Exercice 1 : Méthodes itératives de base

On cherche à résoudre le système $AX = b$ par des méthodes itératives classiques. On testera les fonctions avec A la matrice du Laplacien discret de taille $n \times n$ et b un vecteur de taille n. On prendra $n = 100$.

a) Programmer la méthode de Jacobi. Tracer le logarithme du critère d'arrêt en fonction des itérations.

b) Programmer la méthode SOR. Tracer le nombre d'itérations nécessaire à la convergence en fonction du paramètre de relaxation ω.

c) Programmer la méthode SSOR. Tracer le nombre d'itérations nécessaire à la convergence en fonction du paramètre de relaxation ω.

d) Comparer la vitesse de convergence des trois méthodes précédentes (pour SOR et SSOR on prendra les paramètres optimaux).

e) On cherche à présent à résoudre $AX = 0$. Prendre comme premier itéré le vecteur :

$$X^0 = \sin(\pi x) + \sin(10\pi x) + \sin(30\pi x),$$

où x discrétise l'intervalle $[0, 1]$ et observer chaque X^i obtenu avec la méthode de Jacobi. Expliquer l'adjectif "lisseur" utilisé pour décrire la méthode de Jacobi.

Fonctions Scilab utiles : `triu`, `tril`, `diag`.

Exercice 2 : Méthodes de gradient

On va ici programmer puis comparer les performances de quelques méthodes de gradient pour résoudre l'équation $AX = b$.

a) Programmer la méthode du gradient à pas constant.

- **i)** Pour $A = [2, 0; 0, 1/10]$ et $b = 0$:
  - i. Quelles valeurs de α (le pas) sont admissibles ?
  - ii. Pour $\alpha = 0.98$ puis $\alpha = 0.5$, représenter les différents itérés.
  - iii. Tracer le nombre d'itérations nécessaire à la convergence en fonction de α. Quel est le paramètre optimal ? Comparer avec la valeur optimale théorique.
- **ii)** Pour A, la matrice du Laplacien de taille $n = 100$, montrer la norme du résidu en fonction des itérations en échelle log.

b) Programmer la méthode de gradient à pas optimal. Reprendre les deux cas-tests de la question **a)**.

c) Programmer la méthode du gradient conjugué. Reprendre les deux cas-tests de la question **a)**. Utiliser ensuite la fonction Scilab `pcg`.

Fonctions Scilab utiles : `eigs`, `fcontour`, `champ`.