THÈSE

présentée pour obtenir le grade de

DOCTEUR DE L'ÉCOLE POLYTECHNIQUE

Spécialité :

INFORMATIQUE

par

Pierrick GAUDRY

Titre de la thèse :

ALGORITHMIQUE DES COURBES HYPERELLIPTIQUES ET APPLICATIONS À LA CRYPTOLOGIE

Soutenue le 12 décembre 2000 devant le jury composé de :

M.	Philippe FLAJOLET	Président
MM.	James DAVENPORT	Rapporteurs
	Ming-Deh HUANG	
	Jean-François MESTRE	
Mme	Pascale CHARPIN	Examinateurs
MM.	Jean-Marc COUVEIGNES	
	Gerhard FREY	
	François MORAIN	(Directeur)
	Nigel SMART	

Remerciements

La première personne que je souhaite remercier est mon directeur de thèse, François Morain. Lors de mon stage de DEA, il m'a fait découvrir un domaine de recherche pour lequel il a su me passionner et qui est finalement devenu le sujet de cette thèse. Il est par la suite resté constamment disponible pour me guider et répondre à mes questions ; je lui serai toujours redevable pour ces quelques années où il m'a beaucoup appris.

Le jour de la soutenance de thèse est un événement très important qui marque au moins symboliquement la fin d'une phase dans la vie. Dans ce contexte, le rôle du jury est capital, et je suis très reconnaissant envers les personnes qui ont accepté d'y participer.

Merci tout d'abord à Philippe Flajolet pour avoir présidé ce jury. Il a mené de main de maître le déroulement de la soutenance.

Merci à James Davenport, pour avoir accepté la tâche de rapporteur.

Merci à Ming-Deh Huang qui lui aussi a assumé cette lourde reponsabilité malgré la barrière de la langue.

Je remercie Jean-François Mestre, troisième rapporteur, qui m'a suggéré plusieurs améliorations d'algorithmes. Je suis particulièrement enchanté par la collaboration qui a débuté à cette occasion.

Je remercie chaleureusement Pascale Charpin pour avoir participé au jury et s'être interessée à mon travail.

Merci à Jean-Marc Couveignes : c'est aussi grâce à lui qu'en DEA j'ai débuté sur le sujet, et je suis heureux qu'il fasse aujourd'hui partie du jury.

Je remercie Gerhard Frey pour le mois très enrichissant que j'ai passé dans son équipe à Essen, ainsi que pour avoir accepté d'être dans le jury.

Je remercie Nigel Smart, qui lui aussi a accepté de faire partie du jury, et avec qui j'ai par ailleurs eu grand plaisir à travailler sur la descente de Weil.

La préparation et la rédaction de cette thèse sont indissociables pour moi du lieu dans lequel cela c'est fait : le LIX. J'ai vécu les petits et les grands événements du labo, du traditionnel barbecue de fin d'année au changement de locaux du printemps 1999, et je profite de cette occasion pour remercier Michel Weinfeld qui a dirigé le labo durant mon séjour. Merci aussi à Jean-Marc Steayert qui dirige l'équipe Algorithmique, et qui est toujours disponible pour répondre à des questions de tout ordre.

Le fonctionnement d'un laboratoire est complètement dépendant du personnel non chercheur dont le travail est trop souvent passé sous silence : je remercie Evelyne Rayssac qui assume les tâches administratives, ainsi que les ingénieurs systèmes : Jean-Marc Vinet et Houy Kuoy. C'est aussi grâce à ces personnes que j'ai pu travailler dans de bonnes conditions.

Un labo, c'est de plus un grand nombre de thésards, postdocs, stagiaires. Je remercie tout d'abord Reynald Lercier qui m'a beaucoup appris lors de mon début de thèse. Ensuite Guillaume Hanrot et Emmanuel Thomé avec qui j'ai eu le plaisir de partager mon bureau, et par-là même un bon nombre de réflexions ; Dominique Rossin, Nicolas Gurel, Dominique Poulalhon, Alain Plagne incomparables compagnons de pause-café.

Durant ma thèse, j'ai largement profité de collaborations diverses. Ce fut à chaque fois une expérience extrêmement enrichissante, chacun apportant ses connaissances pour bâtir un résultat. Un grand merci donc à Mireille Fouquet et Robert Harley, avec qui j'ai eu (et j'ai toujours !) grand plaisir à travailler. Merci à Iwan Duursma, à Florian Hess, à Nigel Smart, à Andreas Enge auprès de qui j'ai beaucoup appris. Merci aussi à Éric Schost qui m'a enseigné les rudiments du calcul formel et des bases de Gröbner, les machines que nous avons fait souffrir ensemble s'en souviennent encore !

Certains des calculs que j'ai été amené à faire ont été exécutés sur les machines de l'UMS Medicis. Je remercie Joël Marchand et Teresa Gomez-Diaz pour l'admirable administration de ce centre de calculs. J'en profite aussi pour saluer et remercier les joyeux thésards du GAGE : Gregoire Lecerf, Alexandre Sedoglavic et Anne Fredet pour avoir supporté mes incessantes questions existentielles sur les systèmes polynomiaux.

En dehors du labo, je souhaiterais remercier Daniel Augot pour avoir mis en place l'Action Courbes, ce qui m'a permis de rencontrer une foule de gens intéressants, en particulier les membres du projet Codes. Je remercie aussi Preda Mihăilescu pour sa bonne humeur indéfectible et les discussions que nous avons eues. Merci à Philippe Satgé pour s'être intéressé à mon travail commun avec Éric Schost : sa lecture attentive de l'article et ses commentaires nous ont grandement éclairé. Je remercie Franck Leprevost avec qui j'ai eu l'occasion de discuter à plusieurs reprises au début de ma thèse et qui a eu la patience de répondre à des questions parfois naïves. Enfin, lors de mon séjour en Allemagne, j'ai rencontré Andreas Stein et Steven Galbraith ; ce fut un réel plaisir de confronter nos idées, et de prendre des bières ensemble.

Je terminerai par une pensée pour mes proches, amis et famille, qu'ils soient matheux, informaticiens ou pas du tout. En vrac : mes parents, mes grand-parents, mes sœurs, Fred et Catherine, Bob, Inès, Nanta, Schmürtz, Keumar, Chef, GBS, PBN, cRil, God, Béné et j'en oublie qui ne me pardonneront jamais !

Et pour finir je remercie Émilie, qui, au jour le jour, a partagé mes joies et supporté mes mauvaises humeurs.

Table des matières

Introduction **1**

Partie I Courbes hyperelliptiques et invariants d'Igusa **7**

1 Jacobiennes de courbes hyperelliptiques **9**

1.1 Définition de la Jacobienne 9

1.2 Variétés abéliennes 16

1.3 Courbes sur les corps finis : conjectures de Weil 19

1.4 Automorphismes, supersingularité et p-torsion 22

1.5 Courbes hyperelliptiques 28

1.6 Courbes hyperelliptiques sur $\mathbb{C}$ 31

1.7 Cas particulier du genre 2 35

2 Invariants de Jacobiennes $(2,2)$-décomposables **41**

2.1 Groupe d'automorphismes d'une courbe de genre 2 41

2.2 Cas générique 43

2.3 Groupe diédral 47

2.4 Groupe de Klein 49

2.5 Exemples numériques 50

2.6 La courbe $X_1(13)$ 52

2.7 Annexe : formulaire 53

3 Formes modulaires de Siegel : vers des équations modulaires en genre 2 **55**

3.1 Formes modulaires de Siegel 55

3.2 Application au genre 2 60

3.3 Équations modulaires en dimension 2 65

Partie II Algorithmique, calcul de la cardinalité 71

4 Loi de groupe dans la Jacobienne d'une courbe hyperelliptique 73
4.1 Représentation de Mumford 73
4.2 Algorithme de Cantor 75
4.3 Genre 2 : formules de Spallek et de Harley 77

5 Algorithmes élémentaires pour le calcul de la cardinalité 81
5.1 Algorithmes génériques 81
5.2 Méthodes d'approximation 85
5.3 Estimation de temps de calcul pour les tailles cryptographiques 88

6 Cardinalité de courbes particulières 91
6.1 Courbes de Koblitz et courbes à multiplication complexe 91
6.2 Courbes à multiplication réelle 93
6.3 Opérateur de Cartier-Manin 97

7 Algorithme de Schoof en genre 2 101
7.1 Algorithme de Schoof–Pila–Kampkötter 101
7.2 Polynômes de division de Cantor 104
7.3 Recherche efficace d'un élément de ℓ-torsion 107
7.4 Construction de diviseurs de 2^k-torsion 111
7.5 Combinaison avec d'autres algorithmes, résultats 113

8 Vers une extension Elkies–Atkin en genre supérieur 117
8.1 Construction du polynôme modulaire Ξ_ℓ 117
8.2 Exemples en genre 1 et 2 121
8.3 Motifs de factorisation 125
8.4 Application au calcul de cardinalité 130

Partie III Logarithme discret 133

9 État de l'art du calcul du log discret dans les Jacobiennes 135
9.1 Méthodes pour un groupe générique 135
9.2 Attaque de Frey-Rück 141
9.3 Attaque de Rück 145
9.4 Méthodes sous-exponentielles en genre « grand » 145

10 Algorithme sous-exponentiel générique **149**
10.1 Modèle générique de friabilité . . . 149
10.2 Algorithme . . . 153
10.3 Algèbre linéaire creuse . . . 154
10.4 Complexité . . . 156
10.5 Groupes non cycliques . . . 161

11 Calcul d'index en genre «petit» **165**
11.1 Notion de diviseur friable . . . 165
11.2 Algorithme et complexité . . . 166
11.3 Quelques astuces pour accélérer les calculs . . . 169
11.4 Résultats pratiques . . . 173
11.5 Conséquences cryptographiques . . . 174

12 Calcul d'index en genre 2 **177**
12.1 Description de l'algorithme . . . 177
12.2 Analyse . . . 183
12.3 Mise en pratique . . . 185

13 Application à la Restriction de Weil **189**
13.1 Restriction de Weil d'une courbe elliptique . . . 189
13.2 Théorème d'existence d'une courbe hyperelliptique . . . 193
13.3 Conséquences cryptographiques . . . 194

Conclusion **197**

Bibliographie **199**

Introduction

Historiquement, l'étude des courbes elliptiques a débuté avec pour motivation l'étude d'intégrales. En effet, pour calculer une intégrale de la forme $\int \frac{dx}{\sqrt{x^3+ax+b}}$, liée à la longueur d'un arc d'ellipse, on est naturellement amené à étudier la courbe $y^2 = x^3 + ax + b$. Les acteurs principaux de ces premières études du XIX$^{\mathrm{e}}$ siècle sont Abel, Jacobi, Weierstraß et Riemann. À la fin du XIX$^{\mathrm{e}}$ siècle et au début du XX$^{\mathrm{e}}$, les problèmes liés aux courbes algébriques ont commencé à être abordés d'un point de vue plus algébrique qu'analytique par l'école allemande, très florissante. Celle-ci comprenait parmi d'autres Weber, Artin, Kronecker, Klein, dont les travaux sont encore cités aujourd'hui.

Au cours du XX$^{\mathrm{e}}$ siècle, la formalisation de la géométrie algébrique s'est mise en place. Différentes approches furent tentées avant que les définitions actuelles ne soient adoptées, de sorte que de nombreux ouvrages de cette époque restent écrits dans un langage qui n'a pas perduré. Cette formalisation s'est accompagnée de bons nombres de résultats. Le sujet est désormais assez bien balisé, et des ouvrages de référence se sont imposés [Sil86, Sil94]. Quelques faits marquants sont : le théorème de finitude du rang d'une courbe elliptique sur $\mathbb{Q}$ par Mordell ainsi que sa conjecture sur la finitude d'une courbe de genre supérieur, démontrée bien plus tard par Faltings ; le théorème de Hasse sur le cardinal d'une courbe elliptique sur un corps fini ; les théorèmes de Deuring sur la multiplication complexe ; la généralisation par Weil du théorème de Hasse à des variétés plus générales (courbes et variétés abéliennes) ; les congrès d'Antwerp sur les formes modulaires ; la théorie de la multiplication complexe étendue aux variétés abéliennes par Shimura et Taniyama. Deux conjectures fondamentales ont été énoncées : celle de Birch et Swinnerton-Dyer relie le rang et la fonction L d'une courbe elliptique, et celle de Shimura–Taniyama–Weil affirme que toute courbe elliptique sur $\mathbb{Q}$ est modulaire. La preuve récente de cette dernière implique le théorème de Fermat et a eu un écho jusque dans la presse non spécialisée.

Avec le développement de l'informatique, un nouvel objectif est apparu : savoir résoudre efficacement des problèmes de théorie des nombres. La création de nouveaux algorithmes, leur analyse, leur implantation, tout ceci forme désormais une discipline à part entière : la théorie algorithmique des nombres. Des ouvrages font références sur le sujet [PZ89, Coh93, Coh99] et un congrès international lui est consacré, ANTS [AH94, Coh96, Buh98, Bos00].

Il y a une quinzaine d'années, l'algorithmique sur les courbes elliptiques s'est vraiment épanouie. Deux articles fondateurs sont celui de Schoof [Sch85] donnant le premier algorithme de calcul de cardinalité s'exécutant en temps polynomial déterministe, ainsi que celui de Lenstra [Len87] montrant l'utilité des courbes pour factoriser les entiers. Peu après, les frères Chudnovsky [CC86] et Goldwasser et Kilian [GK86] ont poussé un peu plus loin cette idée afin d'utiliser les courbes elliptiques pour prouver la primalité de presque tous les nombres premiers. Les quinze années suivantes ont été mises à profit pour étendre, modifier, améliorer et surtout rendre efficace en pratique ces algorithmes. Citons Atkin et Morain pour la primalité [AM93], et Elkies, Atkin, Couveignes, Morain, Lercier, Dewaghe, Müller ([Sch95] et [Ler97] sont des références complètes sur le sujet) pour le calcul de cardinalité. Citons aussi le programme de calcul du rang d'une courbe elliptique que Cremona distribue gratuitement [Cre].

Parallèlement à ces améliorations pour les courbes elliptiques, des travaux ont débuté afin d'étendre les résultats et le savoir-faire à des classes plus générales de courbes. Pila [Pil90] est le premier à avoir étendu l'algorithme de Schoof à la dimension supérieure. D'autre part, s'appuyant sur l'algorithme de Cantor [Can87] qui permet de calculer dans les Jacobiennes de courbes hyperelliptiques, de nombreux travaux ont été réalisés spécifiquement pour ces courbes. Notons que pour les courbes plus générales, les travaux de Huang et Ieradi [HI91] fournissent une version effective du théorème de Riemann-Roch et permettent de calculer dans leur Jacobienne, cela restant moins efficace que l'algorithme de Cantor pour les courbes hyperelliptiques. Kampkötter [Kam91] a ainsi donné une version de l'algorithme de Schoof adaptée à ces dernières ; Adleman et Huang [AH92] ont étendu l'algorithme de Goldwasser et Kilian aux courbes de genre 2, prouvant ainsi que la primalité d'un nombre peut être déterminée en temps polynomial probabiliste. Lenstra, Pila et Pomerance [LPP93] ont quant à eux étendu l'algorithme de Lenstra pour la factorisation. Le calcul du rang des Jacobiennes n'est pas encore bien balisé dans le cas général, mais dans le cas du genre 2 [CF96] et quelques autres, c'est désormais faisable.

Ces extensions d'algorithmes provenant des courbes elliptiques sont restées jusqu'ici d'un intérêt surtout théorique et peu d'efforts ont été faits pour les implanter efficacement. L'intérêt pratique est plutôt lié à la découverte par Miller et Koblitz [Mil87, Kob87, Kob89] de l'utilisation possible des courbes elliptiques, puis hyperelliptiques en cryptographie.

Cryptologie

Depuis quelques années la cryptographie est en plein essor et sort des domaines réservés (défense, banque) pour être utilisée de plus en plus par le grand public. Dans ce contexte, la cryptographie à clef publique joue un rôle essentiel car elle est bien adaptée à Internet. L'idée est de rendre le système asymétrique : les clefs vont par paire (publique, privée). La clef publique est distribuée et tout le monde peut s'en servir pour chiffrer des messages que seul le détenteur de la clef privée pourra déchiffrer. Ce genre de système permet aussi la signature électronique : la clef privée est utilisée pour signer le document et tout le monde peut vérifier la validité de cette signature grâce à la clef publique.

La sécurité des protocoles remplissant ces tâches repose sur la difficulté présumée d'un problème mathématique. L'exemple le plus célèbre est celui de la factorisation des nombres entiers, problème qui est mis à profit dans le système RSA [ARS78]. Un autre problème classique est le suivant : on se fixe un groupe cyclique fini $G = \langle g \rangle$ d'ordre N, noté multiplicativement. Étant donné un entier $0 < x < N$, il est aisé de calculer g^x par une méthode binaire (pourvu que la loi de groupe soit calculable). Réciproquement, étant donnés g et h dans G, il peut être délicat de retrouver x tel que $h = g^x$. C'est ce qu'on appelle le problème du logarithme discret.

Le protocole de Diffie–Hellman [DH76] est fondé sur ce problème et permet à deux protagonistes de se fabriquer un secret commun. Ce protocole est très simple et fut le premier à être proposé. Les deux agents sont traditionnellement appelés Alice et Bob et on suppose qu'ils disposent uniquement d'un moyen de communication susceptible d'être espionné. Alice commence par choisir un entier x_A au hasard. Elle calcule $h_A = g^{x_A}$ et l'envoie à Bob. Celui-ci choisit au hasard un entier x_B, calcule $h_B = g^{x_B}$ et l'envoie en retour à Alice. Celle-ci calcule alors $h_B^{x_A}$ et Bob calcule $h_A^{x_B}$. Ces deux valeurs sont égales à $g^{x_A x_B}$ et forment le secret commun h_{AB} d'Alice et Bob. Les seules informations qui circulent sur le canal sont G, N, g, h_A, h_B. On voit tout de suite que si le problème du logarithme discret est facile dans G, alors un espion peut aisément retrouver x_A et x_B et donc h_{AB}. Par contre, s'il n'existe pas d'algorithme rapide (connu) pour le log discret, il semble impossible pour un espion d'obtenir de l'information sur h_{AB}, et Alice et Bob peuvent utiliser cette valeur en toute confiance. Il n'y a pas strictement équivalence entre la difficulté du log discret et la sécurité du protocole [Mau94, MW96]. Toutefois il en existe de nombreux autres, pour faire des tâches variées, et dont la sécurité est plus ou moins bien prouvée sous l'hypothèse de la difficulté du log discret.

Historiquement, les premiers groupes réputés pour avoir un log discret difficile furent les groupes multiplicatifs de corps finis, mais il existe une attaque sous-exponentielle dans ces groupes. Puis les courbes elliptiques et les Jacobiennes de courbes hyperelliptiques furent proposées [Mil87, Kob87, Kob89], ainsi que des variantes à base de groupes de classes de corps de nombres [BW88].

Dans n'importe quel groupe G, il existe un algorithme pour calculer le log discret : on les essaie tous, les uns après les autres. Bien entendu cela n'est faisable que si le groupe n'est pas trop grand. En fait il existe des algorithmes plus performants et qui donnent le résultat en un nombre d'opérations de l'ordre de la racine carrée de l'ordre du groupe. Plus exactement, le temps de calcul est de l'ordre de la racine carrée du plus grand facteur premier du groupe. Il est donc nécessaire que l'ordre du groupe ne soit pas friable.

Ainsi, si l'on veut une sécurité suffisante vis-à-vis de ces attaques inévitables, il est nécessaire d'avoir un groupe d'ordre ayant un facteur de taille au moins 10^{40} et on conseille plutôt 10^{50}. La taille de groupe est importante car elle est directement liée à la taille des clefs publiques et secrètes qui devront être stockées et à la quantité d'information échangée lors des protocoles. Il est ainsi souhaitable d'utiliser des groupes pour lesquels il n'existe pas d'autres attaques que les algorithmes génériques évoqués ci-dessus, ce qui fournit des tailles de clef optimales pour ce type de protocoles.

Pour les groupes multiplicatifs de corps finis, le log discret peut être calculé via une méthode de calcul d'index beaucoup plus rapidement que les méthodes génériques, et il faut compter avec un groupe de taille 10^{300} pour se mettre à l'abri. Les courbes elliptiques résistent mieux en général : la meilleure attaque connue est un algorithme fonctionnant essentiellement dans n'importe quel groupe ; on peut donc avoir des tailles de clef raisonnables. Par contre, il est nécessaire de savoir calculer le cardinal du groupe, et c'est là que l'algorithme de Schoof et ses améliorations entrent en jeu. De manière identique, les Jacobiennes de courbes hyperelliptiques semblent difficiles à attaquer, du moins lorsque le genre reste « petit ». Il existe en effet une attaque sous-exponentielle, découverte par Adleman, DeMarrais, Huang [ADH94]. Celle-ci est valable lorsque le genre est suffisamment grand si bien que dans ce cas la taille de clef à choisir pour garantir une bonne sécurité n'est plus optimale. L'autre problème des courbes hyperelliptiques est le calcul de la cardinalité : les algorithmes en temps polynomial évoqués ci-dessus ne sont pas vraiment utilisables tels quels en pratique.

L'utilisation des courbes elliptiques en cryptographie est désormais banale : des livres font le point sur le sujet [Men93], [BSS99] et des compagnies proposent des logiciels à bases de courbes elliptiques. Les courbes hyperelliptiques n'ont quant à elles pas encore franchi le pas de la recherche théorique vers l'application pratique.

Organisation du document, résultats

Notre mémoire est divisé en trois parties. La première regroupe des prérequis théoriques et quelques calculs concernant les invariants de courbes de genre 2. Le chapitre 1 est consacré à ces prérequis : on rappelle rapidement la construction de la Jacobienne d'une courbe lisse quelconque avant de citer sans démonstration les théorèmes qui nous seront utiles par la suite, notamment les conjectures de Weil. Puis nous insistons un peu plus sur la notion de supersingularité et le lien avec la présence d'automorphismes. En effet, les automorphismes seront assez souvent évoqués dans ce mémoire, et la supersingularité est une caractéristique importante en cryptologie. Pour finir nous rappellerons quelques résultats sur les courbes hyperelliptiques et en particulier les courbes de genre 2.

Le chapitre 2 rapporte un travail effectué en collaboration avec Schost : nous avons donné des formules effectives permettant de relier les invariants d'Igusa d'une courbe de genre 2 ayant une involution non triviale au j-invariant des courbes elliptiques quotients de sa Jacobienne.

Le chapitre 3 relate une tentative de construction d'équations modulaires liant les invariants d'Igusa de courbes de genre 2 ayant des Jacobiennes (ℓ, ℓ)-isogènes. La conclusion est que les équations similaires aux équations modulaires du cas elliptiques sont énormes, même pour $\ell = 2$ pour lequel nous décrivons une méthode de calcul.

La deuxième partie concerne le calcul de la cardinalité de la Jacobienne d'une courbe hyperelliptique sur un corps fini. Ce type de calcul est intéressant pour construire des cryptosystèmes fondés sur le problème du logarithme discret. Les chapitres 4 et 5 sont de courts rappels sur les algorithmes connus pour calculer la loi de groupe dans la Jacobienne d'une courbe hyperelliptique, les algorithmes de calcul de cardinalité pour des groupes génériques et les méthodes d'approximation connues pour les courbes. Notre contribution à ce sujet est une amélioration de la méthode d'approximation utilisant les sous-corps réels.

Dans le chapitre 6, nous montrons comment le calcul de cardinalité peut être facilité pour certaines classes de courbes. Après avoir rappelé les cas bien connus des courbes de Koblitz et à multiplication complexe, nous expliquons que les courbes provenant de la multiplication réelle peuvent aussi être traitées plus facilement que les courbes générales. Nous parlons également de l'utilisation de l'opérateur de Cartier-Manin dans ce contexte, que nous avons remarquée lors d'un travail commun avec Harley.

Le chapitre 7 est dédié à une extension de l'algorithme de Schoof aux courbes de genre 2. En s'appuyant sur les travaux de Pila et Kampkötter et sur les polynômes de divisions de Cantor, nous avons complété tous les détails de manière à pouvoir réellement implanter cette technique. Nous avons alors joint nos efforts à ceux de Harley dont l'algorithme, reposant sur le Paradoxe des Anniversaires, pouvait tirer profit d'information calculée préalablement par notre méthode. Cette collaboration nous a aussi permis de mettre au point une méthode de division par deux dans la Jacobienne et finalement nous avons pu calculer la cardinalité d'une courbe de genre 2 définie sur un corps premier d'ordre environ 10^{19}, soit un groupe de taille environ 10^{38}.

Le chapitre 8 décrit une étude que nous avons menée en commun avec Schost. Nous avons voulu mimer les améliorations d'Elkies–Atkin à l'algorithme original de Schoof, qui consistent à utiliser des équations modulaires et leurs liens avec les isogénies afin d'obtenir de l'information sur la cardinalité. Nous avons construit des équations modulaires mieux adaptées que celles du chapitre 3 et avons été capables de calculer effectivement cette équation pour $\ell = 3$; nous avons ensuite donné les motifs de factorisation sur $\mathbb{F}_p$ en lien avec le Frobenius et avons vérifié expérimentalement ces propriétés. Dans l'état actuel, cela ne mène pas à un algorithme (même théorique) plus rapide que celui du chapitre 7, mais ouvre des horizons pour de plus amples recherches.

La troisième partie est la plus liée à la cryptologie, puisque nous y étudions le problème du logarithme discret, point clef pour évaluer la sécurité de systèmes à base de courbes. Le chapitre 9 décrit l'état de l'art du calcul du logarithme discret dans les Jacobiennes de courbes. Nous décrivons tout d'abord les méthodes fonctionnant dans un groupe générique. Nous insistons sur le fait que la présence d'un automorphisme sur la courbe affaiblit le système (travail commun avec Duursma et Morain). Nous évoquons ensuite rapidement les classes de courbes faibles, puis présentons un petit historique des attaques sous-exponentielles en genre grand. Les chapitres suivants sont consacrés à une variante de ces attaques que nous avons proposée et implantée.

Le chapitre 10 rapporte un travail en collaboration avec Enge où nous avons inscrit notre algorithme dans un cadre très général et nous nous sommes attachés à prouver de manière rigoureuse les complexités obtenues. Nous avons ainsi amélioré les meilleures bornes connues pour le calcul du logarithme discret dans les groupes de classes de corps quadratiques et dans les Jacobiennes de courbes hyperelliptiques de genre grand.

Le chapitre 11 décrit notre algorithme initial, qui apparaît maintenant comme un cas particulier de celui du chapitre 10. L'objectif est plus pratique que théorique : en cryptographie, le genre est fixé et c'est la taille du corps que l'on fait tendre vers l'infini pour obtenir une sécurité suffisante ; c'est donc avec ces hypothèses que nous avons travaillé sur l'algorithme. D'autre part nous avons abandonné l'idée de prouver la complexité pour obtenir une efficacité heuristique optimale. Cette approche, bien adaptée au cas où le genre n'est pas trop grand, nous a permis de casser un cryptosystème de genre 6 proposé par Koblitz.

Dans le chapitre 12 nous avons tenté d'appliquer le calcul d'index à des courbes de genre le plus petit possible : les courbes de genre 2 (les courbes elliptiques restent intouchables pour ce type d'algorithme). De manière assez surprenante, notre variante a pu être adaptée de manière à obtenir une complexité théorique du même ordre de grandeur que les meilleurs algorithmes connus. Toutefois, cela ne présente pas d'intérêt pratique car la complexité en espace n'est pas compétitive.

Le chapitre 13 se réfère à un travail mené en commun avec Heß et Smart. La technique de descente de Weil appliquée au calcul de logarithme discret sur les courbes elliptiques permet de ramener ce problème à un logarithme discret sur une Jacobienne de courbe de genre plus grand, mais sur un corps fini plus petit. Nous avons alors utilisé notre algorithme de logarithme discret et avons ainsi montré qu'une nouvelle classe de courbes elliptiques est sujette à des attaques plus efficaces que les meilleures connues dans le cas général.

Première partie

Courbes hyperelliptiques et invariants d'Igusa

Chapitre 1

Jacobiennes de courbes hyperelliptiques

Ce premier chapitre est consacré au rappel des définitions et des théorèmes nécessaires par la suite. Peu de démonstrations sont données, on se contentera en général de renvoyer à des références classiques. Ce chapitre théorique survole rapidement la construction de la Jacobienne d'une courbe, quelques résultats sur les variétés abéliennes, les conjectures de Weil. On insiste ensuite sur les courbes hyperelliptiques, sur ce qui se passe sur le corps des complexes et pour finir sur le cas très spécifique du genre 2.

1.1 Définition de la Jacobienne

1.1.1 Diviseurs sur une courbe

Soit $\mathcal{C}$ une courbe projective lisse définie sur un corps K algébriquement clos.

Définition 1.1 *Un* diviseur *de $\mathcal{C}$ est une somme formelle finie de points appartenant à $\mathcal{C}$. Ainsi, un diviseur s'écrit*

$$\sum_{P_i \in \mathcal{C}} n_i P_i,$$

où les n_i sont des entiers relatifs presque tous nuls.

L'ensemble des diviseurs est un groupe commutatif, où la loi de groupe est l'addition formelle de points. Ce groupe est noté $\mathrm{Div}(\mathcal{C})$.

Définition 1.2 *Le* degré *d'un diviseur est la somme de ses coefficients :*

$$\deg\left(\sum_{P_i \in \mathcal{C}} n_i P_i\right) = \sum_{P_i \in \mathcal{C}} n_i.$$

Le support *d'un diviseur est l'ensemble fini des points P_i pour lesquels le coefficient n_i est non nul.*

La définition de support a bien un sens car la somme est en fait une somme finie.

Le degré est un homomorphisme de $\mathrm{Div}(\mathcal{C})$ vers $\mathbb{Z}$. Le noyau de cet homomorphisme est l'ensemble des diviseurs de degré 0, noté $\mathrm{Div}^0(\mathcal{C})$. C'est un sous-groupe de $\mathrm{Div}(\mathcal{C})$.

Définition 1.3 *Un diviseur* effectif *est un diviseur D dont tous les coefficients sont positifs. On le note $D \geqslant 0$. Plus généralement, on définit la relation d'ordre partiel $\geqslant$ sur les diviseurs par $D \geqslant D'$ si et seulement si $D - D' \geqslant 0$.*

1.1.2 Diviseurs principaux

Le corps de fonctions de la courbe $\mathcal{C}$ est l'ensemble des fonctions rationnelles de $\mathcal{C}$ vers K. On le note $K(\mathcal{C})$.

Théorème 1.1 *Deux courbes sont isomorphes si et seulement si elles ont deux corps de fonctions isomorphes.*

On trouve une démonstration dans [Ful69, p. 180].

Le théorème précédent indique qu'il est complètement équivalent de travailler avec une vision géométrique des choses : courbes, points, etc... ou de travailler algébriquement, directement sur les corps de fonctions. Nous préférerons la première approche, plus proche de l'intuition (on peut faire des dessins !) ; l'ouvrage de Stichtenoth [Sti93] est une excellente référence pour le lecteur qui préfère une approche non géométrique.

Quelle que soit l'approche choisie, les diviseurs principaux sont intrinsèquement liés au corps de fonctions de $\mathcal{C}$. Pour les définir, il est nécessaire de faire une étude locale de la courbe. Le but est de donner une définition algébrique du fait qu'une fonction s'annule ou a un pôle en un point, éventuellement avec multiplicité.

Théorème 1.2 *Soit P un point de la courbe $\mathcal{C}$ (projective lisse). L'ensemble des fonctions de $K(\mathcal{C})$ qui sont définies au point P est un anneau de valuation discrète noté $\mathcal{O}_P$. Son unique idéal maximal est l'ensemble des fonctions qui s'annulent en P.*

Là encore, une démonstration se trouve dans [Ful69, p. 70].

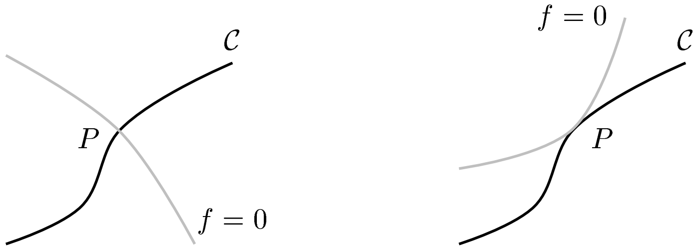
Ainsi, en chaque point, on peut associer à une fonction bien définie en ce point : sa *valuation*, qui est un entier positif ou nul. Cette valuation peut être étendue aux fonctions qui ont un pôle au point considéré.

Définition 1.4 *Soit f une fonction non nulle de $K(\mathcal{C})$ et soit P un point de $\mathcal{C}$. On définit la* valuation *en P de la fonction f, notée* $\mathrm{ord}_P(f)$ *de la façon suivante :*

- *Si f est définie en P et $f(P) \neq 0$, alors* $\mathrm{ord}_P(f) = 0$,
- *Si f est définie en P et $f(P) = 0$, alors* $\mathrm{ord}_P(f) = \nu_P(f)$,
- *Si f a un pôle en P, alors* $\mathrm{ord}_P(f) = -\nu_P(1/f)$,

où ν_P est la valuation de l'anneau local $\mathcal{O}_P$.

Intuitivement, la valuation d'une fonction en un point mesure la multiplicité du zéro de la fonction. Géométriquement cela correspond à la tangence entre la courbe $\mathcal{C}$ et la courbe $f = 0$.



$\nu_P(f) = 1$ $\qquad$ $\nu_P(f) \geqslant 1$

Il reste à former un diviseur ; le théorème suivant prouve que c'est possible.

Théorème 1.3 *Soit f une fonction non nulle de $K(\mathcal{C})$. Alors les points P pour lesquels $\mathrm{ord}_P(f)$ est non nul sont en nombre fini. De plus, le diviseur*

$$\mathrm{div}(f) = \sum_{P_i \in \mathcal{C}} \mathrm{ord}_{P_i}(f)$$

est de degré 0.

Ce dernier résultat peut-être lu « une fraction rationnelle a autant de zéros que de pôles ».

Définition 1.5 *Un diviseur* principal *de $\mathcal{C}$ est un diviseur D tel qu'il existe une fonction f pour laquelle*

$$D = \mathrm{div}(f).$$

Proposition 1.1 *L'application* div *est un homomorphisme du groupe multiplicatif des fonctions non nulles de $K(\mathcal{C})$ vers les diviseurs de degré 0 sur $\mathcal{C}$.*

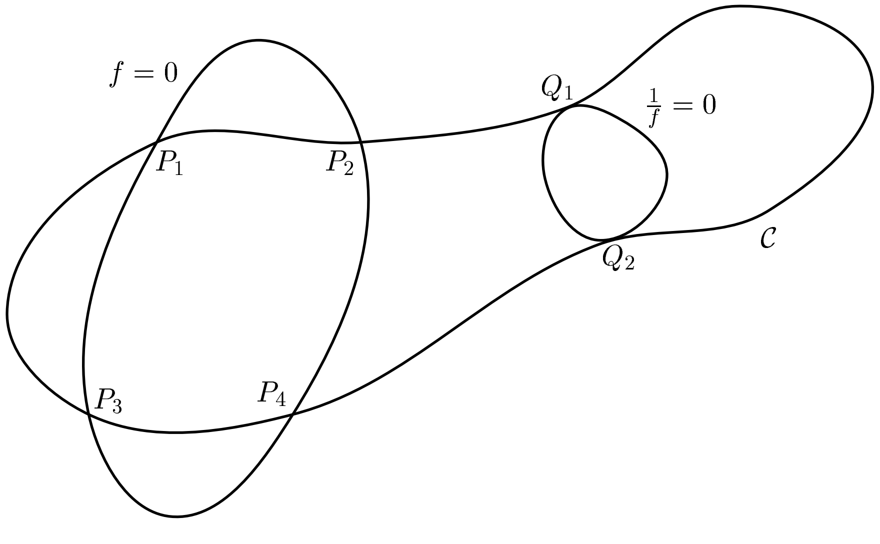
Démonstration. Cela découle directement des propriétés de la valuation en un point. □

L'ensemble des diviseurs principaux est donc un sous-groupe des diviseurs de degré 0. On le note $\mathrm{Pr}(\mathcal{C})$. Soit f une fonction. On découpe souvent $\mathrm{div}(f)$ en la différence de deux diviseurs effectifs :

$$\mathrm{div}(f) = \mathrm{div}_0(f) - \mathrm{div}_\infty(f),$$

où $\mathrm{div}_0(f)$ correspond à l'intersection de $\mathcal{C}$ avec la courbe $f = 0$, et $\mathrm{div}_\infty(f)$ à l'intersection avec $\frac{1}{f} = 0$. Sur le dessin ci-dessous on a ainsi

$$\mathrm{div}(f) = P_1 + P_2 + P_3 + P_4 - (2Q_1 + 2Q_2).$$



1.1.3 Jacobienne et corps de définition

En général les diviseurs de degré 0 ne sont pas tous principaux. Ces derniers forment un sous-groupe de $\mathrm{Div}^0(\mathcal{C})$.

Définition 1.6 *La* Jacobienne *de $\mathcal{C}$ est le groupe des diviseurs de degré zéro quotienté par les diviseurs principaux :*

$$\mathrm{Jac}(\mathcal{C}) = \mathrm{Div}^0(\mathcal{C})/\mathrm{Pr}(\mathcal{C}).$$

Deux diviseurs D et D' qui sont dans la même classe sont dits linéairement équivalents *; on le note*

$$D \sim D'.$$

Avant de donner quelques exemples concrets, nous allons préciser ce qu'il faut adapter dans les définitions précédentes si l'on veut travailler sur un corps non algébriquement clos.

Soit donc K un corps quelconque contenant les coefficients de l'équation de la courbe $\mathcal{C}$ et $\overline{K}$ une clôture algébrique de K. De manière générale, un objet sera dit défini sur K s'il est invariant sous l'action du groupe de Galois $\mathrm{Gal}(\overline{K}/K)$.

Un point de la courbe $\mathcal{C}$ est défini sur K si ses coordonnées sont dans K. Un diviseur est défini sur K s'il est invariant sous l'action du groupe de Galois. Cela ne signifie pas que tous les points qui le composent sont définis sur K ; en effet, l'action de Galois peut permuter les points. Un diviseur sur K est donc une somme de cycle de conjugués de points : si un point défini sur une extension de K est dans le support du diviseur, alors tous ses conjugués y sont aussi avec le même coefficient.

Une fonction est définie sur K si ses coefficients sont dans K ; et il s'ensuit immédiatement que le diviseur d'une fonction sur K est défini sur K.

On note avec l'indice K tous les ensembles d'objets définis sur K : $\mathrm{Div}_K(\mathcal{C})$, $\mathrm{Div}^0_K(\mathcal{C})$, $\mathrm{Pr}_K(\mathcal{C})$.

Définir sans ambiguïté la Jacobienne sur K nécessite un résultat qui prouve que les éléments de la Jacobienne qui sont invariants sous l'action de Galois forment exactement le groupe quotient $\mathrm{Div}^0_K(\mathcal{C})/\mathrm{Pr}_K(\mathcal{C})$.

Théorème 1.4 *Soit $\mathcal{C}$ une courbe définie sur K possédant un point défini sur K et D un diviseur de degré 0 sur K. S'il existe un diviseur principal $\mathrm{div}(f)$ défini sur $\overline{K}$ tel que $D' = D + \mathrm{div}(f)$ soit défini sur K alors il existe une fonction F définie sur K telle que $\mathrm{div}(f) = \mathrm{div}(F)$.*

La preuve de ce résultat repose sur le théorème de Riemann-Roch qui sera énoncé à la section suivante.

La conséquence est que

$$\mathrm{Jac}_K(\mathcal{C}) = \mathrm{Div}^0_K(\mathcal{C})/\mathrm{Pr}_K(\mathcal{C}).$$

L'étude sur un corps non algébriquement clos nécessite l'introduction d'une notion supplémentaire pour les diviseurs.

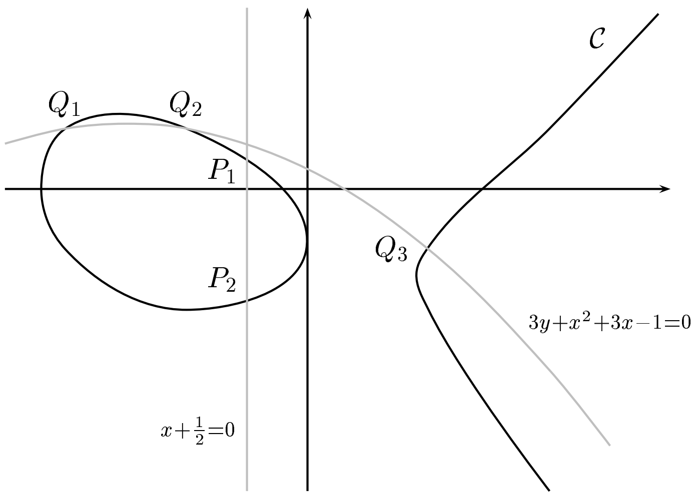
Définition 1.7 *Un diviseur D défini sur K est dit* premier *si*

1. *D est effectif,*
2. *Si D' est effectif, défini sur K et $D' \leqslant D$, alors D' est nul ou égal à D.*

Les diviseurs premiers sont en fait les sommes de tous les conjugués d'un point défini sur une extension de K. Dans le cas d'un corps algébriquement clos, les diviseurs premiers sont exactement les points de la courbe.

Exemple

Considérons la courbe sur $\mathbb{Q}$ d'équation $y^2+xy+2y = x^3+x^2-3x-1$; il s'agit d'une courbe elliptique mise sous forme de Weierstraß. Son allure lorsqu'on la trace sur $\mathbb{R}$ est la suivante :



Pour illustrer les définitions précédentes, il est nécessaire d'avoir une courbe projective. On rajoute donc le point à l'infini, noté ∞ qui complète notre courbe affine en une courbe projective $\mathcal{C}$. Il est facile de vérifier que $\mathcal{C}$ est lisse, même en ∞. La plupart du temps on opérera ainsi : on travaille avec des modèles affines de courbes planes, et l'on manipule les points à l'infini formellement.

Soient P_1 et P_2 les deux points de $\mathcal{C}$ de coordonnées

$$P_1 = \left(-\frac{1}{2}, \frac{-3+\sqrt{19}}{4}\right) \quad \text{et} \quad P_2 = \left(-\frac{1}{2}, \frac{-3-\sqrt{19}}{4}\right).$$

Ces deux points sont conjugués sur $\mathbb{Q}$ et le diviseur $D = P_1 + P_2$ est un diviseur premier de degré 2 défini sur $\mathbb{Q}$.

Soit la fonction f sur $\mathcal{C}$ définie par

$$f = \frac{3y + x^2 + 3x - 1}{x + \frac{1}{2}}.$$

La courbe correspondant à l'annulation du numérateur coupe $\mathcal{C}$ en les trois points $Q_1 = (-2, 1)$, $Q_2 = (-1, 1)$, $Q_3 = (1, -1)$, et celle correspondant au dénominateur coupe $\mathcal{C}$ en P_1 et P_2. Si l'on tient compte des intersections à l'infini, on obtient ainsi

$$\operatorname{div}(f) = Q_1 + Q_2 + Q_3 - P_1 - P_2 - \infty.$$

Ainsi par exemple les deux diviseurs de degré 0 suivants sont linéairement équivalents :

$$P_1 + P_2 - Q_1 - Q_2 \;\sim\; Q_3 - \infty.$$

1.1.4 Théorème de Riemann-Roch

Le théorème de Riemann-Roch est un outil très important pour l'étude des courbes algébriques. Il permet de définir le *genre* de la courbe qui est un invariant fondamental et il fournit l'existence de représentants agréables dans chaque classe de la Jacobienne.

Définition 1.8 *Soit $\mathcal{C}$ une courbe définie sur K et D un diviseur sur K. L'espace $L(D)$ est l'ensemble des fonctions de $K(\mathcal{C})$ dont le diviseur est plus grand que $-D$:*

$$L(D) = \{f \in K(\mathcal{C}),\ f \neq 0,\ \operatorname{div}(f) \geqslant -D\}.$$

Lemme 1.1 *Soit D un diviseur. Si l'on adjoint la fonction nulle, l'espace $L(D)$ est un espace vectoriel de dimension finie. Sa dimension est notée $l(D)$.*

Démonstration. La seule chose à montrer est que la dimension est finie. Nous renvoyons à [Ful69, p. 192], pour une démonstration de ce fait. Le principe sous-jacent est que lorsque l'on rajoute un point à un diviseur, le degré augmente de un, et la dimension de l'espace $L(D)$ augmente d'au plus un. D'où une récurrence qu'il n'est pas difficile d'initialiser. □

Théorème 1.5 *Soit $\mathcal{C}$ une courbe sur un corps K. Il existe un entier g et un diviseur W tels que pour tout diviseur D*

$$l(D) = \deg(D) + 1 - g + l(W - D).$$

L'entier g est appelé le genre *de la courbe et W est un* diviseur canonique*.*

Nous renvoyons à [Ful69] ou [Sti93] pour une démonstration.

Pour une courbe donnée, un diviseur canonique peut être calculé, et donc le théorème de Riemann-Roch donne une valeur exacte pour $l(D)$. Toutefois dans de nombreux cas il suffit de minorer le terme $l(W-D)$ par zéro pour obtenir ce que l'on veut. Cette forme un peu plus faible est ce qu'on appelle le théorème de Riemann.

Le genre g de la courbe est une notion qui admet une interprétation intuitive simple. Si le corps de base est le corps des complexes, alors une courbe projective lisse est en fait une surface de Riemann, et le genre est alors le « nombre de trous » dans cette surface. Par exemple, la sphère de Riemann $\mathbb{P}^1(\mathbb{C})$ est de genre 0 et un tore à un trou est une courbe elliptique (de genre 1).

Revenons à la Jacobienne de $\mathcal{C}$: c'est un groupe de classes, et le problème se pose donc de trouver un représentant canonique pour chaque classe. Le théorème suivant donne déjà l'existence d'une forme agréable d'un représentant :

Théorème 1.6 *Soit P_∞ un point de la courbe $\mathcal{C}$ fixé à l'avance. Pour tout diviseur D de degré zéro, il existe un diviseur effectif E de degré $r \leqslant g$, ne contenant pas P_∞ tel que*

$$D \sim E - rP_\infty.$$

Démonstration. Considérons le diviseur $D' = D + gP_\infty$ de degré g. Le théorème de Riemann-Roch donne $l(D') = 1 + l(W - D') \geqslant 1$, et assure donc l'existence d'une fonction f non nulle telle que $\operatorname{div}(f) + D' \geqslant 0$; notons E' le diviseur effectif $\operatorname{div}(f) + D'$. On a $D + \operatorname{div}(f) = E' - gP_\infty$, donc

$$D \sim E' - gP_\infty.$$

Si l'on prend en compte le fait qu'il faut éliminer les éventuels P_∞ intervenant dans E', on a le résultat. □

Exemple

Une courbe elliptique est une courbe de genre 1. On peut montrer qu'une telle courbe est isomorphe à une courbe admettant un modèle affine sous forme de Weierstraß, c'est-à-dire une équation de la forme $y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$, tout comme dans l'exemple page 13. Appliquons le théorème précédent en prenant pour P_∞ l'unique point à l'infini de la courbe sous cette forme. Chaque classe de la Jacobienne contient un diviseur de la forme $P - P_\infty$ ou bien 0. Le deuxième cas signifie qu'il s'agit de la classe triviale, et dans le cas général, toute classe peut-être représentée par un point de la courbe. Réciproquement, soient P et Q deux points de la courbe, alors $P - P_\infty$ et $Q - P_\infty$ définissent la même classe si et seulement si P et Q sont égaux.

Ainsi la Jacobienne d'une courbe elliptique est la courbe elle-même. La célèbre loi de groupe «corde et tangente» sur la courbe [Cas91] n'est autre qu'un calcul dans la Jacobienne. Quand on trace une droite, on considère le diviseur principal de la fonction donnée par l'équation de cette droite.

Nous donnons un résultat plus précis que le théorème 1.6, dû à Galbraith, Paulus et Smart [GPS00] (voir aussi [Ari99]) et qui fournit un représentant *unique* dans chaque classe.

Théorème 1.7 *Soit P_∞ un point de la courbe $\mathcal{C}$ fixé à l'avance. Pour tout diviseur D de degré zéro, il existe un unique diviseur effectif E de degré minimal m, ne contenant pas P_∞, tel que*

$$D \sim E - mP_\infty.$$

Un tel diviseur minimal est appelé diviseur réduit*, l'entier m est appelé le* poids *du diviseur.*

Démonstration. L'existence d'un tel diviseur est assuré par le théorème précédent. Si $m = 0$ alors D est principal et il n'y a rien à montrer. Nous allons prouver l'unicité dans le cas $m \geqslant 1$.

Soit $D_0 = E - mP_\infty$ une représentation de D avec m minimal. Montrons d'abord que $l(E) = 1$. Supposons que $l(D_0 + (m-1)P_\infty)$ soit non nul. Alors il existe une fonction f telle que $\mathrm{div}(f) + D_0 + (m-1)P_\infty = E' \geqslant 0$, et donc $E' - (m-1)P_\infty$ est une représentation de poids $m-1$ de D, ce qui contredit la minimalité de m. Ainsi $l(D_0 + (m-1)P_\infty) = 0$. Le fait de rajouter un point à un diviseur fait augmenter la dimension de l'espace $L()$ associé d'au plus une unité, donc $l(D_0 + mP_\infty) \leqslant 1$. Or $D_0 + mP_\infty = E$ est un diviseur effectif et les constantes forment un sous-espace vectoriel de $l(D_0 + mP_\infty)$. Il s'ensuit que

$$l(E) = 1.$$

Supposons maintenant qu'il existe E' effectif de degré m tel que $E - mP_\infty \sim E' - mP_\infty$. Alors il existe une fonction f telle que $\mathrm{div}(f) + E = E' \geqslant 0$, donc f appartient à $L(E)$, et donc f est une constante. D'où $\mathrm{div}(f) = 0$, et $E = E'$. □

1.1.5 Diviseur Θ

D'après la section précédente, dès que l'on s'est choisi un point P_∞ sur la courbe, on peut voir la Jacobienne comme un ensemble de diviseurs réduits, qui sont eux-mêmes des sommes formelles d'au plus g points de la courbe. Si la somme se réduit à un seul point, on obtient la proposition suivante :

Proposition 1.2 *Une courbe $\mathcal{C}$ munie d'un point P_∞ s'injecte canoniquement dans sa Jacobienne.*

Par ailleurs la Jacobienne peut-être munie d'une structure de variété algébrique de dimension g. Dans cette variété, les diviseurs réduits de poids g forment un ouvert dense et l'ensemble des diviseurs réduits de poids inférieur à g est une réunion de sous-variétés de dimension inférieure à g.

Définition 1.9 *L'ensemble des diviseurs réduits de poids strictement inférieur à g est appelé* diviseur Θ.

Lorsque le genre est 1, le diviseur Θ se réduit au point P_∞. Lorsque le genre est 2, il s'agit de la courbe, vue comme une sous-variété de sa Jacobienne.

1.2 Variétés abéliennes

La Jacobienne d'une courbe de genre g est une variété algébrique de dimension g munie d'une loi de groupe. De plus cette loi s'exprime localement par des formules algébriques. Un tel objet est appelé une variété abélienne et possède de riches propriétés.

1.2.1 Définition, isogénies

Définition 1.10 *Une* variété abélienne *est une variété projective lisse munie d'une loi de groupe commutatif qui s'exprime localement par des fractions rationnelles.*

Remarque. Il n'est pas nécessaire de mettre la commutativité dans la définition car celle-ci découle des autres propriétés. (cf [Lan59, p. 20])

Proposition 1.3 *Soit $\mathcal{I}$ un homomorphisme d'une variété abélienne A vers une variété abélienne B. Deux quelconques des conditions suivantes impliquent la troisième :*

(i) A et B ont même dimension,

(ii) $\mathcal{I}$ est surjectif sur une clôture algébrique,

(iii) le noyau de $\mathcal{I}$ sur une clôture algébrique est fini.

Si ces conditions sont vérifiées, l'homomorphisme $\mathcal{I}$ est appelé une isogénie.

Le *degré* d'une isogénie est son degré en tant que morphisme de variété algébrique. Soit $\mathcal{I}$ une isogénie de A vers B et soit x un point générique sur A, tous définis sur un corps K. Alors le degré de $\mathcal{I}$ est le degré de l'extension de corps $[K(x) : K(\mathcal{I}(x))]$. Cette extension se décompose en une extension purement inséparable et une extension séparable. On appelle *degré inséparable* et *degré séparable* de $\mathcal{I}$ les degrés respectifs de ces extensions, dont le produit est le degré de l'isogénie.

Notons que dans le cas séparable, le degré correspond au cardinal de la fibre générique sur une clôture algébrique de K.

Le premier exemple d'isogénie est le suivant : soit A une variété abélienne de dimension g, et soit n un entier premier à la caractéristique du corps. Alors la multiplication par n, notée $[n]_A$ et définie par

$$[n]_A(x) = x + x + \cdots + x \quad (n \text{ fois}),$$

est une isogénie de A vers A dont le noyau est de cardinal n^{2g} (cf [Lan59]).

Les isogénies sont des objets très importants. Le point crucial est que s'il existe une isogénie entre A et B, alors il en existe une autre entre B et A.

Théorème 1.8 *Soit $\mathcal{I}$ une isogénie de degré d entre deux variétés abéliennes A et B. Alors il existe une isogénie, notée $\hat{\mathcal{I}}$, de B vers A, telle que*

$$\mathcal{I}\hat{\mathcal{I}} = [d]_A \quad \text{et} \quad \hat{\mathcal{I}}\mathcal{I} = [d]_B,$$

où $[d]_A$ et $[d]_B$ désignent respectivement les multiplications par d sur A et B.

Ainsi la relation «il existe une isogénie entre deux variétés abéliennes» est symétrique. Elle est par ailleurs transitive car la composée de deux isogénies est une isogénie. Finalement c'est une relation d'équivalence, et l'on dira que deux variétés sont isogènes s'il existe une isogénie entre les deux.

Notation. Si A et B sont deux variétés abéliennes isogènes, on note

$$A \sim B.$$

Si elles sont isomorphes, ce qui est plus fort, on note

$$A \cong B.$$

L'ensemble des isogénies d'une variété abélienne A vers une variété abélienne B est noté $\mathrm{Hom}(A, B)$. Si $A = B$, une isogénie est appelée un *endomorphisme* de A. L'ensemble des endomorphismes de A est un anneau noté $\mathrm{End}(A)$.

Ces objets sont des modules sans torsion sur $\mathbb{Z}$, et l'on est souvent amené à considérer leur produit tensoriel par $\mathbb{Q}$ (il s'agit d'une simple extension de scalaires). On notera $\mathrm{Hom}_0(A, B) = \mathrm{Hom}(A, B) \otimes \mathbb{Q}$ et $\mathrm{End}_0(A) = \mathrm{End}(A) \otimes \mathbb{Q}$. Si on remplace A et B par des variétés abéliennes qui leurs sont isogènes, alors $\mathrm{Hom}_0(A, B)$ et $\mathrm{End}_0(A)$ sont inchangés mais $\mathrm{Hom}(A, B)$ et $\mathrm{End}(A)$ peuvent être modifiés.

1.2.2 Théorème de décomposition

La relation d'isogénie étant plus faible que celle d'isomorphisme, on peut introduire une notion d'«irréductibilité» adaptée à cette notion. Toute variété abélienne peut se décomposer de manière unique à isogénie près en produit de variétés abéliennes dites simples.

Théorème 1.9 *Soit A une variété abélienne et B une variété abélienne strictement incluse dans A. Alors il existe une variété abélienne C telle que A soit isogène au produit $B \times C$.*

Nous renvoyons à [Lan59, p. 28] pour une preuve. Notons que l'on a $\dim(A) = \dim(B) + \dim(C)$. Ce théorème dit que dès qu'il existe une sous-variété abélienne propre, alors on peut en trouver une complémentaire.

Définition 1.11 *Une variété abélienne est dite* simple *si elle n'admet pas de sous-variété abélienne autre que $\{0\}$ et elle-même.*

Le théorème de décomposition est donc le suivant :

Théorème 1.10 *Toute variété abélienne est isogène à un produit de variétés abéliennes simples, unique à isogénie près.*

Ainsi toute variété abélienne A est isogène à un produit

$$(A_1 \times \cdots \times A_1) \times \cdots \times (A_m \times \cdots \times A_m),$$

où les A_i sont des variétés abéliennes simples deux-à-deux non isogènes. On peut alors décrire $\mathrm{End}_0(A)$ en fonction des $\mathrm{End}_0(A_i)$. (cf [Lan59, p. 30])

Si $B_i = (A_i \times \cdots \times A_i)$ contient n_i facteurs, alors $\mathrm{End}_0(B_i)$ est l'anneau des matrices de taille n_i à coefficients dans $\mathrm{End}_0(A_i)$. L'algèbre $\mathrm{End}_0(A)$ est alors le produit direct des $\mathrm{End}_0(B_i)$.

Corps de définition

Dans tout ce qui précède, nous avons passé sous silence les problèmes de corps de définition. Si le corps K considéré n'est pas algébriquement clos, deux variétés abéliennes sont dites K-isogènes s'il existe une isogénie *définie sur* K entre les deux. Le théorème de décomposition reste vrai sur K, car si une variété abélienne A contient une sous-variété abélienne propre B définie sur K, alors on peut trouver une variété abélienne C définie sur K telle que A et $B \times C$ sont K-isogènes.

Il est toutefois possible qu'une variété abélienne soit simple sur K mais ne le soit plus sur une clôture algébrique. Cela motive la définition suivante :

Définition 1.12 *Une variété abélienne est* absolument simple *si elle est simple sur toute extension algébrique de son corps de définition.*

1.2.3 Sous-groupes de n-torsion

Nous avons déjà vu que le cardinal du noyau de la multiplication par n est n^{2g}. Par ailleurs, ce noyau est un sous-groupe abélien.

Définition 1.13 *On note $A[n]$ le noyau de la multiplication par n sur une clôture algébrique. Ses éléments sont appelés éléments de n-torsion.*

Le théorème suivant donne la structure de la n-torsion :

Théorème 1.11 *Soit A une variété abélienne de dimension g, et n un entier premier à la caractéristique du corps de base. Alors*

$$A[n] \cong (\mathbb{Z}/n\mathbb{Z})^{2g}.$$

Lorsque n est égal à la caractéristique du corps, le résultat n'est plus vrai.

Théorème 1.12 *Soit A une variété abélienne de dimension g sur un corps K de caractéristique p. Alors*

$$A[p]/\overline{K} \cong (\mathbb{Z}/p\mathbb{Z})^r,$$

où r est un entier tel que $0 \leqslant r \leqslant g$. L'entier r est appelé le p-rang *de A (et par abus de langage, on parle de p-rang d'une courbe pour le p-rang de sa Jacobienne).*

Le rang de la p-torsion est un invariant important. Le cas le plus courant est celui où r est maximal. Les autres cas sont des cas particuliers qu'il conviendra d'étudier.

Définition 1.14 *Une courbe $\mathcal{C}$ de p-rang maximal est appelée* ordinaire. *Une courbe $\mathcal{C}$ dont le p-rang est zéro est appelée* très spéciale.

1.3 Courbes sur les corps finis : conjectures de Weil

Dans cette section, on se fixe un corps fini à $q = p^d$ éléments, noté $\mathbb{F}_q$, et l'on considère une courbe $\mathcal{C}$ de genre g définie sur ce corps. Le nombre de points de la courbe est fini, et la Jacobienne est un groupe fini. L'étude de la fonction Zêta, et les conjectures de Weil donnent des bornes précises sur les cardinalités de ces ensembles. Les preuves de tous les résultats de cette section peuvent être trouvés dans [Sti93].

1.3.1 Fonction Zêta

La fonction Zêta associée à une courbe $\mathcal{C}$ est une série génératrice liée au nombre de points de $\mathcal{C}$ définis sur une extension de degré n.

Définition 1.15 *La* fonction Zêta *de $\mathcal{C}$ est définie par*

$$Z(t) = \exp\left(\sum_{n\geqslant 1} N_n \frac{t^n}{n}\right),$$

où N_n est le nombre de points de $\mathcal{C}$ définis sur $\mathbb{F}_{q^n}$.

En regroupant les points en diviseurs effectifs, on peut réorganiser la série de manière à obtenir une définition équivalente :

Lemme 1.2 *La fonction Zêta de $\mathcal{C}$ vérifie :*

$$Z(t) = \sum_{n\geqslant 0} C_n t^n,$$

où $C_n = \{D \in \mathrm{Div}(\mathcal{C});\; D \geqslant 0, \deg(D) = n\}$ est le nombre de diviseurs effectifs de degré n sur $\mathcal{C}$.

De manière analogue à la fonction ζ de Riemann, on peut transformer cette écriture en un produit Eulérien.

Lemme 1.3 *La fonction $Z(t)$ se réécrit*

$$Z(t) = \prod_{D \text{ premier}} (1 - t^{\deg(D)})^{-1}.$$

Le cas des courbes de genre 0 peut être traité sans trop de problèmes, et nous donnons ici les détails du calcul afin d'illustrer comment les objets introduits jusqu'ici se manipulent.

Exemple de $\mathbb{P}^1(\mathbb{F}_q)$

Pour tout $n \geqslant 1$, le nombre de points sur $\mathbb{P}^1(\mathbb{F}_{q^n})$ est q^n+1, c'est-à-dire le nombre d'éléments dans le corps plus le point à l'infini. La série que l'on obtient est donc

$$Z(t) = \exp\left(\sum_{n\geqslant 1}(q^n+1)\frac{t^n}{n}\right).$$

Après simplification, on obtient

$$Z(t) = \frac{1}{(1-t)(1-qt)}.$$

1.3.2 Théorème de Weil

Dans les années 30, Hasse démontra des bornes sur le nombre de points d'une courbe elliptique sur un corps fini. Généralisant cela, Weil énonça en 1949 de célèbres conjectures concernant la fonction Zêta d'une variété définie sur un corps fini, et les prouva dans le cas particulier des courbes et des variétés abéliennes. Par la suite de nombreux travaux pour étendre le résultat ont été faits par Dwork, Artin, Grothendieck, Deligne. Pour le cas des courbes, on peut en trouver une preuve (essentiellement celle de Bombieri) dans [Sti93].

Théorème 1.13 (Conjectures de Weil) *Soit $\mathcal{C}$ une courbe de genre g sur un corps fini $\mathbb{F}_q$. Sa fonction Zêta $Z(t)$ possède les propriétés suivantes :*

1. **Rationalité :** *$Z(t)$ est une fraction rationnelle.*
2. **Équation fonctionnelle :** $Z(t) = q^{g-1}t^{2g-2}Z(\frac{1}{qt})$.
3. **Hypothèse de Riemann :** *Les inverses des zéros de $Z(t)$ ont pour valeur absolue $\sqrt{q}$.*

Plus précisément, la fonction Zêta peut se mettre sous la forme

$$Z(T) = \frac{L(t)}{(1-t)(1-qt)},$$

où $L(t)$ est un polynôme qui a les propriétés suivantes :

Théorème 1.14 *Le polynôme $L(t) = (1-t)(1-qt)Z(t)$ vérifie :*

1. *C'est un polynôme de degré $2g$ à coefficients entiers.*
2. *Le cardinal de la Jacobienne est $\#\mathrm{Jac}(\mathcal{C}) = L(1)$.*
3. *On a l'équation fonctionnelle $L(t) = q^g t^{2g} L(\frac{1}{qt})$.*
4. *Si l'on écrit $L(t) = a_0 + a_1 t + \cdots + a_{2g}t^{2g}$, alors*
   (a) $a_0 = 1$ *et* $a_{2g} = q^g$,
   (b) $a_{2g-i} = q^{g-i}a_i$ *pour* $0 \leqslant i \leqslant g$.
5. *Si l'on écrit $L(t) = \prod(1-\alpha_i t)$, on peut réarranger les indices de telle sorte que $\alpha_i \alpha_{g+i} = q$, et de plus $|\alpha_i| = \sqrt{q}$.*

Les conséquences des conjectures de Weil sur les cardinalités sont immédiates :

Corollaire 1.1 *Soit $\mathcal{C}$ une courbe de genre g définie sur un corps fini $\mathbb{F}_q$. Alors le nombre de points sur la courbe est borné par*

$$|\#\mathcal{C} - (q+1)| \leqslant 2g\sqrt{q}.$$

Le cardinal de sa Jacobienne est quant à lui borné par

$$(\sqrt{q}-1)^{2g} \leqslant \#\mathrm{Jac}(\mathcal{C}) \leqslant (\sqrt{q}+1)^{2g}.$$

Ces bornes signifient que le nombre de points sur une courbe de genre g est environ q avec un terme d'erreur en $\sqrt{q}$ et le cardinal de sa Jacobienne est environ q^g avec un terme d'erreur en $q^{g-\frac{1}{2}}$.

Un résultat supplémentaire permet de relier la fonction Zêta d'une courbe sur $\mathbb{F}_q$ avec la fonction Zêta de la même courbe, mais considérée sur une extension algébrique finie $\mathbb{F}_{q^r}$.

Théorème 1.15 *Soit $\mathcal{C}$ une courbe de genre g définie sur $\mathbb{F}_q$, et soit r un entier non nul. Notons $L(t) = \prod(1-\alpha_i t)$ le polynôme L associé à la fonction Zêta de $\mathcal{C}$. Alors la fonction Zêta $Z_r(t)$ de la courbe $\mathcal{C}$ sur $\mathbb{F}_{q^r}$ est donnée par*

$$Z_r(t) = \frac{\prod(1-\alpha_i^r t)}{(1-t)(1-q^r t)}.$$

1.3.3 Action de l'endomorphisme de Frobenius

La théorie de Galois sur les corps finis est très simple : toutes les extensions sont cycliques, engendrées par l'automorphisme de Frobenius. Cette action sur les corps finis se transcrit sur les coordonnées des points, puis sur les diviseurs et enfin sur la Jacobienne.

Dans ce qui suit, on se fixe une courbe $\mathcal{C}$ définie sur $\mathbb{F}_q$ et une clôture algébrique $\overline{\mathbb{F}}_q$ de $\mathbb{F}_q$.

Définition 1.16 *L'automorphisme de Frobenius, noté π, est l'automorphisme du corps $\overline{\mathbb{F}}_q$ laissant fixe $\mathbb{F}_q$, défini par*

$$\pi(x) = x^q.$$

Lemme 1.4 *L'automorphisme de Frobenius sur $\overline{\mathbb{F}}_q$ s'étend en une action sur les points de la courbe, puis en un endomorphisme de la Jacobienne. On continue de l'appeler Frobenius et de le noter π.*

Un élément de la Jacobienne est défini sur $\mathbb{F}_q$ si et seulement s'il est invariant sous l'action de $\mathrm{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$, donc si et seulement s'il est invariant sous l'action du Frobenius π. En d'autres termes

$$\mathrm{Ker}(\pi - \mathrm{Id}) = \mathrm{Jac}(\mathcal{C})/\mathbb{F}_q,$$

et l'on en déduit que $\#\mathrm{Jac}(\mathcal{C})/\mathbb{F}_q = \chi(1)$ où $\chi(t)$ est le polynôme caractéristique du Frobenius dans l'anneau des endomorphismes qui est donné par le théorème suivant :

Théorème 1.16 *Le polynôme caractéristique de l'endomorphisme de Frobenius sur $\mathrm{Jac}(\mathcal{C})$, noté $\chi(t)$ est le polynôme réciproque du polynôme $L(t)$ défini à partir de la fonction Zêta de la courbe. C'est donc un polynôme de degré $2g$ à coefficients entiers, dont les racines ont valeur absolue $\sqrt{q}$ et tel que*

$$\#\mathrm{Jac}(\mathcal{C})/\mathbb{F}_q = \chi(1).$$

Cas du genre 0

Pour la droite projective $\mathbb{P}^1(\mathbb{F}_q)$, le polynôme $L(t)$ est constant, égal à 1, et donc

$$\chi(t) = 1.$$

On retrouve alors le fait que la Jacobienne est le groupe trivial n'ayant qu'une seule classe, celle-ci étant définie sur $\mathbb{F}_q$. Le Frobenius est donc égal à l'identité.

Cas du genre 1

C'est le premier cas non trivial. Le polynôme $L(t)$ s'écrit

$$L(t) = 1 + a_1 t + qt^2 = (1 - \alpha_1 t)(1 - \alpha_2 t),$$

et le polynôme caractéristique du Frobenius est de la forme

$$\chi(t) = t^2 - s_1 t + q = (t - \alpha_1)(t - \alpha_2),$$

où α_1 et α_2 sont conjugués complexes, de valeur absolue $\sqrt{q}$. On a donc l'inégalité suivante sur l'entier $s_1 = \alpha_1 + \alpha_2$ (la *trace* de la courbe) :

$$|s_1| \leqslant 2\sqrt{q}.$$

Il est expliqué plus haut que la Jacobienne d'une courbe elliptique est isomorphe à la courbe elle-même, dès que l'on a choisi un point comme élément neutre. Ainsi, si l'on a une courbe elliptique $\mathcal{E}$ définie sur $\mathbb{F}_q$, l'action du Frobenius sur la courbe est décrite par le polynôme $\chi(t)$: pour tout point P défini sur une extension algébrique, on a

$$\pi^2(P) - s_1\pi(P) + qP = 0,$$

où l'addition est celle entre points de la courbe héritant de la structure de Jacobienne, et la multiplication par un entier n'est autre que l'application de l'endomorphisme de multiplication dans la Jacobienne. La notation rigoureuse serait donc :

$$\pi^2(P) - [s_1]\pi(P) + [q]P = 0_{\mathrm{Jac}(\mathcal{E})}.$$

Cas du genre 2

Soit $\mathcal{C}$ une courbe de genre 2 sur $\mathbb{F}_q$. Le polynôme caractéristique de l'endomorphisme de Frobenius sur $\mathrm{Jac}(\mathcal{C})$ est de la forme

$$\chi(t) = t^4 - s_1 t^3 + s_2 t^2 - s_1 q t + q^2,$$

et l'hypothèse de Riemann donne les bornes suivantes pour les entiers s_1 et s_2 :

$$|s_1| \leqslant 4\sqrt{q} \quad \text{et} \quad |s_2| \leqslant 6q.$$

Ainsi le cardinal de la Jacobienne est borné par

$$q^2 - 4q^{\frac{3}{2}} + 6q - 4q^{\frac{1}{2}} + 1 \leqslant \#\mathrm{Jac}(\mathcal{C}) \leqslant q^2 + 4q^{\frac{3}{2}} + 6q + 4q^{\frac{1}{2}} + 1.$$

Là encore, le polynôme caractéristique du Frobenius traduit son comportement sur les éléments de la Jacobienne : pour tout diviseur réduit D, on a

$$\pi^4(D) - [s_1]\pi^3(D) + [s_2]\pi^2(D) - [s_1 q]\pi(D) + [q^2]D = 0_{\mathrm{Jac}(\mathcal{C})}.$$

1.4 Automorphismes, supersingularité et p-torsion

Nous revenons maintenant sur l'étude de $A[p]$ en caractéristique p, et en particulier sur les cas extrêmes où $A[p]$ est dégénéré.

1.4.1 Supersingularité et superspécialité

Dans le cas des courbes elliptiques, les p-rangs possibles sont zéro ou un. Une courbe elliptique de p-rang nul est appelée supersingulière. Cette notion de supersingularité s'étend aux courbes de genre supérieur, mais la définition est un peu différente. En particulier la condition sur le p-rang n'est plus suffisante pour assurer la supersingularité.

Définition 1.17 *Une courbe $\mathcal{C}$ de genre g est dite* supersingulière *si sa Jacobienne est isogène (sur une clôture algébrique) à la puissance g-ième d'une courbe elliptique supersingulière.*

Si on remplace la condition «isogène» par la condition plus forte «isomorphe» on a la définition suivante.

Définition 1.18 *Une courbe $\mathcal{C}$ de genre g est dite* superspéciale *si sa Jacobienne est isomorphe (sur une clôture algébrique) à la puissance g-ième d'une courbe elliptique supersingulière.*

Il est nécessaire d'introduire toutes ces définitions car il existe effectivement des exemples de courbes correspondant précisément à chaque cas. Toutefois en genre 2 les choses se simplifient quelque peu grâce au résultat suivant.

Théorème 1.17 *Toute courbe de genre 2 très spéciale est supersingulière.*

Une preuve de ceci ainsi que de nombreux résultats sur ces problèmes de p-torsion se trouve dans [Yui78]. Nous renvoyons aussi aux thèses de Brock [Bro93] et de Zhu [Zhu97].

Ces définitions et implications sont résumées dans le tableau suivant où E_s désigne une courbe elliptique supersingulière.

Très spéciale p-rang nul	$\Longleftarrow$ $\Longrightarrow$ si $g \leqslant 2$	**Supersingulière** isogène à E_s^g	$\Longleftarrow$ $\Longrightarrow$ si $g = 1$	**Superspéciale** isomorphe à E_s^g

1.4.2 Critère de supersingularité

Le p-rang d'une courbe de genre quelconque sur un corps fini peut être déterminé rapidement dès que l'on connaît le polynôme caractéristique du Frobenius. En effet dans [Sti79], Stichtenoth prouve le théorème suivant.

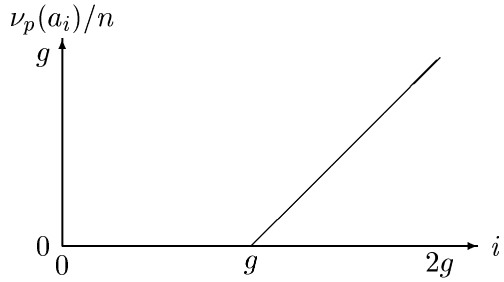
Théorème 1.18 *Soit $\mathcal{C}$ une courbe de genre g sur un corps fini de caractéristique p. Soit $\chi(t) = \sum_{i=0}^{2g} a_{2g-i} t^i$ le polynôme caractéristique du Frobenius sur la Jacobienne de $\mathcal{C}$. Alors le p-rang de $\mathcal{C}$ est donné par*

$$\max\{i \mid a_i \not\equiv 0 \bmod p\}.$$

Un manière plus visuelle de traduire cette proposition est d'introduire le polygône de Newton de $\chi(t)$.

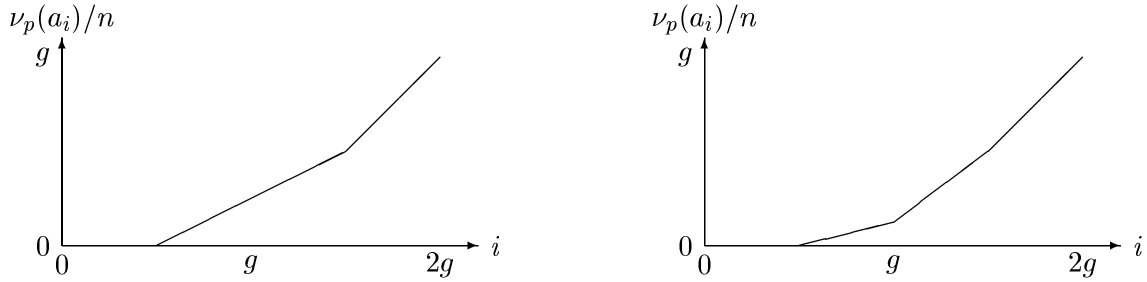
Définition 1.19 *Le* polygone de Newton *de $\chi(t)$ est l'enveloppe convexe inférieure des points $(i, \nu_p(a_i))$ dans un repère cartésien, où ν_p est la valuation p-adique.*

Le cas ordinaire correspond au p-rang maximal, et donc à $\nu_p(a_i) = 0$ pour i allant de 0 à g. Comme on a $a_{2g-i} = q^{g-i}a_i$, on obtient la figure suivante (on a noté $q = p^n$)



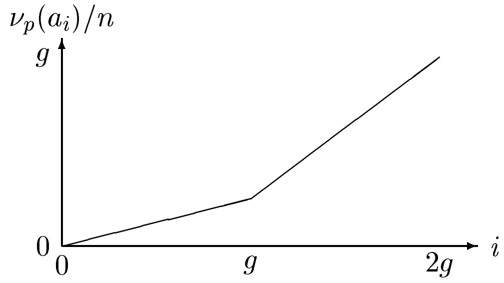
Cas ordinaire

Dans le cas d'une courbe de p-rang non maximal, il existe un $i_0 \leqslant g$ tel que pour tout $i \geqslant i_0$, on a $\nu_p(a_i) > 0$. Les allures possibles pour le polygone de Newton sont alors les suivantes :

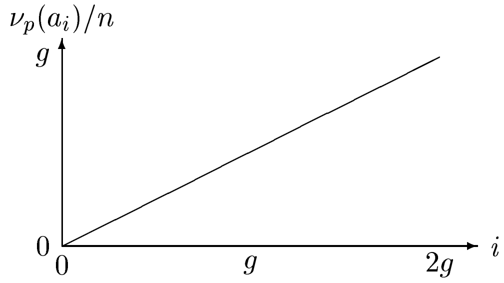


Cas du p-rang non maximal

Dans le cas extrême d'une courbe de p-rang nul, le tracé du polygone de Newton permet de distinguer le cas supersingulier. En effet, dans ce cas le polygone de Newton est composé d'un seul segment, alors que sinon il y en a deux.



Cas très spécial



Cas supersingulier

Tous ces résultats « visuels » sont énoncés et démontrés de manière précise dans l'article de Yui [Yui78]. Nous redonnons seulement le théorème concernant les courbes supersingulières.

Théorème 1.19 *Soit $\mathcal{C}$ une courbe de genre g sur le corps fini à $q = p^n$ éléments. Alors $\mathcal{C}$ est très spéciale si et seulement si toutes les pentes de son polygone de Newton sont strictement positives, et $\mathcal{C}$ est supersingulière si et seulement si toutes les pentes de son polygone de Newton sont égales. Dans ce dernier cas les pentes ont pour valeur $n/2$.*

Ainsi une fois calculé le polynôme caractéristique du Frobenius d'une courbe $\mathcal{C}$, il est aisé de calculer son p-rang et de vérifier si $\mathcal{C}$ est très spéciale ou supersingulière.

Notons de plus que dans le cas supersingulier, la Jacobienne est isogène à une puissance d'une courbe elliptique supersingulière sur une extension finie. Ceci a des conséquences sur le polynôme $\chi(t)$. En effet, sur une extension finie, le $\chi(t)$ de la courbe $\mathcal{C}$ deviendra lui-même une puissance d'un polynôme irréductible de degré 2 définissant la classe d'isogénie de la courbe elliptique en question par le théorème de Tate.

1.4.3 Groupe d'automorphismes d'une courbe

Le but de cette section est de rappeler quelques théorèmes sur le groupe d'automorphismes et en particulier des bornes sur son cardinal. Ces bornes sont différentes selon que la courbe est ordinaire ou non.

Définition d'un automorphisme

Définition 1.20 *Soit $\mathcal{C}$ une courbe de genre g sur un corps K. Un automorphisme K-rationnel de la courbe $\mathcal{C}$ est une fonction λ définie sur K de $\mathcal{C}$ vers $\mathcal{C}$ telle que :*

- *la fonction λ est un morphisme bijectif de variétés algébriques sur $\overline{K}$,*
- *la fonction réciproque est aussi un morphisme de variétés algébriques sur $\overline{K}$.*

Il est important de demander à ce que la notion d'automorphisme soit une notion *géométrique* (i.e. on regarde la clôture algébrique). En effet, si on prenait une définition non géométrique en imposant seulement que λ et λ^{-1} soient des fonctions rationnelles sur K qui soient des bijections de la courbe $\mathcal{C}$ sur K, on aboutirait au problème suivant : si K est un corps fini, alors $\mathcal{C}$ est un ensemble fini et toute permutation de cet ensemble pourrait être vue comme un automorphisme en construisant la formule polynomiale correspondante par interpolation. Le groupe d'automorphismes d'une courbe sur un corps fini ne serait alors qu'un groupe de permutations.

Exemple : Soit $\mathcal{C}$ la courbe $y^2 = x^7 + 1$ définie sur $\mathbb{F}_{29}$. La fonction $(x, y) \mapsto (16\,x, y)$ est un automorphisme de $\mathcal{C}$ d'ordre 7.

Exemple : Soit $\mathcal{C}$ la courbe $y^2 = x^5 + 2x^3 + x + 1$ définie sur $\mathbb{F}_{5^2}$. La fonction « Frobenius de $\mathbb{F}_5$ » définie par $(x, y) \mapsto (x^5, y^5)$ n'est pas un automorphisme de $\mathcal{C}$, car son inverse n'est pas un morphisme géométrique : on peut l'écrire à l'aide de fractions rationnelles seulement sur une extension finie mais pas sur $\overline{\mathbb{F}_{5^2}}$ [Har77, p. 21, Ex 3.2(b)].

En caractéristique nulle, la formule de Hurwitz pour le genre permet de déduire une borne sur le nombre d'automorphismes.

Théorème 1.20 (Hurwitz) *Le groupe d'automorphismes d'une courbe de genre $g \geqslant 2$ définie sur un corps de caractéristique nulle est fini et de cardinal borné par $84(g-1)$.*

Dans le cas général, la borne n'est pas aussi bonne. La démonstration du théorème suivant est donnée dans [Sti73].

Théorème 1.21 (Stichtenoth) *Le groupe d'automorphismes d'une courbe de genre $g \geqslant 2$ définie sur un corps de caractéristique p est fini et de cardinal borné strictement par $16g^4$, à une exception près : la courbe* Hermitienne *$y^q + y = x^{q+1}$ dont le genre est $q(q-1)/2$, et le nombre d'automorphismes est $q^3(q^3+1)(q^2-1)$.*

Dans le cas où la caractéristique du corps est suffisamment grande par rapport au genre, Roquette [Roq70] a montré que la borne d'Hurwitz redevenait vraie.

Théorème 1.22 (Roquette) *Le groupe d'automorphismes d'une courbe de genre $g \geqslant 2$ définie sur un corps de caractéristique $p > g+1$ est fini et de cardinal borné par $84(g-1)$, à une exception près : la courbe $y^p - y = x^2$ dont le genre est $(p-1)/2$, et le nombre d'automorphismes est $2p(p^2-1)$.*

Théorème de Torelli

Jusqu'ici il n'a été question que d'automorphismes sur une courbe $\mathcal{C}$. Le théorème de Torelli montre que le groupe d'automorphismes de la Jacobienne de $\mathcal{C}$ lui est en fait essentiellement isomorphe ; seule l'involution hyperelliptique (définie à la section suivante) pourra perturber légèrement.

Soit $\mathcal{C}$ une courbe de genre g, et soit λ un automorphisme de $\mathcal{C}$. Par linéarité, on peut étendre λ en un automorphisme sur le groupe des diviseurs de degré 0 sur $\mathcal{C}$. De plus l'ensemble des diviseurs principaux est globalement invariant sous l'action de λ. On peut finalement étendre λ à un automorphisme de la Jacobienne de $\mathcal{C}$.

Par exemple, si $\mathcal{C}$ est une courbe hyperelliptique, l'image de l'involution hyperelliptique est l'involution $D \mapsto -D$ dans la Jacobienne.

La réciproque de cette correspondance est assurée par le théorème suivant que l'on peut trouver dans [Mil86].

Théorème 1.23 *Soit $\mathcal{C}$ une courbe injectée dans sa Jacobienne grâce à un point P. On note f cette injection. Soit β un automorphisme de $\mathrm{Jac}(C)$. Alors il existe un automorphisme α de la courbe $\mathcal{C}$, et une constante c dans $\mathrm{Jac}(C)$ tels que*

$$f \circ \alpha = \pm \beta \circ f + c.$$

De plus, c et α sont déterminés de manière unique, ainsi que le signe dans le cas où la courbe n'est pas hyperelliptique.

Lien entre automorphismes et supersingularité

Avoir un grand groupe d'automorphismes a des conséquences sur les propriétés de la courbe. Nous allons évoquer deux phénomènes :

1. Si le groupe d'automorphismes d'une courbe $\mathcal{C}$ est de grand cardinal, alors $\mathcal{C}$ n'est pas ordinaire.

2. Si le groupe d'automorphismes d'une courbe $\mathcal{C}$ contient des éléments d'ordre petit, alors la Jacobienne de $\mathcal{C}$ a tendance à se décomposer.

La borne $16g^4$ de Stichtenoth n'est jamais atteinte pour une courbe ordinaire. En effet, Nakajima donne la borne suivante dans [Nak87] :

Théorème 1.24 (Nakajima) *Soit $\mathcal{C}$ une courbe ordinaire de genre g sur un corps de caractéristique p. Alors le cardinal de son groupe d'automorphismes est borné par $84g(g-1)$.*

La démonstration de ce résultat repose sur la relation entre l'existence de p-sous-groupes de $\mathrm{Aut}(\mathcal{C})$ et la dégénérescence des sous-groupes de p-torsion de la Jacobienne.

À chaque automorphisme σ d'une courbe $\mathcal{C}$ sur un corps K correspond une extension de corps de fonctions. Soit $K(\mathcal{C})$ le corps des fonctions de $\mathcal{C}$, et soit $\mathbb{F}$ le sous-corps de $K(\mathcal{C})$ fixé par σ. L'extension $[K(\mathcal{C}) : \mathbb{F}]$ est galoisienne de degré $\mathrm{ord}(\sigma)$, et la formule du genre de Riemann-Hurwitz relie le genre g de $\mathcal{C}$ au genre $g(\mathbb{F})$ du sous-corps de fonctions $\mathbb{F}$:

$$2g - 2 \geqslant \mathrm{ord}(\sigma)(2g(\mathbb{F}) - 2).$$

Dans cette formule, si $\mathrm{ord}(\sigma)$ est grand, alors $g(\mathbb{F})$ doit être nul, et l'on obtient un recouvrement de $\mathbb{P}^1$. Par contre, si $\mathrm{ord}(\sigma)$ est suffisamment petit par rapport à g, on peut espérer que $\mathbb{F}$ soit un corps de fonctions non trivial. Traduit en termes géométriques, cela signifie qu'il existe un morphisme de $\mathcal{C}$ vers une courbe $\mathcal{C}'$ de genre au moins 1, et donc que la Jacobienne de $\mathcal{C}$ est isogène à un produit de sous-variétés abéliennes dont l'une est la Jacobienne de $\mathcal{C}'$.

Pour plus de détails sur l'existence de telles décompositions provenant d'automorphismes, on consultera l'article de Kani et Rosen [KR89].

Notons pour finir que le groupe d'automorphismes d'une courbe a tendance à contenir beaucoup d'éléments de petit ordre. Le théorème suivant, dû à Kulkarni [Kul87] donne une borne sur les diviseurs premiers du cardinal de $\mathrm{Aut}(\mathcal{C})$.

Théorème 1.25 (Kulkarni) *Soit $\mathcal{C}$ une courbe de genre g. Alors si q est un nombre premier divisant le cardinal du groupe d'automorphismes de $\mathcal{C}$, on a*

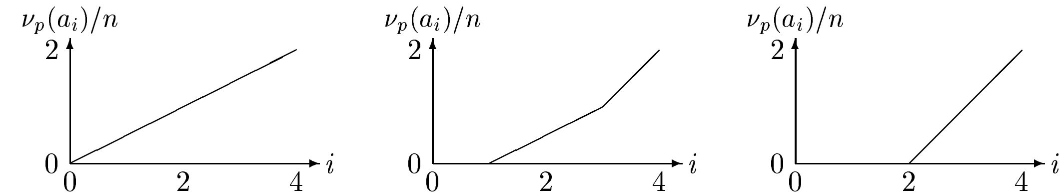
$$q \leqslant 2g + 1.$$

Cas des courbes de genre 2 sur un corps fini

Soit $\mathcal{C}$ une courbe de genre 2 définie sur un corps fini à $q = p^n$ éléments où p est premier. Le polynôme caractéristique du Frobenius de sa Jacobienne est de la forme

$$\chi(t) = t^4 + a_1 t^3 + a_2 t^2 + a_3 t + a_4.$$

Au vu de la section précédente, les polygones de Newton possibles sont au nombre de trois, complètement déterminés par le p-rang (ce qui n'est absolument pas le cas en genre supérieur).



Supersingulier p-rang=1   Ordinaire

Les bornes sur le nombre d'automorphismes se spécialisent en

– Stichtenoth : $\#\mathrm{Aut}(\mathcal{C}) \leqslant 256$.

– Roquette : si $p \geqslant 5$, alors $\#\mathrm{Aut}(\mathcal{C}) \leqslant 84$, sauf pour la courbe $y^2 = x^5 - x$ en caractéristique 5, pour laquelle $\#\mathrm{Aut}(\mathcal{C}) = 120$.

– Nakajima : si $\mathcal{C}$ est ordinaire, alors $\#\mathrm{Aut}(\mathcal{C}) \leqslant 168$.

– Kulkarni : si q premier divise $\#\mathrm{Aut}(\mathcal{C})$, alors $q \leqslant 5$.

En fait, il existe une classification complète des groupes d'automorphismes possibles en genre 2 et le cardinal maximal est 120. Nous reviendrons là-dessus au chapitre 2.

1.5 Courbes hyperelliptiques

1.5.1 Équation et calcul du genre

Nous donnons tout d'abord la définition abstraite d'une courbe hyperelliptique.

Définition 1.21 *Une courbe $\mathcal{C}$ sur un corps K est hyperelliptique s'il existe un morphisme de degré 2 de $\mathcal{C}$ sur une courbe de genre 0.*

On voit immédiatement qu'une courbe elliptique est hyperelliptique : lorsqu'on considère un modèle sous forme de Weierstraß, le morphisme de degré 2 est simplement $(x, y) \mapsto x$. Toute courbe elliptique est munie d'une involution, donnée par l'inversion pour la loi de groupe. Ceci est en fait une propriété générale des courbes hyperelliptiques.

Lemme 1.5 *Soit $\mathcal{C}$ une courbe hyperelliptique et $\varphi : \mathcal{C} \to \mathbb{P}^1$ le morphisme de degré 2. Alors il existe une involution sur $\mathcal{C}$ appelée* involution hyperelliptique *telle qu'un point et son image par l'involution ont même image par φ.*

Notation. L'involution hyperelliptique sera en général notée ι.

Définition 1.22 *Les points fixés par l'involution hyperelliptique sont appelés* points de ramification *de $\mathcal{C}$.*

Remarque. Lorsque la caractéristique est différente de 2, les points de ramification sont en fait les points de Weierstraß d'une courbe hyperelliptique.

Le corps de fonctions de $\mathbb{P}^1(K)$ est simplement le corps des fractions rationnelles $K(x)$. L'existence du morphisme φ correspond donc au fait que le corps de fonctions de $\mathcal{C}$ est une extension de degré 2 de $K(x)$. D'où la forme suivante pour l'équation d'une courbe hyperelliptique :

Lemme 1.6 *Toute courbe hyperelliptique $\mathcal{C}$ sur K admet un modèle affine lisse d'équation*

$$y^2 + h(x)y = f(x),$$

où $h(x)$ et $f(x)$ sont des polynômes à coefficients dans K. L'involution hyperelliptique est alors

$$\iota \ : \ (x, y) \mapsto (x, -y - h(x)).$$

Sous cette forme, le genre de $\mathcal{C}$ est déterminé par les degrés de h et f et le type de la singularité à l'infini.

En caractéristique différente de 2, l'éclatement de cette singularité mène au résultat suivant :

Proposition 1.4 *Soit $\mathcal{C}$ une courbe hyperelliptique de genre g sur un corps de caractéristique différente de 2. Alors $\mathcal{C}$ admet un modèle d'équation $y^2 + h(x)y = f(x)$ avec $\deg f = 2g + 1$ ou $2g + 2$ et $\deg h \leqslant g + 1$.*

Démonstration. Il suffit de construire l'arbre de désingularisation, comme décrit dans [Hac96]. On peut aussi utiliser la formule d'Hurwitz (cf [Har77, p. 301]) : le morphisme de degré 2 admet $2g+2$ points de ramification, en comptant l'infini si le degré de f est impair. On a alors

$$2g(\mathcal{C}) - 2 = 2(2g(\mathbb{P}^1) - 2) + \sum_{P \text{ ramifié}} (e_P - 1),$$

où e_P désigne l'indice de ramification du point P, c'est-à-dire 2 dans notre calcul. Comme $g(\mathbb{P}^1) = 0$, on obtient

$$g(\mathcal{C}) = g.$$

□

En caractéristique 2, cela se complique quelque peu. Nous donnons quelques résultats pour le genre 2, de manière à montrer qu'en général des considérations de degré ne suffisent plus à déterminer le type de singularité à l'infini, et donc le genre :

Proposition 1.5 *Soit $\mathcal{C}$ une courbe hyperelliptique d'équation $y^2 + h(x)y = f(x)$ sur un corps de caractéristique 2. On suppose que cette équation est absolument irréductible et non singulière dans sa partie affine. On note f_i (resp. h_i) le coefficient de x^i dans $f(x)$ (resp. $h(x)$).*

- *Si $\deg f = 5$ et $\deg h \leqslant 2$, alors le genre de $\mathcal{C}$ est 2.*
- *Si $\deg f = 6$ et $\deg h = 1$, alors le genre de $\mathcal{C}$ est 2 si et seulement si f_5 est non nul.*
- *Si $\deg f = 6$ et $\deg h = 2$, alors le genre de $\mathcal{C}$ est 2 si et seulement si $f_5^2 \neq h_2^2 f_6$.*
- *Si $\deg h = 3$ et $\deg f \leqslant 6$, alors le genre de $\mathcal{C}$ est 2.*

Dans les autres cas où $\deg f \leqslant 6$ et $\deg h \leqslant 3$, le genre de $\mathcal{C}$ est au plus 1.

Démonstration. Là encore, la démonstration se réduit à des calculs d'éclatements du point à l'infini. Montrons par exemple la deuxième assertion. Soit $f(x) = f_6x^6 + f_5x^5 + \cdots + f_0$ et $h(x) = h_1x + h_0$, avec f_6 et h_1 non nuls. On commence par homogénéiser et déshomogénéiser l'équation afin de ramener le point à l'infini en $(0,0)$: l'équation homogène est

$$y^2z^4 + (h_1x + h_0z)yz^4 = f_6x^6 + f_5x^5z + \cdots + f_0z^6,$$

et en posant $y = 1$, on obtient

$$\mathcal{C}_0 \ : \ z^4 + (h_1x + h_0z)z^4 = f_6x^6 + f_5x^5z + \cdots + f_0z^6.$$

On étudie le point $P_0 = (0,0) \in \mathcal{C}_0$. C'est un point singulier de multiplicité 4, que l'on éclate en posant $z = tx$. On obtient

$$\mathcal{C}_1 \ : \ t^4 + (h_1 + h_0t)xt^4 = x^2(f_6 + f_5t + \cdots + f_0t^6).$$

Le point P_0 est envoyé sur le point $P_1 = (0,0) \in \mathcal{C}_1$ de multiplicité 2, que l'on éclate par $x = ut$. On obtient

$$\mathcal{C}_2 \ : \ t^2 + (h_1 + h_0t)ut^3 = u^2(f_6 + f_5t + \cdots + f_0t^6).$$

Le point P_1 est envoyé sur le point $P_2 = (0,0) \in \mathcal{C}_2$ de multiplicité 2 que l'on éclate par $u = vt$. On obtient

$$\mathcal{C}_3 \ : \ 1 + (h_1 + h_0t)vt^2 = v^2(f_6 + f_5t + \cdots + f_0t^6).$$

Le point P_2 est envoyé sur $P_3 = (0, \frac{1}{\sqrt{f_6}})$. Par une translation de v, on se ramène au point $(0,0)$: on obtient la courbe

$$\mathcal{C}'_3 \ : \ (h_1 + h_0 t)v't^2 + \frac{1}{\sqrt{f_6}}t^2(h_1 + h_0 t) = v'^2(f_6 + f_5 t + \cdots + f_0 t^6) + \frac{f_5}{f_6}t + \cdots + \frac{f_0}{f_6}t^6,$$

sur laquelle le point $P'_3 = (0,0)$ est non-singulier si et seulement si f_5 est non nul.

Ainsi, si f_5 est non nul, on a terminé la construction de l'arbre de désingularisation et comme on a supposé que le point à l'infini était le seul point singulier de la courbe de départ, on en déduit que le genre est 2 par une formule que l'on peut trouver dans [Hac96, p. 35].

Si f_5 est nul, alors il faut rajouter au moins une étape dans l'arbre de désingularisation, ce qui va diminuer le genre. $\square$

Définition 1.23 *Soit $\mathcal{C}$ une courbe hyperelliptique de genre g, d'équation $y^2 + h(x)y = f(x)$. Lorsque le degré de f est égal à $2g+1$ et le degré de h inférieur ou égal à g, alors le modèle est appelé* imaginaire. *Cela correspond au fait qu'il y a un unique point à l'infini.*

Ce cas est le plus simple et le plus souvent étudié. Nous verrons ci-dessous qu'il est toujours possible de s'y ramener au prix d'une extension du corps de base (au moins en caractéristique différente de 2).

Le cas où (le modèle désingularisé de) la courbe admet deux points à l'infini est appelé *réel.* Le cas où ces deux points sont définis dans une extension du corps de base est appelé imaginaire dans la terminologie classique d'Artin, même si cela se rapproche plus du cas réel. Pour notre part, lorsque nous parlerons de modèle imaginaire par la suite, il s'agira uniquement du cas décrit à la définition ci-dessus.

1.5.2 Forme canonique

Selon le corps de définition, il est parfois possible de simplifier l'équation d'une courbe hyperelliptique.

Corps de caractéristique différente de 2

Soit $\mathcal{C}$ une courbe hyperelliptique de genre g en caractéristique différente de 2. Son équation est de la forme $y^2 + h(x)y = f(x)$. On peut alors effectuer le changement de variables suivant :

$$Y = y + \frac{h(x)}{2}.$$

L'équation devient

$$Y^2 = F(x),$$

où $F(x) = f(x) + \frac{h(x)^2}{4}$ est encore un polynôme de degré $2g+1$ ou $2g+2$. Ainsi à chaque fois que le corps n'est pas de caractéristique 2, on supposera que $h(x) = 0$ dans l'équation générale.

Corps algébriquement clos

Si le corps de base est algébriquement clos et n'est pas de caractéristique 2, on peut « envoyer un point de Weierstraß à l'infini ». Partant d'une équation avec $h(x) = 0$ et $\deg(f) = 2g+2$, le but est de ramener le degré de $f(x)$ à $2g+1$. Soit α une racine de $f(x)$ et β un scalaire quelconque, non racine de $f(x)$ et distinct de α. On effectue le changement de variables suivant :

$$X = \frac{x-\beta}{x-\alpha} \quad \text{et} \quad Y = \frac{y}{(x-\alpha)^{g+1}}.$$

L'équation prend alors la forme

$$Y^2 = F(X),$$

où $F(X)$ est un polynôme de degré $2g+1$. On peut de plus rendre le polynôme $F(X)$ unitaire en opérant une homothétie sur Y de rapport la racine carrée du terme dominant de $F(X)$.

Corps de caractéristique première à $2g+1$

Lorsque l'équation est de la forme

$$y^2 = f(x),$$

où f est unitaire de degré $2g+1$, et que la caractéristique du corps est première à $2g+1$, on peut mettre à 0 le coefficient de degré $2g$ par une translation sur la variable x. On obtient alors une équation de la forme

$$y^2 = x^{2g+1} + f_{2g-1}x^{2g-1} + \cdots + f_0.$$

Remarque importante

Lorsque l'on dira par la suite une phrase du type « soit $\mathcal{C}$ une courbe hyperelliptique d'équation $y^2 = f(x)$ », il faut garder à l'esprit que la courbe que l'on désigne ainsi est un modèle désingularisé de la fermeture projective de la courbe $\mathcal{C}$. Par exemple, lorsque l'on dit que la courbe a deux points à l'infini, cela signifie que l'arbre de désingularisation de l'unique point à l'infini possède deux feuilles. Cet abus de langage est pratique, car il est beaucoup plus facile de manipuler ces équations singulières affines que des modèles projectifs lisses.

1.6 Courbes hyperelliptiques sur $\mathbb{C}$

La Jacobienne d'une courbe hyperelliptique $\mathcal{C}$ de genre g définie sur $\mathbb{C}$ est un groupe de Lie complexe compact de dimension g, elle est donc isomorphe à un tore $\mathbb{C}^g/\Lambda$, où Λ est un réseau. La construction de Λ à partir de $\mathcal{C}$ se fait par des calculs d'intégrales. Le réseau obtenu possède des propriétés particulières, et si l'on veut effectuer la construction inverse, ce n'est pas toujours possible.

1.6.1 Construction de la matrice des périodes

La construction présentée ici est décrite par Mumford dans [Mum83, p. 135] et [Mum84, p. 75].

Soit $\mathcal{C}$ une courbe hyperelliptique complexe de genre g. On peut mettre son équation sous la forme

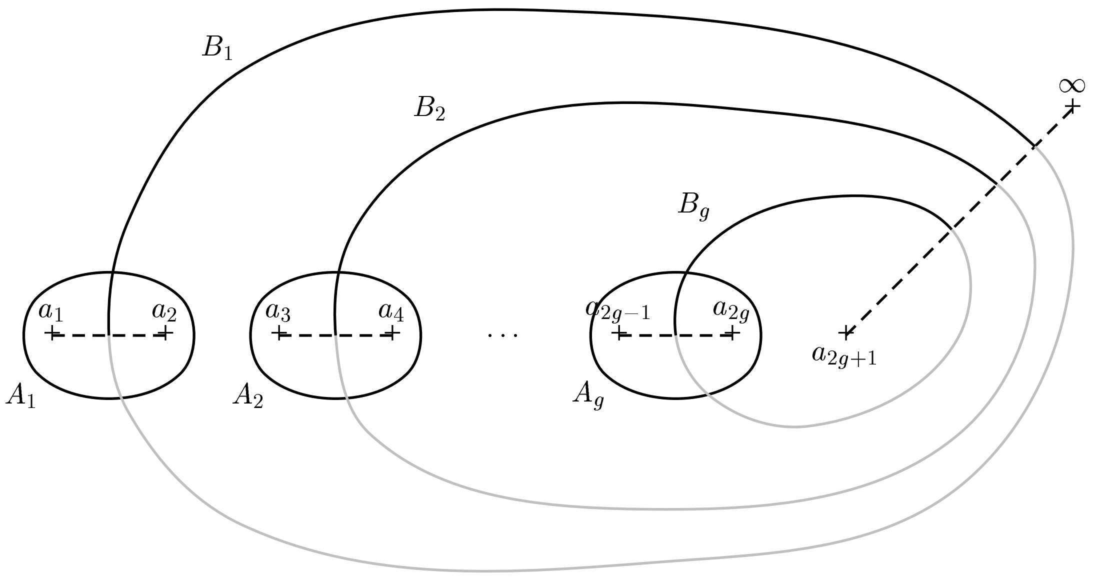
$$y^2 = f(x) = \prod_{1 \leqslant i \leqslant 2g+1} (x - a_i).$$

Le premier groupe d'homologie de la courbe $\mathcal{C}$, vue comme une surface de Riemann, est engendré par $2g$ chemins. Pour construire ces chemins on commence par considérer un ouvert de $\mathbb{C}$ sur lequel la fonction $\sqrt{f(x)}$ est bien définie en tant que fonction holomorphe. On ordonne les indices des a_i de telle sorte que les segments $[a_{2i-1}, a_{2i}]$ pour $1 \leqslant i \leqslant g$ ne se coupent pas. Soit $\mathcal{U}$ l'ouvert complémentaire dans $\mathbb{C}$ des segments $\{[a_{2i-1}, a_{2i}],\ 1 \leqslant i \leqslant g\}$ et d'une demi-droite $[a_{2g+1}, \infty[$ ne coupant pas les segments.

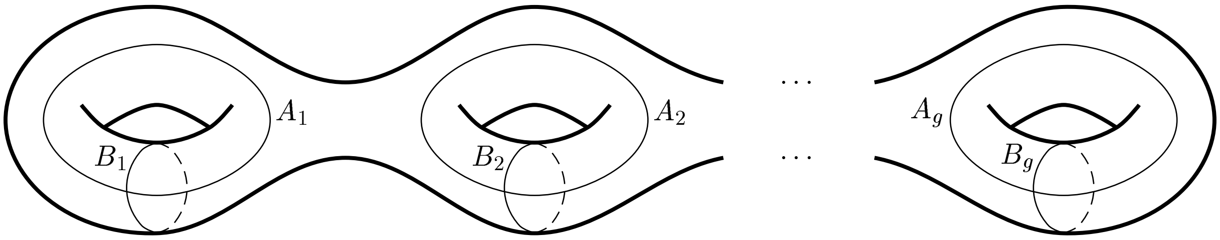
Lemme 1.7 *La fonction $\sqrt{f(x)}$ est holomorphe sur l'ouvert $\mathcal{U}$.*

Démonstration. Lorsque l'on fait un tour autour d'un des a_i en essayant de prolonger par continuité $\sqrt{f(x)}$, on arrive sur l'opposé de la valeur que l'on avait au point de départ. Ainsi faire le tour de deux des a_i permet de retrouver la bonne valeur. La traversée d'un des segments revient à passer à l'opposé. □

On considère les chemins comme sur le dessin ci-dessous dans lequel les lignes pointillées désignent les segments complémentaires de l'ouvert $\mathcal{U}$, et les lignes grisées signifient que l'on prend l'opposé lorsque le chemin croise un segment :



Pour tout $i \neq j$, les chemins A_i et A_j sont disjoints et de même pour B_i et B_j. De plus les chemins A_i et B_j se coupent si et seulement si $i = j$. On a donc bien une base de l'espace d'homologie. Les chemins choisis sont plus compréhensibles si l'on dessine la courbe comme une surface de Riemann à g trous :



Proposition 1.6 *L'espace vectoriel des formes différentielles de première espèce est*

$$\left\{ \frac{P(x)\mathrm{d}x}{y},\ P \text{ polynôme de degré} \leqslant g-1 \right\}.$$

De plus on peut trouver une base $(\omega_i)_{1\leqslant i\leqslant g}$ telle que pour tous i et j

$$\int_{A_i} \omega_j = \delta_{ij}.$$

La preuve de ce résultat se trouve dans [Mum84, p. 77].

Définition 1.24 *La* matrice des périodes *de $\mathcal{C}$ est la matrice $g \times g$ définie par*

$$\Omega = \left(\int_{B_i} \omega_j \right)_{1\leqslant i,j\leqslant g}.$$

On note L_Ω le réseau associé à cette matrice :

$$L_\Omega = \mathbb{Z}^g + \Omega\mathbb{Z}^g.$$

L'intérêt de cette construction est que le théorème suivant permet de relier la matrice des périodes à la Jacobienne de $\mathcal{C}$.

Théorème 1.26 (Abel–Jacobi) *Le tore $\mathbb{C}^g/L_\Omega$ est isomorphe à la Jacobienne de $\mathcal{C}$. Cet isomorphisme est donné explicitement par*

$$\begin{array}{ccc} \mathrm{Jac}(\mathcal{C}) & \to & \mathbb{C}^g/L_\Omega \\ P_1 + \cdots + P_k - k\infty & \mapsto & \left(\sum_{1\leqslant i\leqslant k} \int_\infty^{P_i} \vec{\omega}\right) \bmod L_\Omega \end{array}$$

La preuve de ce théorème se trouve par exemple dans [Kna92, p. 318].

Étant donnée une courbe hyperelliptique sur $\mathbb{C}$, le calcul de la matrice des périodes se fait donc par des calculs d'intégrales le long de contours, puis on peut calculer l'image d'un diviseur par l'isomorphisme d'Abel-Jacobi, là encore en calculant des intégrales. Notons que si l'on change le chemin reliant le point à l'infini à un P_i, la valeur obtenue varie d'un élément de L_Ω, et l'élément du tore est inchangé.

Pour un calcul efficace de la matrice des périodes, on peut remplacer le calcul sur un contour par une intégrale impropre entre deux racines de $f(x)$. Par exemple, pour toute forme différentielle ω,

$$\int_{A_1} \omega = 2 \int_{a_1}^{a_2} \omega.$$

Dans le cas des courbes elliptiques, ces intégrales peuvent s'exprimer comme moyenne arithmético-géométrique de deux nombres, ce qui produit un algorithme très rapide (cf [Coh93, p. 391]). Un algorithme analogue pour les courbes de genre 2 a été développé dans [BM88], d'après un article de Richelot datant d'il y a plus d'un siècle.

1.6.2 Demi-espace de Siegel

Une matrice de périodes n'est pas une matrice quelconque : les intégrales intervenant dans son calcul sont liées par les *relations bilinéaires de Riemann*, ce qui se traduit sur la matrice par le théorème suivant dont on trouvera une démonstration dans [Mum83, p. 139].

Théorème 1.27 *Soit Ω la matrice des périodes d'une courbe de genre g. Alors*

- *la matrice Ω est symétrique,*

– *la partie imaginaire de* Ω *est définie positive, ce qu'on note* $\operatorname{Im}\Omega > 0$.

Définition 1.25 *Le* demi-espace de Siegel *de dimension* g*, noté* $\mathbb{H}_g$ *est défini par*

$$\mathbb{H}_g = \left\{\Omega \in \mathcal{M}_g(\mathbb{C}),\ \Omega = {}^t\Omega \text{ et } \operatorname{Im}(\Omega) > 0\right\}.$$

Dans le cas $g = 1$, on retrouve le demi-plan de Poincaré.

Si la courbe $\mathcal{C}$ est remplacée par une courbe qui lui est isomorphe, alors la matrice des périodes est modifiée par l'action d'une matrice du groupe symplectique. Nous définissons maintenant ce groupe ainsi que son action sur $\mathbb{H}_g$.

Définition 1.26 *Soient* I *et* 0 *les matrices identité et nulle de* $\mathcal{M}_g(\mathbb{C})$*, et* $J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$ *la matrice symplectique de* $\mathcal{M}_{2g}(\mathbb{C})$*. Le* groupe symplectique *de dimension* g *est défini par*

$$\operatorname{Sp}_{2g}(\mathbb{Z}) = \left\{\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathcal{M}_{2g}(\mathbb{Z}),\ \gamma J {}^t\gamma = J\right\}.$$

L'action du groupe symplectique est très proche de l'action bien connue de $\mathrm{SL}_2(\mathbb{Z})$ sur le demi-plan de Poincaré :

Proposition 1.7 *Pour* $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \operatorname{Sp}_{2g}(\mathbb{Z})$ *et* $\Omega \in \mathbb{H}_g$ *on définit une action de groupe par :*

$$\gamma\Omega = (A\Omega + B)(C\Omega + D)^{-1}.$$

Théorème 1.28 *Soient* Ω *et* Ω' *les matrices des périodes de deux courbes* $\mathcal{C}$ *et* $\mathcal{C}'$ *isomorphes. Alors il existe* $\gamma \in \operatorname{Sp}_{2g}(\mathbb{Z})$ *tel que* $\Omega' = \gamma\Omega$.

La réciproque n'est pas vraie, au sens où une matrice Ω dans $\mathbb{H}_g$ ne provient pas forcément d'une courbe : le tore $\mathbb{C}^g/L_\Omega$ est une variété abélienne qui n'est pas toujours la Jacobienne d'une courbe. Ensuite, l'action de $\operatorname{Sp}_{2g}(\mathbb{Z})$ va être invariante pour les classes de variétés abéliennes *principalement polarisées*. Nous ne voulons pas définir cette notion ici. Nous renvoyons à Shimura [Shi68] pour plus de détails. Simplement, lorsque l'on construit la Jacobienne d'une courbe comme précédemment, on obtient une variété abélienne principalement polarisée, la polarisation étant donnée essentiellement par le plongement de la courbe dans sa Jacobienne.

Théorème 1.29 *L'ensemble des classes* $\operatorname{Sp}_{2g}(\mathbb{Z})\backslash\mathbb{H}_g$ *représente les classes d'isomorphismes de variétés abéliennes de dimension* g *principalement polarisées.*

1.6.3 Domaine fondamental

Comme pour le cas elliptique, on définit un domaine fondamental permettant de représenter $\operatorname{Sp}_{2g}(\mathbb{Z})\backslash\mathbb{H}_g$. Il s'agit d'une partie connexe fermée du demi-espace $\mathbb{H}_g$ telle que chaque classe est représentée une fois et une seule, sauf dans le cas où l'on tombe sur la frontière du domaine, on tolère alors plusieurs représentants distincts.

Définition 1.27 *On définit le sous-ensemble* $\mathcal{F}_g$ *de* $\mathbb{H}_g$ *par les conditions suivantes sur* $\Omega = X + iY \in \mathbb{H}_g$ *:*

1. *La matrice symétrique* Y *est réduite au sens de Minkowski,*

2. *On a* $-\frac{1}{2} \leqslant x_{ij} \leqslant \frac{1}{2}$ *pour tous les coefficients de la matrice* $X = (x_{ij})$.

3. *Pour toute matrice symplectique* $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathrm{Sp}_{2g}(\mathbb{Z})$, *on a* $|\det(C\Omega + D)| \geqslant 1$.

Théorème 1.30 *Ce domaine* $\mathcal{F}_g$ *est un domaine fondamental pour le demi-espace de Siegel de dimension* g.

Pour une démonstration de ce théorème, on pourra consulter [Maa71]. Le point clef est de définir la *hauteur* d'un point de $\mathbb{H}_g$ par $h(X + iY) = \det(Y) > 0$. On a alors

$$h\left(\begin{pmatrix} A & B \\ C & D \end{pmatrix} \Omega\right) = |\det(C\Omega + D)|^{-2} h(\Omega).$$

On se ramène donc à étudier les points de hauteur maximale dans une classe d'équivalence.

L'avantage de ce point de vue est qu'il fournit les grandes lignes pour rendre tout cela effectif.

Dans toute la suite, on ne considérera pas d'autre domaine fondamental. On parlera donc *du* domaine fondamental.

Proposition 1.8 *Le domaine fondamental* $\mathcal{F}_g$ *peut être décrit par un nombre fini d'inégalités entre les coefficients de la matrice* Ω.

1.7 Cas particulier du genre 2

1.7.1 Hyperellipticité

Le cas du genre 2 est très particulier : toutes les courbes sont hyperelliptiques.

Théorème 1.31 *Soit* $\mathcal{C}$ *une courbe de genre 2 définie sur un corps* K. *Alors* $\mathcal{C}$ *admet une équation sur* K *de la forme*

$$y^2 + h(x)y = f(x),$$

où h *est un polynôme de degré au plus 3 et* f *un polynôme de degré au plus 6. En particulier,* $\mathcal{C}$ *est hyperelliptique.*

Démonstration. La preuve découle du théorème de Riemann-Roch : on considère un espace $L(D)$ pour un D bien choisi, de sorte que l'on puisse montrer que sa dimension est 11 et qu'il existe deux fonctions x et y telles que cet espace contienne les 12 fonctions

$$\{1, x, x^2, x^3, x^4, x^5, x^6, y, yx, yx^2, yx^3, y^2\}.$$

Celles-ci sont donc liées par une relation de dépendance linéaire et dans les coordonnées données par les fonctions x et y on obtient l'équation voulue (cf [CF96] pour plus de détails). □

1.7.2 Domaine fondamental en dimension 2

Dans le cas de la dimension 2, Gottschling [Got59] a donné explicitement les inégalités décrivant le domaine fondamental (cf proposition 1.8).

On dispose de plus d'un algorithme permettant de ramener un élément quelconque du demi-espace de Siegel dans le domaine fondamental. Cet algorithme est assez proche de celui qui permet de ramener un élément du demi-plan de Poincaré dans le domaine fondamental.

L'algorithme repose sur trois lemmes. Le premier tiré de l'article de Gottschling permet de vérifier le point 3 de la définition pour seulement un nombre fini de matrices, le deuxième montre que l'on peut s'occuper d'abord du point 3 et ensuite des deux points suivants, le troisième est un lemme de finitude qui permet de montrer que l'algorithme termine.

Lemme 1.8 *Un élément de $\Omega \in \mathbb{H}_2$ qui vérifie les inégalités 1 et 2 dans la définition du domaine fondamental vérifie aussi le point 3 pourvu qu'il satisfasse l'inéquation $|det(C\Omega + D)| \geqslant 1$ pour les 19 matrices $M_i = \begin{pmatrix} A_i & B_i \\ C_i & D_i \end{pmatrix}$ suivantes :*

Pour les matrices $M_1, M_2, \ldots, M_{15}$,

$$M_i = \begin{pmatrix} 0 & -I \\ I & D_i \end{pmatrix}, \text{ où } (D_i)_{(1 \leqslant i \leqslant 15)} \text{ décrit } \left\{ \begin{pmatrix} \varepsilon_1 & \varepsilon_2 \\ \varepsilon_2 & \varepsilon_3 \end{pmatrix}, \ \varepsilon_1, \varepsilon_2, \varepsilon_3 \in \{-1, 0, 1\} \right\}$$

Auxquelles on rajoute :

$$M_{16} = \begin{pmatrix} I & -I \\ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix}, \ M_{17} = \begin{pmatrix} I & -I \\ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix},$$

$$M_{18} = \begin{pmatrix} I & 0 \\ \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} & I \end{pmatrix}, \ M_{19} = \begin{pmatrix} -I & 0 \\ \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} & -I \end{pmatrix}.$$

Lemme 1.9 *Soit $\Omega \in \mathbb{H}_2$ vérifiant la condition 3 de la définition du domaine fondamental. Alors pour toute matrice S symétrique entière et toute matrice $A \in \mathrm{GL}_2(\mathbb{Z})$, les matrices $\Omega + S$ et $A\Omega{}^tA$ vérifient encore la condition 3.*

L'utilité de ce lemme réside dans le fait que pour obtenir la réduction de Minkowski on effectue des opérations du type $\Omega \mapsto A\Omega{}^tA$, et pour obtenir la condition 2, des opérations du type $\Omega \mapsto \Omega + S$.

Lemme 1.10 *Étant donné un point $\Omega \in \mathbb{H}_2$ et un réel $\varepsilon > 0$, il n'y a qu'un nombre fini de réels h_0 vérifiant*

1. *$h_0 \geqslant \varepsilon$,*
2. *il existe une matrice $M \in \mathrm{Sp}_4(\mathbb{Z})$ telle que $h_0 = h(M\Omega)$.*

À partir de ces lemmes il est alors aisé de prouver que l'algorithme suivant ramène un point de $\mathbb{H}_2$ dans le domaine fondamental.

Algorithme 1.1 Réduction dans le domaine fondamental
Entrée: Une matrice $\Omega \in \mathbb{H}_2$.
Sortie: Une matrice Ω' du domaine fondamental équivalente à Ω.

1. $\Omega \leftarrow \Omega - [\mathrm{Re}(\Omega)]$, où la notation [.] signifie que l'on remplace les coefficients de la matrice par l'entier le plus proche.

2. Tant que $|\det(C_i\Omega + D_i)| < 1$ pour au moins une des 19 matrices M_i, faire
3. $\quad\Omega \leftarrow M_i\Omega$ pour une de ces matrices,
4. $\quad\Omega \leftarrow \Omega - [\mathrm{Re}(\Omega)]$.
5. Si $\Omega = X + iY$, réduire Y au sens de Minkowski.
6. $\Omega \leftarrow \Omega - [\mathrm{Re}(\Omega)]$.
7. Retourner cette nouvelle matrice Ω.

1.7.3 Invariants d'Igusa

Tout comme les courbes elliptiques sont paramétrées par un invariant usuellement noté j, les courbes de genre 2 sont paramétrées par des invariants, appelés *invariants d'Igusa*. Ces invariants étaient connus au siècle dernier [Cle72, Bol87, Bol88], et Igusa [Igu60] a montré que ceux-ci forment bien un système complet et qu'il est valable en caractéristique quelconque.

Nous rappelons ici brièvement le principe de la construction de ces invariants. En se référera à [Igu60] ou [Mes91a] pour plus de précision.

Définition 1.28 *On appelle* covariant *d'une forme binaire $f = \sum a_i x^{n-i} y^n$, un polynôme*

$$C((a_i), x, y)$$

qui lorsque l'on agit sur f par une transformation inversible du type

$$\begin{cases} x &= \alpha x' + \beta y' \\ y &= \gamma x' + \delta y' \end{cases}$$

est changée ainsi :

$$C((a_i'), x', y') = (\alpha\delta - \beta\gamma)^{-k} C((a_i), x, y).$$

Remarque. Les covariants sont des polynômes homogènes en x et y d'une part, et en les (a_i) d'autre part. Un covariant est donc lui-même une forme binaire. Le degré en x, y est appelé *ordre* du covariant ; et le degré en les (a_i) est appelé *degré* du covariant.

Définition 1.29 *Un covariant ne dépendant pas de x et y (i.e. d'ordre 0) est appelé* invariant.

Remarque. La forme f est elle-même un de ses covariants.

Théorème 1.32 *L'ensemble des covariants d'une forme binaire est une algèbre de type fini.*

Définition 1.30 *Étant données deux formes binaires f et g de degrés respectifs m et n, on peut définir d'autres formes binaires appelées* Überschiebung *de f et g, par la formule*

$$(fg)_k = \frac{(m-k)!(n-k)!}{m!n!} \left(\frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial f}{\partial y}\frac{\partial g}{\partial x} \right)^k,$$

pour k entier supérieur à 1.

Il s'agit de notation symbolique ; on développe par la formule du binôme, puis on regroupe les termes formellement de la manière suivante : $\left(\frac{\partial f}{\partial x}\right)^l \left(\frac{\partial f}{\partial y}\right)^m$ se réécrit $\frac{\partial^{l+m} f}{\partial x^l \partial y^m}$.

Proposition 1.9
1. *Si g et h sont deux covariants d'une forme binaire f, alors pour tout k, l'Überschiebung $(gh)_k$ est aussi un covariant de f.*
2. *Tous les covariants d'une forme binaire f peuvent être obtenus par des Überschiebung successifs à partir de f.*

Le cas des sextiques

Proposition 1.10 *Pour les formes sextiques, il existe cinq invariants A, B, C, D, R, de degrés respectifs 2, 4, 6, 10, 15 tels que*

(i) Les quatre invariants A, B, C, D forment une base des invariants de degré pair.

(ii) Les cinq invariants A, B, C, D, R engendrent tous les invariants.

Définition 1.31 *Les invariants A, B, C, D sont appelés* invariants d'Igusa *de la forme sextique considérée.*

Les invariants d'Igusa s'expriment directement comme fonctions symétriques des racines de la sextique, par exemple l'invariant D est son discriminant. Ceci permet d'avoir des formules très simples pour le calcul de ces invariants.

Dans le cas de la caractéristique différente de 2, nous avons vu que l'équation d'une courbe hyperelliptique de genre 2 est de la forme

$$y^2 = f(x),$$

où f est un polynôme de degré 6 (si f est de degré 5, on peut toujours transformer l'équation de manière à obtenir le degré 6). On peut donc associer à cette courbe la forme sextique $t^6 f(x/t)$. L'intérêt de toute cette construction est qu'un isomorphisme entre courbes hyperelliptiques ayant cette forme d'équation se traduit par une action du type de la définition 1.28. On obtient alors le théorème fondamental suivant :

Théorème 1.33 *En caractéristique différente de 2, deux courbes de genre 2 sont isomorphes sur une clôture algébrique si et seulement si les formes sextiques associées se déduisent l'une de l'autre par une transformation de GL_2, i.e. ont des invariants d'Igusa compatibles.*

Explicitons cette proposition :

Soient $y^2 = f(x)$ et $y'^2 = f'(x')$ les équations de deux courbes de genre 2 isomorphes, telles que l'on passe de l'une à l'autre par la transformation

$$(x', y') = (\frac{ax+b}{cx+d}, \frac{uy}{(cx+d)^3}),$$

où $\Delta = ad - bc$ et u sont non nuls.

Alors pour tout invariant I_s de degré S des formes sextiques, on a

$$I_s(f) = M^s I_s(f'),$$

avec $M = \Delta^3 u^{-2}$.

Remarque. Comme nous l'avons dit, calculer les invariants d'une courbe est une tâche aisée. Réciproquement, étant donnés des invariants, retrouver l'équation d'une courbe correspondante est beaucoup moins facile. Dans [Mes91a], Mestre donne un algorithme effectuant ce travail. Le principal problème est qu'il nécessite de trouver un point rationnel sur une conique. Si le corps de base est un corps fini, c'est très simple, mais si on travaille sur un corps de nombres, cela peut devenir extrêmement difficile. De plus le modèle obtenu est en général très loin d'être minimal.

Invariants absolus

Afin d'obtenir des valeurs qui soient réellement inchangées lorsque l'on reste dans la même classe d'isomorphisme, suivant Igusa, on introduit des quotients d'invariants :

Définition 1.32 *Le quotient de deux invariants de même degré est appelé* invariant absolu. *Sa valeur est caractérisée par la classe d'isomorphisme de courbes de genre 2.*

Les invariants absolus d'une courbe $\mathcal{C}$ sont aussi appelés les *modules* de $\mathcal{C}$.

A priori, trois invariants absolus suffisent pour déterminer la classe d'isomorphisme. Cependant, ceci n'est vrai que localement : il peut être nécessaire de changer de système d'invariants pour recouvrir tous l'espace des classes d'isomorphisme.

Nous choisirons le système de coordonnées suivant, tiré de l'article [Igu62] :

$$\begin{aligned} j_1 &= 144\frac{B}{A^2}, \\ j_2 &= -1728\frac{AB-3C}{A^3}, \\ j_3 &= 486\frac{D}{A^5}. \end{aligned}$$

Ce système n'est plus valable lorsque l'invariant A s'annule ou lorsque la caractéristique du corps est 2 ou 3.

Théorème 1.34 *En caractéristique différente de 2 ou 3, deux courbes de genre 2 ayant des invariants A non nuls sont isomorphes si et seulement si elles ont mêmes triplets d'invariants absolus (j_1, j_2, j_3).*

Remarque. Igusa a proposé un moyen d'étendre ces définitions à des corps de caractéristique quelconque, notamment le cas de la caractéristique 2. Pour cela, il utilise une autre forme canonique pour l'équation d'une courbe de genre 2, valable en toute généralité.

Chapitre 2

Invariants de Jacobiennes $(2,2)$-décomposables

Parmi les courbes de genre 2, celles dont la Jacobienne est décomposable présentent un intérêt particulier. Par exemple, les records actuels de rang ou de torsion sont obtenus pour des courbes de ce type [HLP00]. C'est aussi dans ce cadre que la méthode de Dem'janenko-Manin permet de déterminer tous les points rationnels d'une courbe [Kul99].

Le but du présent chapitre qui reprend essentiellement l'article écrit en collaboration avec É. Schost [GSa] est de rendre explicite le théorème suivant :

Théorème 2.1 *Soit $\mathcal{C}$ une courbe de genre 2 admettant une involution non triviale. Alors il existe au plus deux courbes elliptiques quotients de degré 2 de sa Jacobienne à isomorphisme près.*

Dans ce cas, nous donnons des formules algébriques reliant les modules de $\mathcal{C}$ au j-invariant des courbes elliptiques dont la Jacobienne de $\mathcal{C}$ est le produit.

Les modules des courbes de genre 2 forment une variété de dimension 3. Nous supposerons que A est non nul, et prendrons les coordonnées locales décrites en section 1.7.3. Le cas $A = 0$ sera traité en annexe.

Le corps de base est supposé de caractéristique différente de 2, 3 et 5. On élimine la caractéristique 2 car les invariants absolus que nous avons choisis ainsi que la forme de l'équation d'une courbe hyperelliptique ne sont plus valables dans ce cas. De même, on élimine la caractéristique 3 pour pouvoir choisir un modèle canonique de courbes elliptiques. Pour la caractéristique 5, des groupes d'automorphismes spécifiques apparaissent ; on élimine donc aussi ce cas.

Nous remercions Philippe Satgé pour sa lecture attentive de notre travail. Ses commentaires et suggestions nous ont été d'une grande aide.

2.1 Groupe d'automorphismes d'une courbe de genre 2

Définition 2.1 *La Jacobienne d'une courbe $\mathcal{C}$ de genre 2 est dite (2,2)-décomposable s'il existe une (2,2)-isogénie entre $\mathrm{Jac}(\mathcal{C})$ et un produit de courbes elliptiques $\mathcal{E}_1 \times \mathcal{E}_2$. On dit alors que $\mathcal{E}_1$ est un quotient de degré 2 de $\mathrm{Jac}(\mathcal{C})$.*

Remarque. Le préfixe $(2,2)$ signifie que le noyau de l'isogénie a pour structure de groupe $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.

Le lemme suivant est en substance dans [Igu60].

Lemme 2.1 *Soit $\mathcal{C}$ une courbe de genre 2. Il existe une surjection de l'ensemble des involutions non hyperelliptiques de $\mathcal{C}$ sur les classes d'isomorphisme des courbes elliptiques quotients de degré 2 de la Jacobienne de $\mathcal{C}$. En particulier, la Jacobienne de $\mathcal{C}$ est (2,2)-décomposable si et seulement si $\mathcal{C}$ admet une involution autre que l'involution hyperelliptique.*

Démonstration. Soit τ une involution de $\mathcal{C}$ non hyperelliptique. La courbe $\mathcal{C}$ quotientée par τ est une courbe $\mathcal{E}$ de genre 1 [Igu60]. Alors $\mathcal{E}$ est un quotient de la Jacobienne de $\mathcal{C}$, et on construit ainsi une application qui à chaque involution non triviale de $\mathcal{C}$ associe une classe d'isomorphisme de courbes elliptiques.

Cette application est surjective. En effet, soit $\mathcal{E}$ une courbe de genre 1 quotient de degré 2 de $\mathrm{Jac}(\mathcal{C})$. Il existe alors un morphisme φ de degré 2 de $\mathcal{C}$ sur $\mathcal{E}$. Pour p un point générique de $\mathcal{C}$, la fibre $\varphi^{-1}(\varphi(p))$ est de la forme $\{p, q(p)\}$, où q est une fraction rationnelle de p. On définit alors l'involution τ, qui à p associe $q(p)$. La courbe $\mathcal{E}$ est de genre 1, donc τ n'est pas l'involution hyperelliptique. Cette construction est l'inverse de la précédente. Le morphisme φ n'est pas unique, aussi l'application que l'on construit n'est-elle pas injective. $\square$

Bolza [Bol88], puis Igusa [Igu60] et Lange [Lan76] ont classifié les courbes admettant des automorphismes, en particulier les involutions. Les modules des courbes de genre 2 admettant une involution non triviale forment une sous-variété de dimension 2 de la variété des modules ; nous la noterons $\mathcal{H}_2$. Dans nos coordonnées locales, une équation de cette hypersurface est donnée par la formule ci-dessous, la construction explicite est donnée dans [Mes91a].

$$\begin{aligned}
&839390038939659468275712 j_3^2 + 921141332169722324582400000 j_3^3 + 32983576347223130112000 j_1^2 j_3^2 \\
&+ 182200942574622720 j_3 j_1 j_2^2 - 374813367582081024 j_3 j_1^2 j_2 + 9995023135522160640000 j_3^2 j_1 j_2 \\
&+ 94143178827 j_2^4 - 562220051373121536 j_3 j_2^2 - 562220051373121536 j_3 j_1^3 + 43381176803481600 j_3 j_2^3 \\
&- 71964166575759556608000 j_3^2 j_2 - 388606499509101605683200 j_3^2 j_1 - 1156831381426176 j_1^5 j_3 \\
&- 31381059609 j_1^7 + 62762119218 j_1^4 j_2^2 + 13947137604 j_1^3 j_2^3 - 31381059609 j_1 j_2^4 - 188286357654 j_1^3 j_2^2 \\
&- 6973568802 j_1^6 j_2 + 192612425007458304 j_1^4 j_3 + 94143178827 j_1^6 - 6973568802 j_2^5 \\
&+ 28920784535654400 j_1^2 j_3 j_2^2 + 164848471853230080 j_1^3 j_3 j_2 = 0.
\end{aligned}$$

Une courbe $\mathcal{C}$ de genre 2 admet toujours l'involution hyperelliptique ι, qui commute avec tous ses autres automorphismes. On appellera *groupe réduit d'automorphismes de $\mathcal{C}$* le quotient de son groupe d'automorphismes par $\{1, \iota\}$. On peut alors classifier les points de $\mathcal{H}_2$ par leur groupe réduit d'automorphismes $\mathcal{G}$. On a les cinq possibilités suivantes :

1. Le groupe $\mathcal{G}$ est le groupe diédral D_6, ce qui correspond au point de $\mathcal{H}_2$ associé à la courbe $y^2 = x^6 + 1$.

2. Le groupe $\mathcal{G}$ est le groupe symétrique $\mathfrak{S}_4$, ce qui correspond à la courbe $y^2 = x^5 - x$.

3. Le groupe $\mathcal{G}$ est le groupe diédral D_3. Les points correspondants forment une courbe $\mathcal{D}$ de $\mathcal{H}_2$ privée des 2 points précédents.

4. Le groupe $\mathcal{G}$ est le groupe de Klein V_4. Les points forment une courbe $\mathcal{V}$ de $\mathcal{H}_2$ privée des 2 points de $\mathcal{H}_2$ correspondant aux cas 1 et 2 ; ces points sont les seules intersections de $\mathcal{D}$ et $\mathcal{V}$.

5. Le groupe $\mathcal{G}$ est $\mathbb{Z}/2\mathbb{Z}$, ce qui correspond à l'ouvert dense $\mathcal{U}$ formé de $\mathcal{H}_2$ privé des courbes précédentes. On appellera ce cas le cas générique.

Deux involutions sont conjuguées dans le groupe d'automorphismes si et seulement si les courbes quotients associées sont isomorphes. Ainsi les classes d'isomorphisme de courbes quotients sont en bijection avec les classes de conjugaison non triviales d'éléments d'ordre 2. Le théorème 2.1 dit que ces classes sont au nombre de deux. Un moyen de le prouver serait donc de dénombrer ces classes de conjugaison pour tous les groupes d'automorphismes possibles. Pour cela, on ne peut pas se contenter de regarder le groupe réduit, et cela nécessiterait d'étudier des groupes d'ordre allant jusqu'à 48. Nous présentons ici une approche plus calculatoire, mais qui présente l'avantage de fournir des formules explicites liant les invariants.

Dans chacun des cas, nous allons expliciter l'ensemble des involutions, calculer les invariants des courbes elliptiques associées et vérifier que l'on en obtient en tout au plus 2. On traite les involutions par groupe de deux : τ et $\tau\iota$. En général, ces deux involutions donnent des courbes elliptiques distinctes ; sur la courbe $\mathcal{V}$ et deux points isolés, à chaque paire $(\tau, \tau\iota)$ correspond un seule courbe quotient.

2.2 Cas générique

2.2.1 Forme de Rosenhain

Notre but est d'obtenir explicitement un polynôme de degré 2 donnant les j-invariants en fonction des modules (j_1, j_2, j_3). Le premier pas consiste à obtenir les j-invariants à partir d'une forme de Rosenhain. Nous prenons comme point de départ [Igu60] où l'on trouve la forme de Rosenhain d'une courbe (2,2)-décomposable.

Théorème 2.2 *Soit $\mathcal{C}$ une courbe de genre 2 (2,2)-décomposable. Alors, sur une clôture algébrique de son corps de définition, $\mathcal{C}$ est isomorphe à une courbe d'équation*

$$y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu), \text{ où } \mu = \nu\frac{1-\lambda}{1-\nu},$$

avec λ, ν, μ deux à deux distincts et différents de 0 et 1. La Jacobienne de $\mathcal{C}$ est alors isogène au produit des courbes elliptiques dont les formes de Legendre sont $y^2 = x(x-1)(x-\Lambda)$, avec Λ racine de l'équation

$$\nu^2\lambda^2\Lambda^2 + 2\nu\mu(-2\nu+\lambda)\Lambda + \mu^2 = 0. \tag{2.1}$$

Démonstration. La courbe $\mathcal{C}$ a 6 points de Weierstraß, et un isomorphisme de $\mathcal{C}$ vers une autre courbe sera entièrement déterminé par les images de 3 de ces points. Notons τ une involution non triviale sur $\mathcal{C}$. Soient P_1, P_2, P_3 des points de Weierstraß de $\mathcal{C}$ représentant chaque orbite sous l'action de τ. La courbe $\mathcal{C}'$ obtenue en envoyant P_1, P_2, P_3 sur respectivement 0, 1, ∞ a une équation de la forme

$$y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu).$$

La courbe n'est pas singulière, donc λ, ν, μ sont deux à deux distincts et différents de 0 et 1. On note encore τ l'image par l'isomorphisme de l'involution. Celle-ci permute les nouveaux points de Weierstraß, et quitte à échanger les noms de λ, μ, ν, on a $\tau(0) = \lambda$, $\tau(1) = \mu$ et $\tau(\infty) = \nu$. D'autre part τ est de la forme

$$\tau(x, y) = \left(\frac{ax+b}{cx+d}, \frac{wy}{(cx+d)^3}\right),$$

et le fait que c'est une involution impose $a = -d$ et $w = \pm(ad - bc)^{3/2}$. L'involution τ peut ainsi être déterminée à l'aide des deux images $\tau(0) = \lambda$ et $\tau(\infty) = \nu$. On obtient alors

$$\tau(x, y) = \left(\nu \frac{x - \lambda}{x - \nu}, \frac{u^3 y}{(x - \nu)^3}\right),$$

où

$$u = \pm\sqrt{\nu(\nu - \lambda)}.$$

La relation $\tau(1) = \mu$ fournit alors la première assertion

$$\mu = \nu \frac{1 - \lambda}{1 - \nu}.$$

L'étape suivante est de trouver une courbe isomorphe à $\mathcal{C}'$ pour laquelle l'involution sera $(x, y) \mapsto (-x, y)$. Pour cela on cherche de nouveau les coefficients d'une transformation du type $\varphi : x \mapsto (ax + b)/(cx + d)$ telle que $\varphi(0) = -\varphi(\lambda)$, $\varphi(1) = -\varphi(\mu)$, $\varphi(\infty) = -\varphi(\nu)$. On vérifie que la transformation

$$\varphi(x) = \frac{x - \nu - u}{x - \nu + u},$$

convient ainsi que tous ses multiples. La courbe $\mathcal{C}$ est donc isomorphe à la courbe $\mathcal{C}''$ d'équation $y^2 = (x^2 - x_1^2)(x^2 - x_2^2)(x^2 - x_3^2)$, où

$$x_1 = \varphi(\infty) = 1, \quad x_2 = \varphi(0) = \frac{\nu - u}{\nu + u}, \quad x_3 = \varphi(1) = \frac{1 - (\nu - u)}{1 - (\nu + u)}.$$

Par la suite, le morphisme $(x, y) \mapsto (x^2, y)$ envoie $\mathcal{C}''$ sur la courbe elliptique $\mathcal{E}$ d'équation

$$y^2 = (x - 1)(x - x_2^2)(x - x_3^2),$$

dont une forme de Legendre est donnée par $y^2 = x(x - 1)(x - \Lambda)$ avec

$$\Lambda = \frac{x_2^2 - x_3^2}{1 - x_3^2} = \frac{\mu}{\left(\nu \pm \sqrt{\nu(\nu - \lambda)}\right)^2}.$$

On obtient ainsi le résultat. Remarquons qu'aucun dénominateur ne peut s'annuler et que la courbe $\mathcal{E}$ est bien non singulière, d'après les conditions sur λ, μ, ν. □

Corollaire 2.1 *Soient $\mathcal{C}$ une courbe dont les modules appartiennent à $\mathcal{U}$, et λ, μ, ν définis comme dans le théorème précédent. Alors les j-invariants des courbes elliptiques quotients de la Jacobienne de $\mathcal{C}$ sont les solutions d'une équation*

$$j^2 + c_1(\lambda, \nu) j + c_0(\lambda, \nu) = 0, \tag{2.2}$$

où c_0 et c_1 sont des fractions rationnelles en deux variables.

Démonstration. Le j-invariant d'une courbe elliptique est relié à une forme de Legendre par la relation

$$\Lambda^2(\Lambda - 1)^2 j - 2^8(\Lambda^2 - \Lambda + 1)^3 = 0. \tag{2.3}$$

Le théorème précédent donne deux courbes elliptiques quotients de la Jacobienne de $\mathcal{C}$, et sur $\mathcal{U}$ ce sont les seules. Le polynôme attendu s'obtient ainsi par le calcul du résultant entre les équations 2.3 et 2.1 et en tenant compte de la relation $\mu = \nu \frac{1-\lambda}{1-\nu}$. □

On n'écrira pas ici les valeurs de $c_0(\lambda, \nu)$ et $c_1(\lambda, \nu)$. On connaît l'expression des modules en fonction de λ et ν, on pourrait donc obtenir les coefficients c_0 et c_1 en fonction des modules de $\mathcal{C}$ par un calcul d'élimination. L'approche suivante est moins directe, mais produit des formules plus concises.

2.2.2 Les invariants Ω et Υ

Nous introduisons dans cette partie deux invariants caractéristiques des classes d'isomorphismes de courbes (2,2)-décomposables. Ces grandeurs sont invariantes pour l'action du groupe décrit ci-dessous :

Théorème 2.3 *Soit $\mathcal{C}$ une courbe dont les modules appartiennent à $\mathcal{H}_2$. Alors il existe 24 triplets $(\lambda, \mu = \nu\frac{1-\lambda}{1-\nu}, \nu)$ pour lesquels la courbe d'équation $y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu)$ est isomorphe à $\mathcal{C}$. L'ensemble de ces triplets est muni de l'action transitive d'un groupe isomorphe à l'unique sous-groupe d'ordre 24 de $PGL(2,5)$.*

Démonstration. Le théorème 2.2 fournit un triplet $(\lambda_1, \mu_1, \nu_1)$ satisfaisant les conditions requises. On peut donc supposer que $\mathcal{C}$ est la courbe correspondant à ce triplet. Toute courbe isomorphe à $\mathcal{C}$ est obtenue par une transformation birationnelle

$$x \mapsto \frac{ax+b}{cx+d}.$$

On exige de plus que la courbe soit mise sous forme de Rosenhain, donc la transformation doit envoyer 3 des 6 points de Weierstraß de $\mathcal{C}$ $(0, 1, \infty, \lambda_1, \mu_1, \nu_1)$ sur les points $(0, 1, \infty)$. L'ensemble des transformations homographiques correspondantes forment un groupe d'ordre 120 et le test exhaustif montre que seules 24 des formes obtenues vérifient la condition supplémentaire liant les nouveaux λ, μ et ν.

Notons $(\lambda_i, \mu_i, \nu_i)_{i=1,\ldots,24}$ les triplets solutions. On passe de la forme de Rosenhain donnée par $\{0, 1, \infty, \lambda_i, \mu_i, \nu_i\}$ à une forme donnée par $\{0, 1, \infty, \lambda_j, \mu_j, \nu_j\}$ par applications successives aux 6 éléments des transformations $\sigma_1(x) = 1/x$, $\sigma_2(x) = 1-x$, $\sigma_3(x) = \frac{x-\lambda_i}{1-\lambda_i}$, $\sigma_4(x) = x/\mu_i$. Ces éléments engendrent un groupe isomorphe à l'unique sous-groupe d'ordre 24 de $PGL(2,5)$, qui agit sur les triplets (λ, μ, ν) selon le tableau suivant :

Transformation	σ_1	σ_2	σ_3	σ_4
λ_i	$\frac{1}{\nu_i}$	$1-\mu_i$	$\frac{\lambda_i}{\lambda_i-1}$	$\frac{\lambda_i}{\mu_i}$
μ_i	$\frac{1}{\mu_i}$	$1-\lambda_i$	$\frac{\mu_i-\lambda_i}{1-\lambda_i}$	$\frac{1}{\mu_i}$
ν_i	$\frac{1}{\lambda_i}$	$1-\nu_i$	$\frac{\nu_i-\lambda_i}{1-\lambda_i}$	$\frac{\nu_i}{\mu_i}$

□

Les fonctions symétriques en les 24 triplets $(\lambda_i, \mu_i, \nu_i)$ sont des invariants de la classe d'isomorphisme de $\mathcal{C}$; il est donc naturel d'y chercher 2 invariants caractérisant ces classes.

Définition 2.2 *Soient $\mathcal{C}$ une courbe dont les modules appartiennent à $\mathcal{H}_2$, et les triplets λ_i, μ_i, ν_i définis comme précédemment. On appelle Ω et Υ les deux grandeurs suivantes :*

$$\begin{aligned} \Omega &= \sum_{i=1}^{24} \nu_i^2 \\ \Upsilon &= \sum_{i=1}^{24} \lambda_i \nu_i, \end{aligned}$$

Ces grandeurs ne dépendent que de la classe d'isomorphisme de $\mathcal{C}$.

2.2.3 Formules

Nous allons relier les modules (j_1, j_2, j_3) à ces nouveaux invariants d'un côté, puis écrire le polynôme minimal des j-invariants en fonction de ces invariants. La proposition suivante prouve que Ω et Υ caractérisent les classes d'isomorphisme de ces courbes.

Proposition 2.1 *Soient $\mathcal{C}$ une courbe dont les modules sont dans $\mathcal{H}_2$, et Ω et Υ. Sous réserve qu'elles soient définies, on a alors les égalités suivantes :*

$$
\begin{aligned}
j_1 &= \frac{36(\Omega-2)\Upsilon^2}{(\Omega-8)(2\Upsilon-3\Omega)^2}, \\
j_2 &= -\frac{216\Upsilon^2(\Omega\Upsilon+\Upsilon-27\Omega)}{(\Omega-8)(2\Upsilon-3\Omega)^3}, \\
j_3 &= -\frac{243\Omega\Upsilon^4}{64(\Omega-8)^2(2\Upsilon-3\Omega)^5}.
\end{aligned}
$$

Le système précédent ne peut s'inverser que si (j_1, j_2, j_3) est sur $\mathcal{H}_2$. Dans ce cas, Ω et Υ sont donnés par la proposition suivante :

Proposition 2.2 *Soient $\mathcal{C}$ une courbe dont les modules appartiennent à $\mathcal{H}_2$. Alors les invariants Ω et Υ sont donnés en fonction des modules (j_1, j_2, j_3) par les formules suivantes, lorsqu'elles ont un sens :*

$$
\begin{aligned}
\Omega = {} & (349360128 j_1 j_3 - 29859840 j_3 j_2 + 1911029760000 j_3^2 + 972 j_1^2 j_2 - 110730240 j_1^2 j_3 - 45 j_1 j_2^2 \\
& -12441600 j_1 j_3 j_2 + 6 j_2^3 + 45 j_1^4 - 330 j_1^3 j_2 - 56 j_1^2 j_2^2 - 16 j_1^5)\, / \\
& (-26873856 j_1 j_3 - 14929920 j_3 j_2 + 955514880000 j_3^2 + 3732480 j_1^2 j_3 - 9 j_1 j_2^2 + 4147200 j_1 j_3 j_2 \\
& +3 j_2^3 + 9 j_1^4 - 3 j_1^3 j_2 + 2 j_1^2 j_2^2 - 2 j_1^5), \\[1ex]
\Upsilon = {} & 3/4(162 j_1^4 - 483729408 j_1 j_3 + 171992678400000 j_3^2 + 67184640 j_1^2 j_3 - 36 j_1^5 - 134369280 j_3 j_2 \\
& +162 j_1 j_2^2 + 45 j_2^3 + 35251200 j_1 j_3 j_2 - 45 j_1^3 j_2 - 72 j_1^2 j_2^2 - 6912000 j_3 j_2^2 - 20 j_1 j_2^3 - 4 j_1^4 j_2) \\
& (349360128 j_1 j_3 - 29859840 j_3 j_2 + 1911029760000 j_3^2 + 972 j_1^2 j_2 - 110730240 j_1^2 j_3 - 45 j_1 j_2^2 \\
& -12441600 j_1 j_3 j_2 + 6 j_2^3 + 45 j_1^4 - 330 j_1^3 j_2 - 56 j_1^2 j_2^2 - 16 j_1^5)\, / \\
& \big((27 j_1^4 + 161243136 j_1 j_3 + 1433272320000 j_3^2 - 53498880 j_1^2 j_3 - 9 j_1^5 + 44789760 j_3 j_2 + 486 j_1^2 j_2 \\
& +135 j_1 j_2^2 - 23846400 j_1 j_3 j_2 - 162 j_1^3 j_2 - 81 j_1^2 j_2^2 - 3456000 j_3 j_2^2 - 10 j_1 j_2^3 - 2 j_1^4 j_2) \\
& (-26873856 j_1 j_3 - 14929920 j_3 j_2 + 955514880000 j_3^2 + 3732480 j_1^2 j_3 - 9 j_1 j_2^2 + 4147200 j_1 j_3 j_2 \\
& +3 j_2^3 + 9 j_1^4 - 3 j_1^3 j_2 + 2 j_1^2 j_2^2 - 2 j_1^5)\big).
\end{aligned}
$$

Démonstration. Les formules se vérifient en exprimant j_1, j_2, j_3 ainsi que Ω et Υ en fonction de λ et ν. $\square$

Remarque. Les invariants Ω et Υ sont des fractions rationnelles définies sur la variété $\mathcal{H}_2$. Les formules que nous donnons ne sont pas forcément les plus simples.

Proposition 2.3 *Soit $\mathcal{C}$ une courbe dont les modules sont dans $\mathcal{U}$. Les j-invariants des courbes elliptiques quotients de la Jacobienne de $\mathcal{C}$ sont les racines de $j^2 + c_1 j + c_0$, où c_0 et c_1 sont donnés par les formules suivantes, lorsqu'elles ont un sens :*

$$
\begin{aligned}
c_0 &= \frac{4096\Upsilon^2(\Omega-32)^3}{\Omega^2(\Omega-8)}, \\[1ex]
c_1 &= -\frac{128\Upsilon(\Omega^2-4\Omega\Upsilon+56\Omega-512)}{\Omega(\Omega-8)}.
\end{aligned}
$$

Démonstration. Les coefficients c_0 et c_1 sont déterminés en fonction de λ et ν dans le corollaire 2.1. De même, Ω et Υ sont définis à partir de λ et ν. On vérifie alors l'égalité des deux membres. $\square$

Les deux dernières propositions permettent de réécrire le polynôme minimal 2.2 sous la forme

$$j^2 + c_1(j_1, j_2, j_3)j + c_0(j_1, j_2, j_3) = 0,$$

où $c_1(j_1, j_2, j_3)$ et $c_0(j_1, j_2, j_3)$ sont des fractions rationnelles à trois indéterminées que l'on se garde bien de développer.
Remarque. Les cas d'annulation des dénominateurs sont traités en annexe.

2.3 Groupe diédral

2.3.1 Formules

Nous rappelons sans démonstration ce résultat dû à Igusa [Igu60].

Théorème 2.4 *Soit $\mathcal{C}$ une courbe de genre 2. Le groupe réduit d'automorphismes de $\mathcal{C}$ est D_3 si et seulement si $\mathcal{C}$ est isomorphe à une courbe d'équation*

$$y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu), \text{ où } \mu = \frac{1}{1-\lambda}, \text{ et } \nu = 1 - \frac{1}{\lambda}, \tag{2.4}$$

avec λ différent de 0, de 1 et de $(1 \pm \sqrt{3})/2$.

En termes d'invariants d'Igusa, $\mathcal{C}$ est (2,2)-décomposable de groupe réduit d'automorphismes D_3 si et seulement si ses modules appartiennent à la courbe $\mathcal{D}$ d'équations :

$$\begin{aligned}
j_1 j_2^2 - 297 j_1 j_2 - 90 j_2^2 - 725760 j_1 j_3 + 172800 j_2 j_3 & \\
-2187 j_1 - 243 j_2 + 169641216 j_3 &= 0, \\
7 j_2^3 - 57600 j_3 j_1 j_2 + 8991 j_1 j_2 + 2646 j_2^2 - 34774272 j_3 j_1 - 22394880 j_3 j_2 & \\
-9953280000 j_3^2 + 65610 j_1 + 7290 j_2 - 4901119488 j_3 &= 0, \\
-81 j_1 + 21 j_1^2 - 9 j_2 + 5 j_1 j_2 + 864000 j_3 &= 0.
\end{aligned}$$

Ce résultat s'obtient par un calcul d'élimination, puisque 2.4 permet d'exprimer les modules de $\mathcal{C}$ en fonction de λ. Le résultat suivant montre le théorème 2.1 dans le cas où les modules sont sur $\mathcal{D}$.

Théorème 2.5 *Soit $\mathcal{C}$ une courbe dont les modules sont sur $\mathcal{D}$. Alors les j-invariants des courbes elliptiques quotients de la Jacobienne de $\mathcal{C}$ sont les racines de $j^2 + c_1 j + c_0$, où c_0 et c_1 sont donnés par les formules suivantes :*

$$\begin{aligned}
c_1 &= \frac{3}{8}\frac{-85221 j_1 j_2 - 69228 j_1^2 - 6621 j_2^2 + 6054374400 j_3 j_1 + 692576000 j_3 j_2 - 5952061440 j_3}{j_3(4705 j_2 + 21492 j_1 - 129816)}, \\
c_0 &= -81\frac{2373 j_1^2 + 1412 j_1 j_2 + 210 j_2^2 + 33696000 j_3 j_1 + 4320000 j_3 j_2 - 246067200 j_3}{j_3(5 j_2 + 27 j_1 - 108)}.
\end{aligned}$$

Démonstration. Quand la courbe $\mathcal{C}$ est sous la forme 2.4, les automorphismes réduits peuvent être explicités. Dans le tableau suivant, u désigne $\pm\sqrt{\lambda^2 - \lambda + 1}$.

	transformation	ordre
Id	$(x, y) \mapsto (x, y)$	1
τ_1	$(x, y) \mapsto \left(\frac{\lambda - x}{(\lambda-1)x+1}, \frac{u^3 y}{((\lambda-1)x+1)^3}\right)$	2
τ_2	$(x, y) \mapsto \left(\frac{(\lambda-1)x+1}{\lambda x+1-\lambda}, \frac{u^3 y}{(\lambda x+1-\lambda)^3}\right)$	2
τ_3	$(x, y) \mapsto \left(\frac{\lambda x+1-\lambda}{x-\lambda}, \frac{u^3 y}{(x-\lambda)^3}\right)$	2
ρ_1	$(x, y) \mapsto \left(1 - \frac{1}{x}, \frac{y}{x^3}\right)$	3
ρ_2	$(x, y) \mapsto \left(\frac{1}{1-x}, \frac{y}{(1-x)^3}\right)$	3

On mime la construction effectuée lors de la démonstration du théorème 2.2 : à chaque involution τ_i non triviale de $\mathcal{C}$ on associe une paire de courbes elliptiques. Pour cela, on détermine un isomorphisme φ de $\mathcal{C}$ vers une courbe pour laquelle l'involution τ_i devient $(x, y) \mapsto (-x, y)$. On note $x_1 = 1$, x_2 et x_3 les valeurs prises par φ en 0, 1 et ∞. Les courbes elliptiques sont alors de la forme $y^2 = (x - 1)(x - x_2^2)(x - x_3^2)$. Leurs formes de Legendre sont $y^2 = x(x - 1)(x - \Lambda)$, où $\Lambda = (x_2^2 - x_3^2)/(1 - x_3^2)$ et leurs j-invariants s'en déduisent. Tout ceci est résumé dans le tableau suivant :

involution	isomorphisme	action sur les points de Weierstraß	x_2	x_3	Λ
τ_1	$x \mapsto \frac{(-1-u)x+\lambda}{(-1+u)x+\lambda}$	$(0, \lambda), (1, \nu), (\infty, \mu)$	$\frac{-1-u+\lambda}{-1+u+\lambda}$	$\frac{u+1}{u-1}$	$\Lambda_1 = \lambda(\lambda - 1 - u)^2$
τ_2	$x \mapsto \frac{(\lambda-1-u)x+1}{(\lambda-1+u)x+1}$	$(0, \mu), (1, \lambda), (\infty, \nu)$	$\frac{-1-u+\lambda}{-1+u+\lambda}$	$\frac{\lambda-u}{\lambda+u}$	$\Lambda_2 = \frac{1}{\lambda(\lambda-1+u)^2}$
τ_3	$x \mapsto \frac{x-\lambda+u}{x-\lambda-u}$	$(0, \nu), (1, \mu), (\infty, \lambda)$	$\frac{-1-u+\lambda}{-1+u+\lambda}$	$\frac{\lambda-u}{\lambda+u}$	$\Lambda_3 = \frac{1}{\lambda(\lambda-1+u)^2}$

On note Λ_i' le conjugué de Λ_i obtenu en remplaçant u par $-u$. Alors on a les égalités $\Lambda_2 = \Lambda_3 = 1/\Lambda_1'$, et les j-invariants associés sont les mêmes. Il n'y a donc que deux classes d'isomorphismes de courbes quotients.

Les formules donnant c_0 et c_1 se vérifient en exprimant les modules (j_1, j_2, j_3) en fonction de λ. □

Remarque. On traite dans l'annexe en 2.(c) le cas d'annulation des dénominateurs.

2.3.2 Isogénie entre les courbes elliptiques

Théorème 2.6 *Soit $\mathcal{C}$ une courbe dont les modules sont sur $\mathcal{D}$. Alors les deux courbes elliptiques quotients sont 3-isogènes l'une à l'autre.*

Démonstration. Reprenons les notations de la démonstration précédente. Soit $\mathcal{E}_1$ la courbe elliptique associée à l'involution τ_1 mise sous forme de Legendre $y^2 = x(x - 1)(x - \Lambda_1)$. Son polynôme de 3-division est alors

$$\psi_3(x) = 3x^4 + (-4\Lambda_1 - 4)x^3 + 6\Lambda_1 x^2 - \Lambda_1^2.$$

On vérifie que le polynôme linéaire suivant divise $\psi_3(x)$:

$$S_3(x) = 3x + \lambda - 2(u + 1).$$

Il lui correspond un sous-groupe de $\mathcal{E}_1$ d'ordre 3. Les formules de Vélu [Vél71] permettent alors de déterminer explicitement une courbe 3-isogène à $\mathcal{E}_1$ qui est donnée par

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

où a_2, a_4, a_6 sont définis par la suite de calculs suivants :

$$x_0 = \frac{2(u+1)-\lambda}{3}, \quad t = 6x_0^2 - 4(\Lambda_1+1)x_0 + 2\Lambda_1, \quad u = 4x_0^3 - 4(\Lambda_1+1)x_0^2 + 4\Lambda_1 x_0.$$

$$\begin{array}{rcl} a_2 & = & -(\Lambda_1+1), \\ a_4 & = & \Lambda_1 - 5t, \\ a_6 & = & 4(\Lambda_1+1)t - 7(u+x_0t). \end{array}$$

On vérifie que le j-invariant de cette courbe est le conjugué du j-invariant de $\mathcal{E}_1$. □

Corollaire 2.2 *Soit $\mathcal{C}$ une courbe de genre 2 admettant D_3 comme groupe réduit d'automorphismes. Alors l'anneau d'endomorphisme de la Jacobienne de $\mathcal{C}$ contient un ordre dans l'algèbre de quaternions $(\frac{3,1}{\mathbb{Q}})$. En particulier elle est à multiplication réelle par $\sqrt{3}$.*

Démonstration. Soit $\mathcal{C}$ une courbe de genre 2 admettant D_3 comme groupe d'automorphismes. Alors $\mathrm{Jac}(\mathcal{C})$ est isogène à $\mathcal{E}_1 \times \mathcal{E}_2$, où $\mathcal{E}_1$ et $\mathcal{E}_2$ sont deux courbes elliptiques 3-isogènes. Notons $\mathcal{I} : \mathcal{E}_1 \to \mathcal{E}_2$ une isogénie de degré 3, et $\hat{\mathcal{I}}$ son isogénie duale.

Soit $\mathcal{O}$ l'anneau

$$\mathcal{O} = \left\{ \begin{pmatrix} a & \sqrt{3}b \\ \sqrt{3}c & d \end{pmatrix}, \text{ où } a,b,c,d \in \mathbb{Z} \right\}.$$

Alors l'application qui à $\begin{pmatrix} a & \sqrt{3}b \\ \sqrt{3}c & d \end{pmatrix}$ associe l'endomorphisme de $\mathcal{E}_1 \times \mathcal{E}_2$

$$\begin{array}{rcl} \mathcal{E}_1 \times \mathcal{E}_2 & \to & \mathcal{E}_1 \times \mathcal{E}_2 \\ (P,Q) & \mapsto & ([a]P + [b]\hat{\mathcal{I}}Q, [c]\mathcal{I}P + [d]Q). \end{array}$$

est un homomorphisme injectif d'anneaux.

La multiplication par $\sqrt{3}$ est représentée par exemple par l'endomorphisme

$$(P,Q) \mapsto (\hat{\mathcal{I}}Q, \mathcal{I}P).$$

□

2.4 Groupe de Klein

Comme dans le cas précédent, on commence par rappeler un résultat dû à Igusa.

Théorème 2.7 *Soit $\mathcal{C}$ une courbe de genre 2. Le groupe réduit d'automorphismes de $\mathcal{C}$ est V_4 si et seulement si $\mathcal{C}$ est isomorphe à une courbe d'équation*

$$y^2 = x(x-1)(x+1)(x-\lambda)(x-1/\lambda), \tag{2.5}$$

avec λ différent de 0, de -1 et de 1.

En termes d'invariants d'Igusa, $\mathcal{C}$ admet pour groupe d'automorphismes V_4 si et seulement si ses modules appartiennent à la courbe $\mathcal{V}$ d'équations

$$\begin{aligned}
32j_1j_2^2 - 27j_1j_2 - 54j_2^2 + 4423680j_1j_3 + 14745600j_2j_3 - 13436928j_3 &= 0,\\
64j_2^3 - 78643200j_1j_2j_3 + 243j_1j_2 - 378j_2^2 + 31850496j_1j_3 - 8847360j_2j_3 &\\
-36238786560000j_3^2 + 120932352j_3 &= 0,\\
3j_1^2 - 10j_1j_2 + 18j_2 - 4608000j_3 &= 0.
\end{aligned}$$

Ce résultat s'obtient par un calcul d'élimination.

Théorème 2.8 *Soit $\mathcal{C}$ une courbe de genre 2 admettant V_4 comme groupe réduit d'automorphismes. Alors la Jacobienne de $\mathcal{C}$ est isogène à deux carrés de courbes elliptiques. Ces courbes elliptiques sont 2-isogènes l'une à l'autre. Leurs j-invariants sont les racines de $j^2 + c_1 j + c_0$, où c_0 et c_1 sont donnés par les formules suivantes :*

$$\begin{aligned}
c_1 &= \frac{9}{4}\,\frac{3j_1j_2 - 2j_2^2 + 1866240j_3 + 211200j_3j_1 + 64000j_3j_2}{j_3(-243 + 78j_1 + 20j_2)},\\
c_0 &= 108\,\frac{2560000j_3j_2 + 51j_1j_2 + 30j_2^2 + 768000j_3j_1 + 18662400j_3}{j_3(-243 + 78j_1 + 20j_2)}.
\end{aligned}$$

Démonstration. Quand la courbe $\mathcal{C}$ est sous la forme 2.5, les automorphismes réduits peuvent être explicités. Dans le tableau suivant, u désigne $\pm\sqrt{1-\lambda^2}$ et $\overline{u}$ désigne $\pm\sqrt{\lambda^2-1}$.

	transformation	ordre
Id	$(x,y) \mapsto (x,y)$	1
τ_1	$(x,y) \mapsto \left(\frac{x-\lambda}{\lambda x-1}, \frac{u^3 y}{(\lambda x-1)^3}\right)$	2
τ_2	$(x,y) \mapsto \left(\frac{\lambda x-1}{x-\lambda}, \frac{\overline{u}^3 y}{(x-\lambda)^3}\right)$	2
ρ	$(x,y) \mapsto \left(\frac{1}{x}, \frac{iy}{x^3}\right)$	4

On applique la même démarche que dans la démonstration du théorème 2.5. Cela nous mène au tableau suivant, qui nous donne les j-invariants des courbes quotients.

invol.	isomorphisme	action sur les points de Weierstraß	x_2	x_3	Λ	j
τ_1	$\varphi_1(x) = \frac{(1-u)x-\lambda}{(1+u)x-\lambda}$	$(0,\lambda),(1,-1),(\infty,\frac{1}{\lambda})$	$\frac{\lambda+u-1}{\lambda-u-1}$	$\frac{1-u}{1+u}$	$\Lambda_1 = \frac{\lambda^2(1-\lambda)}{(\lambda-1-u)^2}$	$J_1 = 64\frac{(4-l^2)^3}{l^4}$
τ_2	$\varphi_2(x) = \frac{(-\lambda-\bar{u})x+1}{(-\lambda+\bar{u})x+1}$	$(0,\frac{1}{\lambda}),(1,-1),(\infty,\lambda)$	$\frac{\lambda+\bar{u}-1}{\lambda-\bar{u}-1}$	$\frac{\lambda+\bar{u}}{\lambda-\bar{u}}$	$\Lambda_2 = \frac{\lambda-1}{\lambda(\lambda-1-\bar{u})^2}$	$J_2 = 64\frac{(4l^2-1)^3}{l^2}$

Notons Λ_i' les valeurs conjuguées des Λ_i obtenues en remplaçant u par $-u$ et $\bar{u}$ par $-\bar{u}$. Alors $\Lambda_1 + \Lambda_1' = 1$ et $\Lambda_2 + \Lambda_2' = 1$, ce qui explique que les j-invariants des courbes correspondantes sont inchangés. Ainsi la Jacobienne de $\mathcal{C}$ est (2,2)-isogène à des produits $\mathcal{E}_1 \times \mathcal{E}_1$ et $\mathcal{E}_2 \times \mathcal{E}_2$, donc aussi à $\mathcal{E}_1 \times \mathcal{E}_2$. Enfin, les courbes $\mathcal{E}_1$ et $\mathcal{E}_2$ sont 2-isogènes l'une à l'autre car le couple (J_1, J_2) annule l'équation modulaire de degré 2.

2.5 Exemples numériques

2.5.1 Cas générique

Soit la courbe $\mathcal{C}$ définie sur $\mathbb{Q}$ par l'équation

$$y^2 = x^6 - x^5 + x^4 - x^2 - x - 1.$$

Ses invariants d'Igusa sont

$$j_1 = \frac{2^3 \times 3^2 \times 5 \times 13}{37^2}, \; j_2 = -\frac{2^3 \times 3^3 \times 11 \times 13}{37^3}, \; j_3 = \frac{3^5 \times 53^2}{2^8 \times 37^5}.$$

On vérifie qu'ils sont dans l'ouvert $\mathcal{U}$, et donc $\mathrm{Jac}(\mathcal{C})$ est isogène au produit de deux courbes elliptiques. Sur cet exemple, trouver ces courbes en passant par une forme de Rosenhain nécessiterait de passer dans une extension de $\mathbb{Q}$ de degré 24. Les propositions 2 et 3 permettent de calculer directement

$$c_0 = \frac{2^{14} \times 5^6 \times 37^3}{53^2}, \; c_1 = \frac{2^8 \times 3^4 \times 47}{53},$$

et les j-invariants des courbes elliptiques sont définis sur le corps $\mathbb{Q}(i)$ par

$$j = -\frac{2^7 \times 3^4 \times 47}{53} \pm \frac{2^8 \times 7 \times 11 \times 181}{53} i.$$

Notons que 53 est un facteur du discriminant de la courbe. Il n'est donc pas surprenant de le retrouver au dénominateur des valeurs de j et dans l'expression de j_3.

2.5.2 Courbe $\mathcal{D}$

L'exemple suivant est tiré de l'article [Kul95], où Kulesz construit une courbe ayant de nombreux points rationnels. Soit la courbe $\mathcal{C}$ définie sur $\mathbb{Q}$ par l'équation

$$y^2 = 1412964(x^2 - x + 1)^3 - 8033507x^2(x-1)^2.$$

Ses invariants d'Igusa sont

$$\begin{aligned} j_1 &= \frac{3^2 \times 149 \times 167 \times 239^2 \times 3618470803 \times 33613^2}{757^2 \times 76832154757^2}, \\ j_2 &= -\frac{3^3 \times 239^2 \times 33613^2 \times 195593 \times 31422316507485410373257}{757^3 \times 76832154757^3}, \\ j_3 &= -\frac{2^{22} \times 3^{1}7 \times 5^9 \times 7^6 \times 47^3 \times 89^3 \times 239^4 \times 33613^4}{757^5 \times 76832154757^5}. \end{aligned}$$

On vérifie qu'ils sont $\mathcal{D}$, et donc le groupe d'automorphismes réduit de la courbe est D_3. En fait la manière dont cette courbe est construite dans [Kul95] implique directement ce résultat. Là encore, écrire une forme de Rosenhain pour cette courbe nécessiterait de passer dans une extension de $\mathbb{Q}$. Nos formules donnent directement les j-invariants des courbes elliptiques quotients :

$$j = -\frac{239 \times 33613 \times 84333563^3}{2^{24} \times 3^4 \times 5^9 \times 7^2 \times 47^3 \times 89}, \quad \text{et} \quad j' = \frac{19^3 \times 67^3 \times 239 \times 349^3 \times 33613}{2^8 \times 3^{12} \times 5^3 \times 7^6 \times 47 \times 89^3}.$$

2.5.3 Courbe $\mathcal{V}$

Dans [LM97], Leprévost et Morain ont étudié la courbe $\mathcal{C}_\theta$ définie sur $\mathbb{Q}(\theta)$ par

$$y^2 = x(x^4 - \theta x^2 + 1),$$

avec comme objectif des calculs de sommes de caractères. Ses invariants d'Igusa sont

$$j_1 = 144\frac{9\theta^2 - 20}{(3\theta^2+20)^2}, \; j_2 = -3456\frac{27\theta^2 - 140}{(3\theta^2+20)^3}, \; j_3 = 243\frac{\theta^2 - 4}{(3\theta^2+20)^5}.$$

On vérifie qu'ils annulent les polynômes définissant $\mathcal{V}$, et donc le groupe d'automorphismes réduit de la courbe est V_4. On trouve alors les j-invariants des courbes elliptiques quotients :

$$j = 64\frac{(3\theta - 10)^3}{(\theta - 2)(\theta + 2)^2} \quad \text{et} \quad j' = 64\frac{(3\theta + 10)^3}{(\theta + 2)(\theta - 2)^2}.$$

Notons que les courbes E_θ et E'_θ données dans [LM97] :

$$y^2 = x(x^2 \pm 4x + 2 - \theta),$$

ont toutes les deux le même j-invariants j'. Les autres courbes quotients, d'invariant j sont les courbes d'équations :

$$y^2 = x(x^2 \pm 4x + 2 + \theta).$$

2.6 La courbe $X_1(13)$

Lors d'une recherche exhaustive de courbes sur $\mathbb{Q}$ de petite hauteur dont la Jacobienne a un gros groupe de torsion, nous avons trouvé que la courbe $\mathcal{C}$ d'équation

$$y^2 = x^6 - 2x^5 + x^4 - 2x^3 + 6x^2 - 4x + 1$$

a une Jacobienne dont le nombre de points est divisible par 19 lorsqu'on la réduit modulo les 15 plus petits nombres premiers de bonne réduction. Ceci est un bon indice pour que cette Jacobienne ait de la 19-torsion rationnelle.

Après quelques calculs, on vérifie que le diviseur $D_{19} = (1,1) - (1,-1)$ est d'ordre 19.

En fait, cette courbe n'est autre que la courbe modulaire $X_1(13)$, dont on trouve par exemple l'équation dans [Rei86].

Après avoir calculé ses invariants,

$$j_1 = 7137, \quad j_2 = 218673, \quad j_3 = -\frac{41067}{4},$$

on constate que la courbe est dans $\mathcal{H}_2$ et que son groupe d'automorphismes est D_6. Nos formules fournissent alors les j-invariants des courbes elliptiques quotients : ils sont racines de

$$j^2 + 61763j - 74559407,$$

dont le discriminant est 13×17787^2. Soit donc K le corps de nombres $\mathbb{Q}(\sqrt{13})$. Les courbes elliptiques sont donc définies sur K et ont pour j-invariant

$$j = -\frac{61763 \pm 17787\sqrt{13}}{2}.$$

Toutefois, cela ne signifie pas que l'on peut transporter le point de 19-torsion sur les courbes elliptiques. En effet, le morphisme de $\mathcal{C}$ vers ces courbes n'est a priori pas défini sur le même corps que les courbes elles-mêmes.

2.7 Annexe : formulaire

Nous complétons l'étude précédente par les formules traitant des cas suivants :

– Le groupe réduit d'automorphismes $\mathcal{G}$ n'est ni $\mathcal{D}$, ni $\mathcal{V}$, ni $\mathbb{Z}/2\mathbb{Z}$: ce sont les deux points traités en 2.(a) et 2.(b) ci-dessous.

– Un dénominateur s'annule. Sur la courbe $\mathcal{D}$ cela se produit en un point traité en 2.(c), dans le cas générique, deux courbes doivent être étudiées, en 2.(f) et 2.(g).

– Le covariant A est nul, et les invariants choisis ne sont plus adaptés. On en choisit deux autres, et on procède à la même étude exhaustive.

Nous regroupons ces formules sous forme d'un algorithme prenant en entrée une courbe quelconque de genre 2, dont la Jacobienne est (2,2)-décomposable, et qui retourne le polynôme minimal des j-invariants des courbes elliptiques quotients de la Jacobienne.

1. Calculer les covariants A, B, C, D de la courbe, et vérifier que $R = 0$.
2. Cas où A est non nul : calculer j_1, j_2, j_3.
   (a) Si $(j_1, j_2, j_3) = (\frac{81}{20}, -\frac{729}{200}, \frac{729}{25600000})$, alors le groupe d'automorphismes est D_6 et retourner $j(j - 54000)$.
   (b) Si $(j_1, j_2, j_3) = (-\frac{36}{5}, \frac{1512}{25}, \frac{243}{200000})$, alors le groupe d'automorphismes est $\mathfrak{S}_4$, et retourner $j - 8000$.
   (c) Si $(j_1, j_2, j_3) = (\frac{24297228}{885481}, -\frac{81449284536}{833237621}, -\frac{57798021931029}{47220229240364864})$, alors le groupe d'automorphismes est D_3 et retourner $j^2 + \frac{471690263168}{658503} j - \frac{8094076887461888}{57289761}$.
   (d) Si (j_1, j_2, j_3) annulent les polynômes définissant $\mathcal{D}$, alors le groupe d'automorphismes est D_3, et retourner les j calculés en section 2.3.
   (e) Si (j_1, j_2, j_3) annulent les polynômes définissant $\mathcal{V}$, alors le groupe d'automorphismes est V_4, et retourner les j calculés en section 2.4.
   (f) Si (j_1, j_2, j_3) vérifient
   $$\begin{aligned} 331776 j_3 - j_2^2 - 24 j_1 j_2 - 144 j_1^2 &= 0, \\ 9 j_1 + j_2 &= 0, \end{aligned}$$
   alors le groupe d'automorphismes est $\mathbb{Z}/2\mathbb{Z}$ et retourner
   $$j^2 + \frac{150994944 j_3}{j_2 + 12 j_1} j - \frac{260919263232 j_3}{j_2 + 12 j_1}.$$
   (g) Si (j_1, j_2, j_3) vérifient
   $$\begin{aligned} j_2^5 + 54 j_2^4 - 322486272 j_2^2 j_3 + 481469424205824 j_3^2 &= 0, \\ 18 j_1 + 5 j_2 &= 0, \end{aligned}$$
   alors le groupe d'automorphismes est $\mathbb{Z}/2\mathbb{Z}$ et retourner $j^2 + c_1 j + c_0$, où
   $$\begin{aligned} c_0 &= -\frac{125}{9559130112} \frac{(-j_2^2 - 24 j_1 j_2 - 144 j_1^2 + 16257024 j_3)^2}{j_3^2} \\ c_1 &= \frac{(-j_2^2 - 24 j_1 j_2 - 144 j_1^2 + 16257024 j_3)(2723051520 j_3 - 289 j_2^2 - 6936 j_1 j_2 - 41616 j_1^2)}{2064772104192 j_3^2}, \end{aligned}$$

(h) Sinon on est dans le cas « générique », et retourner les j calculés en section 2.2

3. Cas où $A = 0$

   (a) Si $B = 0$ et $C^5 = 4050000D^3$, alors le groupe d'automorphismes est $\mathbb{Z}/2\mathbb{Z}$, retourner $(j - 4800)(j - 8640)$.

   (b) Si $C = 0$ et $B^5 = 3037500D^2$, alors le groupe d'automorphismes est $\mathbb{Z}/2\mathbb{Z}$, retourner $(j - 160)(j + 21600)$.

   Calculer les invariants

$$t_1 = \frac{3}{512}\frac{CD}{B^4} \quad \text{et} \quad t_2 = 1536\frac{BC}{D}.$$

   (c) Si $(t_1, t_2) = (1/576000, -460800)$, alors le groupe d'automorphismes est V_4, et retourner $j^2 + 7200j + 13824000$.

   (d) Si $(t_1, t_2) = (-1/864000, -172800)$, alors le groupe d'automorphismes est D_3, et retourner $j^2 + 55200j - 69984000$.

   (e) On est dans le cas « générique » pour $A = 0$. Calculer

$$\begin{aligned} \Omega &= -4\frac{-238878720000t_1 + 1555200t_2t_1 + 7t_2^2t_1 + 2t_2}{477757440000t_1 + 2073600t_2t_1 + t_2^2t_1 - t_2}, \\ \Upsilon &= \frac{3}{2}\Omega. \end{aligned}$$

   puis c_0 et c_1 par les formules de la section 2.2. Retourner $j^2 + c_1j + c_0$.

Chapitre 3

Formes modulaires de Siegel : vers des équations modulaires en genre 2

Dans ce chapitre nous nous proposons de décrire une approche pour construire des équations liant les invariants d'Igusa de courbes dont les Jacobiennes sont isogènes. La stratégie est de mimer ce que l'on sait faire pour les courbes elliptiques. Autant que faire ce peut, nous traiterons le cas des courbes de genre quelconque, puis nous nous restreindrons au genre deux, seul cas pour lequel la théorie des invariants est complète.

Nous nous référerons la plupart du temps à l'ouvrage de Freitag [Fre83]. Voir aussi le livre de Klingen [Kli90] pour une référence en langue anglaise.

3.1 Formes modulaires de Siegel

La définition des formes modulaires de Siegel est très proche de la définition des formes modulaires classiques, de même les propriétés de structure se transcrivent bien. Toutefois le passage à la dimension supérieure ne se fait pas aussi bien pour ce qui concerne les développements en séries. En effet le développement en série de Fourier ne donne pas immédiatement une série entière ou une série de Laurent, et de toute façon il y a un écueil théorique profond : le corps des fractions des séries entières à plusieurs variables ne correspond pas aux séries de Laurent.

3.1.1 Formes et fonctions modulaires

On suppose que la dimension g est fixée, et l'on cherche à étudier les fonctions de $\mathbb{H}_g$ dans $\mathbb{C}$ qui sont invariantes sous l'action de $\mathrm{Sp}_{2g}(\mathbb{Z})$. Pour cela on introduit les formes modulaires qui, faute d'être invariantes sont des fonctions dont on contrôle le comportement. Par analogie avec le cas classique de la dimension 1, on notera τ l'élément courant de l'espace $\mathbb{H}_g$; il ne faut pas perdre de vue qu'il s'agit d'une matrice.

Définition 3.1 *Une* forme modulaire *de poids r est une fonction $f : \mathbb{H}_g \to \mathbb{C}$ qui vérifie :*

1. *La fonction f est holomorphe sur $\mathbb{H}_g$.*
2. *Pour toute matrice $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathrm{Sp}_{2g}(\mathbb{Z})$ et tout $\tau \in \mathbb{H}_g$, on a*

$$f(M\tau) = \det(C\tau + D)^r f(\tau).$$

3. *Pour tout $Y_0 > 0$, la fonction f est bornée sur le domaine $Y \geqslant Y_0$.*

Le troisième point est une généralisation de la condition « holomorphe aux pointes » que l'on a dans le cas elliptique. Lorsque $g > 1$, cette condition n'est pas nécessaire dans la définition car elle découle des deux premiers points ; c'est le principe de Koecher (cf Freitag [Fre83, p. 44]).

Donnons quelques éléments sur la structure de l'ensemble des formes modulaires.

Proposition 3.1 *Pour tout entier r, l'ensemble des formes modulaires de poids r est un espace vectoriel sur $\mathbb{C}$.*

L'ensemble des formes modulaires est un anneau gradué : si f et f' sont des formes modulaires de poids respectifs r et r', alors le produit ff' est une forme modulaire de poids $r + r'$.

On a de plus un résultat de finitude pour la dimension des formes modulaires de poids fixé en tant qu'espace vectoriel.

Théorème 3.1 *Il existe une constante A_g ne dépendant que de g telle que la dimension de l'espace vectoriel des formes modulaires de poids r soit inférieure à $A_g r^{\frac{g(g+1)}{2}}$.*

Ce théorème est à l'origine du théorème de dépendance algébrique qui suit un peu plus bas. On en trouvera une démonstration dans [Fre83, p. 52].

Définition 3.2 *Une* fonction modulaire *est une fonction méromorphe $f : \mathbb{H}_g \to \mathbb{C}$ qui s'écrit comme quotient de deux formes modulaires de même poids.*

On peut se demander si cette définition est bien ce que l'on veut. En effet on pourrait aussi décider que les fonctions modulaires sont les fonctions méromorphes invariantes sous l'action de $\mathrm{Sp}_{2g}(\mathbb{Z})$. Le théorème suivant montre que ces deux définitions se rejoignent.

Théorème 3.2 *Toute fonction modulaire est une fonction méromorphe de $\mathbb{H}_g$ invariante sous l'action de $\mathrm{Sp}_{2g}(\mathbb{Z})$. Réciproquement toute fonction méromorphe de $\mathbb{H}_g$ invariante sous l'action de $\mathrm{Sp}_{2g}(\mathbb{Z})$ peut s'écrire comme quotient de deux formes modulaires de même poids, c'est donc une fonction modulaire.*

On pourra trouver une démonstration de ce théorème dans Siegel [Sie73].

L'ensemble des fonctions modulaires est un corps contenant $\mathbb{C}$. Dans le cas elliptique, un résultat classique est que ce corps est l'ensemble des fractions rationnelles en la fonction j (invariant modulaire).

On a un résultat analogue pour la dimension supérieure.

Théorème 3.3 *Le degré de transcendance du corps des fonctions modulaires de dimension g par rapport à $\mathbb{C}$ est $\frac{g(g+1)}{2}$. Ainsi toute famille de $\frac{g(g+1)}{2} + 1$ fonctions modulaires est algébriquement dépendante.*

On ne peut pas donner de théorème général plus précis car il peut arriver que le corps des fonctions modulaires ne soit pas un corps de fonctions rationnelles. Freitag [Fre83, p. 163], cite le cas où $g \equiv 0 \mod 24$, comme exemple de ce phénomène.

Toutefois, pour le cas du genre 2, nous verrons plus loin que les fonctions modulaires forment bien un corps de fonctions rationnelles.

3.1.2 Formes modulaires classiques

Les théorèmes précédents ne sont pas constructifs. Ce paragraphe fournit quelques exemples de formes modulaires, les Theta-constantes et les séries d'Eisenstein, afin d'illustrer ce qui précède. Dans la suite, les séries d'Eisenstein seront étudiées plus en profondeur.

Theta-Constantes

Grâce aux fonctions Theta, on peut fabriquer des formes modulaires de Siegel. Cette construction est faite dans Freitag [Fre83] ; pour plus de détails sur les vastes propriétés de ces fonctions, on consultera Mumford [Mum83] [Mum84] [Mum91].

Définition 3.3 *Pour tout couple de vecteurs entiers $a, b \in \mathbb{Z}^g$, on définit la fonction* Theta-constante *de caractéristique (a, b) (ou Thetanullwert en allemand) par*

$$\theta_{a,b}(\tau) = \sum_{k \in \mathbb{Z}^g} e^{i\pi\left({}^t(k+\frac{1}{2}a)\tau(k+\frac{1}{2}a)+{}^tbk\right)},$$

où τ est une matrice de $\mathbb{H}_g$.

On voit facilement que $\theta_{a,\tilde{b}}(\tau) = \theta_{a,b}(\tau)$ dès que $b \equiv \tilde{b} \mod 2$, et que

$$\theta_{a+2\tilde{a},b}(\tau) = (-1)^{{}^tab}\,\theta_{a,b}(\tau).$$

Comme par la suite, on s'intéressera uniquement aux carrés de Theta-constantes, on se ramène au cas où les Theta-caractéristiques (a, b) sont définies modulo 2.

Définition 3.4 *Une Theta-caractéristique (a, b) sera dite* paire *si ${}^tab \equiv 0 \mod 2$, et* impaire *sinon.*

Théorème 3.4 *Si la Theta-caractéristique (a, b) est impaire, alors la Theta-constante associée $\theta_{a,b}(\tau)$ est identiquement nulle.*

Théorème 3.5 *Soit*

$$k_g = \begin{cases} 8 & si\ g = 1, \\ 2 & si\ g = 2, \\ 1 & si\ g \geqslant 3. \end{cases}$$

La fonction

$$\Delta_g(\tau) = \prod_{(a,b)\ paire} \theta_{a,b}(\tau)^{k_g}$$

est une forme modulaire non nulle de poids

$$\begin{array}{rl} 12 & si\ g = 1, \\ 10 & si\ g = 2, \\ (2^g+1)2^{g-2} & si\ g \geqslant 3. \end{array}$$

Séries d'Eisenstein

Les séries d'Eisenstein sont une généralisation de celles existant en dimension 1. Elles nous permettront plus tard de décrire toutes les formes modulaires en dimension 2.

Définition 3.5 *Pour tout entier r pair, et supérieur à $g+1$, on définit la série d'Eisenstein E_r par*

$$E_r(\tau) = \sum_{\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \Gamma_0 \backslash \mathrm{Sp}_{2g}(\mathbb{Z})} \det (C\tau + D)^{-r},$$

où le sous-groupe Γ_0 est défini par

$$\Gamma_0 = \left\{ \begin{pmatrix} A & B \\ 0 & D \end{pmatrix} \in \mathrm{Sp}_{2g}(\mathbb{Z}) \right\}.$$

Proposition 3.2 *La série d'Eisenstein E_r est une forme modulaire de poids r.*

Pour une preuve de ce résultat, on consultera [Fre83].

3.1.3 Développement en série de Fourier

Une forme modulaire est périodique en un certain sens, il est donc naturel de s'intéresser au développement de Fourier découlant de cette périodicité. Même si celui-ci n'est pas aussi simple qu'en dimension 1, il se révèle un outil essentiel pour l'analyse des formes modulaires.

Pour ce paragraphe, on pourra trouver les démonstrations complètes dans Maaß [Maa71].

Proposition 3.3 *Toute forme modulaire f admet un développement de Fourier de la forme*

$$f(\tau) = \sum_T a(T) e^{2i\pi\sigma(T\tau)},$$

où la somme est prise sur toutes les matrices T symétriques semi-entières (i.e. les éléments diagonaux sont entiers, et les autres sont dans $\frac{1}{2}\mathbb{Z}$). Le symbole σ désigne la trace (Spur en allemand).

En notant $\tau = X + iY$, et x_{kl} et y_{kl} les termes de X et Y en ligne k et colonne l, le coefficient $a(T)$ s'exprime sous la forme

$$a(T) = e^{2\pi\sigma(TY)} \int_{-\frac{1}{2} \leqslant x_{kl} \leqslant \frac{1}{2}} f(X+iY) e^{2i\pi\sigma(TX)} dX.$$

Cette proposition regroupe en fait plusieurs résultats. Le point de départ est le fait suivant : une forme modulaire f vérifie $f(\tau + S) = f(\tau)$ pour toute matrice entière S. Pour appliquer la théorie générale de Fourier, on décompose alors τ en $X + iY$; et pour Y fixé on a :

$$f(X+iY) = \sum_\chi a(\chi, Y)\chi(X),$$

où la somme est prise sur l'ensemble des caractères du groupe des matrices symétriques complexes quotienté par le sous-groupe des matrices symétriques entières. On vérifie facilement que ces caractères sont les applications $X \mapsto \sigma(TX)$, où T est une matrice symétrique semi-entière. Si l'on oublie les problèmes de convergence, on comprend l'allure de la formule de la proposition. Il reste toutefois à élucider la non-dépendance de $a(T,Y)$ en Y. Ceci résulte de l'holomorphie de f que l'on peut exprimer par

$$\frac{1}{2}\left(\frac{\partial}{\partial x_{kl}} + i\frac{\partial}{\partial y_{kl}}\right) f = 0,$$

pour k et l décrivant les numéros de ligne et colonne des éléments des matrices X et Y. Ces équations fournissent des équations différentielles du premier ordre en $a(T,Y)$ que l'on résout en

$$a(T,Y) = a(T)e^{-2\pi\sigma(TY)}.$$

En regroupant les morceaux on tombe bien sur la formule annoncée.

Proposition 3.4 *Soit f une forme modulaire ayant un développement de Fourier de la forme*

$$f(\tau) = \sum_T a(T)e^{2i\pi\sigma(T\tau)}.$$

Alors si $a(T) \neq 0$, la matrice T est nécessairement semi-définie positive (i.e. ses valeurs propres sont positives ou nulles).

Ce résultat qui permet de se rapprocher un peu du développement en série des formes modulaires en dimension 1 conduit de plus à généraliser la notion de forme parabolique.

Définition 3.6 *Une forme modulaire f est dite* forme parabolique *si les seuls coefficients de Fourier non nuls apparaissent quand T est définie positive.*

3.1.4 L'opérateur Φ de Siegel

L'opérateur Φ de Siegel permet de transformer une forme modulaire de degré g (i.e agissant sur $\mathbb{H}_g$) en une forme modulaire de degré $g-1$, tout en conservant le poids. Son intérêt est qu'il se comporte bien vis-à-vis des coefficients de Fourier, et que de plus il envoie les séries d'Eisenstein sur d'autres séries d'Eisenstein.

Définition 3.7 *Soit $g \geqslant 2$, et soit $f : \mathbb{H}_g \to \mathbb{C}$ une fonction telle que pour tout $\tau \in \mathbb{H}_{g-1}$, la limite suivante existe*

$$\lim_{t\to+\infty} f\begin{pmatrix} \tau & 0 \\ 0 & it \end{pmatrix}.$$

On définit alors une fonction $\Phi.f$ sur $\mathbb{H}_{g-1}$ par

$$\forall \tau \in \mathbb{H}_{g-1}, \quad \Phi.f(\tau) = \lim_{t\to+\infty} f\begin{pmatrix} \tau & 0 \\ 0 & it \end{pmatrix}.$$

Théorème 3.6 *L'opérateur Φ définit une application linéaire des formes modulaires de degré g de poids r vers les formes modulaires de degré $g-1$ de poids r.*

Théorème 3.7 *Si une forme modulaire f admet un développement de Fourier*

$$f(\tau) = \sum_T a(T)e^{2i\pi\sigma(T\tau)},$$

où les matrices T sont de taille g, alors la forme modulaire $\Phi.f$ admet le développement de Fourier suivant :

$$\Phi.f(\tau) = \sum_{T'} a\begin{pmatrix} T' & 0 \\ 0 & 0 \end{pmatrix} e^{2i\pi\sigma(T'\tau)},$$

où cette fois-ci les matrices T' sont de taille $g-1$, et $\tau \in \mathbb{H}_{g-1}$.

On peut alors remarquer que les formes paraboliques sont précisément les formes modulaires qui sont dans le noyau de Φ.

Proposition 3.5 *Pour tout g, on note $E_r^{(g)}$ la série d'Eisenstein de degré g, de poids r (pour $g = 1$, on considère les séries d'Eisenstein normalisées). Alors on a*

$$\Phi.E_r^{(g)} = E_r^{(g-1)}.$$

3.2 Application au genre 2

3.2.1 Structure de l'anneau gradué des formes modulaires

Dans le cas du genre 2, la structure de l'anneau gradué des formes modulaires peut être entièrement décrite grâce aux séries d'Eisenstein. C'est pourquoi dans la suite on étudiera plus particulièrement ces séries.

Le théorème suivant, dû à Igusa [Igu62] décrit toutes les formes modulaires en genre 2.

Théorème 3.8 *En genre 2, l'anneau gradué des formes modulaires est engendré sur $\mathbb{C}$ par les séries d'Eisenstein φ_4, φ_6, φ_{10}, φ_{12} (qui sont algébriquement indépendantes).*

Ce résultat permet d'écrire toute forme modulaire de poids r comme un polynôme homogène en $(\varphi_4,\ \varphi_6,\ \varphi_{10},\ \varphi_{12})$ affectés des poids $(4,\ 6,\ 10,\ 12)$. On dispose donc d'un moyen de calculer la dimension de l'espace vectoriel des formes de poids donné.

Proposition 3.6 *La dimension N_r de l'espace vectoriel complexe des formes modulaires de poids r est égal au nombre de solutions entières positives ou nulles de l'équation*

$$r = 4p + 6q + 10s + 12t.$$

Igusa donne en appendice de son article une formule close explicite pour N_r. Celle-ci est obtenue en passant par des résidus de séries génératrices. La formule obtenue n'est pas très simple.

Le théorème de structure des formes modulaires nous permet de déduire une forme explicite pour le corps des fonctions modulaires similaire au cas elliptique.

Théorème 3.9 *Considérons les fonctions modulaires suivantes*

$$g_1 = \varphi_4\varphi_6\varphi_{10}^{-1},\ g_2 = \varphi_4^3\varphi_{12}^{-1},\ g_3 = \varphi_6^2\varphi_{12}^{-1}.$$

Alors le corps des fonctions modulaires est $\mathbb{C}(g_1, g_2, g_3)$.

En effet tout monôme $\varphi_4^p\varphi_6^q\varphi_{10}^s\varphi_{12}^t$ avec $4p + 6q + 10s + 12t = 0$ peut s'exprimer par un monôme de la forme $g_1^l g_2^m g_3^n$, avec $l, n, m \in \mathbb{Z}$. Comme les fonctions modulaires sont des quotients de polynômes homogènes de même degré, elles peuvent ensuite s'exprimer comme fraction rationnelle en les g_1, g_2, g_3.

3.2.2 Changement de variables de Siegel

Afin de mettre le développement en série de Fourier d'une forme modulaire sous une forme plus agréable, Siegel [Sie55] a proposé un changement de variables. Celui-ci existe en toute dimension ; en dimension 1, cela revient à poser classiquement $q = e^{2i\pi\tau}$. Explicitons le changement de variables en dimension 2. Pour un élément $\tau = \begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix} \in \mathbb{H}_2$, on note

$$w_1 = z_1 - 2z_2,\ w_2 = z_2,\ w_3 = z_3 - z_1 \text{ et } q_k = e^{2i\pi w_k}.$$

Alors pour toute matrice $T = \begin{pmatrix} t_1 & t_2 \\ t_2 & t_3 \end{pmatrix}$ symétrique entière semi-positive, on a

$$e^{2i\pi\sigma(T\tau)} = q_1^{t_1+t_3} q_2^{2(t_1+t_3)+2t_2} q_3^{t_3}.$$

Le fait que T soit symétrique entière semi-positive nous assure que les puissances des q_i

$$u_1 = t_1 + t_3,\ u_2 = 2(t_1+t_3) + 2t_2,\ u_3 = t_3,$$

sont des entiers positifs ou nuls et qu'ils vérifient de plus

$$u_1 \geqslant u_3 \quad \text{et} \quad 2\left(u_1 - \sqrt{u_3(u_1-u_3)}\right) \leqslant u_2 \leqslant 2\left(u_1 + \sqrt{u_3(u_1-u_3)}\right).$$

Donnons de plus les formules de passage dans le sens inverse :

$$t_1 = u_1 - u_3,\ t_2 = \frac{u_2}{2} - u_1,\ t_3 = u_3.$$

On résume tout cela dans la proposition suivante.

Proposition 3.7 *Toute forme modulaire f admet un développement de la forme :*

$$f(\tau) = \sum_{u_3 \geqslant 0} \sum_{u_1 \geqslant u_3} \sum_{2\left(u_1 - \sqrt{u_3(u_1-u_3)}\right) \leqslant u_2 \leqslant 2\left(u_1 + \sqrt{u_3(u_1-u_3)}\right)} a(u_1,u_2,u_3) q_1^{u_1} q_2^{u_2} q_3^{u_3}.$$

L'avantage de cette écriture est que l'on exprime une forme modulaire comme une série entière en les trois variables q_1, q_2, q_3. De plus, on vérifie facilement que si l'élément τ sur lequel agit la forme modulaire est dans le domaine fondamental décrit en section 1.6.3, alors

$$|q_1|,\ |q_2|,\ |q_3| \leqslant 1.$$

Cas des formes paraboliques :

Si de plus f est une forme parabolique, alors certains des coefficients du développement sont nuls. En particulier si l'une des conditions suivantes se produit, le coefficient $a(u_1, u_2, u_3)$ est nul :

1. $u_3 = 0$, car alors $t_3 = 0$,

2. $u_1 = u_3$, car alors $t_1 = 0$,

3. u_2 atteint ses bornes, car alors il en est de même pour t_2.

3.2.3 Coefficients de Fourier des séries d'Eisenstein

Méthode de calcul

L'article de Maaß [Maa78] donne des relations linéaires entre les coefficients de Fourier des séries d'Eisenstein de degré 2 de poids inférieur ou égal à 12 (ce qui est suffisant grâce au théorème de structure). Ces relations fournissent directement un algorithme récursif pour calculer ces coefficients.

Avant de donner ces relations, remarquons que l'on connaît déjà ces coefficients de Fourier dans le cas où la matrice T est de déterminant nul. En effet l'opérateur Φ de Siegel permet alors de se ramener à la dimension 1. On obtient

$$a\begin{pmatrix} t_1 & t_2 \\ t_2 & t_3 \end{pmatrix} = \frac{-2k}{B_k}\sigma_{k-1}(\mathrm{pgcd}(t_1,t_3)),$$

où B_k est le $k^{\text{ème}}$ nombre de Bernoulli et $\sigma_k(h)$ est la somme des puissances $k^{\text{ème}}$ des diviseurs de h. Rappelons que les nombres de Bernoulli sont définis par le développement en série entière suivant :

$$\frac{z}{e^z-1} = \sum_k \frac{B_k}{k!}z^k.$$

Dans la suite, on s'intéresse donc au calcul des coefficients pour les matrices T de rang 2.

Théorème 3.10 *Soit φ_k la série d'Eisenstein de degré 2, de poids k, et soit $a(T)$ son coefficient de Fourier en la matrice symétrique semi-positive T. On définit les rationnels*

$$A_0(\varphi_k,h) = a_k(1)\left(a_k(h)+b_k(h)\right) \quad \textit{et} \quad A_2(\varphi_k,h) = \frac{a_k(1)}{k}\left(h\left(a_k(h)+b_k(h)\right)+d_k(h)\right),$$

où

$$a_k(h) = \frac{-2k}{B_k}\sigma_{k-1}(h),$$

$$b_k(h) = 2^{10}\cdot 3^5\cdot 5^2\cdot 7^2\cdot 131^{-1}\cdot 593^{-1}\cdot 691^{-1}\cdot\tau(h), \quad \textit{si} \quad k=12, \quad \textit{et} \quad b_k(h)=0 \quad \textit{sinon},$$

$$d_k(h) = 2^{10}\cdot 3^5\cdot 19\cdot 43867^{-1}\cdot\tau(h), \quad \textit{si} \quad k=10, \quad \textit{et} \quad d_k(h)=0 \quad \textit{sinon}.$$

Les notations B_k désignent les nombres de Bernoulli, $\sigma_k(h)$ la somme des puissances $k^{\text{ème}}$ des diviseurs de h, et $\tau(h)$ la fonction de Ramanujan définie par

$$q\prod_{n\geqslant 1}(1-q^n)^{24} = \sum_{h\geqslant 1}\tau(h)q^h.$$

On a alors les relations suivantes entre certaines valeurs de $a(T)$ et les scalaires $A_0(\varphi_k,h)$ et $A_2(\varphi_k,h)$:

$$A_0(\varphi_k,h) = a\begin{pmatrix} 1 & 0 \\ 0 & h \end{pmatrix} + 2\sum_{0<t\leqslant\sqrt{h}} a\begin{pmatrix} 1 & 0 \\ 0 & h-t^2 \end{pmatrix} + 2\sum_{0\leqslant t\leqslant\frac{-1+\sqrt{1+4h}}{2}} a\begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & h-t(t+1) \end{pmatrix},$$

$$A_2(\varphi_k,h) = \sum_{0<t\leqslant\sqrt{h}} a\begin{pmatrix} 1 & 0 \\ 0 & h-t^2 \end{pmatrix}(2t)^2 + \sum_{0\leqslant t\leqslant\frac{-1+\sqrt{1+4h}}{2}} a\begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & h-t(t+1) \end{pmatrix}(2t+1)^2.$$

Les valeurs de $A_0(\varphi_k, h)$ et $A_2(\varphi_k, h)$ sont facilement calculables, on peut donc déjà calculer successivement tous les coefficients

$$a\begin{pmatrix} 1 & 0 \\ 0 & h \end{pmatrix} \quad \text{et} \quad a\begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & h \end{pmatrix}$$

avec $h \geqslant 0$ pour les séries d'Eisenstein. À partir de là, on peut obtenir tous les autres coefficients grâce aux théorèmes suivants :

Théorème 3.11 *Soit $a(T)$ le coefficient du développement en série de Fourier d'une forme modulaire de poids k pair. Alors pour toute matrice U unimodulaire*

$$a({}^tUTU) = a(T).$$

En d'autres termes, si $T = \begin{pmatrix} t_1 & t_2 \\ t_2 & t_3 \end{pmatrix}$, $a(T)$ est une fonction de la classe d'équivalence unimodulaire de T, et ne dépend que de $4|t_1t_3 - t_2^2|$ et de $e(T) = \mathrm{pgcd}(t_1, 2t_2, t_3)$.

On peut désormais calculer les coefficients $a(T)$ pour $e(T) = 1$. Le théorème suivant permet de construire les éléments manquants.

Théorème 3.12 *Les coefficients de Fourier d'une série d'Eisenstein vérifient :*

$$\text{si } T = \begin{pmatrix} t_1 & t_2 \\ t_2 & t_3 \end{pmatrix},\ a_k(T) = \sum_{d|e(T),\ d>0} d^{k-1} a_k \begin{pmatrix} 1 & \frac{t_2}{d} \\ \frac{t_2}{d} & \frac{t_1t_3}{d^2} \end{pmatrix}.$$

Début du développement en série entière

Pour les séries d'Eisenstein de poids inférieur ou égal à 12, on donne le début du développement en série après avoir effectué le changement de variables. On tronque le développement au degré total inférieur à 10.

$$\begin{aligned}
\varphi_4(\tau) \quad = \quad & 1 + 240q_1q_2^2 + 240q_1q_2^2q_3 + 240q_1^2q_2^2q_3 + 13440q_1^2q_2^3q_3 + 2160q_1^2q_2^4 + 30240q_1^2q_2^4q_3 \\
& +2160q_1^2q_2^4q_3^2 + 13440q_1^2q_2^5q_3 + 30240q_1^3q_2^4q_3 + 240q_1^2q_2^6q_3 + 30240q_1^3q_2^4q_3^2 + 138240q_1^3q_2^5q_3 \\
& +6720q_1^3q_2^6 + 138240q_1^3q_2^5q_3^2 + 181440q_1^3q_2^6q_3 + 2160q_1^4q_2^4q_3^2 + 13440q_1^4q_2^5q_3 + \cdots \\
\varphi_6(\tau) \quad = \quad & 1 - 504q_1q_2^2 - 504q_1q_2^2q_3 - 504q_1^2q_2^2q_3 + 44352q_1^2q_2^3q_3 - 16632q_1^2q_2^4 + 166320q_1^2q_2^4q_3 \\
& -16632q_1^2q_2^4q_3^2 + 44352q_1^2q_2^5q_3 + 166320q_1^3q_2^4q_3^1 - 504q_1^2q_2^6q_3 + 166320q_1^3q_2^4q_3^2 + 2128896q_1^3q_2^5q_3 \\
& -122976q_1^3q_2^6 + 2128896q_1^3q_2^5q_3^2 + 3792096q_1^3q_2^6q_3^1 - 16632q_1^4q_2^4q_3^2 + 44352q_1^4q_2^5q_3 + \cdots \\
\varphi_8(\tau) \quad = \quad & 1 + 480q_1q_2^2 + 480q_1q_2^2q_3 + 480q_1^2q_2^2q_3 + 26880q_1^2q_2^3q_3 + 61920q_1^2q_2^4 + 175680q_1^2q_2^4q_3 \\
& +61920q_1^2q_2^4q_3^2 + 26880q_1^2q_2^5q_3 + 175680q_1^3q_2^4q_3^1 + 480q_1^2q_2^6q_3 + 175680q_1^3q_2^4q_3^2 \\
& +6727680q_1^3q_2^5q_3 + 1050240q_1^3q_2^6 + 6727680q_1^3q_2^5q_3^2 + 15914880q_1^3q_2^6q_3 + 61920q_1^4q_2^4q_3^2 \\
& +26880q_1^4q_2^5q_3 + \cdots \\
\varphi_{10}(\tau) \quad = \quad & 1 - 264q_1q_2^2 - 264q_1q_2^2q_3 - 264q_1^2q_2^2q_3 + \frac{227244864}{43867}q_1^2q_2^3q_3 - 135432q_1^2q_2^4 \\
& + \frac{2626026480}{43867}q_1^2q_2^4q_3 - 135432q_1^2q_2^4q_3^2 + \frac{227244864}{43867}q_1^2q_2^5q_3 + \frac{2626026480}{43867}q_1^3q_2^4q_3 - 264q_1^2q_2^6q_3 \\
& + \frac{2626026480}{43867}q_1^3q_2^4q_3^2 + \frac{306175997952}{43867}q_1^3q_2^5q_3 - 5196576q_1^3q_2^6 + \frac{306175997952}{43867}q_1^3q_2^5q_3^2 \\
& + \frac{950818774752}{43867}q_1^3q_2^6q_3 - 135432q_1^4q_2^4q_3^2 + \frac{227244864}{43867}q_1^4q_2^5q_3 + \cdots
\end{aligned}$$

$$\begin{aligned}\varphi_{12}(\tau) \quad = \quad & 1 + \frac{65520}{691} q_1 q_2^2 + \frac{65520}{691} q_1 q_2^2 q_3 + \frac{65520}{691} q_1^2 q_2^2 q_3 + \frac{22266840960}{53678953} q_1^2 q_2^3 q_3 + \frac{134250480}{691} q_1^2 q_2^4 \\ & + \frac{456798756960}{53678953} q_1^2 q_2^4 q_3 + \frac{134250480}{691} q_1^2 q_2^4 q_3^2 + \frac{22266840960}{53678953} q_1^2 q_2^5 q_3 + \frac{456798756960}{53678953} q_1^3 q_2^4 q_3 \\ & + \frac{65520}{691} q_1^2 q_2^6 q_3 + \frac{456798756960}{53678953} q_1^3 q_2^4 q_3^2 + \frac{162868282536960}{53678953} q_1^3 q_2^5 q_3 + \frac{11606736960}{691} q_1^3 q_2^6 \\ & + \frac{162868282536960}{53678953} q_1^3 q_2^5 q_3^2 + \frac{661522702800960}{53678953} q_1^3 q_2^6 q_3 + \frac{134250480}{691} q_1^4 q_2^4 q_3^2 + \frac{22266840960}{53678953} q_1^4 q_2^5 q_3 \cdots\end{aligned}$$

3.2.4 Lien avec les invariants d'Igusa

On reprend les invariants d'Igusa A, B, C, D définis au chapitre 1, ainsi que les invariants absolus :

$$j_1 = 2^4 3^2 B/A^2, \ j_2 = 2^6 3^3 (3C - AB)/A^3, \ j_3 = 2 \cdot 3^5 D/A^5.$$

L'intérêt est de faire le lien avec les formes modulaires, ce qui est fait grâce au théorème suivant tiré de [Igu62] :

Théorème 3.13 *Les trois invariants absolus d'Igusa s'expriment en tant que fonctions modulaires sous la forme*

$$j_1 = \frac{\varphi_4 \chi_{10}^2}{\chi_{12}^2}, \ j_2 = \frac{\varphi_6 \chi_{10}^3}{\chi_{12}^3}, \ j_3 = \frac{\chi_{10}^6}{\chi_{12}^5},$$

où φ_k désigne la série d'Eisenstein de poids k, et χ_k une forme parabolique de poids k normalisée :

$$\chi_{10} = -43867 \cdot 2^{-12} \cdot 3^{-5} \cdot 5^{-2} \cdot 7^{-1} \cdot 53^{-1} (\varphi_4 \varphi_6 - \varphi_{10}),$$

$$\chi_{12} = 131 \cdot 593 \cdot 2^{-13} \cdot 3^{-7} \cdot 5^{-3} \cdot 7^{-2} \cdot 337^{-1} (3^2 7^2 \varphi_4^3 + 2 \cdot 5^3 \varphi_6^2 - 691 \varphi_{12}).$$

Pour vérifier les calculs, on peut faire les opérations suivantes. Partant d'une courbe hyperelliptique de genre 2, on peut calculer ses invariants absolus tout d'abord de manière purement algébrique, par des formules closes, et d'autre part en passant par les fonctions modulaires. Pour cela, on calcule de manière approchée la matrice des périodes τ de la courbe et on ramène τ dans le domaine fondamental. Par la suite, on peut évaluer les séries d'Eisenstein en cette valeur de τ grâce à leur développement en série entière, puis on vérifie que l'on retombe bien sur des valeurs approchées des invariants absolus.

Exemple numérique :

Considérons la courbe de genre 2 suivante, tirée de la thèse de Spallek [Spa94, p. 80, ex 1],

$$y^2 = x^5 + 23x^4 + 122x^3 - 386x^2 - 3019x + 187.$$

Sa matrice des périodes est donnée par la théorie de la multiplication complexe. On peut aussi la recalculer de manière approchée en évaluant les intégrales autour de contours, comme décrit en page 31

$$\tau_0 = \begin{pmatrix} \sqrt{2 + \sqrt{2}} i & \frac{1}{2}\sqrt{4 - 2\sqrt{2}} i \\ \frac{1}{2}\sqrt{4 - 2\sqrt{2}} i & \frac{1}{2}\sqrt{2 + \sqrt{2}} i \end{pmatrix}.$$

On applique l'algorithme 1.1, page 36 pour ramener cette matrice au domaine fondamental. On obtient :

$$\tau = \begin{pmatrix} \sqrt{4 - 2\sqrt{2}} i & \sqrt{2} - 1 \\ \sqrt{2} - 1 & 2\sqrt{2 - \sqrt{2}} i \end{pmatrix}.$$

Le calcul direct des invariants absolus à partir de l'équation de la courbe donne :

$$j_1 = \frac{20}{9},\ j_2 = \frac{8}{9},\ j_3 = \frac{1}{3779136}.$$

Pour évaluer j_1, j_2, j_3 avec la valeur de τ, pour chaque j_i, on calcule le développement en série du numérateur et du dénominateur, que l'on évalue en τ, et on effectue finalement la division des nombres complexes obtenus. À partir des développements des séries d'Eisenstein jusqu'au degré 40, on obtient :

$$\begin{aligned} j_1 &= 2.2222222222222222183 - 1.477\ 10^{-18}i \\ j_2 &= 0.8888888888888884347 + 1.223\ 10^{-17}i \\ j_3 &= 2.64609\ 10^{-7} - 1.223\ 10^{-13}i \end{aligned}$$

qui se rapprochent des valeurs exactes.

3.2.5 Développement en série des invariants absolus

Les invariants j_1, j_2, j_3 définis plus haut s'expriment comme quotient de formes modulaires de même poids. Chacune de ces formes modulaires s'exprime comme un polynôme en les séries d'Eisenstein dont on connaît le développement en série. Pour obtenir un développement en série des invariants, il « suffit » donc de faire le quotient de deux séries entières en 3 indéterminées. Le problème est que ce n'est pas toujours possible : le corps de fraction des séries entières n'est plus l'anneau des séries de Laurent quand on a plus d'une indéterminée.

Toutefois pour le cas qui nous intéresse, ça se passe bien grâce à la proposition suivante.

Proposition 3.8 *Pour toute forme parabolique, on peut mettre $q_1^2q_2^3q_3$ en facteur dans le développement en série de Fourier défini plus haut.*

Pour le calcul des invariants, on doit diviser par la forme parabolique χ_{12}. Le développement de celle-ci débute par $\frac{1}{12}q_1^2q_2^3q_3$. Ainsi, en mettant ce terme en facteur grâce à la proposition, on se ramène à inverser une série ayant un terme constant non nul, ce qui est tout à fait possible par la même méthode que pour les séries à une indéterminée. Les invariants admettent donc un développement en série de Laurent (le calcul montre qu'il s'agit en fait d'un développement en série entière).

Début du développement :

$$\begin{aligned} j_1 &= 9 - 216q_2 - 216q_1q_2 + 3456q_2^2 - 47304q_2^3 - 216q_1q_2q_3 + 10368q_1q_2^2 \\ &\quad +597888q_2^4 + 3456q_1^2q_2^2 - 238680q_1q_2^3 + 10368q_1q_2^2q_3 + \cdots \end{aligned}$$

$$\begin{aligned} j_2 &= -27 + 972q_2 - 21384q_2^2 + 972q_1q_2 + 376164q_2^3 - 34992q_1q_2^2 + 972q_1q_2q_3 \\ &\quad -21384q_1^2q_2^2 + 910764q_1q_2^3 - 34992q_1q_2^2q_3 - 5828112q_2^4 + \cdots \end{aligned}$$

$$\begin{aligned} j_3 &= q_1^2q_2^3q_3\big(\tfrac{243}{4} - \tfrac{7533}{2}q_2 + \tfrac{525123}{4}q_2^2 - \tfrac{7533}{2}q_1q_2 - \tfrac{7533}{2}q_1q_2q_3 + 283338q_1q_2^2 \\ &\quad -3411720q_2^3 + \tfrac{525123}{4}q_1^2q_2^2 - 11756583q_1q_2^3 + 283338q_1q_2^2q_3 + 73876860q_2^4 + \cdots\big) \end{aligned}$$

3.3 Équations modulaires en dimension 2

Toujours dans le but d'étendre les propriétés des formes modulaires de genre 1, on s'intéresse maintenant aux équations modulaires. Dans le cas elliptique, celles-ci permettent de relier par une équation polynomiale les j-invariants de deux courbes elliptiques isogènes.

3.3.1 Le sous-groupe $\Gamma_0(N)$

Il s'agit tout d'abord d'introduire les sous-groupes de congruence de $\mathrm{Sp}_4(\mathbb{Z})$.

Définition 3.8 *Pour N un entier strictement positif, soit $\Gamma(N)$ l'ensemble des matrices $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathrm{Sp}_4(\mathbb{Z})$ vérifiant $\gamma \equiv \mathrm{Id} \mod N$. Cet ensemble $\Gamma(N)$ forme un sous-groupe de $\mathrm{Sp}_4(\mathbb{Z})$ appelé le* sous-groupe principal de congruence de niveau *N.*

Tout sous-groupe de $\mathrm{Sp}_4(\mathbb{Z})$ contenant $\Gamma(N)$ pour un entier N est appelé sous-groupe de congruence*.*

Définition 3.9 *L'ensemble des matrices $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ de $\mathrm{Sp}_4(\mathbb{Z})$ vérifiant $C \equiv 0 \mod N$ forment un sous-groupe de $\mathrm{Sp}_4(\mathbb{Z})$ que l'on note $\Gamma_0(N)$ (c'est un sous-groupe de congruence).*

Ce sous-groupe est très similaire à son homonyme elliptique. En effet c'est lui qui permettra de construire les équations modulaires liées aux isogénies, et ce grâce à la proposition suivante.

Proposition 3.9 *Si on a f une fonction modulaire, alors $F(\tau) = f(N\tau)$ est une fonction invariante sous l'action de $\Gamma_0(N)$. La fonction F est appelée une fonction modulaire pour $\Gamma_0(N)$.*

Démonstration. Soient $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \Gamma_0(N)$, et $\tau \in \mathbb{H}_2$.

Comme $\gamma \in \Gamma_0(N)$, on a $C \equiv 0 \mod N$, et donc $C = NC'$ où C' est une matrice entière. On vérifie aisément que $\begin{pmatrix} A & NB \\ C' & D \end{pmatrix}$ appartient à $\mathrm{Sp}_4(\mathbb{Z})$.

On a alors

$$\begin{aligned} F(\gamma.\tau) &= f(N.\gamma\tau) \\ &= f(N.(A\tau + B)(C\tau + D)^{-1}) \\ &= f((A.(N\tau) + N.B)(C'.(N\tau) + D)^{-1}) \\ &= f(N\tau) = F(\tau). \end{aligned}$$

La fonction F est donc bien invariante sous l'action de $\Gamma_0(N)$. □

Pour tout sous-groupe Γ d'index fini dans $\mathrm{Sp}_4(\mathbb{Z})$, on note $\mathbb{C}(\Gamma)$ le corps des fonctions méromorphes de $\mathbb{H}_2$, invariantes sous l'action de Γ. En particulier $\mathbb{C}(\mathrm{Sp}_4(\mathbb{Z}))$ représente les fonctions modulaires.

Théorème 3.14 *Si Γ est un sous-groupe de $\mathrm{Sp}_4(\mathbb{Z})$ d'index k, alors le corps $\mathbb{C}(\Gamma)$ est une extension algébrique finie de degré k de $\mathbb{C}(\mathrm{Sp}_4(\mathbb{Z}))$.*

La démonstration de ce théorème (qui fait appel à la théorie de Galois) se trouve dans Freitag [Fre83].

On connaît la structure du corps des fonctions modulaires (c'est un corps de fonctions rationnelles), le théorème nous donne donc la structure de $\mathbb{C}(\Gamma_0(N))$. Il reste alors à calculer explicitement cette extension. C'est le rôle des équations modulaires.

3.3.2 Les équations modulaires

On applique le théorème précédent à $\mathbb{C}(\Gamma_0(N))$. Explicitons les objets rentrant en jeu. Le corps de base est $\mathbb{C}(\mathrm{Sp}_4(\mathbb{Z}))$, c'est-à-dire le corps des fonctions modulaires. D'après la section 3, ce corps peut s'écrire $\mathbb{C}(g_1, g_2, g_3)$. On peut facilement montrer que c'est encore vrai pour les invariants absolus d'Igusa que nous avons noté j_1, j_2, j_3, afin de coller aux notations standards du cas elliptique.

Maintenant définissons les fonctions

$$\begin{aligned} J_1(\tau) &= j_1(N\tau), \\ J_2(\tau) &= j_2(N\tau), \\ J_3(\tau) &= j_3(N\tau). \end{aligned}$$

Celles-ci appartiennent à $\mathbb{C}(\Gamma_0(N))$, grâce à la proposition 3.9. Le théorème 3.14 nous assure donc l'existence d'équations polynomiales reliant ces fonctions :

Définition 3.10 *Il existe des polynômes en 4 indéterminées ψ_1^N, ψ_2^N, ψ_3^N tels que :*

$$\begin{aligned} \psi_1^N(J_1(\tau), j_1(\tau), j_2(\tau), j_3(\tau)) &= 0, \\ \psi_2^N(J_2(\tau), j_1(\tau), j_2(\tau), j_3(\tau)) &= 0, \\ \psi_3^N(J_3(\tau), j_1(\tau), j_2(\tau), j_3(\tau)) &= 0. \end{aligned}$$

Ces équations sont appelées équations modulaires.

Plus généralement, l'ensemble des polynômes en 6 variables qui s'annulent en $J_1, J_2, J_3, j_1, j_2, j_3$ est un idéal de dimension 3 appelé *idéal modulaire* et qui contient les équations modulaires.

Le théorème nous donne de plus une borne sur le degré de l'extension de corps, donc une borne sur le degré des polynômes ψ_1, ψ_2, ψ_3 en la première variable.

La symétrie du système laisse penser que cette borne est valable aussi pour les autres variables[1].

Proposition 3.10 *Le degré des polynômes ψ_1, ψ_2, ψ_3 en la première variable est majoré par $[\mathrm{Sp}_4(\mathbb{Z}) : \Gamma_0(N)]$. Dans le cas où $N = p$ est premier, on a*

$$[\mathrm{Sp}_4(\mathbb{Z}) : \Gamma_0(p)] = (p+1)(p^2+1).$$

3.3.3 Lien avec les isogénies

Lorsque l'on multiplie la matrice des périodes τ d'une courbe $\mathcal{C}$ de genre 2 par l'entier N, la Jacobienne de $\mathcal{C}$ est transformée en une variété abélienne qui lui est (N,N)-isogène. On obtient donc la caractérisation suivante :

Théorème 3.15 *Soient $\mathcal{C}$ et $\mathcal{C}'$ deux courbes de genre 2 d'invariants d'Igusa respectifs j_1, j_2, j_3 et J_1, J_2, J_3. Alors les Jacobiennes de $\mathcal{C}$ et $\mathcal{C}'$ sont (N,N)-isogènes si et seulement si ces 6 invariants annulent tous les polynômes de l'idéal modulaire.*

1. Nous n'avons pas été capable de prouver rigoureusement que c'était effectivement le cas. Mener le calcul jusqu'au bout permettra de s'en assurer.

3.3.4 Algorithme naïf pour le calcul explicite

Une fois que l'on connaît l'existence des équations modulaires, ainsi que des bornes raisonnables pour le degré en chaque variable, un moyen pour déterminer cette équation est de remplacer les indéterminées par le développement en série de $j_i(\tau)$, en gardant comme inconnues les coefficients de l'équation. Ainsi si on a les développements à un ordre suffisant, il suffit de résoudre un système linéaire pour trouver l'équation modulaire cherchée.

Par exemple, pour trouver le polynôme ψ_1^2 tel que

$$\psi_1^2(j_1(2\tau), j_1(\tau), j_2(\tau), j_3(\tau)) = 0,$$

on écrit

$$\psi_1^2(X_1, Y_1, Y_2, Y_3) = \sum_{0 \leqslant l,m,n,p \leqslant 15} c_{lmnp} X_1^l Y_1^m Y_2^n Y_3^p,$$

et on remplace les j_i par leur développement en série. En écrivant que chaque terme du développement obtenu doit être nul, on obtient des équations linéaires en les c_{lmnp}.

Dans cet exemple, le nombre d'inconnues est $16^4 = 65536$, ce qui rend le calcul peu praticable compte-tenu de la précision nécessaire pour les séries.

3.3.5 Amélioration pour le cas $N = 2$

Dans l'optique d'obtenir au moins certains coefficients de l'équation modulaire pour $N = 2$, on peut essayer d'étendre des techniques classiques de calcul d'équations modulaires elliptiques. Décrivons la procédure pour ψ_1^2.

Soit $K = \mathbb{C}(j_1(\tau), j_2(\tau), j_3(\tau))$ le corps des fonctions modulaires, et considérons ψ_1^2 comme un polynôme à coefficients dans K. C'est alors un polynôme univarié noté $\overline{\psi_1^2}$, et vérifiant $\overline{\psi_1^2}(j_1(2\tau)) = 0$. On met pour l'instant de coté le coefficient dominant de ce polynôme que l'on considère donc unitaire.

On dispose en fait d'autres racines de ψ_1^2. En effet, si $\{\gamma_1, \ldots, \gamma_{15}\}$ est un ensemble de représentants des classes à droite de $\Gamma_0(2)$ dans $\mathrm{Sp}_4(\mathbb{Z})$, alors les $j_1(2\gamma_i\tau)$ sont les conjugués de $j_1(2\tau)$, et sont donc les racines de $\overline{\psi_1^2}$. (Cela revient à remplacer τ par $\gamma_i\tau$ dans l'équation $\psi_1^2(j_1(2\tau), j_1(\tau), j_2(\tau), j_3(\tau)) = 0$.) Ainsi on a

$$\overline{\psi_1^2}(X) = \prod_{i=1}^{15} (X - j_1(2\gamma_i\tau)).$$

Les coefficients de l'équation modulaire sont donc les fonctions symétriques élémentaires en les $(j_1(2\gamma_i\tau))_{1\leqslant i\leqslant 15}$.

Détermination explicite de représentants des cosets

Pour trouver des représentants des classes dans le cas $N = 2$, une recherche brutale suffit. On énumère tous les éléments de $\mathrm{Sp}_4(\mathbb{F}_2)$, et on garde un sous-ensemble de ces matrices qui soit maximal et libre. En effet le fait que deux matrices de $\mathrm{Sp}_4(\mathbb{Z})$ sont dans la même classe se voit sur leur réduction modulo 2. Soient $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ et $M' = \begin{pmatrix} A' & B' \\ C' & D' \end{pmatrix}$ deux matrices de $\mathrm{Sp}_4(\mathbb{Z})$. Celles-ci sont dans la même classe si et seulement si MM'^{-1} appartient à $\Gamma_0(2)$. Ce qui se traduit par $C^tD' \equiv D^tC' \mod 2$.

Une fois que l'on a les quinze représentants dans $\mathrm{Sp}_4(\mathbb{F}_2)$, on remonte ceux-ci dans $\mathrm{Sp}_4(\mathbb{Z})$. Ces γ_i étant obtenus, on multiplie leurs deux premières lignes par $N = 2$, afin d'avoir des matrices M_i telles que

$$\psi_1^2(X) = \prod_{i=1}^{15} (X - j_1(M_i\tau)).$$

On peut alors multiplier à gauche les matrices M_i par des matrices symplectiques quelconques sans changer la valeur de $j_1(M_i\tau)$, car j_1 est une fonction modulaire. On utilise cette liberté afin de garantir la convergence (en tant que série formelle) du développement de $j_1(M_i\tau)$.

Le tableau 3.1 donne les matrices que nous avons choisies.

Autres pistes envisageables

Nous n'avons pas été capable de mener ces calculs jusqu'au bout par cette approche : les séries en trois variables à manipuler deviennent très vite énormes et on atteint les limites de la technologie. Toutefois, les progrès matériels et logiciels laissent supposer que cela sera faisable dans un proche avenir.

Un autre moyen de calculer ces équations modulaires pour $N = 2$ serait d'engendrer suffisamment de couples de courbes isogènes, d'évaluer leurs invariants, puis d'interpoler. Cette phase d'interpolation paraît elle aussi assez difficile avec la technologie actuelle, mais pas complètement irréalisable. Le problème est d'engendrer les courbes. Une piste à suivre (idée de R. Harley) est l'isogénie de Richelot (cf [CF96] et [BM88]) qui est utilisée dans les techniques de descente pour calculer le rang de la Jacobienne sur $\mathbb{Q}$.

Il est aussi probable que des exemples de courbes à multiplication complexe ou a multiplication réelle trouvées dans la littérature permettront de rajouter des conditions. En effet, si une courbe a sa Jacobienne qui est $(2, 2)$-isogène à elle-même, son anneau d'endomorphisme contiendra un élément de norme 4, ce qui peut aider à deviner la « diagonale » de l'idéal modulaire.

matrice γ_i	matrice de transformation P_i	matrice finale $M_i = P_i.2.\gamma_i$
$\begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$	$\begin{pmatrix} 2I & 0 \\ 0 & I \end{pmatrix}$
$\begin{pmatrix} \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix} & \begin{matrix} -1 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 1 & 0 \\ 0 & 0 \end{matrix} & I \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} 0 & 2 \\ 1 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 3 & 0 \end{matrix} \\ 0 & \begin{matrix} 0 & 1 \\ 2 & 0 \end{matrix} \end{pmatrix}$
$\begin{pmatrix} 0 & -I \\ I & I \end{pmatrix}$	$\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$	$\begin{pmatrix} I & I \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} \begin{matrix} 1 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & -1 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix} & I \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} 2 & 0 \\ 0 & 1 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 3 \end{matrix} \\ 0 & \begin{matrix} 1 & 0 \\ 0 & 2 \end{matrix} \end{pmatrix}$
$\begin{pmatrix} 0 & \begin{matrix} 0 & -1 \\ -1 & 1 \end{matrix} \\ \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix} & I \end{pmatrix}$	$\begin{pmatrix} -I & \begin{matrix} 0 & 1 \\ 1 & -1 \end{matrix} \\ \begin{matrix} -1 & -1 \\ -1 & 0 \end{matrix} & 0 \end{pmatrix}$	$\begin{pmatrix} I & \begin{matrix} 0 & 3 \\ 3 & -3 \end{matrix} \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} \begin{matrix} 1 & 0 \\ -1 & 0 \end{matrix} & \begin{matrix} 1 & -1 \\ -1 & 0 \end{matrix} \\ \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} & I \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 1 & -1 \\ 2 & 1 & 1 & -2 \\ -1 & -1 & 1 & -1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} 2 & 0 \\ 1 & -1 \end{matrix} & \begin{matrix} 3 & -3 \\ 3 & -6 \end{matrix} \\ 0 & \begin{matrix} 1 & 1 \\ 0 & -2 \end{matrix} \end{pmatrix}$
$\begin{pmatrix} 0 & \begin{matrix} 1 & -1 \\ -1 & 0 \end{matrix} \\ \begin{matrix} 0 & 1 \\ 1 & 1 \end{matrix} & I \end{pmatrix}$	$\begin{pmatrix} -I & \begin{matrix} -1 & 1 \\ 1 & 0 \end{matrix} \\ \begin{matrix} 0 & -1 \\ -1 & -1 \end{matrix} & 0 \end{pmatrix}$	$\begin{pmatrix} I & \begin{matrix} -3 & 3 \\ 3 & 0 \end{matrix} \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} 0 & \begin{matrix} 0 & -1 \\ -1 & 0 \end{matrix} \\ \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} & I \end{pmatrix}$	$\begin{pmatrix} -I & \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \\ \begin{matrix} 0 & -1 \\ -1 & 0 \end{matrix} & 0 \end{pmatrix}$	$\begin{pmatrix} I & \begin{matrix} 0 & 3 \\ 3 & 0 \end{matrix} \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} 0 & -I \\ I & \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} -1 & -1 \\ -1 & -1 \end{matrix} & I \\ -I & 0 \end{pmatrix}$	$\begin{pmatrix} I & \begin{matrix} 3 & 3 \\ 3 & 3 \end{matrix} \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & -1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} 2 & 0 \\ 1 & -1 \end{matrix} & \begin{matrix} 3 & 3 \\ 0 & 0 \end{matrix} \\ 0 & \begin{matrix} 1 & 1 \\ 0 & -2 \end{matrix} \end{pmatrix}$
$\begin{pmatrix} 0 & -I \\ I & \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix} \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} 0 & 0 \\ 0 & -1 \end{matrix} & I \\ -I & 0 \end{pmatrix}$	$\begin{pmatrix} I & \begin{matrix} 0 & 0 \\ 0 & 3 \end{matrix} \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} 0 & -I \\ I & \begin{matrix} 1 & 0 \\ 0 & 0 \end{matrix} \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} -1 & 0 \\ 0 & 0 \end{matrix} & I \\ -I & 0 \end{pmatrix}$	$\begin{pmatrix} I & \begin{matrix} 3 & 0 \\ 0 & 0 \end{matrix} \\ 0 & 2I \end{pmatrix}$
$\begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} 0 & 2 \\ 1 & 0 \end{matrix} & 0 \\ 0 & \begin{matrix} 0 & 1 \\ 2 & 0 \end{matrix} \end{pmatrix}$
$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \begin{matrix} 2 & 0 \\ 0 & 1 \end{matrix} & 0 \\ 0 & \begin{matrix} 1 & 0 \\ 0 & 2 \end{matrix} \end{pmatrix}$

TAB. 3.1: *Matrices des cosets*

Deuxième partie

Algorithmique, calcul de la cardinalité

Chapitre 4

Loi de groupe dans la Jacobienne d'une courbe hyperelliptique

Dans ce court chapitre, nous rappelons les méthodes connues pour calculer efficacement dans les Jacobiennes de courbes hyperelliptiques. Le cas des courbes n'admettant qu'un point à l'infini (modèle imaginaire) est le plus facile. Le cas général est plus compliqué, et ne sera pas traité ici.

4.1 Représentation de Mumford

Soit $\mathcal{C}$ une courbe hyperelliptique de genre g sur un corps K, donnée par une équation affine de la forme

$$y^2 + h(x)y = f(x).$$

La première chose lorsque l'on veut calculer dans un groupe, est de trouver une représentation adéquate de ses éléments. L'idée naturelle est de se fixer un point rationnel P_∞ sur la courbe, puis d'utiliser le théorème 1.6 (ou même 1.7) afin d'avoir un représentant de la classe du diviseur que l'on veut manipuler. On peut alors stocker tous les points qui définissent ce diviseur grâce à leurs coordonnées. Cette approche présente l'inconvénient de nécessiter de travailler dans des extensions finies du corps K.

Dans le cas où (le modèle désingularisé de) la courbe $\mathcal{C}$ n'a qu'un seul point à l'infini, c'est un candidat naturel pour jouer le rôle de P_∞, et les théorèmes de représentation unique prennent une forme agréable.

Théorème 4.1 *Soit $\mathcal{C}$ une courbe hyperelliptique de genre g d'équation $y^2 + h(x)y = f(x)$ avec $\deg f = 2g+1$ et $\deg h \leqslant g$, i.e. ayant un unique point à l'infini, que l'on notera ∞. Alors toute classe de $\mathrm{Jac}(\mathcal{C})$ contient un unique diviseur D vérifiant*

1. *il existe E effectif de degré r inférieur ou égal à g tel que $D = E - r\infty$;*
2. *le point ∞ n'apparaît pas dans E ;*
3. *si $P = (x, y)$ apparaît dans E, alors $\iota(P) = (x, -y - h(x))$ n'y apparaît pas ;*
4. *les points de ramifications apparaissent avec un coefficient au plus 1 dans E.*

Un diviseur sous cette forme est appelé diviseur réduit*. Si l'on relâche la condition $r = \deg E \leqslant g$, on n'a plus unicité et on dit que le diviseur est* semi-réduit.

Démonstration. L'existence découle du théorème 1.6. En effet, ce théorème donne directement un diviseur E effectif vérifiant les conditions 1 et 2. Si dans E apparaissent simultanément P et $\iota(P)$, on peut les enlever, car le diviseur $P + \iota(P) - 2\infty$ est principal : c'est le diviseur de la fonction $x - x_P$. De même, si un point de ramification apparaît avec un coefficient d'au moins 2, on peut retirer 2 à ce coefficient, et donc tout nombre pair : on se ramène à un coefficient 0 ou 1.

L'unicité est moins immédiate. Une démonstration dans le cas de la caractéristique nulle est donnée par Mumford [Mum84, p. 30]. On peut adapter ses arguments au cas général. Soit $D = E - r\infty$ un diviseur réduit ($r \leqslant g$). Supposons qu'il existe un diviseur réduit $D' = E' - r'\infty$ linéairement équivalent à D avec $r' < r$. Cela signifie qu'il existe une fonction φ telle que

$$E - r\infty = E' - r'\infty + \operatorname{div}(\varphi).$$

Écrivons $E' = \sum n_i P_i$ avec $P_i = (x_i, y_i)$, et soit δ la fonction $\prod (x - x_i)^{n_i}$. On a alors

$$\operatorname{div}(\delta) = E' + \iota(E') - 2r'\infty,$$

où $\iota(E')$ désigne le diviseur obtenu en prenant les images par ι de chaque point formant E'. La fonction $\varphi\delta$ a donc pour diviseur

$$\operatorname{div}(\varphi\delta) = E + \iota(E') - (r + r')\infty.$$

Cette fonction n'a de pôles qu'en ∞, c'est donc un polynôme ; de plus ce pôle est d'ordre inférieur à $2g$ car $r' < r \leqslant g$, donc ce polynôme ne fait pas intervenir la variable y. En effet, en réduisant par l'équation de la courbe, on peut supposer que le polynôme s'écrit $\sigma(x) + y\tau(x)$, mais l'ordre de y à l'infini est $2g + 1$ donc impair, alors que celui de x est 2. Il ne peut donc pas y avoir de compensation de valuations entre $\sigma(x)$ et $y\tau(x)$ pour des raisons de parité et le pôle engendré par $y\tau(x)$ est d'ordre trop élevé à moins que $\tau(x) = 0$. Ainsi on a

$$\varphi(x, y)\delta(x) = \sigma(x).$$

La fonction φ est donc elle-même une fraction rationnelle en x uniquement, et la réduction du diviseur D en D' ne peut donc consister qu'en des simplifications triviales du type $P + \iota(P) - 2\infty \sim 0$. Celles-ci étant déjà effectuées dans D, il n'existe pas de D' avec un poids inférieur. $\square$

Remarque. Cette définition de *diviseur réduit* coïncide avec celle donnée par le théorème 1.7 dans le cadre d'une courbe un peu plus générale.

Une représentation efficace de cet élément canonique se trouve dans le livre de Mumford [Mum84] qui l'attribue à Jacobi.

Proposition 4.1 *Reprenons les conditions du théorème précédent. Soit $D = E - r\infty$ un diviseur réduit ou semi-reduit, avec $E = \sum n_i P_i$, où $P_i = (x_i, y_i)$. Alors D peut être représenté de manière unique par deux polynômes $u(x)$ et $v(x)$ définis par*

$$u(x) = \prod_i (x - x_i),$$

et $v(x)$ est l'unique polynôme de degré inférieur à $r = \deg u$ tel que pour tout i, $v(x_i) = y_i$, et $u(x)$ divise $v^2(x) + v(x)h(x) - f(x)$.

Cette représentation peut être comprise de la manière suivante : le premier polynôme décrit les abscisses des points de E, et le deuxième « interpole » les ordonnées. La condition de divisibilité permet de prendre en compte les éventuelles multiplicités. Notons que le fait d'avoir interdit la présence de deux points distincts de même abscisse est nécessaire pour définir $v(x)$.

En accord avec les définitions générales du chapitre 1, le degré du polynôme u est appelé le *poids* du diviseur, et le diviseur est dit *premier* si le polynôme u est irréductible.

Remarque. Si la caractéristique est différente de 2, le polynôme $h(x)$ dans l'équation de la courbe peut être choisi égal à zéro. La condition de divisibilité devient $u \mid v^2 - f$. Posons alors

$$w = \frac{v^2 - f}{u}.$$

La forme quadratique $uX^2 + 2vX + w$ est alors de discriminant $-f$. L'algorithme de Cantor, tenant compte de cette analogie avec les corps quadratiques imaginaires va mimer la composition et la réduction des formes quadratiques à la Gauß.

Notation. Un diviseur réduit ou semi-réduit dans sa représentation de Mumford est noté

$$D = \langle u(x), v(x) \rangle.$$

4.2 Algorithme de Cantor

Le problème est le suivant : étant donnés deux diviseurs réduits dans la représentation de Mumford, trouver le diviseur réduit représentant leur somme dans la Jacobienne (toujours sous forme de Mumford). Nous décrivons ici une solution proposée par Cantor [Can87] (voir aussi [Kob89] pour le cas de la caractéristique 2).

Tout d'abord le lemme suivant traite le problème de l'opposé.

Lemme 4.1 *Soit $\mathcal{C}$ une courbe hyperelliptique de genre g d'équation $y^2 + h(x)y = f(x)$ avec $\deg f = 2g+1$ et $\deg h \leqslant g$. Soit $D = \langle u(x), v(x) \rangle$ un diviseur réduit (ou semi-réduit). Alors son opposé dans* $\mathrm{Jac}(\mathcal{C})$ *est donné sous forme de Mumford par*

$$-D = \langle u(x),\ -v(x) - h(x) \bmod u(x) \rangle\,.$$

Démonstration. Soit $P = (x_P, y_P)$ un point de la courbe. Le diviseur de la fonction $x - x_P$ est $P + \iota(P) - 2\infty$. Ainsi l'opposé du diviseur réduit $D = P - \infty$ est $-D = \iota(P) - \infty$. Prenant en compte le fait que $\iota(P)$ a pour coordonnées $(x_P, -y_P - h(x_P))$, on obtient le résultat pour les diviseurs réduits de poids 1, et la formule générale s'ensuit. □

L'algorithme d'addition dans la Jacobienne se décompose en deux parties : la *composition* calcule un diviseur semi-réduit représentant la somme de deux diviseurs, la *réduction* transforme un diviseur semi-réduit en un diviseur réduit.

Algorithme 4.2 COMPOSITION
Entrée: Deux diviseurs semi-réduits $D_1 = \langle u_1(x), v_1(x) \rangle$ et $D_2 = \langle u_2(x), v_2(x) \rangle$.
Sortie: Un diviseur semi-réduit D_3 tel que $D_3 \sim D_1 + D_2$ dans $\mathrm{Jac}(\mathcal{C})$.

1. Par deux calculs de pgcd étendus, construire s_1, s_2, s_3 tels que

$$d = \mathrm{pgcd}(u_1, u_2, v_1 + v_2 + h) = s_1 u_1 + s_2 u_2 + s_3(v_1 + v_2 + h);$$

2. $u_3 \leftarrow u_1 u_2 / d^2$;

3. $v_3 \leftarrow (s_1 u_1 v_2 + s_2 u_2 v_1 + s_3(v_1 v_2 + f))/d \bmod u_3$;

4. Retourner $D_3 = \langle u_3(x), v_3(x) \rangle$.

Dans le cas le plus simple où D_1 et D_2 n'ont pas de point en commun, le calcul consiste simplement à regrouper les deux ensembles de g points en un ensemble de $2g$ points. Le polynôme u_3 de la somme est donc le produit $u_1 u_2$, et le polynôme $v_3 = s_1 u_1 v_2 + s_2 u_2 v_1 \bmod u_3$ est une sorte d'interpolation de Lagrange. Dans le cas général, on peut montrer que les formules de l'algorithme consistent exactement à traiter les cas de points multiples, ou éventuellement opposés qu'il faut éliminer. Nous renvoyons à [Can87] pour une preuve détaillée.

Algorithme 4.3 RÉDUCTION
Entrée: Un diviseur semi-réduit $D = \langle u(x), v(x) \rangle$.
Sortie: Un diviseur réduit $D' \sim D$.

1. Tant que $\deg u(x) > g$, faire
2. $\quad u \leftarrow (f - hv - v^2)/u$;
3. $\quad v \leftarrow -h - v \bmod u$;
4. Retourner $D' = \langle u(x), v(x) \rangle$.

La phase de réduction est celle qui fait réellement intervenir la structure de Jacobienne : on fait baisser le poids du diviseur semi-réduit en ajoutant des diviseurs principaux. Si l'on part d'un diviseur de poids $r > g$, alors $\deg u = r$ et $\deg v < r$. Une itération de la boucle va donc faire baisser le poids de 2 (sauf éventuellement la dernière qui ne le fait baisser que de 1). Là encore, nous renvoyons à [Can87] pour une preuve de l'algorithme.

Complexité

Partant de deux diviseurs réduits (donc de poids borné par g), on commence par les composer : on obtient un diviseur semi-réduit de poids au plus $2g$, puis on réduit celui-ci pour obtenir un diviseur réduit.

Notons $M(x)$ le nombre d'opérations dans le corps de base nécessaires pour multiplier deux polynômes de degré au plus x. Le calcul du pgcd de deux polynômes de degré au plus x peut alors s'effectuer en $M(x) \log x$ opérations [vzGG99].

L'étape de composition de deux diviseurs réduits fait intervenir un nombre fini de pgcd et de multiplications de polynômes de degré au plus $2g$. Ainsi la complexité de cette phase est

$$C_{comp} = O(M(g) \log g)$$

opérations dans K.

Comme nous l'avons déjà remarqué, chaque itération dans la phase de réduction fait baisser le poids de 2. On part d'un diviseur de poids au plus $2g$, et l'on veut atteindre un poids inférieur ou égal à g. Cela nécessite donc au plus $O(g)$ étapes. Chaque itération coûte $O(M(g))$. La complexité de la réduction est donc

$$C_{red} = O(gM(g))$$

opérations dans K.

Remarque. Dans l'article original de Cantor, un second algorithme de réduction est proposé, dont la complexité est asymptotiquement meilleure. Cela permet de rabaisser le coût de la réduction au même niveau que celui de la composition : $O(M(g)\log g)$. Cet algorithme repose sur l'évaluation rapide de fractions continues. Toutefois, nous nous intéresserons principalement aux courbes de genre « pas trop grand » pour lesquelles le premier algorithme reste plus efficace.

Remarque. La valeur de $M(g)$ dépend de l'algorithme choisi pour multiplier les polynômes. L'algorithme naïf donne $M(g) = O(g^2)$, la méthode de Karatsuba descend à $M(g) = O(g^{1.58\cdots})$, et les méthodes rapides à base de transformée de Fourier rapide (FFT) donnent

$$M(g) = O(g \log g \log \log g).$$

Dès les petits degrés, Karatsuba permet de gagner par rapport à la méthode naïve. Par contre, la FFT ne sera pas utile dans notre contexte.

4.3 Genre 2 : formules de Spallek et de Harley

Pour un genre fixé, il est possible de dérouler les algorithmes. En particulier, pour le genre 2, cela donne des formules qui restent de taille raisonnable. Les différents cas à envisager correspondent

– aux différents poids des diviseurs en entrée ;

– aux différents branchements possibles dans les algorithmes de pgcd.

Pour chaque cas il est possible de trouver des formules permettant d'obtenir la somme de deux diviseurs avec peu de calculs. Le fait qu'il soit nécessaire d'envisager une famille de formules plutôt qu'une seule n'est pas vraiment étonnant : si l'on note A la variété abélienne Jacobienne de la courbe, on veut définir une fonction φ de $A\times A$ dans A. Cette fonction est régulière et s'exprime localement par une fraction rationnelle. Toutefois, la variété A est projective et on ne peut pas espérer obtenir une formule valable partout : il faut recouvrir $A\times A$ par des ouverts et donner une expression différente de φ pour chacun des ouverts. Dans le cas des courbes elliptiques, on avait déjà deux formules distinctes pour la somme de deux points différents et le doublement. En genre 2, la quantité de cas à envisager est plus grande (et augmenterait encore en genre supérieur).

Approche de Spallek

Dans sa thèse [Spa94], Spallek a donné de telles formules en caractéristique impaire. Pour les calculer, elle n'a pas vraiment utilisé l'algorithme de Cantor. Son approche consiste à écrire les conditions que doit vérifier la sortie de l'algorithme de composition, puis à résoudre formellement le système ainsi construit de manière à obtenir des formules closes. Il y a différents systèmes à résoudre de manière à traiter tous les branchements possibles. Elle procède ensuite de même pour la réduction.

Pour fixer les idées, nous allons donner le système correspondant au cas générique (le plus fréquent) pour la composition. Soit $\mathcal{C}$ une courbe hyperelliptique de genre 2 d'équation $y^2 = f(x)$. Soient $D = \langle u(x), v(x)\rangle$ et $D' = \langle u'(x), v'(x)\rangle$ deux diviseurs réduits génériques, c'est à dire que $D = P_1 + P_2 - 2\infty$ et $D' = P_1' + P_2' - 2\infty$ où les quatre points P_1, P_2, P_1', P_2' ont des abscisses différentes. Alors le diviseur semi-réduit $R = \langle u^*(x), v^*(x)\rangle$, composition des diviseurs D et D' est le diviseur $R = P_1 + P_2 + P_1' + P_2' - 4\infty$. Ainsi, à partir des coordonnées de D et D' sous forme de Mumford, on peut trouver les coordonnées des points P_i et P_i', puis reconstruire le diviseur R sous forme de Mumford.

On note $u(x) = x^2 + u_1 x + u_0$, $v(x) = v_1 x + v_0$, $P_i = (x_i, y_i)$, et les mêmes notations avec des «primes» pour D' et P_i'. La représentation de Mumford de R est alors donnée par

$$u^*(x) = (x - x_1)(x - x_2)(x - x_1')(x - x_2'),$$

$$v^*(x_i) = y_i \quad \text{et} \quad v^*(x_i') = y_i' \quad \text{pour } i = 1, 2.$$

Ce dernier système se résout facilement, par exemple par les formules d'interpolation de Lagrange. Les formules ainsi obtenues sont en les coordonnées des points. Pour se ramener aux coordonnées de D et D', on a les relations suivantes :

$$u_0 = x_1 x_2, \quad u_1 = -(x_1 + x_2), \quad v_0 = \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2}, \quad v_1 = \frac{y_1 - y_2}{x_1 - x_2},$$

et de même avec les «primes». Les coefficients de $u^*(x)$ et $v^*(x)$ s'expriment comme des fractions rationnelles symétriques en (x_1, y_1) et (x_2, y_2) d'une part, et en (x_1', y_1') et (x_2', y_2') d'autre part. On peut donc les exprimer à l'aide uniquement de u_0, u_1, v_0, v_1, et u_0', u_1', v_0', v_1', et l'on obtient les formules souhaitées pour la composition.

La réduction s'obtient par le même type de calcul. Finalement, Spallek propose une famille de formules couvrant la plupart des cas possibles. Pour une opération dans la Jacobienne, le coût annoncé par Spallek est d'environ 40 multiplications et 1 inversion. Toutefois, les formules qu'elle donne contiennent plutôt une cinquantaine de multiplications et 2 inversions. Quoiqu'il en soit, cela améliore grandement l'algorithme de Cantor.

Formules de Harley

Récemment, Harley [Hara] a amélioré ces formules. Son approche diffère de celle de Spallek : comme dans l'algorithme de Cantor, Harley manipule les polynômes plutôt que les coefficients séparément. Toutefois bon nombre d'opérations de base peuvent être économisées, du fait que l'on sait que certaines divisions sont exactes, que l'on peut appliquer la méthode de Karatsuba plutôt que la multiplication naïve, que certaines parties des polynômes intermédiaires ne sont pas nécessaires. De plus, les pgcd étendus de polynômes sont déroulés et la composition est vue comme un théorème des restes chinois, ou une itération de Newton dans le cas du doublement.

Pour illustrer la stratégie de Harley, nous allons détailler le doublement d'un diviseur générique de poids 2. Les formules finales comprenant tous les cas possibles sont disponibles à l'adresse [Hara].

On part d'un diviseur réduit en représentation de Mumford $D = \langle u(x), v(x) \rangle$, sur une courbe de genre 2 d'équation $y^2 = f(x)$. Le résultat de la composition est le diviseur semi-réduit $[2]D = \langle u'(x), v'(x) \rangle$, où $u'(x) = u(x)^2$, et $v'(x)$ est le polynôme interpolant les points composant D avec multiplicité 2. Plus exactement, cela se traduit par le fait que $u(x)^2$ divise $v'(x)^2 - f(x)$, et que $v'(x) \equiv v(x) \mod u(x)$. On peut donc voir $v'(x)$ comme la racine carrée de $f(x)$ remontée modulo $u(x)^2$ à partir de $v(x)$ qui est une racine de $f(x)$ modulo $u(x)$. C'est pourquoi $v'(x)$ s'obtient par une itération de Newton correspondant à cette racine carrée :

$$v'(x) = v(x) - \frac{f - v^2}{2v} \bmod u(x)^2.$$

On va calculer $v'(x)$ en base $u(x)$: seul le deuxième «chiffre» est inconnu. Soit $k(x)$ tel que $f(x) - v^2(x) = u(x)k(x)$. Ce polynôme est le résultat d'une division exacte, et se calcule donc au prix de 3 multiplications et 1 carré. Le deuxième chiffre de $v'(x)$ est alors $s(x) = \frac{k(x)}{2v(x)} \mod u(x)$, ce qui peut se calculer au prix de 10 multiplications et 1 inversion d'un element r que l'on calcule avec 3 muliplications et 2 carrés. En fait, cet élément r est calculé dès le début, car sa non-nullité traduit le caractère générique du diviseur D. Notons de plus qu'un des 2 carrés nécessaires pour calculer r peut être réutilisé pour le calcul de $k(x)$.

On passe maintenant à la phase de réduction. Dans l'état actuel, on n'a pas vraiment calculé $u'(x) = u(x)^2$, et on connaît $v'(x)$ uniquement en base $u(x)$. D'après l'algorithme de Cantor, la réduction se ramène aux opérations

$$\begin{aligned} U(x) &= \tfrac{f(x)-v'^2(x)}{u^2(x)}, \\ V(x) &= -v'(x) \mod U(x). \end{aligned}$$

La première équation se réécrit

$$U(x) = \frac{(v+su)^2 - f}{u^2} = \frac{(v^2-f) + 2suv + s^2u^2}{u^2} = s^2 - \frac{k-2sv}{u}.$$

Là encore, la division étant exacte, le calcul de $\frac{k-2sv}{u}$ se fait au prix d'une seule multiplication, et le calcul de $U(x)$ nécessite 3 carrés supplémentaires pour le calcul de $s^2(x)$. Pour rendre $U(x)$ unitaire, il faut faire une inversion et deux multiplications. Pour finir, le calcul de $V(x)$ nécessite 6 multiplications.

Au total, 2 inversions, 5 carrés, et 25 multiplications sont nécessaires pour le doublement d'un diviseur. Notons une fois de plus que les nombres d'opérations annoncés à chaque étape ne sont pas immédiats ; par exemple, toute multiplication de polynômes est faite avec l'algorithme de Karatsuba.

Finalement, pour une courbe de genre 2 sur un corps de caractéristique impaire, les formules de Harley fournissent une algorithme dont le coût en nombre d'opérations dans le corps de base est le suivant :

	multiplications	inversions
addition	27	2
double	30	2

Remarque. Il n'est pas certain que le minimum soit atteint par les formules de Harley. Déterminer la meilleure manière d'évaluer une famille de fractions rationnelles en de nombreuses variables est un problème compliqué en général ; une recherche naïve de la meilleure stratégie n'est pas envisageable pour les formules qui nous intéressent.

Remarque. Récemment, pour les courbes de genre supérieur, Nagao [Nag00] a proposé des algorithmes plus efficaces que ceux de Cantor. En particulier, des formules utilisant des coordonnées projectives sont employées afin d'éviter certaines inversions, ce qui peut s'avérer très important pour des raisons théoriques (mise sous forme d'arbre de calcul SLP) ou pratiques (le coût d'une inversion est souvent bien supérieur au coût d'une multiplication).

Chapitre 5

Algorithmes élémentaires pour le calcul de la cardinalité

Pour calculer le cardinal d'un groupe, il existe des méthodes génériques au sens où elles s'appliquent dès que l'on dispose d'un algorithme pour la loi de groupe et de bornes sur la cardinalité. Dans ce chapitre nous rappelons ces méthodes, ainsi que la méthode d'approximation qui consiste à améliorer les bornes au prix d'un précalcul. Nous proposons une petite amélioration de cette approximation, s'appuyant sur l'utilisation du sous-corps réel de l'anneau des endomorphismes.

5.1 Algorithmes génériques

On se fixe un groupe abélien G noté additivement, d'élément neutre noté 0, pour lequel on dispose

1. d'algorithmes efficaces pour additionner et prendre l'opposé ;
2. d'un moyen de tirer des éléments aléatoires uniformément dans G ;
3. de bornes explicites sur l'ordre de G :

$$A \leqslant \mathrm{ord} G \leqslant B.$$

Les algorithmes génériques pour le calcul de l'ordre d'un groupe sont de deux types :

- Algorithme « pas de bébé, pas de géant », dû à Shanks.
- Algorithmes reposant sur le paradoxe des anniversaires, initialement dus à Pollard, et rendus effectifs par van Oorschot et Wiener [vOW99], Stein–Teske [ST99a] et Harley [GH00].

Ces algorithmes requièrent $O(\sqrt{B-A})$ opérations dans le groupe et l'avantage de la deuxième classe d'algorithmes est que l'espace nécessaire peut être rendu arbitrairement petit, contrairement à la méthode de Shanks.

5.1.1 Méthode de Shanks

Cette célèbre méthode « pas de bébé, pas de géant » fut initialement proposée par Shanks dans le contexte des groupes de classes de corps quadratiques imaginaires [Sha71].

Notons N l'ordre du groupe que l'on cherche à déterminer. Soit $w = B - A$ la taille de l'intervalle dans lequel on cherche N. L'idée de Shanks est de chercher l'ordre d'un élément aléatoire x dans G par une méthode astucieuse. On sait que $N \cdot x = 0$. Soit W un paramètre entier dans l'intervalle $[1, w-1]$, on écrit alors

$$N = A + n_0 + n_1 W,$$

avec $0 \leqslant n_0 \leqslant W$ et $0 \leqslant n_1 \leqslant \lceil \frac{w}{W} \rceil$ et l'on a

$$(A + n_0) \cdot x = (-W n_1) \cdot x.$$

L'algorithme consiste à précalculer tous les membres de gauche possibles (les pas de bébés), puis à calculer les membres de droites possibles les uns après les autres en vérifiant à chaque fois si l'on ne retombe pas sur un des éléments déjà calculés.

Algorithme 5.4 Pas de bébé, pas de géant de Shanks
Entrée: Un élément x d'un groupe G, des bornes A et B sur l'ordre de G, un paramètre W.
Sortie: Un multiple de l'ordre de x.

$w \leftarrow B - A$; $y \leftarrow A \cdot x$; $S \leftarrow \{(y, 0)\}$;
Pour ν_0 allant de 1 à W, */* Pas de bébé */*
    $y \leftarrow y + x$;
    Si $y = 0$, alors Retourner $A + \nu_0$;
    $S \leftarrow S \cup \{(y, \nu_0)\}$;
$z \leftarrow -W \cdot x$; $y \leftarrow z$;
Pour ν_1 allant de 1 à $\lceil \frac{w}{W} \rceil$, */* Pas de géant */*
    Si y appartient à S, alors
        $n_0 \leftarrow \nu_0$ correspondant à y dans S ;
        Retourner $A + n_0 + \nu_1 W$;
    $y \leftarrow y + z$;

Une implantation efficace de cet algorithme nécessite une structure de données adéquate pour gérer l'ensemble S. Grâce à des méthodes de hachage performantes, on peut supposer que toute opération se fait en temps constant. Nous n'insisterons pas sur cet aspect.

La valeur de W doit être ajustée de manière à équilibrer les temps d'exécution des deux phases. En première approximation, il faut choisir W de l'ordre de $\sqrt{w}$, mais en fonction de l'implantation et du comportement de la structure S pour l'inclusion et la recherche d'un élément, il est nécessaire d'ajuster cette valeur.

Une fois que l'on a obtenu un multiple de l'ordre de x, on peut obtenir l'ordre exact de x au prix de la factorisation de ce dernier. Si un seul multiple de l'ordre de x se trouve dans l'intervalle $[A, B]$, on en déduit immédiatement l'ordre du groupe G. Toutefois, des cas pathologiques peuvent se produire pour lesquels, même en prenant le ppcm de l'ordre de plusieurs éléments aléatoires, on n'arrive pas à conclure. Il existe une méthode pour traiter tous les cas, dont le principe est de garder en mémoire les sous-groupes déjà calculés. Nous renvoyons à [Coh93, p. 236] (une erreur s'est glissée dans cet algorithme : un errata est disponible sur la page web de l'auteur).

Complexité

On suppose que le coût de la gestion de la structure de données S est négligeable. Le coût des «pas de bébé» est $O(W)$ opérations dans le groupe, et celui des «pas de géant» est $O(w/W)$ opérations. Ainsi, en prenant $W = O(\sqrt{w})$ comme annoncé plus haut, le coût total de l'algorithme est de $O(\sqrt{w}) = O(\sqrt{B-A})$ opérations dans le groupe. L'espace mémoire requis est lui aussi de l'ordre de $O(\sqrt{B-A})$.

C'est ce dernier point qui est le plus vite limitant en pratique. Le but de l'algorithme de la section suivante est d'éviter ce stockage important.

5.1.2 Algorithme «paradoxe des anniversaires»

Nous nous contentons de donner l'idée générale de cette méthode. Pour les détails d'implantation, ainsi que des estimations précises de complexité, nous renvoyons à [ST99a], [GH00], [Harb].

Comme pour la méthode de Shanks, on commence par déterminer un multiple de l'ordre d'un élément aléatoire du groupe. On espère ensuite que cette information suffit pour conclure, grâce aux bornes dont on dispose. Notons qu'à la différence de la méthode de Shanks, il n'existe pas de version de cet algorithme fonctionnant tout le temps : si l'exposant du groupe est trop petit, on ne peut pas espérer déterminer l'ordre du groupe sans utiliser de mémoire.

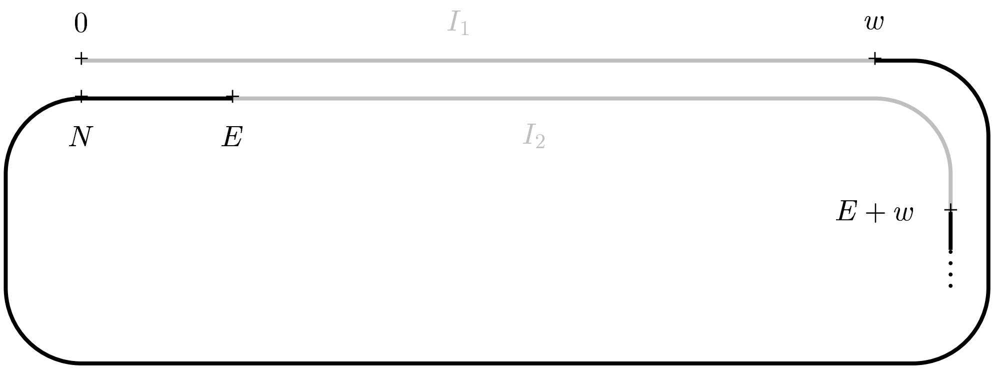
Reprenons les notations précédentes : soit x un élément de G d'ordre N inconnu. On sait que N est dans un intervalle de largeur w. Soit $E = \frac{A+B}{2}$ le centre de l'intervalle. L'ensemble

$$\{i \cdot x,\ i \in \mathbb{N}\}$$

est un ensemble fini, car le sous-groupe engendré par x est cyclique. On définit deux intervalles

$$I_1 = \{i \cdot x,\ i \in [0, w]\} \quad \text{et} \quad I_2 = \{i \cdot x,\ i \in [E, E+w]\}.$$

Les bornes sur N certifient alors que I_1 et I_2 se recouvrent sur au moins une largeur de taille $w/2$. Le principe des algorithmes «paradoxes des anniversaires» est alors de calculer des éléments aléatoires dans I_1 et dans I_2, jusqu'à ce que l'on tombe sur une *collision*. En effet, si l'on a trouvé deux entiers $i_1 \in [0, w]$ et $i_2 \in [E, E+w]$ tels que $i_1 \cdot x = i_2 \cdot x$, alors $i_2 - i_1$ est un entier non nul divisant l'ordre de x.



Ainsi, une première méthode consiste à calculer et stocker des couples $(i_1 \cdot x, i_1)$ d'une part et $(i_2 \cdot x, i_2)$ d'autre part, pour un grand nombre de valeurs aléatoires de $i_1 \in [0, w]$ et $i_2 \in [E, E+w]$, jusqu'à obtenir un point commun entre ces deux ensembles. Ce qui va rendre cette méthode bien plus performante en pratique est l'utilisation d'une marche pseudo-aléatoire de manière à calculer un nouvel élément « aléatoire » au prix d'une unique opération de groupe et surtout, l'utilisation de *points distingués* qui permettent de réduire arbitrairement la taille mémoire requise, au prix d'un surcoût de calcul négligeable.

Ces deux stratégies, marche pseudo-aléatoire et points distingués, seront décrites en détail dans le chapitre 9 dans le contexte du logarithme discret.

Pour l'analyse, nous allons idéaliser la situation et supposer qu'on tire un élément en alternance dans I_1 ou dans I_2, avec une distribution de probabilité uniforme. Une collision dans $J = I_1 \cap I_2$ mène au résultat si et seulement si elle se fait entre deux éléments provenant d'intervalles distincts. On peut donc modéliser le problème ainsi : on dispose d'un ensemble J de taille n dans lequel on tire uniformément des éléments. Lors d'une collision, on a gagné avec probabilité $1/2$. On note t_p le temps d'attente moyen pour obtenir p collisions. Le nombre moyen d'éléments à tirer dans J avant d'avoir gagné est donc

$$\sum_{p=1}^{\infty} \frac{t_p}{2^p}.$$

Le calcul de t_p est du type « paradoxe des anniversaires », le cas classique étant $p = 1$. Dans tous ces calculs, on admet que les temps d'attente sont de l'ordre de $\sqrt{n}$ (ce qui est démontré dans [FS96], pages 424), et nous nous permettons de ne pas étudier précisément les termes d'erreurs dans les approximations.

La probabilité que la première collision se produise à l'étape k est

$$P_k = \frac{k}{n} \prod_{1 \leqslant i \leqslant k-1} \left(1 - \frac{i}{n}\right) \approx \frac{k}{n} e^{-\frac{k^2}{2n}}.$$

D'où

$$t_1 = \sum_{k=1}^{\infty} k P_k \approx \int_0^{\infty} \frac{x^2}{n} e^{-\frac{x^2}{2n}} dx = \sqrt{\frac{\pi n}{2}}.$$

Intuitivement, la deuxième collision se produit après un temps d'attente plus court que deux fois le temps pour une seule collision, car une fois la première collision atteinte, la table est déjà bien remplie. Plus précisément, la probabilité que la deuxième collision se produise à l'étape k est

$$\begin{aligned} Q_k &= \sum_{k' < k} \prod_{i=1}^{k'-1} \left(1 - \frac{i}{n}\right) \frac{k'}{n} \prod_{i=k'+1}^{k-1} \left(1 - \frac{i}{n}\right) \frac{k}{n} \\ &\approx \frac{k}{n} e^{-\frac{k^2}{2n}} \sum_{k' < k} \frac{k'}{n} \\ &\approx \frac{k^3}{2n^2} e^{-\frac{k^2}{2n}}. \end{aligned}$$

D'où un temps d'attente

$$t_2 \approx \sum_{k=1}^{\infty} k Q_k \approx \int_0^{\infty} \frac{x^4}{2n^2} e^{-\frac{x^2}{2n}} dx = \frac{3}{2} \sqrt{\frac{\pi n}{2}}.$$

De manière générale, la probabilité que la p-ième collision se produise à l'étape k est

$$\begin{aligned} R_k(p) &\approx k \sum_{k_1<\cdots<k_{p-1}<k} \frac{k_1 \cdots k_{p-1}}{n^p} e^{-\frac{k^2}{2n}} \\ &\approx \frac{k^{2p-1}}{n^p 2^{p-1}(p-1)!} e^{-\frac{k^2}{2n}}. \end{aligned}$$

D'où un temps d'attente moyen

$$t_p \approx \sum_{k=1}^{\infty} k R_k \approx \int_0^{\infty} \frac{x^{2p}}{n^p 2^{p-1}(p-1)!} e^{-\frac{x^2}{2n}} dx = \frac{(2p)!}{2^{2p-1} p!(p-1)!} \sqrt{\frac{\pi n}{2}}.$$

Pour finir, on a

$$\sum_{p=1}^{\infty} \frac{t_p}{2^p} = \sqrt{\frac{\pi n}{2}} \sqrt{2} = \sqrt{\pi n}.$$

Soit α le coefficient de recouvrement de I_1 et I_2, c'est-à-dire que la taille de J est $n = \frac{w}{\alpha}$. Par construction, on a $\alpha \in [1, 2]$. Le nombre d'éléments à tirer dans I_1 ou dans I_2 avant de trouver le résultat est donc

$$\alpha \sqrt{\pi \frac{w}{\alpha}} = \sqrt{\alpha \pi w}.$$

En général, le cardinal de la Jacobienne est plus proche du centre de l'intervalle de Hasse–Weil que des bords, et α est plus proche de 1 que de 2. Une étude heuristique de la répartition du cardinal dans l'intervalle de recherche se trouve dans [ST00a]. Ainsi, le nombre d'éléments aléatoires à construire est de l'ordre de

$$2\sqrt{w} = 2\sqrt{B - A}.$$

Si, comme évoqué ci-dessus, on utilise une marche pseudo-aléatoire pour construire les éléments, on s'éloigne quelque peu du modèle idéal utilisé pour l'analyse, mais chaque élément coûte une unique opération dans le groupe. En pratique, on constate que la constante 2 est une bonne estimation, même après avoir rajouté la marche pseudo-aléatoire et les points distingués.

Remarque. Notons que cette méthode se parallélise particulièrement bien : si l'on dispose de m machines, le temps de calcul est divisé par m, et la quantité d'informations circulant est très faible. Il est toutefois nécessaire de bien régler les paramètres de la marche aléatoire et des points distingués de manière à obtenir la bonne constante dans le $O()$. Nous renvoyons aux références citées plus haut pour ces détails très importants en pratique.

5.2 Méthodes d'approximation

Cette section concerne les bornes que l'on peut espérer obtenir sur la taille du groupe dans le cas des Jacobiennes de courbes (cf [Elk98]).

5.2.1 Bornes sur les coefficients de $\chi(t)$

Les algorithmes décrits ci-dessus s'appliquent directement au calcul du cardinal de la Jacobienne d'une courbe hyperelliptique. Des premières bornes sur le cardinal sont données par le théorème de Weil : si $\mathcal{C}$ est une courbe de genre g définie sur un corps fini $\mathbb{F}_q$, alors

$$(\sqrt{q} - 1)^{2g} \leqslant \#\mathrm{Jac}(\mathcal{C}) \leqslant (\sqrt{q} + 1)^{2g},$$

ce qui fait un intervalle de largeur $O(q^{g-\frac{1}{2}})$. On obient ainsi les complexités suivantes pour le calcul de la cardinalité en fonction du genre de la courbe

genre	1	2	3	4	5	6	g
complexité	$q^{1/4}$	$q^{3/4}$	$q^{5/4}$	$q^{7/4}$	$q^{9/4}$	$q^{11/4}$	$q^{\frac{2g-1}{4}}$

Ces complexités sont calculées pour un genre fixé, en faisant tendre uniquement q vers l'infini. Asymptotiquement, on peut donner une estimation plus précise de la taille de l'intervalle :

$$w = 4gq^{g-\frac{1}{2}} + O(q^{g-1}).$$

Cette largeur d'intervalle est la taille de référence pour un algorithme en racine carrée.

Revenons au polynôme caractéristique $\chi(t)$ de l'endomorphisme de Frobenius. Sa valeur en 1 donne le cardinal de $\mathrm{Jac}(\mathcal{C})$, toutes ses racines ont valeurs absolues $\sqrt{q}$ et il est de la forme

$$\chi(t) = t^{2g} - s_1 t^{2g-1} + \cdots (-1)^g s_g t^g + (-1)^{g+1} s_{g-1} q t^{g-1} + \cdots - s_1 q^{g-1} t + q^g.$$

On a donc la borne suivante sur ses coefficients :

$$|s_i| \leqslant \binom{2g}{i} q^{i/2}.$$

Ainsi, si l'on connaît les $k-1$ premières valeurs des s_i, on a réduit la taille de l'intervalle à

$$w' = 2\binom{2g}{k} q^{g-\frac{k}{2}} + O(q^{g-\frac{k+1}{2}}). \tag{5.1}$$

5.2.2 Approximation

Il est possible de calculer les premiers coefficients du polynôme $\chi(t)$ de manière indépendante des algorithmes en racine carrée. En effet, ces coefficients sont très liés au nombre de points de la courbe sur le corps de base $\mathbb{F}_q$ ainsi que sur les premières extensions.

Reprenant les notations de la page 19, une petite manipulation de séries donne

$$\log t^{2g}\chi(1/t) = \log L(t) = \sum_{n\geqslant 1} \left(N_n - (q^n+1)\right) \frac{t^n}{n}, \tag{5.2}$$

où N_n est le nombre points de la courbe $\mathcal{C}$ sur $\mathbb{F}_{q^n}$, en tenant compte du point à l'infini.

On obtient ainsi des formules pour calculer de proche en proche les s_i à partir des N_i :

$$\begin{aligned} s_1 &= q+1-N_1, \\ s_2 &= \tfrac{s_1^2+N_2-(q^2+1)}{2}, \\ s_3 &= \tfrac{s_1^3-3s_1s_2-N_3+q^3+1}{3}, \\ &\vdots \end{aligned}$$

Le calcul des N_i se fait par une méthode naïve : pour chaque élément x du corps $\mathbb{F}_{q^i}$ on cherche si l'équation $y^2 + h(x) = f(x)$ de degré 2 en y admet 0, 1 ou 2 solutions dans $\mathbb{F}_{q^i}$. On ajoute ce nombre de solutions à un compteur que l'on a initialisé à 1 pour tenir compte du point à l'infini. Cette méthode nécessite $O(q^i)$ opérations de base et résolutions d'équations quadratiques dans le corps $\mathbb{F}_{q^i}$.

Pour une courbe de genre g, il est donc préférable de calculer quelques-uns des premiers s_i par cette méthode naïve afin de réduire l'intervalle de recherche avant de se lancer dans un algorithme en racine carrée.

Supposons que l'on précalcule les $k-1$ premières valeurs des s_i. La complexité de ce précalcul est $O(q^{k-1})$. Ensuite, d'après l'équation 5.1, la complexité de la deuxième phase en racine carrée est $O(q^{\frac{2g-k}{4}})$. Pour minimiser la complexité totale, on impose

$$k-1 = \frac{2g-k}{4},$$

ce qui conduit à une valeur $k = \frac{2g+4}{5}$, et donc une complexité de

$$O(q^{\frac{2g-1}{5}}).$$

Bien entendu, la valeur de k doit être un entier supérieur ou égale à 1 pour que ce calcul ait un sens ; dans [ST99b], on trouve les formules exactes qui dépendent de la congruence de g modulo 5 .

La constante dans la largeur de l'intervalle, donnée par la formule 5.1, est à première vue égale à $2\binom{2g}{k}$. En fait, la connaissance des s_i jusqu'à l'indice $k-1$ donne des informations sur s_k. La formule 5.2 donne s_k en fonction des s_i d'indice inférieur et de N_k :

$$s_k = \varphi(s_1, \ldots, s_{k-1}) + (-1)^k \frac{N_k - (q^k+1)}{k},$$

où la fonction φ est aisément calculable à partir du développement en série de $\log(1+x)$. Connaissant les valeurs s_i impliquées, la borne de Hasse-Weil permet d'encadrer N_k et donc s_k. La largeur de l'intervalle obtenu est ainsi

$$w'' = \frac{4g}{k} q^{g-\frac{k}{2}} + O(q^{g-\frac{k+1}{2}}).$$

Exemples

Pour les courbes de genre 1 et 2, la complexité des méthodes en racine carrée est inférieure à $O(q)$ qui est le coût minimal d'une approximation. Cette dernière n'est donc pas rentable.

En genre 3, le précalcul du terme s_1 fait baisser la complexité totale du calcul du nombre de points de $O(q^{5/4})$ à $O(q)$. Le tableau suivant donne les différentes complexités pour les différents niveaux d'approximation :

niveau d'approximation k	1	2	3
coût approx + coût racine carrée	$0 + q^{5/4}$	$q + q$	$q^2 + q^{3/4}$

Produit Eulérien tronqué

Une autre manière d'attaquer le problème de l'approximation, spécifique au cas hyperelliptique, est d'utiliser l'écriture de la fonction Zêta comme un produit Eulérien (cf page 19). Cette méthode, due à Stein et Williams consiste à transférer cette écriture en une écriture de $\chi(t)$ comme un produit Eulérien que l'on peut alors tronquer pour estimer sa valeur en 1. Les bornes obtenues sont essentiellement les mêmes que par le calcul ci-dessus.

Nous renvoyons à [SW99] et [ST99b] pour une description précise de cette approche. Notons qu'en utilisant l'écriture Eulérienne à l'aide du caractère quadratique associé à la courbe hyperelliptique, les auteurs peuvent appliquer une méthode de crible très efficace pour accélérer la phase d'approximation.

5.2.3 Utilisation du sous-corps réel

Reprenons le cas du genre 3, qui est le premier pour lequel l'approximation est rentable. Le calcul de N_1 donne s_1, et les bornes sur N_2 donnent des bornes sur s_2. On peut en fait obtenir un peu plus d'information en utilisant le fait que le polynôme $\chi(t)$ doit définir un corps à multiplication complexe.

On suppose que la Jacobienne de $\mathcal{C}$ est simple, donc que le polynôme $\chi(t)$ est irréductible. Si ce n'est pas le cas, certaines des inégalités écrites ci-dessous doivent être prises au sens large.

On calcule facilement que le sous-corps totalement réel du corps à multiplication complexe est défini par le polynôme

$$P(X) = X^3 - s_1X^2 + (s_2 - 3q)X - (s_3 - 2qs_1).$$

Une condition nécessaire sur $P(X)$ pour qu'il définisse une extension totalement réelle est que son polynôme dérivé $P'(X)$ admette deux racines réelles, donc ait un discriminant positif. On obtient ainsi la condition suivante :

$$s_1^2 - 3s_2 + 9q > 0,$$

qui se réécrit en

$$N_2 - (q^2 + 1) < 6q - \frac{s_1^2}{3}.$$

On obtient donc les bornes suivantes sur $s_2 - \frac{s_1^2}{2} = \frac{1}{2}(N_2 - (q^2 + 1))$:

$$-3q < s_2 - \frac{s_1^2}{2} < 3q - \frac{s_1^2}{6},$$

la minoration provient simplement de la borne de Hasse-Weil, comme dans l'approximation classique, et la majoration, due à la présence du sous-corps réel est meilleure.

La qualité de cette nouvelle borne est variable selon la valeur obtenue pour s_1 lors du précalcul. Si s_1 est nul, on retombe sur la borne de Hasse-Weil. Si par contre $|s_1|$ est grand, on peut réduire l'intervalle jusqu'à un facteur 2, ce qui se traduit par un gain d'un facteur $\sqrt{2}$ pour l'algorithme en racine carrée, soit environ 29%. Ces cas extrêmes pour s_1 sont relativement rares : ils correspondent aux courbes ayant un grand nombre de points ou un petit nombre de points.

5.3 Estimation de temps de calcul pour les tailles cryptographiques

Nous allons prendre l'hypothèse de travail que l'on peut effectuer 10^{10} opérations dans le groupe par jour. Cela correspond à ce que Harley a pu obtenir grâce à ses formules optimisées dans le cas d'une courbe de genre 2, sur une station de travail Alpha, 500MHz. Dans le cas du genre supérieur, on doit perdre un facteur constant, que l'on va négliger, le but étant de donner des ordres de grandeur.

On suppose de plus que les phases d'approximation éventuelle coûtent moins que la phase en racine carrée, pourvu que la complexité soit au plus la même.

Sous ces conditions, on obtient les temps de calculs suivants :

$\#\mathrm{Jac} \setminus g$	2	3	4
10^{30}	100 jours	5 jours	70 jours
10^{40}	1500 ans	30 ans	1100 ans
10^{50}	9 M ans	60000 ans	6 M ans

En cryptographie, on s'intéresse à des groupes de taille au minimum 10^{40}, et la taille conseillée en général est plutôt 10^{50}. On constate ainsi que même en parallèlisant le calcul, il n'est pas raisonnable d'espérer calculer jusqu'à des tailles cryptographiques par les méthodes en racine carrée, même aidées de l'approximation.

Remarque. Lorsque l'on travaille avec un modèle réel de la courbe, i.e. avec deux points à l'infini, la structure est plus compliquée : on n'a plus un groupe, mais une *infrastructure*, c'est-à-dire que chaque classe d'idéaux est un cycle d'idéaux réduits. On peut alors définir la multiplication et réduction d'idéaux classique, qui coûte en pratique le même nombre d'opérations que la composition et la réduction de diviseurs à la Cantor dans le modèle imaginaire. Mais on dispose aussi d'une opération plus simple qui consiste à avancer dans un cycle en restant dans la même classe. Cette opération coûte environ $4g$ fois moins d'opérations dans le corps que l'opération de base sur les idéaux. En tirant parti de cette propriété, on peut accélérer le calcul de cardinalité d'un facteur $2\sqrt{g}$ par rapport à une courbe dans un modèle imaginaire. Nous renvoyons à [ST00b] pour un exposé précis de cette approche.

Notons que pour le genre 2, les formules ne sont pas autant optimisées dans le cas réel que dans le cas imaginaire, et le facteur $2\sqrt{2}$ théorique ne se retrouve pas en pratique.

Remarque. Une fois connu le cardinal de la Jacobienne, une question naturelle est sa structure en tant que groupe. La première étape est de factoriser le cardinal, on ne peut donc pas rêver de le faire en temps polynomial. Une fois cette étape franchie, calculer la structure est une tâche faisable si le groupe est cyclique ou presque cyclique, ce qui est très probable pour une courbe aléatoire. Toutefois, dans le cas d'un groupe comprenant au moins deux gros facteurs cycliques en somme directe, la détermination de la structure peut s'avérer délicate, et d'autant plus si l'on demande des générateurs. Pour des considérations sur la difficulté de ce problème, nous renvoyons à [OS92]. Dans cet article il est expliqué comment on peut tirer parti du couplage de Weil dans ce contexte (voir aussi page 141).

Chapitre 6

Cardinalité de courbes particulières

6.1 Courbes de Koblitz et courbes à multiplication complexe

Cette première section rappelle brièvement deux cas connus pour lesquels le calcul du nombre de points peut être accéléré. La situation la plus simple est lorsque la courbe est définie sur un corps fini petit, mais qu'elle est étudiée sur une extension ; comme dans le cas des courbes elliptiques « de Koblitz », il est alors aisé de calculer la cardinalité. Le second cas concerne les courbes à multiplication complexe. Là encore, cela se passe comme pour les courbes elliptiques, mais de nombreuses complications sont à prévoir.

6.1.1 Calcul de la cardinalité par le théorème de Weil

Soit $\mathcal{C}$ une courbe de genre g définie sur le corps fini $\mathbb{F}_q$. Comme expliqué dans le chapitre 1, le polynôme caractéristique du Frobenius, et donc aussi le cardinal de la Jacobienne peuvent être déterminés à partir de la connaissance du nombre de points sur la courbe définis sur les corps $\mathbb{F}_{q^i}$ pour i allant de 1 à g. Les valeurs de q et g sont supposées suffisamment petites pour que ce calcul soit faisable. Le polynôme $\chi(t)$ est alors obtenu sous la forme

$$\chi(t) = \sum_{i=0}^{2g} a_i t^i,$$

que l'on peut réécrire en faisant apparaître ses racines :

$$\chi(t) = \prod_{i=0}^{2g} (t - \alpha_i).$$

Soit n un entier positif et soit $\mathcal{C}_n$ la courbe $\mathcal{C}$ considérée comme une courbe sur $\mathbb{F}_{q^n}$. Le théorème de Weil donne alors facilement le polynôme caractéristique $\chi_n(t)$ du Frobenius sur la Jacobienne de la courbe $\mathcal{C}_n$, en fonction de $\chi(t)$, celui correspondant à la courbe $\mathcal{C}$: il est obtenu en élevant toutes les racines à la puissance n :

$$\chi_n(t) = \prod_{i=0}^{2g} (t - \alpha_i^n).$$

En pratique, pour calculer $\chi_n(t)$, on commence par déterminer des valeurs approchées des racines α_i de $\chi(t)$. Puis on élève celles-ci à la puissance n, et on reconstruit $\chi_n(t)$. Les coefficients exacts de ce polynôme sont obtenus en arrondissant les coefficients approchés à l'entier le plus proche. Si la précision est suffisante, on obtient ainsi le cardinal de la Jacobienne de $\mathcal{C}_n$ de manière très rapide.

Si l'on désire faire tous les calculs intermédiaires de manière exacte, il existe des formules de récurrence permettant de calculer les $\chi_n(t)$ sans faire appel au calcul des racines complexes (cf [Lan]).

6.1.2 Utilisation de la multiplication complexe

L'idée est de réduire une courbe définie sur un corps de nombres modulo un idéal premier. Pour cela, le théorème suivant est très utile :

Théorème 6.1 *Soit A une variété abélienne définie sur un corps de nombres K. Alors pour presque tout idéal premier $\mathfrak{p}$ de l'anneau des entiers de K,*

1. *la variété abélienne A se réduit modulo $\mathfrak{p}$ en une variété abélienne $\overline{A}$ définie sur le corps résiduel $\mathbb{F}_q$ de $\mathfrak{p}$,*
2. *on a l'homomorphisme injectif d'anneau* $\mathrm{End}(A) \to \mathrm{End}(\overline{A})$,
3. *cet homomorphisme conserve les degrés des isogénies.*

Un idéal premier pour lequel le théorème est vérifié sera dit *de bonne réduction.*

La démonstration de ce résultat se trouve dans [ST61] : c'est la proposition 12, page 95 utilisée avec la proposition 25, page 109.

Remarque. La condition 1 contient le fait que la loi de groupe reste bien définie modulo $\mathfrak{p}$. D'autre part, la définition précise de « presque tout idéal » est donnée en page 101 du même ouvrage.

Définition 6.1 *Un corps k est dit à* multiplication complexe *s'il est une extension imaginaire d'un corps de nombres totalement réel.*

En anglais, on abrège en « CM field », et souvent on dit aussi « corps CM » en français.

Définition 6.2 *Une courbe de genre g est à multiplication complexe si l'anneau d'endomorphismes de sa Jacobienne est un ordre dans un corps CM.*

Dans le cas des courbes elliptiques, l'utilisation de la multiplication complexe pour calculer la cardinalité est bien connue [Mor91, Kob92, Miy93, LZ94, CTT94]. Les objets en jeu sont les ordres dans les corps quadratiques imaginaires. Dans le cas général, le théorème de Weil donne la proposition suivante :

Proposition 6.1 *Soit $\mathcal{C}$ une courbe de genre g sur un corps de nombres K, ayant multiplication complexe par un corps k, et soit $\mathfrak{p}$ un idéal premier de bonne réduction. Soit φ l'homomorphisme de $End_0(\mathrm{Jac}(\bar{\mathcal{C}}))$ vers K, et soit π le Frobenius sur $\mathrm{Jac}(\bar{\mathcal{C}})$ que l'on suppose simple. Alors $\varphi(\pi)$ est un élément de norme q^g ayant tous ses conjugués de valeur absolue $\sqrt{q}$. Un tel élément de k est appelé nombre de Weil.*

En utilisant cette proposition il est possible de construire des courbes sur un corps fini pour lesquelles le cardinal de la Jacobienne est facile à calculer. Le principe est le suivant. Partant d'une courbe $\mathcal{C}$ sur un corps de nombres K ayant multiplication complexe par un corps k connu, on s'arrange pour construire un nombre de Weil π de k de norme q^g, correspondant à un idéal premier $\mathfrak{p}$ connu. Une fois ce nombre trouvé, son polynôme caractéristique donne le nombre de points de la Jacobienne de la courbe réduite, aux problèmes de tordues près.

Les principales difficultés de cette technique sont d'une part de construire des courbes sur des corps de nombres ayant multiplication complexe, et d'autre part de trouver rapidement des nombres de Weil.

Ceci a été largement étudié et mis en application par Spallek dans sa thèse [Spa94]. On consultera aussi [vW99] et par exemple [CMNT97].

6.2 Courbes à multiplication réelle

Nous nous intéressons maintenant au cas où l'on dispose de moins d'information sur l'anneau d'endomorphismes. Dans le cas du genre 2, le calcul du cardinal de la Jacobienne pour laquelle l'anneau d'endomorphismes est partiellement connu peut être accéléré par rapport aux méthodes génériques. Cette section décrit un algorithme en $O(\sqrt{q})$ pour les courbes à multiplication réelle, à comparer avec la complexité $O(q^{3/4})$ dans le cas général.

6.2.1 Définition d'une courbe RM

Une courbe est à multiplication réelle si son anneau d'endomorphismes est un ordre dans un corps réel. Plus précisément on a la définition suivante.

Définition 6.3 *Soit $\mathcal{C}$ une courbe de genre g définie sur un corps de nombres K dont la Jacobienne est absolument simple. Soit k_0 un corps de nombres totalement réel de degré g. La courbe $\mathcal{C}$ est dite* à multiplication réelle *par le corps k_0 si l'anneau d'endomorphismes de* $\mathrm{Jac}(\mathcal{C})$ *est un ordre dans k_0.*

Avec cette définition, toute courbe elliptique n'ayant pas multiplication complexe est à multiplication réelle par $\mathbb{Q}$. Bien entendu c'est en genre supérieur que cela devient intéressant.

Si l'on considère une courbe elliptique sur un corps fini, il est impossible que son anneau d'endomorphismes soit seulement $\mathbb{Z}$ car le Frobenius vient perturber les choses. En genre supérieur, cela se généralise avec la multiplication réelle.

Lemme 6.1 *Soit $\mathcal{C}$ une courbe de genre g sur un corps fini $\mathbb{F}_q$ dont la Jacobienne est absolument simple. Alors son anneau d'endomorphismes contient un ordre dans un corps à multiplication complexe de degré $2g$. En particulier, ce n'est jamais un ordre dans un corps totalement réel.*

Démonstration. L'endomorphisme de Frobenius $\pi : x \mapsto x^q$ a un polynôme caractéristique $\chi(t)$ de degré $2g$ à coefficients entiers. Ce polynôme est irréductible sur $\mathbb{Q}$ car sinon la Jacobienne se décomposerait. Soit k le corps de nombres $\mathbb{Q}[t]/(\chi(t))$. Les conjectures de Weil impliquent que k est une extension imaginaire d'un corps totalement réel k_0 de degré g. Donc $\mathcal{C}$ est à multiplication complexe par un ordre de k. □

Le lemme suivant précise le comportement d'une courbe à multiplication réelle lorsqu'on la réduit modulo un idéal premier.

Lemme 6.2 *Soit $\mathcal{C}$ une courbe sur un corps de nombres K à multiplication réelle par k_0, et soit $\mathfrak{p}$ un idéal premier de bonne réduction pour* $\mathrm{Jac}(\mathcal{C})$. *Alors la Jacobienne de la courbe réduite $\overline{\mathcal{C}}$ est soit décomposée, soit à multiplication complexe par un corps contenant k_0.*

Démonstration. Le théorème 6.1 assure que $\mathrm{End}_0(\mathrm{Jac}(\overline{\mathcal{C}}))$ contient k_0. Si la Jacobienne est simple, alors elle contient de plus un corps k à multiplication complexe. Ce corps contient donc un sous-corps réel de degré g qui ne peut-être que k_0. □

6.2.2 Calcul de la cardinalité en genre 2

Pour simplifier, $\mathcal{C}$ désigne non plus la courbe sur le corps de nombres (qui sera maintenant notée $\tilde{\mathcal{C}}$) mais la courbe réduite ; K désignera toujours un corps de nombres.

Soit $\tilde{\mathcal{C}}$ une courbe de genre 2 sur K à multiplication réelle par un corps $k_0 = \mathbb{Q}(\sqrt{d})$, et soit $\mathfrak{p}$ un idéal premier de bonne réduction. On note $\mathcal{C}$ la réduction de $\tilde{\mathcal{C}}$ définie sur le corps résiduel $\mathbb{F}_q$ et on se pose la question de calculer $\#\mathrm{Jac}(\mathcal{C})/\mathbb{F}_q$. On suppose que $\mathrm{Jac}(\mathcal{C})$ est absolument simple.

Le polynôme caractéristique de l'endomorphisme de Frobenius sur $\mathcal{C}$ est de la forme

$$\chi(t) = t^4 - s_1 t^3 + s_2 t^2 - s_1 q t + q^2,$$

où $|s_1| \leqslant 4\sqrt{q}$ et $|s_2| \leqslant 6q$. Calculer le cardinal de la Jacobienne revient à trouver s_1 et s_2. Or $\chi(t)$ est un polynôme irréductible qui doit définir un corps de nombres k à multiplication complexe admettant k_0 comme sous-corps totalement réel.

Ce sous-corps réel est en fait engendré par l'endomorphisme $\pi + \bar{\pi}$, où π est une racine de $\chi(t)$. Le polynôme caractéristique de $\pi + \bar{\pi}$ est

$$\begin{aligned} P(t) &= t^2 - \mathrm{Tr}(\pi + \bar{\pi})t + \mathrm{N}(\pi + \bar{\pi}) \\ &= t^2 - s_1 t + (s_2 - 2q) \end{aligned}$$

dont le discriminant est

$$\mathrm{disc}(P(t)) = s_1^2 - 4s_2 + 8q > 0.$$

Par construction de la courbe $\mathcal{C}$, on sait que $\mathbb{Q}(t)/(P(t)) = k_0 = \mathbb{Q}(\sqrt{d})$. Donc le discriminant de $P(t)$ est égal à d à un facteur carré près. Ainsi il existe un entier n tel que

$$s_1^2 - 4s_2 + 8q = n^2 d,$$

et l'on peut borner n grâce aux bornes sur s_1 et s_2. On trouve

$$n \in \{1, \ldots, \sqrt{48q/d}\}.$$

L'algorithme de calcul du nombre de points procède alors comme suit : au lieu de chercher s_1 et s_2 comme au chapitre 5, il est plus efficace de chercher s_1 et n. À chaque couple de valeurs de s_1 et de n correspond une expression pour le polynôme $\chi(t)$, et donc un candidat pour $\#\mathrm{Jac}(\mathcal{C}) = \chi(1)$ que l'on teste sur un diviseur aléatoire. En organisant bien les calculs, le nombre d'opérations à effectuer est en $O(\sqrt{q})$.

Plus précisément, soit D un diviseur de $\mathrm{Jac}(\mathcal{C})$. Son ordre divise $\chi(1)$, donc aussi $4\chi(1)$. On obtient donc

$$[4(q^2+1) + 4s_2 - 4s_1(q+1)]D = 0,$$

ce que l'on réécrit en remplaçant $4s_2$ par sa valeur en termes de n et s_1 :

$$[4(q^2+2q+1)-n^2d]D = [4(q+1)s_1 - s_1^2]D,$$

puis, en simplifiant :

$$[n^2d]D = [(s_1-2(q+1))^2]D.$$

Dans un premier temps, on précalcule tous les termes de gauche pour toutes les valeurs de n dans $\{1,\dots,\sqrt{48q/d}\}$. Ensuite on calcule le terme de droite pour toutes les valeurs de s_1 dans l'intervalle $\{-4\sqrt{q},\dots,4\sqrt{q}\}$, et lorsqu'il y a égalité, on trouve un candidat pour l'ordre de D et donc un candidat pour $\#\mathrm{Jac}(C)$. Avec une grande probabilité, un seul candidat subsistera après plusieurs choix pour le diviseur D.

Pour évaluer tous les termes de gauche, on utilise l'égalité triviale $(n+1)^2 = n^2+2n+1$ de manière à calculer ceux-ci de proche en proche au prix d'un nombre fini d'additions dans la Jacobienne à chaque étape. La même astuce peut être appliquée au membre de droite. Nous renvoyons à [Knu98, p. 488], pour plus de détails sur cette stratégie d'évaluation d'un polynôme sur une progression arithmétique. Finalement, la complexité de cette méthode est de $O(\sqrt{q})$ opérations dans la Jacobienne.

Algorithme 6.5 CARDINALITÉ D'UNE COURBE RM
Entrée: Une courbe $\mathcal{C}$ sur K de genre 2 ayant multiplication réelle par $\mathbb{Q}(\sqrt{d})$, et un idéal premier $\mathfrak{p}$ de bonne réduction.
Sortie: Une liste d'entiers contenant le cardinal de $\mathrm{Jac}(\mathcal{C})$ sur le corps résiduel $\mathbb{F}_q$.

1. $D \leftarrow$ diviseur aléatoire sur $\mathrm{Jac}(\mathcal{C})/\mathbb{F}_q$; $T \leftarrow \{\}$; $R \leftarrow \{\}$;
2. Pour n allant de 1 à $\sqrt{48q/d}$, faire
3. $\quad E \leftarrow [n^2d]D$;
4. $\quad$ Si $E = 0$, alors $R \leftarrow R \cup \{n^2d\}$;
5. $\quad$ Sinon $T \leftarrow T \cup \{(E,n)\}$;
6. Pour s_1 allant de $-4\sqrt{q}$ à $4\sqrt{q}$, faire
7. $\quad F \leftarrow [(s_1-2(q+1))^2]D$;
8. $\quad$ Si $F = 0$, alors $R \leftarrow R \cup \{(s_1-2(q+1))^2\}$;
9. $\quad$ Sinon, si F appartient à T, alors
10. $\qquad N \leftarrow n$ correspondant à F dans T ;
11. $\qquad s_2 \leftarrow (s_1^2 + 8q - N^2d)$ div 4 ;
12. $\qquad R \leftarrow R \cup \{q^2+1+s_2-s_1(q+1)\}$;
13. Retourner R.

6.2.3 Courbes RM dans la littérature

Appliquer l'algorithme ci-dessus suppose la connaissance de courbes sur $\mathbb{Q}$ ayant multiplication réelle. Des exemples de telles courbes sont données par les courbes *modulaires.*

Définition 6.4 *Une courbe $\mathcal{C}$ sur $\mathbb{Q}$ est dite* modulaire *si sa Jacobienne est isogène à un quotient de la Jacobienne de $X_0(N)$. L'entier N est appelé le* niveau *de la courbe $\mathcal{C}$.*

Théorème 6.2 *Toute courbe modulaire est à multiplication réelle.*

La preuve de ceci se trouve dans [Shi71].

Ce théorème s'applique en fait à toute variété abélienne isogène à un quotient de la Jacobienne de $X_0(N)$, même si elle n'est pas elle-même une Jacobienne de courbe. La réciproque de ce théorème est en fait une version étendue de la conjecture de Shimura–Taniyama–Weil (qui est désormais prouvée dans le cas elliptique).

Conjecture 6.1 (Shimura–Taniyama–Weil généralisée) *Toute variété abélienne à multiplication réelle est modulaire.*

Une première voie pour construire des courbes ayant multiplication réelle est donc de partir d'une courbe $X_0(N)$ pour un entier N fixé, de décomposer sa Jacobienne en produits de variétés abéliennes simples, puis de chercher parmi ces quotients quels sont ceux qui sont des Jacobiennes de courbes. Ensuite il reste le problème de calculer effectivement ces courbes, ainsi que le corps à multiplication réelle associé.

Cette construction due à Shimura [Shi71] a été rendue effective par les élèves du Professeur Frey : Wang, Basmaji, Weber, Müller [Wan95, Bas96, Web97, FM98]. Il est désormais possible d'effectuer toutes les tâches énumérées ci-dessus, au moins dans le cas où les courbes sont hyperelliptiques de genre au plus 5.

Remarque. Le problème de décider si une variété abélienne principalement polarisée est la Jacobienne d'une courbe, connu sous le nom de problème de Schottky n'est pas résolu dans le cas général. Toutefois, dans le cas hyperelliptique on dispose d'un critère effectif. Ce critère se traduit par l'annulation de certaines Theta-Constantes (cf page 57), et est assez simple en pratique. Nous renvoyons à [Web97] pour une description complète.

Une autre réponse partielle au problème de Schottky est donnée par Arita [Ari00] : il fournit un algorithme qui « peut trouver » une courbe C_{ab} dont la Jacobienne est une variété abélienne donnée. Cette méthode est heuristique, mais fonctionne en pratique dans certains cas.

Exemple : Dans [FM98] on trouve parmi d'autres la courbe de genre 2

$$y^2 = x^5 - x^4 + x^3 + x^2 - 2x + 1,$$

de niveau 188, ayant multiplication réelle par $\mathbb{Q}(\sqrt{5})$.

On trouve par ailleurs dans la littérature d'autres courbes à multiplication réelle [Mes91b, Ben99]. Je remercie aussi Müller pour m'avoir indiqué les travaux de Brumer [Bru95] où l'on trouve la famille de courbes hyperelliptiques $\mathcal{C}_{b,c,d}$ définies par

$$y^2 + (x^3 + x + 1 + c(x^2 + x))y = b + (1 + 3b)x + (1 - bd + 3b)x^2 + (b - 2bd - d)x^3 - bdx^4,$$

qui sont à multiplication réelle par $\mathbb{Q}(\sqrt{5})$.

6.2.4 Exemples numériques

Nous avons implantée une version non optimisée de notre algorithme 6.5.

Sur le corps à $p = 1000003$ éléments, on considère la courbe de Brumer pour $b = 919713$, $c = 357376$, $d = 853160$. Après une petite transformation on peut la mettre sous la forme

$$y^2 = x^5 + 852905x^4 + 341672x^3 + 188213x^2 + 907556x + 857756.$$

Avec $d = 5$, l'algorithme donne

$$s_1 = -1624, \quad s_2 = 2482630 \quad \text{et} \quad \#\mathrm{Jac} = 1001632489136.$$

Nous avons vérifié que $[1001632489136].D = 0$ pour plusieurs diviseurs D aléatoires.

Un exemple un peu plus grand, avec la courbe de [FM98] : sur le corps à $p = 10^{11} + 3$ éléments, la courbe $y^2 = x^5 - x^4 + x^3 + x^2 - 2x + 1$ a un polynôme caractéristique donné par l'algorithme 6.5 avec $d = 5$:

$$s_1 = -69353 \quad \text{et} \quad s_2 = -32400572093$$

Et donc $\#\mathrm{Jac} = 10000006935867599705329$.

Nous avons de plus vérifié que le nombre d'opérations effectuées dans la Jacobienne est en $O(\sqrt{p})$, ce qui est meilleur que $O(p^{3/4})$ par la méthode élémentaire.

Remarque. Lors de la construction des courbes modulaires, il est expliqué dans [FM98] comment construire le polynôme caractéristique de l'opérateur de Hecke de degré p pour un nombre premier p quelconque. Cela se fait au prix de $O(p)$ opérations élémentaires. Ceci permet en particulier de déduire le polynôme caractéristique du Frobenius sur une Jacobienne quotient d'une variété $J_0(N)$. Un point particulièrement intéressant de cette approche est que la complexité ne dépend pas du genre de la courbe. Pour le genre 2, cela n'est pas rentable car les méthodes élémentaires sont plus rapides (et notre algorithme encore plus), mais à partir du genre 3, cela s'avère compétitif. On peut se demander dans quelle mesure on peut étendre notre algorithme utilisant le sous-corps réel à des courbes de genre supérieur, et par exemple si la complexité de $O(\sqrt{p})$ se conserve en genre 3. Jusqu'à présent, nous ne sommes pas parvenu à traiter ce cas. Le principal obstacle est que la théorie des corps cubiques est beaucoup moins simple que celle des corps quadratiques.

6.3 Opérateur de Cartier-Manin

Le but de cette section est la description d'un algorithme qui permet de calculer le cardinal de la Jacobienne d'une courbe hyperelliptique de genre quelconque modulo la caractéristique du corps de définition. La complexité est telle que cela ne fonctionne que lorsque la caractéristique est petite, et l'information donnée n'est donc que partielle. Cette méthode est un résultat obtenu conjointement avec R. Harley.

6.3.1 Matrice de Hasse–Witt et théorème de Manin

En 1957, Cartier [Car57] a défini un opérateur sur les formes différentielles d'une courbe de genre g définie sur un corps de caractéristique $p \neq 0$. Dans une base bien choisie cet opérateur est en fait représenté par la matrice de Hasse–Witt, matrice qui avait déjà été étudiée quelques décennies auparavant [HW36]. Dans [Man65], Manin a par la suite montré que cet opérateur était étroitement lié au comportement de l'endomorphisme de Frobenius sur la Jacobienne de la courbe. C'est ce résultat qui est appliqué ici au cas des courbes hyperelliptiques tel que décrit dans [Yui78].

Le théorème suivant donne l'expression de la matrice de Hasse–Witt à partir de l'équation d'une courbe hyperelliptique. Du point de vue pratique, on peut prendre ceci comme une définition.

Théorème 6.3 *Soit $y^2 = f(x)$ avec $\deg f = 2g + 1$ l'équation d'une courbe hyperelliptique de genre g en caractéristique $p \neq 2$. Soit c_i le coefficient en x^i du polynôme $f(x)^{(p-1)/2}$. Alors la matrice de Hasse-Witt est donnée par*

$$A = (c_{ip-j})_{1 \leqslant i,j \leqslant g}.$$

Pour une matrice $A = (a_{ij})$, on notera $A^{(p)}$ l'élévation terme à terme à la puissance p de A, c'est-à-dire la matrice (a_{ij}^p).

Théorème 6.4 *Soit $\mathcal{C}$ une courbe de genre g définie sur un corps fini $\mathbb{F}_{p^n}$. Soit A la matrice de Hasse-Witt de $\mathcal{C}$ et soit $A_\pi = AA^{(p)} \cdots A^{(p^{n-1})}$. Soit $\kappa(t)$ le polynôme caractéristique de la matrice A_π, et $\chi(t)$ le polynôme caractéristique de l'endomorphisme de Frobenius. Alors*

$$\chi(t) \equiv (-1)^g t^g \kappa(t) \mod p.$$

Ainsi le calcul de $\chi(t)$ modulo p est immédiat, dès que l'on a calculé la matrice de Hasse–Witt, ce qui se fait en élevant $f(x)$ à la puissance $(p-1)/2$. Cette dernière opération nécessite $O(M(gp) \log p)$ opérations dans le corps de base, et ne peut donc être faite que pour de relativement petites valeurs de p.

6.3.2 Exemples numériques

Soit $\mathcal{C}$ la courbe définie par

$$y^2 = x^5 + ux^4 + u^2x + 1,$$

sur le corps $\mathbb{F}_{3^4} = \mathbb{F}_3[u]/(u^4 + u^2 + u + 1)$. Pour $p = 3$, élever à la puissance $\frac{p-1}{2}$ est immédiat, et l'on obtient la matrice de Hasse–Witt suivante :

$$A = \begin{pmatrix} 0 & 1 \\ u^2 & u \end{pmatrix}.$$

On calcule ensuite sa matrice « norme » :

$$A_\pi = \begin{pmatrix} u^3 + 2u + 1 & 2u^3 + u^2 + 2 \\ u^3 + 2u^2 + u + 2 & 2u^3 + u \end{pmatrix},$$

et son polynôme caractéristique est $X^2 + 2X + 1$. On en déduit donc

$$\chi(t) \equiv t^4 + 2t^3 + t^2 \mod 3,$$

et $\#\mathrm{Jac}(\mathcal{C}) \equiv 1 \mod 3$.

Un exemple un peu plus grand :

$$y^2 = x^5 + ux^4 + u^2x + 1,$$

sur le corps $\mathbb{F}_{p^5} = \mathbb{F}_p[u]/(u^5 + u + 11)$, où $p = 1000003$. Cette fois-ci, l'élévation à la puissance $\frac{p-1}{2}$ est coûteuse : 43 heures avec Magma sur un Pentium 450 MHz. On obtient

$$\chi(t) \equiv t^4 + 29948t^3 + 110573t^2 \mod p,$$

et $\#\mathrm{Jac}(\mathcal{C}) \equiv 140522 \mod p$.

Chapitre 7

Algorithme de Schoof en genre 2

Dans les chapitres précédents nous avons vu des méthodes pour calculer l'ordre de la Jacobienne d'une courbe. Toutefois, soit elles ne s'appliquent que dans des cas très particuliers, soit elles ont une complexité exponentielle. Le but de ce chapitre est de décrire un algorithme inspiré de ceux de Pila et Kampkötter qui s'exécute en temps polynomial en la taille du corps. En théorie il fonctionne pour des courbes hyperelliptiques de genre quelconque, toutefois nous insisterons sur le cas du genre 2 pour lequel les polynômes de division de Cantor permettent de rendre praticable cet algorithme.

7.1 Algorithme de Schoof–Pila–Kampkötter

7.1.1 Historique

En 1985, Schoof [Sch85] a proposé un algorithme pour calculer le cardinal d'une courbe elliptique sur un corps fini fonctionnant en temps *polynomial déterministe*. Quinze ans plus tard, après les améliorations théoriques et pratiques d'Atkin, Elkies, Couveignes, Morain, Lercier, Müller, Dewaghe, il est désormais possible de compter le nombre de points de courbes elliptiques définies sur un corps fini premier ayant jusqu'à environ 10^{500} éléments et en caractéristique deux [2] jusqu'à 2^{1999}. Concernant les courbes plus générales, la question est résolue pareillement en théorie.

En 1990, Pila [Pil90] a publié un algorithme pour calculer le polynôme caractéristique du Frobenius sur une variété abélienne définie sur un corps fini. Cet algorithme est très théorique, mais fonctionne en temps polynomial déterministe. Du moins il est polynomial en la taille du corps fini, la dépendance en la dimension étant nettement moins bonne. L'algorithme prend en entrée une variété abélienne donnée par un système d'équations homogènes dans un espace projectif, ainsi qu'un ensemble de fractions rationnelles définissant la loi de groupe sur un recouvrement de l'espace par des ouverts. Si l'on veut l'appliquer à la Jacobienne d'une courbe, il faut donc commencer par trouver ces équations. Pour les courbes hyperelliptiques de genre 2, Flynn [Fly90] a construit celles-ci : la Jacobienne se plonge dans un espace projectif de dimension 15 et est décrite par 72 équations de degré 2. En d'autres termes, l'algorithme de Pila a un temps d'exécution polynomial, mais dès le genre 2, l'exposant est extrêmement élevé.

Durant la même période, Kampkötter [Kam91] s'est attaché à construire un algorithme spécifique aux courbes hyperelliptiques. Une bonne partie de son travail a consisté à calculer des

2. Notons que récemment Satoh [Sat00] a découvert un nouvel algorithme pour la petite caractéristique complètement différent de l'approche de Schoof. La complexité théorique est très bonne et de nouveaux records ont été obtenus, jusqu'à 2^{8009} [FGH00].

formules de récurrence pour décrire les éléments de ℓ-torsion. Son algorithme, bien qu'ayant une complexité bien meilleure que celui de Pila n'était pas encore prêt à l'époque pour être applicable en pratique.

Récemment, Adleman et Huang [AH96] puis Huang et Ierardi [HI98] ont amélioré les travaux de Pila. Les algorithmes proposés fonctionnent pour une courbe quelconque, singulière ou non. Là encore, il s'agit d'algorithmes théoriques et même si la complexité est bien meilleure que celle de l'algorithme de Pila, il n'est pas envisageable de traiter des exemples en vraie grandeur.

Dans tout ce chapitre, $\mathcal{C}$ désigne une courbe hyperelliptique définie sur un corps fini $\mathbb{F}_q$, où $q = p^d$ est impair. Le polynôme caractéristique de la Jacobienne de $\mathcal{C}$ est noté $\chi(t)$, et l'on cherche à calculer

$$\#\mathrm{Jac}(\mathcal{C}) = \chi(1).$$

7.1.2 Principe

Bref rappel de l'algorithme original de Schoof

Soit E une courbe elliptique définie sur un corps fini $\mathbb{F}_q$. Le polynôme caractéristique du Frobenius sur E est de la forme $\chi(t) = t^2 - ct + q$, où c est la *trace* de E. Soit ℓ un nombre premier impair distinct de la caractéristique. Alors le sous-groupe de ℓ-torsion de E est décrit par le *polynôme de division* $\psi_\ell(x)$, qui vérifie

$$P = (x, y) \in E[\ell] \iff \psi_\ell(x) = 0.$$

Ce polynôme est de degré $\frac{\ell^2-1}{2}$ et est calculable efficacement grâce à des formules de récurrence. Pour tout $P \in E[\ell]$, on a $\chi(\pi)(P) = 0$, et donc

$$\pi(\pi(P)) - [c \bmod \ell]\pi(P) + [q \bmod \ell]P = 0.$$

On construit alors formellement un élément de ℓ-torsion en prenant l'extension $\mathbb{F}_q[x]/(\psi_\ell(x))$, et on cherche pour quelle valeur de $c \bmod \ell$ l'équation ci-dessus est vérifiée. On déduit ainsi la valeur de $c \bmod \ell$ en temps polynomial en ℓ.

Le théorème de Hasse donne par ailleurs la borne $|c| \leqslant 2\sqrt{q}$. Ainsi après $O(\log q)$ nombres premiers ℓ (qui sont de taille $O(\log q)$), on a suffisament d'information pour retrouver c par le théorème des restes Chinois.

Extension au genre supérieur

L'analogue de l'algorithme de Schoof en genre supérieur consiste aussi à calculer $\chi(t)$ modulo de petits nombres premiers ℓ en travaillant dans $\mathrm{Jac}[\ell]$. Une fois cela fait pour suffisamment de nombres premiers (ou puissances de nombres premiers), $\chi(t)$ peut être reconstruit par le théorème Chinois. Grâce aux bornes sur les coefficients s_i de $\chi(t)$ provenant du théorème de Weil il est suffisant de considérer $\ell = O(\log q)$. En pratique, on ne peut pas conclure par cette seule méthode : on calcule seulement l'information modulo des ℓ aussi grands que possible, puis on combine avec les autres méthodes décrites précédemment.

Soit ℓ un entier premier distinct de la caractéristique. Alors le sous-groupe des éléments de ℓ-torsion a pour structure $\mathrm{Jac}[\ell] \cong (\mathbb{Z}/\ell\mathbb{Z})^{2g}$. De plus le Frobenius π agit linéairement sur ce sous-groupe et le théorème de Tate [Tat66] établit que le polynôme caractéristique de l'endomorphisme induit est précisément le polynôme $\chi(t)$ dont les coefficients ont été réduits modulo ℓ. Ainsi, calculer les éléments de $\mathrm{Jac}[\ell]$ et l'action du Frobenius dessus est suffisant pour obtenir $\chi(t)$ modulo ℓ.

Le lemme suivant, dû à Kampkötter simplifie le problème, au moins en genre 2.

Lemme 7.1 *En genre 2, si ℓ est une puissance d'un nombre premier impair, alors l'ensemble* $\mathrm{Jac} \setminus \Theta$ *contient une base de* $\mathrm{Jac}[\ell]$ *en tant que module libre sur* $\mathbb{Z}/\ell\mathbb{Z}$.

Démonstration. Nous reproduisons ici la preuve de Kampkötter.

Soit $\{D_1, D_2, D_3, D_4\}$ une base de $\mathrm{Jac}[\ell]$. Alors si D_i appartient à Θ, c'est un diviseur de poids 1, donc $2D_i$ est un diviseur de poids 2 et n'appartient pas à Θ. De plus, remplacer D_i par $2D_i$ ne change pas le fait que l'on a une base, car D_i est d'ordre ℓ qui est impair. □

Génériquement, dans le cas d'une courbe de genre supérieur à 2, le lemme reste vrai : avoir un élément de torsion dans Θ est un événement rare, et dans la suite, on supposera que cela n'arrive pas.

Idéal de ℓ-division

Un élément $D = \langle u(x), v(x)\rangle$ de $\mathrm{Jac} \setminus \Theta$ est un diviseur de poids g ; on notera ses coordonnées

$$D = \langle x^g + u_{g-1}x^{g-1} + \cdots + u_0,\ v_{g-1}x^{g-1} + v_{g-2}x^{g-2} + \cdots + v_0\rangle.$$

La condition $[\ell]D = 0$ peut être exprimée par un ensemble fini d'équations rationnelles en les u_i et v_i. Plus précisément, il existe un idéal I_ℓ de l'anneau de polynômes $\mathbb{F}_q[U_0, \ldots U_{g-1}, V_0, \ldots, V_{g-1}]$ tel que D appartienne à $\mathrm{Jac}[\ell] \setminus \Theta$ si et seulement si $f(u_0, \ldots, u_{g-1}, v_0, \ldots, v_{g-1}) = 0$ pour tout polynôme f appartenant à (un système de générateurs de) l'idéal I_ℓ. Dans [Kam91], Kampkötter donne des formules explicites pour des polynômes multivariés engendrant I_ℓ. Toutefois le système de coordonnées est légèrement différent.

Définition 7.1 *L'idéal I_ℓ est appelé* idéal de ℓ-division.

Supposons que l'on a construit I_ℓ. Alors on peut représenter un élément générique de $\mathrm{Jac}[\ell]\setminus\Theta$ par l'élément

$$D = \langle x^g + U_{g-1}x^{g-1} + \cdots + U_0,\ V_{g-1}x^{g-1} + V_{g-2}x^{g-2} + \cdots + V_0\rangle$$

sur l'algèbre quotient $\mathbb{F}_q[U_0, \ldots, U_{g-1}, V_0, \ldots, V_{g-1}]/I_\ell$. L'action du Frobenius peut être calculée sur cet élément, et il est possible de trouver son polynôme minimal par recherche exhaustive. Le polynôme caractéristique est alors facile à déduire (au moins dans le cas où ℓ est premier), et l'on a trouvé $\chi(t)$ modulo ℓ.

On peut prouver que cette méthode employée par Pila et Kampkötter a un temps d'exécution polynomial. Toutefois travailler dans l'algèbre quotient n'est pas facile en pratique : cela nécessite des calculs de bases de Gröbner qui sont extrêmement coûteux. Dans la suite nous allons voir comment on peut éviter les manipulations d'idéaux dans le cas du genre 2 de manière à obtenir un algorithme qu'on peut utiliser en pratique. Le point clef est de réussir à calculer efficacement des éléments de ℓ-torsion. En supposant que l'on sache faire cela, l'algorithme est le suivant où $s_1, \ldots, s_g$ désignent les coefficients de $\chi(t)$ cherchés :

$$\chi(t) = t^{2g} - s_1t^{2g-1} + s_2t^{2g-2} + \cdots + (-1)^g s_g t^g + (-1)^{g-1}qt^{g-1} + \cdots - q^{g-1}s_1t + q^g.$$

Algorithme 7.6 Algorithme de Schoof en genre g
Entrée: Une courbe hyperelliptique $\mathcal{C}$ de genre g définie sur $\mathbb{F}_q$, un nombre premier ℓ.
Sortie: $\#\mathrm{Jac}(\mathcal{C}) \bmod \ell$.

1. $S \leftarrow$ ensemble de tous les g-uplets possibles $(s_1, \ldots, s_g)$ à valeur dans $(\mathbb{Z}/\ell\mathbb{Z})^g$.

2. Tant que l'ensemble des $\chi(1) \bmod \ell$ possibles pour les g-uplets restant dans S n'est pas un singleton faire :
   (a) Construire un nouveau diviseur D de ℓ-torsion,
   (b) Pour chaque g-uplet dans S, si pour le $\chi(t) \bmod \ell$ correspondant on a $\chi(\pi)(D) \neq 0$, enlever le g-uplet de S.
3. Retourner le seul $\chi(1) \bmod \ell$ restant.

Le temps de calcul de cet algorithme dépend grandement du coût de l'étape 2.a. L'analyse précise sera faite en section 7.3.4.

7.2 Polynômes de division de Cantor

Dans [Can94], Cantor définit les *polynômes de division* des courbes hyperelliptiques, généralisant ainsi le cas elliptique. Il donne aussi des formules de récurrence efficaces pour les calculer.

Ces polynômes sont très liés à l'idéal de division I_ℓ, mais ils permettront de travailler essentiellement avec des polynômes univariés dans le cas du genre 2. On peut interpréter ces polynômes de division comme un moyen de représenter l'idéal I_ℓ directement sous une forme plus agréable : les variables sont un peu plus séparées, et dans le cas du genre 2, on obtient presque directement une base de Gröbner pour un ordre lexicographique.

7.2.1 Construction des polynômes de division en genre quelconque

Cette section est assez technique et l'on peut sans inconvénient passer à la section suivante, page 107. Les notations qui y sont introduites sont celles de l'article original de Cantor [Can94] et ne seront pas utilisées ailleurs. On s'autorise donc des conflits éventuels avec les notations des autres sections et chapitres.

Partant d'un point générique $P = (x_P, y_P)$ sur la courbe, le but est de calculer le couple de polynômes représentant l'élément de la Jacobienne $[\ell]\langle x - x_P, y_P\rangle$ sous forme de Mumford : $\langle u(x), v(x)\rangle = \left\langle \delta_\ell\left(\frac{x_P - x}{4y_P^2}\right), \varepsilon_\ell\left(\frac{x_P - x}{4y_P^2}\right)\right\rangle$.

On commence par faire un changement de variables :

$$x = x_P - z,\ \ E(z) = f(x_P - z),$$

de telle sorte que le point dont on veut l'exponentiation soit $P_0 = (0, -y_P)$ sur la courbe $y^2 = E(z)$.

On pose alors

$$S(z) = \sqrt{E(z)},$$

vu comme un développement en série en $z = 0$. Ce développement existe car $E(0) \neq 0$; sinon le point considéré est un point de 2-torsion.

Le point clef de la construction est la recherche de deux polynômes $A_\ell(z)$ et $B_\ell(z)$ tels que

$$z^\ell \text{ divise } A_\ell(z) - B_\ell(z)S(z),$$

$$2\deg A_\ell(z) \leqslant \ell + g,\ \text{ et } 2\deg B_\ell(z) \leqslant \ell - g - 1.$$

Nous allons montrer que le problème est alors à peu près résolu.

Soit D le diviseur des zéros de la fonction $\Phi(z,y) = A_\ell(z) - B_\ell(z)y$. Le diviseur D est effectif de degré $\ell + h$. En effet, la fonction Φ a un zéro de degré ℓ en P_0. On peut écrire D sous la forme $D = D' + \ell \cdot P_0$, où D' est un diviseur positif de degré h. De plus, $D - (\ell + h)\,\infty$ est un diviseur principal (par définition de D), donc est nul dans la Jacobienne. Ainsi $D' + \ell \cdot P_0 - (\ell + h)\,\infty \sim 0$, et en regroupant on obtient : $D' - h\,\infty \sim -\ell \cdot (P_0 - \infty)$. Si $h \leqslant g$, alors D' est le diviseur réduit de $[\ell](x,y)$.

Soit $D_\ell(z) = (A_\ell(z)^2 - B_\ell(z)^2 E(z))/z^\ell$. Les conditions imposées sur A_ℓ et B_ℓ font que D_ℓ est un polynôme de degré inférieur ou égal à g. De plus tout point de la courbe autre que P_0 et qui est zéro de Φ donnera un zéro de D_ℓ. Ce qui prouve $h \leqslant g$. Ainsi les abscisses (coordonnées en z) des points du diviseur D' seront les zéros du polynômes D_ℓ ; D_ℓ constitue donc le premier polynôme du couple qui représentera $\ell.(0,-y)$ (ce qui donnera δ_ℓ après changement de variable).

Trouver les polynômes A_ℓ et B_ℓ est un problème du type approximation de Padé. Il s'agit, étant donnés une série $S(z) = \sum_{j=0}^\infty s_j z^j$, et deux entiers positifs m et n, de trouver des polynômes non nuls $u(z)$ et $v(z)$ de degré respectifs m et n, tels que z^{m+n+1} divise $u(z) - v(z)S(z)$.

Une solution est obtenue par la démarche suivante : soit $u_{mn}(z,S) = \det U_{mn}(z,S)$, avec

$$U_{mn}(z,S) = \begin{pmatrix} s_{m-n+1} & s_{m-n+2} & \cdots & s_{m+1} \\ s_{m-n+2} & s_{m-n+3} & \cdots & s_{m+2} \\ \vdots & \vdots & \ddots & \vdots \\ s_m & s_{m+1} & \cdots & s_{m+n} \\ z^n S_{m-n}(z) & z^{n-1} S_{m-n+1}(z) & \cdots & S_m(z) \end{pmatrix},$$

où l'on a défini $S_j(z) = \sum_{i=0}^j s_j z^j$.

Soit $v_{mn}(z,S) = \det V_{mn}(z,S)$, avec

$$V_{mn}(z,S) = \begin{pmatrix} s_{m-n+1} & s_{m-n+2} & \cdots & s_{m+1} \\ s_{m-n+2} & s_{m-n+3} & \cdots & s_{m+2} \\ \vdots & \vdots & \ddots & \vdots \\ s_m & s_{m+1} & \cdots & s_{m+n} \\ z^n & z^{n-1} & \cdots & 1 \end{pmatrix}.$$

Posons ensuite

$$m_\ell = \left\lfloor \frac{\ell + g}{2} \right\rfloor, \quad n_\ell = \left\lfloor \frac{\ell - g - 1}{2} \right\rfloor.$$

Alors les solutions sont :

$$A_\ell(z) = u_{m_\ell n_\ell}(z), \quad B_\ell(z) = v_{m_\ell n_\ell}(z).$$

Ces formules permettent donc de déterminer les polynômes de divisions (elles ne sont valables que si ℓ est supérieur à $g+1$, mais les autres cas sont faciles).

Tout le problème est ensuite de calculer *rapidement* ces formules. Pour cela on consultera l'article de Cantor, qui permet d'aboutir aux formules de récurrence que nous allons maintenant donner pour le genre 2.

Formules de récurrence en genre 2

On commence par calculer récursivement un polynôme que Cantor note $\psi_\ell(x)$. Notons que ce polynôme n'est pas le même que celui qui sera construit dans le chapitre suivant : cette notation ne va servir que dans cette sous-section. La formule qui permet le calcul est

$$\psi_s\psi_r\psi_{s+r}\psi_{s-r} = \det\begin{pmatrix} \psi_{s-2}\psi_r & \psi_{s-1}\psi_{r+1} & \psi_s\psi_{r+2} \\ \psi_{s-1}\psi_{r-1} & \psi_s\psi_r & \psi_{s+1}\psi_{r+1} \\ \psi_s\psi_{r-2} & \psi_{s+1}\psi_{r-1} & \psi_{s+2}\psi_r \end{pmatrix}.$$

Si l'on veut calculer ψ_ℓ pour une grande valeur de ℓ, on applique cette formule pour $r = \lfloor \ell/2 \rfloor$ et $s = \lceil \ell/2 \rceil$ et on itère. Ainsi en un nombre polynomial d'étapes, on obtient le résultat.

Pour le calcul de δ_ℓ, on utilise deux polynômes intermédiaires α_ℓ et γ_ℓ. Nous noterons $\varphi\{\iota_2\}$ le polynôme obtenu en tronquant le polynôme φ, pour ne garder que les termes de degré *au plus 2*. On a alors

$$\delta_\ell(x_P, x) = (\alpha_\ell(x_P, x)\ \gamma_\ell(x_P, x))\{\iota_2\},$$

où α_ℓ et γ_ℓ sont définies par récurrence de la manière suivante :

$$-\alpha_{r+s-1}\psi_{s-r}\psi_{s-1}\psi_{r-1} = \det\begin{pmatrix} (\alpha_{r-2}\alpha_s)\{\iota_2\} & \psi_{r-2}\psi_s & \psi_{r-1}\psi_{s+1} \\ (\alpha_{r-1}\alpha_{s-1})\{\iota_2\} & \psi_{r-1}\psi_{s-1} & \psi_r\psi_s \\ (\alpha_r\alpha_{s-2})\{\iota_2\} & \psi_r\psi_{s-2} & \psi_{r+1}\psi_{s-1} \end{pmatrix}.$$

$$-\gamma_{r+s-1}\psi_{s-r}\psi_r\psi_s = \det\begin{pmatrix} \psi_{r-2}\psi_s & \psi_{r-1}\psi_{s+1} & (\gamma_{r-1}\gamma_{s+1})\{\iota_2\} \\ \psi_{r-1}\psi_{s-1} & \psi_r\psi_s & (\gamma_r\gamma_s)\{\iota_2\} \\ \psi_r\psi_{s-2} & \psi_{r+1}\psi_{s-1} & (\gamma_{r+1}\gamma_{s-1})\{\iota_2\} \end{pmatrix}.$$

Toutes ces formules sont valides pour $s \geqslant r \geqslant 4$ et ne sont utilisables que si $s - r \geqslant 2$. En effet ψ_{s-r} doit être non nul. Tout ceci signifie que pour initialiser les récurrences il faut calculer « à la main » les 7 ou 8 premiers termes selon les cas ; ce qui se fait en résolvant formellement le problème.

Pour simplifier les calculs, on a quelques résultats supplémentaires : le coefficient dominant de $\delta_\ell(x)$ est $-(4f(x_P))^2\psi_\ell(x_P)^2$; et le terme constant est $\delta_\ell(0) = -\psi_{\ell-1}\psi_{\ell+1}$. Donc, pour le calcul de δ_ℓ, il ne manque plus que le coefficient en x, que l'on peut obtenir grâce à $\alpha_\ell\{\iota_1\}$ et $\gamma_\ell\{\iota_1\}$. On peut ainsi tronquer les formules de récurrence au degré inférieur à 1 en x.

On dispose aussi de formules plus simples qui permettent de calculer le terme d'indice ℓ en fonction du terme d'indice $\ell - 1$ pour $\alpha_\ell\{\iota_1\}$ et $\gamma_\ell\{\iota_1\}$:

$$\gamma_\ell[0] = \psi_{\ell+1}, \qquad \alpha_\ell[0] = -\psi_{\ell-1},$$
$$\gamma_\ell[1]\ \psi_\ell = \gamma_{\ell-1}[1]\ \psi_{\ell+1} + \psi_{\ell-1}\psi_{\ell+2},$$
$$\alpha_\ell[1]\ \psi_{\ell-2} = \alpha_{\ell-1}[1]\ \psi_{\ell-1} + \psi_\ell\psi_{\ell-3},$$

où $p[h]$ désigne le coefficients de degré h en x dans le polynôme p.

Ces formules fournissent un moyen de calcul *non polynomial* des polynômes de divisions, mais sont toutefois intéressantes dans le cas où l'on doit calculer *tous* les polynômes γ_ℓ et α_ℓ jusqu'à un rang donné. Dans la pratique, on utilisera ces formules simplifiées : on n'atteint pas la limite où l'algorithme polynomial est meilleur que l'algorithme exponentiel.

Il reste à donner la formule qui permet de calculer ε_ℓ :

$$\varepsilon_\ell(x) = -\frac{y_P x(\psi_{\ell-1}^2\delta_{\ell+1}(x) - \psi_{\ell+1}^2\delta_{\ell-1}(x))}{\psi_{\ell-1}\psi_\ell^2\psi_{\ell+1}} \bmod \delta_\ell(x).$$

Cette formule n'est valable que pour $\ell \geqslant 3$, toutefois pour les valeurs inférieures, on calcule ε_ℓ sans difficultés.

7.2.2 Cas du genre 2

Dans le cas du genre 2, la construction de Cantor donne 6 suites de polynômes $d_0^{(\ell)}, d_1^{(\ell)}, d_2^{(\ell)}$ et $e_0^{(\ell)}, e_1^{(\ell)}, e_2^{(\ell)}$ telles que pour un diviseur $P = \langle x - x_P, y_P \rangle$ de poids 1 en position générale on a

$$[\ell]P = \left\langle x^2 + \frac{d_1^{(\ell)}(x_P)}{d_0^{(l)}(x_P)}x + \frac{d_2^{(\ell)}(x_P)}{d_0^{(\ell)}(x_P)},\ y_P\left(\frac{e_1^{(\ell)}(x_P)}{e_0^{(\ell)}(x_P)}x + \frac{e_2^{(\ell)}(x_P)}{e_0^{(\ell)}(x_P)}\right)\right\rangle. \tag{7.1}$$

Les degrés de ces polynômes de division sont

d_0	d_1	d_2	e_0	e_1	e_2
$2\ell^2-1$	$2\ell^2-2$	$2\ell^2-3$	$3\ell^2-2$	$3\ell^2-2$	$3\ell^2-3$

Un diviseur réduit $D = \langle u(x), v(x)\rangle$ de poids 2, c'est-à-dire n'appartenant pas à Θ, peut être écrit comme une somme de deux diviseurs réduits de poids 1 ; on a $D = P_1 + P_2$ avec $P_1 = \langle x - x_1, y_1\rangle$ et $P_2 = \langle x - x_2, y_2\rangle$ où x_1 et x_2 sont les racines de $u(x)$ et $y_i = v(x_i)$. Clairement

$$[\ell]D = [\ell]P_1 + [\ell]P_2.$$

Le diviseur D est de ℓ-torsion si et seulement si $[\ell]D = 0$, se qui se réécrit

$$[\ell]P_1 = -[\ell]P_2.$$

Deux diviseurs en représentation de Mumford sont opposés si et seulement si leurs polynômes $u(x)$ sont égaux et leurs polynômes $v(x)$ sont opposés. En utilisant la forme 7.1 pour $[\ell]P_1$ et $[\ell]P_2$, on obtient 4 équations polynomiales en les 4 inconnues x_1, x_2, y_1, y_2. Le point crucial est que deux de ces équations ne font intervenir que les deux variables x_1 et x_2.

Ainsi on obtient un idéal similaire à I_ℓ mais représenté sous forme agréable : on peut éliminer x_2 dans les deux équations bivariées en calculant des résultants, on obtient ainsi un polynôme en l'unique variable x_1, et pour chaque racine x_1 il n'est pas difficile de retrouver les valeurs correspondantes pour x_2, y_1 et y_2.

7.3 Recherche efficace d'un élément de ℓ-torsion

7.3.1 Construction d'un polynôme annulant x_1

On commence par calculer les polynômes de ℓ-division de Cantor grâce aux formules de la section précédente. Cette première phase prend un temps négligeable par rapport à ce qui suit. Le système de 4 équations s'écrit

$$\left\{\begin{array}{rclcl} E_1(x_1,x_2) & = & d_1(x_1)d_2(x_2) - d_1(x_2)d_2(x_1) & = & 0, \\ E_2(x_1,x_2) & = & d_0(x_1)d_2(x_2) - d_0(x_2)d_2(x_1) & = & 0, \\ F_1(x_1,x_2,y_1,y_2) & = & y_1e_1(x_1)e_0(x_2) + y_2e_1(x_2)e_0(x_1) & = & 0, \\ F_2(x_1,x_2,y_1,y_2) & = & y_1e_2(x_1)e_0(x_2) + y_2e_2(x_2)e_0(x_1) & = & 0, \end{array}\right. \tag{7.2}$$

et l'on va éliminer x_2 entre les deux premières équations. Le polynôme $(x_1 - x_2)$ est un facteur commun de E_1 et E_2, et ce facteur est un parasite : il ne correspond pas à un diviseur de ℓ-torsion.

S'il y a un autre facteur commun de E_1 et E_2 cela signifie qu'il existe un diviseur de ℓ-torsion non nul sur Θ. Ce cas se produit rarement, et doit être traité séparément car tous les degrés annoncés pour les polynômes sont perturbés. Nous nous plaçons donc dans le cas générique où seul $(x_1 - x_2)$ divise les deux équations.

On note encore $E_1(x_1, x_2)$ et $E_2(x_1, x_2)$ les deux premières équations divisées par $(x_1 - x_2)$. On élimine alors x_2 en calculant le résultant suivant :

$$R(x_1) = \mathrm{Res}_{x_2}(E_1(x_1, x_2), E_2(x_1, x_2)).$$

Ce calcul peut être effectué par un calcul de sous-résultants bivariés [vzGG99]. Toutefois la forme particulière des équations permet de faire mieux.

Le polynôme $R(x_1)$ que l'on cherche à calculer est divisible par une grande puissance de $d_2(x_1)$. En effet, si $d_2(x_1) = 0$ alors les expressions de E_1 et E_2 s'annulent en les racines de $d_2(x_2)$. Un petit calcul prenant en compte le fait que l'on a divisé par $(x_1 - x_2)$ donne que la puissance de d_2 dans R_1 est $\delta = 2\ell^2 - 3$. Supposons que le corps de base est suffisamment grand. On peut alors spécialiser E_1 et E_2 en de nombreuses valeurs distinctes de x_1. En substituant ξ_i pour x_1, le système devient deux polynômes *univariés* en x_2, pour lesquels on calcule le résultant r_i. Après avoir calculé un nombre de paires (ξ_i, r_i) supérieur au degré de

$$\tilde{R}(x_1) = R(x_1)/(d_2(x_1))^\delta,$$

on reconstruit $\tilde{R}(x_1)$ par interpolation. Connaissant les degrés de d_0, d_1, d_2, un simple calcul fournit

$$\deg \tilde{R}(x_1) = 4\ell^4 - 10\ell^2 + 6.$$

Algorithme 7.7 Calcul du résultant
Entrée: Une courbe hyperelliptique $\mathcal{C}$ de genre 2 définie sur $\mathbb{F}_q$, un nombre premier ℓ tel que $4\ell^4 - 10\ell^2 + 6 < q$.
Sortie: Le polynôme $\tilde{R}(x_1)$.

1. Calculer d_0, d_1, d_2 pour la courbe et le ℓ donnés.
2. Pour $4\ell^4 - 10\ell^2 + 7$ valeurs distinctes de $\xi_i \in \mathbb{F}_q$, faire :
   (a) Évaluer d_0, d_1, d_2 en ξ_i : soient ∂_0, ∂_1, ∂_2 les valeurs obtenues,
   (b) Calculer les polynômes $E_1^i(x) = (\partial_1 d_2(x) - \partial_2 d_1(x))/(x - \xi)$ et $E_2^i(x) = (\partial_0 d_2(x) - \partial_2 d_0(x))/(x - \xi)$.
   (c) Calculer r_i le résultant de $E_1^i(x)$ et $E_2^i(x)$.
   (d) Diviser r_i par $(\partial_2)^{2\ell^2 - 3}$.
3. Interpoler $\tilde{R}(x_1)$ à partir des valeurs (ξ_i, r_i).
4. Retourner $\tilde{R}(x_1)$.

7.3.2 Élimination des parasites

Comme mentionné précédemment, il y a ℓ^4 diviseurs de ℓ-torsion. Génériquement, tous sont de poids deux excepté le diviseur nul et donc le polynôme minimal de x_1 devrait être de degré $\ell^4 - 1$. Le degré de $\tilde{R}(x_1)$ est trop grand d'un facteur 4. Cela signifie qu'il y a encore beaucoup de facteurs parasites dus au fait que l'on n'a pris en compte que les conditions sur les abscisses x_1, x_2 et que l'on n'a rien imposé sur les ordonnées y_1, y_2. Deux stratégies peuvent alors être employées : on peut décider de vivre avec ces parasites et passer à l'étape suivante ou bien on peut calculer un autre résultant de manière à les éliminer (et obtenir ainsi un polynôme du degré attendu $\ell^4 - 1$). Le choix dépend des vitesses relatives du calcul de résultant et de recherche de racines pour les polynômes du corps considéré.

Afin d'éliminer les parasites, on construit une troisième équation impliquant x_1 et x_2 à partir des équations F_1 et F_2 du système 7.2 : celles-ci donnent toutes les deux une expression de y_1/y_2 en fonction de x_1 et x_2, on peut donc les égaler et l'on obtient

$$E_3(x_1, x_2) = e_1(x_1)e_2(x_2) - e_1(x_2)e_2(x_1) = 0,$$

que l'on divise là encore par $(x_1 - x_2)$.

Après avoir opéré le même genre d'astuces que précédemment, le résultant entre $E_1(x_1, x_2)$ et $E_3(x_1, x_2)$ donne un polynôme $\tilde{S}(x_1)$ de degré $12\ell^4 - 30\ell^2 + 18$ dont le pgcd avec $\tilde{R}(x_1)$ est de degré $\ell^4 - 1$ (du moins en général : quelques parasites peuvent subsister dans de rares cas). Dans la suite, on notera encore ce pgcd $\tilde{R}(x_1)$ pour simplifier.

7.3.3 Reconstruction de la solution modulo ℓ

Une fois $\tilde{R}(x_1)$ calculé, deux stratégies sont envisageables pour reconstituer un diviseur de ℓ-torsion. La première consiste à travailler modulo le polynôme $\tilde{R}(x_1)$ en entier, et la deuxième à rechercher d'abord des petits facteurs.

Calcul modulo $\tilde{R}(x_1)$

Cette approche ne fonctionne que dans le cas où l'on a éliminé les parasites. Le polynôme $\tilde{R}(x_1)$ permet de définir une extension algébrique de $\mathbb{F}_q$. A priori, il n'est pas irréductible, donc $A = \mathbb{F}_q[x_1]/(\tilde{R}(x_1))$ n'est pas un corps mais une somme directe de corps. Les lois d'anneau dans A sont bien définies, mais l'inverse d'un élément n'existe pas toujours. Pour inverser un élément de A, on procède comme si A était un corps : l'algorithme d'Euclide étendu donne la réponse en temps polynomial, sauf dans le cas où l'élément à inverser est un diviseur de zéro auquel cas le pgcd obtenu à la fin de l'algorithme d'Euclide n'est pas égal à 1. Ainsi on dispose d'une méthode efficace qui retourne soit l'inverse de élément souhaité, soit un facteur de $\tilde{R}(x_1)$. Ce dernier cas est improbable, mais s'il se produit, on peut recommencer les calculs en travaillant séparément dans les deux algèbres correspondant aux deux facteurs de $\tilde{R}(x_1)$ que l'on a découverts. Dans la suite, nous supposerons pour simplifier que toutes les inversions se passent bien.

L'élément x_1 de l'algèbre A représente la première coordonnée d'une solution générique du système 7.2. On peut alors reporter sa valeur afin de déterminer x_2 puis y_1 et y_2. Toutes ces coordonnées sont obtenues dans l'algèbre A. On forme ensuite le diviseur $D = P_1 + P_2$ où $P_1 = (x_1, y_1)$ et $P_2 = (x_2, y_2)$. Cet élément est désormais notre diviseur de ℓ-torsion auquel on appliquera le Frobenius afin de trouver les coefficients de $\chi(t) \mod \ell$.

Factorisation partielle de $\tilde{R}(x_1)$

En pratique, si l'on n'a pas éliminé les parasites, ou si l'on ne veut pas travailler dans l'algèbre A, on peut rechercher des facteurs de $\tilde{R}(x_1)$: on factorise $\tilde{R}(x_1)$, et pour chaque facteur irréductible on construit l'extension de $\mathbb{F}_q$ associée de manière à obtenir une racine X_1 de $\tilde{R}(x_1)$. Ensuite on substitue cette racine dans E_1 et E_2 et le pgcd donne la valeur correspondante X_2

de x_2 (définie sur le même corps). En utilisant l'équation de la courbe on obtient les ordonnées Y_1 et Y_2 au signe près et qui peuvent être dans une extension de degré 2. On obtient les deux diviseurs de poids 1 $P_1 = \langle x - X_1, Y_1 \rangle$ et $P_2 = \langle x - X_2, Y_2 \rangle$. Il reste à vérifier si $[\ell](P_1 + P_2) = 0$ ou $[\ell](P_1 - P_2) = 0$. Si aucun des deux n'est nul, on est parti d'un facteur parasite : on jette ce facteur et on repart d'un autre. Dans le cas favorable, on obtient un diviseur de ℓ-torsion D avec lequel on peut vérifier l'équation du Frobenius.

Par cette méthode, il peut être nécessaire de réitérer la procédure avec un autre diviseur de ℓ-torsion. En effet, un seul peut ne pas être suffisant pour obtenir une seule possibilité pour $\chi(t)$ mod ℓ. Afin d'accélérer la procédure, on commence par les facteurs de $\tilde{R}(x_1)$ de degré le plus petit.

7.3.4 Algorithme final, complexité

Une fois un connu un élément D de ℓ-torsion, on calcule

$$[s_1]\pi^3(D) + [qs_1 \bmod \ell]\pi(D),$$

pour tout $s_1 \in [0, \ell - 1]$ et

$$\pi^4(D) + [s_2]\pi^2(D) + [q^2 \bmod \ell]D,$$

pour tout $s_2 \in [0, \ell - 1]$. On garde seulement les paires (s_1, s_2) pour lesquelles les expressions sont égales.

S'il ne reste plus qu'une seule paire (s_1, s_2) ou s'il y en a plusieurs, mais correspondant toutes à la même valeur de $\chi(1)$ mod ℓ, alors il n'est pas nécessaire de continuer avec un nouveau facteur.

Algorithme 7.8 Étape ℓ dans l'algorithme de Schoof en genre 2
Entrée: Une courbe hyperelliptique $\mathcal{C}$ de genre 2 définie sur $\mathbb{F}_q$, un nombre premier ℓ.
Sortie: La valeur de $\chi(1)$ mod ℓ.

1. Calculer $\tilde{R}(x_1)$ par l'algorithme 7.7,
2. *(facultatif)* Éliminer les parasites dans $\tilde{R}(x_1)$,
3. Reconstruire un élément de ℓ-torsion D à partir de $\tilde{R}(x_1)$,
4. Pour chaque paire (s_1, s_2) restante, vérifier l'équation du Frobenius pour D.
5. Calculer l'ensemble des valeurs possibles de $\#\mathrm{Jac}/\mathbb{F}_q$ mod ℓ à partir des valeurs restantes de (s_1, s_2). S'il reste plusieurs valeurs, retourner à l'étape 3 pour construire un nouvel élément de torsion. S'il n'en reste qu'une, la retourner.

Pour évaluer le temps d'exécution de l'algorithme, on compte le nombre d'opérations dans le corps de base $\mathbb{F}_q$. Les facteurs $\log^\alpha \ell$ seront négligés, et l'on note $M(x)$ le nombre d'opérations nécessaire pour multiplier deux polynômes de degré x. Nous supposerons ici que les techniques d'arithmétique rapide de polynômes sont employées (cf [vzGG99]).

Le calcul de $\tilde{R}(x_1)$ requiert $O(\ell^4)$ calculs de résultants chacun d'eux pouvant se faire en $O(1)M(\ell^2)$ opérations et un calcul d'interpolation d'un polynôme de degré $O(\ell^4)$, ce qui peut se faire en $M(\ell^4)$ opérations. Au total, calculer $\tilde{R}(x_1)$ coûte

$$O(\ell^4)M(\ell^2) + O(1)M(\ell^4).$$

L'élimination des parasites se fait par un calcul similaire, puis par un pgcd. Cela ne rajoute donc rien dans la complexité.

Par la suite, on suppose que l'on choisit de calculer modulo $\tilde{R}(x_1)$, et que toutes les inversions se passent bien (sinon, on doit effectuer deux fois le calcul sur des données deux fois plus petites, et le coût est donc du même ordre). Une opération dans l'algèbre A est une manipulation de polynômes de degré $O(\ell^4)$, et peut donc se faire en $M(\ell^4)$ opérations dans $\mathbb{F}_q$. Pour reconstituer le diviseur D de ℓ-torsion, on doit reporter la valeur de x_1 dans le système 7.2, ce qui coûte $O(\ell^2)$ opérations dans A, et donc $O(\ell^2)M(\ell^4)$ opérations dans $\mathbb{F}_q$. Ensuite, la loi de groupe dans la Jacobienne se fait en un nombre fini d'opérations dans A, et l'application du Frobenius en $O(\log q)$ opérations dans A. Ensuite on a ℓ valeurs de s_1 à tester, et de même pour s_2. Finalement, le coût du calcul de $\chi(1) \bmod \ell$ est

$$O(\ell^4)M(\ell^2) + O(\log q)M(\ell^4)$$

opérations dans le corps de base.

Évaluons maintenant la complexité de l'algorithme en entier, si l'on veut calculer le cardinal de la Jacobienne uniquement à partir d'informations modulaires. On doit alors traiter des nombres premiers ℓ en quantité $O(\log q)$, ayant une taille maximale en $O(\log q)$. Nous supposerons que la multiplication rapide de polynômes à l'aide de transformée de Fourier rapide est employée et donc $M(x) = O(x)$ (si l'on ignore les facteurs logarithmiques). Ainsi le coût de l'algorithme est heuristiquement de $O(\log^7 q)$ opérations dans $\mathbb{F}_q$. Chaque opération peut être faite en $O(\log^2 q)$ opérations bit-à-bit. On obtient donc le cardinal de la Jacobienne d'une courbe de genre 2 définie sur $\mathbb{F}_q$ en temps $O(\log^9 q)$.

7.4 Construction de diviseurs de 2^k-torsion

Le but de cette section est de montrer comment obtenir de l'information sur $\#\mathrm{Jac}(\mathcal{C})$ modulo de petites puissances de 2. La factorisation de f donne immédiatement de l'information. Pour aller plus loin, on itère une méthode pour « diviser par deux » des diviseurs dans la Jacobienne. Cela mène rapidement à des diviseurs définis sur de grandes extensions, ce qui fait que le temps de calcul croît exponentiellement. En pratique on utilise cette technique pour obtenir de l'information partielle.

Les diviseurs d'ordre 1 et 2 sont précisément les $D = \langle u(x), 0\rangle$ pour lesquels $u(x)$ divise $f(x)$ et est de degré au plus g. Quand f a n facteurs irréductibles, alors il a 2^n facteurs en tout. Exactement la moitié d'entre eux ont degré au plus g car f est sans facteurs carré de degré $2g+1$. Ainsi le nombre de diviseurs d'ordre 1 ou 2 est 2^{n-1}, et $2^{n-1} \mid \#\mathrm{Jac}(\mathcal{C})$. De plus quand f est irréductible la 2-torsion rationnelle est triviale et $\#\mathrm{Jac}(\mathcal{C})$ est impair.

Par exemple, en genre 2 on obtient le tableau suivant :

Motif de factorisation de f	#Jac
(5)	1 mod 2
(4)(1)	0 mod 2
(3)(2)	0 mod 2
(3)(1)(1)	0 mod 4
(2)(2)(1)	0 mod 4
(2)(1)(1)(1)	0 mod 8
(1)(1)(1)(1)(1)	0 mod 16

7.4.1 Diviser par deux dans la Jacobienne

Nous décrivons cette méthode pour une courbe de genre g quelconque. Par la suite nous nous restreindrons au genre 2, seul cas pour lequel les calculs sont praticables.

Soit $D = \langle u(x), v(x)\rangle$ un diviseur non nul. Il s'agit de calculer un diviseur Δ tel que $[2]\Delta = D$. Notons qu'il y a 2^{2g} solutions, différant l'une de l'autre d'un diviseur de 2-torsion. En général Δ est défini sur une extension du corps de définition de D.

En écrivant $\Delta = \langle \tilde{u}(x), \tilde{v}(x)\rangle$, on peut déduire une expression rationnelle formelle pour le diviseur $[2]\Delta$ en utilisant l'algorithme de Cantor ou les formules de Harley. Ensuite, en égalisant cette expression avec celle de D on obtient un ensemble de $2g$ équations polynomiales en les $2g$ inconnues $\tilde{u}_i$ et $\tilde{v}_i$ avec les $2g$ paramètres u_i et v_i. En fait, il y a g^2 systèmes différents correspondant aux différents poids possibles de D et Δ.

Nous considérons le cas le plus fréquent où D et Δ sont tous les deux de poids g. Le système correspondant a au plus 2^{2g} solutions et celles-ci peuvent être obtenues en construisant une base de Gröbner pour un ordre lexicographique. On factorise ensuite le dernier polynôme de la base et on propage la solution dans les autres polynômes. Si le diviseur D est défini sur le corps de base, tout cela peut se faire en temps polynomial en $\log q$. Nous aurons toutefois à traiter des diviseurs dans des extensions de plus en plus grandes.

Afin d'accélérer les calculs dans le cas où D est défini sur une grande extension, il est possible d'éviter les calculs répétés de bases de Gröbner en calculant dès le début une seule base de Gröbner générique pour le système où les coefficients de D sont gardés comme des paramètres. Comme la division par 2 est algébrique sur $\mathbb{F}_q$ (car la courbe est elle-même définie sur $\mathbb{F}_q$), la base générique est elle aussi définie sur $\mathbb{F}_q$. Après ce précalcul, on peut diviser par 2 un diviseur D quelconque, même défini sur une grande extension en substituant ses coefficients dans la base générique pour obtenir la base de Gröbner spécialisée directement.

Ce calcul de base générique est dû à Éric Schost [Sch]. Pour sa construction, ce dernier a utilisé le logiciel `Kronecker` écrit par Grégoire Lecerf [Lec99]. Ce logiciel se comporte très bien pour résoudre ce type de problèmes (relever des systèmes spécialisés vers des systèmes génériques), et il est probable que ce calcul n'aurait pas pu être fait par les algorithmes classiques de calcul de base de Gröbner.

Exemple

Soit $\mathcal{C}$ la courbe d'équation

$$y^2 = x^5 + 1597\,x^4 + 1041\,x^3 + 5503\,x^2 + 6101\,x + 1887,$$

sur le corps $\mathbb{F}_p$ avec $p = 10^{17} + 3$. Nous allons donner tous les diviseurs *rationnels* d'ordre une puissance de 2, c'est-à-dire ceux définis sur $\mathbb{F}_p$. Deux facteurs irréductibles de $f(x)$ ont degré au plus 2 :

$$f_1 = x + 28555025517563816 \quad \text{et} \quad f_2 = x + 74658844563359755,$$

Ainsi il y a trois diviseurs rationnels d'ordre exactement 2 : $P_1 = \langle f_1, 0\rangle$, $P_2 = \langle f_2, 0\rangle$ et $P_1 + P_2$. La méthode de division par deux appliquée à P_1 permet de trouver quatre diviseurs rationnels d'ordre 4. Ce sont $\langle u, v\rangle$ et $\langle u, -v\rangle$ où :

$$\begin{aligned} u &= x^2 + 1571353025997967\,x + 12198441063534328 \\ v &= 32227723250469108\,x + 68133247565452990 \end{aligned}$$

et :

$$\begin{aligned} u &= x^2 + 70887725815800572\,x + 94321182398888258 \\ v &= 42016761890161508\,x + 3182371156137467. \end{aligned}$$

Il y a 16 solutions en tout mais les autres sont dans une extension de $\mathbb{F}_p$ (les bases de Gröbner sont trop grandes pour être écrites sur du papier !) La méthode de division par deux appliquée à P_2 et $P_1 + P_2$ ne fournit pas de nouveaux éléments rationnels d'ordre 4. En continuant ainsi, on trouve 8 diviseurs d'ordre 8, 16 d'ordre 16, 32 d'ordre 32, et c'est tout. Ainsi le 2-sous-groupe de la Jacobienne définie sur $\mathbb{F}_p$ est de la forme $(\mathbb{Z}/2) \times (\mathbb{Z}/32)$ et donc $\#\mathrm{Jac}(\mathcal{C}) \equiv 64 \bmod 128$.

Ce type de recherche exhaustive dans le corps de base permet de déterminer l'exacte puissance de 2 qui divise $\#\mathrm{Jac}/\mathbb{F}_p$. On peut aller plus loin en utilisant les diviseurs de torsion que l'on construit pour les reporter dans l'équation du Frobenius.

7.4.2 Algorithme pour calculer $\#\mathrm{Jac}(\mathcal{C}) \bmod 2^k$

Maintenant nous allons dans des extensions de $\mathbb{F}_q$ pour trouver des diviseurs de 2^k-torsion et les substituer dans $\chi(t)$ pour déterminer les valeurs de ses coefficients modulo 2^k de la même manière que dans l'algorithme de Schoof. On peut ainsi déduire la valeur de $\#\mathrm{Jac}(\mathcal{C}) \bmod 2^k$ pour des valeurs croissantes de k. Nous présentons une version idéalisé de l'algorithme pour le genre 2.

Algorithme 7.9 Cardinalité modulo les puissances de 2
Entrée: Une courbe hyperelliptique $\mathcal{C}$ de genre 2.
Sortie: Les valeurs successives de $\#\mathrm{Jac}(\mathcal{C}) \bmod 2^k$.

1. Factoriser f pour trouver un diviseur de 2-torsion. Le diviser par 2 pour obtenir un diviseur D_4 de 4-torsion.
2. Trouver la paire $(s_1, s_2) \bmod 4$ pour laquelle $\chi(D_4) = 0$. Mettre k à 2.
3. Calculer la base de Gröbner générique pour la division par 2 dans $\mathrm{Jac}(\mathcal{C})$.
4. Construire un diviseur $D_{2^{k+1}}$ de 2^{k+1}-torsion de la manière suivante : substituer les coefficients de D_{2^k} dans le système, calculer une racine du polynôme éliminant dans une extension de degré minimal et la propager dans le système.
5. Pour chaque paire $(s_1, s_2) \bmod 2^{k+1}$ compatible avec la paire précédemment calculée modulo 2^k, brancher $D_{2^{k+1}}$ dans χ et trouver la paire pour laquelle $\chi(D_{2^{k+1}}) = 0$.
6. Poser $k = k + 1$, et revenir à l'étape 4.

Il s'agit d'un algorithme idéalisé au sens où il restera fréquemment plusieurs paires (s_1, s_2) après avoir vérifié l'équation du Frobenius pour un diviseur de 2^k-torsion. On peut éliminer les mauvais candidats en construisant d'autres diviseurs de 2^k-torsion. Il peut s'avérer coûteux de les éliminer tous quand les diviseurs nécessaires sont dans des extensions de plus en plus grandes ; une stratégie alternative est de continuer et d'espérer que les mauvais candidats seront éliminés plus tard, avec un diviseur de 2^{k+1}-torsion.

Dans cet algorithme on peut sauter l'étape 3 et calculer une base de Gröbner à chaque passage à l'étape 4. Toutefois la base de Gröbner générique est plus efficace et permet de calculer une ou deux itérations de plus dans le même temps de calcul.

7.5 Combinaison avec d'autres algorithmes, résultats

Nous avons implanté en Magma les algorithmes de ce chapitre. Cette programmation en langage de haut niveau n'est pas trop pénalisante pour ce type d'algorithme où le temps de calcul est principalement dû à des opérations sur de grands polynômes. En effet, Magma dispose d'algorithmes rapides pour multiplier les polynômes, à base de transformée de Fourier rapide. Afin de compléter ces calculs en des « records » nous avons jumelé nos efforts avec R. Harley dont l'implantation de l'algorithme Paradoxe des Anniversaires est particulièrement efficace.

7.5.1 Combinaison avec l'algorithme Paradoxe des Anniversaires

Le résultat suivant a été obtenu avec Robert Harley et a donné lieu à une publication [GH00]. Soit $\mathcal{C}$ la courbe « aléatoire » d'équation

$$\begin{aligned} y^2 = x^5 &+ 3141592653589793238\ x^4 + 4626433832795028841\ x^3 \\ &+ 9716939937510582097\ x^2 + 4944592307816406286\ x \\ &+ 2089986280348253421\ , \end{aligned}$$

sur le corps fini à $p = 10^{19} + 51$ éléments. L'ordre de sa Jacobienne est

$$\#\mathrm{Jac}(\mathcal{C}) = 99999999982871020671452277000281660080,$$

et le polynôme caractéristique du Frobenius a pour coefficients :

$$s_1 = 1712898036 \quad \text{et} \quad s_2 = 11452277089352355350.$$

La première étape du calcul est la factorisation de $f(x)$. Il y a 3 facteurs irréductibles, donc on sait déjà que $\#\mathrm{Jac} \equiv 0 \bmod 4$.

La deuxième étape est le relèvement de diviseurs d'ordre une puissance de 2. Le calcul de la base de Gröbner générique pour la division par deux (effectué par É. Schost) prend environ une heure sur une station de travail Alpha. Ensuite nous avons relevé les diviseurs plusieurs fois, et testé à chaque fois l'équation du Frobenius. Dans le tableau suivant, nous donnons le degré de l'extension dans laquelle il y a un diviseur de 2^k-torsion et l'information sur #Jac que l'on obtient (temps de calcul sur un Pentium 450).

#Jac	deg d'ext	#Jac	deg d'ext	temps
0 mod 2	1	16 mod 32	16	
0 mod 4	1	48 mod 64	32	
0 mod 8	4	48 mod 128	64	5000 sec
0 mod 16	8	176 mod 256	128	9 heures

L'étape suivante est le calcul modulo $\ell \in \{3, 5, 7, 11, 13\}$ par l'algorithme de Schoof. Nous n'avons pas éliminé les parasites aussi avons nous dû trouver des facteurs de $\tilde{R}(x_1)$. La table suivante donne pour chaque ℓ le degré des polynômes $\tilde{R}(x_1)$ et la plus petite extension dans laquelle on trouve un diviseur de ℓ-torsion (temps de calcul sur un Pentium 450).

ℓ	degré de $\tilde{R}(x_1)$	degré d'ext	#Jac	temps
3	240	2	1 mod 3	1200 sec
5	2256	1	0 mod 5	300 sec
7	9120	6	4 mod 7	12 heures
11	57360	1	0 mod 11	19 heures
13	112560	7	9 mod 13	205 heures

Le temps de calcul pour $\ell = 3$ est étonnement grand dans cette table. Pour cette courbe, un événement malchanceux qui devient rare quand ℓ devient grand, s'est produit. En effet, après avoir testé l'équation du Frobenius pour *tous* les éléments de 3-torsion, plusieurs candidats (s_1, s_2) restent en lice, correspondant à plusieurs possibilités pour #Jac mod 3. Ce que cela signifie c'est que le polynôme minimal de π n'est pas son polynôme caractéristique. Chaque candidat restant pour (s_1, s_2) donne un multiple du polynôme minimal. En prenant leur pgcd, on obtient le polynôme minimal exact pour lequel il est ensuite facile de déduire le polynôme caractéristique et #Jac mod 3. Ce problème est expliqué en détail dans [Kam91].

Dans notre cas, il reste 3 paires après avoir testé tous les diviseurs de 3-torsion, ce qui donne les candidats suivants pour #Jac mod 3.

(s_1, s_2) mod 3	#Jac mod 3	$\chi(t)$ mod 3
$(0, 2)$	1	$t^4 - t^2 + 1$
$(1, 2)$	2	$t^4 - t^3 - t^2 - t + 1$
$(2, 2)$	0	$t^4 + t^3 - t^2 + t + 1$

Le troisième cas est impossible car si $\#\mathrm{Jac} \equiv 0 \bmod 3$ alors on aurait trouvé un élément de 3-torsion rationnel dès le début. Afin de décider entre les deux derniers choix, on calcule le polynôme minimal, qui se trouve être $t^2 + 1$ et ainsi le polynôme caractéristique ne peut être que $(t^2 + 1)^2$. Finalement on a $\#\mathrm{Jac} \equiv 1 \bmod 3$.

Toutefois pour faire ce calcul nous avons été obligé de construire tous les diviseurs de 3-torsion, ce qui explique pourquoi le temps de calcul est plus élevé que pour $\ell = 5$ où nous avons trouvé un diviseur de 5-torsion rationnel et avons immédiatement déduit $\#\mathrm{Jac} = 0 \bmod 5$.

La fin de calcul se fait par un calcul de Paradoxe des Anniversaires. La largeur de l'intervalle de Hasse-Weil est environ 2.5×10^{29}. L'espace de recherche est réduit d'un facteur $2^8 \times 3 \times 5 \times 7 \times 11 \times 13 = 3843840$ ce qui laisse 6.6×10^{22} candidats. La recherche a été faite par Robert Harley en parallèle sur une dizaine de stations de travail Alpha et a nécessité 5×10^{11} opérations dans la Jacobienne. Sur une seule station à 500 MHz cette dernière phase aurait duré environ 50 jours.

7.5.2 Cas de la petite caractéristique : combinaison avec Cartier-Manin

Soit $\mathcal{C}$ la courbe d'équation

$$y^2 = x^5 + x^4 + x^3 + x^2 + ux + 1,$$

définie sur le corps $\mathbb{F}_{3^{30}} = \mathbb{F}_3[u]/(u^{30} + u - 1)$. La cardinalité de sa Jacobienne est

$$\#\mathrm{Jac} = 42391156018493425614913594804.$$

La valeur modulo 3 peut être obtenue indifféremment par l'opérateur de Cartier-Manin ou par l'algorithme de Schoof. L'exemple suivant illustre mieux l'avantage de Cartier-Manin : on choisit un corps de caractéristique relativement grande de sorte que le calcul modulo p ne soit pas faisable par l'algorithme de Schoof.

Soit $\mathcal{C}$ la courbe d'équation

$$y^2 = x^5 + x^4 + x^3 + x^2 + x + u,$$

sur le corps $\mathbb{F}_{p^4} = \mathbb{F}_p[u]/(u^4 - 17)$, où $p = 2^{16} - 15$.

L'opérateur de Cartier-Manin permet de calculer $\#\mathrm{Jac} \equiv 58976 \bmod p$ en 17 minutes, et en finissant par avec les autres méthodes on obtient

$$\#\mathrm{Jac} = 339659790214687297284652908385855015466.$$

7.5.3 Cas d'une courbe à multiplication réelle

En combinant l'algorithme de Schoof avec l'algorithme 6.5 de la page 95, il est en principe possible de faire un exemple de taille cryptographique $\#\mathrm{Jac} \approx 2^{160}$, au prix d'un calcul de $O(1)$ semaines. Ce calcul sera entrepris durant les prochains mois. Le but est de trouver une telle courbe dont l'ordre serait presque premier... Bien entendu, grâce à l'algorithme de Schoof, on peut rapidement éliminer les courbes pour lesquelles l'ordre est divisible par un petit facteur.

Chapitre 8

Vers une extension Elkies–Atkin en genre supérieur

L'algorithme à la Schoof décrit dans le chapitre précédent est polynomial mais avec un degré élevé. Ceci est en partie dû au fait que pour traiter la ℓ-torsion, on doit manipuler des polynômes de degré $O(\ell^{2g})$. L'idée naturelle qui vient alors est de tenter de transcrire ce qui se fait habituellement pour les courbes elliptiques : réduire le degré des polynômes à manipuler en utilisant des équations modulaires, tout ceci en lien avec les isogénies de degré ℓ. Les équations modulaires décrites au chapitre 3 présentent deux inconvénients dans ce contexte. Le premier est qu'elles correspondent à des isogénies de degré ℓ^2 d'un type particulier, ce qui était nécessaire pour garantir que la variété abélienne à l'arrivée était bien une Jacobienne, mais qui nous pénalise ici. Le deuxième problème est la taille de ces équations. Même pour $\ell = 2$, le nombre de coefficients est énorme.

Le but du présent chapitre est de décrire une autre famille d'équations modulaires mieux adaptées à notre problème de cardinalité. La construction est purement algébrique et ne fait pas intervenir de formes de Siegel, ce qui permet de la faire (au moins en théorie) en genre quelconque, bien que la théorie des invariants ne sont pas complète au delà du genre 2. Cette construction se rapproche de celle effectuée il y a une dizaine d'années par Charlap, Coley et Robbins [CCR91] dans le cas elliptique.

Tout ceci est un travail en cours que nous poursuivons en collaboration avec É. Schost [GSb].

8.1 Construction du polynôme modulaire Ξ_ℓ

8.1.1 Description générale

L'idée initiale est très simple : les isogénies de degré ℓ sont en correspondance bijective avec les sous-groupes d'ordre ℓ de la Jacobienne. Ainsi plutôt que d'essayer de paramétrer les variétés abéliennes ℓ-isogènes à notre Jacobienne, on va préférer paramétrer les sous-groupes d'ordre ℓ. Lorsqu'on quotiente une Jacobienne par un sous-groupe, on obtient une variété abélienne qui en général n'est pas une Jacobienne. Le premier avantage de cette nouvelle approche est qu'elle évite de manipuler ces variétés abéliennes. Le deuxième avantage est que le nombre d'inconnues va être réduit par rapport à ce que l'on avait au chapitre 3.

Dans ce chapitre, nous allons utiliser le langage de l'algèbre commutative et du calcul formel. Nous renvoyons à [CLO98] pour les définitions des notions évoquées.

La stratégie employée est la suivante : étant donnée une courbe, on peut définir son idéal de

ℓ-division qui décrit tous les diviseurs de ℓ-torsion de poids g. Génériquement, il y en a $\ell^{2g}-1$. Ces diviseurs peuvent être regroupés en une réunion finie de $\frac{\ell^{2g}-1}{\ell-1}$ ensembles L_i, tous de cardinal $\ell-1$, de telle sorte que chaque L_i est un sous-groupe d'ordre ℓ privé de 0. On choisit une coordonnée arbitraire T_i pour paramétrer ces L_i, qui génériquement sera séparante. On définit alors le polynôme modulaire associé à la courbe comme étant le produit $\prod(T-T_i)$. C'est un polynôme de degré $\frac{\ell^{2g}-1}{\ell-1}$ en T. Comme ses racines sont liées aux sous-groupes d'ordre ℓ, la manière dont il se factorise est liée au cardinal de la Jacobienne modulo ℓ.

Il s'avère que les coefficients sont des fractions rationnelles en les coefficients de l'équation de la courbe, que l'on peut calculer (par interpolation par exemple). C'est le polynôme multivarié obtenu en chassant les dénominateurs dont on va étudier les propriétés.

Dans ce qui suit on va supposer que la caractéristique du corps est suffisament grande pour que tout fonctionne bien. Ainsi on peut choisir un modèle de courbe hyperelliptique avec un coefficient en moins, et on s'autorise la division par n'importe quel entier de taille bornée.

8.1.2 Construction de ψ_ℓ, propriétés

Une manière de dire que les paramètres sont en position générique est de travailler sur le corps des fractions rationnelles en ces paramètres. C'est ce que nous allons faire ici.

Soit $g \geqslant 1$ un entier, et soit $K = \mathbb{Q}(f_0, f_1, \ldots, f_{2g-1})$ le corps des fractions rationnelles en $2g$ variables. Soit $\mathcal{C}$ la courbe hyperelliptique sur K d'équation $y^2 = x^{2g+1} + f_{2g-1}x^{2g-1} + \cdots + f_1 x + f_0$. Comme dans les chapitres précédents, on utilise la représentation de Mumford pour un élément D de la Jacobienne de $\mathcal{C}$:

$$D = \langle u(x), v(x)\rangle = \langle x^g + u_{g-1}x^{g-1} + \cdots + u_0, v_{g-1}x^{g-1} + v_{g-2}x^{g-2} + \cdots + v_0\rangle.$$

Soit ℓ un nombre premier impair fixé pour toute la section. On rappelle la définition de l'idéal de ℓ-division pour la courbe $\mathcal{C}$:

Définition 8.1 *L'idéal de ℓ-division de $\mathcal{C}$, noté I_ℓ est l'ensemble des polynômes P de $K[u_0, \ldots, u_{g-1}, v_0, v_1, \ldots, v_{g-1}]$ tels que pour tout diviseur $D = \langle u(x), v(x)\rangle$ de ℓ-torsion,*

$$P(u_0, u_1, \ldots, u_{g-1}, v_0, v_1, \ldots, v_{g-1}) = 0.$$

Si un diviseur D est un élément de ℓ-torsion, alors son opposé l'est aussi. Ceci suggère de « quotienter » l'idéal I_ℓ par l'involution hyperelliptique. Cette démarche est analogue à ce qui se fait pour les courbes elliptiques où la ℓ-torsion est entièrement déterminée par un polynôme univarié décrivant les abscisses des points.

Une façon de modifier l'idéal I_ℓ est d'effectuer le changement de variables suivant :

$$\mathfrak{v}_0 = v_0^2 \quad \text{et} \quad \mathfrak{v}_i = v_0 v_i \quad \text{pour} \quad 1 \leqslant i \leqslant g-1.$$

Le nouveau système de variables ne change pas lorsque l'on passe d'un diviseur à son opposé. Réciproquement, revenir au système initial nécessite de choisir une racine carrée, ce qui revient à choisir un des deux diviseurs parmi les deux opposés.

Définition 8.2 *L'idéal de ℓ-division modifié de $\mathcal{C}$, noté I_ℓ^* est l'ensemble des polynômes P de $K[u_0, u_1, \ldots, u_{g-1}, \mathfrak{v}_0, \mathfrak{v}_1, \ldots, \mathfrak{v}_{g-1}]$ tels que pour tout diviseur $D = \langle u(x), v(x)\rangle$ de ℓ-torsion,*

$$P(u_0, u_1, \ldots, u_{g-1}, v_0^2, v_0v_1, \ldots, v_0v_{g-1}) = 0.$$

Les points de la variété associée à l'idéal I_ℓ^* représentent exactement les couples $\{D, -D\}$ de diviseurs de ℓ-torsion qui sont de poids g. Sur une clôture algébrique, il y a en tout ℓ^{2g} diviseurs de ℓ-torsion, parmi lesquels on trouve naturellement le diviseur nul. Les autres sont génériquement de poids g et se groupent par deux. Ainsi la variété de I_ℓ^* contient $\frac{\ell^{2g}-1}{2}$ points. D'autre part, génériquement le coefficient u_{g-1} du polynôme $u(x)$ est différent pour chaque point (il en est de même des autres, d'ailleurs). Ceci peut être résumé dans la proposition suivante :

Proposition 8.1 *L'idéal I_ℓ^* est un idéal de dimension zéro et de degré $\frac{\ell^{2g}-1}{2}$. La variable u_{g-1} est un élément séparant les points de la variété associée.*

Plus précisément, on peut utiliser la variable u_{g-1} pour décrire de manière agréable l'idéal I_ℓ^*.

Proposition 8.2 *Il existe $2g$ polynômes $\varphi_i(U)$, univariés, à coefficients dans K, de degré inférieur à $\frac{\ell^{2g}-1}{2}$, tels que l'idéal I_ℓ^* soit engendré par les polynômes suivants :*

$$\begin{cases} u_{g-1}^{(\ell^{2g}-1)/2} + \varphi_1(u_{g-1}), \\ u_{g-2} + \varphi_2(u_{g-1}), \\ \vdots \\ u_0 + \varphi_g(u_{g-1}), \\ v_{g-1} + \varphi_{g+1}(u_{g-1}), \\ \vdots \\ v_0 + \varphi_{2g}(u_{g-1}). \end{cases}$$

Démonstration. Par construction, l'idéal est radical (on l'a défini comme l'idéal d'une variété) et de dimension zéro. La variable u_{g-1} sépare les points de la variété, nous sommes donc dans les conditions d'application du *Shape lemma* (cf [CLO98], ex 16, page 62) qui donne exactement la représentation voulue. $\square$

Ainsi, partant d'une racine du polynôme minimal de l'élément u_{g-1}, on déduit facilement les valeurs des autres variables en substituant dans le système. C'est ce polynôme minimal qui va être essentiellement le polynôme ψ_ℓ que l'on cherche à construire.

Théorème 8.1 *Pour tout nombre premier impair ℓ et pour tout entier $g \geqslant 1$, il existe un polynôme multivarié sur $\mathbb{Q}$, noté $\psi_\ell(U, f_0, f_1, \ldots, f_{2g-1})$ qui s'annule précisément quand il existe un diviseur de ℓ-torsion $D = P_1 + P_2 + \cdots + P_g - g\infty$ sur la Jacobienne de la courbe $y^2 = x^{2g+1} + f_{2g-1}x^{2g-1} + \cdots + f_1 x + f_0$, pour lequel la somme des abscisses des P_i vaut U. Son degré en U est $\frac{\ell^{2g}-1}{2}$.*

Démonstration. En multipliant le polynôme $U^{(\ell^{2g}-1)/2} + \varphi_1(U)$ par tous les dénominateurs intervenants dans les coefficients, on obtient un polynôme multivarié ψ_ℓ en les variables $U, f_0, f_1, \ldots, f_{2g-1}$. La variable u_{g-1} décrit la somme des abscisses des points intervenants dans un diviseur. Ainsi les propriétés souhaitées sont obtenues par construction de l'idéal I_ℓ^*. $\square$

Proposition 8.3 *Le polynôme ψ_ℓ est homogène si l'on affecte le poids 1 à la variable U et le poids $2g+1-i$ à la variable f_i.*

Démonstration. Soit t un nombre rationnel non nul, soit $\mathcal{C}$ la courbe d'équation $y^2 = f(x) = x^{2g+1} + f_{2g-1}x^{2g-1} + \cdots + f_1x + f_0$ et soit $\tilde{\mathcal{C}}$ d'équation $\tilde{y}^2 = \tilde{f}(\tilde{x}) = \tilde{x}^{2g+1} + \tilde{f}_{2g-1}\tilde{x}^{2g-1} + \cdots + \tilde{f}_1\tilde{x} + \tilde{f}_0$, où $\tilde{f}_i = f_i t^{2g+1-i}$. Alors l'application

$$\phi \ : \ (x,\, y) \mapsto (\tilde{x} = tx,\, \tilde{y} = t^{2g+1}y)$$

est un isomorphisme entre $\mathcal{C}$ et $\tilde{\mathcal{C}}$.

Soit u une racine de $\psi_\ell(U, f_0, \ldots, f_{2g-1})$. Il existe un diviseur de ℓ-torsion $D = P_1 + \cdots + P_g - g\infty$ de $\mathrm{Jac}(\mathcal{C})$ tel que u soit la somme des abscisses des P_i. Le diviseur $\tilde{D}$ défini par $\phi(P_1) + \cdots + \phi(P_g) - g\infty$ est un diviseur de ℓ-torsion de $\mathrm{Jac}(\tilde{\mathcal{C}})$, et la somme des abscisses est $\tilde{u} = tu$. Ainsi on a

$$\forall t \in \mathbb{Q}, \quad \psi_\ell(u, f_0, \ldots, f_{2g-1}) = 0 \implies \psi_\ell(tu, t^{2g+1}f_0, \ldots, t^2 f_{2g-1}) = 0.$$

Et ψ_ℓ est homogène pour les poids annoncés. □

8.1.3 Action sur les racines de ψ_ℓ

À chaque racine u de ψ_ℓ, on peut associer une paire de diviseurs de ℓ-torsion que l'on notera $\pm D_u$. De plus d'après la proposition 8.2 les coefficients de $\pm D_u$ s'expriment comme des fractions rationnelles en u.

Proposition 8.4 *Soient g, ℓ, $K = \mathbb{Q}(f_0, f_1, \ldots, f_{2g-1})$ et ψ_ℓ définis comme ci-dessus. Pour tout entier k dans $[1, \frac{\ell-1}{2}]$, il existe un polynôme $h_k(U)$ à coefficients dans K tel que si u est la racine de ψ_ℓ correspondant au diviseur de ℓ-torsion $\pm D_u$, alors $h_k(u)$ est la racine de ψ_ℓ correspondant au diviseur $[\pm k]D_u$.*

Démonstration. Soit u une racine de ψ_ℓ et $\pm D_u$ le diviseur associé. La multiplication par un entier k dans la Jacobienne commute avec l'involution : $\pm[k]D_u = [\pm k]D_u$, et donc calculer les coefficients de la paire $\pm[k]D_u$ nécessite seulement de connaître $\pm D_u$. De plus la multiplication s'exprime à l'aide de fractions rationnelles en les coefficients, qui eux-mêmes sont des fractions rationnelles en u. Ainsi il existe une fraction rationnelle en u qui exprime la racine de ψ_ℓ correspondant à $\pm[k]D_u$. La réduction de cette fraction modulo ψ_ℓ fournit le polynôme h_k cherché. □

Bien entendu, on peut composer deux polynômes h_k, et réduire modulo ψ_ℓ. On obtient alors un autre polynôme h_k.

Proposition 8.5 *Pour tous entiers k_1 et k_2 dans $[1, \frac{\ell-1}{2}]$, on a*

$$h_{k_1} \circ h_{k_2}(U) \pmod{\psi_\ell(U)} \ = \ h_{k_2} \circ h_{k_1}(U) \pmod{\psi_\ell(U)} \ = \ h_k,$$

où k est celui des deux entiers $(k_1k_2 \mod \ell)$ et $(-k_1k_2 \mod \ell)$ qui se trouve dans $[1, \frac{\ell-1}{2}]$.

On peut comprendre les h_k comme une description explicite d'un sous-groupe d'ordre $(\ell-1)/2$ du groupe de Galois du polynôme ψ_ℓ. Cette action sur les racines est cyclique, et est en fait isomorphe à $\mathbb{F}_\ell^*/\{\pm 1\}$.

Ce qu'on va calculer est une résolvante de ψ_ℓ pour cette action. En d'autres termes, on va calculer le polynôme minimal d'une fonction des racines invariante sous l'action des h_k.

8.1.4 Définition et propriétés de Ξ_ℓ

Définition 8.3 *Le polynôme minimal de l'élément $T = \sum_{1 \leqslant k \leqslant (\ell-1)/2} h_k(U)$ dans l'algèbre quotient $K[(u_i), (\mathfrak{v}_i)]/I_\ell^*$ associée à l'idéal de division modifié est un polynôme unitaire à coefficients dans K. Le polynôme de $\mathbb{Q}[T, f_0, \ldots, f_{2g-1}]$ obtenu en remultipliant par tous les dénominateurs est appelé* polynôme modulaire *et noté $\Xi_\ell(T)$.*

De manière plus visuelle, quand la variable U parcourt toutes les racines de ψ_ℓ, la variable T (c'est une sorte de trace) prend un certain nombre de valeurs distinctes t_i. Le polynôme $\Xi_\ell(T)$ est alors de la forme $\alpha \prod_i (T - t_i)$.

Théorème 8.2 *Le degré en T du polynôme modulaire est*

$$\deg_T \Xi_\ell = \frac{\ell^{2g} - 1}{\ell - 1}.$$

Le polynôme modulaire est homogène si l'on met le poids 1 à la variable T et le poids $2g + 1 - i$ à la variable f_i.

Démonstration. On se place dans une clôture algébrique de K. Les $(\ell^{2g} - 1)/2$ racines de ψ_ℓ se regroupent en $d = (\ell^{2g} - 1)/(\ell - 1)$ valeurs possibles pour T. Notons $(t_i)_{1 \leqslant i \leqslant d}$ ces différentes valeurs possibles et $(u_{i,j})_{1 \leqslant i \leqslant d,\ 1 \leqslant j \leqslant (\ell-1)/2}$ les racines de ψ_ℓ correspondantes. Le polynôme minimal de la multiplication par T dans l'algèbre quotient admet tous les t_i comme racines. De plus, pour tout i, l'endomorphisme $T - t_i$ s'annule sur tous les $(u_{i,j})_{1 \leqslant j \leqslant (\ell-1)/2}$. La dimension de l'espace propre associé à t_i est donc maximale, la multiplication par T est diagonalisable, et le polynôme minimal est sans racines multiples. D'où le degré annoncé.

L'homogénéité se montre comme dans la proposition 8.3. □

8.2 Exemples en genre 1 et 2

8.2.1 Rappel du cas elliptique : CCR

Dans le cas du genre 1, le polynôme $\psi_\ell(U)$ n'est autre que le polynôme de ℓ-division. Il est de degré $(\ell^2 - 1)/2$. Pour une courbe elliptique

$$y^2 = x^3 + Ax + B,$$

les polynômes sont les suivants pour les premières valeurs de ℓ :

$$\psi_3(U) = 3U^4 + 6AU^2 + 12BU - A^2,$$

$$\begin{aligned}\psi_5(U) = {} & 7U^{24} + 308AU^{22} + 3944BU^{21} - 2954A^2U^{20} - 112ABU^{19} + (-19852A^3 - \\ & 42896B^2)U^{18} - 92568A^2BU^{17} + (-35231A^4 - 571872AB^2)U^{16} + (-31808A^3B - \\ & 829696B^3)U^{15} + (-82264A^5 - 615360A^2B^2)U^{14} + (-161840A^4B - \\ & 2132480AB^3)U^{13} + (-111916A^6 - 297472A^3B^2 - 928256B^4)U^{12} + (-608160A^5B - \\ & 2603776A^2B^3)U^{11} + (-42168A^7 - 1192800A^4B^2 - 3293696AB^4)U^{10} + \\ & (-425712A^6B - 3727360A^3B^3 - 1555456B^5)U^9 + (15673A^8 - 831936A^5B^2 - \\ & 7069440A^2B^4)U^8 + (-53824A^7B - 1314560A^4B^3 - 7127040AB^5)U^7 + (14756A^9 - \\ & 190400A^6B^2 - 2293760A^3B^4 - 2809856B^6)U^6 + (57288A^8B - 168448A^5B^3 - \\ & 3698688A^2B^5)U^5 + (1302A^{10} + 134400A^7B^2 + 394240A^4B^4 - 3039232AB^6)U^4 + \\ & (1680A^9B + 152320A^6B^3 + 831488A^3B^5 - 802816B^7)U^3 + (196A^{11} + 3696A^8B^2 + \\ & 96768A^5B^4 + 544768A^2B^6)U^2 + (392A^{10}B + 7168A^7B^3 + 64512A^4B^5 + \\ & 229376AB^7)U - A^{12} + 160A^9B^2 + 3328A^6B^4 + 24576A^3B^6 + 65536B^8.\end{aligned}$$

Les polynômes $\Xi_\ell(T)$ sont de degré $\ell+1$. Le calcul donne

$$
\begin{aligned}
\Xi_3(T) &= 3T^4+6AT^2+12BT-A^2,\\
\Xi_5(T) &= T^6+20AT^4+160BT^3-80A^2T^2-128ABT-80B^2,\\
\Xi_7(T) &= T^8+84AT^6+1512BT^5-1890A^2T^4-9072ABT^3+(-21168B^2+644A^3)T^2\\
&\quad +5832A^2BT-567A^4,\\
\Xi_{11}(T) &= T^{12}+550AT^{10}+27500BT^9-103125A^2T^8-1650000ABT^7+(-13688400B^2+\\
&\quad 645700A^3)T^6+20625000A^2BT^5+(35793120AB^2-11407385A^4)T^4+\\
&\quad (34041920B^3-58614160A^3B)T^3+(-175832976A^2B^2-2177802A^5)T^2+\\
&\quad (-235016704AB^3+1351692A^4B)T-110680064B^4+6297984A^3B^2-321651A^6.
\end{aligned}
$$

Nous retrouvons les polynômes modulaires construits par Charlap, Coley et Robbins [CCR91] dans le but d'accélérer l'algorithme de Schoof ($\Xi_\ell(T)$ est noté $U_\ell(x)$ dans leur article).

On constate que ces polynômes sont plus petits que les équations modulaires classiques $\Phi_\ell(X,Y)$ reliant les j-invariants de deux courbes ℓ-isogènes. Il y a moins de termes, et les coefficients sont des entiers plus petits. Par exemple

$$
\begin{aligned}
\Phi_3(X,Y) &= 1855425871872000000000(X+Y)-770845966336000000XY+452984832000000(X^2+Y^2)\\
&\quad +8900222976000XY(X+Y)+2587918086X^2Y^2+36864000(X^3+Y^3)\\
&\quad -1069956XY(X^2+Y^2)+2232X^2Y^2(X+Y)-X^3Y^3+X^4+Y^4,\\
\Phi_5(X,Y) &= 141359947154721358697753474691071362751004672000\\
&\quad +53274330803424425450420160273356509151232000(X+Y)\\
&\quad -264073457076620596259715790247978782949376XY\\
&\quad +6692500042627997708487149415015068467200(X^2+Y^2)\\
&\quad +36554736583949629295706472332656640000XY(X+Y)\\
&\quad +280244777828439527804321565297868800(X^3+Y^3)\\
&\quad +5110941777552418083110765199360000X^2Y^2\\
&\quad -192457934618928299655108231168000XY(X^2+Y^2)\\
&\quad +26898488858380731577417728000X^2Y^2(X+Y)+128473313284142445625344 0(X^4+Y^4)\\
&\quad +1285417989068288163840 00XY(X^3+Y^3)-441206965512914835246100X^3Y^3\\
&\quad +383083609779811215375X^2Y^2(X^2+Y^2)+107878928185336800X^3Y^3(X+Y)\\
&\quad +1963211489280(Y^5+X^5)+1665999364600X^4Y^4-246683410950XY(X^4+Y^4)\\
&\quad +2028551200X^2Y^2(X^3+Y^3)-4550940X^3Y^3(X^2+Y^2)+3720X^4Y^4(X+Y)-X^5Y^5\\
&\quad +X^6+Y^6.
\end{aligned}
$$

8.2.2 Étude du terme dominant

Dans le polynôme $\Xi_\ell(T)$, on a éliminé les dénominateurs des coefficients, le terme dominant en la variable T est donc a priori un polynôme en les f_i. Le résultat suivant donne des cas où ce terme est constant.

Proposition 8.6 *Dans le cas du genre 1, le polynôme $\Xi_\ell(T)$ est unitaire sur $\mathbb{Q}$. Il reste unitaire si on le transforme en un polynôme à coefficients dans $\mathbb{Z}$, sauf si $\ell=3$, auquel cas, le terme dominant est 3.*

Dans le cas du genre 2, le polynôme $\Xi_3(T)$ est unitaire sur $\mathbb{Q}$. Ce n'est plus le cas pour $\ell \geqslant 5$.

Démonstration. La première partie est démontrée dans [CCR91].

Étudions le cas des courbes de genre 2. Le terme dominant est un polynôme en f_0, f_1, f_2, f_3 qui s'annule lorsque l'on est dans un cas dégénéré, c'est-à-dire lorsque le nombre de sous-groupes d'ordre ℓ représentés par les racines de Ξ_ℓ est inférieur à $\ell^3+\ell^2+\ell+1$. Cela peut se produire uniquement lorsqu'il existe un diviseur de ℓ-torsion non-nul qui n'est pas de poids 2. En effet de tels diviseurs ne sont pas pris en compte dans la construction et leur présence va perturber les degrés.

Dans le cas $\ell = 3$, on est ramené à étudier le cas où $[3]P = 0$, où P est un diviseur de poids 1, c'est-à-dire un point de la courbe. On peut le réécrire $[2]P = -P$. Ainsi, si P n'est pas un point de ramification, on a deux écritures réduites d'un même diviseur, ce qui n'est pas possible. D'autre part, si P est de ramification, alors $[2]P = 0$, et ce n'est pas un élément de 3-torsion. Ainsi le terme dominant de Ξ_3 est un polynôme qui ne s'annule jamais, et est donc constant.

Dans le cas général où $\ell \geqslant 5$, l'équation $[\ell]P = 0$ peut se traduire en termes de polynômes de Cantor (cf page 107) :

$$[\ell](x,y) = 0 \iff d_1(x) = d_0(x) = 0,$$

où d_1 et d_0 sont des polynômes dont les coefficients sont des fractions rationnelles en les coefficients de la courbe. L'élimination de x dans ces deux équations produit un polynôme $\Pi_\ell(f_0, f_1, f_2, f_3)$ dont l'annulation traduit l'existence d'un diviseur de ℓ-torsion de poids 1. Ce polynôme doit donc apparaître à une certaine puissance dans le terme dominant de Ξ_ℓ. □

8.2.3 Calcul effectif de Ξ_3 en genre 2

Le polynôme modulaire $\Xi_3(T)$ en genre 2 est de la forme suivante :

$$\Xi_3(T) = T^{40} + \sum_{0 \leqslant i \leqslant 39} P_i(f_0, f_1, f_2, f_3) T^i,$$

où les P_i sont des polynômes. On a de plus une condition d'homogénéité sur Ξ_3 qui se traduit par le fait que P_i est homogène de degré $40 - i$ (avec les poids de la proposition 8.3).

Par exemple P_{39} est homogène de degré 1. Or aucun des f_i n'est de degré 1, donc $P_{39} = 0$. De même de nombreux monômes n'apparaissent pas.

Proposition 8.7 *Le nombre de monômes de degré d que l'on peut former à partir des variables T, f_0, f_1, f_2, f_3 affectées des poids respectifs $1, 5, 4, 3, 2$ est le coefficient en z^d dans le développement en série de la fonction*

$$\frac{1}{(1-z)(1-z^2)(1-z^3)(1-z^4)(1-z^5)}.$$

La valeur asymptotique de ce coefficient est $\frac{d^4}{4!5!}$.

Démonstration. C'est un problème classique que l'on trouve par exemple traité dans [FS93]. □

Pour $\ell = 3$, on a $d = 40$ et le nombre de monômes possible est 1747. Une méthode possible est donc d'instancier les f_i en suffisamment de quadruplets distincts, de calculer le polynôme modulaire pour ces valeurs en utilisant les méthodes exposées au chapitre 7, puis d'interpoler chacun des P_i.

Toutefois, Éric Schost a trouvé une méthode plus astucieuse. On commence par calculer le polynôme $\Xi_3(T)$ instancié en des valeurs aléatoires des f_i, modulo un nombre premier ($p = 9001$ par exemple). Ensuite, l'idéal I_3 sert de base de calcul pour effectuer une itération de Newton de manière à remonter les coefficients sur $\mathbb{F}_p$ en des coefficients sur $\mathbb{Q}_p$ jusqu'à une précision suffisante ; enfin, toujours par une itération de Newton, on relève ces scalaires en des polynômes en les f_i. Finalement, on reconnaît les élément de $\mathbb{Q}_p$ comme des rationnels. Cette méthode est probabiliste à plusieurs points de vue : tout d'abord, si la valeur instanciée de départ n'est pas générique, alors le calcul est entièrement faux ; cela se détecte assez facilement en pratique, car les degrés attendus sont connus. D'autre part, lors du relèvement vers $\mathbb{Q}_p$, il faut choisir une certaine précision. Pour cela, en théorie on doit avoir une estimation de la taille des coefficients de manière à être sûr de pouvoir reconnaître les bons rationnels à la fin. Les bornes théoriques étant démesurées, on se contente de remonter jusqu'à une précision à laquelle cela semble se stabiliser.

Une fois un résultat conjecturé, il est ensuite facile de vérifier qu'il a les propriétés très particulières exposées à la section suivante, ce qui suffit pour se convaincre que c'est bien le résultat cherché.

Finalement, le temps de calcul nécessaire est d'environ 15000 secondes sur une station de travail Alpha EV6 à 500 Mhz. Pour plus de détails sur ces calculs et les nombreuses techniques qui les ont rendus possibles, on consultera la thèse d'Éric Schost [Sch00].

Le polynôme $\Xi_3(T)$ est disponible à l'adresse

`http://www.lix.polytechnique.fr/Labo/Pierrick.Gaudry/publis/eqmod3g2`.

Le début est :

$$\begin{aligned}
&T^{40} + 312 f_3 T^{38} - 7904 f_2 T^{37} + (-16344 f_3^2 + 183488 f_1) T^{36} + (337536 f_3 f_2 - 4179456 f_0) T^{35} + \\
&(345456 f_3^3 + 464064 f_3 f_1 - 2381088 f_2^2) T^{34} + (-6044256 f_3^2 f_2 - 196322688 f_3 f_0 + 33608832 f_2 f_1) T^{33} + \\
&(-3966084 f_3^4 - 58451616 f_3^2 f_1 + 52440768 f_3 f_2^2 + 1324005696 f_2 f_0 - 364387392 f_1^2) T^{32} + \\
&(57911232 f_3^3 f_2 + 280858752 f_3^2 f_0 + 591085824 f_3 f_2 f_1 - 241107200 f_2^3 - 1357926912 f_1 f_0) T^{31} + \\
&(26487216 f_3^5 + 388684800 f_3^3 f_1 - 378838944 f_3^2 f_2^2 + 8135750592 f_3 f_2 f_0 - 8711658240 f_3 f_1^2 + \\
&1407612480 f_2^2 f_1 - 30436734528 f_0^2) T^{30} + (-308962080 f_3^4 f_2 + 15709314816 f_3^3 f_0 - 7039348992 f_3^2 f_2 f_1 + \\
&1847884416 f_3 f_2^3 - 21546021888 f_3 f_1 f_0 - 4295003904 f_2^2 f_0 + 7496418816 f_2 f_1^2) T^{29} + (-97855128 f_3^6 + \\
&717110304 f_3^4 f_1 + 731068032 f_3^3 f_2^2 - 75564061056 f_3^2 f_2 f_0 - 47621349504 f_3^2 f_1^2 + 55550744448 f_3 f_2^2 f_1 + \\
&141450046848 f_3 f_0^2 - 8868753152 f_2^4 - 240969913344 f_2 f_1 f_0 + 76579431936 f_1^3) T^{28} + (815181696 f_3^5 f_2 - \\
&7457389056 f_3^4 f_0 - 11717124096 f_3^3 f_2 f_1 + 4346795520 f_3^2 f_2^3 + 144606394368 f_3^2 f_1 f_0 - \\
&71162772480 f_3 f_2^2 f_0 + 16981051392 f_3 f_2 f_1^2 - 11416397824 f_2^3 f_1 + 131172009984 f_2 f_0^2 - \\
&150350708736 f_1^2 f_0) T^{27} + (140091120 f_3^7 - 9497570112 f_3^5 f_1 + 1617994080 f_3^4 f_2^2 - \\
&202079060352 f_3^3 f_2 f_0 - 85220137728 f_3^3 f_1^2 + 146302604928 f_3^2 f_2^2 f_1 + 357349905792 f_3^2 f_0^2 - \\
&35760872448 f_3 f_2^4 - 857345121792 f_3 f_2 f_1 f_0 + 254363526144 f_3 f_1^3 + 16342396416 f_2^3 f_0 + \\
&4166138880 f_2^2 f_1^2 + 1294107121152 f_1 f_0^2) T^{26} + \cdots
\end{aligned}$$

Calcul pour $\ell > 3$

Le degré en T du polynôme $\Xi_\ell(T)$ est $d = \ell^3 + \ell^2 + \ell + 1$. En négligeant le fait que le degré total croît encore à cause du terme dominant, on a déjà un nombre de monômes qui est en $O(\ell^{12})$. Par exemple, pour $\ell = 5$, on a $d = 156$ et le nombre de monômes est de l'ordre de 250000. Le calcul paraît donc beaucoup plus compliqué que pour $\ell = 3$. Et dès $\ell = 7$, on commence à avoir des problèmes pour *stocker* l'hypothétique résultat !

8.2.4 Remarques sur $\ell = 2$

Le polynôme Ξ_ℓ n'a été défini que pour ℓ impair. On pourrait se demander quel est son équivalent pour $\ell = 2$. En effet, du point de vue des isogénies, le degré 2 ne présente rien de particulier. La restriction que l'on a effectuée est due au fait que l'on quotiente les sous-groupes de ℓ-torsion par l'involution hyperelliptique, ce qui n'a pas de sens pour $\ell = 2$ où l'opposé d'un élément est l'élément lui-même. Déjà dans le cas elliptique, il est courant de séparer les polynômes de division de degré pair et impair.

Dans le cas qui nous intéresse, on classifie les isogénies de degré ℓ dans le but de déterminer le cardinal de la Jacobienne modulo ℓ. Pour $\ell = 2$, nous avons vu au chapitre précédent que la factorisation du polynôme $f(x)$ de l'équation de la courbe fournit déjà les éléments de 2-torsion, et donc le cardinal modulo 2. D'une certaine manière, on peut dire que le polynôme Ξ_2 n'est autre que le polynôme $f(x)$, étant donné que l'on y lit les éléments de 2-torsion, et donc les noyaux des 2-isogénies.

8.3 Motifs de factorisation

Un fois le polynôme Ξ_ℓ précalculé, on peut substituer les valeurs $f_0, f_1, \ldots$ correspondant à une courbe que l'on veut étudier. On suppose que le corps de base est un corps fini. Nous allons voir que la manière dont Ξ_ℓ se décompose en facteurs irréductibles est alors très liée au comportement du Frobenius sur les éléments de ℓ-torsion, et donc au cardinal de la Jacobienne modulo ℓ.

8.3.1 Théorie

Soit $\mathcal{C}$ une courbe hyperelliptique de genre g définie sur un corps fini $\mathbb{F}_q$. On note encore $\Xi_\ell(T)$ le polynôme modulaire dans lequel on a substitué les coefficients de $\mathcal{C}$.

Théorème 8.3 *Soit D un diviseur de ℓ-torsion de poids g dans $\mathrm{Jac}(\mathcal{C})$ sur une clôture algébrique. Soit V l'espace vectoriel sur $\mathbb{F}_\ell$ engendré par les conjugués de D :*

$$V = \mathrm{Vect}_{\mathbb{F}_\ell} \left\{ \pi^n(D);\ n \in \mathbb{N} \right\}.$$

Soit $P(t)$ le polynôme caractéristique de π restreint à $V \subseteq \mathrm{Jac}[\ell]$.

Alors le degré de l'extension dans laquelle se trouve la racine de $\Xi_\ell(T)$ associée au sous-groupe engendré par D est (divisible par)

$$\mathrm{ord}^*(P) = \min \left\{ k \in \mathbb{N}^* \textit{ tel que } \deg \left(t^k \bmod P(t) \right) = 0 \right\}.$$

Démonstration.

Le degré de l'extension dans laquelle se trouve la racine associée à D est l'entier k minimum tel que $\pi^k(D) = \lambda D$, pour un λ quelconque de $\mathbb{F}_\ell^*$. En effet cela équivaut à $\pi^k(\langle D \rangle) = \langle D \rangle$.

L'existence d'un tel λ pour un k donné se traduit par le fait que le degré de $t^k \bmod P(t)$ est nul. □

Remarque. Comme nous l'avons noté, l'extension est a priori seulement divisible par $\mathrm{ord}^*(P)$. En effet, $\Xi_\ell(T)$ est le polynôme minimal d'un paramètre qui ne prend en compte qu'une seule des coordonnées des diviseurs en représentation de Mumford. Il se peut que cette coordonnée soit dans un sous-corps du corps de définition du diviseur, auquel cas un des facteurs va se décomposer en conséquence. Ce phénomène ne se produit que très rarement en général et jamais pour les courbes elliptiques. Dans la suite, nous supposerons toujours qu'il n'y a pas de telle décomposition.

Il reste ensuite à déterminer le nombre de diviseurs de ℓ-torsion D qui correspondent à chaque facteur possible du polynôme caractéristique. Ceci est aisé dès que l'on connaît la réduction de π en temps qu'endomorphisme de $\mathbb{F}_\ell$-espace vectoriel.

Corollaire 8.1 *La réduction de π restreint à* $\mathrm{Jac}[\ell]$ *détermine*

1. *le polynôme caractéristique de π modulo ℓ, donc aussi* $\#\mathrm{Jac} \bmod \ell$,
2. *le motif de factorisation de $\Xi_\ell(T)$.*

Dans le contexte du calcul de la cardinalité, on part du motif de factorisation de $\Xi_\ell(T)$, cela restreint les choix possibles pour la réduction de π, et donc $\#\mathrm{Jac} \bmod \ell$ ne peut prendre que certaines valeurs.

Le cas elliptique est bien connu (au moins pour les polynômes modulaires classiques) : on trouve par exemple dans [Sch95] le théorème suivant (cf aussi [Ler97]).

Théorème 8.4 *Soit E une courbe elliptique non-supersingulière définie sur $\mathbb{F}_q$, d'invariant j_E différent de* 0 *et* 1728. *Soit $\chi(t) = t^2 - s_1 t + q$ le polynôme caractéristique du Frobenius sur E et soit $f_1(T) \cdots f_s(T)$ la factorisation dans $\mathbb{F}_q[T]$ de $\Phi_\ell(j_E, T)$. Alors les ensembles de degrés possibles pour les f_i sont*

1. $(1, \ell)$ *ou* $(1, 1, \ldots, 1)$ *si $\chi(t)$ est un carré modulo ℓ, i.e. si* $s_1^2 - 4q \equiv 0 \bmod \ell$,
2. $(1, 1, r, \ldots, r)$ *si $\chi(t)$ est scindé modulo ℓ, i.e. si $s_1^2 - 4q$ est un carré de $\mathbb{F}_\ell^*$,*
3. $(r, \ldots, r)$ *si $\chi(t)$ est irréductible modulo ℓ, i.e. si $s_1^2 - 4q$ n'est pas un carré de $\mathbb{F}_\ell^*$.*

L'entier r est le plus petit entier tel que

$$\forall P \in E[\ell],\ \exists \kappa \in \mathbb{N},\ \pi^r(P) = \kappa P.$$

8.3.2 Table des motifs pour $g = 2$

Nous donnons ici toutes les réductions possibles de l'endomorphisme π ainsi que les motifs associés dans le cas d'une courbe de genre 2 dont la Jacobienne est simple.

Nous traitons complètement le cas où la matrice de π est semblable à la matrice

$$M = \begin{array}{|c|c|}\hline A_2 & 0 \\ \hline 0 & \begin{array}{|c|c|}\hline b & 0 \\ \hline 0 & b \\ \hline \end{array} \\ \hline \end{array},$$

où A_2 désigne une matrice 2×2 dont le polynôme caractéristique est irréductible sur $\mathbb{F}_\ell$, et b est un scalaire. Les autres cas s'étudient de manière similaire.

Désignons par (D_1, D_2, D_3, D_4) une base de $\mathrm{Jac}[\ell]$ dans laquelle la matrice de π est comme ci-dessus.

Pour tout $0 \neq D \in \langle D_3, D_4 \rangle$, on a $\pi(D) = bD$, et les sous-groupes d'ordre ℓ inclus dans $\langle D_3, D_4 \rangle$ correspondent donc à des racines de $\Xi_\ell(T)$ définies sur $\mathbb{F}_\ell$. Leur nombre est égal au nombre de droites dans $\langle D_3, D_4 \rangle$, c'est-à-dire $\frac{\ell^2-1}{\ell-1} = \ell + 1$.

Pour tout $0 \neq D \in \langle D_1, D_2 \rangle$, l'espace vectoriel engendré par les conjugués de D est $V = \langle D_1, D_2 \rangle$ et le polynôme caractéristique de π restreint à V est celui de la matrice A_2. La racine de $\Xi_\ell(T)$ associée au sous-groupe engendré par D est donc dans une extension de degré ord*(Polynôme caractéristique de A_2), que l'on note ord(A_2) pour simplifier. Leur nombre est là encore égal à $\ell+1$ et donc ord(A_2) divise $\ell+1$. Finalement, on a $(\ell+1)/\mathrm{ord}(A_2)$ facteurs de degré ord(A_2) dans $\Xi_\ell(T)$.

Pour tout $D = E + F$ où $0 \neq E \in \langle D_1, D_2\rangle$ et $0 \neq F \in \langle D_3, D_4\rangle$, l'espace vectoriel engendré par les conjugués de D est $V = \langle D_1, D_2, F\rangle$ et le polynôme caractéristique de π sur V est celui de A_2 multiplié par $(X-b)$. On note ord$(A_2 b)$ la valeur de ord* en ce polynôme ; c'est le degré de l'extension dans laquelle vivent les racines. Le nombre de droites engendrées par de tels D est $(\ell-1)(\ell+1)^2$.

En résumé, $\Xi_\ell(T)$ contient

- $\ell+1$ facteurs linéaires,
- $(\ell+1)/\mathrm{ord}(A_2)$ facteurs de degré ord(A_2),
- $(\ell-1)(\ell+1)^2/\mathrm{ord}(A_2 b)$ facteurs de degré ord$(A_2 b)$.

Avant de donner une table de tous les motifs possibles correspondant à toutes les réductions possibles de la matrice de π, on peut éliminer certains cas qui ne peuvent de toute façon pas se produire, ou alors seulement dans des cas dégénérés.

Théorème 8.5 *Soit $\mathcal{C}$ une courbe de genre 2 sur $\mathbb{F}_q$ dont la Jacobienne est simple et soit ℓ un nombre premier. Alors $\chi(t) \bmod \ell$ n'a pas de facteur irréductible de degré 3. De plus si $\chi(t) \bmod \ell$ a une racine triple, alors c'est obligatoirement une racine quadruple.*

Démonstration. (Merci à P. Mihăilescu pour ses remarques.)

La Jacobienne de $\mathcal{C}$ est simple donc $\chi(t)$ est irréductible et le corps $\mathbb{Q}[t]/(\chi(t))$ est à multiplication complexe : c'est une extension quadratique imaginaire d'un corps quadratique réel noté K_0. Ainsi les racines de $\chi(t)$ s'obtiennent par le procédé suivant : on calcule les deux racines τ_1 et τ_2 du polynôme $P(\tau)$ définissant K_0, puis pour chaque τ_i, on doit calculer les deux racines d'un polynôme Q_{τ_i} à coefficients dans $\mathbb{Q}(\tau_i)$ de degré 2.

On opère maintenant de même pour chercher les racines de $\chi(t)$ modulo ℓ :

- Si $P(\tau)$ est irréductible modulo ℓ, alors τ_1 et τ_2 sont conjugués dans $\mathbb{F}_{\ell^2}$.
  - Si $Q_{\tau_1}(t)$ est irréductible sur $\mathbb{F}_{\ell^2}$, alors ses racines sont dans $\mathbb{F}_{\ell^4}$ et $\chi(t)$ est irréductible modulo ℓ.
  - Si $Q_{\tau_1}(t) = (t-t_1)(t-t_2)$ avec t_1 et t_2 dans $\mathbb{F}_{\ell^2}$, alors leurs conjugués sont les racines de $Q_{\tau_2}(t)$, et $\chi(t)$ a deux facteurs irréductibles de degré 2.
- Si $P(\tau)$ a deux facteurs $(\tau-\tau_1)$ et $(\tau-\tau_2)$ dans $\mathbb{F}_\ell[\tau]$,
  - Si $Q_{\tau_i}(t)$ est irréductible sur $\mathbb{F}_\ell$, alors ses racines sont conjuguées et forment un facteur irréductible de degré 2 de $\chi(t) \bmod \ell$.
  - Si $Q_{\tau_i}(t)$ a deux racines t_1 et t_2 dans $\mathbb{F}_\ell$, on a deux facteurs linéaires dans $\chi(t) \bmod \ell$.

Ainsi aucun facteur de degré 3 ne peut apparaître.

Supposons maintenant que $\chi(t)$ se scinde sur $\mathbb{F}_\ell$ en $(t-t_1)(t-t_2)^3$. Alors le théorème de Weil assure que

$$t_1 t_2 \equiv q \bmod \ell, \quad \text{et} \quad t_2^2 \equiv q \bmod \ell.$$

Mais alors $t_2(t_2 - t_1) \equiv 0 \bmod \ell$, et pour déduire $t_1 \equiv t_2 \bmod \ell$, il suffit de s'assurer que t_2 ne s'annule pas modulo ℓ. L'annulation de t_2 impliquerait celle de q modulo ℓ, ce qui contredit que ℓ est un nombre premier distinct de la caractéristique. $\square$

Pol. carac.	Matrice	Motif du polynôme modulaire
Un facteur de degré 4		
(4)	$\boxed{A_4}$	$(\underbrace{\mathrm{ord}(A_4),\dots,\mathrm{ord}(A_4)}_{\ell^3+\ell^2+\ell+1})$
Deux facteurs de degré 2		
$(2)^2$	$\begin{array}{\|c\|c\|}\hline A_2 & 0 \\ \hline 0 & A_2 \\ \hline\end{array}$	$(\underbrace{\mathrm{ord}(A_2),\dots,\mathrm{ord}(A_2)}_{\ell^3+\ell^2+\ell+1})$
$(2)^2$	$\begin{array}{\|c\|c\|}\hline A_2 & * \\ \hline 0 & A_2 \\ \hline\end{array}$	$(\underbrace{\mathrm{ord}(A_2),\dots,\mathrm{ord}(A_2)}_{\ell+1},\ \underbrace{\mathrm{ord}(A_2^2),\dots,\mathrm{ord}(A_2^2)}_{\ell^2(\ell+1)})$
$(2)(2)$	$\begin{array}{\|c\|c\|}\hline A_2 & 0 \\ \hline 0 & B_2 \\ \hline\end{array}$	$(\underbrace{\mathrm{ord}(A_2),\dots,\mathrm{ord}(A_2)}_{\ell+1},\ \underbrace{\mathrm{ord}(B_2),\dots,\mathrm{ord}(B_2)}_{\ell+1},\ \underbrace{\mathrm{ord}(A_2B_2),\dots,\mathrm{ord}(A_2B_2)}_{(\ell-1)(\ell+1)^2})$
Un facteur de degré 2, deux de degré 1		
$(2)(1)^2$	$\begin{array}{\|c\|c\|}\hline A_2 & 0 \\ \hline 0 & \begin{array}{c\|c} b & 0 \\ \hline 0 & b \end{array} \\ \hline\end{array}$	$(\underbrace{1,\dots,1}_{\ell+1},\ \underbrace{\mathrm{ord}(A_2),\dots,\mathrm{ord}(A_2)}_{\ell+1},\ \underbrace{\mathrm{ord}(A_2b),\dots,\mathrm{ord}(A_2b)}_{(\ell-1)(\ell+1)^2})$
$(2)(1)^2$	$\begin{array}{\|c\|c\|}\hline A_2 & 0 \\ \hline 0 & \begin{array}{c\|c} b & * \\ \hline 0 & b \end{array} \\ \hline\end{array}$	$(1,\ \ell,\ \underbrace{\mathrm{ord}(A_2),\dots,\mathrm{ord}(A_2)}_{\ell+1},\ \underbrace{\mathrm{ord}(A_2b),\dots,\mathrm{ord}(A_2b)}_{(\ell-1)(\ell+1)},\ \underbrace{\mathrm{ord}(A_2b^2),\dots,\mathrm{ord}(A_2b^2)}_{\ell(\ell-1)(\ell+1)})$
$(2)(1)(1)$	$\begin{array}{\|c\|c\|}\hline A_2 & 0 \\ \hline 0 & \begin{array}{c\|c} b & 0 \\ \hline 0 & c \end{array} \\ \hline\end{array}$	$(1,\ 1,\ \underbrace{\mathrm{ord}(A_2),\dots,\mathrm{ord}(A_2)}_{\ell+1},\ \underbrace{\mathrm{ord}(bc),\dots,\mathrm{ord}(bc)}_{\ell-1},\ \underbrace{\mathrm{ord}(A_2b),\dots,\mathrm{ord}(A_2b)}_{(\ell+1)(\ell-1)},$ $\underbrace{\mathrm{ord}(A_2c),\dots,\mathrm{ord}(A_2c)}_{(\ell+1)(\ell-1)},\ \underbrace{\mathrm{ord}(A_2bc),\dots,\mathrm{ord}(A_2bc)}_{(\ell-1)^2(\ell+1)})$

Pol. carac.	Matrice	Motif du polynôme modulaire
Quatre facteurs de degré 1		
$(1)^4$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & 0 \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} a & 0 \\ \hline 0 & a \end{array} \\ \hline \end{array}$	$(\underbrace{1, \dots, 1}_{\ell^3+\ell^2+\ell+1})$
$(1)^4$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & * \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} a & 0 \\ \hline 0 & a \end{array} \\ \hline \end{array}$	$(\underbrace{1, \dots, 1}_{\ell^2+\ell+1}, \underbrace{\ell, \dots, \ell}_{\ell^2})$
$(1)^4$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & * \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} a & * \\ \hline 0 & a \end{array} \\ \hline \end{array}$	$(\underbrace{1, \dots, 1}_{\ell+1} \underbrace{\ell, \dots, \ell}_{\ell^2+\ell})$
$(1)^2(1)^2$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & 0 \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} b & 0 \\ \hline 0 & b \end{array} \\ \hline \end{array}$	$(\underbrace{1, \dots, 1}_{2\ell+2}, \underbrace{\mathrm{ord}(ab), \dots, \mathrm{ord}(ab)}_{(\ell-1)(\ell+1)^2})$
$(1)^2(1)^2$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & * \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} b & 0 \\ \hline 0 & b \end{array} \\ \hline \end{array}$	$(\underbrace{1, \dots, 1}_{\ell+2}, \ell, \underbrace{\mathrm{ord}(ab), \dots, \mathrm{ord}(ab)}_{(\ell-1)(\ell+1)}, \underbrace{\mathrm{ord}(a^2b), \dots, \mathrm{ord}(a^2b)}_{\ell(\ell-1)(\ell+1)})$
$(1)^2(1)^2$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & * \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} b & * \\ \hline 0 & b \end{array} \\ \hline \end{array}$	$(1, 1, \ell, \ell, \underbrace{r, \dots, r}_{\ell-1}, \underbrace{s_1, \dots, s_1}_{\ell(\ell-1)}, \underbrace{s_2, \dots, s_2}_{\ell(\ell-1)}, \underbrace{t, \dots, t}_{\ell^2(\ell-1)})$ où $r = \mathrm{ord}(ab)$, $s_1 = \mathrm{ord}(ab^2)$, $s_2 = \mathrm{ord}(a^2b)$, $t = \mathrm{ord}(a^2b^2)$
$(1)^2(1)(1)$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & 0 \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} b & 0 \\ \hline 0 & c \end{array} \\ \hline \end{array}$	$(\underbrace{1, \dots, 1}_{\ell+3}, \underbrace{r, \dots, r}_{\ell-1}, \underbrace{s_1, \dots, s_1}_{\ell^2-1}, \underbrace{s_2, \dots, s_2}_{\ell^2-1}, \underbrace{t, \dots, t}_{(\ell^2-1)(\ell-1)})$ où $r = \mathrm{ord}(bc)$, $s_1 = \mathrm{ord}(ab)$, $s_2 = \mathrm{ord}(ac)$, $t = \mathrm{ord}(abc)$
$(1)^2(1)(1)$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & * \\ \hline 0 & a \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} b & 0 \\ \hline 0 & c \end{array} \\ \hline \end{array}$	$\Big(1, 1, 1, \ell, \underbrace{r_1, \dots, r_1}_{\ell-1}, \dots, \underbrace{r_3, \dots, r_3}_{\ell-1}, \underbrace{s_1, \dots, s_1}_{\ell(\ell-1)}, \underbrace{s_2, \dots, s_2}_{\ell(\ell-1)},$ $\underbrace{t, \dots, t}_{(\ell-1)^2} \underbrace{u, \dots, u}_{\ell(\ell-1)^2}\Big)$ où $r_1 = \mathrm{ord}(bc)$, $r_2 = \mathrm{ord}(ab)$, $r_3 = \mathrm{ord}(ac)$, $s_1 = \mathrm{ord}(a^2b)$, $s_2 = \mathrm{ord}(a^2c)$, $t = \mathrm{ord}(abc)$, $u = \mathrm{ord}(a^2bc)$
$(1)(1)(1)(1)$	$\begin{array}{\|c\|c\|}\hline \begin{array}{c\|c} a & 0 \\ \hline 0 & b \end{array} & 0 \\ \hline 0 & \begin{array}{c\|c} c & 0 \\ \hline 0 & d \end{array} \\ \hline \end{array}$	$\Big(1, 1, 1, 1, \underbrace{r_1, \dots, r_1}_{\ell-1}, \dots, \underbrace{r_6, \dots, r_6}_{\ell-1}, \underbrace{s_1, \dots, s_1}_{(\ell-1)^2}, \dots,$ $\underbrace{s_4, \dots, s_4}_{(\ell-1)^2}, \underbrace{t, \dots, t}_{(\ell-1)^3}\Big)$ où $r_1 = \mathrm{ord}(ab)$, $r_2 = \mathrm{ord}(ac)$, $r_3 = \mathrm{ord}(ad)$, $r_4 = \mathrm{ord}(bc)$, $r_5 = \mathrm{ord}(bd)$, $r_6 = \mathrm{ord}(cd)$, $s_1 = \mathrm{ord}(abc)$, $s_2 = \mathrm{ord}(abd)$, $s_3 = \mathrm{ord}(acd)$, $s_4 = \mathrm{ord}(bcd)$, $t = \mathrm{ord}(abcd)$

Cas de $\ell = 3$

Dans le cas de $\ell = 3$, on peut regarder pour chacun des cas du tableau précédent les différentes valeurs numériques possibles. En regroupant tout cela selon le cardinal de la Jacobienne modulo 3, on obtient le théorème suivant :

Théorème 8.6 *Soit $\mathcal{C}$ une courbe de genre 2 sur un corps fini $\mathbb{F}_q$ de caractéristique p différente de 3. Alors les motifs de factorisation de $\Xi_3(T)$ impliquent les valeurs de $\#\mathrm{Jac}(\mathcal{C}) \bmod 3$ données dans les tableaux suivants :*

$q \equiv 2 \bmod 3$	
motif	$\#\mathrm{Jac}(\mathcal{C}) \bmod 3$
$(10)^4$	2
$(2)^2(6)^6$	1
$(4)(12)^3$	1
$(2)^{20}$	1
$(1)^2(2)(4)(8)^4$	0
$(1)^2(2)(3)^2(6)^5$	0
$(1)^8(2)^{16}$	0
$(1)^5(2)^4(3)(6)^4$	0
$(4)^{10}$	$1, 2$

$q \equiv 1 \bmod 3$	
motif	$\#\mathrm{Jac}(\mathcal{C}) \bmod 3$
$(5)^8$	$1, 2$
$(1)(2)^2(3)(4)^2(12)^2$	$0, 2$
$(1)^4(2)^2(4)^8$	$0, 2$
$(1)(3)^4(9)^3$	$0, 1$
$(1)^4(3)^{12}$	$0, 1$
$(1)^{40}$	$0, 1$
$(4)^{10}$	2
$(2)^2(6)^6$	1
$(2)^{20}$	1
$(1)^4(3)^{12}$	0
$(1)^2(2)(3)^2(6)^5$	0
$(1)^5(2)^4(3)(6)^4$	0
$(1)^8(2)^{16}$	0

8.3.3 Vérification expérimentale

Soit $\mathcal{C}$ une courbe aléatoire sur un petit corps fini. On peut alors calculer le motif de factorisation de $\Xi_3(T)$ pour cette courbe ainsi que le polynôme caractéristique du Frobenius par une méthode naïve, puisque le corps est petit. Nous avons ainsi trouvé des exemples pour chacun des motifs et chacun des polynômes caractéristiques possibles, sauf pour quelques cas très rares où le nombre de racines est grand. Ces exemples sont regroupés dans les tableaux 8.1 et 8.2.

8.4 Application au calcul de cardinalité

Algorithme à la Atkin

Dans le cas des courbes elliptiques, Atkin [Atk92] a proposé d'utiliser les motifs de factorisation du polynôme modulaire afin de compter le nombre de points. Le principe est complètement indépendant du genre :

1. Tant que l'on n'a pas suffisamment d'information, faire
   (a) Choisir un nouveau ℓ premier ;
   (b) Calculer le motif de factorisation de $\Xi_\ell(T)$;
   (c) Calculer les valeurs de $\chi(t) \bmod \ell$ compatibles avec ce motif ;
2. Déterminer la seule valeur possible de $\chi(t)$ compatible avec les informations modulaires.

$f(x)$	$\chi(t) \bmod \ell$	motif de $\Xi_\ell(T)$
$x^5+24x^3+17x+34$	$(t+1)(t+2)(t^2+2t+2)$	$(1)^2(2)(4)(8)^4$
$x^5+33x^3+2x^2+2x+21$	$(t+1)(t+2)(t^2+t+2)$	$(1)^2(2)(4)(8)^4$
$x^5+24x^3+17x^2+35x+26$	$(t+1)^2(t+2)^2$	$(1)^2(2)(3)^2(6)^5$
$x^5+x^3+34x^2+30x+8$	t^4+t^3+2t+1	$(10)^4$
$x^5+29x^3+40x^2+26x+13$	t^4+2t^3+t+1	$(10)^4$
$x^5+12x^3+7x^2+3x+9$	$(t^2+t+2)(t^2+2t+2)$	$(4)^{10}$
$x^5+39x^3+9x^2+26x+5$	$(t^2+1)^2$	$(2)^2(6)^6$
$x^5+27x^3+2x^2+5x+38$	$(t^2+2t+2)^2$	$(4)(12)^3$
$x^5+8x^3+16x^2+40x+4$	$(t^2+t+2)^2$	$(4)(12)^3$
$x^5+8x^3+14x^2+17x+31$	$(t+1)^2(t+2)^2$	$(1)^8(2)^{16}$
$x^5+2x^3+2x^2+25x+19$	$(t^2+t+2)^2$	$(4)^{10}$
$x^5+10x^3+39x^2+25x+32$	$(t^2+2t+2)^2$	$(4)^{10}$
$x^5+35x^3+12x^2+19x+1$	$(t^2+1)^2$	$(2)^{20}$

TAB. 8.1: *Exemples sur* $\mathbb{F}_p$ *avec* $p = 41 \equiv 2 \bmod 3$

L'utilisation de cet algorithme nécessite d'avoir précalculé les polynômes modulaires, ce qui est tout à fait faisable en genre 1.

Supposons pour le moment que nous disposons du polynôme $\Xi_\ell(T)$ générique. Le coût du calcul modulo ℓ se décompose ainsi :

1. L'évaluation du polynôme en les coefficients de la courbe nécessite environ autant d'opérations qu'il y a de monômes.
2. La factorisation d'un polynôme de degré $O(\ell^{2g-1})$, ou du moins la détection de son motif de factorisation (cf [Sho95]).
3. Pour chaque valeur de $\chi(t) \bmod \ell$, déterminer si le motif correspond.

La phase 2 et la phase 3 coûtent moins cher que l'algorithme de Schoof dans lequel on doit travailler avec des polynômes de degré $O(\ell^{2g})$, en supposant que comme pour le genre 2 on a réussi à éviter tout calcul avec les idéaux.

Par contre, la phase 1 peut devenir problématique. Dans le cas des courbes elliptiques, le nombre de monômes est en $O(\ell^2)$, donc l'évaluation n'est pas un problème. En genre 2, le nombre de monômes croît en $O(\ell^{12})$, ainsi l'évaluation devient la part la plus coûteuse de l'algorithme, loin devant la factorisation. La constante dans le $O()$ est petite, si bien que le cas $\ell = 3$ est encore faisable. Toutefois, dès $\ell = 5$ le coût risque d'être prohibitif.

Comment faire baisser la complexité

Un moyen de parvenir à faire baisser la complexité plus avant serait de calculer $\Xi_\ell(T)$ instancié directement avec la courbe dont on a besoin en utilisant un nombre d'opérations moindre que pour construire l'idéal de ℓ-division.

L'objectif est le suivant : partant du système 7.2 provenant des polynômes de Cantor, qui s'exprime à l'aide de polynômes univariés de degré $O(\ell^2)$, on veut obtenir un polynôme univarié de degré $O(\ell^3)$ entièrement déterminé par ce système. Il ne paraît pas inenvisageable de trouver une méthode qui permette d'effectuer le calcul sans jamais utiliser de polynôme de degré $O(\ell^4)$,

$f(x)$	$\chi(t)$ mod ℓ	motif de $\Xi_\ell(T)$
$x^5+58x^3+34x^2+29x+44$	$(t^2+1)^2$	$(2)^2(6)^6$
$x^5+21x^3+20x^2+42x+15$	$(t^2+1)^2$	$(2)^{20}$
$x^5+41x^3+18x^2+7x+24$	$(t+1)^2(t^2+1)$	$(1)^4(2)^2(4)^8$
$x^5+60x^3+5x^2+10x+5$	$(t+1)^2(t^2+1)$	$(1)(2)^2(3)(4)^2(12)^2$
$x^5+30x^3+32x^2+46x+4$	$(t+2)^2(t^2+1)$	$(1)(2)^2(3)(4)^2(12)^2$
$x^5+56x^3+17x^2+5x+10$	$(t+2)^2(t^2+1)$	$(1)^4(2)^2(4)^8$
$x^5+17x^3+50x^2+10x+38$	$(t+1)^4$	$(1)(3)^4(9)^3$
$x^5+15x^3+5x^2+33x+40$	$(t+1)^4$	$(1)^4(3)^{12}$
$x^5+34x^3+24x^2+9x+3$	$(t+2)^4$	$(1)(3)^4(9)^3$
$x^5+8x^3+30x^2+56x+42$	$(t+2)^4$	$(1)^4(3)^{12}$
$x^5+x^3+6x^2+4x+20$	$(t+1)^2(t+2)^2$	$(1)^2(2)(3)^2(6)^5$
$x^5+39x^3+51x^2+20x+29$	$(t+1)^2(t+2)^2$	$(1)^5(2)^4(3)(6)^4$
$x^5+56x^3+27x^2+4x+11$	$(t+1)^2(t+2)^2$	$(1)^8(2)^{16}$
$x^5+26x^3+31x^2+44x+18$	$t^4+2t^3+t^2+2t+1$	$(5)^8$
$x^5+58x^3+36x^2+55x$	$t^4+t^3+t^2+t+1$	$(5)^8$
$x^5+2x^3+7x^2+56x+12$	$(t^2+t+2)(t^2+2t+2)$	$(4)^{10}$

TAB. 8.2: *Exemples sur* $\mathbb{F}_p$ *avec* $p = 61 \equiv 1 \bmod 3$

ce qui signifierait très probablement une complexité en $O(M(\ell^3))$, c'est-à-dire bien plus faible que la factorisation qui suivrait.

Pour le moment, nous n'avons pas trouvé le moyen de faire cela.

Troisième partie

Logarithme discret

Chapitre 9

État de l'art du calcul du log discret dans les Jacobiennes

9.1 Méthodes pour un groupe générique

Définition 9.1 *Le* problème du logarithme discret *est le suivant :*

Soit G un groupe abélien fini noté additivement. Soit D_1 un élément de G d'ordre N et D_2 un élément du sous-groupe cyclique engendré par D_1. Le but est de trouver l'entier λ dans l'intervalle $[0, N-1]$ tel que

$$D_2 = \lambda \cdot D_1.$$

L'entier λ est appelé logarithme discret de D_2 en base D_1, *et est noté* $\log_{D_1}(D_2)$.

Au chapitre 5, nous avons vu des algorithmes permettant de calculer le cardinal d'un groupe générique, c'est-à-dire des algorithmes qui ne s'appuient que sur la structure de groupe, sans utiliser de propriétés plus particulières au groupe qu'on étudie. Nous allons décrire le même type d'algorithme, mais cette fois ci dans le but de résoudre le problème du logarithme discret. Notons qu'il existe une borne inférieure sur le nombre moyen d'opérations dans le groupe qu'un tel algorithme générique doit faire pour résoudre le log discret [Nec94, Sho97]. Les algorithmes ci-dessous atteignent cette borne et sont donc en ce sens optimaux.

9.1.1 Réduction de Pohlig-Hellman

Pohlig et Hellman [PH78] ont ramené le problème du logarithme discret dans un groupe cyclique d'ordre quelconque à des problèmes de log discret dans des sous-groupes d'ordre premier divisant l'ordre du groupe initial.

Soit $N = \prod p_i^{e_i}$ la décomposition en facteurs premiers de l'ordre de D_1. Nous supposons que cette factorisation est connue, ce qui n'est pas déraisonnable compte-tenu du fait qu'actuellement les algorithmes de factorisation sont bien plus efficaces que ceux de logarithme discret générique.

On veut résoudre $D_2 = \lambda \cdot D_1$. Pour tout i, on multiplie cette équation par le cofacteur de $p_i^{e_i}$:

$$\frac{N}{p_i^{e_i}} \cdot D_2 = (\lambda \bmod p_i^{e_i}) \, \frac{N}{p_i^{e_i}} \cdot D_1,$$

où l'entier λ est pris modulo l'ordre de $B_i = \frac{N}{p_i^{e_i}} \cdot D_1$. Ainsi, trouver $\lambda \bmod p_i^{e_i}$ est faisable par un calcul de log discret dans le sous-groupe engendré par B_i, dont l'ordre est $p_i^{e_i}$. Si l'on a effectué ce calcul pour tout i, il est alors facile de reconstruire λ par le théorème Chinois.

On a donc ramené un calcul de log discret dans un groupe d'ordre composé N à un nombre polynomial de calculs de log discrets dans des sous-groupes d'ordre des puissances de nombres premiers. On suppose donc dorénavant que l'ordre de D_1 est $N = p^e$, où p est un nombre premier.

Écrivant λ en base p, on a

$$D_2 = \lambda D_1 = \left(\lambda_0 + p\lambda_1 + \cdots + \lambda_{e-1}p^{e-1}\right) D_1.$$

On multiplie alors ceci par p^{e-1} et l'on obtient

$$p^{e-1}D_2 = \lambda_0 p^{e-1} D_1.$$

Trouver λ_0 revient donc à calculer un log discret dans le sous-groupe engendré par $p^{e-1}D_1$ qui est d'ordre p. Supposons que cela a été effectué, on peut alors calculer $D_2' = D_2 - \lambda_0 D_1$. Cet élément vérifie

$$D_2' = \left(\lambda_1 + \cdots + \lambda_{e-1}p^{e-2}\right) pD_1,$$

et les autres coefficients de λ peuvent être calculé par une résolution de log discret dans le sous-groupe engendré par pD_1 dont l'ordre est p^{e-1}.

En itérant ce processus, on voit facilement que le calcul de λ revient à résoudre e log discret dans des sous-groupes d'ordre p.

Ainsi, au prix d'un nombre polynomial d'opérations dans le groupe, un calcul de log discret se ramène à $O(\log N)$ calculs de log discrets dans des sous-groupes d'ordre premier divisant N.

Une conséquence de cette réduction est que si l'ordre du groupe ne contient pas de grand facteur premier (entier friable), alors le calcul du logarithme discret est facilité. Nous supposerons dorénavant dans tout problème de logarithme discret que l'ordre du groupe N est un nombre premier.

9.1.2 Méthodes en racine carrée

Pas de bébé, pas de géant

La méthode de Shanks que nous avons étudiée à la section 5.1.1 peut s'adapter presque directement au problème du log discret. En fait il s'avère qu'elle s'applique à de nombreuses situations où l'on cherche un entier dans un intervalle et que l'on dispose d'une certaine structure de groupe liée au problème ; on réduit alors le nombre d'opérations à la racine carrée de ce qui serait nécessaire par une recherche exhaustive. Dans notre contexte, celle-ci consisterait à essayer toutes les valeurs de λ les unes après les autres, jusqu'à trouver la bonne. Cela nécessite donc $O(N)$ opérations dans le groupe. Comme pour le calcul de l'ordre d'un élément, la stratégie repose sur l'écriture de λ en base $\sqrt{N}$: soient $\lambda_1 \in \left[0,\ \lfloor\sqrt{N}\rfloor - 1\right]$ et $\lambda_2 \in \left[0,\ \lceil\sqrt{N}\rceil\right]$ tels que

$$\lambda = \lambda_0 + \left\lfloor\sqrt{N}\right\rfloor \lambda_1.$$

L'égalité $\lambda \cdot D_1 = D_2$ se réécrit donc

$$\lambda_0 \cdot D_1 = D_2 - \lambda_1 \left\lfloor\sqrt{N}\right\rfloor \cdot D_1.$$

On peut alors précalculer tous les membres de gauche possibles (pas de bébé), les stocker dans une table, puis calculer tous les membres de droite (pas de géant) en cherchant à chaque fois si l'élément est dans la table. Nous ne redonnons pas ici l'algorithme détaillé. La principale difficulté est la gestion efficace de la structure de données nécessaire et comme dans pour le calcul de la cardinalité, il peut être nécessaire de modifier la valeur $\sqrt{N}$ comme base d'écriture de λ.

L'exécution de cet algorithme nécessite en moyenne $O(\sqrt{N})$ opérations dans le groupe. Plus précisément, en moyenne la solution est trouvée au milieu de la phase «pas de géant» et le nombre moyen d'opérations est $\frac{3}{2}\sqrt{N}$. Le problème majeur de cette méthode, comme pour le calcul du nombre de points est que la complexité en espace est la même que la complexité en temps, c'est-à-dire $O(\sqrt{N})$. De plus la parallélisation n'est pas faisable sans de nombreuses communications.

Méthode Rho et consorts

La méthode Rho, due à Pollard [Pol78], est un algorithme probabiliste qui permet de résoudre le problème du log discret avec une complexité heuristique moyenne qui est la même que pour la méthode de Shanks, mais qui ne nécessite quasiment aucun espace mémoire. Nous n'allons pas rappeler la version originale de l'algorithme, mais préférer les variantes qui présentent l'avantage d'être facilement parallélisable [GLV98, vOW99, WZ99, DGM99].

Le point clef de la méthode est l'introduction d'une marche pseudo-aléatoire dans le groupe. On commence par précalculer un nombre r de *décalages*: pour tout j compris entre 1 et r, soit

$$T_j = a_j D_1 + b_j D_2,$$

où a_j et b_j sont des entiers choisis aléatoirement dans l'intervalle $[0, N-1]$. On se donne ensuite une fonction de hachage $\mathcal{H}$ du groupe G vers l'intervalle $[1, r]$. On peut alors définir une fonction pseudo-aléatoire f de la manière suivante :

$$f(R) = R + T_{\mathcal{H}(R)}.$$

Et la marche pseudo-aléatoire annoncée est obtenue en itérant cette fonction. Le point initial

$$R_0 = \alpha_0 D_1 + \beta_0 D_2,$$

est obtenu en tirant les α_0 et β_0 aléatoirement dans l'intervalle $[0, N-1]$, puis pour tout i,

$$R_{i+1} = f(R_i) = R_i + T_{\mathcal{H}(R_i)}.$$

Une particularité importante de cette marche aléatoire est qu'elle permet de maintenir une expression en termes des éléments D_1 et D_2: si l'on sait que $R_i = \alpha_i D_1 + \beta_i D_2$, alors on peut calculer $\alpha_{i+1} = \alpha_i + a_{\mathcal{H}(R_i)} \bmod N$ et $\beta_{i+1} = \beta_i + b_{\mathcal{H}(R_i)} \bmod N$ de sorte que $R_{i+1} = \alpha_{i+1} D_1 + \beta_{i+1} D_2$.

Comme le groupe est fini, il existe deux indices i et j tels que $R_i = R_j$. Utilisant l'expression de ces éléments en termes de D_1 et D_2, on obtient

$$(\alpha_i - \alpha_j) D_1 = (\beta_j - \beta_i) D_2.$$

Si β_i et β_j sont deux entiers différents modulo N (N est supposé premier), alors le résultat est donné par

$$\lambda = \frac{\alpha_i - \alpha_j}{\beta_j - \beta_i} \bmod N.$$

L'algorithme est donc le suivant :

Algorithme 9.10 MÉTHODE RHO
Entrée: D_1 d'ordre N premier, $D_2 \in \langle D_1 \rangle$, un paramètre r, et une fonction de hachage $\mathcal{H}$.
Sortie: Le log discret de D_2 en base D_1.

1. Construire les décalages $T_j = a_j D_1 + b_j D_2$ pour $1 \leqslant j \leqslant r$;

2. Initier la marche aléatoire :
3. $\quad R \leftarrow \alpha D_1 + \beta D_2$;
4. $\quad S \leftarrow \{(R, \alpha, \beta)\}$;
5. Tant que le résultat n'est pas trouvé :
6. $\quad j \leftarrow \mathcal{H}(R)$;
7. $\quad R,\ \alpha,\ \beta\ \leftarrow R + T_{\mathcal{H}(R)},\ \alpha + a_{\mathcal{H}(R)},\ \beta + b_{\mathcal{H}(R)}$;
8. $\quad$ Si R est déjà dans S avec les coefficients α', β' alors
9. $\quad\quad$ Si $\beta - \beta'$ est inversible modulo N, retourner $\frac{\alpha-\alpha'}{\beta'-\beta} \bmod N$.
10. $\quad S \leftarrow S \cup \{(R, \alpha, \beta)\}$;

L'analyse en moyenne du temps de calcul de cet algorithme peut-être faite sous l'hypothèse que la fonction f est aléatoire, i.e. que la suite des éléments R que l'on calcule est aléatoire. Des travaux théoriques [SS85] et des expériences pratiques [Tes98] montrent qu'en prenant $r \geqslant 20$, cette hypothèse est raisonnable. Le nombre moyen d'itérations avant d'avoir une collision, obtienu par un calcul similaire à celui de la page 84 est

$$\sqrt{\frac{\pi N}{2}}.$$

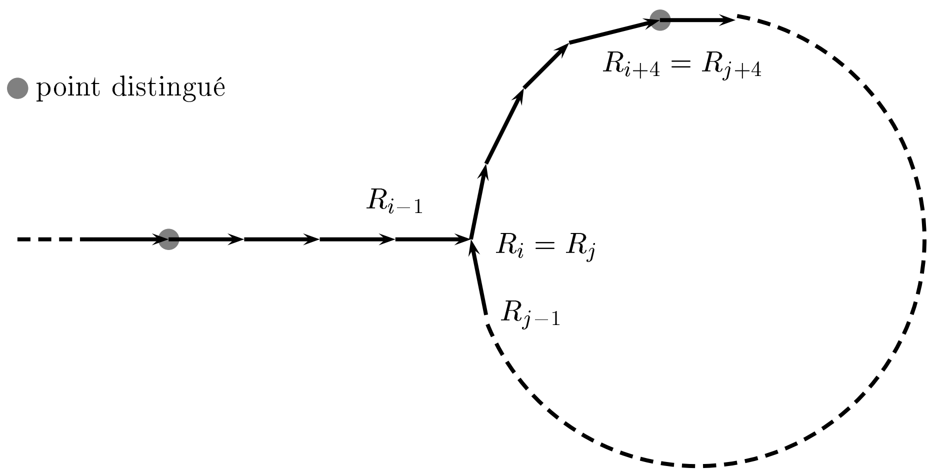
Cette collision produit le log discret avec une probabilité $1 - \frac{1}{N}$, c'est-à-dire presque à coup sûr pour des valeurs non triviales de N.

Toutefois, tel qu'il est présenté, l'algorithme nécessite toujours une mémoire aussi grande que pour la méthode de Shanks. Ce qui va changer les choses est l'utilisation des *points distingués*. On se fixe une propriété arbitraire sur les éléments du groupe, qui se produit avec une probabilité θ. En général cette propriété sera du type « les k premiers bits dans la représentation interne de l'élément sont à zéro », ce qui arrive avec probabilité 2^{-k}. Notons que cette notion de points distingués a été introduite pour la première fois par Quisquater et Delascaille [DQ90] pour la recherche de collision dans le système de cryptographie symétrique DES.

On modifie l'algorithme en ne faisant les étapes 8, 9 et 10 que si R est distingué. Le point clef est que si une collision intervient entre la i-ème et la j-ème valeur de R, celle-ci ne sera pas détectée tout de suite : il faudra attendre le premier point distingué après cette collision. Comme la fonction f est déterministe, pour tout n, le $(i+n)$-ième terme et le $(j+n)$-ième sont identiques, dès que le i-ème et le j-ème le sont.

Grâce à cette astuce des points distingués, le stockage est réduit d'un facteur θ^{-1}, au prix de θ^{-1} itérations supplémentaires. Pour un temps de calcul supplémentaire négligeable, on peut donc ramener l'espace mémoire nécessaire à une quantité négligeable.

Par ailleurs, cet algorithme se parallélise très bien. Les décalages T_j sont précalculés par un serveur central qui les envoie à tous les processeurs, ainsi que la propriété distinguante choisie. Chacun d'eux tire un point de départ aléatoire pour la marche aléatoire, effectue l'algorithme ci-dessus, mais se contente d'envoyer les points distingués au serveur quand il en trouve. Ainsi si l'on dispose de M processeurs, le temps de calcul total est divisé par M, avec une quantité d'information échangée négligeable.



9.1.3 Utilisation des automorphismes

Supposons que l'on dispose d'un automorphisme σ d'ordre m agissant sur le groupe engendré par D_1. Supposons de plus que le coût du calcul de l'image d'un point par σ est négligeable devant celui de la loi de groupe. On peut alors adapter la méthode Rho en créant une marche aléatoire sur les *orbites* sous l'action de σ plutôt que sur les éléments eux-mêmes. La taille de l'ensemble parcouru est divisée par m, ce qui fait que le nombre d'opérations est réduit d'un facteur $\sqrt{m}$.

Plus précisément : à partir de σ, on définit une relation d'équivalence $\simeq$ par

$$T \simeq U \iff \exists\, i,\ T = \sigma^i(U),$$

et on note

$$\overline{T} = \{\sigma^i(T),\ i \in \mathbb{N}\}$$

l'orbite de l'élément T. La fonction de hachage $\mathcal{H}$ utilisée pour calculer la fonction pseudo-aléatoire f doit être dorénavant une fonction qui ne dépend que de $\overline{T}$, et non plus de T. De même, la propriété distinguante doit être une propriété des orbites. Une façon de faire est de calculer les m éléments de l'orbite de T et de choisir comme représentant de $\overline{T}$ l'élément le plus petit pour une relation d'ordre arbitraire (en utilisant la représentation en machine, on peut convertir les éléments en des entiers).

Afin de garder un lien avec les éléments D_1 et D_2 lorsque l'on applique σ, on calcule l'entier η tel que

$$\forall T \in \langle D_1 \rangle,\ \sigma(T) = \eta \cdot T.$$

Un tel entier existe : $\sigma(D_1)$ appartient au sous-groupe engendré par D_1, donc il existe η tel que $\sigma(D_1) = \eta \cdot D_1$; et cet entier η vérifie la propriété demandée pour tout T, car σ est compatible avec la loi de groupe. Ainsi, si l'on a

$$R = \alpha D_1 + \beta D_2,$$

alors on garde l'expression en D_1 et D_2 lorsqu'on applique σ :

$$\sigma^i(R) = (\alpha\eta^i \bmod N)D_1 + (\beta\eta^i \bmod N)D_2.$$

Application aux courbes

Quand le groupe G est une Jacobienne de courbe hyperelliptique, il existe toujours un automorphisme d'ordre 2 très rapide à calculer : l'inversion dans le groupe, qui est donnée par l'involution hyperelliptique sur les points. Ainsi, un facteur $\sqrt{2}$ peut toujours être gagné par rapport à la méthode Rho classique, en itérant sur les orbites $\{R, -R\}$.

Une complication apparaît alors : de petits cycles peuvent se produire, ne donnant pas le log discret. Soit R_1 un diviseur dans la marche aléatoire qui est le représentant de son orbite $\{-R_1, R_1\}$. Supposons que la fonction f appliquée à R_1 revient à ajouter l'élément T, et qu'il faille prendre l'opposé pour tomber sur le représentant de $f(R_1) = R_1 + T$. Alors

$$R_2 = \overline{f(R_1)} = \overline{R_1 + T} = -R_1 - T.$$

Il peut alors se produire que la fonction f appliquée à R_2 nécessite d'ajouter de nouveau le même décalage T (cela se produit avec probabilité $\frac{1}{r}$). Alors

$$R_3 = \overline{f(R_2)} = \overline{-R_1 - T + T} = \overline{-R_1} = R_1,$$

et l'on tombe sur une collision qui manifestement ne donnera pas le log discret. Ce phénomène doit être pris en compte afin d'éviter de surcharger le serveur avec des collisions inutiles.

Le deuxième exemple classique d'automorphisme sur les courbes est l'automorphisme de Frobenius. Nous avons vu dans la partie 2 que calculer le cardinal des Jacobiennes n'est pas une tâche facile. Il est souvent proposé dans la littérature d'utiliser des courbes de Koblitz, i.e. des courbes définies sur $\mathbb{F}_p$ mais considérées sur une extension $\mathbb{F}_{p^n}$, pour lesquelles l'automorphisme de corps $x \mapsto x^p$ s'étend en un automorphisme d'ordre n sur la Jacobienne. C'est pourquoi ces courbes sont légèrement plus fragiles que les courbes générales : le calcul du log discret par la méthode Rho peut-être accéléré par un facteur $\sqrt{n}$.

Nous avons proposé [DGM99] d'autres exemples d'automorphismes provenant de la multiplication complexe. Il s'agit là encore de l'utilisation de propriétés spéciales de la courbe qui avaient facilité le calcul du cardinal de la Jacobienne. Les exemples de références sont les courbes elliptiques de la forme $y^2 = x^3 + ax$ et $y^2 = x^3 + b$ qui sont à multiplication complexe respectivement par $\mathbb{Z}[i]$ et par $\mathbb{Z}[\rho]$, où $i^2 = \rho^3 = 1$. En effet, pour ces courbes on dispose des automorphismes $(x, y) \mapsto (-x, iy)$ et $(x, y) \mapsto (\rho x, y)$, ce qui permet de gagner respectivement un facteur $\sqrt{2}$ et $\sqrt{3}$ par rapport à des courbes quelconques.

En genre supérieur, les courbes proposées dans la littérature présentent fréquemment des automorphismes. Le tableau suivant donne quelques exemples de telles courbes.

Auteur	Équation de la courbe	Corps	Automorphismes	m
Koblitz [Kob89],	$Y^2 + Y = X^5 + X^3$	$\mathbb{F}_{2^n}$	Frob + $\begin{cases} X \mapsto X + 1 \\ Y \mapsto Y + X^2 \end{cases}$	$4n$
[Kob90], [Kob97]	$Y^2 + Y = X^5 + X^3 + X$	$\mathbb{F}_{2^n}$	Frobenius	$2n$
	$Y^2 + Y = X^{2g+1} + X$	$\mathbb{F}_{2^n}$	Frobenius	$2n$
	$Y^2 + Y = X^{2g+1}$	$\mathbb{F}_{2^n}$	Frobenius	$2n$
Buhler–Koblitz [BK98] Chao et al. [CMST97]	$Y^2 + Y = X^{2g+1}$ (et ses twists)	$\mathbb{F}_p$ avec $p \equiv 1\,(2g+1)$	multiplication par ζ_{2g+1}	$2(2g+1)$
Sakai–Sakurai [SS98]	$Y^2 + Y = X^{13} + X^{11} + X^9 + X^5 + 1$	$\mathbb{F}_{2^{29}}$	Frob + $\begin{cases} X \mapsto X + 1 \\ Y \mapsto Y + X^6 + X^5 \\ \quad + X^4 + X^3 + X^2 \end{cases}$	4×29
Duursma–Sakurai [DS00]	$Y^2 = X^p - X + 1$	$\mathbb{F}_{p^n}$	Frob + $\begin{cases} X \mapsto X + 1 \\ Y \mapsto Y \end{cases}$	$2np$

9.2 Attaque de Frey-Rück

Les courbes elliptiques supersingulières sont plus faibles que les courbes elliptiques quelconques, à cause d'une attaque dite *réduction MOV*, du nom de ses auteurs Menezes, Okamoto, Vanstone [MOV93]. Cette attaque a été généralisée à des courbes de genre supérieur par Frey et Rück [FR94]. Nous présentons ici cette dernière méthode, qui contient en fait la précédente.

9.2.1 Couplage de Tate

Soit $\mathcal{C}$ une courbe définie sur un corps K et soit f une fonction sur $\mathcal{C}$. Il est possible d'évaluer f en un point de la courbe : on obtient un scalaire ou l'infini si la valuation de f en ce point est négative. Plus généralement, si D est un diviseur dont le support est disjoint de $\mathrm{div}(f)$, il est possible de définir $f(D)$. Pour cela, on écrit $D = \sum n_i P_i$, et on défini $f(D)$ en étendant linéairement sa valeur en les points :

$$f(\sum n_i P_i) = \prod f(P_i)^{n_i}.$$

L'hypothèse sur les supports disjoints assure que le résultat est un scalaire non nul et évite les problèmes d'indétermination.

Définition 9.2 *Soit $\mathcal{C}$ une courbe de genre g définie sur un corps K, soit N un entier premier à la caractéristique de K et soient D_1 et D_2 deux diviseurs de degré 0 de support disjoints. On suppose que la classe de D_1 dans la Jacobienne est d'ordre N.*

Soit f une fonction sur $\mathcal{C}$ telle que $\mathrm{div}(f) = N \cdot D_1$. *Alors le* couplage de Tate *de D_1 et D_2 est défini par*

$$\{D_1, D_2\}_N = f(D_2).$$

Dans [FR94], le théorème suivant est prouvé :

Théorème 9.1 *Soient $\overline{D_1}$ et $\overline{D_2}$ deux éléments de la Jacobienne de $\mathcal{C}$, avec $\overline{D_1}$ d'ordre N. Soient D_1 et D_2 des représentants des éléments $\overline{D_1}$ et $\overline{D_2}$ ayant des supports disjoints. On peut alors poser*

$$\{\overline{D_1}, \overline{D_2}\}_N = \{D_1, D_2\}_N.$$

Cela définit un scalaire unique à une puissance N-ième près dans K.

De plus ce couplage est une application bilinéaire, qui est non dégénérée si K contient les racines N-ièmes de l'unité.

Ainsi, étant donnés deux éléments de la Jacobienne, le premier étant de N-torsion, on peut fabriquer un scalaire modulo les puissances N-ièmes. Ce scalaire peut être calculé en pratique de manière efficace. Tout d'abord, trouver des représentants à supports disjoints n'est pas vraiment un problème, puisqu'on peut ajouter des diviseurs principaux autant que l'on veut. Ensuite, le point clef est le calcul d'une fonction f telle que $\mathrm{div}(f) = N \cdot D_1$.

On suppose que l'on sait calculer dans la Jacobienne : soit l'on dispose d'une version effective du théorème de Riemann-Roch pour la courbe considérée, soit il s'agit d'une courbe hyperelliptique et l'on utilise l'algorithme de Cantor. Partant de D_1 vu comme un élément de la Jacobienne, on peut calculer N fois ce diviseur et on obtient le diviseur nul (par hypothèse sur D_1). Cette multiplication par N se fait par des additions et doublements successifs en $O(\log N)$ étapes. À chaque fois que l'on réduit le diviseur en utilisant le théorème de Riemann-Roch, on ajoute en fait le diviseur d'une fonction. En gardant en mémoire toutes les fonctions dont on ajoute ou retranche le diviseur, on a ainsi la description de la fonction f cherchée.

Par exemple, dans le cas des courbes elliptiques, pour réduire un diviseur de poids 2 en un diviseur de poids 1, on fait passer une droite par les deux points, cette droite coupe la courbe en un troisième point et on prend le symétrique (c'est la fameuse loi «corde et tangente»). Les fonctions dont on ajoute les diviseurs sont dans ce cas données par les équations des droites que l'on doit tracer. Nous renvoyons à [Men93] pour des exemples sur les courbes elliptiques, le cas des courbes générales n'est pas plus compliqué pourvu que l'on sache calculer dans la Jacobienne.

Une fois que l'on connaît la fonction f, l'évaluation en D_2 est aisée. Toutefois, la fonction f a un degré de l'ordre de $O(N)$ pour son numérateur et son dénominateur, et la stocker sous forme développée n'est possible qu'au prix d'un espace et d'un temps de calcul de l'ordre de $O(N)$ au minimum. Pour éviter ce coût prohibitif deux stratégies sont envisageables :

- Il est possible de stocker f sous forme factorisée : on a alors $O(\log N)$ facteurs (éventuellement à de grandes puissances). Ce stockage facilite de plus l'évaluation et rend polynomial le temps de calcul du couplage de Tate.
- Au lieu de stocker la fonction f, on l'évalue au fur et à mesure en D_2 à chaque étape de la multiplication par N. Cette méthode est plus simple à implanter, et est plus rapide à l'exécution. Voir [FMR98] pour des exemples.

Finalement, on a le résultat suivant :

Proposition 9.1 *Soit $\mathcal{C}$ une courbe pour laquelle on sait calculer dans la Jacobienne. Alors le calcul du couplage de Tate d'ordre N de deux diviseurs nécessite $O(\log N)$ d'opérations dans le corps de base et dans la Jacobienne.*

9.2.2 Algorithme de réduction

Le couplage de Tate peut être utilisé pour transférer un problème de logarithme discret d'ordre N dans la Jacobienne d'une courbe vers un problème de logarithme discret dans la plus petite extension du corps de base contenant les racines N-ièmes de l'unité.

Algorithme 9.11 Réduction Frey–Rück
Entrée: Une courbe $\mathcal{C}$ définie sur $\mathbb{F}_q$, D_1 d'ordre N dans la Jacobienne de $\mathcal{C}$ et $D_2 \in \langle D_1 \rangle$.
Sortie: Le log discret de D_2 en base D_1.

1. Déterminer k minimal tel que N divise $q^k - 1$;
2. Choisir $E \in \mathrm{Jac}(\mathcal{C})/\mathbb{F}_{q^k}$ aléatoirement ;
3. Calculer les couplages de Tate $d_1 = \{D_1, E\}_N$ et $d_2 = \{D_2, E\}_N$;
4. Si d_1 est une puissance N-ième, retourner en 2 ;
5. Résoudre le problème de log discret $d_2 = d_1^\lambda$, avec λ défini modulo N ;
6. Retourner λ.

Cet algorithme permet donc de tirer parti de l'existence d'algorithmes sous-exponentiels pour le log discret dans les corps finis. C'est grâce à cette méthode que Frey et Rück ont pu casser certains cryptosystèmes proposés dans le premier article de Koblitz sur la crypographie hyperelliptique. Ces courbes, définies sur des extensions du corps $\mathbb{F}_2$ avaient pour équation $y^2 + y = x^5 + x^3$, $y^2 + y = x^5 + x^3 + x$, et $y^2 + y = x^5$. Notons que ces courbes sont en fait supersingulières, elles sont donc isogènes (éventuellement sur une extension) à un produit de courbes elliptiques supersingulières (voir page 22).

Le problème du logarithme discret peut aussi être attaqué par cette voie : on transfère le problème en un problème sur une courbe elliptique, puis par la réduction MOV, on peut se ramener à un corps fini. Ces deux approches sont équivalentes au sens où l'extension de $\mathbb{F}_q$ dans laquelle on est obligé de travailler est la même.

9.2.3 Exemple déroulé en genre 2

Soit $\mathcal{C}$ la courbe sur $\mathbb{F}_2$ d'équation

$$y^2 + y = x^5 + x^3.$$

Le cardinal de sa Jacobienne est 13 et on cherche le logarithme discret de

$$D_1 = \langle x^2 + x, 1\rangle \quad \text{et} \quad D_2 = \langle x^2 + 1, 0\rangle.$$

Afin d'expliciter les calculs intermédiaires, nous allons donner les fonctions qui entrent en jeu, même si elles ne sont pas stockées en pratique, mais évaluées à la volée.

Commençons par rechercher une fonction f_1 telle que

$$\mathrm{div}(f_1) = 13 \cdot D_1.$$

Pour cela on applique la méthode binaire de manière à multiplier par 13 le diviseur D_1 suivant l'algorithme de Cantor. Cependant on garde à chaque étape de réduction la fonction qui entre en jeu. On a tout d'abord

$$2 \cdot D_1 = \langle x, 0\rangle + \mathrm{div}(\varphi_1),$$

où $\varphi_1 = \frac{y+1}{x}$. Ensuite

$$4 \cdot D_1 = \langle x^2, 0\rangle + \mathrm{div}(\varphi_1^2).$$

Et un dernier doublement donne

$$8 \cdot D_1 = \langle x^2 + x, x + 1\rangle + \mathrm{div}(\varphi_1^4 \varphi_2),$$

où $\varphi_2 = \frac{y+x^3}{x(x+1)}$. On écrit ensuite $13 = 1 + 4 + 8$, et on obtient finalement

$$\begin{aligned}
13 \cdot D_1 &= \langle x^2 + x, 1\rangle + \langle x^2, 0\rangle + \langle x^2 + x, x + 1\rangle + \mathrm{div}(\varphi_1^6 \varphi_2) \\
&= \langle x^2 + x, x\rangle + \langle x^2 + x, x + 1\rangle + \mathrm{div}(x) + \mathrm{div}(\varphi_1^6 \varphi_2) \\
&= \mathrm{div}(x(x+1)) + \mathrm{div}(x) + \mathrm{div}(\varphi_1^6 \varphi_2) \\
&= \mathrm{div}(x^2(x+1)\varphi_1^6 \varphi_2).
\end{aligned}$$

La fonction f_1 cherchée est donc

$$f_1 = \frac{(y + x^3)(y + 1)^6}{x^5}.$$

De manière similaire pour D_2, la méthode binaire donne

$$2 \cdot D_2 = \langle x^2 + x, 1\rangle + \mathrm{div}(\varphi_3),$$

où $\varphi_3 = \frac{y+x^3+x}{x^2+x}$. Ensuite

$$4 \cdot D_2 = \langle x, 0 \rangle + \mathrm{div}(\varphi_3^2 \varphi_1).$$

Puis

$$8 \cdot D_2 = \langle x^2, 0 \rangle + \mathrm{div}(\varphi_3^4 \varphi_1^2).$$

Et pour finir

$$\begin{aligned} 13 \cdot D_2 &= \langle x^2+1, 0 \rangle + \langle x, 0 \rangle + \langle x^2, 0 \rangle + \mathrm{div}(\varphi_1^3 \varphi_3^6) \\ &= \langle x^2, 1 \rangle + \langle x^2, 0 \rangle + \mathrm{div}(\frac{y}{x^2}) + \mathrm{div}(\varphi_1^3 \varphi_3^6) \\ &= \mathrm{div}(x^2) + \mathrm{div}(\frac{y}{x^2}) + \mathrm{div}(\varphi_1^3 \varphi_3^6) \\ &= \mathrm{div}(y \varphi_1^3 \varphi_3^6) \end{aligned}$$

La fonction f_2 cherchée est donc

$$f_2 = \frac{y(y+1)^3(y+x^3+x)^6}{x^9(x+1)^6}.$$

Pour appliquer l'algorithme de Frey–Rück, il faut maintenant choisir un diviseur aléatoire sur une extension de $\mathbb{F}_2$ contenant les racines 13-ième de l'unité. Dans notre cas, il faut donc se placer sur

$$\mathbb{F}_{2^{12}} = \mathbb{F}_2[t]/(t^{12} + t^3 + 1).$$

On considère le diviseur $E = P_1 - P_2$, où

$P_1 = (t^{10}+t^9+t^8+t^7+t^4+t^2+t,\ t^8+t^7+t^6+t^5+t^4+t^2)$ et $P_2 = (t^{10}+t^8+t^2,\ t^7+t^6+t^5+t^4+t^2+1)$.

On calcule alors facilement

$$d_1 = f_1(E) = \frac{f_1(P_1)}{f_1(P_2)} = t^{11} + t^{10} + t^3 + t^2 + 1,$$

et

$$d_2 = f_2(E) = \frac{f_2(P_1)}{f_2(P_2)} = t^{10} + t^8 + t^5 + t^2 + t.$$

Et on vérifie que l'on a d'une part

$$\frac{\log(d_2)}{\log(d_1)} \bmod 13 = 7,$$

et d'autre part

$$D_2 = [7]D_1.$$

On a donc bien ramené un problème de log discret hyperelliptique à un log discret classique dans un corps fini.

Remarque. Dans l'exemple ci-dessus, les fonctions que l'on calcule sont normalisées en le point à l'infini. Si l'on procède ainsi, il n'est plus utile de choisir un diviseur E de degré 0. Cette condition était nécessaire pour lever l'ambiguïté due au fait que les fonctions ne sont a priori définies qu'à une constante multiplicative près. Ainsi, calculer avec des fonctions normalisées permet de choisir E égal à un diviseur formé d'un seul point aléatoire, ce qui économise la moitié des calculs.

9.3 Attaque de Rück

L'attaque par couplage de Tate (de même que la réduction MOV) ne fonctionne que lorsque l'ordre N de D_1 est premier à la caractéristique du corps. Dans le cas contraire, pour les courbes elliptiques, une attaque a été découverte indépendamment par Smart [Sma99], Semaev [Sem98] et Satoh–Araki [SA98]. Comme pour la section précédente, l'attaque pour les courbes plus générales contient celle pour les courbes elliptiques.

Le théorème de Rück [Rüc99] est le suivant :

Théorème 9.2 *Soit $\mathcal{C}$ une courbe de genre g définie sur un corps fini de caractéristique p. Le problème du logarithme discret dans un sous-groupe d'ordre p de* $\mathrm{Jac}(\mathcal{C})$ *peut être résolu au prix de $O(\log p)$ opérations dans le corps de base et dans la Jacobienne.*

Le principe est le suivant : soit D un diviseur de degré 0 dont la classe dans la Jacobienne est d'ordre p, alors il existe une fonction f telle que $\mathrm{div}(f) = p \cdot D$. La différentielle holomorphe df/f peut être exprimée à l'aide d'un paramètre local t en le point P_∞ qui a servi à injecter la courbe dans sa Jacobienne : $df/f = \frac{\partial f/\partial t}{f} dt$. On peut alors définir la série

$$\frac{\partial f/\partial t}{f} = \sum_{i=0}^{\infty} a_i t^i,$$

où les a_i sont des scalaires. Le point crucial est que la fonction qui à la classe de D associe les $2g-1$ premiers coefficients $(a_0, \ldots, a_{2g-2})$ est un homomorphisme ϕ de groupes additifs. Le problème du logarithme discret ainsi transféré est facilement soluble.

Le calcul de cet homomorphisme ϕ peut être effectué efficacement par une technique similaire à celle employée pour le couplage de Tate.

9.4 Méthodes sous-exponentielles en genre « grand »

En 1994, Adleman, DeMarrais et Huang [ADH94] ont proposé la première méthode sous-exponentielle pour attaquer le log discret pour une grande classe de courbes. Leur algorithme, non prouvé dans sa forme originale, repose sur un calcul d'index, et donc sur une notion de friabilité dans les Jacobiennes de courbes. Malheureusement (ou heureusement, c'est selon !) cette notion de friabilité ne prend du sens que lorsque le genre de la courbe est suffisamment grand, et n'est donc absolument pas utilisable pour les courbes elliptiques.

9.4.1 Fonction sous-exponentielle, règles de calcul

On va définir une fonction, dite *sous-exponentielle* qui est une fonction plus petite qu'une fonction exponentielle, mais pas encore polynomiale.

Définition 9.3 *Les fonctions de la forme*

$$L_N(a, c) = \exp\left(c\,(\log N)^a (\log\log N)^{1-a}\right),$$

où c est un réel positif, et a est un réel dans l'intervalle $[0, 1]$ *sont appelées sous-exponentielles.*

Si $a = 0$, alors $L_N(0, c) = (\log N)^c$ est polynomial en la taille de N ; si $a = 1$, alors $L_N(1, c) = N^c$ est exponentiel en la taille de N. Les autres valeurs de a mesurent en quelque sorte si l'on est plus proche d'une fonction polynomiale ou exponentielle. Tous les algorithmes sous-exponentiels connus le sont pour $a = \frac{1}{2}$ ou $a = \frac{1}{3}$. Souvent on omet d'écrire ce paramètre a, après avoir précisé s'il s'agissait de l'une ou l'autre de ces valeurs.

Lemme 9.1 *Lors du calcul avec des complexités sous-exponentielles, on a les règles suivantes : pour $a > 0$ et c, c_1, c_2 des constantes positives,*

$$L_N(a, c_1) \cdot L_N(a, c_2) = L_N(a, c_1 + c_2),$$

$$L_N(a, c_1) + L_N(a, c_2) \in \Theta(L_N(a, \max(c_1, c_2))),$$
$$(\log N)^k \cdot L_N(a, c) \in L_N(a, c + o(1)).$$

De plus, si $b < a$, alors

$$L_N(a, c_1) \cdot L_N(b, c_2) \in L_N(a, c_1 + o(1)).$$

9.4.2 Définition de la notion de friabilité à la ADH

La définition d'un diviseur friable s'applique à un diviseur semi-réduit. Dans sa forme originale, elle ne concerne que les courbes hyperelliptiques.

Définition 9.4 *Soit $\mathcal{C}$ une courbe hyperelliptique de genre g définie sur un corps fini $\mathbb{F}_q$. Soit S un entier supérieur ou égal à 1. Soit $D = \langle u(x), v(x) \rangle$ un diviseur semi-réduit en représentation de Mumford. Le diviseur D est dit S-friable si le polynôme $u(x)$ n'a que des facteurs irréductibles de degré au plus S.*

Une définition équivalente permet d'étendre cette notion à n'importe quelle courbe. Cela permet aussi de mieux comprendre sa signification.

Proposition 9.2 *Un diviseur D est S-friable si et seulement si les points de la courbe qui le constituent sont définis sur une extension de degré au plus S du corps de base.*

On voit tout de suite que cette définition n'est d'aucune utilité en genre 1 : un élément de la Jacobienne est en fait un point de la courbe, donc tous les éléments sont 1-friables et a fortiori S-friables pour tout $S > 1$. Plus le genre est grand, plus on a de marge de manœuvre pour choisir S de manière à avoir la proportion souhaitée d'éléments friables ; c'est pourquoi la complexité sous-exponentielle ne sera accessible que pour les courbes de genre suffisamment grand.

9.4.3 Principe général de l'algorithme

L'algorithme de log discret est un classique calcul d'index. Il comporte quatre phases :

1. Construire la *base de facteurs*, c'est-à-dire l'ensemble des éléments $\{p_1, \ldots, p_n\}$ de la Jacobienne S-friables ;

2. Trouver « suffisamment » de relations entre les éléments de la base :

$$\sum e_i p_i = 0.$$

3. Par de l'algèbre $\mathbb{Z}$-linéaire, déterminer une base $\{b_1, \dots b_r\}$ de la Jacobienne en tant que $\mathbb{Z}$-module et exprimer les p_i sur cette base ;

4. Exprimer D_1 et D_2 en fonction des p_i, puis sur la base des b_i et en déduire le log discret.

Dans ces quatre phases, les trois premières constituent un calcul que l'on doit effectuer une fois pour chaque courbe. Ensuite pour chaque problème de log discret dans la Jacobienne de cette courbe, on n'a que la phase 4 à faire.

Le coût de la phase 1 est en général négligeable. Les phases 2 et 3 sont les deux phases critiques. Trouver des relations est d'autant plus facile que les éléments friables sont nombreux, donc la borne de friabilité S doit être la plus grande possible ; par ailleurs l'algèbre linéaire sera plus simple si la base de facteurs est petite, donc la borne S doit être la plus petite possible pour rendre faisable cette étape. Tout l'art des calculs d'index est de réduire au maximum les complexités de ces deux phases par des techniques diverses, puis de choisir la meilleure borne S possible de manière à minimiser le coût total de l'algorithme.

En général, le coût de la phase 4 est du même ordre que le coût requis pour trouver une relation dans la phase 2.

Dans ce type d'algorithme, le choix optimal de S est *sous-exponentiel* en la taille de l'entrée : il existe a (en général $a = \frac{1}{2}$) tel qu'en choisissant une borne de friabilité $S = L_N(a, c)$, on peut obtenir une complexité $L_N(a, c')$ pour la phase de recherche de relations, puis $L_N(a, c'')$ pour l'algèbre linéaire, où c' et c'' dépendent de c. Il reste alors à choisir c pour minimiser c' et c''.

9.4.4 Variantes proposées dans la littérature

Les différentes variantes proposées dans la littérature portent sur le moyen de trouver les relations. Pour l'algèbre linéaire, il s'agit toujours d'un calcul de *Forme Normale de Smith* (abrégé en SNF, d'après les initiales anglaises) ; nous renvoyons à [Coh93] pour tout ce qui concerne ce type de calcul qui est une sorte de pivot de Gauß dans lequel les inversions sont remplacées par des pgcd.

Algorithme original ADH

L'algorithme ne fonctionne qu'en caractéristique impaire. Soit $\mathcal{C}$ hyperelliptique d'équation $y^2 = f(x)$.

Le moyen proposé par ADH pour trouver des relations est assez compliqué : on fabrique une fonction sur la courbe, on calcule ensuite son diviseur en espérant qu'il soit semi-réduit et S-friable. Voici comment cela est fait : soient $A(x)$ et $B(x)$ deux polynômes aléatoires. La fonction $\varphi(x, y) = A(x) + yB(x)$ sur $\mathcal{C}$ aura un diviseur S-friable si et seulement si sa *norme* est S-friable, cette norme étant définie par

$$\mathrm{N}(\varphi) = A(x)^2 + f(x)B(x)^2.$$

Heuristiquement, cette norme est friable avec la même probabilité que pour un polynôme aléatoire du même degré. Une fois une fonction de norme friable détectée, on en déduit un relation entre des éléments de la base de facteurs, et donc entre leurs logarithmes discrets.

Cette stratégie nécessite de trouver les paramètres adéquats pour la borne de friabilité et les degrés de $A(x)$ et $B(x)$. Avec quelques heuristiques, on en déduit une complexité (non prouvée)

$$L_{q^g}\left(\frac{1}{2}, c\right),$$

pour un genre g supérieur à $\log q$.

À notre connaissance, cet algorithme n'a pas été implanté tel quel.

Variante de Flassenberg et Paulus

Reprenant l'approche d'ADH, Flassenberg et Paulus [FP99] ont amélioré la recherche de relations en utilisant un crible. La méthode est toujours de rechercher des fonctions $\varphi(x, y)$ ayant un diviseur friable, mais cette fois-ci une méthode de crible permet de trouver plus rapidement parmi un ensemble de fonctions lesquelles conviennent. Cela permet donc de baisser la borne de friabilité et de réduire le coût global de l'algorithme.

Les auteurs ont implanté leur algorithme et obtenu des temps de calcul. Les plus grands exemples traités sont de genre 12 pour un corps à 11 éléments, et de genre 6 sur un corps à 101 éléments. Le résultat est alors obtenu en quelques heures. D'après ces expériences, il n'est pas aisé de voir à partir de quand cette méthode devient plus rapide qu'un algorithme générique en racine carrée.

Travaux de Müller–Stein–Thiel et de Enge

Les algorithmes précédents restant heuristiques, Müller, Stein et Thiel [MST99] d'une part et Enge [Eng99] d'autre part se sont attachés à en fournir une version prouvée. La méthode employée pour trouver des relations est inspirée de l'algorithme de Haffner-McCurley [HM89] : on construit des éléments aléatoires dans le groupe en espérant qu'ils soient friables. Les clefs pour prouver rigoureusement la complexité sont une analyse précise de la proportion d'éléments friables dans la Jacobienne d'une courbe hyperelliptique [ES00], une randomisation suffisante de la construction d'éléments aléatoires, et une version prouvée du calcul de SNF. La complexité obtenue est

$$L_{q^g}\left(\frac{1}{2}, 2.04\right).$$

Ces travaux sont les premiers prouvant réellement qu'il existe un algorithme sous-exponentiel pour le log discret quand le genre devient grand ; d'autre part, dans [Eng99], la complexité fait intervenir le rapport entre le genre et la taille du corps, donnant une première idée de ce que signifie « genre grand » en pratique. Toutefois, cet algorithme n'a pas été implanté.

Approche de Galbraith–Paulus–Smart

Dans leur article sur les courbes superelliptiques [GPS00], Galbraith, Paulus et Smart ont décrit une technique reposant sur l'algorithme de Haffner-McCurley permettant de résoudre le problème du logarithme discret pour une courbe superelliptique, c'est-à-dire une courbe d'équation $y^n = f(x)$ n'ayant qu'un seul point à l'infini. La complexité sous-exponentielle n'est qu'heuristique, car pour ces courbes, on ne dispose pas de théorème sur la densité des éléments friables dans la Jacobienne. Notons que leur approche fonctionne en fait sur n'importe quelle courbe pour laquelle on sait calculer efficacement dans la Jacobienne et qui ne possède qu'un seul point à l'infini.

Chapitre 10

Algorithme sous-exponentiel générique

Ce chapitre est consacré à la description d'un algorithme générique pour résoudre le logarithme discret dans le contexte suivant :

- le groupe est cyclique, d'ordre connu,
- une notion de friabilité est disponible dans le groupe.

L'exemple de référence pour un tel groupe est le groupe multiplicatif d'un corps fini. La plupart des algorithmes à la Haffner-McCurley permettant de résoudre le logarithme discret dans d'autres groupes ne supposent pas le premier point et requièrent donc une première phase durant laquelle la structure du groupe est déterminée. Nous allons montrer que cette hypothèse supplémentaire (principalement la connaissance de l'ordre du groupe) permet de décrire un calcul d'index dont le temps de calcul est prouvable et améliore les meilleures bornes connues pour les groupes de classes comme énoncé dans le théorème ci-dessous. Ces résultats ont été obtenus conjointement avec A. Enge [EG00].

Théorème 10.1 *Il existe un algorithme probabiliste résolvant un problème de logarithme discret dans la Jacobienne d'une courbe hyperelliptique de genre g sur un corps fini $\mathbb{F}_q$ quand le groupe est cyclique d'ordre connu et $g/\log q$ tend vers l'infini. Le temps de calcul moyen est*

$$L_{q^g}\left(\frac{1}{2}, \sqrt{2}+o(1)\right).$$

Sous l'hypothèse de Riemann généralisée, il existe un algorithme probabiliste résolvant un problème de logarithme discret dans le groupe de classes d'un corps quadratique imaginaire de discriminant $-D$ quand le groupe est cyclique d'ordre connu N. Le temps de calcul moyen est

$$L_N\left(\frac{1}{2}, \sqrt{2}+o(1)\right) = L_D\left(\frac{1}{2}, 1+o(1)\right).$$

10.1 Modèle générique de friabilité

L'algorithme que nous allons décrire est *générique* au sens où il s'applique à tous les groupes connus où une notion de friabilité est disponible. Nous allons donner un modèle abstrait de friabilité, inspiré par les *formations arithmétiques* de Knopfmacher [Kno75].

10.1.1 Hypothèses générales

Soit G un groupe cyclique d'ordre connu N. En général, on s'attend à ce que les éléments de G soient représentés par des suites de $O(\log N)$ bits et l'on peut mesurer les complexités comme des fonctions de N. Cependant, il est possible d'avoir des groupes pour lesquels les éléments sont représentés par des suites de bits de taille $O(\log N')$ pour une valeur de N' plus grande que N. Par exemple, les éléments des Jacobiennes de courbes de genre g sur $\mathbb{F}_2$ sont donnés par $\Theta(\log N')$ bits où $N' = 2^g$, alors que la borne inférieure sur la taille du groupe donnée par la borne de Hasse-Weil n'est que $(\sqrt{2}-1)^g \leqslant 1$. C'est pourquoi nous étudierons les complexités en fonction de N' qui est plus lié à la taille des entrées que l'ordre du groupe N. Notons qu'en pratique, seuls les cas où N et N' sont du même ordre de grandeur sont intéressants.

Par ailleurs, les algorithmes considérés seront sous-exponentiels en $\log N'$, donc chaque étape nécessitant un temps de calcul polynomial en $\log N'$ peut être comptée comme une unité, sachant que cela rajoute seulement $o(1)$ dans la constante. On introduit donc la notation de [vzGG99] : pour une fonction positive f de N', $O^{\sim}(f)$ est l'ensemble des fonctions qui sont dans $O(f)$ à un facteur près, borné par une puissance de $\log N'$.

L'idée de friabilité est que les éléments du groupe G doivent se comporter essentiellement comme des entiers ou des polynômes, c'est-à-dire admettre une décomposition unique en « somme d'éléments premiers ». Pour modéliser ce phénomène, nous supposons qu'il existe un monoïde abélien libre $\mathbb{M}$ sur un ensemble dénombrable $\mathcal{F}$, dont les éléments sont appelés *premiers*, ainsi qu'une relation d'équivalence $\sim$ sur $\mathbb{M}$ compatible avec la loi de composition, tout ceci tel que

$$G \;\simeq\; (\mathbb{M}/\sim) \;\simeq\; \left(\mathbb{N}^{\mathcal{F}}/\sim\right).$$

Ainsi chaque élément de $\mathbb{M}$ a une décomposition unique en somme de premiers. On suppose que pour chaque élément de G on dispose d'un représentant canonique dans $\mathbb{M}$, de sorte que G hérite de la propriété de décomposition unique. On suppose de plus que ces représentants sont de taille $O^{\sim}(1)$ bits et que l'arithmétique dans G (addition, négation et test d'égalité) peut se faire en manipulant ces représentants en temps polynomial en $\log N'$.

On suppose de plus qu'il existe un homomorphisme de monoïdes $\deg : \mathbb{M} \to \mathbb{R}_+$, qui associe à chaque élément de $\mathbb{M}$ (et donc de G) sa *taille*. Comme $\mathbb{M}$ est libre sur $\mathcal{F}$, un tel homomorphisme est donné par l'assignation d'un réel positif à chaque élément premier et on étend la définition additivement. Intuitivement, la taille $\deg(m)$ et la longueur de la suite de bits représentant m en machine sont essentiellement les mêmes, à une constante multiplicative près. On impose de plus $\deg(p) \geqslant 1$ pour $p \in \mathcal{F}$ et $\deg(g) \in O^{\sim}(1)$ pour tout $g \in G$; ceci assure que le nombre d'éléments premiers dans la décomposition d'un élément de G est en $O^{\sim}(1)$.

Pour une *borne de friabilité* $S \in \mathbb{N}$ on note $\mathcal{F}_S$ l'ensemble des premiers de taille au plus S, de cardinal noté n_S et $\mathbb{M}_S$ l'ensemble des éléments de $\mathbb{M}$ de taille au plus S, de cardinal noté n'_S. L'ensemble $\mathcal{F}_S$ est appelé *base de facteurs*.

Un élément de G est dit S-friable si sa décomposition ne met en jeu que des premiers dans $\mathcal{F}_S$. Comme la taille des éléments de G est en $O^{\sim}(1)$, la distinction entre friables et non-friables n'a d'intérêt que pour $S \in O^{\sim}(1)$.

Table récapitulative des notations

Ce chapitre est assez technique, et nous donnons donc une table récapitulative des notations, spécifique à celui-ci.

notation	signification
$\mathcal{F}$	ensemble dénombrable : les *premiers*
$\mathbb{M}$	monoïde libre sur $\mathcal{F}$
$\sim$	relation d'équivalence sur $\mathbb{M}$
$G = \mathbb{M}/\sim$	groupe dans lequel on résout le problème
N	ordre du groupe, supposé connu
N'	entier lié à la taille de l'entrée
deg	homomorphisme *taille*, de G vers $\mathbb{R}_+$
S	borne de friabilité
$\mathcal{F}_S$	premiers de taille au plus S
n_S	cardinal de $\mathcal{F}_S$
$\mathbb{M}_S$	éléments de $\mathbb{M}$ de taille au plus S
n'_S	cardinal de $\mathbb{M}_S$

10.1.2 Hypothèses algorithmiques

D'un point de vue algorithmique, on suppose que n'_S est fini, que les éléments de taille au plus S peuvent être énumérés en temps polynomial en S et linéaire en n'_S, et que la primalité d'un élément $m \in \mathbb{M}$ peut être testée en temps polynomial en $\deg(m)$ et linéaire en $n'_{\deg(m)}$ (la méthode naïve « trial division » peut être employée). Ainsi $\mathcal{F}_S$ peut être construit en temps polynomial en S et quadratique en n'_S. En pratique, un crible d'Eratosthène diminue la complexité en n'_S ; cependant les algorithmes étudiés seront au mieux quadratiques en n'_S.

Pour des raisons techniques, nous supposons que $\log n_S \in O^{\sim}(1)$.

Le test de S-friabilité d'un élément de G, et le cas échéant la décomposition en éléments de $\mathcal{F}_S$ peuvent être faits en temps $O^{\sim}(n_S)$ par « trial division » par les éléments de $\mathcal{F}_S$. Dans tous les cas considérés en pratique, on peut faire mieux, ce qui améliore les complexités.

En ce qui concerne les entiers, le meilleur test de friabilité connu, qui est sous-exponentiel en $\log n'_S$, est non-déterministe et peut faillir au sens où il peut ne pas reconnaître un entier friable. C'est pourquoi nous étendons quelque peu notre modèle en demandant que le test de friabilité rejette tous les éléments non friables et reconnaisse un élément friable avec une probabilité d'erreur qui ne dépasse pas $1/2$.

10.1.3 Exemples

Corps finis premiers $G = \mathbb{F}_p^*$

Ici G peut être représenté comme $(\mathbb{N}^*, \cdot)/\sim$, où $m_1 \sim m_2$ si et seulement si $p | m_1 - m_2$, et $\mathcal{F}$ est l'ensemble des nombres premiers naturels. La taille d'un élément est donnée par son logarithme en base 2, $\deg(m) = \log_2 m$, et $N' = N = p - 1$.

Corps finis de caractéristique 2, $G = \mathbb{F}_{2^k}^*$

Ici G peut être représenté comme $(\mathbb{F}_2[X] \backslash \{0\}, \cdot)/\sim$, où $f_1 \sim f_2$ si et seulement si $f | f_1 - f_2$ pour un polynôme irréductible f fixé de degré k dans $\mathbb{F}_2[X]$, et $\mathcal{F}$ est l'ensemble des polynômes irréductibles sur $\mathbb{F}_2$. La taille d'un élément est son degré au sens usuel et $N' = N = 2^k - 1$.

Corps finis de la forme $G = \mathbb{F}_{p^k}^*$, p premier

Ici G peut être représenté par les polynômes de degré inférieur à k sur $\mathbb{F}_p$. Soit $\mathbb{F}_p[X]'$ l'ensemble des polynômes unitaires sur $\mathbb{F}_p$. Remarquant que tout polynôme est le produit de son coefficient dominant et d'un polynôme unitaire, G peut être représenté comme $(\mathbb{N}^*, \cdot) \times (\mathbb{F}_p[X]', \cdot)/\sim$, où $(m_1, f_1) \sim (m_2, f_2)$ si et seulement si $p | m_1 - m_2$ et $f | f_1 - f_2$ pour un polynôme irréductible fixé de degré k sur $\mathbb{F}_p$. L'ensemble des premiers $\mathcal{F}$ est donné par la réunion des nombres premiers naturels et de l'ensemble des polynômes unitaires irréductibles sur $\mathbb{F}_p$. La taille d'un élément est $\deg(m_1, f_1) = \log_2 m_1 + \deg f_1$, et $N' = N = p^k - 1$. Notons que ces définitions sont compatibles avec les exemples précédents.

Groupes de classes de corps de nombres

Soit K un corps de nombres et $\mathcal{O}$ son anneau des entiers. Alors le groupe de classes G de K est défini par $\mathbb{M}/\sim$, où $\mathbb{M}$ est l'ensemble des idéaux de $\mathcal{O}$ (c'est le monoïde libre sur l'ensemble $\mathcal{F}$ des idéaux premiers), et $\sim$ est induite par le sous-monoïde des idéaux principaux. La taille d'un idéal est donnée par le logarithme de sa norme. Si K a un groupe d'unités de rang 0, alors chaque classe d'idéaux contient un unique idéal *réduit* qui peut être calculé en temps polynomial à partir d'un représentant quelconque de la classe pourvu que $(K : \mathbb{Q})$ soit fixé. Cet idéal réduit constitue le représentant canonique de la classe.

Le cas particulier des corps quadratiques imaginaires $\mathbb{Q}(\sqrt{D})$ de discriminant $D < 0$ remplit ces conditions. Soit $\omega = \frac{D+\sqrt{D}}{2}$, de polynôme minimal $X^2 - DX + \frac{D^2-D}{4}$, un générateur d'une base d'entiers. Alors $\mathcal{F}_S$ peut être construit en énumérant tous les nombres premiers $p \leqslant 2^S$ et en résolvant l'équation $y_p^2 - Dy_p + \frac{D^2-D}{4} \equiv 0 \pmod{p}$, ce qui se fait en temps polynomial par un algorithme probabiliste. Si cette équation n'a pas de solution, alors p est inerte et $p\mathcal{O}$ est un idéal premier principal, il peut donc être omis de la base de facteurs. Si l'équation a une solution double $y_p = 0$, alors p est ramifié et (p, ω) est le seul idéal premier au dessus de p dans $\mathcal{O}$. Finalement, si l'équation a deux solutions y_p et $\overline{y}_p$, alors p est décomposé et $\mathfrak{p} = (p, \omega - y_p)$ et $\overline{\mathfrak{p}} = (p, \omega - \overline{y}_p)$ sont les deux idéaux premiers au dessus de p dans $\mathcal{O}$.

Un idéal réduit $\mathfrak{a} = (a, \omega - b)$ avec $a | b^2 - Db + \frac{D^2-D}{4}$ est S-friable si et seulement si tous les diviseurs premiers de a sont majorés par 2^S. Soit p un diviseur premier de a et ν l'exposant de p dans a. Alors, comme l'idéal est réduit, on peut montrer que p n'est pas inerte, donc il existe un idéal de la forme $\mathfrak{p} = (p, \omega - y_p)$ au dessus de p dans $\mathcal{O}$. Si $y_p \equiv b \pmod{p}$, alors l'idéal intervient avec multiplicité ν dans la décomposition de $\mathfrak{a}$, sinon, $\overline{\mathfrak{p}}$ intervient avec multiplicité ν. Ainsi le test de friabilité et la décomposition d'un idéal en idéaux premiers se réduisent aux mêmes problèmes sur les entiers rationnels. Là encore, $N' = N$.

Jacobiennes de courbes sur les corps finis

Soit $\mathcal{C}$ une courbe plane irréductible sur $\mathbb{F}_q$ et G sa Jacobienne. Une fois que l'on s'est fixé un point sur la courbe, le monoïde des diviseurs effectifs s'envoie dans la Jacobienne et la relation d'équivalence $\sim$ est induite par les diviseurs principaux, comme décrit au chapitre 1. L'ensemble $\mathcal{F}$ est l'ensemble des diviseurs premiers et la taille est donnée par le degré usuel.

Le cas des courbes hyperelliptiques est l'analogue des corps de nombres quadratiques. Soit $\mathcal{C}$ une courbe hyperelliptique d'équation $y^2 + h(x)y + f(x) = 0$ avec $h \in \mathbb{F}_q[x]$ de degré au plus g et $f \in \mathbb{F}_q[x]$ unitaire de degré $2g+1$. Alors la représentation de Mumford permet de donner chaque élément de la Jacobienne de $\mathcal{C}$ par un couple de polynômes $\langle u(x), v(x) \rangle$, avec $\deg v < \deg u \leqslant g$. Sa taille est le degré de $u(x)$, c'est-à-dire le poids du diviseur réduit. Les éléments premiers sont ceux pour lesquels u est irréductible sur $\mathbb{F}_q[x]$. L'algorithmique dans le groupe se fait par l'algorithme de Cantor en temps polynomial et le test de friabilité et la décomposition en éléments premiers se ramenant au problème identique pour les polynômes sur $\mathbb{F}_q$ peuvent se faire en temps polynomial probabiliste.

La taille de l'entrée est donc $O(\log N')$ où $N' = q^g$. L'hypothèse $N \in O^{\sim}(N')$ est vérifiée car $N \leqslant (2g+1)q^g$. Pour q premier, cette borne est due à Artin [Art24, p. 24, formule (8)], et les arguments s'étendent aisément au cas général en remplaçant le caractère d'Artin par le caractère quadratique général.

10.2 Algorithme

L'algorithme est décrit pour un groupe cyclique dont l'ordre est connu mais non nécessairement premier. À la différence de l'approche Pohlig–Hellman, le problème initial n'est pas décomposé en une série de problèmes dans des sous-groupes d'ordre premier. En effet, il est nécessaire de travailler dans le groupe entier de manière à garantir une proportion suffisante et prouvable d'éléments friables. Cependant l'ordre du groupe est tout de même factorisé de manière à faciliter la phase d'algèbre linéaire.

Au chapitre suivant, où nous considérons le point de vue pratique plutôt que l'analyse prouvée, nous retournerons à l'approche Pohlig–Hellman.

Algorithme 10.12 Calcul d'index générique
Entrée: Un générateur g_1 d'un groupe cyclique G d'ordre N et un élément $g_2 \in G$.
Sortie: Un log discret $\log_{g_1} g_2$.

1. Choisir une borne de friabilité S et construire la *base de facteurs* $\mathcal{F}_S = \{p_1, \ldots, p_n\}$ avec $n = n_S$. Soit $k = \lceil \log_2 n + \log_2 \log_2 N \rceil + 1$.

2. Construire une matrice $A = (a_{ij}) \in (\mathbb{Z}/N\mathbb{Z})^{n \times (2kn)}$ comme suit. Pour $j = 1, \ldots, kn$, choisir au hasard uniformément α_j, $\beta_j \in \mathbb{Z}/N\mathbb{Z}$ jusqu'à ce que $\alpha_j g_1 + \beta_j g_2$ soit S-friable, et écrire
$$\alpha_j g_1 + \beta_j g_2 = \sum_{i=1}^{n} a_{ij} p_i. \tag{10.1}$$
Pour $j = kn+1, \ldots, 2kn$, écrire $j = (k+l)n + m$ avec $0 \leqslant l \leqslant k-1$, $1 \leqslant m \leqslant n$, et choisir au hasard uniformément α_j, $\beta_j \in \mathbb{Z}/N\mathbb{Z}$ jusqu'à ce que $\alpha_j g_1 + \beta_j g_2 - p_m$ soit S-friable ; alors écrire
$$\alpha_j g_1 + \beta_j g_2 = p_m + \sum_{i=1}^{n} b_{ij} p_i = \sum_{i=1}^{n} a_{ij} p_i. \tag{10.2}$$
(En pratique on crée seulement un peu plus de n colonnes du premier type, cf chapitre suivant. Le second type de relation est là pour rendre possible une preuve rigoureuse du temps de calcul.)

3. Par la procédure probabiliste décrite en Section 10.3, essayer de trouver un vecteur non nul $\gamma = (\gamma_1, \ldots, \gamma_{2kn}) \in \mathrm{Ker}(A)$. (Durant cette phase, N est factorisé.) Si la procédure échoue, retourner en 2.

4. Si $\sum_{j=1}^{2kn} \beta_j\gamma_j$ est inversible dans $\mathbb{Z}/N\mathbb{Z}$, alors retourner

$$-\left(\sum_{j=1}^{2kn} \beta_j\gamma_j\right)^{-1}\left(\sum_{j=1}^{2kn} \alpha_j\gamma_j\right);$$

sinon revenir en 2.

À l'étape 4, si l'algorithme termine alors il retourne le bon logarithme discret de g_2 en base g_1. Le fait que $\gamma \in \mathrm{Ker}(A)$ signifie que

$$0 = \sum_{j=1}^{2kn} a_{ij}\gamma_j \quad \forall i = 1, \ldots, n.$$

En multipliant ces équations par p_i et en les additionnant, on obtient

$$0 = \sum_{j=1}^{2kn}\left(\sum_{i=1}^{n} a_{ij}p_i\right)\gamma_j = \left(\sum_{j=1}^{2kn} \alpha_j\gamma_j\right) g_1 + \left(\sum_{j=1}^{2kn} \beta_j\gamma_j\right) g_2.$$

Comme g_1 et g_2 sont tous les deux des éléments de N-torsion, la multiplication par l'inverse de $\sum_{j=1}^{2kn} \beta_j\gamma_j$ dans $\mathbb{Z}/N\mathbb{Z}$, s'il existe, montre l'exactitude du résultat.

10.3 Algèbre linéaire creuse

Comme $\mathrm{Rg}\, A \leqslant n$, il est possible de trouver un vecteur non nul $\gamma \in \mathrm{Ker}(A)$. La manière d'effectuer cela efficacement mérite de plus amples explications. D'un coté il est souhaitable d'exploiter la structure creuse de la matrice, qui n'a que $O^{\sim}(1)$ termes par colonne et les algorithmes correspondants peuvent échouer avec une certaine probabilité. D'autre part une complication est introduite par le fait que N n'est pas supposé premier de sorte que $\mathbb{Z}/N\mathbb{Z}$ n'est pas forcément un corps.

10.3.1 Algorithme de Lanczos randomisé

Afin d'exploiter la structure creuse de la matrice, on peut utiliser l'algorithme de Lanczos ; nous nous appuyons sur le résultat suivant, qui s'obtient immédiatement d'après le théorème 6.2 dans [EK97].

Théorème 10.2 *Soit $\mathbb{F}_q$ le corps fini à q éléments, soit $A \in \mathbb{F}_q^{n\times d}$ une matrice de rang r avec ω termes non nuls et soit $b \in \mathbb{F}_q^n$. Il existe un algorithme probabiliste qui retourne un vecteur $x \in \mathbb{F}_q^d$ tel que $Ax = b$ ou bien renvoie un message d'erreur. L'algorithme nécessite $O(r(\omega + d))$ opérations dans $\mathbb{F}_q$ et a une probabilité d'échec d'au plus $\frac{11d^2-d}{2(q-1)}$.*

Le principe de l'algorithme est de perturber toutes les données par des éléments aléatoires de manière à pouvoir borner la probabilité d'échec de l'algorithme de Lanczos. Nous utiliserons l'anglicisme « randomisation » pour décrire cette méthode de perturbation aléatoire.

Le vecteur solution retourné par l'algorithme peut être rendu uniformément distribué parmi toutes les solutions possibles en randomisant le membre de droite de la manière suivante (en fait, cette randomisation fait déjà partie de l'algorithme dans [EK97]) :

– Choisir y uniformément dans $\mathbb{F}_q^d$;

– Résoudre $A\overline{x} = b + Ay$ par l'algorithme précédent ;

– Retourner $x = \overline{x} - y$.

Si y varie dans une classe fixée de $\mathbb{F}_q^d/\mathrm{Ker}A$, alors $\overline{x}$ ne dépend pas de y, et x est distribué uniformément dans l'espace des solutions $\overline{x} + \mathrm{Ker}A$. Ainsi, la même assertion a lieu quand y n'appartient pas à une classe fixée.

10.3.2 Uniformisation de la probabilité d'échec

Quand q est petit par rapport à d, il n'est pas possible d'appliquer le théorème directement car la probabilité d'échec devient trop grande. La solution est de transférer le problème dans une extension du corps de base. Cette idée n'est pas nouvelle, elle est évoquée dans [LO90], et [KS91], et est utilisée en pratique depuis longtemps. Toutefois nous n'avons pas trouvé tous les détails dans la littérature et donnons donc ici une preuve du résultat.

Théorème 10.3 *Soit $A \in \mathbb{F}_q^{n\times d}$ une matrice de rang r ayant ω entrées non nulles et $b \in \mathbb{F}_q^n$. Il existe un algorithme probabiliste qui retourne un vecteur $x \in \mathbb{F}_q^d$ tel que $Ax = b$ ou renvoie une erreur. Le temps d'exécution de l'algorithme est en $O(r(\omega+d)\log^2(dq))$, et sa probabilité d'échec est au plus $\frac{1}{2}$. De plus le vecteur résultat est uniformément distribué parmi toutes les solutions possibles.*

Démonstration. Dans la situation du théorème 10.2, soit p la caractéristique de $\mathbb{F}_q$, $\nu = \min\{l : q^l > 11d^2, p \nmid l\}$ et $q' = q^\nu$. Alors $q^{\nu-2} \leqslant 11d^2$, de sorte que $q' \in O(d^2q^2)$. Le principe est de résoudre un système linéaire sur $\mathbb{F}_{q'}$ et de projeter la solution sur une solution $x \in \mathbb{F}_q^d$ de $Ax = b$.

Pour la projection, on utilise la fonction trace $\mathrm{Tr} : \mathbb{F}_{q'} \to \mathbb{F}_q$, qui est un homomorphisme d'espaces vectoriels sur $\mathbb{F}_q$ et agit sur $\mathbb{F}_q$ comme une multiplication par ν. Soit $b' = \nu' b \in \mathbb{F}_q^d$ avec $\nu\nu' \equiv 1 \pmod p$, de sorte que $\nu b' = b$. La valeur ν' existe car pgcd $(\nu, p) = 1$ et peut être calculée par l'algorithme d'Euclide étendu en temps $O(\log \nu \log p)$, ce qui, tout comme la multiplication de b par ν' est négligeable par rapport à la résolution de système.

On résout $Ax' = b'$ par l'algorithme de [EK97]. La probabilité de succès de cette étape est au moins $1 - \frac{11d^2-d}{2(q'-1)} \geqslant \frac{1}{2}$. Soit $x = \mathrm{Tr}(x')$. Alors par linéarité de la trace on déduit que $Ax = \mathrm{Tr}(Ax') = \mathrm{Tr}(b') = \nu b' = b$. De plus, toutes les solutions $x \in \mathbb{F}_q^d$ de $Ax = b$ peuvent être obtenues de cette manière et toutes avec la même probabilité. Plus précisément, pour une solution x, l'ensemble des solutions de $Ax' = b'$ sur $\mathbb{F}_{q'}^d$ qui s'envoient sur x par la fonction trace est donné par $\nu' x + (\mathrm{Ker}A \cap \mathrm{Ker}\mathrm{Tr})$, dont la cardinalité $(q')^{\dim(\mathrm{Ker}A\cap\mathrm{Ker}\mathrm{Tr})}$ est indépendante de x. □

10.3.3 Application à l'algorithme de log discret

Dans le cas où N est premier, le théorème 10.3 résout le calcul d'algèbre linéaire de l'algorithme. Sinon, on factorise N, on calcule γ modulo p^ν pour tous les $p^\nu \| N$ et on combine les résultats par le théorème Chinois. Ces calculs modulo p^ν peuvent être décomposés en ν itérations modulo p par une procédure de lifting : supposons qu'une solution non nulle $\gamma_1 \in \{0, \ldots, p^e-1\}^{2kn}$ de l'équation $Ax \equiv 0 \pmod{p^e}$ soit connue, par exemple $A\gamma_1 = p^e\delta$ avec $\delta \in \mathbb{Z}^{2kn}$ et supposons qu'il existe une solution γ_2 de $Ax \equiv \delta \pmod p$. Alors $p^e\gamma_2 - \gamma_1$ est une solution non nulle de $Ax \equiv 0 \pmod{p^{e+1}}$. Si tous les calculs modulo un nombre premier retournent un vecteur uniformément distribué parmi toutes les solutions possibles, alors le résultat combiné est aussi uniformément distribué dans le noyau de A.

Cependant, considérant la forme normale de Smith de la matrice A, il est facile de voir que la procédure de lifting peut échouer si (et seulement si) $\mathrm{Rg}_{\mathbb{Q}} A \neq \mathrm{Rg}_{\mathbb{Z}/p\mathbb{Z}} A$ car alors l'équation $Ax \equiv \delta \pmod p$ n'a pas nécessairement de solution. C'est la raison pour laquelle, suivant [Pom87], nous avons construit une matrice A d'une manière particulière : on engendre bien plus que les $n+1$ colonnes nécessaires en pratique et on introduit les éléments de la base canonique p_m dans la matrice. En effet, il est prouvé dans le lemme 4.1 et la remarque subséquente de [Pom87] qu'ainsi la matrice a rang maximal sur $\mathbb{Z}/p\mathbb{Z}$ avec grande probabilité. Nous rappelons ce lemme avec nos notations.

Lemme 10.1 *Soit V un espace vectoriel sur un corps $\mathbb{F}$ avec $\dim V = n < \infty$. Soit $\mathcal{S}$ un ensemble fini de vecteurs de V et $b_1, \ldots, b_n$ une base de V. Soit $k \in \mathbb{N}^*$. Soient $2kn$ tirages indépendants d'éléments de $\mathcal{S}$ avec une distribution de probabilité arbitraire sur $\mathcal{S}$; on note $v_1, \ldots, v_{kn}, w_1, \ldots, w_{kn}$ les vecteurs obtenus. Soit V' le sous-espace de V engendré par $v_1, \ldots, v_{kn}$, et les vecteurs $b_j + w_{(j-1)k+i}$ pour $j = 1, \ldots, n$ et $i = 1, \ldots, k$.*

Alors avec probabilité au moins $1 - \frac{n}{2^{k-1}}$, on a $V = V'$.

Dans notre cas, l'espace vectoriel V est l'espace des vecteurs colonnes de taille n à coefficients dans $\mathbb{Z}/p\mathbb{Z}$, la base est la base canonique, et l'ensemble $\mathcal{S}$ est l'ensemble des vecteurs colonnes représentant un élément friable de G. Les vecteurs engendrant V' correspondent précisément aux vecteurs formant la matrice A. Ainsi la probabilité que le lifting soit possible dans $\mathbb{Z}/p\mathbb{Z}$ est au moins $1 - \frac{n}{2^{k-1}}$.

Il y a au plus $\frac{\log_2 N}{2}$ premiers p distincts dont le carré divise N, donc la probabilité que le lifting soit possible pour chacun d'eux est au moins

$$1 - \frac{n \log_2 N}{2^k} \geqslant \frac{1}{2}$$

avec notre choix de k. Dans ce cas, en répétant $\log_2(2\log_2 N)$ fois l'algorithme du théorème 10.3, on obtient une solution d'un problème modulo un nombre premier avec probabilité au moins

$$1 - \frac{1}{2^{\log_2(2\log_2 N)}} = 1 - \frac{1}{2\log_2 N}.$$

Comme de tels sous-problèmes sont en quantité au plus $\log_2 N$, on obtient une solution modulo N avec probabilité au moins $\frac{1}{2}$.

Combinant tout cela, l'étape 3 est passée avec succès avec une probabilité d'au moins $\frac{1}{4}$, auquel cas le vecteur solution est uniformément distribué dans tout le noyau.

10.4 Complexité

10.4.1 Probabilité de succès et temps de calcul

Pour estimer la probabilité de succès de l'algorithme lors d'une exécution des étapes 2 à 4, on suppose que l'étape 2 a été accomplie avec succès, l'étude de cette étape étant reportée plus bas.

Comme montré ci-dessus, l'étape 3 est accomplie avec succès avec probabilité au moins $\frac{1}{4}$. L'algorithme peut aussi échouer si $\sum_{j=1}^{2kn} \beta_j \gamma_j$ n'est pas inversible dans $\mathbb{Z}/N\mathbb{Z}$ à l'étape 4. Toutefois, cela se produit avec une probabilité suffisamment faible. Pour un $j \leqslant kn$ donné, et un β_j quelconque, comme g_1 est un générateur de G et α_j est uniformément distribué, l'élément

$\alpha_j g_1 + \beta_j g_2$ est uniformément distribué parmi tous les éléments du groupe. C'est encore vrai pour $j > kn$ et $\alpha_j g_1 + \beta_j g_2 - p_m$. Ainsi, la matrice A et le vecteur β sont des variables aléatoires indépendantes, de sorte que γ et β sont aussi indépendants.

Soit p un diviseur premier de N. Comme γ est uniformément distribué parmi tous les vecteurs du noyau, la probabilité que $\gamma \not\equiv 0 \pmod p$ est au moins $1 - \frac{1}{p}$. Alors l'espace orthogonal à $\gamma \bmod p$ dans $\mathbb{Z}^{2kn}$ a dimension $2kn - 1$, et la probabilité conditionnelle que $\beta \bmod p$ n'est pas orthogonal à $\gamma \bmod p$ est au moins $1 - \frac{1}{p}$. Ainsi $\sum_{j=1}^{2kn} \beta_j \gamma_j$ est inversible dans $\mathbb{Z}/N\mathbb{Z}$ avec probabilité au moins

$$\prod_{p|N} \left(1 - \frac{1}{p}\right)^2 = \left(\frac{\varphi(N)}{N}\right)^2.$$

D'après (3.41) dans [RS62] on a $\frac{\varphi(N)}{N} \in \Omega(1/\log\log N)$.

Ainsi la probabilité de succès d'une exécution des étapes 2 à 4 est en

$$\Omega\left(\frac{1}{(\log\log N)^2}\right).$$

Pour simplifier, on note $n' = n'_S$ le nombre d'éléments dans le monoïde $\mathbb{M}$ dont la taille est majorée par S.

Avec nos hypothèses sur le groupe, $\mathcal{F}_S$ peut être construit en temps $O^\sim(n'^2)$.

Soit N_S le nombre d'éléments S-friables dans G, et soit t_s et t_d des bornes sur le temps de calcul pour le test de friabilité et respectivement, pour la décomposition d'un élément friable en éléments premiers. Le temps pour calculer une combinaison linéaire de g_1 et g_2 et tester sa friabilité est en $O^\sim(t_s)$; cela doit être répété en moyenne $\frac{N}{N_S}$ fois jusqu'à trouver un élément friable. Cet élément est reconnu avec une probabilité au moins $1/2$, donc en moyenne au plus deux répétitions de la procédure précédente sont nécessaires pour remplir une colonne de la matrice. Ainsi le temps moyen de calcul pour l'étape 2 est en

$$O^\sim\left(n\left(\frac{N}{N_S}t_s + t_d\right)\right) \subseteq O^\sim\left(n\frac{N}{N_S}t_s + n^2\right)$$

car $2kn,\ t_d \in O^\sim(n)$.

Soit t_f une borne sur le temps de calcul nécessaire à la factorisation de l'entier N. Comme expliqué à la section 10.3, $\log_2(2\log_2 N)\log_2 N \in O^\sim(1)$ exécutions de l'algorithme du théorème 10.3 sont nécessaires à l'étape 3. Le nombre de termes dans chaque colonne de A est en $O^\sim(1)$, ainsi l'étape 3 s'exécute en temps $O^\sim(t_f + n^2)$.

Finalement, l'étape 4 peut être effectuée en $O^\sim(n)$.

Comme seulement $O((\log\log N)^2) \subseteq O^\sim(1)$ répétitions des étapes 2 à 4 sont nécessaires en moyenne et $n \leqslant n'$, le temps de calcul total de l'algorithme est en

$$O^\sim\left(t_f + n'^2 + n'\frac{N}{N_S}t_s\right). \tag{10.3}$$

(Dans tous les cas considérés, n et n' diffèrent seulement par un facteur polynomial en $\log N'$, i.e. $O^\sim(n) = O^\sim(n')$, de sorte que l'on ne perd rien en remplaçant n par n'.)

Exemples

1. $G = \mathbb{F}_p^*$, p premier
   Avec les algorithmes déterministes de Pollard et Strassen [Pol74, Str76] on a $t_s \in O^\sim(\sqrt{n'})$.

Une méthode probabiliste plus efficace a été prouvée en utilisant les courbes hyperelliptiques. Le test de [LPP93] reconnaît (et décompose) un nombre friable avec probabilité au moins $1/2$ en temps $t_s \in O^{\sim}(L_{n'}(2/3, c))$, où c est une constante positive. Ainsi le temps de calcul total est en

$$O^{\sim}\left(t_f + n'^2 + n' L_{n'}(2/3, c)\frac{N}{N_S}\right).$$

2. $G = \mathbb{F}_{2^k}^*$
   Maintenant $t_s \in O^{\sim}(1)$, car un test de friabilité peut être effectué en temps polynomial déterministe par la « distinct degree factorisation » du polynôme représentant l'élément. Précisément, soit $f \in \mathbb{F}_2[X]'$ l'élément à tester, et $g = \frac{f}{\operatorname{pgcd}(f, f')}$ sa partie sans facteur carré. Alors f est S-friable si et seulement si g l'est. Comme $X^{2^i} - X$ est le produit de tous les polynômes irréductibles de degré divisant i dans $\mathbb{F}_2[X]'$, g est friable si et seulement si

   $$g = \operatorname{ppcm}\left(\left\{\operatorname{pgcd}(g, X^{2^i} - X)\ :\ i = 1, \ldots, S\right\}\right).$$

   Calculant $X^{2^i} - X \bmod g$ par des carrés et réductions successifs modulo g, cela peut être testé en temps polynomial en S et $\deg f \in O(\log N)$. Ainsi le temps de calcul total de l'algorithme est en

   $$O^{\sim}\left(t_f + n'^2 + n'\frac{N}{N_S}\right).$$

3. $G = \mathbb{F}_{p^k}$, p premier
   Un élément $(m, f) \in \mathbb{N}^* \times \mathbb{F}_p[X]'$ est S-friable si et seulement si m et f sont S-friables. La friabilité de m peut être testée en temps $O^{\sim}(L_p(2/3, c))$ comme dans le premier exemple et la friabilité de f en temps $O^{\sim}(1)$ comme dans le deuxième. Ainsi le temps de calcul total est en

   $$O^{\sim}\left(t_f + n'^2 + n' L_p(2/3, c)\frac{N}{N_S}\right).$$

4. Groupes de classes de corps quadratiques imaginaires
   Le test de friabilité et la décomposition en éléments premiers se réduisent au cas des entiers naturels, aussi l'analyse du premier exemple montre que le temps de calcul est en

   $$O^{\sim}\left(t_f + n'^2 + n' L_{n'}(2/3, c)\frac{N}{N_S}\right).$$

5. Jacobiennes de courbes hyperelliptiques
   Ici le test de friabilité et la décomposition se réduisent au cas des polynômes unitaires sur un corps fini, l'analyse du deuxième exemple s'applique et donne un temps de calcul en

   $$O^{\sim}\left(t_f + n'^2 + n'\frac{N}{N_S}\right).$$

10.4.2 Sous-exponentialité

Sauf mention du contraire, le niveau de sous-exponentialité sera toujours $1/2$, aussi note-t-on $L_{N'}(c)$ pour $L_{N'}(\frac{1}{2}, c)$.

Supposons que l'on dispose d'un résultat du type : « La borne S peut être choisie de telle sorte que

$$n' \in O(L_{N'}(\rho + o(1)))$$

et

$$\frac{N}{N_S} \in O(L_{N'}(\sigma + o(1)))$$

pour certaines constantes ρ, $\sigma > 0$.»

L'hypothèse $N \in O^{\sim}(N')$ implique que $L_N(c) \in O(L_{N'}(c) + o(1))$. Prenant en compte que N peut être factorisé en temps moyen $O(L_N(1 + o(1))) \subseteq O(L_{N'}(1 + o(1)))$ par l'algorithme présenté dans [LP92] et introduisant un exposant τ tel que $t_s \in O^{\sim}(n'^{\tau})$, on peut spécialiser (10.3) pour obtenir

$$O\left(L_{N'}\left(\max(1, 2\rho, (1 + \tau)\rho + \sigma) + o(1)\right)\right).$$

En fait, les constantes pour tous les exemples ci-dessous sont supérieures ou égales à 1, de sorte que la factorisation de N n'a pas d'influence sur le temps de calcul.

Exemples

1. $G = \mathbb{F}_p^*$, p premier ; $N' = N = p - 1$
   Avec les notations usuelles $\psi(x, y)$ pour le nombre d'entiers entre 1 et x dont tous les facteurs premiers sont majorés par y, on a $\frac{N}{N_S} = \frac{N}{\psi(N, 2^S)}$. Soit $S = \lceil \log(L_N(\rho)) \rceil$, de sorte que $n' = 2^S \in [L_N(\rho), 2L_N(\rho)]$. Alors le lemme 3.1 dans [Pom87] montre que $\frac{N}{N_S} \in O(L_N(\sigma + o(1)))$ avec $\sigma = \frac{1}{2\rho}$. De plus, de $n' \in O(L_N(\rho))$ on déduit $L_{n'}(2/3, c) \in L_N(o(1))$. Le temps de calcul de l'algorithme est donc en

   $$O\left(L_N\left(\max\left(2\rho, \rho + \frac{1}{2\rho}, 1\right) + o(1)\right)\right)$$

   pour tout $\rho > 0$; le choix optimal $\rho = 1/\sqrt{2}$ donne un temps de calcul en

   $$O(L_N(\sqrt{2} + o(1))).$$

   C'est précisément la complexité du plus rapide algorithme connu décrit dans [Pom87].

2. $G = \mathbb{F}_{2^k}^*$; $N' = N = 2^k - 1$
   Soit $N_q(d, m)$ le nombre de polynômes unitaires de degré d sur $\mathbb{F}_q$ dont les facteurs premiers sont de degré au plus m. Alors $\frac{N}{N_S} \leqslant \frac{2^k}{N_2(k-1, S)}$. Soit $S = \lceil \log(L_N(\rho)) \rceil$, de sorte que $n' \in \Theta(L_N(\rho))$. Le théorème 2.1 dans [LP98] montre que $N_2(k-1, S) \in \frac{2^{k-1}}{u^{(1+o(1))u}}$ pour $u = \frac{k-1}{S} \leqslant \frac{1}{\rho}\sqrt{\frac{\log N}{\log\log N}} \leqslant \frac{1}{\rho}\sqrt{\log N}$. Ainsi, un court calcul montre que $\frac{N}{N_S} \in O(L_N(\sigma + o(1)))$ pour $\sigma = \frac{1}{2\rho}$, et le temps de calcul de l'algorithme est en $O\left(L_N\left(\max\left(2\rho, \rho + \frac{1}{2\rho}, 1\right) + o(1)\right)\right)$ pour tout $\rho > 0$. Le choix optimal $\rho = 1/\sqrt{2}$ donne un temps de calcul en

   $$O(L_N(\sqrt{2} + o(1))),$$

   ce qui correspond au plus rapide algorithme connu décrit dans [Pom87] et [LP98].

3. $G = \mathbb{F}_{p^k}$, p premier ; $N' = N = p^k - 1$
   Remarquons tout d'abord qu'avec la représentation polynomiale que nous avons choisie, il est impossible d'obtenir un temps de calcul sous-exponentiel pour un $k \geqslant 2$ fixé et $p \to \infty$. Si l'on fixe $S = 0$, alors, seules les constantes ont une chance d'être friables, et $\frac{N}{N_S} \geqslant \frac{p^k - 1}{p - 1} \geqslant p^{k-1}$ est exponentiel en N. Si $S \geqslant 1$, alors les p polynômes linéaires unitaires

sont inclus dans la base de facteur, qui est donc de taille exponentielle. Aussi doit-on se restreindre aux cas où p est suffisamment petit par rapport à k.

Soit $S = \lceil \log_p(L_N(\rho)) \rceil$, on obtient l'estimation

$$\frac{N}{N_S} \leqslant \frac{p}{\psi(p-1, 2^S)} \cdot \frac{p^{k-1}}{N_p(k-1, S)} \in O\left(pL_N\left(\frac{1}{2\rho} + o(1)\right)\right)$$

comme dans le deuxième exemple, ce qui introduit un facteur p indésirable. De plus, comme on doit arrondir S à l'entier supérieur, il n'est plus toujours vrai que $n' \in O(L_N(\rho))$. En fait,

$$\begin{aligned}
n' &= \sum_{i=0}^{S} \left|\{f \in \mathbb{F}_p'[X] : \deg f = i\}\right| \cdot \left|\{m \in \{1, \ldots, p-1\} : \log_2 m \leqslant S - i\}\right| \\
&= \sum_{i=0}^{S} p^i \min\left\{p-1, 2^{S-i}\right\} \\
&\leqslant \sum_{i=0}^{S-1} p^{i+1} + p^S \\
&\in O\left(p^S\right) \\
&\subseteq O(pL_N(\rho)).
\end{aligned}$$

Un premier cas particulier est lorsque $p \in O(\log N)$ ou plus généralement $p \in O^{\sim}(1)$, ce qui implique $n' \in O^{\sim}(L_N(\rho))$ et $L_p(2/3, c) \in L_N(o(1))$. Ainsi l'analyse de l'exemple 2 s'applique sans modification.

Plus généralement, on doit s'assurer que p est sous-exponentiel en $\log N = k \log p$. Suivant les idées de [Eng99], on considère le cas $k \geqslant \vartheta \log p$ où ϑ est une constante, de sorte que $p \leqslant L_N\left(\frac{1}{\sqrt{\vartheta}}\right)$. Alors $n' \in O\left(L_N\left(\rho + \frac{1}{\sqrt{\vartheta}}\right)\right)$ et $\frac{N}{N_S} \in O\left(L_N\left(\frac{1}{2\rho} + \frac{1}{\sqrt{\vartheta}} + o(1)\right)\right)$ pour la même valeur de S qu'au dessus, $L_p(2/3, c) \in L_N(o(1))$ et le temps de calcul total est en

$$O\left(L_N\left(\max\left\{2\rho + \frac{2}{\sqrt{\vartheta}}, \rho + \frac{1}{2\rho} + \frac{2}{\sqrt{\vartheta}}, 1\right\} + o(1)\right)\right).$$

Le choix optimal pour ρ est $\frac{\sqrt{2}}{2}$, ce qui donne un temps de calcul de

$$O\left(L_N\left(\sqrt{2} + \frac{2}{\sqrt{\vartheta}} + o(1)\right)\right).$$

Asymptotiquement pour $\vartheta \to \infty$ (e.g., pour p fixé), on retrouve le temps de calcul du deuxième exemple.

Dans [AD93], Adleman et DeMarrais décrivent un algorithme avec un temps d'exécution heuristique sous-exponentiel pour $p > k$. Ils représentent le corps fini comme l'anneau des entiers d'un corps de nombres modulo un idéal premier (voir aussi [Sem95]). La question de savoir s'il existe un algorithme sous-exponentiel prouvé pour tous les corps finis est un problème ouvert intéressant.

4. Groupes de classes des corps quadratiques imaginaires
Soit D le discriminant du corps de nombres. Dans [Sey87], proposition 4.4, il est montré, sous l'hypothèse de Riemann généralisée que $\frac{N}{N_S} \in O\left(L_{|D|}\left(\frac{1}{4c}+o(1)\right)\right)$ pour $S = \lceil \log L_{|D|}(c) \rceil$. Par un théorème de Siegel [Sie35], $\log |D| \in (2+o(1)) \log N$ de sorte que $L_{|D|}(c+o(1)) = L_N(\sqrt{2}c + o(1))$. Prenant $S = \lceil \log L_N(\rho) \rceil$, on déduit $n \in O(L_N(\rho))$ et $\frac{N}{N_S} \in O\left(L_N\left(\frac{1}{2\rho}+o(1)\right)\right)$. Reprenant l'analyse du premier exemple, on obtient un temps de calcul en
$$O\left(L_N\left(\sqrt{2}+o(1)\right)\right) = O\left(L_{|D|}\left(1+o(1)\right)\right)$$
sous l'hypothèse de Riemann généralisée. Ceci améliore le temps de calcul en $O(L_{|D|}(\sqrt{2}+o(1)))$ de l'algorithme décrit dans [HM89] pour calculer la structure du groupe de classes dans utiliser d'information supplémentaire.

5. Jacobiennes de courbes hyperelliptiques
Dans ce cas $N' = q^g$. Soit $S = \lceil \log_q L_{q^g}(\rho) \rceil$, on a $n \leqslant 2qL_{q^g}(\rho)$. Là encore nous suivons [Eng99] et considérons seulement les instances pour lesquelles $g \geqslant \vartheta \log q$ où ϑ est une constante, de sorte que $q \leqslant L_{q^g}\left(\frac{1}{\sqrt{\vartheta}}\right)$. Dans [ES00] il est montré qu'alors $N_S \geqslant q^g L_{q^g}\left(-\frac{1}{2\rho}\right)$, ainsi $\frac{N}{N_S} \in O^{\sim}\left(\frac{N'}{N_S}\right) \subseteq O\left(L_{q^g}\left(\frac{1}{2\rho}+o(1)\right)\right)$, et le temps de calcul de l'algorithme est en
$$O\left(L_{q^g}\left(\max\left\{2\rho+\frac{2}{\sqrt{\vartheta}}, \rho+\frac{1}{2\rho}+\frac{1}{\sqrt{\vartheta}}, 1\right\}+o(1)\right)\right).$$
Une analyse similaire à celle de l'exemple 3 montre que le choix optimal de ρ est
$$\min\left\{\frac{\sqrt{2}}{2}, \sqrt{\frac{1}{2}+\frac{1}{4\vartheta}}-\frac{1}{2\sqrt{\vartheta}}\right\} = \sqrt{\frac{1}{2}+\frac{1}{4\vartheta}}-\sqrt{\frac{1}{4\vartheta}},$$
ce qui donne un temps de calcul en
$$O\left(L_{q^g}\left(\sqrt{2}\left(\sqrt{1+\frac{1}{2\vartheta}}+\sqrt{\frac{1}{2\vartheta}}\right)+o(1)\right)\right).$$
Cela améliore le temps de calcul en
$$O\left(L_{q^g}\left(\frac{5}{\sqrt{6}}\left(\sqrt{1+\frac{3}{2\vartheta}}+\sqrt{\frac{3}{2\vartheta}}\right)+o(1)\right)\right)$$
de [Eng99], et asymptotiquement pour $\vartheta \to \infty$ (e.g., pour q constant), la constante dans la fonction sous-exponentielle est encore la même que pour l'exemple 2.

10.5 Groupes non cycliques

À la différence des groupes multiplicatifs de corps finis, les groupes de classes sont parfois non cycliques. Par exemple le nombre de facteurs cycliques dans la Jacobienne d'une courbe hyperelliptique peut être deux fois le genre. C'est pourquoi il est important de savoir calculer des logarithmes discrets dans un sous-groupe cyclique $H = \langle g_1 \rangle$ d'un groupe abélien G donné. Heuristiquement, cela ne pose pas de problèmes : l'algorithme décrit ci-dessus fonctionne encore.

Avec l'hypothèse que l'appartenance à un sous-groupe et la friabilité sont deux notions indépendantes, i.e. la proportion d'éléments friables est la même dans H que dans G, l'analyse reste valable.

Mais nous nous intéressons ici à des temps de calcul prouvé, et les résultats de friabilité utilisés ne s'appliquent qu'au groupe en entier. Dans cette section, nous proposons quelques approches pour remédier à cette situation.

10.5.1 Perturbation par des éléments du supplémentaire

La situation la plus simple se produit lorsque pgcd $\left(|H|, \frac{|G|}{|H|}\right) = 1$; alors H admet un supplémentaire H' dans G, i.e., $G = H \times H'$. Supposons qu'il est possible de tirer aléatoirement des éléments h_j de H' selon une distribution uniforme en temps polynomial en $\log N'$. C'est par exemple le cas si l'on sait le faire pour tout le groupe G. Un autre situation favorable est le cas où l'on connaît une base de H'. (Dans ce contexte, une base de H' est un ensemble $\{b_1, \ldots, b_r\}$ tel que H' est la somme directe $\langle b_1 \rangle \times \cdots \times \langle b_r \rangle$. Le cardinal r d'une base n'est pas un invariant de H', mais il est majoré par $\log_2 |H'|$.)

Alors $\alpha_j g_1 + \beta_j g_2 + h_j$ est uniformément distribué dans G et indépendant de β, de sorte que l'algorithme peut être poursuivi avec ces éléments à la place de $\alpha_j g_1 + \beta_j g_2$. Si l'algèbre linéaire n'échoue pas, alors

$$\left(\sum_{j=1}^{2kn} \alpha_j \gamma_j\right) g_1 + \left(\sum_{j=1}^{2kn} \beta_j \gamma_j\right) g_2 = -\sum_{j=1}^{2kn} \gamma_j h_j \in H \cap H' = \{0\},$$

et

$$\log_{g_1} g_2 = \left(\sum_{j=1}^{2kn} \beta_j \gamma_j\right)^{-1} \left(\sum_{j=1}^{2kn} \alpha_j \gamma_j\right)$$

comme précédemment. De même l'analyse du temps de calcul est inchangée.

Bien que cette situation semble très spéciale, elle est typique de ce qui se passe en cryptographie, où H est supposé avoir un grand ordre premier et le cofacteur $\frac{|G|}{|H|}$ est petit, si bien que $|H|$ et $\frac{|G|}{|H|}$ sont automatiquement premiers entre eux. De plus, pour $\frac{|G|}{|H|}$ polynomial en $\log N'$, la structure et en particulier une base de $H' \simeq G/H$ peut être déterminée en temps polynomial (voir par exemple [Coh93]), et les hypothèses de cette section sont satisfaites.

10.5.2 Utilisation d'une base de G

Supposons qu'une base $\{b_1, \ldots, b_r\}$ de G ainsi que les ordres $e_1, \ldots, e_r$ de ses éléments soient connus. Alors le problème du logarithme discret peut être résolu en deux étapes. Au lieu d'écrire directement g_2 comme un multiple de g_1 on exprime d'abord g_1 comme une combinaison linéaire des éléments de la base et on procède ensuite de même pour g_2. Le log discret se déduit alors au prix de quelques opérations modulo les e_i.

Afin d'écrire g_1 en termes des b_i, une petite variante de l'algorithme permet d'utiliser les propriétés de friabilité. Pour un $j \leqslant kn$ donné, on tire aléatoirement des α_{ij} et β_j jusqu'à ce que $\sum \alpha_{ij} b_i + \beta_j g_1$ soit S-friable et on écrit cet élément comme $\sum a_{ij} p_i$. De même, pour $j > kn$ on tire aléatoirement des entiers jusqu'à ce que $\sum \alpha_{ij} b_i + \beta_j g_1 - p_m$ soit S-friable. Là encore, les éléments de G dont on teste la friabilité sont distribués uniformément dans G et sont indépendants de β, de sorte que l'analyse de la section 10.4.1 peut être appliquée. Ainsi avec grande probabilité, un vecteur non nul du noyau est obtenu et g_1 est exprimé comme une combinaison linéaire $g_1 = \sum \gamma_i b_i$. La même méthode donne $g_2 = \sum \delta_i b_i$. On essaye ensuite de résoudre le système d'équations $\delta_i \equiv l\gamma_i \pmod{e_i}$. Si c'est possible, alors l est le log discret cherché. Sinon, g_2 n'appartient pas au sous-groupe cyclique engendré par g_1.

Chapitre 11

Calcul d'index en genre « petit »

L'algorithme du chapitre précédent s'applique aux courbes hyperelliptiques et prouve qu'il existe un algorithme qui s'exécute en temps sous-exponentiel en q^g, dès que le genre est suffisamment grand par rapport au corps fini considéré. Toutefois, pour les applications pratiques, et notamment dans un contexte cryptographique, il est plus pertinent de considérer le genre fixe et de faire tendre la taille du corps q vers l'infini. Ce chapitre décrit d'une part les modifications à apporter par rapport à l'algorithme général afin de prendre en compte ce changement de point de vue et d'autre part des améliorations qui permettent d'accélérer les calculs en pratiques, même si l'on doit troquer les preuves pour des heuristiques.

Cette version pratique de l'algorithme a été découverte avant le cadre général théorique mis en place avec A. Enge. Les résultats ont donné lieu à la publication [Gau00].

11.1 Notion de diviseur friable

La notion d'élément friable est essentiellement celle proposée initialement par ADH. Le point crucial est que l'on considère toujours la borne de friabilité égale à $S = 1$.

Définition 11.1 *Soit $\mathcal{C}$ une courbe hyperelliptique de genre g définie sur un corps fini $\mathbb{F}_q$. Soit $D = \langle u(x), v(x) \rangle$ un diviseur semi-réduit en représentation de Mumford. Le diviseur D est dit friable si le polynôme $u(x)$ est scindé sur $\mathbb{F}_q$.*

Autrement dit, un diviseur est friable si et seulement si les points qui le composent sont définis sur $\mathbb{F}_q$.

Lorsque l'on fait tendre q vers l'infini en laissant g fixé, la taille du groupe tend vers l'infini, et la proportion optimale d'éléments friables est sous-exponentielle en cette taille de groupe. Même en prenant $S = 1$ qui est le choix minimal, la proportion d'éléments friables est environ $\frac{q}{q^g}$, ce qui est exponentiel en la taille du groupe (ne pas oublier que g est fixé). C'est pourquoi le choix $S = 1$ est optimal : on prend un nombre d'éléments friables minimal et qui est déjà trop grand par rapport à ce qu'il faudrait.

Bien entendu, si l'on veut effectuer un vrai calcul pour des valeurs de q et g données, il faut vérifier que l'on est bien dans la zone où $S = 1$ est optimal. Pour donner un ordre de grandeur, ceci est avéré pour un genre borné par 8 ou 9, dès que q est trop grand pour qu'une attaque en racine carrée puisse fonctionner.

Définition 11.2 *La base de facteur est l'ensemble des diviseurs de poids 1. On la note $\mathcal{F}$, et son cardinal w.*

11.2 Algorithme et complexité

La méthode employée est essentiellement la même qu'au chapitre précédent, mais, étant intéressé par des applications pratiques plus que par des complexités prouvées, nous allons abandonner toutes les randomisations, et autres astuces à la Pomerance qui ne sont absolument pas nécessaires en pratique. D'autre part, afin de construire les relations le plus efficacement possible, nous allons réintroduire une marche pseudo-aléatoire similaire à celle employée dans la méthode Rho.

11.2.1 Algorithme

Pour simplifier, nous supposons que la Jacobienne est d'ordre presque premier, et que l'on doit calculer le log discret de D_2 par rapport à D_1 dont l'ordre N est un grand nombre premier.

La marche pseudo-aléatoire est définie comme dans la section 9.1.2. Soit $R_0 = \alpha_0 D_1 + \beta_0 D_2$ le point de départ de la marche, où R_0 est le diviseur réduit obtenu par l'algorithme de Cantor, et α_0 et β_0 sont des entiers aléatoires. Pour j allant de 1 à r, on précalcule les décalages

$$T_j = a_j D_1 + b_j D_2.$$

La marche est alors définie par $R_{i+1} = R_i + T_{\mathcal{H}(R_i)}$, où $\mathcal{H}$ est une fonction de hachage ; en pratique, celle-ci est obtenue par les derniers bits de la représentation interne des diviseurs sous forme de Mumford. Chaque nouvel élément nécessite une unique opération dans la Jacobienne. De plus, à chaque étape, on garde une représentation de R_{i+1} comme $\alpha_{i+1}D_1 + \beta_{i+1}D_2$, où α_{i+1} et β_{i+1} sont des entiers modulo N.

La méthode Rho classique consiste à attendre une collision $R_{i_1} = R_{i_2}$, ce qui fournit le log discret $\lambda = -(\alpha_{i_1} - \alpha_{i_2})/(\beta_{i_1} - \beta_{i_2}) \bmod N$. Utilisons la friabilité : pour chaque R_i dans la marche aléatoire, on peut tester s'il est friable. Dans l'affirmative, on peut l'exprimer sur la base de facteurs, et sinon, on le jette. Ainsi on extrait une sous-suite de la suite R_i dont tous les éléments sont friables. On note encore R_i cette sous-suite. On peut alors mettre le résultat de ce calcul dans une matrice M, chaque ligne représentant un élément de la base de facteur et chaque colonne étant un diviseur réduit friable R_i exprimé sur cette base : pour une colonne i, on a $R_i = \sum_k m_{ki} g_k$, où $M = (m_{ki})$.

De cette manière on recueille $w+1$ colonnes constituant une matrice $w \times (w+1)$. Ainsi le noyau de M est de dimension au moins 1. Par de l'algèbre linéaire, on trouve un vecteur non-nul de ce noyau, ce qui correspond à une combinaison linéaire non triviale des R_i : on a une famille (γ_i) telle que $\sum_i \gamma_i R_i = 0$. Revenant à l'expression des R_i en fonction de D_1 et D_2, on obtient : $\sum_i \gamma_i(\alpha_i D_1 + \beta_i D_2) = 0$, et donc

$$\lambda = -\frac{\sum_i \gamma_i \alpha_i}{\sum_i \gamma_i \beta_i}.$$

Le log discret est alors trouvé avec grande probabilité (le dénominateur est nul avec probabilité $1/N$).

Algorithme 11.13 Log discret dans les courbes hyperelliptiques
Entrée: Un diviseur D_1 d'une courbe de genre g sur $\mathbb{F}_q$, d'ordre premier $N = \mathrm{ord}(D_1)$, un diviseur $D_2 \in \langle D_1 \rangle$, et un paramètre r.
Sortie: Un entier λ tel que $D_2 = \lambda D_1$.

1. /* construction de la base de facteurs $\mathcal{F}$ */
   Pour chaque polynôme irréductible unitaire u_i sur $\mathbb{F}_q$ de degré 1, essayer de trouver v_i tel

que $\langle u_i, v_i \rangle$ soit un diviseur sur la courbe. S'il y a une solution, stocker $g_i = \langle u_i, v_i \rangle$ dans $\mathcal{F}$ (on ne met que l'un des deux diviseurs opposés dans la base).

2. /* Initialisation de la marche aléatoire */
   Pour j allant de 1 à r, choisir a_j et b_j aléatoirement dans $[1..N]$, et calculer $T_j \leftarrow a_j D_1 + b_j D_2$.
   Choisir α_0 et β_0 au hasard dans $[1..N]$ et calculer $R_0 \leftarrow \alpha_0 D_1 + \beta_0 D_2$.
   $k \leftarrow 1$.

3. /* Boucle principale */

   (a) /* Recherche d'un diviseur friable */
   Calculer $j \leftarrow \mathcal{H}(R_0)$, $R_0 \leftarrow R_0 + T_j$, $\alpha_0 \leftarrow \alpha_0 + a_j \mod N$, et $\beta_0 \leftarrow \beta_0 + b_j \mod N$.
   Répéter cette étape jusqu'à ce que $R_0 = \langle u_0(z), v_0(z) \rangle$ soit un diviseur friable.

   (b) /* Expression de R_0 sur la base $\mathcal{F}$ */
   Factoriser $u_0(z)$ sur $\mathbb{F}_q$, et déterminer les positions des facteurs dans la base $\mathcal{F}$.
   Stocker le résultat comme une colonne $R_k = \sum m_{ki} g_i$ d'une matrice $M = (m_{ki})$.
   Stocker les coefficients $\alpha_k = \alpha_0$ et $\beta_k = \beta_0$.
   Si $k < \#\mathcal{F} + 1$, alors soit $k \leftarrow k + 1$, et retourner à l'étape 3.a.

4. /* Algèbre linéaire */
   Calculer un vecteur non nul (γ_k) dans le noyau de la matrice M. Le calcul peut être effectué dans le corps $\mathbb{Z}/N\mathbb{Z}$.

5. /* Solution */
   Retourner $\lambda = -(\sum \alpha_k \gamma_k)/(\sum \beta_k \gamma_k) \mod N$. (Si le dénominateur est nul, retourner à l'étape 2.)

11.2.2 Points critiques de l'algorithme

Dans la première étape, il faut construire la base de facteurs, et pour cela, il faut trouver, s'il existe, un polynôme v correspondant à un polynôme u irréductible donné. Cela peut être converti en la résolution d'une équation de degré 2 sur $\mathbb{F}_q$, ce qui est faisable rapidement, quel que soit le corps fini.

L'initialisation de la marche aléatoire consiste en quelques opérations dans le groupe ; après cela, le calcul de chaque nouveau diviseur R_i ne coûte qu'une seule opération dans le groupe.

Un point crucial est le test de la friabilité d'un diviseur, c'est-à-dire de décider si un polynôme de degré g (le u du diviseur) est scindé sur $\mathbb{F}_q$. Une façon de faire est d'effectuer le début d'une factorisation de u, comme décrit en page 158 : en calculant $\gcd(X^q - X, u(X))$, on obtient le produit de tous les facteurs irréductibles de u de degré 1. Ainsi, si le degré de ce produit est égal au degré de u, cela prouve que u est scindé sur $\mathbb{F}_q$. La réciproque est vrai sous réserve que u soit sans facteurs carrés.

Lorsqu'un diviseur friable est détecté, la factorisation peut être terminée par les algorithmes probabilistes classiques, ou bien on peut essayer de diviser par tous les éléments de la base de facteurs (*trial division*).

L'algèbre linéaire est le point capital. La matrice obtenue est creuse, et il y a au plus g termes non nuls dans chaque colonne. Les algorithmes d'algèbre linéaire creuse comme l'algorithme de Lanczos [LO90] ou de Wiedemann [Wie86] peuvent donc être utilisés, de manière à obtenir une solution en temps quadratique en la taille de la matrice (au lieu de cubique par la méthode du pivot de Gauß).

De multiples autres optimisations peuvent être effectués en pratique. Elles seront décrites en section 11.3.

11.2.3 Analyse

Probabilité pour un diviseur d'être friable

La proposition suivante donne la proportion d'éléments friables et la probabilité de friabilité lors de la marche aléatoire. C'est le résultat principal pour l'analyse de complexité.

Proposition 11.1 *La proportion d'éléments friables dans la Jacobienne d'une courbe hyperelliptique de genre g sur $\mathbb{F}_q$ tend vers $1/g!$ quand q tend vers l'infini.*

Démonstration. Cette proposition repose sur la borne de Hasse-Weil : le nombre de points d'une courbe de genre g sur un corps fini à q éléments est égal à $q+1$ avec une erreur d'au plus $2g\sqrt{q}$, i.e. pour q tendant vers l'infini on peut négliger celle-ci. De plus la cardinalité de sa Jacobienne est q^g avec une erreur de l'ordre de $2gq^{g-\frac{1}{2}}$. Ici l'approximation est vraie lorsque q est suffisamment grand comparé à $4g^2$, ce qui est le cas à g fixé et q tendant vers l'infini.

D'après la définition de diviseur réduit pour les courbes hyperelliptiques, les diviseurs friables sont en nombre comparable avec les multiensembles de g points de la courbe. L'erreur introduite provient des points de ramification et est donc négligeable. On a donc environ $q^g/g!$ diviseurs friables et la proportion cherchée est donc $1/g!$. □

Complexité

La complexité de l'algorithme va être exponentielle en la taille de q, aussi allons nous compter le nombre d'opérations qui peuvent être effectuées en temps polynomial. Ces opérations sont de quatre sortes : on notera c_J le coût d'une opération de groupe dans la Jacobienne, c_q le coût d'une opération dans le corps de base, $c_{q,g}$ le coût d'une opération sur les polynômes de degré g définis sur $\mathbb{F}_q$ (y compris la factorisation), et c_N le coût d'une opération dans $\mathbb{Z}/N\mathbb{Z}$, où $N \approx q^g$ est l'ordre de la Jacobienne. On considère les différentes étapes de l'algorithme 11.13.

Étape 1. Pour construire la base de facteurs, on doit effectuer q résolutions d'une équation de degré 2 sur $\mathbb{F}_q$. Ainsi la complexité est $O(qc_q)$.

Étape 2. L'initialisation de la marche aléatoire nécessite un nombre polynomial d'opérations dans le groupe. On a donc $O((\log N)c_J)$ pour cette étape.

Étape 3. On doit répéter $\#\mathcal{F} = O(q)$ fois les étapes 3.a. et 3.b.
Étape 3.a. Le calcul d'un nouvel élément dans la marche aléatoire coûte une addition dans la Jacobienne et deux additions modulo N et le test de friabilité coûte la première étape de DDF. Par la proposition 11.1, on doit calculer $g!$ diviseurs en moyenne avant d'en trouver un qui soit friable et de sortir de l'étape 3.a. Ainsi le coût de cette étape est $O(g!(c_J + c_N + c_{q,g}))$.
Étape 3.b. La factorisation complète du polynôme, nécessaire pour exprimer le diviseur sur la base peut être faite de manière polynomiale probabiliste par l'algorithme de Berlekamp par exemple (cf [vzGG99, p. 365]). On a alors une complexité de $O(c_{q,g})$.

Ainsi la complexité de l'étape 3. est $O(qg!(c_J + c_n + c_{q,g})) + O(qc_{q,g})$.

Étape 4. Cette étape d'algèbre linéaire concerne une matrice de taille $O(q)$ et de poids $O(gq)$, les coefficients étant dans $\mathbb{Z}/N\mathbb{Z}$. Ainsi l'algorithme de Lanczos fournit une solution en temps $O(gq^2c_N)$.

Étape 5. Cette dernière étape nécessite $O(q)$ multiplications modulo N, et une inversion. D'où un coût $O(qc_N)$. Comme au chapitre précédent, on peut montrer que la probabilité d'échec est $1/N$.

Finalement, la complexité totale de l'algorithme est $O(g!qc_J) + O((g!q + gq^2)(c_n + c_{q,g})) + O(qc_q)$. Par l'algorithme de Cantor c_J est polynomial en $g \log q$, et les algorithmes classiques dans les corps finis et sur les polynômes donnent c_N polynomial en $N = g \log q$, c_q polynomial en $\log q$ et $c_{q,g}$ polynomial en $g \log q$. Ainsi toutes ces opérations peuvent être effectuées en temps borné par un polynôme en $g \log q$.

Théorème 11.1 *Sous l'hypothèse que la marche pseudo-aléatoire est purement aléatoire vis-à-vis de la friabilité, l'algorithme nécessite $O(q^2 + g!q)$ opérations de complexité polynomiale en $g \log q$ et si l'on considère le genre g fixé, le coût de l'algorithme est $O(q^2 \log^\gamma q)$.*

11.3 Quelques astuces pour accélérer les calculs

11.3.1 Recherche de relations

Parallélisation

La phase de construction de la matrice peut aisément se paralléliser : chaque processeur a sa propre marche aléatoire et envoie les colonnes qu'il a trouvées à un serveur central qui se chargera de l'algèbre linéaire. Le précalcul de la base de facteurs peut être fait par une machine puis le résultat envoyé à chaque noeud, ou bien, si le coût est négligeable, chaque processeur le refait. Ainsi, en utilisant M machines, on divise le coût de la recherche par M, ce qui est optimal.

Utilisation du théorème de Swan

Une optimisation intéressante a été suggérée par François Morain. Dans [Swa62], Swan donne un théorème qui relie la parité du nombre de facteurs irréductibles d'un polynôme sur un corps fini au fait que son discriminant soit un carré ou non dans le corps local correspondant.

Théorème 11.2 *Soit $F(x)$ un polynôme unitaire de degré n à coefficients entiers sur un corps $\mathfrak{p}$-adique K. Soit $f(x)$ le polynôme obtenu en réduisant les coefficients modulo $\mathfrak{p}$. Supposons que $f(x)$ est sans facteur carré et soit r le nombre de facteurs irréductibles de $f(x)$. Alors $r \equiv n \mod 2$ si et seulement si le discriminant de $F(x)$ est un carré dans K.*

Pour le test de friabilité, un premier calcul testant si le discriminant est un carré peut être effectué, et dans la moitié des cas cela suffit pour éliminer un polynôme non friable. Si ce premier test est positif, alors on effectue le test classique par DDF.

Cette technique est rentable si et seulement si le test de Swan coûte moins que la moitié du temps du test classique. En caractéristique impaire, c'est toujours le cas (lorsque q est assez grand), mais en caractéristique 2, l'estimation du temps de calcul est plus compliquée car certains calculs doivent être effectués dans une extension de $\mathbb{Z}/8\mathbb{Z}$ et aucune bibliothèque ne fournit de code optimisé pour ce type d'anneau. Notons que cette complication pour la caractéristique 2 n'est pas vraiment surprenante car dans le corps fini $\mathbb{F}_{2^n}$ tout élément est un résidu quadratique et il n'est pas évident de transcrire le théorème de Swan en un algorithme pratique.

Nous avons implanté cette technique de Swan ainsi que la parallèlisation. Le gain obtenu par Swan pour le test de friabilité est de l'ordre de 30 à 40% en caractéristique impaire, mais n'apporte rien en caractéristique 2.

Les autres améliorations proposées ci-dessous n'ont pas été implantées.

Addition d'un élément de poids faible

Lors des précalculs des décalages T_j servant à définir la fonction aléatoire, il est possible de construire des T_j aléatoires en grande quantité jusqu'à en trouver qui soient de poids inférieur à g. Ensuite, chaque pas dans la marche aléatoire consistant en l'ajout d'un des T_j sera accéléré, car l'opération dans la Jacobienne fera intervenir un diviseur de poids g (en général) et un diviseur de poids inférieur. On peut espérer ainsi réduire le nombre de réductions dans l'algorithme de Cantor.

Le nombre de T_j que l'on doit calculer en moyenne avant d'en trouver un de poids $g-1$ est de l'ordre de $O(q)$, ce qui n'est pas négligeable. Une manière de faire pour être sûr de gagner du temps par cette technique est de commencer l'algorithme avec une marche aléatoire quelconque, comme précédemment. Au fur et à mesure de la boucle principale, on teste la friabilité des diviseurs (et donc on commence à remplir la matrice), mais on vérifie aussi si l'on a un diviseur de poids inférieur. Dans l'affirmative, on remplace un des T_j de poids g par ce diviseur de poids moindre. On espère ainsi gagner quelques pourcents dans le temps de calcul. Bien entendu, il est alors complètement impossible de prouver que la marche aléatoire est encore suffisamment aléatoire.

Addition d'un élément de la base

Pour aller plus loin avec cette idée d'accélérer la marche aléatoire en ajoutant des éléments de petit poids, on peut essayer d'ajouter des éléments de la base, qui sont de poids 1.

Pour cela, on initialise la marche aléatoire de la manière habituelle, en précalculant des décalages. Ensuite, partant d'un élément R_i dans la marche aléatoire, s'il est non friable, on ajoute un élément g_j de la base de facteurs. On teste alors la friabilité de ce nouvel élément. S'il n'est pas friable, on ajoute de nouveau le *même* élément g_j, et ainsi de suite. On construit donc une suite

$$S_t = R_i + kg_j,$$

que l'on peut supposer se comporter comme une suite aléatoire vis-à-vis de la notion de friabilité. On espère donc trouver un indice t tel que S_t soit friable. Une colonne de la matrice peut alors être remplie : on a

$$S_t = \sum m_{ki} g_i = t g_j + \alpha_i D_1 + \beta_i D_2,$$

donc

$$\alpha_i D_1 + \beta_i D_2 = \sum m_{ki} g_i - t g_j.$$

Le gain de cette méthode est que dans la plupart des étapes on n'ajoute qu'un élément de poids 1 au diviseur courant, ce qui est bien plus rapide que d'ajouter un diviseur quelconque (spécialement si le genre devient assez grand). La contrepartie est que la matrice est un peu plus dense : au lieu d'avoir g termes par colonnes, on en a maintenant $g+1$, l'un d'eux étant de l'ordre de $g!$. Seules des expériences pratiques peuvent décider dans quel contexte cela est rentable.

Autres améliorations possibles

Quelques autres approches pourraient permettre de gagner encore sur le temps de calcul du remplissage de la matrice :

- Il est possible d'utiliser un crible de corps de fonctions comme décrit par Flassenberg et Paulus [FP99] afin de remplir notre matrice. De nombreuses expériences devront être menées afin de déterminer si ce crible peut être plus rapide que la marche aléatoire avec test de friabilité.

- La représentation «réelle» d'une courbe hyperelliptique pourrait être utilisée plutôt que l'imaginaire, car l'arithmétique peut y être accélérée dans certains contextes.
- Des variantes du type «large prime» sont envisageables lorsque le genre devient un peu trop grand pour garder la borne de friabilité égale à 1.

11.3.2 Algèbre linéaire

L'algèbre linéaire est une phase critique de l'algorithme car elle n'est pas facilement parallélisable et nécessite d'avoir toute la matrice disponible en mémoire centrale.

Élimination Gaussienne structurée

Avant d'exécuter l'algorithme de Lanczos, un prétraitement peut être effectué sur la matrice (voir [DW98] [Cav99]). Cette étape de filtrage (aussi appelée élimination Gaussienne structurée) consiste en les tâches suivantes :

- Éliminer les lignes vides.
- Éliminer les lignes ayant exactement un terme et la colonne correspondante.
- Si le nombre de colonnes est plus grand que le nombre de lignes plus 1, éliminer une colonne (choisie au hasard ou grâce à un critère heuristique).
- Essayer le début d'un pivot de Gauß, le pivot étant choisi de manière à minimiser l'augmentation de poids de la matrice. S'arrêter lorsque cela augmente le coût prévisionnel de l'algorithme de Lanczos.

Nous n'avons pas poussé à fond cette optimisation. Notre implantation du quatrième point n'est pas encore satisfaisante, et n'a pas été employée pour les résultats pratiques de la section suivante.

Il est possible d'évaluer le nombre moyen de lignes ayant un poids donné, ce qui donne une idée de ce que peut apporter l'élimination Gaussienne structurée.

Proposition 11.2 *Soit M une matrice creuse ayant w lignes et $w+1$ colonnes, avec exactement k termes dans chaque colonne. Alors le nombre moyen de lignes ayant exactement l termes est asymptotiquement*

$$n_l(k,w) = \frac{wk^l e^{-k}}{l!}.$$

Algorithme de Lanczos

Nous avons choisi l'algorithme de Lanczos de préférence à celui de Wiedemann car il ne nécessite que $2n$ produits de la matrice par un vecteur, alors que $3n$ sont nécessaires pour la technique de Wiedemann. Nous renvoyons à [LO90] pour une comparaison précise de ces deux algorithmes.

Le choix de représentation de la matrice en machine est capital : il faut éviter de perdre de l'espace en utilisant des pointeurs superflus et les données doivent être suffisamment contiguës pour minimiser les défauts de page. Aussi, il n'est plus possible à ce stade d'utiliser des langages de haut niveau.

Par ailleurs, il est possible de réduire un peu le coût des opérations dans le corps de base en effectuant une réduction paresseuse, comme expliqué dans [DW98].

Nous avons utilisé une représentation classique des éléments du corps fini. Pour plus d'efficacité, on pourrait employer la représentation de Montgomery.

Une amélioration capitale serait de paralléliser efficacement la phase d'algèbre linéaire. Les algorithmes de Lanczos et de Wiedemann par blocs fournissent des réponses théoriques partielles. Toutefois, une investigation plus poussée est nécessaire pour déterminer si cela s'avérera rentable (cf [Tho]).

11.3.3 Utilisation des automorphismes

Comme expliqué plus haut, la méthode Rho de Pollard peut être accélérée d'un facteur $\sqrt{m}$ s'il existe un automorphisme d'ordre m calculable rapidement sur la courbe. Pour notre algorithme, la présence d'un automorphisme est encore plus intéressante :

Proposition 11.3 *L'algorithme peut profiter de la présence d'un automorphisme d'ordre m comme suit : la recherche de relations est accélérée par un facteur m et l'algèbre linéaire d'un facteur m^2.*

De plus l'automorphisme n'a pas besoin d'être évalué aussi rapidement que pour la méthode Rho. Une évaluation en temps polynomial est suffisante.

L'idée est de garder dans la base de facteurs seulement un représentant pour chaque orbite sous l'action de l'automorphisme. Ainsi la taille de la base est réduite d'un facteur m, de sorte que le nombre de relations à construire est réduit du même facteur, et l'algèbre linéaire d'un facteur m^2.

On suppose que la Jacobienne est cyclique d'ordre premier $N = \mathrm{ord}(D_1)$, et l'on note σ l'automorphisme d'ordre m sur $\mathcal{C}$ que l'on étend par linéarité à $Jac(\mathcal{C})$. Alors $\sigma(D_1)$ appartient à $Jac(\mathcal{C}) = \langle D_1 \rangle$, et il existe un entier η tel que $\sigma(D_1) = \eta D_1$. De plus, σ étant un automorphisme de groupe, pour tout $D \in Jac(\mathcal{C})$, $D = kD_1$ et on a $\sigma(D) = \sigma(kD_1) = k\sigma(D_1) = k\eta D_1 = \eta D$.

Supposons maintenant que l'on n'a gardé dans la base qu'un seul élément pour chaque orbite. Soit $R = P_1 + P_2 + \cdots + P_k = \alpha D_1 + \beta D_2$ la décomposition d'un diviseur friable en diviseurs premiers de degré 1. Pour chaque i, il existe une puissance de σ telle que le diviseur premier P_i est égal à $\sigma^{l_i}(g_i)$, où g_i est un élément de la base de facteurs réduite. Alors on peut écrire $R = \eta^{l_1}(g_1) + \cdots + \eta^{l_k}(g_k)$, et l'on a une relation dans une matrice de taille m fois moindre.

Pour le cas général où la Jacobienne n'est pas cyclique et où l'on travaille dans un sous-groupe d'ordre N, on doit travailler quelque peu pour justifier les calculs, mais en pratique on fait essentiellement la même chose.

Proposition 11.4 *Supposons que l'ordre N de D_1 est premier et que N^2 ne divise pas la cardinalité de la Jacobienne. Alors toutes les opérations sur les coefficients de la matrice peuvent être faites dans le corps $\mathbb{Z}/N\mathbb{Z}$. (Même si certains éléments de la base de facteurs ne sont pas d'ordre N).*

Démonstration. Soit $\#\mathrm{Jac}(\mathcal{C}) = NN_2$, avec N_2 premier à N. La matrice $M = (m_{ki})$ est définie par $R_k = \sum_i m_{ki} g_i$ pour tout k. Supposons que l'on a résolu l'algèbre linéaire modulo N. Alors soit $(\gamma_1, \ldots, \gamma_{w+1})$ un vecteur non nul du noyau de M modulo N, i.e. pour tout i on a

$\sum_k \gamma_k m_{ki} \equiv 0 \mod N$. Les γ_k sont à valeurs dans $\mathbb{Z}/N\mathbb{Z}$, et on les considère comme des entiers entre 0 et $N-1$. Alors

$$\sum_k \gamma_k R_k = \sum_k \gamma_k \sum_i m_{ki} g_i = \sum_i \left(\sum_k \gamma_k m_{ki} \right) g_i.$$

En multipliant les deux membres de cette équation par N_2, on obtient

$$N_2 (\sum_k \gamma_k R_k) = \sum_i N_2 (\sum_k \gamma_k m_{ki}) g_i.$$

Pour tout i, le coefficient devant g_i est nul modulo $N_2 N$, qui est l'ordre de la Jacobienne, ainsi le membre de droite est nul. On en déduit que $\sum_k \gamma_k R_k$ est un diviseur d'ordre divisant N_2. Cependant, tous les R_k appartiennent au sous-groupe engendré par D_1, et sont donc d'ordre N. Ainsi $\sum_k \gamma_k R_k = 0$ dans $\mathrm{Jac}(\mathcal{C})$, et la solution modulo N donne une solution à notre problème. □

Réduction de la base

Dans le cas où le genre est très petit, les deux phases sont déséquilibrées : la recherche de relations coûte beaucoup moins que l'algèbre linéaire. Une stratégie consiste alors à réduire artificiellement la base de facteurs : on ne garde qu'une proportion arbitraire des diviseurs. Supposons que l'on a ainsi réduit la base d'un facteur n. Alors la probabilité d'obtenir une relation lors de la marche aléatoire est réduite d'un facteur n^g et le coût de la construction de la matrice augmente d'un facteur n^{g-1}. Par contre l'algèbre linéaire est accélérée d'un facteur n^2. On voit donc tout de suite que ceci ne sera utile que si le genre est suffisamment petit.

Cependant, dans ce contexte, R. Harley nous a fait remarquer que cela peut tout de même réduire la complexité (heuristique) de l'algorithme total. En effet, si l'on équilibre les deux phases en choisissant n de l'ordre de $q^{\frac{1}{g+1}}$, on obtient un temps de calcul en $O(q^{\frac{2g}{g+1}})$.

11.4 Résultats pratiques

Nous avons implanté la recherche de relations dans le système de calcul formel Magma [BC97]. Pour l'algèbre linéaire, nous avons utilisé le langage C, avec la bibliothèque ZEN [CL98] pour les étapes non critiques (i.e. les opérations qui sont effectuées un nombre linéaire de fois), et pour les autres (i.e. les opérations dans le produit matrice-vecteur et les produits scalaires), nous avons utilisé des appels directs à des routines assembleur des bibliothèques GMP [Gra96] et BigNum [HSV89]. En effet une représentation compacte de la matrice ne permet pas d'utiliser ZEN sans surcoût notable.

Le premier exemple est un cryptosystème proposé par Buhler et Koblitz [BK98]. Nous avons pris les valeurs recommandées par Koblitz dans son livre [Kob98], i.e. nous avons travaillé sur la courbe $y^2 + y = x^{13}$, sur un corps fini premier d'ordre p plus grand que $5,000,000$, et $p \equiv 1 \mod 13$. Cette courbe a un automorphisme d'ordre 13 provenant de la multiplication complexe, ce qui aide le calcul de l'ordre de la Jacobienne, mais aide aussi notre attaque.

Le tableau suivant donne les informations concernant cette courbe.

corps	$\mathbb{F}_{5026243}$
équation	$y^2 + y = x^{13}$
genre	6
$\#J$	$13^3 \times 7345240503856807663632202049344834001 \approx 10^{40}$

Dans le tableau suivant nous donnons les temps de calcul pour un calcul de log discret. Ces temps ont été obtenus sur un Pentium II 450 MHz avec 128 Mo. Durant la phase d'algèbre linéaire (l'étape nécessitant le plus de mémoire), l'espace utilisé était environ 60Mo.

cardinal de la base de facteurs	$193,485$
temps pour construire la base	1638 sec
nombre de pas aléatoires	$201,426,284$
nombre de « early abort » par Swan	$100,721,873$
nombre de relations calculées	$281,200$
proportion d'éléments friables ($g!$)	716.3 (720)
temps total pour recueillir les relations	$513,870$ sec = **6 jours**
temps pour écrire les relations sur la base	$8,822$ sec
temps pour prétraiter la matrice	1218 sec
taille de la matrice	$165,778 \times 165,779$
temps total pour Lanczos	$780,268$ sec = **9 jours**

Notre algorithme ne dépend pas de la caractéristique du corps. Nous avons testé notre implantation sur une courbe de genre 6 définie sur $\mathbb{F}_{2^{23}}$. Cette courbe a été obtenue en étendant les scalaires à partir d'une courbe sur $\mathbb{F}_2$. C'est pourquoi l'automorphisme de Frobenius peut être utilisé pour accélérer l'attaque. La taille de la Jacobienne est environ 10^{41}. Une telle courbe n'est pas attaquable par les variantes parallèles de la méthodes Rho ; en effet, même en utilisant les automophismes, environ 2^{63} opérations dans la Jacobienne seraient nécessaires.

Nous donnons les mêmes indications que pour la courbe précédente.

corps	$\mathbb{F}_{2^{23}}$
équation	$y^2 + (x+1)y = x^{13} + x^{11} + x^8 + x^7 + x^5 + x^4 + x + 1$
genre	6
$\#J$	$2^3 \times 7 \times 622571845211703438355012489904899949517 7 \approx 10^{41}$

cardinal de la base de facteurs	$182,462$
temps pour construire la base	6575 sec
nombre de pas aléatoires	$165,732,450$
nombre de relations calculées	$231,000$
proportion d'éléments friables ($g!$)	717.5 (720)
temps total pour recueillir les relations	$797,073$ sec = **9 jours**
temps pour écrire les relations sur la base	$12,057$ sec
temps pour prétraiter la matrice	880 sec
taille de la matrice	$162,873 \times 162,874$
temps total pour Lanczos	$1,038,534$ sec = **12 jours**

11.5 Conséquences cryptographiques

Notre algorithme est relativement simple à implanter et s'analyse bien pour un genre fixé pas trop grand. Le temps de calcul est de l'ordre de $O(q^2)$, que l'on peut comparer à $O(q^{g/2})$ pour la méthode Rho. Ainsi, à partir du genre 5, les cryptosystèmes peuvent être attaqués plus efficacement. Le tableau suivant donne les complexités correspondantes :

g	1	2	3	4	5	6	7
Rho	$q^{1/2}$	q	$q^{3/2}$	q^2	$q^{5/2}$	q^3	$q^{7/2}$
Index	q^2	q^2	q^2	q^2	q^2	q^2	q^2

Au vu des expériences menées, cet algorithme est efficace en pratique. Ainsi il ne semble pas pertinent d'utiliser de cryptosystème hyperelliptique de genre autre que 2, 3 ou 4, car pour un genre supérieur on est obligé d'augmenter la taille de la clef afin de garantir un niveau de sécurité donné.

Le cas particulier du genre 4 est intéressant : la première analyse donne une complexité équivalente à celle de la méthode Rho. Toutefois, l'emploi de la réduction de la base donne une complexité heuristique de $O(q^{8/5})$ qui est bien meilleure que $O(q^2)$. Un exemple d'un tel calcul a été effectué dans le cadre de la descente de Weil et sera commenté au chapitre 13.

Chapitre 12

Calcul d'index en genre 2

La résolution du problème du logarithme discret par un calcul d'index dans les Jacobiennes de courbes fonctionne bien quand le genre est grand et l'étude du chapitre précédent montre que la limite se situe autour de $g = 4$. Toutefois dans le cas particulier du genre 2, il est possible de ramener la complexité du calcul d'index au même niveau que les meilleurs algorithmes connus.

Dans ce chapitre, nous allons étudier cet algorithme spécifique au genre 2. L'intérêt de celui-ci est plutôt théorique que pratique. En effet l'espace mémoire requis est énorme : du même ordre que pour la méthode de Shanks. Néanmoins la constante peut être un petit peu meilleure que celles des méthodes génériques, et il est intéressant de constater que le calcul d'index peut s'appliquer au genre 2 avec une bonne complexité.

12.1 Description de l'algorithme

12.1.1 Principe

Le principe de l'algorithme est un calcul d'index à la manière du chapitre précédent, mais au lieu de stocker les informations recueillies dans une matrice, une structure de donnée performante est utilisée afin de pouvoir effectuer la recherche d'un élément du noyau en temps *linéaire*. Il n'est guère surprenant qu'une telle technique existe : cela revient à éliminer efficacement dans une matrice qui ne comporte que deux entrées par ligne. Nous allons abandonner la terminologie de l'algèbre linéaire et préférer celle des graphes. En effet, il va s'avérer qu'en moyenne on trouve la solution bien avant d'avoir une matrice ayant une ligne de plus que de colonnes et l'analyse est facilitée par l'utilisation du point de vue « graphes ».

Soit $\mathcal{C}$ une courbe hyperelliptique de genre 2 sur un corps fini $\mathbb{F}_q$. Soient D_1 un élément de $\mathrm{Jac}(\mathcal{C})$ d'ordre premier N, et soit D_2 un élément du sous-groupe engendré par D_1 dont on cherche le log discret. Le cas où l'ordre n'est pas premier fonctionne facilement de manière identique, mais les statistiques concernant les constantes semblent assez perturbées dès que l'ordre est composé.

Soit G le graphe dont les sommets sont les points de $\mathcal{C}$ à coordonnées dans $\mathbb{F}_q$, et initialement sans arêtes. Après avoir initialisé une marche pseudo-aléatoire dans le sous-groupe engendré par D_1, on obtient une suite d'éléments

$$R_i = \alpha_i D_1 + \beta_i D_2,$$

parmi lesquels ne sont conservés que les diviseurs friables qui, ici, sont des sommes de deux points rationnels sur la courbe. Pour chaque élément friable trouvé, une arête est ajoutée dans le graphe G entre les deux points P_1 et P_2 tels que $R_i = P_1 + P_2 - 2\infty$. Cette arête est étiquetée par le couple (α_i, β_i) de manière à garder un lien avec le problème de départ.

Le graphe G a $O(q)$ sommets, donc au pire, au bout de $O(q)$ arêtes ajoutées, un cycle apparaît. Si le cycle est formé d'un nombre *pair* d'arêtes, alors un peu retrouver le logarithme discret cherché avec grande probabilité. En effet, pour i allant de 1 à $2k$ notons $S_i = P_i+P_{i+1}-2\infty = \alpha_i D_1+\beta_i D_2$ les diviseurs correspondant aux arêtes d'un cycle de longueur $2k$, avec $P_{k+1} = P_1$. Alors en prenant la somme alternée, on obtient

$$\sum_{1\leqslant i\leqslant 2k} (-1)^i S_i = 0 = \left(\sum_{1\leqslant i\leqslant 2k} \alpha_i\right) D_1 + \left(\sum_{1\leqslant i\leqslant 2k} \beta_i\right) D_2.$$

Et la solution en découle pourvu que le coefficient devant D_2 soit inversible modulo N :

$$\log_{D_1}(D_2) = -\frac{\sum_{1\leqslant i\leqslant 2k} \alpha_i}{\sum_{1\leqslant i\leqslant 2k} \beta_i}.$$

Il est nécessaire d'avoir un cycle d'ordre pair sinon il est impossible de trouver une combinaison linéaire des S_i qui s'annule. En effet la matrice correspondante est

$$\begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \end{pmatrix},$$

dont le déterminant vaut 0 si la taille est paire, et 2 sinon. Dans ce dernier cas, la matrice est inversible et le seul moyen d'obtenir une combinaison linéaire nulle des lignes est la combinaison triviale.

12.1.2 Algorithme Union–Find

La littérature contient de nombreux algorithmes efficaces pour résoudre divers problèmes dans les graphes. Pour notre calcul de log discret, nous désirons insérer des arêtes dans un graphe et détecter le premier cycle qui se produit, tout cela le plus rapidement possible, et en gardant suffisamment d'informations pour pouvoir reparcourir le cycle afin de reconstruire la solution.

L'algorithme *Union–Find* que l'on peut trouver par exemple décrit dans [Sed88] pages 441–449, peut être adapté pour remplir cette tâche. À l'origine, son rôle est de maintenir une structure où l'on rajoute dynamiquement des arêtes, en pouvant à tout moment détecter si deux sommets sont dans la même composante connexe.

Les sommets du graphe sont numérotés de 1 à m et à l'initialisation un tableau T de taille m est alloué et rempli avec des zéros. Dans cette structure, les propriétés qui vont être conservées sont les suivantes :

- Si $T[x]$ est un entier positif, alors il existe un chemin dans le graphe entre le sommet x et le sommet $T[x]$,
- Si $T[x]$ est nul, alors le sommet x est isolé dans le graphe,
- Si $T[x]$ est un entier négatif, alors la composante connexe du sommet x contient $-T[x]+1$ éléments,
- Pour tout x, en itérant $x :\ = T[x]$ tant que $T[x]$ est positif, on obtient une suite finie se terminant sur un $T[x]$ négatif.

De manière plus visuelle, on maintient à jour une forêt. Chaque arbre représente une composante connexe dans laquelle chaque élément x pointe sur son «père» $T[x]$ qui pointe sur son «grand-père» $T[T[x]]$, et ainsi de suite jusqu'à ce que l'on atteigne l'aïeul commun $\mathcal{A}(x)$ de la composante connexe. Pour cet aïeul, $T[\mathcal{A}(x)]$ contient l'opposé du nombre d'éléments autre que lui-même.

Lorsqu'une nouvelle arête entre x et y se présente, il est aisé de calculer l'aïeul $\mathcal{A}(x)$ de x et celui $\mathcal{A}(y)$ de y en remontant dans les arbres à l'aide du tableau T. Si $\mathcal{A}(x) = \mathcal{A}(y)$, alors l'arête que l'on veut insérer va créer un cycle car x et y sont déjà reliés par un chemin. Le test du premier cycle est donc très simple.

Dans le cas où l'arête ne crée pas de cycle, il faut insérer cette nouvelle information tout en maintenant la structure. Le but est de fusionner deux arbres pour en faire un seul (c'est le *union* du nom de l'algorithme). La première idée qui vient à l'esprit est de mettre à jour la case $\mathcal{A}(x)$ du tableau T de manière à ce que l'arbre sous $\mathcal{A}(x)$ devienne un sous-arbre de $\mathcal{A}(y)$. Mettre à jour les cardinalités n'est pas difficile.

Pour gagner en efficacité par rapport à cette première approche, deux techniques peuvent être ajoutées :

1. L'équilibrage de poids,
2. La compression de chemins.

L'équilibrage de poids consiste simplement à choisir lequel des deux arbres deviendra un sous-arbre de l'autre en fonction de leur poids respectif. L'heuristique choisie est que l'arbre le moins lourd devienne un sous-arbre du plus lourd.

La compression de chemins permet quant à elle de maintenir une profondeur très faible dans les arbres sans pour autant les parcourir en entier. L'idée est qu'après avoir détecté l'aïeul de x, on peut refaire une passe sur le chemin qui mène de x à $\mathcal{A}(x)$ en faisant pointer tous les éléments (dont x) directement sur $\mathcal{A}(x)$.

Tarjan a prouvé que ces deux techniques rendent l'algorithme très efficace.

Théorème 12.1 *L'algorithme Union–Find avec équilibrage de poids et compression de chemins nécessite $O(E\alpha(E))$ opérations pour construire une structure à E arêtes, où α désigne l'inverse de la fonction d'Ackermann.*

La fonction α est une fonction qui tend vers l'infini, mais tellement lentement que pour toute application pratique elle apparaît comme une constante inférieure à 4.

12.1.3 Détails de l'algorithme

L'ossature de l'algorithme est la suivante :

Algorithme 12.14 Log discret par Union-Find
Entrée: Une courbe $\mathcal{C}$ de genre 2 sur $\mathbb{F}_q$, un élément D_1 d'ordre premier N dans $\mathrm{Jac}(\mathcal{C})$, et un élément D_2 dans $\langle D_1 \rangle$.
Sortie: Le logarithme discret de D_2 en base D_1.

1. Initialiser la marche pseudo-aléatoire.
2. Initialiser le tableau de Union–Find.

3. Tant que le log discret n'est pas trouvé faire
   (a) Itérer la marche pseudo-aléatoire jusqu'à trouver un diviseur $D = P_1+P_2 = \alpha D_1+\beta D_2$ friable.
   (b) Appeler la routine Union-Find pour stocker l'arête correspondante.
   (c) Si l'arête ferme un cycle, essayer de de calculer le log discret, et le retourner en cas de succès.

La construction d'une marche pseudo-aléatoire est détaillée au chapitre 9, nous nous concentrerons donc sur la façon adéquate d'implanter Union–Find.

Plutôt que de mettre tous les points de la courbe dans le graphe, on peut quotienter par l'involution hyperelliptique, et ne stocker que l'un des deux points qui partagent la même abscisse. Ceci permet de réduire la taille du graphe d'un facteur 2 et de faciliter la numérotation des sommets. On suppose que l'on a un moyen d'envoyer $\mathbb{F}_q$ sur l'intervalle $[1, q]$, ce qui est particulièrement aisé dans le cas des corps premiers et de caractéristique 2. Alors la case x du tableau représente celui des deux points (x, y) et (x, y') dont l'ordonnée est la plus petite. Pour un entier $x \in [1, q]$, on notera P_x ce point de la courbe.

Le tableau ne sera pas un simple tableau d'entiers comme dans le Union-Find générique, mais un tableau dont chaque case contient un quadruplet $(y,\ \alpha,\ \beta,\ \varepsilon)$, où y est un entier qui représente l'indice du père dans le tableau, α et β sont des entiers modulo N et ε est un entier dans $\{-1, 1\}$. Plus précisément, si $T[x]$ contient $(y,\ \alpha,\ \beta,\ \varepsilon)$, cela signifie que

$$P_x + \varepsilon P_y - 2\infty = \alpha D_1 + \beta D_2, \tag{12.1}$$

et donc que P_x et P_y sont liés par une arête. Le signe ε est la contrepartie à payer pour avoir divisé la taille du graphe par deux.

Il s'agit maintenant de voir comment les différentes partie de Union–Find s'adaptent de manière à conserver la propriété ci-dessus. Supposons que la marche aléatoire vient de nous fournir un diviseur friable

$$P_1 + P_2 = \alpha D_1 + \beta D_2.$$

Désormais le point à l'infini est sous-entendu dans l'écriture de telles expressions. On cherche alors le diviseur P_x de même abscisse que P_1 et P_y de même abscisse que P_2. Soient ε_x et ε_y les entiers de $\{-1, 1\}$ tels que

$$\varepsilon_x P_x + \varepsilon_y P_y = \alpha D_1 + \beta D_2.$$

On suppose que P_x et P_y sont distincts, ce qui est équivalent à supposer que P_1 et P_2 sont distincts. Si ce n'est pas le cas, on n'utilise pas ce diviseur, et on reprend la marche aléatoire.

Recherche des aïeux et compression de chemins

La recherche de l'aïeul se fait classiquement en remontant dans l'arbre. On garde au passage la liste des éléments rencontrés sur le chemin. On reparcourt alors cette liste en sens inverse en faisant pointer tous les éléments directement sur l'aïeul et en maintenant un étiquetage valide.

Algorithme 12.15 AieulCompresse
Entrée: Un entier $x \in [1, q]$, et le tableau T.
Sortie: L'aïeul $\mathcal{A}(x)$ de x.
Effet de bord : le tableau T compressé, en maintenant la règle 12.1.

1. $C \leftarrow$ une liste contenant initialement x ; $X \leftarrow x$;

2. */* Recherche de l'aïeul */*
   TANT QUE $T[X][1] > 0$ FAIRE
   - $X \leftarrow T[X][1]$;
   - Rajouter X à la fin de la liste C ;
3. $\ell \leftarrow$ longueur (C) ;
4. */* Compression du chemin */*
   SI $\ell \geqslant 3$ ALORS
   - $\alpha \leftarrow T[C[\ell-1]][2]$; $\beta \leftarrow T[C[\ell-1]][3]$; $\varepsilon \leftarrow T[C[\ell-1]][4]$;
   - POUR i allant de $\ell-2$ à 1 FAIRE
     - $\alpha \leftarrow T[C[i]][2] - T[C[i]][4]\alpha \mod n$;
     - $\beta \leftarrow T[C[i]][3] - T[C[i]][4]\beta \mod n$;
     - $\varepsilon \leftarrow -T[C[i]][4]\varepsilon$;
     - $T[C[i]] \leftarrow [X, \alpha, \beta, \varepsilon]$;
5. RETOURNER X.

Dans la fonction UNION-FIND, cette fonction AIEULCOMPRESSE va être utilisée deux fois de suite : d'abord pour x, puis pour y. Dans la suite, on stockera ces résultats dans les variables $X \leftarrow$ AIEULCOMPRESSE(x), et $Y \leftarrow$ AIEULCOMPRESSE(y).

Remarquons qu'on doit faire la première compression *avant* de calculer l'aïeul du deuxième élément, car sinon la compression du premier chemin risque de modifier le chemin que l'on aura stocké lors de la deuxième recherche d'aïeul.

Par la suite, l'algorithme diffère selon que X et Y sont égaux, auquel cas on a un cycle qu'il faut analyser, ou non, et alors il faut ajouter l'arête entre les deux arbres.

Un aïeul commun : tentative de calcul du log discret

La compression de chemin a remonté x et y à au plus une case de leur aïeul commun X. Ainsi le cycle que l'on doit étudier a au plus trois sommets. La distinction pair-impair n'est plus valable depuis que l'on a regroupé deux points dans un sommet. La condition de validité d'un cycle se transcrit toutefois facilement en une condition sur les ε qui interviennent.

Supposons que le tableau T contient les informations suivantes :

$$P_x + \varepsilon_1 P_X = \alpha_1 D_1 + \beta_1 D_2 \quad \text{et} \quad P_y + \varepsilon_2 P_X = \alpha_2 D_1 + \beta_2 D_2.$$

Lorsque l'on rajoute l'arête

$$\varepsilon_x P_x + \varepsilon_y P_y = \alpha D_1 + \beta D_2,$$

on peut éliminer de manière à obtenir

$$(\varepsilon_1 \varepsilon_x + \varepsilon_2 \varepsilon_y) P_X = (\varepsilon_x \alpha_1 + \varepsilon_y \alpha_2 - \alpha) D_1 + (\varepsilon_x \beta_1 + \varepsilon_y \beta_2 - \beta) D_2.$$

Ce qui fournit une relation entre D_1 et D_2 (et donc le log discret avec grande probabilité) à condition que $\varepsilon_1 \varepsilon_x + \varepsilon_2 \varepsilon_y$ soit nul, ce qui se produit avec une chance sur deux. L'algorithme est donc le suivant :

Algorithme 12.16 TESTCYCLE
Entrée: Deux entiers x et y dans $[1, q]$, deux entiers ε_x et ε_y dans $\{-1, 1\}$, α et β tels que $\varepsilon_x P_x + \varepsilon_y P_y = \alpha D_1 + \beta D_2$, le tableau T, et le père commun X de x et y.
Sortie: Le logarithme discret ou -1 si le calcul a échoué.

1. SI $x = X$ ALORS $\varepsilon_1 \leftarrow -1$; $\alpha_1 \leftarrow 0$; $\beta_1 \leftarrow 0$;
   SINON $\varepsilon_1 \leftarrow T[x][4]$; $\alpha_1 \leftarrow T[x][2]$; $\beta_1 \leftarrow T[x][3]$;
2. SI $y = X$ ALORS $\varepsilon_2 \leftarrow -1$; $\alpha_2 \leftarrow 0$; $\beta_2 \leftarrow 0$;
   SINON $\varepsilon_2 \leftarrow T[y][4]$; $\alpha_2 \leftarrow T[y][2]$; $\beta_2 \leftarrow T[y][3]$;
3. SI $\varepsilon_x \varepsilon_1 + \varepsilon_y \varepsilon_2 = 0$ ALORS
   - $\lambda \leftarrow \varepsilon_x \beta_1 + \varepsilon_y \beta_2 - \beta \mod n$;
   - SI λ n'est pas inversible modulo N ALORS RETOURNER -1 ;
   - $\mu \leftarrow \varepsilon_x \alpha_1 + \varepsilon_y \alpha_2 - \alpha \mod n$;
   - RETOURNER $-\mu/\lambda \mod n$;
4. RETOURNER -1 ;

Deux aïeux disjoints : union des deux arbres

L'union des deux arbres ne présente pas de difficulté, si ce n'est qu'il faut prendre garde à bien maintenir à jour les étiquettes.

Algorithme 12.17 UNIONARBRES
Entrée: Deux entiers x et y dans $[1, q]$, deux entiers ε_x et ε_y dans $\{-1, 1\}$, α et β tels que $\varepsilon_x P_x + \varepsilon_y P_y = \alpha D_1 + \beta D_2$, le tableau T, et le père X de x et le père y de Y.
Sortie: Aucune.
Effet de bord : Mise à jour du tableau T, en maintenant la règle 12.1.

1. SI $x = X$ ALORS $\varepsilon_1 \leftarrow -1$; $\alpha_1 \leftarrow 0$; $\beta_1 \leftarrow 0$;
   SINON $\varepsilon_1 \leftarrow T[x][4]$; $\alpha_1 \leftarrow T[x][2]$; $\beta_1 \leftarrow T[x][3]$;
2. SI $y = Y$ ALORS $\varepsilon_2 \leftarrow -1$; $\alpha_2 \leftarrow 0$; $\beta_2 \leftarrow 0$;
   SINON $\varepsilon_2 \leftarrow T[y][4]$; $\alpha_2 \leftarrow T[y][2]$; $\beta_2 \leftarrow T[y][3]$;
3. $\varepsilon_X \leftarrow \varepsilon_1 \varepsilon_x$; $\varepsilon_Y \leftarrow \varepsilon_2 \varepsilon_y$;
4. $\alpha_{XY} \leftarrow \varepsilon_x \alpha_1 + \varepsilon_y \alpha_2 - \alpha$; $\beta_{XY} \leftarrow \varepsilon_x \beta_1 + \varepsilon_y \beta_2 - \beta$;
5. SI $T[Y][1] < T[X][1]$ ALORS
   - $T[Y][1] \leftarrow T[Y][1] + T[X][1] - 1$;
   - $T[X] \leftarrow [Y,\ \varepsilon_X \alpha_{XY} \mod n,\ \varepsilon_X \beta_{XY} \mod n,\ \varepsilon_X \varepsilon_Y]$;

   SINON
   - $T[X][1] \leftarrow T[X][1] + T[Y][1] - 1$;
   - $T[Y] \leftarrow [X,\ \varepsilon_Y \alpha_{XY} \mod n,\ \varepsilon_Y \beta_{XY} \mod n,\ \varepsilon_X \varepsilon_Y]$;

12.2 Analyse

12.2.1 Modèle employé, et hypothèses

Pour analyser l'algorithme précédent, on fait l'hypothèse usuelle que la marche pseudo-aléatoire est vraiment aléatoire. Nous supposerons de plus que la Jacobienne est cyclique d'ordre N premier, et donc que le sous-groupe engendré par D_1 est la Jacobienne tout entière. Avec ces hypothèses, lorsque l'on tombe sur un diviseur friable, il est tiré au sort de manière uniforme et indépendante parmi tous les diviseurs friables de $\mathrm{Jac}(\mathcal{C})$. De plus, on n'utilise pas les cas où le diviseur friable est un point double et où le diviseur n'est constitué que d'un point.

Ainsi, le modèle de graphe aléatoire correspondant est un graphe où l'on tire aléatoirement de manière uniforme et indépendante des arêtes entre des points *distincts*.

12.2.2 Complexité en moyenne

Lemme 12.1 *Dans une Jacobienne de courbe de genre 2 sur $\mathbb{F}_q$, le nombre d'éléments friables de la forme $P_1 + P_2 - 2\infty$ avec P_1 différent de P_2 est $\frac{q^2}{2} + O(q^{3/2})$. Le nombre total d'éléments est $q^2 + O(q^{3/2})$.*

Ce lemme découle directement de la borne de Hasse-Weil et nous dit qu'en moyenne, dans notre marche aléatoire, on tombe une fois sur deux sur un diviseur qui correspond à une arête du graphe.

Lemme 12.2 *Lorsqu'un cycle se forme, la probabilité qu'il fournisse le logarithme discret est $\frac{1}{2}(1 - \frac{1}{N})$.*

Démonstration. La condition de « parité » sur le graphe est convertie à la section précédente en une condition sur les ε. Pour que le logarithme discret puisse être trouvé, il faut et il suffit que

1. L'expression $\varepsilon_1\varepsilon_x + \varepsilon_2\varepsilon_y$ soit nulle,

2. La valeur de $\lambda = \varepsilon_x\beta_1 + \varepsilon_y\beta_2 - \beta$ soit inversible modulo N.

La première condition se produit avec probabilité $1/2$. En effet, ε_1 et ε_2 sont des variables aléatoires qui ne dépendent que de x, de y et de l'arbre. Les variables aléatoires ε_x et ε_y sont donc des variables aléatoires indépendantes de ces premières données, et la probabilité d'obtenir l'annulation est $1/2$.

Ensuite, pour montrer que λ est inversible modulo N, on utilise le même type d'argument, avec β aléatoire indépendant des autres variables. (cf aussi l'argumentation de la page 157). □

La partie délicate est d'analyser au bout de combien de temps en moyenne un cycle se produit, puis de combiner avec le lemme précédent pour en déduire le temps moyen avant la découverte du logarithme discret.

L'analyse exacte n'a pas été faite. Toutefois, en se fondant sur des travaux de Flajolet, Knuth et Pittel, on peut obtenir une analyse heuristique qui fournit des résultats assez proches de ceux observés lors d'expériences.

Dans [FKP89], les auteurs font une analyse en moyenne du temps d'attente avant la création de la première composante cyclique dans des graphes aléatoires suivant deux modèles (cet article contient aussi des analyses en moyenne de beaucoup de caractéristiques de cette composante). Les modèles sont les suivants :

- le *modèle uniforme* où à chaque étape on tire uniformément et indépendament deux sommets x et y. L'arête non-orientée (x, y) est alors ajoutée au graphe. Si $x = y$, ou si l'arête existe déjà, on considère que l'on a trouvé un cycle (de longueur 1 ou 2).
- le *modèle permuté* où l'on tire uniformément et indépendament parmi les arêtes (x, y) avec $x \neq y$ et (x, y) qui n'existe pas encore dans le graphe.

Le théorème est le suivant :

Théorème 12.2 (Flajolet, Knuth, Pittel) *Le nombre moyen d'arêtes quand un graphe aléatoire à q sommets atteint sa première composante cyclique est $\frac{1}{3}q + O(q^{5/6})$ dans le modèle uniforme et $\frac{1}{2}(1 - \hat{p}_3)q + O(q^{5/6})$ dans le modèle permuté, où $\hat{p}_3$ est une constante proche de* 0.1216.

Malheureusement, le modèle uniforme ne convient pas, car pour le log discret, une arête du type (x, x) ne produit jamais un cycle fournissant le log discret ; et le modèle permuté ne convient pas car il rejette les répétitions d'arêtes, alors que celles-ci donne le log discret avec probabilité $1 - 1/N$. Le modèle qui nous intéresse doit donc se situer entre les deux, et le temps d'attente avant le premier cycle est vraisemblablement inclus entre les temps d'attente des deux modèles.

Ensuite, vient se rajouter le problème lié au fait que le premier cycle ne suffit pas toujours pour conclure. Toutefois, heuristiquement, une fois que l'on a atteint un stade où le premier cycle est « mûr » pour apparaître, le second ne doit pas suivre de loin car la forêt est déjà bien remplie.

En combinant cela avec le théorème de Tarjan qui donne le coût de la gestion de la structure Union–Find, on trouve un coût heuristique :

Conjecture 12.1 *Soit $\mathcal{C}$ une courbe de genre 2 sur $\mathbb{F}_q$ dont la Jacobienne est cyclique d'ordre premier. Sous des hypothèses heuristiques raisonnables, le coût moyen de l'algorithme de log discret dans* $\mathrm{Jac}(\mathcal{C})$ *est $2\kappa q$ opérations de groupe dans la Jacobienne et tests de friabilité, et $O(q\alpha(q))$ opérations supplémentaires pour gérer la structure, où κ est une constante de l'ordre de* 0.25.

Si l'on néglige le fait que le premier cycle n'est pas toujours le bon, le modèle uniforme donne $\kappa = 0.16$ et le modèle permuté donne $\kappa = 0.22$.

À titre de comparaison, la méthode Rho nécessite en moyenne $0.88q$ opérations de groupe (d'après la formule $\sqrt{\frac{\pi N}{2 \times 2}}$, où $N \approx q^2$), et la méthode de Shanks nécessite en moyenne $0.75q$ opérations de groupe (en utilisant l'involution hyperelliptique).

Bien entendu, la méthode Rho et ses variantes parallèlisables restent les seules praticables pour des exemples très grands, car elles ne nécessitent quasiment pas de mémoire.

12.2.3 Coût du test de friabilité

Dans la Jacobienne d'une courbe de genre 2, le test de friabilité revient à tester si un polynôme $u(x) = x^2 + u_1 x + u_2$ de degré 2 est scindé ou non, et si oui à donner ses racines. Nous allons étudier seulement le cas des corps premiers et le cas de la caractéristique 2.

Corps premier $\mathbb{F}_p$

Le discriminant de $u(x)$, donné par $\Delta = u_1^2 - 4u_2$ coûte essentiellement 1 carré. Tester si Δ est un carré se fait de manière très efficace par un calcul de symbole de Legendre en utilisant la réciprocité quadratique, ce qui fait une complexité de $O(\log^2 p)$.

Pour l'extraction de la racine carrée, la complexité depend de la valuation de 2 dans $p-1$. Si celle-ci n'est pas trop grande, on obtient une complexité de $O(\log^3 p)$.

Corps $\mathbb{F}_{2^n}$

Le test de friabilité consiste en la résolution d'une équation de degré 2. En caractéristique 2, cela se fait tout d'abord par le calcul de la trace d'une certaine quantité appelée aussi discriminant, ce qui coûte $O(n^2)$ opérations pour calculer le discriminant, puis $O(n)$ pour la trace. L'équation a des solutions si et seulement si cette trace est nulle.

Ensuite le calcul de la solution se ramène à un produit d'une matrice précalculée par un vecteur, ce qui se fait en $O(n^2)$ opérations. Nous renvoyons à [McE95] pour plus de détails sur ces algorithmes.

12.2.4 Utilisation d'automorphismes

L'algorithme présenté ci-dessus utilise l'involution hyperelliptique afin de réduire la taille du graphe de moitié, avec en contrepartie une probabilité d'échec de 1/2 lors de la découverte d'un cycle. Comme le deuxième cycle coûte beaucoup moins cher à découvrir que le premier, le gain est presque d'un facteur 1/2 dans le temps d'exécution et la mémoire requise.

Plus généralement, si la courbe $\mathcal{C}$ présente un automorphisme σ d'ordre m, on peut regrouper m points d'une même orbite sous l'action de σ en un seul sommet du graphe. Dans la structure Union–Find, les ε appartiennent alors à un ensemble de racines m-ième de l'unité. L'algorithme peut alors être recopié mot pour mot. La seule différence est que maintenant la probabilité qu'un cycle donne le logarithme discret est $\frac{1}{m}(1-\frac{1}{N})$, et donc qu'il faut attendre en moyenne le m-ième cycle avant d'avoir le résultat. Toutefois, là encore heuristiquement, le plus dur est d'atteindre le premier cycle. Les autres doivent suivre assez rapidement ensuite. Le gain de l'utilisation d'automorphismes est donc de l'ordre de m en temps et en espace, ce qui est à comparer avec la méthode Rho où le gain apporté par un automorphisme d'ordre m est seulement $\sqrt{m}$.

Il faut noter que pour une implantation pratique, il faut trouver un moyen de numéroter les éléments choisis comme représentant des classes de manière efficace, ce qui dépendera bien sûr du corps de définition et de l'automorphisme.

12.3 Mise en pratique

12.3.1 Quelques statistiques

Nous avons implanté notre algorithme en Magma avec pour but de vérifier les heuristiques faites dans l'analyse afin de confirmer que l'on a bien une complexité en $2\kappa q$ opérations avec une valeur de κ proche de la théorie.

Cas où Jac($\mathcal{C}$) **est cyclique d'ordre premier**

$y^2 = x^5 + 4618x^4 + 7736x^3 + 427x^2 + 9390x + 1348$ $q = 10007 \quad N = 102050869$				
(1000 essais)	min	max	moyenne	κ_{exp}
nombre d'opérations dans Jac($\mathcal{C}$)	651	6043	4665.1	0.233
nombre de cycles infructueux	0	10	1.025	
proportion d'éléments friables	0.4817	0.5353	0.5098	

$y^2 = x^5 + 26119x^4 + 45517x^3 + 95137x^2 + 96222x + 51634$ $q = 100003 \quad N = 10006879567$				
(1000 essais)	min	max	moyenne	κ_{exp}
nombre d'opérations dans Jac($\mathcal{C}$)	3667	54590	45356.2	0.226
nombre de cycles infructueux	0	11	1.049	
proportion d'éléments friables	0.4921	0.5153	0.5004	

$y^2 = x^5 + 156083x^4 + 952399x^3 + 641296x^2 + 941181x + 546291$ $q = 1000003 \quad N = 999649724383$				
(100 essais)	min	max	moyenne	κ_{exp}
nombre d'opérations dans Jac($\mathcal{C}$)	182788	519951	451913.4	0.226
nombre de cycles infructueux	0	6	0.98	
proportion d'éléments friables	0.4981	0.5012	0.4997	

$y^2 = x^5 + 3408650x^4 + 3817692x^3 + 1845808x^2 + 4237685x + 4123069$ $q = 5000011 \quad N = 24985522767991$				
(10 essais)	min	max	moyenne	κ_{exp}
nombre d'opérations dans Jac($\mathcal{C}$)	744296	2531646	2270840	0.227
nombre de cycles infructueux	1	2	0.5	
proportion d'éléments friables	0.4988	0.5003	0.4997	

On constate que les proportions d'éléments friables et le nombre de cylces infructueux sont très proches de ce que donne la théorie. En effet, une probabilité de cycle infructueux de 1/2 donne un nombre moyen d'échecs de 1, car en cas de premier cycle inutile, on a de nouveau une probabilité de 1/2 d'échec pour le suivant. Ainsi le nombre moyen théorique est $\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\cdots = 1$.

Les constantes κ observées sont aux alentours de 0.23. Un point intéressant à noter est que l'écart-type ne semble pas trop mauvais, et que l'on ne s'éloigne pas trop de la valeur moyenne.

Cas où l'on est dans un sous-groupe de Jac($\mathcal{C}$)

Lorsque $N = \mathrm{ord}(D_1)$ est un diviseur de $\#\mathrm{Jac}(\mathcal{C})$, le modèle du graphe aléatoire n'est plus vrai : seulement une partie des arêtes peuvent être obtenues. Si la quantité d'arêtes possibles est faible, le cycle que l'on obtiendra sera presque sûrement un cycle formé de deux fois la même arête. C'est-à-dire que l'on retombe sur quelque chose de ressemblant à la méthode Rho. On peut donc s'attendre à une complexité qui dépend de N et non du cardinal du groupe. Ceci est confirmé par des expériences pratiques résumées dans le tableau suivant :

$y^2 = x^5 + 138662x^4 + 782649x^3 + 166700x^2 + 188833x + 481080$ $q = 1000003 \quad N = 5519686061 \quad \#\mathrm{Jac} = 181 \times N$			
(1000 essais)	min	max	moyenne
nombre d'opérations dans Jac($\mathcal{C}$)	2919	194254	75261
nombre de cycles infructueux	0	1	0.004
proportion d'éléments friables	0.4864	0.5105	0.4995

$y^2 = x^5 + 119628x^4 + 465127x^3 + 927173x^2 + 579002x + 818311$ $q = 1000003 \quad N = 11823841 \quad \#\mathrm{Jac} = 84547 \times N$			
(1000 essais)	min	max	moyenne
nombre d'opérations dans Jac($\mathcal{C}$)	177	10391	3585
nombre de cycles infructueux	0	0	0
proportion d'éléments friables	0.4476	0.5423	0.5002

On constate que le temps de calcul est de l'ordre de $\sqrt{N}$. Par ailleurs, le nombre de cycles infructueux est devenu négligeable, ce qui est effectivement le cas lorsque le cycle consiste en deux fois la même arête.

12.3.2 Spéculations pour une implantation efficace

Pour une implantation efficace, le principal problème est d'économiser la mémoire. Soit $\mathcal{C}$ une courbe fixée sur $\mathbb{F}_q$ de genre 2 dont la Jacobienne est cyclique d'ordre premier $N \approx q^2$. La quantité minimale d'informations que doit contenir une case du tableau est la suivante :

– Le numéro de la case du père : $\lceil \log_2 q \rceil$ bits.

– Les valeurs α et β : ce sont des entiers modulo N, ils nécessitent donc $\lceil \log_2 N \rceil$ bits. Soit au total : $2\lceil 2 \log_2 q \rceil$ bits.

– Le signe ε : 1 bit.

Ainsi une case du tableau doit avoir une taille d'environ $5 \log_2 q$ bits.

Le nombre de case théoriquement nécessaire est le nombre de points sur $\mathbb{F}_q$ divisé par deux, donc environ $\frac{q}{2}$. Toutefois cela suppose d'avoir trouvé un moyen de numéroter les points de manière très efficace, et on pense aussitôt aux techniques de hachage. Pour obtenir un hachage performant, il faut cependant accorder un peu de marge en mémoire, et donc reperdre un facteur 1.25 par exemple. Si on ajoute à cela le surcoût de la gestion de ce hachage, il ne paraît pas évident que l'on va gagner par rapport à l'utilisation d'un tableau indexé par l'abscisse des points qui perd un facteur 2.

q	10^4	10^5	10^6	10^7	10^8	10^9	10^{10}
#Jac($\mathcal{C}$)	10^8	10^{10}	10^{12}	10^{14}	10^{16}	10^{18}	10^{20}
mémoire nécessaire	80 ko	1 Mo	12 Mo	140 Mo	1.6 Go	18 Go	200 Go

Au vu de cet espace mémoire énorme, même pour des exemples relativement petit, on se rend compte qu'il est inutile de tenter une implantation optimisée de cet algorithme.

Étudions maintenant le cas d'une courbe de genre 2 définie sur $\mathbb{F}_2$ et vue sur $\mathbb{F}_{2^k}$ (courbe de Koblitz). La composition du Frobenius et de l'involution hyperelliptique donne un automorphisme d'ordre $2k$ et donc le tableau ne contient que $m = \frac{2^k}{2k}$ cases. Cette fois-ci on n'échappe pas au hachage sous peine de perdre beaucoup d'espace. Le stockage mémoire est le suivant :

- Le numéro de la case du père : $\lceil \log_2 m \rceil$ bits.
- Les valeurs α et β : ce sont des entiers modulo $N \approx 2^{2k}$, ils nécessitent donc $2k$ bits. Soit au total : $4k$ bits.
- La racine de l'unité ε : on peut la coder sur $\log_2(2k)$ bits.

Ainsi une case du tableau doit avoir une taille d'environ $5k$ bits. Ce qui donne les valeurs suivantes pour la mémoire nécessaire :

k	13	17	19	23	29	31	37
$\#\mathrm{Jac}(\mathcal{C})$	$7\,10^7$	$2\,10^{10}$	$3\,10^{11}$	$7\,10^{13}$	$3\,10^{17}$	$5\,10^{18}$	$2\,10^{22}$
mémoire nécessaire	3 ko	40 ko	160 ko	2.6 Mo	160 Mo	650 Mo	42 Go

On constate donc, que la réduction de mémoire grâce aux automorphismes n'est pas suffisante pour rendre utilisable l'algorithme pour des exemples « grandeur nature ».

Chapitre 13

Application à la Restriction de Weil

L'idée d'utiliser la restriction de Weil dans un cadre cryptographique est due à Frey [Fre98] qui voyait ainsi un moyen de déguiser une courbe elliptique en une variété abélienne en apparence compliquée. Les applications cryptographiques étaient d'une part la création de systèmes à brèche secrète et d'autre part la découverte de nouvelles classes de courbes faibles.

Cette idée à ensuite été reprise par Galbraith et Smart [GS99] qui ont étudié plus en détail le cas de la caractéristique 2. Nous présentons ici les progrès récents effectués par Heß et Smart ainsi que notre contribution sur le sujet [GHS00].

13.1 Restriction de Weil d'une courbe elliptique

13.1.1 Principe général

Le restriction de Weil a une définition abstraite dont nous ne parlerons pas ici. Nous allons préférer une approche plus terre-à-terre qui présente l'avantage de ne pas nécessiter de théorie pourvu que l'on admette les théorèmes. Par ailleurs, cette approche est constructive.

Soit $\mathcal{E}$ une courbe elliptique définie sur un corps fini $\mathbb{F}_{q^n}$, extension de degré n du corps $\mathbb{F}_q$. La restriction de Weil de $\mathcal{E}$ est une variété abélienne de dimension n définie sur le corps $\mathbb{F}_q$. Pour construire cette variété on voit $\mathbb{F}_{q^n}$ comme un espace vectoriel sur $\mathbb{F}_q$ de dimension n. On écrit ensuite chaque indéterminée dans l'équation de la courbe à l'aide d'une base de $\mathbb{F}_{q^n}$. En identifiant coordonnées par coordonnées, on obtient ainsi une famille de n équations en $2n$ inconnues, ce qui définit bien une variété de dimension n.

Plus précisément, supposons que $\mathbb{F}_{q^n}$ est défini par

$$\mathbb{F}_{q^n} = \mathbb{F}_q[t]/(f(t)),$$

où $f(t)$ est un polynôme unitaire irréductible de degré n. Supposons que le corps de base est de caractéristique différente de 2 et 3 et que $\mathcal{E}$ a pour équation

$$Y^2 = X^3 + \alpha X + \beta.$$

La base choisie est donc formée par les puissances de t inférieures à n. On écrit les inconnues X, Y et les paramètres α, β sur cette base :

$$X = \sum_{0 \leqslant i < n} x_i t^i, \quad Y = \sum_{0 \leqslant i < n} y_i t^i, \quad \alpha = \sum_{0 \leqslant i < n} a_i t^i, \quad \beta = \sum_{0 \leqslant i < n} b_i t^i.$$

À partir de l'expression de Y, on peut calculer Y^2 comme polynôme en t, puis réduire ce polynôme modulo $f(t)$, de sorte que Y^2 s'exprime comme un polynôme en t de degré inférieur à n à coefficients dans les polynômes en les y_i sur $\mathbb{F}_q$. De même on peut écrire le membre de droite de l'équation de $\mathcal{E}$ comme un polynôme en t de degré inférieur à n à coefficients dans les polynômes en les x_i, a_i, b_i sur $\mathbb{F}_q$. Les deux membres étant égaux, on identifie les coefficients des deux polynômes en t. On a

$$0 = Y^2 - (X^3 + \alpha X + \beta) = \sum_{0 \leqslant j < n} \varphi_j(a_i, b_i, x_i, y_i) t^j,$$

où les φ_j sont des polynômes, ce qui engendre le système de n équations

$$\forall\ 0 \leqslant j < n, \quad \varphi_j(a_i, b_i, x_i, y_i) = 0.$$

On travaille pour une courbe $\mathcal{E}$ fixée, aussi les a_i et b_i sont-ils des constantes. Les inconnues, au nombre de $2n$, sont donc les x_i et y_i ; et la variété ainsi définie est (a priori) de dimension n.

Nous allons admettre le résultat suivant :

Théorème 13.1 *La variété A ainsi construite est de dimension n et peut être munie d'une structure de variété abélienne induite par la loi de groupe sur la courbe elliptique. De plus si l'on note $\chi_{\mathcal{E}}(t)$ le polynôme caractéristique du Frobenius sur $\mathcal{E}$ et $\chi(t)$ celui de cette variété abélienne, on a*

$$\chi(t) = \chi_{\mathcal{E}}(t^n).$$

Par construction, les points de $\mathcal{E}$ et de A sont en bijection, et c'est en ce sens que la loi de groupe sur $\mathcal{E}$ en induit une sur A.

Lorsque la courbe elliptique est elle-même définie sur le corps de base (courbe de Koblitz), on a un résultat supplémentaire :

Proposition 13.1 *Soit $\mathcal{E}$ une courbe elliptique définie sur $\mathbb{F}_q$. Soit A la restriction de Weil de $\mathcal{E}$ considérée comme définie sur $\mathbb{F}_{q^n}$. Alors $\mathcal{E}$ apparaît comme sous-variété abélienne de A. La variété abélienne B telle que $A \sim E \times B$ est simple sur $\mathbb{F}_q$.*

L'intérêt de la restriction de Weil est que si A est la Jacobienne d'une courbe hyperelliptique, on a deux structures de groupes isomorphes, et travailler dans l'une ou dans l'autre revient au même d'un point de vue sécurité cryptographique. D'un côté cela permet de construire des courbes hyperelliptiques pour lesquelles la sécurité est la même que pour la courbe elliptique dont elle provient, de l'autre, on peut espérer attaquer ainsi des cryptosystèmes elliptiques en utilisant les attaques sous-exponentielles décrites dans ce mémoire.

Le problème est qu'il est extrêmement improbable que A soit une Jacobienne. On va donc demander un peu moins : si l'on trace une courbe $\mathcal{C}$ sur A, alors la Jacobienne de $\mathcal{C}$ contient une sous-variété isogène à une sous-variété de A, et l'on peut espérer s'en sortir en travaillant dans $\mathrm{Jac}(\mathcal{C})$ au lieu de A. On est confronté aux problèmes suivants :

1. Comment trouver une courbe dont le genre (et donc la dimension de $\mathrm{Jac}(\mathcal{C})$) ne sera pas trop grand par rapport à la dimension de A?

2. Peut-on garantir que cette courbe sera hyperelliptique?

3. Comment transporter le problème du logarithme discret à travers ces isogénies?

Le problème 1 est que si le genre de $\mathcal{C}$ est trop grand, on va être amené à travailler dans un groupe pour lequel seulement un tout petit sous-groupe nous interessera, ce qui ne peut se faire sans perte d'efficacité.

Le problème 2 est lié au fait que dans l'état actuel de l'algorithmique, les courbes hyperelliptiques sont de loin les plus faciles à manipuler. Ceci est en partie en train d'évoluer et l'on peut raisonnablement envisager que des courbes plus générales : superelliptiques [GPS00] ou C_{ab} [Ari99] seront bientôt d'utilisation aussi aisée.

Le problème 3 signifie que pour relier la sécurité des crypotosystèmes basés sur $\mathcal{E}$ et sur $\mathcal{C}$ une condition nécessaire et suffisante est de pouvoir transférer un problème de log discret de l'un vers l'autre.

Avec une construction spécifique à la caractéristique 2, due à Galbraith et Smart, Heß a pu donner des réponses satisfaisantes à ces problèmes dans certains cas. On peut alors appliquer notre algorithme du chapitre 11 et trouver des familles de courbes elliptiques plus faibles que les courbes générales.

Nous allons maintenant décrire cette construction, en commençant par des exemples, et nous donnerons le théorème d'existence d'une courbe hyperelliptique.

13.1.2 Exemples en caractéristique 2

On se place sur un corps $\mathbb{F}_{q^n}$ de caractéristique 2 pour lequel il existe une base sur $\mathbb{F}_q$ de la forme

$$\{\theta, \theta^2, \theta^4, \theta^8, \ldots, \theta^{2^{n-1}}\},$$

avec de plus

$$\theta + \cdots + \theta^{2^{n-1}} = 1,$$

et l'on considère une courbe elliptique $\mathcal{E}$ d'équation

$$Y^2 + XY = X^3 + \beta.$$

Ces restrictions sur le corps choisi ont pour but de simplifier les formules obtenues lorsque l'on effectue les calculs de la restriction de Weil.

On commence par écrire X et Y sur la base choisie :

$$X = \sum_{0 \leqslant i < n} x_i \theta^{2^i}, \quad Y = \sum_{0 \leqslant i < n} y_i \theta^{2^i}.$$

Plutôt que de calculer les équations de A qui sont assez compliquées, on calcule directement le système définissant une courbe $\mathcal{C}$ tracée sur A. De manière à maintenir un degré pas trop grand, en espérant que le genre suivra, on coupe la variété A par les hyperplans de coordonnées

$$x_0 = x_1 = x_2 = \cdots = x_{n-1} = x.$$

Ces $n-1$ équations supplémentaires ramènent donc à une courbe. Le choix de la base est tel que

$$X = \sum_{0 \leqslant i < n} x_i \theta^{2^i} = x_0 \sum_{0 \leqslant i < n} \theta^{2^i} = x \in \mathbb{F}_q,$$

d'où $X^3 \in \mathbb{F}_q$ et

$$X^3 = \sum_{0 \leqslant i < n} x^3 \theta^{2^i}.$$

De même

$$XY = \sum_{0 \leqslant i < n} x y_i \theta^{2^i}.$$

Par ailleurs, on a

$$Y^2 = \sum_{0 \leqslant i < n} y_{i-1}^2 \theta^{2^i}.$$

Si l'on note b_i les coordonnées de β, il est alors facile de voir que la courbe $\mathcal{C}$ est définie par le système

$$\left\{ \begin{array}{rcl} y_{n-1}^2 + xy_0 + x^3 + b_0 & = & 0 \\ y_0^2 + xy_1 + x^3 + b_1 & = & 0 \\ & \vdots & \\ y_{n-2}^2 + xy_{n-1} + x^3 + b_{n-1} & = & 0 \end{array} \right.$$

dans lequel on peut éliminer de proche en proche les variables y_1, y_2, $\ldots$ jusqu'à n'avoir plus qu'une équation en $y = y_0$ et en x. Cette équation prend la forme

$$y^{2^n} + x^{2^n-1} y + \sum_{0 \leqslant i < n} x^{2^n + 2^i} + g(x) = 0,$$

où $g(x)$ est un polynôme de degré au plus 2^n.

Cas où $n = 2$

L'équation devient

$$\mathcal{C}_2 : y^4 + x^3 y + x^6 + x^5 + b_0 x^2 + b_1^2 = 0.$$

Si la courbe $\mathcal{E}$ est définie sur le corps de base, c'est-à-dire si $b_0 = b_1$, alors $\mathcal{C}_2$ a deux composantes irréductibles dont une est $\mathcal{E}$ et l'autre sa tordue quadratique. Dans les autres cas, la courbe est irréductible, de genre 2, et est donc hyperelliptique. C'est le cas idéal où la dimension de $\mathrm{Jac}(\mathcal{C})$ est égale à n.

Cas où $n = 4$

$$C_4 : y^{16} + x^{15} y + x^{24} + x^{20} + x^{18} + x^{17} + b_0 x^{14} + b_3^2 x^{12} + b_2^4 x^8 + b_1^8 = 0.$$

Experimentalement, Smart a remarqué que lorsque cette courbe est irréductible, le genre est toujours au plus 8. La courbe est réductible lorsque $b_3 = b_0 + b_1 + b_2$, et alors une des composantes est donnée par

$$C_{4a} : y^8 + x^4 y^4 + x^6 y^2 + x^7 y + x^{12} + x^9 + b_0 x^6 + (b_2^2 + b_1^2) x^4 + b_1^4 = 0.$$

Lorsque cette dernière courbe est elle-même irréductible, elle semble avoir toujours un genre au plus 4 et être de plus hyperelliptique.

C'est en s'appuyant sur ces observations que Heß et Smart ont démontré que ce phénomène se généralisait.

13.2 Théorème d'existence d'une courbe hyperelliptique

Théorème 13.2 *Soit $\mathcal{E}$ une courbe elliptique d'équation*

$$y^2 + xy = x^3 + \alpha x^2 + \beta$$

définie sur $\mathbb{F}_{q^n}$, où q est une puissance de 2. Soit $\mathcal{C}$ la courbe obtenue en coupant la restriction de Weil de $\mathcal{E}$ par les hyperplans correspondant à « $x \in \mathbb{F}_q$ ». Alors il existe un entier $m \in [1..n]$ tel que $\mathcal{C}$ ait une composante irréductible qui est hyperelliptique de genre

$$2^{m-1} \quad ou \quad 2^{m-1} - 1.$$

De plus, ce théorème est constructif au sens où m et $\mathcal{C}$ sont effectivement calculables et on peut transporter les points par un homomorphisme effectif de $\mathcal{E}$ vers $\mathrm{Jac}(\mathcal{C})$.

Nous allons donner seulement une esquisse de la démonstration. La preuve complète se trouve dans [GHS00].

On étudie tout d'abord la géométrie de la courbe $\mathcal{C}$, et donc on peut se placer sur une extension de $\mathbb{F}_q$. En l'occurence, en se plaçant sur $\mathbb{F}_{q^n}$, on peut montrer que $\mathcal{C}$ est birationnellement équivalente à

$$\mathcal{D} : \begin{cases} w_0^2 + xw_0 + x^3 + \alpha_0 x^2 + \beta_0 = 0, \\ \qquad\vdots \\ w_{n-1}^2 + xw_{n-1} + x^3 + \alpha_{n-1} x^2 + \beta_{n-1} = 0. \end{cases}$$

où $\alpha_i = \sigma^i(\alpha)$ et $\beta_i = \sigma^i(\beta)$, sont les n conjugués de α et β.

On utilise alors le point de vue « corps de fonctions ». Soit

$$F_i = \mathbb{F}_{q^n}(x)[w_i]/(w_i^2 + xw_i + x^3 + \alpha_i x^2 + \beta_i)$$

le corps de fonction de degré 2 sur $\mathbb{F}_{q^n}(x)$ défini par la i-ième équation. Soit F le compositum des corps F_i. Le degré de F est une puissance de 2 inférieure ou égale à n que l'on note m :

$$[F : \mathbb{F}_{q^n}(x)] = 2^m.$$

Ainsi $\mathcal{D}$ a 2^{m-n} composantes irréductibles ayant pour corps de fonctions F.

Les équations des corps F_i peuvent être transformées en équations de la forme

$$s_i^2 + s_i + z^{-1} + \alpha_0 + \sqrt{\beta_0} z = 0,$$

où $z = 1/x$ et la théorie des extensions d'Artin–Schreier donne alors m comme dimension d'un espace vectoriel sur $\mathbb{F}_2$, au moins dans le cas où $\alpha = 0$ ou 1 : on a alors

$$m = \dim_{\mathbb{F}_2}\left(\mathrm{Vect}_{\mathbb{F}_2}\left\{(1, \sqrt{\beta_0}), \ldots, (1, \sqrt{\beta_{n-1}})\right\}\right).$$

Ainsi il est extrêmement aisé de calculer ce paramètre.

Quitte à réordonner, on peut supposer que le corps F est le compositum des m premiers F_i et après quelques changements de variables et manipulations d'équations, on constate que le corps F est le compositum de F_0 et des corps $(L_i)_{1 \leqslant i \leqslant m-1}$ d'équations de la forme

$$t_i^2 + t_i + \delta_i z + \gamma_i,$$

où t_i est une nouvelle variable et δ_i et γ_i sont des constantes effectivement calculables. Soit L le compositum $L_1 \cdots L_{m-1}$.

On a $F = F_0 L$, et on peut montrer que L est une extension rationnelle de $\mathbb{F}_{q^n}(z)$, donc est de genre 0. En fait ce calcul peut être fait explicitement et l'on dispose d'un élément c tel que z est un polynôme en c et $L = \mathbb{F}_{q^n}(c)$.

Il est alors aisé d'en déduire que F est hyperelliptique. Le calcul du genre, un peu plus compliqué, est lié au degré du polynôme qui lie c à $z = 1/x$.

Il reste ensuite à transporter les points, de $\mathcal{E}$ vers la courbe de corps de fonctions F, et de redescendre sur le corps $\mathbb{F}_q$. Sans rentrer dans les détails, cela se fait grâce à la conorme qui s'applique aux diviseurs et donc aussi aux classes de diviseurs :

$$\mathrm{Cl}^0(\mathbb{F}_{q^n}(\mathcal{E})) \xrightarrow{\mathrm{Con}_{F/\mathbb{F}_{q^n}(\mathcal{E})}} \mathrm{Cl}^0(F) \xrightarrow{\mathrm{N}_{F/F'}} \mathrm{Cl}^0(F'),$$

où F' est le sous-corps de F fixé par le Frobenius.

13.3 Conséquences cryptographiques

13.3.1 Construction de cryptosystèmes sûrs

Partant d'une courbe elliptique $\mathcal{E}$ définie sur un corps fini $\mathbb{F}_{q^2}$ de caractéristique 2, on peut construire la restriction de Weil de $\mathcal{E}$, puis une courbe hyperelliptique $\mathcal{C}$ de genre 2 sur $\mathbb{F}_q$ dont la Jacobienne lui sera isogène.

Le cardinal de $\mathrm{Jac}(\mathcal{C})$ est alors plus facile à déterminer que dans le cas d'une courbe quelconque. En effet, par l'algorithme de Schoof et les améliorations de Couveignes et Lercier ou par l'algorithme de Satoh, il est possible de calculer le cardinal de $\mathcal{E}$ sur $\mathbb{F}_{q^n}$ en un temps très raisonnable (quelques secondes pour une taille cryptographique). La restriction de Weil et l'isogénie conservent le cardinal, donc la valeur calculée est aussi l'ordre de $\mathrm{Jac}(\mathcal{C})$ sur $\mathbb{F}_q$.

On peut alors se servir de la courbe $\mathcal{C}$ pour bâtir un cryptosystème basé sur le problème du logarithme discret dans $\mathrm{Jac}(\mathcal{C})$.

La sécurité de ce système est exactement la même que si l'on utilise la courbe $\mathcal{E}$, puisque le problème du log discret peut être transféré d'un groupe à l'autre. En particulier, une attaque de Frey–Rück sera possible sur $\mathcal{E}$ si et seulement si elle est possible sur $\mathcal{C}$, et le degré de l'extension sera identique. De même, une attaque du type Smart–Araki–Satoh–Semaev est possible sur $\mathcal{E}$, si et seulement si l'attaque de Rück fonctionne sur $\mathcal{C}$.

Un exemple d'une courbe de genre 2 sûre définie sur $\mathbb{F}_{2^{81}}$ est donnée dans [GHS00].

Cette méthode permet donc de fabriquer des cryptosystèmes de genre 2. Notons toutefois que par ce biais, on n'agrandit qu'artificiellement la famille des groupes utilisables : on ne construit aucunement des solutions de remplacement pour le cas où les courbes elliptiques seraient cassées.

13.3.2 Attaque de cryptosystèmes elliptiques

Prenant le contrepied du point de vue précédent, on peut attaquer des systèmes elliptiques grâce à la restriction de Weil. En effet, il a été montré au chapitre 11 qu'il existe un algorithme résolvant un problème de log discret dans la Jacobienne d'une courbe hyperelliptique de genre g sur $\mathbb{F}_q$ en temps $O(q^2)$ lorsque g est fixé et que q tend vers l'infini. En rajoutant cet algorithme à la sortie du calcul de la restriction de Weil et du transport des points, on obtient le résultat suivant :

Théorème 13.3 *Pour une proportion significative des courbes elliptiques définies sur un corps* *fini $\mathbb{F}_{q^n}$ de caractéristique 2, il existe un algorithme résolvant le problème du logarithme discret en temps $O(q^2)$, lorsque $n \geqslant 4$ est fixé et q tend vers l'infini.*

Précisons tout de suite que la dépendance en n est calamiteuse : la constante dans le $O()$ cache un facteur $n!$ qui ne peut pas être négligé en pratique. Ceci fait que l'on dispose seulement d'une petite fenêtre entre $n \geqslant 4$ et la non-faisabilité pratique.

Le théorème ne s'applique pas à toutes les courbes, car il existe des cas particuliers pour lesquels la restriction de Weil ne permettra pas de garder suffisamment d'informations. C'est la cas par exemple pour les courbes de Koblitz. En effet, dans ce cas la paramètre m est toujours égal à 1, ce qui signifie que la méthode décrite ci-dessus va nous ramener à la courbe elliptique de départ, oubliant la partie intéressante de la variété abélienne A. Toutefois, génériquement, on n'a pas de problème.

En utilisant la réduction de base dans le calcul d'index, il est expliqué page 173 que la complexité heuristique peut être ramenée à $O(q^{\frac{2g}{g+1}})$. Avec cette amélioration, on peut comparer la méthode restriction de Weil + calcul d'index à la méthode Rho sur la courbe elliptique de départ dont la complexité est $O(q^{n/2})$. On étudie pour le cas limite où $n = 4$, selon le genre obtenu pour $\mathcal{C}$:

Courbe	C_4	C_4	C_{4a}
n, g	4,8	4,7	4,4
Rho sur $\mathcal{E}(\mathbb{F}_{q^n})$	q^2	q^2	q^2
Index sur Jac($\mathcal{C}$)	$q^{16/9}$	$q^{7/4}$	$q^{8/5}$

Ainsi le cas le plus favorable est le cas C_{4a} qui se produit avec probabilité $1/q$. Nous avons experimenté cette attaque pour une courbe de ce type.

Soit le corps $\mathbb{F}_q = \mathbb{F}_{2^{21}}$ défini par $\mathbb{F}_2[w]/(w^{21} + w^2 + 1)$ et soit l'extension de degré $n = 4$ de $\mathbb{F}_q$ définie par $\mathbb{F}_{q^n} = \mathbb{F}_q[\theta]/(\theta^4 + \theta^3 + \theta^2 + \theta + 1)$. On considère la courbe elliptique

$$\mathcal{E} : Y^2 + XY = X^3 + b_0\theta + b_1\theta^2 + b_2\theta^4 + b_3\theta^8,$$

où

$$b_0 = 0, \quad b_1 = w^{1127280}, \quad b_2 = w^{171398}, \quad b_3 = w^{1370436}.$$

Notons que $b_3 = b_0 + b_1 + b_2$, de sorte que l'on est dans le cas C_{4a}. Par l'algorithme de Schoof, on trouve

$$\#\mathcal{E} = 2^4 \times 1208925819614311295169073.$$

Le processus de la section 13.1.2 produit la courbe hyperelliptique sur $\mathbb{F}_q$ d'équation $y^2 + h(x)y = f(x)$ où

$$h(x) = x^4 + w^{624429}x^3 + w^{1248858}x^2 + w^{1442662}x + w^{386860}$$

et

$$f(x) = x^9 + w^{1859582}x^6 + w^{293124}x^4 + w^{1783647}x^3 + w^{1541982}x^2 + w^{1370912}x + w^{1888298}.$$

Nous avons alors appliqué la méthode de calcul d'index pour des diviseurs aléatoires de cette Jacobienne. La base de facteurs contient a priori $\approx 2^{20}$ éléments, ce qui paraît un peu trop. Nous avons alors utilisé la réduction de base en décidant que l'on ne garderait que les éléments dont la représentation en machine a ses trois bits les plus significatifs à zéro. On a alors une base de facteurs de taille 131294 $\approx 2^{17}$. Finalement le calcul a pris l'équivalent d'environ 31 semaines sur un Pentium II 450 MHz pour la phase de recherche de relations (effectuée en parallèle), puis 64.4 heures pour l'algèbre linéaire sur la même machine.

Remarquons que la recherche de relations a été programmée dans un langage de haut niveau (C++ avec la bibliothèque Lidia) et qu'une implantation soignée, spécifique au corps considéré permettrait vraisemblablement de réduire cette première phase[3]. La deuxième phase a quant à elle été programmée de manière plus fine.

On peut estimer que le temps de calcul pour résoudre le même problème par la méthode Rho sur la courbe elliptique est de quelques dizaines de semaines pour la même machine. On a donc atteint la limite pour laquelle la restriction de Weil devient plus efficace que la méthode Rho.

Remarque. L'attaque exposée dans ce chapitre n'est pas une attaque générale sur toutes les courbes elliptiques en caractéristique 2 : seule une famille très particulière de courbes est touchée. Les conclusions à retenir sont

– Il ne faut pas utiliser de courbes dont on ne connaît pas la provenance : cette courbe peut avoir été choisie pour disposer d'une faiblesse. Les courbes non générées aléatoirement sont à proscrire.

– Une courbe définie sur un corps $\mathbb{F}_{2^n}$ où n est composé laisse plus de liberté pour une attaque par restriction de Weil. Il faut préférer les extensions de degré premier.

3. Smart [Sma00] a récemment effectué de nouvelles expériences, avec une nouvelle implantation.

Conclusion

Dans ce mémoire, nous avons présenté notre contribution à l'étude de l'algorithmique des courbes hyperelliptiques. L'objectif à terme est de pouvoir traiter toutes les instances, sans se limiter aux cas particuliers comme les courbes à multiplications complexes ou les courbes de Koblitz. Dans le cas des courbes elliptiques, on dispose déjà d'un savoir-faire suffisant pour ouvrir la voie à des applications pratiques et on peut espérer que nos travaux contribueront à l'utilisation des courbes hyperelliptiques pour le même type d'applications.

Jusqu'ici, hormis pour quelques cas particuliers, seuls des algorithmes de complexité exponentielle étaient employés pour calculer la cardinalité d'une courbe hyperelliptique. Des algorithmes en temps polynomial étaient connus, mais considérés comme non compétitifs. Nous avons montré que ces algorithmes peuvent être améliorés de sorte qu'ils se révèlent utiles en pratique : cela produit un gain substantiel par rapport aux performances obtenues auparavant, et il est désormais possible de traiter des Jacobiennes de taille $\approx 10^{38}$. En s'inspirant du savoir-faire pour les courbes elliptiques, nous avons abordé de nouvelles approches pour accélérer les calculs de cardinalité, dans lesquelles les équations modulaires liées aux isogénies en genre 2 jouent un rôle important. S'il n'est toujours pas envisageable de traiter des courbes aléatoires de taille cryptographique, la poursuite dans cette voie pourra s'avérer fructueuse.

D'autre part, les équations modulaires pourront être utilisées dans un autre contexte : Satoh [Sat00] a récemment proposé un nouvel algorithme pour le calcul du nombre de points d'une courbe elliptique bien adapté au cas des corps finis de petite caractéristique. Cette méthode, reposant sur le relèvement canonique de la courbe dans les p-adiques, s'est révélée très efficace en pratique [FGH00]. Il est naturel de tenter de généraliser l'algorithme de Satoh au genre 2.

En ce qui concerne le logarithme discret, nous avons sensiblement amélioré les attaques sous-exponentielles de manière à les rendre appliquables même pour des courbes de genre plus petit que ce qui était communément admis. Auparavant, des arguments heuristiques suggéraient de ne pas considérer les courbes de genre supérieur à 6. Nous avons rabaissé cette limite au genre 4, par des analyses de complexités illustrées par des expériences pratiques. Ainsi l'utilisation de courbes de genre supérieur à 4 n'est-elle plus pertinente dans un contexte cryptographique.

Cette attaque des courbes de genre « pas trop grand » a permis de plus de mettre à jour une nouvelle classe de courbes elliptiques faibles, grâce à la restriction de Weil. Il est désormais recommandé de ne pas utiliser de courbes définies sur une extension de $\mathbb{F}_2$ de degré non premier. Dans ce contexte de la restriction de Weil, il serait intéressant d'aller plus loin et d'étudier le log discret sur des courbes plus générales que les courbes hyperelliptiques.

Bibliographie

[AD93] L. M. Adleman and J. DeMarrais, *A subexponential algorithm for discrete logarithms over all finite fields*, Math. Comp. **61** (1993), no. 203, 1–15.

[ADH94] L. M. Adleman, J. DeMarrais, and M.-D. Huang, *A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields*, ANTS-I (L. Adleman and M.-D. Huang, eds.), Lecture Notes in Comput. Sci., vol. 877, Springer–Verlag, 1994, pp. 28–40.

[AH92] L. M. Adleman and M.-D. A. Huang, *Primality testing and Abelian varieties over finite fields*, Lecture Notes in Math., vol. 1512, Springer–Verlag, 1992.

[AH94] L. Adleman and M.-D. Huang (eds.), *ANTS-I*, Lecture Notes in Comput. Sci., vol. 877, Springer–Verlag, 1994, 1st Algorithmic Number Theory Symposium - Cornell University, May 6–9, 1994.

[AH96] L. Adleman and M.-D. Huang, *Counting rational points on curves and abelian varieties over finite fields*, ANTS-II (H. Cohen, ed.), Lecture Notes in Comput. Sci., vol. 1122, Springer–Verlag, 1996, pp. 1–16.

[AM93] A. O. L. Atkin and F. Morain, *Elliptic curves and primality proving*, Math. Comp. **61** (1993), no. 203, 29–68.

[Ari99] S. Arita, *Algorithms for computations in Jacobians of C_{ab} curve and their application to discrete-log-based public key cryptosystems*, Proceedings of Conference on The Mathematics of Public Key Cryptography, Toronto, June 12–17, 1999.

[Ari00] S. Arita, *Construction of secure C_{ab} curves using modular curves*, ANTS-IV (W. Bosma, ed.), Lecture Notes in Comput. Sci., vol. 1838, Springer–Verlag, 2000, pp. 113–126.

[ARS78] L. M. Adleman, R. L. Rivest, and A. Shamir, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21** (1978), no. 2, 120–126.

[Art24] E. Artin, *Quadratische Körper im Gebiet der höheren Kongruenzen II*, Math. Z. **19** (1924), 207–246.

[Atk92] A. O. L. Atkin, *The number of points on an elliptic curve modulo a prime*, Serie of e-mails to the `NMBRTHRY` mailing list, 1992.

[Bas96] J. Basmaji, *Ein Algorithmus zur Berechnung von Hecke-Operatoren und Anwendungen auf modulare Kurven*, Ph.D. thesis, Universität Gesamthochschule Essen, 1996.

[BC97] W. Bosma and J. Cannon, *Handbook of Magma functions*, 1997, `http://www.maths.usyd.edu.au:8000/u/magma/`.

[Ben99] P. Bending, *Curves of genus 2 with $\sqrt{2}$ multiplication*, 1999.

[BK98] J. Buhler and N. Koblitz, *Lattice basis reduction, Jacobi sums and hyperellitic cryptosystems*, Bull. Austral. Math. Soc. **58** (1998), 147–154.

[BM88] J.-B. Bost and J.-F. Mestre, *Moyenne arithmético-géométrique et périodes de courbes de genre 1 et 2*, Gaz. Math. Soc. France **38** (1988), 36–64.

[Bol87] O. Bolza, *Darstellung der rationalen ganzen Invarianten des Binärform sechsten Grades durch die Nullwerte der zugehörigen ϑ-Funktionen*, Math. Ann. **30** (1887), 478–495.

[Bol88] O. Bolza, *On binary sextics with linear transformations onto themselves*, Amer. J. Math. **10** (1888), 47–70.

[Bos00] W. Bosma (ed.), *ANTS-IV*, Lecture Notes in Comput. Sci., vol. 1838, Springer–Verlag, 2000, Fourth Algorithmic Number Theory Symposium, Leiden, The Netherlands, July 2000.

[Bro93] B. Brock, *Superspecial curves of genera two and three*, Ph.D. thesis, Princeton University, 1993.

[Bru95] A. Brumer, *The rank of $J_0(N)$*, Astérisque **228** (1995), 41–68.

[BSS99] I. Blake, G. Seroussi, and N. Smart, *Elliptic curves in cryptography*, London Math. Soc. Lecture Note Ser., vol. 265, Cambridge University Press, 1999.

[Buh98] J. P. Buhler (ed.), *ANTS-III*, Lecture Notes in Comput. Sci., vol. 1423, Springer–Verlag, 1998, Third Algorithmic Number Theory Symposium, Portland, Oregon, USA, June 1998.

[BW88] J. Buchmann and H. C. Williams, *A key-exchange system based on imaginary quadratic fields*, J. of Cryptology **1** (1988), 107–118.

[Can87] D. G. Cantor, *Computing in the Jacobian of an hyperelliptic curve*, Math. Comp. **48** (1987), no. 177, 95–101.

[Can94] D. G. Cantor, *On the analogue of the division polynomials for hyperelliptic curves*, J. Reine Angew. Math. **447** (1994), 91–145.

[Car57] P. Cartier, *Une nouvelle opération sur les formes différentielles*, C. R. Acad. Sci. Paris Sér. I Math. **244** (1957), 426–428.

[Cas91] J.W.S. Cassels, *Lectures on elliptic curves*, LMS student Texts, vol. 24, Cambridge University Press, 1991.

[Cav99] S. Cavallar, *Strategies in filtering in the Number Field Sieve*, Extended abstract, conference MPKC, Toronto, June 1999.

[CC86] D. V. Chudnovsky and G. V. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factorization tests*, Adv. in Appl. Math. **7** (1986), 385–434.

[CCR91] L. S. Charlap, R. Coley, and D. P. Robbins, *Enumeration of rational points on elliptic curves over finite fields*, Draft, 1991.

[CF96] J. W. S. Cassels and E. V. FLynn, *Prolegomena to a middlebrow arithmetic of curves of genus 2*, London Math. Soc. Lecture Note Ser., vol. 230, Cambridge University Press, 1996.

[CL98] F. Chabaud and R. Lercier, *Zen, a new toolbox for computing in finite extensions of finite rings*, February 1998, distributed with the ZEN package at `http://www.dmi.ens.fr/~zen`.

[Cle72] A. Clebsch, *Theorie der Binären Algebraischen Formen*, B.G. Teubner, Leipzig, 1872.

[CLO98] D. Cox, J. Little, and D. O'Shea, *Using algebraic geometry*, Graduate Texts in Mathematics, vol. 185, Springer-Verlag, 1998.

[CMNT97] J. Chao, N. Matsuda, O. Nakamura, and S. Tsujii, *Cryptosystems based on CM abelian variety*, Proc. Symposium on Cryptography and Information Security, 1997.

[CMST97] J. Chao, N. Matsuda, J. Sato, and S. Tsujii, *Efficient construction of secure hyperelliptic discrete logarithm problems of large genera*, Proc. Symposium on Cryptography and Information Security, 1997, Fukuoka, Japan.

[Coh93] H. Cohen, *A course in algorithmic algebraic number theory*, Graduate Texts in Mathematics, vol. 138, Springer–Verlag, 1993, Second corrected printing, 1995.

[Coh96] H. Cohen (ed.), *ANTS-II*, Lecture Notes in Comput. Sci., vol. 1122, Springer–Verlag, 1996, 2nd Algorithmic Number Theory Symposium - Université de Bordeaux, May 18–23, 1996.

[Coh99] H. Cohen, *Advanced topics in computational number theory*, Graduate Texts in Mathematics, vol. 193, Springer–Verlag, 1999.

[Cre] J. Cremona, `mwrank` *program*, Available at `http://www.maths.nottingham.ac.uk/personal/jec/`.

[CTT94] J. Chao, K. Tanada, and S. Tsujii, *Design of elliptic curves with controllable lower boundary of extension degree for reduction attacks*, Advances in Cryptology – CRYPTO '94 (Y. Desmedt, ed.), Lecture Notes in Comput. Sci., vol. 839, Springer-Verlag, 1994, pp. 50–55.

[DGM99] I. Duursma, P. Gaudry, and F. Morain, *Speeding up the discrete log computation on curves with automorphisms*, Advances in Cryptology – ASIACRYPT '99 (K.Y. Lam, E. Okamoto, and C. Xing, eds.), Lecture Notes in Comput. Sci., vol. 1716, Springer-Verlag, 1999, pp. 103–121.

[DH76] W. Diffie and M. E. Hellman, *New directions in cryptography*, IEEE Trans. Inform. Theory **IT-22-6** (1976), 644–654.

[DQ90] J.-P. Delescaille and J.-J. Quisquater, *How easy is collision search? Application to DES*, Advances in Cryptology – EUROCRYPT'89 (J.-J. Quisquater, ed.), Lecture Notes in Comput. Sci., vol. 434, 1990, pp. 429–434.

[DS00] I. Duursma and K. Sakurai, *Efficient algorithms for the jacobian variety of hyperelliptic curves* $y^2 = x^p - x + 1$ *over a finite field of odd characteristic* p, Proceedings of the "International Conference on Coding Theory, Cryptography and Related Areas", Springer-Verlag, 2000, Guanajuato, Mexico on April, 1998, pp. 73–89.

[DW98] T. Denny and D. Weber, *The solution of McCurley's discrete log challenge*, Advances in Cryptology – CRYPTO'98 (H. Krawczyk, ed.), Lecture Notes in Comput. Sci., vol. 1462, 1998, pp. 458–471.

[EG00] A. Enge and P. Gaudry, *A general framework for subexponential discrete logarithm algorithms*, Research Report LIX/RR/00/04, LIX, 2000.

[EK97] W. Eberly and E. Kaltofen, *On randomized Lanczos algorithms*, ISSAC '97 (Wolgang W. Küchlin, ed.), 1997, Proceedings of the International Symposium on Symbolic and Algebraic Computation, July 21, Maui, pp. 176–183.

[Elk98] N. Elkies, *Elliptic and modular curves over finite fields and related computational issues*, Computational Perspectives on Number Theory (D.A. Buell and J.T. Teitelbaum, eds.), AMS/International Press, 1998, Proceedings of a Conference in Honor of A.O.L. Atkin, pp. 21–76.

[Eng99] A. Enge, *Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time*, Combinatorics and Optimization Research Report CORR 99-04, University of Waterloo, February 1999, To appear in *Math. Comp.*; available at http://-cacr.math.uwaterloo.ca/techreports/1999/corr99-04.ps.

[ES00] A. Enge and A. Stein, *Smooth ideals in hyperelliptic function fields*, Combinatorics and Optimization Research Report CORR 2000-08, University of Waterloo, January 2000, Available at http://cacr.math.uwaterloo.ca/techreports/2000/corr2000-08.ps.

[FGH00] M. Fouquet, P. Gaudry, and R. Harley, *An extension of Satoh's algorithm and its implementation*, J. Ramanujan Math. Soc. **15** (2000), 281–318.

[FKP89] P. Flajolet, D. Knuth, and B. Pittel, *The first cycles in an evolving graph*, Discrete Math. **75** (1989), 167–215.

[Fly90] E.V. Flynn, *The Jacobian and formal group of a curve of genus 2 over an arbitrary ground field*, Math. Proc. Cambridge Philos. Soc. **107** (1990), 425–441.

[FM98] G. Frey and M. Müller, *Arithmetic of modular curves and applications*, Tech. Report 20, Institut für Experimentelle Mathematik, 1998.

[FMR98] G. Frey, M. Müller, and H.-G. Rück, *The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems*, Tech. Report 23, Institut für Experimentelle Mathematik, 1998.

[FP99] R. Flassenberg and S. Paulus, *Sieving in function fields*, Experiment. Math. **8** (1999), no. 4, 339–349.

[FR94] G. Frey and H.-G. Rück, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Math. Comp. **62** (1994), no. 206, 865–874.

[Fre83] E. Freitag, *Siegelsche Modulfunktionen*, Grundlehren der mathematischen Wissenschaften, vol. 254, Springer–Verlag, 1983.

[Fre98] G. Frey, *How to disguise an elliptic curve (Weil descent)*, Talk at Waterloo workshop ECC'98, 1998, `http://cacr.math.uwaterloo.ca/conferences/1998/ecc98/slides.html`.

[FS93] P. Flajolet and R. Sedgewick, *The average case analysis of algorithms : complex asymptotics and generating functions*, Research Report 2026, INRIA, 1993.

[FS96] P. Flajolet and R. Sedgewick, *An introduction to the analysis of algorithms*, Addison–Wesley, 1996.

[Ful69] W. Fulton, *Algebraic curves*, Math. Lec. Note Series, W. A. Benjamin Inc, 1969.

[Gau00] P. Gaudry, *An algorithm for solving the discrete log problem on hyperelliptic curves*, Advances in Cryptology – EUROCRYPT 2000 (B. Preneel, ed.), Lecture Notes in Comput. Sci., vol. 1807, Springer–Verlag, 2000, pp. 19–34.

[GH00] P. Gaudry and R. Harley, *Counting points on hyperelliptic curves over finite fields*, ANTS-IV (W. Bosma, ed.), Lecture Notes in Comput. Sci., vol. 1838, Springer–Verlag, 2000, pp. 313–332.

[GHS00] P. Gaudry, F. Hess, and N. Smart, *Constructive and destructive facets of Weil descent on elliptic curves*, To appear in J. Crypt., 2000.

[GK86] S. Goldwasser and J. Kilian, *Almost all primes can be quickly certified*, Proc. 18th STOC, ACM, 1986, May 28–30, Berkeley, pp. 316–329.

[GLV98] R. Gallant, R. Lambert, and S. Vanstone, *Improving the parallelized Pollard lambda search on binary anomalous curves*, `http://www.certicom.com/chal/download/paper.ps`, 1998.

[Got59] Gottschling, *Explizite Bestimmung der Ranflächen des Fundamentalbereiches der Modulgruppe zweiten Grades*, Math. Ann. **138** (1959), 103–124.

[GPS00] S. Galbraith, S. Paulus, and N. Smart, *Arithmetic on superelliptic curves*, Preprint, 2000.

[Gra96] T. Granlund, *The GNU Multiple Precision arithmetic library – 2.0.2*, GNU, 1996, distributed with the gmp package at `ftp://prep.ai.mit.edu/pub/gnu/gmp-M.N.tar.gz`.

[GSa] P. Gaudry and É. Schost, *Invariants des quotients de la jacobienne d'une courbe de genre 2*, Preprint. `http://www.lix.polytechnique.fr/Labo/Pierrick.Gaudry/jinvar.ps.gz`.

[GSb] P. Gaudry and É. Schost, *Modular equations in higher genus*, In preparation.

[GS99] S. Galbraith and N. Smart, *A cryptographic application of Weil descent*, Cryptography and Coding, 7th IMA Conference, Lecture Notes in Comput. Sci., vol. 1746, springer–Verlag, 1999, Full paper is HP-LABS Technical Report (Number HPL-1999-70), pp. 191–200.

[Hac96] G. Haché, *Construction effective de codes géométriques*, Ph.D. thesis, Université de Paris VI, 1996.

[Hara] R. Harley, *Fast arithmetic on genus 2 curves*,
Available at `http ://cristal.inria.fr/~harley/hyper/`.

[Harb] R. Harley, *Thèse de doctorat*, In preparation, Inria Rocquencourt.

[Har77] R. Hartshorne, *Algebraic geometry*, Graduate Texts in Mathematics, vol. 52, Springer–Verlag, 1977.

[HI91] M.-D. Huang and D. Ierardi, *Efficient algorithms for the Riemann-Roch problem and for addition in the jacobian of a curve*, 32nd annual symposium on foundations of computer science, IEEE, october 1991, pp. 678–687.

[HI98] M.-D. Huang and D. Ierardi, *Counting points on curves over finite fields*, J. Symbolic Comput. **25** (1998), 1–21.

[HLP00] E. Howe, F. Leprevost, and B. Poonen, *Large torsion subgroups of split Jacobians of curves of genus two or three*, Forum Math. **12** (2000), 315–364.

[HM89] J. L. Haffner and K. S. McCurley, *A rigorous subexponential algorithm for computation of class groups*, J. Amer. Math. Soc. **2** (1989), no. 4, 837–850.

[HSV89] J.-C. Hervé, B. Serpette, and J. Vuillemin, *BigNum: A portable and efficient package for arbitrary-precision arithmetic*, Tech. Report 2, Digital Paris Research Laboratory, May 1989.

[HW36] H. Hasse and E. Witt, *Zyklische unverzweigte Erweiterungskörper vom primzahlgrade p über einem algebraischen Funktionenkörper der Charakteristik p*, Monatsch. Math. Phys. **43** (1936), 477–492.

[Igu60] J. Igusa, *Arithmetic variety of moduli for genus two*, Ann. of Math. (2) **72** (1960), 612–649.

[Igu62] J. Igusa, *On Siegel modular forms of genus two*, Amer. J. Math. **84** (1962), 175–200.

[Kam91] W. Kampkötter, *Explizite Gleichungen für Jacobische Varietäten hyperelliptischer Kurven*, Ph.D. thesis, Universität Gesamthochschule Essen, August 1991.

[Kli90] H. Klingen, *Introductory lectures on Siegel modular forms*, Cambridge studies in advanced mathematics, vol. 20, Cambridge University Press, 1990.

[Kna92] A. Knapp, *Elliptic curves*, Princeton University Press, 1992.

[Kno75] J. Knopfmacher, *Abstract analytic number theory*, North-Holland Mathematical Library, vol. 12, North-Holland Publishing Company, Amsterdam, 1975.

[Knu98] D. E. Knuth, *The Art of Computer Programming: Seminumerical algorithms*, 3rd ed., Addison-Wesley, 1998.

[Kob87] N. Koblitz, *Elliptic curve cryptosystems*, Math. Comp. **48** (1987), no. 177, 203–209.

[Kob89] N. Koblitz, *Hyperelliptic cryptosystems*, J. of Cryptology **1** (1989), 139–150.

[Kob90] N. Koblitz, *A family of jacobians suitable for discrete log cryptosystems*, Advances in Cryptology – CRYPTO '88 (S. Goldwasser, ed.), Lecture Notes in Comput. Sci., vol. 403, Springer–Verlag, 1990, pp. 94–99.

[Kob92] N. Koblitz, *CM-curves with good cryptographic properties*, Advances in Cryptology – CRYPTO '91 (Joan Feigenbaum, ed.), Lecture Notes in Comput. Sci., vol. 576, Springer-Verlag, 1992, pp. 279–287.

[Kob97] N. Koblitz, *A very easy way to generate curves over prime fields for hyperelliptic cryptosystems*, Rump-session Crypto'97, 1997.

[Kob98] N. Koblitz, *Algebraic aspects of cryptography*, Algorithms and Computation in Mathematics, vol. 3, Springer–Verlag, 1998.

[KR89] E. Kani and M. Rosen, *Idempotent relations and factors of Jacobians*, Math. Ann. **298** (1989), 307–327.

[KS91] E. Kaltofen and B. D. Saunders, *On Wiedemann's method of solving sparse linear systems*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (Berlin) (H. F. Mattson, T. Mora, and T. R. N. Rao, eds.), Lecture Notes in Comput. Sci., vol. 539, Springer-Verlag, 1991, pp. 29–38.

[Kul87] R. Kulkarni, *Symmetries of surfaces*, Topology **26** (1987), no. 2, 195–203.

[Kul95] L. Kulesz, *Courbes algébriques de genre 2 possédant de nombreux points rationnels*, C. R. Acad. Sci. Paris Sér. I Math. **321** (1995), 91–94.

[Kul99] L. Kulesz, *Application de la méthode de Dem'janenko-Manin à certaines familles de courbes de genre 2 et 3*, J. Number Theory **76** (1999), 130–146.

[Lan] T. Lange, *Efficient arithmetic on hyperelliptic Koblitz curves*, In preparation.

[Lan59] S. Lang, *Abelian varieties*, Interscience Tracts in Pure and Applied Mathematics, no. 7, Interscience Publishers, 1959.

[Lan76] H. Lange, *Über die Modulvarietät der Kurven vom Geschlecht 2*, J. Reine Angew. Math. **281** (1976), 80–96.

[Lec99] G. Lecerf, *Kronecker, polynomial equation system solver, reference manual*, 1999, `http://www.gage.polytechnique.fr/~lecerf/software/kronecker`.

[Len87] H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (2) **126** (1987), 649–673.

[Ler97] R. Lercier, *Algorithmique des courbes elliptiques dans les corps finis*, Thèse, École polytechnique, June 1997.

[LM97] F. Leprévost and F. Morain, *Revêtements de courbes elliptiques à multiplication complexe par des courbes hyperelliptiques et sommes de caractères*, J. Number Theory **64** (1997), 165–182.

[LO90] B. A. LaMacchia and A. M. Odlyzko, *Solving large sparse linear systems over finite fields*, Advances in Cryptology – CRYPTO '90 (A. J. Menezes and S. A. Vanstone, eds.), Lecture Notes in Comput. Sci., vol. 537, Springer–Verlag, 1990, pp. 109–133.

[LP92] H. W. Lenstra Jr. and C. Pomerance, *A rigorous time bound for factoring integers*, J. Amer. Math. Soc. **5** (1992), no. 3, 483–516.

[LP98] R. Lovorn Bender and C. Pomerance, *Rigorous discrete logarithm computations in finite fields via smooth polynomials*, Computational Perspectives on Number Theory : Proceedings of a Conference in Honor of A. O. L. Atkin (D. A. Buell and J. T. Teitelbaum, eds.), AMS/IP Studies in Advanced Mathematics, vol. 7, American Mathematical Society, International Press, 1998, pp. 221–232.

[LPP93] H. W. Lenstra, Jr., J. Pila, and C. Pomerance, *A hyperelliptic smoothness test, I*, Philos. Trans. Roy. Soc. London Ser. A **345** (1993), 397–408.

[LZ94] G.-J. Lay and H. G. Zimmer, *Constructing elliptic curves with given group order over large finite fields*, ANTS-I (L. Adleman and M.-D. Huang, eds.), Lecture Notes in Comput. Sci., vol. 877, Springer-Verlag, 1994, pp. 250–263.

[Maa71] Maaß, *Siegel modular forms and Dirichlet series*, Lecture Notes in Math., vol. 216, Springer–Verlag, 1971.

[Maa78] Maaß, *Lineare Relationen für die Fourierkoeffizienten einiger Modulformen zweiten Grades*, Math. Ann. **232** (1978), 163–175.

[Man65] J. I. Manin, *The Hasse-Witt matrix of an algebraic curve*, Trans. Amer. Math. Soc. **45** (1965), 245–264.

[Mau94] U. M. Maurer, *Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms*, Advances in Cryptology – CRYPTO '94 (Y. G. Desmedt, ed.), Lecture Notes in Comput. Sci., vol. 839, Springer-Verlag, 1994, pp. 271–281.

[McE95] R. McEliece, *Finite fields for computer scientists and engineers*, Kluwer Academic Publishers, 1995, Second Printing.

[Men93] Alfred J. Menezes, *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 1993.

[Mes91a] J.-F. Mestre, *Construction de courbes de genre 2 à partir de leurs modules*, Effective methods in algebraic geometry (T. Mora and C. Traverso, eds.), Progr. Math., vol. 94, Birkhäuser, 1991, Proc. Congress in Livorno, Italy, April 17–21, 1990, pp. 313–334.

[Mes91b] J.-F. Mestre, *Familles de courbes hyperelliptiques à multiplications réelles*, Arithmetic Algebraic Geometry, Progr. Math., vol. 89, 1991, pp. 193–208.

[Mil86] J. S. Milne, *Jacobian varieties*, Arithmetic Geometry (G. Cornell and J. H. Silverman, eds.), Springer–Verlag, 1986, pp. 167–212.

[Mil87] V. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptology – CRYPTO '86 (A. M. Odlyzko, ed.), Lecture Notes in Comput. Sci., vol. 263, Springer-Verlag, 1987, pp. 417–426.

[Miy93] A. Miyaji, *Elliptic curves over F_p suitable for cryptosystems*, Advances in Cryptology – AUSCRYPT '92 (J. Seberry and Y. Zheng, eds.), Lecture Notes in Comput. Sci., vol. 718, Springer-Verlag, 1993, pp. 479–491.

[Mor91] F. Morain, *Building cyclic elliptic curves modulo large primes*, Advances in Cryptology – EUROCRYPT '91 (D. Davies, ed.), Lecture Notes in Comput. Sci., vol. 547, Springer–Verlag, 1991, pp. 328–336.

[MOV93] A. Menezes, T. Okamoto, and S. A. Vanstone, *Reducing elliptic curves logarithms to logarithms in a finite field*, IEEE Trans. Inform. Theory **39** (1993), no. 5, 1639–1646.

[MST99] V. Müller, A. Stein, and C. Thiel, *Computing discrete logarithms in real quadratic congruence function fields of large genus*, Math. Comp. **68** (1999), no. 226, 807–822.

[Mum83] D. Mumford, *Tata lectures on theta I*, Progr. Math., vol. 28, Birkhauser, 1983.

[Mum84] D. Mumford, *Tata lectures on theta II*, Progr. Math., vol. 43, Birkhauser, 1984.

[Mum91] D. Mumford, *Tata lectures on theta III*, Progr. Math., vol. 97, Birkhauser, 1991.

[MW96] U. M. Maurer and S. Wolf, *Diffie-Hellman oracles*, Advances in Cryptology – CRYPTO '96 (N. Koblitz, ed.), Lecture Notes in Comput. Sci., vol. 1109, Springer-Verlag, 1996, pp. 268–282.

[Nag00] K. Nagao, *Improving group law algorithms for Jacobians of hyperelliptic curves*, ANTS-IV (W. Bosma, ed.), Lecture Notes in Comput. Sci., vol. 1838, Springer–Verlag, 2000, pp. 439–447.

[Nak87] S. Nakajima, *p-ranks and automorphism groups of algebraic curves*, Trans. Amer. Math. Soc. **303** (1987), no. 2, 595–607.

[Nec94] V. Nechaev, *Complexity of a determinate algorithm for the discrete logarithm*, Mathematical Notes **55** (1994), no. 2, 165–172.

[OS92] T. Okamoto and K. Sakurai, *Efficient algorithms for the construction of hyperelliptic cryptosystems*, Advances in Cryptology – CRYPTO '91 (J. Feigenbaum, ed.), Lecture Notes in Comput. Sci., vol. 576, Springer–Verlag, 1992, pp. 267–278.

[PH78] S. Pohlig and M. Hellman, *An improved algorithm for computing logarithms over* GF(p) *and its cryptographic significance*, IEEE Trans. Inform. Theory **IT–24** (1978), 106–110.

[Pil90] J. Pila, *Frobenius maps of abelian varieties and finding roots of unity in finite fields*, Math. Comp. **55** (1990), no. 192, 745–763.

[Pol74] J. M. Pollard, *Theorems on factorization and primality testing*, Math. Proc. Cambridge Philos. Soc. **76** (1974), 521–528.

[Pol78] J. M. Pollard, *Monte Carlo methods for index computation* mod p, Math. Comp. **32** (1978), no. 143, 918–924.

[Pom87] C. Pomerance, *Fast, rigorous factorization and discrete logarithm algorithms*, Discrete Algorithms and Complexity, Proceedings of the Japan–US Joint Seminar, June 4–6, 1986, Kyoto, Japan (Orlando) (D. S. Johnson, T. Nishizeki, A. Nozaki, and H. S. Wolf, eds.), Perspectives in Computing, Academic Press, 1987, pp. 119–143.

[PZ89] M. Pohst and H. Zassenhaus, *Algorithmic algebraic number theory*, Cambridge Univ. Press, 1989.

[Rei86] M. Reichert, *Explicit determination of nontrivial torsion structures of elliptic curves over quadratic number fields*, Math. Comp. **46** (1986), no. 174, 637–658.

[Roq70] P. Roquette, *Abschätzung der Automorphismenanzahl von Funktionenkörpern bei Primzahlcharacteristik*, Math. Z. **117** (1970), 157–163.

[RS62] J. B. Rosser and L. Schoenfeld, *Approximate formulas for some functions of prime numbers*, Illinois J. Math. **6** (1962), 64–94.

[Rüc99] H. G. Rück, *On the discrete logarithm in the divisor class group of curves*, Math. Comp. **68** (1999), no. 226, 805–806.

[SA98] T. Satoh and K. Araki, *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves*, Comment. Math. Helv. **47** (1998), no. 1, 81–92.

[Sat00] T. Satoh, *The canonical lift of an ordinary elliptic curve over a finite field and its point counting*, J. Ramanujan Math. Soc. **15** (2000), 247–270.

[Sch] É. Schost, *Computing parametric geometric resolutions*, Preprint.
Avaliable at `http://www.medicis.polytechnique.fr/gage/notes/2000.html`.

[Sch85] R. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Math. Comp. **44** (1985), 483–494.

[Sch95] R. Schoof, *Counting points on elliptic curves over finite fields*, J. Théor. Nombres Bordeaux **7** (1995), 219–254.

[Sch00] É. Schost, *Sur la résolution des systèmes polynomiaux à paramètres*, Ph.D. thesis, École polytechnique, 2000.

[Sed88] R. Sedgewick, *Algorithms*, second ed., Addison–Wesley, 1988.

[Sem95] I. A. Semaev, *Computation of discrete logarithms in an arbitrary finite field*, Discrete Math. Appl. **5** (1995), no. 2, 107–116.

[Sem98] I. A. Semaev, *Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curves in characteristic p*, Math. Comp. **67** (1998), no. 221, 353–356.

[Sey87] M. Seysen, *A probabilistic factorization algorithm with quadratic forms of negative discriminant*, Math. Comp. **48** (1987), no. 178, 757–780.

[Sha71] D. Shanks, *Class number, a theory of factorization, and genera*, Proc. Symp. Pure Math. vol. 20, AMS, 1971, pp. 415–440.

[Shi68] G. Shimura, *Automorphic functions and number theory*, Lecture Notes in Math., vol. 54, Springer–Verlag, 1968.

[Shi71] G. Shimura, *Introduction to the theory of automorphic functions*, Princeton, 1971.

[Sho95] V. Shoup, *A new polynomial factorization algorithm and its implementation*, J. Symbolic Comput. **20** (1995), 363–397.

[Sho97] V. Shoup, *Lower bounds for discrete logarithms and related problems*, Advances in Cryptology – EUROCRYPT '97 (W. Fumy, ed.), Lecture Notes in Comput. Sci., vol. 1233, Springer–Verlag, 1997, pp. 256–266.

[Sie35] C. Siegel, *Über die Classenzahl quadratischer Zahlkörper*, Acta Arith. **1** (1935), 83–86.

[Sie55] C. Siegel, *Zur Theorie der Modulfunktionen n-ten Grades*, Communications on Pure and Applied Mathematics **8** (1955), 677–681.

[Sie73] C. Siegel, *Automorphic functions and abelian integrals*, Topics in complex function theory III, Wiley-Interscience, 1973.

[Sil86] J. H. Silverman, *The arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 106, Springer–Verlag, 1986.

[Sil94] J. H. Silverman, *Advanced topics in the arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 151, Springer–Verlag, 1994.

[Sma99] N. Smart, *The discrete logarithm problem on elliptic curves of trace one*, J. of Cryptology **12** (1999), no. 3, 193–196.

[Sma00] N. Smart, *How secure are elliptic curves over composite extension fields ?*, Tech. Report CSTR-00-017, University of Bristol, 2000.

[Spa94] A.-M. Spallek, *Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen*, Ph.D. thesis, Universität Gesamthochschule Essen, July 1994.

[SS85] J. Sattler and C. Schnorr, *Genrating random walks in groups*, Ann. Univ. Sci. Budapest. Sect. Comput. **6** (1985), 65–79.

[SS98] Y. Sakai and K. Sakurai, *Design of hyperelliptic cryptosystems in small charatcteristic and a software implementation over* $\mathbb{F}_{2^n}$, Advances in Cryptology – ASIACRYPT '98 (K. Ohta and D. Pei, eds.), Lecture Notes in Comput. Sci., vol. 1514, Springer–Verlag, 1998, pp. 80–94.

[ST61] G. Shimura and Y. Taniyama, *Complex multiplication of abelian varieties and its applications to number theory*, Publ. Math. Soc. Japan, vol. 6, Math. Soc. Japan, 1961.

[ST99a] A. Stein and E. Teske, *Catching kangaroos in function fields*, Proc. of The Mathematics of Public Key Cryptography, Toronto, June 1999.

[ST99b] A. Stein and E. Teske, *Explicit bounds and heuristics on class numbers in hyperelliptic function fields*, Tech. Report CORR 1999-26, Department of Combinatorics and Optimization, University of Waterloo, 1999.

[ST00a] A. Stein and E. Teske, *Explicit bounds and heuristics on class numbers in hyperelliptic function fields*, Preprint, 2000.

[ST00b] A. Stein and E. Teske, *The parallelized Pollard's kangaroo method in real quadratic function fields*, Tech. Report CORR 2000-35, Department of Combinatorics and Optimization, University of Waterloo, 2000.

[Sti73] H. Stichtenoth, *Über die Automorphismengruppe eienes algebraischen Funktionenkörpers von Primzahlcharacteristik. Teil I*, Arch. Math. **24** (1973), 527–544.

[Sti79] H. Stichtenoth, *Die Hasse-Witt-Invariante eines Kongruenzfunktionenkörpers*, Arch. Math. **33** (1979), 357–360.

[Sti93] H. Stichtenoth, *Algebraic function fields and codes*, Springer–Verlag, 1993.

[Str76] V. Strassen, *Einige Resultate über Berechnungskomplexität*, Jahresbericht der Deutschen Mathematiker-Vereinigung **78** (1976), 1–8.

[SW99] A. Stein and H. Williams, *Some methods for evaluating the regulator of a real quadratic function field*, Experiment. Math. **8** (1999), no. 2, 119–133.

[Swa62] R. G. Swan, *Factorization of polynomials over finite fields*, Pacific J. Math. **12** (1962), 1099–1106.

[Tat66] J. Tate, *Endomorphisms of Abelian varieties over finite fields*, Invent. Math. **2** (1966), 134–144.

[Tes98] E. Teske, *Speeding up Pollard's rho method for computing discrete logarithms*, ANTS-III (J. P. Buhler, ed.), Lecture Notes in Comput. Sci., vol. 1423, Springer–Verlag, 1998, pp. 541–554.

[Tho] É. Thome, *Improving the sequential stage of block Wiedemann algorithm*, Preprint.

[Vél71] J. Vélu, *Isogénies entre courbes elliptiques*, C. R. Acad. Sci. Paris Sér. I Math. **273** (1971), 238–241, Série A.

[vOW99] P. C. van Oorschot and M. J. Wiener, *Parallel collision search with cryptanalytic applications*, J. of Cryptology **12** (1999), 1–28.

[vW99] P. van Wamelen, *Examples of genus two CM curves defined over the rationals*, Math. Comp. **68** (1999), no. 225, 307–320.

[vzGG99] J. von zur Gathen and J. Gerhard, *Modern computer algebra*, Cambridge University Press, 1999.

[Wan95] X. Wang, *2-dimensional simple factors of $J_0(N)$*, Manuscripta Math. **87** (1995), no. 2, 179–197.

[Web97] H.-J. Weber, *Hyperelliptic simple factors of $J_0(N)$ with dimension at least 3*, Experiment. Math. **6** (1997), no. 4, 273–287.

[Wie86] D. H. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory **IT–32** (1986), no. 1, 54–62.

[WZ99] M. J. Wiener and R. J. Zuccherato, *Faster attacks on elliptic curve cryptosystems*, Selected Areas in Cryptography '98 (S. Tavares and H. Meijer, eds.), Lecture Notes in Comput. Sci., vol. 1556, Springer-Verlag, 1999.

[Yui78] N. Yui, *On the jacobian varietes of hyperelliptic curves over fields of characteristic $p > 2$*, J. Algebra **52** (1978), 378–410.

[Zhu97] H. Zhu, *Supersingular abelian vaieties over finite fields*, Ph.D. thesis, University of California at Berkeley, 1997.

Résumé

L'étude algorithmique des courbes hyperelliptiques est la suite naturelle de celle des courbes elliptiques qui est maintenant bien avancée. La plupart des algorithmes connus pour les courbes elliptiques ainsi que leurs applications à la cryptographie peuvent être étendus plus ou moins facilement aux Jacobiennes de courbes hyperelliptiques.

Dans une première partie, nous étudions certains aspects des invariants d'Igusa, qui généralisent le j-invariant d'une courbe elliptique. Pour les Jacobiennes $(2,2)$-décomposables, nous relions les invariants d'Igusa aux j-invariants des courbes elliptiques quotients par des formules explicites. Par ailleurs nous étudions ces invariants sous l'angle des formes modulaires de Siegel dans le but de calculer des équations modulaires.

La deuxième partie est consacrée à des algorithmes de calcul de cardinalité d'une courbe hyperelliptique sur un corps fini. Ce calcul est une étape nécessaire lorsque l'on désire mettre en œuvre un cryptosystème hyperelliptique. Hormis les algorithmes génériques qui peuvent s'appliquer à des groupes autres que des Jacobiennes, nous proposons une version effective des algorithmes à la Schoof en genre 2. Nous présentons aussi un premier pas vers des améliorations du type Elkies-Atkin, qui ont fait leur preuve dans le cas des courbes elliptiques.

La troisième partie traite d'algorithmes de calcul de logarithme discret. Ce problème, réputé difficile, est la clef de voûte des cryptosystèmes : si l'on sait le résoudre en temps raisonnable, le système est fragile. Après un bref état de l'art, nous présentons des algorithmes utilisant les idées classiques de calcul d'index. En tirant parti des spécificités des problèmes provenant de la cryptographie, nous démontrons par des résultats de complexité ainsi que des expériences pratiques que les systèmes à base de courbes de genre supérieur ou égal à 4 ne sont pas sûrs. De plus, combiné avec les techniques de descente de Weil, ceci permet d'attaquer certains cryptosystèmes elliptiques.

Mots-clés: Courbes hyperelliptiques, invariants d'Igusa, algorithme de Schoof en genre 2, logarithme discret, calcul d'index, équations modulaires.

Abstract

The study of algorithmical aspects of hyperelliptic curves is the natural continuation of the case of elliptic curves, which is now well advanced. Most of the algorithms known for elliptic curves and their applications to cryptography can be more or less easily extended to Jacobians of hyperelliptic curves.

In a first part, we investigate some aspects of Igusa's invariants which generalize the j-invariant of elliptic curves. For $(2,2)$-reducible Jacobians, we relate by explicit formulae the Igusa's invariants to the j-invariants of the quotient elliptic curves. Besides, we study these invariants by the way of Siegel modular forms with a view toward computing modular equations.

The second part is dedicated to algorithms for computing the cardinality of a hyperelliptic curve over a finite field. This computation is the first step when one want to use these curves for cryptography. Beside generic algorithms which can be applied to other groups than Jacobians, we propose an effective version of *à la* Schoof algorithms in genus 2. We present also a first step toward an Elkies-Atkin approach, which has proven to be successful in the elliptic case.

The third part deals with algorithms for discrete logarithm computations. The security of some cryptosystems relies on this problem, which is considered to be difficult: if one can solve it in a reasonable amount of time, then the system is weak. After a brief state of the art, we present some algorithms based on the classical ideas of index-calculus. Taking advantage of the particularities of the problems coming from cryptography, we demonstrate with complexity results and practical experiments that the systems based on curves of genus greater or equal to 4 are not secure. Moreover, when combined with the techniques of Weil descent, this allows to attack some elliptic cryptosystems.

Keywords: Hyperelliptic curves, Igusa invariants, genus 2 Schoof's algorithm, discrete logarithm, index-calculus, modular equations.