# I Vade-Mecum

**Règles :** Les travaux sont effectués en binôme : un travail copié ou effectué en collaboration entre $N$ binômes divise les notes des binômes concernés par $N$.
Cette première partie du projet se déroule sur 3h30, de 9h à 12h30. Le rapport et les codes sources devront être envoyés par mail (ou rendu sur papier pour les rapports manuscrits) **avant 12h30**.

**Objectif et contenu :** Le projet consiste en la résolution mathématique d'un problème et sa mise en oeuvre informatique. Le **rapport** devra donc d'une part répondre aux questions théoriques posés dans l'énoncé et d'autre part expliquer les choix d'implémentation. Ce deuxième point nécessite tout d'abord la description précise des fonctions implémentées (les types des arguments et les propriétés mathématiques qu'ils doivent vérifier, la définition mathématique de la sortie, la description du fonctionnement de l'algorithme, ...). Ensuite, les résultats obtenus dans les applications numériques devront être vérifiés, analysés et commentés.
Les **codes sources (fichiers sci et sce)** seront réalisés de manière modulaire, les variables auront des noms explicites, ou au moins univoque. On vérifiera qu'ils sont compilables et que les sorties obtenues à partir d'un historique vide sont celles attendues.
**NB : Il est recommandé de lire intégralement l'énoncé en premier lieu.**

# II Epitome

Les méthodes itératives utilisées pour résoudre les systèmes linéaires $Ax = b$, avec $A \in \mathcal{G}l_n(K)$ et $b \in \mathcal{M}_{n,1}(K)$, construisent une suite de vecteurs $(x^{(n)})$. Ces méthodes ont en commun le fait que chaque itération nécessite un nombre d'opérations du même ordre de grandeur que celui nécessaire à faire un produit matrice $\times$ vecteur.

Si la suite $x^{(n)}$ converge vers la solution $x$, on approche $x$ numériquement par :
— soit $x^{(n_0)}$ où $n_0$ est fixé a priori,
— soit par le premier élément $x^{(k)}$ vérifiant un test d'arrêt.
Le test d'arrêt dit "standard" est :

$$\frac{\left\|Ax^{(k)} - b\right\|}{\|b\|} \leqslant \varepsilon$$

où $\varepsilon$ est fixé assez petit (par exemple $10^{-6}$) et $\|.\|$ une norme vectorielle. On utilise en général $\|.\|_\infty$ ou $\|.\|_2$. En posant $e^{(k)} = x - x^{(k)}$, l'inégalité précédente implique $\left\|e^{(k)}\right\| \leqslant \|\!|A^{-1}|\!\|\,\|\!|A|\!\|\,\|x\|\,\varepsilon$.

# III Méthodes de Jacobi, Gauss-Seidel, SOR

Pour les applications numériques, on définit $b_0 = {}^t(1, 1, 1, 1)$,

$$B_0 = \begin{pmatrix} -2 & -1 & -1 & -1 \\ -2 & -2 & -3 & -2 \\ -1 & 2 & 3 & 1 \\ 3 & 1 & 1 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 2 & 2 & 1 \\ -1 & 2 & 1 & 0 \\ 0 & 1 & -2 & 2 \\ 1 & -2 & -2 & 4 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 2 & -1 & 1 & 1 \\ 1 & -1 & 1 & 0 \\ -1 & -1 & 2 & 2 \\ 1 & -2 & -2 & 2 \end{pmatrix}.$$

**Méthodes itératives à nombre d'itérations fixe.**

1. Écrire une fonction `Iter` prenant en argument une matrice carrée $B$, un vecteur $b$, un entier $n$ et le vecteur inital $x^{(0)}$ et qui retourne le vecteur $x^{(n)}$, $n$-ème itéré de la méthode itérative
$$x^{(k+1)} = Bx^{(k)} + b.$$
Tester cette fonction avec $n = 100$, $x^{(0)} = {}^t(0,0,0,0)$, $b = b_0$, $B = B_0$ puis $B = \frac{1}{2}B_0$.
2. Écrire une fonction `Jac`, d'arguments $A$ et $b$, renvoyant la matrice d'itération $J$ et le vecteur $\tilde{b}$ de la méthode de Jacobi appliquée au problème $Ax = b$.
Écrire une fonction `GS` équivalente pour la méthode de Gauss-Seidel.
Tester ces deux fonctions avec $b = {}^t(1,1,1,1)$ et $A = A_1$ puis $A = A_2$.
3. Ecrire une fonction `IterJac` renvoyant le vecteur $x^{(n)}$ obtenu après $n$ itérations de la méthode de Jacobi appliquée au problème $Ax = b$ et initialisée à un vecteur $x^{(0)}$.
Écrire une fonction `IterGS` équivalente pour la méthode de Gauss-Seidel.
4. Calculer les 100ème, 200ème et 300ème vecteurs de la suite obtenue par chacune des méthodes (Jacobi et Gauss-Seidel) et pour chacun des deux problèmes $A_1x = b$ et $A_2x = b$. On optimisera le nombre de calculs.
Que peut-on déduire empiriquement de ces résultats ?
En utilisant la fonction `spec` de `Scilab` et en citant le théorème du cours utilisé, démontrer ces résultats.
5. Reprendre les questions 2, 3 et 4 pour la méthode SOR avec l'argument supplémentaire $\omega$. Trouver (théoriquement ou empiriquement) des valeurs de $\omega$ pour que la méthode SOR appliquée au problème $A_1x = b$ et $A_2x = b$ converge.

**Méthodes itératives avec test d'arrêt.**

6. Implémenter la résolution approchée de $Ax = b$ par les méthodes de Jacobi, Gauss-Seidel et SOR avec un test d'arrêt standard. On prendra soin de prescrire un nombre maximal d'itérations a priori (afin de forcer l'arrêt si la méthode diverge). Les fonctions devront retourner la solution approchée et le nombre d'itérations effectuées.
Tester chaque méthode sur un des deux exemples $A_1x = b$ et $A_2x = b$ pour laquelle la méthode considérée converge. Commenter.
7. Tester ces fonctions sur la matrice de discrétisation à 20 points du Laplacien en dimension 1 :
$$A = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \in \mathcal{M}_{20}(\mathbb{R}),$$
les autres paramètres étant laissés au choix.
Pour la méthode SOR, tester différentes valeurs du paramètre $\omega$. Déterminer expérimentalement la valeur du paramètre qui permet la convergence la plus rapide. Quelle est la vitesse de convergence asymptotique dans ce cas.