MULTIPLICAÇÃO DE MATRIZES E A PERMUTAÇÃO DE ÍNDICES

Caíque da Silva Machado

12 de julho de 2019

Resumo

Para este artigo será realizado o estudo das possibilidades para os índices dos laços do algoritmo clássico para multiplicação de matrizes, onde as operações aritméticas se encontram no laço mais interno. A ferramenta computacional escolhida para escrever o algoritmo e avaliar o seu desempenho foi o *MATLAB*.

Palavras Chaves: Multiplicação de Matrizes, Permutações e Algoritmos.

1 Introdução

O produto entre matrizes é uma operação frequente e costuma ser realizada no processo de resolução de sistemas lineares e não lineares. A definição desta operação é: seja $A = (a_{ij})$, uma matriz $m \times p$ e $B = (b_{ij})$, uma matriz $p \times n$. O produto de A por B, denotado por AB é a matriz $C = (c_{ij}), m \times n$ definida por $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + ... + a_{ip}b_{pj} = \sum_{k=1}^{p} a_{ik}b_{kj}$, para $1 \leq i \leq m, 1 \leq j \leq n$ [1]. É importante observar que o produto AB está definido apenas quando o número de colunas de A é igual ao número de linhas de B.

2 Definição do Problema

Dada a multiplicação $C = A \times B$ sendo que $A_{n\times n}$ e $B_{n\times n}$ são matrizes quadradas. Esta operação, apesar de ser simples, é custosa quando as matrizes envolvidas são densas e de ordem elevada. Sabendo - se que o algoritmo intuitivo, que realiza o produto, faz uso de três laços de repetição aninhados será realizado neste trabalho o estudo das curvas de complexidade para cada combinação dos índices dos laços, afim de se avalar o desempenho do método que realiza o produto para cada permutação.

3 O algoritmo de multiplicação

Para este artigo será feito o uso do algoritmo clássico para multiplicação de matrizes, onde as operações aritméticas se encontram no laço mais interno. A ferramenta computacional escolhida para escrever o algoritmo e avaliar o seu desempenho foi o *MATLAB*. Abaixo, no algoritmo 1, tem -se descrito o método clássico.

Algoritmo 1 : Algoritmo clássico de multiplicação de matrizes.

```
for I = 1:N
  for J = 1:N
    for K = 1:N
      C(I, J) = C(I, J) + A(I, K) * B(K, J);
    end
  end
end
```

4 Estudo das Possibilidades

A primeira permutação escolhida foi obtida colocando os índices na ordem JIK e tanto para o modo clássico quanto para a primeira permutação, cada produto interno que gera um elemento da matriz C, é calculado no laço mais interno, porem a permutação JIK realiza a multiplicação na forma vetorizada [2]. Isto pode ser descrito graficamente usando um diagrama, como foi introduzido por Dongarra [3]. Abaixo, nas figuras 1 e 2, tem - se o diagrama de acesso a memória para as opções escolhidas.

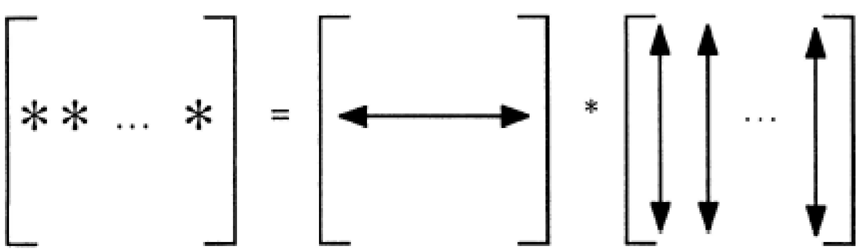

Figura 1: Acesso a memoria para os índices IJK

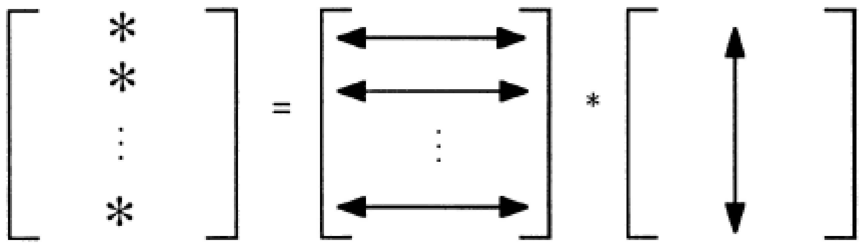

Figura 2: Acesso a memoria para os índices JIK

É importante, observar que, independentemente da escolha da ordem dos índices os valores dos elementos da matriz contendo a solução do produto não são afetados, ou seja, o resultado do produto é o mesmo.

Para a combinação de índices KIJ, uma linha de B é multiplicada com um elemento de A e o resultado é usado na atualização da linha de C. Esta opção também manipula as linhas da matriz e portanto não é preferível [2], porem para a ordem KJI multiplica - se uma coluna de A com um elemento de B para atualizar uma coluna de C. Esta ordem não apresenta conflitos de armazenamento e portanto é mais eficiente que as opções anteriores [2]. No entanto, deve - se notar que, como o índice J varia, outra coluna de C deve ser carregada e armazenada [2].

Para esta combinação de índices é possível notar que o número de operações para os laços mais internos será $2N^2+N$, uma vez que, será necessário 1 operação de leitura da coluna K de A, N operações de leitura e N operações de escrita na coluna J de C [2].

Para as combinações IKJ e JKI, tem -se que, para a primeira opção os elementos de B são acessados coluna por coluna o que pode provocar conflitos no armazenamento. Para a segunda opção, tem - se que, o laço mais interno se encontra vetorizado de maneira eficiente, uma vez que, o produto realizado no laço mais interno é feito continuamente sobre o vetor armazenado [2]. Abaixo, nas figuras 3 e 4, tem - se o diagrama de acesso a memória para estas duas ultimas combinações.

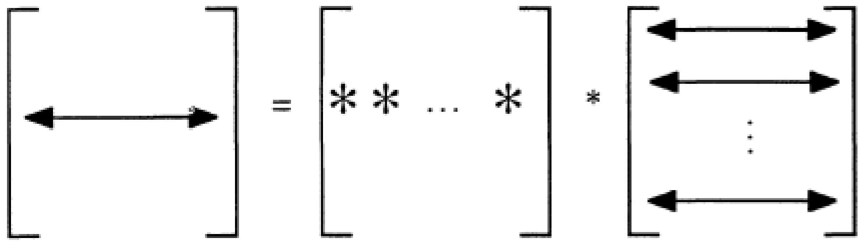

Figura 3: Acesso a memoria para os índices IKJ

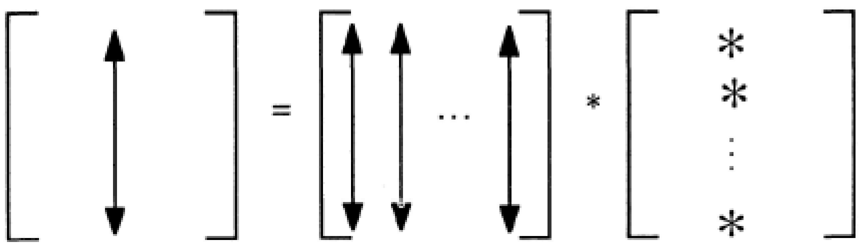

Figura 4: Acesso a memoria para os índices JKI

O número de operações realizado pela combinação JKI é de $N^2 + 2N$, uma vez que, este faz N leituras das colunas de A e faz 1 operação leitura e 1 de escrita da coluna J de C [2].

5 Resultados

Uma vez que, foi realizado o estudo das possibilidades para os índices dos laços de repetição utilizados no algoritmo clássico, foi elaborado um programa, que será apresentado no apêndice deste trabalho, para realizar a multiplicação de duas matrizes densas e de ordem elevada, afim de se avaliar o desempenho do algoritmo para cada tipo de combinação de índice estudada.
A ordem das matrizes utilizadas é 20×20, um valor considerado apropriado para o teste, e na figura 5, abaixo, é apresentado o gráfico contendo o resultado do algoritmo, sendo que, é importante observar que o parâmetro utilizado na comparação das possibilidades é o tempo necessário para se realizar a multiplicação das matrizes.

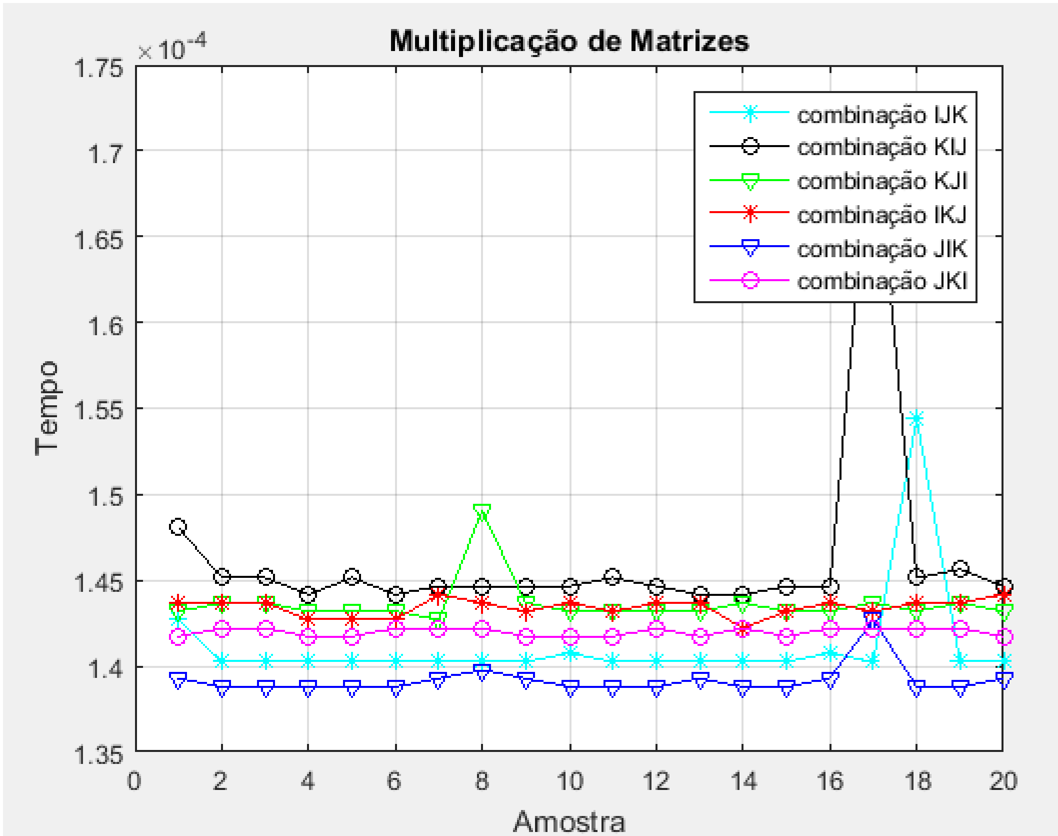

Figura 5: Comparação entre os tempos para cada combinação

6 Conclusão

Enfim, apos o estudo das possibilidades observa - se que, no resultado obtido pelo programa, a combinação JIK apresenta o melhor tempo para a atividade de multiplicar as matrizes. Isto se deve, ao fato desta combinação de índices vetorizar os laços o que proporcionou o melhor desempenho.

Referências

[1] R. Mauro, F. Márcia, Álgebra Linear aula 6: Matriz, CEDERJ 2005.

[2] C. Lacor, Programming aspects and algorithms, 2010.

[3] Dongarra, J. J., Redesingning Linear Algebra algoritns. Proc, 1, Int. call. on vector and parallel computing in scient. appl., bulletin de la direction des studes et Recherches, Serie C, nº 1 (1983), 51 - 59.

7 Apêndice

Para este trabalho foi desenvolvido um programa para avaliar o desempenho do algoritmo de multiplicação de matrizes quando se permuta os índices dos laços de repetição que o compõe. Abaixo, no algoritmo 2, está apresentado este programa.

```
Algoritmo 2: Multiplicação de Matrizes com permutação de índices

clear all; clc; close all;

N = 100;%tamanho da amostra
P = 6;%número de possibilidades
A = rand(N, N);
B = rand(N, N);
C = zeros(N, N);
t = zeros(P, N);
%inicializa indice
amostra = 1;

while amostra <= N
  tempo = 1;

  tic
  for I=1:N
    for J=1:N
      for K=1:N
        C(I,J)=C(I,J)+A(I,K)*B(K,J);
      end
   end
  end
  t(tempo, amostra) = toc;
  tempo = tempo + 1;
  C = zeros(N, N);
  tic
  for K=1:N
    for I=1:N
      for J=1:N
        C(I,J)=C(I,J)+A(I,K)*B(K,J);
      end
    end
  end
  t(tempo, amostra) = toc;
  tempo = tempo + 1;
  C = zeros(N, N);
```

```
tic
for K=1:N
  for J=1:N
    for I=1:N
      C(I,J)=C(I,J)+A(I,K)*B(K,J);
    end
  end
end
t(tempo, amostra) = toc;
tempo = tempo + 1;
   C = zeros(N, N);

tic
for I=1:N
  for K=1:N
    for J=1:N
      C(I,J)=C(I,J)+A(I,K)*B(K,J);
    end
  end
end
t(tempo, amostra) = toc;
  tempo = tempo + 1;
C = zeros(N, N);

tic
for J=1:N
  for I=1:N
    for K=1:N
      C(I,J)=C(I,J)+A(I,K)*B(K,J);
    end
  end
end
t(tempo, amostra) = toc;

tempo = tempo + 1;
C = zeros(N, N);

tic
for J=1:N
  for K=1:N
    for I=1:N
      C(I,J)=C(I,J)+A(I,K)*B(K,J);
    end
  end
end
t(tempo, amostra) = toc;
```

```
  amostra = amostra + 1;
end

plot(t(1,1:N),'c-*');
hold on
plot(t(2,1:N),'k-o');
hold on
plot(t(3,1:N),'g-v');
hold on
plot(t(4,1:N),'r-*');
hold on
plot(t(5,1:N),'b-v');
hold on
plot(t(6,1:N),'m-o');
ax = gca;
ax.XTick = 1:N;
title('Multiplicação de Matrizes');
xlabel('Amostra');
ylabel('Tempo');
legend('combinação IJK','combinação KIJ', 'combinação KJI',
'combinação IKJ', 'combinação JIK', 'combinação JKI');
grid on;
```