

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Comparação entre o método de Gauss-Jacobi e de Gauss-Seidel em suas formas iterativa e matricial

Estefanie Rayane Silva de Oliveira¹Alexsandro Pinheiro de Moura²Matheus da Silva Menezes³Ivan Mezzomo⁴Stefeson B. M.⁵

Departamento de Ciências, Matemática e Estatística, UFERSA, Mossoró - RN

1 Introdução

Métodos numéricos contribuem na resolução de problemas envolvendo sistemas de equações lineares, presentes nas diferentes áreas do conhecimento humano [2]. Dentre eles, destacam-se os métodos iterativos estacionários de Gauss-Jacobi e de Gauss-Seidel, os quais divergem entre si pelo simples fato de que no constructo do segundo, utilizamos na função de iteração os valores de cálculo já atualizados na iteração corrente, para encontrar a solução posterior [3].

Ambos os métodos podem ser escritos na sua forma matricial. Para Gauss-Jacobi a função de iteração é dada por $x^{k+1} = D^{-1}(I + S)x^k + D^{-1}b$. De modo semelhante, no de Gauss-Seidel se transforma em $x^{k+1} = D^{-1}b - D^{-1}Ix^{k+1} - D^{-1}Sx^k$, sendo que D, I e S correspondem a matriz diagonal, a matriz triangular inferior e a matriz triangular superior, respectivamente [2].

Em trabalhos que utilizam alguma implementação computacional de métodos numéricos, a estratégia utilizada pode influenciar diretamente no tempo de processamento do método, impactando em sua análise de desempenho. Os métodos de Gauss-Seidel e Gauss-Jacobi podem ser implementados de duas maneiras distintas: (a) em sua forma matricial, utilizando as operações matriciais previamente definidas em ambientes de programação matemática, tais como Matlab, Scilab, Octave dentre outros; ou (b) Utilizando implementação através de comandos básicos de programação, onde o programador deve controlar todos os aspectos e operações do método. Dessa forma, pretendemos analisar se existe diferença significativa de desempenho e tempo de processamento entre essas duas formas de implementação, executando os mesmos nos softwares Matlab, versão 4.51 e Scilab, versão 6.0, nas suas formas iterativa e matricial [2], visando verificar qual programa, método e

¹estefanierayane159@gmail.com²alexsandropinheiro3@gmail.com³matheus@ufersa.edu.br⁴imezzomo@ufersa.edu.br⁵stefeson@ufersa.edu.br

forma de resolução possuem melhores resultados e se as diferenças são ou não relevantes. Utilizou-se como critério de parada um limite de 5000 iterações ou uma precisão relativa de $D_r = 10^{-8}$. Utilizou-se quatro matrizes [1], com características dispostas na tabela 1. Os testes foram realizados em um computador com processador Intel® Core i7-3610QM, 6 GB de RAM e sistema operacional Windows 10.

Tabela 1: Dados dos Problemas Considerados

Nome	Tamanho	Estrutura	Diagonal Dominante
m30	30×30	Real não-simétrica	sim
pde225	225×225	Real não-simétrica	não
DWB512	512×512	Real não-simétrica	sim
PDE900	900×900	Real não-simétrica	não

Tabela 2: Resultados Obtidos no Experimento

SCILAB								
Método	Gauss-Jacobi				Gauss-Seidel			
Forma	Iterativo		Matricial		Iterativo		Matricial	
Problema	T(s)	iter	T(s)	iter	T(s)	iter	T(s)	iter
m30	0,045	11	0,002	11	0,036	8	0,0008	8
pde225	289,63	1305	6,500	1305	139,470	633	0,0397	633
DWB512	16,26	14	0,558	14	9,859	8	0,052	8
PDE900	1352,48	407	74,801	407	738,69	223	0,253	223
MATLAB								
m30	0,0009	11	0,002	11	0,00037	8	0,00055	8
pde225	486,83	1305	0,1478	1305	238,47	633	0,0404	633
DWB512	0,1235	14	0,4784	14	0,0712	8	0,0305	8
PDE900	10,9146	407	58,558	407	5,7912	223	0,1678	223

A partir dos dados dispostos na tabela 2, podemos observar, que o método de Gauss-Seidel apresenta melhores resultados, convergindo em tempo menor e menos iterações. Além disso, a forma matricial se mostrou mais eficiente em ambos os métodos e em todas as matrizes testadas, independente do *software*. Percebemos também, em algumas matrizes, uma grande diferença no tempo de execução do Scilab para o Matlab, sendo que este último teve menores tempos de processamento. No entanto, cabe ressaltar que há casos em que o inverso é observado, como por exemplo, a matriz *pde225*. Tais resultados foram satisfatórios e podem servir de embasamento para auxiliar a escolha da estratégia e *software* de forma a obter o melhor desempenho possível.

Referências

- [1] R. F. Boisvert and R. Pozo and K. Remington and R.F. Barrett, J. J. Dongarra. *Matrix Market: A Web Resource for Test Matrix Collections*. Proceedings of the IFIP TC2/WG2.5 Working Conference on Quality of Numerical Software: Assessment and Enhancement, 1997
- [2] R. L. Burden and D. Faires. *Análise Numérica, 10.a ed.* Cengage, São Paulo, 2008.
- [3] S. Chapra. *Métodos Numéricos Aplicados com MATLAB para Engenheiros e Cientistas. 3.a edição*, Bookman, Porto Alegre, 2013.