Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

A Influência da Alocação de Memória e Precisão na Resolução de Sistemas Lineares

Rafaela C. Brum[1]
Maria Clícia S. de Castro[2]
Cristiane O. Faria[3]
Programa de Pós-Graduação em Ciências Computacionais, UERJ, Rio de Janeiro, RJ

Este trabalho realiza um estudo comparativo entre alguns dos métodos diretos de resolução de sistemas lineares mais conhecidos na literatura [2, 3, 5] avaliando o uso de precisão simples e dupla na definição das variáveis. Os métodos avaliados são: Eliminação Gaussiana, Método de Gauss-Jordan, Decomposição LU e LDU, Decomposição de Cholesky e Decomposição QR por ortogonalização de Gram-Schmidt.

A precisão das variáveis define quantas casas decimais são usadas para representar computacionalmente os valores numéricos. Quanto mais casas decimais pudermos utilizar, mais preciso são os resultados e os erros numéricos vindos do processo de truncamento são reduzidos. Além disso, dados iniciais bem definidos em relação a sua precisão podem não ser bem armazenados no computador, uma vez que este utiliza a base binária para as suas operações. Definir as variáveis com precisão dupla nos permite armazenar o dobro de casas decimais que as variáveis com precisão simples e por isso, normalmente é a melhor escolha. Porém, para o computador significa o dobro de tempo para acessar cada variável. Então, neste trabalho estamos interessados em avaliar o que é mais vantajoso e importante, tempo na resolução ou uma solução mais precisa.

Os experimentos foram feitos com matrizes quadradas simétricas e definidas positivas de ordem 500 até 3000. As matrizes de entrada foram geradas a partir de uma matriz A com números aleatórios de 0 a 999. Para transformá-la em uma matriz simétrica, multiplicamos pela sua transposta. Depois, somamos a ordem (500 até 3000) na diagonal principal, para garantir que a matrizes possuem a propriedade de ser definida positiva. Para podermos comparar os resultados das duas precisões mais facilmente, multiplicamos essas matrizes criadas pelo vetor com valor 1 em todas as posições para achar o vetor dos valores independentes (vetor B). Assim, os sistemas tem a solução trivial como resposta.

Os testes foram realizados numa máquina Intel Core i7 de 2,80 GHz, com 4 núcleos, 8 GB de memória principal, 8 MB de memória cache, sistema operacional Ubuntu versão 14.04. Cada método foi implementado em linguagem C, compilado com gcc versão 4.8.4.

Em todos os métodos, o tempo de execução usando precisão simples ou dupla foi similar para as matrizes menores. No entanto, o tempo aumentou a medida que o tamanho da

[1]rafinhacbrum@gmail.com
[2]mariaclicia@gmail.com
[3]cofaria@ime.uerj.br

massa de dados de entrada crescia. Porém, todos os métodos tiveram valores diferentes para as incógnitas na precisão simples e somente a Decomposição QR na precisão dupla.

Na Tabela 1 mostramos a norma euclidiana do erro entre a solução exata e a solução aproximada para todos os métodos considerando uma matriz de ordem 2000. Podemos observar que a Decomposição QR é a que mais sofre com propagação de erros (erro de precisão na precisão dupla). Este fato ocorre porque a Decomposição QR é a que mais utiliza operações de multiplicação e, por isso, escala a quantidade de dígitos muito rapidamente. Assim, comprovamos experimentalmente que a multiplicação é uma das operações que mais sofre com arredondamentos e truncamentos, o que é apresentado em [4].

Tabela 1: Normas das duas precisões para uma matriz da ordem 2000

Método direto	Precisão simples	Precisão dupla
Eliminação Gaussiana	29,32	0
Gauss-Jordan	29,32	0
Decomposição LU	469,49	0
Decomposição LDU	469,48	0
Decomposição Cholesky	644,55	0
Decomposição QR	4719,97	59,60

Neste trabalho podemos observar que o aumento de tempo de execução ao definir as variáveis com precisão simples e dupla não foi significativo para essa massa de dados. Já os erros encontrados foram altos. Assim, concluímos que é mais vantajoso usarmos variáveis com precisão dupla de memória e garantirmos uma solução mais precisa. Como sequência deste trabalho pretendemos aumentar o tamanho da matriz e verificar se o mesmo comportamento continuará.

Referências

[1] A. F. G. Ascencio e E. A. V. Campos. *Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java.* Pearson Prentice Hall, São Paulo, 2008. ISBN: 978-85-7605-148-0.

[2] J. W. Demmel. *Applied Numerical Linear Algebra.* SIAM, 1997. ISBN:978-08-987-1389-3.

[3] G. H. Golub and C. F. van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1996. ISBN: 0-8018-5414-8.

[4] M. A. G. Ruggiero e V. L. da Rocha. *Cálculo numérico: aspectos teóricos e computacionais.* Makron Books do Brasil, São Paulo, 1997.

[5] D. S. Watkins. *Fundamentals of matrix computations.* Wiley-Interscience, New York, 2002. ISBN: 978-0-471-21394-9.