Méthodes numériques – TD2

Premiers pas avec MatLab (suite)

Le but de cette séance est d'aller une peu plus loin dans la compréhension des mécaniques de MatLab. Pour le compte-rendu (commun avec le TD précédent), vous pouvez toujours prendre des captures d'écran et/ou sauvegarder les commandes que vous avez rentrées.

1 Grandes matrices

*La fonction **rand** permet de créer des matrices remplies aléatoirement. Avec les fonctions **tic** and **toc**, il est possible de mesure le temps d'exécution d'un code. Par exemple tic;<code> ; toc.*

1. Créez une matrice carrée A de taille 1000x1000, ainsi qu'un vecteur b de taille 1000.
2. Calculer la solution du système résultant avec la fonction directe A\b et en utilisant les différentes factorisations. Mesurez le temps de calcul.
3. Comment expliquer ces différences ?
4. Utilisez A pour créer une matrice symétrique S à diagonale positive.
5. Calculer la solution du système résultant avec la fonction directe S\b et en utilisant les différentes factorisations. Mesurez le temps de calcul.
6. Comment expliquer ces différences ?

2 Minimisation aux moindres carrés – suite de TD1.7

*Nous allons ici explorer les limites de l'approche développée dans la question TD1.7 pour l'approximation des données avec des polynômes. Pour cela nous utiliserons les nouvelles données data2. Dans tous les cas, on pourra se référer à la fonction **polyfit** qui implémente l'approximation par un polynôme et **polyval** qui permet son évaluation.*

1. Affichez sur une même figure les données d'origine et l'approximation (cf la documentation de plot).
2. Que constatez-vous lorsque l'on augmente le degré maximal ? Vérifiez votre conditionnement.
3. Pour améliorer la précision, une première idée est d'éviter d'avoir des matrices trop grandes. Pour cela nous utiliserons le processus suivant
   Trouver la meilleure approximation avec un polynôme constant $P_0 = a_0$
   Pour chaque k>= 1, trouver la meilleure approximation avec $P_k=a_k*x^k+b_k*P_{k-1}$
   1. Implémentez cette solution
   2. Expliquez le comportement
4. Une autre solution se base sur le fait que les plus grandes erreurs ont lieu sur les bords.
   1. Calculez l'approximation avec le polynôme $P_k= \Sigma\, a_k*(x-m)^k$ avec m moyenne des valeurs x en entrée.
   2. Transformez le résultat obtenu sous la forme $P_k= \Sigma\, b_k\, x^k$
   3. Que constatez-vous.
5. Que proposeriez-vous pour améliorer l'approximation ?