Um comparativo entre as execuções de implementações sequencial e paralela do método de Gauss-Jacobi

Matheus da Silva Menezes
matheus@ufersa.edu.br

Ivan Mezzomo
imezzomo@ufersa.edu.br

Paulo Henrique Lopes Silva **
phenrique@ufersa.edu.br

João Paulo Caraú Oliveira *
jpauloco@gmail.com

Raimundo Leandro Andrade Marques*
leanmarqs@hotmail.com

Universidade Federal Rural do Semi-Árido UFERSA, Campus Angicos

RESUMO

Os sistemas de equações lineares estão associados a muitos problemas nos campos da engenharia e da ciência [4], mais de 75% dos problemas matemáticos encontrados em aplicações científicas e industriais envolvem a resolução de sistemas lineares em algum estágio [5].

Os sistemas computacionais baseados em arquiteturas *multicore* são uma realidade presente na computação de alto desempenho e procuram melhorar o desempenho computacional na resolução de problemas através do uso de vários processadores trabalhando em paralelo [3]. As mudanças introduzidas por arquiteturas *desse tipo* criaram a necessidade de desenvolver algoritmos que utilizem o *hardware* de forma mais eficiente [1]. Nesse sentido, o presente trabalho apresenta uma comparação para o método Gauss-Jacobi para encontrar solução de sistemas lineares de forma sequencial e paralela em Linguagem C e também uma comparação com os resultados obtidos através do SciLab para os mesmos problemas, também objetivamos uma melhoria no tempo para execução dos cálculos.

O método iterativo de Gauss-Jacobi pertence à classe dos métodos iterativos estacionários sendo regidos por critérios de parada que controlam o processo de cálculo da solução do sistema.

Considere a "i-ésima" equação pertencente a um sistema de ***n*** equações lineares:

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i.$$

Em que, $a_{i,j}$ e b_i são constantes e x_j é uma das "n" variáveis pertencentes à solução do sistema.

Nesse esquema, o valor da "i-ésima" variável (x_i) na "k-ésima" interação (x_i^k) do método de Gauss-Jacobi pode ser calculada pelas equações descritas no Quadro 1:

Quadro 1 – Atualização das variáveis no método de Gauss-Jacobi

Método de Gauss-Jacobi
$x_i^{(k)} = (b_i - \sum_{j \neq i} a_{i,j} x_j^{(k-1)}) / a_{i,i}$

No método de Gauss-Jacobi, a atualização das ***n*** variáveis que definem o sistema pode ser feita de maneira aleatória em cada iteração pois dependem apenas dos valores encontrados na solução anterior. Com isso, pode-se usar multiprocessamento na medida em que a atualização de variáveis distintas dentro da mesma interação pode ser feita em paralelo. Partindo dessa premissa, além de implementarmos o método Gauss-Jacobi de forma sequencial que já é convencionalmente utilizada, desenvolvemos uma implementação paralela para o método segundo as seguintes condições: cada iteração usará dois processadores trabalhando simultaneamente para encontrar os (x_i) que são as variáveis do sistema; os sistemas serão divididos em duas partes, tendo as equações com índice $i \leq n/2$ calculadas pelo processador 1 e as equações com $n/2 < i \leq n$ sendo calculadas pelo processador 2.

O paralelismo neste trabalho foi utilizado segundo as especificações do MPI (Message Passing Interface) [7], com 3 processadores, sendo 1 para processador para coordenar as tarefas (*Master)* que serão realizadas por outros 2 processadores (*Worker)*[7]

Visando analisar a funcionalidade do algoritmo proposto, foi desenvolvida a implementação do método de Gauss-Jacobi, utilizando linguagem de programação C através do Microsoft Visual

* Bolsistas de Iniciação Científica UFERSA
** Professor na UFERSA Mossoró

Studio em uma máquina com processador Intel Core i5, 4GB de RAM e rodando o Windows 7 32 bits.

O teste dos algoritmos foi realizado mediante resolução de dois sistemas lineares, sendo o problema 1 com dimensão 225 x 225, o problema 2 de dimensão 1000 x 1000. Os critérios de parada adotados foram um erro relativo com precisão de 10^{-12} ou um limite máximo de 50000 iterações, sendo que o algoritmo está programado para parar a operação com o que acontecer primeiro. A Tabela 1 mostra o tempo levado para atingir a iteração de convergência e a quantidade de iterações realizadas por cada um dos métodos na resolução dos sistemas considerados:

Tabela1 – Resultados Obtidos

Algoritmo	**Sequencial (s)**	**Paralelo (s)**
Problema 1 (225 x 225)	2,879	2,305
Problema 2 (1000 x 1000)	23,629	18,248

Ambas as execuções sequencial e paralela convergiram no tempo observado conforme tabela. Para os tempos observados, realizamos vários testes e apresentamos a média aritmética. Os resultados também nos mostram que o método paralelo obteve ganho significativo, em relação ao método sequencial acima de 15%. Comparando também com testes realizados para os mesmos problemas numa máquina de mesmo porte, com o SciLab 5.3.2, por [8], vimos que o autor afirma que o SciLab levou 520 segundos para encontrar a solução para o problema 1 e 88000 segundos para o problema 2 sem convergir, mostrando que em nossos testes os resultados foram bem mais satisfatórios.

A diminuição do tempo se deve pelos seguintes pontos: a execução paralela que permite o cálculo de duas variáveis em cada iteração; a diminuição dos erros de arredondamento, pois na Linguagem C podem-se utilizar cerca de 308 casas decimais [9] e o SciLab utiliza apenas 16 casa decimais [10].

Portanto melhoramos o desempenho com a execução em linguagem C diminuindo os erros de arredondamento, e a utilização de paralelismo através da biblioteca MPI permitiu o cálculo de 2 variáveis por vez apresentando ganhos em relação ao tempo de execução.

Palavras-chave: *Métodos iterativos, resolução de sistemas de equações lineares, paralelismo, Gauss-Jacobi*

Referências

[1] Baboulin, Marc; Dongarra, Jack; Tomov, Stanimere. "Some Issues in Dense Linear Algebra for Multicore and Special Purpose" Architectures LAPACK Working Note #200;

[2] Blas: "Basic linear algebra subprograms". http://www.netlib.org/blas/, acesso em 15/01/2012, 19h;

[3] Buttari, Alfredo. Langou, Julien; Kurzak, Jakub; Dongarra Jack. "A Class of Parallel Tiled Linear Algebra Algorithms for Multicore Architectures". LAPACK Working Note # 191;

[4] Burden, Richard L.; Faires J. Douglas. "Análise Numérica. 8. ed. São Paulo": Cengage Learning, 2008;

[5] Leon, S. J. "Álgebra Linear: com aplicações." 8. ed. Rio de Janeiro: LTC, 2011;

[6] Netlib, "método iterativo de Gauss Jacobi", disponível em http://netlib.org/linalg/html_templates/node12.html, acesso em 15/01/2012, 16h ;

[7] "Message Passing Interface (MPI)", disponível em https://computing.llnl.gov/tutorials/mpi/#What, acesso em 27/02/2014, 23h;

[8] Oliveira, João Paulo Caraú; "Proposta Preliminar de um Método Iterativo Híbrido para a Resolução de Sistemas de Equações Lineares", Trabalho de Conclusão de Curso, Universidade Federal Rural do Semi Árido – Campus Angicos, (2014);

[9] Neto, Samuel Dias. "Linguagem C – tipos de dados". http://homepages.dcc.ufmg.br/~joaoreis/Site%20de%20tutoriais/c_bas/c_tipos.html acessado em 13/04/2014, 23h;

[10] "Introdução ao SciLab". http://www.dca.ufrn.br/~meneghet/FTP/MCEC/Transp01.pdf acessado em 13/04/2014, 23h.

