TD 5 - Algèbre linéaire : factorisation de Cholesky

La décomposition de Cholesky d'une matrice symétrique, définie positive est une adaptation de la décomposition LU.
Plus précisément, on peut énoncer le résultat suivant :

proposition
Soit A une matrice symétrique, définie positive i.e. $A^T = A$ et pour tout $V \in \mathbb{R}^n$, $V \neq 0$, $V^T AV > 0$. Il existe alors une unique matrice triangulaire inférieure, B, dont les éléments diagonaux sont strictement positifs et telle que : $A = BB^T$.

Pour déterminer la matrice B, on effectue un calcul direct du produit BB^Tavec :

$$B = \begin{pmatrix} b_{11} & 0 & & 0 \\ b_{21} & b_{22} & \ddots & \\ \vdots & \vdots & \ddots & \vdots \\ & & & 0 \\ b_{n1} & b_{n2} & & b_{nn} \end{pmatrix}$$

On obtient les égalités :

$$a_{ij} = \sum_{k=1}^{j} b_{ik} b_{jk} \ \text{ pour } 1 \leq i \leq n \, \text{et} \, 1 \leq j \leq i.$$

On en déduit :

$$b_{ii}^2 = a_{ii} - \sum_{k=1}^{i-1} b_{ik}^2$$

$$b_{ki} = b_{ii}^{-1} \left(a_{ki} - \sum_{\ell=1}^{i-1} b_{k\ell} b_{i\ell} \right) \ \text{ pour } i < k \leq n$$

Travail demandé :
Programmer la décomposition de Cholesky.

1. Pour ce qui concerne les sommes $\sum_{k=1}^{i-1} b_{ik}^2$ et $\sum_{\ell=1}^{i-1} b_{k\ell} b_{i\ell}$ intervenant dans les formules ci-dessus on pourra utiliser une instruction de la forme : sum (U.*V) ce qui évite d'écrire une boucle for ... end.
2. Le programme doit afficher la matrice B.
   Effectuer le calcul BB^T dans la console pour vérifier le résultat.

Script Scilab

```
// factorisation de Cholesky
// -----------------------------------------------------
n=evstr(x_mdialog('dimension',['n'],['3']));
A=ones(n,n)+eye(n,n);
B=zeros(n,n);
// -----------------------------------------------------
B(1,1)=sqrt(A(1,1));
for i=1:n-1
    for k=1:n-i
        B(i+k,i)=(A(i+k,i)-sum(B(i+k,1:i-1).*B(i,1:i-1)))/B(i,i);
    end
    B(i+1,i+1)=sqrt(A(i+1,i+1)-sum(B(i+1,1:i).^2));
end
// affichage de la matrice triangulaire inférieure B
disp('matrice B :');
disp(B);
// vérification que le produit B*B' redonne A
disp('matrice A :');
C=B*B';
disp(C);
```