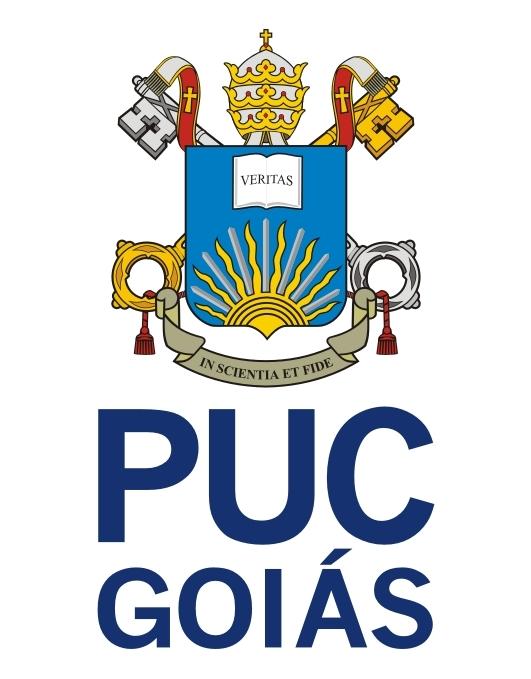
PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO

GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



Ítalo Fernandes Gonçalves

**AED - DECOMPOSIÇÃO DE CHOLESKY E SPLINES CÚBICAS**

Goiânia, 2018

Ítalo Fernandes Gonçalves

**Decomposição de Cholesky e splines cúbicos**

Relatório técnico científico apresentado com atividade extra disciplinar - AED na disciplina CMP1058 - fundamentos IV, no curso de Engenharia da Computação, na Pontifícia Universidade Católica de Goiás.

Goiânia, 2018

**RESUMO**

Este trabalho apresenta a decomposição de Cholesky juntamente com um exemplo de aplicação, um sistema linear simples, que será resolvido por meio desta técnica apresentada, no que se refere a terceira aed. Ainda neste, será apresentado também uma aplicação de spline cúbico, contemplando a quarta aed.

**INTRODUÇÃO**

Será apresentado a solução de um sistema linear, usando como base a decomposição de Cholesky. Também será apresentado a técnica de Splines cúbicos, em uma de suas aplicações mais clássicas.

**DESENVOLVIMENTO -** *decomposição de cholesky*

A decomposição de Cholesky decompõe uma matriz A na forma A = LL’ , onde L é uma matriz triangular inferior com elementos da diagonal principal estritamente positivos.

Uma matriz A é dita definida positiva se A é simétrica e se x’ \* Ax > 0, para todo x != 0.

**Teorema**  (a) Se A[n×n] é uma matriz simétrica, então A possui ‘n’ autovalores reais e ‘n’ autovetores ortonormais.

(b) Uma matriz simétrica A é definida positiva se, e somente se, seus autovalores são reais e positivos.

(c) Uma matriz simétrica A é definida positiva se, e somente se, cada uma de suas submatrizes principais tem determinante positivo.

(d) Uma matriz simétrica A é definida positiva se, e somente se, o processo de eliminação de Gauss pode ser realizado sem permutação de linhas ou colunas e tem todos os elementos pivots positivos.

O teorema de Cholesky diz que uma matriz simétrica A é definida positiva se, e somente se, pode ser fatorada como LL’ , onde L é uma matriz triangular inferior com elementos positivos na diagonal.

Temos que:

A = L \* L’, onde L’ é a matriz transposta de L.

**CÁLCULO DA MATRIZ ‘L’**

Seja a matriz genérica L e L’, multiplicando ambas e igualando a matriz genérica A e posteriormente isolando o termos A(i,j), chegamos a 2 casos, quando i é igual a j e quando i é diferente de j. Isto é:

Então fazemos estes casos para a matriz triangular inferior de A, e assim geramos a matriz L.

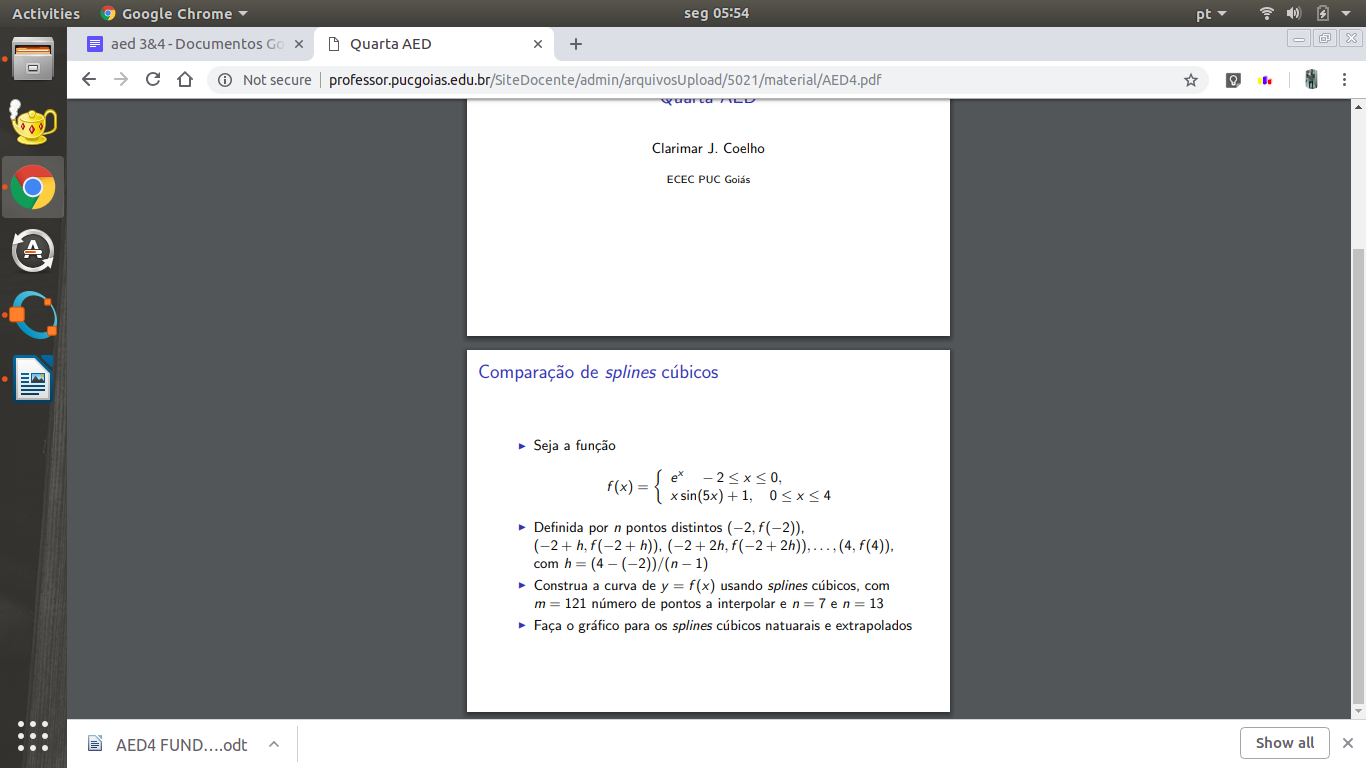
Segue em anexo o código feito no octave, juntamente com o printscreen da tela, como definido na terceira AED.

**RESULTADOS**

Foi possível resolver o sistema linear com o método de decomposição apresentado. Ele pode ser enxergado como uma versão modificada da eliminação de gauss. Foi confirmado os resultados no próprio octave, resolvendo o sistema linear de forma direta.

**SPLINES CÚBICOS - QUARTA AED**

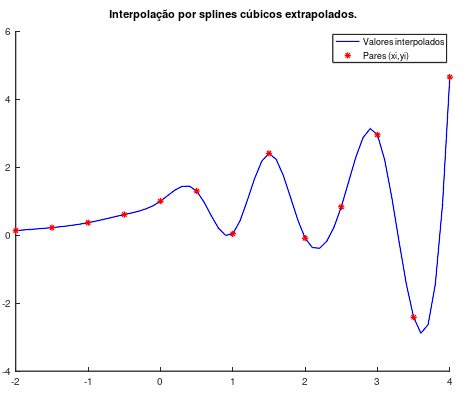
Seja o problema, descrito a seguir:



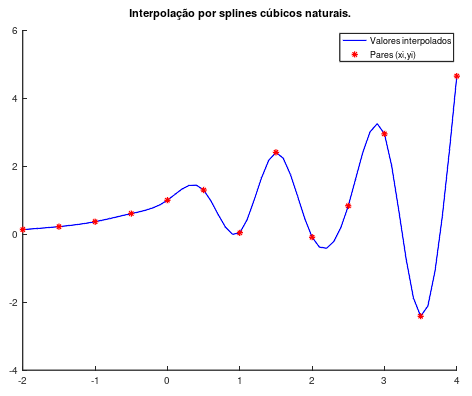
**RESULTADOS**

Foi usado o software octave para construção dos gráficos, com a função já existente neste próprio. O algoritmo foi também implementado, mas não apresentou resultados concisos.

**Splines cúbicos Extrapolados**

****

**Splines cúbicos naturais**

****

**REFERÊNCIAS BIBLIOGRÁFICAS**

<http://wwwp.fc.unesp.br/~arbalbo/Iniciacao_Cientifica/sistemaslineares/teoria/3_Met_Cholesky.pdf>

<http://www.dma.uem.br/kit/arquivos/arquivos_pdf/cholesky.pdf>

**ANEXOS**

Implementação do método de CHOLESKY no octave:

***function*** *[L] = cholesky(A)*

*[l,c] = size(A);*

*tam = l;*

*% inicializa matriz L*

***for*** *j = 1 : tam*

***for*** *i = 1 : tam*

*L(i,j) = 0;*

***end***

***end***

*% realiza cálculos*

***for*** *i = 1 : tam*

***for*** *j = 1 : i*

***if*** *i == j % CASO 1 -> i == j*

*temp = 0;*

***for*** *k = 1 : (i-1)*

*temp = temp + L(i,k)\*L(i,k); % formula*

***end***

*L(i,i) = sqrt(A(i,i) - temp);*

***else*** *% CASO 2 -> i != J*

*temp = 0;*

***for*** *k = 1 : (j-1)*

*temp = temp + L(i,k)\*L(j,k);*

***end***

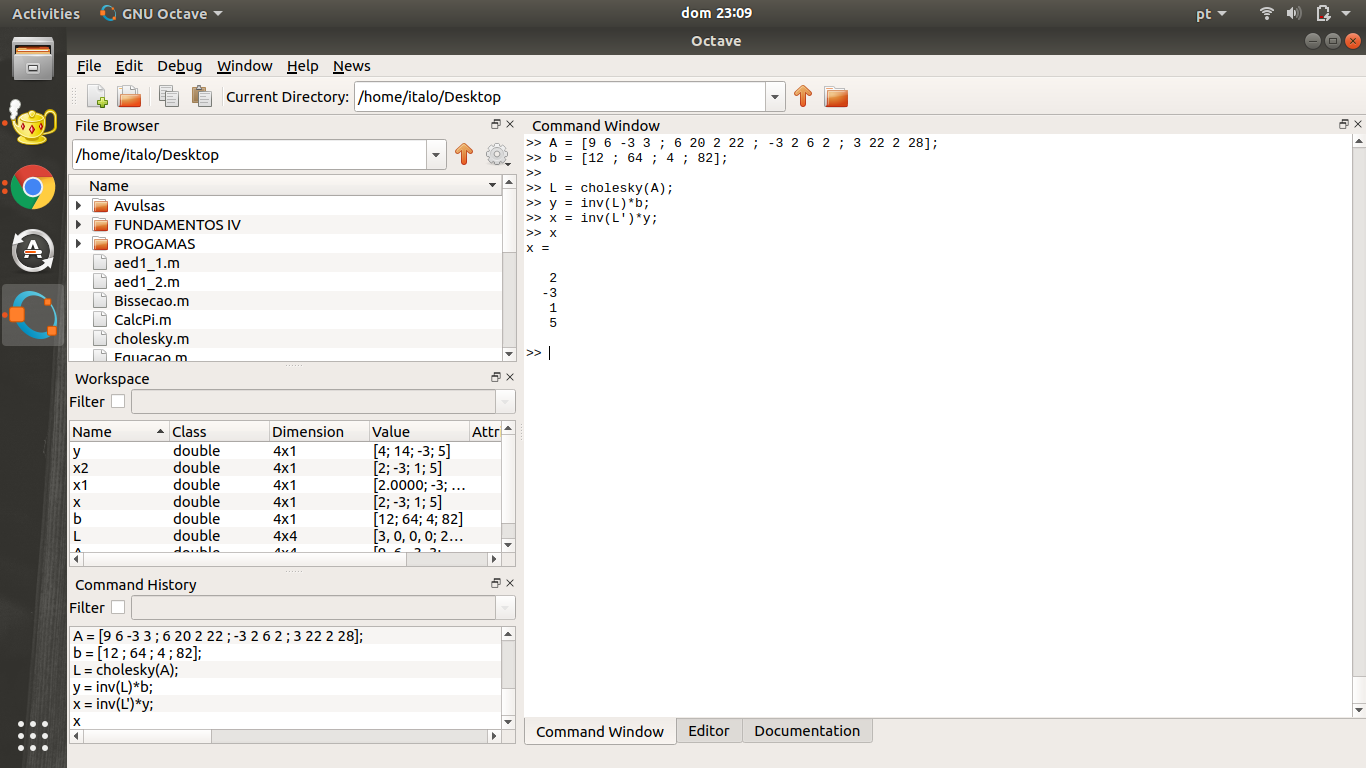
*L(i,j) = (A(i,j)-temp)/L(j,j); % formula*

***end***

***end***

***end***

***end***

******