \{(x_i, y_j) : x_i = ih, 0 \leq i \leq N, y_j = jh, 0 \leq j \leq N\}$$



(a) Malha

(b) Ordem lexográfica

Figura 2: Domínio discreto

A fronteira e o interior deste conjunto são respectivamente definidos por:

$$\partial\Omega_h = \{(x_i, y_j) = (ih, jh) : i \in \{0, N\} \text{ ou } j \in \{0, N\}\}$$

$$\Omega_h = \{(x_i, y_j) : x_i = ih, 1 \leq i \leq N-1, y_j = jh, 1 \leq j \leq N-1\}$$

Nosso objetivo é aproximar u nos pontos $(x_i, y_j) \in \Omega_h$. Para os índices iguais a 0 ou a N (isto é, em $\partial\Omega_h$) a condição de contorno do problema é fornecida e portanto conhecemos u nesses pontos. Denotamos por u_{ij} a aproximação da função u no ponto $x_i = ih, y_j = jh$ para $i, j = 1, 2, \cdots, N-1$, isto é, $u_{ij} \approx u(x_i, y_j)$. Também denotamos $f_{ij} = f(x_i, y_j)$ e $g_{ij} = g(x_i, y_j)$.

O método de diferenças finitas consiste em aproximar as derivadas usando expansões em séries de Taylor truncadas. Para obter a discretização da segunda derivada em x e em y vamos utilizar a combinação das Séries de Taylor truncadas avançada e retrógrada:

$$u(x_{i+1}, y_j) = u(x_i, y_j) + hu_x(x_i, y_j) + \frac{h^2}{2}u_{xx}(x_i, y_j) + \frac{h^3}{3!}u_{xxx}(x_i, y_j) + \frac{h^4}{4!}u_{xxxx}(\alpha_i, y_j) \tag{4}$$

$$u(x_{i-1}, y_j) = u(x_i, y_j) - hu_x(x_i, y_j) + \frac{h^2}{2}u_{xx}(x_i, y_j) - \frac{h^3}{3!}u_{xxx}(x_i, y_j) + \frac{h^4}{4!}u_{xxxx}(\beta_i, y_j) \tag{5}$$

Onde $\alpha_i \in]x_{i-1}, x_i[$ e $\beta \in]x_i, x_{i+1}[$. Somando (4) e (5), obtemos:

$$u_{xx}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_i, y_j)}{h^2} + \tau_x(x_i, y_j) \tag{6}$$

Onde $\tau_x(x_i, y_j) = \mathcal{O}(h^2)$ é o erro da discretização local cometido por esta aproximação no ponto (x_i, y_j). Analogamente, obtemos que a segunda derivada em relação a y é dada por:

$$u_{yy}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2} + \tau_y(x_i, y_j) \tag{7}$$

Substituindo (6) e (7) em (3) temos:

$$\frac{-u(x_{i-1},y_j)-u(x_i,y_{j-1})+4u(x_i,y_j)-u(x_{i+1},y_j)-u(x_i,y_{j+1})}{h^2}=f(x_i,y_j)+\tau_h(x_i,y_j) \tag{8}$$

Onde $\tau_h(x_i,y_j)=\tau_x(x_i,y_j)+\tau_y(x_i,y_j)=\mathcal{O}(h^2)$.

Ignorando o erro de discretização local, temos que as aproximações nos pontos interiores satisfazem a seguinte relação:

$$\frac{-u_{i-1,j}-u_{i,j-1}+4u_{i,j}-u_{i+1,j}-u_{i,j+1}}{h^2}=f(x_i,y_j) \tag{9}$$

Podemos ainda reescrever esta equação como um sistema linear contendo $(N-1)^2$ equações e $(N-1)^2$ incógnitas, dado por:

$$\begin{aligned} -u_{i-1,j}-u_{i,j-1}+4u_{i,j}-u_{i+1,j}-u_{i,j+1}=h^2f_{ij},\ \text{para } i,j=1,\cdots,N-1\ (\text{em }\Omega_h)\\ u_{ij}=g_{ij},\ \text{para } i\in\{0,N\}\ \text{ou}\ j\in\{0,N\}\ (\text{em }\partial\Omega_h)\end{aligned} \tag{10}$$

de forma que, utilizando a enumeração lexicográfica (veja na Figura 2), temos que as aproximações são obtidas através da resolução do seguinte sistema linear.

Para ilustrar, se tomarmos $N=4$, teríamos que o nosso sistema poderia ser escrito matricialmente como:

$$\left(\begin{array}{ccc|ccc|ccc} 4&-1&0&-1&0&0&0&0&0\\ -1&4&-1&0&-1&0&0&0&0\\ 0&-1&4&0&0&-1&0&0&0\\ \hline -1&0&0&4&-1&0&-1&0&0\\ 0&-1&0&-1&4&-1&0&-1&0\\ 0&0&-1&0&-1&4&0&0&-1\\ \hline 0&0&0&-1&0&0&4&-1&0\\ 0&0&0&0&-1&0&-1&4&-1\\ 0&0&0&0&0&-1&0&-1&4 \end{array}\right)\left(\begin{array}{c}u_6\\u_7\\u_8\\u_{11}\\u_{12}\\u_{13}\\u_{16}\\u_{17}\\u_{18}\end{array}\right)=\left(\begin{array}{c}h^2f_6+u_1+u_5\\h^2f_7+u_2\\h^2f_8+u_3+u_9\\h^2f_{11}+u_{10}\\h^2f_{12}\\h^2f_{13}+u_{14}\\h^2f_{16}+u_{15}+u_{21}\\h^2f_{17}+u_{22}\\h^2f_{18}+u_{19}+u_{23}\end{array}\right)$$

No caso para N qualquer, o sistema pode ser escrito matricialmente como:

$$\left(\begin{array}{ccccccc} T&-I&&&&\cdots&0\\ -I&T&-I&&&\ddots&\vdots\\ &-I&T&-I&&&\\ &&\ddots&\ddots&\ddots&&\\ &&&-I&T&-I&\\ \vdots&\ddots&&&-I&T&-I\\ 0&\cdots&&&&-I&T \end{array}\right)\left(\begin{array}{c}U_1\\\vdots\\\vdots\\\vdots\\\vdots\\U_{N-1}\end{array}\right)=\left(\begin{array}{c}F_1\\\vdots\\\vdots\\\vdots\\\vdots\\F_{N-1}\end{array}\right) \tag{11}$$

Onde I denota a matriz identidade $(N-1)\times(N-1)$ e T é a matriz tridiagonal $(N-1)\times(N-1)$ dada por:

$$T = \begin{pmatrix} 4 & -1 & & & & \\ -1 & 4 & -1 & & & \\ & -1 & 4 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 4 & -1 \\ & & & & -1 & 4 \end{pmatrix} \tag{12}$$

E também temos para $k = 1, \cdots, N-1$:

$$U_k = (u_{k1}, u_{k2}, \cdots, u_{k,n-1}) \in \mathbb{R}^{N-1} \tag{13}$$

Naturalmente, as questões a serem abordadas a partir de agora são:

- Este sistema sempre possui solução?
- Qual a maneira mais eficiente de resolver o sistema associado?

Uma outra questão seria se as aproximações obtidas convergem para a solução do problema quando $h \to 0$. Isto de fato ocorre e a convergência é de ordem 2. Não iremos demonstrar isto aqui, indicamos (Strikwerda, 1990) para o leitor interessado.

4 Existência de Solução do Sistema Linear

Considere as seguintes definições:

Definição 4.1. Uma matriz $A = [a_{i,j}]$ $n \times n$ é dita irredutível se não existir uma matriz de permutação P tal que:

$$PAP^T = \begin{pmatrix} B & C \\ 0 & D \end{pmatrix}$$

onde $k < n$, $B = [b_{ij}]$ $k\times$, $C = [c_{ij}]$ $k\times(n-k)$ e $D = [d_{ij}]$ e $(n-k)\times(n-k)$. Ou seja, a matriz não pode ser decomposta de forma que um conjunto de variáveis fique independente das outras.

Definição 4.2. Seja n um inteiro. Uma matriz $A = [a_{ij}]$ $n \times n$ é dita diagonal dominante se:

$$|a_{ii}| > \sum_{j=1, i\neq i}^{n} |a_{ij}|, \forall i = 1, \cdots, n$$

e é dita fracamente diagonal dominante se:

$$|a_{ii}| \geq \sum_{j=1, i\neq i}^{n} |a_{ij}|, \forall i = 1, \cdots, n$$

e vale a desigualdade estrita para alguma linha.

Teorema 4.1. *(Existência de Solução do sistema). Toda matriz diagonal dominante é inversível e toda matriz fracamente diagonal dominante e irredutível é inversível.*

Demonstração. Seja A uma matriz diagonal dominante e suponha, por absurdo, que $\det(A) = 0$. Então existe um vetor $x \neq 0$ tal que $Ax = 0$. Seja i o índice tal que:

$$|x_i| = M = \max_j |x_j| > 0$$

Temos que a i-ésima linha do sistema $Ax = 0$ é dada por:

$$\sum_{j=1}^{n} a_{ij}x_j = 0 \Rightarrow x_i = -\sum_{j\neq i}^{n} \frac{a_{ij}}{a_{ii}}x_j$$

Tomando o módulo dos dois lados, temos que:

$$|x_i| = \left|\sum_{j\neq i}^{n} \frac{a_{ij}}{a_{ii}}x_j\right| \leq \sum_{j\neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|}|x_j| \leq M \underbrace{\sum_{j\neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|}}_{<1} < M \text{ (absurdo!)}$$

Logo $\det(A) \neq 0$. No caso em que a matriz é fracamente diagonal dominante, note que

$$\sum_{j\neq i}^{n} \frac{|a_{ij}|}{|a_{ii}|} = M \iff |x_i| = M \text{ quando } a_{ij} \neq 0$$

Como a matriz é irredutível, podemos reordenar os índices onde $|x_j| = M$ de forma a colocá-los em sequência crescente. Como existe pelo menos uma linha onde a desigualdade é estrita, nesta linha caímos novamente no caso anterior. □

Observe no exemplo para $N = 4$ (página 4) que os pontos próximos da fronteira possuem a desigualdade estrita enquanto que o ponto interior u_{12} possui a entrada da diagonal igual a soma dos outros elementos da linha 12. Além disso, é possível mostrar que a matriz do sistema (11) (ou equivalentemente, do sistema (10)) é irredutível, de forma que o resultado a seguir garante a existência e unicidade de solução para os sistemas em estudo.

5 Métodos Iterativos para a resolução de sistemas lineares

Antes de iniciarmos o estudo de métodos iterativos para a resolução de sistemas lineares, uma pergunta interessante a ser feita é: Por que utilizar métodos iterativos para resolver sistemas lineares?

Sabemos que as aproximações do problema (3) serão obtidas através da resolução de um sistema linear cuja matriz possui $(N-1)^2$ elementos, onde apenas 5 deles são não nulos em cada uma das linhas. Utilizando o método de Eliminação de Gauss seriam necessárias em torno de $(N-1)^6$ operações para construir as aproximações do problema em estudo.

Para exemplificar o tempo de processamento que seria gasto, considere uma malha com $N = 100$ (ou seja, $h = 0.01$) e que todas as operações realizadas demorem o mesmo tempo de execução de uma multiplicação. Desta forma, utilizando um computador capaz de realizar 10^{10} multiplicações por segundo seriam necessários aproximadamente 95 segundos. Porém, aumentando o valor para $N = 200$ ($h = 0.005$) o tempo necessário aumenta para aproximadamente 1 hora e 43 minutos e para e $N = 500$ ($h = 0.002$) o tempo aumenta para mais de 17 dias. Com isto, podemos concluir que o método de Eliminação de Gauss não é uma boa estratégia para a resolução deste problema. Portanto, iremos procurar por um método com custo computacional baixo e que seja convergente.

Considere $A = [a_{ij}]$ $n \times n$ uma matriz real tal que $\det(A) \neq 0$ e o sistema linear $Ax = b$. Nossa estratégia será escrever $A = M - N$, de modo que seja possível escrever um método iterativo da forma:

$$Mx - Nx = b \Rightarrow Mx^{k+1} = b + Nx^k$$

Considere também a hipótese de que $\det(M) \neq 0$. Assim, caso tal processo seja convergente, então o limite do processo será a solução do sistema $Ax = b$ pois, se

$$\lim_{k\to\infty} x_k = \overline{x} \Rightarrow M\overline{x} = b + N\overline{x} = (M-N)\overline{x} = b \Rightarrow A\overline{x} = b$$

Definição 5.1. Seja A uma matriz diagonalizável. Dizemos que o maior autovalor (em módulo) de uma matriz A é o raio espectral da matriz, denotado por $\rho(A)$.

Proposição 5.1. *(Condição necessária para a convergência dos métodos iterativos) . Seja $A = M - N$, onde A e M são inversíveis. Um método iterativo será convergente se, e somente se, $\rho(M^{-1}N) < 1$.*

Demonstração. Definindo $e_k = \overline{x} - x_k$ temos que

$$Me_{k+1} = Ne_k \Rightarrow e_{k+1} = M^{-1}Ne^k \Rightarrow e_k = (M^{-1}N)^k e_0$$

Logo,

$$e_k \to 0 \iff (M^{-1}N)^k \to 0$$

Mas isto ocorre se, e somente se, $\rho(M^{-1}N)^k < 1 \iff \rho(M^{-1}N) < 1$ □

Escrevendo $A = L + D + U$, onde L é a parte triangular inferior (lower), D a diagonal da matriz e U a parte triangular superior (upper), podemos escrever os métodos de Jacobi e Gauss-Seidel:

- Método de Jacobi: $M = D, N = -(L+U)$. O método será convergente se, e somente se $\rho(D^{-1}(L+U)) < 1$

- Método de Gauss-Seidel: $M = L + D, N = -U$. O método será convergente se, e somente se $\rho((D+L)^{-1}U) < 1$

Gostaríamos agora de saber se estes métodos podem ser aplicados para a resolução do sistema (10). O teorema a seguir fornece algumas condições nas quais os métodos de Jacobi e Gauss-Seidel são convergentes.

Teorema 5.1. *Se A é uma matriz irredutível e fracamente diagonal dominante, então os métodos de Jacobi e Gauss-Seidel são convergentes.*

Demonstração.

- Método de Jacobi: Seja λ um autovalor da matriz $D^{-1}(L+U)$. Temos que:

  $$\det(D^{-1}(L+U)-\lambda I) = 0 \Rightarrow \det(D^{-1}(L+U-\lambda D)) = 0 \Rightarrow \det(D^{-1})\det((L+U-\lambda D)) = 0$$

  Como A é fracamente diagonal dominante e irredutível, temos que $\det(D) \neq 0$ e, portanto, $\det(D^{-1}) \neq 0$. Logo

  $$\det((L+U-\lambda D)) = 0$$

  Caso $|\lambda| \geq 1$ temos que $(L+U-\lambda D)$ é irredutível e fracamente diagonal dominante. Pelo Teorema (4.1), temos que a matriz é inversível e, portanto, $\det((L+U-\lambda D)) \neq 0$. Logo, $|\lambda| < 1$ para todos os autovalores de $D^{-1}(L+U)$ de modo que $\rho(D^{-1}(L+U)) < 1$. Pela Proposição (5.1), temos que o método será convergente.

- Método de Gauss-Seidel: Seja λ um autovalor da matriz $D^{-1}(L+U)$. Temos que:

  $$\det((D+L)^{-1}U - \lambda I) = 0 \Rightarrow \det((D+L)^{-1}(U - \lambda(D+L))) = 0$$

  $$\Rightarrow \det((D+L)^{-1})\det(U - \lambda(D+L)) = 0$$

  Por hipótese $\det((D+L) \neq 0$, de modo que $\det(U - \lambda(D+L)) = 0$. Caso $\lambda \geq 1$ temos que $(U - \lambda(D+L))$ é irredutível e fracamente diagonal dominante. Novamente pelo Teorema (4.1) a matriz $U - \lambda(D+L)$ é inversível e $det(U-\lambda(D+L)) \neq 0$. Logo, $|\lambda| < 1$ para todos os autovalores de $(D+L)^{-1}U$ de modo que

  $$\rho(-(D+L)^{-1}U) < 1$$

  Pela Proposição (5.1), segue a convergência do método.

□

Agora que sabemos que ambos os métodos são convergentes para o sistema (10), seria interessante que pudéssemos estimar a velocidade de convergência de cada um desses métodos. Para isto, considere $\overline{x}$ a solução do sistema $Ax = b$

, x_k a aproximação obtida na k-ésima iteração do método, $\lambda_1 < \lambda_2 < \cdots < \lambda_n = \rho(M^{-1}N)$ e $v_1, v_2, \cdots, v_n$ os autovalores e autovetores da matriz $M^{-1}N$. Assim:

$$x_k - \overline{x} = (M^{-1}N)^k(x_0 - \overline{x})$$

Como os autovetores formam uma base para o $\mathbb{R}^n$, podemos escrever:

$$x_0 - \overline{x} = \sum_{i=1}^{n} c_i v_i$$

Fazendo $k \to \infty$, temos:

$$(M^{-1}N)^k(x_0 - \overline{x}) = \sum_{i=1}^{n} \lambda_i^k c_i v_i = \lambda_n^k \sum_{i=1}^{n-1} \left(\frac{\lambda_i}{\lambda_n}\right)^k c_i v_i + c_n \lambda_n^k v_n$$

Desta forma, podemos concluir que:

$$\|x_{k+1} - \overline{x}\| \approx \rho(M^{-1}N)^k \|x_0 - \overline{x}\|$$

e a taxa de convergência das aproximações é controlada pelo valor de $\rho(M^{-1}N)$. Assim, para que um dado método iterativo apresente rápida convergência para a solução do problema devemos ter que $\rho(M^{-1}N)$ deve ser pequeno o suficiente para que poucas iterações sejam suficientes para fornecer boas aproximações.

Vamos considerar a matriz do sistema (10). Para o método de Jacobi, pode-se mostrar que:

$$\rho(M^{-1}N) = \cos(\pi h) \approx 1 - \frac{\pi^2 h^2}{2} < 1$$

o que mostra que o método de Jacobi possui uma baixa velocidade de convergência. De maneira análoga, obtemos que para o método de Gauss-Seidel

$$\rho(M^{-1}N) = \cos^2(\pi h) \approx 1 - \pi^2 h^2$$

de forma que o método de Gauss-Seidel possui a velocidade de convergência um pouco maior do que o método de Jacobi . Em ambos os casos, quando $h \to 0$ a velocidade de convergência da solução vai diminuindo, pois o raio espectral de ambas as matrizes tendem a 1. Ou seja, quanto mais refinada a malha, menor a velocidade de convergência dos métodos de Jacobi e de Gauss-Seidel.

Visando aumentar a velocidade de convergência vamos inserir um parâmetro livre no problema de forma que com a manipulação desse parâmetro seja possível aumentar a velocidade de convergência dos métodos numéricos. Dado o sistema $Ax = b$ e considerando $\omega \in \mathbb{R}$, vamos escrever a $k+1$-ésima iteração do método como uma combinação ponderada entre a k -ésima aproximação e a $k+1$-ésima aproximação obtida pelo método de Gauss-Seidel. Isto equivale a escrever:

$$x_i^{k+1} = \frac{\omega}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k\right) + (1-\omega)x_i^k$$

Na forma matricial, temos que:

$$(D + \omega L)x^{k+1} = ((1 - \omega)D - \omega U)x^k + \omega b$$

onde a matriz de iteração do método será dada por

$$S = (D + \omega L)^{-1}((1 - \omega)D - \omega U)$$

e o método será convergente se, e somente se, $\rho(S) < 1$. Este método conhecido como Sucessive Over-Relaxation ou SOR. Os resultados a seguir apresentam os resultados de convergência e de velocidade de convergência do método SOR.

As demonstrações não serão feitas aqui, pois utilizam alguns conceitos que fogem ao escopo de uma primeira apresentação. Ao leitor interessado, a referência (Stoer and Bulirsch, 2002) contém as provas dos resultados a seguir:

Teorema 5.2. *Se o método SOR é convergente, então* $0 < \omega < 2$.

Teorema 5.3. *O parâmetro ótimo para o método SOR quando aplicado a matriz do sistema gerado por (8) é dado por:*

$$\omega = \frac{2}{1 + \sin(\pi h)}$$

Com este parâmetro, o raio espectral da matriz de iteração é:

$$\rho(S) = \frac{1 - \sin(\pi h)}{1 + \sin(\pi h)} \approx 1 - 2\pi h$$

Assim como nos métodos de Jacobi e de Gauss Seidel, observa-se que o raio espectral da matriz tende a 1 quando h tende a 0. Porém, neste método a convergência é linear, enquanto que nos outros métodos a convergência é quadrática.

6 Implementação computacional

Na resolução numérica do Problema (3) iremos utilizar no vetor de soluções do sistema dois índices, um representando a linha e outro representando a coluna. A aplicação dos métodos de Jacobi, Gauss-Seidel e SOR seguem a mesma estrutura:

1. Crie matrizes U^{old}, U^{new} e F de dimensões $N \times N$.
2. Inicialize U^{old} com alguma aproximação inicial;
3. Inicialize o lado direito do sistema F com as informações da função f ;
4. Utilize a função g para atribuir os valores aos contornos do domínio em U^{old} (isto é, para os índices $i, j = 0$ ou N);

5. Atualize os pontos internos de U^{new} de acordo com o método utilizado (Jacobi, Gauss Seidel ou SOR);

6. Calcule a diferença entre as duas iterações (ver critério de parada);

7. Se a diferença for maior do que uma tolerância TOL, faça $U^{old} = U^{new}$ retorne ao passo (5). Caso contrário, encerre.

6.1 Critério de parada

Como critério de parada vamos avaliar a diferença entre duas iterações, e paramos se esta diferença for pequena. Assim, dada uma tolerância inicial TOL, iremos considerar como critério de parada a condição

$$\|U_k - U_{k-1}\|_h \leq TOL$$

onde a norma $\|\cdot\|_h$ é dada por:

$$\|U\|_h = h\left[\sum_{i,j=1}^{N-1} |U_{ij}|^2\right]^{\frac{1}{2}}$$

Esta é uma medida de variação quadrática média entre iterações.

Observe que a tolerância TOL pode ser uma estimativa muito pequena, de modo que o algoritmo pode demorar muito para convergir. Desta forma, é aconselhável que seja colocado também uma restrição no número máximo de passos a ser executado pelo código.

7 Tarefa

Você deverá implementar os métodos de Jacobi, Gauss-Seidel e SOR para a equação de Poisson. As linguagens permitidas são: C, Python e Octave. Com o seu programa, você deverá resolver os exercícios abaixo.

Os parâmetros do programa são N, a função f, a condição de fronteira g, a tolerância TOL e o número máximo de iterações $MAXITER$.

Junto com o programa, você deverá entregar um relatório (em pdf) contendo as discussões pedidas nos exercícios.

Caso você queira usar uma outra linguagem ou tenha dúvidas, escreva um email para o monitor (luansantos@ime.usp.br).

7.1 Exercícios

Considere α [1] o último algarismo do seu número USP e $L = 1$.

1. Para testarmos a implementação computacional realizada, considere o problema (3) para

[1] Se ele for zero, considere α como o primeiro dígito diferente de zero do seu número USP

(a) $f = 0$ e $u(x, y) = \alpha e^x \sin(y)$ para $(x, y) \in \partial\Omega$.

(b) $f(x, y) = 2\alpha\pi^2 \cos(\pi x) \sin(\pi y)$ e $u(x, y) = \alpha \cos(\pi x) \sin(\pi y)$ para $(x, y) \in \partial\Omega$.

- Resolva numericamente os problemas *a*) e *b*) para $N = 2^3, 2^4, \cdots, 2^9$ utilizando os métodos de Jacobi, Gauss-Seidel e SOR para a resolução do sistema linear associado. Utilize a aproximação inicial nula e o critério de parada com $TOL = 10^{-5}\ h$.
- Compare a relação entre os valores de N e a quantidade de iterações necessárias (e o tempo computacional) para a resolução utilizando cada um dos métodos.
- Compare as soluções obtidas com a solução exata do problema (observe que a função u descrita no contorno é a própria solução do problema). Isto é, calcule o erro máximo após o método ter convergido:

$$erro = \max_{i,j} |u(x_i, y_j) - U_{ij}|$$

Onde $u(x_i, y_j)$ é a solução exata e U_{ij} é a solução numérica. O que acontece com o erro máximo para cada valor de N? Faça um gráfico ou uma tabela que mostre o erro máximo para cada N.

2. Vamos agora utilizar o algoritmo implementado para a simulação de um modelo real. Considere o problema (3) para $V_A(x, 1) = 110V$, $V_B(x, 0) = 0V$, $V_C(0, y) = V_D(1, y) = 110 \sin(\frac{\pi}{2} y)$ nos seguintes casos:

   - Duas placas contendo ar entre elas (permissividade $\varepsilon \approx 8.84 \times 10^{-12}$) e com densidade constante $\rho_v = 100 \times 10^{-12}$.
   - Duas placas paralelas contendo baquelita (permissividade $\varepsilon \approx 75 \times 10^{-12}$) e com densidade dada por $\rho_v = 10 \sin(\pi(x + y)) \times 10^{-8}$.

   Em ambos casos apresente gráficos das soluções.

3. Nas questões 1 e 2, utilizando o método SOR com

$$\omega = \frac{2}{1 + \sin(\pi h)}$$

compare a quantidade de iterações necessárias (e o tempo computacional) em relação aos métodos de Jacobi e de Gauss-Seidel. Teste para outros valores de ω. O que pode-se observar?

7.2 Observações sobre a implementação computacional

1. Toda a notação apresentada no enunciado considera que os índices começam em 0 (zero). Caso queira utilizar uma linguagem interpretada onde os índices comecem em 1 (um) tenha muito cuidado com os índices. Utilize precisão dupla em todos os cálculos.

2. Antes de realizar o método iterativo, deve-se inicializar os valores das fronteiras $U_{0,i}, U_{i,0}, U_{i,N}$ e $U_{N,i}$, $i = 1, \cdots, N$ utilizando os valores de $u(x, y)$ conhecidos em $\partial\Omega$.

3. Observe que o método de Jacobi para a matriz do sistema (11) pode ser feito num looping da forma:

   $$U_{ij}^{new} = \frac{1}{4}(U_{i-1,j}^{old} + U_{i+1,j}^{old} + U_{i,j-1}^{old} + U_{i,j+1}^{old} + h^2 f_{ij}) \text{ para } i, j = 1, \cdots N-1$$

   Escreva loopings semelhantes para os métodos de Gauss Seidel e SOR. Não armazene a matriz do sistema (11). Isto gastaria muita memória sem necessidade. Também não use as matrizes L, D e U no seu programa.

Referências

Sadiku, M. N. (2010). *Elements of Electromagnetics.*

Stoer, J. and Bulirsch, R. (2002). *Introduction to numerical analysis (Third edition).*

Strikwerda, J. (1990). Finite difference schemes and partial difference equations. *Mathematics of Computation - Math. Comput.*, 55.