

Résultats des tests (résolution de systèmes linéaires)

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

Note > Les classes Matrice et Vecteur ont été testées lors de l'étape précédente.

SysDiagonal

A:= {3, 0, 0, 0} {0, 7, 0, 0} {0, 0, -2, 0} {0, 0, 0, -5}

b := { 1, 7, -6, 8 }

Déroulé:

- résolution $Ax=b$, puis calcul des trois normes pour $Ax-b$
- Les normes sont bien inférieures au epsilon, donc le test est positif.

Explication brève:

Pour résoudre ce système, on crée une instance de la classe SysDiagonal avec le tableau de valeurs ci-dessus, puis on appelle la méthode de résolution sur cette instance.

La méthode de résolution de la classe SysDiagonal ne vérifie pas si la matrice du système est bien diagonale, elle se comporte comme si c'est le cas. Cela, nous permettra de l'intégrer plus facilement dans la fonction de résolution partielle de la classe Helder.

Attention: la méthode de résolution peut émettre une exception si une division par 0 est effectuée.

Puis, une fois que le vecteur solution est calculé. On calcule le produit de A et x (vecteur solution), et puis on lui soustrait b, pour enfin convertir le résultat vers un vecteur afin de calculer les normes. Si les normes sont supérieures au epsilon numérique défini, alors un message d'erreur est affiché dans la console.

SysTriangInf

A:= {3, 0, 0} {7, 2, 0} {5, 10, 3}

b := {1, 3, 2}

Déroulé:

- résolution $Ax=b$, puis calcul des trois normes pour $Ax-b$
- Les normes sont bien inférieures au epsilon, donc le test est positif.
- idem pour SysTriangInfUnite

Explication brève:

Pour résoudre le système, on crée une instance de SysTriangInf avec la matrice spécifiée et le vecteur b. Puis, on fait appel à la méthode de résolution sur cette instance.

Attention: la méthode de résolution peut émettre une exception si une division par 0 est effectuée.

La méthode de résolution ne vérifie pas que la matrice est bien TriangInf pour une meilleure intégration avec la classe Helder.

Puis, comme expliqué précédemment, on calcule Ax-b et puis on transforme le résultat en un vecteur, afin de pouvoir calculer les normes.

Pour ce qui est de la classe SysTriangInfUnite, la méthode de résolution est strictement la même, sauf que dans celle-ci on ne divise pas par 1.

SysTriangSup

A:= {3, 4, 5} {0, 1, 7} {0, 0, 8}

b := {3, 7, 2}

Déroulé:

- résolution Ax=b, puis calcule des trois normes pour Ax-b
- Les normes sont bien inférieures au epsilon, donc le test est positif.
- idem pour SysTriangSupUnite

Explication brève:

Pour résoudre le système, on crée une instance de la classe SysTriangSup et puis on appelle la méthode de résolution. Celle-ci aussi ne vérifie pas si la matrice est bien SysTriangSup.

Attention: la méthode de résolution peut émettre une exception si une division par 0 est effectuée.

Une fois que le vecteur résultat est trouvé, on calcule Ax-b, puis on transforme celui-ci en une instance de Vecteur pour pouvoir calculer les normes.

Pour ce qui est de la classe SysTriangSupUnite, la méthode de résolution est strictement la même, sauf que dans celle-ci on ne divise pas par 1.

Helder

Premier Test, $A^2x=b$ A:= { 1, 2, 3, 4} {-2, 3, 6, -1} { 3, 4, -6, 2}
{-5, 2, 3, -1}

b := {3, 7, 2, -1}

Déroulé:

- factorisation de A en LDR.
- Vérification que $A = LDR$

Deuxième Test, $A^2x=b$ $A := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$b := \{3, 5, 8\}$

Déroulé:

- factorisation de A en LDR.
- Résolution du système $Ay=b$, où $y = Ax$
- Puis, résolution du système $Ax=y$
- Calcule de la norme de A^2x-b , et vérifier qu'elle sont bien inférieures epsilon

Explication brève:

La classe Helder contient trois méthodes principales: - factorLDR(), pour factoriser la matrice de l'instance actuelle en LDR. - résolutionPartielle(), pour résoudre le système partant de la factorisation LDR. - resolution(), qui factorise puis résout le système, et cela en appelant les deux autres méthodes.

Pour factoriser la matrice du système actuel en LDR, l'algorithme vu en TD est utilisé. Pour résoudre partant de la factorisation LDR, il suffit d'utiliser les classes précédemment définies et cela en passant la matrice du système actuel vers leurs constructeurs.

i.e, Pour résoudre $Ax=b$: - On factorise A en LDR, donc $Ax=b$ devient $LDRx=b$
 - On résout $Lz=b$, avec SysTriangInfUnite - On résout $Dy=z$, avec SysDiagonal -
 On résout $Rx=y$, avec SysTriangSupUnite

Puis, on calcule la norme de $Ax-b$, et on vérifie qu'elle est inférieure à epsilon.

Aussi, pour bien tester la classe Helder, on a utilisé $A^2x=b$ comme système, pour le résoudre: - On factorise A en LDR, donc le système devient $LDR(LDRx)=b$ -
 On résout $LDRw=b$ - Puis, on résout $LDRx=w$

Pour y arriver, on utilise la méthode setSecondMembre pour changer le second membre du système, afin de réutiliser la décomposition de A.

Note: - D'autres tests ont été effectués au fur et à mesure pour vérifier le bon fonctionnement des autres méthodes présentes dans les différentes classes.