**Université de Lorraine - UFR MIM - 2015/2016**
**Cours MATLAB**

# MATLAB 3

*Algèbre linéaire numérique - Calcul scientifique*

J-P. Croisille

**1- Matrice du laplacien 1D**
1) Entrer en mode plein la matrice $T$ du laplacien en dimension 1, `t=2*diag(d1)-diag(d2,1)-diag(d2,-1);` avec `n=10; d1=ones(n,1); d2=ones(n-1,1);`
2) Calculer les valeurs propres de $T$.
3) Vérifier que les valeurs propres sont

$$\lambda_k = (2\sin(k\pi/(2(n+1))))^2 \;\; k = 1, n \tag{1}$$

4) Vérifier que le temps calcul croît linéairement en fonction de $N$. On utilisera les commandes `tic` et `toc`.

**2- Résolution de** $Tx = b$
1) Entrer la matrice du laplacien de taille $n = 10, 50, 100, 500, 1000$.
2) Entrer `b=ones(n,1);`.
3) Résoudre la système $Tx = b$ en utilisant `x=T\b;`.

**3- Matrice du laplacien 2D**
1) Le produit kroneckerien de deux matrices $A \in \mathbb{M}_{m,n}$ et $B \in \mathbb{M}_{p,q}$ est la matrice $C = A \otimes B \in \mathbb{M}_{mp,nq}$

$$C = A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & ... & a_{1,n}B \\ ... & & & \\ ... & & & \\ a_{m,1}B & a_{m,2}B & ... & a_{m,n}B \end{bmatrix} \tag{2}$$

Autrement dit,

$$C_{ii',jj'} = A_{i,j}B_{i',j'} \tag{3}$$

avec $1 \leq i \leq m$, $1 \leq i' \leq p$, $1 \leq j \leq n$, $1 \leq j' \leq q$.
2) On fixe $n \geq 1$. On pose Entrer la matrice identité $I_n$ en utilisant la commande `eye`. Calculer la matrice $D_N = T_n \otimes I_n + I_n \otimes T_n$ par `d=kron(t,id)+kron(id,t)`.
2) Calculer les valeurs propres de $D_N$. Vérifier qu'elles coïncident avec

$$\mu_{k,l} = \lambda_k + \lambda_l, \quad 1 \leq k, l \leq n \tag{4}$$

**4- Résolution de** $(T \otimes I + I \otimes T)x = b$
On considère le second membre $b \in \mathbb{R}^{n^2}$ tel que $b_i = 1$. On va résoudre de plusieurs façons le système linéaire

$$Dx = b \tag{5}$$

avec $D = (T \otimes I + I \otimes T)$. 1) Essayer d'abord `x=D\b`. Evaluer le temps calcul en fonction de l'entier $N$. On utilisera les commandes `tic` et `toc`.
2) Utiliser la méthode du gradient conjugué, qui s'obtient dans MATLAB avec la commande `cgs`.
3) Effectuer un préconditionnement diagonal. Evaluer à nouveau la courbe $N \mapsto T(N)$.