Rapport sur la résolution des systèmes tridiagonaux avec la méthode de Thomas

---

@Author: Massiles Ghernaout, L3 informatique, Université du Havre.

Implémentation des classes:

`Mat3Diag.java:` Cette classe contient les constructeur demandés, mais aussi une méthode pour le calcule du produit entre un vecteur et une matrice tridiagonale. Chaque matrice tridiagonale instanciée avec cette classe aura un tableau de trois lignes et n == dim(matrice) colonnes.

La première ligne du tableau est pour la sur-diagonale. La deuxième est pour la diagonale, et la dernière est pour la sous-diagonale. D'ailleurs, la méthode de calcul de produit, exploite cette conception pour faciliter et accélérer le calcul.

`Thomas.java:` Cette classe hérite de la classe abstraite `SysLinAbstract.java`, et donc c'est une classe qui représente un système linéaire. Ici, le système fait entrer en jeu une matrice tridiagonale, et pour cela on peut utiliser le constructeur qui prend une matrice carré équivalente à la matrice tridiagonale de notre système (avec des coefficients à 0), et aussi un tableau de trois lignes et n == dim(matrice) colonnes qui contiendront que les diagonales. Cela est une conséquence du fait que Le constructeur de SysLin vérifie si la matrice passée en paramètre est bien carrée.

Le tableau des diagonales est un attribut d'instance, et son rôle est de simplifier le calcul pour les tests.

Implémentation des tests :

Déroulé:

- Instancier une matrice Mat3Diag d'un ordre donné.
- Instancier une matrice carrée équivalente à la matrice tridiagonale.
- Instancier un vecteur qui sera notre second membre.
- Instancier la classe Thomas avec la matrice carrée équivalente et le second membre.
- Résolution du système en passant par la méthode de résolution de la classe Thomas.java
- Calcule des normes de Ax-b, (la soustraction du produit de la matrice tridiag et du vecteur solution d'une part, et du second membre d'autre part)
- Affichage de l'état des tests.

Résultat des tests:

Notes:

Le cas de test utilisé est l'exemple du TD 2.

Les résultats sont positifs, ce qui veut dire que la norme L1 et L infinie de Ax-b est bien inférieure au epsilon numérique prédéfini.

```
b :={-2, -2, -2, 23}

A :=
                    {2,    -1,  0,    0},
                    {-1,    2, -1,    0},
                    {0,    -1,  2,   -1},
                    {0,     0, -1,    2},


tridiag(A) :=
                    {-1, -1, -1,0},     // sur diagonale (cn  = 0)
                    {2, 2, 2, 2 },      // diagonale
                    {0,-1, -1, -1},     // sous diagonal (a0 = 0)


# après la résolution

solution := {0.9999999, 3.9999999, 8.9999999, 16.0 }


# calcul de Ax-b et des normes

Ax := { -2.0, -2.000000000000001, -2.0000000000000018, 23.0 }

Ax-b := {0.0, -8.881784197001252E-16, -1.7763568394002505E-15, 0.0 }

-> norme L1       < epsilon
-> norme L infini < epsilon
```