

Comparação entre o Método de Jacobi e de Gauss-Seidel. Comparação entre os métodos diretos e iterativos. Introdução aos Sistemas não lineares

MS211 – Cálculo Numérico

Giuseppe Romanazzi

29 Outubro 2020

Introdução

Na parte inicial desta aula apresentaremos uma comparação entre os métodos iterativos Jacobi Gauss-Seidel e entre eles e os métodos diretos para a resolução de sistemas lineares.

Depois introduziremos os sistemas não lineares.

Conteúdo

- 1 Comparação completa entre o método de Jacobi e de Gauss-Seidel
- 2 Comparação entre os métodos diretos e os métodos iterativos
- 3 Sistemas de equações não lineares (Sistemas não lineares)
 - Exemplos
 - Matriz Jacobiana, normas e controle dos métodos iterativos

Método de Jacobi, revisão

A expressão explícita do método escrito pode ser escrita na forma vetorial como $x^{(k+1)} = G_J x^{(k)} + c_J = (I - D^{-1}A)x^{(k)} + D^{-1}b$ e na forma escalar explícita como

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

por cada $i = 1, \dots, n$.

Método de Gauss-Seidel

Usando a matriz de iteração $G_{GS} = -(D + L)^{-1}U = I - (D + L)^{-1}A$, e o vetor $c_{GS} = (D + L)^{-1}b$, o método de Gauss-Seidel no é (na forma vetorial)

$$x^{(k+1)} = G_{GS}x^{(k)} + c_{GS},$$

A sua expressão linha por linha (na forma escalar)

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

Forma escalar do método de Gauss-Seidel

$$\left\{ \begin{array}{l} x_1^{(k+1)} = \frac{1}{a_{11}} (b_1 - a_{12}x_2^{(k)} \dots - a_{1n}x_n^{(k)}) \\ x_2^{(k+1)} = \frac{1}{a_{22}} (b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} \dots - a_{2n}x_n^{(k)}) \\ \dots \\ x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) \\ \dots \\ x_n^{(k+1)} = \frac{1}{a_{nn}} (b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} \dots - a_{nn-1}x_{n-1}^{(k+1)}) \end{array} \right.$$

Algoritmo de Jacobi

Algorithm 1 Algoritmo do passo $k + 1$ de Jacobi

Require: $y = x^{(k)}$, $A = (a_{ij})$, $b = (b_i)$

for $i = 1, \dots, n$ **do**

$x_i \leftarrow b_i$;

for $j = 1, \dots, i - 1$ **do**

$x_i \leftarrow x_i - a_{ij}y_j$;

end for

for $j = i + 1, \dots, n$ **do**

$x_i \leftarrow x_i - a_{ij}y_j$;

end for

$x_i \leftarrow x_i / a_{ii}$;

end for

Output: x

▷ Se $i = 1$ este ciclo não tem efeito

Algoritmo de Gauss Seidel

Algorithm 2 Algoritmo do passo $k + 1$ de Gauss-Seidel

Require: $x = x^{(k)}$, $A = (a_{ij})$, $b = (b_i)$

for $i = 1, \dots, n$ **do**

$x_i \leftarrow b_i$;

for $j = 1, \dots, i - 1$ **do**

$x_i \leftarrow x_i - a_{ij}x_j$;

end for

for $j = i + 1, \dots, n$ **do**

$x_i \leftarrow x_i - a_{ij}x_j$;

end for

$x_i \leftarrow x_i / a_{ii}$;

end for

Output: x

▷ Se $i = 1$ este ciclo não tem efeito

Comparação entre o método de Jacobi e de Gauss-Seidel

- O método de Gauss-Seidel resulta ser mais rápido de Jacobi (necessita de poucas iterações para ter convergência), e isso acontece quando a matriz A satisfaz o critério das linhas.
- Se a matriz A tem todos os valores não nulos e a matriz de iteração de Jacobi $G_J = I - D^{-1}A$ tem coeficientes positivos ou nulos, então vale $\|G_{GS}\| < \|G_J\|$.
Neste caso se o método de Jacobi converge, também Gauss-Seidel converge e convergirá mais rapidamente.

Comparação entre o método de Jacobi e Gauss-Seidel

- Note que a iteração $x_i^{(k+1)}$ de Gauss-Seidel depende explicitamente das $x_j^{(k+1)}, j = 1, \dots, i - 1$, que são esperadas ser mais próximas às x_j da solução, respeito às $x_j^{(k)}$ do passo k anterior usadas em Jacobi.
- Esta rapidez de Gauss-Seidel respeito Jacobi pode ser verificada observando que a condição $\|d_r^{(k)}\| < \varepsilon$ acontece num número inferior de iterações respeito ao Jacobi.
- Porém esta rapidez sabe se que a convergência é mais provável que aconteça para o método de Jacobi que para Gauss-Seidel $\{A \text{ matrizes para que Gauss-Seidel converge a solução do sistema}\} \subset \{A \text{ matrizes para que Jacobi converge a solução do sistema}\}$

- O método de Gauss Seidel é mais eficiente no uso da memória, não necessitando arrays a mais respeito aqueles de input. Notamos em vez que o método de Jacobi precisa um vetor a mais ($y \leftarrow x^{(k)}$) para ser implementado. Gauss Seidel em vez pode sobrescrever o vetor de input $x^{(k)}$ para obter a $x^{(k+1)}$
- Uma iteração de Jacobi pode ser obtida mais rapidamente respeito aquela de Gauss-Seidel. Isso porque o método de Jacobi é associado a um algoritmo vetorial.

Isso é, cada componente $x_i^{(k+1)}$ pode ser computada independente das outras $x_j^{(k+1)}$. Observa por exemplo que eu posso computar $x_1^{(k+1)}$, $x_2^{(k+1)}$ ao mesmo tempo sem esperar de obter $x_1^{(k)}$ para poder computar $x_2^{(k+1)}$

Comparação entre os métodos diretos e iterativos

Os métodos diretos:

- determinam num número (conhecido) finito de passos, na aritmética exata (com um número infinito de dígitos), a solução exata do sistema linear
- Na aritmética finita a solução pode distar daquela certa (algoritmos dos métodos diretos podem ser instáveis). O problema é que qualquer método direto vai ser instável por algum tipo de matriz. Por isso é necessário aplicar estratégias (como o pivotamento) para reduzir a instabilidade do método.
- São métodos custosos, sobre tudo quando a matriz A tem uma estrutura (distribuição de elementos não nulos) não conhecida a priori. Exemplo: Eliminação de Gauss tem um custo de $O(n^3)$, se em vez a matriz for triangular podemos resolver o sistema com um custo menor $O(n^2)$.
- Necessitam bastante memória, podem necessitar de todas as n^2 locações de memória da estrutura da matriz inicial . . .

Fill-in do método de eliminação de Gauss

$$A = \begin{pmatrix} x & x & x & x \\ x & 0 & x & x \\ x & 0 & 0 & 0 \\ x & x & 0 & 0 \end{pmatrix} \longrightarrow A^{(1)} = \begin{pmatrix} x & x & x & x \\ m & x & x & x \\ m & x & x & x \\ m & x & x & x \end{pmatrix}$$

Comparação com os métodos iterativos

Os métodos iterativos:

- precisam que sejam analisadas as condições de convergência
- processo iterativo, num número infinito de passo chegaremos a solução exata, mas podemos chegar num número finito de passos a uma aproximação com a tolerância pequena qualquer
- não precisam de mais memória (não há fill-in)
- são métodos mais estáveis que os diretos
- Tem custo computacional bem baixo quando a matriz apresenta muitos zeros.

Esta é uma das vantagens principais respeito aos métodos diretos.

Sistemas de equações não lineares

Estes sistemas de equações são muito comuns nos problemas reais, eles aparecem cada vez que as variáveis procuradas não dependem linearmente entre eles.

Por exemplo, eles podem surgir quando precisarmos de encontrar dois ou mais valores x, y, z, \dots que descrevem quantidades nos

- fenômenos naturais (como a temperatura, pressão, massa, velocidade, tempo, distancia)
- problemas geométricos (coordenadas, inclinações, ângulos)
- problemas industriais (peso, resistência de materiais, temperatura, pontos de ruptura, pontos críticos, tensão elétrica, percentagens, tempos, etc).

Sistemas de equações não lineares

Se as variáveis x, y, z, \dots não fossem ligadas entre eles através equações lineares (ou seja através equações onde aparecem somente a soma e multiplicação para constantes, tipo $a_i x + b_i y + c_i z = d, i = 1, 2, \dots$) então estaremos necessariamente no caso onde é preciso resolver varias equações não lineares ao mesmo tempo.

Cada conjunto de equações não lineares que foram satisfeitas ao mesmo tempo de incógnitas x, y, z, \dots ou x_1, x_2, x_3, \dots chama se sistema de equações não lineares, ou simplesmente sistema não linear.

O sistema linear $Ax = b$ não é claramente um sistema não linear. Note que $ax_1 * x_2 + bx_3 = c$ é uma equação não linear

$$\text{assim como } x_1^2 + x_2 - 3 = x_3$$

$$\cos(x_1) + x_2 = -2,$$

$$\log(x_2) + x_3 = 1,$$

$$e^{x_3} - x_1 = 4x_2, \dots$$

Sistemas não lineares, $F(x) = 0$

Suponha de ter n incógnitas que indicamos com x_1, x_2, \dots, x_n e de ter n equações não lineares entre as variáveis, estaremos no caso de ter o seguinte sistema não linear para resolver:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

A função f_i descreve a equação não linear i .

Por exemplo se for $n = 2$ e $f_1(x_1, x_2) = x_1x_2 + x_1 + 1$,

$f_2(x_1, x_2) = x_1^2 - 2$ é como se quiséssemos encontrar x_1, x_2 tais que

$$\begin{cases} x_1x_2 - x_1 + 1 = 0 \\ x_1^2 - 2 = 0 \end{cases} \quad \text{que tem duas soluções}$$

$$(x_1, x_2) = (\sqrt{2}, 1 - \frac{1}{\sqrt{2}}) \text{ e } (-\sqrt{2}, 1 + \frac{1}{\sqrt{2}}).$$

Sistemas não lineares, $F(x) = 0$

Note que o sistema

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \iff F(x) = 0$$

pode ser escrito simplesmente como $F(x) = 0$, onde $x \in \mathbb{R}^n$ é um vetor de dimensão n que tem como elementos os x_i incógnitos:

$$x = (x_1, x_2, \dots, x_n)^t.$$

A função F transforma os vetores em \mathbb{R}^n em outros vetores em \mathbb{R}^n ,

ou seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$,

satisfaz

$$F(x) = F(x_1, \dots, x_n) = (f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))^t$$

Exemplos de sistemas não lineares, interpretação geométrica

Diferentemente dos sistemas lineares, não é conhecido um critério para determinar a-priori se um sistema não linear de n equações do tipo $f_i(x) = 0$ admite nenhuma, uma ou mais soluções.

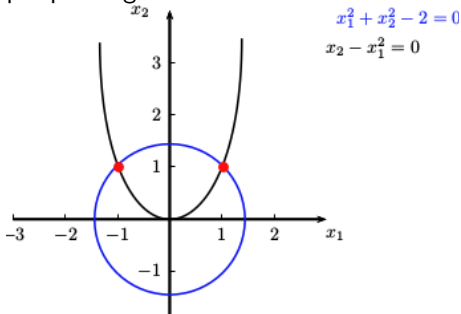
Por exemplo um sistema de duas equações não lineares e de duas incógnitas x_1, x_2 pode ter zero, uma, duas, três, ..., k soluções ou até infinitas soluções.

É possível achar quantas soluções o sistema tem, fazendo o gráfico dos pontos (x_1, x_2) que satisfazem as equações $f_i(x_1, x_2) = 0$ com $i = 1, 2$ e ver quantas interseções existem.

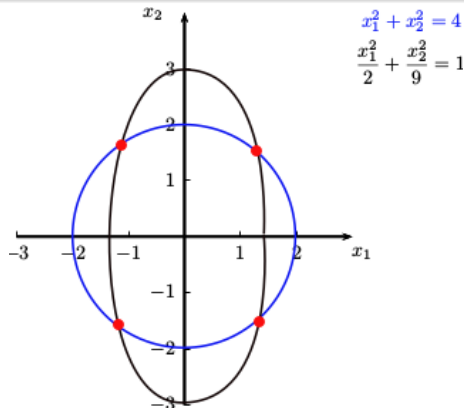
Exemplos de sistemas com $n = 2$ equações não lineares

$$\begin{cases} f_1(x_1, x_2) = x_1^2 + x_2^2 - 2 = 0 \\ f_2(x_1, x_2) = x_2 - x_1^2 = 0 \end{cases} \quad \text{Notamos que a primeira equação}$$

descreve o conjunto de pontos que distam $\sqrt{2}$ da origem $(0, 0)$. Ou seja o gráfico de $f_1(x_1, x_2) = 0$ descreve a circunferência de centro $(0, 0)$ e raio $\sqrt{2}$. Em vez a segunda equação descreve os pontos que se encontram na parábola de equação $x_2 = x_1^2$ que tem vértex na origem. Observando os gráficos notamos que temos duas soluções do sistema, porque os gráficos intersecam-se em dois pontos.



Exemplos de sistemas com $n = 2$ equações não lineares

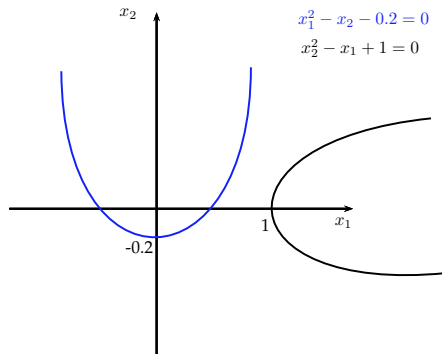


$$\begin{cases} x_1^2 + x_2^2 = 4 \\ \frac{x_1^2}{2} + \frac{x_2^2}{9} = 1 \end{cases}$$

Este sistema tem 4 soluções. Note que as soluções são as interseções da circunferência de raio 2 e centro na origem e de um elipse de equação $\frac{x_1^2}{2} + \frac{x_2^2}{9} = 1$ que então tem semi eixos de medida $\sqrt{2}$ ao longo do eixo x_1 e de medida 3 ao longo do eixo x_2 . A elipse é centrada na origem.

$$\begin{cases} x_1^2 - x_2 = 0.2 \\ x_2^2 - x_1 + 1 = 0 \end{cases}$$

A primeira equação pode ser escrita como $x_2 = x_1^2 - 0.2$ portanto descreve os pontos ao longo de uma parábola com vértex em $(0, -0.2)$ com concavidade por cima. A segunda equação pode ser escrita como $x_1 = x_2^2 + 1$ e portanto é uma parábola com concavidade a direita no plano $x_1\hat{x}_2$ e vértex em $(1, 0)$. Portanto não existem interseções, veja o gráfico. O sistema não tem então soluções.



Matriz Jacobiana de F

Seja $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ a função que descreve o sistema não linear e $x \in D$ um ponto no domínio da F .

Definição Matriz Jacobiana

Chama-se matriz jacobiana de F no ponto x o simplesmente Jacobiana de F em x a seguinte matriz de dimensão n

$$J_F(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \dots & \frac{\partial f_2}{\partial x_n}(x) \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1}(x) & \frac{\partial f_n}{\partial x_2}(x) & \dots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix} = \begin{pmatrix} \nabla f_1^t(x) \\ \nabla f_2^t(x) \\ \vdots \\ \nabla f_n^t(x) \end{pmatrix}$$

Denotado com $\nabla f_i(x) \in \mathbb{R}^n$ o gradiente de f em x ou seja,
 $\nabla f_i(x) = \left(\frac{\partial f_i}{\partial x_1}(x) \quad \frac{\partial f_i}{\partial x_2}(x) \quad \dots \quad \frac{\partial f_i}{\partial x_n}(x) \right)^t$. Temos também

$$J_F(x) = \left(\nabla f_1(x)^t \quad \nabla f_2(x)^t \quad \dots \quad \nabla f_n(x)^t \right)^t.$$

O Jacobiano estende o conceito de derivada para funções n -dimensionais

Exemplo de Jacobiano associado a um sistema não linear

$$\begin{cases} -2x_1^3 + 2x_2 + x_3 = 2 \\ 4x_1 - 2x_2 = -1 \\ -x_3 + x_1 = 0 \end{cases}$$

Observamos que o sistema pode ser escrito assim

$$\begin{cases} f_1(x) = -2x_1^3 + 2x_2 + x_3 - 2 = 0 \\ f_2(x) = 4x_1 - 2x_2 + 1 = 0 \\ f_3(x) = x_1 - x_3 = 0 \end{cases}$$

Portanto o Jacobiano de $F = (f_1 \quad f_2 \quad f_3)^t$ num ponto $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ é

$$J_F(x) = \begin{pmatrix} -6x_1^2 & 2 & 1 \\ 4 & -2 & 0 \\ 1 & 0 & -1 \end{pmatrix}$$