Université Bordeaux 1

Master Mathématiques Pures (ALGANT)

Mémoire de master 2

Théorie et pratique de la méthode des points d'Heegner

Anna Morra

Directeur du mémoire : Professeur Henri Cohen

19 juin 2006

Remerciements

Je voudrais remercier :
- Monsieur Cohen pour tout le temps qu'il m'a dedié pour l'étude du sujet, la programmation de l'algorithme et l'écriture du mémoire.
- Monsieur Belabas pour m'avoir aidée dans la programmation et m'avoir appris des astuces utiles pour mon travail.
- Monsieur Delaunay pour avoir accepté de me rencontrer pendant son sejour à Bordeaux.
- Monsieur Cremona, qui m'a éclairci les idées par e-mail.
- les professeurs Languasco et Garuti, pour m'avoir soutenue et incouragée et aussi pour s'être occupés des affaires burocratiques italiennes à ma place.
- Alexandre pour m'avoir aidée dans la correction de l'ortographe dans le mémoire.
- Ma famille et mes amis, qui m'ont soutenue et encouragée tout au long de cette experience.

Introduction

Soit E une courbe elliptique de rang 1.
La méthode des points d'Heegner nous permet, si le conducteur N de cette courbe n'est pas trop grand (disons $N < 10^8$), de calculer un point d'ordre infini sur E.
Au premier chapitre, on décrit les bases théoriques nécessaires pour comprendre cet algorithme ; parmi celles on rappelle le théorème de Mordell-Weil sur la structure du groupe abelien des points d'une courbe elliptique sur un corps, le théorème de Weil sur la fonction L associée à une courbe elliptique, la conjecture BSD et la définition de hauteur canonique.
Dans le deuxième chapitre, on s'intéresse plus spécifiquement aux points d'Heegner, à la théorie de la multiplication complexe et au théorème de Gross-Zagier.
Le troisième chapitre est dedié à la description de l'algorithme des points d'Heegner et aux possibles améliorations, dont le plus importantes sont dûes à Cremona-Silverman et Delaunay-Watkins.
Enfin, dans le quatrième chapitre on donne des résultats numériques de récherches de points rationnels sur des courbes elliptiques de conducteur assez grand, obtenus grâce à l'implementation en PARI GP de l'algorithme.

Table des matières

Chapitre 1

Introduction aux courbes elliptiques

1.1 Courbes elliptiques et invariants fondamentaux

Une courbe elliptique sur un corps est la donnée d'une courbe algébrique lisse sur un corps K (c'est-à-dire une variété algébrique projective de dimension 1 sans points singuliers de genre 1), et d'un point défini sur K, qu'on appelle *point rationnel*.

Grâce au théorème de Riemann-Roch on peut se passer de cette définition abstraite. En effet, ce théorème implique qu'il existe un modèle plan de la courbe avec équation projective, dite *équation générale de Weierstrass*, de la forme :

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3,$$

avec les $a_i \in K$. On va se servir de cette nouvelle définition pour nos objectifs.

Dans ce cas là, on a automatiquement un point rationnel sur la courbe : $\mathcal{O} = (0, 1, 0)$, qui est le seul point à l'infini.

En général, si L est une extension du corps K, on denote par $E(L)$ l'ensemble des points sur L qui satisfont l'équation de la courbe E.

On peut définir les valeurs b_2, b_4, b_6, b_8, c_4 et c_6 associées à la courbe comme :

$$\begin{aligned} b_2 &= a_1^2 + 4a_2 \\ b_4 &= a_1a_3 + 2a_4 \\ b_6 &= a_3^2 + 4a_6 \\ b_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\ c_4 &= b_2^2 - 24b_4 \\ c_6 &= -b_2^3 + 36b_2b_4 - 216b_6, \end{aligned}$$

ce qui nous permet, si la caractéristique du corps K est différente de 2 et 3, de faire un changement de cordonnées qui nous donne une équation de la forme $ZY^2 = X^3 + aXZ^2 + bZ^3$, (où $a = -c_4/48$ et $b = -c_6/864$) qu'on appelle *équation simple de Weierstrass*. Quand la courbe est décrite par cette équation, on peut définir son *discriminant* par

$$\mathrm{disc}(E) = -16(4a^3 + 27b^2).$$

Donc quand la courbe est donnée par son équation générale de Weierstrass on définit son discriminant comme le discriminant de l'équation simple associée et on obtient la formule :

$$\mathrm{disc}(E) = \frac{c_4^3 - c_6^2}{1728},$$

qui est valable seulement si la caractéristique est différente de 2 et 3. Mais on peut aussi exprimer le discriminant comme

$$\mathrm{disc}(E) = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6,$$

et cette dernière formule est valable pour toute caractéristique.
Souvent, il sera plus pratique de travailler en cordonnées affines, avec l'équation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

en se rappelant, tout de même, du point à l'infini.

1.2 La loi de groupe

Un résultat très important, dû à Fermat, est que toute courbe elliptique a une structure naturelle de groupe Abelien.

Théorème 1.2.1 *Soit E une courbe elliptique sur un corps K, définie par une équation générale de Weierstrass*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

On peut définir une loi d'addition sur la courbe en imposant que le point à l'infini $\mathcal{O}$ soit l'identité du groupe, et que $P_1 + P_2 + P_3 = \mathcal{O}$ si et seulement si les P_i sont (projectivement) alignés. On obtient que :

1. *cette addition donne à E une structure de groupe Abelien ;*
2. *si $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ sont deux points sur E, différents de $\mathcal{O}$, alors leur somme est égale à $\mathcal{O}$ si et seulement si $x_1 = x_2$ et $y_2 = -y_1 - a_1x - a_3$, et en tous les autres cas, elle est donnée par le point $P_3 = (x_3, y_3)$, avec*

$$\begin{aligned} x_3 &= m(m + a_1) - x_1 - x_2 - a_2 \\ y_3 &= m(x_1 - x_3) - y_1 - a_1x_3 - a_3, \end{aligned}$$

où

$$m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{si } P_1 \neq P_2 \\ \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x + a3} & \text{si } P_1 = P_2. \end{cases}$$

1.3 Courbes elliptiques dégénérées

Proposition 1.3.1 *Si une équation générale de Weierstrass a des singularités, alors elle en a exactement une, qui se trouve sur la droite $2y + a_1x + a_3z = 0$. En particulier, si la caractéristique de K est différente de 2 et 3, alors la singularité se trouve sur l'axe x.*

On va maintenant considérer les types possibles de singularités.
Supposons, par simplicité, que la caractéristique soit différente de 2 (mais en caractéristique 2 on a une situation semblable), pour pouvoir se réduire à $a_1 = a_3 = 0$. On a donc une équation affine du type $y^2 = P(x)$, où $P(x) = x^3 + a_2x^2 + a_4x + a_6$, et on sait que la singularité est sur l'axe x, donc le point singulier est du type $P_0 = (\alpha, 0)$, où α est une racine du polynôme $P(x)$. Donc, pour que cela donne une singularité il faut que α soit une racine multiple.

Si l'on considère maintenant l'équation d'une tangente à la courbe en P_0, elle est de la forme $ny + m(x - \alpha) = 0$. Comme P_0 est un point singulier, il faut que la tangente ait pour multiplicité au moins 3 en P_0. Cela implique que n doit être différent de 0, et donc on peut supposer $n = -1$. En remplaçant

cela dans l'équation de la courbe, on obtient $m^2(x-\alpha^2) = (x-\alpha^2)(x-\beta)$, où β est la troisième racine de $P(x)$, c'est-à-dire $(x-\alpha)^2(x-\beta-m^2)$. Donc la multiplicité en P est plus grande ou égale à 3 si et seulement si $m^2 = \alpha - \beta$. On a donc trois possibilités :

1. *"cusp"* (pointe) : quand α est racine triple de $P(x)$ on a une seule tangente de pente $m = 0$;
2. *point double avec tangentes définies sur* K : si $\alpha - \beta$ est un carré sur K ;
3. *point double avec tangentes non définies sur* K : quand $\alpha - \beta \neq 0$ n'est pas un carré sur K.

En tous les trois cas, on peut se réduire à une équation de la forme $y^2 = x^2(x-\beta)$, où la singularité se trouve maintenant dans l'origine des axes, et on peut paramétrer les points non singuliers par la droite projective $\mathbb{P}'(K)$:

$$\left(\beta + \frac{1}{t^2}, \frac{\beta}{t} + \frac{1}{t^3}\right),$$

où il faut exclure eventuellement les t tels que $\frac{1}{t^2} = -\beta$. Le point à l'infini est donné par $t = 0$. Dans cette situation on a :

Proposition 1.3.2 *L'ensemble* $G = E(K) - \{P_0\}$ *a une structure naturelle de groupe :*

1. $G \simeq (K, +)$ *quand* P_0 *est une pointe* **(réduction additive)** *;*
2. $G \simeq (K^*, \cdot)$ *quand* P_0 *est un point double avec tangentes définies sur* K **(réduction multiplicative déployée)** *;*
3. G *est isomorphe au groupe multiplicatif des éléments de norme* 1 *dans l'extension quadratique de* K *engendrée par les pentes des tangentes en* P_0*, quand* P_0 *est un point double avec tangentes non définies sur* K **(réduction multiplicative non déployée)**.

1.4 Isogénies

On veut définir une application naturelle entre courbes elliptiques, c'est-à-dire une application qui respecte les trois structures dont chaque courbe elliptique est équipée :

1. E est une courbe algébrique ;
2. E est une variété analytique de dimension 1 sur $\mathbb{C}$, c'est-à-dire une surface de Riemann compacte ;

3. E a une structure de groupe Abelien.

Grâce à un théorème de Riemann qui dit que chaque surface de Riemann est algébrique, on peut se réduire aux points (1) et (3). En outre, on a le théorème suivant :

Théorème 1.4.1 *Soient E et E' deux courbes elliptiques, avec éléments identités $\mathcal{O}$ et $\mathcal{O}'$ réspectivement, et soit ϕ un morphisme de courbes algébriques de E à E' (ϕ est définie par des fonctions rationnelles). Alors ϕ est un homomorphisme de groupes de E à E' si et seulement si $\phi(\mathcal{O}) = \mathcal{O}'$. (Donc ϕ préserve la loi de groupe !!!).*

On est donc portés à la suivante :

Définition 1.4.2 *Soient E et E' deux courbes elliptiques avec éléments identités $\mathcal{O}$ et $\mathcal{O}'$ réspectivement.*
Une isogénie ϕ de E dans E' est un morphisme de courbes algébriques de E en E' tel que $\phi(\mathcal{O}) = \mathcal{O}'$.
Une isogénie non constante est une isogénie telle que il existe $P \in E$ avec $\phi(P) \neq \mathcal{O}'$. On dit que E et E' sont isogènes si il existe une isogénie non constante entre E et E'.

Théorème 1.4.3 *Soit ϕ une isogénie non constante de E dans E' :*

1. *si K est algébriquement clos, alors ϕ est surjective ;*
2. *ϕ est une application finie, c'est-à-dire la fibre de chaque point de E' est constante et finie.*

On en déduit que ϕ induit une application injective du corps des fonctions de E' à celui de E sur quelque clôture algébrique du corps base. Le degré de la correspondante extension de corps est fini et est appelé le *degré* de ϕ.
Si l'extension de corps est séparée, alors le degré est égal aussi à la cardinalité d'une fibre, c'est-à-dire à $|\ker(\phi)|$.
S'il existe une isogénie non constante ϕ de E dans E' de degré m (une *m-isogénie*), alors on dit que E et E' sont m-isogènes.

Il faut remarquer que deux courbes elliptiques isogènes sont très semblables, mais pas isomorphes. Par exemple, deux courbes elliptiques isogènes sur $\mathbb{Q}$ ont le même rang et la même fonction L, mais pas forcement le même sous-groupe de torsion.

Proposition 1.4.4 *Soit ϕ une isogénie non constante de E dans E' de degré m. Il existe une isogénie ψ de E' dans E, appelée l'isogénie duale de ϕ, telle que*

$$\psi \circ \phi = [m]_E \text{ et } \phi \circ \psi = [m]_{E'},$$

où $[m]$ est l'application "multiplication par m" sur la courbe correspondante.

1.5 Courbes elliptiques sur $\mathbb{C}$

Une courbe elliptique sur $\mathbb{C}$ est isomorphe (autant analytiquement qu'algébriquement) au quotient de $\mathbb{C}$ par un réseau de dimension 2, $\Lambda = \omega_1\mathbb{Z} + \omega_2\mathbb{Z}$. On définit la fonction $\wp$ de Weierstrass $\wp_\Lambda(z)$ associée à Λ par la formule :

$$\wp_\Lambda(z) = \frac{1}{z^2} + \sum_{\omega \in \Lambda - \{0\}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right),$$

qui définit une fonction double-périodique (Λ-périodique) méromorphe sur $\mathbb{C}$ avec poles seulement pour $z \in \Lambda$.
$\wp_\Lambda(z)$ satisfait l'équation algébrique différentielle :

$$\wp'_\Lambda(z)^2 = 4\wp_\Lambda(z)^3 - g_2(\Lambda)\wp_\Lambda(z) - g_3(\Lambda),$$

avec $g_2(\Lambda) = 60 \sum_{\omega \in \Lambda - \{0\}} \frac{1}{\omega^4}$ et $g_3(\Lambda) = 140 \sum_{\omega \in \Lambda - \{0\}} \frac{1}{\omega^6}$.
On a donc un isomorphisme explicite entre $\mathbb{C}/\Lambda$ et une courbe elliptique sur $\mathbb{C}$, qui envoye un représentant $z \in \mathbb{C}$ d'un élément de $\mathbb{C}/\Lambda$ dans le couple $(\wp_\Lambda(z), \wp'_\Lambda(z))$, et toutes les courbes elliptiques sur $\mathbb{C}$ peuvent être obtenues de cette manière. La loi du groupe de la courbe elliptique est simplement induite par l'addition sur $\mathbb{C}$. Le point à l'infini c'est le seul pôle de $\wp$. L'isomorphisme inverse est appelé logarithme elliptique.

1.6 Courbes elliptiques sur $\mathbb{F}_q$

1.6.1 Comptage des points

Lemme 1.6.1 *Soit $q = p^k$, p premier, k impair. Soit ρ le seul caractère multiplicatif d'ordre 2 sur $\mathbb{F}_q$, soit $y^2 = f(x)$ l'équation d'une courbe elliptique sur $\mathbb{F}_q$, où $f(x)$ est un polynôme de degré 3. Alors $|E(\mathbb{F}_q)| = q + 1 - a_q$, $a_q = -\sum_{x \in \mathbb{F}_q} \rho(f(x))$.*
En particulier, si $q = p$ premier, alors $|E(\mathbb{F}_p)| = p + 1 - a_p$, avec

$$a_p = -\sum_{x \in \mathbb{F}_p} \left(\frac{f(x)}{p} \right).$$

Théorème 1.6.2 (Hasse) *Soit E une courbe elliptique sur un corps fini $\mathbb{F}_q$. Alors on a*

$$q + 1 - 2\sqrt{q} \leq |E(\mathbb{F}_q)| \leq q + 1 + 2\sqrt{q}.$$

En outre, il est facile de montrer que le groupe $E(\mathbb{F}_q)$ est le produit de au plus deux groupes cycliques, et si l'on écrit $E(\mathbb{F}_q) \simeq \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z}$, avec $d_2|d_1$, alors $d_2|q-1$.

On veut maintenant compter les points d'une courbe dégénérée sur $\mathbb{F}_p$:

Exemple

On considère une courbe elliptique E définie sur $\mathbb{Q}$ par une équation de Weierstrass à coefficients entiers, et l'on considère sa réduction modulo un premier p, c'est-à-dire la courbe définie sur $\mathbb{F}_p$ par la même équation mais avec les coefficients réduits modulo p. On a que la courbe est dégénérée si et seulement si p en divise le discriminant. En ce cas là, on peut facilement calculer les $a_p = p + 1 - |E(\mathbb{F}_p)|$ par la Proposition 1.3.2, et l'on obtient :

1. $a_p = 0$ si la réduction est additive ;
2. $a_p = 1$ si la réduction est multiplicative déployée ;
3. $a_p = -1$ si la réduction est multiplicative non déployée.

□

1.6.2 Fonctions L locales

On peut définir la fonction zéta de Hasse-Weil attachée à E comme

$$\zeta_q(E,T) = \exp\left(\sum_{n\geq 1} \frac{|E(\mathbb{F}_{q^n})|}{n} T^n\right).$$

On peut montrer que $\zeta_q(E,T)$ est une fonction rationelle, et plus précisement :

$$\zeta_q(E,T) = \frac{1 - a_q T + qT^2}{(1-T)(1-qT)}.$$

Quand $q = p$ premier on définit

$$L_p(E,T) = \frac{1}{1 - a_p T + pT^2}.$$

Les L_p seront utilisés comme facteurs d'Euler locaux dans la définition de la fonction L globale d'une courbe elliptique définie sur $\mathbb{Q}$.

1.6.3 Structure de $\mathrm{End}(E)$

L'anneau $\mathrm{End}(E)$ des $\overline{\mathbb{F}_q}$-endomorphismes d'une courbe elliptique définie sur $\mathbb{F}_q$ est un peu plus compliqué que celui d'une courbe définie sur $\mathbb{C}$ (voir le

Paragraphe 1.10.2). En effet, End(E) contient un endomorphisme additionel, l'automorphisme de Frobenius :

$$\phi : (x, y) \longrightarrow (x^q, y^q).$$

On peut montrer que ϕ (comme isogénie de E dans E) satisfait une équation quadratique avec discriminant plus petit ou égal à zéro, et il est facile de montrer que $\phi \neq [m]$, pour tout $m \in \mathbb{Z}$, donc le discriminant est strictement négatif. Par conséquent, sur $\mathbb{F}_q$, l'anneau End(E) contient toujours un ordre dans un corps quadratique imaginaire, donc E a toujours multiplication complexe.

1.7 Tordues quadratiques

Définition 1.7.1 *Soit E une courbe elliptique sur un corps K de caractéristique différente de 2 définie par une équation de la forme $y^2 = f(x)$ ($f(x)$ polynôme cubique). Soit $D \in K^*$. La* **tordue quadratique** *de E par D est la courbe elliptique E_D avec équation :*

$$Dy^2 = f(x).$$

Soit $f(x) = x^3 + ax^2 + bx + c$. On a deux cas :

1. *soit D est un carré dans K^* alors on pose $Y = y\sqrt{D}$ en obtenant la même équation que celle initiale ;*
2. *sinon on multiplie l'équation par D^3 et l'on pose $Y = D^2y$, $X = Dx$, ce qui donne la nouvelle équation :*

$$Y^2 = X^3 + aDX^2 + bD^2X + cD^3.$$

Proposition 1.7.2 *Soit E une courbe elliptique définie sur $\mathbb{F}_p$ (p premier impair) par une équation de Weierstrass de la forme $y^2 = f(x)$ ($f(x)$ polynôme cubique). Soit $D \in \mathbb{F}_p^*$, soit E_D la tordue quadratique de E par D et soient $a_p(E) = p + 1 - |E(\mathbb{F}_p)|$ et $a_p(E_D)$ défini de la même manière. Alors*

$$a_p(E_D) = \left(\frac{D}{p}\right) a_p(E).$$

En général ce résultat est vrai pour tout corps fini $\mathbb{F}_q$ de caractéristique impaire, en remplaçant $\left(\frac{D}{p}\right)$ par $\rho(D)$, où ρ est l'unique caractère multiplicatif d'ordre 2 de $\mathbb{F}_q$.

1.8 Courbes elliptiques sur $\mathbb{Q}$

On va maintenant s'occuper des aspects diophantiens des courbes elliptiques. Une des questions qu'on peut se poser à ce sujet c'est l'existence et la structure des points rationnels sur la courbe. A ce propos on sait, grâce au théorème de Fermat, que $E(\mathbb{Q})$ a une structure de groupe Abelien, et on a un autre résultat très important, dû à Mordell :

Théorème 1.8.1 (Mordell) *Soit E une courbe elliptique sur $\mathbb{Q}$. Alors $E(\mathbb{Q})$ est un groupe Abelien de type fini. Plus précisement, on a :*

$$E(\mathbb{Q}) = E_t(\mathbb{Q}) \oplus G,$$

où $E_t(\mathbb{Q})$ est le sous-groupe de torsion de $E(\mathbb{Q})$ (c'est-à-dire l'ensemble des points $T \in E(\mathbb{Q})$ tels que il existe $k \in \mathbb{Z} - \{0\}$, avec $kT = \mathcal{O}$) et G est engendré par un nombre fini r de points P_i d'ordre infini. Donc $G \simeq \mathbb{Z}^r$ et tout point $P \in E(\mathbb{Q})$ peut s'écrire comme

$$P = T + \sum_{i=1}^{r} a_i P_i,$$

avec $T \in E_t(\mathbb{Q})$, $a_i \in \mathbb{Z}$. L'ordre r est appelé **rang algébrique** *de la courbe E.*

Remarques

- Ce théorème a été étendu par Weil au cas des corps de nombres.
- Il existe un algorithme pour calculer la partie de torsion du groupe de Mordell-Weil ($E(\mathbb{Q})$).
- Par contre, on n'est pas capable, en général, de calculer les générateurs P_i de la partie infinie, et même pas de calculer le rang r ! (on conjecture que r n'est pas borné, et le plus grand r connu vaut 24)

Un autre problème qu'on peut se poser c'est d'étudier les *points entiers* sur une courbe. En ce cas là, il faut faire attention, car deux équations équivalentes pour la courbe peuvent donner deux ensembles différents de points entiers, donc ce concept est strictement lié à l'équation donnée.
A cet égard, on a un théorème de Siegel :

Théorème 1.8.2 *Soit $f(x, y) = 0$ l'équation affine d'une cubique non singulière plane à coefficients entiers. Il existe seulement un nombre fini de couples $(a, b) \in \mathbb{Z}^2$ tels que $f(a, b) = 0$.*

Remarques

- Ce théorème n'est pas du tout effectif : il ne nous permet pas de trouver ces points, ni de connaître combien il y en a.
- En pratique, on est capable de calculer les points entiers, en utilisant le logarithme elliptique et LLL, mais à condition de connaître les générateurs du groupe de Mordell-Weil.

1.8.1 Le modèle minimal d'une courbe elliptique

Soit E une courbe elliptique donnée par une équation générale de Weierstrass :

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

Soit $p \in \mathbb{Z}$ un nombre premier. Si p ne divise pas le discriminant de la courbe, on sait que la courbe a bonne réduction en p. Par contre, si p divise $\mathrm{disc}(E)$, on sait que l'équation réduite modulo p donne une courbe dégénérée, mais il se peut bien que le premier p devienne acceptable si on utilise un autre modèle de la même courbe, c'est-à-dire en appliquant une transformation birationnelle à l'équation.

On a un résultat fondamental, qui nous dit qu'il existe un modèle minimal de la courbe, en forme générale de Weierstrass, avec les $a_i \in \mathbb{Z}$, qui, parmi ses propriétés, a le discriminant le plus petit possible : si p divise ce discriminant, alors p divise le discriminant de toute équation de la courbe.

Cette propriété est dûe en large partie au fait que $\mathbb{Z}$ est un domaine principal. Si la courbe était définie sur un corps de nombres avec nombre de classes strictement plus grand que 1, elle n'aurait pas forcement eu de modèle minimal.

1.8.2 Fonction L associée a une courbe elliptique

Soit E une courbe elliptique donnée par une équation générale de Weierstrass.

Quand un premier p ne divise pas le discriminant (minimal) de la courbe, alors cette équation définit une courbe elliptique sur $\mathbb{F}_p$ à laquelle est associé le facteur d'Euler local $L_p(E,T)$. Si p divise le discriminant on a une unique singularité qui peut être de trois types, à chacun desquels on peut associer aussi un facteur local :

1. réduction additive : $L_p(E,T) = 1$,
2. réduction multiplicative déployée : $L_p(E,T) = \frac{1}{1-T}$,
3. réduction multiplicative non déployée : $L_p(E,T) = \frac{1}{1+T}$.

On peut donc réunir le trois formules dans la suivante : $L_p(E,T) = \frac{1}{1-a_pT}$, alors pour tout premier p on peut écrire :

$$L_p(E,T) = \frac{1}{1 - a_pT + \chi(p)T^2},$$

où $\chi(p)$ est égal à 1 si on a bonne réduction en p, à 0 sinon.
On peut enfin définir la fonction L globale de la courbe E comme :

$$L(E,s) = \prod_p L_p(E,p^{-s}) = \prod_p \frac{1}{1 - a_p p^{-s} + \chi(p)p^{1-2s}}.$$

Il est facile, grâce à l'inégalité de Hasse, de montrer que ce produit d'Euler (qui est évidemment une série de Dirichlet) est absolument convergent si $\mathrm{Re}(s) > 3/2$.
On a le théorème suivant :

Théorème 1.8.3 *Soient E et E' deux courbes elliptiques définies sur $\mathbb{Q}$. Si E et E' sont isogènes sur $\mathbb{Q}$, alors $L(E,s) = L(E',s)$.*
Inversement, si $L(E,s) = L(E',s)$, alors E et E' sont isogènes.

La première partie du théorème n'est pas trop difficile à prouver, par contre l'autre a été une conjecture (la conjecture d'isogénie) pendant longtemps, et est dévenue un théorème grâce à la preuve de Faltings. C'est un théorème très important du point de vue théorique.

Un autre théorème très important et extrêmement difficile à démontrer c'est le théorème de Wiles et autres, qui prouve la conjecture de Taniyama-Shimura-Weil :

Théorème 1.8.4 (Wiles et autres) *La fonction $L(E,s)$ possède une continuation analytique à tout le plan complexe à une fonction holomorphe. En outre, il existe un entier N (qui a les mêmes diviseurs premiers que le discriminant minimal de la courbe et qui divise ce dernier), appelé le* **conducteur** *de la courbe, tel que si l'on pose :*

$$\Lambda(E,s) = N^{s/2}(2\pi)^{-s}\Gamma(s)L(E,s),$$

on a l'équation fonctionnelle :

$$\Lambda(E,2-s) = \varepsilon(E)\Lambda(E,s),$$

où $\varepsilon(E) = \pm 1$ est appelé le signe de l'équation fonctionnelle.

Remarques

- Il existe un algorithme de Tate, qui permet de trouver le modèle minimal de la courbe et son conducteur.
- Il y a aussi un algorithme plus récent pour calculer $\varepsilon(E)$, dû à Mestre-Henniart et Halberstadt.

Enfin, on a une conjecture très importante liée aux courbes elliptiques sur $\mathbb{Q}$:

Conjecture 1.8.5 (Birch et Swinnerton-Dyer (BSD)) *Soit E une courbe elliptique sur $\mathbb{Q}$. Le rang algébrique de E est égal à l'ordre d'annulation de $L(E,s)$ en $s=1$. En outre :*

$$\lim_{s\to 1}\frac{L(E,s)}{(s-1)^r}=\frac{L^{(r)}(E,1)}{r!}=\omega_1(E)\frac{|Ш(E)|R(E)c_\infty(E)\prod_{p|N}c_p(E)}{|E_t(\mathbb{Q})|^2},$$

où $\omega_1(E)$ est la période réelle de E, $Ш(E)$ est le groupe de Tate-Shafarevich, $R(E)$ le régulateur, les $c_p(E)$ sont des petits entiers appelés nombres de Tamagawa et $c_\infty(E)$ est le nombre de composantes connexes de $E(\mathbb{R})$ (toutes ces valeurs sont calculées à partir du modèle minimal de la courbe E).

Remarques

- Toutes les quantités du membre de droite sont computables, sauf qu'il n'y a pas un algorithme connu pour calculer $R(E)$ et on ne connaît pas $|Ш(E)|$: en général on ne sait même pas s'il est fini, sauf pour $r\leq 1$.
- Les seuls cas prouvés de la conjecture sont pour rang analytique 0 et 1. Si le rang analytique est plus grand ou égal à 2 on ne sait absolument rien, même pas pour des cas spéciaux.

1.8.3 Une propriété des points rationnels

Proposition 1.8.6 *Soit R un anneau principal, avec corps de fractions K, et soit E une courbe elliptique donnée par une équation générale de Weierstrass :*

$$Y^2+a_1XY+a_3Y=X^3+a_2X^2+a_4X+a_6,$$

avec les $a_i\in R$, et soit $P=(X,Y)\in E(K)$ un point affine sur K.
Alors, il existe $M,N,D\in R$ tels que

$$X=\frac{M}{D^2}\ \text{et}\ Y=\frac{N}{D^3},\ \text{avec}\ \gcd(M,D)=\gcd(N,D)=1.$$

Evidemment cette propriété est valable pour le cas $R=\mathbb{Z}$ et $K=\mathbb{Q}$ et elle est très utile en pratique (comme on verra plus tard, dans la méthode des points de Heegner).

1.8.4 Hauteur naïve et canonique

Soit E une courbe elliptique sur $\mathbb{Q}$, et soit $P \in E(\mathbb{Q})$ un point donné. On peut être interessé à savoir si ce point est d'ordre fini ou pas. Cela est possible grâce à la caractérisation des points de torsion, mais si, par exemple, on a deux points P et Q et l'on sait qu'ils ne sont pas de torsion, alors on voudrait aussi savoir s'ils sont indépendants ou pas. Pour cela on a besoin de la définition de *hauteur canonique*.

On va commencer par la définition de *hauteur naïve* d'un nombre rationnel $x \neq 0$. Si l'on écrit $x = \frac{n}{d}$ avec $\gcd(n, d) = 1$, alors on définit

$$h(x) = \max(\log(|n|)), \log(|d|)),$$

où l'on pose $\log(0) = -\infty$.
En général, si $P \in E(\mathbb{Q})$ ayant pour cordonnées affines $\left(\frac{m}{d^2}, \frac{n}{d^3}\right)$, avec $\gcd(m, d) = \gcd(n, d) = 1$, on définit la hauteur naïve du point P par :

$$h(P) = \max(\log(|m|), \log(|d^2|)).$$

A partir de cette définition, on peut définir la hauteur canonique du point P comme :

$$\widehat{h}(P) = \lim_{k \to \infty} \frac{h(kP)}{k^2},$$

où l'on a $\widehat{h}(P) \geq 0$ pour tout $P \in E(\mathbb{Q})$, car $h(P) \geq 0$ pour tout P.

Théorème 1.8.7 *La limite ci-dessus existe, et définit une fonction non négative $\widehat{h}(P)$ sur $E(\mathbb{Q})$ avec les propriétés suivantes :*

1. **(Forme quadratique :)** *la fonction $\widehat{h}(P)$ est une forme quadratique sur $E(\mathbb{Q})$.*
   Si on définit
   $$\langle P, Q \rangle = \frac{\widehat{h}(P+Q) - \widehat{h}(P) - \widehat{h}(Q)}{2},$$
   alors $\langle P, Q \rangle$ est une forme bilinéaire symétrique sur $E(\mathbb{Q})$ telle que $\langle P, P \rangle = \widehat{h}(P)$, donc $\widehat{h}(kP) = k^2 \widehat{h}(P)$.
2. **(Non dégénérée :)** *On a $\widehat{h}(P) = 0$ si et seulement si $P \in E_t(\mathbb{Q})$, donc $\widehat{h}$ induit une forme quadratique définie positive sur le groupe Abelien libre de type fini $E(\mathbb{Q})/E_t(\mathbb{Q})$.*
3. **(Indépendance :)** *Les points $(P_i)_{1 \leq i \leq n}$ dans $E(\mathbb{Q})$ sont linéairement indépendants dans $E(\mathbb{Q})/E_t(\mathbb{Q})$ si et seulement si le déterminant de la matrice ("height pairing") $M = ((\langle P_i, P_j \rangle)_{1 \leq i,j \leq n})$ est différent de zéro.*

4. **(Borne :)** *Il existe une constante calculable explicitement $C(E)$, qui ne dépend que de E, telle que pour tout $P \in E(\mathbb{Q})$ on a $|\widehat{h}(P) - h(P)| \leq C(E)$ (voir le théorème suivant pour les détails).*
5. **(Finitude :)** *Pour tout $B > 0$ il existe seulement un nombre fini de points $P \in E(\mathbb{Q})$ tels que $\widehat{h}(P) \leq B$.*

On va maintenant s'occuper de la constante $C(E)$.

Théorème 1.8.8 *Soit E une courbe elliptique sur $\mathbb{Q}$ définie par une équation générale de Weierstrass. Posons :*

$$\mu(E) = \frac{\log(|\operatorname{disc}(E)|) + \log^+(j(E))}{6} + \log^+(b_2/12) + \log(2^*), \qquad (1.1)$$

où $log^+(x) = \log(\max(e, |x|))$ et $2^* = 2$ si $b_2 \neq 0$, 1 sinon, et $j(E)$ est le j-invariant de la courbe (voir le Paragraphe 1.10.1).
Alors pour $P \in E(\mathbb{Q})$ on a :

$$-\frac{h(j(E))}{12} - \mu(E) - 1.946 \leq \widehat{h}(P) - h(P) \leq \mu(E) + 2.14.$$

1.9 Paramétrisation modulaire

Si $L(E, s) = \sum_{n \geq 1} a_n n^{-s}$, alors le théorème de Taniyama-Weil nous dit que $f_E(\tau) = \sum_{n \geq 1} a_n q^n$, où $q = e^{2i\pi\tau}, \tau \in \mathcal{H}$, est une forme modulaire de poids 2 sur $\Gamma_0(N)$.

Comme $f_E(\tau)d\tau$ est une differentielle holomorphe, alors l'intégral $\widetilde{\phi}(\tau) = 2i\pi \int_\infty^\tau f_E(z)dz$ (où ∞ est le point à l'infini du demi-plan supérieur) est indépendant du parcours choisi, donc il définit une application de $\mathcal{H}$ dans $\mathbb{C}$. explicitement, si $\tau \in \mathcal{H}$, alors on a

$$\widetilde{\phi}(\tau) = \sum_{n \geq 1} \frac{a_n}{n} q^n.$$

Le fait que f_E est une forme modulaire est équivalent au fait que $\widetilde{\phi}$ induit une application analytique ϕ de $X_0(N)$ dans $\mathbb{C}/\Lambda$ où $X_0(N) = (\mathcal{H} \cup \mathbb{P}_1(\mathbb{Q}))/\Gamma_0(N)$ est la courbe modulaire associée à $\Gamma_0(N)$ et Λ est le réseau formé par les périodes de f_E, qui correspondent aux valeurs de ϕ pour $\tau = \gamma(\infty)$ et $\gamma \in \Gamma_0(N)$ (qui peuvent être calculés avec la formule integrale et pas avec la série).

Le réseau Λ est très souvent un sous-réseau du réseau Λ_E associé au modèle minimal de la courbe E. En ce cas là $\widetilde{\phi}$ induit une application analytique

ϕ de $X_0(N)$ dans $\mathbb{C}/\Lambda_E$. On peut donc composer ϕ avec $(\wp, \wp')$ pour obtenir des points sur la courbe $E(\mathbb{C})$.
Au fait, il pourrait arriver aussi que Λ ne soit pas un sous-réseau de Λ_E, ce qui arrive si la *"constante de Manin"* de la courbe n'est pas 1, mais ce cas arrive rarement en pratique, et il est facilement traitable.

Donc pour tout $\tau \in \mathcal{H}$ on peut associer un $\varphi(\tau) \in E(\mathbb{C})$, où $\varphi = \wp \circ \phi$ est une application de $X_0(N)$ dans $E(\mathbb{C})$, appelée *paramétrisation modulaire*, et le théorème de Wiles nous assure que cette application existe et est unique au signe près.

1.10 j-invariant et multiplication complexe

1.10.1 Le j-invariant d'une courbe elliptique

Soit E une courbe elliptique définie sur $\mathbb{C}$. On a vu au Paragraphe 1.5 qu'on peut écrire son équation comme :

$$y^2 = 4x^3 - g_2 x - g_3.$$

On définit le j-invariant (invariant modulaire) de E :

$$j(E) = \frac{1728 g_2^3}{(g_2^3 - 27 g_3^2)} = \frac{c_4^3}{\mathrm{disc}(E)}.$$

Une des propriétés fondamentales de j est que cette fonction caractérise la classe d'isomorphisme de la courbe E sur $\mathbb{C}$:

Théorème 1.10.1 *Deux courbes elliptiques E et E' sont isomorphes sur $\mathbb{C}$ si et seulement si $j(E) = j(E')$.*

Soit $E = \mathbb{C}/\Lambda$ une courbe elliptique sur $\mathbb{C}$. Alors, Λ peut être engendré (comme $\mathbb{Z}$-module) par deux nombres complexes, ω_1 et ω_2, $\mathbb{R}$-linéairement indépendants. Soit $\tau = \omega_2/\omega_1$. Quitte à changer l'ordre des ω_i on peut supposer $\mathrm{Im}(\tau) > 0$. Or, multiplier le réseau Λ par un nombre complexe ne change pas la classe d'isomorphisme de la courbe E, donc on a $j(E) = j(E_\tau)$, où $E_\tau = \mathbb{C}/(\mathbb{Z} + \tau\mathbb{Z})$. On va noter dorénavant $j(\tau) = j(E_\tau)$. Cela définit une fonction j sur $\mathcal{H} = \{\tau \in \mathbb{C} | \, \mathrm{Im}(\tau) > 0\}$.

Soit $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$. Alors, le réseau engendré par $a\tau + b$ et $c\tau + d$ est le même que celui engendré par 1 et τ donc on obtient l'invariance modulaire de $j(\tau)$:

Théorème 1.10.2 *Pour tout élément* $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$ *on a*

$$j\left(\frac{a\tau+b}{c\tau+d}\right) = j(\tau).$$

On a un résultat très important :

Proposition 1.10.3 *La fonction j est une application bijective de la compactification de $\mathcal{H}/\mathrm{SL}_2(\mathbb{Z})$ dans le plan projectif complexe $\mathbb{P}_1(\mathbb{C})$, c'est-à-dire $j(\tau)$ assume une et une seule fois toute valeur possible (y compris ∞) sur $\overline{\mathcal{H}/\mathrm{SL}_2(\mathbb{Z})}$.*

1.10.2 Multiplication complexe

On considère l'ensemble $\mathrm{End}(E)$ des endomorphismes d'une courbe elliptique et on remarque qu'il est un anneau, avec l'opération d'addition induite par celle de E et la multiplication donnée par la composition d'endomorphismes. Cet anneau contient l'application $[m]$ pour tout $m \in \mathbb{Z}$ et on peut facilement montrer que toutes ces applications sont distinctes, donc $\mathbb{Z} \subset \mathrm{End}(E)$.

Le résultat fondamental est que soit $\mathrm{End}(E) = \mathbb{Z}$, soit $\mathrm{End}(E)$ est un ordre dans un corps quadratique imaginaire, c'est-à-dire $\mathrm{End}(E) \simeq \mathbb{Z} + \tau\mathbb{Z}$, avec $\tau = \frac{D+\sqrt{D}}{2}$ pour un certain D discriminant fondamental négatif. Dans ce dernier cas, on dit que E a multiplication complexe par l'ordre de discriminant D.

Théorème 1.10.4 (Multiplication complexe) *Soit $\tau \in \mathcal{H}$ un nombre quadratique de discriminant D (discriminant de la forme quadratique associée). Alors la courbe elliptique $E_\tau = \mathbb{C}/(\mathbb{Z} + \tau\mathbb{Z})$ a multiplication complexe par un ordre contenu dans le corps quadratique $\mathbb{Q}(\tau)$ et le j-invariant $j(\tau) = j(E_\tau)$ est un entier algébrique. En particulier, le degré de $j(\tau)$ est exactement $h(D)$ où $h(D)$ est le nombre de classes de l'ordre imaginaire quadratique de discriminant D.*

Conséquence
$j(\tau) \in \mathbb{Q}$ (et plus précisement $\in \mathbb{Z}$) si et seulement si $h(D) = 1$. Cela ne peut arriver que pour 13 discriminants possibles :

$$-3, -4, -7, -8, -11, -19, -43, -67, -163$$

(qui correspondent à des corps quadratiques), et

$$-12, -16, -27, -28$$

(qui correspondent à des ordres quadratiques).
Les valeurs de la fonction j correspondantes aux corps quadratiques ci-dessus sont :

$$\begin{aligned}
j\left(\frac{1+i\sqrt{3}}{2}\right) &= 0 \\
j(i) &= 1728 \\
j\left(\frac{1+i\sqrt{7}}{2}\right) &= -3375 \\
j(i\sqrt{2}) &= 8000 \\
j\left(\frac{1+i\sqrt{11}}{2}\right) &= -32768 \\
j\left(\frac{1+i\sqrt{19}}{2}\right) &= -884736 \\
j\left(\frac{1+i\sqrt{43}}{2}\right) &= -884736000 \\
j\left(\frac{1+i\sqrt{67}}{2}\right) &= -147197952000 \\
j\left(\frac{1+i\sqrt{163}}{2}\right) &= -262537412640768000,
\end{aligned}$$

et pour les ordres quadratiques on a :

$$\begin{aligned}
j(i\sqrt{3}) &= 54000 \\
j(2i) &= 287496 \\
j\left(\frac{1+3i\sqrt{3}}{2}\right) &= -12288000 \\
j\left(i\sqrt{7}\right) &= 16581375.
\end{aligned}$$

Quand on a une courbe elliptique à multiplication complexe, il est facile de calculer les coefficients de la fonction L associée : en effet, on a théorème suivant :

Théorème 1.10.5 *Soit E une courbe elliptique à multiplication complexe par un ordre quadratique imaginaire de discriminant D et soit p un nombre*

premier. Alors on a :

$$|E(\mathbb{F}_p)| = p + 1 - a_p,$$

où a_p est donné par :

1. *si p est inerte $\left(\left(\frac{D}{p}\right) = -1\right)$ alors $a_p = 0$;*
2. *si p se factorise en produit de premiers, disons $p = \pi\overline{\pi}$, alors $a_p = \pi + \overline{\pi}$ pour un choix adapté de π.*

Remarque
Si $D < -4$, alors il n'existe que deux choix possibles pour π, car l'ordre quadratique a deux unités. Ces choix donnent deux valeurs opposées de a_p. Une d'elles est le a_p correcte pour E, l'autre donne a_p pour la courbe E tordue par un non-résidue quadratique.
Si $D = -4$ ou $D = -3$, alors on a 4 (respectivement 6) choix pour π, correspondants aussi à des courbes tordues.

Chapitre 2

Théorie des points d'Heegner

2.1 Points d'Heegner : définition et propriétés

Définition 2.1.1
- *Soit* $\tau \in \mathcal{H} = \{z \in \mathbb{C} | \operatorname{Im}(z) > 0\}$. *On dit que* τ *est un* **point de multiplication complexe (CM)** *s'il est racine d'une équation quadratique de la forme* $A\tau^2 + B\tau + C = 0$, *avec* $A, B, C \in \mathbb{Z}$, $B^2 - 4AC < 0$.
- *Si* $\gcd(A, B, C) = 1$ *et* $A > 0$ *(ce qui implique l'unicité de cette forme quadratique) alors* $(A, B, C) = Ax^2 + Bxy + Cy^2$ *est la forme quadratique (définie positive) associée à* τ, *et on appelle discriminant de* τ *le discriminant de cette forme :* $\Delta(\tau) = B^2 - 4AC$.
- *Soit* $N \geq 1$ *un entier fixé. On dit que* τ *est un* **point d'Heegner** *de niveau* N *si* $\Delta(N\tau) = \Delta(\tau)$.

Notations
- $\Gamma = \mathrm{SL}_2(\mathbb{Z})$
- $\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma \middle| c \equiv 0 \mod N \right\}$
- $\Gamma_1(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma_0(N) \middle| a \equiv 1 \mod N \right\}$

Proposition 2.1.2 *Si* $\gamma \in \mathrm{SL}_2(\mathbb{Z})$, *alors* $\Delta(\gamma(\tau)) = \Delta(\tau)$, *pour tout* $\tau \in \mathcal{H}$ *(en particulier, cela est vrai pour* $\gamma \in \Gamma_0(N)$*).*
Si $\gamma \in \Gamma_0(N)$ *et* τ *est un point d'Heegner de niveau* N, *alors il en est de même pour* $\gamma(\tau)$.

Preuve
La première partie de la Proposition est évidente, on va prouver la deuxième : on a que $\Delta(\tau) = \Delta(N\tau)$, car τ est un point d'Heegner, et grâce à la première partie de la Proposition on sait aussi que $\Delta(\tau) = \Delta(\gamma(\tau))$ et que $\Delta(N\tau) = \Delta(\gamma(N\tau))$. Il suffit donc de prouver que $\Delta(\gamma(N\tau)) = \Delta(N\gamma(\tau))$, mais cela est facile si on se rappelle que :

$$\Gamma_0(N) = \Gamma \cap \begin{pmatrix} N & 0 \\ 0 & 1 \end{pmatrix}^{-1} \Gamma \begin{pmatrix} N & 0 \\ 0 & 1 \end{pmatrix}.$$

□

Proposition 2.1.3 *Soit $\tau \in \mathcal{H}$ une racine irrationelle de la forme quadratique (A, B, C) avec discriminant $D < 0$.*
τ est un point d'Heegner de niveau N si et seulement si $N|A$ et une des conditions suivantes (qui sont équivalentes) est satisfaite :

1. $\gcd(A/N, B, CN) = 1$,
2. $\gcd(N, B, AC/N) = 1$,
3. *il existe un $F \in \mathbb{Z}$ tel que $B^2 - 4NF = D$ et* $\gcd(N, B, F) = 1$.

Preuve
$\tau = \frac{-B+\sqrt{D}}{2A}$, alors $N\tau = \frac{-NB+N\sqrt{D}}{2A}$ et on veut qu'il soit de la forme $\frac{-B'+\sqrt{D}}{2A'}$. On obtient que $A = NA'$ et $B = B'$, donc $N|A$ et $N\tau$ est racine de l'équation $A/N(N\tau)^2 + B(N\tau) + CN + 0$, qui a pour discriminant $\Delta = B^2 - 4AC$, donc il faut que cette équation soit minimale, c'est-à-dire $\gcd(A/N, B, CN) = 1$, ce qui prouve la Proposition avec la première condition. L'équivalence avec les deux autres propriétés est facile à démontrer. □

Corollaire 2.1.4 *Soit τ un point d'Heegner de niveau N et discriminant D. On a que $W(\tau) = -\frac{1}{N\tau}$ est aussi un point d'Heegner de même niveau et de même discriminant.*

Preuve
Si (A, B, C) est la forme quadratique associée à τ, alors $(CN, -B, A/N)$ est celle associée a $-\frac{1}{N\tau}$, et on conclut grâce à la Proposition précédente. □

Dorénavant on va supposer D discriminant fondamental.
On rappelle que dans ce cas le groupe de classes $Cl(K)$ de $K = \mathbb{Q}(\sqrt{D})$ est en bijection avec les classes des formes quadratiques définies positives

(A, B, C) de discriminant D, modulo l'action de $\mathrm{SL}_2(\mathbb{Z})$. La bijection est celle qui à chaque forme (A, B, C) associe l'idéal $\mathbb{Z} + \tau\mathbb{Z}$, où $\tau = \frac{-B+\sqrt{D}}{2A}$.

Proposition 2.1.5 *Soit τ un point d'Heegner de discriminant D et niveau N.*
Si D est un discriminant fondamental, alors la condition $\gcd(N, B, F) = 1$ *de la Proposition précédente est automatiquement satisfaite et pour tout* $p|\gcd(D, N)$ *on a* $v_p(N) = 1$.

Preuve
Soit p un premier tel que $p|\gcd(N, B, F)$. Alors $p^2|B^2 - 4NF = D$, ce qui implique $p = 2$ et $D/4 \equiv 2, 3 \mod 4$, vu que D est un discriminant fondamental. Mais alors $(B/2)^2 = D/4 + NF \equiv D/4 \equiv 2, 3 \mod 4$, ce qui est impossible.
Enfin, si $p|\gcd(D, N)$ et $p^2|N$, alors, vu que $B^2 - 4NF = D$, on a que $p|B$, donc $p^2|D$, donc comme dans la prèmiere partie, on déduit que $p = 2$, mais alors $4|N$, d'où $(B/2)^2 \equiv D/4 + NF \equiv 2, 3 \mod 4$, absurde. □

Proposition 2.1.6 *Il existe une bijection entre les classes modulo $\Gamma_0(N)$ des points d'Heegner de niveau N et discriminant D et les couples $(\beta, [\mathfrak{a}])$, avec $\beta \in \mathbb{Z}/2N\mathbb{Z}$, tel que $b^2 \equiv D \mod 4N$ pour tout soulèvement de β à $b \in \mathbb{Z}$ et $[\mathfrak{a}] \in Cl(K)$, classe d'idéaux.*

Preuve
La démonstration est semblable a celle de l'équivalence entre classes d'idéaux et formes quadratiques (qu'on peut trouver dans [Coh1]), on décrit seulement la bijection :

- étant donné le couple $(\beta, [\mathfrak{a}])$, on lui associe la forme quadratique (A, B, C) qui est en bijection avec $[\mathfrak{a}]$ et telle que $N|A$ et $B \equiv \beta \mod 2N$ et on obtient l'élément $\tau = \frac{-B+\sqrt{D}}{2A}$.
- Si l'on a τ on peut lui associer la forme quadratique (A, B, C) et enfin $\beta \equiv B \mod 2N$ et $\mathfrak{a} = \mathbb{Z} + \tau\mathbb{Z}$.

□

2.1.1 Multiplication complexe

Soit K un corps de nombres. Le *corps de classes de Hilbert* est une extension de K, qui parmi ses propriétés est une extension abelienne de K, et son

groupe de Galois est canoniquement isomorphe au groupe des classes $Cl(K)$, où l'isomorphisme est donné par une application complétement explicite :

$$Art : Cl(K) \to Gal(H/K).$$

Donc tout élément de $Gal(H/K)$ peut s'écrire comme $Art([\mathfrak{a}])$ pour une unique classe d'idéaux $[\mathfrak{a}]$. On va noter par h^σ l'action d'un élément $\sigma \in Gal(H/K)$ sur un élément $h \in H$.
On va utiliser le théorème fondamental de la multiplication complexe :

Théorème 2.1.7 *Soit $\tau = (\beta, [\mathfrak{a}])$ un point d'Heegner de niveau N et discriminant D. Soient $K = \mathbb{Q}(\sqrt{D})$ et H son corps de classes de Hilbert. Alors,*

$$\varphi(\tau) \in E(H)$$

et on a les propriétés suivantes :

1. *pour tout $[\mathfrak{b}] \in Cl(K)$:*

$$\varphi((\beta, [\mathfrak{a}]))^{Art([\mathfrak{b}])} = \varphi((\beta, [\mathfrak{a}\mathfrak{b}^{-1}]))$$

   (loi de réciprocité de Shimura) *;*

2. $$\varphi(\mathrm{W}(\beta, [\mathfrak{a}])) = \varphi((-\beta, [\mathfrak{a}\mathfrak{n}^{-1}])),$$

   où $\mathfrak{n} = N\mathbb{Z} + \frac{B+\sqrt{D}}{2}\mathbb{Z}$, B est un entier congru à β modulo $2N$;

3. $$\overline{\varphi((\beta, [\mathfrak{a}]))} = \varphi((-\beta, [\mathfrak{a}^{-1}])).$$

Donc on sait qu'en utilisant τ on peut obtenir un point de la courbe à cordonnées algébriques, dans H. Cela est dû à la multiplication complexe et généralise le fait que $e^{2i\pi q}$, avec $q \in \mathbb{Q}$ quelconque, donne des nombres algébriques.

On va calculer la trace P de $\varphi(\tau)$:

$$P = \sum_{\sigma \in Gal(H/K)} \varphi((\beta, [\mathfrak{a}]))^\sigma = \sum_{[\mathfrak{b}] \in Cl(k)} \varphi((\beta, [\mathfrak{a}\mathfrak{b}^{-1}])) = \sum_{[\mathfrak{b}] \in Cl(K)} \varphi((\beta, [\mathfrak{b}])).$$

On sait par la théorie de Galois que $P \in E(K)$, mais si le signe de l'équation fonctionelle ε est -1 on peut dire plus :

Lemme 2.1.8 *Si $\varepsilon = -1$ alors $P \in E(\mathbb{Q})$.*

Preuve
$\varepsilon = -1$ est équivalent à $\varphi \circ \mathrm{W} = \varphi$, donc on a :

$$\overline{\varphi((\beta, [\mathfrak{b}]))} = \overline{\varphi(\mathrm{W}(\beta, [\mathfrak{b}]))} = \overline{\varphi((-\beta, [\mathfrak{b}\mathfrak{n}^{-1}]))} = \varphi((\beta, [\mathfrak{b}^{-1}\mathfrak{n}])),$$

d'où :

$$\overline{P} = \sum_{[\mathfrak{b}] \in Cl(K)} \varphi((\beta, [\mathfrak{b}^{-1}\mathfrak{n}])) = \sum_{[\mathfrak{b}] \in Cl(K)} \varphi((\beta, [\mathfrak{b}])) = P,$$

et on conclut que $P \in E(\mathbb{Q})$. □
On sait ainsi comment obtenir un point dans $E(\mathbb{Q})$, mais il est possible que le point trouvé soit de torsion. Au fait, si le rang de la courbe est strictement plus grand que 1, le point est toujours de torsion (cela vient du théorème de Gross-Zagier et du travail de Kolyvagin). En outre, on peut prouver de façon similaire au Lemme précédent, que si $\varepsilon = 1$ alors $P + \overline{P}$ est de torsion. On en conclut qu'on ne peut appliquer la méthode des points d'Heegner que si le rang est égal à 1.

2.2 Le théorème de Gross-Zagier

Soit E une courbe elliptique définie sur $\mathbb{Q}$ par une équation simple de Weierstrass :

$$y^2 = x^3 + ax^2 + bx + c,$$

et soit E_D sa tordue quadratique par D :

$$Dy^2 = x^3 + ax^2 + bx + c,$$

où D est un discriminant fondamental, et il est un carré modulo $4N$. Alors, on sait, grâce à la Proposition 2.1.5 que si $p | \gcd(D, N)$, $p^2 \nmid N$, et donc on en déduit que le conducteur de E_D est

$$N_D = \frac{ND^2}{\gcd(N, D)}$$

et

$$L(E_D, s) = \sum_{n \geq 1} \left(\frac{D}{n}\right) \frac{a_n}{n^s}.$$

On rappelle aussi qu'on a définit une hauteur canonique $\widehat{h}$ sur $E(\mathbb{Q})$ (voir le Paragraphe 1.8.8).

Théorème 2.2.1 (Gross-Zagier) *Si* $\gcd(D, 2N) = 1$ *et* $D \neq 3$*, alors la hauteur canonique du point* P *obtenu par la méthode d'Heegner est :*

$$\widehat{h}(P) = \frac{\sqrt{|D|}}{4\,\mathrm{Vol}(E)} L'(E, 1) L(E_D, 1).$$

Donc on peut facilement vérifier si le point P a hauteur 0, c'est-à-dire s'il est de torsion.

En outre, si le rang de la courbe est 1, on sait que $R(E) = \widehat{h}(G)$, où G est un générateur de la partie infinie du groupe de Mordell-Weil, donc on peut écrire $P = lG+T$, avec $l \in \mathbb{Z}, T \in E_t(\mathbb{Q})$, et on en déduit que $\widehat{h}(P) = l^2\widehat{h}(G)$. En utilisant la formule donnée par la Conjecture 1.8.5 (BSD), on obtient la formule :

$$\frac{l^2}{|Ш(E)|} = \omega_1(E) \frac{c(E)\sqrt{|D|}}{4\,\mathrm{Vol}(E)|E_t(\mathbb{Q})|^2} L(E_D, 1),$$

où $c(E)$ est le produit des nombres de Tamagawa (y compris c_∞). En général on ne connait pas la valeur de $|Ш(E)|$, mais on sait qu'elle est très souvent égale à 1, et que, si le rang est 1, elle est toujours le carré d'un petit entier.

Maintenant on va donner une conjecture de Gross-Zagier qui nous permet de considerer le cas général, avec $\gcd(D, 2N)$ quelconque et D discriminant fondamental quelconque.

Conjecture 2.2.2 (Gross-Zagier) *Soit* E *une courbe elliptique de rang analytique* 1*, et soit* D *un discriminant fondamental négatif, qui est un carré modulo* $4N$*. Supposons que* $L(E_D, 1) \neq 0$ *et que pour tout* $p|\gcd(D, N)$ *on ait* $a_p = -1$*. Alors*

$$\frac{l^2}{|Ш(E)|} = \omega_1(E) \frac{c(E)\sqrt{|D|}(w(D)/2)^2}{4\,\mathrm{Vol}(E)|E_t(\mathbb{Q})|^2} 2^{\omega(\gcd(D,N))} L(E_D, 1), \tag{2.1}$$

où $w(D)$ *est le nombre de racines de l'unité dans* $\mathbb{Q}(\sqrt{D})$ *(c'est-à-dire,* $w(-3) = 6$*,* $w(-4) = 4$ *et* $w(D) = 2$ *pour* $D < 4$*) et* $\omega(\gcd(D, N))$ *est le nombre de facteurs premiers distincts de* $\gcd(D, N)$*.*

La condition $a_p = -1$ pour $p|\gcd(D, N)$ est nécessaire pour obtenir un point d'Heegner et pour la validité de la formule. En outre, elle implique que $\varepsilon(E_D) = 1$, ce qui nous permet de calculer $L(E_D, 1)$ par la formule :

$$L(E_D, 1) = 2 \sum_{n \geq 1} \frac{a_n}{n} e^{-2\pi n/\sqrt{N_D}}$$

Chapitre 3

Aspects algorithmiques de la méthode des points d'Heegner

3.1 L'algorithme de base

On va maintenant décrire une simple version de l'algorithme des points d'Heegner, qui permet, étant donné le modèle minimal d'une courbe elliptique de rang 1, de calculer un point de non torsion sur la courbe.

On a programmé cet algorithme en `PARI GP`, donc on va supposer, dorénavant, d'avoir à disposition certains outils de calcul, qui sont déjà programmés dans le logiciel.

Algorithme 3.1.1

1. Tout d'abord, il faut calculer un certain nombre de valeurs associées à la courbe E, en particulier, son conducteur N, les générateurs du réseau Λ_E, notés par $\omega_1(E)$ et $\omega_2(E)$, le volume $\mathrm{Vol}(E)$ du parallélogramme fondamental et $L'(E,1)$, donné par la formule :
$$L'(E,1) = 2\sum_{n\geq 1} \frac{a_n(E)}{n} E_1\left(\frac{2\pi n}{\sqrt{N}}\right),$$
où $E_1(x) = \int_x^\infty \frac{e^{-t}}{t} dt$.
2. On fait une boucle sur les discriminants fondamentaux D tels que D est un carré modulo $4N$ et $a_p = -1$ pour tout $p | \gcd(D,N)$.
Pour chaque D on calcule
$$L(E_D,1) = 2\sum_{n\geq 1} \frac{a_n}{n}\left(\frac{D}{n}\right) \exp\left(\frac{-2\pi n}{\sqrt{ND^2/\gcd(D,N)}}\right),$$

et l'on vérifie que $L(E_D, 1)$ soit différent de 0 (dans le cas contraire, on passe au D suivant).

3. On choisit $\beta \in \mathbb{Z}/(2N)\mathbb{Z}$ tel que $D \equiv \beta^2 \mod 4N$ et on calcule
$$m^2 = \omega_1(E)\frac{c(E)\sqrt{|D|}(w(D)/2)^2 2^{\omega(\gcd(D,N))}}{4\,\mathrm{Vol}(E)|E_t(\mathbb{Q})|^2}L(E_D, 1)$$
(m^2 correspond à $\frac{l^2}{|Ш(E)|}$ dans la formule (2.2.2)), et on vérifie que m^2 soit différent de 0 (sinon on passe au D suivant).

4. On utilise le sous-algorithme 3.3.1 (présenté ci-dessous) pour calculer une liste $\mathcal{L}$ de répresentants des classes des formes quadratiques définies positives de discriminant D, (A, B, C), telles que $N|A$, A minimal et $B \equiv \beta \mod 2N$.
On accouple les formes (A, B, C) et (A', B', C') quand (A', B', C') est équivalent à $(CN, B, A/N)$.

5. On calcule la précision nécessaire par la formule :
$$p = \left\lceil \frac{d + \log(2) + \log(|\mathcal{L}|) + 2K(E)}{\log(10)} \right\rceil$$
où $K(E) = \frac{h(j(E))}{12} + \mu(E) + 1.946$ (voir la formule (1.1) et le Paragraphe 3.2.2),
$$d = 2|Ш(E)|R(E) = 2\frac{|E_t(\mathbb{Q})|^2 L'(E, 1)}{c(E)\omega_1(E)},$$
où la derniére égalité est obtenue grâce à la conjecture BSD.

6. On calcule
$$z = \sum_{(A,B,C)\in\mathcal{L}} \varphi\left(\frac{-B + \sqrt{D}}{2A}\right) \in \mathbb{C},$$
en utilisant la formule $\varphi(\tau) = \sum_{n\geq 1} \frac{a_n}{n} q^n$ ($q = e^{2i\pi\tau}$) et le fait que $\varphi\left(\frac{-B'+\sqrt{D}}{2A'}\right) = \overline{\varphi\left(\frac{-B+\sqrt{D}}{2A}\right)}$ si (A, B, C) et (A', B', C') ont été accouplés au pas 4.

7. Soit e l'exposant du groupe $E_t(\mathbb{Q}), l = \gcd(e, m^\infty) = \gcd(e, m^3), m' = ml$. Pour tout couple $(u, v) \in [0, m' - 1]^2$ on pose
$$z_{u,v} = \frac{lz + u\omega_1(E) + v\omega_2(E)}{m'}$$
et on calcule $x = \wp(z_{u,v})$. Pour tout (u, v) tel que $x \in \mathbb{R}$ on vérifie si x est proche à un nombre rationnel $X = \frac{m}{d^2}$ ayant pour dénominateur un carré, et en ce cas là on vérifie s'il existe un rationnel $Y = \frac{n}{d^3}$ tel que $(\frac{m}{d^2}, \frac{n}{d^3}) \in E(\mathbb{Q})$.

3.2 Remarques et améliorations à l'algorithme

3.2.1 Accouplement des formes

En utilisant la troisième partie du Théorème 2.1.7 et en remarquant que si la classe d'idéaux $[\mathfrak{a}]$ correspond à la forme (A, B, C), alors $[\mathfrak{a}^{-1}\mathfrak{n}]$ correspond à $(CN, B, A/N)$, on peut accoupler les formes quadratiques de la liste $\mathcal{L}$ et calculer $\varphi(\tau)$ pour environ la moitié d'elles.

3.2.2 Précision

Pour qu'il soit possible de reconnaître z avec la méthode des fractions continues, on a besoin que l'erreur soit

$$\varepsilon < \frac{1}{2q^2},$$

où q est le dénominateur de la fraction cherchée. En outre, vu que on somme $|\mathcal{L}|$ termes, l'erreur sur chacun d'eux doit être

$$\varepsilon' < \frac{1}{2q^2|\mathcal{L}|},$$

mais alors la précision nécessaire doit être

$$p > \frac{\log(2) + 2\log(q) + \log(|\mathcal{L}|)}{\log(10)}$$

Or, si P est le point qu'on va détecter, alors

$$\log(q) \leq h(P) \leq \widehat{h}(P) + K(E).$$

Enfin, on rappelle que le point P va être de la forme $P = \sqrt{|Ш(E)|}G + T$, où G est un générateur de la partie infinie du groupe de Mordell-Weil. Ainsi $h(P) = |Ш(E)|h(G) = |Ш(E)|R(E)$, ce qui prouve la formule donnée.
Dans les calculs des séries de la forme $\sum_{n\geq 1} x_n e^{kn}$,où k est une constante, on s'arrête évidemment dès que $e^{kn} < 10^{-p}$, c'est-à-dire $n > \frac{p}{k\log(10)}$.
Enfin, pour le calcul de $L'(E,1) = \sum_{n\geq 1} \frac{a_n}{n} E_1(kn)$, on utilise la minoration suivante :

$$E_1(x) = \int_x^\infty \frac{e^{-t}}{t} dt \leq e^{-x}$$

et on se ramène au cas précedent.

On remarque que le calcul le plus coûteux de l'algorithme est celui de z, qui demande environ $\frac{Ap}{\pi\sqrt{|D|}}$ termes.

3.2.3 Choix du disciminant fondamental

On remarque que le choix du plus petit discriminant fondamental n'est pas toujours le meilleur. En effet, vu que le nombre de termes de la série à calculer est un multiple de $A_{max}/\sqrt{|D|}$, où A_{max} est le plus grand des premiers coeffiecients des formes $(A, B, C) \in \mathcal{L}$. Donc, surtout si l'on utilise la méthode d'Atkin-Lehner (voir le Paragraphe 3.4), qui permet de garder petits les A, indépendamment du discriminant fondamental D, on est interessé à augmenter la taille de D, de façon que le nombre de termes à calculer diminue. Par contre, il faut remarquer aussi que pour le calcul de $L(E_D, 1)$ on a besoin d'un nombre de termes proportionnel à D, donc on ne peut pas non plus l'augmenter trop.

3.2.4 Boucle sur les $z_{u,v}$

On va maintenant expliquer le pas 7 de l'algorithme.
Soit $P \in E(\mathbb{Q})$ le point correspondant à $z \in \mathbb{C}/\Lambda_E$. P est de la forme $P = mG + T$, où $T \in E_t(\mathbb{Q})$ et G générateur de la partie infinie. Si l'ordre de T est premier avec m, on peut écrire $P = mG + mT'$, pour un certain $T' \in E_t(\mathbb{Q})$, donc le point $G + T'$ correspond à $z/m + \omega$, avec $\omega \in \Lambda_E/m$, ce qui explique les $z_{u,v}$.
Si, par contre, m n'est pas premier avec l'ordre de T, on a quand même $lP = m'G + lT$ (avec les notations de l'algorithme), et l'ordre de lT est premier avec m', donc on peut appliquer la méthode précedente avec lz à la place de z.

En outre, on peut se réduire à calculer seulement m' ou $2m'$ $z_{u,v}$ au lieu que $(m')^2$. En effet, on cherche un point $z = \lambda_1\omega_1(E) + \lambda_2\omega_2(E) \in \mathbb{C}/\Lambda_E$, qui correspond à un point à cordonnées réelles dans E, et cela est verifié si et seulement si :

1. $\lambda_2 \in \mathbb{Z}$ si $\mathrm{disc}(E) < 0$,
2. $\lambda_2 \in \frac{1}{2}\mathbb{Z}$ si $\mathrm{disc}(E) > 0$.

On obtient donc les points :

1. si $\mathrm{disc}(E) < 0$: $z_u = \frac{(l\,\mathrm{Re}(z) + u\omega_1(E))}{m'} + o\frac{\omega_1(E)}{2l}$, avec $o = \frac{\mathrm{Im}(z)}{\mathrm{Im}(\omega_2(E))}$ pour $u = 0, \ldots, m' - 1$,
2. si $\mathrm{disc}(E) > 0$: $z_u = \frac{l\,\mathrm{Re}(z) + u\omega_1(E)}{m'}$ et $z'_u = \frac{l\,\mathrm{Re}(z + u\omega_1(E))}{m'} + \frac{\omega_2(e)}{2}$ pour les mêmes u.

3.2.5 Calcul des séries comme polynômes

Du point de vue computationnel, pour évaluer une série $\sum_{n\geq 1} x_n q^n$ il est plus rapide de calculer le vecteur des x_n, et après l'évaluer comme polynôme (en multipliquant enfin par q le résultat obtenu), plutôt que sommer terme à terme, car il existe des algorithmes très rapides pour l'évaluation des polynômes, surtout si le vecteur des coefficients contient beaucoup de zéros, comme il arrive souvent dans notre cas, quand on travaille avec les a_n.

3.2.6 Calcul des a_n par récurrence

Quand le conducteur de la courbe, N, est très grand, il peut arriver que le nombre de termes à calculer pour la série des $\varphi(\tau)$ soit plus grand que la longueur maximale consentie pour un vecteur, appelons-la `L`.
En ce cas là, il faut travailler par récurrence, en essayant d'être le plus astucieux possible...
Soit $\texttt{nmax} < \texttt{L}^2$ le nombre de termes à calculer, et soit $\texttt{fois} = \lceil \texttt{nmax}/\texttt{L} \rceil$ le nombre minimal de vecteurs nécessaires pour contenir tous les coefficients (dans nos applications, $\texttt{L} \sim 16 * 10^6, \texttt{nmax} \sim 10^8$).

On commence par calculer le premier vecteur `vs`, avec les a_n de 1 à L (cela est très rapide à calculer en PARI GP, grâce à la fonction `ellan` (qui utilise Shank-Mestre pour les courbes générales, et Cornacchia pour les courbes à multiplication complexe, voir le Paragraphe 3.6).
On va garder le vecteur `vs` jusqu'à la fin des calculs.

Pour chacun des vecteurs suivants, on calcule les a_n en utilisant les formules :

$$\begin{array}{lcll} a_{pq} & = & a_p a_q & \text{si } \gcd(p,q) = 1 \\ a_{p^k} & = & a_p a_{p^{k-1}} - \chi(p) p a_{p^{k-2}} & \text{si } p \text{ premier,} \end{array}$$

où $\chi(p) = \begin{cases} 0 & \text{si } p \text{ est un premier de mauvaise réduction,} \\ 1 & \text{sinon.} \end{cases}$

- Si p premier $\geq \texttt{LL} = \sqrt{\texttt{nmax}+1}$, alors on sait qu'il ne peut apparaître que à la puissance 1, et on peut donc calculer tous les a_{pj} contenus dans le vecteur comme $a_p a_j$ où j est (pour nos valeurs de `L` et `nmax`) $<$ `L`, donc a_j est dans le vecteur `vs`.
- Pour ce qui concerne les autres n, qui ont tous leurs facteurs premiers $<$ `L`, on peut, par exemple, les decomposer comme `PQ` où `Q` est p^e, p le plus grand premier dans la factorisation. Alors, soit `P` soit `Q` est $<$ `L`, donc au moins une des deux valeurs de a_n est repérable dans le vecteur `vs`, et on appelle donc très rarement la fonction pour calculer les a_k, qui est plûtot couteuse en termes de temps.

Remarque
Il existe un algorithme, dû à Buhler et Gross, ([BG]), qui permet de stocker un nombre minimal de valeurs ($\sqrt{\texttt{nmax}}$) et de calculer tous les a_n avec le nombre minimal de multiplications possible. On s'est donc inspiré à celui-là, mais on n'a pas pû l'utiliser exactement tel qu'il est, car il ne calcule pas les a_n dans l'ordre (ce qui nous est très pratique pour l'évaluation comme polynômes) et il est très coûteux du point de vue du temps nécessaire pour l'évaluation des polynômes.

3.2.7 La constante de Manin et le groupe de Tate-Shafarevich

Dans l'algorithme, on suppose que la constante de Manin et la cardinalité du groupe de Tate-Shafarevich soient 1, mais il pourrait arriver que cela ne soit pas le cas.

Si $|Ш(E)| \neq 1$ au lieu que trouver un générateur du groupe de Mordell-Weil, on risque d'en trouver un multiple. Cela n'est pas grave pour notre but, vu que à partir d'un point d'ordre infini, il est facile de trouver un générateur, par contre cela va nous demander une précision plus haute, ce qui n'est pas du tout agréable dans l'algorithme. Donc il est mieux, dans cette éventualité, de diviser z par des petits entiers. D'autre coté, si la constante de Manin est différente de 1, alors il faut multipliquer z par des petits entiers, qui techniquement répresentent le degré d'isogénie entre la courbe E et la courbe de Weil forte dans sa même classe d'isogénie.

3.3 Sous-algorithmes

Sous-algorithme 3.3.1 *(calcule la liste $\mathcal{L}$)*

Etant donné un discriminant fondamental D et β tel que $\beta^2 \equiv D \mod 4N$, on calcule la liste $\mathcal{L}$ utilisée dans l'algorithme :

1. (*Initialisation :*) on calcule le nombre de classes de formes de discriminant D, $h(D)$.
   On initialise $\mathcal{L}_r, \mathcal{L} \leftarrow \emptyset$ et on choisit b tel que $b \equiv \beta \mod 2N$.
2. (*On remplit les listes :*) on pose $R \leftarrow \frac{b^2-D}{4N}$ et pour tout diviseur positif d de R :
   - on pose $f \leftarrow (dN, b, R)$, $f_r \leftarrow$ la forme quadratique réduite equivalente à f.
   - si $f_r \notin \mathcal{L}_r$, $\mathcal{L}_r \leftarrow \mathcal{L}_r \cup \{f_r\}$, on calcule $f' = (A', B', C')$ équivalente à f, toujours avec $N|A'$ et $B' \equiv B \mod 2N$, mais avec A' minimal (on utilise le Sous-algorithme 3.3.2) et on pose $\mathcal{L} \leftarrow \mathcal{L} \cup \{f'\}$

3. (*Fin ?*) Si $|\mathcal{L}| < h(D)$, $b \leftarrow b + 2N$ et on retourne au pas 2, sinon FIN.

Sous-algorithme 3.3.2 *(pour calculer une forme quadratique f' équivalente à f mais avec A minimal)*

Etant donné $f = (A, B, C)$ une forme quadratique définie positive de discriminant $D < 0$ avec $N|A$, on calcule $f' = (A', B', C')$ équivalente à f, avec $N|A, B' \equiv B \mod 2N$ et A' minimal.

1. (*Initialisation :*) on pose $u \leftarrow \frac{-B}{2A/N}, v_2 \leftarrow \frac{|D|}{(2A/N)^2}$.
   Avec une méthode raisonnable quelconque, on cherche un couple (c_0, d_0), avec $\gcd(c_0 N, d_0) = 1$ et tel que $(c_0 u + d_0)^2 + c_0^2 v_2 = m_0$ soit le plus petit possible pour la méthode choisie (par exemple, on peut poser $c_0 \leftarrow 0, d_0 \leftarrow 1$ et $m_0 \leftarrow 1$).
   On pose $L \leftarrow \sqrt{m_0/v_2}$; si $L \leq 1$ on sort le résultat (A, B, C) et on termine l'algorithme, sinon, on pose $c \leftarrow 0$ et on va au pas suivant.
2. (*Boucle sur c :*) on pose $c \leftarrow c + 1$. Si $c \geq L$ on va au pas 3, sinon on pose $d \leftarrow \lfloor -cu \rceil^*$ (l'entier le plus proche à $-cu$, premier avec cN) et $r \leftarrow (cu + d)^2 + c^2 v_2$.
   Si $r < m_0$, on pose $m_0 \leftarrow r, c_0 \leftarrow c, d_0 \leftarrow d$ et $L \leftarrow \sqrt{m_0/v_2}$. Aller au pas 2.
3. (*Trouver la forme :*) si $m_0 = 1$, on retourne (A, B, C), sinon on calcule (par l'algorithme d'Euclide étendu) a_0, b_0 tels que $a_0 d_0 - b_0 N c_0 = 1$, on retourne $f' = (A', B', C') = f(a_0 x + b_0 y, c_0 N x + d_0 y)$ et on termine l'algorithme.

Preuve de l'algorithme
Soit $\tau = \frac{-B+\sqrt{D}}{2A}$ la racine complexe associée à la forme quadratique (A, B, C), et soient $x = \mathrm{Re}(\tau) = -\frac{B}{2A}$, $y = \mathrm{Im}(\tau) = \frac{\sqrt{|D|}}{2A}$. Trouver une forme quadratique (A', B', C'), équivalente à (A, B, C) mais avec A' minimal équivaut à trouver τ' avec partie imaginaire maximale. On remarque que les rélations $N|A'$ et $B \equiv B' \mod 2N$ sont gardées par $\gamma \in \mathrm{SL}_2(\mathbb{Z})$ si et seulement si $\gamma \in \Gamma_0(N)$, donc γ est de la forme $\begin{pmatrix} a & b \\ Nc & d \end{pmatrix}$.

Dans ce cas là, $\mathrm{Im}(\gamma(\tau)) = \frac{\mathrm{Im}(\tau)}{|Nc\tau+d|^2}$. Alors on veut $|Nc\tau + d|^2$ minimal. On a :

$$|Nc\tau + d|^2 = (Ncx + d)^2 + N^2 c^2 y^2 = (cu + d)^2 + c^2 v_2,$$

avec les mêmes notations que dans l'algorithme. On peut obtenir trivialement la quantité m_0, en posant $c = c_0$ et $d = d_0$ (voir pas 1). Donc si l'on

veut qu'elle soit $< m_0$, il faut que $c < \sqrt{\frac{m_0}{v_2}} = L$. Dès qu'on a fixé c, il y a un seul d qui donne la valeur minimale : l'entier le plus proche a $-cu$ premier avec cN (car au pas 3 on doit appliquer l'algorithme d'Euclide étendu). □

3.4 Opérateurs d'Atkin-Lehner

Notations

On définit :

$$\begin{aligned}
\mathcal{H}_N^D &= \{\text{points d'Heegner de discriminant } D \text{ et niveau } N\},\\
\mathcal{H}_N^D(\beta) &= \{\tau \in \mathcal{H}_N^D, \text{ tels que, si } (A,B,C) \text{ est la forme quadratique associée à } \\
&\quad \tau, \text{ alors } B \equiv \beta \mod 2N\}.\\
\widehat{\mathcal{H}}_N^D(\beta) &= \mathcal{H}_N^D(\beta)/\Gamma_0(N).
\end{aligned}$$

Soit τ un point d'Heegner de discriminant D et niveau N, auquel est associée la forme quadratique (A,B,C), et soit Q un diviseur positif de N tel que $\gcd(Q, N/Q) = 1$. Alors, il existe $u, v \in \mathbb{Z}$ tels que la matrice $W_Q = \begin{pmatrix} uQ & v \\ N & Q \end{pmatrix}$ ait pour determinant Q.

On appelle W_Q une matrice d'Atkin-Lehner pour le diviseur Q de N (plus en général on pourrait prendre $W_Q = \begin{pmatrix} uQ & v \\ wN & zQ \end{pmatrix}$ avec $\det(W_Q) = Q$, mais il est facile de prouver que toutes ces matrices sont équivalentes modulo l'action de $\Gamma_0(N)$).

On peut vérifier que $W_Q^2 = Q\gamma$, avec $\gamma \in \Gamma_0(N)$, donc l'action de W_Q par transformations fractionnelles linéaires (c'est-à-dire $W_Q(\tau) = \frac{uQ\tau+v}{N\tau+Q}$) est une involution pour les fonctions invariantes par $\Gamma_0(N)$. Dans notre cas, $W_Q^2(\tau)$ est donc dans l'orbite de τ par l'action de $\Gamma_0(N)$.

Si on applique W_Q à la forme (A,B,C) on obtient (A_1,B_1,C_1), où :

$$\begin{aligned}
A_1 &= AQu^2 + BNu + (N^2/Q)C\\
B_1 &= 2Auv + B(N/Q)v + QBu + 2CN\\
C_1 &= (A/Q)v^2 + Bv + CQ.
\end{aligned}$$

Or, on peut vérifier que $N|A_1$ et $\gcd(A_1/N, B_1, C_1N) = 1$, donc $W_Q(\tau)$ est encore un point d'Heegner de niveau N et discriminant D.

En outre on a que $\phi = \pm\phi \circ W_Q$ et donc

$$\varphi(\tau) = \pm\varphi(W_Q(\tau)) + (\text{un point de torsion}),$$

où le signe est donné par $\varepsilon_Q = \prod_{p|Q} \varepsilon_p$, où ε_p est le "root number" local de E en p, donc on a

$$P = \sum_{\tau \in \widehat{\mathcal{H}}_N^D(\beta)} \varphi(\tau) = \sum_{\tau \in \widehat{\mathcal{H}}_N^D(\beta)} \varepsilon_Q \varphi(W_Q(\tau)) + (\text{un point de torsion})$$

On veut donc utiliser ces opérateurs pour remplacer les répresentants τ qu'on utilise dans l'algorithme 3.1.1 par des autres répresentants, de la forme $W_Q(\tau')$, mais qui aient partie imaginaire plus grande (A_1 plus petit) et donc qui nécessitent moins de termes à calculer. Le seul problème est que l'action de W_Q ne préserve pas la rélation $B_1 \equiv \beta \mod 2N$.

Soit (A, B, C) une forme quadratique de discriminant D, avec $B \equiv \beta'$ mod $2N$ où β' est une racine carrée quelconque de $D \mod 4N$. On veut savoir quelles sont les transformations W_Q qui envoyent (A, B, C) dans une forme (A_1, B_1, C_1) telle que $B_1 \equiv \beta \mod 2N$. Soit p un diviseur premier de N. On peut remarquer que :

- Si $p | \gcd(D, N)$, alors $v_p(\gcd(D, N)) = 1$ et toutes les racines carrées de $D \mod 4N$ sont $\equiv 0 \mod p$, donc $\beta \equiv \beta' \mod p$;
- si $p^k \parallel Q$, alors $B_1 \equiv -B \mod p^k$;
- si $p^k \parallel N/Q$, alors $B_1 \equiv B \mod p^k$;
- si $p \nmid \gcd(D, N)$, alors $p \nmid B_1, B, \beta, \beta'$ donc on n'a que deux racines carrées de $D \mod p^k$ (β et $-\beta$ évidemment) sauf dans le cas $p = 2$ et $v_2(N) \geq 2$, où on a quatre racines carrées $\mod 2^{k+1}$ (il y a un 2 qui vient du facteur 2 de $2N$). En ce cas là on a $\beta' \in \{\beta, -\beta, \beta+2^k, -\beta+2^k\}$ et si β' est une des deux dernières racines, il n'y a aucun moyen de l'envoyer sur β par une transformation W_Q.

Donc on construit

$$Q = d \prod_{p^k \parallel N, \beta' \not\equiv \beta \mod p^k} p^k,$$

avec $d | \gcd(D, N)$ et, quitte à vérifier $\beta' \equiv \pm\beta \mod 2^{v_2(N)+1}$, on est sûr d'obtenir un point d'Heegner dans $\widehat{\mathcal{H}}_N^D(\beta)$.
On a un analogue du Théorème 2.1.7 :

$$\varphi(W_Q((\mathfrak{l}, \beta))) = \varepsilon_Q \varphi((\mathfrak{l}\mathfrak{q}^{-1}, \beta')) + (\text{un point de torsion}),$$

avec $\mathfrak{q} = (Q\mathbb{Z} + \frac{-\beta'+\sqrt{D}}{2}\mathbb{Z})$. En outre si $\mathfrak{m} \sim (\mathfrak{l}\mathfrak{n}^{-1})$ (donc $\overline{\varphi((\mathfrak{l}, \beta))} = \varphi((\mathfrak{m}, \beta))$), mais ici $\mathfrak{q}^{-1} = \mathfrak{q}$, donc dans $\mathbb{C}/\Lambda_E$, à un point de torsion près,

on a :

$$\begin{aligned} & \overline{\varphi(W_Q((\mathfrak{l},\beta)))} = \varepsilon_Q \overline{\varphi((\mathfrak{lq}^{-1},\beta'))} = \varepsilon_Q \varphi(((\mathfrak{lq}^{-1}\mathfrak{n})^{-1},\beta')) \\ = \quad & \varepsilon_Q \varphi(((\mathfrak{lqn})^{-1},\beta')) = \varphi(W_Q(((\mathfrak{ln})^{-1},\beta))) = \varphi(W_Q((\mathfrak{m},\beta))) \end{aligned}$$

donc $(\mathfrak{l},\beta)$ peut être accouplé si et seulement si $W_Q((\mathfrak{l},\beta))$ peut.

On va maintenant décrire l'algorithme pour trouver une liste de formes quadratiques (avec leur poids) avec petits coefficients A, qui peut remplacer les deux sous-algorithmes utilisés pour l'Algorithme 3.1.1.
L'idée est de parcourir les formes (aN,b,c) avec a petit, et essayer de les envoyer dans une forme (A,B,C) avec $B \equiv \beta \mod 2N$ par une transformation W_Q.

Sous-algorithme 3.4.1 *Etant donnés D, N on trouve des bons réprésentants des τ.*

1. Choisir $\beta \in \mathbb{Z}/2N\mathbb{Z}, \beta^2 \equiv D \mod 4N$. Poser $U \leftarrow \emptyset, R \leftarrow \emptyset$.
2. Tant que $|R| \neq |Cl(\mathbb{Q}(\sqrt{D}))|$:
3. faire une boucle sur a de 1 à ∞ et $b \in \mathbb{Z}$ racine carrée de $D \mod 4N$ ($b \equiv \pm\beta \mod 2^{k+1}$, où $2^k \parallel N$) ;
4. faire une boucle sur les solutions s de l'équation $Ns^2 + bs + \frac{(b^2-D)}{4N} \equiv 0 \mod a$;
5. soit $f = \left(aN, b+2Ns, \frac{((b+2Ns)^2-D)}{4aN}\right)$;
6. faire une boucle sur les diviseurs positifs d de $\gcd(D,N)$
7. poser $Q \leftarrow d\prod_{(p^k \| N, b \not\equiv \beta \mod p^k)} p^k$, $g \leftarrow \frac{W_Q(f)}{Q}$ $(g \in \mathcal{H}_N^D(\beta))$.
8. Calculer les formes réduites équivalentes à g et $\overline{g}$ et si elles ne sont pas déjà dans R, les ajouter à R et ajouter à U f avec poids ε_Q si $g \sim \overline{g}$, $2\varepsilon_Q$ sinon.

Donc on trouve $z = \sum_{f \in U} \mathrm{wt}(f)\varphi(\tau_f)$, où $\mathrm{wt}(f)$ est le poids associé à f dans l'algorithme précedent.

3.5 La méthode de Cremona-Silverman

Je voudrais remercier Monsieur Cremona pour avoir eu la gentilesse de m'expliquer personellement l'idée de sa méthode.
A la fin de l'algorithme des points d'Heegner on doit reconnaître un nombre rationnel à partir d'un réel en précision flottante, en utilisant les fractions

continues. Cela nous impose de doubler la précision par rapport à la dimension du dénominateur de la fraction qui nous intéresse. En outre, si l'on utilise les fractions continues, on ne peut pas profiter du fait qu'on sait en avance que ce dénominateur va être un carré.

Cremona, en lisant un article de Silverman, "Computing rational points on rank 1 elliptic curves via L-series and canonical heights", a eu l'idée qu'on va décrire.

Soit $P \in E(\mathbb{Q})$ le point qu'on veut reconnaître. On connaît son approximation réelle, mais aussi sa hauteur canonique, grâce au théorème de Gross-Zagier (Théorème 2.2.1).
Or, on peut écrire la hauteur canonique d'un point comme une somme de hauteurs locales :

$$\widehat{h}(P) = \widehat{h}_\infty(P) + \sum_{p \text{ premier}} \widehat{h}_p(P)$$

On sait que la hauteur locale $\widehat{h}_p(P)$ est nulle si P n'a pas mauvaise réduction en p, donc il suffit de se restreindre aux premiers p qui divisent N.

Soit d le dénominateur de l'abscisse du point P. Si $p|d$, alors $\widehat{h}_p(P) = v_p(d)\log(p)$. Par conséquent on obtient la formule suivante :

$$\widehat{h}(P) = \widehat{h}_\infty(P) + \log(d) + \sum_{p|N, p\nmid d} \widehat{h}_p(P).$$

Le théorème suivant donne les formules pour calculer les hauteurs locales :

Théorème 3.5.1 *Soit E une courbe elliptique définie sur $\mathbb{Q}$ par une équation générale de Weierstrass, qui est minimale en p, et soit $P = (x, y) \in E(\mathbb{Q})$. Définissons les fonctions ψ_2 et ψ_3 par :*

$$\begin{aligned}\psi_2(P) &= 2y + a_1x + a_3\\ \psi_3(P) &= 3x^4 + b_2x^3 + 3b_4x^2 + 3b_6x + b_8.\end{aligned}$$

On a :

1. *Si $v_p(3x^2 + 2a_2x + a_4 - a_1y) \leq 0$ ou $v_p(2y + a_1x + a_3) \leq 0$, alors*

$$\widehat{h}_p(P) = \max(0, -v_p(x))\log(p),$$

2. *sinon, si $v_p(c_4) = 0$, alors soient $N = v_p(\mathrm{disc}(E))$ et $M = \min(v_p(\psi_2(P)), \lfloor \frac{1}{2}N \rfloor)$, alors*

$$\widehat{h}_p(P) = \frac{M(M-N)}{N}\log(p),$$

3. *sinon, si* $v_p(\psi_3(P)) \geq 3v_p(\psi_2(P))$, *alors*

$$\widehat{h}_p(P) = -\frac{2}{3} v_p(\psi_2(P)) \log(p),$$

4. *sinon,*

$$\widehat{h}_p(P) = -\frac{1}{4} v_p(\psi_3(P)) \log(p).$$

Comme conséquence de ce théorème on a que, étant donné un premier p divisant le conducteur d'une courbe elliptique E, on peut se restreindre à un ensemble fini de hauteurs locales $\widehat{h}_p(P)$ possibles, pour tout point P (voir le Paragraphe 3.5.1).
En outre, il existe un algorithme, qui ne dépend que de l'approximation réelle de l'abscisse de P (voir [Coh1] ou [Cre]) pour calculer la composante à l'infini de la hauteur canonique, $\widehat{h}_\infty(P)$. Donc, quitte à considérer un ensemble fini de hauteurs locales possibles pour les premiers p qui divisent le conducteur N, on peut calculer :

$$\log(d) = \widehat{h}(P) - \widehat{h}_\infty(P) - \sum_{p|N, p\nmid d} \widehat{h}_p(P).$$

Enfin, comme $d = c^2$ est un carré, on a :

$$2\log(c) = \log(c^2) = \widehat{h}(P) - \widehat{h}_\infty(P) - \sum_{p|N, p\nmid d} \widehat{h}_p(P),$$

et comme ça on peut calculer le dénominateur de la fraction cherchée avec une précision qui est environ la moitié de celle réquise par l'algorithme de base.
Une fois trouvé le dénominateur d, il faut récupérer le numérateur n, et pour cela il faut faire un peu plus attention. En effet, on ne peut plus profiter du fait qu'on cherche un carré, et donc la précision suffisante pour trouver d pourrait se révéler insuffisante pour n, mais cela généralement peut se résoudre facilement en décalant de quelques unités la valeur trouvée $\widetilde{n} = \lfloor xd \rceil$ où x est l'approximation réelle de l'abscisse du point P.

Remarque

Quand on utilise la méthode de Cremona-Silverman, il faut faire attention à calculer $L'(E,1)$ avec une précision suffisante. En effet, dans l'algorithme de base on n'a pas besoin d'une bonne précision, vu que $L'(E,1)$ sert juste à nous donner la précision nécessaire pour le calcul de φ, donc un arrondissement par excès ne peut pas être dangereux. Par contre, quand on utilise la méthode de Cremona, on a besoin de calculer avec une bonne précision la hauteur canonique du point qu'on va trouver, et cela dépend de $L'(E,1)$.

3.5.1 Les sous-algorithmes

Sous-algorithme 3.5.1 *Calcul de l'ensemble fini des hauteurs locales possibles.*

1. On initialise $R \leftarrow [0]$,
2. on fait une boucle sur les premiers $p|N$,
3. si $p^2|\operatorname{disc}(E)$ on calcule le vecteur V des $\widehat{h}_p(P)$ possibles avec le sous-algorithme 3.5.2,
4. on remplace $R \leftarrow R'$ où R' et un vecteur de longueur $\operatorname{len}(R') = \operatorname{len}(R)\operatorname{len}(V)$ (ici len denote la longueur d'un vecteur) rempli avec toutes les sommes possibles d'un élément de R et un élément de V,
5. à la fin de la boucle, on retourne R.

Remarques

- On a besoin de calculer les hauteurs locales possibles pour le premier p seulement si $p^2|\operatorname{disc}(E)$. En effet, si le point P a mauvaise réduction en p on a deux possibilités :
  1. Soit $p|c_4$, mais alors $p^2|N$ et donc $p^2|\operatorname{disc}(E)$,
  2. sinon $p \nmid c_4$ mais alors la seule possibilité de mauvaise réduction pour P en p est le cas 2) du théorème 3.5.1. Mais alors on a $N = 1$ et $M = 0$, donc il ne nous reste que $\widehat{h}_p(P) = 0$.
- En utilisant ce sous-algorithme, il se peut qu'on calcule aussi la hauteur local pour les premiers p qui divisent le dénominateur de l'abscisse du point qu'on veut trouver ; cela ne doit pas nous inquiéter, car, comme on verra dans le sous-algorithme 3.5.2, dans chaque vecteur des hauteurs locales possibles il y a 0 donc une des possibilités sera quand même la bonne.

Sous-algorithme 3.5.2 *Etant donnés une courbe elliptique E et un premier p, on calcule l'ensemble fini des hauteurs locales possibles $\widehat{h}_p(P)$, d'un point $P \in E(\mathbb{Q})$ quelconque.*

1. On calcule le type de Kodaira K de la courbe E au premier p.
2. Si $K = \mathrm{I}_m$, alors on calcule $n = v_p(\operatorname{disc}(E))$, $n' = \lfloor \frac{m}{2} \rfloor$ et on retourne $V = [0, \ldots, \frac{j^2}{n-j}\log(p), \ldots, \frac{n'^2}{n-n'}\log(p)]$ (j va de 0 à n') ;
3. si $K = \mathrm{I}_m^*$ alors on retourne $V = [0, -1, -1 - m/4]\log(p)$;
4. si $K = \mathrm{II}$ ou $K = \mathrm{II}^*$ ou $K = \mathrm{I}_0^*$: on retourne $V = [0]$.

5. si $K = \mathrm{III}$, on retourne $V = [0, -1/2]\log(p)$;
6. si $K = \mathrm{IV}$, on retourne $V = [0, -2/3]\log(p)$;
7. si $K = \mathrm{III}^*$ on retourne $V = [0, -3/2]\log(p)$;
8. si $K = \mathrm{IV}^*$ on retourne $V = [0, -4/3]\log(p)$.

Remarque Pour les types de Kodaira I,II,II* et I_m la liste des hauteurs locales possibles dérive directement du Théorème 3.5.1, par contre, pour les autres types de Kodaira la démonstration que la liste est bien celle donnée par l'algorithme ci-dessus est un peu plus laborieuse : en fait, dans les autres cas on peut se réduire, sans perte de généralité, au cas ou $P = (0, 0)$ et sa réduction est additive. A ce point là il suffit d'étudier les possibles valuation en p de a_3 (si on est dans le cas (3) du theoreme 3.5.1, c'est-à-dire si le type de Kodaira est IV ou IV*) ou b_8 (dans le cas (4) du théorème, i. e. pour les types de Kodaira III, III*, I_0^* et I_m^*), en suivant les étapes de l'algorithme de Tate (voir [Sil]), et pour cela on renvoye à l'article de Cremona, Prickett et Siksek, [CPS].

Sous-algorithme 3.5.3 *Etant donné $z \in \mathbb{C}/\Lambda_E$ correspondant à un point rationnel de la courbe E on le reconnaît avec la méthode de Cremona-Silverman :*

1. On calcule l'approximation réelle de l'abscisse du point P cherché : $x = \wp(z)$;
2. on calcule la hauteur à l'infini du point P (en ne connaissant que x) $\widehat{h}_\infty(P)$;
3. on calcule le vecteur C des $\frac{\widehat{h}(P) - \widehat{h}_\infty(P) - \lambda_i}{2}$ où les λ_i sont les hauteurs locales possibles calculées avec l'algorithme 3.5.1.
4. on fait une boucle sur les éléments du vecteur C et pour chacun d'eux on essaye de reconnaître l'abscisse du point P comme une fraction ayant pour dénominateur $\lfloor C[i]^2 \rceil$.

3.6 Calcul des a_p sur une courbe à multiplication complexe

Comme on a vu au paragraphe 1.10, si la courbe elliptique E a multiplication complexe par un ordre quadratique imaginaire, alors on peut profiter du théorème 1.10.5 pour avoir un algorithme très rapide pour le calcul des a_p :

Sous-algorithme 3.6.1 *Etant donné une courbe E et un premier p, on calcule a_p.*

1. Si $p < 100$ on calcule a_p avec la fonction de Jacobi et on termine l'algorithme, sinon on continue.
2. On vérifie si $j(E) \in \{j(\tau) | \tau$ les 13 nombres quadratiques imaginaires associés aux ordres de multiplication complexe$\}$, si oui on pose $D =$ le discriminant de l'ordre quadratique trouvé et on continue, sinon il faut calculer a_p avec les méthodes habituelles car E n'a pas de multiplication complexe.
3. Si $D \notin \{-3, -4, -8, -16\}$, on vérifie que $\left(\frac{D}{p}\right) \neq -1$ (si cela est le cas on retourne 0 et on termine l'algorithme) et on cherche a et b tels que $a^2 - Db^2 = 4p$ (avec l'algorithme de Cornacchia). Cela est équivalent à $\frac{a+b\sqrt{D}}{2}\frac{a-b\sqrt{D}}{2} = p$, donc grâce au théorème 1.10.5 on sait que, quitte à changer le signe de $\pi = \frac{a+b\sqrt{D}}{2}$, on aura : $a_p = a$.
4. Di $D = -3, -4, -8, -16$, l'algorithme est un peu plus compliqué mais l'idée est la même : on cherche à factoriser $p = \pi\overline{\pi}$ avec $\pi = \frac{a+b\sqrt{D}}{2}$ de façon que a_p soit égal à a, quitte à le multiplier par une racine de l'unité dans le corps quadratique $\mathbb{Q}(\sqrt{D})$.
5. On retourne a.

Chapitre 4

Quelques calculs avec la méthode des points d'Heegner

4.1 Courbes associées aux nombres congruents

On sait que si n est un nombre congruent, alors la courbe elliptique donnée par l'équation

$$y^2 = x^3 - n^2 x$$

a pour rang 1.

Cette classe de courbes permet des calculs plus rapides par rapport à des courbes quelconques, car toute courbe associée à un nombre congruent est dotée de multiplication complexe, donc le calcul des a_n, qui est la partie qui prend la plupart du temps dans l'algorithme, va être beaucoup plus rapide.

Exemple 4.1.1 *On va commencer par chercher un point d'ordre infini sur la courbe* $E : y^2 = x^3 - 157^2 x$.

La courbe E a pour discriminant $\mathrm{disc}(E) = 958468597212736$ et pour conducteur $N = 788768$.
On calcule tout d'abord la valeur de $L'(E, 1)$ et on trouve

$$L'(E, 1) = 11.425944500734015260343961 18$$

(on travaille au début en précision 28).
Après on cherche un discriminant fondamental négatif D tel que D soit un carré modulo $4N$, et que a_p soit égal à -1 pour tout premier $p | \gcd(D, N)$. Le premier D qui satisfait ces conditions est -31 mais en calculant $L(E_{-31}, 1)$

on trouve qu'il est 0, c'est-à-dire la tordue quadratique par -31 n'a pas pour rang 0, et donc elle ne va pas marcher pour nos objectifs.
On passe donc au discriminant fondamental suivant, $D = -39$, et on vérifie que en ce cas là $L(E_{-39}, 1)$ est bien différent de zéro.
On cherche une liste $\mathcal{L}$ de formes quadratiques définies positives de discriminant D par le sous-algorithme 3.4.1, ce qui nous donne :

$$\mathcal{L} = \{f_1 = (788768, 1275547, 515684), f_2 = (1577536, 1275547, 257842)\},$$

où chacune des deux formes a pour poids 2.
On calcule la précision nécessaire, qui est 29 si l'on utilise les améliorations dûes à Delaunay-Watkins et Cremona-Silverman.
On calcule enfin

$$z = 2\,\mathrm{Re}(\varphi(f_1) + \varphi(f_2))$$

(où $\varphi(f) = \varphi(\tau(f))$ avec $\tau(f) = \frac{-B+\sqrt{D}}{2A}$, si $f = (A, B, C)$, $D = B^2 - 4AC$) et on trouve $z = -5.6391112750083176600769616631$, ou bien, en réduisant modulo la période réelle ω_1, $z = 0.010989034871117828903024377062$. On calcule aussi $m = \frac{l}{\sqrt{|Ш(E)|}} = 4$.

A ce point là on peut trouver le dénominateur de l'abscisse du point cherché avec la méthode de Cremona-Silverman, $d = 2825630694251145858025$, ensuite reconnaître l'abscisse
$x = -16613623166818526754080 4/2825630694251145858025$ et enfin le point P cherché, qui a hauteur canonique $\widehat{h}(P) = 54.60$:

$$\left(-\frac{16613623166818526754080 4}{2825630694251145858025}, \frac{16766162445683433540481211146978 2006}{15020109520013551810876147023 5125}\right).$$

Remarque :
Cette courbe est le premier exemple significatif avec lequel on a testé l'algorithme qu'on était en train d'implémenter. Initialement, c'est à dire sans aucune amélioration à l'algorithme de base, cela requierait le double de la précision donnée dans l'exemple, et prenait un temps d'environ une minute et 43 secondes.

Maintenant, après avoir implementé les améliorations de Delaunay-Watkins et Cremona-Silverman et en ayant compilé en `C` la fonction crée en `gp` (en modifiant manuellement la gestion de la mémoire) on a besoin de la moitié de la précision initiale, et on a réussi à réduire le temps des calculs à environ 7 secondes, c'est à dire moins d'un dixième du temps que cela prenait au début.

Exemple 4.1.2 *On va considerer maintenant une courbe avec conducteur plus grand, qui nous a rendu indispensable de travailler par récurrence sur les* a_n *(vu que le nombre de termes nécessaires pour le calcul des* φ *est* 26373340*) et d'améliorer la gestion de la mémoire faite par* `gp`.

On considère la courbe

$$E : y^2 = x^3 - 373^2 x.$$

Elle a pour discriminant $\mathrm{disc}(E) = 172358602780396096$ et pour conducteur $N = 4452128$.
Le premier discriminant fondamental qui satisfait nos conditions est $D = -7$, mais on a $L(E_{-7}, 1) = 0$, donc on passe au D suivant, c'est-à-dire $D = -31$, et celui-là donne une tordue quadratique de rang 0.
La liste $\mathcal{L}$ est donnée par :

$$\mathcal{L} = \{f_1 = (4452128, 8719537, 4269325), f_2 = (4452128, 7141169, 2863591)\},$$

avec poids réspectivement 1 et 2.
On trouve donc $z = -1.42518326222058367065723646463676464375266404 - 0.13576499386946671481866471518556359344453548 4i$, qui réduit modulo ω_1 et ω_2 donne $z = -0.06753332352591652247058931278112870930730827 79$.

On calcule aussi $m = \frac{l}{\sqrt{|Ш(E)|}} = 4$. A partir de cela, en utilisant la méthode de Cremona-Silverman on obtient le point P, de hauteur canonique $\widehat{h}(P) = 90.78$. Son abscisse est :

$$x = -\frac{26821864374337055617547436886939517 6964}{67785461744298149414991653534241702 25}$$

et son ordonnée est :

$$y = \frac{41174881115960557363822796228215014576995245560839387280966}{17648379667315310098307681890164392953422127043848359625}.$$

Remarque : Ce calcul, qui aurait été impossible avec la méthode de base, est maintenant possible en un peu plus qu'une minute.

4.2 Récherche de points rationnels sur des courbes sans multiplication complexe

Quand la courbe E qu'on considère n'a pas de multiplication complexe, les calculs deviennent plus lents, mais on peux quand même étudier des courbes avec conducteur environ 10^8.

4.2.1 Points rationnels avec hauteur canonique élevée

On va commencer par deux exemples où on va trouver des points rationnels de grande hauteur canonique, respectivement 139.17 et 207, mais qui ont conducteur assez petit, de l'ordre de 10^5.

Exemple 4.2.1 *On considère la courbe*

$$E : y^2 + xy = x^3 - x^2 - 751055859x - 7922219731979.$$

Cette courbe a pour discriminant $\mathrm{disc}(E) = -88105384606705423220736$ et pour conducteur $N = 11682$.
On calcule $L'(E,1) = 4.013671856334650766897620263$ et on choisit le discriminant fondamental $D = -8$.
On trouve la liste $\mathcal{L} = \{f = (11682, 7244, 1123)\}$, avec $\mathrm{wt}(f) = 1$. On vérifie que la précision nécessaire est 70, et on calcule

$$z = -0.540330064224636478330902378256956873507981399025645700409467856560722 3 \\ -1.475500791419810864025119974932737874960865677890307462938774699400487i$$

ou bien, en réduisant modulo le réseau engendré par ω_1 et ω_2 :

$$z = -0.006807029831013577303682014851989187869912516356197409526080948848615448.$$

Enfin, en utilisant l'algorithme de Cremona, on trouve le point $P = (X_P, Y_P)$, avec :

$$X_P = \frac{59083304348120361249634159120027026593412059174649381755087 15}{15220376368374676634206259927599005212016051725014908208 9}$$

$$Y_P = \frac{86534389787820682966636929945300376332526538620580656032808219915433139849264790585912631 57}{18777513706194836989285546274566522715333535975432238786875725932873448715886945141 87},$$

qui a pour hauteur canonique $\widehat{h}(P) = 139.17$.
Remarque Ce calcul ne prend que 6 secondes.

Exemple 4.2.2 *Soit E la courbe elliptique donnée par l'équation :*

$$y^2 + xy = x^3 + x^2 - 1417922740x - 20796724086960.$$

Elle a pour discriminant $\mathrm{disc}(E) = -43835976099463181784973312 00$ et pour conducteur $N = 20650$.
On calcule $L'(E,1) = 5.088747865274074141976080138$.
On utilise le discriminant fondamental $D = -24$ et on trouve la liste $\mathcal{L} = \{f_1 = (20650, 18324, 4065), f_2 = 20650, 18324, 4065)\}$.
On calcule que la précision nécessaire est 100.

On calcule

$z = -3.778252665437689154151053016779950208195474347077267885825241611670579029370760865307952151610167162$
$-4.291824339405899790291778802331763567388834565352719465500289941552779608144892247214008892324660063i$

c'est-à-dire

$z = -0.004828502248281707064368338102355889436034303403178937162035723728646687404758761361387049278576231024,$

si l'on réduit modulo ω_1 et ω_2. Donc on obtient le point $P = (X_P, Y_P)$, avec :

$X_P = 505062837595750143720506428511947314678896967286422083646852988896489974505522820126868807744 1/$
$703209176911306272589254194250747008902385577832260247765651674861582331257361204648321$

$Y_P = 358865033051809105017875527004855987654935284305935788193213729521288036440311636992663440913989039$
$464080756097358075043110999066380376389317/186477652984180683104560487827396947246335470628191797$
$916761981480385629699747865971843098721394146447844893799803477347066007383 69$

Remarque Ce calcul prend environ 12 secondes.

4.2.2 Courbes elliptiques avec conducteur assez grand

Exemple 4.2.3 *On commence par étudier la courbe E donnée par l'équation*

$$y^2 + xy + y = x^3 + x^2 + 33x + 113,$$

qui a pour conducteur $N = 3283526$.

On calcule $L'(E, 1) = 42.761739188120392833000528 78$ et on utilise le discriminant fondamental $D = -103$.
On obtient la liste

$$\begin{aligned}\mathcal{L} &= \{f_1 = (3283526, 2550185, 495157), f_2 = (3283526, 5051021, 1942486),\\ &\quad f_3 = (3283526, 4730677, 1703908)\},\end{aligned}$$

avec $\mathrm{wt}(f_1) = 2$, $\mathrm{wt}(f_2) = -2$, $\mathrm{wt}(f_3) = 1$.
Il faut travailler avec précision 10.
On trouve $z = -5.735329262$, ou bien, réduit modulo ω_1, $z = 0.7534891338$, ce qui donne le point

$$P = \left(-\frac{193569}{101761}, \frac{237584308}{32461759}\right),$$

qui a hauteur canonique $\widehat{h}(P) = 13.18$.
Remarque Pour la recherche de ce point, l'algorithme prend 1 minute et quelques secondes.

Exemple 4.2.4 *On considère la courbe*

$$E : y^2 + y = x^3 + x^2 + 42x + 248.$$

Elle a pour conducteur $N = 28349787$.
On a $L'(E, 1) = 4.22652624450674475706417848 3$.
On choisit le discriminant fondamental $D = -8$ et on trouve la liste $\mathcal{L} = \{f = (28349787, 48697364, 20912267)\}$, avec $\mathrm{wt}(f) = 2$.
On travaille en précision 6 et on obtient $z = -0.634951$, ce qui nous donne enfin le point $P = (11, 46)$, de hauteur canonique $\widehat{h}(P) = 2.83$.
Remarque Avec ce choix du discriminant fondamental $D = -8$, l'algorithme employe 7 minutes et 45 secondes pour trouver le point P. Par contre, si on fait un choix plus astucieux de D, comme $D = -995$, on peut diminuer le nombre de termes nécessaires pour le calcul de la série, et on obtient le point P en 2 minutes et 40 secondes, donc moins de la moitié du temps.

4.3 Calculs de la cardinalité du groupe de Tate-Shafarevich des tordue quadratiques

Quand on calcule la tordue quadratique d'une courbe elliptique E de rang 1 par un discriminant fondamental choisi comme dans l'algorithme des points d'Heegner, on trouve une courbe E_D de rang 0.

On s'intéresse à calculer la cardinalité : $|Ш(E_D)|$.

En effet, on peut utiliser la formule BSD pour le cas rang 0 :

$$L(E_D, 1) = \omega_1(E_D)\frac{|Ш(E_D)|c(E_D)}{|(E_D)_t(\mathbb{Q})|^2},$$

qui nous donne :

$$|Ш(E_D)| = \frac{L(E_D, 1)|(E_D)_t(\mathbb{Q})|^2}{\omega_1(E_D)c(E_D)}$$

Exemples

1. Si on considère la courbe de rang 1 avec le conducteur le plus petit, c'est-à-dire $E : y^2 + y = x^3 - x$, qui a pourconducteur $N = 37$, les discriminant fondamentaux possibles (jusqu'à -1000) dans l'algorithme sont 125, et on trouve que $|Ш(E_D)|$ vaut : `1` 75 fois (60%), `4` 13 fois (10.4%), `9` 22 fois (17.6%), `16` une fois (0.8%), `25` 11 fois (8.8%) et `49` 3 fois (2.4%).

2. Soit E la courbe elliptique d'équation $y^2 - y = x^3 + x^2$, qui a pour conducteur $N = 43$. On trouve 132 discriminants fondamentaux (jusqu'à -1000), et on calcule les $|Ш(E_D)|$ correspondants. On trouve `1` 66 fois (50%), `4` 19 fois (14.4%), `9` 25 fois (19%), `16` 8 fois (6%), `25` 8 fois (6%), `36` une fois (0.8%), `49` 4 fois (3%) et `121` une fois (0.8%).
3. Soit E la courbe elliptique d'équation $y^2 + xy + y = x^3 - x^2$ et conducteur $N = 53$. En faisant les mêmes calculs on trouve 131 discriminants fondamentaux, et les $|Ш(E_D)|$ correspondants valent : `1` 66 fois (50.4%), `4` 21 fois (16%), `9` 25 fois (19%), `16` 6 fois (4.6%), `25` 8 fois (6.1%), `36` 2 fois (1.5%) et `49` 3 fois (2.3%).

Annexe A

Le code du programme en PARI GP

```
install(poleval,GG);
install(polevalshort,GGL);
install(ellheightoo,GGp);
install(veceint12,GGGp);

\\ ********** CREMONA LAMBDALIST **********

lambda1(E,p,discr)=
{
  local(kod,m,n,nl);
  kod=elllocalred(E,p)[2];
  if(kod>4,
    m=kod-4;
    n=valuation(discr,p);
    nl=1+floor(m/2);
    return(vector(nl,j,(j-1)^2/n-(j-1))*log(p)),
    if(kod<-4,
      m=-kod-4;
      return([0,-1,-1-m/4]*log(p)),
    if((kod==2)||(kod==-2)||(kod==-1),
      return([0]),
    if(kod==3,
      return([0,-1/2]*log(p)),
```

```
    if(kod==4,
      return([0,-2/3]*log(p)),
    if(kod==-3,
      print([0,-3/2]*log(p));
      return([0,-3/2]*log(p)),
    if(kod==-4,
      return([0,-4/3]*log(p)),
  )))))))
}

lambdalist(E,N,discr)=
{
  local(np,ans,nans,i,p,ll,nll,nnewans,newans);
  plist=factor(N)[,1];
  np=length(plist);
  ans=[0];nans=1;
  for(j=1,np,
      p=plist[j];
      if(discr%(p*p)==0,
         ll=lambda1(E,p,discr);
         nll=length(ll);
         nnewans=nans*nll;
         newans=vector(nnewans);
         i=1;
         for(k=1,nans,
             for(l=1,nll,
                 newans[i]=ans[k]+ll[l];
                 i++;
                )
             );
         nans=nnewans;ans=newans)
      );
  return(ans);
}

\\ *********  RADMOD  **********

/* donne le vecteur des racines carrées de a mod p^e, p premier impair */

radmodpe(a,p,e)=
```

```
{
  local(pe,fe,ce,res,v,a1,y);
  pe=p^e;
  a=a%pe;
  if(a==0,
    fe=e>>1;
    ce=e-fe;
    res=Mod(1,pe)*p^ce*vector(p^fe,i,i-1);
    return(res);
  );
  v=valuation(a,p);
  a1=a/(p^v);
  if(v%2 || kronecker(a1,p)==-1,return([]));
  y=truncate(sqrt(a1+O(p^(e-v))));
  return(Mod(1,pe)*p^(v/2)*concat(vector(p^(v/2),i,
         y+(i-1)*p^(e-v)),vector(p^(v/2),i,-y+(i-1)*p^(e-v))));
}


/* donne le vecteur des racines carrées de a mod 2^e */

radmod2e(a,e)=
{
  local(p,pe,pe2,fe,ce,res,v,a1,y);
  p=2;
  if(e==1,return([Mod(a,p)]));
  pe=1<<e;
  pe2=1<<(e-1);
  a=a%pe;
  if(a==0,
    fe=e>>1;
    ce=e-fe;
    res=Mod(1,pe2)*p^ce*vector(p^(fe-1),i,i-1);
    return(res);
  );
  v=valuation(a,p);
  if(v%2,return([]));
  a1=a/(p^v);
  if(e-v==1,
    return(Mod(1,pe2)*p^(v/2)*vector(p^(v/2-1),i,a1+(i-1)*p^(e-v)));
```

```
  );
  if(e-v==2,
    if(a1%4!=1,return([]));
    res=Mod(1,pe2)*p^(v/2)*vector(p^(e/2-1),i,2*i-1);
    return(res);
  );
  if(a1%8!=1,return([]));
  y=truncate(sqrt(a1+O(p^(e-v+1))));
  res=Mod(1,pe2)*p^(v/2)*concat(vector(p^(v/2),i,
        y+(i-1)*p^(e-v-1)),vector(p^(v/2),i,-y+(i-1)*p^(e-v-1)));
  return(res);
}

/* racine carree de z=Mod(a,N) pour tout entier N */

radmod(z)=
{
  local(a,N,fa,lpr,liexp,lfa,p,ex,res,v,leres,lv);
  v=[];
  a=lift(z);
  N=z.mod;
  fa=factor(N);
  lpr=fa[,1];
  liexp=fa[,2];
  lfa=length(lpr);
  for(i=1,lfa,
    p=lpr[i];
    ex=liexp[i];
    if(p==2,
      res=radmod2e(a,ex),
      res=radmodpe(a,p,ex)
    );
    if(res==[],return([]));
    if(i==1,
      v=res,
      leres=length(res);
      lv=length(v);
      v=vector(leres*lv,i,chinese(res[i%leres+1],v[ceil(i/leres)]));
    )
  );
```

```
  return(v);
}

\\ ****************************************

pgfp(n)=
{
 local(list,l);
 list=factor(n)[,1];
 l=length(list);
 return(list[l]);
}

\\l'entier le plus proche de x,premier avec y

nip(x,y)=
{
  local(n,c);
  n=round(x);
  if(n>x,c=-1,c=1);
  while(gcd(n,y)!=1,n+=c;c=-(c+sign(c)););
  return(n);
}

solmod(a,b,c,n)=
{
  local(solm);
  solm=[];
  for(i=0,n-1,
    if((a*i^2+b*i+c)%n==0,
      solm=concat(solm,i)));
  return(solm);
}

Wq(f,Q,N)=
{
  local(bez,u,v,A,B,C,A1,B1,C1);
  if(N%Q!=0,error("Q doesn't divide N"));
  if(gcd(Q,N/Q)!=1,error("gcd(Q,N/Q)!=1"));
  bez=bezout(Q,-N/Q);
```

```
  u=bez[1];
  v=bez[2];
  A=component(f,1);
  B=component(f,2);
  C=component(f,3);
  A1=A*Q*u^2+B*N*u+N^2*C/Q;
  B1=2*A*u*v+B*N/Q*v+Q*B*u+2*C*N;
  C1=A/Q*v^2+B*v+C*Q;
  return(Qfb(A1,B1,C1));
}

cong(f,N)=
  qfbred(Qfb(component(f,1)/N,-component(f,2),component(f,3)*N));

isin(el,v,len)=
{
  local(fn);
  fn=0;
  for(l=1,len,if((v[l]==el),fn=1));
  return(fn);
};

algo(E,D,N)=
{
  local(beta,U,R,h,found,bvec,pgcd,len,V,primi,exponents,L,a,b,
    F,M,sol,lsol,s,f,Q,g,gr,gcr,gc,gcrapp,grapp,wt,Qp,lq,sg);
  if(!issquaremod(D,4*N),error("D n est pas un carre mod 4N"));
  bvec=lift(radmod(Mod(D,4*N)));
  pgcd=gcd(D,N);
  len=length(bvec);
  beta=bvec[1];
  h=qfbclassno(D);
  R=vector(h);
  U=[];
  V=factor(N);
  primi=V[,1];
  exponents=V[,2];
  L=length(primi);
  found=0;
  a=0;
```

```
  while(1,
    a++; \\boucle sur a
    for(i=1,len, \\boucle sur  b
      b=bvec[i];
      F=1;
      for(pr=1,L,
        M=primi[pr]^exponents[pr];
        if((b%M)!=(beta%M),F*=M;));
      sol=solmod(N,b,(b^2-D)/(4*N),a);
      lsol=length(sol);
      for(j=1,lsol,\\ boucle sur les solutions
        s=sol[j];
        f=Qfb(a*N,b+2*N*s,((b+2*N*s)^2-D)/(4*a*N));
        fordiv(pgcd,d,
          Q=d*F;
          g=Wq(f,Q,N);
          if(component(g,2)%(2*N)==beta,
          gc=cong(g,N);
          gr=qfbred(g);
          gcr=qfbred(gc);
          grapp=isin(gr,R,found);
          gcrapp=isin(gcr,R,found);
          if(!grapp || !gcrapp,
            wt=0;
            Qp=factor(Q)[,1];lq=length(Qp);
            sg=prod(j=1,lq,ellrootno(E,Qp[j]));
            if(!grapp,
            found++;R[found]=gr;
            wt+=sg;);
            if(gr!=gcr && !gcrapp,
              found++;R[found]=gcr;
              wt+=sg;);
            U=concat(U,[[f,wt]]););
            if(found==h,
              print("List = ",U);
            return(U));
  )))))
};

issquaremod(a,b)=
```

```
{
  local(v,fact,e,start,l);
  v=factor(b);
  fact=v[,1];
  e=v[,2];
  l=length(e);
  start=1;
  if(fact[1]==2,
    start=2;
    if(e[1]==2,if(a%4!=0 && a%4!=1,return(0)));
    if(e[1]>=3,if(a%8!=0 && a%8!=4 && a%8!=1,return(0))));
  for(i=start,l,if(kronecker(a,fact[i])==-1,return(0)));
  return(1);
};

w(D)=
{
  if(D==-3,return(6));
  if(D==-4,return(4));
  return(2);
};

omeg(n)=
{
  return(length(factor(n)[,1]));
};

logp(x)=
{
  return(log(max(exp(1),abs(x))));
};

deux_etoile(x)=
{
  if(x==0,return(1));
  return(2);
};


hf(x)=
```

```
{
  return(log(max(abs(numerator(x)),abs(denominator(x)))));
};

Kell(E)=
{
  local(b2,mu);
  b2=E.b2;
  mu=(log(abs(E.disc))+logp(E.j))/6+logp(b2/12)+log(deux_etoile(b2));
  return(hf(E.j)/12+mu+1.946);
};


apk(ap,k,p,N)=
{
  if(N%p==0,
    return(ap^k));
  a=1;
  b=ap;
  for(i=2,k,
    t=b;
    b=ap*b-p*a;
    a=t;
    );
  return(b);
}


Phi(E,N,nmaxv,qv,len:small,L:small,D,gotLd,gotdL,Ld,dL,qld,nmaxld,nmaxdl)=
{
  local(vs,nmax,phi,fois:small,done,last,lung,ql,
    qtemp,qldl,qldtemp,wei,vs2,todo,ap);
  nmax=vecmax(nmaxv);
  if(!gotdL,dL/=2;if(nmax<nmaxdl,nmax=nmaxdl));
  if(!gotLd,Ld/=2;if(nmax<nmaxld,nmax=nmaxld));
    if(nmax>=L^2,print("L too small");
      L=(ceil(sqrt(nmax))+1):small;print("new L",L));
  phi=vector(len,i,0.);
  fois=ceil(nmax/L):small;
```

```
 \\ case k=1

 lung=min(nmax,L);
 until(1,vs=ellan(E,lung));until(1,vs2=vector(lung,j,vs[j]/j));
 for(i=1,len,
   phi[i]+=polevalshort(vs2,qv[i],min(nmaxv[i],lung)));

\\ case k>1

 if(fois>1,
   default(primelimit,min(10^9,nmax));
   ql=vector(len,j,qv[j]^L);
   qtemp=vector(len,j,1);
   qldl=0;
   qldtemp=0;
   if(!gotLd,
     qldl=qld^L;
     qldtemp=1;);
   for(k=2,fois,
     done=(k-1)*L;
     lung=if(k==fois,nmax%L,L);
     todo=done+lung;
     /*je veux un vecteur de longueur lung,
       contenant an/n pour n=done+1...done+lung*/
     vs2=vector(lung,i,1);
     en=done+1;
     LL=ceil(sqrt(nmax+1));
     until(1,
       forprime(p=LL,min(todo,10^9),
       ap=ellap(E,p);
       for(j=ceil((done+1)/p),todo\p,
         vs2[p*j-done]=ap*vs[j];
       ));
       for(p=10^9+1,todo,
       if(isprime(p),
         ap=ellap(E,p);
         for(j=ceil((done+1)/p),todo\p,
           vs2[p*j-done]=ap*vs[j];
     ))));
     for(i=1,lung,
```

```
      if(vs2[i]==1,
        fa=factor(en);
        plist=fa[,1];
        elist=fa[,2];
        llen=length(plist);
        if(plist[llen]<LL,
          R=en;
          while(R>L,
          vs2[i]*=apk(vs[plist[llen]],elist[llen],plist[llen],N);
          R/=(plist[llen]^elist[llen]);
          llen--;
          );
        vs2[i]*=vs[R]));
        vs2[i]/=en;
        en++;
    );
    qtemp=vector(len,j,qtemp[j]*ql[j]);
    for(i=1,len,
      if(nmaxv[i]>done,
         phi[i]+=polevalshort(vs2,qv[i],
          if(nmaxv[i]<todo,nmaxv[i]%L,lung))*qtemp[i]));
      if(!gotdL && done<nmaxdl,
        last=if(nmaxdl<k*L,nmaxdl%L,lung);
        tt=ceil((last-done)/10^6):small;
        for(i=1,tt,
          beg=done+(i-1)*10^6+1;
          end=min(done+i*10^6,nmaxdl);
          until(1,wei=veceint12(2*Pi/sqrt(N),beg,end));
          for(n=beg,end,if(vs2[n-done],dL+=vs2[n-done]*wei[n-beg+1])));
            if(!gotLd && done<nmaxld,
            qldtemp*=qldl;
            if(nmaxld<k*L,lung=nmaxld%L);
            until(1,vs2=vector(lung,j,vs2[j]*kronecker(D,j+done)));
            Ld+=polevalshort(vs2,qld,lung)*qldtemp;
  ))));
  if(!gotdL,dL*=2);
  if(!gotLd,Ld*=2*qld);
  phi=vector(len,i,phi[i]*qv[i]);
  return([phi,Ld,dL]);
}
```

```
punto(E,z,height,expo,lambdas,l,P)=
{
  local(pt,x,r,cvec,cs,c,X,Y,p,e);
  pt=real(ellztopoint(E,z));
  x=pt[1];
  r=2*ellheightoo(E,pt);
  cvec=vector(l,i,exp((height-r-lambdas[i])/2));
  for(i=1,l,
    cs=round(cvec[i],&e);
    if(cs>cvec[i],c=-1,c=1);
    if(cs,
    X=round(cs^2*x,&e);Y=ellordinate(E,X/cs^2);
    XX=X;
    k=1;
    while(Y==[] && k<50,
      XX=X+k;
      Y=ellordinate(E,XX/cs^2);
      if(Y==[],
        XX=X-k;
        Y=ellordinate(E,XX/cs^2));
        k++;
      );
      if(Y!=[],p=[XX/cs^2,Y[1]];
      if(length(ellpow(E,p,expo))>1,return(p))));
      cs+=c;
      if(cs,
        X=round(cs^2*x,&e);
        XX=X;
        Y=ellordinate(E,XX/cs^2);
        k=1;
        while(Y==[] && k<50,
          XX=X+k;
          Y=ellordinate(E,XX/cs^2);
          if(Y==[],
          XX=X-k;
          Y=ellordinate(E,X/cs^2));
          k++;
       );
       if(Y!=[],p=[XX/cs^2,Y[1]];
```

```
         if(length(ellpow(E,p,expo))>1,return(p)))));
  return([]);
}

 calcq(f)=
{
  local(A,B,C,D,tau);
  A=component(f,1);B=component(f,2);C=component(f,3);
  D=B^2-4*A*C;
  tau=(-B+sqrt(D))/(2*A);
  return(exp(2*I*Pi*tau));
};




goodred(p,N)={if(N%p==0,return(0),return(1))};


heegner(e)=
{
  local(E,om1,om2,Et,tors,volE,Ered,N,c,epsilon,dL,vv,nmaxv,
   qv,d,P,ok,pgcd,Ld,len:small,z,phi,expo,l,m1,zu,m,discr,
   comp1,comp2,Amax,height,L:small,Imax,Jmax,om,KE,nmax,q,mc,
   mcapp,list,ph,gotLd,alg,Dvec,apok,Dlen,D,vc,nmaxld,qld,
   lambdas,llam,gotdL,wei,fois:small,beg,end);
  L=round(1.6*10^7):small;
  Imax=1;\\Tate shafarevitch
  Jmax=1;\\Manin constant
  default(realprecision,28);
  E=ellinit(e);
  print("We consider the curve E = ",vector(5,i,E[i]));
  nmaxld=0;
  qld=0;
  discr=E.disc;
  print("Wich has discriminant = ",discr);
  om1=E.omega[1];
  om2=E.omega[2];
  if(discr>0,om=2*om1,om=om1);
  tors=elltors(E);
```

```
Et=tors[1];
if(length(tors[2])==0,expo=1,
if(length(tors[2])==1,expo=tors[2][1],
expo=max(tors[2][2],tors[2][1])));
Ered=ellglobalred(E);
N=Ered[1];
print("And conductor N = ",N);
c=Ered[3];
epsilon=ellrootno(E);
volE=E.area;
KE=Kell(E);
ok=1;
gotdL=0;
if(Ered[2]!=[1,0,0,0],error("l'equation n'est pas minimale!"));
if(epsilon!=-1,error("epsilon different de -1!!!"));

\\ ***********  calcul de dL  **********

nmaxdl=ceil(28*log(10)*sqrt(N)/(2*Pi));
\\ il faut calculer dL avec une bonne precision pour avoir
\\ une bonne precision pour la hauteur canonique!!!
if(nmaxdl<=L,gotdL=1,nmaxdl=L);
until(1,vv=ellan(E,nmaxdl);vv=vector(nmaxdl,j,vv[j]/j));
fois=ceil(nmaxdl/10^6):small;
dL=0.;
for(i=1,fois,
  beg=(i-1)*10^6+1;
  end=min(i*10^6,nmaxdl);
  until(1,wei=veceint12(2*Pi/sqrt(N),beg,end));
  for(n=beg,end,if(vv[n],dL+=vv[n]*wei[n-beg+1])));
  dL*=2;
  print("dL = ",dL);
  if(abs(dL)<10^(-2),ok=0);
  if(ok,
    Dvec=[];
    forstep(D=-3,-1000,-1,
    if(isfundamental(D) && issquaremod(D,4*N),
      default(realprecision,28);
      pgcd=gcd(N,D);
      apok=1;
```

```
  fordiv(pgcd,p,if(isprime(p) && ellap(E,p)!=-1,apok=0));
  if(apok,Dvec=concat(Dvec,D))));
Dlen=length(Dvec);
for(i=1,Dlen,
  ok=1;
  D=Dvec[i];
  pgcd=gcd(D,N);
  if(i%3==1,
    nmax=ceil(28*log(10)*sqrt(N*Dvec[min(i+2,Dlen)]^2/
      vecmin(vector(min(3,Dlen-i+1),j,gcd(Dvec[i+j-1],N))))/(2*Pi));
    until(1,vv=ellan(E,min(nmax,L)));
  );

  \\ **********  calcul de  Ld  **********

  gotLd=0;
  nmax=ceil(28*log(10)*sqrt(N*D^2/pgcd)/(2*Pi));
  if(nmax<=L,gotLd=1,nmax=L);
  Ld=0.;
  q=exp(-2*Pi/sqrt(N*D^2/pgcd));
  qt=1;
  B=1000000;
  qb=q^B;
  foisB=ceil(nmax/B);
  for(i=1,foisB,
  until(1,vc=vector(if(i==foisB,nmax%B,B),i,vv[i]/i*kronecker(D,i)));
  Ld+=poleval(vc,q)*qt;qt*=qb;
  );
  Ld*=2*q;
  if(abs(Ld)<10^(-27),ok=0);\\else
  if(ok,

  \\ **********  list & flag **********

    list=algo(E,D,N);
    len=length(list);
    Amax=0;
    for(i=1,len,if(component(list[i][1],1)>Amax,
         Amax=component(list[i][1],1)));
```

```
 \\ **********   changement de  precision  **********

 d=2*(Et^2*dL)/(c*om);
 P=ceil((d+log(len)+2*KE)/(2*log(10)));
 print("We must work with precision = ",P);
 default(realprecision,P);
 E=ellinit(e);
 om1=E.omega[1];
 om2=E.omega[2];
 if(discr>0,om=2*om1,om=om1);
 if(P>28,
   gotdL=0;
   nmax=ceil(P*log(10)*sqrt(N)/(2*Pi));
   if(nmax<=L,gotdL=1,nmax=L);
   until(1,vv=ellan(E,nmax));until(1,vv=vector(nmax,j,vv[j]/j));
   fois=ceil(nmax/10^6):small;
   dL=0.;
   for(i=1,fois,
     beg=(i-1)*10^6+1;
     end=min(i*10^6,nmax);
     until(1,wei=veceint12(2*Pi/sqrt(N),beg,end));
     for(n=beg,end,if(vv[n],dL+=vv[n]*wei[n-beg+1])));
     dL*=2;
);
z=0.;

\\ **********   calcul de  z  **********

nmaxv=vector(len,i,ceil(((P)*log(10)*component(list[i][1],1))/
   (Pi*sqrt(-D))));
qv=vector(len,i,calcq(list[i][1]));
if(!gotLd,
  nmaxld=ceil((P)*log(10)*sqrt(N*D^2/pgcd)/(2*Pi));
qld=exp(-2*Pi/sqrt(N*D^2/pgcd)));
if(!gotdL,
  nmaxdl=ceil((P)*log(10)*sqrt(N)/(2*Pi)));
alg=Phi(E,N,nmaxv,qv,len,L,D,gotLd,gotdL,Ld,dL,qld,nmaxld,nmaxdl);
phi=alg[1];
Ld=alg[2];
dL=alg[3];
```

```
height=(Et^2*dL)/(c*om);
for(i=1,len,
  if(list[i][2]%2!=0,
    z+=list[i][2]*phi[i],
    z+=list[i][2]*real(phi[i])));
print("z = ",z);
comp2=imag(z)/imag(om2);comp1=(real(z)-comp2*real(om2))/om1;
comp1=round(comp1);comp2=round(comp2);
z=z-(comp1*om1+comp2*om2);
print("zred = ",z);
mc=om*(c*sqrt(-D)*(w(D)/2)^2)/(4*volE*Et^2)*2^(omeg(pgcd))*Ld;
mcapp=round(mc);
m=sqrtint(mcapp);
print("m = ",m);
if(m!=0,
  lambdas=lambdalist(E,N,discr);
  llam=length(lambdas);
  l=gcd(expo,m^3);
  m1=m*l;
  for(i=1,Imax,
    for(j=1,Jmax,
      for(u=0,m1-1,
        if(discr>0,
          zu=j/i*(l*real(z)+u*om1)/m1;
          ph=punto(E,zu,height,expo,lambdas,llam,P);
          if(ph!=[],
            print("found a non torsion point in ",gettime(),"ms");
            return(ph));
          zu=j/i*((l*real(z)+u*om1)/m1+om2/2);
          ph=punto(E,zu,height,expo,lambdas,llam,P);
          if(ph!=[],
            print("found a non torsion point in",gettime(),"ms");
            return(ph)),
        \\else
          zu=j/i*((l*real(z)+u*om1)/m1-((imag(z)/imag(om2))*om1/2)/m);
          ph=punto(E,zu,height,expo,lambdas,llam,P);
          if(ph!=[],
            print("found a non torsion point in ",gettime(),"ms");
            return(ph))
    )));
```

```
            error("I get out of the loop without finding a point,
              something didn't work!!!!!!!!!!!");
  )))))
}
```

Bibliographie

[BG] J. Buhler et B. Gross, *Arithmetic on elliptic curves with complex multiplication. II.*, Inventiones matematicae, vol. 79, pp. 11-30, 1985.

[Coh1] H. Cohen,*A Course in Computational Algebraic Number Theory*, GTM 138, Springer-Verlag, 1997.

[Coh3] H. Cohen, *Number Theory*, to be published.

[Cre] J. Cremona, *Algorithms for Modular Elliptic Curves*, Cambridge Univ. Press, second ed., 1997.

[CPS] J. Cremona, M. Prickett et Samir Siksek, *Height Difference Bounds For Elliptic Curves over Number Fields*, Journal of Number Theory 116(1) (2006), 42-68.

[Del] C. Delaunay, *Formes modulaires et invariants de courbes elliptiques définies sur Q*, Thèse de Doctorat, Université Bordeaux I, 2002.

[Kna] . Knapp, *Elliptic Curves*, Princeton Univ. Press, Princeton, New Jersey, 1992.

[Kob] . Koblitz, *Introduction to Elliptic Curves and Modular Forms*, GTM 97, Springer-Verlag, New York, 1984.

[Sil] J. H. Silverman, *Advanced Topics in the arithmetic of Elliptic Curves*, GTM 151, Springer-Verlag, 1994.

[ST] J. H. Silverman et J. Tate, *Rational Points on Elliptic Curves*, UTM, Springer-Verlag, 1992.

[Wat] M. Watkins, *Some Remarks on Heegner Point Computations*.