Extrait du Les nouvelles technologies pour l'enseignement des mathématiques <u>http://revue.sesamath.net/spip.php?article360</u>

Algèbre linéaire et géométrie dynamique

- N°27 - Novembre 2011 -

Date de mise en ligne : jeudi 13 octobre 2011

Description:

Les vecteurs sont des objets géométriques, et l'utilisation d'un logiciel de géométrie dynamique pour les enseigner est intéressante, parce qu'elle contribue à les rendre un peu moins abstraits pour les élèves.

Les nouvelles technologies pour l'enseignement des mathématiques

Il n'est pas possible de "dessiner" un vecteur, ni une matrice, dans un logiciel de géométrie dynamique. Mais on peut dessiner un représentant d'un vecteur, le manipuler dans le logiciel, et illustrer les matrices carrées d'ordre 2 par la géométrie analytique (ce sont des endomorphismes du plan vectoriel).

En bas d'article figurent des compléments d'informations de l'auteur d'un des logiciels cités.

Vecteurs

La notion de vecteur remonte à <u>Hermann Günther Grassmann</u> au milieu du 19e siècle, donc plutôt récemment à l'échelle de l'histoire des maths. Pour son contemporain Hamilton, un espace vectoriel est un espace affine dont on a fixé un point (l'origine) et un vecteur est la différence entre deux points. Lorsque les vecteurs sont apparus dans l'enseignement des maths au 20e siècle, c'est avec la traditionnelle définition de "une direction, un sens, une longueur". L'égalité des vecteurs pose alors un problème : comment peut-on considérer que deux flèches sont égales alors qu'elles ne sont manifestement pas au même endroit ? Les "maths modernes" ont alors trouvé la parade, en définition des "direction-sens-longueur". Dans le programme actuel de Seconde, l'organisation est la suivante :

- 1. On commence par définir les translations, à partir de la notion de parallélogramme ("À tout point C du plan, on associe, par la translation qui transforme A en B, l'unique point D tel que [AD] et [BC] ont même milieu."), et on leur "associe" un vecteur ;
- 2. La configuration du parallélogramme est alors répétée, à propos de l'égalité des vecteurs : Une nouvelle manière de dire que les vecteurs sont des translations ;
- L'addition des vecteurs correspond donc à la loi de groupe sur les translations, la somme correspondant à la composée de translations ("La somme des deux vecteurs ... est le vecteur associé à la translation résultant de l'enchaînement des translations ...")
- 4. Ensuite on parle d'emblée des coordonnées du vecteur. L'importance du tableur incite alors à *définir* un vecteur par ses coordonnées ; en effet, dans ce contexte, ni l'égalité de vecteurs, ni la somme ne présentent de problèmes de conceptualisation ; et
- 5. Le produit d'un vecteur par un réel est, lui, défini par les coordonnées. On gagne quelque chose à faire tout le cours sur les vecteurs de <u>manière entièrement analytique</u> : Du temps... Par contre on risque de semer la confusion entre un point (qui a deux coordonnées puisqu'en Seconde, la géométrie est repérée) et un vecteur (qui lui aussi, a deux coordonnées). On va voir plus bas comment on peut éviter ce piège.

DrGeoll

Le début se fait sans grande surprise, avec le besoin de sélectionner d'abord l'origine, puis l'extrémité du vecteur. Seulement les deux points ne sont pas créés à la volée, on doit d'abord les créer avant de sélectionner l'outil *vecteur*, et en plus on doit sélectionner deux fois l'outil *point*. C'est peut-être <u>un geste en faveur des élèves dyspraxiques</u> ?

Ceci dit, une fois les deux points créés, il suffit de les sélectionner pour avoir le vecteur les joignant :



Mais c'est à ce stade que DrGeoII est purement génial : il suffit d'attraper le vecteur avec la souris pour placer le représentant où on veut :



Il n'y a pas mieux pour montrer la distinction à faire entre vecteurs et représentants ! Et comme on peut s'y attendre, on reste dans le monde de la géométrie dynamique, où le mouvement ultérieur d'un point avec la souris modifie le représentant en conséquence, même à distance.

Pour additionner deux vecteurs avec DrGeoII, on doit évidemment créer un second vecteur (donc, au préalable, deux autres points) :

$\Theta \Theta \Theta$		Vecteur	s	O
Fichier P	Points Lignes	Transformations	Numériques	Macro-construction
	×	× ×	*	
Sélectionn	ner et déplacer u	n objet		

Seulement voilà : L'outil *somme* n'existe pas dans le menu de DrGeoII. On peut alors le créer par un script en <u>Smalltalk</u>, facile à écrire parce que l'instruction *Smalltalk* est notée par un "+" :

\varTheta 🖯 🖯 Éditer ou	créer un 🛛 🔿
DrGeoUserScripts all examples scripts private	Instance ? SommeAvec: example3: myCounter myFunction myParabola
Browse	Senders
Somme: vecteur an "additionne deux v v1 v2 v1:=vecteur direct v2:=autre direction ^v1 + v2.	rd: autre vecteurs" tion. on.

Une fois qu'on a entré *Control+S* pour enregistrer cet énorme algorithme d'addition, il subsiste encore un problème : À chaque terme de la somme, on a envoyé le message *direction* qui lui demande de dire ses coordonnées. En définitive, on a additionné les coordonnées des vecteurs, et en appelant le script, on obtient donc les coordonnées de la somme (en rouge ci-dessous), et non un représentant de celle-ci :



Pour dessiner la somme des deux vecteurs comme un vecteur, on doit donc

- créer l'origine (point de coordonnées 0 et 0, en bleu ci-dessus)
- créer l'extrémité (point de coordonnées, celles qui sont affichées ; il suffit pour cela de sélectionner l'outil *point par coordonnées* puis de cliquer sur les coordonnées en rouge)
- créer le vecteur à partir des points précédents.

Mais à ce stade, on peut déplacer le représentant de la somme où on veut, et la modification d'un des termes entraîne <u>immédiatement</u> la mise à jour de l'affichage des coordonnées ainsi que celle du représentant de la somme.

Et, dans la logique d'<u>EIAH</u> qui caractérise DrGeoII, il est à ce stade très facile de transformer la somme de vecteurs en une macro, en choisissant les deux vecteurs comme objets initiaux et leur somme comme objet final. Dans la même logique, il y a peu de modifications à apporter à la manipulation précédente pour construire une macro de multiplication de vecteur par un nombre.

++++ TracenPoche

Avec TracenPoche, les vecteurs sont créés plus rapidement parce que leur origine et extrémité sont créées à la volée (sauf qu'il faut les nommer).

Comme on peut le voir en manipulant la figure ci-dessous, il est possible de déplacer le représentant du vecteur avec la souris, l'origine et l'extrémité suivent le mouvement.

Pour additionner deux vecteurs vAB et vCD de la figure, on doit passer par le script

vecteur w=vAB+vCD;

Seulement on ne voit pas le vecteur somme parce qu'il a juste été calculé. Alors il reste à ajouter à la figure ses origine et extrémité (cette dernière obtenue comme image de l'origine *O* par la translation de vecteur *w*) :

```
O=point(0,0);
t=translation(w);
M=image(t,0);
reprw=vecteur(0,M) {rouge};
```

C'est tout ce qu'il y a à faire pour obtenir la figure que voici : <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,65,0" width="600" height="600" > <embed src="plugins/tep_plugin/tepweb600600.swf?script=IMG/txt_somvectTeP.txt" width="600" height="600" align="middle" quality="high" bgcolor="#99cccc" swLiveConnect=true id="script1" allowScriptAccess="sameDomain" type="application/x-shockwave-flash" pluginspage="http://www.macromedia.com/go/getflashplayer" />

Cette figure est aussi dynamique que possible : On peut bouger les points qui définissent les vecteurs, mais aussi bouger les vecteurs avec la souris, par déplacement parallèle. La somme ne peut pas bouger (on n'est pas dans *GeoLicia*!) mais on peut déplacer son représentant en bougeant l'origine O de celui-ci, ce qui est possible parce que ladite origine est libre dans le plan (ce que ne sont ni l'extrémité, ni le vecteur).

++++ MathGraph32

MathGraph32 semble être le premier des logiciels de GD à avoir incorporé <u>JLaTeXMath</u>, ce qui lui donne un avantage pour l'affichage dans la figure d'expressions faisant intervenir les noms des vecteurs.

Après avoir créé deux points, on peut les joindre par un vecteur, et pour peu qu'on aie deux vecteurs, on peut les additionner en choisissant l'outil "point par somme de vecteurs" puis en sélectionnant les deux vecteurs :



Comme on pouvait se permettre de l'espérer, la résultante va suivre la souris jusqu'à ce qu'on ait choisi un point (créé au préalable), mais, originalité de MathGraph32, cette résultante est montrée comme décomposée en ses deux termes :



Autre méthode

Le passage par les coordonnées est possible aussi avec MathGraph32 :

- 1. Créer un repère
- 2. Choisir l'outil abscisse et l'appliquer à chacun des 4 points
- 3. Idem avec l'outil ordonnée (on peut aller plus vite avec l'outil affixe)
- 4. Calculer par soustractions les coordonnées des deux vecteurs
- 5. Calculer leur somme
- 6. Appliquer l'outil translation par coordonnées

MathGraph32 possède aussi un outil pour créer un point par multiplication d'un vecteur par un nombre.

Pour l'affichage, on peut créer les milieux des segments, leur attacher des expressions en LaTeX puis les cacher :



++++ Xcas

Une fois que la figure est créée, on passe en mode *point* pour créer des points, qui sont automatiquement nommés A, B, C et D. Ensuite on passe en mode *vecteur* pour créer les vecteurs \$\overrightarrow{AB}\$ et \$\overrightarrow{CD}\$.

Inconvénient : Ceux-ci sont automatiquement nommés *E* et *F* par Xcas ; mais il suffit de remplacer au clavier leurs noms par *u* et *v* pour les voir apparaître :



Après, comme on le voit en bas de la figure ci-dessus, on entre

w:=u+v

pour créer la somme des deux vecteurs, et la nommer du même coup. Par défaut, Xcas l'attache au point A :



Une fois que w est créé, on peut accéder à son abscisse en entrant

abscissa(w)

ce qui permet de conjecturer la méthode de calcul de ses coordonnées.

Mais attention à ne pas essayer de bouger les représentants des vecteurs avec la souris ! Dans ce cas, ils se transforment en points et deviennent invisibles.

++++ JSXGraph

<u>JSXGraph</u> n'est pas vraiment un logiciel de GD, en raison de l'absence d'outils de création d'objets et de possibilité de sauvegarde de la figure [2]. Il se pose plutôt en afficheur de figures dynamiques. Ce qui n'empêche pas de l'utiliser pour enseigner les vecteurs !

L'addition des vecteurs est illustrée par cet exemple sur les nombres complexes. En voici le principe :

1) Dans JSXGraph, un vecteur est un segment muni d'une flèche. Si A et B sont deux points créés dans la figure, on crée le vecteur les joignant par

var u=figure.create('arrow',[A,B]);

Alors le vecteur *u* possède une équation cartésienne (puisque c'est un morceau de droite !) mais on peut accéder à l'abscisse de son origine par

u.point1.X()

avec des notations analogues pour l'extrémité et pour leur ordonnée. Une fois la flèche construite, on peut avec la souris, déplacer les deux points A et B et le vecteur se met à jour. Mais on ne peut pas bouger le vecteur avec la souris.

2) Pour avoir un représentant mobile du vecteur, on peut, toujours à partir des points A et B, faire

```
var droite=figure.create('line',[A,B],{visible:false});
var origine=figure.create('point',[0,0],{name:'0'});
var u=figure.create('arrowparallel',[droite,origine]);
```

Ainsi le représentant du vecteur \$\overrightarrow{AB}\$ partant de O est mobile, en bougeant son origine O avec la souris.

 3) Pour additionner deux vecteurs, on crée une translation puis on l'applique à D (avec le vecteur \$\overrightarrow{AB}\$). La démarche est très voisine de celle avec TracenPoche :

```
var t=figure.create('transform',[["X(B)-X(A)","Y(B)-Y(A)"],origine],{type:'translation'});
var D=figure.create('point',[origine,t]);
```

Ceci dit, JSXGraph peut importer des figures créées par GeoNext, lequel permet de créer des vecteurs aussi, avec des points créés à la volée :



Et il a aussi un outil *représentant* qui, après la sélection d'un vecteur et d'une origine, construit un représentant de celui-ci :



Avec ce choix de l'origine, on peut utiliser l'outil *quatrième point d'un parallélogramme* pour additionner les deux vecteurs :



Cette figure, une fois enregistrée puis importée dans JSXGraph, peut alors être manipulée en ligne.

++++ GeoGebra

Lorsque Grassmann inventait le calcul vectoriel, il notait [ab] le vecteur allant du point a vers le point b. Mais une coutume française consiste à appeler notation de Grassmann [3] celle consistant à noter

- A+u l'image de A par la translation de vecteur u ;
- B-A le vecteur allant de A vers B
- (aA+bB)/(a+b) le barycentre de A(a) et B(b)

Cette notation présente de nombreux avantages de par sa concision. Exemples :

- 1. La formule de Chasles se démontre ainsi : (B-A)+(C-B)=(C-B)+(B-A)=C-B+B-A=C-A ;
- 2. La droite (AB) est l'ensemble des A+tu où t est réel, ce qui permet de résumer en une seule ligne sa représentation paramétrique ;
- 3. la notation (A+B)/2 pour le milieu de [AB] montre immédiatement le lien entre le milieu et la moyenne, ainsi que la raison pour laquelle cette notion pose des problèmes en caractéristique 2...

D'ailleurs, cette notation a été réinventée par un élève de Seconde ; très clairement pour lui, il voulait résumer l'expression "on additionne les coordonnées de A et celles de u" par le signe "+" (sans doute l'habitude du tableur où on additionne des colonnes assez régulièrement).

À propos de notation, l'usage moderne de mettre des flèches au-dessus des noms des vecteurs a mis <u>du temps à se</u> <u>généraliser</u> en raison d'une forte résistance des imprimeurs ! Ainsi, Grassmann notait les vecteurs avec des lettres minuscules.

Il est très probable que Markus Hohenwarter ait suivi des cours notés avec la "notation de Grassmann", puisque cette fameuse notation est utilisée dans GeoGebra depuis l'acte de naissance de celui-ci il y a 10 ans.

On sent une longue réflexion sur l'anticipation du logiciel, avec des conventions très logiques et qui font gagner énormément de temps pour créer et manipuler des vecteurs. Ainsi, alors qu'il est possible de créer un vecteur par deux points comme dans tout logiciel de GD (avec la création des points à la volée), on constate que le nom des points est en majuscule, alors que celui des vecteurs, comme pour toute ligne d'ailleurs, est en minuscule. Donc, si on tape

A=(2,1)

on crée un point A de coordonnées (2,1), alors que si on tape

u=(2,1)

c'est un vecteur *u* qui est créé, et dans la "fenêtre algèbre", ce vecteur est affiché sous forme d'une matrice colonne (à laquelle il est d'ailleurs noté comme égal). En résumé, les points sont des matrices ligne, et les vecteurs des matrices colonne. C'est bien comme ça que nous notons leurs coordonnées au tableau, non ?

Pour additionner deux vecteurs u(2,1) et v(-3,1), après les avoir créés, il suffit d'entrer

w=u+v

seulement le vecteur somme est attaché à l'origine du repère (mais avec l'outil *translation* on peut avoir un autre représentant, attaché à un autre point M, par exemple en tapant "P=M+w") :



Le représentant du vecteur *v* par contre peut être déplacé avec la souris [4]. Ce qui permet par exemple de passer de la configuration "parallélogramme" ci-dessus à la configuration "Chasles" :



Pour multiplier le vecteur u par 3, il suffit d'entrer

3*u

et le nouveau vecteur est dessiné, ses coordonnées affichées, et mises à jour à chaque modification de *u*. Et bien entendu pour avoir l'image de A par la translation de vecteur u, on entre B=A+u.

De nombreux risques de confusion sont dénoncés <u>ici</u> et <u>là</u> mais il faut avouer que c'est quand même bien pratique ! Ceci dit, expliquer aux élèves pourquoi ils ne peuvent pas donner à un vecteur un nom en majuscule, c'est moins rigolo ! Et leur dire que l'addition des points, également permise par GeoGebra, ne l'est pas dans le cours de maths [<u>5</u>], c'est encore une autre histoire.

Autre problème de notation : Le produit scalaire est appelé en anglais "dotproduct" et de fait, usuellement noté par un point. Ce choix n'a pas été fait dans GeoGebra, le symbole de multiplication "*" lui ayant été préféré :

0	GeoGebra - Error
	Invalid input: u.v
	OK

Ce qui pose un problème de cohérence lorsqu'on passe aux nombres complexes : On peut additionner, soustraire et diviser des points (parce que GeoGebra les confond avec leurs affixes) mais pour les multiplier il faut aller chercher une multiplication notée tensoriellement quelque part dans les tréfonds du menu des caractères spéciaux...

Une parade serait de créer un texte, le mettre en mode LaTeX, et là, placer un point comme produit scalaire. Nouveauté de GeoGebra 4 : On peut insérer dans le code LaTeX des "objets" de la figure qui sont affichés en LaTeX, ici les coordonnées des vecteurs et le produit scalaire :

$vec{u}.vec{v} = \overline{x(u)}$	times(x(v)) +	y(u) \times y(v) =a \$
•	III	
🖌 LaTeX formula 🗸	Symbols 👻	Objects 🗸
π		
nanananananananananananananananananana		
Fleview		
$\vec{u}.\vec{v} = 2 \times (-3) + 1 \times$	1 = -5	

L'effet est à la fois "pro" et dynamique (l'affichage du produit scalaire change si on bouge les points à la souris) :



++++ CaRMetal

<u>CaRMetal</u> possède un objet *vecteur* depuis la version 3.7.2 [6]. Et comme cet objet a été créé par <u>un prof de maths</u> <u>en lycée</u>, on peut s'attendre à ce que CaRMetal gère les vecteurs de manière plus conforme à l'enseignement français, que ne le ferait un logiciel étranger. En particulier, bien que la syntaxe des expressions de CaRMetal ressemble à celle de GeoGebra, CaRMetal n'utilise pas la "notation de Grassmann".

On peut créer un vecteur par JavaScript, mais le plus simple est de passer par deux clics qui créent l'origine, puis l'extrémité, à la volée (sauf choix contraire) :

Aspo Nom	ect <u>Numé</u> : u	rique Condition		<r> <g> </g></r>		<mark>0</mark> □,	-	
					(CaRMetal		
chier	Édition	Construction	Affichage	Macros	Javascript	Exercice	Aide	
				0	ū		10	

Dans CaRMetal, le vecteur a deux noms : Ci-dessus, le nom officiel est u ce qui permet facilement de l'utiliser (par exemple avec l'expression x(u) qui donne son abscisse), alors que le nom affiché est le même u mais surmonté d'une flèche. Ceci est un compromis entre la facilité d'utilisation et la qualité de l'affichage. Et il reste la possibilité de remplacer cet affichage par quelque chose de plus élaboré en utilisant le champ "alias" (c'est l'alias qui sera affiché à la place du nom officiel).

Si on affiche la "valeur", on obtient comme valeur d'un vecteur, ses coordonnées :



Au passage l'affichage des coordonnées du point se fait également sans le signe "=" qui agace tant d'utilisateurs de GeoGebra [7] :



En cliquant sur l'onglet "numérique" du vecteur, on accède non seulement à sa longueur [8] mais aussi à ses coordonnées, qu'il est possible de modifier au clavier :



On peut déplacer le représentant du vecteur, soit en bougeant son origine, soit en bougeant la flèche avec la souris :



Pour additionner deux vecteurs de coordonnées respectives (2 ;1) et (-3 ;1) on peut commencer par les créer et les nommer respectivement *u* et *v*, puis créer un vecteur au hasard (*w*, en rouge ci-dessous) :

Aspe	ct Numé	rique Condition	nel X	-1,571	5976331360	0944		
Nom	: w	<alias></alias>		: 2,2721 Fixe	893491124	26		
					(CaRMetal		
hier	Édition	Construction	Affichage	Macros	Javascript	Exercice	Aide	
				Q				
				P				
				\overrightarrow{w}	\backslash	()		
						$\vec{u} \begin{pmatrix} 2 \\ 1 \end{pmatrix}$	2)	
						(1		
				~	<u> </u>	~		
			\vec{v}	-3	0	0		
				1)				

puis de lui donner comme abscisse, la somme de celles de u et de v, soit x(u)+x(v) (le calcul de l'ordonnée la plus logique est alors fait automatiquement par CaRMetal) :



Et après avoir créé une somme de deux vecteurs, on peut encore, comme avec DrGeoII, déplacer le représentant de la somme !

On peut multiplier un vecteur par un réel (bien entendu, éventuellement variable comme un curseur, l'abscisse d'un point etc.) avec une méthode analogue, ou alors en utilisant JavaScript, qui, outre les instructions *SetColor*, *SetAlias* etc. de tous les autres objets, possède des instructions pour accéder à des informations sur le vecteur :

- 1. Origin(u) qui retourne un point ;
- 2. *Extremity(u)* qui retourne un point ;
- 3. X(u) qui retourne un nombre (l'abscisse du vecteur u);
- 4. et Y(u) qui retourne un nombre (l'ordonnée)

En plus, l'instruction Move("u", x, y) permet d'affecter au vecteur u les coordonnées x et y. Par exemple, avec la figure ci-dessus, si on veut ajouter le double de u, on peut créer un vecteur d'origine A (et dont l'extrémité est initialisée à (0,0) dans la ligne 1), puis (ligne 2) afficher le nom de son extrémité) et enfin lui donner les coordonnées voulues (ligne 3) :

```
1 t=Vector("A",Point(0,0));
2 Println(Extremity(t));
3 Move(t,"2*x(u)","2*y(u)");
```

Il va de soi que c'est bien plus facile à faire directement qu'avec JavaScript, mais ces instructions permettent d'utiliser JavaScript pour tracer des champs de vecteurs. Avec le script ci-dessus, en cliquant sur le feu vert, on obtient ceci (qui permet de savoir que l'extrémité du vecteur s'appelle *P3*) :

00	
Р3	

et cela (qui permet de savoir que le vecteur nouvellement créé s'appelle v4) :

Nom	ect Numé	érique Condition		<r> <g> </g></r>		<mark>0</mark> □,	-	B 0,20 Å
					(CaRMetal		
ichier	Édition	Construction	Affichage	Macros	Javascript	Exercice	Aide	
0			4	v	w i	$\vec{l} \left(\begin{array}{c} 2\\ 1 \end{array} \right)^{\vec{v_4}}$	$\begin{pmatrix} 4\\2 \end{pmatrix}$	

Pour le produit scalaire, il suffit de créer une expression et de lui donner la bonne valeur [9] :

	Exp : x(u)*x(v)+y(u)*y(v) Curseur : -5 Commentaire : Valeur
	000
3	
-5	

Après ça il est facile de transformer le produit scalaire en macro, en prenant les deux vecteurs comme initiaux, et l'expression comme final, puis de comparer avec d'autres macros donnant d'autres expressions du produit scalaire. Par exemple, le carré de la norme peut se définir algébriquement comme

x(u)*x(u)+y(u)*y(u)

ou alors géométriquement comme

u*u

Le déterminant de deux vecteurs peut être défini, comme le produit scalaire, par une expression.

Si le point M est attaché à une courbe (cercle, conique, représentation graphique de fonction, ...), on peut définir un vecteur ayant M pour origine, et lui donner les coordonnées (d(x(M)),d(y(M))) pour découvrir (ou faire découvrir) expérimentalement la notion de vecteur tangent et la <u>géométrie différentielle</u>...

Matrices

Lorsque <u>Arthur Cayley</u> inventait les matrices au milieu du 19e siècle (donc à peu près en même temps que Grassmann inventait avec Hamilton le calcul vectoriel), cela faisait plus de 100 ans que <u>Gabriel Cramer</u> les inversait, et 3 siècles que <u>Girolamo Cardano</u> calculait des déterminants d'ordre 2. Cayley utilisait des traits verticaux pour délimiter ses matrices, puis Wedderburn dédoublait les traits verticaux : Là encore, la notation a mis du temps à faire son chemin parmi les barricades des imprimeurs. Aujourd'hui, les anglosaxons utilisent des crochets et en Europe, c'est plutôt les parenthèses (mais celles-ci tendent à se généraliser si on en croit l'usage de LaTeX). Les traits verticaux de Cayley sont maintenant utilisés pour les déterminants, comme <u>ici</u>.

L'ancien programme de spé maths en 1ère ES suggérait une approche qui a fait ses preuves :

- 1. Commencer par distinguer les vecteurs ligne des vecteurs colonne, et généraliser à d'autres dimensions que 2 et 3 ;
- 2. Faire le lien entre les vecteurs ligne et colonne, avec la notion de transposition ;
- 3. Redéfinir le produit scalaire des vecteurs ligne X et Y comme produit de X par la transposée de Y;
- 4. Passer aux matrices carrées, et leur généraliser le produit précédent, en l'appliquant à chaque ligne de la matrice : Le produit d'une matrice carrée par un vecteur colonne est un vecteur colonne ;
- 5. enfin généraliser au produit de deux matrices carrées, puis de deux matrices rectangulaires ; ce qui mène aux notions de puissance et d'inverse d'une matrice carrée, puis à la résolution de systèmes, etc.

JSXGraph

JSXGraph permet d'entrer une matrice dynamique en définissant ses coefficients par des curseurs, et peut afficher le produit de vecteurs colonnes par la matrice, comme le montre <u>cet exemple</u>. Et grâce à son mode LaTeX, il peut afficher la matrice elle-même dans la figure, comme on peut le voir <u>ici</u>.

L'intérêt des logiciels de géométrie dynamique pour les matrices est la possibilité de piloter certains de leurs coefficients par un curseur.

++++ Tableur

Cet onglet, un peu hors sujet, est surtout destiné à préparer au suivant qui montre comment GeoGebra fait comme

les tableurs classiques mais en plutôt mieux (ne serait-ce que pour l'affichage des matrices).

Donc, un tableur comme celui de <u>Libre Office</u> possède des fonctions matricielles, une matrice étant reconnue comme une zone de cellules. Libre Office peut calculer

- des sommes de matrices de même dimensions
- des produits de matrices dont les dimensions correspondent
- des transposées de matrices
- des déterminants et inverses de matrices carrées.

Et on peut utiliser l'un des langages de programmation disponibles [10] pour créer d'autres fonctions sur les matrices

Le problème étant celui de l'affichage, les parenthèses n'étant pas encore fournies avec le tableur pour l'instant. Alors pour l'instant, un bon moyen de mettre une matrice en exergue est de la border, colorier etc. pour la distinguer du reste de la feuille.

Ci-dessous on va vérifier visuellement que **Z** est un <u>sous-groupe à un paramètre</u> de SL(2,**Z**) ; le paramètre sera un curseur à valeurs entières (ou plutôt deux, un par matrice ; chaque curseur ayant la même couleur que la matrice qu'il pilote) :

	Α	В	С	D	E	
1	1	2				
2	0 1					
3						
4	1	3				
5	0	1				
6						

Pour afficher le produit des deux matrices, on peut utiliser la multiplication matricielle de Libre Office, et entrer avec *Ctrl+Shift+Entrée* ce qui a pour effet de créer toute la matrice (on voit que Libre Office a mis la formule entre accolades pour la rendre matricielle ; on ne voit pas que les matrices ont été sélectionnées avec la souris) :

A 8		▼ foo	Σ Ξ {	=MMULT(A1:	32;A4:B5)}	
	A	В	С	D	E	
1	1	2				
2	0	1				
3						
4	1	3				
5	0	1				
6						
7						
8	1	5				
9	0	1				
10						

En manipulant les curseurs, on peut émettre une conjecture sur le produit de ces matrices triangulaires. Mais pour peu qu'on ait utilisé la multiplication matricielle de <u>CMathOOoCAS</u>, on peut même la démontrer, cette conjecture ! Il

suffit pour cela de remplacer les entrées cursorisées par des lettres a et b et on a la preuve par le tableur formel :



CMathOOoCAS implémente aussi des fonctions matricielles qui n'existent pas nativement dans le tableur :

- puissances de matrices carrées (nécessaires pour les activités présentées dans les onglets suivants)
- valeurs propres des matrices carrées (données sous forme d'expressions formelles)
- vecteurs propres des matrices (calcul formel aussi)

Et le calcul formel sur les matrices permet par exemple de résoudre un système dont les coefficients sont eux-mêmes <u>formels</u>. Voici ce que ça donne avec des matrices, le principe étant de condenser l'écriture du système sous la forme *AX=b* où *A* est une matrice carrée (les coefficients du système), *X* un vecteur colonne (celui des inconnues) et *b* un autre vecteur colonne (le second membre). Alors la solution du système s'obtient en multipliant *b* à gauche par l'inverse de *A*, ce qui, avec des curseurs, donne quelque chose comme ça :



Procédé de fabrication

Les coefficients du système sont dans A1 à B2 et sont pilotés par des curseurs ; le second membre est dans C1 et

C2 et ses composantes sont, elles aussi, commandées par des curseurs. Les coefficients de la première équation sont en bleu, ceux de la seconde équation en rouge.

L'affichage à droite est obtenu avec des formules (comme =A1) mélangées à du texte fixe (les x, y et opérations). Il permet de mieux lire le système et n'a pas d'utilité pour la suite.

L'inverse et la solution ont été obtenus en appliquant les fonctions matricielles de Libre Office, et la solution a été mise en forme avec des formules comme =*A9* dans *B13*.

En manipulant les curseurs, on voit toute une foule de systèmes défiler, dont certains ont des solutions non décimales, mais calculées à la vitesse des curseurs.

Si maintenant on remplace les fonctions matricielles de Libre Office par celles de CMathOOoCAS, et le =A9 de B13 par un =*simplifier(A9)*, on a l'affichage exact de la solution du système :

	Α	В	С	D	E	F	G	Н	I J	K L	М
1	4	4	-7						4 x+	4 v=	-7
2	5	2	3						5 x+	2 y=	3
3											
4	inverse :				•		•				
5	-1/6	1/3									
6	5/12	-1/3			•		• • • • • • • • • • • • • • • • • • •				
7											
8	produit :				•		•				
9	-7*(-1)/6+	1/3*3									
10	-7*5/12+(1)/3*3									
11											
12	solution :				•		•				
13	x=	13/6									
14	y=	-47/12									
15											
16											

Et on peut même rendre littéral l'un (ou plusieurs) des coefficients :

	Α	В	С	D	E	F	G	Н	I J	K L M
1	-5	a	-2						-5 x+	a y= -2
2	10	2	-4						10 x+	2 y= -4
3										
4	inverse :				•					
5	2/(-10*a-10)	(-a)/(-10*a-10)								
6	-10/(-10*a-10)	-5/(-10*a-10)					▶			
7										
8	produit :				4		►			
9	-2*2/(-10*	a-10)-4*(-	a)/(-10*a-	10)						
10	-2*(-10)/(-	10*a-10)-	4*(-5)/(-10)*a-10)		•			
11										
12	solution :				•		Þ			
13	x=	(-2*a+2)	(5*a+5)							
14	y=	-4/(a+1)								
15										

Ce qui montre que la solution du système

\$\$-5x+ay=-2\$\$

\$\$10x+2y=-4\$\$

est

 $x=\frac{-2a+2}{5a+5}$

\$\$y=-\frac{4}{a+1}\$\$

Bonus : Le corrigé du Bac ES Amérique du Nord mai 2008 :



Comme dans l'onglet sur les chaînes de Markov, on a préféré ici l'écriture des vecteurs de probabilités en colonne plutôt qu'en ligne, ce qui fait que la matrice de Markov est la transposée de celle du sujet de Bac.

La feuille de calcul numérique permet de calculer les puissances de cette matrice et donc de conjecturer l'existence d'une probabilité stationnaire (la limite de la suite). C'était d'ailleurs le seul attendu du sujet de Bac ES. Mais la feuille de calcul formel permet de calculer la valeur exacte de cette limite, et donc d'être plus exhaustif sur cet exercice que ne l'attendait son auteur. On apprend donc qu'après beaucoup de feux, Mathurin a

- 178 chances sur 209 de passer au vert ;
- une chance sur 19 de passer à l'orange ;

• et donc 20 chances sur 209 de s'arrêter au feu rouge.

++++ GeoGebra

Pour GeoGebra, un vecteur n'est pas un objet de même nature qu'une matrice ligne ou colonne, malgré une certaine similitude dans l'affichage :

Algèbre
$$\mathbf{v} \in \mathbb{D}^n \times$$
 \bigcirc Objets libres \bigcirc $\mathbf{u} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ \bigcirc $\mathbf{v} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ \bigcirc $\mathbf{v} = \{3, 2\}$

Le code qui a permis cela est

```
u=(3,2)
v={{3},{2}}
w={3,2}
```

La couleur verte est attribuée par GeoGebra aux listes, c'est donc ce que sont *v* et *w*. Plus précisément, *w* est une liste de deux nombres (donc aucun rapport avec l'algèbre linéaire) et *v* est une liste de listes, donc une matrice. On peut aussi créer *v* avec

$v=Transpose[\{\{3,2\}\}]$

la matrice entre crochets étant une matrice ligne.

D'ailleurs GeoGebra ne possède pas de concaténation pour assembler une matrice carrée en collant bout à bout deux vecteurs colonne (comme le fait par exemple <u>Euler Math Toolbox</u>).

Ceci dit, et malgré l'abondance d'accolades nécessaires, il est très facile de créer une matrice carrée avec GeoGebra : Avec le tableur ! Ainsi, on commence par créer deux curseurs entiers *a* et *b*, puis, dans le tableur, on écrit 1 dans les cellules A1 et B2, 0 dans la cellule A2 et *a* (la valeur du curseur) dans la cellule B1 :

Algebra	*≣ ⊡" X 👌	Graphics	Ø	*≣ ⊡" ×	Spre	adsheet		
E Free Objects		(-4.3, 6.3)				А	В	
– 🥥 a = 12					1	1	12	
$- \bigcirc \mathbf{b} = -1$					2	0	1	
Dependent Objects			а	= 12	3			
				-	4			
			b = -1		5			
			•		6			
	3			3	_			

Ensuite, après avoir sélectionné les 4 cellules avec la souris, on effectue un simple clic droit qui permet de créer une matrice à partir de la plage de cellules (il n'y a pas de limite aux dimensions de la matrice); ce qui, en modifiant la matrice créée (ici, A), permet de voir sa syntaxe :

Algebra ■ Free Objects → ② a = 12	Graphics	↑ Object Properties
<pre></pre>	Objects Clist Clist Objects A Clist Objects A A Clist Objects A Clist Objects A Clist Objects A Clist Objects A Clist Objects A	Basic Color Style Advanced Scripting Name: A Definition: {{A1, B1}, {A2, B2}} Caption:

On peut alors s'inspirer de cette syntaxe pour créer la matrice B avec

B={{1,b},{0,1}}

On a désormais deux éléments variables du groupe **G** des matrices triangulaires supérieures avec des 1 sur la diagonale et un entier au-dessus.

Pour vérifier l'homomorphisme entre Z et G, il suffit de créer le produit des deux matrices avec

P=A*B

puis de manipuler les curseurs :

Algebra	*≣ ⊡" X	Graphics
Free Objects		
– 🥥 a = 12		
└─		
🗏 Dependent Objects		a = 12
$- \circ \mathbf{A} = \begin{pmatrix} 1 & 12 \\ 0 & 1 \end{pmatrix}$ $- \circ \mathbf{B} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$		b = -1
$- \circ P = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$		

On peut aussi calculer l'inverse de la matrice A avec

Inverser[A]

GeoGebra nomme cet inverse C :



Et on peut aussi calculer une puissance de A avec

A^n

(*n* peut être un curseur...)

Ce qui permet de conjecturer que \$\left(\begin{array}{rr}1&1\\0&1\end{array}\right)^n=\left(\begin{array}{rr}1&n\\0&1\end{array}\right)\$ (très bon exercice de démonstration par récurrence d'ailleurs).

La résolution du système *AX=b* en multipliant *b* par l'inverse de *A*, vue dans l'onglet précédent, est presque immédiate par GeoGebra, et peut se gérer dynamiquement (avec des équations de droites par deux points). C'est par cette méthode que GeoGebra détermine la conique par 5 points (il y a un système de 5 équations à résoudre) et, plus généralement, les courbes algébriques, comme la <u>cubique par 9 points</u>.

++++ CaRMetal

Comme GeoGebra et MathGraph32, CaRMetal utilise la bibliothèque libre <u>JLaTeXMath</u> de Calixte Denizet, et donc comme GeoGebra et MathGraph32, CaRMetal peut afficher dans la figure des matrices dynamiques (dont certains coefficients peuvent varier en manipulant la figure). Cette possibilité était déjà présente avec l'ancienne bibliothèque LaTeX, et avait déjà été illustrée dans <u>un autre article de cette revue</u>.

Comme une matrice est, algorithmiquement parlant, un tableau de nombres, les <u>CaRScript</u>s offrent une ouverture intéressante vers le monde des matrices. Ci-dessous on va voir comment on peut définir une matrice avec ses deux vecteurs colonne. Le point de départ sera donc la construction des deux vecteurs. Et pour améliorer l'affichage, on va mettre les incréments de leurs origines et extrémités à 1, ce qui aura pour effet de limiter à des positions entières leurs coordonnées :

Aspect	Numérique	Conditionnel	X :	1,0		
Nom	^	-Aliana III	Υ:	-1,0		
Nom :	<u>A</u>	-Allasz		Fixe Incréme	nt: 1,0	0

Une fois que les 4 points ont été construits, il suffit de les joindre pour avoir deux vecteurs, donc une matrice :



Ensuite, c'est au tour de JavaScript d'entrer en lice, avec la construction d'un tableau, qui sera modifié en matrice. Pour cela il suffit de cliquer sur l'icône correspondante :

	J	avas	scrip	t	
cos	sin	tan	acos	asin	atan
abs	ceil	floor	rnd	min	max,
π	е	x ⁿ	e^{x}	In	$\sqrt{\mathbf{x}}$
ou	et	no	0	Щ	6

Le script obtenu révèle la syntaxe des matrices en JavaScript (qui est assez proche de celle de GeoGebra, avec des crochets à la place des accolades). Le premier script, qui modifie une variable texte appelée "TA" et préconstruite, fabrique une chaîne de caractères en LaTeX et l'affiche dans la figure :

	🥺 🕎 🔮
1	A = [[1,2],[3,4]];
2	aff="\$\\left(\\begin{array}{rr}";
3	aff+=A[0][0]+"&"+A[0][1]+"\\\\";
4	aff+=A[1][0]+"&"+A[1][1]+"\\end{array}";
5	aff+="\\ right)\$ ";
6	<pre>SetText("TA",aff);</pre>

En cliquant sur le feu vert, on obtient ceci :



Maintenant qu'on peut afficher le contenu d'une matrice, on peut rendre celle-ci dynamique, en donnant à ses coefficients les coordonnées des deux vecteurs :

 $\mathbf{1} A = [[X("u"), X("v")], [Y("u"), Y("v")]];$ 2 aff="\$\\left(\\begin{array}{rr}"; **3** aff+=A[0][0]+"&"+A[0][1]+"\\\\"; 4 aff+=A[1][0]+"&"+A[1][1]+"\\end{array}"; 5 aff+="\\right)\$"; 6 SetText("TA",aff);

Pour que l'affichage de la matrice soit mis à jour à chaque mouvement d'un des points, il suffit de le demander au gestionnaire des scripts :

Gestionnaire de scripts 🛞		
Cliquer ou glisser : A;B;C;D		
Le script se lance à l'ouverture de la figure		
Renommer : matrice1	9	
	n and	
	$\vec{v} \begin{pmatrix} -1 \\ 2 \end{pmatrix}$	$\vec{u}\begin{pmatrix} 2\\ 2 \end{pmatrix} \qquad B$
		(-)
	U	
		o A

Après ça, en bougeant un des points, on a quelque chose comme ça :



Cette activité basée sur un "CaRAScript", présente l'avantage de montrer le lien qu'il y a entre matrices et endomorphismes, et l'importance des bases pour décrire les endomorphismes. C'est donc une approche géométrique des matrices qui est suggérée ici, mais que peut-on attendre d'autre avec des logiciels de géométrie dynamique ?

On peut alors recréer les opérations sur les matrices avec des fonctions JavaScript, comme par exemple le produit :

1	<pre>function ProdMat(A,B){</pre>
2	a=A[0][0]*B[0][0]+A[0][1]*B[1][0];
3	b=A[0][0]*B[0][1]+A[0][1]*B[1][1];
4	c=A[1][0]*B[0][0]+A[1][1]*B[1][0];
5	d=A[1][0]*B[0][1]+A[1][1]*B[1][1];
6	<pre>return Array([a,b],[c,d]);</pre>
7	}
8	
9	<pre>M = new Array([1,2],[0,1]);</pre>
10	<pre>N = new Array([1,3],[0,1]);</pre>
11	<pre>Println(ProdMat(M,N));</pre>

Toutefois, les créer comme des macros peut être préférable, en particulier pour illustrer les puissances de matrices comme des systèmes dynamiques, comme on le verra dans l'onglet suivant. D'autres exemples sont téléchargeables ci-dessous.

++++ Chaînes de Markov

Pour mieux montrer l'intérêt de la géométrie dynamique avec les matrices, on va prendre un exemple tridimensionnel, en l'occurence le sujet de spécialité maths du bac ES 2011.

Cliquer ici pour voir le sujet

EXERCICE 2

Candidats ayant suivi l'enseignement de spécialité

5 points

Chaque année, une association de cyclotourisme prépare de nouveaux circuits. Pour satisfaire ses nombreux membres, elle élabore des circuits de différents niveaux : « niveau facile », « niveau moyen » et « niveau difficile ».

Au premier janvier 2010, l'association a fait son bilan :

- 20 % de ses adhérents ont choisi le niveau facile, noté A
- 70 % de ses adhérents ont choisi le niveau moyen, noté B
- 10 % de ses adhérents ont choisi le niveau difficile, noté C

Pour répondre aux attentes des adhérents et les fidéliser sur le long terme, une enquête est effectuée.

Il s'avère que, d'une année à l'autre :

- parmi les adhérents ayant choisi le niveau A, 40 % restent à ce niveau et 60 % passent au niveau B,
- parmi les adhérents ayant choisi le niveau B, 70% restent à ce niveau et 20% reviennent au niveau A et les autres passent passent au niveau C,
- parmi les adhérents ayant choisi le niveau C, 85 % restent à ce niveau et les autres reviennent au niveau B.

On note :

- · A l'état « l'adhérent a choisi le niveau A »,
- B l'état « l'adhérent a choisi le niveau B »,
- C l'état « l'adhérent a choisi le niveau C ».

Pour *n* entier naturel positif ou nul, on note $P_n = (a_n \ b_n \ c_n)$ la matrice ligne donnant l'état probabiliste de la répartition dans les différents niveaux (indiqués dans l'ordre donné dans l'énoncé au premier janvier de l'année 2010 + *n*. Ainsi $P_0 = (0, 2 \ 0, 7 \ 0, 1)$.

On se décide se baser uniquement sur ces résultats pour prévoir l'évolution de la répartition à partir du premier janvier 2010 (on néglige donc les nouveaux abonnés et les départs).

- 1. Représenter cette situation par un graphe probabiliste de sommets A, B et C.
- 2. Reproduire et compléter la matrice de transition *M* de ce graphe probabiliste, en respectant l'ordre alphabétique des sommets.

$$M = \begin{pmatrix} \cdots & \cdots & 0 \\ 0, 2 & \cdots & \cdots \\ \cdots & 0, 15 & \cdots \end{pmatrix}$$

3. Une seule des trois matrices Q, R, T ci-dessous correspond à l'état probabiliste stable.

$$Q = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 3 & 1 \\ \end{pmatrix} \quad R = \begin{pmatrix} 1 & 1 & 1 \\ 6 & 2 & 1 \\ \end{bmatrix} \quad T = \begin{pmatrix} 1 & 4 \\ 5 & 5 & 0 \\ \end{pmatrix}$$

Le président de l'association affirme que 50 % des adhérents choisiront après un certain nombre d'années le niveau B. Cette affirmation est-elle correcte ?

Choix de la notation

Matriciellement, ce genre d'exercice peut se traiter de deux manières :

- 1. En multipliant (à droite) un vecteur ligne de probabilités par une matrice carrée, dont les lignes sont des probabilités conditionnelles ;
- 2. En multipliant à gauche un vecteur colonne par la transposée de la matrice précédente.

Dans les deux cas, le problème revient à calculer les puissances de la matrice de transition, mais le second formalisme a été choisi ici parce qu'il revient à étudier les itérés d'un endomorphisme, ce qui montre mieux à la fois

l'aspect géométrique du problème et son aspect <u>système dynamique</u>. Or il se trouve que la notation utilisée dans l'enseignement supérieur et dans les sujets de Bac est la première [<u>11</u>]. Le choix fait ici n'est donc pas standard. Cependant, le passage d'un formalisme à l'autre se fait par transposition des matrices et de leur produit.

La matrice ci-dessous sera la transposée de celle de l'énoncé, et les vecteurs seront considérés comme des matrices colonne et non des matrices lignes comme dans l'énoncé.

La plupart des sujets du Bac ES portent sur des matrices d'ordre 2, mais comme les vecteurs vérifient tous x+y=1, les orbites d'un point sont toutes situées dans la même partie du plan, à savoir l'intersection de la droite d'équation x+y=1 et du premier quadrant (un segment). Visuellement, ce n'est pas terrible. Aussi a-t-on choisi ici un problème sur trois niveaux, qui est en apparence un problème de dimension 3. En apparence seulement parce que cette fois-ci, les vecteurs (x,y,z) sont tous dans le plan d'équation x+y+z=1, ou plus précisément dans l'intersection de ce plan et du premier octant, qui est un triangle équilatéral :



On peut donc déplacer le problème dans le plan de ce triangle, et dans ce cas le point (x,y,z) est représenté par le barycentre des sommets du triangle affectés des coefficients x, y et z. Ce qui est hors programme mais facilite la facture de la figure :

- On commence par placer un point P à l'intérieur du triangle (tout simplement en cliquant sur "dans l'objet") ;
- On construit les triangles APB, BPC et CPA, et on les cache ;
- On crée trois expressions égales aux quotients des aires de ces triangles cachés par celle d'*ABC*. Par exemple, x0 est poly2/poly1 ;
- On crée les expressions x1, y1 et z1 par produit matriciel ; par exemple, x1 est 0,4*x0+0,2*y0 ;
- On ajoute le point défini comme barycentre de A(x1), B(y1) et C(z1); en fait on définit son abscisse comme x1*x(A)+y1*x(B)+z1*x(C) et on laisse CaRMetal adapter son ordonnée;
- On crée une macro ayant les trois expressions x0, y0, z0 (et les trois sommets du triangle) comme initiaux, et les expressions x1, y1, z1 et le point nouvellement créé comme finaux ; on rend "fixes" les trois sommets du triangle pour qu'il ne soit pas nécessaire de les cliquer à chaque fois ;
- On applique répétitivement la macro, et il ne reste plus qu'à cacher les expressions et mettre un peu de LaTeX par ci-par là pour avoir la figure, sur laquelle on constate à la fois en regardant les matrices et vecteurs, et les

points, que

- 1. La suite des vecteurs converge (ça se voit surtout sur les points) ;
- 2. La limite semble indépendante du vecteur initial.

Voici la figure (qu'il est préférable de télécharger et manipuler en local, la version en ligne reconnaissant mal la molette de la souris) :

<applet archive="plugins/auto/eva_geometrie/scripts/CaRMetal.jar" code="Main.class" width="1024" height="658" mayscript="true">

Pour finir le corrigé de l'exercice, il ne reste plus qu'à placer le point P sur le point E...

Pas facile, hein ?

En fait il faut télécharger la figure (en bas de l'article) et l'ouvrir avec CaRMetal. Puis, détacher le point P du triangle, mettre son incrément à zéro, et le rendre magnétiquement attiré par le point E. Après on arrive mieux à le coller sur E :



L'avantage de cette projection du problème dans un hyperplan est qu'on peut traiter aussi des problèmes à quatre probabilités, en regardant une suite qui converge dans un tétraèdre régulier. Mais pour cela il faut utiliser des barycentres dans l'espace et là, on risque d'être un peu hors programme...

++++ Exemples CaRMetal

Le diaporama téléchargeable ci-dessous illustre le paradigme des matrices comme couples de vecteurs, avec des macros qu'on peut récupérer dans les figures du dossier :



diaporama CaRMetal Les matrices vues comme couples de vecteurs, et une introduction progressive de la notion de produit.

La dernière figure résume un système de 2 équations à deux inconnues, à une matrice carrée et un vecteur (le second membre), et on peut résoudre dynamiquement un tel système en utilisant les macros en question.

Le diaporama s'inspire de la théorie de Grassmann puisqu'il définit le déterminant comme une aire orientée et pas comme le résultat d'un calcul.

Le diaporama ci-dessous est la base d'un exercice inédit jusqu'ici, basé sur l'idée qu'un fichier numérique, lors de son téléchargement, est une suite de 0 et de 1 qu'on peut considérer comme aléatoires, et peut être modélisé par un processus de Bernoulli : Ce modèle du canal binaire bruité est utile en théorie des codes correcteurs d'erreur. Mais aussi en Terminale puisque

- le processus est un cas particulier de chaîne de Markov : C'est ici qu'interviennent les matrices ;
- Les variables aléatoires suivent une loi binomiale, ce qui fait que les matrices sont d'ordre 2 :



matrices et probas conditionnelles Une activité inédite aux saveurs d'ISN (bits aléatoires de Bernoulli sur un canal bruité)

On notera la présence d'un arbre pondéré dynamique dans la seconde figure...

Post-scriptum :

Réponse de l'auteur d'un des logiciels

Bonjour.

Je me réjouis de voir qu'on parle (un peu) de MathGraph32 dans cet article. Mais comme, à le lecture de cet article, je constate que certains logiciels sont toujours plus mis en avant, je me permets d'apporter quelques compléments sur MathGraph32 et les vecteurs.

L'outil translation permet de créer l'image d'un objet par une translation. On clique en premier, soit sur un vecteur, soit sur deux points (origine et extrémité du vecteur). Un vecteur clignote alors à l'écran et il suffit de cliquer sur un objet pour avoir son image (y compris les arcs de cercles ou polygones). On peut dans la foulée recliquer sur un autre objet pour avoir son image (y compris cliquer sur l'objet image précédemment créé). Ainsi la notion de translation est intimement liée à la notion de vecteur. A noter aussi l'outil parallélogramme qui permet de compléter le quatrième point d'un parallélogramme en cliquant sur les trois premiers.

Il est fait mention du calcul de produit scalaire avec d'autres logiciels (CarMetal, Geogebra) sans dire que MathGraph32 le permet de façon native. Et inutile d'être dans un repère. Il suffit que la figure possède une longueur unité. Par exemple, on crée quatre points libres, deux vecteurs en cliquant sur origine et extrémité. On clique ensuite sur l'icône mesure de produit scalaire puis sur les deux vecteurs. Une boîte de dialogue s'ouvre alors pour demander le nom du calcul représentant ce produit scalaire et c'est tout (les points doivent avoir un nom : s'ils n'en ont pas MathGraph32 leur affectera un nom automatiquement).

MathGraph32 permet aussi, à l'aide de constructions prédéfinies, de <u>mesurer les coordonnées d'un vecteur</u> dans un repère (et MathGraph32 peut gérer un nombre quelconque de repères dans une figure). Par exemple, pour mesurer les coordonnées d'un vecteur en les affichant sur la figure sous forme d'un couple, il suffit de créer un vecteur puis d'utiliser le menu Constructions - Utiliser une construction prédéfinie, d'ouvrir le dossier nommé Mesure et de choisir par exemple la construction nommée Mesure Coordonnées Vecteur Avec Affichage. Il suffit alors de cliquer sur le vecteur. On voit alors apparaître en haut et à gauche un affichage LaTeX représentant ce couple. MathGraph32 propose aussi de renommer les objets finaux numériques qui sont les deux coordonnées du vecteur et sont nommées par défaut a et b. Cette mesure pourra soit être affichée sur la figure soit être utilisée dans d'autres calculs. Dans l'article, sur le lien autre méthode, on pourrait dire que les coordonnées peuvent être mesurées automatiquement avec la construction prédéfinie nommée Mesure Coordonnées Vecteur.

A noter qu'on peut aussi mesurer les coordonnées du vecteur sans les afficher et qu'il existe des constructions analogues pour mesurer ou afficher l'affixe d'un vecteur ou d'un point (puisque MathGraph32 travaille avec les nombres complexes de façon native). Il est bien sûr possible de créer soi-même ses propres constructions pour réaliser ce type de mesure.

Yves Biton.

[1] pour la relation d'équipollence entre bipoints, vu comme ça, ça fait peur, mais en fait c'est juste une façon formelle d'exprimer les idées du programme actuel de Seconde

[2] Ceci dit, il a un langage de création de figure qu'on peut manipuler avec des boutons et il peut suivre la souris

[3] apparemment elle remonterait à Hamilton, en 1843, dans son article sur les quaternions.

[4] avec des surprises parfois : Lorsqu'on approche la souris du vecteur, GeoGebra devine si on veut saisir le vecteur, son origine ou son extrémité ; si on ne fait pas attention, l'origine est bougée ce qui a pour effet de modifier les coordonnées du vecteur ; mais il est facile de modifier celles-ci en cliquant sur "propriétés".

[5] à moins qu'on y réintroduise les barycentres...

[6] Avant ça, le vecteur était un segment fléché hérité de C.a.R. et il était difficile d'accéder à ses coordonnées.

[7] Cette fonctionnalité n'est pas nouvelle dans CaRMetal, mais avant on "devait" ajouter le caractère "underscore" à la fin de l'alias du point ; la nouveauté c'est que le signe "=" est absent par défaut.

[8] celle-ci aussi peut être rendue fixe, voire entrée au clavier, ce qui permet instantanément d'avoir un vecteur normé en entrant 1 comme longueur.

[9] qui, contrairement à GeoGebra, est toujours égale au produit des normes des vecteurs et du cosinus de leur angle, parce que, contrairement à GeoGebra, le repère de CaRMetal est toujours orthonormé.

[10] actuellement, Java, JavaScript, Python et le Basic maison, lequel est très doué pour communiquer avec le tableur (il a été fait pour ça aussi).

[11] Elle présente l'avantage que les multiplications se font de gauche à droite, et se prête donc mieux à une étude purement matricielle, au contraire de l'ordre de la composition des fonctions.