**Parcours : LAGE-TC**

**Semestre : 2**

**AU : 2016-2017**

*Réf. : GE-085*


Abdelbacet Mhamdi

Ingénieur en Génie Électrique

Technologue en GE à l'ISET de Bizerte


# Calcul Scientifique : Matlab/Octave

ATELIERS CORRIGÉS


**Institut Supérieur des Études Technologiques de Bizerte**

Disponible à l'adresse : https ://github.com/a-mhamdi/isetbz/

# Code d'honneur

The University of North Carolina at Chapel Hill
Department of Physics and Astronomy
http://physics.unc.edu/undergraduate-program/labs/general-info/

"During this course, you will be working with one or more partners with whom you may discuss any points concerning laboratory work. However, you must write your lab report, in your own words.

Lab reports that contain identical language are not acceptable, so do not copy your lab partner's writing.

If there is a problem with your data, include an explanation in your report. Recognition of a mistake and a well-reasoned explanation is more important than having high-quality data, and will be rewarded accordingly by your instructor. A lab report containing data that is inconsistent with the original data sheet will be considered a violation of the Honor Code.

Falsification of data or plagiarism of a report will result in prosecution of the offender(s) under the University Honor Code.

On your first lab report you must write out the entire honor pledge :

---

**The work presented in this report is my own, and the data was obtained by my lab partner and me during the lab period.**

---

On future reports, you may simply write "*Laboratory Honor Pledge*" and sign your name."

# Table des matières

# Atelier # 1 | Calcul matriciel

**Manipulation nº 1 :**

Écrire les instructions MATLAB permettant d'exécuter les commandes suivantes :

**a)** Trouver la valeur de $t \in [0,\ 2\pi]$ qui maximise $\cos(t) + \sin(t)$. On considère un pas de $t$ de $0.01$ ;

**b)** Créer un vecteur ligne composé de $100$ nombres tirés aléatoirement suivant la loi uniforme sur l'intervalle $[0,\ 5]$ ;

**c)** Trouver ensuite l'indice et la valeur au sein de ce vecteur qui s'approche le plus de $\pi$.

*(Utiliser la commande* `find`*.)*

Command Window

```
>> t = 0:0.01:2*pi
>> y = cos(t)+sin(t)
>> t(y == max(y))
>>
§ >> u = 5*rand(1, 100)
>>
>> y = abs(u-pi)
>> idx = find(y == min(y))
>> disp(u(idx))
```

**Manipulation nº 2 :**

**a)** Créer la variable $t$ contenant tous les instants d'échantillonnage entre $0$ et $0.1$ sec, par pas de $1$ msec (soit $10^{-3}$ sec) ;

**b)** Afficher en fonction du temps $t$ le vecteur $\overrightarrow{\omega}_1$ contenant les valeurs de la fonction

$$x(t) = 2\cos(2\pi f t + \varphi_1)$$

pour une fréquence $f = 50$ Hz, une phase $\varphi_1 = \pi/3$ ;

**c)** Afficher la courbe en bleu avec une épaisseur de ligne de $2$ et faire apparaître les échantillons par des étoiles. Renseigner les axes et mettre un titre ;

**d)** Sur le même graphique, afficher en fonction du temps et en rouge le vecteur $\overrightarrow{\omega}_2$ contenant les valeurs de la fonction

$$y(t) = 2\cos(2\pi f_1 t + \varphi_1)\sin(2\pi f_2 t + \varphi_2)$$

pour les fréquences $f_1 = 50$ Hz, $f_2 = 300$ Hz, les phases $\varphi_1 = \pi/3$ et $\varphi_2 = \pi/4$ ;

**e)** Mettre une épaisseur de ligne de $2$ et faire apparaître les échantillons par des ronds ;

**f)** Changer le titre et mettre une légende.

Command Window

```
>> t = 0:1e-3:0.1
>> o1 = 2*cos(2*pi*50*t+pi/3)
>> plot(t, o1, 'b', 'linewidth', 2, 'marker', '*')
>> xlabel('Temps( sec)'), ylabel('x')
>> title("Evolution de x(t)")
>> o2 = 2*cos(2*pi*50*t+pi/3).*sin(2*pi*300*t+pi/4)
>> hold on
>> plot(t, o2, 'r', 'linewidth', 2, 'marker', 'o')
>> ylabel('x et y')
>> title("Evolution de x(t) et de y(t)")
>> legend('x(t)', 'y(t)')
```

**Manipulation Nº 3 :**

Le signal échelon est défini par $x(t) = 1$ si $t > 0$ et $x(t) = 0$ sinon. Écrivez les instructions MATLAB permettant de créer et d'afficher ce signal sur l'intervalle de temps $[-5, 5]$ avec un pas de $0.01$ sec.

Command Window

```
>> t = -5:0.01:5
>> x(t>0) = 1
>> plot(t, x)
>> grid
>> xlabel('Temps (sec)')
>> ylabel('x(t)')
>> title('Signal échelon')
```

**Manipulation Nº 4 :**

Écrire les commandes MATLAB correspondantes.

**a)** Créer une matrice $\mathcal{Z}$ sachant que :

- la $1^{\text{ère}}$ ligne de $\mathcal{Z}$ est $\overrightarrow{a} = [10, -20, 30, -40]$;
- la $2^{\text{ème}}$ ligne de $\mathcal{Z}$ est le double de $\overrightarrow{a}$ ;
- la $3^{\text{ème}}$ ligne de $\mathcal{Z}$ est le carré de chaque élément de $\overrightarrow{a}$.

**b)** Trouver la taille de $\mathcal{Z}$ ;

**c)** Calculer le nombre des éléments positifs, négatifs et nuls de la matrice $\mathcal{Z}$ ;

**d)** Calculer le nombre des éléments pairs et impairs de la matrice $\mathcal{Z}$ ;

**e)** Trouver le max, min de $\mathcal{Z}$.

Command Window

```
>> a = [10 -20 30 -40]
>> Z = [a; 2*a; a.^2]
>>
>> size(Z)
>>
>> length(Z(Z<0)) % Eléments négatifs
>> length(Z(Z>0)) % Eléments positifs
>> length(Z(Z == 0)) % Eléments nuls
```

```
>>
>> length(Z(mod(Z, 2) == 0)) % Eléments pairs
>> length(Z(mod(Z, 2) == 1)) % Eléments impairs
>>
>> max(max(Z))
>> min(min(Z))
```

**Manipulation Nº 5 :**

Soient les deux fonctions

$$f(x) = xe^{-x}\cos(2x) \quad \text{et} \quad g(x) = e^{-\frac{x^2}{2}}$$

pour $x \in [0,\ \pi]$ avec un pas $p = \dfrac{2\pi}{100}$.

**a)** Calculer $f(x)$ et $g(x)$ ;

**b)** Représenter leurs courbes sur la même figure.

Command Window

```
>> x = 0:2*pi/100:pi
>> f = x.*exp(-x).*cos(2*x)
>> g = exp(-x.^2/2)
>>
>> plot(x, f, x, g)
>> legend('f(t)', 'g(t)')
```

**Manipulation Nº 6 :**

Soit la matrice

$$\mathcal{A} = \begin{pmatrix} 7 & 5 & 8 & 8 & 3 \\ 7 & 1 & 8 & 4 & 7 \\ 1 & 2 & 1 & 6 & 6 \\ 7 & 4 & 8 & 1 & 8 \end{pmatrix}.$$

**a)** Extraire dans un vecteur $\overrightarrow{e}$ tous les éléments de $\mathcal{A}$ qui sont supérieurs à $6$ ;

**b)** Extraire dans un vecteur $\overrightarrow{f}$ toutes les valeurs de $\mathcal{A}$ qui sont inférieures à $6$ en leur rajoutant $1$.

Command Window

```
>> A = [7 5 8 8 3; 7 1 8 4 7; 1 2 1 6 6; 7 4 8 1 8]
>> e = A(A>6)
>> f = A(A>6) + 1
```

**Manipulation Nº 7 :**

**a)** Écrire les commandes qui demandent à l'utilisateur deux entiers $n \geq 2$ et $m \geq 2$, et génère une matrice $\mathcal{M}$ de nombres pseudo-aléatoires uniformes entre $0$ et $1$ de taille $n \times m$ ;

**b)** Construire une sous matrice avec uniquement les éléments de $\mathcal{M}$ à un indice de ligne et de colonne pair ;

**c)** Construire une matrice $\mathcal{A}$ telle que pour tout indice $i \in \{1 \cdots 3\}$ et $j \in \{1 \cdots 3\}$, le terme

$$a_{ij} = 2i - j;$$

**d)** Construire une matrice $\mathcal{B}$ telle que pour tout indice $i \in \{1 \cdots 3\}$ et $j \in \{1 \cdots 4\}$, le terme

$$b_{ij} = 2i - \frac{j}{3}.$$

Command Window

```
>> n = input('Donner un entier >= 2 :\n')
>> m = input('Donner un entier >= 2 :\n')
>> M = rand(n, m)
>>
>> idx_Row = 2:2:n-mod(n, 2) % Les indices pairs de lignes
>> idx_Col = 2:2:m-mod(m, 2) % Les indices pairs de colonnes
>>
>> M(idx_Row, idx_Col)
>>
>> I = 1:3
>> J = 1:3
>> A = 2*[I' I' I']-[J; J; J] % Octave : A = 2*I'-J
>> I = 1:3
>> J = 1:4
>>
>> B = 2*[I' I' I' I']-1/3*[J; J; J; J] % Octave : B = 2*I'-1/3*J
```

**Manipulation № 8 :**

Soit la matrice et les vecteurs colonnes suivants :

$$\mathcal{A} = \begin{pmatrix} 1/8 & 1/4 & 1/5 \\ -1/4 & 0 & 5/8 \\ 1/5 & -5/8 & 1/8 \end{pmatrix}, \quad \overrightarrow{b} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \quad \overrightarrow{u}_1 = \begin{pmatrix} 5 \\ 2 \\ 4 \end{pmatrix}. \tag{1.1}$$

On définit pour $n \geq 1$, la suite $\overrightarrow{u}_{n+1} = \mathcal{A}\overrightarrow{u}_n + \overrightarrow{b}$.

**a)** Construire une boucle qui calcule les 100 premiers termes de la suite $\left(\overrightarrow{u}_n\right)$;

**b)** Représenter graphiquement l'évolution du module $\|\overrightarrow{u}_n\|_2$ en fonction de l'indice $n$;

$$\left(\textit{On rappelle que } \|\overrightarrow{u}\|_2 = \left\| \begin{matrix} u_x \\ u_y \\ u_z \end{matrix} \right\|_2 = \sqrt{u_x^2 + u_y^2 + u_z^2}\ .\right)$$

**c)** Changer le vecteur $\overrightarrow{u}_1$ par $\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}$ et re–calculer et re–afficher l'évolution de la suite $\left(\overrightarrow{u}_n\right)$.

Command Window

```
>> A = [1/8 1/4 1/5; -1/4 0 5/8; 1/5 -5/8 1/8]
>> b = [1 -1 1]'
>> U(:, 1) = [5 2 4]'
```

```
>>
>> for k = 2:100
U(:, k) = A*U(:, k-1) + b
end
>>
>> Module = sqrt(sum(U.^2, 1))
>> plot(Module)
>>
>> U(:, 1) = [2 1 0]'
>> for k = 2:100
U(:, k) = A*U(:, k-1) + b
end
>>
>> Module = sqrt(sum(U.^2, 1))
>> hold on
>> plot(Module)
```

**Manipulation № 9 :**

**a)** Construire de trois manières différentes, le vecteur ligne $\mathcal{X}$ de taille 100 qui comporte les 100 premiers carrés des nombres entiers ;

**b)** Construire sans effectuer de boucles la matrice $\mathcal{M}$ de dimension 10×10 donnant les résultats de la table de multiplication ;

**c)** Extraire la matrice formée de la table de multiplication des 4 premiers entiers impairs (en ligne) par les 4 premiers entiers pairs (en colonnes).

**Les instructions données par la suite ont été testées sous GNU OCTAVE.**

Command Window

```
>> X = (0:99).^2
>> v = 0:9
>> M = v'.*v
>> M(2:2:8, 1:2:7)
```

**Manipulation № 10 :**

Soient les vecteurs $\mathcal{X} = \begin{pmatrix} 3 \\ 2 \\ 6 \\ 8 \end{pmatrix}$ et $\mathcal{Y} = \begin{pmatrix} 4 \\ 1 \\ 3 \\ 5 \end{pmatrix}$.

**a)** Ajouter la somme des éléments de $\mathcal{X}$ à $\mathcal{Y}$ ;

**b)** Élever chaque élément de $\mathcal{X}$ à la puissance spécifiée par l'élément correspondant de $\mathcal{Y}$ ;

**c)** Diviser chaque élément de $\mathcal{Y}$ par l'élément correspondant de $\mathcal{X}$. Le résultat trouvé devrait être assigné à la variable $\mathcal{Z}$ ;

**d)** Multiplier chaque élément de $\mathcal{Z}$ par lui même et assigner le résultat à $\Omega$ ;

**e)** Calculer $\mathcal{X}^{\mathsf{T}}\mathcal{Y} - \Omega$.

§

Command Window

```
>> X = [3 2 6 8]'
>> Y = [4 1 3 5]'
>> Y += sum(X)
>> X.^Y
>> Z = Y.^X
>> Omega = Z.^2
>> X'*Y-Omega
```

**Manipulation № 11 :**

Écrivez les instructions permettant de générer un vecteur ligne contenant tous les entiers pairs entre $456$ et $222$ rangé du plus grand au plus petit. Combien y en a-t-il de ces entiers pairs ?

§

Command Window

```
>> X = 456:-2:222
>> length(X)
```

**Manipulation № 12 :**

Soit le vecteur $\mathcal{X} = [0,\ \pi/10,\ 2\pi/10,\ \cdots,\ 2\pi]$ :

**a)** Tracer sur un même graphique les trois courbes

$$\mathcal{Y}_{11} = \sin(\mathcal{X}), \quad \mathcal{Y}_{12} = \sin(\mathcal{X} - 0.3) \quad \text{et} \quad \mathcal{Y}_{13} = \sin(\mathcal{X} - 0.5),$$

de telle sorte que la courbe 1 soit une ligne continue rouge, la courbe 2 des cercles bleus, et la courbe 3 des pointillés noirs ;

**b)** Définir

$$\mathcal{Y}_{21} = \sin(\mathcal{X}), \quad \mathcal{Y}_{22} = \cos(\mathcal{X}),$$

puis utiliser `subplot(2,1,1)` et `subplot(2,1,2)` pour tracer sur une même figure les deux graphes des fonctions sinus et cosinus, l'un en dessous de l'autre.

§

Command Window

```
>> X = 0:pi/10:2*pi
>> Y11 = sin(X)
>> Y12 = sin(X-0.3)
>> Y13 = sin(X-0.5)
>> plot(X, Y11, 'r')
>> hold on
>> plot(X, Y12, 'ob')
>> plot(X, Y13, ':k')
>> Y21 = sin(X)
>> Y22 = cos(X)
>> subplot(2, 1, 1)
>> plot(X, Y21)
>> subplot(2, 1, 2)
>> plot(X, Y22)
```

**Manipulation № 13 :**

Pour tout $t \in \left[0,\ \frac{1}{2}\right]$ et par pas de $0.01$, on considère le signal

$$x(t) = \sin\left(2\pi f_0 t + \frac{\pi}{7}\right),$$

où $f_0 = 6$Hz. Compléter les instructions MATLAB permettant de créer et d'afficher graphiquement la fonction $x$.

Command Window

```
>> T = 0:0.01:0.5
>> X = sin(12*pi*T+pi/7)
>> plot(T, X)
>> xlabel('Temps (sec)')
>> ylabel('x(t)')
>> title('Graphe de x au cours du temps')
>> grid on;
```

**Manipulation № 14 :**

On définit $n!$ comme étant le produit des entiers inférieurs à $n$ et strictement positifs. Écrire l'instruction MATLAB qui permet de calculer $\ln(123!)$. On rappelle que :

$$\begin{aligned} \ln(n!) &= \ln\left(\prod_{k=1}^{n} k\right) \\ &= \sum_{k=1}^{n} \ln(k). \end{aligned} \tag{1.2}$$

Command Window

```
>> LnFact123 = sum(log(1:123))
```

**Manipulation № 15 :**

Écrire le code MATLAB qui permet de créer les variables suivantes :

**a)** $\mathcal{Z}$ : matrice de dimensions $4 \times 5$ dont les éléments pris en colonne sont les nombres allant de $0$ à $-1.9$ par pas de $-0.1$ ;

**b)** $\mathcal{F}$ : matrice de dimensions $9 \times 5$ dont la partie haute est la matrice $\mathcal{Z}$ précédente et le reste est nul ;

**c)** $\mathcal{D}$ : matrice diagonale $6 \times 6$ dont les éléments diagonaux valent $10^{-3}$ ;

**d)** $\mathcal{U}$ : matrice de dimensions $3 \times 6$ dont les éléments sont des tirages pseudo–aléatoires uniformes sur l'intervalle $[0,\ 1]$ ;

**e)** $\Sigma_1$ : somme des colonnes de $\mathcal{F}$ ;

**f)** $\Sigma_2$ : somme de tous les éléments de $\mathcal{U}$ ;

**g)** $\overrightarrow{v}$ : vecteur ligne de longueur $1000$ dont les éléments sont des tirages pseudo–aléatoires gaussiens de moyenne nulle et d'écart–type $2$ ;

**h)** $\mu$ : moyenne empirique de $\overrightarrow{v}$ ;

**i)** $\sigma^2$ : variance empirique de $\overrightarrow{v}$.

§
Command Window
```
>> Z = reshape(0:-0.1:-1.9, 4, 5)
>> F = zeros(9, 5)
>> F(1:4, :) = Z
>> D = 1e-3*eye(6)
>> U = rand(3, 6)
>> Sigma1 = sum(F, 2)
>> Sigma2 = sum(sum(U))
>> v = 2*randn(1, 1000)
>> mu = mean(v)
>> sigma2 = var(v)
```

**Manipulation № 16 :**

**a)** Définir une matrice $\mathcal{M}$ aléatoire selon une loi uniforme à trois lignes et sept colonnes ;

**b)** Combien de nombres dans cette matrice sont plus grand que 0.5 ? que 0.8 ? Où sont–ils situés ?

**c)** Construire alors la matrice $\mathcal{P}$ obtenue à partir de la matrice $\mathcal{M}$ en remplaçant tous les nombres de $\mathcal{M}$ inférieurs à 0.4 par 0, et ceux supérieurs à 0.4 par 1 ;

**d)** Construire de même la matrice $\mathcal{Q}$ obtenue à partir de la matrice $\mathcal{M}$ en remplaçant tous les nombres de $\mathcal{M}$ inférieurs à 0.5 par -3 et tous les nombres supérieurs à 0.5 par 14.

§
Command Window
```
>> M = rand(3, 7);
>> length(M(M>=0.5));
>> [r5_idx, c5_idx] = find(M>=0.5);
>> length(M(M>=0.8));
>> [r8_idx, c8_idx] = find(M>=0.8);
>> P = M; P(P<=0.4) = 0; P(P>=0.4) = 1;
>> Q = M; Q(Q<=0.5) = -3; Q(Q>=0.5) = 14;
```

**Manipulation № 17 :**

**a)** Construire le vecteur colonne $x = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$ ;

**b)** Calculer $\mathcal{P}$ définie par $p_{ij} = x_i x_j$ ;

**c)** Calculer $m = \sum_{k=1}^{4} x_i^2$ ;

**d)** Créer le vecteur ligne $\overrightarrow{a}$ en prenant que la $1^{\text{ère}}$ ligne de $\mathcal{P}$ ;

**e)** Créer le vecteur ligne $\overrightarrow{b}$ en prenant que la $2^{\text{ème}}$ colonne de $\mathcal{P}$ ;

**f)** Créer la matrice $\mathcal{C}$ en accolant verticalement les deux vecteurs lignes $\overrightarrow{a}$ et $\overrightarrow{b}$.

§
Command Window
```
>> X = [1; 2; 3; 4]
```

```
>> P = X*X'
>> m = sum(X.^2)
>> a = P(1, :)
>> b = P(:, 2)'
>> C = [a; b]
```

**Manipulation №18 :**

**a)** Définir la variable $\mathcal{X} = \frac{\pi}{4}$, et calculer $\mathcal{Y}_1 = \sin(\mathcal{X})$ et $\mathcal{Y}_2 = \cos(\mathcal{X})$, puis $\mathcal{Z} = \tan(\mathcal{X})$ à partir de $\mathcal{Y}_1$ et $\mathcal{Y}_2$ ;

**b)** Définir la variable $\mathcal{X} = \left[\frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}\right]$, et calculer $\mathcal{Y}_1 = \sin(\mathcal{X})$ et $\mathcal{Y}_2 = \cos(\mathcal{X})$ ;

**c)** Calculer alors $\mathcal{Z} = \tan(\mathcal{X})$ en utilisant exclusivement les vecteurs $\mathcal{Y}_1$ et $\mathcal{Y}_2$ précédents ;

**d)** Définir la variable $\mathcal{X} = [0 : 0.1 : 2\pi]$. Combien y a-t-il de valeurs dans ce vecteur ? Afficher la courbe du sinus.

Command Window

```
>> X = pi/4; Y1 = sin(x); Y2 = cos(X); Z = Y1/Y2;
>> X = [pi/6 pi/4 pi/3]; Y1 = sin(X); Y2 = cos(X);
>> tanX = Y1./Y2;
>> X = 0:.1:2*pi;
>> length(X);
>> plot(X, sin(X)); grid on; xlabel('X'); ylabel('sin(X)');
>> title('La courbe du sinus');
```

# Atelier # 2 | Système d'équations linéaires

**Manipulation N° 1 :**

Déterminer toutes les matrices $\mathcal{X} \in \mathcal{M}_{(2,2)}(\mathbb{R})$ telles que $\mathcal{A} = \mathcal{B}\mathcal{X}$. On donne

$$\mathcal{A} = \begin{pmatrix} 4 & 3 \\ 6 & 5 \\ 4 & 3 \end{pmatrix} \quad \text{et} \quad \mathcal{B} = \begin{pmatrix} 2 & 3 \\ 3 & 4 \\ 2 & 3 \end{pmatrix}.$$

On ignore la dernière ligne de $\mathcal{A}$ et de $\mathcal{B}$.

Command Window

```
>> A =  [4 3; 6 5]
>> B =  [2 3; 3 4]
>>
>> det(B)  % Tester l'inversibilité de B !
>> X = B\A
```

**Manipulation N° 2 :**

Trouver le point d'intersection de deux lignes droites $\Delta_1$ et $\Delta_2$ définies dans le système de coordonnées cartésiennes par :

$$\begin{cases} \Delta_1 : & 4x + 3y = 24 \\ \Delta_2 : & 3x + 4y = 25. \end{cases}$$

Command Window

```
>> A = [4 3; 3 4]
>> b = [24; 25]
>> det(A)
>> A\b
>> fprintf("Les deux droites s'intersectent\
au point (%.1f, %.1f).\n", ans)
```

**Manipulation N° 3 :**

$24$ crayons et $12$ gommes à effacer coûtent $9.6$ dinars. $20$ crayons et $15$ gommes à effacer coûtent $10$ dinars. Quels sont les prix du crayon et de la gomme ? Décrire le problème d'abord par un système d'équa-

tions.

On dénote par $c$ et $g$ le nombre de crayons et de gommes à effacer respectivement. Le problème se ramène à l'écriture suivante :

$$\begin{cases} 24c + 12g &= 9.6 \\ 20c + 15g &= 10 \end{cases}$$

Command Window

```
>> A = [24 12; 20 15]
>> b = [9.6; 10]
>> det(A)
>> A\b
```

**Manipulation № 4 :**

Vous contre un athlète ! Vous pouvez courir $0.2$ km chaque minute. L'athlète peut courir $0.5$ km chaque minute. Vous débutez la course $6$ minutes plus tôt. Jusqu'où pouvez-vous aller avant que l'athlète ne vous attrape ?

La distance en km, parcourue par un athlète est $0.5y$. La vôtre exprimée en km, est $(1.2 + 0.2x)$. Le système d'équations est donné donc par :

$$(S) \quad \begin{cases} x - y &= 0 \\ 0.2x - 0.5y &= -1.2. \end{cases}$$

Command Window

```
>> A = [1 -1; 0.2 -0.5]
>> b = [0; -1.2]
>> det(A)
>> T = A\b
>> (T + [6; 0]).*[.2; .5] % Distances parcourues
```

**Manipulation № 5 :**

Soit la matrice $\mathcal{A} \in \mathbb{R}^{n\times n}$ ayant la forme suivante :

$$\mathcal{A} = \begin{pmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & 0 & a_{n-1} & b_n \end{pmatrix}$$

On note

- $b = (b_i)_{1\leq i\leq n}$ les coefficients diagonaux ;
- $a = (a_i)_{1\leq i\leq n-1}$ les coefficients sous–diagonaux ;

- $c = (c_i)_{1 \le i \le n-1}$ les coefficients sur-diagonaux de la matrice $\mathcal{A}$.

Afin de résoudre un système $\mathcal{A}\overrightarrow{x} = \overrightarrow{v}$, on peut appliquer une factorisation sous la forme $\mathcal{LU}$, avec :

$$\mathcal{L} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ l_1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & l_{n-1} & 1 \end{pmatrix} \quad \text{et} \quad \mathcal{U} = \begin{pmatrix} d_1 & u_1 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & u_{n-1} \\ 0 & \cdots & \cdots & 0 & d_n \end{pmatrix}$$

On donne l'algorithme de cette factorisation.

**Entrée :**

$\overrightarrow{a} \in \mathbb{R}^{n-1}$, $\overrightarrow{b} \in \mathbb{R}^n$ et $\overrightarrow{c} \in \mathbb{R}^{n-1}$

**Sortie :**

$\overrightarrow{l} \in \mathbb{R}^{n-1}$, $\overrightarrow{d} \in \mathbb{R}^n$ et $\overrightarrow{u} \in \mathbb{R}^{n-1}$.

**Initialisation :**

$n \leftarrow$ length$(\overrightarrow{b})$
$d_1 \leftarrow b_1$
Pour $i \leftarrow 1$ à $n-1$ faire
    $l_i \leftarrow \frac{a_i}{d_i}$
    $u_i \leftarrow c_i$
    $d_{i+1} \leftarrow b_{i+1} - l_i u_i$
Fin

**a)** Construire la matrice $\mathcal{A}$, donnée par :

$$\mathcal{A} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}.$$

**b)** Extraire les vecteurs $\overrightarrow{a}$, $\overrightarrow{b}$ et $\overrightarrow{c}$ à partir de la matrice $\mathcal{A}$ ;

**c)** Compléter le code qui permet d'obtenir les vecteurs $\overrightarrow{l}$, $\overrightarrow{d}$ et $\overrightarrow{u}$ à partir de vecteurs $\overrightarrow{a}$, $\overrightarrow{b}$ et $\overrightarrow{c}$ ;

**d)** Former par la suite les deux matrices $\mathcal{L}$ et $\mathcal{U}$ ;

**e)** Écrire un code qui permet de résoudre le système triangulaire inférieur $\mathcal{L}\overrightarrow{y} = \overrightarrow{v}$, où

$$\overrightarrow{v} = (0,\ 1,\ 2,\ -1,\ 5)^{\mathsf{T}} ;$$

**f)** Écrire un code qui permet de résoudre le système triangulaire supérieur $\mathcal{U}\overrightarrow{x} = \overrightarrow{y}$.

Command Window

```
>> A = [2 -1 0 0 0; -1 2 -1 0 0; 0 -1 2 -1 0; 0 0 -1 2 -1; 0 0 0 -1 2]
>>
>> n = length(A)
>> for i  = 1:n
b(i) = A(i, i)
end
>> for i = 1:n-1
a(i) = A(i+1, i)
c(i) = A(i, i+1)
end
>>
>> d(1) = b(1)
>> for i = 1:n-1
l(i) = a(i)/d(i)
u(i) = c(i)
d(i+1) = b(i+1)-l(i)*u(i)
end
>> L = zeros(n, n), U = zeros(n, n)
>> for i = 1:n
L(i, i) = 1
U(i, i) = d(i)
end
>> for i = 1:n-1
L(i+1, i) = l(i)
U(i, i+1) = u(i)
end
>>
>> v = [0 1 2 -1 5]'
>> y = zeros(n, 1)
>> for i = 1:n
y(i) = 1/L(i, i)*(v(i)-L(i, :)*y)
end
>>
>>  x = zeros(n, 1)
>> for i = n:-1:1
x(i) = 1/U(i, i)*(y(i)-U(i, :)*x)
end
```

**Manipulation № 6 :**

Mettre le problème suivant sous forme de système d'équations. On se propose de déterminer le nombre de crayons et de bocaux. La devinette est comme suit :

**a)** Si on met $9$ crayons dans chaque bocal, il reste $2$ bocaux ;

**b)** Si on met $6$ crayons dans chaque bocal, il reste $3$ crayons.

Soient $x$ et $y$ les nombres de crayons et et de bocaux respectivement. Le système d'équations résultant est comme suit :

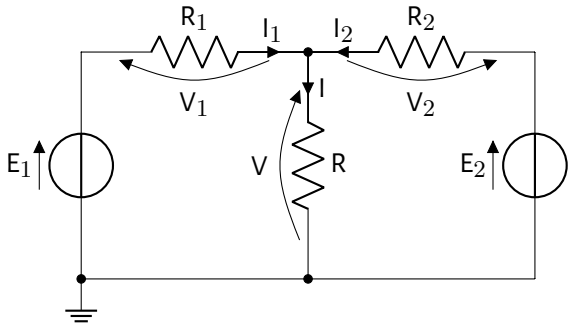
$$\begin{cases} 9(y-2) &= x \\ 6y &= x-3 \end{cases}$$

Command Window

```
>> A = [1 -9; 1 -6]
>> b = [-18; 3]
>> det(A)
>> Res = A\b
>> fprintf('Le nombre de crayons est %i.\n', Res(1))
>> fprintf('Le nombre de bocaux est %i.\n', Res(2))
```

**Manipulation № 7 :**

Soit le circuit électrique ci-dessous. On se donne $E_1 = E_2 = 10$ volts, $R_1 = R_2 = 2.2k\Omega$ et $R = 1k\Omega$. Le système d'équations résultant après application des lois de Kirchoff est donné par Eq. 2.1. Calculer le courant I qui parcourt la résistance R.




$$\begin{cases} E_1 - R_1 I_1 - RI &= 0 \\ E_2 - R_2 I_2 - RI &= 0 \\ I &= I_1 + I_2. \end{cases} \tag{2.1}$$

Command Window

```
>> A = [2.2 0 1; 0 2.2 1; -1 -1 1];
>> b = [10; 10; 0];
>> det(A)
>> X = A\b
>> I = X(end)*1e-3
```

**Manipulation № 8 :**

Résoudre le système linéaire, d'inconnues $x$, $y$ et $z$ suivant

$$\begin{cases} 6x + y - 5z = 10 \\ 2x + 2y + 3z = 11 \\ 4x - 9y + 7z = 12. \end{cases}$$

§

Command Window

```
>> A = [6 1 -5; 2 2 3; 4 9 7]
>> b = [10; 11; 12]
>> det(A)
>> A\b
```

# Atelier # 3 | Intégration numérique

**Manipulation N° 1 :**

Résoudre numériquement pour $t$ dans $[-\pi/2.25,\ \pi/2.25]$ l'équation différentielle suivante :

$$y^{(1)} + \tan(t)y \ = \ \sin(2t), \quad y(t = -\pi/2.25) = 1.$$

M-File

```
function   [    dot_y    ] = Func1( t, y)
     dot_y = -tan(t)*y+sin(2*t)
end
```

Command Window

```
>> [t, y] = ode45(@Func1, [-pi/2.25 pi/2.25], 1);
```

**Manipulation N° 2 :**

Résoudre numériquement pour tout $t$ dans $[0,\ 5]$ , l'équation différentielle suivante :

$$\dot{X} \ = \ \begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix} X, \quad \text{avec} \quad X_0 \ = \ \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

M-File

```
function [dot_X] = Func(t, X)
    dot_X = [3 4; -4 3]*X
end
```

Command Window

```
>> [t, X] = ode45(@Func, [0 5], [0; 1])
```

**Manipulation N° 3 :**

On définit une équation différentielle du premier ordre par :

$$\frac{dy}{dt} \ = \ \mathcal{F}(t, y), \quad y\,(t = t_0) = y_0.$$

Pour résoudre numériquement cette équation, on peut avoir recours à l'approximation suivante :

$$y\,(t + \Delta) \ = \ y(t) + \Delta \frac{dy}{dt},$$

tout en fixant un pas fini de temps $\Delta$, un instant $t_i$ est donné par $t_i = t_0 + i\Delta$ et on dénote par $y_i = y(t = t_i)$. On obtient :

$$y_{i+1} = y_i + \Delta\mathcal{F}(t_i, y_i).$$

On se propose de résoudre l'équation différentielle suivante :

$$\frac{dy}{dt} = y, \quad \text{avec} \quad y_0 = 2,$$

sur l'intervalle de temps $[0,\ 5]$ avec un pas $\Delta$ de $0.1$sec.

**a)** Écrire le code qui permet d'implémenter cette fonction ;

**b)** Compléter les commandes qui permettent de résoudre l'équation précédente en se référant à l'approximation donnée en début de l'exercice ;

**c)** Utiliser la commande ode45 pour résoudre la même équation et représenter les deux solutions sur la même figure ;

**d)** Refaire le même travail pour

$$\frac{dy}{dt} + y^2 = 0, \quad \text{avec} \quad y_0 = 1.$$

§

M-File

```
function [  doty    ] = func1(t, y)
        doty = y;
end
```

§

M-File

```
>> delta = 0.1; y0 = 1; t1 = (0:delta:5)';
>> n = length(t1); y1 = [y0; zeros(n-1, 1)];
>> for i = 1:n-1
y1(i+1) = y1(i) + delta*func1(t1(i), y1(i));
end
```

§

Command Window

```
>> [t2, y2] = ode45(@func1, t1, 2);
>> plot(t1, y1, t2, y2);
>> legend('Approximation', 'ODE45');
```

§

M-File

```
function [  doty    ] = func2(t, y)
        doty = -y^2;
end
```

§

Command Window

```
>> delta = 0.1; t1 = (0:delta:5)';
>> n = length(t1);
>> y1 = [1; zeros(n-1, 1)];
>>for i = 1:n-1
y1(i+1) = y1(i) + delta*func2(t1(i), y1(i));
end
>> [t2, y2] = ode45(@func2, t1, 1);
```

```
>> plot(t1, y1, t2, y2);
>> legend('Approx', 'ODE45');
```

# Atelier # 4 | Interpolation polynomiale

**Manipulation №1 :**

**a)** Calculer le polynôme d'interpolation de Lagrange $\mathcal{P}_n$ de la fonction sinus définie sur l'intervalle $[0,\ 2\pi]$ par pas de $0.1$ associé à $n+1$ points équidistants. On prend $x_0 = 0$ et $x_n = 2\pi$ pour $n = 3,\ 4,\ 5$ & $6$;

*(Utiliser la commande* `linspace`*.)*

**b)** Le tracer en superposant les résultats pour les différentes valeurs de $n$ et tracer sur le même graphique la courbe du sinus;

**c)** Faire de même pour la fonction $1/(1+x^2)$ sur l'intervalle $[-5,\ 5]$ par pas de $0.1$, avec $x_0 = -5$ et $x_n = 5$ pour $n = 3,\ 4,\ 5$ & $6$.

§

Command Window

```
>> X = 0:0.1:2*pi; Y = sin(X);
>>
>> X3 = linspace(0, 2*pi, 4); P3 = polyfit(X3, sin(X3), 3);
>> X4 = linspace(0, 2*pi, 5); P4 = polyfit(X4, sin(X4), 4);
>> X5 = linspace(0, 2*pi, 6); P5 = polyfit(X5, sin(X5), 5);
>> X6 = linspace(0, 2*pi, 7); P6 = polyfit(X6, sin(X6), 6);
>>
>> Y3 = polyval(P3, X); Y4 = polyval(P4, X);
>> Y5 = polyval(P5, X); Y6 = polyval(P6, X);
```

§

Command Window

```
>> plot(X, Y, X, Y3, X, Y4, X, Y5, X, Y6)
>> legend('Sin', 'P_3', 'P_4', 'P_5', 'P_6')
```

§

Command Window

```
>> X = -5:0.1:5; Y = 1./(1+X.^2);
>>
>> X3 = linspace(-5, 5, 4); P3 = polyfit(X3, 1./(1+X3.^2), 3);
>> X4 = linspace(-5, 5, 5); P4 = polyfit(X4, 1./(1+X4.^2), 4);
>> X5 = linspace(-5, 5, 6); P5 = polyfit(X5, 1./(1+X5.^2), 5);
>> X6 = linspace(-5, 5, 7); P6 = polyfit(X6, 1./(1+X6.^2), 6);
>>
>> Y3 = polyval(P3, X); Y4 = polyval(P4, X);
>> Y5 = polyval(P5, X); Y6 = polyval(P6, X);
>>
>> plot(X, Y, X, Y3, X, Y4, X, Y5, X, Y6)
>> legend('1/(1+x^2)', 'P_3', 'P_4', 'P_5', 'P_6')
```

Nous proposons dans ce document quelques ateliers corrigés sur les méthodes et outils du calcul scientifique et d'analyse numérique. Ils portent essentiellement sur :

① l'identification de l'intérêt du calcul scientifique réalisé avec un outil logiciel performant et multi-plateforme : `Matlab` ou `Octave` ;

② le potentiel de l'application du calcul matriciel ;

③ la modélisation & la résolution d'équations linéaires ;

④ l'intégration numérique des équations différentielles ordinaires ;

⑤ la formulation & la résolution des problèmes d'interpolation polynomiale.

`Matlab` ; `Octave` ; algèbre linéaire ; intégration numérique ; interpolation polynomiale.