On veut illustrer un théorème dû à Gauss. Soit $p \in \mathbb{N}^*$ un nombre premier vérifiant $p \equiv 1 \pmod 3$; alors, la première partie (assez facile) du théorème de Gauss affirme que $4p$ s'écrit $t^2 + 27u^2$ et cette écriture est unique si l'on impose $t \equiv 2 \pmod 3$ et $u > 0$. Cette écriture est liée à la factorisation de p dans $\mathbb{Z}[j]$, $p = \pi\overline{\pi}$; on peut imposer $\pi \equiv 1 \pmod 3$: en posant $\pi = x + jy$, on a $\pi\overline{\pi} = x^2 - xy + y^2$ donc $4\pi\overline{\pi} = (2x - y)^2 + 3y^2$, $t = 2x - y = \pi + \overline{\pi} \equiv 2 \pmod 3$ et $y \equiv 0 \pmod 3$: c'est l'écriture $4p = t^2 + 27u^2$. La deuxième partie du théorème de Gauss (le coeur du théorème) affirme que le nombre de points de la cubique projective $\{(x : y : z) \in \mathbb{P}_2(\mathbb{F}_p) \mid x^3 + y^3 + z^3 = 0\}$ est égal à $p + 1 - t$.

Sur un corps $\Bbbk$ de caractéristique distincte de 3, les deux cubiques projectives suivantes :

$$x^3 + y^3 + z^3 = 0, \qquad Y^2Z - YZ^2 = X^3 - 7Z^3 \quad (\text{en } \{Z = 1\}\text{-affine : } Y^2 - Y = X^3 - 7)$$

sont isomorphes. Cela vient du fait que les deux applications linéaires $\Bbbk^3 \to \Bbbk^3$ sont **projectivement** réciproques l'une de l'autre :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} -3z \\ 5x - 4y \\ x + y \end{pmatrix}, \qquad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -Y - 4Z \\ Y - 5Z \\ 3X \end{pmatrix}$$

Il y a une manière d'introduire cela mais c'est trop long à expliquer en quelques lignes. On peut se contenter de faire des vérifications (ce qui n'est certes guère motivant) :

$$A = \begin{pmatrix} 0 & 0 & -3 \\ 5 & -4 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \qquad \det(A) = -3^3, \qquad B = \frac{1}{3}\widetilde{A} = \begin{pmatrix} 0 & -1 & -4 \\ 0 & 1 & -5 \\ 3 & 0 & 0 \end{pmatrix}$$

D'autre part, A, B induisent deux bijections réciproques entre les deux cubiques projectives comme le prouve l'expression de $x^3 + y^3 + z^3$ en fonction de X, Y, Z :

$$x^3 + y^3 + z^3 = -27\left[Y^2Z - YZ^2 - (X^3 - 7Z^3)\right]$$

Pourquoi tout ceci ? Parce que cela va permettre à `magma` de calculer efficacement le nombre de points $(x : y : z) \in \mathbb{P}_2(\mathbb{F}_p)$ vérifiant $x^3 + y^3 + z^3 = 0$ en se ramenant à la courbe elliptique de Weierstrass $Y^2 - Y = X^3 - 7$ (j'ai fait $Z = 1$). Note : sur une cubique de Weierstrass, en particulier $Y^2 - Y = X^3 - 7$, il y a un seul point à l'infini, à savoir le point $(0 : 1 : 0)$.

Dans le programme `magma` ci dessous, on fait calculer le nombre de points de deux manières. Ou bien en utilisant la courbe elliptique sur $\mathbb{F}_p$ d'équation $Y^2 - Y = X^3 - 7$. Ou bien en factorisant p dans $\mathbb{Z}[j]$. Dans cette deuxième méthode, on utilise le fait que la réduction modulo 3 : $\mathbb{Z}[j] \to \mathbb{Z}[j]/\langle 3\rangle$ induit un isomorphisme $U(\mathbb{Z}[j]) = \langle -j\rangle = \{\pm 1, \pm j, \pm j^2\} \simeq U(\mathbb{Z}[j]/\langle 3\rangle)$: ceci se vérifie facilement en considérant la "réduction centrée modulo 3" : $z \bmod 3 = a + bj$ avec $a, b \in \{-1, 0, 1\}$. De ceci, on déduit le fait que pour tout $z \in \mathbb{Z}[j]$ non divisible par $1 - j$ i.e. premier avec 3, il existe z' associé à z vérifiant $z' \equiv 1 \pmod 3$; dans la pratique, on peut prendre $z' = \varepsilon^{-1}z$ où ε est l'inversible de $\mathbb{Z}[j]$ défini par $\varepsilon = z \bmod 3$. Une fois obtenue l'écriture $p = \pi\overline{\pi}$ avec $\pi \equiv 1 \pmod 3$, on tient l'écriture convoitée $4p = t^2 + 27u^2$ avec $t \equiv 2 \pmod 3$.

Note pour moi : on a une inclusion stricte d'anneaux $\mathbb{Z}[\sqrt{-3}] \subsetneq \mathbb{Z}[j]$ et le premier est d'indice 2 dans le second. Mais les normes des éléments des deux anneaux sont les mêmes c'est-à-dire que les deux ensembles $\{u^2 + 3v^2 = N(u + v\sqrt{-3}) \mid u, v \in \mathbb{Z}\}$ et $\{x^2 + xy + y^2 = N(x - jy) \mid x, y \in \mathbb{Z}\}$ sont les mêmes (bien sûr, on peut remplacer $x^2 + xy + y^2$ par $x^2 - xy + y^2 = N(x + jy)$). C'est évident dans un sens car il suffit d'utiliser que $\sqrt{-3} = 2j + 1$ donc :

$$u + v\sqrt{-3} = u + v + 2vj \Rightarrow u^2 + 3v^2 = (u + v)^2 - 2v(u + v) + (2v)^2$$

Dans l'autre sens, on utilise $x + jy = (x - y/2) + \sqrt{-3}/2y$ ce qui donne :

$$x^2 - xy + y^2 = \left(x - \frac{y}{2}\right)^2 + \frac{3}{4}y^2 = \left(y - \frac{x}{2}\right)^2 + \frac{3}{4}x^2$$

Ceci fournit le résultat si x ou y est pair. Si x, y sont impairs, on considère $j(x + jy)$ qui a même norme $x^2 - xy + y^2$ que $x + jy$. Mais $j(x + jy) = jx + j^2y = -y + j(x - y)$ d'où l'identité :

$$x^2 - xy + y^2 = y^2 + y(x - y) + (x - y)^2$$

et on gagne cette fois car $x - y$ est pair.

~/AGREGATION/PROBLEMES/GAUSS_JACOBI/*

```
clear ;

E := EllipticCurve([0, 0, -1, 0, -7]) ;  // y^2 - y = x^3 - 7

Qj<sqrt_moins_3> := QuadraticField(-3) ;  j := (-1 + sqrt_moins_3) / 2 ;

Generer_premier_congru_a_1_modulo_3_superieur_a := function(k)
    // k est un entier. Retourne un premier p >= k verifiant p = 1 mod 3
    p := k + (1-k) mod 3 ;  // de cette facon p = 1 modulo 3
    while not IsPrime(p) do p := p+3 ; end while ;
    return p ;
end function ;

G := Generer_premier_congru_a_1_modulo_3_superieur_a ;  // abreviation

jCoordinates := function(z)
   // z = u + v\/-3  à exprimer en fonction de j = (-1 + \/-3)/2
   // \/-3 = 2j + 1 d'où u + v\/-3 = u + (2j + 1)v = u+v + 2vj
   u, v := Explode(Eltseq(z)) ;
   return Explode(ChangeUniverse([u+v, 2*v], IntegerRing())) ;
end function ;

// Reste modulo 3 centré : 0 -> 0, 1 -> 1, 2 -> -1
mod3 := map <Z -> Z | x :-> case < x_mod_3 | 2 : -1, default : x_mod_3 > where x_mod_3 is x mod 3>
            where Z is IntegerRing() ;


p := G(Random(10^70, 10^75)) ;  "p =", p ;
time  _, pi := NormEquation(Qj, p) ;  /* V2.7 to V2.8.3*/   pi := pi[1] ;
// Il faut faire en sorte que pi = x + jy soit égal à 1 modulo 3

for epsilon in [1, -1, j, -j, j^2, -j^2] do
   x, y := jCoordinates(epsilon * pi) ;
   if x mod 3 eq 1 and y mod 3 eq 0 then break ; end if ;
end for ;

// Ou encore :
x, y := jCoordinates((x0 + j*y0)^-1 * pi)
       where x0 is mod3(x) where y0 is mod3(y) where x, y is jCoordinates(pi) ;

// 4p = t^2 + 3*y^2  avec t = 2x - y [= 2 modulo 3]

"Nb de points sur x^3 + y^3 + z^3 mod p via Gauss :" ; p + 1 - (2*x - y) ;
"Et via magma :" ; time Order(ChangeRing(E, GF(p))) ;
```

Résultats

```
p = 964341711958550192827299293073616528784371257294699900711922410107576544907
Time: 1.250
Nb de points sur x^3 + y^3 + z^3 mod p via Gauss :
964341711958550192827299293073616528822732351528944197362646643281876625087
Et via magma :
964341711958550192827299293073616528822732351528944197362646643281876625087
Time: 0.510
```