Trabalho Prático de Introdução a Programação

Sthefanie J. G. Passo[1]

[1]Instituto de Computação - Universidade Federal do Amazonas (UFAM)
[1]Av. Rodrigo Otávio, 6200 - Japiim - 69077-000 – Manaus – AM – Brasil

sjgp@icomp.ufam.edu.br

Abstract. *This article came with one fast study about the story and application of determinants, it property, theorems and easier rules to do algebra calculus. The principal objective was to develop easy strategies to calculate one determinant looking for a codification tool. The linear algebra notion in this tool made the optimization method strategy happen, it was 66% faster than the not optimized.*

Resumo. *Este artigo traz um breve estudo sobre a história e aplicação dos determinantes, suas propriedades, teoremas e regras facilitadoras para cálculos algébricos. O objetivo principal foi desenvolver estratégias mais fáceis de calcular um determinante através de uma ferramenta codificada. A noção de álgebra linear nesta ferramenta proporcionou a estratégia de métodos com otimização, o qual superou 66% em rapidez em relação ao não otimizado.*

1. Introdução e História

Historicamente a matemática iniciou seu maior desenvolvimento no Oriente, onde os chineses quando compunham seus diagramas representavam sistemas lineares e seus coeficientes em barras de bambu sobre os quadrados de um tabuleiro. Assim foi descoberto o método de resolução por eliminação - que consiste em anular coeficientes por meior de operações elementares (séc. III a.C.).

No decorrer dos séculos, matemáticos se destacaram ao aplicar técnicas para cálculo de determinantes de ordem 2. Posteriormente o escocês Colin Maclaurin (1698-1746) resolveu um sistema de N equações e N incógnitas por meio de determinantes - publicada em 1748 - e paralelo a isso o suíço Gabriel Cramer (1704-1752) defendeu o mesmo método de Maclaurin através da introdução e análise das curvas planas (1750) em conexão determinou os coeficientes da cônica geral. Assim ficou conhecida a Regra de Cramer mesmo que, em parte, a mesma não foi descoberta exclusivamente por ele (DOMINGUES, 2018).

O processo estabelecido de sinais dos termos de um determinante foi sistematizado pelo francês Étienne Bézout em 1764. Já o importante teorema de Laplace foi demonstrado em 1772, para a aplicação do teorema deve-se escolher uma linha ou coluna para calcular o determinante. Assim cada elemento é multiplicado pelo seu cofator correspondente, e quando todos os elementos forem multiplicados ocorre a soma algébrica destes, obtendo a determinante da matriz completa (OLIVEIRA, 2015).

No século seguinte o alemão Carl G. J. Jacobi (1804-1851) contribuiu para a consolidação da teoria dos determinantes. Essa teoria foi simplificada graças a Jacobi quando disse que o determinante de uma matriz não se altera quando somamos aos elementos de uma fila uma combinação linear dos elementos correspondentes de filas paralelas (VIRTUOUS, 2018).

A organização do trabalho é distribuida em Apresentação de Resultados com os métodos implementados e suas otimizações básicas e avançadas. Após isso é feita uma descrição da máquina utilizada, caso queira reproduzir os mesmos resultados encontrados no artigo.

2. Apresentação dos Resultado

Ao aplicar métodos de cálculo determinantes para matrizes de ordem N, usando propriedades de álgebra linear e programação, houve a aplicação de técnicas com otimização básica e avançada. Tais métodos foram descritos neste capítulo, assim como foi feita a análise de resultados na execução dos mesmos. Por fim, com base nos resultados houve a conclusão de quão vantajoso foi otimizar tais métodos.

2.1 Métodos Implementados

2.1.1 Otimização Básica

Com o objetivo de otimizar os cálculos de determinante, foram utilizados métodos com regras para reduzir a quantidade de passos recursivos em matrizes de médio e grande porte. Para isso, houve uma otimização básica que detecta a linha ou coluna com mais zeros.

O método **linhaParaInicio()** cria um vetor que verifica quantos zeros há em cada linha, por exemplo, se em uma matriz a linha um (1) tiver um (1) números zero (0), o vetor do método terá na posição um (1) o número um (1), assim como descreve a Figura 1 .

$$\begin{vmatrix} 5 & 2 & 3 \\ 4 & 0 & 2 \\ 10 & 4 & 3 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 \end{vmatrix}$$

Figura 1. Exemplo de vetor criado após uma matriz 3x3 ser verificada no método linhaParaInicio(). FONTE (o autor, 2018).

Do mesmo modo, o método **colunaParaInicio()** cria um vetor que contabiliza quantos zeros há em cada coluna, e contabiliza em cada posição do vetor. Após esta verificação, o método retorna um objeto do tipo vetor para verificação.

O método **detOrdemNBasica(Matriz mat)** verifica qual a linha e qual a coluna que há mais zeros de acordo com os vetores dos métodos linhaParaInicio() e colunaParaInicio() respectivamente. Em seguida a linha e coluna com mais zeros são comparadas para que a com maior quantidade tenha a posição escolhida a fim de ser utilizada como cofator.

No que diz respeito ao cofator, quando este for diferente de zero, a Regra de Laplace é feita normalmente; quando o cofator for igual a zero, a regra não é aplicada - resposta é igual a resposta anterior - pois ao multiplicar qualquer determinante por zero o resultado é zero.

2.1.2 Otimização Avançada

Na otimização avançada o método **detPro()** controla os comandos. Dentre eles estão os mesmos da detBasica() e outros mais, veja a seguir:

- **detOrdem3(this):** é utilizada a Regra de Sarrus que faz os somatórios dos produtos das diagonais principais com o somatório dos produtos das diagonais secundárias. Em seguida subtrai o primeiro valor pelo segundo, obtendo o determinante e evitando métodos recursivos. O método retorna a resposta do determinante.
- **nulaLinhaColuna(this)**: se a matriz possuir uma linha ou coluna inteira com elementos zero (0) então toda a matriz é nula e seu determinante é zero (0). O método analisa a matriz caso isso ocorra e retorna verdadeiro se o determinante for zero (0) (RIBEIRO, 2017).
- **matTriangular(this)**: o método verifica se todos os elementos acima da diagonal principal ou abaixo da diagonal principal são zero. Se forem, então o determinante é zero (0) e o método retorna verdadeiro (RIBEIRO, 2017).
- **matMultipla(this)**: no método há verificação de linhas e colunas. Se duas linhas de uma matriz são iguais, então seu determinante é nulo. Se duas colunas de uma matriz são proporcionais, então seu determinante é nulo. Caso o determinante seja nulo, o método retorna verdadeiro (VIRTUOUS, 2018).

Após isso, se a matriz passar por estes métodos e não for encontrada o determinante, então o método **controleChio(this)** será aplicado, o qual contém os seguintes métodos.

- **matChio(this)**
- **encontraUm(this)**
- **forcaUm(this)**

De início verifica-se a posição (0,0) da matriz e se o elemento é um (1) (Figura 2).

$$A_{4x4} = \begin{pmatrix} 1 & 2 & 0 & 2 \\ 5 & 2 & 2 & 1 \\ 1 & -5 & -2 & 0 \\ 6 & 7 & 4 & 7 \end{pmatrix}$$

Figura 2. Verificação do primeiro elemento da matriz 4x4. FONTE (OLIVEIRA, 2016).

Se o primeiro elemento for um (1), há a aplicação da Regra de Chió – **matChio(this)** – na qual multiplica-se o elemento da primeira linha com seu correspondente na mesma coluna. Em conseguinte, multiplica-se o elemento da primeira coluna com seu correspondente na linha, conforme a Figura 3. Assim se obterá uma matriz 4x4 por uma nova matriz 3x3 que terá o determinante igual ao da anterior.

$$A_{3x3} = \begin{pmatrix} 2-5.2 & 2-5.0 & 1-5.2 \\ -5-1.2 & -2-1.0 & 0-1.2 \\ 7-6.2 & 4-6.0 & 7-6.2 \end{pmatrix} = \begin{pmatrix} -8 & 2 & -9 \\ -7 & -2 & -2 \\ -5 & 4 & -5 \end{pmatrix}$$

Figura 3. Subtraindo os elementos suprimidos e obtendo a nova matriz de ordem 3x3 após a aplicação da Regra de Chió. FONTE (OLIVEIRA, 2016).

Pode haver casos que as matrizes tenham o elemento 1 mas não na posição (0,0). Para isso o método **encontraUm(this)** foi implementado. Quando o método encontra o número um (1), são aplicadas as propriedades dos determinantes a fim de mover esse elemento para a posição (0,0) e ser aplicada a Regra de Chió.

Nas propriedades dos determinantes pode-se entender que, ao trocarmos duas linhas ou duas colunas de lugar, o sinal do determinante fica negativo (IGM, 2009). Já, se houver duas trocas, o sinal fica positivo. Com isso foi feito um controle para verificar quantas trocas foram feitas para adequação do sinal.

Por fim, foi criado um método para tentar gerar o número 1 a fim de que este seja colocado na posição (0,0) – encontraUm(this) - e viabilize aplicação a Regra de Chió – matChio(this). No método **forcaUm(this)** tem-se a utilização do Teorema de Jacobi no qual diz que o determinante de uma matriz não se altera se somarmos (ou subtrairmos) uma coluna em combinação linear de outra coluna. Veja o exemplo da aplicação da regra na Figura 4:

$$\begin{vmatrix} 5 & 2 & 3 \\ 4 & 1 & 2 \\ 10 & 4 & 3 \end{vmatrix} = \begin{vmatrix} 5-2.2 & 2 & 3 \\ 4-1.2 & 1 & 2 \\ 10-4.2 & 4 & 3 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 2 & 4 & 3 \end{vmatrix} = 9$$

Figura 4. Exempo - Teorema de Jacobi. FONTE: adaptado (VIRTUOUS, 2018).

2.2 Máquina Utilizada

- Sistema Operacional: Deepin (Copyright 2011-2017 Wuhan Deepin Technology Co., Ltd.)
- Edição: 15.5 Desktop

- Tipo: 64Bit
- Processador: Intel® Core(TM) 5-5200U CPU @ 2.20GHz x 4
- Memória: 7.71GB
- Disco: 931.51GB

2.3 Resultados

Após a implementação dos métodos houve a coleta de dados em três matrizes distintas para obter resultados estatísticos representativos. A análise seguiu a metodologia que consistiu na aplicação do método em três matrizes de mesma ordem e com números randômicos de zero (0) a vinte (20), em seguida tirou-se a média aritmética dos resultados em cada método. No que diz respeito a otimização básica e avançada, notou-se uma grande vantagem, pois com a otimização básica o tempo reduziu mais que a metade em muitos casos (Tabela 1).

Tabela 1. Análise de resultados na aplicação dos métodos com matrizes de ordem N em nanossegundos. FONTE: (o autor, 2018).

Ordem	Det. sem otimizar (ns)	Det. básica (ns)	Det. avançada (ns)
5	611337,6667	215701	199291
7	6874224,667	1775756,667	1178215,667
9	85541208,00	38217329,33	36784170,67
11	507584686762,667	400892887040	400535792298,67
13	699111571456	544998457344	543823659008
15	168286999293185	162178769798242	156097801773139

Com a otimização avançada o processamento foi ainda mais eficiente por conta dos métodos básicos e ajuste fino para reduzir a quantidade de passos recursivos. Sabendo que há passos recursivos, foi feito e ajustado um gráfico com unidades em log para melhor visibilidade (Figura 5).

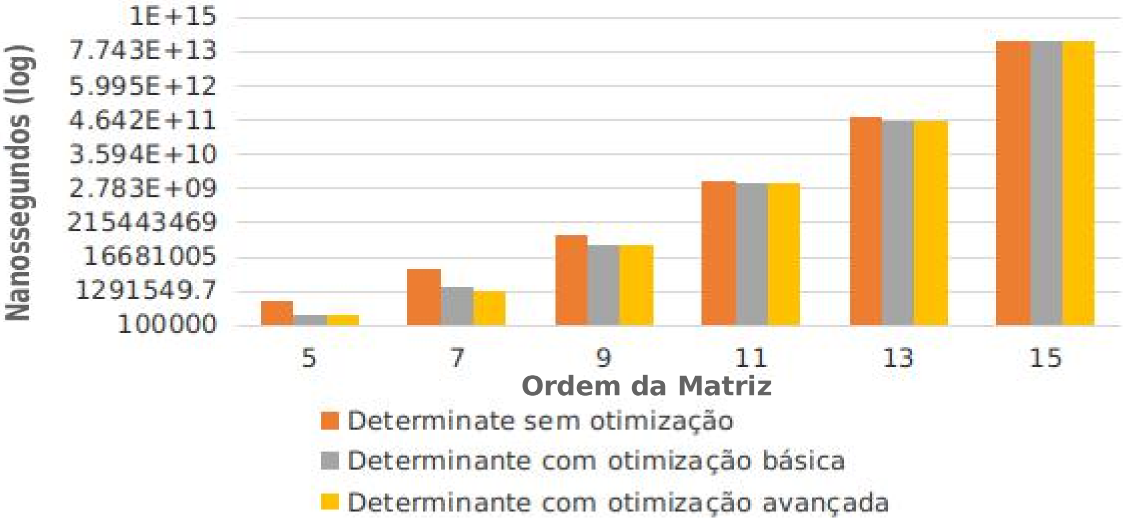

Figura 5. Média de *delay* dos métodos com matrizes de ordem N em nanossegundos (log). FONTE: (o autor, 2018).

Após isso foi ajustado outro gráfico do tipo linha, a fim de verificar o comportamento sequencial de uma matriz para outra, observando o tempo de cada uma e o padrão - em escala logarítmica (Figura 6).

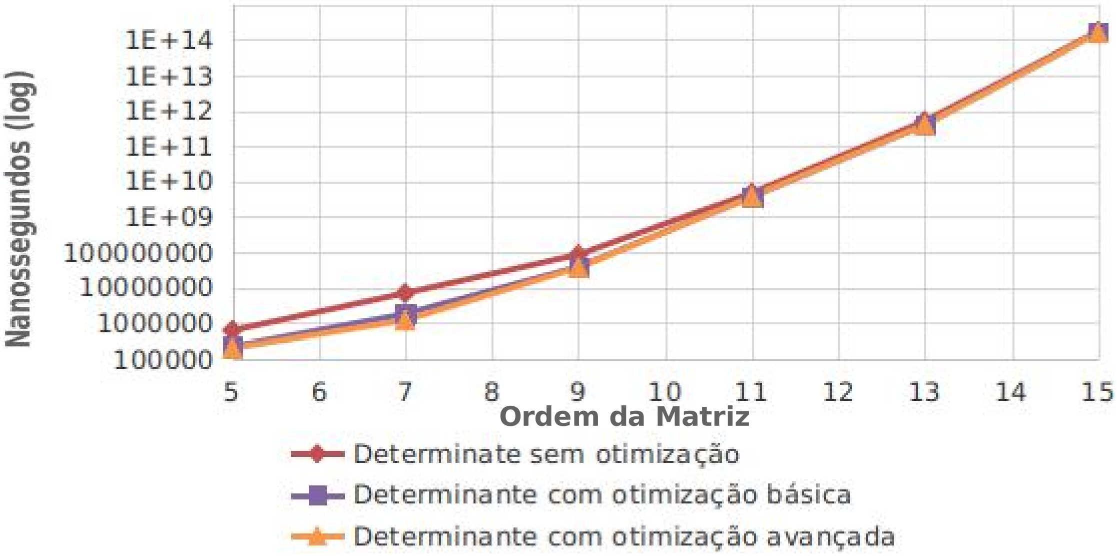

Figura 6. Análise de resultados para verificação do comportamento com matrizes de ordem N em nanossegundos (log). FONTE: (o autor, 2018).

Algo a se notar é que o método recursivo aumenta o tempo para execução de forma exponencial. No entanto essa técnica é muito utilizada na programação, principalmente quando há a necessidade de simplificar um problema, de forma que o mesmo seja dividido em subproblemas, minimizando a manutenção do código. Recursividade é um método chave para construção de muitos algoritmos importantes, e uma parte fundamental da programação dinâmica (GOULARTE, 2015).

Para fins de interação com o usuário, foi feito o método **fimEmotion()** baseado em uma cena do filme Star ASCIIMATION no terminal (JASEN, 2015). Tal cena não é completamente compatível para impressão do java, em detrimento disso, foram necessárias algumas adaptações. Foi trocado o caracter ‘ \ ’ por ‘ | ‘ ou ‘ [‘ dependendo do caso, de maneira a formar o desenho dos robôs de Star Wars de um jeito criativo e compreensível.

O método sem otimização resolveu todos os cálculos de forma eficiente, já quando houve a otimização básica conseguiu-se eliminar alguns passos recursivos. Em conseguinte, com a otimização avançada a rapidez e eficiência dos cálculos melhorou ainda mais, pois, reduziu a ordem da matriz fazendo com que os passos recursivos não fossem tão solicitados, o que contribuiu na resolução do determinante de forma mais ágil e efetiva.

No entanto, para matrizes de ordem menor que 5x5 não é perceptível grande vantagem em relação ao método sem otimização. Ordens maiores que esta apresentam um bom resultado nos métodos de otimização, mas os melhores resultados computados foram em matrizes de ordem 5 (cinco) e 7 (sete) que reduziram o tempo em praticamente dois terços (2/3) do habitual. Como as matrizes maiores necessitam de passos recursivos exponencialmente, os métodos otimizados fazem uma diferença expressiva por conta de reduzirem muito o tamanho da matriz. Com essa otimização o tempo de execução passa a ser uma ordem de grandeza menor na potência, dependendo da ordem da matriz, o que induz uma vantagem inegável quando se trata de eficácia e eficiência em termos de otimização.

3. Referências

Domingues, H. H. (2018) "Origens dos sistemas lineares e determinantes" In: Só Matemática, https://www.somatematica.com.br/historia/sistemas.php. Consultado em 13/06/2018 às 02:14.

Goularte, C. (2015) "Função recursiva (Recursive function)" In: Código Master, http://www.codigomaster.com.br/desenvolvimento/funcao-recursiva-recursive-function . Consultado em 11/06/2018 às 19:36.

IGM, Equipe. (2009) "Propriedades dos Determinantes" In: Instituto Goiano de Matemática, http://www.igm.mat.br/aplicativos/index.php?option=com_content&view=article&id=148%3Apropriedadesdosdetermnantes&catid=41%3Aconteudosal&Itemid=38. Consultado em 11/06/2018 às 18:49.

Jansen, S. (2015) "Star ASCIIMATION" In: http://www.asciimation.co.nz/. Consultado em 1/06/2018 às 20:30.

Ribeiro, A. (2017) "Matrizes e Determinantes – Propriedades dos determinantes" In: Mundo Educação, https://mundoeducacao.bol.uol.com.br/matematica/propriedades-dos-determinantes.htm. Consultado em 11/06/2018 às 15:49.

Virtuous. (2018) "Deterinantes" In: Só matemática, https://www.somatematica.com.br/emedio/determinantes/determinantes4.php. Consultado em 11/06/2018 às 16:00.

Oliveira, G. A. (2015) "Teorema de Laplace" In: Brasil Escola, https://brasilescola.uol.com.br/matematica/teorema-laplace.htm. Consultado em 13/06/2018 às 01:52.

Oliveira, G. A. (2016) "Matrizes e Determinantes – Regra de Chió nos cálculos dos determinantes" In: Mundo Educação, https://mundoeducacao.bol.uol.com.br/matematica/regra-chio-nos-calculos-dos-determinantes.htm. Consultado em 11/06/2018 às 18:49.