

PROJETO COMPUTACIONAL #6 – MS993/MT404 – 2S2016 – IMECC/UNICAMP

Matemática Aplicada (Não haverá atividades extra para o projeto #6)

Atenção: Este projeto #6 poderá ser desenvolvido individualmente ou com no máximo 3 (três) estudantes.

Atividade(*)	Temas, palavras-chave	Disponibilização	Entrega
Projeto #6 (P6) David S. Watkins, Fundamentals of Matrix Computations, New Jersey: John Wiley & Sons (3 ed., 2010). Gene Golub and Charles Van Loan. Matrix computations, 3rd ed., Johns Hopkins Univ. Press (1996). Roger Horn and Charles Johnson. Matrix analysis, Cambridge, MA, Cambridge Univ. Press (1985).	The SVD and the Least Squares Problem, Applications, Image Compression (Audio and Video Compression ?), The Discrete Least Squares Problem, Orthogonal Matrices, Rotators, and Reflectors, Solution of the Least Squares Problem, The Gram-Schmidt Process, QR Decomposition, Sensitivity of the Least Squares Problem, Francis algorithm, Least Squares Solution from Normal Equations, Norms: Quantifying Error and Distance, Calculating the QR-factorization - Householder Transformations, Rank Deficiency: Numerical Loss of Orthogonality, Computations in floating point and exact arithmetic.	18/Nov	9/Dez

The primary learning goal of the project. We are primarily focused in the SVD and the QR procedures for numerically solving Least Squares problems, along with applications to image compression. ((*) **check also directory auxFILESp6**)

TASK 1. Como tratar a compressão de imagens coloridas, e em preto e branco, usando SVD ?

Siga as instruções nas dicas 1 e 2 logo abaixo. Utilize várias imagens coloridas em seu projeto para explorar diferentes cores. Imagens/fotos de natureza são muitas boas para essa finalidade. Por exemplo, digite “nature” na opção “image” do seu navegador para encontrar um conjunto de imagens de sua preferência. É obrigatório encontrar ao menos mais uma aplicação concreta para as fatorações SVD e QR e reportar neste projeto computacional.

Dica 1: Podemos usar “Singular Value Decomposition” (SVD) para aplicação de compressão de imagens; os códigos executados em sala podem ser encontrados no link a seguir http://www.ime.unicamp.br/~ms512/sites/default/files/material-didatico/MS512-Applicacao_SVD.pdf .

Clique nesse link acima e depois siga as instruções para reproduzir os experimentos numéricos e aplicações apresentados (note que no primeiro link apresentado no arquivo acima tem uma explicação sobre os gráficos gerados em sala junto com uma explicação adicional sobre o processo de compressão de imagens, verifique !). Observa-se que no caso discutido em sala de aula, foram consideradas figuras em escala de cinza (note que no link acima as figuras “tiger” e “cat” são coloridas, mas foram convertidas antes da aplicação SVD).

Dica 2: Assim, uma pergunta natural seria: como tratar a compressão de imagens em figuras coloridas ? Para ajudar a responder essa pergunta, estude o material “Singular Value Decomposition, Investigation in Mathematics, Singular Value Decomposition, Rebekah Wheeler 10214927, 28th March 2013”, que está disponível no link seguinte (ver também em auxFILESp6): <http://mathsforall.co.uk/documents/Singular%20Value%20Decomposition%20by%20Becky%20Wheeler.pdf>

Considere também uma leitura do material disponível no link que segue logo abaixo (ou ver também [SVD_paper.pdf](#) no diretório **auxFILESp6**):

http://msemac.redwoods.edu/~darnold/math45/laproj/fall2006/iancraig/SVD_paper.pdf **Remark:** In this moment, you should to do a research and verify if it is possible to use/extend ideas of image compression but now to **audio and video Compression** by using the SVD framework or some related theory of computational linear algebra as building block.

TASK 2. Essa atividade refere-se a “**fatoração/decomposição QR**” e ao “**algoritmo QR (ou algoritmo de Francis)**”. Pede-se fazer os seguintes itens:

(a) Veja o diretório “auxFILESp6” e procure por “materialQR”; ver também “Vídeos científicos e outros materiais” no site http://www.ime.unicamp.br/~ms512/sites/default/files/material-didatico/MS512-Aplicacao_SVD.pdf

Em seguida, execute os códigos em matlab “runQRgivensRot.m” e “runQRhouseholderRef.m”; esses códigos realizam a decomposição $A=QR$ via transformações ortogonais pelas fatorações de Givens e Householder, respectivamente. Uma vez colocado a dimensão da matriz “A”, esses códigos realizam a fatoração QR. Para efeito de comparação, ao final, é mostrada a matriz “R” sobreposta na matriz “A”, computadas pelos programas acima e também uma matriz “R” calculada pela função do MATLAB. Note que na execução desses programas é mostrado a sequência de matrizes a partir de “A” até “R”; veja que a matriz “A” é gerada de forma aleatória.

Modifique cada um dos programas “runQRgivensRot.m” e “runQRhouseholderRef.m” para imprimir, em cada nível da iteração, suas respectivas matrizes de transformação ortogonal, a saber, “Transformação Givens” e “Transformação Householder”.

TASK 3. Considere os dois textos a seguir (**disponíveis no diretório “auxFILESp6”**):

- David S. Watkins, Understanding the QR algorithm *SIAM REVIEW*, 24(4) (1982) 427-440.
- David S. Watkins, The QR Algorithm Revisited, *SIAM REVIEW*, 50(1) (2008)133-145.

Com base nesses dois textos, faça uma descrição dos principais argumentos para demonstração de convergência do método QR (**algoritmo de Francis**) via sua equivalência com o método de iterações simultâneas, que é fundamentalmente uma generalização dos métodos das potências.

Dicas:

- Faça primeiro a leitura do “David S. Watkins, Understanding the QR algorithm *SIAM REVIEW*, Vol. 24, No. 4, pp. 427-440, 1982”, que resume bem toda a discussão para estabelecer a equivalência entre “Simultaneous Iteration \Leftrightarrow QR Algorithm”.
- No livro “L. N. Trefethen & D. Bau III – Numerical Linear Algebra, SIAM, 1997”, procure por “Simultaneous Iteration \Leftrightarrow QR Algorithm”, Lecture 28, Theorem 28.2 (página 215), Theorem 28.3 (página 216) e Theorem 28.4 (página 218).
- No livro “D. S. Watkins, Fundamentals of Matrix Computations, New Jersey: John Wiley & Sons, SEGUNDA EDIÇÃO (2002), ver “Section 6.2 Subspace Iteration, Simultaneous Iteration, and the QR Algorithm”, páginas 420-427.