

Resolução de Sistemas Lineares com o Método de Gauss-Seidel

Felipe Alves Siqueira
Universidade de São Paulo (USP)
Instituto de Ciências Matemáticas e de Computação (ICMC)

Abril 2018

1 Descrição do Problema

Neste trabalho, é implementado o método numérico iterativo de Gauss-Seidel para resolução aproximada de sistemas de equações lineares. O algoritmo codificado utilizando a linguagem de Python 3 foi testado a partir de casos de testes solicitados pela especificação original do problema.

2 Método numérico de Gauss-Seidel

O método numérico iterativo de Gauss-Seidel, dado um sistema de equações do tipo $Ax = b$ com n equações e n incógnitas, a ideia é calcular resoluções aproximadas do sistema a cada iteração que dependam não somente da iteração anterior (como no método de Jacobi), mas que também dependam dos cálculos de uma mesma iteração nas equações anteriores. Tal método, assim como o método de Jacobi (que não será apresentado neste trabalho), é uma importante metodologia para a resolução de problemas que apresentam grandes matrizes esparsas pois apresentam um custo computacional tangível. A equação 1 mostra como o sistema $Ax = b$ pode ser reorganizado de modo a permitir o cálculo iterativo de sua solução, como propõe o método de Gauss-Seidel.

$$\begin{cases} x_1^{(k+1)} = (b_1 - x_2^{(k)}a_{1,2} - \dots - x_n^{(k)}a_{1,n})/a_{1,1} \\ \dots \\ x_n^{(k+1)} = (b_n - x_1^{(k)}a_{n,1} - \dots - x_{n-1}^{(k)}a_{n,n-1})/a_{n,n} \end{cases} \quad (1)$$

A equação 1 pode ser convenientemente reescrita na forma da equação 2, que representa como tal método foi efetivamente implementado no código da solução proposta.

$$x^{(k+1)} = L^{-1}(b - (A - L)x^{(k)}) \quad L = \begin{bmatrix} a_{1,1} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \quad (2)$$

Este método evita o cálculo custoso de A^{-1} , por exemplo, além de ser um algoritmo “*anytime*”, isto é, sempre possui uma resposta atual válida independentemente do seu estágio de execução, podendo ser interrompido a qualquer momento. Quanto mais iterações o algoritmo executa, mais precisa será a resposta, supondo que a entrada do problema convirja para uma solução. Como pode ser observado, o método exige uma solução aproximada inicial x_0 como entrada. Neste trabalho, caso esta solução aproximada não seja explicitada pelo usuário do programa, assume-se $x_0 = \vec{0} \in \mathbb{R}^n$.

Computacionalmente falando, o método de Gauss-Seidel é mais eficiente que o método de Jacobi num ambiente sem paralelismo, pois utiliza de valores previamente calculados para novos cálculos. Após o início da execução do método numérico a ordem das equações não pode ser alterada, tornando o método de Gauss-Seidel não paralelizável. O método de Jacobi não possui esta restrição, e é superior ao método de Gauss-Seidel em ambientes com recursos para programação paralela.

3 Critérios de parada

Foram adotados dois critérios de parada para a finalização do código, sendo uma delas um número máximo de iterações n definido pelo usuário. Caso não seja explicitamente definido, é assumido $n = 100$. O segundo critério de parada também é definido pelo usuário, sendo um erro máximo ϵ permitido numa iteração do método numérico em questão. Caso não seja explicitado pelo usuário tal valor, assume-se $\epsilon = 10^{-10}$.

Para o cálculo do erro numa iteração do método, o programa apresenta duas alternativas: Erro de Norma infinita E_∞ (equação 4) e Erro Relativo E_R (5). Ambos usam o conceito de Norma Infinita de um vetor, definida como segue: seja $V = [v_1 \dots v_n]^T \in \mathbb{R}^n$. Define-se sua Norma Infinita como o seu maior elemento em módulo, como mostra a equação 3.

$$\|V\|_\infty = \arg \max_{1 \leq i \leq n} |v_i| \quad (3)$$

A partir da definição de norma infinita, temos o Erro Norma Infinita (equação 4), utilizado como critério de parada padrão do método numérico de Gauss-Seidel neste trabalho.

$$E_\infty(x^{(k+1)}, x^{(k)}) = \|x^{(k+1)} - x^{(k)}\|_\infty \leq \epsilon \ll 1 \quad (4)$$

O programa também disponibiliza o Erro Relativo (equação 5) como critério de parada, caso o usuário deseje.

$$E_R(x^{(k+1)}, x^{(k)}) = \frac{\|x^{(k+1)} - x^{(k)}\|_\infty}{\|x^{(k+1)}\|_\infty} \leq \epsilon \ll 1 \quad (5)$$

4 Critérios de convergência

Existem duas maneiras simples de verificar que, dada uma matriz de coeficientes A e um vetor solução b , o método numérico de Gauss-Seidel converge independentemente da aproximação inicial x_0 . Ambos os critérios serão apresentados nesta seção.

4.1 Critério das Linhas

Se a matrix A for diagonalmente dominante, como define a equação 6, então o método de Gauss-Seidel converge independentemente da aproximação inicial x_0 .

$$a_{i,i} > \sum_j |a_{i,j}|, \quad j \in \{1, \dots, n\} / \{i\} \quad (6)$$

4.2 Positiva Definida

Se A for uma matriz positiva definida, isto é:

- Simétrica
- Todo os autovalores positivos
- $x^T A x > 0, \quad \forall x \in \mathbb{R}^n / \{\vec{0}\}$
- Todos os pivôs, obtidos através da Eliminação de Gauss, são positivos
- Pode-se realizar a fatoração de Cholesky, fato que comentado na próxima seção.

Então o método de Gauss-Seidel converge, independentemente da aproximação inicial x_0 .

5 Matriz Pentadiagonal definida para o problema

A especificação do problema também envolve uma matriz particular A, definida pela equação 7.

$$A = \begin{cases} a_{i,i} = 4, & i = 1, \dots, n \\ a_{i+1,i} = a_{i,i+1} = -1, & i = 1, \dots, n-1 \\ a_{i+3,i} = a_{i,i+3} = -1, & i = 1, \dots, n-3 \\ a_{i,j} = 0, & c.c. \end{cases} \quad (7)$$

Tal matriz A possui algumas características interessantes:

- Esparsa: se $n = 100$, a proporção de elementos nulos em $A \in \mathbb{R}^{100 \times 100}$ é de 0.9508.
- Simétrica: $A = A^T$ pela natureza de sua definição.
- Positiva definida ($x^T A x > 0, \quad \forall x \in \mathbb{R}^n / \{\vec{0}\}$): tal fato pode ser verificado efetuando a fatoração de Cholesky onde $A = LL^T$ e L é triangular inferior, assumindo que a matriz é positiva definida. Como exemplo, para $n = 6$ temos $A = LL^T$ tal que

$$L \approx \begin{bmatrix} 2.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.5 & 1.936492 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -0.516398 & 1.932184 & 0.0 & 0.0 & 0.0 \\ -0.5 & -0.129099 & -0.552052 & 1.85164 & 0.0 & 0.0 \\ 0.0 & -0.516398 & -0.138013 & -0.617213 & 1.825742 & 0.0 \\ 0.0 & 0.0 & -0.517549 & -0.154303 & -0.63901 & 1.81659 \end{bmatrix}$$

6 Código da solução

6.1 Visão geral

Segue uma versão simplificada da implementação do método de Gauss-Seidel. Este recorte enfatiza a implementação efetiva do método numérico, objeto de interesse deste trabalho, e procura prover a maior legibilidade possível ao leitor, não correspondendo exatamente a implementação real anexada a este relatório. Todas as funções auxiliares necessárias para o funcionamento correto deste código devem ser analisadas diretamente do código fonte.

```
n = ArowNum # Numero de linhas de A
# Inicia as matrizes L e U, ambas n x n
L = minit(n, n)
U = minit(n, n)

# A matriz L e triangular inferior, com todos os ele-
# mentos sobre e abaixo da diagonal principal de A.
# U e triangular superior, com todos os elementos aci-
# ma da diagonal principal de A.
for i in range(n):
    for j in range(i, n):
        L[j, i] = A[j, i]
        U[i, j] = A[i, j] if i != j else 0.0

Linv = L.I # Calcula a Inversa de L, assumindo a[i][i] != 0
xCur = x0 # Inicia a aproximacao xAtual como x0 dado
i = 0
err = epsilon * 2.0 + 1.0
# Inicia o metodo numerico iterativo de Gauss-Seidel
while i < itMax and err > epsilon:
    i += 1

    # Calcula a proxima aproximacao:
    xNew = Linv * (b - U * xCur)

    # Calcula o erro Norma Infinita da itera-
    # cao atual
    err = error(xNew, xCur)

    # Atualiza a solucao aproximada atual
    xCur = xNew

# Retorna a ultima solucao aproximada
return xCur
```

6.2 Instruções de execução

Para executar o código, basta utilizar o interpretador Python 3 (testado pelo autor na versão 3.5.2). Note que é necessária a biblioteca “Numpy” para o funcionamento do programa. Existem alguns parâmetros que devem ser especificados através da linha de comando para a execução correta de cada experimento. Uma execução do código completa pode comportar os seguintes parâmetros:

```
gaussSeidel.py
[-a matrixFilepath] [-b vectorFilepath] [-x x0Filepath]
[-n matrixSize] [-i itMax] [-e epsilon]
[-s (show error each iteration)] [-r (use Relative Error)]
```

Ao executar o programa sem especificar nenhum parâmetro, o mesmo irá utilizar dos valores padrão de cada opção para executar um caso de teste simples. Neste caso, é usada a matriz pentadiagonal com dimensão $n=10$, o vetor solução b é definido por $b_i = i^{-1}$, $i = 1, \dots, n$ e a aproximação inicial x_0 é tal que $x_0 = \vec{0} \in \mathbb{R}^{10}$.

Para especificar caminhos de arquivo de entrada, use as opções “-a” para a matrix dos coeficientes A , “-b” para o vetor solução b e “-x” para a aproximação inicial x_0 . Note que as opções “-a” e “-b” devem obrigatoriamente serem especificadas em conjunto. Caso não seja especificado explicitamente, a solução inicial x_0 é tida como o vetor nulo com dimensão compatível com as matrizes A e b . Caso o usuário deseje usar os valores padrão de matrizes A e b , porém com dimensão diferente de 10, basta especificar um novo valor utilizando a opção “-n”. Note que esta opção é ignorada quando as opções “-a” e “-b” são especificadas.

Pode-se alterar o erro máximo ϵ permitido dentro de uma iteração do método numérico com a opção “-e”, e o número máximo de iterações permitido com a opção “-i”. O quadro 1 sintetiza todas as opções descritas nesta subseção.

Na seção de resultados, os comandos de execução de cada caso de teste são expostos para permitir a reprodução de cada teste facilmente pelo leitor.

6.3 Funcionalidades extras

Esta seção é reservada para comentar as algumas das opções extras que o programa disponibiliza.

- -h ou --help: utilize esta opção para exibir ajuda do programa.
- -s: exibe o erro obtido para toda iteração durante a execução do método numérico.
- -r: usa o Erro Relativo E_R (equação 5) ao invés do Erro Norma Infinita E_∞ (equação 4) como critério de parada.

Quadro 1: Sumarização dos parâmetros essenciais para a experimentação correta com o programa.

Opção	Valor padrão	Usada para
-a	Equação 7	Caminho do arquivo que contém os valores da matriz de coeficientes A .
-b	$b_i = 1.0/i$	Caminho do arquivo que contém os valores do vetor solução b .
-x	$\vec{0} \in \mathbb{R}^n$	Caminho do arquivo que contém os valores da aproximação inicial x_0 .
-n	10	Dimensão de A e b padrão quando “-a” e “-b” não forem especificados.
-i	100	Numero máximo de iterações permitido.
-e	10^{-10}	Erro máximo permitido dentro em uma iteração.

7 Resultados e Comentários

O quadro 2 mostra os parâmetros usados em cada um dos testes requisitados pela especificação do problema. As subseções que seguem expõem os resultados obtidos a partir destas execuções.

Quadro 2: Configuração dos parâmetros do algoritmo para cada caso de teste

#	n	ϵ	maxIt	$X^{(0)}$	\mathbf{b}_i	\mathbf{A}
1	50	10^{-13}	6000	$\vec{0} \in \mathbb{R}^{50}$	$b_i = \sum_{j=1}^{50} a_{i,j}$	Pentadiagonal
2	100	10^{-13}	6000	$\vec{0} \in \mathbb{R}^{100}$	$b_i = \sum_{j=1}^{100} a_{i,j}$	Pentadiagonal
3	100	10^{-10}	10000	$\vec{0} \in \mathbb{R}^{100}$	$b_i = i^{-1}$	Pentadiagonal

Quadro 3: Resultados dos testes propostos na tabela 2.

#	Iteração final com E_∞	E_∞
1	1450	$9.880984919163893 \times 10^{-14}$
2	5263	$9.969802761133906 \times 10^{-14}$
3	4036	$9.986189652977373 \times 10^{-11}$
#	Iteração final com E_R	E_R
1	1450	$9.880984919168365 \times 10^{-14}$
2	5263	$9.969802761144364 \times 10^{-14}$
3	3748	$9.970973533825043 \times 10^{-11}$

7.1 Caso de teste 1

Neste caso de teste, a saída esperada fora $\mathbf{x} = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^{50}$. Tal resultado foi obtido com sucesso, com os erros de Norma Infinita e Relativo como mostra na tabela 3.

Comando bash (entrada para interpretador do python 3) para fácil reprodução (*copy-paste*):

```
# Erro norma infinita (sem -r)
python3 gaussSeidel.py -a testcases/1A.in \
    -b testcases/1b.in -i 6000 -e 1e-13

# Erro relativo (com -r)
python3 gaussSeidel.py -a testcases/1A.in \
    -b testcases/1b.in -i 6000 -e 1e-13 -r
```

7.2 Caso de teste 2

Tal teste é bastante similar ao teste 1, porém com um número maior de dimensões ($n = 100$). O resultado esperado também fora $\mathbf{x} = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^{100}$, com os erros correspondentes na tabela 3.

Comando bash (entrada para interpretador do python 3) para fácil reprodução (*copy-paste*):

```
# Erro norma infinita (sem -r)
python3 gaussSeidel.py -a testcases/2A.in \
    -b testcases/2b.in -i 6000 -e 1e-13

# Erro relativo (com -r)
python3 gaussSeidel.py -a testcases/2A.in \
    -b testcases/2b.in -i 6000 -e 1e-13 -r
```

7.3 Caso de teste 3

Neste caso de teste, uma preparação especial foi necessária, como mostra a configuração correspondente na tabela 2. O método numérico levou alguns milhares de iterações para a convergência, que foi obtida com sucesso. Tal resultado já era esperado, dado que a matriz pentadiagonal A respeita os critérios de convergência do método de Gauss-Seidel, como visto na seção 4. Os erros obtidos neste teste foram expostos na tabela 3.

Como pode ser observado, os erros são relativamente pequenos, portanto a resposta é suficientemente precisa. Interessante notar que, ao pedir erro Norma Infinita ou Relativo “nulos” ($\epsilon = 0.0$), o algoritmo leva exatamente 6911 iterações para ser finalizado em ambas as situações. Vale ressaltar que o erro “zero” não é matematicamente zero, levando em conta a limitação natural da capacidade de representação de ponto flutuante pelo padrão IEEE de precisão dupla (64 bits, sendo 1 de sinal, 11 de expoente e 52 de mantissa) e de sua aritmética numa arquitetura de processador de 64 bits.

Comando bash (entrada para interpretador do python 3) para fácil reprodução (*copy-paste*):

```
# Erro norma infinita (sem -r)
python3 gaussSeidel.py -n 100 -i 10000 -e 1e-10

# Erro relativo (com -r)
python3 gaussSeidel.py -n 100 -i 10000 -e 1e-10 -r
```

8 Considerações finais

O código produzido durante este trabalho está disponível em <https://github.com/FelSiq/Mathematica/>