

Mention Physique - L2 - Année 2010–2011
Licence de Sciences et Technologies

LP 207: Mathématiques pour physiciens 2

TP N°1 : Algèbre linéaire 1

A Calculs de matrices et de déterminants.

A1 Premiers calculs

1) Dans *Mathematica*, définir les matrices suivantes (voir le petit mémo à la fin de cette feuille) :

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}. \quad (1)$$

Calculer, avec *Mathematica*, les expressions

$$a) \quad AB \quad (2)$$

$$b) \quad BA \quad (3)$$

$$c) \quad AB - BA \quad (4)$$

Commenter le résultat.

2) Pour chacune des matrices M ci-dessous, calculer avec *Mathematica* le déterminant $\det M$ et la matrice inverse (M^{-1}) , si elle existe. Vérifier, avec le résultat pour M^{-1} fourni par *Mathematica*, l'équation $MM^{-1} = I$, en faisant la multiplication des matrices avec *Mathematica*.

$$M_1 = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}, \quad M_3 = \begin{pmatrix} -1 & 2 & -1 \\ 0 & -1 & 2 \\ 0 & 0 & -1 \end{pmatrix}.$$

3) Faire calculer par *Mathematica* les déterminants suivants, en s'efforçant de présenter le résultat sous la forme la plus simple et factorisée possible

$$\begin{vmatrix} 1 & x & x^2 & x^3 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1^2 & 2^2 & 3^2 & 4^2 \end{vmatrix}, \quad \begin{vmatrix} a^2 & ab & ab & b^2 \\ ab & a^2 & b^2 & ab \\ ab & b^2 & a^2 & ab \\ b^2 & ab & ab & a^2 \end{vmatrix}, \quad \begin{vmatrix} a & a & a & a \\ a & b & b & b \\ a & b & c & c \\ a & b & c & d \end{vmatrix}.$$

A2 Procédures : Puissances et Commutateurs

1) Le commutateur de deux matrices A, B est défini par

$$[A, B] = AB - BA.$$

a) Écrire une procédure qui retourne le *commutateur* de deux matrices A et B . Vérifier que votre procédure retourne le bon résultat pour les matrices A et B du A1-1).

b) Soit R_3 la matrice de rotation dans $\mathbb{R}^3$ d'angle $\pi/4$ autour de l'axe Oz . Soit P la matrice qui effectue une permutation cyclique des trois axes Ox, Oy, Oz : $P_{21} = 1, P_{32} = 1, P_{13} = 1$, les autres éléments nuls. La matrice de rotation d'angle $\pi/4$ autour de l'axe Oy est $R_2 = P^{-1}.R_3.P$. Calculer avec *Mathematica* le commutateur $[R_2, R_3]$. Conclusion?

2) Écrire une procédure qui retourne l'*anti-commutateur* de deux matrices A et B , que l'on définit par

$$\{A, B\} = AB + BA.$$

3) On rappelle l'expression des matrices de Pauli:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

À l'aide des procédures définies ci-dessus, vérifier que pour tout $i \neq j$, on a $\{\sigma_i, \sigma_j\} = 0$, et que $\sigma_i^2 = I$.

A3 Exponentielles de matrices

L'exponentielle d'une matrice A est définie comme pour les nombres complexes, en posant

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

1) Avec la procédure *Mathematica* `MatrixExp[A]`, vérifier pour tout $\alpha \in \mathbb{R}$ que

$$e^{i\alpha\sigma_j} = \cos(\alpha)I + i\sin(\alpha)\sigma_j$$

pour $j = 1, 2, 3$, où σ_j sont les matrices de Pauli du A2.

2) Écrire une procédure `POWER(A,p)` qui retourne la puissance p -ième d'une matrice A .

3) Écrire une procédure `EXP(A,K)` avec comme variables d'entrée $A \in M_n(\mathbb{C})$ et $K \in \mathbb{N}$ qui retourne la somme

$$\exp_K(A) = \sum_{k=0}^K \frac{A^k}{k!}.$$

4) À l'aide de la procédure `EXP` ci-dessus, calculer la valeur numérique de $\exp_K(i\frac{\pi}{3}\sigma_2)$ pour différentes valeurs de K , et comparer le résultat à la valeur théorique de $e^{i\frac{\pi}{3}\sigma_2}$.

Pourriez-vous proposer une méthode pour quantifier cette comparaison?

A4 Déterminants

On se propose dans cet exercice de résoudre par *Mathematica* l'exercice C-b) du TD 2.

1) Définir avec une procédure *Mathematica* la matrice suivante, de taille $n \times n$, qui n'a que trois diagonales non nulles :

$$M_n = \begin{pmatrix} 1 & -a & 0 & 0 & \dots & 0 \\ -a & 1 & -a & 0 & \dots & 0 \\ 0 & -a & 1 & -a & \dots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -a & 1 & -a \\ 0 & & & \dots & -a & 1 \end{pmatrix}$$

2) Calculer $D_n = \det M_n$ pour $n = 3, 4, 5$. à l'aide de *Mathematica*.

3) En développant D_n suivant la première colonne, on montre que D_n satisfait la relation de récurrence suivante:

$$D_n = D_{n-1} - a^2 D_{n-2}.$$

On peut réexprimer cette relation sous forme matricielle, en écrivant :

$$\begin{pmatrix} D_n \\ D_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & -a^2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} D_{n-1} \\ D_{n-2} \end{pmatrix}. \quad (5)$$

a) Que valent D_1 et D_2 ?

b) Écrire une procédure *Mathematica* `Dnapx[n,a]` qui calcule D_n par la relation de récurrence (5) ci dessus (on emploiera `Expand` ou `Simplify` dans le résultat du calcul matriciel), et calculer avec `Dnapx[n,a]` les valeurs de D_n pour $n = 3, 4, 5$.

4) Pour connaître le temps de calcul utilisé par *Mathematica* pour effectuer une instruction ou fonction, on fait :

instruction // Timing

On prend $n = 50$. Trouver le temps que met *Mathematica* pour calculer D_n directement, comme à la question 2. Trouver ensuite le temps mis par *Mathematica* pour calculer D_n avec votre procédure `Dnapx`, et comparer.

B Réseau électrique

1) On considère le quadripôle de la figure 1 (a). On calcule en utilisant la loi d'Ohm, et on admettra ici, que les intensités et les tensions i_1, i_0, V_1, V_0 sont reliées par

$$\begin{pmatrix} V_0 \\ V_1 \end{pmatrix} = \begin{pmatrix} R_1 + R_3 + R & -R \\ R & -(R_2 + R_4 + R) \end{pmatrix} \begin{pmatrix} i_0 \\ i_1 \end{pmatrix}$$

Résoudre (par `Solve[...]`) ces relations dans les variables V_1, i_1 en fonction de V_0 et i_0 et en déduire la *matrice de transfert* T :

$$\begin{pmatrix} V_1 \\ i_1 \end{pmatrix} = T \begin{pmatrix} V_0 \\ i_0 \end{pmatrix}$$

(**Attention !** : les grandeurs i_0 et V_0 ne peuvent pas être choisies indépendamment : les conditions aux limites aux deux extrémités du circuit vont fournir des conditions supplémentaires, comme on va voir.)

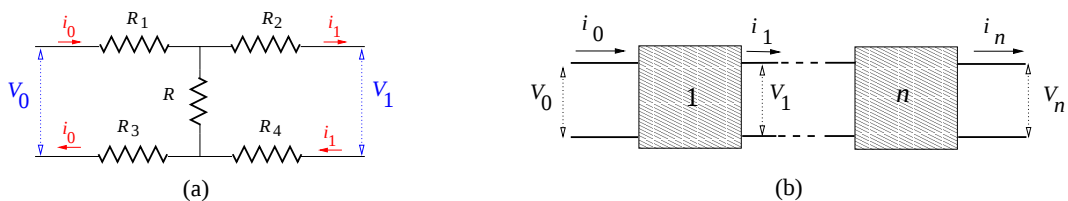


Figure 1: (a) : un quadripôle ; (b) n quadripôles en cascade.

2) On construit alors un réseau fait d'une suite de n circuits du type précédent, tous avec les mêmes résistances $R_1, \dots, R_4$ et R , connectés en "cascade" (Figure 1 (b)). Quelle est la matrice T_n qui relie V_n, i_n à V_0, i_0 ?

3) Application numérique : $R_1 = R_2 = 10 \Omega$, $R_3 = R_4 = 20 \Omega$, $R = 100 \Omega$. Calculer $T = T_1, T_2, T_3, T_4, T_{10}$. La matrice T_n vous semble-t-elle avoir une limite quand $n \rightarrow \infty$? (On étudiera prochainement une méthode permettant de calculer précisément cette limite.)

4) On suppose qu'un ampèremètre de résistance négligeable est branché à l'extrémité droite du réseau, tandis qu'une tension $V_0 = 10V$ est imposée à l'extrémité gauche, par exemple grâce à un générateur. Quelle est la valeur de V_n ? celle de i_n mesurée par l'ampèremètre ? celle de i_0 pour $n = 4$, puis pour $n = 10$? Qu'en concluez-vous ?

5) Même question si on remplace l'ampèremètre par un voltmètre de résistance supposée infinie : valeurs de V_n , de i_n et de i_0 , pour $n = 4$, puis $n = 10$?

C Matrice de Gram (*question bonus*)

On considère les polynômes $P_n(u)$ d'une variable u , définis par $P_0(u) = 1$, $P_1(u) = u$, puis par la relation de récurrence

$$(2n+1)uP_n(u) = (n+1)P_{n+1}(u) + nP_{n-1}(u) \quad (6)$$

Ce sont les *polynômes de Legendre*.

1. Écrire une procédure pour calculer ces polynômes.
2. On définit la matrice G dont l'élément (i, j) est l'intégrale

$$G_{ij} = \int_{-1}^1 P_i(u)P_j(u)du$$

La calculer par *Mathematica* pour $1 \leq i, j \leq N = 10$. Qu'observe-t-on ? Vérifier cette conjecture en poussant le calcul jusqu'à $N = 20$.

N.B. Les intégrales G_{ij} peuvent être considérées comme les produits scalaires des vecteurs P_i , dans l'espace vectoriel des polynômes en u . La matrice G est appelée *matrice de Gram* de ces vecteurs.

D Commandes utiles de *Mathematica*

Mathematica comporte plusieurs milliers de commandes avec souvent un grand nombre d'options pour chacune. Nous présentons ici quelques unes des commandes qui vous serviront le plus.

D1 Interactivité

Toutes les commandes doivent être suivies de [Majuscules] [Entrée]

Ctrl-. pour arrêter un calcul
ou commande Abort dans le menu Evaluation

D2 Généralités

<i>commande</i>	commande avec impression du résultat
<i>commande</i> ;	commande sans impression du résultat
<i>x = expression</i>	assignation immédiate
<i>x := expression</i>	assignation différée
Simplify[<i>expression</i>]	simplifier une expression
<i>expression</i> // Simplify	même chose
FullSimplify[<i>expression</i>]	plus puissant
Factor[<i>expression</i>]	factoriser une expression
(* <i>phrase(s)</i> *)	commentaires
Clear[<i>var1</i>]	effacer variable
%	résultat précédent (historiquement)
Ctrl-L	entrée précédente (géographiquement)

D3 Indices et puissances

x_i ou Subscript[*x*,*i*] ou *x* Ctrl-_ *i* *x_i*
x^j ou *x* Ctrl^*j* *x^j*

D4 Listes

<code>x = Range[1,11,2]</code>	liste d'expressions séparées par une valeur constante
<code>tab = Table[{i,Sqrt[i]},{i,1,10}]</code>	liste 2D
<code>tab // TableForm</code>	présentation de la liste 2D sous forme de tableau
<code>Length[x]</code>	dimension principale d'une liste
<code>Dimensions[tab]</code>	dimensions d'un tableau
<code>Total[x]</code>	somme des éléments d'une liste

D5 Boucles et conditions

<code>Do[... , {i,1,Length[liste]}]</code>	boucle do
<code>x = If[y>0,1,0]</code>	condition simple : $x = 1$ si $y > 0$, sinon $x = 0$
<code>x = If[y>0 && y != 2,1,0]</code>	condition double $y > 0$ et $y \neq 2$
<code>x = If[y>0 y <= 2,1,0]</code>	condition double $y > 0$ ou $y \leq 2$

D6 Équations

<code>Solve[{eqn1==0, eqn2==0},{x,y}]</code>	résout les équations eqn1 et eqn2 dans les variables x et y
<code>NSolve[{eqn1==0, eqn2==0},{x,y}]</code>	résout numériquement ces équations

D7 Fonctions et procédures

Une fonction est un petit programme qui effectue des instructions en fonction d'un ou plusieurs paramètres d'entrée. La structure d'une procédure est la suivante :

<code>f[x_,y_] := Sin[x+3y]</code>	fonction simple
<code>f[x_,y_:0] := Sin[x+3y]</code>	fonction au 2nd argument optionnel (0 par défaut)
<code>f[n_Integer] := Prime[n]</code>	fonction à argument entier
<code>f[x_,y_] := Module[{u=3,v=4}, u x + v y]</code>	fonction avec variables locales u et v

Les arguments des fonctions/procédures sont des variables *locales*.

D8 Algèbre linéaire

$V=\{u,v\}$	vecteur (liste)
$V // \text{MatrixForm}$	présente V sous forme vectorielle
$V+W$	somme de vecteurs
VW	vecteur dont les éléments sont les produits des éléments de V et W
$V.W$	produit scalaire de deux vecteurs
$A=\{\{a,b\},\{c,d\}\}$	définit la matrice $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ [ligne par ligne]
$A // \text{MatrixForm}$	présente A sous forme matricielle
$\text{Array}[a,\{n,p\}]$	matrice $n \times p$ faite de a
$\text{IdentityMatrix}[n]$	matrice identité $n \times n$
$A[[1]]$	première ligne de A
$A[[A11,1]]$	première colonne de A
$A[[1,2]]$	$A_{1,2}$
$A[[1;;2,2;;3]]$	sous-matrice des 2 premières lignes et 2 dernières colonnes
$A.V$	produit de matrice par vecteur
$A.B$	produit de deux matrices
$V.M.V$	donne un scalaire
$\text{Det}[A]$	déterminant de A
$\text{Inverse}[A]$	matrice inverse de A
$\text{Tr}[A]$	trace de A
$\text{Transpose}[A]$	transposée de A
$\text{Eigenvalues}[A]$	valeurs propres de A
$\text{Eigenvectors}[A]$	vecteurs propres de A

D9 Raccourcis

Echap a Echap	α (etc.)
Echap th Echap	θ (etc.)
Echap ii Echap	$i = \sqrt{-1}$
Echap dd Echap x	dx
Echap d Echap x	∂x
Echap int Echap	$\int$
Echap inf Echap	∞

D10 Aide

?Fonction	description brève d'une fonction
??Fonction	description plus détaillée d'une fonction
F1 sur chaîne de caractères sélectionnée	description très détaillée

D11 Entrées et sorties

Print[<i>expression</i>]	imprimer <i>expression</i> dans le <i>Notebook</i>
<i>expression</i> = Import[" <i>préfixe.extension</i> "]	importer données ou image
Export[" <i>préfixe.extension</i> ", <i>expression</i>]	exporter données ou image

D12 Erreurs communes

- Définir une expression déjà définie mais avec une dimension différente
- Oublier d'employer des variables locales dans ses fonctions/procédures
- Oublier espace entre deux variables