

TRACÉ DE LIGNES ET DE COURBES PLANES

J.-P. ZANOTTI

TABLE DES MATIÈRES

Avertissement	1
1. Espaces vectoriels	1
2. Espaces affines	6
3. Espaces projectifs	8
4. Transformations dans le complété projectif du plan réel	9
5. Tracé de segments de droites	11
6. Tracé d'arcs de cercles	12
7. Courbes de Bézier	13
8. Courbes de Bézier par morceaux	17
9. Courbes B -splines	19
10. Courbes B -splines uniformes	21
11. Surfaces de Bézier	21

AVERTISSEMENT

Les exercices de travaux dirigés, ou les exercices de travaux pratiques sont précédés d'un symbole qui qualifie la difficulté de l'exercice. Le \heartsuit signifie que l'exercice est une simple application des définitions et ne doit (devrait) demander que quelques minutes. Il doit *impérativement être fait* car il vous permet de vous assurer que vous avez bien saisi la notion qui a été introduite. La simple dague \dagger signifie un minimum de réflexion, tandis que la double dague \ddagger indique un exercice plus délicat (sans pour autant être infaisable). Les passages qui sont des compléments de cours ou qui ne sont pas fondamentaux sont écrits dans une police plus petite et peuvent être passés en première lecture.

Les objets que l'on manipule en infographie sont principalement, à l'échelle "atomique", des points et des vecteurs, nous allons donc faire quelques rappels d'ordre général sur les espaces vectoriels, les espaces affines et les espaces projectifs, puis nous nous focaliserons sur l'espace qui nous concerne, à savoir le plan euclidien réel $\mathbf{R} \times \mathbf{R}$.

1. ESPACES VECTORIELS

La notion récurrente dans toute cette partie du cours est celle d'espace vectoriel. Ce type de structure algébrique intervient à tout moment dans ce cours, il est donc indispensable d'en comprendre le sens.

Définition 1. Soit K un corps commutatif. On appelle K -espace vectoriel tout triplet $(E, +, \cdot)$ où $+$ est une loi de composition interne et \cdot une loi de composition externe à domaine d'opérateurs dans un corps K satisfaisant aux conditions suivantes :

- (1) $(E, +)$ est un groupe commutatif;
- (2) $\forall \lambda \in K, \forall (u, v) \in E \times E, \quad \lambda.(u + v) = (\lambda.u) + (\lambda.v)$;
- (3) $\forall (\lambda, \mu) \in K \times K, \forall u \in E, \quad (\lambda + \mu).u = (\lambda.u) + (\mu.u)$;
- (4) $\forall (\lambda, \mu) \in K \times K, \forall u \in E, \quad \lambda.(\mu.x) = (\lambda\mu).x$;
- (5) $\forall u \in E, \quad 1.u = u$.

Les éléments du corps K sont appelés des *scalaires*. On rappelle qu'une *combinaison linéaire* d'une famille $(u_i)_{i \in I}$ est le vecteur u défini par la somme

$$u := \sum_{i \in I} \alpha_i u_i$$

où la famille $(\alpha_i)_{i \in I}$ d'éléments de K est presque nulle (seul un nombre fini de scalaires α_i sont non-nuls). Notons que cette dernière condition est indispensable sans quoi la somme ci-dessus n'aurait pas de sens.

Une combinaison linéaire est une construction élémentaire et fondamentale dans les espaces vectoriels puisqu'elle fait intervenir les deux lois de composition de l'espace, l'addition de deux vecteurs et la multiplication d'un vecteur par un scalaire.

Une famille $(u_i)_{i \in I}$ d'éléments d'un espace vectoriel E est dite *libre* si aucun vecteur de la famille ne peut s'écrire comme combinaison linéaire des autres, autrement dit si

$$\sum_{i \in I} \alpha_i u_i = 0 \Rightarrow \forall i \in I, \alpha_i = 0.$$

Un *sous-espace vectoriel* d'un K -espace vectoriel E est une partie F de E stable pour les lois externes et internes induites. Une famille $(u_i)_{i \in I}$ d'éléments d'un espace vectoriel E est dite *génératrice* d'un sous-espace vectoriel F de E si tout vecteur de F peut s'exprimer comme une combinaison linéaire d'éléments de cette famille. Une famille libre et génératrice est appelée *base* du sous-espace F . Si cette base est finie, son cardinal est appelé *dimension* de l'espace vectoriel F . Cette notion de base d'un espace vectoriel est elle aussi fondamentale, puisque l'existence d'une base permet de construire *tous* les vecteurs de l'espace à l'aide de combinaisons linéaires des vecteurs de la base.

Considérons à présent une base $B = \{b_1, \dots, b_n\}$ d'un K -espace vectoriel E de dimension n et u un vecteur de E . Puisque B est une base, il existe un n -uplet $(\lambda_1, \dots, \lambda_n)$ de K^n tel que u est une combinaison linéaire des vecteurs de B avec ces scalaires : $u = \lambda_1 b_1 + \lambda_2 b_2 + \dots + \lambda_n b_n$. On appelle alors *coordonnées* de u dans la base B le n -uplet $(\lambda_1, \dots, \lambda_n)$ et on écrit

$$u : (\lambda_1, \dots, \lambda_n)_B.$$

Quand aucune ambiguïté n'est possible sur la base, on omet de faire référence à B .

Intéressons nous maintenant à la construction du plan. Dès lors que l'on dispose d'un corps commutatif, on est en mesure de construire très facilement un espace vectoriel, comme le montre l'exercice 1.

Exercice 1 ♡ Montrez que si K est un corps commutatif de loi additive $+$ et de loi multiplicative \times , l'ensemble $E = K^n$ (le produit cartésien de n copies du corps K) muni de la loi interne \oplus et de la loi externe \odot sur K suivantes est un K -espace vectoriel :

$$(u_1, \dots, u_n) \oplus (v_1, \dots, v_n) := (u_1 + v_1, \dots, u_n + v_n)$$

$$\lambda \odot (u_1, \dots, u_n) := (\lambda \times u_1, \dots, \lambda \times u_n)$$

Pour cette construction, la famille de vecteurs $\{e_1, \dots, e_n\}$, où e_i est le vecteur dont toutes les composantes sont nulles sauf la i -ème qui vaut 1, est manifestement libre. En effet

$$\sum_{i=1}^n \alpha_i e_i = (\alpha_1, \dots, \alpha_n) = 0 \iff \forall i \in \{1, \dots, n\}, \alpha_i = 0.$$

D'autre part elle est génératrice car tout vecteur $u = (u_1, \dots, u_n)$ de K^n peut s'écrire $\sum_{i=1}^n u_i e_i$. C'est donc une base de K^n appelée *base canonique*. Les coordonnées de u dans la base canonique sont

$$u : (u_1, \dots, u_n).$$

Quand $n = 2$ et $K = \mathbf{R}$, l'espace ainsi construit s'appelle le *plan réel*. Nous le noterons à l'avenir \mathcal{P} . La base canonique du plan est constituée des vecteurs $(1, 0)$ et $(0, 1)$.

Exemple 1. L'idée sous-jacente à la notion de base est très simple et nous allons observer brièvement un exemple d'espace vectoriel et de base un peu moins classique que ceux que l'on trouve en général dans les ouvrages de mathématiques. Tout infographiste sait que l'on peut coder les couleurs sur 24 bits, ou plus précisément sur 3 octets, chaque octet correspondant à l'intensité que l'on veut associer à chacune des couleurs fondamentales : le rouge le vert et le bleu (codage **RVB**). Ces 24 bits permettent de coder $2^{24} = 16.777.216$ couleurs ce qui est suffisant pour notre capacité à les distinguer. En notant E l'espace des couleurs et R, V, B les couleurs rouge, vert et bleu respectivement, la couleur **jaune** est obtenue par la *combinaison linéaire*

$$J = 255R + 255V + 0B.$$

En considérant que les scalaires vivent dans le corps fini \mathbf{F}_{2^8} (le corps contenant $2^8 = 256$ éléments), l'espace des couleurs E devient un \mathbf{F}_{2^8} -espace vectoriel de dimension 3. En effet, les trois couleurs $\{R, V, B\}$ forment une famille *génératrice* de E puisqu'elles permettent de construire toutes les autres et *libre* puisqu'aucune d'entre elles ne peut s'obtenir comme mélange (combinaison linéaire) des deux autres. La famille $\{R, V, B\}$ est donc une base de E de dimension 3. Ainsi $J : (255, 255, 0)$ dans cette base.

Quand on dispose d'ensembles muni d'une même structure algébrique, il est naturel de s'intéresser aux applications qui ne bouleversent pas cette structure, les *morphismes*. Dans la situation qui nous concerne, les morphismes d'espaces vectoriels sont généralement appelés *applications linéaires*. Soient donc E et F deux K -espaces vectoriels et f une application de E dans F . Pour que f soit une application linéaire, elle doit d'une part respecter la structure additive : pour deux vecteurs x et y , il faut que $f(x+y) = f(x) + f(y)$; d'autre part la structure multiplicative : on se donne un vecteur x et un scalaire λ et il faut que $f(\lambda x) = \lambda f(x)$.

On appelle *image* d'une application linéaire $f : E \rightarrow F$, l'image de l'espace E tout entier, i.e. $f(E) := \{f(x), x \in E\}$. On appelle *noyau* de f l'ensemble des éléments de E qui ont pour image le vecteur nul, soit $\{x \in E, f(x) = 0\}$. On note respectivement ces ensembles $\text{im}(f)$ et $\text{ker}(f)$.

Exercice 2 ♡ Soit f une application linéaire $f : E \rightarrow F$. Montrez que $\text{im}(f)$ est un sous-espace vectoriel de F et que $\text{ker}(f)$ est un sous-espace vectoriel de E .

Nous supposons que le calcul matriciel de base est acquis, rappelons simplement le lien entre le calcul matriciel et la représentation d'une application linéaire f d'un K -espace vectoriel E dans un K -espace vectoriel F .

Définition 2. Soient E et F deux K -espaces vectoriels de dimensions finies respectives $p > 0$ et $n > 0$ et de bases respectives e et f . Soit φ une application linéaire de E dans F . On appelle matrice représentative de φ dans les bases e et f la matrice (a_{ij}) déterminée par

$$\forall i \in \{1, \dots, p\}, \forall j \in \{1, \dots, n\}, \varphi(e_j) = \sum_{i=1}^p a_{ij} f_i.$$

De manière plus imagée, la i -ème colonne de la matrice est l'image par φ du vecteur e_j exprimée dans la base f ,

$$(1) \quad \begin{matrix} \varphi & e_1 & e_2 & \dots & e_p \\ f_1 & a_{11} & a_{12} & \dots & a_{1p} \\ f_2 & a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ f_n & a_{n1} & a_{n2} & \dots & a_{np} \end{matrix}$$

Les composantes du vecteur $\varphi(u)$ s'obtiennent alors simplement à l'aide du produit matriciel Mu où M désigne la matrice ci-dessus et u est représenté en colonne.

Les applications linéaires qui vont nous intéresser dans la suite seront des applications de \mathcal{P} dans lui-même, autrement dit de $\mathbf{R} \times \mathbf{R}$ dans $\mathbf{R} \times \mathbf{R}$. Le lecteur vérifiera aisément que la composition de deux applications linéaires est encore une application linéaire, cette composition se traduisant efficacement du point de vue matriciel par la multiplication des matrices correspondantes, propriété hautement intéressante du point de vue algorithmique.

Exercice 3 ♡ Soient f et g deux applications linéaires de \mathcal{P} dans \mathcal{P} . Montrez que l'application composée $h = g \circ f$ est linéaire et a pour représentation matricielle le produit des matrices M_g et M_f dans cet ordre, où M_f (resp. M_g) désigne la représentation matricielle de l'application f (resp. g).

On rappelle qu'une application linéaire est inversible (et cela n'a de sens que si les espaces d'arrivée et de départ ont même dimension) si le déterminant de sa représentation matricielle n'est pas nul. Dans le cas général en dimension n , le calcul du déterminant se fait de manière récursive : si A est une matrice carrée $n \times n$, on peut développer ce déterminant suivant les colonnes ou les lignes de la matrices, soit respectivement :

$$\det(A) := \sum_{i=1}^n a_{i,j} (-1)^{i+j} \det(A_{i,j}^-), \quad \forall j, 1 \leq j \leq n.$$

$$\det(A) := \sum_{j=1}^n a_{i,j} (-1)^{i+j} \det(A_{i,j}^-), \quad \forall i, 1 \leq i \leq n.$$

où $A_{i,j}^-$ désigne la matrice $(n-1) \times (n-1)$ obtenue en privant la matrice A de sa i -ème ligne et de sa j -ème colonne. Dans le cas particulier du plan où $n = 2$,

$$A := \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \det(A) = ad - bc.$$

L'ensemble des applications linéaires inversibles d'un espace vectoriel E dans lui-même constituent un groupe pour la composition des applications appelé *groupe linéaire* et noté $GL(E)$. L'élément neutre est l'application identique $x \mapsto x$ du plan dans lui-même (elle est bien linéaire) et la composition des applications est associative. Ce groupe n'est *pas* commutatif. Parallèlement, les matrices associées forment également un groupe pour la multiplication matricielle, ce groupe est isomorphe à $GL(E)$ auquel il s'identifiera. Dans le cas particulier où $E = \mathbf{R}^n$, ce groupe se note $GL_n(\mathbf{R})$.

Si l'on s'intéresse tant aux applications linéaires en infographie, c'est justement parce que l'on est capable de les décrire de manière très concise sous forme matricielle et que les calculs se limitent à quelques sommes et produits. La transposition de la structure matricielle dans le langage informatique est évidemment la structure de tableau.

La seule structure d'espace vectoriel nous permet d'étudier deux premières transformations élémentaires : les *translations* et les *homothéties* liées respectivement à la loi additive interne et la loi multiplicative externe.

Définition 3. Soit E un K -espace vectoriel et v un vecteur de E . On appelle translation de vecteur v , l'application $t_v : E \rightarrow E$ définie par

$$t_v(u) = u + v.$$

Définition 4. Soit E un K -espace vectoriel et λ un scalaire de K . On appelle homothétie de rapport λ , l'application $h_\lambda : E \rightarrow E$ définie par

$$h_\lambda(u) = \lambda u.$$

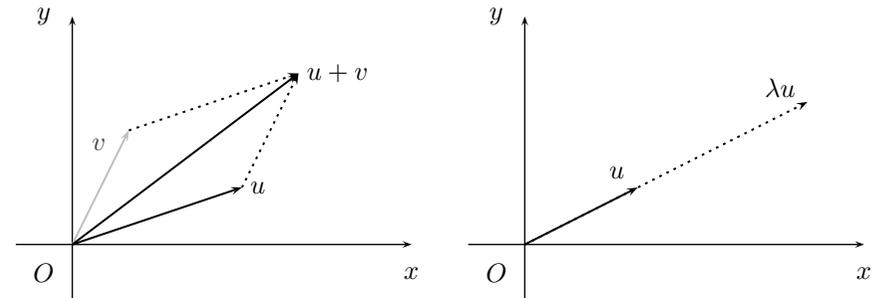


FIG. 1. À gauche : translation de vecteur v . À droite : homothétie de rapport λ

Il faut noter qu'aucune flèche n'a été utilisée "au-dessus" des vecteurs. Ceci est motivé par le fait que dans ce cours, nous exploitons principalement les propriétés algébriques de la structure d'espace vectoriel et que l'on peut se passer avec profit de la présentation des vecteurs comme on a pu le voir dans le secondaire. Notons néanmoins que la représentation des vecteurs dans le plan à l'aide de flèches est particulièrement commode pour appréhender les opérations. Ainsi, dans la base canonique $(1, 0)$, $(0, 1)$, le vecteur $u : (x, y)$ qui n'est que le point de coordonnées (x, y) se dessine comme une flèche qui part de l'origine $(0, 0)$ et se termine en (x, y) . On pourra observer dans la figure 1, comment ces flèches mettent en perspective la translation et l'homothétie.

Exercice 4 ♡ Montrez que dans un K -espace vectoriel, une translation de vecteur u n'est *pas* une application linéaire. *A contrario*, montrez que l'homothétie de rapport λ est linéaire.

La structure d'espace vectoriel est le support de base pour faire de la géométrie, mais il faut se doter d'autres outils pour manipuler des objets du plan. Deux notions sont fondamentales, la notion de distance et la notion d'angle, toutes deux obtenues grâce au *produit scalaire* :

Définition 5. On appelle produit scalaire l'application $\phi : K^n \times K^n \rightarrow K$ définie par

$$(2) \quad \phi(u, v) := u_1v_1 + u_2v_2 + \cdots + u_nv_n \quad \text{si } u = (u_1, \dots, u_n) \text{ et } v = (v_1, \dots, v_n).$$

On note souvent $(u | v)$ le nombre réel $\phi(u, v)$.

Quand l'espace ambiant est le plan réel \mathcal{P} , le produit scalaire est bien sûr défini sur $\mathcal{P} \times \mathcal{P}$ par $\phi(u, v) := u_1v_1 + u_2v_2$.

Exercice 5 ♡ Montrez que le produit scalaire est une forme bilinéaire symétrique non-dégénérée, c'est-à-dire :

- (1) $\phi(u + v, w) = \phi(u, w) + \phi(v, w)$ et $\phi(u, v + w) = \phi(u, v) + \phi(u, w)$ (bilinéarité).
- (2) $\phi(u, v) = \phi(v, u)$ (symétrie).
- (3) Soit $u \in \mathcal{P}$, si $\forall v \in \mathcal{P}$, $\phi(u, v) = 0$ alors $u = 0$. (non-dégénérescence).

Un produit scalaire permet de définir une *norme* puis une *distance* dites *induites*. Si u et v désignent deux vecteurs du plan,

$$(3) \quad \|u\| := \sqrt{(u | u)} \quad \text{et} \quad d(u, v) := \|v - u\|.$$

la norme et la distance ainsi définies sont la norme et la distance *euclidiennes*. On montrera à titre d'exercice le théorème

Théorème 6 (Inégalité de Cauchy-Schwarz). *Pour tous vecteurs u et v de \mathbf{R}^2 on a l'inégalité suivante :*

$$|(u | v)| \leq \|u\| \cdot \|v\|.$$

Exercice 6 ♡ Montrez que l'application de $\mathcal{P} \rightarrow \mathbf{R}$ définie par (3) est bien une norme, i.e. qu'elle satisfait les axiomes suivants :

- (1) $\forall u \in \mathbf{R} \times \mathbf{R}$, $\|u\| \geq 0$ et $\|u\| = 0 \Leftrightarrow u = 0$;
- (2) $\forall \lambda \in \mathbf{R}$, $\forall (u, v) \in \mathbf{R} \times \mathbf{R}$, $\|\lambda u\| = |\lambda| \|u\|$;
- (3) $\|u + v\| \leq \|u\| + \|v\|$ (inégalité triangulaire).

Pour l'inégalité triangulaire, utilisez l'inégalité de Cauchy-Schwarz. Montrez que l'application de $\mathbf{R}^2 \times \mathbf{R}^2 \rightarrow \mathbf{R}$ définie par (3) est bien une distance, i.e. qu'elle satisfait les axiomes suivants :

- (1) $\forall (u, v) \in \mathbf{R} \times \mathbf{R}$, $d(u, v) \geq 0$ et $d(u, v) = 0 \Leftrightarrow u = v$;
- (2) $\forall (u, v) \in \mathbf{R} \times \mathbf{R}$, $d(u, v) = d(v, u)$;
- (3) $\forall (u, v, w) \in \mathbf{R}^3$, $d(u, w) \leq d(u, v) + d(v, w)$.

Un vecteur est dit *normé* ou *unitaire* si $\|u\| = 1$. Deux vecteurs u et v sont *orthogonaux* si leur produit scalaire est nul, autrement dit si $(u | v) = 0$. Ils sont *orthonormés* s'ils sont orthogonaux et si chacun des vecteurs est normé. Pour toute partie A du plan \mathcal{P} , on définit l'orthogonal de la partie A comme l'ensemble des vecteurs orthogonaux à A , soit $A^\perp = \{u \in \mathcal{P}; (u | v) = 0, \forall v \in A\}$. Notons que A^\perp est un sous-espace vectoriel de \mathcal{P} même si A ne l'était pas, en effet si $(u, v) \in A^\perp \times A^\perp$, $\lambda \in \mathbf{R}$ et w est un vecteur quelconque de A , alors $(u + v | w) = (u | w) + (v | w) = 0$ et $(\lambda u | w) = \lambda(u | w) = 0$, donc A^\perp est stable pour les lois de \mathcal{P} .

Exercice 7 ♡ Soit A une partie quelconque d'un K -espace vectoriel E . Montrez que $(A^\perp)^\perp$ est le plus petit sous-espace vectoriel de E qui contient A .

On rappelle qu'un *hyperplan* est le noyau d'une forme linéaire, c'est donc un sous-espace vectoriel de codimension 1, ou encore de dimension $n - 1$ si n est la dimension de l'espace ambiant. Par exemple, si u désigne un vecteur non-nul d'un K -espace vectoriel $E = K \times \cdots \times K$, l'ensemble u^\perp est un hyperplan puisque l'application $E \rightarrow \mathbf{R}$ définie par $v \mapsto (u | v)$ est linéaire par bilinéarité du produit scalaire. Dans le cas du plan réel, l'hyperplan u^\perp est appelé *droite vectorielle*. Si $u = (a, b)$, alors les vecteurs $v = (x, y)$ orthogonaux à u satisfont

$$(4) \quad ax + by = 0.$$

Il s'agit là de l'équation d'une droite vectorielle D . Si $u = (x, y)$ est un vecteur non-nul et appartient à la droite D , i.e. s'il satisfait (4), alors tout vecteur λu qui lui est colinéaire ($\lambda \neq 0$) satisfait l'équation (4) également. N'importe lequel de ces vecteurs est appelé *vecteur directeur* de la droite D . On dit également que la droite D est *engendrée* par "son" vecteur directeur; En effet, si $u \neq 0$, $\{u\}$ forme une famille génératrice de D puisque D est l'ensemble des combinaisons linéaires de cette famille réduite à un élément. Ainsi, en notant $M(\lambda) := (x(\lambda), y(\lambda))$ le point du plan défini par

$$\begin{cases} x(\lambda) := \lambda x, \\ y(\lambda) := \lambda y. \end{cases}$$

on définit ce que l'on appelle une *représentation paramétrique* de la droite D .

Une matrice A est dite *orthogonale* si les vecteurs formés par les deux colonnes de la matrice sont orthonormés. Ceci équivaut à ce que l'application linéaire représentée par cette matrice conserve le produit scalaire, i.e.

$$(5) \quad \forall (u, v) \in \mathcal{P}, \quad (Au | Av) = (u | v).$$

Exercice 8 ♡ Démontrez qu'une matrice A est orthogonale si et seulement si l'application linéaire représentée par cette matrice conserve le produit scalaire (équation (5)).

Ces matrices sont inversibles et l'inverse d'une matrice orthogonale A est sa matrice transposée tA qui est elle-même orthogonale. D'autre part, le produit de deux matrices orthogonales est une matrice orthogonale, les matrices orthogonales forment donc un sous-groupe noté $\mathbf{O}_2(\mathbf{R})$ *non-commutatif* du groupe $\mathrm{GL}_2(\mathbf{R})$ des matrices inversibles. On peut caractériser ces matrices :

Proposition 7. *Les matrices orthogonales sont de la forme*

$$\begin{pmatrix} a & -b \\ b & a \end{pmatrix} \quad \text{ou} \quad \begin{pmatrix} a & b \\ b & -a \end{pmatrix} \quad \text{avec } a^2 + b^2 = 1.$$

Exercice 9 † Démontrez la proposition ci-dessus. Indication : montrez que tA est l'inverse de A donc à gauche et à droite. Montrez que le déterminant d'une matrice orthogonale est ± 1 . Trouvez un contre-exemple à la réciproque.

Les matrices orthogonales de déterminant 1 forment un sous-groupe *commutatif* de $\mathbf{O}_2(\mathbf{R})$ appelé *groupe spécial orthogonal* et noté $\mathbf{SO}_2(\mathbf{R})$. Les matrices de ce groupe sont appelées des matrices orthogonales directes ou *matrices de rotation* et sont de la première forme dans la proposition ci-dessus.

Nous allons à présent introduire le corps des nombres complexes \mathbf{C} et expliciter pourquoi il joue un rôle important dans la géométrie du plan réel. Il existe plusieurs constructions de \mathbf{C} , nous présentons celle qui est la plus proche de nos préoccupations. Pour construire un corps, il faut un ensemble, deux lois de compositions internes, la première, commutative, est classiquement appelée *addition*, l'autre, qui n'est pas nécessairement commutative, est appelée *multiplication*. L'ensemble que nous allons considérer est encore une fois le plan $\mathbf{R} \times \mathbf{R}$, l'addition est définie par la classique addition composantes à composantes et il reste à définir le produit de deux nombres complexes $u = (x, y)$ et $v = (x', y')$, ce qui est obtenu grâce à

$$uv := (xx' - yy', xy' + yx').$$

Cette multiplication est associative et commutative, elle a pour élément neutre $(1, 0)$ que l'on note 1. On note $i := (0, 1)$. Tout élément $u = (x, y)$ admet un inverse

$$(6) \quad u^{-1} := \frac{1}{x^2 + y^2}(x, -y).$$

Après toutes les vérifications nécessaires, le lecteur se convaincra que $\mathbf{R} \times \mathbf{R}$ muni de ces deux lois est bien un corps. Il est appelé corps des nombres *complexes* et noté \mathbf{C} .

Exercice 10 † Montrez que \mathbf{C} est un \mathbf{R} -espace vectoriel et que 1 et i forment une base.

L'exercice 10 montre que tout nombre complexe $u = (x, y)$ s'écrit $u = x.1 + y.i$, donc $u : (x, y)$ dans la base $\{1, i\}$. La première composante de ce vecteur est appelée *partie réelle* notée $\Re(u)$ et la seconde *partie imaginaire* notée $\Im(u)$. Le *module* d'un nombre complexe $u = (x, y)$ est défini à partir de la norme euclidienne sur $\mathbf{R} \times \mathbf{R}$:

$$(7) \quad |u| := \sqrt{x^2 + y^2}.$$

On appelle *conjugué* d'un nombre complexe $u = (x, y)$, le nombre $\bar{u} = (x, -y)$.

Exercice 11 † Montrez que $u\bar{u} = |u|^2$. En déduire (6).

En raison de la multiplicativité du module, i.e. $|uv| = |u||v|$, l'ensemble des nombres complexes dont le module est 1 (qui sont donc des vecteurs unitaires si l'on considère la structure de \mathbf{R} -espace vectoriel de \mathbf{C}), forme un sous-groupe du groupe multiplicatif de \mathbf{C} que l'on note \mathbf{U} et qui est appelé *groupe unitaire*.

Proposition 8. *L'application $m : \mathbf{C} \rightarrow M_2(\mathbf{R})$ définie par*

$$(8) \quad u = (a, b) \mapsto \begin{pmatrix} a & -b \\ b & a \end{pmatrix}.$$

est un morphisme de \mathbf{R} -algèbres. Elle induit un isomorphisme de groupes du groupe unitaire \mathbf{U} sur le groupe spécial orthogonal $\mathbf{SO}_2(\mathbf{R})$.

L'*exponentielle complexe* est l'application de \mathbf{C} dans \mathbf{C} définie par

$$(9) \quad \exp(u) := \sum_{n \geq 0} \frac{u^n}{n!}.$$

Cette définition a un sens car cette série entière converge avec un rayon de convergence infini (autrement dit, elle converge quelle que soit la valeur de u). On peut montrer que l'exponentielle est additive, i.e. $\exp(u + u') = \exp(u)\exp(u')$. On note e le nombre de *Neper* défini par $\exp(1)$. L'exponentielle est une fonction continue et grâce à l'additivité, on montre que pour tout nombre réel x , $\exp(x) = e^x$. Ainsi, par convention, on note e^u plutôt que $\exp(u)$ l'exponentielle du nombre complexe u . On démontre les relations suivantes :

$$\begin{aligned} - \bar{e^u} &= e^{\bar{u}}; \\ - |e^u| &= e^{\Re(u)}, \text{ i.e. si } u = x + iy, |e^{x+iy}| = e^x \text{ et en particulier } |e^{iy}| = 1. \end{aligned}$$

On définit respectivement les fonctions *cosinus* et *sinus* de la variable réelle θ comme la partie réelle et la partie imaginaire du nombre complexe $e^{i\theta}$:

$$\begin{aligned} \cos \theta &:= \Re(e^{i\theta}) = \frac{1}{2}(e^{i\theta} + e^{-i\theta}) \\ \sin \theta &:= \Im(e^{i\theta}) = \frac{1}{2i}(e^{i\theta} - e^{-i\theta}) \end{aligned}$$

Exercice 12 ♡ Calculez $e^{i\theta}$ et $e^{-i\theta}$ en fonction de $\cos \theta$ et $\sin \theta$. Montrez que $\cos^2 \theta + \sin^2 \theta = 1$.

Proposition 9. L'application $\theta \mapsto e^{i\theta}$ est un morphisme du groupe additif de \mathbf{R} dans le groupe multiplicatif de \mathbf{C}^* . L'image de ce morphisme est le groupe unitaire \mathbf{U} . Son noyau est un sous-groupe discret non-trivial de $(\mathbf{R}, +)$.

Exercice 13 † Démontrez la première partie de la proposition.

Le noyau de ce morphisme est donc l'ensemble des nombres réels θ tels que $e^{i\theta} = 1$. On déduit de la deuxième partie de la proposition que ce noyau est de la forme $\tau\mathbf{Z}$ où τ est un nombre réel > 0 . La périodicité de l'application $\theta \mapsto e^{i\theta}$ en découle et τ est sa plus petite période.

Définition 10. Le nombre π est la moitié de la période de la fonction $\theta \mapsto e^{i\theta}$.

Exercice 14 † Montrez les formules trigonométriques suivantes à partir de l'additivité de l'exponentielle :

$$\begin{aligned} \cos(a + b) &= \cos a \cos b - \sin a \sin b \\ \sin(a + b) &= \sin a \cos b + \cos a \sin b \end{aligned}$$

En déduire que l'application $R : \mathbf{R} \rightarrow \mathbf{SO}_2(\mathbf{R})$ définie par

$$R(\theta) := \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

est un morphisme surjectif de groupes périodique de période 2π .

On peut à présent définir la notion d'angle. Pour cela, on considère deux vecteurs unitaires u et v . D'après la proposition 8, il existe une unique matrice A de $\mathbf{SO}_2(\mathbf{R})$ telle que $v = Au$, on l'appelle *angle* des vecteurs u et v . D'après la proposition précédente, il existe $\theta \in \mathbf{R}$ (pas nécessairement unique), tel que $A = R(\theta)$, on dit alors que θ est une *mesure* de l'angle A . Ceci se traduit par

$$v = e^{i\theta} u.$$

On appelle donc *rotation d'angle* θ l'application $\text{rot}_\theta : \mathcal{P} \rightarrow \mathcal{P}$ définie par $u \mapsto e^{i\theta} u$, ce qui se traduit sous forme matricielle, en notant $u = (x, y)$, par

$$(10) \quad (x, y) \mapsto \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

On parlera parfois de rotation *vectorielle*.

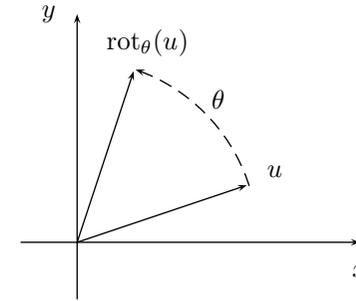


FIG. 2. Rotation d'angle θ

2. ESPACES AFFINES

Définition 11. Soit E un ensemble et V un K -espace vectoriel. Soit $\varphi : E \times E \rightarrow V$ une application satisfaisant :

- (1) $\forall (A, B) \in E \times E$ l'application $B \mapsto \varphi(A, B)$ est bijective ;
- (2) $\forall (A, B, C) \in E \times E \times E$, $\varphi(A, B) + \varphi(B, C) = \varphi(A, C)$.

Le couple (E, φ) est appelé *espace affine défini sur V* . Cette structure est généralement obtenue directement à partir d'un espace vectoriel E à l'aide de l'application $\varphi(A, B) := B - A$ et dans ce cas, il s'agit de la structure d'espace affine naturelle de E . Les éléments d'un espace affine sont appelés *des points*.

Nous travaillerons toujours dans l'espace affine naturel associé à $\mathbf{R} \times \mathbf{R}$ appelé *plan affine*. Par conséquent, les points de cet espace affine sont aussi des vecteurs, tout dépend du point de vue auquel on se place. La notation \mathcal{P} désignera donc également le plan affine. Si A et B sont deux points du plan affine, nous noterons \overrightarrow{AB} le vecteur $B - A$. La distance entre deux points est naturellement définie comme la distance entre les deux vecteurs, i.e. $d(A, B) = \|\overrightarrow{AB}\|$, que l'on notera plus souvent $|AB|$.

La première (resp. seconde) coordonnée d'un point $M = (x, y)$ dans le plan affine est appelée *abscisse* (resp. *ordonnée*). Soit Δ une droite vectorielle et $A = (x_A, y_A)$ un point du plan affine. On appelle *droite affine de direction Δ contenant A* l'ensemble des points $M = (x, y)$ de \mathcal{P} tels que $\overrightarrow{AM} \in \Delta$. Autrement dit si Δ est d'équation $ax + by = 0$, on doit avoir $a(x - x_A) + b(y - y_A) = 0$, soit

$$ax + by - (ax_A + by_A) = 0.$$

En notant que si A décrit le plan \mathcal{P} , alors $-(ax_A + by_A)$ décrit \mathbf{R} tout entier, l'équation générale d'une droite affine est obtenue pour trois scalaires a, b et c par

$$ax + by + c = 0.$$

Quand on parle de droite sans autre précision, c'est toujours d'une droite affine dont il est question.

Définition 12. Soient A et B deux points du plan. On appelle segment AB l'ensemble défini par

$$[A, B] := \{(1 - u)A + uB, u \in [0, 1]\}.$$

Exercice 15 ♡ Montrez qu'une droite affine est la translattée d'une droite vectorielle. Montrez qu'il existe une unique droite qui passe par deux points distincts A et B . On parle alors de la droite AB que l'on note (AB) .

Deux droites sont dites *parallèles* si elles ont mêmes directions. Deux droites sont dites *orthogonales* ou *perpendiculaires* si leurs directions sont orthogonales.

Exercice 16 † Montrez que toute droite est parallèle à elle-même. Montrez que pour que deux droites distinctes soient parallèles il faut et il suffit qu'elles n'aient aucun point commun. Montrez que deux droites qui ne sont pas parallèles ont un unique point commun.

Soit D une droite et M un point du plan \mathcal{P} . La droite orthogonale à D et qui passe par M rencontre la droite D en un point H (qui peut être confondu avec A si $A \in D$).

Proposition 13. Le point H est l'unique point de D le plus proche de M , autrement dit le seul point qui réalise le minimum de la fonction $P \mapsto |MP|$.

Ce point H est appelé le *projeté orthogonal* de M sur D . L'application $M \mapsto H$ est appelé *projection orthogonale du plan sur la droite D* .

Exercice 17 ♡ Montrez le théorème de Pythagore, à savoir, que pour que deux droites AB et AC soient orthogonales, il faut et il suffit que

$$AB^2 + AC^2 = BC^2.$$

Dans ce cas, on dit que le triangle ABC est rectangle en A .

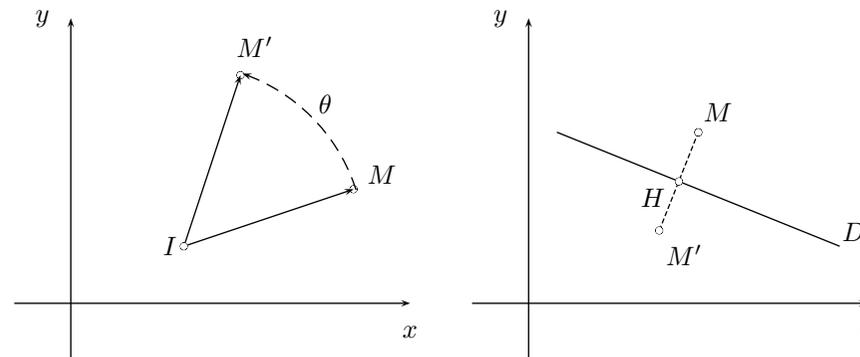


FIG. 3. À gauche : rotation de centre I et d'angle θ . À droite : symétrie orthogonale par rapport à la droite D .

La translation d'un point de l'espace affine est définie exactement de la même manière que dans un espace vectoriel, par contre on peut définir la rotation autour d'un point du plan affine, la rotation *vectorielle* que nous avons étudiée plus haut consistant à tourner autour de l'origine :

Définition 14. On appelle rotation d'angle θ autour d'un point I de \mathcal{P} , l'application $\text{rot}_{(I,\theta)} : \mathcal{P} \rightarrow \mathcal{P}$ qui associe à un point M l'unique point M' tel que

$$(11) \quad \overrightarrow{IM'} = \text{rot}_\theta(\overrightarrow{IM}).$$

où rot_θ désigne la rotation vectorielle d'angle θ . Quand $\theta = \pi$, la rotation $\text{rot}_{(I,\pi)}$ s'appelle symétrie de centre I .

Soit H la projection orthogonale d'un point M sur une droite D et M' le point défini par $\overrightarrow{MM'} = 2\overrightarrow{MH}$. L'application $\text{sym}_D : \mathcal{P} \rightarrow \mathcal{P}$ définie par $M \mapsto M'$ est appelée *symétrie orthogonale* par rapport à la droite D , ou encore symétrie d'axe D .

Exercice 18 ‡ Démontrez la proposition 13. Montrez que les translations et les rotations conservent les angles et les distances. Montrez qu'une symétrie orthogonale conserve les distances et change les angles en leurs opposés. Quels sont les points fixes d'une symétrie orthogonale ?

Exercice 19 † On appelle *milieu* d'un segment $[A, B]$, l'unique point du plan I tel que $\overrightarrow{IA} + \overrightarrow{IB} = 0$. Il vérifie $\overrightarrow{AB} = 2\overrightarrow{AI}$. Supposons que $A \neq B$. Montrez que l'ensemble des points du plan qui sont à égale distance entre A et B est la droite orthogonale à (AB) qui passe par le milieu du segment $[A, B]$. On l'appelle la *médiatrice* du segment AB .

Exercice 20 ‡ Démontrez que les isométries du plan, i.e. les applications de \mathcal{P} dans \mathcal{P} qui conservent les distances ($|f(A)f(B)| = |AB|$) sont le produit (composition) de 0, 1, 2 ou 3 symétries orthogonales.

Définition 15. Considérons une famille M_1, \dots, M_n de n points du plan et une famille $\lambda_1, \dots, \lambda_n$ de n nombres réels. Soit Λ la somme de ces n nombres. On appelle barycentre de la famille de points M_i le point G défini par

$$(12) \quad G := \frac{1}{\Lambda} \sum_{i=1}^n \lambda_i M_i.$$

Les scalaires λ_i sont appelés poids ou coefficients de pondération du système de points. On dit que la pondération est normalisée quand $\Lambda = 1$.

Exercice 21 ♡ Montrez que le barycentre d'un système de points est l'unique point qui satisfait

$$\sum_{i=1}^n \lambda_i \overrightarrow{GM_i} = 0.$$

3. ESPACES PROJECTIFS

Une des motivations pour la construction des espaces projectifs est de trouver un cadre théorique adapté à l'étude des perspectives. Tout étudiant a appris ici même que deux droites parallèles ne se coupent pas, alors que l'architecte sait pertinemment qu'elles se coupent à l'infini, ce que le dessin en perspective montre bien. Alors quel résultat est vrai? En fait, les deux. Tout dépend de la géométrie que l'on considère. L'utilisation des espaces projectifs en infographie 2D est motivée essentiellement par la simplification des transformations planes. Plus précisément, la translation, qui n'est pas une transformation linéaire dans le plan, le devient dans le plan projectif. Ce résultat, apparemment anodin a pourtant une conséquence fort intéressante pour l'infographiste, la seule transformation élémentaire du plan qui échappait à une écriture matricielle peut enfin être composée de manière efficace dans les programmes. Cependant, l'étude des perspectives étant largement abordée par les infographistes 3D, il nous a semblé qu'une brève introduction aux espaces projectifs n'était pas inutile dans le cadre de ce cours "2D" !

Si deux droites parallèles dans le plan affine ne se coupent pas, on remarque néanmoins qu'elles ont un point commun (dans tous les sens du terme!), leur direction. L'idée maîtresse pour réaliser l'intersection "à l'infini" de deux droites parallèles est d'intégrer intelligemment cette composante directionnelle dans l'écriture des points.

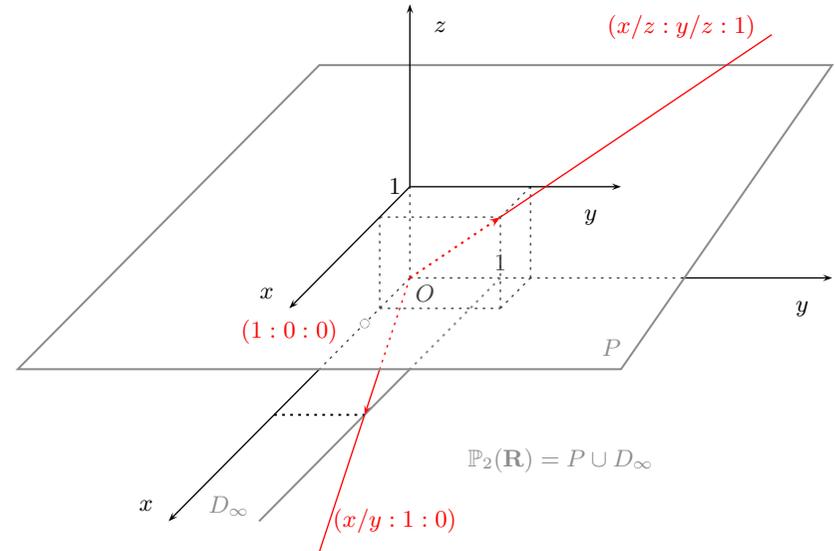


FIG. 4. Le plan projectif $\mathbb{P}_2(\mathbf{R})$ (en gris).

Définition 16. Soit E un K -espace vectoriel. On appelle espace projectif associé à E , l'ensemble des classes d'équivalence de $E^* := E \setminus \{0\}$ pour la relation de colinéarité

$$v \mathcal{R} u \Leftrightarrow \exists \lambda \in K, v = \lambda u.$$

On note $\mathbb{P}(E)$ cet espace et $\mathbb{P}_n(\mathbf{R})$ l'espace $\mathbb{P}(\mathbf{R}^{n+1})$.

Si E est un espace vectoriel de dimension finie $n + 1$ et de base $e = \{e_1, \dots, e_{n+1}\}$ (typiquement $E = \mathbf{R}^{n+1}$), alors pour tout élément $\chi \in \mathbb{P}(E)$, il existe au moins un élément $x : (x_1, x_2, \dots, x_{n+1})$ de E dont la classe d'équivalence soit χ . On dit alors que $x : (x_1, x_2, \dots, x_{n+1})$ est un système de coordonnées homogènes de χ relativement à la base e . Pour distinguer un système de coordonnées du $(n + 1)$ -uplet, on note plutôt $(x_1 : x_2 : \dots : x_{n+1})$. Réciproquement, un $(n + 1)$ -uplet $x = (x_1, x_2, \dots, x_{n+1})$ de K^{n+1} définit un unique point χ de $\mathbb{P}(E)$ qui admet x comme système de coordonnées homogènes.

Exercice 22 ♡ Montrez que la relation de colinéarité est bien une relation d'équivalence.

Nous allons nous intéresser plus particulièrement à la construction du *plan projectif*, i.e. à l'espace $\mathbb{P}_2(\mathbf{R})$ qui est l'ensemble des classes d'équivalence de $\mathbf{R}^3 \setminus \{0\}$ pour la relation de colinéarité. Soit $u = (x, y, z)$ un vecteur de \mathbf{R}^3 . Sa classe d'équivalence est constituée de tous les vecteurs de la forme $(\lambda x, \lambda y, \lambda z)$ et si l'on suppose que $z \neq 0$, on peut choisir $\lambda = 1/z$ et donc $(x/z, y/z, 1)$ comme représentant. Ainsi, quand (x, y) décrit \mathbf{R}^2 , le vecteur $(x/z, y/z, 1)$ décrit le plan affine P de \mathbf{R}^3 d'équation $z = 1$ que l'on peut naturellement identifier au plan réel $\mathbf{R} \times \mathbf{R}$ (plan en gris dans la figure 4).

Il reste à considérer les classes des vecteurs $u = (x, y, 0)$. Les vecteurs colinéaires à $(x, y, 0)$ sont de la forme $(\lambda x, \lambda y, 0)$ et si $y \neq 0$, en posant $\lambda = 1/y$, le point $(x/y, 1, 0)$ est un représentant de la classe. Ainsi quand x décrit \mathbf{R} , le vecteur $(x/y, 1, 0)$ décrit la droite affine de \mathbf{R}^3 d'équation $z = 0, y = 1$. Pour conclure cette construction, il ne reste plus qu'à déterminer la classe des vecteurs du type $(x, 0, 0)$. Par construction, on a exclu le vecteur nul de l'espace projectif, donc $x \neq 0$, ainsi $(1, 0, 0)$ représente la classe de $(x, 0, 0)$.

La réunion des classes $(x/y, 1, 0)$ et $(1, 0, 0)$ est appelée *droite à l'infini* et notée D_∞ qui n'est rien d'autre que $\mathbb{P}_1(\mathbf{R})$.

On peut donc résumer tout ceci en disant que $\mathbb{P}_2(\mathbf{R}) = P \cup D_\infty$ et de manière plus générale, on pourrait montrer que $\mathbb{P}_n(K) = K^n \cup \mathbb{P}_{n-1}(K)$. Dans le cas, $\mathbb{P}_{n-1}(K)$ s'appelle *l'hyperplan à l'infini*. Notons que le choix de $z = 1$ et $y = 1$ pour cette représentation est parfaitement arbitraire et nous aurions pu faire cette construction avec toute autre représentation. Observons à présent en quoi cette nouvelle géométrie résout le problème de l'intersection de droites parallèles, mais pour cela il faut avant tout définir ce qu'est une droite dans le cadre du plan projectif. Soient a, b et c trois scalaires. Considérons le plan vectoriel de \mathbf{R}^3 d'équation

$$(13) \quad ax + by + cz = 0.$$

Ce plan coupe le plan P sur une droite affine D d'équation

$$ax + by + c = 0, \quad z = 1$$

puisque le plan P a pour équation $z = 1$. On définit donc une *droite projective* comme l'ensemble des classes d'équivalence des points du plan vectoriel de \mathbf{R}^3 d'équation (13). Si l'on observe la figure 5, les droites projectives D et D' sont les deux plans qui coupent le plan P en deux droites affines parallèles. Ces deux plans ont pour intersection une droite vectorielle qui coupe le plan projectif $\mathbb{P}_2(\mathbf{R})$ sur la droite à l'infini D_∞ . Ainsi, deux droites projectives parallèles se coupent à l'infini. Il faut bien noter que ceci est une *modélisation* d'une idée, ainsi les "droites" D et D' se coupent à l'infini parce que le vocabulaire a été bien choisi. Pour conclure ce paragraphe, on peut donc étudier les objets affines à travers leurs homologues dans un espace projectif. Ainsi le plan projectif $\mathbb{P}_2(\mathbf{R})$ est appelé le *complété projectif* du plan affine \mathcal{P} .

Pour une droite affine D d'équation $ax + by + c = 0$, son complété projectif est la droite projective d'équation (13). Pour la construction de la partie plane P de $\mathbb{P}_2(\mathbf{R})$, nous avons supposé que $z \neq 0$. Dans ce cas on dit que l'on a *homogénéisé* la droite D

en z . Si nous avons homogénéisé en y , la droite projective associée à D aurait eu pour équation $ax + cy + bz = 0$. Ce choix est parfaitement arbitraire.

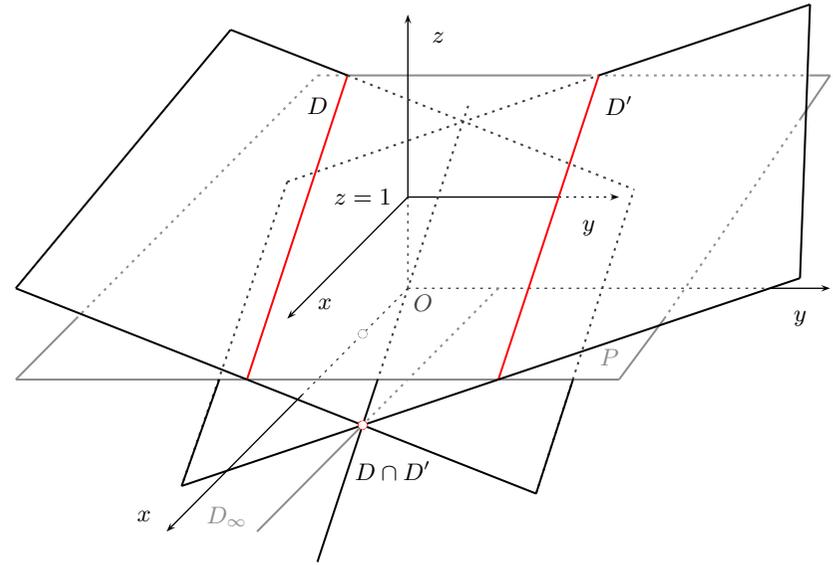


FIG. 5. Intersection de deux droites parallèles dans le plan projectif.

4. TRANSFORMATIONS DANS LE COMPLÉTÉ PROJECTIF DU PLAN RÉEL

Les transformations dans le plan réel désignent des applications du plan dans lui-même et sont évidemment fondamentales pour l'infographie. Nous avons déjà rencontré plusieurs de ces transformations : les homothéties, les translations, les rotations. En voici l'écriture matricielle dans l'espace projectif :

– homothétie de rapport λ :

$$\begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

– homothétie différentielle de rapports (λ_x, λ_y) :

$$\begin{pmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

– translation de vecteur (x_t, y_t) :

$$\begin{pmatrix} 1 & 0 & x_t \\ 0 & 1 & y_t \\ 0 & 0 & 1 \end{pmatrix}$$

– rotation d'angle θ :

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

– cisaillement de rapport λ le long des axes x (resp. y) :

$$\begin{pmatrix} 1 & \lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ \lambda & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

– symétrie par rapport à l'axe des x (resp. y) :

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Exercice 23 † Calculez la représentation matricielle en coordonnées homogènes d'une symétrie orthogonale par rapport à une droite d'équation $ax + by + c = 0$. Même question pour une rotation d'angle θ et de centre $I = (a, b)$.

Exercice 24 † La représentation d'une courbe sur un écran nécessite quelques transformations à partir de sa représentation dans le plan réel \mathcal{P} . Cette opération se fait en deux étapes : la première est une *normalisation*, qui consiste à comprimer et/ou à dilater une portion du plan réel, la *fenêtre d'observation*, dans l'*espace normalisé*, c'est-à-dire le carré unité $[0, 1] \times [0, 1]$. Cette normalisation est la composée d'une translation T , qui ramène la fenêtre d'observation de la courbe à l'origine O du repère, et d'une homothétie différentielle H (i.e. une homothétie dont le rapport est différent en x et en y). Cette fenêtre d'observation est délimitée par son coin inférieur gauche (x_{\min}, y_{\min}) et son coin supérieur droit (x_{\max}, y_{\max}) . La deuxième opération est la transformation de la courbe normalisée vers une *fenêtre écran* de hauteur MaxY et de longueur MaxX . Cette deuxième transformation est la composition d'une symétrie S par rapport à la droite d'équation $y = 1/2$, puis d'une seconde homothétie différentielle H' qui renvoie la courbe dans la fenêtre écran (figure 6).

- (1) Écrivez les matrices représentant les transformations T, H, S et H' dans l'espace projectif.
- (2) Calculez la composition $H' \circ S \circ H \circ T$ de ces transformations.
- (3) Concluez que le point (x, y) aura pour coordonnées

$$\begin{pmatrix} \frac{\text{MaxX}}{x_{\max} - x_{\min}}(x - x_{\min}) \\ \frac{\text{MaxY}}{y_{\max} - y_{\min}}(y_{\max} - y) \end{pmatrix}$$

dans la fenêtre écran.

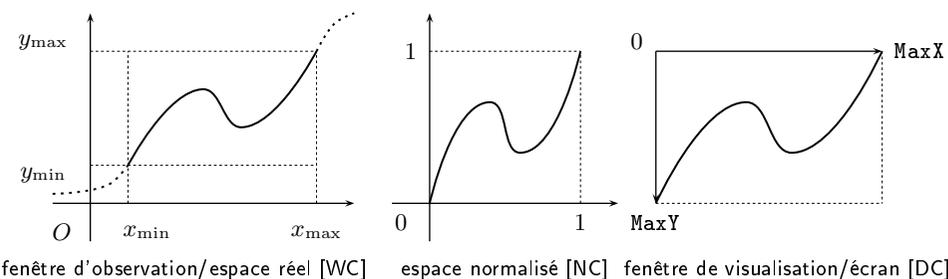


FIG. 6. Représentation sur une fenêtre écran d'une courbe plane. [WC=World Coordinates, NC=Normalized Coordinates, DC=Display Coordinates]

Remarque : si les rapports hauteur/largeur de la fenêtre d'observation et de la fenêtre écran sont distincts, la courbe sera bien entendu déformée.

Travaux Pratiques 1 ♡ Écrivez un programme `poly` en `C` dont les paramètres de la ligne de commande sont : `xmin xmax nb a0 a1 ... an` et qui affiche `nb` points de la courbe polynomiale définie par le polynôme $P(X) = a_0X + a_1X + \dots + a_nX^n$. Ces `nb` points (x, y) sont obtenus en évaluant les `nb` abscisses uniformément réparties entre `xmin` et `xmax`. Le programme doit renvoyer sur la sortie standard `nb` lignes composées de deux nombres de type `float` séparés par un espace, ces deux nombres représentant le couple $(x, p(x))$. Pour l'évaluation d'un réel x , on utilise la factorisation de *Hörner* du polynôme $P(X)$:

$$\begin{aligned} P(X) &= a_0 + a_1X + a_2X^2 + \dots + a_nX^n \\ &= a_0 + X(a_1 + X(a_2 + X(a_3 + \dots + X(a_{n-1} + Xa_n) \dots))) \end{aligned}$$

Cette évaluation est implantée en `C` de la manière suivante :

```
float eval_poly(float *P, float x, unsigned char n)
{unsigned char i;
 float y;

 y = 0;
 for (i = 0; i <= n; i++)
     y = x * y + P[n - i];
 return y;
}
```

Travaux Pratiques 2 ♥ Écrivez un programme qui lit les coordonnées de N points du plan dans un fichier texte (chaque ligne du fichier contient l'abscisse x et l'ordonnée y d'un point séparés par un espace) et qui trace ces points dans une fenêtre écran dont vous fixerez la taille. Vous déterminerez automatiquement les coordonnées de la fenêtre d'observation en calculant la plus petite et la plus grande abscisse ainsi que la plus petite et la plus grande ordonnée. Modifiez votre programme pour qu'il propose de sauvegarder dans un fichier les coordonnées normalisées de ces points. Utilisez les résultats de l'exercice 24 et le TP 1.

5. TRACÉ DE SEGMENTS DE DROITES

Nous allons étudier un algorithme très simple et très commun pour tracer un segment de droite $[AB]$, l'algorithme de *Bresenham*. On suppose que A et B sont des points de la fenêtre écran dans laquelle on veut faire le tracé, autrement dit les coordonnées de deux pixels. Néanmoins il sera nécessaire de mettre ce segment en équation, leurs coordonnées respectives (x_A, y_A) et (x_B, y_B) sont donc supposées être le centre des pixels A et B quand on quadrille le plan. D'autre part, nous allons supposer que ce segment appartient au premier octant du plan euclidien centré en A , autrement dit la première part de "pizza" dans l'ordre trigonométrique quand on coupe le plan en 8 part égales autour de A . On supposera pour terminer que $x_A < x_B$. Dans toute la suite, y compris la section suivante sur le tracé de cercles, un pixel est une surface carrée du plan euclidien de coté 1 définie par les coordonnées de son centre.

L'algorithme de Bresenham a trois qualités remarquables en algorithmique géométrique :

- (1) Il est *incrémental*, c'est-à-dire qu'à chaque étape les nouvelles positions sont obtenues à partir de la dernière position calculée. Le tracé se fait en incrémentant un compteur qui décrit une des deux coordonnées du pixel à traiter, le calcul de l'autre coordonnée étant justement l'objet de l'algorithme;
- (2) Les données sont de type *entier*, il n'y a pas de calculs coûteux d'arrondis;
- (3) Les opérations arithmétiques se limitent à des additions et des soustractions, ce qui réduit encore le coût algorithmique puisqu'aucun produit/division n'est utilisé.

On se limite au premier octant pour deux raisons : la première est que l'on peut obtenir un tracé dans n'importe quel autre octant avec des symétries très simples (voir l'exercice 25), la seconde est que l'algorithme de Bresenham repose sur l'hypothèse que la pente de la droite est strictement inférieure à 100%, (i.e. $\pi/4$) pour obtenir le résultat suivant :

Lemme 17. *Soit M un point quelconque du segment $[A, B]$ (dans le premier octant) d'abscisse entière p . Si q désigne l'ordonnée du pixel $P = (p, q)$ qui contient M , alors $[A, B]$ traverse le pixel $L := (p + 1, q)$ ou le pixel $H := (p + 1, q + 1)$ (ou inclusif).*

Démonstration. Utiliser l'hypothèse du premier octant. □

L'algorithme de Bresenham s'appuie sur le résultat du lemme 17 et consiste donc "simplement" à incrémenter une abscisse x initialisée à x_A et à sélectionner l'un des deux pixels L ou H . Le critère qui va déterminer le choix est le suivant : on calcule le point d'intersection I entre le segment $[AB]$ et le segment $[LH]$; on choisit le pixel L si la distance $d(I, L)$ est inférieure à la distance $d(I, H)$ et le pixel H dans le cas contraire. Il reste à présent à triturer les expressions de ces distances pour simplifier le critère qui permettra de prendre la décision.

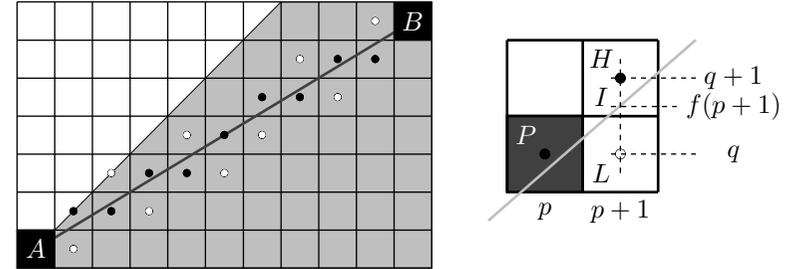


FIG. 7. Illustration de l'algorithme de Bresenham. Le premier octant est matérialisé en gris, les centres des pixels L et H par des disques, le noir désignant le pixel choisi par l'algorithme.

On désigne par f la fonction de \mathbf{R} dans \mathbf{R} telle que la droite (AB) soit définie par l'équation cartésienne $y = f(x)$. On définit alors les deux quantités dont le quotient est la pente de cette droite :

$$(14) \quad \Delta_x := x_B - x_A, \quad \Delta_y := y_B - y_A.$$

Si $P := (p, q)$ est le dernier pixel choisi par l'algorithme (initialement $P := A$), le prochain a $p + 1$ pour abscisse et donc $f(p + 1)$ pour ordonnée. Le signe de la longueur $|HI| - |IL|$ permet de déterminer quel est le pixel à allumer, L s'il est positif et H s'il est négatif. On a $|HI| - |IL| = 2(q - f(p + 1)) + 1$ et comme $f(p + 1) = f(p) + \Delta_y/\Delta_x$, on obtient

$$|HI| - |IL| = 2(q - f(p) - \Delta_y/\Delta_x) + 1$$

On peut multiplier l'inégalité $|HI| - |IL| > 0$ par Δ_x qui est positif pour obtenir le test

$$(15) \quad 2\Delta_x(q - f(p)) + (\Delta_x - 2\Delta_y) > 0.$$

Les quantités $2\Delta_x$ et $\Delta_x - 2\Delta_y$ sont évidemment calculées une fois pour toute à l'initialisation de l'algorithme et le test (15) devient très simple. On pourrait donc en rester là, mais il reste une multiplication que l'on peut encore éviter. Notons R_p la partie gauche de cette inégalité. À l'initialisation du processus, $q = f(p)$ et donc $R_0 = \Delta_x - 2\Delta_y$. Le

premier test se limite donc à déterminer le signe de $\Delta_x - 2\Delta_y$. La quantité R_{p+1} vaut respectivement :

$$R_{p+1} = \begin{cases} 2\Delta_x(q+1 - f(p+1)) + \Delta_x - 2\Delta_y, & \text{si on a allumé } H. \\ 2\Delta_x(q - f(p+1)) + \Delta_x - 2\Delta_y, & \text{si on allumé } L. \end{cases}$$

Plutôt que de recalculer R_{p+1} ex-nihilo, nous allons calculer la différence $R_{p+1} - R_p$ qui va s'avérer très simple :

$$R_{p+1} - R_p = \begin{cases} 2(\Delta_x - \Delta_y), & \text{si on a allumé } H. \\ -2\Delta_y, & \text{si on a allumé } L. \end{cases}$$

L'algorithme devient très simple à présent, on initialise le processus en calculant le signe de $R_0 = \Delta_x - 2\Delta_y$ et selon le résultat, on calcule R_1 en ajoutant l'une des deux quantités précalculées.

Exercice 25 ♡ Trouvez des critères simples (au sens du coût algorithmique) pour déterminer dans quel octant se trouve un point (x, y) du plan.

Exercice 26 ♡ Écrivez l'algorithme de Bresenham sans qu'aucun produit ou rapport ne soit calculé (la multiplication par 2 est négligée car elle correspond à un shift des bits du registre associé d'un cran vers la droite). Déclarez une variable R qui jouera le rôle de R_p ci-dessus initialisée à $\Delta_x - 2\Delta_y$.

6. TRACÉ D'ARCS DE CERCLES

Dans le même esprit que Bresenham, nous trouvons l'algorithme de Michener pour le tracé de cercles. Nous supposons que le cercle est centré en l'origine dans le repère de la fenêtre écran, une simple translation permettant de centrer le cercle en tout autre point du plan. De même le rayon r du cercle est relatif à un nombre de pixels. Avant d'étudier cet algorithme, notons qu'il est possible de réaliser un tracé avec un coût algorithmique réduit en utilisant la fonction de rotation (10) sous forme matricielle qui ne nécessite qu'un seul calcul de cosinus et de sinus. On évitera bien entendu de décrire un cercle de rayon r à l'aide de fonctions paramétrées $r \cos \theta$ et $r \sin \theta$.

Le tracé se fait dans le sens des aiguille d'une montre et commence par le pixel au sommet du cercle. En supposant que le tracé se limite au second octant, on peut montrer (cf. lemme 18) que lors du tracé, le prochain pixel à allumer est soit le pixel juste à droite du pixel courant (incrément de l'abscisse uniquement) soit le pixel au sud-est (incrément de l'abscisse et décrétement de l'ordonnée). Autrement dit, on incrémente systématiquement l'abscisse, le décrétement ou non de l'ordonnée étant laissé à la charge de l'algorithme.

Le but du lemme suivant est donc de montrer que dans le second octant, le cercle de centre O et de rayon r traverse au plus deux pixels contigus de même abscisse p .

Exercice 27 ♠ Soit $P = (p, q)$ un pixel du second octant et C un cercle de centre O et de rayon r . Quelle est l'abscisse p_{\max} la plus grande telle que la droite d'équation $x = p_{\max}$ coupe C dans le deuxième octant ?

Lemme 18. On se fixe une abscisse entière $0 \leq p \leq p_{\max}$ et on note $P = (p, q)$ le pixel le plus proche du cercle de centre O et de rayon r . Alors le pixel d'abscisse $p+1$ le plus proche du cercle est le pixel $L := (p+1, q)$ ou le pixel $H := (p+1, q-1)$.

Démonstration. Il suffit de montrer que H est à l'extérieur du cercle et L à l'intérieur, autrement dit que le cercle passe entre les points H et L . On le montre avec une récurrence sur p . C'est évidemment vrai pour $p = 0$ (dans ce cas, $q = r$). Soit y_p (resp. y_{p+1}) l'ordonnée du point du cercle d'abscisse p (resp. $p+1$). La tangente en tout point du cercle sur le second octant a une pente inférieure à 100%, le cercle étant sous la tangente, on a donc $y_p - y_{p+1} \leq 1$, autrement dit que quand on se déplace sur le cercle d'une unité vers la droite, on descend d'au plus d'une unité, le résultat souhaité s'en déduit directement. □

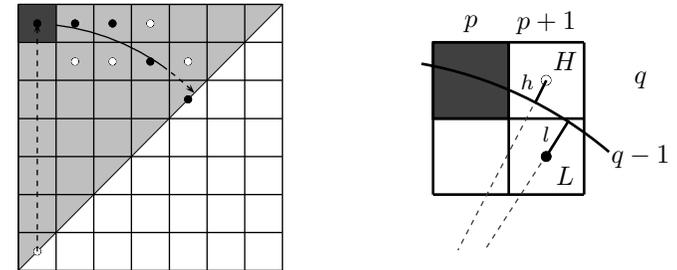


FIG. 8. Illustration de l'algorithme de Michener. Le second octant est matérialisé en gris, les centres des pixels par des cercles, le noir désignant le pixel choisi par l'algorithme.

Logiquement le choix devrait se faire en comparant les distances h et l des pixels H et L par rapport au cercle (voir figure (8)), mais nous allons voir qu'il est plus avantageux en terme de calculs de comparer les quantités $\bar{h} := |OH|^2 - r^2$ et $\bar{l} := r^2 - |OL|^2$.

Lemme 19.

$$\bar{h} > \bar{l} \iff h > l.$$

Démonstration. On a

$$\begin{aligned}(h+r)^2 &= |OH|^2 = q^2 + p^2 \\ (r-l)^2 &= |OL|^2 = (q-1)^2 + p^2.\end{aligned}$$

En sommant les deux quantités :

$$\begin{aligned}(h+r)^2 + (r-l)^2 &= |OH|^2 + |OL|^2 \\ &= \bar{h} + r^2 + r^2 - \bar{l}.\end{aligned}$$

Et en simplifiant les deux expressions, on obtient :

$$\bar{h} - \bar{l} = h^2 + l^2 + 2r(h-l).$$

Il est donc évident que $h-l \geq 0$ entraîne $\bar{h} - \bar{l} \geq 0$, mais dans quelle mesure peut-on avoir $h-l < 0$ avec $\bar{h} - \bar{l} \geq 0$? Si cette situation est possible, cela entraînerait une erreur de diagnostic sur la position effective du cercle par rapport aux deux pixels H et L . Notons $f(h, l) := h^2 + l^2 + 2r(h-l)$. Un rapide calcul montre que

$$f(h, l) = (h+r)^2 + (l-r)^2 - 2r^2,$$

 et la portion du plan défini par les points (h, l) satisfaisant l'inégalité $f(h, l) > 0$ est donc l'extérieur du cercle de centre $(-r, r)$ et de rayon $r\sqrt{2}$. Il faut donc à présent chercher l'intersection du demi-plan défini par $h < l$ et l'extérieur de ce cercle pour trouver les points (h, l) qui peuvent poser un problème dans le test.

7. COURBES DE BÉZIER

Les courbes et les surfaces dites de *Bézier* ont été développées par P. Bézier dans les années 70 pour la création d'éléments de surface chez Renault pour tracer la "peau" des véhicules, c'est-à-dire la carrosserie. Le principe est relativement simple, on se donne un ensemble de $n+1$ points du plan appelés *points de contrôle* et on cherche à construire une courbe plane qui soit la plus proche de ces points et inscrite dans l'enveloppe convexe de ces $n+1$ points. L'idée consiste à construire une famille de points qui sont les barycentres de notre famille de $n+1$ points de contrôle. Mais avant de définir cette courbe, revenons sur quelques éléments fondamentaux :

Comment allons-nous représenter une courbe plane? Si on fait le choix classique d'une représentation cartésienne, i.e. les points $M = (x, y)$ de la courbe sont définis par une équation du type

$$(16) \quad y = f(x),$$

alors nous sommes confrontés à un problème insurmontable du point de vue de l'infographie, il est impossible de dessiner toutes les courbes! En effet, un simple cercle ne peut pas être représenté de cette manière. La raison est extrêmement simple, f étant une fonction, il n'est pas possible de lui associer deux images. C'est pourtant ce que l'on

voudrait pouvoir faire et même au-delà. On pourrait alors utiliser une représentation *implicite* de la courbe, en considérant les points $M = (x, y)$ solutions de l'équation

$$(17) \quad f(x, y) = 0,$$

mais cela sous-entend que pour un x fixé (ou symétriquement pour un y) il faut résoudre l'équation (17) en y (ou x) ce qui n'est pas toujours possible de manière algébrique si l'expression de f est trop compliquée. En revanche, cette question est immédiatement réglée en faisant le choix d'une représentation paramétrique des points de la courbe :

$$(18) \quad M(u) = \begin{pmatrix} X(u) \\ Y(u) \end{pmatrix}, \quad u \in I.$$

On a donc une application $M : I \subseteq \mathbf{R} \rightarrow \mathbf{R} \times \mathbf{R}$. Ainsi, le cercle de rayon r et de centre $(0, 0)$ a pour représentation paramétrique :

$$(19) \quad M(u) = \begin{pmatrix} r \cos u \\ r \sin u \end{pmatrix}, \quad u \in [0, 2\pi].$$

Un autre avantage de cette représentation est que l'on peut définir naturellement une *orientation* de la courbe. En effet, l'application M est définie sur un intervalle I de \mathbf{R} et il est légitime de considérer que l'orientation de la courbe est obtenue en faisant croître le paramètre u dans cet intervalle. Autrement dit, on considère que le point $M(u)$ "précède" le point $M(v)$ si et seulement si $u \leq v$, ce que l'on note $M(u) \prec M(v)$.

Introduisons maintenant une famille de polynômes qui va servir à définir les courbes de Bézier, les *polynômes de Bernstein* :

Définition 20. Soit n un entier strictement positif. On appelle p -ème polynôme de Bernstein d'ordre n le polynôme de $\mathbf{R}[T]$ défini par :

$$B_{n,p}(T) := \binom{n}{p} T^p (1-T)^{n-p}$$

où $\binom{n}{p}$ est le p -ème coefficient binomial, i.e. $n!/(p!(n-p)!)$.

Il faut noter qu'un polynôme n'est pas une application! On utilise en fait la *fonction polynomiale* associée au polynôme de Bernstein pour laquelle on donnera le domaine de définition.

Proposition 21. Les polynômes de Bernstein satisfont l'égalité suivante :

$$(20) \quad \sum_{p=0}^n B_{n,p}(u) = 1 \quad \forall u \in [0, 1].$$

Preuve. La formule du binôme de Newton nous donne pour tout $0 \leq p \leq n$:

$$(T + (1-T))^n = 1 = \sum_{p=0}^n \binom{n}{p} T^p (1-T)^{n-p} = \sum_{p=0}^n B_{n,p}(T).$$

□

On peut alors construire une courbe \mathcal{C} dont les points $M(u)$ sont les barycentres des points de contrôle M_p , $0 \leq p \leq n$ affectés respectivement des coefficients $B_{n,p}(u)$, $0 \leq p \leq n$, u variant dans l'intervalle $[0, 1]$ (l'égalité (20) évite simplement de normaliser la pondération puisque la somme des coefficients est 1). Notons que toute permutation non-triviale des points de contrôle aboutit à des barycentres différents.

Définition 22. On appelle courbe de Bézier de degré n , de points de contrôle M_p , $0 \leq p \leq n$, la courbe paramétrique définie par les points suivants :

$$(21) \quad M(u) = \sum_{p=0}^n B_{n,p}(u)M_p, \quad u \in [0, 1].$$

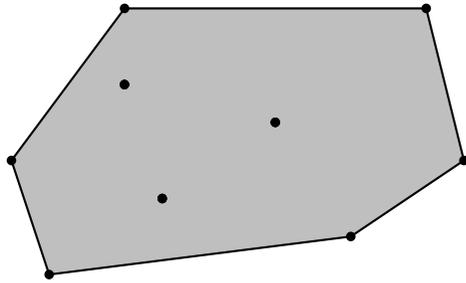


FIG. 9. L'enveloppe convexe des points est matérialisée par la zone grise du dessin.

Nous allons montrer qu'une courbe de Bézier est incluse dans l'enveloppe convexe de ses points de contrôle. D'après la définition 12, le segment $[A, B]$ est l'ensemble des barycentres des points A et B pondérés par des valeurs positives. D'autre part, une partie P d'un \mathbf{R} -espace vectoriel E est dite convexe si et seulement si

$$\forall(A, B) \in P \times P, [A, B] \subseteq P.$$

Ainsi, si deux points A et B appartiennent à F , alors tout le segment appartient à F . De manière très imagée, l'enveloppe convexe d'un ensemble de points du plan est la région intérieure délimitée par un élastique que l'on aurait tendu puis relâché autour de tous ces points. Plus formellement, il est évident que l'intersection de deux convexes est une partie convexe, on peut donc écrire

Définition 23. On appelle enveloppe convexe d'une famille (finie ou infinie) \mathcal{M} de points d'un espace vectoriel, l'intersection de toutes les parties convexes qui contiennent \mathcal{M} . On la note $\text{env}(\mathcal{M})$.

Il est faux de dire qu'il s'agit de la plus petite partie convexe (au sens de l'inclusion) qui contient \mathcal{M} , en effet, l'inclusion n'est pas une relation d'ordre totale, on pourrait donc au mieux parler de convexes minimaux contenant \mathcal{M} et on n'aurait pas l'unicité.

Théorème 24. L'enveloppe convexe d'une famille finie $\mathcal{M} = \{M_1, M_2, \dots, M_n\}$ de points d'un \mathbf{R} -espace vectoriel est l'ensemble des barycentres de ces points, i.e.

$$(22) \quad \text{env}(\mathcal{M}) = \left\{ \sum_{i=0}^n u_i M_i; u_i \geq 0, \sum_{i=0}^n u_i = 1 \right\}.$$

Démonstration. L'ensemble des barycentres des points M_i est évidemment une partie convexe qui contient ces points, elle contient donc l'enveloppe convexe de ces points. Réciproquement, soit M un barycentre des points M_i . On peut construire M de proche en proche en calculant la succession de barycentres de deux points uniquement, donc dans l'enveloppe convexe qui contient tous les M_i . \square

Corollaire 25. Une courbe de Bézier est contenue dans l'enveloppe convexe de ses points de contrôle.

Preuve. L'expression (21) montre que les fonctions associées aux polynômes de Bernstein sont toutes positives sur l'intervalle $[0, 1]$, il suffit alors de remarquer que l'enveloppe convexe est constituée de tous les barycentres des M_i , or d'après les expressions (20) et (21), la courbe de Bézier est une famille particulière de barycentres des points de contrôle. \square

Exercice 28 \heartsuit Montrez que l'on a les valeurs particulières des fonctions polynomiales de Bernstein suivantes ($0 < p < n$, $n \neq 0$) :

$$\begin{aligned} B_{n,0}(0) &= 1 & B_{n,0}(1) &= 0 \\ B_{n,p}(0) &= 0 & B_{n,p}(1) &= 0 \\ B_{n,n}(0) &= 0 & B_{n,n}(1) &= 1 \end{aligned}$$

En déduire que les courbes de Bézier passent nécessairement par les points de contrôle extrémaux, i.e. par M_0 et M_n .

Exercice 29 \dagger Écrivez un algorithme en $\Theta(n)$ pour calculer les coefficients des polynômes de Bernstein de degré n en utilisant le triangle de Pascal (on suppose que le triangle de Pascal est précalculé).

Théorème 26. Les polynômes de Bernstein $B_{n,p}(T)$, $0 \leq p \leq n$, forment une base de l'espace $\mathbf{R}_n[T]$ des polynômes de degré au plus n sur \mathbf{R} .

Exercice 30 \ddagger Démontrez le théorème 26. Déduisez de ce résultat comment déterminer les n points de contrôle de la courbe de Bézier égale à une courbe polynomiale de degré n fixée.

On peut donc représenter tout polynôme de degré inférieur à n à l'aide d'une combinaison linéaire de polynômes de Bernstein. Cette propriété est évidemment fondamentale, elle exprime le fait que les polynômes de Bernstein de degré n permettent de représenter toutes les courbes obtenues à l'aide de polynômes de degrés inférieurs à n .

Notons au passage que l'espace des fonctions polynomiales à coefficients dans \mathbf{Q} est dense dans l'espace des fonctions continues réelles, autrement dit, on peut approcher d'aussi près que l'on veut n'importe quelle courbe à l'aide de courbes polynomiales à coefficients dans \mathbf{Q} . C'est l'objet du théorème de Weierstrass et du corollaire qui suivent.

Théorème 27 (Weierstrass). *Soient a et b deux nombres réels. Toute fonction continue $f : [a, b] \rightarrow \mathbf{R}$ peut être approchée uniformément d'aussi près que l'on veut par des fonctions polynomiales. Autrement dit, le \mathbf{R} -espace vectoriel des fonctions polynomiales associé à $\mathbf{R}[X]$ est partout dense dans $\mathcal{C}^0([a, b], \mathbf{R})$ pour la norme uniforme $\| \cdot \|_\infty$ définie par*

$$\|f\|_\infty = \sup_{x \in [a, b]} |f(x)|.$$

Preuve. voir \square

Corollaire 28. $\mathbf{Q}[X]$ est partout dense dans $\mathcal{C}^0([a, b], \mathbf{R})$.

Preuve. On veut montrer que pour tout $\epsilon > 0$, il existe un polynôme $q(X)$ à coefficients dans \mathbf{Q} tel que

$$\|f(x) - q(x)\|_\infty < \epsilon.$$

D'après le théorème de Weierstrass 27, il existe un polynôme $p(X) = a_0 + a_1X + \dots + a_nX^n$ à coefficients réels tel que

$$(23) \quad \|f(x) - p(x)\|_\infty < \frac{\epsilon}{2}.$$

On sait par ailleurs \square que \mathbf{Q} est dense dans \mathbf{R} pour la topologie associée à la norme usuelle $| \cdot |$. Alors, pour tout $i \in \{1, \dots, n\}$, il existe une suite $(a_n^{(i)})_j, j \in \mathbf{N}$, telle que

$$(24) \quad \forall \epsilon_i > 0, \exists N_i \in \mathbf{N}, \forall n \geq N_i, |a_n^{(i)} - a_i| < \epsilon_i.$$

Notons $q(X)$ le polynôme $a_m^{(0)} + a_m^{(1)}X + \dots + a_m^{(n)}X^n$. On en déduit qu'il existe toujours un entier M tel que pour tout $m \geq M$ on a

$$(25) \quad \|p(x) - q(x)\|_\infty < \frac{\epsilon}{2}.$$

On conclut avec (23) et l'inégalité triangulaire :

$$\begin{aligned} \|f(x) - q(x)\|_\infty &= \|f(x) - p(x) + p(x) - q(x)\|_\infty \\ &\leq \|f(x) - p(x)\|_\infty + \|p(x) - q(x)\|_\infty \\ &< \epsilon \end{aligned}$$

Mais quel rapport y-a-t-il entre ces propriétés de densité *a priori* purement mathématiques et l'infographie? Le théorème de Weierstrass nous dit que l'on peut se contenter de tracer des courbes polynomiales car elles permettent d'approcher d'aussi près que l'on veut n'importe quelle courbe continue, on ne perd donc rien en généralité. La densité de \mathbf{Q} dans \mathbf{R} nous permet

d'affirmer qu'il est inutile de considérer des polynômes à coefficients dans \mathbf{R} et que l'on peut se contenter de coefficients rationnels ce qui est bien pratique quand on écrit des programmes, car rappelons le, les nombres réels ne sont représentés en machines que par un très petit nombre de rationnels.

Exemple 2. Considérons le cas d'une cubique de Bézier ($n = 3$) dont les 4 points de contrôle sont $M_0 = (0, 1)$, $M_1 = (2, 5)$, $M_2 = (5, 4)$ et $M_3 = (6, 0)$. Les polynômes de Bernstein sont les suivants :

$$B_{3,0}(T) = -T^3 + 3T^2 - 3T + 1$$

$$B_{3,1}(T) = 3T^3 - 6T^2 + 3T$$

$$B_{3,2}(T) = -3T^3 + 3T^2$$

$$B_{3,3}(T) = T^3$$

On obtient donc la courbe cubique définie par les coordonnées paramétriques suivantes (voir figure 10) :

$$\begin{cases} X(u) = -3u^3 + 3u^2 + 6u \\ Y(u) = 2u^3 - 15u^2 + 12u + 1 \end{cases}$$

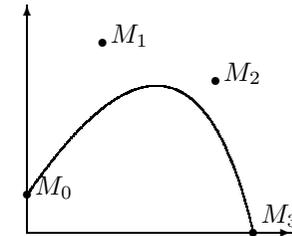


FIG. 10. Courbe de Bézier cubique

Exercice 31 † Considérons la matrice \mathcal{B} des coefficients de Bernstein qui est définie comme la matrice carrée d'ordre $n + 1$ définie par

$$(26) \quad \mathcal{B}_{i,j} = B_{n,j}^{(i)}, \quad 0 \leq i, j \leq n.$$

où $B_{n,j}^{(i)}$ désigne le coefficient de degré i du polynôme de Bernstein $B_{n,j}(T)$. Exemple : dans le cas des courbes de Bézier cubiques, la matrice \mathcal{B} est la suivante (voir exemple 2) :

$$\mathcal{B} = \begin{pmatrix} +1 & +0 & +0 & +0 \\ -3 & +3 & +0 & +0 \\ +3 & -6 & +3 & +0 \\ -1 & +3 & -3 & +1 \end{pmatrix}$$

Montrez que si $x = (x_0, x_1, \dots, x_n)$ et $y = (y_0, y_1, \dots, y_n)$ désignent respectivement le vecteur formé par les abscisses (ordonnées) des points de contrôle M_0, M_1, \dots, M_n dans cet ordre, alors les produits $\mathcal{B}x$ et $\mathcal{B}y$ sont les vecteurs dont les composantes sont les coefficients des polynômes $X(T)$ et $Y(T)$ qui définissent respectivement l'abscisse et l'ordonnée du point générique $M(u) = (X(u), Y(u))$ de la courbe de Bézier.

Ecrivez un algorithme qui calcule la matrice \mathcal{B} en supposant que les coefficients binomiaux sont précalculés dans une table à deux entrées. Calculez la complexité de cet algorithme. Ecrivez un algorithme qui calcule les coordonnées du point $M(u)$ en utilisant la factorisation de Hörner pour l'évaluation des fonctions polynomiales X et Y .

Les courbes de Bézier n'ont pas été définies directement à partir de l'expression (21) mais avec un procédé algorithmique, la formalisation est venue ensuite, c'est ce que nous allons étudier. À partir des $n + 1$ points de contrôle, on peut construire les n barycentres des paires $\{M_i, M_{i+1}\}$, pour $i \in \{0, \dots, n\}$ affectés des coefficients $1 - u$ et u , c'est à dire les n points

$$(1 - u)M_i + uM_{i+1}, \quad i \in \{0, \dots, n\}.$$

On peut alors recommencer à calculer deux-à-deux les $n - 1$ barycentres des n points obtenus (toujours avec les mêmes coefficients) et ainsi de suite jusqu'à n'obtenir qu'un seul point. C'est cette méthode qui est représentée dans la figure 11.

A *posteriori* en formalisant le procédé, on définit la famille de points $M_{i,j}$, $0 \leq i \leq n$ et $0 \leq j \leq n - i$ définis récursivement par les relations suivantes :

$$(27) \quad M_{i,j} = \begin{cases} M_i & \text{si } j = 0, \\ (1 - u)M_{i,j-1} + uM_{i+1,j-1} & \text{si } j \neq 0. \end{cases}$$

La base récurrente est donnée par les points de contrôle M_i , quand $j = 0$. L'indice i désigne l'ordre du point dans la séquence de barycentres, et j désigne la génération de barycentre (la génération 0 correspondant aux $n + 1$ points de contrôle).

On montre alors (exercice 32) que le dernier point obtenu $M_{0,n}$ décrit, en fonction du coefficient de pondération u , la courbe de Bézier associée aux points de contrôle M_p , $0 \leq p \leq n$ (c'est l'objet de l'exercice 32). C'est ainsi qu'apparaissent les coefficients binomiaux des polynômes de Bernstein.

Exemple 3. Dans le cas d'une courbe de Bézier cubique, on a les points suivants :

$$\begin{aligned} M_{0,0} &= M_0 & M_{0,1} &= (1 - u)M_{0,0} + uM_{1,0} & M_{0,2} &= (1 - u)M_{0,1} + uM_{1,1} \\ M_{1,0} &= M_1 & M_{1,1} &= (1 - u)M_{1,0} + uM_{2,0} & M_{1,2} &= (1 - u)M_{1,1} + uM_{2,1} \\ M_{2,0} &= M_2 & M_{2,1} &= (1 - u)M_{2,0} + uM_{3,0} \\ M_{3,0} &= M_3 \end{aligned}$$

et $M_{0,3} = (1 - u)M_{0,2} + uM_{1,2}$.

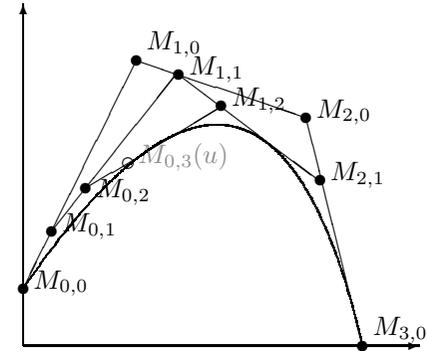


FIG. 11. Algorithme de calcul récursif d'une courbe de Bézier

Exercice 32 ‡ Démontrez que la courbe $M_{0,n}(u)$, $0 \leq u \leq 1$ est bien la courbe de Bézier associée aux points de contrôle M_p , $0 \leq p \leq n$ où $M_{0,n}(u)$ est défini par la relation de récurrence (27). Pour cela, montrez que l'arbre binaire de racine $M_{0,n}$ dont les nœuds sont les points $M_{i,j}$ avec pour fils gauche le point $M_{i,j-1}$ et pour fils droit le point $M_{i+1,j-1}$ est parfaitement équilibré de hauteur n . Montrez que l'indexation des 2^n feuilles A_0, \dots, A_{2^n-1} correspond en binaire au parcours effectué pour atteindre ces feuilles en partant de la racine de l'arbre et en codant 0 pour le fils gauche et 1 pour le fils droit.

On rappelle que le poids binaire $wt(k)$ d'un entier k est égal au nombre de 1 dans sa sa décomposition binaire. Par exemple $wt(13) = wt(1101) = 3$. Utilisez le poids binaire des indices pour faire votre calcul.

Exercice 33 † Déduisez de la relation (27) un algorithme récursif pour tracer une courbe de Bézier. Ecrivez le programme correspondant. Montrez que certains calculs sont effectués plusieurs fois et indiquez une méthode itérative qui évite ces calculs superflus (trouvez un ordre de calcul des $M_{i,j}$ qui assure que les deux autres valeurs intervenant dans le calcul ont déjà été calculées). Comparez alors la complexité du tracé des courbes de Bézier entre les trois algorithmes suivants :

- (1) implantation directe à partir de l'expression (21).
- (2) avec la définition récursive (27).
- (3) avec la méthode itérative qui évite les calculs superflus.

Travaux Pratiques 3 † On veut écrire un programme qui trace des courbes de Bézier. Ce programme doit demander à l'utilisateur de saisir le nombre $n+1$ de points de contrôle de la courbe puis saisir les $n+1$ couples correspondants. Pour l'affichage, le programme doit demander à l'utilisateur de saisir le nombre de points à tracer et la dimension de la clôture (fenêtre écran). L'utilisateur doit pouvoir modifier tout au long de l'exécution du programme un point de contrôle et voir la modification induite sur la courbe précédente. Pour cela tracer la courbe de référence et la courbe modifiée avec des couleurs différentes. Proposez également à l'utilisateur de choisir l'une des trois méthodes de calcul. Application : tracez la courbe de l'exemple précédent à partir de l'expression (21).

8. COURBES DE BÉZIER PAR MORCEAUX

Le défaut des courbes de Bézier comme nous venons de les définir est que le degré des polynômes qui interviennent dans la définition du point $M(u)$ de la courbe est égal à n . D'autre part, toute modification sur un point de contrôle de la courbe affecte l'ensemble de la courbe, il faut donc retracer la courbe entièrement. Pour ces différentes raisons, on préfère construire des courbes de Bézier *par morceaux*, i.e. on obtient une courbe qui est la jonction de courbes de Bézier définies à partir d'un nombre restreint de points de contrôle. Ainsi, outre les calculs qui sont simplifiés, la modification d'un point de contrôle affectera *localement* la forme de la courbe générale, puisque cette modification ne se fait que sur une des courbes de Bézier. En général, on utilise des courbes de Bézier cubiques (le cas quadratique manque singulièrement de souplesse et pose des problèmes pour la jonction des portions de courbe).

On définit aisément des portions de courbes à l'aide de Bézier cubiques, mais encore faut-il joindre correctement les morceaux ainsi obtenus. Nous allons mettre en évidence les conditions pour faire une "bonne" jonction. Il est clair que la première condition pour deux courbes de Bézier \mathcal{C}_1 et \mathcal{C}_2 est que le dernier point de contrôle de la courbe \mathcal{C}_1 soit égal au premier point de contrôle de la courbe \mathcal{C}_2 , autrement dit, la courbe doit être continue!

Définition 29. *Dans une courbe de Bézier par morceaux, les points de jonction des différentes courbes de Bézier sont appelés des nœuds.*

Malheureusement cette première condition de continuité sur la courbe (condition de continuité C^0) est hautement insuffisante, en tout cas pour les besoins traditionnels de l'infographie. Si on se contentait de cette condition, on accepterait des courbes dont l'aspect serait quelque peu chaotique! Il est légitime d'imposer que les tangentes aux points de jonction soient les mêmes pour les deux courbes. Nous allons étudier ce que cette nouvelle condition impose comme contraintes sur notre courbe.

La tangente en un point $M(a)$ d'une courbe définie paramétriquement par les points $M(u)$, $u \in I$ introduits dans le système (18) est donnée par la droite paramétrée

suivante :

$$(28) \quad T(u) = M(a) + uM'(a) = \begin{pmatrix} X(a) + uX'(a) \\ Y(a) + uY'(a) \end{pmatrix}, \quad u \in \mathbf{R}.$$

où $X'(a)$ désigne la valeur de la dérivée de la fonction $u \mapsto X(u)$ au point a et $Y'(a)$ la valeur de la dérivée de $u \mapsto Y(u)$ au point a . On a obtenu cette droite, en calculant le vecteur dérivé de M en a et en translatant la droite vectorielle associée par le vecteur $M(a)$.

Considérons à présent deux courbes de Bézier \mathcal{C}_1 et \mathcal{C}_2 de même degré n (on peut cependant faire les mêmes démonstrations sans cette hypothèse sans pour autant compliquer les calculs). Soient M_0, M_1, \dots, M_n les $n+1$ points de contrôle de la première courbe et $M_n, M_{n+1}, \dots, M_{2n}$ les $n+1$ points de contrôle de la seconde courbe. On rappelle que la première condition avait imposé que le dernier point de contrôle de la première courbe soit égal au premier point de contrôle de la seconde. L'équation de la tangente en M_n est obtenue pour $u=1$ avec la courbe \mathcal{C}_1 et pour $u=0$ avec la courbe \mathcal{C}_2 . En notant $M_{(1)}(u)$ le point générique de la courbe \mathcal{C}_1 et $M_{(2)}(u)$ celui de \mathcal{C}_2 , on déduit du système (28) la condition de continuité du premier ordre C^1 :

$$(29) \quad M'_{(1)}(1) = M'_{(2)}(0).$$

Il suffit alors de calculer les dérivées des polynômes de Bernstein pour les différentes valeurs de p en 0 et en 1, on a

$$(30a) \quad B'_{n,0}(T) = -n(1-T)^{n-1}$$

$$(30b) \quad B'_{n,n}(T) = nT^{n-1}$$

$$(30c) \quad B'_{n,p}(T) = \binom{n}{p} T^{p-1} (1-T)^{n-p-1} (p-nT), \quad 0 < p < n$$

On en déduit les valeurs

$$\begin{aligned} B'_{n,0}(0) &= -n & B'_{n,n}(1) &= n \\ B'_{n,1}(0) &= n & B'_{n,n-1}(1) &= -n \\ B'_{n,p}(0) &= 0 & B'_{n,p}(1) &= 0 \end{aligned}$$

et on exprime alors la condition (29) sous la forme

$$n(M_n - M_{n-1}) = n(M_{n+1} - M_n),$$

c'est à dire

$$(31) \quad M_n = \frac{1}{2}(M_{n-1} + M_{n+1}).$$

Autrement dit, les trois points M_{n-1} , M_n et M_{n+1} sont alignés et le nœud M_n est situé au milieu des deux autres points. Cependant, est-ce que la condition de continuité du premier ordre suffit? Cela dépend en fait du degré de finesse que l'on veut obtenir. Supposons par exemple que l'on trace la courbe obtenue en joignant un segment de

droite avec un arc de cercle (figure 12). Imaginons maintenant un mobile qui se déplace le long de cette droite, que va-t-il se passer quand il va aborder la courbe? Ce mobile

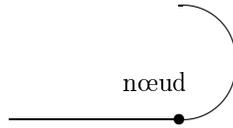


FIG. 12. Jonction C^1 entre deux courbes

va voir la composante verticale de son accélération passer de 0 à une valeur positive à l’instant précis où il va entamer la courbe, cela sous-entend une variation brutale de l’accélération. La condition de continuité à l’ordre 0 impose une fonction *position* continue (le mobile ne se téléporte pas dans un ravin), la condition de continuité du premier ordre impose une fonction *vitesse* continue du véhicule. Au second ordre elle impose une fonction *accélération* continue. On peut évidemment aller plus loin, tout dépend donc du type de courbe que l’on veut obtenir.

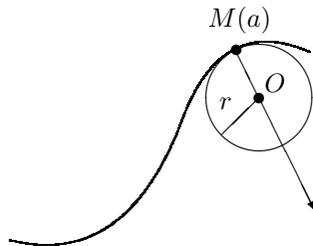


FIG. 13. Cercle osculateur et rayon de courbure

La continuité du second ordre impose en fait que le *rayon de courbure* de la courbe au nœud soit le même des deux cotés. Très schématiquement, en un point particulier $M(a)$ d’une courbe, si elle est C^2 , il existe un vecteur normal intérieur (qui se dirige dans la concavité de la courbe). Si on place un cercle tangent au point $M(a)$, dont le centre est sur la droite passant par $M(a)$ de vecteur directeur le vecteur normal intérieur, on peut augmenter son rayon tant que localement, autour de $M(a)$, la courbe ne coupe pas la tangente en d’autres points que $M(a)$. En passant à la limite, le rayon finalement obtenu est ce rayon de courbure. Le cercle en question est appelé *cercle osculateur* de

la courbe en $M(a)$ et son rayon est donné par la formule suivante :

$$(32) \quad r(a) = \frac{X'(a)Y''(a) - Y'(a)X''(a)}{(X'(a)^2 + Y'(a)^2)^{\frac{3}{2}}}$$

La continuité du second ordre impose également des conditions sur les points de contrôle :

Exercice 34 † Démontrez que la continuité du second ordre sur un nœud M_n de deux courbes de Bézier ayant pour points de contrôle respectifs les points M_0, M_1, \dots, M_n et $M_n, M_{n+1}, \dots, M_{2n}$ impose la condition suivante sur les points de contrôle des deux courbes :

$$(33) \quad M_{n-2} - 2M_{n-1} - M_n = M_{n+2} - 2M_{n+1} - M_n$$

En utilisant la condition de continuité du premier ordre (31), montrez que l’on a l’égalité suivante pour le troisième point de contrôle de la seconde courbe :

$$(34) \quad M_{n+2} = M_{n-2} + 4(M_n - M_{n-1}).$$

En conclusion, si on impose une continuité du second ordre pour des courbes de Bézier cubiques, le seul point de contrôle que l’on peut effectivement modifier est le dernier point puisque les trois premiers sont fixés par la condition de continuité. Pour pallier ce handicap, on peut évidemment augmenter le nombre de points de contrôle et par conséquent le degré des polynômes.

Exercice 35 † Calculez le vecteur unitaire normal au point $M(a)$, $a \in [0, 1]$ à une courbe de Bézier d’ordre n . Pour cela, calculez l’équation de la tangente au point $M(a)$.

Travaux Pratiques 4 † Faites la jonction de la courbe de Bézier de l’exemple 2 avec la courbe de Bézier cubique définie par le dernier point de contrôle (12, 3) en respectant une continuité du second ordre. Tracez la courbe obtenue.

9. COURBES B-SPLINES

Le mot *spline* signifie *latte*. L'origine de la terminologie vient de la construction des bateaux : on plantait des clous sur une surface plane et on intercalait une latte flexible entre ces clous qui prenait alors une forme optimale en ce sens qu'elle minimise l'énergie. Les courbes ainsi obtenues servaient à modeler les coques des navires.

Le principe pour définir les courbes *B-splines* est le même que pour les courbes de Bézier mais les $n + 1$ polynômes de Bernstein sont remplacés par $n + 1$ fonctions polynomiales par morceaux. Un des objectifs des *B-splines* est de limiter l'influence d'un point de contrôle à une portion de la courbe pour éviter de refaire un tracé complet en cas de modification de ce point, de la même manière que pour des courbes de Bézier par morceaux. On veut donc une modification *locale* et non *globale* de la courbe. Contrairement aux courbes de Bézier où le degré des polynômes de Bernstein est directement lié au nombre de points de contrôle puisqu'il vaut n , le degré m des fonctions polynomiales *B-splines* est fixé par l'utilisateur, on veut simplement que $m \leq n$.

On se donne encore une fois un ensemble de $n + 1$ points de contrôle M_0, M_1, \dots, M_n et m un entier qui désigne le degré commun des polynômes *B-splines*. En plus des $n + 1$ points de contrôle, il faut définir une séquence croissante de $n + m + 2$ valeurs réelles appelée *séquence nodale* ou encore *vecteur nœud* de la courbe :

$$(35) \quad u_0 \leq u_1 \leq \dots \leq u_{n+m+1}.$$

Notons dès à présent que cette suite est croissante mais pas nécessairement équirépartie. D'autre part, il est possible que certains u_i adjacents soient identiques. Plus précisément, un nœud u_i est dit de *multiplicité* k , s'il existe une sous-liste maximale $(u_j)_{j \in J}$ de longueur k (i.e. $\#J = k$) de $(u_0, u_1, \dots, u_{n+m+1})$ telle que $\forall j \in J, u_j = u_i$. Plus simplement on compte le nombre maximal d'occurrences successives du nœud u_i . On la note $\nu(u_i)$. Le point $M(u)$ qui décrit la courbe *B-spline* est alors défini par l'équation

$$(36) \quad M(u) = \sum_{p=0}^n N_{p,m}(u)M_p, \quad u \in [u_m, u_{n+1}[.$$

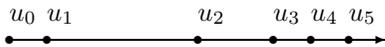


FIG. 14. Séquence nodale d'une courbe de *B-spline* (ici $n + m + 1 = 5$)

Les fonctions *B-splines* $N_{i,j}$, $0 \leq i \leq n$, $0 \leq j \leq m$, sont définies récursivement. La base récurrente est donnée par $N_{i,0}(u) = \mathbf{1}_{[u_i, u_{i+1}[}(u)$, où $\mathbf{1}_{[u_i, u_{i+1}[}$ désigne l'indicatrice de l'intervalle $[u_i, u_{i+1}[$, c'est-à-dire :

$$\mathbf{1}_{[u_i, u_{i+1}[}(u) = \begin{cases} 1, & \text{si } u \in [u_i, u_{i+1}[. \\ 0, & \text{sinon.} \end{cases}$$

L'auto-définition pour une fonction polynomiale *B-spline* de *classe* j est donnée par la formule suivante :

$$(37) \quad N_{i,j}(u) = \frac{u - u_i}{u_{i+j} - u_i} N_{i,j-1}(u) + \frac{u_{i+1+j} - u}{u_{i+1+j} - u_{i+1}} N_{i+1,j-1}(u).$$

Cette définition récursive indique au passage pourquoi $m \leq n$, et il faut remarquer que l'indice j désigne de la même manière que pour les fonctions de Bézier, la génération de fonctions. On pose par convention pour ces calculs que $0/0 = 0$. De la même façon que dans l'exercice 33.(3), pour éviter de répéter certains calculs, il est préférable de s'assurer pour calculer $N_{i,j}$ que $N_{i-1,j+1}$ et $N_{i,j+1}$ ont déjà été calculés, ce qui est possible grâce au parcours dans le sens de lecture du tableau ci-dessous :

	$\mathbf{1}_{[u_0, u_1]}$	$\mathbf{1}_{[u_1, u_2]}$	$\mathbf{1}_{[u_2, u_3]}$	$\mathbf{1}_{[u_{n+m-1}, u_{n+m}]}$	$\mathbf{1}_{[u_{n+m}, u_{n+m+1}]}$
1	$N_{0,1}$	$N_{1,1}$	$N_{2,1}$	$N_{n+m-1,1}$	
	⋮							
m	$N_{0,m}$	$N_{1,m}$	$N_{2,m}$...	$N_{n,m}$			

TAB. 1. Table de construction des splines.

On rappelle que le *support* d'une fonction f définie sur un ensemble E est l'adhérence de la partie S de E sur laquelle la fonction est non-nulle. On le note $\text{supp}(f)$. La démonstration de la proposition suivante est laissée en exercice (voir exercice 36).

Pour ceux qui ont totalement oublié leur cours d'analyse de premier cycle, l'adhérence d'une partie de \mathbf{R} est le plus petit "fermé" qui contient la partie en question. Pour les besoins de ce cours d'infographie, on peut se contenter de noter que l'adhérence d'un intervalle $]a, b[,]a, b]$ ou encore $[a, b[$ est tout simplement l'intervalle $[a, b]$.

Proposition 30. *Le support de la fonction B-spline $N_{i,j}$ vaut*

$$(38) \quad \text{supp}(N_{i,j}) = [u_i, u_{i+j+1}].$$

D'autre part, $N_{i,j}(u) > 0$ si $u \in]u_i, u_{i+j+1}[$ et $N_{i,j}(u) = 0$ si $u = u_i$ ou $u = u_{i+j+1}$.

Exercice 36 † Démontrez la proposition précédente à l'aide d'un raisonnement par récurrence.

On a donc $\text{supp}(N_{i,j-1}) = [u_i, u_{i+j}]$ et $\text{supp}(N_{i+1,j-1}) = [u_{i+1}, u_{i+j+1}]$. A partir de la formule (37) on observe qu'une fonction B -spline $N_{i,j}$ est la moyenne pondérée des B -splines $N_{i,j-1}$ et $N_{i+1,j-1}$ avec pour coefficients de pondération respectifs, le rapport entre la distance de u à u_i (u_{i+j+1}) et la largeur du support de $N_{i,j-1}$ ($N_{i+1,j-1}$).

Plus important, si l'on veut montrer qu'une courbe B -spline est incluse dans l'enveloppe convexe de ses points de contrôle, il faut que les points de cette courbe soient des barycentres de ses points de contrôle (voir preuve de la proposition 25).

Théorème 31. *Les fonctions B -splines satisfont l'égalité suivante pour tout m :*

$$(39) \quad \sum_{p=0}^n N_{p,m}(u) = 1, \quad \forall u \in [u_m, u_{n+1}[.$$

Preuve. Pour $m = 0$, c'est évident puisque les B -splines $N_{p,0}$ sont des indicatrices à supports disjoints. Supposons que ce soit vrai pour $m - 1$, alors

$$\sum_{p=0}^n N_{p,m}(u) = \sum_{p=0}^n \frac{u - u_p}{u_{p+m} - u_p} N_{p,m-1}(u) + \sum_{p=1}^{n+1} \frac{u_{p+m} - u}{u_{p+m} - u_p} N_{p,m-1}(u)$$

où l'on a simplement fait le changement d'indexation $p + 1 \mapsto p$ dans la deuxième somme. Mais $N_{0,m-1}$ vaut 0 pour $u \in [u_m, u_{n+1}]$, en effet $\text{supp}(N_{0,m-1}) = [u_0, u_m[$. De la même façon, sur cet intervalle $N_{n+1,m-1}$ vaut 0. On peut donc rajouter le premier terme à la deuxième somme et le deuxième terme à la première somme et les regrouper pour obtenir :

$$\begin{aligned} \sum_{p=0}^n N_{p,m}(u) &= \sum_{p=0}^{n+1} \frac{u_{p+m} - u_p}{u_{p+m} - u_p} N_{p,m-1}(u) \\ &= \sum_{p=0}^{n+1} N_{p,m-1}(u). \end{aligned}$$

Et l'hypothèse de récurrence permet de conclure. □

Corollaire 32. *Une courbe B -spline est contenue dans l'enveloppe convexe de ses points de contrôle.*

Preuve. Voir démonstration du corollaire 25. □

Exemple 4. Pour des fonctions B -splines de classe $m = 0, 1, 2$ et 3 , on obtient les courbes de la figure 15. Notons que dans la situation où les nœuds sont *équirépartis* (B -splines uniformes, voir §10), la courbe B -spline $N_{p+k,m}$ est obtenue en translatant la courbe $N_{p,m}$, de k la largeur du pas.

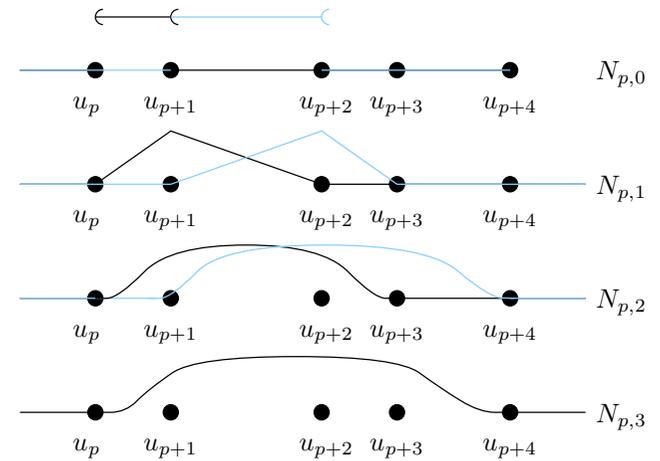


FIG. 15. Fonctions B -splines pour $m = 0, 1, 2$ et 3

Travaux Pratiques 5 † Écrivez un programme qui trace les courbes B -splines de classe m . Essayez de tracer ces courbes avec des nœuds de multiplicité supérieure à 1.

Contrairement aux courbes de Bézier, où la continuité en tout point est de classe C^n , n désignant le degré commun des polynômes de Bernstein, les courbes B -splines ne sont pas des polynômes mais des polynômes par morceaux, il faut donc calculer la continuité aux nœuds de ces courbes (ailleurs, ce sont des polynômes et le résultat est évident). On admettra le théorème suivant :

Théorème 33. *Une courbe B -spline est de classe de continuité $C^{m-\nu(u_i)}$ au nœud u_i , où $\nu(u_i)$ désigne l'ordre de multiplicité de u_i .*

Ce résultat est très important, il permet de tracer des courbes avec une continuité arbitraire, on peut ainsi inclure des points de rebroussement ou même avoir une coupure de la courbe si $\nu(u_i) = m$.

D'autre part, en définissant la multiplicité d'un point de contrôle de manière similaire à la multiplicité d'un nœud, on peut rapprocher la courbe d'un point de contrôle particulier en augmentant sa multiplicité et même passer par ce point si sa multiplicité atteint $m - 1$ où m est la classe de la courbe B -spline (voir exercice 37).

Exercice 37 ‡ Montrer qu'une courbe B -spline de classe m contenant un point de contrôle d'ordre $m - 1$ passe par ce point de contrôle.

Exemple 5. Dans la courbe B -spline ci-dessous, il y a 8 points de contrôle M_1, M_2, \dots, M_7 . Le degré des fonctions splines est fixé à $m = 3$ et on a posé $M_0 = M_1$ et $M_6 = M_7$, la courbe passe donc par ces “deux” points :

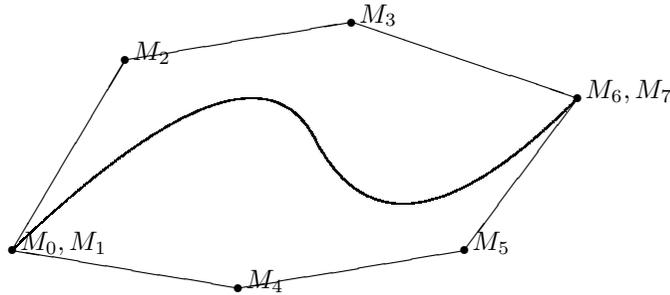


FIG. 16. Courbe B -spline de degré $m = 3$ (l’enveloppe convexe est tracée).

Exercice 38 † Quelle courbe B -spline obtient-on avec des fonctions B -splines de classe 0? De classe 1?

Exemple 6. On peut produire des courbes fermées très facilement en jouant sur la multiplicité des points de contrôle.

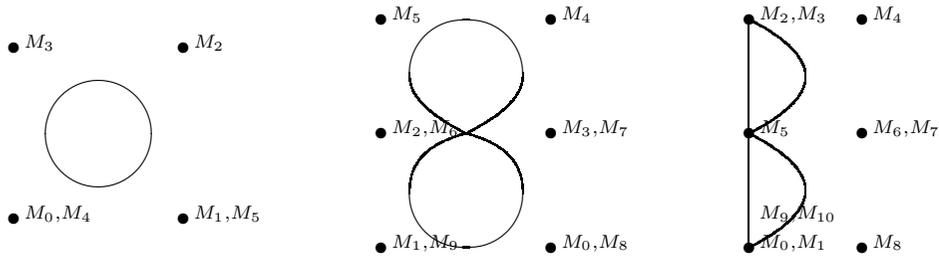


FIG. 17. Courbes B -splines fermées pour $m = 3$

Exercice 39 † Montrez que les fonctions B -splines $N_{i,k+1}$ de classe $k + 1$ telles que chaque nœud apparaît avec une multiplicité $k + 1$ sont exactement les polynômes de Bernstein $B_{i,k}$.

10. COURBES B -SPLINES UNIFORMES

Les courbes B -splines les plus utilisées sont celles pour lesquelles le vecteur nœud est *uniforme*, i.e. $\forall i \in [0, \dots, n + m], u_{i+1} - u_i = K$ ou éventuellement 0 quand on veut diminuer la continuité. Dans ce cas on peut poser $u_i = i$, et on obtient à la place de (37) :

$$(40) \quad N_{i,j}(u) = \frac{1}{j} \left((u - i)N_{i,j-1}(u) + (i + 1 + j - u)N_{i+1,j-1}(u) \right).$$

Pour terminer avec ces courbes B -splines, notons que d’un point de vue technique, on ne peut pas réellement considérer qu’une modification d’un point de contrôle sur une courbe de Bézier soit globale et locale sur une B -spline. En effet, formellement une courbe de Bézier est définie sur un segment et la modification d’un point de contrôle se répercute donc sur tout le segment, c’est exactement la même chose pour une B -spline! D’ailleurs, faire des jonctions de courbes de Bézier revient à les définir sur plusieurs segments et là on constate bien que les modifications restent locales. Ce qui distingue plus justement les Béziers des B -splines c’est que pour conserver une certaine classe de continuité, les courbes de Bézier imposent la position des points de contrôle, alors que dans le cas des B -splines, les points de contrôle ne sont pas modifiés par la continuité en chaque nœud. D’autre part, le degré des polynômes B -splines est fixé au départ et n’est lié au nombre de points de contrôle que par l’inégalité $m \leq n$.

11. SURFACES DE BÉZIER

Le principe est le même pour les surfaces que pour les courbes, mais les points de contrôle sont définis sur une grille rectangulaire qui a subi une déformation élastique dans l’espace.

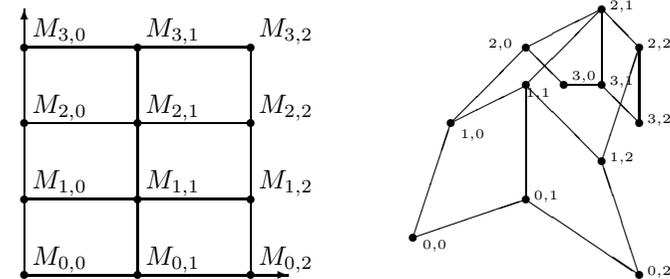


FIG. 18. Grille planeire et points de contrôle associés

La représentation paramétrique de la surface nécessite évidemment un deuxième degré de liberté et dépend donc de deux paramètres u et v :

$$(41) \quad M(u, v) = \begin{pmatrix} X(u, v) \\ Y(u, v) \\ Z(u, v) \end{pmatrix}$$

Définition 34. On appelle surface de Bézier de degré nm , associée aux points de contrôle $M_{p,q}$, $0 \leq p \leq n$, $0 \leq q \leq m$, la surface paramétrique définie par les points suivants :

$$(42) \quad M(u, v) = \sum_{p=0}^n \sum_{q=0}^m B_{n,p}(u) B_{m,q}(v) M_{p,q}, \quad (u, v) \in [0, 1] \times [0, 1].$$

De la même façon que pour les courbes, les surfaces de Bézier habituellement employées sont des surfaces dites *bicubiques*, i.e. $n = m = 3$, et on emploie les mêmes techniques pour construire des surfaces complexes à partir de surfaces élémentaires bicubiques. La jonction entre deux surfaces de Bézier se fait sur un des “bords” des deux surfaces, autrement dit, il n’y a plus 1 point de contrôle en commun, mais n ou m points en fonction du bord où l’on fait la jonction. Les conditions de continuité du premier et du second ordre s’obtiennent de la même façon que pour les courbes, les calculs sont simplement plus fastidieux.

Nous donnons seulement l’équation paramétrique du plan tangent au point $M(a, b)$. On obtient deux vecteurs tangents à la surface $(u, v) \mapsto M(u, v)$ en (a, b) en dérivant $M(u, v)$ suivant les directions u et v , le plan vectoriel est donc engendré par ces deux vecteurs. On obtient finalement le plan affine $(u, v) \mapsto T(u, v)$ en translatant le plan vectoriel par le vecteur $M(a, b)$, d’où :

$$(43) \quad T(u, v) = \begin{pmatrix} X(a, b) + uX'_u(a, b) + vX'_v(a, b) \\ Y(a, b) + uY'_u(a, b) + vY'_v(a, b) \\ Z(a, b) + uZ'_u(a, b) + vZ'_v(a, b) \end{pmatrix}, \quad (u, v) \in \mathbf{R} \times \mathbf{R}.$$

Exercice 40 † Exprimer les conditions de continuité du premier et du second ordre entre deux surfaces de Bézier bicubiques (pensez à utiliser les résultats obtenus sur les courbes).

Remarque 7. Notons que le bord d’une surface de Bézier est une courbe de Bézier, les conditions (31) et (34) sur les points de jonctions sur ce bord sont donc strictement identiques à la situation plane. En effet l’ajout d’une troisième coordonnée à un point de contrôle ne fait que définir une courbe dans l’espace plutôt qu’une courbe plane.