
Algèbre linéaire avancée II
printemps 2016

Exercice de Programmation 1


Exercice 1. *On met en pratique l'algorithme 1.2 des notes du cours. À partir du code fourni ci-dessous, votre tâche c'est de d'ajouter la fonction suivante : obtenir une matrice non singulière P telle que $P^T \cdot A \cdot P$ soit une matrice diagonale, étant donnée une matrice symétrique $A \in K^{n \times n}$, où la caractéristique de K n'est pas 2.*

On utilise Sage. Si vous ne connaissez pas bien Python ou Sage, ne vous en faites pas, car l'exercice est assez simple. Quasiment tout est déjà préparé pour vous, il ne vous reste qu'à remplir quelques lignes de code.

Comment s'y mettre avec Sage

On veut utiliser *SageMath* (ou *Sage* en abrégé), un logiciel optimal pour un grand nombre de calculs standard en mathématiques. Sage nous permet, par exemple, d'exécuter des opérations sur les lignes et colonnes de matrices, les inverser, etc. Ses commandes sont nettement plus intuitives par rapport à d'autres langages de programmation.

Pour bien commencer, le présent guide vise à vous expliquer en tout détail la démarche à suivre, jusqu'à ce que vous soyez confortable avec Sage. Point avantageux : il ne vous est nécessaire d'installer aucun logiciel, car Sage fonctionne sur votre navigateur d'internet.

1. **Obtenez un compte** La première chose à faire c'est créer un compte. Sage étant gratuit, ceci n'est qu'une formalité. Pour ce faire, visitez <https://cloud.sagemath.com/> et choisissez n'importe quelle option. Soit vous créez un nouveau compte, soit vous vous connectez avec un compte Google, GitHub, ou même Facebook ou Twitter.
2. **Créez un nouveau projet** Dans le compte, on va commencer un nouveau projet. Un projet peut contenir plusieurs documents, comme un dossier. Cliquez sur le bouton *Projects* dans le côté supérieur gauche de la page; puis sur *Create new project...* Vous devez nommer votre projet (par exemple "DiagMat"). Il est optionnel d'ajouter une description du projet; on vous conseille d'y décrire ce que vous voulez faire, pour référence future (par exemple "diagonaliser une matrice avec l'algorithme du cours"). Après que vous cliquiez sur le bouton *Create project*, votre projet apparaîtra dans la liste.
3. **Démarrer le projet** Une fois le projet créé, vous devez dire à Sage qu'on va travailler la-dessus. Vous voyez, dans le côté droit du projet créé, le symbol , suivi de *Stopped*. Pour démarrer le projet, cliquez sur ce dernier. (Si ceci ne marche pas, jetez un coup d'œil sur *Settings*, dans le côté supérieur gauche. Ici vous avez plusieurs options de réglage, et en particulier vous pouvez enregistrer votre progrès. Néanmoins, le projet est enregistré automatiquement très souvent.)
4. **Ajouter du code au projet** Heureusement vous ne commencez pas avec un projet vide. Un bon morceau du programme est déjà écrit, et vous le trouvez sur <https://cloud.sagemath.com/projects/1519bdae-0dcd-42f8-8d04-69cfc2b0e9c1/files/>

[Diagonalize%20a%20Matrix%20via%20congruence%20operations.html](#) Lorsque vous ouvrez le lien, un nouveau projet apparaît comme onglet dans le haut de la page. Ceci n'est pas votre projet, donc vous ne pouvez pas l'éditer. Mais vous pouvez copier son contenu et le coller dans votre projet. Pour cela, cliquez sur le bouton *Copy Jupyter Notebook to your Project...* Un dialogue s'ouvre dans l'écran, avec le fichier à copier à gauche (ce fichier finit par *.ipynb), suivi de *Select a project...* et *Home directory*. Dans le menu déroulant au milieu, choisissez le projet que vous venez de créer comme destination. Pour finir, cliquez sur *Copy 1 item*. On n'a pas besoin du fichier html. Si vous cliquez sur votre projet *et rafraîchissez la page*, vous y trouverez le nouveau fichier.

5. **Commencer** Vous voyez maintenant l'exercice qui a été préparé pour vous. Il s'agit des morceaux de texte, interrompus par de boîtes grises qui contiennent du code de programmation. Ces boîtes sont appelées des cellules (en anglais *cells*). Une cellule contient du code de programmation, qui est le langage avec lequel on donne des instructions à l'ordinateur. En cliquant sur une cellule, vous pouvez modifier ce contenu. Vous pouvez faire l'ordinateur exécuter le contenu d'une cellule en cliquant sur *Cell > Run* dans le côté supérieur de la page (la "boîte à outils"). Le résultat (le "output") apparaîtra au dessous de la cellule sélectionnée. Pourtant dans ce fichier, les cellules contiennent des morceaux de code qui sont censés être lus et exécutés comme un tout. Ainsi, on veut utiliser le bouton *Cell > Run All*.
6. **Votre tâche** Le code est presque complet, il ne reste que deux sections à compléter. Essayez de comprendre toutes les commandes; elles sont pour la plupart très intuitives. Votre tâche c'est d'insérer les bons morceaux de code, afin qu'on obtienne la matrice P correcte comme résultat. Si vous le faites correctement, la toute dernière ligne du résultat devrait lire "The code seems to be correct".
7. **Davantage sur Sage** Pour se familiariser avec Sage et ses commandes, et connaître tout ce que vous pouvez faire avec, il y a plusieurs guides en ligne, par exemple http://doc.sagemath.org/html/en/a_tour_of_sage/, et <http://doc.sagemath.org/html/fr/tutorial/> (en français). Visitez aussi <http://www.sagemath.org/help.html> pour plus d'information. Cela pourrait être une bonne idée d'installer Sage dans votre ordinateur; c'est à vous de décider.