

**Rapport Matlab**

**TP1**

http://www.nersc.gov/assets/Vis/matlablogo.jpg

**ALGEBRE LINEAIRE**

Exercice 4 : Matrice dérivée seconde

1. Construction de la matrice dérivée seconde d’ordre :

On sait qu’une matrice dérivée seconde d’ordre 3 se présente sous la forme :

Ainsi, pour construire un équivalent à l’ordre, on place des en sa diagonale, encadrée par des 1 dans une matrice initialement composée de zéros :

A = diag((-2)\*ones(1,n)); % Le diagonale principale

A = A + diag(ones(1,n-1),1); % Le diagonale au-dessus de -2

A = A +diag(ones(1,n-1),-1); % Le diagonale en dessous de -2

1. Vecteurs propres de :

Grâce à la commande, on calcule les valeurs propres de la matrice dérivée seconde.

Ensuite grâce à une boucle on affiche les 5 derniers vecteurs propres :

[P,D] = eig(A); % Calcul des valeurs propres

l =length(P); % l corresponds au nombre de valeurs propres

for i =1:5

subplot(2,3,i);

plot(P(:,l-5+i));

str =fprintf('%d ',l-5+i);

title(str);

end



Exercice 6 : Isométries du plan

1. On a : et avec : et

Et :

On voit clairement que Tr est l’image de tr par une rotation d’angle et une translation (a,b) .

1. Tracé de l’image d’un triangle par la transformation ci-dessus :



tr = [0, 1, 2 ; 0, 1, 0];

fill(tr(1,:),tr(2,:),'b'); % affichage du triangle initial

n = length(tr);

a = rand();

b = rand();

alpha = rand();

rotation = [cos(alpha),-sin(alpha) ; sin(alpha),cos(alpha)];

translation = [a,b]' \* ones(1,n);

Tr = rotation \*tr + translation; % Transformation du triangle

hold on;

fill(Tr(1,:),Tr(2,:),'g'); % affichage du triangle transformé

legend('Triangle initial','Triangle transformé');

1. Généralisation de la transformation à un polygone en prenant pour exemple des feuilles :



tr1 = [0,0.5,0.5,5,5,9,7,9,7,6 ,2,3,1,0 ; % Polygone d'une moitié de feuille d'érable

0,0 ,2 ,1,2,4,6,8,8,10,5,12,10,12];

tr2 = [-tr1(1,:) ; tr1(2,:)]; % Autre moitié de la feuille

tr2 = fliplr(tr2); % on prend son symétrique

tr=[tr1 , tr2]; % collage des 2 parties

fill(tr(1,:),tr(2,:),'b'); % affichage de la feuille initiale

n = length(tr);

for i = 1:500;

min = -100;

max = 100;

a = (max-min)\*rand() + min;

b = (max-min)\*rand() + min;

alpha = (max-min)\*rand() + min;

rotation = [cos(alpha),-sin(alpha) ; sin(alpha),cos(alpha)];

translation = [a,b]' \* ones(1,n);

Tr = rotation \*tr +translation; % feuille transformée

hold on;

r\_min = 0.2;

r\_max= 0.7;

r = (r\_max-r\_min)\*rand() + r\_min; % limitation des tons de rouge

g\_min = 0;

g\_max= 0.4;

g = (g\_max-g\_min)\*rand() + g\_min; % limitation des tons de vert

fill(Tr(1,:),Tr(2,:),[r,g,0]); % affichage des feuilles

grid on;

end

Exercice 7 :

1. On a :

Les 3 droites nous donnent le système :

On résout le système et on trouve : avec et

Sous Matlab, on code :

M =[1,0;0,1;1,1]

N= [0;1;2]

X = N\M

Affichage : X = 0.4000 0.6000 Résultat invraisemblable

1. Tomographie réduite
2. Résolution du système pour le 1er cas :

M = [1,1,1,0,0,0,0,0,0,0 %a1

0,0,0,1,1,1,1,0,0,0 %a2

0,0,0,0,0,0,0,1,1,1 %a3

1,0,0,1,0,0,0,0,0,0 %a4

0,1,0,0,1,0,0,1,0,0 %a5

0,0,1,0,0,1,0,0,1,0 %a6

0,0,0,0,0,0,1,0,0,1 %a7

0,0,0,1,0,0,0,1,0,0 %a8

1,0,0,0,1,0,0,0,1,0 %a9

0,1,0,0,0,1,0,0,0,1 %a10

0,0,1,0,0,0,1,0,0,0 %a11

];

N =[3 4 3 2 3 3 2 2 3 3 2]';

Xk = ones(10,1);

X =M\N;

Nk=M\*Xk

On trouve :

X = [ 2 ,0 ,1 ,0 ,1 ,2 ,1 ,2 ,0 ,1]'

1. Résolution du système pour le 2ème cas :

N = [6 22 27 5 15 18 17 12 15 18 10]';

X = M\N;

for i = 1:10

Xk(i) = i;

end

Nk=M\*Xk;

On trouve :

X = [5, 0, 1, 0, 3, 10, 9, 12, 7, 8]'

1. Noyau de M :

On utilise le commande pour trouver le noyau de M .

On trouve :

ker(M) = Vect 0.0113 0.4329

-0.4156 -0.2779

0.4043 -0.1549

-0.0113 -0.4329

0.4043 -0.1549

0.0113 0.4329

-0.4043 0.1549

0.0113 0.4329

-0.4156 -0.2779

0.4043 -0.1549

Commentaires :

On ne trouve pas ce que on est censé trouve dans le deux cas. Ceci est due a la limitation de la division matricielle en Matlab. Notre matrice M ici n'est pas une matrice carrée inversible. Donc, Matlab essaye de nous donne la meilleur réponse quelle peut. En effet, ice que Matlab nous donne est X = X₀ + K avec X₀ la bonne réponse et  .

En regardant le noyau de M, on remarque qu'on pourrait construire des vecteurs pour corriger les fautes.