

Plan_Projectif

June 9, 2023

1 Plans projectifs

Marc Lorenzi

5 juin 2023

```
[1]: import matplotlib.pyplot as plt
import math
```

1.1 1. Introduction

1.1.1 1.1 Qu'est-ce qu'un plan projectif ?

L'un des premiers objectifs de la géométrie est de parler de points et de droites. Tout le monde sait, ou croit savoir ce que sont un point ou une droite. En réalité, on n'a pas besoin de savoir ce que c'est : c'est la clé de *l'approche axiomatique*. Pour faire de la géométrie, il suffit de posséder un ensemble P dont les éléments sont appelés *points*, un ensemble D dont les éléments sont appelés *droites*, et une relation entre les points et les droites, la relation *d'incidence* (intuitivement le point est sur la droite, ou la droite passe par le point). On se donne ensuite un certain nombre d'axiomes vérifiés par la relation d'incidence.

Il existe plusieurs systèmes d'axiomes possibles, conduisant à des géométries différentes. La géométrie qui nous intéressera ici est la *géométrie projective plane*. Voici les axiomes.

Définition. Un *plan projectif* est un triplet $\mathbb{P} = (P, D, I)$ où P et D sont deux ensembles et $I \subseteq P \times D$ vérifiant les propriétés suivantes.

(P1) Pour tous $p, p' \in P$ distincts, il existe un unique $d \in D$ tel que pId et $p'Id$.

(P2) Pour tous $d, d' \in D$ distincts, il existe un unique $p \in P$ tel que pId et pId' .

(P3) Il existe un ensemble $Q \subseteq P$ de cardinal 4 tel que pour tout $d \in D$, il existe au plus deux éléments p de Q tels que pId .

Les éléments de P sont appelés les *points*, ceux de D sont les *droites*. Lorsque pId , on dit que p est *incident* à d ou que d est *incidente* à p .

Commentaires. L'axiome P1 est un grand classique. En géométrie élémentaire il s'énonce : « par deux points il passe une unique droite ». L'axiome P2 est un peu plus surprenant : deux droites se coupent en un unique point. Ainsi, en géométrie projective, le parallélisme n'existe pas. L'axiome

$P3$ est là pour éviter des cas dégénérés. L'ensemble Q dont l'existence est assurée par cet axiome s'appelle un *quadrangle*.

Dans ce notebook, nous nous focaliserons sur les plans projectifs *finis*, c'est à dire ceux pour lesquels l'ensemble P est fini. Plusieurs questions se posent.

1. Quels sont les entiers m pour lesquels il existe un plan projectif ayant m points ?
2. Comment construire un plan projectif fini ?
3. Si un plan projectif possède m points, combien possède-t-il de droites ?
4. Combien de points sont incidents à une droite donnée ? Combien de droites sont incidentes à un point donné ?

On ne connaît pas, à l'heure actuelle la réponse à la question 1.

Nous allons nous intéresser dans ce notebook à une famille de plans projectifs fabriqués à partir du corps $\mathbb{Z}/n\mathbb{Z}$ des entiers modulo n , où n est un nombre premier. Dans ce cas particulier, nous allons pouvoir répondre à toutes les questions.

1.1.2 1.2 Le plan de Fano

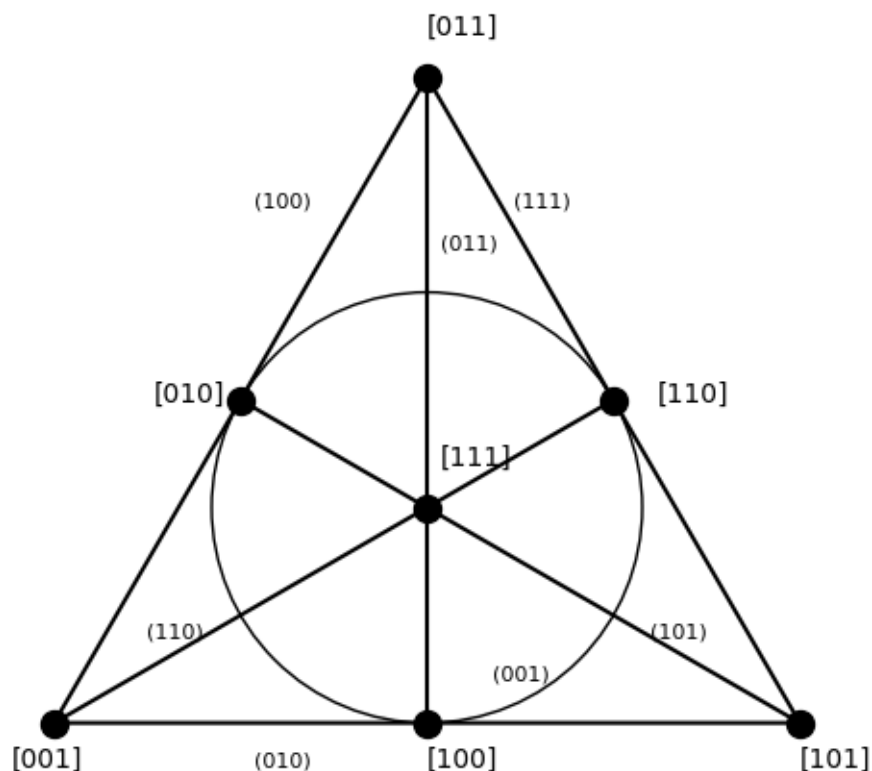
Commençons par examiner un exemple très simple (en fait le plus simple) de plan projectif fini, le *plan de Fano*. En voici une représentation graphique.

```
[2]: def plot_fano():
    r3 = math.sqrt(3) / 2
    x1, x2, x3 = -r3, r3, 0
    y1, y2, y3 = -1/2, -1/2, 1
    x4, x5, x6 = (x1 + x2) / 2, (x1 + x3) / 2, (x2 + x3) / 2
    y4, y5, y6 = (y1 + y2) / 2, (y1 + y3) / 2, (y2 + y3) / 2
    xs = [x1, x2, x3, 0, x4, x5, x6]
    ys = [y1, y2, y3, 0, y4, y5, y6]
    plt.plot(xs, ys, 'ok', ms=10)
    plt.plot([x1, x2, x3, x1], [y1, y2, y3, y1], 'k')
    plt.plot([x1, x6], [y1, y6], 'k')
    plt.plot([x2, x5], [y2, y5], 'k')
    plt.plot([x3, x4], [y3, y4], 'k')
    d = math.sqrt(x4 ** 2 + y4 ** 2)
    circle = plt.Circle((0, 0), d, color='k', fill=False)
    plt.gca().add_patch(circle)
    plt.text(x3, y3+.1, '[011]')
    plt.text(x5-.2, y5, '[010]')
    plt.text(x1-.1, y1-.1, '[001]')
    plt.text(x4, y4-.1, '[100]')
    plt.text(x2, y2-.1, '[101]')
    plt.text(x6+.1, y6, '[110]')
    plt.text(.03, .1, '[111]')

    plt.text(0.2, 0.7, '(111)', fontsize=8)
    plt.text(-0.4, 0.7, '(100)', fontsize=8)
```

```
plt.text(-0.4, -0.6, '(010)', fontsize=8)
plt.text(0.03, 0.6, '(011)', fontsize=8)
plt.text(0.52, -0.3, '(101)', fontsize=8)
plt.text(-0.65, -0.3, '(110)', fontsize=8)
plt.text(0.15, -0.4, '(001)', fontsize=8)
plt.axis('off')
plt.axis('equal')
plt.savefig('fano2.png', bbox_inches='tight')
```

```
[3]: plot_fano()
```



Le plan de Fano est $\mathbb{P} = (P, D, I)$ où $P = D = \{001, 010, 011, 100, 101, 110, 111\}$. Il possède donc 7 points et 7 droites. I est la relation d'incidence. Sur le dessin, si $p \in P$ et $d \in D$ on a pId lorsque le point p est sur la ligne d . Par exemple, $[111]I(101)$.

Sur le dessin, on a conventionnellement représenté les points par des ... points, et les droites par des lignes (6 segments et 1 cercle). Les droites sont étiquetées par leur valeur entre parenthèses, les points par leur valeur entre crochets.

Ce n'est pas évident à voir, mais il y a une « formule » pour la relation d'incidence I . Si p et p' sont deux points distincts, l'unique droite incidente à p et p' est $p \wedge p'$ où $\wedge$ désigne le « produit vectoriel » modulo 2. Par exemple, l'unique droite incidente à $[001]$ et $[101]$ est (010) car

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \wedge \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

De même, Si d et d' sont deux droites distinctes, l'unique point incident à d et d' est $d \wedge d'$. Par exemple, l'unique point incident à (001) et (101) est $[010]$.

Chaque point du plan de Fano est incident à 3 droites, et chaque droite est incidente à 3 points. Si p et p' sont deux points distincts incidents à une droite d , le troisième point de d est $p + p'$, où l'addition est faite bit par bit modulo 2. De même, si d et d' sont deux droites distinctes incidentes à un point p , la troisième droite incidente à p est $d + d'$. Par exemple,

$$001 + 101 = 100$$

Ainsi,

- Les points $[001]$, $[101]$ et $[100]$ sont les 3 points incidents à la droite (010) .
- Les droites (001) , (101) et (100) sont les 3 droites incidentes au point $[010]$.

Pour l'axiome $P3$, on peut par exemple considérer l'ensemble

$$Q = \{[001], [101], [011], [111]\}$$

Chaque droite autre que (001) est incidente à exactement deux points de Q . La droite (001) n'est incidente à aucun point de Q . L'ensemble Q vérifie donc $P3$.

1.1.3 1.3 Notations

Dans ce notebook, n désigne un nombre premier.

On pose également $m = n^2 + n + 1$. L'entier m apparaîtra régulièrement.

Nous noterons $\mathbb{F} = \mathbb{Z}/n\mathbb{Z}$ le corps des entiers modulo n . L'ensemble $\mathbb{F}^3$ est un $\mathbb{F}$ -espace vectoriel de dimension 3. Comme $\mathbb{F}$ est fini, de cardinal n , $\mathbb{F}^3$ est de cardinal n^3 .

Enfin, nous poserons $E = \mathbb{F}^3 \setminus \{0\}$.

1.1.4 1.4 Le corps $\mathbb{F}$ en Python

Avant de parler de plans projectifs, mettons en place des outils pour manipuler facilement les éléments de $\mathbb{F}$. Écrivons une classe `F`. Un objet de cette classe sera alors créé par une instruction du genre `x = F(4)`, pour créer la classe de 4 dans $\mathbb{Z}/11\mathbb{Z}$.

Comme la classe `F` dépend du nombre premier n , nous allons en fait écrire une fonction `corps_fini` qui prend l'entier n en paramètre et qui renvoie la classe `F`.

Le calcul de l'inverse d'un élément de $\mathbb{F}$ peut être effectué au moyen de l'algorithme d'Euclide étendu, effectué dans la fonction `bezout`. Cette dernière prend en paramètres deux entiers relatifs. Elle renvoie un triplet (u, v, d) tel que $ua + vb = d$ et $d = a \wedge b$.

```
[4]: def bezout(a, b):
    u1, v1, d1 = 1, 0, a
    u2, v2, d2 = 0, 1, b
    while d2 != 0:
        q = d1 // d2
        u, v, d = u1 - q * u2, v1 - q * v2, d1 - q * d2
        u1, v1, d1 = u2, v2, d2
        u2, v2, d2 = u, v, d
    return u1, v1, d1
```

```
[5]: bezout(123, 457)
```

```
[5]: (-26, 7, 1)
```

```
[6]: -26 * 123 + 7 * 457
```

```
[6]: 1
```

La fonction `inv_mod` prend en paramètres deux entiers a et m tels que $a \wedge m = 1$. Elle renvoie un entier u tel que $ua \equiv 1 \pmod{m}$.

```
[7]: def inv_mod(a, m):
    u, v, d = bezout(a, m)
    if d != 1: raise Exception('non inversible')
    else: return u % m
```

```
[8]: inv_mod(123, 457)
```

```
[8]: 431
```

```
[9]: (123 * 431) % 457
```

```
[9]: 1
```

La fonction `corps_fini` prend en paramètre un nombre premier n . Elle renvoie la classe `F` qui permet la manipulation des entiers modulo n . On y a prévu toutes les opérations élémentaires dans $\mathbb{F}$ (addition, soustraction, multiplication, division). On a également prévu de pouvoir opérer entre un objet de la classe `F` et un entier (un objet de type `int`).

```
[10]: def corps_fini(n):

    class F:

        modulo = n

        def __init__(self, a): self.val = a % F.modulo

        def __repr__(self):
```

```

        return str(self.val)

    def __eq__(self, y):
        if isinstance(y, int): y = F(y)
        return self.val == y.val

    def __req__(self, y): return self == y

    def __add__(self, y):
        if isinstance(y, int): y = F(y)
        x = (self.val + y.val) % F.modulo
        return F(x)

    def __radd__(self, y): return self + y

    def __neg__(self):
        x = (-self.val) % F.modulo
        return F(x)

    def __sub__(self, y):
        return self + (-y)

    def __rsub__(self, y): return -(self - y)

    def __mul__(self, y):
        if isinstance(y, int): y = F(y)
        x = (self.val * y.val) % F.modulo
        return F(x)

    def __rmul__(self, y): return self * y

    def inverse(self):
        z = inv_mod(self.val, F.modulo)
        return F(z)

    def __truediv__(self, y):
        if isinstance(y, int): y = F(y)
        return self * y.inverse()

    def __rtruediv__(self, y): return F(y) / self

return F

```

Voici par exemple les tables de multiplication et de division dans $\mathbb{Z}/13\mathbb{Z}$.

```
[11]: F = corps_fini(13)
```

```
[12]: n = 13
for x in range(n):
    for y in range(n):
        print('%3s' % (F(x) * F(y)), end='')
    print()
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9 10 11 12
0 2 4 6 8 10 12 1 3 5 7 9 11
0 3 6 9 12 2 5 8 11 1 4 7 10
0 4 8 12 3 7 11 2 6 10 1 5 9
0 5 10 2 7 12 4 9 1 6 11 3 8
0 6 12 5 11 4 10 3 9 2 8 1 7
0 7 1 8 2 9 3 10 4 11 5 12 6
0 8 3 11 6 1 9 4 12 7 2 10 5
0 9 5 1 10 6 2 11 7 3 12 8 4
0 10 7 4 1 11 8 5 2 12 9 6 3
0 11 9 7 5 3 1 12 10 8 6 4 2
0 12 11 10 9 8 7 6 5 4 3 2 1
```

```
[13]: n = 13
for x in range(1, n):
    for y in range(1, n):
        print('%3s' % (F(x) / F(y)), end='')
    print()
```

```
1 7 9 10 8 11 2 5 3 4 6 12
2 1 5 7 3 9 4 10 6 8 12 11
3 8 1 4 11 7 6 2 9 12 5 10
4 2 10 1 6 5 8 7 12 3 11 9
5 9 6 11 1 3 10 12 2 7 4 8
6 3 2 8 9 1 12 4 5 11 10 7
7 10 11 5 4 12 1 9 8 2 3 6
8 4 7 2 12 10 3 1 11 6 9 5
9 11 3 12 7 8 5 6 1 10 2 4
10 5 12 9 2 6 7 11 4 1 8 3
11 12 8 6 10 4 9 3 7 5 1 2
12 6 4 3 5 2 11 8 10 9 7 1
```

Remarquer les 1 et les 12 sur les deux diagonales de la table de division. On a en effet pour tout $x \in \mathbb{Z}/n\mathbb{Z}$ non nul, $\frac{x}{x} = 1$ et $\frac{-x}{x} = -1$.

1.2 2. Points

1.2.1 2.1 Une relation d'équivalence

Définissons sur $E = \mathbb{F}^3 \setminus \{0\}$ une relation $\simeq$ en posant

$$u \simeq v \iff \exists \lambda \in \mathbb{F}^*, v = \lambda u$$

Proposition. $\simeq$ est une relation d'équivalence sur E .

Démonstration.

- Réflexivité. Soit $u \in E$. On a $u = 1u$, donc $u \simeq u$.
- Symétrie. Soient $u, v \in E$. Supposons $u \simeq v$. Il existe $\lambda \in \mathbb{F}^*$ tels que $v = \lambda u$. On a donc $u = \frac{1}{\lambda}v$, donc $v \simeq u$.
- Transitivité. Soient $u, v, w \in E$. Supposons $u \simeq v$ et $v \simeq w$. Il existe $\lambda, \mu \in \mathbb{F}$ tels que $v = \lambda u$ et $w = \mu v$. On a donc $w = (\mu\lambda)u$, donc $u \simeq w$.

Définition. Un *point* est un élément de l'ensemble quotient $P = E / \simeq$.

Un point p est un ensemble de triplets d'éléments de $\mathbb{F}$ pas tous nuls. Précisément, il existe $u \in E$ tel que

$$p = \{\lambda u : \lambda \in [1, n-1]\}$$

Un point est ainsi un ensemble fini de cardinal $n-1$.

Proposition. $|P| = n^2 + n + 1$.

Démonstration. Comme $\simeq$ est une relation d'équivalence, l'ensemble P des points est une partition de E . On a donc

$$E = \bigcup_{p \in P} p$$

Cette union est disjointe (partition) donc

$$|E| = \sum_{p \in P} |p|$$

L'ensemble E est de cardinal $n^3 - 1 = (n-1)(n^2 + n + 1)$ et chaque point est un ensemble de cardinal $n-1$. On a donc

$$(n-1)(n^2 + n + 1) = \sum_{p \in P} (n-1) = (n-1)|P|$$

d'où le résultat en divisant par $n-1$.

```
[14]: def card_P(n):
      return n ** 2 + n + 1

[15]: for n in [2, 3, 5, 7, 11, 13]:
      print('%3d%5d' % (n, card_P(n)))
```


2	7
3	13
5	31
7	57
11	133
13	183

1.2.2 2.2 Représentant normalisé

Soit $p \in P$. Il existe $u \in E$ tel que

$$p = \{\lambda u : \lambda \in [1, n-1]\}$$

Nous noterons $p = [u]$. Si $u = (x, y, z)$, nous écrirons $[x, y, z]$ au lieu de $[(x, y, z)]$.

Il n'y a pas unicité du triplet (x, y, z) . Le point p possède donc, d'une certaine manière, $p-1$ triplets de coordonnées. Aucun de ces triplets n'est préférable à un autre, mais nous pouvons décider d'un représentant particulier. Choisissons la méthode suivante :

- Si $z \neq 0$, choisissons $(x/z, y/z, 1)$.
- Si $z = 0$ et $y \neq 0$, choisissons $(x/y, 1, 0)$.
- Si $z = 0$ et $y = 0$, alors $x \neq 0$. Choisissons $(1, 0, 0)$.

Il est maintenant facile de décider si deux points sont égaux. C'est le cas si et seulement si ils ont le même représentant normalisé.

1.2.3 2.3 Les points en Python

Voici la classe DP. Pourquoi ce nom ? Lorsque nous aurons défini les droites la raison sera évidente. Lors de l'appel du constructeur, le point est normalisé, ce qui permet de tester facilement si deux points sont égaux.

```
[16]: class DP:

    def __init__(self, F, x, y, z):
        self.corps = F
        if isinstance(x, int): self.x = self.corps(x)
        else: self.x = x
        if isinstance(y, int): self.y = self.corps(y)
        else: self.y = y
        if isinstance(z, int): self.z = self.corps(z)
        else: self.z = z
        self.normaliser()

    def __repr__(self):
        return ' [%s,%s,%s]' % (self.x, self.y, self.z)

    def __eq__(self, q):
```

```

    return self.x == q.x and self.y == q.y and self.z == q.z

def normaliser(self):
    if self.z != 0:
        self.x = self.x / self.z
        self.y = self.y / self.z
        self.z = self.corps(1)
    elif self.y != 0:
        self.x = self.x / self.y
        self.y = self.corps(1)
    else:
        self.x = self.corps(1)

def vecteur(self):

    return (self.x, self.y, self.z)

```

```

[17]: F = corps_fini(7)
      p = DP(F, 1, 2, 0)
      q = DP(F, 4, 1, 4)
      p == q

```

[17]: False

```

[18]: p = DP(F, 5, 3, 5)
      q = DP(F, 4, 1, 4)
      p == q

```

[18]: True

1.2.4 2.4 Énumérer les points

La fonction `enum_points` énumère les éléments de P . Cette énumération établit un ordre conventionnel entre les points.

- On énumère d'abord les n^2 points de la forme $[x, y, 1]$, dans l'ordre lexicographique du couple (x, y) :

$$(0, 0), (0, 1), \dots, (0, n-1), (1, 0), \dots, (n-1, n-1)$$

- On énumère ensuite les points de la forme $[x, 1, 0]$.
- On termine par le point $[1, 0, 0]$.

```

[19]: def enum_points(F):
      n = F.modulo
      for x in range(n):

```

```

    for y in range(n):
        yield DP(F, x, y, 1)
for x in range(n):
    yield DP(F, x, 1, 0)
yield DP(F, 1, 0, 0)

```

```

[20]: F = corps_fini(5)
      for p in enum_points(F): print(p, end=' ')

```

```

[0,0,1] [0,1,1] [0,2,1] [0,3,1] [0,4,1] [1,0,1] [1,1,1] [1,2,1] [1,3,1] [1,4,1]
[2,0,1] [2,1,1] [2,2,1] [2,3,1] [2,4,1] [3,0,1] [3,1,1] [3,2,1] [3,3,1] [3,4,1]
[4,0,1] [4,1,1] [4,2,1] [4,3,1] [4,4,1] [0,1,0] [1,1,0] [2,1,0] [3,1,0] [4,1,0]
[1,0,0]

```

Il est également facile, étant donné un entier $k \in [0, n^2 + n]$ de trouver quel est le k ième point de l'énumération ci-dessus.

- Si $k < n^2$, c'est $[x, y, 1]$ où $x = \lfloor \frac{k}{n} \rfloor$ et $y = k \bmod n$.
- Si $n^2 \leq k < n^2 + n$, c'est $[k - n^2, 1, 0]$
- Enfin, si $k = n^2 + n$, c'est $[1, 0, 0]$.

```

[21]: def point_numero(F, k):
      n = F.modulo
      n2 = n ** 2
      if k < n2:
          x = k // n
          y = k % n
          return DP(F, x, y, 1)
      elif k < n2 + n:
          return DP(F, k - n2, 1, 0)
      else:
          return DP(F, 1, 0, 0)

```

```

[22]: F = corps_fini(5)
      for k in range(card_P(5)):
          print(point_numero(F, k), end=' ')

```

```

[0,0,1] [0,1,1] [0,2,1] [0,3,1] [0,4,1] [1,0,1] [1,1,1] [1,2,1] [1,3,1] [1,4,1]
[2,0,1] [2,1,1] [2,2,1] [2,3,1] [2,4,1] [3,0,1] [3,1,1] [3,2,1] [3,3,1] [3,4,1]
[4,0,1] [4,1,1] [4,2,1] [4,3,1] [4,4,1] [0,1,0] [1,1,0] [2,1,0] [3,1,0] [4,1,0]
[1,0,0]

```

On retrouve évidemment la même chose qu'avec l'énumération.

1.3 3. Droites

1.3.1 3.1 Introduction

Définition. Une droite est un élément de P .

Cette définition peut au premier abord paraître terrible. Une droite, c'est donc un point ? Eh bien oui. Pour commencer, posons gratuitement $D = P$.

On comprend mieux maintenant la notation `DP` pour notre classe. les objets de cette classe sont des points, mais aussi des droites. Inutile, donc d'écrire du code pour manipuler les droites.

Proposition. $|D| = n^2 + n + 1$.

Démonstration. Évidemment, vu que $D = P$.

```
[23]: F = corps_fini(7)
      d1 = DP(F, 1, 2, 2)
      d2 = DP(F, 4, 1, 1)
      d1 == d2
```

```
[23]: True
```

1.3.2 3.2 Incidence

Nous allons définir une relation I entre les points et les droites, la *relation d'incidence*.

Notation. Pour tous $u = (x, y, z)$ et $v = (x', y', z') \in \mathbb{F}^3$,

$$\langle u, v \rangle = xx' + yy' + zz'$$

On note $u \perp v$ lorsque $\langle u, v \rangle = 0$.

Proposition. Pour tous $u, v \in \mathbb{F}^3$ et $\lambda, \mu \in \mathbb{F}^*$, $u \perp v$ si et seulement si $(\lambda u) \perp (\mu v)$.

Démonstration. C'est évident, car $\langle \lambda u, \mu v \rangle = \lambda \mu \langle u, v \rangle$.

Cette propriété montre que la relation $\perp$ « passe au quotient ». Elle justifie la définition suivante.

Définition. Soient $p = [u] \in P$ et $d = [v] \in D$. Le point p est incident à la droite d si $u \perp v$.

Si $p = [x, y, z]$ et $d = [a, b, c]$, on a donc pId si et seulement si

$$ax + by + cz = 0$$

Remarque. L'égalité ci-dessus est symétrique en p et d . Nous dirons donc aussi que la droite d est incidente au point p .

```
[24]: def prod_scal(u, v):
      x1, y1, z1 = u
      x2, y2, z2 = v
```

```
return x1 * x2 + y1 * y2 + z1 * z2
```

```
[25]: def incidents(p, d):
      return prod_scal(p.vecteur(), d.vecteur()) == 0
```

```
[26]: F = corps_fini(7)
      p = DP(F, 2, 3, 3)
      d = DP(F, 3, 1, 4)
      incidents(p, d)
```

```
[26]: True
```

1.4 4. Le plan projectif

Définition. Le *plan projectif* est $\mathbb{P} = (P, D, I)$.

Nous allons évidemment prouver que tous les axiomes de plan projectif sont vérifiés par $\mathbb{P}$.

1.4.1 4.1 La matrice d'incidence

Rappelons que $m = n^2 + n + 1$. Soient $p_0, \dots, p_{m-1}$ une énumération de P et $d_0, \dots, d_{m-1}$ une énumération de D . La matrice d'incidence de $\mathbb{P}$ (relative à ces deux énumérations) est la matrice $A \in \mathcal{M}_n(\mathbb{Z})$ définie pour $i, j \in [0, m-1]$ par $A_{ij} = 1$ si p_i est incident à d_j et $A_{ij} = 0$ sinon.

```
[27]: def matrice_incidence(F):
      n = F.modulo
      m = card_P(n)
      A = m * [None]
      for i in range(m): A[i] = m * [0]
      for i in range(m):
          for j in range(m):
              if incidents(point_numero(F, i), point_numero(F, j)):
                  A[i][j] = 1
      return A
```

```
[28]: A = matrice_incidence(corps_fini(3))
      A
```

```
[28]: [[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1],
      [0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1],
      [0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1],
      [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0],
      [0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0],
      [0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0],
```

```

[0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0],
[1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0],
[1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0],
[1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]]

```

1.4.2 4.2 Dénombrements

Regardons la matrice d'incidence de $\mathbb{P}$ que nous avons obtenue pour $n = 3$. Elle possède quatre 1 sur chaque ligne et chaque colonne : chaque point est incident à 4 droites et chaque droite est incidente à 4 points. Ceci se généralise facilement.

Combien y a-t-il de points incidents à une droite donnée (ou, ce qui revient au même, de droites incidentes à un point donné) ?

Donnons-nous une droite $d = [a, b, c]$ où $(a, b, c) \in E$. Les points incidents à d sont les points $[x, y, z]$ vérifiant $ax + by + cz = 0$. Considérons le plan F de l'espace vectoriel $\mathbb{F}^3$ d'équation

$$(F) \quad aX + bY + cZ = 0$$

F est un $\mathbb{F}$ -espace vectoriel de dimension 2, il est donc isomorphe à $\mathbb{F}^2$, qui est de cardinal n^2 . Ainsi, $|F| = n^2$. En enlevant le triplet $(0, 0, 0)$, on a donc

$$|F \setminus \{0\}| = n^2 - 1 = (n - 1)(n + 1)$$

Cet ensemble est la réunion disjointe de classes modulo $\simeq$, chacune de cardinal $n - 1$. Il est donc l'union de $n + 1$ classes, c'est à dire de $n + 1$ points de $\mathbb{P}$. Nous venons de montrer le résultat suivant.

Proposition. Pour toute droite d de $\mathbb{P}$, d est incidente à $n + 1$ points.

On a bien sûr gratuitement la proposition sur les points.

Proposition. Pour tout point p de $\mathbb{P}$, p est incident à $n + 1$ droites.

1.4.3 4.3 Points incidents à deux droites distinctes

En géométrie affine, deux droites distinctes peuvent s'intersecter en un unique point, ou alors avoir une intersection vide si elles sont parallèles. En géométrie projective, le parallélisme n'existe pas.

Proposition. Soient d et d' deux droites distinctes. Il existe un unique point incident à d et d' .

Démonstration. Posons $d = [a, b, c]$ et $d' = [a', b', c']$. Considérons les plans de $\mathbb{F}^3$ d'équations

$$(F) \quad ax + by + cz = 0$$

$$(F') \quad a'x + b'y + c'z = 0$$

Les droites d et d' étant distinctes, les plans F et F' sont distincts. Leur intersection $F \cap F'$ est donc une droite (au sens usuel dans l'espace vectoriel $\mathbb{F}^3$) δ . Il existe $(x_0, y_0, z_0) \in \mathbb{F}^3 \setminus \{0\}$ tel que

$$\delta = \{\lambda(x_0, y_0, z_0) : \lambda \in \mathbb{F}\}$$

En enlevant le triplet $(0,0,0)$, on constate que $\delta \setminus \{0\}$ est un point p de $\mathbb{P}$. Ainsi, d et d' sont incidentes à un et un seul point, le point p .

Notation. Nous noterons $p = d \cap d'$. Nous appellerons p *l'intersection* de d et d' .

Le calcul de $d \cap d'$ se fait facilement par un « produit vectoriel » : $d \cap d' = p$ où

$$p = [yz' - y'z, zx' - z'x, xy' - x'y]$$

```
[29]: def prod_vect(u, v):
      x1, y1, z1 = u
      x2, y2, z2 = v
      return (y1 * z2 - y2 * z1, z1 * x2 - z2 * x1, x1 * y2 - x2 * y1)
```

```
[30]: def intersection(F, d1, d2):
      x, y, z = prod_vect(d1.vecteur(), d2.vecteur())
      return DP(F, x, y, z)
```

```
[31]: F = corps_fini(7)
      d1 = DP(F, 2, 1, 2)
      d2 = DP(F, 3, 1, 1)
      print(intersection(F, d1, d2))
```

[1,3,1]

1.4.4 4.4 Droites incidentes à deux points distincts

Dans notre plan projectif, il y a symétrie complète entre les points et les droites. Nous avons donc gratuitement le théorème suivant.

Proposition. Soient p et p' deux points distincts de $\mathbb{P}$. Il existe une unique droite incidente à p et p' .

Notation. Nous noterons (pp') l'unique droite incidente à p et p' .

```
[32]: def droite_incidente(F, p1, p2):
      return intersection(F, p1, p2)
```

Voici la liste des 21 paires de points distincts du plan de Fano, avec l'unique droite qui leur est incidente.

```
[33]: def str_fano(p):
      return str(p.x) + str(p.y) + str(p.z)
```

```
[34]: def str_point(p):
      return '[' + str_fano(p) + '']
```

```
[35]: def str_droite(d):
      return '(' + str_fano(d) + '')
```

```
[36]: n = 2
      F = corps_fini(n)
      m = card_P(n)
      for i in range(m):
          p1 = point_numero(F, i)
          for j in range(i + 1, m):
              p2 = point_numero(F, j)
              d = droite_incidente(F, p1, p2)
              print(str_point(p1), end=' ')
              print(str_point(p2), end=' ')
              print(str_droite(d))
```

```
[001] [011] (100)
[001] [101] (010)
[001] [111] (110)
[001] [010] (100)
[001] [110] (110)
[001] [100] (010)
[011] [101] (111)
[011] [111] (011)
[011] [010] (100)
[011] [110] (111)
[011] [100] (011)
[101] [111] (101)
[101] [010] (101)
[101] [110] (111)
[101] [100] (010)
[111] [010] (101)
[111] [110] (110)
[111] [100] (011)
[010] [110] (001)
[010] [100] (001)
[110] [100] (001)
```

1.4.5 4.5 Retour à la matrice d'incidence.

Soit A la matrice d'incidence de $\mathbb{P}$. Rappelons que $m = n^2 + n + 1$.

Proposition. $AA^T = nI_m + J_m$ où I_m est la matrice identité d'ordre m et J_m est la matrice de taille m qui ne contient des 1.

Démonstration. Soient $i, j \in [0, m - 1]$. Pour tout $k \in [0, m - 1]$, on a $A_{ik}A_{jk} = 1$ si p_i et p_j

sont incidents à d_k , et 0 sinon. Ainsi, $\sum_{k=0}^{m-1} A_{ik}A_{jk}$ est le nombre de droites incidentes à p_i et p_j .

- Cas 1, $i = j$. p_i est incident à exactement $n + 1$ droites, donc

$$\sum_{k=0}^{m-1} A_{ik}A_{jk} = n + 1$$

- Cas 2, $i \neq j$. p_i et p_j sont incidents à exactement une droite, donc

$$\sum_{k=0}^{m-1} A_{ik}A_{jk} = 1$$

Finalement,

$$AA^T = \begin{pmatrix} n+1 & 1 & 1 & \dots & 1 \\ 1 & n+1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & & \vdots \\ 1 & 1 & \dots & n+1 & 1 \\ 1 & 1 & \dots & 1 & n+1 \end{pmatrix}$$

Faisons un petit test.

```
[37]: def prod_mat(A, B):
    p, q = len(A), len(A[0])
    q1, r = len(B), len(B[0])
    assert q == q1
    C = p * [None]
    for i in range(p):
        C[i] = r * [0]
    for i in range(p):
        for j in range(r):
            for k in range(q):
                C[i][j] += A[i][k] * B[k][j]
    return C
```

```
[38]: def transposee(A):
    p, q = len(A), len(A[0])
    C = q * [None]
    for i in range(q):
        C[i] = p * [0]
    for i in range(q):
        for j in range(p):
            C[i][j] = A[j][i]
    return C
```

```
[39]: A = matrice_incidence(corps_fini(3))
      prod_mat(A, transposee(A))
```

```
[39]: [[4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
      [1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
      [1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
      [1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1],
      [1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1],
      [1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 1],
      [1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1],
      [1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1],
      [1, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1],
      [1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1],
      [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1],
      [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 1],
      [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4]]
```

Proposition. $\det(A) = \pm(n+1)n^{\frac{1}{2}n(n+1)}$.

Démonstration. Soit $B = AA^T \in \mathcal{M}_m(\mathbb{Z})$. On a

$$\det(B) = \begin{vmatrix} n+1 & 1 & 1 & \dots & 1 \\ 1 & n+1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & & \vdots \\ 1 & 1 & \dots & n+1 & 1 \\ 1 & 1 & \dots & 1 & n+1 \end{vmatrix}$$

Soustrayons la première ligne à toutes les autres pour obtenir

$$\det(B) = \begin{vmatrix} n+1 & 1 & 1 & \dots & 1 \\ -n & n & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ -n & 0 & \dots & n & 0 \\ -n & 0 & \dots & 0 & n \end{vmatrix}$$

Ajoutons à la première colonne la somme de toutes les autres. Comme $m+n = (n+1)^2$,

$$\det(B) = \begin{vmatrix} (n+1)^2 & 1 & 1 & \dots & 1 \\ 0 & n & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & n & 0 \\ 0 & 0 & \dots & 0 & n \end{vmatrix}$$

On a là un déterminant de matrice triangulaire. On en déduit

$$\det(B) = (n+1)^2 n^{m-1} = (n+1)^2 n^{n(n+1)}$$

Or, $\det(B) = \det(A)^2$, d'où le résultat.

Remarque. Le signe exact de $\det(A)$ n'a pas grand intérêt. En effet la matrice A n'est définie qu'à condition d'avoir fixé une énumération des éléments de P et de D . Un changement d'énumération conduit à des échanges de lignes et de colonnes de la matrice, et donc à un changement de signe de son déterminant.

```
[40]: def detA(n):  
      return (n + 1) * n ** (n * (n + 1) // 2)
```

```
[41]: for n in [2, 3, 5, 7]:  
      print('%3d %d' % (n, detA(n)))
```

```
2 24  
3 2916  
5 183105468750  
7 3679892292357919687814408
```

1.4.6 4.6 Quadrangles et quadrilatères

Il nous reste à montrer que $\mathbb{P}$ vérifie l'axiome $P3$ des plans projectifs.

Définition. Un *quadrangle* est un ensemble Q de 4 points tel que toute droite est incidente à au plus deux points de Q .

Définition. Un *quadrilatère* est un ensemble Q de 4 droites tel que tout point est incident à au plus deux droites de Q .

Proposition. Il existe un quadrangle et un quadrilatère.

Démonstration. Étant données les symétries entre points et droites, l'existence d'un quadrangle équivaut à celle d'un quadrilatère. Concentrons nous sur les quadrangles. Considérons les 4 points $p_1 = [0, 0, 1]$, $p_2 = [1, 0, 1]$, $p_3 = [0, 1, 1]$ et $p_4 = [1, 1, 1]$. Montrons que l'ensemble Q de ces 4 points est un quadrangle. Soit $d = [a, b, c]$ une droite, où $(a, b, c) \neq (0, 0, 0)$.

Supposons que d est incidente à, par exemple, p_2 , p_3 et p_4 . On a alors

$$\begin{cases} a + c & = & 0 \\ b + c & = & 0 \\ a + b + c & = & 0 \end{cases}$$

d'où, facilement, $a = b = c = 0$. Contradiction. On laisse en exercice les trois autres cas, tout aussi impossibles.

1.5 5. Conclusion

Nous avons montré au fil de ce notebook que le triplet $\mathbb{P}$ créé à partir du corps $\mathbb{Z}/n\mathbb{Z}$ était un plan projectif fini. En réalité, la construction que nous avons faite et les théorèmes que nous avons montré restent valables en remplaçant $\mathbb{F}$ par n'importe quel corps fini.

Existe-t-il des plans projectifs autres que ceux-ci ? La réponse est oui. Dans un plan projectif fini quelconque (c'est à dire vérifiant les axiomes $P1$, $P2$ et $P3$), on peut montrer la propriété suivante.

Proposition. Soit $\mathbb{P} = (P, D, I)$ un plan projectif fini. Il existe un unique entier $n \geq 2$ tels que

- $|P| = |D| = n^2 + n + 1$.
- Pour toute droite d , d est incidente à $n + 1$ points.
- Pour tout point p , p est incident à $n + 1$ droites.

L'entier n est appelé *l'ordre* du plan projectif.

Dans ce notebook, nous avons montré que pour tout n premier, il existe au moins un plan projectif d'ordre n . En étendant nos résultats à des corps finis quelconques, on peut montrer que pour tout n premier et tout $k \geq 1$, il existe un plan projectif d'ordre n^k . Les entiers $n \leq 14$ pour lesquels nous avons montré qu'il existe un plan projectif d'ordre n sont donc

$$2, 3, 4, 5, 7, 8, 9, 11, 13$$

Manquent à cette liste : 6, 10, 12 et 14.

Citons le théorème suivant.

Théorème de Bruck-Ryser. Soit $n \geq 2$. On suppose que $n \equiv 1 \pmod{4}$ ou $n \equiv 2 \pmod{4}$. S'il existe un plan projectif fini d'ordre n , alors il existe $a, b \in \mathbb{N}$ tels que $n = a^2 + b^2$.

Les entiers 6 et 14 sont congrus à 2 modulo 4, mais ne sont pas la somme de deux carrés. Il n'existe donc pas de plan projectif d'ordre 6 ou 14.

Remarquons que $10 \equiv 2 \pmod{4}$ mais $10 = 3^2 + 1^2$ est une somme de deux carrés. Le théorème de Bruck-Ryser reste donc muet sur le statut de l'entier 10. Lam, Thiel et Swiercz ont démontré en 1989 qu'il n'existe pas de plan projectif d'ordre 10. Leur démonstration traitait un certain nombre de cas avec l'aide d'ordinateurs. Le calcul a pris 800 heures. L'entier 10 est ainsi le plus petit contre-exemple à la réciproque du théorème de Bruck-Ryser.

Comme $12 \equiv 0 \pmod{4}$, le théorème de Bruck-Ryser ne peut pas s'appliquer à $n = 12$. Existe-t-il un plan projectif d'ordre 12 ? La question reste à ce jour une question ouverte.

Bibliographie

- A. Pál, *An Introduction to Finite Projective Planes*, Imperial College, London, 2018
- C.W.H. Lam, L. Thiel, S. Swiercz, *The Non-existence of Finite Projective Planes of Order 10*, Canadian Journal of Mathematics, Vol. XLI, No 6, pp 1117-1123, 1989
- R.H. Bruck, H.J. Ryser, *The Nonexistence of Certain Finite Projective Planes*, Canadian Journal of Mathematics, Vol. I, pp. 88-93, 1948