**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**

**MÉTODOS NUMÉRICOS COMPUTACIONAIS**

**Lista de exercícios 01**

Thiago Henrique Gonçalves Mello - 201612060188 - thiagohgmello@gmail.com

**Questão 1)** Para resolução da primeira questão da lista de exercícios, foi desenvolvida uma rotina em linguagem C. Primeiramente, criou-se um arquivo *header* nomeado “matrix.h” com o intuito de separar as rotinas desenvolvidas para cada tipo de problema. Nele, foram inseridas as funções responsáveis pela verificação da possibilidade de multiplicação entre matrizes, inicializador de matrizes, multiplicador e uma rotina específica para desalocar matrizes uma vez que estas são criadas dinamicamente.

A verificação é feita através da comparação entre o número de colunas da primeira matriz na multiplicação e o número de linhas da segunda. O tamanho dos elementos multiplicados é definido pelo usuário exigindo então uma requisição de dados inicialmente.

Com os tamanhos especificados, passa-se então para a multiplicação das matrizes, caso a operação possa ser realizada. Uma rotina em MATLAB foi feita com o intuito de comparar os resultados oriundos da rotina em C com precisões de ponto flutuante (*float*) e *double*. Para explorar de forma mais completa as vantagens e desvantagens de determinado tipo de variável, as entradas de cada uma das matrizes foram determinadas a partir de uma função matemática, a saber:

em que é uma variável semi-aleatória criada através da função “rand” da biblioteca *stdlib*.

A norma implementada para comparação de resultados foi a norma de Frobenius, uma vez que esta expressa de forma similar a média geométrica dos elementos da matriz.

A título de comparação entre as duas precisões, vários tamanhos de matrizes distintos foram gerados aleatoriamente e então comparadas com o resultado oriundo da rotina em MATLAB.

Com a precisão *float*, 5 tamanhos distintos de matrizes foram apresentados com os erros da norma de Frobenius expressos na Tabela I.

Tabela I. Erros percentuais sobre as normas de Frobenius considerando precisão float. Comparação entre programa C e resultado do MATLAB.

|  |  |  |
| --- | --- | --- |
| Tamanho das Matrizes | | Erro (%) |
| A | B |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

De forma similar, o mesmo procedimento com matrizes do mesmo tamanho, porém com possibilidade de serem diferentes, foi realizado considerando precisão estendida (*double*). Os resultados estão na Tabela II.

Tabela II. Erros percentuais sobre as normas de Frobenius considerando precisão double. Comparação entre programa C e resultado do MATLAB.

|  |  |  |
| --- | --- | --- |
| Tamanho das Matrizes | | Erro (%) |
| A | B |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Comparando os dados, vê-se que, mesmo apresentando erros baixos, variáveis *float* cometem erros cerca de um milhão de vezes superiores aos casos em que se utiliza *double*. Para matrizes maiores como por exemplo a última experimentada, os erros absolutos com números de ponto flutuante chegaram na casa de milhares de unidades, o que pode representar grandezas demasiadamente grandes para problemas específicos.

Outro ponto que merece ser ressaltado é o efeito da propagação do erro. Para o caso de multiplicação, houve apenas dois pontos de combinação de incertezas: 1) no cálculo do elemento em e 2) no produto entre os elementos, entretanto, caso houvesse mais operações subsequentes, a propagação ocasionaria problemas maiores, como será visto em outras questões da lista.

A desvantagem em se utilizar tamanhos maiores de variáveis ficou explícita ao avaliar o arquivo .txt gerado como saída do produto. Para o caso de precisão *double*, o arquivo teve cerca de de memória, o que impossibilitava até sua abertura em programas para edição deste tipo de extensão.

Portanto, o que se pode concluir neste problema é que, caso não seja necessária uma precisão exorbitante, variáveis do tipo float são interessantes.