UNIVERSITÉ DE LIMOGES
ÉCOLE DOCTORALE Science - Technologie - Santé
Faculté des Sciences et Techniques
Département de Mathématiques, Laboratoire XLIM

Année 2008 Thèse N°46 – 2008

Thèse
pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ DE LIMOGES
Discipline : Mathématiques et ses applications
soutenue et présentée par

Adrien POTEAUX
le 15 octobre 2008.

Calcul de développements de Puiseux et application au calcul du groupe de monodromie d'une courbe algébrique plane.

Thèse codirigée par **Moulay BARKATOU** et **Marc RYBOWICZ**

Jury

Président	
André GALLIGO	Professeur.
Rapporteurs	
André GALLIGO	Professeur.
Grégoire LECERF	Chargé de recherche CNRS.
Mark van HOEIJ	Professeur.
Examinateurs	
Laureano Gonzalez-Vega	Professeur .
Claude Quitté	Maître de conférences.
Moulay Barkatou	Professeur (directeur de thèse).
Marc Rybowicz	Maître de conférences (directeur de thèse).

Table des matières

Table des figures

Remerciements

Je remercie tout d'abord Marc Rybowicz pour m'avoir proposé ce sujet de thèse, et surtout pour m'avoir accompagné durant ces années de thèse. Sans lui ce travail n'aurait pu voir le jour, et son investissement dans ce travail a été d'une grande aide. Les mots sur cette page ne suffiront à exprimer ma gratitude.

Je tiens également à remercier Moulay Barkatou pour avoir encadré cette thèse, ainsi que pour ses conseils au fur et a mesure des années.

Mark van Hoeij a initié la motivation de ces travaux de thèse, de part ses travaux avec Bernard Deconinck. Il a de plus suivi mes travaux, apportant parfois de précieux conseils. Pour toutes ces choses, mais également pour avoir accepté de rapporter mes travaux, je l'en remercie.

André Galligo s'est intéressé à mes travaux avec enthousiasme, et a ensuite accepté la charge de rapporteur. Merci à lui.

Je remercie également Grégoire Lecerf pour avoir effectué une relecture minutieuse de cette thèse, ainsi que pour les longues discussions scientifiques que nous avons pu avoir à ce sujet.

Claude Quitté et Lauréano Gonzalez Vega m'ont fait l'honneur de participer à mon jury, je les en remercie.

Jacques-Arthur Weil a encadré mes enseignements en tant que tuteur durant mon monitorat. Il a notamment partagé le cours de géométrie analytique, me permettant de faire mon premier cours magistral lors de mon année d'ATER. Cela aura été une expérience très enrichissante. Je le remercie donc pour cela, mais aussi pour ses conseils qu'il aura pu me prodiguer durant cette thèse, et notamment les différentes répétitions d'exposés auxquelles il aura pu participer.

Je remercie également l'ensemble des enseignants avec qui j'ai pu partager les cours tout au long de cette année, et notamment Anne Bellido, Pascale Sénéchaud, Thierry Berger, Guilhem Castagnos et Stéphane Vinatier.

Lors des voyages effectués au fur et à mesure de cette thèse, j'ai été accueilli par différentes personnes et différents laboratoires. Je tiens ici à les remercier, et plus particulièrement l'équipe Algo, Delphine Boucher, Greg Reid et Rob Corless lors de mon passage à London, et dernièrement l'équipe Galaad et le laboratoire Dieudonné. Je remercie également Olivier Ruatta, Thomas Cluzeau, Joris van der Hoeven, Marc Mezzarobba et Eckhard Pfluegel pour les discussions scientifiques que nous avons pu avoir.

Mes remerciements vont également aux secrétaires du département de Mathématiques, dont l'aide est précieuse pour l'ensemble des démarches administratives. Merci donc à Yolande, Sylvie, Patricia, et plus récemment Aurélie.

Enfin, avant de quitter le monde du travail pour celui des loisirs, je tiens à remercier l'ensemble des doctorants avec qui j'ai partagé quelques moments de détente, que ce soit dans le cadre de l'ADDMUL ou dans un cadre plus privé. Merci donc à Samuel, Nicolas, Laurent, Romain, Sandrine, Daouda, Elsa, Christophe, Pierre-Louis, Julien, Aurore, et également à Morgan.

Outre le travail qu'a représentée cette thèse, cela aura également été une longue tranche de vie durant laquelle j'ai connu de nombreux moments forts. Je tiens tout d'abord à remercier l'ensemble de ma famille, tout simplement d'exister et de m'accompagner au fur et à mesure de ma vie. C'est toujours un grand plaisir d'aller s'y ressourcer. Le 27 août 2006 aura été un jour crucial de ma vie, me permettant de retrouver une part de moi. La folle épopée de la Toussaint qui l'a suivi l'aura été tout autant. Mille mercis Agathe de l'avoir partagée avec moi. Je tiens également à remercier tout particulièrement Blandine pour m'avoir grandement aidé dans ce processus de reconstruction. Merci aussi à Lydie et Jean Claude pour m'accueillir dans chacune de mes virées familiale. Merci à la fratrie Maurice, Isabelle et Jean Claude, Blandine et Dominique, Olivier et Maïté pour leur accueil occasionnel.

De l'autre côté de la famille, je remercie Claire et Jean Luc pour m'avoir toujours accueilli chez eux comme chez moi, et Julie (et Jérémy) d'avoir ensuite pris le relais. Merci aussi à Séverine pour les discussions que nous avons pu avoir. Et merci à toute la clique « du méchoui » pour m'avoir toujours accueilli de manière si chaleureuse.

Enfin, merci à mes parents pour me soutenir chacun à leur façon.

Ces années de thèse m'auront également valu de nombreuseus rencontres. Merci à l'ensemble des membres de la Citadelle du Jeu pour toutes ces soirées et tous ces week end de détente que nous avons pu partager. Un merci tout particulier à ceux qui sont devenus mes amis et qui m'ont ainsi supporté tout

au long de la thèse : merci donc à Abel, Alexandre (et Coralie) et Cécile. Merci aussi à tous ceux qui se sont investis activement dans cette association pour lui permettre de vivre.

Merci aussi à ceux qui *ensemble, font avancer le jeu*, et tout particulièrement à Christian, JBeuh, Émilie, Kévin, Olivier et Claude pour le soutien, les discussions, l'accueil et tous ces moments super que l'on a pu avoir.

Et avant de conclure ces remerciements, je tiens à remercier Tania pour tout ce qu'elle m'aura apporté.

La mort a frappé deux fois durant ma thèse. Manuel Bronstein fut celui qui m'a fait découvrir le monde de la recherche lors de mon stage de DEA. C'est ce stage qui m'a donné goût à ce métier, et quelque part ce travail n'aurait pas existé sans son apport. Je le remercie pour tout ce qu'il m'a apporté. Ma tante Claire a elle été emportée par la loterie du cancer. Elle a toujours été là pour moi, notamment dans les moments difficiles de ma vie. J'ai toujours été accueilli chez elle comme chez moi. Ces deux évènements m'ont profondément marqué, et je tiens à leur dédier cette thèse.

Enfin, je finirai par un mot à l'attention de mon oncle Jean Claude et de ma mère. La loterie du cancer vous frappe maintenant. J'espère que vous passerez cette épreuve et je pense fort à vous.

Introduction

Notons K un corps de nombres algébriques, c'est-à-dire une extension finie de $\mathbb{Q}$, et considérons une courbe algébrique plane de $\mathbb{C}^2$:

$$\mathcal{C} = \{(x_0, y_0) \in \mathbb{C}^2 \mid F(x_0, y_0) = 0\}$$

où $F \in K[x, y]$ est un polynôme sans facteur carré.

Dans [DvH01], Mark van Hoeij et Bernard Deconinck décrivent le calcul de la matrice des périodes d'une courbe algébrique plane, qui est une première étape pour obtenir une version effective du célèbre théorème d'Abel-Jacobi (voir par exemple [Mir95] ou [For81, section 20]). Ce théorème permet de répondre à des questions telles que « Un diviseur de degré zéro est-il un diviseur de fonction », qui est importante en calcul formel. Par exemple, elle apparaît lorsque l'on calcule la primitive d'une fonction algébrique, et notamment dans la partie logarithmique de cette primitive [Ris69, Dav81, Tra84, Bro90, Ber95]. Elle est aussi utile dans l'étude des solutions algébriques d'équations différentielles ordinaires [BD79], ou pour le calcul du groupe de Galois différentiel [CS98]. De plus, l'application d'Abel a aussi des applications en physique, notamment pour construire des solutions particulières des équations KdV, KP et NLS [DS98, DP07, Pat07].

Pour calculer une base canonique de l'homologie de la surface de Riemann (nécessaire au calcul des périodes), la méthode proposée dans [DvH01] utilise un résultat de Tretkoff et Tretkoff [TT84], qui réduit la construction de cette base au calcul du groupe de monodromie du revêtement associé à la courbe $\mathcal{C}$. Néanmoins, leur algorithme de calcul de monodromie (paquetage `algcurves`, commande `monodromy` de Maple) ne s'avère pas complètement fiable, et nécessite parfois une intervention humaine pour terminer (voir la partie 3.2.1). Par exemple, si l'on considère le polynôme $F(x, y) = y^4 - 200\,y^2 + 40\,y - 2 - x$, l'algorithme `monodromy` nécessite 60 chiffres de précision pour rendre un résultat. Tout appel à cette fonction effectué avec une précision moindre renvoie ainsi un message d'erreur. Néanmoins, nous verrons qu'il est possible de calculer le groupe de monodromie de ce polynôme en utilisant seulement 10

chiffres de précision.

Pour résoudre ces problèmes, nous proposons un nouvel algorithme pour calculer le groupe de monodromie d'une courbe algébrique plane. Contrairement à l'approche proposée dans [DvH01], nous n'avons pas cherché à éviter les points critiques, mais au contraire nous utilisons une méthode mixte symbolique et numérique consistant notamment à calculer les développements de Puiseux au-dessus des points critiques. Malheureusement, calculer symboliquement les développements de Puiseux par l'algorithme classique de Newton-Puiseux peut être coûteux, même sur des exemples simples. De plus, les résultats peuvent contenir des nombres rationnels de grande taille, qui nécessitent donc une précision importante si l'on veut les évaluer numériquement. Par exemple, le polynôme de degré 6 en y, $F(x,y) = (y^3 - x)((y - 1)^2 - x)(y - 2 - x^2) + x^2y^5$, a pour discriminant en y $\Delta_F(x) = x^3P(x)$, où $P(x)$ est un polynôme irréductible sur $\mathbb{Q}$ de degré 23. Si l'on souhaite calculer symboliquement les développements de Puiseux au-dessus des racines de ce polynôme, on doit donc travailler dans une extension de degré 23. De plus, le premier terme de la série de Puiseux ainsi calculé contient des fractions rationnelles de 136 chiffres. P.G. Walsh a montré que, pour tout $\epsilon > 0$, la partie singulière des développements de Puiseux peut être calculée en $O(d_y^{32+\epsilon} d_x^{4+\epsilon} (\log h)^{2+\epsilon})$ opérations binaires [Wal00], où d_y et d_x sont respectivement les degrés en y et x du polynôme F, et h est la hauteur de F. Même si cette borne n'est probablement pas fine, elle n'est pas encourageante et confirme les observations expérimentales. Malheureusement, une utilisation numérique de l'algorithme de Newton-Puiseux pour calculer les développements de Puiseux (voir le chapitre 2) au-dessus d'un point critique n'est pas simple. En effet, si le point critique est remplacé par une approximation, l'algorithme retourne des séries approchées ayant un rayon de convergence très petit, et surtout ne contenant pas certaines informations importantes, telles que les indices de ramifications.

Pour éviter ces calculs symboliques, nous introduisons une nouvelle approche symbolique-numérique : l'information exacte est obtenue par des calculs modulo un nombre premier p « bien choisi », de telle sorte que la structure des solutions soit préservée par réduction modulaire ; ensuite, on utilise la structure des solutions pour guider le calcul numérique des séries de Puiseux. Ainsi, la taille des coefficients est contrôlée, et les instabilités numériques sont réduites. Et surtout, les informations exactes importantes (notamment pour le calcul du groupe de monodromie), telles que les indices de ramifications et les multiplicités d'intersection des branches, sont conservées.

Cette thèse est divisée de la manière suivante :

Dans le **chapitre 1**, nous rappelons les principaux résultats sur les courbes algébriques planes dont nous aurons besoin par la suite.

Dans le **chapitre 2**, nous détaillons un algorithme symbolique-numérique pour calculer les développements de Puiseux. Ce chapitre présente plusieurs contributions :

- On introduit les notions de « polygone de Newton générique » et d' « arbre de polygones ». La seconde notion contient l'ensemble les informations exactes dont nous aurons besoin pour guider les calculs numériques. La première notion permet de simplifier les explications et les preuves, notamment pour ce qui concerne la réduction modulaire.
- On étudie la réduction modulaire des développements de Puiseux, ce qui nous amène à un critère de bonne réduction pour le choix d'un « bon premier » p, qui nous permet d'obtenir l'arbre de polygones en utilisant une arithmétique modulaire. On décrit de plus des algorithmes probabilistes nous permettant d'obtenir un tel premier p de petite taille, c'est-à-dire logarithmique en la taille des entrées.
- On améliore les bornes connues sur le nombre d'opérations modulaires à effectuer pour calculer l'arbre des polygones en un point critique, puis en l'ensemble des points critiques. À l'aide de l'étude de complexité des algorithmes probabilistes pour trouver le premier p, on obtient une complexité sur le nombre d'opérations binaires pour calculer l'arbre des polygones au-dessus de l'ensemble des points critiques.
- On explique comment utiliser l'information donnée par l'arbre de polygones pour pouvoir calculer une approximation numérique des développements de Puiseux.

Le **chapitre 3** décrit la stratégie que nous utilisons pour calculer le groupe de monodromie de la courbe $\mathcal{C}$. Celle-ci repose sur trois principes :

- Nous utilisons un arbre de recouvrement minimum pour réduire la longueur totale des chemins que nous aurons à suivre. Nous proposons une méthode pour obtenir des chemins suivant cet arbre qui soient homotopes à ceux utilisés classiquement, comme dans [TT84].
- Nous relions les fibres entre deux points intermédiaires à l'aide de développements en série tronqués à un ordre contrôlé. Nous donnons notamment des bornes sur les ordres de troncation afin d'avoir des connexions fiables. De plus, nous calculons de tels développements au-dessus de points réguliers, mais aussi au-dessus des points critiques. Ces derniers décrivent les fonctions lorsqu'elles sont prolongées le long d'un arc de cercle autour d'un point critique, et nous donnent la monodromie locale.
- Enfin, nous étudions la complexité du prolongement analytique pour

obtenir un compromis entre les ordres de troncation et le nombre de points intermédiaires à utiliser pour chaque arête de l'arbre. Nous donnons aussi une borne sur le nombre total de points intermédiaires à considérer en utilisant cette méthode.

Enfin, il est à noter que, comme nous voyons ce calcul de groupe de monodromie comme une étape du calcul de l'application d'Abel, notre stratégie est influencée par cet objectif. Par exemple, l'utilisation de développements en série tronqués sera utile pour obtenir des bornes d'erreur pour calculer les périodes de la courbe $\mathcal{C}$. De plus, l'utilisation de développements de Puiseux peut s'avérer utile lors du calcul de l'application d'Abel [DP07, Pat07].

Notations générales

- L un corps.
- $\mathrm{car}(L)$ sa caractéristique.
- K un corps de nombres (extension finie de $\mathbb{Q}$).
- $\overline{K}$ et $\overline{L}$ leur clôture algébrique.
- $F = \sum_{k=0}^{d_y} a_k(x) y^k$ un polynôme à deux variables x et y.
- d_y son degré en y.
- d_x son degré en x.
- D son degré total.
- $\mathcal{C}$ la courbe algébrique asociée à F.
- Δ_F le discriminant du polynôme F en y.
- R_F le résultant de F et F_y en y.
- $\alpha_1, \ldots, \alpha_n$ les points critiques finis de $\mathcal{C}$.
- $\delta(x_0)$ est la distance de x_0 à son plus proche point critique (x_0 excepté s'il est lui même un point critique).
- $\mathcal{F}(x_0)$ est la fibre en x_0.
- Si e désigne un entier positif, ζ_e est une racine primitive e-ième de l'unité. Ces racines primitives sont choisies de telle façon que $\zeta_{ab}^b = \zeta_a$.
- $[i, e]$, noté $[i]$ si e peut se déduire du contexte, est l'automorphisme :

$$\begin{array}{rccc} [i,e] & : & \overline{L}((x^{1/e})) & \to & \overline{L}((x^{1/e})) \\ & & x^{1/e} & \mapsto & \zeta_e^i x^{1/e} \end{array}$$

- Si $S = \sum_{k=n}^{\infty} \beta_k x^{k/e}$ et $r \geq n/e$ est un élément du corps $\mathbb{Q}$, alors $\widetilde{S}^r = \sum_{k=n}^{N} \beta_k x^{k/e}$, où $N = \max\{k \in \mathbb{Z} \mid \frac{k}{e} \leq r\}$.
- $M(N)$ désigne le nombre d'opérations arithmétiques nécessaires pour multiplier deux polynômes de degrés inférieurs à N.
- $\mathrm{tc}(S)$ est le coefficient de plus bas degré de S.
- arg la fonction argument d'un nombre complexe. Cette notation utilise la détermination principale $]-\pi, \pi]$.
- $\mathrm{ht}(F)$ désigne la hauteur du polynôme F.
- $\mathcal{I}(F)$ est l'entier $v_y(F(0, y))$, utile pour les polygones de Newton.

Chapitre 1

Courbes algébriques planes

1.1 Corps de fonctions algébriques

Dans cette partie, L désigne un corps de caractéristique $\mathrm{car}(L) \geq 0$, $\overline{L}$ une clôture algébrique de L, et $F \in L[x, y]$ un polynôme à coefficients dans L, que nous supposerons absolument irréductible dans $L[x, y]$, c'est à dire irréductible dans $\overline{L}[x, y]$. On notera d_x et d_y les degrés de F en respectivement x et y, et D le degré total du polynôme F. Nous supposerons de plus que $d_x > 0$ et $d_y > 0$.

Le polynôme F étant irréductible sur L, le quotient $E = L(x)[y]/(F(x, y))$ est un corps, extension de degré d_y de $L(x)$, que l'on appelle **corps de fonctions algébriques** sur L, et que l'on notera souvent E/L pour rappeler le corps de base. De plus, le polynôme F étant supposé absolument irréductible, le corps des constantes de E, formé de l'ensemble des éléments de E algébriques sur L, est égal à L [Eic66, section III.6].

1.1.1 Places d'un corps de fonctions algébriques

Pour illustrer le but de cette partie, nous commençons par considérer le corps de fonctions $L(x)/L$ en supposant sur cet exemple que L est algébriquement clos. Soit $f = \frac{g}{h} \in L(x)$, où $(g, h) \in L[x]^2$ sont non nuls et premiers entre eux. Un **zéro** de f est un élément $\beta \in L$ tel que $g(\beta) = 0$, et un **pôle** de f est un élément $\beta \in L$ tel que $h(\beta) = 0$. De plus, si $\beta \in L$, alors il existe un unique $m \in \mathbb{Z}$ et un unique $u \in L(x)$ tels que $f(x) = (x - \beta)^m u(x)$, où u est défini en β et vérifie $u(\beta) \neq 0$. Si $m > 0$ (respectivement $m < 0$), alors β est un **zéro** (respectivement **pôle**) de f **d'ordre** m.

Afin de généraliser cette notion de pôle et de zéro aux corps de fonctions quelconques, nous allons dans cette partie introduire la notion de place d'un corps de fonctions au sens de Chevalley [Che51].

Définition 1. *Un **anneau de valuation** $\mathcal{O}$ de E/L est un sous-anneau de E tel que $L \subsetneq \mathcal{O} \subsetneq E$ et :*

$$\forall s \in E, \quad s \notin \mathcal{O} \;\Rightarrow\; s^{-1} \in \mathcal{O}$$

*On appelle **place** de $\mathcal{O}$ l'ensemble $\mathfrak{P} = \mathcal{O} \backslash \mathcal{O}^\times$ (où $\mathcal{O}^\times$ désigne l'ensemble des éléments inversibles de $\mathcal{O}$).*

Proposition 1. *Un anneau de valuation $\mathcal{O}$ d'un corps de fonctions E/L est un anneau local. La place $\mathfrak{P} = \mathcal{O} \backslash \mathcal{O}^\times$ est son unique idéal maximal. Cet idéal est principal, c'est-à-dire qu'il existe $t \in \mathcal{O}$ tel que $\mathfrak{P} = t\mathcal{O}$. Un tel t est appelé un **paramètre local** de $\mathfrak{P}$ ou aussi **uniformisante** de $\mathfrak{P}$. Le corps $\mathcal{O}/\mathfrak{P}$ est appelé le **corps résiduel** de $\mathcal{O}$.*

Démonstration. voir [Che51, chapitre 1, section2] □

Par définition, chaque anneau de valuation a une unique place. Réciproquement, si $\mathcal{O}_1$ et $\mathcal{O}_2$ sont deux anneaux de valuation de E/L de même place $\mathfrak{P}$, alors $\mathcal{O}_1 = \mathcal{O}_2 = \{\beta \in E \,|\, \beta\mathfrak{P} \subset \mathfrak{P}\}$. Il y a donc une bijection entre l'ensemble des places d'un corps de fonctions E/L et l'ensemble des anneaux de valuation de ce même corps de fonction. Dans la suite, si $\mathfrak{P}$ est une place de E/L, nous noterons $\mathcal{O}_\mathfrak{P}$ l'anneau de valuation de E/L de place $\mathfrak{P}$.

Exemple 1. *Considérons le cas particulier $E = L(x)$ (correspondant au cas où $d_y = 1$). Alors on peut montrer (voir par exemple [Che51, chapitre 1, section 3]), que tout anneau de valuation de E/L est :*

- *soit $\mathcal{O}_P = \left\{ \frac{g}{h} \,|\, g, h \in L[x], P \text{ ne divise pas } h \right\}$ avec $P \in L[x]$ un polynôme irréductible,*
- *soit $\mathcal{O}_\infty = \left\{ \frac{g}{h} \,|\, g, h \in L[x], \deg(g) \leq \deg(h) \right\}$.*

Le paramètre local de l'anneau de valuation $\mathcal{O}_P$ (respectivement $\mathcal{O}_\infty$) est alors le polynôme irréductible P (respectivement $\frac{1}{x}$). Enfin, si L n'est pas algébriquement clos et si γ est une racine de P, alors le corps résiduel de $\mathcal{O}_P$ est isomorphe à $L(\gamma)$. Le corps résiduel de $\mathcal{O}_\infty$ est quant à lui isomorphe à L.

Définition 2. *Soit $\mathfrak{P}$ une place de E/L et t un paramètre local de cette place. Pour tout élément u de E, on définit l'**ordre de u en la place $\mathfrak{P}$** comme étant :*

$$\nu_{\mathfrak{P}}(u) := \begin{cases} \max\left\{k \in \mathbb{Z} \mid \frac{u}{t^k} \in \mathcal{O}_{\mathfrak{P}}\right\} & si\ u \neq 0 \\ +\infty & sinon \end{cases}$$

L'ordre en la place $\mathfrak{P}$ de u ne dépend pas du paramètre local t choisi pour le définir, puisque si t_1 et t_2 sont deux paramètres locaux de la place $\mathfrak{P}$, alors $\mathfrak{P} = t_1\mathcal{O}_{\mathfrak{P}} = t_2\mathcal{O}_{\mathfrak{P}}$. De ce fait, t_1/t_2 et t_2/t_1 sont des éléments de $\mathcal{O}_{\mathfrak{P}}$, et on a $\frac{u}{t_1^k} = \frac{u}{t_2^k}\left(\frac{t_2}{t_1}\right)^k \in \mathcal{O}_{\mathfrak{P}} \Leftrightarrow \frac{u}{t_2^k} = \frac{u}{t_1^k}\left(\frac{t_1}{t_2}\right)^k \in \mathcal{O}_{\mathfrak{P}}$.

Rappelons que si E est un corps, alors une **valuation discrète** ν sur E est une application surjective $\nu : E \to \mathbb{Z} \cup \{\infty\}$ telle que :

1. $\nu(x_0) = \infty \Leftrightarrow x_0 = 0$,
2. $\nu(x_0y_0) = \nu(x_0) + \nu(y_0)$ pour tout $(x_0, y_0) \in E^2$,
3. $\nu(x_0 + y_0) \geq \min\{\nu(x_0), \nu(y_0)\}$ pour tout $(x_0, y_0) \in E^2$.

Proposition 2. *Soit E/L un corps de fonctions et $\mathfrak{P}$ une place de E/L. L'application $\nu_{\mathfrak{P}}$ est une valuation du corps E. De plus, on a :*

$$\mathcal{O}_{\mathfrak{P}} = \{u \in E \mid \nu_{\mathfrak{P}}(u) \geq 0\}$$

La place $\mathfrak{P}$ vérifie :

$$\mathfrak{P} = \{u \in E \mid \nu_{\mathfrak{P}}(u) > 0\}$$

Si l'on note $m = \nu_{\mathfrak{P}}(f)$, la place $\mathfrak{P}$ sera un **zéro** (respectivement **pôle**) **d'ordre** m de la fonction f si $m > 0$ (respectivement $m < 0$). On peut ainsi remarquer que $\mathfrak{P}$ est un zéro (respectivement pôle) de f si et seulement si $f \in \mathfrak{P}$ (respectivement $1/f \in \mathfrak{P}$).

Pour finir cette partie, nous allons définir la notion d'extension de places de $L(x)/L$, ainsi que la notion de ramification d'une place de E au-dessus d'une place de $L(x)$.

Définition 3. *Soit E/L un corps de fonctions, $\mathfrak{p}$ une place de $L(x)/L$, et $\mathfrak{P}$ une place de E/L. On dit que $\mathfrak{P}$ est **au-dessus** de $\mathfrak{p}$ si $\mathcal{O}_{\mathfrak{p}} \subset \mathcal{O}_{\mathfrak{P}}$.*

Proposition 3. *Pour toute place $\mathfrak{p}$ de $L(x)/L$, il existe au moins une place $\mathfrak{P}$ de E/L au-dessus de $\mathfrak{p}$. De plus, si $\mathfrak{P}$ est au-dessus de $\mathfrak{p}$, alors on a :*

$$\mathfrak{P} \cap L(x) = \mathfrak{p} \ \text{ et } \ \mathcal{O}_{\mathfrak{P}} \cap L(x) = \mathcal{O}_{\mathfrak{p}}.$$

Définition 4. *Soit $\mathfrak{P}$ une place de E/L située au-dessus d'une place $\mathfrak{p}$ de $L(x)/L$. Notons t une uniformisante de la place $\mathfrak{p}$. Alors l'**indice de ramification** de la place $\mathfrak{P}$ au-dessus de la place $\mathfrak{p}$ est l'entier $e_{\mathfrak{P}} = \nu_{\mathfrak{P}}(t)$.*

Théorème 1. *Soit E/L un corps de fonctions et $\mathfrak{p}$ une place $L(x)/L$. Alors il existe un nombre fini de places de E/L au-dessus de $\mathfrak{p}$, que nous noterons $\mathfrak{P}_1, \ldots, \mathfrak{P}_r$. De plus, si pour tout $1 \leq i \leq r$, on note $e_{\mathfrak{P}_i}$ l'indice de ramification de la place $\mathfrak{P}_i$ au-dessus de la place $\mathfrak{p}$, et $f_{\mathfrak{P}_i} = [\mathcal{O}_{\mathfrak{P}_i}/\mathfrak{P}_i : \mathcal{O}_{\mathfrak{p}}/\mathfrak{p}]$ le degré de l'extension formée par le corps résiduel de $\mathcal{O}_{\mathfrak{P}_i}$ sur le corps résiduel de $\mathcal{O}_{\mathfrak{p}}$, alors on a :*

$$\sum_{i=1}^{r} e_{\mathfrak{P}_i} f_{\mathfrak{P}_i} = d_y$$

Démonstration. Il s'agit du théorème 1 de [Che51, chapitre 4, section 1] □

Proposition 4 (formule d'Hurwitz)**.** *On suppose que l'hypothèse (1.1) de la section 1.1.4 est vérifiée. Si pour une place $\mathfrak{P}$ de E/L au-dessus d'une place $\mathfrak{p}$ de $L(x)/L$, on note $e_{\mathfrak{P}}$ son indice de ramification, $f_{\mathfrak{P}} = [\mathcal{O}_{\mathfrak{P}}/\mathfrak{P} : \mathcal{O}_{\mathfrak{p}}/\mathfrak{p}]$ et $d_{\mathfrak{p}}$ le degré de la place $\mathfrak{p}$, c'est-à-dire le degré de l'extension engendrée par son corps résiduel sur L, alors le **genre** g du corps de fonction E/L vérifie la relation suivante :*

$$g = 1 - d_y + \frac{1}{2} \sum_{\mathfrak{p}} d_{\mathfrak{p}} \sum_{\mathfrak{P}|\mathfrak{p}} f_{\mathfrak{P}}(e_{\mathfrak{P}} - 1)$$

où les sommes sont prises respectivement sur l'ensemble des places $\mathfrak{p}$ de $L(x)/L$ et l'ensemble des places $\mathfrak{P}$ de E/L qui divisent $\mathfrak{p}$.

Démonstration. Cette formule provient de [Che51] : c'est une conséquence du corollaire 2 de la section VI.2. Les hypothèses sur la caractéristique du corps nous permettent en effet, à l'aide du théorème 7 de la section IV.8, de remplacer les exposants différentiels par $e_{\mathfrak{P}} - 1$ dans la définition de la différente. □

1.1.2 Paramétrisations et branches

Notons $\mathcal{C} = \{(x_0, y_0) \in \overline{L}^2 \mid F(x_0, y_0) = 0\}$ la courbe algébrique définie par le polynôme F sur le corps L. Cette courbe $\mathcal{C}$ est un ensemble de points. Néanmoins, certains de ces points sont « spéciaux ». Par exemple, si l'on considère la courbe $\mathcal{C}_1$ définie par le polynôme $F_1(x, y) = y^2 - x^3 - x^2 \in \mathbb{R}[x]$ (voir la figure 1.1), on peut voir que le point $(0, 0)$ est un point particulier,

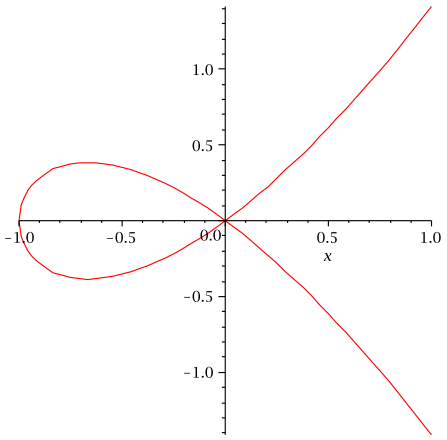

FIG. 1.1 – $\mathcal{C}_1 = \{(x_0, y_0) \in \mathbb{R}^2 \,|\, y_0^2 = x_0^3 + x_0^2\}$

puisqu'il y a deux « branches » de $\mathcal{C}_1$ (l'une ayant une tangente de pente 1, l'autre une tangente de pente -1) en ce point.

Le but de cette section est de définir cette notion de branche. Afin de mieux expliciter les branches infinies de $\mathcal{C}$, nous considérons le plan projectif $\mathbb{P}^2(L)$, ainsi que le polynôme $\hat{F}(x, y, z) = z^D F(\frac{x}{z}, \frac{y}{z})$, où D est le degré total de F, qui définit la **courbe algébrique plane projective** complétée de $\mathcal{C}$:

$$\hat{\mathcal{C}} = \{(x : y : z) \in \mathbb{P}^2(L) | \hat{F}(x : y : z) = 0\}$$

Les points à l'infini de $\hat{\mathcal{C}}$ sont les points de $\hat{\mathcal{C}}$ qui sont des points à l'infini de $\mathbb{P}^2(L)$, c'est à dire les points de la forme $(x_0 : y_0 : 0)$ avec $(x_0, y_0) \in L^2$. Les autres points de $\hat{\mathcal{C}}$ sont les points affines de $\hat{\mathcal{C}}$, et l'ensemble des points affines de $\hat{\mathcal{C}}$ s'identifie à $\mathcal{C}$.

Définition 5. *Une **paramétrisation** $R(T)$ de $\mathcal{C}$ est définie comme étant un couple $(\tilde{x}(T), \tilde{y}(T)) \in (\overline{L}((T)))^2$ tel que :*

1. *$\tilde{x}(T)$ et $\tilde{y}(T)$ ne sont pas tous les deux dans $\overline{L}$.*
2. *$F(\tilde{x}(T), \tilde{y}(T)) = 0$ dans $\overline{L}((T))$.*

*Deux paramétrisations $(\tilde{x}(T), \tilde{y}(T))$ et $(\hat{x}(T), \hat{y}(T))$ de $\mathcal{C}$ sont dites **équivalentes** si il existe $u \in L[[T]]$ de valuation T-adique 1 tel que :*

$$(\tilde{x}(T), \tilde{y}(T)) = (\hat{x}(u(T)), \hat{y}(u(T)))$$

*Une paramétrisation $(\tilde{x}(T), \tilde{y}(T))$ de $\mathcal{C}$ est dite **irréductible** s'il n'existe aucun $u \in L[[T]]$ de valuation T-adique supérieure ou égale à 2 tel que $(\tilde{x}(T), \tilde{y}(T)) \in (\overline{L}[[u(T)]])^2$.*

*Le **corps des coefficients** de R est l'extension de L engendrée par les coefficients de $\tilde{x}(T)$ et $\tilde{y}(T)$.*

Exemple 2. *Considérons le polynôme $F(x,y) = y^2 - 2x^2 \in \mathbb{Q}[x,y]$ et notons $\mathcal{C}$ la courbe algébrique associée. Le couple $R_0(T) = (T^2, \sqrt{2}T^2)$ est une paramétrisation de $\mathcal{C}$. Cette paramétrisation n'est pas irréductible, puisqu'elle appartient à $L((T^2))$. Une paramétrisation irréductible de F est donnée par $R_1(T) = (T, \sqrt{2}T)$. Le corps des coefficients de cette paramétrisation est $\mathbb{Q}(\sqrt{2})$. Une paramétrisation équivalente à R_1 est $R_2(T) = (T/\sqrt{2}, T)$, puisque $R_2(T) = R_1(T/\sqrt{2})$.*

De la même façon, on peut définir une paramétrisation de $\hat{\mathcal{C}}$ comme étant un triplet $(\tilde{x}(T) : \tilde{y}(T) : \tilde{z}(T)) \in \mathbb{P}^2(\,L((T))\,)$ tel que $\hat{F}(\tilde{x}(T) : \tilde{y}(T) : \tilde{z}(T)) = 0$ et $(\tilde{x}(T) : \tilde{y}(T) : \tilde{z}(T)) \notin \mathbb{P}^2(\overline{L})$. Les notions d'équivalence et d'irréductibilité sont alors similaires. De plus, si l'on choisit une paramétrisation de telle sorte que $\min(v_T(\tilde{x}), v_T(\tilde{y}), v_T(\tilde{z})) = 0$, alors le centre de la paramétrisation $(\tilde{x}(T) : \tilde{y}(T) : \tilde{z}(T))$ est défini comme étant le triplet $(\tilde{x}(0) : \tilde{y}(0) : \tilde{z}(0)) \in \mathbb{P}^2(L)$.

Comme le corps L est supposé algébriquement clos, on peut montrer que pour toute paramétrisation $(\tilde{x}(T) : \tilde{y}(T) : \tilde{z}(T))$ de $\hat{\mathcal{C}}$, on a $\tilde{z} \neq 0$ [DRSS95, Lemme 5.1]. De ce fait, il existe une bijection entre les paramétrisations de $\mathcal{C}$ et celles de $\hat{\mathcal{C}}$.

On distinguera deux types de paramétrisations :

- Une paramétrisation $R(T) = (\tilde{x}, \tilde{y})$ de $\mathcal{C}$ sera dite **finie** si elle appartient à $(\overline{L}[[T]])^2$. Dans ce cas, on appellera **centre** de cette paramétrisation le point $(\tilde{x}(0), \tilde{y}(0))$ de $\mathcal{C}$, et l'on dira que la paramétrisation $R(T)$ est **au-dessus** de $\tilde{x}(0)$. Une telle paramétrisation correspond à la paramétrisation $(\tilde{x} : \tilde{y} : 1)$ centrée en $(\tilde{x}(0) : \tilde{y}(0) : 1)$ de $\hat{\mathcal{C}}$.
- Une paramétrisation $R(T) = (\tilde{x}, \tilde{y})$ sera dite **infinie** si elle n'appartient pas à $(\overline{L}[[T]])^2$. On peut classer ces dernières en trois catégories :
  1. Si $\tilde{x}(T) \in \overline{L}[[T]]$, alors on dira que la paramétrisation R est une paramétrisation infinie au-dessus de $\tilde{x}(0)$. Une telle paramétrisation correspond à la paramétrisation $(\tilde{y}^{-1}(T)\tilde{x}(T) : 1 : \tilde{y}^{-1}(T))$ centrée en $(0 : 1 : 0)$ de $\hat{\mathcal{C}}$.
  2. Si $\tilde{y}(T) \in \overline{L}[[T]]$, alors la paramétrisation R est dite **au-dessus de l'infini**. Elle correspond à la paramétrisation de $\hat{\mathcal{C}}$ centrée en $(1 : 0 : 0)$ et égale à $(1 : \tilde{x}^{-1}(T)\tilde{y}(T) : \tilde{x}^{-1}(T))$.
  3. Sinon, R est une paramétrisation infinie au-dessus de l'infini, qui correspond à une paramétrisation de $\hat{\mathcal{C}}$ centrée en $(1 : 1 : 0)$, $(1 : 0 : 0)$ ou $(0 : 1 : 0)$.

Définition 6. *On appelle* ***branche*** *de $\mathcal{C}$ une classe d'équivalence de paramétrisations irréductibles.*

Les notions de paramétrisation finie ou infinie, de centre de paramétrisation, et de paramétrisation au-dessus d'un point (fini ou non) s'étend trivialement aux branches de $\mathcal{C}$.

1.1.3 Lien entre les places et les branches

Après avoir définies les places du corps de fonctions E/L et les branches de l'ensemble $\mathcal{C} = \{(x_0, y_0) \in \overline{L}^2 \,|\, F(x_0, y_0) = 0\}$, nous allons maintenant montrer qu'il existe une bijection naturelle entre les places de E/L et les branches de $\mathcal{C}$.

Considérons une paramétrisation (quelconque) $R(T) = (\tilde{x}(T), \tilde{y}(T))$ de $\mathcal{C}$. Cette paramétrisation définit un morphisme de corps :

$$\begin{array}{rccc} \phi_R \ : & E & \to & \overline{L}((T)) \\ & f(x,y) & \mapsto & f(\tilde{x}(T), \tilde{y}(T)) \end{array}$$

En composant ϕ_R avec la valuation ν_T de $\overline{L}((T))$, on obtient ainsi une valuation du corps E, que l'on notera à nouveau ν_T. On peut alors définir les ensembles $\mathfrak{P}_R = \{f \in E \,|\, \nu_T(f) > 0\}$ et $\mathcal{O}_{\mathfrak{P}_R} = \{f \in E \,|\, \nu_T(f) \geq 0\}$. Il est facile de vérifier que $\mathcal{O}_{\mathfrak{P}_R}$ est un anneau de valuation de E/L, et que $\mathfrak{P}_R$ est la place qui lui est associée. De plus, deux paramétrisations équivalentes de $\mathcal{C}$ définissent la même place de E/L. Ainsi, nous avons défini une application Ψ, qui, à chaque branche de $\mathcal{C}$, associe une place de E/L.

De la même façon, soit $\mathfrak{P}$ une place de E/L et t un paramètre local. On peut se fixer un système de représentants du corps résiduel $\mathcal{O}_{\mathfrak{P}}/\mathfrak{P}$, que nous noterons $E_{\mathfrak{P}}$. Alors on peut montrer que tout élément $u \in E$ s'écrit de manière unique sous la forme $u = \sum_{i=m}^{\infty} u_i t^i$, où $m = \nu_{\mathfrak{P}}(u)$ et les éléments u_i appartiennent à $E_{\mathfrak{P}}$: u_m est le représentant du projeté de u/t^m par la projection canonique de $\mathcal{O}_{\mathfrak{P}}$ sur son corps résiduel, u_{m+1} est le représentant du projeté de $(u - u_m t^m)/t^{m+1}$, u_{m+2} celui de $(u - (u_m t^m + u_{m+1} t^{m+1}))/t^{m+2}$ etc. Le corps résiduel pouvant être vu comme une extension algébrique finie de L, on peut définir une application :

$$\begin{array}{rccc} \phi_t \ : & E & \to & \overline{L}((T)) \\ & u = \sum_{i=m}^{\infty} u_i t^i & \mapsto & \sum_{i=m}^{\infty} \bar{u}_i T^i \end{array}$$

où $\bar{u}_i$ représente la classe modulo $\mathfrak{P}$ de u_i. En particulier, si l'on considère x et y comme des éléments de E, on peut définir $\tilde{x}_t = \phi_t(x)$ et $\tilde{y}_t = \phi_t(y)$.

Lemme 1. *La paire $(\tilde{x}_t, \tilde{y}_t)$ est une paramétrisation irréductible de $\mathcal{C}$.*

Démonstration. voir [DRSS95, section 11.4]. □

Ainsi, pour chaque place $\mathfrak{P}$ de E, on peut définir une branche de $\mathcal{C}$. On définit ainsi une application Ψ', qui à chaque place de E/L associe une branche de $\mathcal{C}$.

Proposition 5. *Si le corps L est algébriquement clos, alors les applications Ψ et Ψ' sont des applications réciproques.*

Démonstration. voir [DRSS95, section 11.4]. □

Il s'ensuit trivialement le corollaire suivant :

Corollaire 1. *Si le corps L est algébriquement clos, alors il y a une bijection entre l'ensemble des places de E/L et l'ensemble des branches de $\mathcal{C}$.*

1.1.4 Développements de Puiseux

Dans cette partie, nous relâchons quelque peu nos hypothèses : nous supposons ici que le polynôme $F = \sum_{k=0}^{d_y} a_k(x)y^k \in L[x,y]$ considéré est sans facteur carré (et donc pas forcément irréductible). De plus, nous faisons l'hypothèse suivante :

$$\mathrm{car}(L) = 0 \quad \text{ou} \quad \mathrm{car}(L) > d_y \tag{1.1}$$

Nous rappelons les résultats sur les développements de Puiseux (classiques et rationnels) en x, ainsi que leur lien avec la factorisation du polynôme F dans $L((x))[y]$, $\overline{L}((x))[y]$ et $\overline{L((x))}[y]$. Pour simplifier les notations, nous nous plaçons au-dessus du point $x = 0$. Cette hypothèse ne réduit nullement l'étude faite, puisque l'on peut s'y ramener à l'aide d'un changement de variable $x \leftarrow x + x_0$ (ou $x \leftarrow 1/x$ dans le cas où l'on considère le point $x_0 = \infty$).

Nous commençons par fixer quelques notations :

- Si e désigne un entier positif, nous noterons ζ_e une racine primitive e-ième de l'unité dans $\overline{L}$. Ces racines primitives seront choisies de telle façon que $\zeta_{ab}^b = \zeta_a$.
- Étant donné un entier naturel $e \geq 1$, pour tout $0 \leq i \leq e-1$, on notera $[i,e]$ l'automorphisme :

$$\begin{array}{rccc} [i,e] \ : & \overline{L}((x^{1/e})) & \to & \overline{L}((x^{1/e})) \\ & x^{1/e} & \mapsto & \zeta_e^i x^{1/e} \end{array}$$

Quand l'entier e peut être déduit du contexte, on notera plus simplement $[i]$ à la place de $[i, e]$. Si $S \in \overline{L}((x^{1/e})$, l'image de S par $[i]$ sera notée $S^{[i]}$.

- Si $S = \sum_{k=n}^{\infty} \beta_k x^{k/e}$ est une série en des puissances fractionnaires de $L((x^{1/e}))$, et r est un nombre rationnel supérieur à $\frac{n}{e}$, $\widetilde{S}^r$ désigne la série tronquée $\widetilde{S}^r = \sum_{k=n}^{N} \beta_k x^{k/e}$, où $N = \max\{k \in \mathbb{Z} \mid \frac{k}{e} \leq r\}$.

Développements de Puiseux classiques

Théorème 2 (Puiseux). *Soit F un polynôme sans facteur carré de $L[x, y]$ tel que $d_y = \deg_y(F) > 0$.*

- *Si la condition (1.1) est vérifiée, alors il existe $e_1, \ldots, e_s$ des entiers positifs vérifiant $\sum_{i=1}^{s} e_i = d_y$ tel que F (vu comme un polynôme univarié en y) possède d_y racines distinctes dans $\overline{L((x))}$, qui s'écrivent de la manière suivante :*

$$S_{ij}(x) = \sum_{k=n_i}^{\infty} \beta_{ik}\, \zeta_{e_i}^{jk}\, x^{\frac{k}{e_i}}$$

  où $1 \leq i \leq s$, $0 \leq j \leq e_i - 1$, $n_i \in \mathbb{Z}$ et $\beta_{in_i} \neq 0$. De plus, l'ensemble des coefficients $\{\beta_{ik}\}$ est inclus dans une extension finie de L.
- *Si $car(L) = 0$, on a de plus :*

$$\overline{L(x)} \subset \overline{L((x))} \simeq \bigcup_{e \in \mathbb{N}^*} \overline{L}((x^{1/e}))$$

Démonstration. Voir par exemple [BK86, section 8.3]. □

La factorisation de F sur $\overline{L((x))}[y]$ est alors :

$$F(x, y) = a_{d_y}(x) \prod_{i=1}^{s} \prod_{j=0}^{e_i - 1} (y - S_{ij}(x))\,.$$

Définition 7. *Ces d_y séries de Laurent fractionnaires sont appelés les **développements de Puiseux** de F au-dessus de 0. L'entier e_i est l'**indice de ramification** de la série S_{ij}. Si $e_i > 1$, on dit que la série S_{ij} est **ramifiée**. Une série S_{ij} telle que $n_i \geq 0$ sera dite **définie en** $x = 0$.*

Remarque 1. *Il est à noter que l'hypothèse (1.1) sur la caractéristique du corps est importante. Par exemple, le polynôme $F(x, y) = y^2 + xy + x \in$*

$\mathbb{F}_2[x,y]$ *vu comme un polynôme univarié en la variable* y *a deux racines distinctes dans* $\overline{\mathbb{F}_2((x))}$ *:*

$$S_1(x) = \sum_{k=1}^{\infty} x^{\frac{2^k-1}{2^k}} \text{ et } S_2(x) = S_1(x) + x.$$

Ceci montre que les séries solutions de F *ne sont pas toujours des séries de Puiseux dans le cas* $car(L) \leq d_y$*. Pour plus de détails sur ce cas, nous renvoyons le lecteur à [Gri95].*

Exemple 3. *Soit* $L = \mathbb{Q}$ *le corps des nombres rationnels et* $F_1(x,y) = y^3 - x$*. Il existe trois séries solutions du polynôme* F_1 *au dessus de* 0*, qui sont les séries finies* $S_{10}(x) = x^{1/3}, S_{11}(x) = \zeta_3 x^{1/3}$ *et* $S_{12}(x) = \zeta_3^2 x^{1/3}$*. Ces trois séries ont un indice de ramification égal à* 3*.*

Exemple 4. *Soit* $L = \mathbb{Q}$ *et* $F_2(x,y) = (y^3-x)\,((y-1)^2-x)\,(y-2-x^2)+x^2\,y^5$*. Les* 6 *séries de Puiseux du polynôme* F_2 *au dessus de* $x = 0$ *sont donnés par les séries dont les premiers termes sont :*

- $S_{1j}(x) = \zeta_3^j x^{\frac{1}{3}} + \frac{1}{6}\,x^3 + \frac{5}{12}\,\zeta_3^j\,x^{\frac{10}{3}} + \cdots$, $j = 0, 1, 2$,
- $S_{2j}(x) = 1 + x^{\frac{1}{2}}\zeta_2^j\frac{1}{2}\,x^{\frac{3}{2}} + \frac{3}{2}\,x^2\zeta_2^j\frac{19}{8}\,x^{\frac{5}{2}} + 2\,x^3 + \cdots$, $j = 0, 1$,
- $S_{30}(x) = 2 - 3\,x^2 - \frac{9}{2}\,x^3 + \cdots$,

et qui ont pour indices de ramification respectifs $3, 2$ *et* 1*.*

Un nombre arbitraire de termes de ces séries de Puiseux peut être calculé à l'aide de l'algorithme de Newton-Puiseux. Nous détaillerons ce dernier dans le chapitre 2.

Pour tout entier positif $e \leq d_y$, notre hypothèse (1.1) sur la caractéristique du corps L implique que le groupe de Galois $\mathbb{G}_e$ de $\overline{L}((x^{1/e}))/\overline{L}((x))$ est cyclique et engendré par $[1, e]$: $x^{1/e} \mapsto \zeta_e x^{1/e}$. Ainsi, $\mathbb{G}_{e_i}$ permute les éléments de $S_i = \{S_{ij}(x)\}_{0 \leq j \leq e_i - 1}$ de manière cyclique.

Définition 8. *On appelle l'ensemble* S_i *un* ***cycle*** *de* F *au-dessus de* 0*.*

Remarque 2. *Les séries* $(S_{ij})_{0 \leq j \leq e_i - 1}$ *peuvent être facilement déduites de n'importe quel élément de* S_i *: il suffit de faire agir* $[1, e_i]$ $e_i - 1$ *fois sur cet élément. Ainsi, il suffit de calculer un élément de chaque cycle pour obtenir ensuite l'ensemble des séries de Puiseux au-dessus de* 0*.*

Le regroupement par cycle des développements de Puiseux permet de raffiner la factorisation de F sur $\overline{L((x))}[y]$:

Proposition 6. *La factorisation de F sur $\overline{L}((x))$ est donnée par :*

$$F(x,y) = a_{d_y}(x) \prod_{i=1}^{s} \bar{F}_i(x,y)$$

où les polynômes

$$\bar{F}_i(x,y) = \prod_{j=0}^{e_i-1} (y - S_{ij}(x))$$

sont irréductibles dans $\overline{L}((x))[y]$.

Définition 9. *L'**indice de régularité** r_{ij} de la série de Puiseux S_{ij} dans F est le plus petit entier relatif N tel que $\widetilde{S_{ij}}^{\frac{N}{e_i}} = \widetilde{S_{uv}}^{\frac{N}{e_i}}$ implique $(u,v) = (i,j)$. La série tronquée $\widetilde{S_{ij}}^{\frac{r_{ij}}{e_i}}$ est la **partie singulière** de S_{ij} dans F.*

Autrement dit, r_{ij} est le nombre minimum de termes nécessaire pour distinguer S_{ij} des autres séries de Puiseux au-dessus de 0.

Exemple 5. *Soit F le polynôme minimal sur $\mathbb{Q}(x)$ de la série $S(x) = x^{5/6} + x$. Ce polynôme a un cycle, dont un élément est la série $S(x)$. L'indice de régularité de ce développement est égal à 5, le terme de $S(x)$ qui fait apparaître la ramification de la série.*

Il est important de noter que, comme l'on ne suppose pas ici que F est irréductible dans $L[x,y]$, l'entier r_{ij} ne dépend pas uniquement de S_{ij}, mais aussi de F, comme le montre les deux exemples suivants :

Exemple 6. *Soit F le produit des deux polynômes minimaux sur $\mathbb{Q}(x)$ des séries $S_1(x) = x^{5/6} + x$ et $S_2(x) = x^{5/6} + x^{11/12}$. Ce polynôme a deux cycles, dont les séries S_1 et S_2 sont des représentants. Les indices de régularité de ces développements dans F sont respectivement 6 et 11.*

Exemple 7. *Le polynôme $F(x,y) = (y - 1 - 2x - x^2)(y - 1 - 2x - x^7) \in \mathbb{Q}[x,y]$ a deux séries de Puiseux au-dessus de 0, $S_{1,0}(x) = 1 + 2x + x^2$ et $S_{2,0}(x) = 1 + 2x + x^7$, qui sont toutes deux finies. L'indice de régularité de ces deux séries dans F est égal à 2.*

Si la partie singulière des développements de Puiseux est connue, alors un changement de variable définit un polynôme bivarié pour lequel le point 0 est régulier. Ainsi, les autres termes de la série peuvent être calculés « rapidement », en utilisant l'algorithme des itérations quadratiques de Newton

[KT78, vzGG99] (voir le chapitre 2), ou en utilisant des algorithmes détendus [vdH02].

Il est possible de caractériser l'indice de régularité de la manière suivante :

Proposition 7. *Notons $Y_1, \ldots, Y_{d_y}$ les d_y développements de Puiseux définis au-dessus de 0, ayant pour indices de ramification respectifs $e_1, \ldots, e_{d_y}$. Alors pour tout $1 \leq i \leq d_y$, r_i est l'indice de régularité de la série Y_i si et seulement si :*

1. *Pour tout $j \in \{1, \ldots, d_y\}$ différent de i, $\widetilde{Y_i}^{\frac{r_i}{e_i}} \neq \widetilde{Y_j}^{\frac{r_i}{e_i}}$.*
2. *Il existe $j \in \{1, \ldots, d_y\}$ différent de i tel que $\widetilde{Y_i}^{\frac{r_i-1}{e_i}} = \widetilde{Y_j}^{\frac{r_i-1}{e_i}}$.*

Démonstration. Le premier point est trivial. Le second est dû à la minimalité de r_i. □

Corollaire 2. *Les éléments d'un même cycle S_i ont le même indice de régularité, que l'on le notera r_i.*

Démonstration. Notons les séries de Puiseux S_{ij} ($0 \leq j \leq e_i - 1$, $1 \leq i \leq s$), comme dans le théorème 2. Soit r l'indice de régularité de la série S_{i0}. Alors, d'après la proposition 7, on a :

$$\widetilde{S_{i0}}^{\frac{r}{e_i}} \neq \widetilde{S_{uv}}^{\frac{r}{e_i}} \text{ pour tout } (u,v) \neq (i,0)$$

et

$$\text{il existe } (u,v) \neq (i,0) \text{ tel que } \widetilde{S_{i0}}^{\frac{r-1}{e_i}} = \widetilde{S_{uv}}^{\frac{r-1}{e_i}}$$

Si l'on note e le plus petit multiple commun aux indices de ramifications e_i, $1 \leq i \leq s$, alors on a $S_{ij}(x) = S_{i0}^{[j,e]}(x)$. De ce fait, en conjuguant la relation précédente par $[j,e]$ pour tout $0 \leq j \leq e_i - 1$, on obtient le résultat. □

Développements de Puiseux rationnels

Pour faire des calculs dans des extensions de L les plus petites possibles, et pour utiliser les relations de conjugaison au-dessus de L, D. Duval a introduit le concept de « développements de Puiseux rationnels sur L » [Duv87]. Nous décrivons ici ce concept.

Remarque 3. *Une variante de ce concept de « développements de Puiseux rationnels sur L » a été introduite dans la littérature, et notamment dans [Duv89, Wal99]. La définition de [Duv87] nous a semblé la plus appropriée à notre contexte.*

Dans le cas où F est irréductible, nous notons $\{\mathfrak{P}_i\}_{1\leq i\leq r}$ les places de E/L divisant (x) et k_i le corps résiduel de la place $\mathfrak{P}_i$. Nous utilisons aussi l'application Ψ définie dans la partie 1.1.3.

Définition 10 (Développements de Puiseux rationnels)**.**

- *Si le polynôme F est irréductible dans $L[x,y]$, un* ***système de développements de Puiseux rationnels sur*** *L au-dessus de 0 de F est un ensemble de paramétrisations irréductibles $\{R_i\}_{1\leq i\leq r}$ de la forme :*

$$R_i(T) = (\tilde{x}_i(T), \tilde{y}_i(T)) = \left(\lambda_i T^{e_i}, \sum_{k=n_i}^{\infty} \beta_{ik} T^k\right) \in \overline{L}((T))^2$$

  avec $e_i > 0$, $n_i \in \mathbb{Z}$, et $\beta_{in_i} \neq 0$ tels que :

  1. *L'application Ψ définit une bijection de l'ensemble $\{R_i\}_{1\leq i\leq r}$ vers l'ensemble $\{\mathfrak{P}_i\}_{1\leq i\leq r}$. On supposera que les $\mathfrak{P}_i$ sont numérotés de telle sorte que $\mathfrak{P}_i = \mathfrak{P}_{R_i} = \Psi(R_i)$.*
  2. *Le corps des coefficients de R_i est isomorphe à k_i.*

- *Si le polynôme F est sans facteur carré, un* ***système de développements de Puiseux rationnels sur*** *L au-dessus de 0 de F est la réunion des systèmes de développements de Puiseux rationnels sur L des facteurs irréductibles de F dans $L[x,y]$.*

Les nombres λ_i, les entiers e_i, ainsi qu'un nombre arbitraire de termes des séries $\tilde{y}_i$ peuvent être calculés à l'aide de l'algorithme de Newton-Puiseux rationnel introduit par Dominique Duval [Duv87, Duv89]. Nous détaillerons cet algorithme dans le chapitre 2.

Les développements de Puiseux rationnels présentent plusieurs avantages sur les développements de Puiseux classiques :

- La représentation est plus compacte, puisque l'on ne représente qu'un élément par classe de conjugaison du corps des coefficients, et que l'extension engendrée par le corps des coefficients des développements de Puiseux rationnels est minimal.
- Ils décrivent bijectivement les places du corps de fonctions algébriques $L(x)[y]/(F(x,y))$, et le corps résiduel d'une telle place est donné (à isomorphisme près) par le corps des coefficients du développement de Puiseux rationnel associé.

Exemple 8. *Considérons le polynôme $F(x,y) = y^2 - 2x \in \mathbb{Q}[x,y]$. Ce polynôme a deux développements de Puiseux classiques en 0 :*

$$S_{11}(x) = \sqrt{2}x^{1/2} \text{ et } S_{12}(x) = -\sqrt{2}x^{1/2}$$

Le corps des coefficients de ces séries est $\mathbb{Q}(\sqrt{2})$*. Un système de développements de Puiseux rationnel de* F *au-dessus de* 0 *est :*

$$R_1(T) = (T^2/2, T),$$

qui a pour corps de coefficients $\mathbb{Q}$*.*

Les développements de Puiseux classiques peuvent facilement être obtenus à partir d'un système de développements de Puiseux rationnels (on note ici f_i le degré de l'extension engendrée par le corps des coefficients de R_i sur L) :

1. Chaque développement de Puiseux rationnel R_i a exactement f_i conjugués sur L, que l'on notera R_i^σ $(1 \leq \sigma \leq f_i)$.

$$R_i^\sigma(T) = (\tilde{x}_i^\sigma(T), \tilde{y}_i^\sigma(T)) = \left(\lambda_i^\sigma T^{e_i}, \sum_{k=n_i}^{\infty} \beta_{ik}^\sigma T^k\right)$$

2. Chaque R_i^σ définit alors un développement de Puiseux $\tilde{y}_i^\sigma((x/\lambda_i^\sigma)^{1/e_i})$. L'ensemble des séries ainsi définies forme un ensemble de représentants des cycles $\{S_i\}_{1 \leq i \leq s}$ de F au-dessus de 0.
3. Les d_y développements de Puiseux sont finalement obtenus en utilisant l'action de $\mathbb{G}_{e_i}$, $1 \leq i \leq s$.

Là aussi, on peut montrer que les indices de régularité de toutes les séries de Puiseux correspondant au même développement de Puiseux rationnel sont égaux. On peut donc définir la partie singulière d'un développement de Puiseux rationnel R_i comme étant le couple :

$$\left(\lambda_i T^{e_i}, \sum_{k=n_i}^{r_i} \beta_{ik} T^k\right)$$

où r_i est l'indice de régularité d'un des développements de Puiseux associé à R_i.

Théorème 3. *La factorisation complète de* $F(x, y)$*, considéré comme un polynôme univarié en* y*, sur les corps* $L((x))$*,* $\overline{L}((x))$ *et* $\overline{L((x))}$ *est donnée par :*

$$F(x, y) = a_{d_y}(x) \prod_{i=1}^{r} F_i(x, y) \text{ dans } L((x))[y],$$

$$F_i(x,y) = \prod_{\sigma=1}^{f_i} F_i^\sigma(x,y) \ \textit{dans}\ \overline{L}((x))[y],$$

et

$$F_i^\sigma = \prod_{j=0}^{e_i-1} \left(y - \tilde{y}_i^\sigma \left(\zeta_{e_i}^j \left(\frac{x}{\lambda_i^\sigma} \right)^{1/e_i} \right) \right) \ \textit{dans}\ \overline{L((x))}[y].$$

Démonstration. voir [Duv87] □

On retrouve alors l'égalité fondamentale du théorème 1 :

Corollaire 3. *On a l'égalité suivante :*

$$\sum_{i=1}^{r} e_i f_i = d_y$$

Il s'agit bien de la même égalité, puisque l'on s'est placé au-dessus de 0, qui correspond à la place $\mathfrak{p} = (x)$, dont le corps résiduel est isomorphe à L.

Enfin, nous concluons cette partie en remarquant que les développements de Puiseux permettent de calculer les indices de ramification des places de E/L sur $L(x)$, et ainsi (par la formule d'Hurwitz donnée dans la proposition 4), le genre de la courbe algébrique plane $\mathcal{C}$.

1.2 Point de vue analytique

Soient K un corps de nombres, $\overline{K}$ sa clôture algébrique, et considérons une courbe algébrique plane de $\mathbb{C}^2$,

$$\mathcal{C} = \{(x_0, y_0) \in \mathbb{C}^2 \mid F(x_0, y_0) = 0\},$$

où $F(x,y) = \sum_{k=0}^{d_y} a_k(x) y^k \in K[x,y]$ est un polynôme sans facteur carré. Comme dans la section 1.1, nous noterons d_x et d_y les degrés de F en respectivement x et y, et D son degré total.

Étant donné un point $x_0 \in \mathbb{C}$ du plan complexe, on définit la **fibre** $\mathcal{F}(x_0)$ de F en ce point comme étant l'ensemble des racines complexes du polynôme à une variable $F(x_0, y)$. Nous distinguerons alors les points du plan complexe de la manière suivante : un point x_0 sera dit **critique** si la fibre $\mathcal{F}(x_0)$ contient strictement moins de d_y éléments (c'est-à-dire qu'il y a une racine multiple ou une branche à l'infini au dessus de x_0). Les points critiques sont en nombres

fini, puisque ce sont les racines de $R_F(x) = \texttt{Resultant}_y(F, F_y)$ le résultant de F et F_y (dérivée par rapport à la variable y du polynôme F) en la deuxième variable y. On notera $\{\alpha_1, \ldots, \alpha_n\}$ l'ensemble des points critiques affines de F, et étant donné un point x_0 du plan complexe, on définit $\delta(x_0)$ comme étant la distance entre x_0 et son plus proche point critique (x_0 excepté s'il est lui même un point critique). Les points non critiques seront dits **réguliers**.

Notons $X_0 = \mathbb{C}\backslash\{\alpha_1, \ldots, \alpha_n\}$ l'ensemble $\mathbb{C}$ privé de l'ensemble des points critiques du polynôme F. Si l'on restreint la courbe $\mathcal{C}$ aux points $x_0 \in X_0$, on obtient une surface de Riemann non compacte $\mathcal{R}_0 = \{(x_0, y_0) \in X_0 \times \mathbb{C} \,|\, F(x_0, y_0) = 0\}$ [For81, chapitre 1, section 1].

Remarque 4. *On peut compactifier cette surface de Riemann [Mir95, For81] pour obtenir une surface de Riemann compacte $\mathcal{R}$, homéomorphe à la somme connexe de g tores, où g est le genre de la courbe $\mathcal{C}$.*

Si l'on note

$$\begin{array}{rccc} p_x \ : & \mathcal{R}_0 & \rightarrow & X_0 \\ & (x_0, y_0) & \mapsto & x_0 \end{array}$$

la projection sur la première coordonnée x de la surface de Riemann $\mathcal{R}_0$, alors pour tout point x_0 de X_0, il existe un voisinage V_0 de x_0 dans X_0 tel que $p_x^{-1}(V_0)$ soit constitué de l'union disjointe de d_y ensembles ouverts de $\mathcal{R}_0$. On définit ainsi un revêtement de X_0 à d_y feuillets, que l'on notera $(\mathcal{C}, x)$.

Dans cette partie, nous allons rappeler quelques propriétés bien connues de ce revêtement $(\mathcal{C}, x)$. Nous commençons par définir la notion d'homotopie dans un ouvert $X \subset \mathbb{C}$, définissant ainsi le groupe fondamental d'un tel ensemble X. Puis nous montrerons dans le cas $X = X_0$ comment le groupe fondamental de X_0 permet de définir le groupe de monodromie de la courbe $\mathcal{C}$.

1.2.1 Homotopie dans $X \subset \mathbb{C}$

Nous commençons cette partie en rappelant brièvement quelques résultats de topologie liés à l'homotopie de courbes. Pour plus de détails sur le sujet, le lecteur pourra se référer à [For81, pages 13 à 20]. Nous nous limiterons à considérer l'homotopie de chemins dans X, ouvert de $\mathbb{C}$.

Un **chemin** γ sur X est une courbe paramétrée :

$$\begin{array}{rccc} \gamma \ : [0,1] & \rightarrow & X \\ t & \mapsto & \gamma(t) \end{array}$$

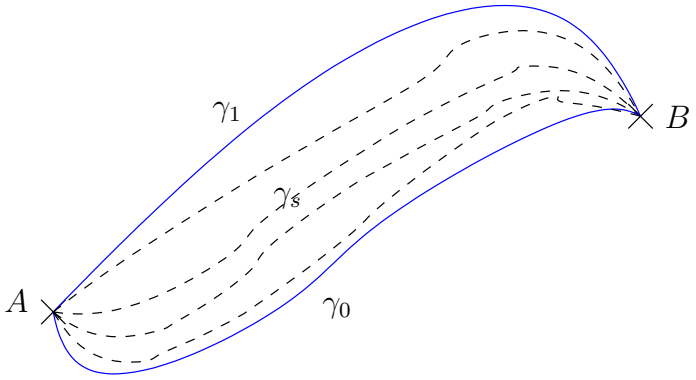

FIG. 1.2 – Déformation de γ_0 à γ_1

Le point $A = \gamma(0)$ sera appelé point de départ du chemin, et le point $B = \gamma(1)$ point d'arrivée. On dira que le chemin γ va de A à B.

Définition 11. *Soient deux points A et B de X. Deux chemins γ_0 et γ_1 allant de A à B sont dits **homotopes** s'il existe une application continue $H : [0,1] \times [0,1] \to X$ qui vérifie les propriétés suivantes :*

$$\begin{array}{ll} (i) & H(t,0) = \gamma_0(t) \quad \forall t \in [0,1] \\ (ii) & H(t,1) = \gamma_1(t) \quad \forall t \in [0,1] \\ (iii) & H(0,s) = A \text{ et } H(1,s) = B \quad \forall s \in [0,1] \end{array}$$

Remarque 5. *Si l'on pose $\gamma_s(t) := H(t,s)$, alors chaque γ_s est un chemin de A à B. La famille de chemins $(\gamma_s)_{0 \leq s \leq 1}$ est alors appelée la déformation du chemin γ_0 au chemin γ_1, ou encore l'homotopie de γ_0 à γ_1. La figure 1.2 illustre ce phénomène.*

Théorème 4. *Soient A et B deux points de X. Alors la notion d'homotopie est une relation d'équivalence sur l'ensemble des chemins allant de A à B.*

Démonstration. voir [For81, théorème 3.2] □

Un chemin $\gamma : [0,1] \to X$ vérifiant $\gamma(0) = \gamma(1)$ est appelé un **lacet**. Un lacet ayant A pour point de départ et d'arrivée est **homotopiquement nul** s'il est homotope au chemin constant égal à A (l'application constante $\gamma_0 : [0,1] \to X$ tel que $\forall t \in [0,1],\ \gamma_0(t) = A$).

Définition 12. *Soient A, B et C trois points de l'ensemble X, γ un chemin de A vers B et γ' un chemin de B vers C.*

1. *Le* ***chemin produit*** $\gamma \cdot \gamma' : [0,1] \to X$ *de* A *vers* C *est défini par :*

$$\gamma \cdot \gamma'(t) := \begin{cases} \gamma(2t) & si \ \ 0 \leq t \leq \frac{1}{2} \\ \gamma'(2t-1) & si \ \ \frac{1}{2} \leq t \leq 1. \end{cases}$$

2. *Le* ***chemin réciproque*** $\gamma^{-1} : [0,1] \to X$ *de* B *vers* A *est défini par :*

$$\gamma^{-1}(t) := \gamma(1-t) \ \ \forall\, t \in [0,1].$$

En d'autres termes, $\gamma \cdot \gamma'$ correspond à la concaténation des chemins γ et γ', et γ^{-1} correspond au chemin γ, mais dans le sens opposé.

Théorème 5. *Soit* $X \subset \mathbb{C}$ *et* $a \in X$*. L'ensemble* $\pi_1(X,a)$ *des classes d'homotopie des lacets de* X *avec pour point de base* a *associé à l'opération de produit de chemins forment un groupe.*

Démonstration. voir [For81, théorème 3.8] □

Définition 13. $\pi_1(X,a)$ *est le* ***groupe fondamental*** *de* X *de base* a*.*

Nous concluons cette partie en donnant le groupe fondamental de l'ensemble X_0 : on choisit a tel que pour tout $i \neq j$, les points a, α_i et α_j ne soient pas alignés, et l'on ordonnera les points $\alpha_1, \ldots, \alpha_n$ par arguments croissants par rapport à a : $i < j$ si et seulement si $\arg(b_i - a) < \arg(b_j - a)$, où arg est la fonction qui a un nombre complexe associe son argument (n'importe quel choix de détermination convient). La figure 1.3 illustre un tel choix.

Théorème 6. *Soient* a *un point de base comme ci-dessus. Si l'on note* $X_0 = \mathbb{C}\backslash\{\alpha_1, \ldots, \alpha_n\}$*, alors le groupe fondamental de* X_0 *avec pour point de base* a *est un groupe libre engendré par* n *lacets* $\gamma_1, \ldots, \gamma_n$ *homotopes dans* X_0 *à ceux de la figure 1.3 : chaque lacet* γ_i *fait le tour du point critique* α_i*, et d'aucun autre point critique* α_j*.*

1.2.2 Groupe de monodromie d'une courbe algébrique plane

Dans cette partie, nous considérons l'ouvert $X_0 = \mathbb{C}\backslash\{\alpha_1, \ldots, \alpha_n\}$ du plan complexe privé de l'ensemble des points critiques. On considère un point de base a comme décrit précédemment, et nous gardons les hypothèses sur l'ordre des points critiques, ainsi que les notations introduites depuis le début de la section 1.2. Le revêtement $(\mathcal{C}, x)$ vérifie la propriété de **relèvements de**

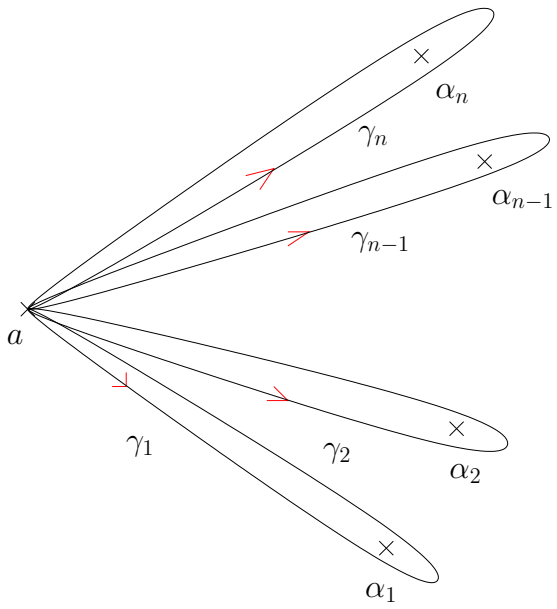

FIG. 1.3 – Lacets engendrant $\pi_1(X_0, a)$

chemins : pour tout chemin $\gamma : [0,1] \to X_0$ et tout antécédent $y_0 \in \mathcal{F}(\gamma(0))$ de $\gamma(0)$, il existe un unique chemin γ_0 sur $\mathcal{R}_0$ tel que $\gamma_0(0) = y_0$ et $p_x \circ \gamma_0 = \gamma$.

Notons $\mathcal{F}(a) = \{y_1, \ldots, y_{d_y}\}$ la fibre en le point de base a. Si γ est un lacet dans X_0 ayant pour point de base a, alors pour tout $1 \leq i \leq d_y$, le relèvement γ_i du lacet γ sur $\mathcal{R}_0$ tel que $\gamma_i(0) = y_i$ est un chemin ayant pour point de départ y_i et pour point d'arrivée $y_j \in \mathcal{F}(a)$, avec j qui n'est pas forcément égal à i. Ce point d'arrivée dépend uniquement du point de départ y_i et de la classe d'homotopie dans X_0 du chemin γ. De plus, comme les chemins γ_i sont uniques, on a $\mathcal{F}(a) = \{\gamma_i(1)\}_{1 \leq i \leq d_y}$. Ainsi, le chemin γ engendre une permutation σ de $\{1, \ldots, d_y\}$ telle que :

$$\gamma_i(1) = y_{\sigma(i)} = \gamma_{\sigma(i)}(0).$$

On obtient ainsi un morphisme du groupe fondamental de X_0 dans $\mathcal{S}_{d_y}$:

$$\begin{array}{rccc} \Psi & : & \pi_1(X_0, a) & \to & \mathcal{S}_{d_y} \\ & & \gamma & \mapsto & \sigma. \end{array}$$

L'image de ce morphisme définit un groupe $\mathcal{M}$. Changer le point de base a ou la numérotation de la fibre $\mathcal{F}(a)$ définit un conjugué de $\mathcal{M}$ dans $\mathcal{S}_d$ [Mir95]. Étant donné que nous cherchons n'importe lequel de ces conjugués,

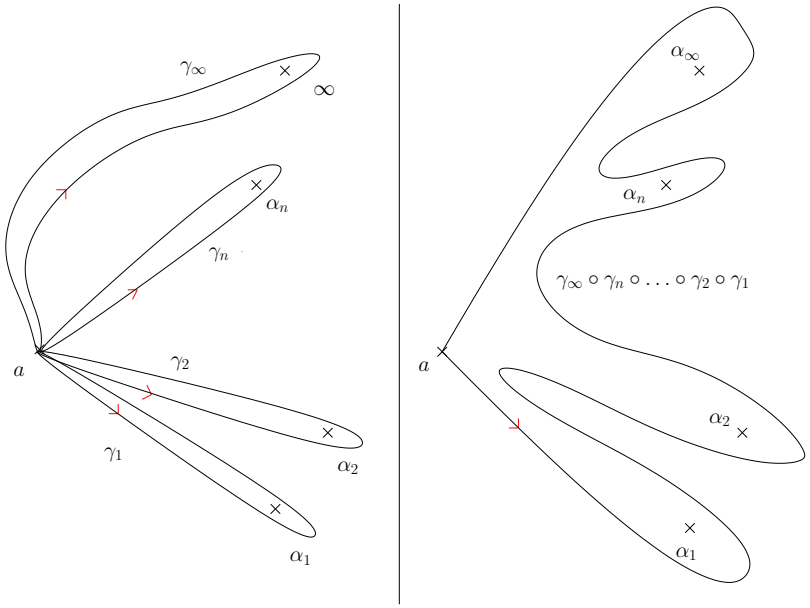

FIG. 1.4 – Homotopie du chemin $\gamma_\infty \cdot \gamma_n \cdot \ldots \cdot \gamma_1$

nous noterons $\mathcal{M}$ quel que soit le point de base choisi et quelque soit la numérotation de la fibre $\mathcal{F}(a)$.

Définition 14. *On appelle* ***groupe de monodromie*** *du revêtement* $(\mathcal{C}, x)$ *ce groupe* $\mathcal{M}$.

Considérons maintenant des lacets $\gamma_1, \ldots, \gamma_n$ homotopes à ceux de la figure 1.3. D'après le théorème 6, ces lacets engendrent le groupe fondamental $\pi_1(X_0, a)$. Nous cherchons donc à calculer les n permutations $\sigma_i = \Psi(\gamma_i), 1 \leq i \leq n$.

Remarque 6. *La définition habituelle du groupe de monodromie prend en compte le point* $x = \infty$*, c'est à dire que l'on considère le revêtement de* $\mathbb{P}^1(\mathbb{C})$ *privé des points critiques de* F*, et défini par* F *et* p_x*. Néanmoins, si le point* $x = \infty$ *est un point critique (c'est-à-dire que* 0 *est un point critique du polynôme* $x^{d_x} F(1/x, y)$*), alors le lacet* γ_∞ *de point de base* a*, qui entoure le point* $x = \infty$ *et aucun autre point critique, engendre une permutation* $\sigma_\infty = \Psi(\gamma_\infty)$ *non triviale. Mais on a alors* $\gamma_\infty \cdot \gamma_n \cdot \ldots \cdot \gamma_1 = Id$ *sur* $\mathbb{P}^1(\mathbb{C}) \backslash \{\alpha_1, \ldots, \alpha_n, \infty\}$*, comme l'illustre la figure 1.4. Le groupe de monodromie est donc bien engendré par les permutations* $\sigma_1, \ldots, \sigma_n$*, que le point* $x = \infty$ *soit un point critique ou non.*

Dans la suite de cette thèse, nous considérerons des chemins γ_i homotopes à ceux de la figure 1.3. Ce choix n'est pas anodin : ce sont précisément ces chemins qui sont utilisés par Tretkoff et Tretkoff [TT84] pour calculer le groupe d'homologie de la surface de Riemann compacte $\mathcal{R}$.

Chapitre 2

Calcul de développements de Puiseux

Soit K un corps de nombre, et $F(x,y) = \sum_{k=0}^{d_y} a_k(x)y^k \in K[x,y]$ un polynôme à deux variables défini sur ce corps, que nous supposerons sans facteur carré. Comme précédemment, nous noterons d_x et d_y les degrés de F respectivement en x et y, et D son degré total. Nous faisons les hypothèses suivantes :

- $d_x > 0$ et $d_y > 0$,
- le polynôme F est séparable et primitif en y.

Dans le chapitre 1, nous avons introduit la notion de développements de Puiseux (classique et rationnel) du polynôme F. Dans ce chapitre, nous décrivons une nouvelle stratégie symbolique-numérique pour calculer une approximation numérique de ces développements de Puiseux. Ceci est un travail en commun avec Marc Rybowicz, dont la partie modulaire a été publiée à l'occasion de la conférence ISSAC 2008 [PR08].

Calculer de tels développements au-dessus d'un point régulier est tout à fait possible à l'aide d'un algorithme purement numérique : on peut appliquer numériquement l'algorithme des itérations quadratiques de Newton [KT78, vzGG99], que nous rappellerons dans la partie 2.1.1. Malheureusement, au-dessus d'un point critique x_0, l'algorithme de Newton-Puiseux [Wal78] ou l'algorithme de Newton-Puiseux rationnel [Duv87, Duv89], que nous présenterons dans les sections 2.1.3 et 2.1.4, ne peuvent être appliqués numériquement. En effet, ces algorithmes nécessitent la connaissance exacte des polygones de Newton et des racines des polynômes caractéristiques (voir la partie 2.1.2). De ce fait, si le point critique, qui est un nombre algébrique, est remplacé par une approximation numérique, alors les polygones de New-

ton deviennent triviaux, et l'algorithme retourne des séries approchées ayant de très petits rayons de convergence, et qui de plus n'apportent pas certaines informations importantes, telles que les indices de ramification.

D'un autre côté, les méthodes symboliques sont en pratique assez lentes : la croissance des coefficients, ainsi que les extensions engendrées par les algorithmes augmentent considérablement les temps de calcul. Ainsi, comme le degré du discriminant Δ_F en y de F est en $O(D^2)$, le point critique peut être un nombre algébrique de grand degré, et les développements de Puiseux peuvent avoir des coefficients dans une extension de K de degré $O(D^3)$. Finalement, quand les coefficients des développements de Puiseux sont exprimés comme des combinaisons linéaires sur $\mathbb{Q}$, la taille des nombres rationnels impliqués peut devenir excessive. De ce fait, une évaluation numérique de ces coefficients nécessite parfois un grand nombre de chiffres de précision, du fait d'annulations numériques importantes.

Exemple 9. *Considérons le polynôme* $F(x,y) = (y^3-x)\,((y-1)^2-x)\,(y-2-x^2)+x^2\,y^5 \in \mathbb{Q}[x,y]$, *de degré total* 7. *Son discriminant en* y *est de la forme* $\Delta_F(x) = x^3 P(x)$, *où* $P(x)$ *est un polynôme irréductible de degré 23 sur* $\mathbb{Q}$. *De ce fait, les développements de Puiseux rationnels calculés au-dessus d'une racine de* $P(x)$ *ont des coefficients dans une extension de degré* 23 *sur* $\mathbb{Q}$. *De plus, des nombres rationnels de 136 chiffres décimaux apparaissent dans le premier terme de ces développements.*

Exemple 10. *Le polynôme* $F(x,y) = y^3 - x^5 - 2\,(10\,x-1)^2$ *a pour discriminant* $\Delta_F(x) = -27\,(x^5 - 200\,x^2 + 40\,x - 2)^2$. *Lors du calcul du groupe de monodromie (voir le chapitre 3), nous avons besoin de calculer les développements de Puiseux au-dessus d'une racine* γ *de* $x^5 - 200\,x^2 + 40\,x - 2$ *à l'ordre* 13/3. *L'appel de fonction* `algcurves[puiseux]`$(F, x=\gamma, y, T, 14/3)$ *de Maple 11 retourne une paramétrisation dont les coefficients incluent des nombres rationnels de* 39 *chiffres décimaux. Dans la stratégie de calcul du groupe de monodromie décrite dans le chapitre 3, nous sommes amenés à évaluer les développements de Puiseux au-dessus de la racine* γ *ayant pour approximation* $-2.990197594 + 5.065348139\,I$ *en le point* $x_0 = -3.608291497 + 3.799011105\,I$. *Si l'on effectue cette évaluation avec 10 chiffres de précision, on obtient une approximation de la fibre dont les éléments sont de l'ordre de* 10^{23}. *La même évaluation faite avec une précision de 20 chiffres décimaux retourne une approximation de la fibre dont les éléments sont de l'ordre de* 10^{12}. *Il faut finalement utiliser une précision de 40 chiffres décimaux pour obtenir une approximation ayant une précision suffisante pour effectuer la connexion avec la fibre* $\mathcal{F}(x_0)$.

Nous verrons qu'il est possible d'établir une telle connexion en utilisant

uniquement 10 *chiffres de précision pour les calculs.*

P. G. Walsh a montré [Wal00] qu'il existe un algorithme qui, pour tout $\epsilon > 0$, calcule la partie singulière des développements de Puiseux en au plus $O(d_y^{32+\epsilon} d_x^{4+\epsilon} (\log \mathrm{ht}(F))^{2+\epsilon})$ opérations binaires, où $\mathrm{ht}(F)$ est la hauteur du polynôme F. Même si elle n'est probablement pas fine, cette borne a tendance à confirmer les observations expérimentales.

Pour contourner ces difficultés, nous procédons de la manière suivante : l'information exacte est obtenue en utilisant l'algorithme de Newton-Puiseux rationnel modulo un nombre premier p bien choisi, nous permettant d'obtenir l'« arbre des polygones » (voir la section 2.1.5). Puis, à l'aide de cette information exacte, nous montrerons comment suivre l'algorithme de Newton-Puiseux numériquement. Ce chapitre sera découpé de la manière suivante :

- Dans la section 2.1, nous décrivons les algorithmes symboliques de calculs de développements en série, ainsi que les définitions nécessaires à leur description.
- Puis nous étudierons les éléments caractéristiques des développements de Puiseux pour obtenir un critère de bonne réduction de ces développements, ceci nous permettant d'obtenir l'arbre des polygones à l'aide de calculs modulaires (section 2.2).
- Ensuite, nous nous intéressons dans la partie 2.3 à la complexité arithmétique et binaire de la partie modulaire de notre algorithme.
- Enfin, nous expliquerons dans la section 2.4 comment utiliser l'arbre des polygones pour pouvoir calculer numériquement les développements de Puiseux.

Notre travail apporte plusieurs contributions :

- Nous généraliserons la notion de polygone de Newton, en définissant les « polygones de Newton génériques » et le « polygones de Newton exceptionnel » (section 2.1.2). Ces notions nous permettront de simplifier les explications et les preuves, notamment en ce qui concerne la réduction modulaire (voir les lemmes 8 et 9).
- Nous définissons la notion d'« arbre de polygones » $\mathcal{T}(F)$, qui contient l'information exacte dont nous aurons besoin pour guider les calculs numériques de la section 2.4. Nous expliquons comment obtenir ce dernier à partir de l'algorithme de Newton-Puiseux rationnel (voir la partie 2.1.5).
- L'étude des développements de Puiseux nous conduit à un critère de bonne réduction, qui nous permet de trouver un nombre premier p et

un idéal premier $\mathfrak{p}$ de corps résiduel isomorphe à $\mathbb{F}_{p^{t_0}}$ tel que, si $\overline{F}$ est la réduction de F modulo $\mathfrak{p}$, alors l'arbre des polygones associé à l'algorithme `RNPuiseux`$(\overline{F}, \mathbb{F}_{p^{t_0}})$ est précisément $\mathcal{T}(F)$ (voir le théorème 12).

- Ensuite, la partie 2.2.4 décrit différents algorithmes pour obtenir ce nombre premier p, en précisant la taille du nombre premier ainsi calculé. On obtient notamment des algorithmes probabilistes qui retournent un nombre premier p de taille petite (c'est-à-dire logarithmique en les entrées pour l'algorithme de type Las Vegas, et également logarithmique en la probabilité d'erreur autorisée pour l'algorithme de type Monte-Carlo ; voir les propositions 20 et 21 de la partie 2.2.4).
- Enfin, nous donnons de nouvelles estimations de complexité pour l'algorithme `RNPuiseux` au-dessus des corps finis, ainsi qu'une complexité pour le calcul de $\mathcal{T}(F)$ au-dessus d'un point critique, puis au-dessus de l'ensemble des points critiques. L'étude de complexité des algorithmes permettant de trouver le nombre premier p, ainsi que les bornes sur la taille de ce nombre premier, nous conduisent à des bornes sur le nombre d'opérations binaires nécessaires pour calculer $\mathcal{T}(F)$, que ce soit au-dessus de 0 ou de chaque point critique.

Tout comme dans la présentation des développements de Puiseux dans le chapitre 1, nous supposerons que le point x_0 considéré est 0, afin de simplifier les notations. Nous rappelons qu'un changement de variable $x \leftarrow x + x_0$ (ou $x \leftarrow 1/x$ si $x_0 = \infty$) permet de se ramener à une telle situation.

Nous concluons cette introduction en résumant les résultats classiques de complexité que nous utiliserons dans cette thèse. Nos études de complexité se contenteront de compter le nombre d'opérations effectuées dans le corps des coefficients. De plus, nous supposons que l'accès aux coefficients des polynômes considérés dans nos algorithmes peut se faire en temps constant (par exemple en utilisant une table à double entrée pour représenter les polynômes bivariés). Enfin, nous utiliserons les notations suivantes :

- Nous noterons $O^{\sim}$ les estimations de complexité ne prenant pas en compte les termes logarithmiques : plus précisément, pour tout couple de fonctions (f, g), la notation $f \in O^{\sim}(g)$ signifie qu'il existe $m > 0$ tel que $f/g \in O(\log(g)^m)$ [vzGG99, définition 25.8].
- $M(N)$ représente classiquement le nombre d'opérations arithmétiques suffisant pour calculer le produit de deux polynômes ayant un degré inférieur ou égal à N. Nous rappelons que $M(N) \in O(N^2)$ si l'on utilise la multiplication naïve, et que ce coût peut être réduit à $M(N) \in O(N \log N \log\log N) \subset O^{\sim}(N)$ en utilisant la multiplication rapide ba-

sée sur la transformée de Fourier [SS71, Sch77, F07] (voir également [BCS97, sections 2.1 et 2.2]).
- Pour simplifier les notations dans la section 2.3, on introduit la quantité $\tilde{M}(N) = M(N)/N$, de telle sorte que $\tilde{M}(N) \in O(N)$ ou $\tilde{M}(N) \in O(\log N \log\log N)$ selon la multiplication utilisée.

2.1 Algorithmes symboliques

Dans cette partie, nous décrivons trois algorithmes pour calculer les développements en série de F au-dessus de 0. Dans le cas où 0 est régulier, nous utiliserons l'algorithme des itérations quadratiques de Newton pour calculer ces développements. Ceci sera utile pour le calcul du groupe de monodromie (voir le chapitre 3), ainsi que pour calculer les termes des développements de Puiseux une fois que l'on aura calculé leur partie singulière. Nous décrivons aussi les algorithmes de Newton-Puiseux classique et rationnel. En effet, ces deux algorithmes seront utilisés dans notre approche symbolique-numérique : en ce qui concerne les calculs modulaires, il est intéressant de faire des calculs dans l'extension la plus petite possible, et le phénomène non contrôlé de croissance des coefficients (voir la section 2.1.7) n'ont pas d'impacts sur les corps finis ; d'un autre côté, les calculs numériques n'impliquent pas d'extension, et il est donc préférable d'utiliser l'algorithme classique. Nous expliquerons également comment les coefficients calculés par ces deux méthodes sont liés, et nous ferons quelques remarques sur la croissance des coefficients engendrée par ces algorithmes.

Tout au long de cette partie, L désignera un corps de caractéristique quelconque que nous noterons $\mathrm{car}(L)$, et $F(x,y) \in L[x,y]$ un polynôme tel que la condition (1.1), page 26, soit satisfaite. Nous supposerons dans ce chapitre que les corps L considérés sont tels qu'il existe un algorithme de factorisation de polynôme au-dessus de ces corps. Enfin, nous réutilisons les notations et hypothèses de la section 1.1 du chapitre 1.

2.1.1 Itérations quadratiques de Newton

Dans cette partie, nous supposons que le point 0 est régulier. Nous rappelons alors la description de l'algorithme des itérations quadratiques de Newton [KT78]. Bien que ces résultats soient classiques, nous rappellerons quelques preuves pour une meilleure compréhension de l'algorithme. Cette section est fortement inspirée de [vzGG99, chapitre 9].

Soit y_0 un élément de la fibre $\mathcal{F}(0)$. Comme 0 est un point régulier, la dérivée en y du polynôme F ne s'annule pas en $(0, y_0)$: $F_y(0, y_0) \neq 0$. Ainsi, $F_y(x, y_0)$ est inversible modulo x. L'algorithme des itérations quadratiques de Newton vient principalement du résultat classique suivant :

Proposition 8 (Convergence quadratique de l'itération de Newton)**.**

Soient $\varphi(y) \in L[[x]][y]$, $u \in L[[x]]$ telle que $\varphi(u) = 0$ et N un entier strictement positif.

Si $\varphi'(\widetilde{u}^{N-1})$ est inversible modulo x^N, alors on a :

$$\widetilde{u}^{2N-1} \equiv \widetilde{u}^{N-1} - \frac{\varphi(\widetilde{u}^{N-1})}{\varphi'(\widetilde{u}^{N-1})} \mod x^{2N},$$

et $\varphi'(\widetilde{u}^{2N-1})$ est inversible modulo x^{2N}

Lemme 2. *Soit N un entier strictement positif et $u \in L[x]$. Alors u est inversible modulo x^N si et seulement si il est inversible modulo x.*

Démonstration. voir [vzGG99, corollaire 9.13]. □

Démonstration. (de la proposition 8)
Comme $\varphi'(u)$ est inversible modulo x^{2N} (voir le lemme 2), le polynôme $v = \widetilde{u}^N - \frac{\varphi(\widetilde{u}^N)}{\varphi'(\widetilde{u}^N)} \mod x^2 N$ est bien défini. En utilisant un développement de Taylor à l'ordre 2 de φ en $y = v$, on a :

$$\varphi(v) = \varphi(\widetilde{u}^N) - \frac{\varphi(\widetilde{u}^N)}{\varphi'(\widetilde{u}^N)}\varphi'(\widetilde{u}^N) + \left(\frac{\varphi(\widetilde{u}^N)}{\varphi'(\widetilde{u}^N)}\right)^2 \psi(\widetilde{u}^N)$$

pour un certain polynôme $\psi \in L[x][y]$. En considérant cette égalité modulo x^{2N}, comme x^{2N} divise $\left(\frac{\varphi(\widetilde{u}^N)}{\varphi'(\widetilde{u}^N)}\right)^2$, on obtient $\varphi(v) \equiv 0 \mod x^{2N}$, et donc $v = \widetilde{u}^{2N-1}$. Enfin, étant donné que $\widetilde{u}^{2N-1} \equiv \widetilde{u}^{N-1} \mod x^N$, on a

$$\varphi'(\widetilde{u}^{2N-1}) \equiv \varphi'(\widetilde{u}^{N-1}) \mod x^N,$$

et le lemme 2 nous permet d'affirmer que $\varphi'(\widetilde{u}^{2N-1})$ est inversible modulo x^{2N}. □

Ainsi, si l'on connaît les N premiers termes d'un développement en série $u(x)$ de F, alors on peut calculer en une itération de Newton les N termes

suivant de la série $u(x)$. Néanmoins, cela nécessite de calculer l'inverse de $\varphi'(u)$ modulo x^{2N}. Ce dernier calcul peut lui aussi se faire en utilisant l'itération de Newton, puisque le calcul de l'inverse de $u(x) \in L[x]$ peut se voir comme la recherche d'une racine du polynôme $u(x)y - 1$:

Lemme 3. *Soit $u \in L[x]$ un polynôme et notons $v \in L[[x]]$ l'inverse de u. Alors :*

$$\widetilde{v}^{2N-1} \equiv 2\widetilde{v}^{N-1} - \widetilde{u}^{2N-1}\left(\widetilde{v}^{N-1}\right)^2 \mod x^{2N}$$

Démonstration. Notons $w = 2\widetilde{v}^{N-1} - \widetilde{u}^{2N-1}\left(\widetilde{v}^{N-1}\right)^2$.

Comme x^N divise $\widetilde{u}^{N-1}\widetilde{v}^{N-1} - 1$, il divise aussi $\widetilde{u}^{2N-1}\widetilde{v}^{N-1} - 1$.

Ainsi, x^{2N} divise $(\widetilde{u}^{2N-1}\widetilde{v}^{N-1} - 1)^2 = -\widetilde{u}^{2N-1}w + 1$.

On a donc $wu \equiv 1 \mod x^{2N}$, et donc $\widetilde{v}^{2N-1} \equiv w \mod x^{2N}$. □

En combinant ces deux résultats et en faisant une étude plus fine des résultats de congruence, on obtient un algorithme décrit dans [vzGG99, chapitre 9]. Celui-ci calcule u modulo x, puis modulo x^2, $x^4, \ldots, x^{2^{l-1}}$ où $l = \lceil \log_2(N) \rceil$, et enfin modulo x^N.

Nous proposons ci-dessous une variante bien connue de cet algorithme, l'algorithme `Newton-quadratique`. Ce dernier est présenté de manière récursive afin de mieux utiliser la décomposition en base 2 de l'entier N : si l'on souhaite $u(x) \mod x^N$, on calculera d'abord $u(x) \mod x^{\lceil \frac{N}{2} \rceil}$ etc. Cette approche bien connue permet de généraliser les résultats de complexité présentés dans le théorème 9.25 de [vzGG99], où N est supposé être une puissance de 2, aux valeurs de N quelconques. En d'autres termes, cela permet de « lisser » la courbe de complexité de l'algorithme.

Enfin, nous traitons également le premier appel à l'algorithme différemment, puisque l'on n'a pas besoin de la donnée de l'inverse de $F_y(x, u(x))$ en sortie, et que l'on veut donc éviter son calcul lors de la dernière étape du calcul. Nous supposons donc être capable de distinguer un appel initial d'un appel récursif, comme par exemple en utilisant un argument booléen supplémentaire.

`Algorithme Newton-quadratique(`F`,`y_0`,`N`)`

`Entrée :`

F : un polynôme à deux variables pour lequel le point 0 est régulier.

y_0 : un élément de la fibre $\mathcal{F}(0)$.

N : un entier > 0.

`Sortie :`

– Si l'appel est récursif, un couple de polynômes (u, v) tel que $F(x, u(x)) \equiv 0 \mod x^N$, $v(x) \equiv 1/F_y(x, u(x)) \mod x^N$ et $u(0) = y_0$.
– Sinon, un polynôme u tel que $F(x, u(x)) \equiv 0 \mod x^N$ et $u(0) = y_0$.

```
Début
  Si N = 1 alors
    Si l'appel est initial
      Retourner y_0
    Sinon
      Retourner (y_0, 1/F_y(0, y_0))
    Fin
  Fin
  (u, v) ← Newton-quadratique(F, y_0, ⌈N/2⌉)
  u ← u − F(x, u)v mod x^N
  Si l'appel est initial alors Retourner u Fin
  Sinon
    v ← 2v − F_y(x, u)v^2 mod x^N
    Retourner (u, v)
  Fin
Fin
```

Théorème 7. *L'algorithme* `Newton-quadratique` *renvoie la sortie escomptée.*

Démonstration. Si $N > 1$, alors $\lceil N/2 \rceil < N$. L'algorithme construit donc une suite strictement décroissante qui converge vers 1 en un nombre fini (égal à $\lceil \log_2(N) \rceil + 1$) d'étapes. De ce fait, l'algorithme termine. De plus, si $N = 1$, alors la sortie $(u, v) = (y_0, 1/F_y(0, y_0))$ (ou $u = y_0$ dans le cas où l'appel est initial) vérifie bien la condition souhaitée.

Soit maintenant un entier $N > 1$. On suppose que l'appel de fonction `Newton-quadratique`$(F, y_0, \lceil \frac{N}{2} \rceil)$ retourne un résultat correct, et nous cherchons maintenant à montrer qu'alors l'appel `Newton-quadratique`(F, y_0, N) retourne lui aussi le résultat attendu. D'après notre hypothèse de récurrence, on a un couple (u, v) tel que $u(0) = y_0$, $F(x, u(x)) \equiv 0 \mod x^{\lceil N/2 \rceil}$ et $v(x) \equiv 1/F_y(x, u(x)) \mod x^{\lceil N/2 \rceil}$. Alors $w(x) := u(x) - F(x, u(x))v(x)$ vérifie (en utilisant la formule de Taylor comme dans la preuve de la proposition 8) :

$$F(x, w(x)) \equiv F(x, u(x))(1 - v(x)\, F_y(x, u(x))) \mod x^N$$

Or, $v(x) \equiv F_y(x, u(x)) \mod x^{\lceil N/2 \rceil}$, donc on a $1 - v(x)\, F_y(x, u(x)) \equiv 0 \mod x^{\lceil N/2 \rceil}$. Comme $F(x, u(x))$ est lui aussi congru à 0 modulo $x^{\lceil N/2 \rceil}$, et

vu que $N \leq 2\lceil N/2 \rceil$, on a donc $F(x, w(x)) \equiv 0 \mod x^N$, ce qui montre le résultat pour le cas où l'appel est initial. Dans le second cas, le lemme 3 nous permet de conclure. Par récurrence, le théorème est donc prouvé. □

Théorème 8. *L'algorithme* `Newton-quadratique` *calcule les N premiers termes d'une série solution de F au-dessus de 0 en $(3d_y+3/2)M(N)+O(d_yN)$ opérations dans le corps L.*

Démonstration. Il suffit de faire un raisonnement similaire à celui effectué dans la preuve du théorème 9.25 de [vzGG99]. □

Enfin, nous notons qu'il est également possible de calculer rapidement de telles séries en utilisant des algorithmes détendus [vdH02].

2.1.2 Polygones de Newton génériques et polynômes caractéristiques

Les notions de polygone de Newton et de polynôme caractéristique sont les outils principaux de l'algorithme de Newton-Puiseux (classique ou rationnel). Dans cette partie, nous rappelons ces définitions bien connues et introduisons une variante de la première notion qui s'avère plus pratique, et qui est cruciale pour le critère de bonne réduction que nous présenterons dans la partie 2.2

Notons $F(x, y) = \sum_{i,j} a_{ij} x^j y^i$ un polynôme de $L[[x]][y]$ tel que x ne divise pas F. Nous définissons la notion de polygone de Newton de F de la manière suivante :

Définition 15. *Pour chaque couple (i, j) de $Supp(F) = \{(i, j) \in \mathbb{N}^2 \,|\, a_{ij} \neq 0\}$, notons $Q_{ij} = \{(i', j') \in \mathbb{R}^2 \mid i' \geq i \text{ et } j' \geq j\}$. Alors le* ***polygone de Newton*** *$\mathcal{N}(F)$ de F est la partie formée des sommets et arêtes finies de l'enveloppe convexe de*

$$Q(F) = \cup_{(i,j)\in Supp(F)} Q_{ij}.$$

Chaque Q_{ij} est un quart de plan supérieur droit. Les arêtes du polygone $\mathcal{N}(F)$ sont donc toutes de pente négative. Plus précisément, si l'on note $\mathcal{I}(F)$ l'entier naturel $v_y(F(0, y))$, on peut décrire $\mathcal{N}(F)$ de la manière suivante :

- Si $F(x, 0) \neq 0$, $\mathcal{N}(F)$ est formé de la suite d'arêtes de $Q(F)$ joignant $(0, v_x(F(x, 0)))$ à $(\mathcal{I}(F), 0)$,
- Si $F(x, 0) = 0$, le point $(0, v_x(F(x, 0)))$ est remplacé par le sommet de plus petite abscisse de $Q(F)$.

En particulier, on ne considère pas comme faisant partie de $\mathcal{N}(F)$ une arête verticale et une arête horizontale de $Q(F)$. Ceci peut très bien définir un polygone de Newton trivial. Par exemple, $F(x, y) = y$ définit un polygone $\mathcal{N}(F) = (1, 0)$.

Ce polygone de Newton permet de calculer l'ensemble des racines de F qui s'annulent en $x = 0$. Il correspond aux appels récursifs de l'algorithme de Newton-Puiseux. Pour calculer l'ensemble des développements de Puiseux au-dessus de $x = 0$, et notamment les développements non définis en $x = 0$, la première étape nécessite d'utiliser une autre définition du polygone de Newton :

Définition 16. *Si l'on note $\mathcal{H}$ l'enveloppe convexe de Supp(F), alors le* ***polygone de Newton initial*** *$\mathcal{N}_0(F)$ est la partie inférieure de $\mathcal{H}$.*

Le polygone de Newton initial peut ainsi contenir des arêtes ayant une pente nulle ou positive. Plus précisément, si l'on note $v = v_x(a_{d_y})$ la valuation x-adique du coefficient de tête de F, et $i_0 = \deg_y(F(0, y))$, alors $\mathcal{N}_0(F)$ est la concaténation de $\mathcal{N}(F)$, $[(\mathcal{I}(F), 0), (i_0, 0)]$ et de la suite d'arêtes de $\mathcal{H}$ joignant $(i_0, 0)$ à (d_y, v). En particulier, si F est unitaire en y, alors $\mathcal{N}_0(F)$ est la concaténation de $\mathcal{N}(F)$ et $[(\mathcal{I}(F), 0), (d_y, 0)]$.

Avant de donner plus d'exemples de polygones de Newton, nous introduisons deux variantes de ces polygones : le « polygone de Newton générique » correspond au polygone de Newton $\mathcal{N}(F)$, et le « polygone de Newton exceptionnel » correspond au polygone de Newton initial $\mathcal{N}_0(F)$. Ces variantes nous permettront de traiter de manière homogène les développements finis et infinis, simplifiera les spécifications des algorithmes (notamment en ce qui concerne les indices de régularité), et est surtout nécessaire pour les résultats que nous donnons sur la réduction modulaire.

Définition 17. *Le* ***polygone de Newton générique*** *$\mathcal{GN}(F)$ du polynôme F est obtenu en restreignant $\mathcal{N}(F)$ aux arêtes ayant une pente supérieure ou égale à -1, et en joignant le point restant le plus à gauche à l'axe vertical par une arête de pente -1.*

En d'autres termes, on ajoute un point fictif $(0, j_0)$ à Supp(F) pour masquer toutes les arêtes ayant une pente strictement infèrieure à -1.

Exemple 11. *Considérons le polynôme $F_1(x, y) = y^7 + x^2y^2 + xy^4 + x^8 + x^6 + y^3x^2 + y^5x + y^3x^4$. Dans la figure 2.1, le support de F_1 est représenté par des croix, $\mathcal{GN}(F_1)$ est dessiné avec un trait continu, alors que la partie masquée de $\mathcal{N}(F_1)$ est représentée par une ligne en pointillés.*

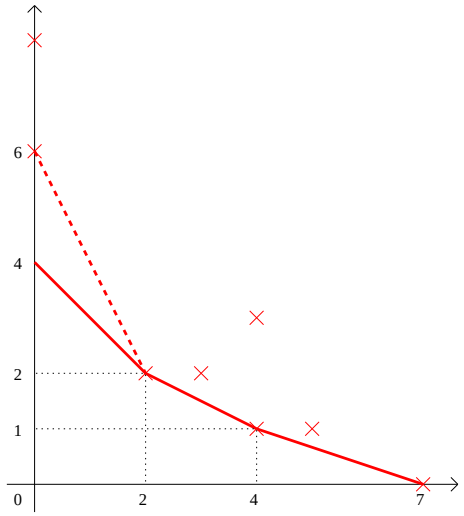

FIG. 2.1 – $\mathcal{GN}(F_1)$ et $\mathcal{N}(F_1)$

Exemple 12. *Considérons le polynôme* $F_2(x,y) = y^8 + 3x^2y^3 + xy^5 + yx^8 + 2x^6 + 4y^2x^3 + y^5x^2 + y^5x^3 + y^3x^4$ *et la figure 2.2. La pente* -1 *est prolongée jusqu'à l'axe vertical.*

Exemple 13. *Considérons le polynôme* $F_3(x,y) = y$. *Alors* $\mathcal{GN}(F_3)$ *est formé de l'unique arête joignant* $(0,1)$ *à* $(1,0)$.

Remarque 7. *Ce concept de polygone de Newton générique est à notre connaissance nouveau dans la littérature. Néanmoins, Mark van Hoeij nous a indiqué que son implantation de l'algorithme de Newton-Puiseux, disponible depuis Maple V.5 (*`algcurves[puiseux]`*), utilise implicitement ce concept. Sa motivation est d'accroître l'efficacité de l'algorithme : à chaque étape récursive, il est possible de faire les calculs modulo une puissance de* x *bien choisie, obtenant ainsi précisément le polygone générique de l'étape suivante. Ce code est utilisé pour le calcul d'anneaux d'entiers [vH94], mais cette technique d'implantation n'a pas été publiée.*

Concernant le polygone de Newton initial, nous introduisons la variante suivante :

Définition 18. *Le* ***polygone de Newton exceptionnel*** $\mathcal{EN}(F)$ *est l'enveloppe convexe inférieure de* $Supp(F) \cup \{(0,0)\}$

En d'autres termes, la partie $\mathcal{N}(F)$ de $\mathcal{N}_0(F)$ est remplacée par l'arête $[(0,0),(\mathcal{I}(F),0)]$, qui prolonge donc éventuellement la pente nulle de $\mathcal{N}_0(F)$. En particulier, si le polynôme F est unitaire, alors $\mathcal{EN}(F) = [(0,0),(d_y,0)]$.

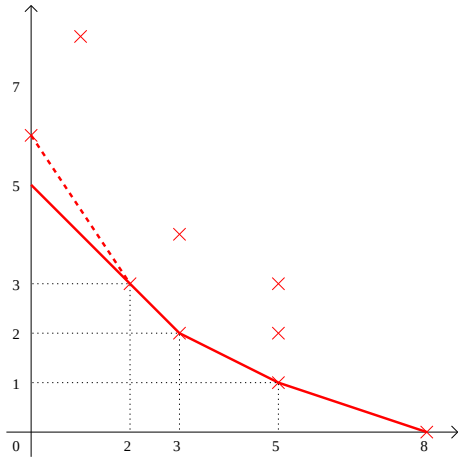

FIG. 2.2 – $\mathcal{GN}(F_2)$ et $\mathcal{N}(F_2)$

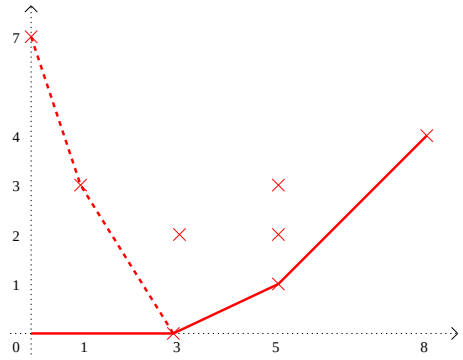

FIG. 2.3 – $\mathcal{EN}(F_4)$ et $\mathcal{N}_0(F_4)$

Exemple 14. *Soit* $F_4(x,y) = x^4y^8 + 3x^2y^3 + xy^5 + 2x^7 + 4x^3y + x^2y^5 - x^3y^5 + 5y^3$. *Sur la figure 2.3,* $\mathcal{EN}(F_4)$ *est dessiné avec un trait continu, alors que la partie masquée de* $\mathcal{N}_0(F_4)$ *est représentée par une ligne en pointillés.*

À une arête Δ d'un polygone de Newton (classique, initial, générique ou exceptionnel) correspondent trois entiers q, m, l, avec q strictement positif et q et m premiers entre eux, tels que Δ appartient à la droite d'équation $qj + mi = l$. Si Δ est l'arête horizontale du polygone, alors $m = l = 0$, et l'on choisit $q = 1$.

Définition 19. *On définit le* ***polynôme caractéristique*** *d'une arête* Δ

comme étant :

$$\phi_\Delta(T) = \sum_{(i,j)\in\Delta} a_{ij} T^{\frac{i-i_0}{q}}$$

où i_0 est la plus petite valeur i telle que $(i,j) \in \Delta$.

Remarque 8. *Si l'on utilise le polygone classique ou initial, alors $\phi_\Delta(T)$ ne peut pas s'annuler en $T = 0$, alors que les variantes que nous proposons permettent une telle annulation si Δ est une arête fictive (ou contient une partie fictive). Dans ce cas, la multiplicité de 0 en tant que racine de $\phi_\Delta(T)$ est la longueur de cette arête fictive (ou de la portion fictive de cette arête) ajoutée.*

Pour conclure cette partie, nous rappelons les liens entre le polygone de Newton de F et les polygones de Newton de ses facteurs dans $L[[x]][y]$:

Lemme 4. *Si F est un polynôme irréductible de $L[[x]][y]$ et $F(0,0) = 0$, alors $\mathcal{GN}(F)$ a une unique arête Δ, et ϕ_Δ a une unique racine.*

Démonstration. Pour le polygone de Newton classique, on peut trouver une telle preuve dans [BK86]. L'extension au polygone de Newton générique est triviale. □

Lemme 5. *Soient F_1 et F_2 deux éléments de $L[[x]][y]$. Alors $\mathcal{GN}(F_1F_2)$ peut se construire en joignant deux à deux les différentes arêtes de $\mathcal{GN}(F_1)$ et $\mathcal{GN}(F_2)$, placées pertinemment, c'est-à-dire par pente croissante à partir du point $(\mathcal{I}(F_1) + \mathcal{I}(F_2), 0)$. De plus, le polynôme caractéristique d'une arête Δ de pente $-m/q$ de $\mathcal{GN}(F_1F_2)$ est le produit des polynômes caractéristiques associés aux arêtes de pente $-m/q$ de $\mathcal{GN}(F_1)$ et $\mathcal{GN}(F_2)$. En particulier, si $F_1(0,0) \neq 0$ (cas où $\mathcal{GN}(F_1)$ est réduit au point $(0,0)$), alors $\mathcal{GN}(F_1F_2) = \mathcal{GN}(F_2)$.*

Démonstration. Pour les polygones de Newton classiques, voir [BK86]. Pour les polygones de Newton génériques, on peut procéder de la manière suivante : si nécessaire, on ajoute un monôme cx^{n_1} (respectivement cx^{n_2}) à F_1 (respectivement F_2), où c est une constante indéterminée, de telle sorte que $\mathcal{GN}(F_i) = \mathcal{N}(F_i)$. Ensuite, il suffit d'appliquer le résultat dans le cas classique, puis de poser $c = 0$ pour obtenir $\mathcal{GN}(F_1F_2)$. □

Les algorithmes de Newton-Puiseux (classique et rationnel) décrits dans les deux sections suivantes effectuent des changements de variable successifs, déterminés par les triplets (q, m, l) correspondant aux arêtes Δ des polygones de Newton, ainsi qu'aux racines de ϕ_Δ. Ils retournent un ensemble de triplets $\{(G_i(x,y), P_i(x), Q_i(x,y))\}_i$ tels que :

- $G_i \in \overline{L}[x,y]$,
- $P_i(x)$ est un monôme de la forme $\lambda_i x^{e_i}$ avec $\lambda_i \in \overline{L}$,
- $Q_i(x,y) = Q_{i0}(x) + y\,x^{r_i}$, où r_i est l'indice de régularité du développement, et $(P_i(T), Q_{i0}(T))$ est la partie singulière de la paramétrisation de F,
- il existe des entiers L_i tels que l'on ait :

$$G_i(x,y) = F(P_i(x), Q_i(x,y))/x^{L_i},\ G_i(0,0) = 0 \text{ et } G_{iy}(0,0) \neq 0.$$

Par le théorème des fonctions implicites, la troisième condition nous assure qu'il existe une unique série S telle que $G_i(x, S(x)) = 0$ et $S(0) = 0$. La paramétrisation de F correspondante est alors $R_i(T) = (P_i(T), Q_i(T, S(T)))$. On calculera cette série S à l'aide de l'algorithme `Newton-quadratique` décrit dans la section 2.1.1. Enfin, il est possible que y divise G_i, ce qui correspond à un développement fini de F.

2.1.3 Algorithme de Newton-Puiseux classique

Nous commençons par donner une variante de l'algorithme de Newton-Puiseux classique [Wal78] pour calculer la partie singulière des développements de Puiseux. Notre algorithme numérique (voir la section 2.4) est basé sur cette méthode. Cette version de l'algorithme retourne exactement un représentant pour chaque cycle au-dessus de 0. L'algorithme est donné de manière récursive et sa sortie est donnée sous une forme paramétrique, comme décrit ci-dessus. Le premier appel (non récursif) à l'algorithme doit être traité séparément (il nécessite l'utilisation de $\mathcal{EN}(F)$ à la place de $\mathcal{GN}(F)$, voir la partie 2.2 pour une explication). Là aussi, on suppose que l'on est capable de distinguer un appel initial d'un appel récursif.

`CNPuiseux(`F`)`
`Entrée :`
  F : *Un polynôme sans facteur carré dans* $\overline{L}[x,y]$*, de degré* $d_y \geq 1$*, tel que* $F(0,y) \neq 0$.
`Sortie :`
  Un ensemble de triplets $\{[G_i, P_i, Q_i]\}_i$ *représentant :*
  - *les cycles de* F *au-dessus de 0, pour l'appel initial,*
  - *les cycles de* F *qui s'annulent en* $x = 0$*, pour les appels récursifs.*
`Début`
  `Si l'appel est récursif alors`
    `Si` $\mathcal{I}(F) = 1$ `alors Retourner` $\{[F,x,y]\}$ `Fin`
    $\mathcal{N} \leftarrow \mathcal{GN}(F)$

```
  Sinon
    N ← EN(F)
  Fin
  R ← {}
  Pour chaque arête Δ de N faire
    Calculer q, m, l et φ_Δ
    Pour chaque racine (distincte) ξ de φ_Δ faire
      α ← ξ^{1/q}
      F_0(x,y) ← F(x^q, x^m(α+y))/x^l
      Pour chaque [G,P,Q] dans CNPuiseux(F_0) faire
        R ← R ∪ {[G, P^q, P^m(α+Q)]}
      Fin
    Fin
  Fin
  Retourner R
Fin.
```

Proposition 9. *L'algorithme* `CNPuiseux` *retourne la sortie escomptée. En particulier, il retourne précisément la partie singulière des développements de Puiseux.*

Démonstration. Si l'on remplace les polygones génériques et exceptionnels par les polygones classiques, on obtient l'algorithme classique de Newton-Puiseux (voir par exemple [Wal78]). De plus, le seul changement provoqué par l'ajout des polygones génériques est l'apparition d'éventuelles racines ξ nulles. Dans ce cas, l'algorithme se contente de factoriser x^m dans la série retournée, et la factorisation correspondante est effectuée au niveau du polynôme F. Dans le cas où $\mathcal{I}(F_0) \neq 1$, la sortie est donc inchangée par l'utilisation des polygones génériques. Dans le cas contraire, l'algorithme s'est contenté de calculer le monôme correspondant à l'indice de régularité du développement de Puiseux (voir l'exemple 16). □

Nous allons maintenant maintenant regarder quelques exemples qui nous permettront d'illustrer l'intérêt de l'introduction des variantes du polygone de Newton.

Exemple 15. *Considérons le polynôme* $F(x,y) = y^3 - x^5 \in \mathbb{Q}[x,y]$. *L'algorithme* `CNPuiseux`*(F) rend un triplet avec* $P_1(x) = x^3$ *et* $Q_1(x,y) = x^0(0 + x^3(0+x^2(1+y))) = x^5 + x^5\,y$. *Le premier coefficient nul vient du polygone de Newton exceptionnel* $\mathcal{EN}(F) = [(0,0),(0,3)]$. *Le second correspond à l'arête fictive de* $\mathcal{GN}(F)$ *introduite durant le premier appel récursif. Le fait d'ajouter*

ces nombres 0 *dans les séries peut paraître inefficace, mais cette astuce n'a pas d'impact sur la complexité et simplifiera les preuves dans la partie 2.2. En pratique, on peut néanmoins utiliser le polygone de Newton classique si nécessaire.*

Exemple 16. *Considérons à nouveau le polynôme* $F(x,y) = (y-1-2x-x^2)(y-1-2x-x^7) \in \mathbb{Q}[x,y]$ *de l'exemple 7 du chapitre 1. En appliquant l'algorithme* `CNPuiseux`*, on obtient deux triplets avec :*

$$\begin{aligned}(P_1,Q_1) &= (x, x^0(1+x(2+x(1+y))) = (x, 1+2x+x^2+x^2y)\\ (P_2,Q_2) &= (x, x^0(1+x(2+x(0+y))) = (x, 1+2x+x^2y)\end{aligned}$$

On peut remarquer ici que l'utilisation du polygone de Newton générique nous permet d'obtenir directement l'indice de régularité 2 *de la série* $x+x^7$ *dans* F*. L'utilisation du polygone de Newton classique ne fournit pas directement une telle information.*

Les entiers q, m et l viennent directement des polygones de Newton. Ils doivent être calculés de manière exacte, puisque par exemple l'entier q contribue à l'indice de ramification, dont nous avons besoin dans notre stratégie de calcul du groupe de monodromie (voir le chapitre 3). De plus, il est trivial que la plupart du temps, si l'on remplace ξ par une approximation numérique, le changement de variable dans `CNPuiseux` définit un polynôme F_0 qui a un polygone de Newton trivial réduit à un unique point $(0,0)$. Il n'est alors pas facile de retrouver le véritable polygone, puisqu'il faut alors décider quels coefficients de F_0 sont des approximations de 0 (et doivent donc être ignorés). Enfin, on a la proposition suivante :

Proposition 10. *Soit* F *un polynôme satisfaisant les hypothèses d'entrée de* `CNPuiseux`*.*

- *L'entier* $\mathcal{I}(F)$ *correspond au nombre de développements de Puiseux de* F *au-dessus de* 0 *qui s'annulent en* $x=0$*.*
- *L'entier* $\mathcal{I}(F_0)$ *est égal à la multiplicité de* ξ *dans* ϕ_Δ*.*

Démonstration. Voir [Duv89] □

La seconde propriété de la proposition 10 montre que ϕ_Δ possède en général des facteurs multiples. Si l'on travaille avec des polynômes approchés, déterminer les racines distinctes de ϕ_Δ et leur multiplicité peut être problématique. Néanmoins, si l'on suppose que l'ensemble des polygones de Newton successifs est obtenu par un autre moyen (ce qui signifie, d'après la proposition 10, que l'on connaît aussi les multiplicités des polynômes caractéristiques) et qu'ils sont alors donnés en entrée, on peut :

1. Extraire des coefficients de F ceux qui sont utiles au calcul de $\mathcal{GN}(F)$: les coefficients se situant en dessous de $\mathcal{GN}(F)$ sont forcément nuls, on peut donc les ignorer.
2. En déduire une approximation de ϕ_Δ.
3. Trouver les grappes de racines de ϕ_Δ correspondant aux multiplicités escomptées.
4. Pour chaque grappe, déduire une approximation de ξ, puis appliquer le changement de variable et procéder récursivement.

Avec une telle approche, on obtiendra des développements de Puiseux approchés ayant des indices de ramification corrects. Pour calculer les données exactes, c'est-à-dire les polygones de Newton successifs, nous utiliserons des calculs modulo un nombre premier p bien choisi (voir la partie 2.2). Puis nous expliquerons dans la section 2.4 comment établir une correspondance entre ces calculs modulaires et les calculs numériques.

2.1.4 Algorithme de Newton-Puiseux rationnel

Nous décrivons maintenant un algorithme, dû à D. Duval [Duv87, Duv89], qui calcule les parties singulières de développements de Puiseux rationnels au-dessus de 0. Nous utiliserons cet algorithme lors de nos calculs modulaires. Nous commençons par donner deux algorithmes intermédiaires, dont nous ne donnons que les entrées et sorties :

`Factorisation(`L`,`ϕ`)`

`Entrée` :
L : *un corps*
ϕ : *un polynôme univarié de* $L[z]$.
`Sortie` : *Un ensemble de couples* $\{(\phi_i, k_i)\}_i$ *tels que les* ϕ_i *sont des polynômes unitaires irréductibles et distincts de* $L[z]$ *et* $\phi = c\prod_i \phi_i^{k_i}$ *avec* $c \in L$.

Nous rappelons que nous avons supposé en introduction de ce chapitre l'existence d'un tel algorithme. En pratique, nous utiliserons l'algorithme `RNPuiseux` sur les corps finis, corps pour lesquels cette hypothèse est vérifiée.

`Bézout(`q`,`m`)`

`Entrée` : q *et* m *deux entiers positifs*
`Sortie` : *Un couple d'entiers* (u, v) *tel que* $uq - mv = 1$. *Si* $q = 1$, *on impose* $v = 0$ *et* $u = 1$.

```
Algorithme RNPuiseux(L,F)
Entrée :
     L : Un corps.
     F : Un polynôme sans facteur carré dans L[x,y], de degré d_y ≥ 1,
         tel que F(0,y) ≠ 0.
Sortie :
     Un ensemble de triplets {[G_i,P_i,Q_i]}_i représentant :
     - les développements de Puiseux rationnels sur L de F au-dessus
       de 0, pour l'appel initial,
     - les développements de Puiseux rationnels sur L de F centrés
       en (0,0), pour les appels récursifs.
Début
  Si l'appel est récursif alors
    Si I(F) = 1 alors Retourner {[F,x,y]} Fin
    N ← GN(F)
  sinon
    N ← EN(F)
  Fin
  R ← {}
  Pour chaque arête Δ de N faire
    Calculer q, m, l et φ_Δ
    (u,v) ← Bézout(q,m)
    Pour chaque (f,k) dans Factorisation(L,φ_Δ) faire
      ξ ← n'importe quelle racine de f
      F_0(x,y) ← F(ξ^v x^q, x^m(ξ^u + y))/x^l
      Pour chaque [G,P,Q] dans RNPuiseux(L(ξ),F_0) faire
        R ← R ∪ {[G, ξ^v P^q, P^m(ξ^u + Q)]}
      Fin
    Fin
  Fin
  Retourner R
Fin.
```

Proposition 11. *L'algorithme* `RNPuiseux` *retourne la sortie escomptée. En particulier, il retourne précisément la partie singulière des développements de Puiseux.*

Démonstration. Le raisonnement est similaire à la preuve de la proposition 9, avec les deux modifications suivantes : si l'on utilise les polygones classiques, on obtient l'algorithme de D. Duval [Duv89] ; de plus, la répartition de la racine ξ dans le polynôme F introduite par D. Duval demande une précaution

supplémentaire quand cette racine est nulle : si v est strictement positif, le changement de variable définit un polynôme univarié en y, et si v est strictement négatif, le changement de variable n'est pas défini. Il faut donc prendre $v = 0$ quand $\xi = 0$. Ceci est bien le cas de part la spécification de l'algorithme `Bézout` : ξ ne peut valoir zéro que quand la pente est entière, c'est-à-dire $q = 1$. □

On peut remarquer que les algorithmes classiques et rationnels sont très similaires : ils ne diffèrent que par la façon dont les racines des polynômes caractéristiques sont traitées, ainsi que dans la définition du polynôme F_0. Ainsi, si l'on applique l'algorithme `RNPuiseux` sur le corps $\overline{L}$, et que l'on change la ligne $(u, v) \leftarrow$ `Bézout`(q, m) de l'algorithme par $(u, v) \leftarrow (1/q, 0)$ au fur et à mesure de l'algorithme, on retrouve l'algorithme `CNPuiseux`.

Exemple 17. *Soit* $F(x, y) = (y^2 - 2x^3)(y^2 - 2x^2)(y^3 - 2x) \in \mathbb{Q}[x, y]$. `RNPuiseux`$(\mathbb{Q}, F)$ *retourne trois développements :*

$$\begin{aligned}(P_1, Q_1) &= (2x^2, x^0(0 + 2x^2(0 + x(2 + y)))) = (2x^2, 4x^3 + 2x^3y)\\ (P_2, Q_2) &= (4x^3, x^0(0 + x(2 + y))) = (4x^3, 2x + xy)\\ (P_3, Q_3) &= (x, x^0(0 + x(\sqrt{2} + y))) = (x, \sqrt{2}x + xy)\end{aligned}$$

Les deux premiers développements ont $\mathbb{Q}$ *comme corps de coefficients, et pour indices de ramifications respectivement* 2 *et* 3. *Le troisième correspond à une place ayant un corps résiduel isomorphe à* $\mathbb{Q}(\sqrt{2})$. *L'algorithme* `RNPuiseux` *appliqué au corps* $\mathbb{Q}(\sqrt{2})$ *retourne un développement supplémentaire :*

$$(P_4, Q_4) = (x, x^0(0 + x(-\sqrt{2} + y))) = (x, -\sqrt{2}x + xy).$$

Dans [Duv87, Duv89], il est précisé que le système D5 [DDD85, Duv87] permet d'éviter les factorisations des polynômes caractéristiques. Dans notre cas, nous utilisons cet algorithme uniquement dans la partie modulaire de notre stratégie, et il existe des algorithmes efficaces pour factoriser un polynôme au-dessus d'un corps fini. De ce fait, étant donné que la taille des premiers p que nous utilisons est petite, il n'est pas problématique de factoriser les polynômes caractéristiques (voir la partie 2.3.1 sur la complexité de la partie modulaire, et la partie 2.2.4 pour la taille des premiers utilisés par notre algorithme).

Remarque 9. *Il est important de noter que, contrairement aux développements de Puiseux classiques, les développements de Puiseux rationnels ne sont pas canoniquement définis. En effet, si l'on remplace* T *par* βT *dans* $R_i(T) = (\tilde{x}_i(T), \tilde{y}_i(T))$, *avec* β *choisi dans le corps des coefficients de* R_i,

alors on obtient un autre développement de Puiseux rationnel correspondant à la même place. Le choix de β peut avoir des conséquences considérables sur la taille des coefficients du développement, et donc sur l'efficacité des algorithmes. Nous détaillerons ce point dans la partie 2.1.6.

2.1.5 Arbre de polygones

À un appel de l'algorithme `RNPuiseux`(L, F), nous associons un arbre étiqueté. Par définition, la **profondeur** d'un sommet v est le nombre d'arêtes sur le chemin allant de la racine à v. En particulier, la racine a une profondeur nulle. Nous étiquetterons les sommets de l'arbre de profondeur paire avec des polygones, et ceux de profondeur impaire par des partitions d'entiers. De la même manière, les arêtes sont étiquetées alternativement avec les arêtes des polygones de Newton et des couples d'entiers (k, f) où k est la multiplicité de la racine ξ et $f = [L(\xi) : L]$. Ainsi, une arête de l'arbre correspond au choix d'une arête du polygone ou bien au choix d'une racine du polynôme caractéristique.

Plus précisément, l'arbre est construit récursivement à partir de la racine comme suit (voir la figure 2.4). Les sommets de profondeur paire correspondent aux appels de fonction. Si f est un polynôme de $L[z]$ ayant pour décomposition sans facteur carré $f = c\prod_{i=1}^{r} {f_i}^{k_i}$ (c'est-à-dire que les k_i sont des entiers positifs deux à deux distincts, et les f_i sont des polynômes de degrés strictement positifs et deux à deux premiers entre eux), alors nous noterons $[f] = ({k_1}^{\deg f_1} \dots {k_r}^{\deg f_r})$ la partition de $\deg f$ définie par cette décomposition (c'est-à-dire que la multiplicité k_i est répétée $\deg f_i$ fois).

- Un sommet v de profondeur paire l est étiqueté avec un polygone $\mathcal{P}$, qui est égal à $\mathcal{EN}(F)$ pour la racine ($l = 0$), et $\mathcal{GN}(F)$ pour les appels récursifs ($l > 0$).
- À chaque arête Δ de $\mathcal{P}$ correspond une arête allant de v à un sommet de profondeur $l + 1$. On étiquette cette arête avec Δ (représenté par les points définissant le segment).
- Un fils (sommet de profondeur $l + 1$) est étiqueté avec la partition $[\phi_\Delta]$.
- À chaque choix de racine ξ de ϕ_Δ effectué par l'algorithme correspond une arête allant du sommet de profondeur $l + 1$ à un sommet de profondeur $l+2$. L'arête est étiquetée avec la paire (k, f) ou k est la multiplicité de la racine ξ et $f = [L(\xi) : L]$.
- Ensuite, on procède récursivement : un sommet de profondeur $l + 2$ est la racine de l'arbre associé à l'appel de fonction `RNPuiseux`$(L(\xi), F_0)$ où F_0 est le polynôme obtenu pour le choix de l'arête Δ et le choix de la

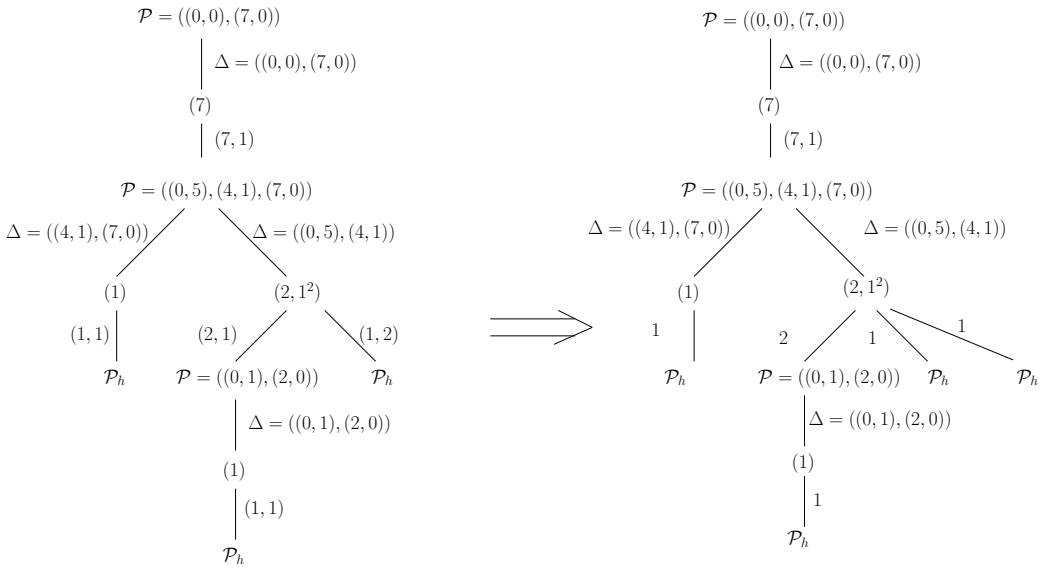

FIG. 2.4 – Les arbres de polygone $\mathcal{RT}(\mathbb{Q}, F)$ et $\mathcal{T}(F)$ pour l'exemple 17.

racine ξ.

Les feuilles de l'arbre sont des sommets de profondeur paire étiquetés par des polygones ayant une unique arête $\mathcal{P}_h = [(0,1),(1,0)]$. On peut remarquer que les racines ξ ne font pas partie de l'arbre. Comme la décomposition sans facteur carré est un sous-produit de la factorisation sur $\overline{L}$, l'arbre étiqueté peut être obtenu sans coût supplémentaire. Si l est la profondeur de l'arbre d'appels de fonctions engendré par `RNPuiseux`(L, F), alors l'arbre étiqueté construit aura une profondeur égale à $2l$.

Pour un appel de fonction `CNPuiseux`(F), on définit un arbre similaire, avec pour seule différence qu'une arête allant d'une partition à un polygone n'est étiquetée que par la multiplicité k ; le corps dans lequel sont effectuées les factorisations étant $\overline{L}$, toutes les extensions de corps sont de degré 1.

Définition 20. *À chaque appel de fonction* `RNPuiseux`(F, L) *(respectivement* `CNPuiseux`(F)*), l'arbre étiqueté associé sera noté* $\mathcal{RT}(L, F)$ *(respectivement* $\mathcal{T}(F)$*). Dans les deux cas, l'arbre est appelé* ***l'arbre de polygones*** *associé à l'appel de fonction.*

Nous montrons dans la partie 2.4 que $\mathcal{T}(F)$ est précisément l'information exacte dont nous avons besoin pour conduire nos calculs numériques.

Proposition 12. *L'arbre* $\mathcal{T}(F)$ *peut facilement être obtenu à partir de l'arbre* $\mathcal{RT}(L, F)$ *comme suit : il suffit de dupliquer* f *fois chaque arête étiquetée*

(k, f) (ainsi que les sous-arbres de cette arête), puis de remplacer l'étiquette (k, f) par l'étiquette k.

Démonstration. Cela vient directement du fait que $\mathcal{T}(F) = \mathcal{RT}(\overline{L}, F)$ (voir l'introduction de la partie 2.1.6 pour cette égalité). □

La figure 2.4 illustre ce procédé.

2.1.6 Des développements de Puiseux classiques aux développements de Puiseux rationnels

On peut remarquer (voir [Duv89]) que les polygones de Newton et les multiplicités des racines des polynômes caractéristiques obtenus au fur et à mesure des calculs sont les mêmes avec les deux algorithmes `CNPuiseux` et `RNPuiseux` ci-dessus. Cette remarque s'étend facilement aux polygones de Newton génériques et exceptionnels, ainsi qu'à leur polynômes caractéristiques associés. Néanmoins, en général, les valeurs prises par les racines non nulles des polynômes caractéristiques diffèrent. Dans cette partie, nous allons étudier les liens entre les coefficients des développements de Puiseux classiques et rationnels. Cela nous apportera une meilleure compréhension de l'algorithme rationnel, notamment en ce qui concerne la croissance des coefficients qu'il engendre (voir la partie 2.1.7), ainsi qu'un critère de réduction pour les développements de Puiseux rationnels (voir le corollaire 6).

Notons $(\beta_1, m_1, q_1), \ldots, (\beta_h, m_h, q_h)$ la suite de triplets rencontrés dans le calcul d'un développement de Puiseux classique en utilisant l'algorithme `CNPuiseux`. Plus précisément, β_i est une racine q_i-ième du i-ième polynôme caractéristique, et $-m_i/q_i$ est la pente de l'arête du polygone de Newton générique correspondant. La sortie de l'algorithme est alors :

$$\begin{aligned} P(x) &= x^{q_1 q_2 \cdots q_h} = x^e \\ Q(x, y) &= x^{m_1 q_2 \cdots q_h}(\beta_1 + x^{m_2 q_3 \cdots q_h}(\beta_2 + \cdots + x^{m_h}(\beta_h + y)\cdots)) \end{aligned}$$

de telle manière qu'un élément du cycle correspondant s'écrit :

$$S(x) = x^{\frac{m_1}{q_1}}(\beta_1 + x^{\frac{m_2}{q_1 q_2}}(\beta_2 + \cdots + x^{\frac{m_h}{q_1 q_2 \cdots q_h}}(\beta_h + \cdots)\cdots)) \tag{2.1}$$

D'un autre côté, notons $(\xi_1, m_1, q_1), \ldots, (\xi_h, m_h, q_h)$ la suite de triplets rencontrés lors du calcul d'un développement de Puiseux rationnel en utilisant l'algorithme `RNPuiseux`. Ici aussi, ξ_i est une racine du i-ième polynôme

caractéristique, et $-m_i/q_i$ est la pente de l'arête du polygone de Newton générique correspondant. Nous noterons (u_i, v_i), $1 \leq i \leq h$ les couples d'entiers retournés par l'algorithme `Bézout`.

Comme nous utilisons les polygones de Newton génériques et exceptionnels à la place des polygones classiques, certains des ξ_i et β_i peuvent être nuls. Si $\xi_i = \beta_i = 0$, alors $q_i = 1$ (car la pente associée est alors égale à -1 ou 0), et donc $v_i = 0$ (voir la procédure `Bézout`). Dans la suite, on définit $0^0 = 1$, de telle manière que si $\xi_i = \beta_i = 0$, alors $\xi_i^{v_i} = \beta_i^{v_i} = 1$, et qu'ainsi les expressions considérées ont un sens et les égalités sont correctes.

Proposition 13. *Il existe un développement de Puiseux classique comme ci-dessus et un ensemble d'entiers $\{e_{ij}\}_{1 \leq j < i \leq h}$ tels que :*

$$\xi_i = \beta_i^{q_i} \prod_{j=1}^{i-1} \beta_j^{v_j e_{ij}}.$$

Démonstration. Posons $x_0 = x$ et $y_0 = y$ et considérons les transformations effectuées par l'algorithme :

$$\begin{array}{rcl} x_{i-1} & = & \xi_i^{v_i} x_i^{q_i} \\ y_{i-1} & = & x_i^{m_i}(\xi_i^{u_i} + y_i) \end{array}$$

On définit alors (n'importe quel choix pour la racine e-ième est acceptable) :

$$\mu_i = \prod_{j=1}^{i} \xi_j^{-\frac{v_j}{q_j q_{j+1} \cdots q_i}} \quad 1 \leq i \leq h,$$

de telle manière que l'on puisse écrire :

$$x_i = \mu_i x^{\frac{1}{q_1 q_2 \cdots q_i}}.$$

On peut alors exprimer la série tronquée calculée par l'algorithme de la manière suivante :

$$Q(x,0) = x_1^{m_1}(\xi_1^{u_1} + x_2^{m_2}(\xi_2 + x_3^{m_3}(\xi_3^{u_3} + \ldots x_{h-1}^{m_{h-1}}(\xi_{h-1}^{u_{h-1}} + x_h^{m_h}\xi_h^{u_h}) \cdots))).$$

En utilisant la définition des x_i, et en identifiant coefficient par coefficient cette expression avec (2.1), on voit qu'il existe un développement de Puiseux classique tel que :

$$\beta_i \quad = \quad \xi_i^{u_i} \textstyle\prod_{j=1}^{i} \mu_j^{m_j} \qquad (2.2)$$

Si l'on pose $\beta_0 = \xi_0 = 1$, on a donc :

$$\beta_i \beta_{i-1}^{-1} = \xi_{i-1}^{-u_{i-1}} \xi_i^{u_i} \mu_i^{m_i} \quad 1 \leq i \leq h$$

En introduisant $\theta_i = \beta_i \beta_{i-1}^{-1} \xi_{i-1}^{u_{i-1}}$, on a alors :

$$\begin{aligned} \mu_i^{q_i} &= \mu_{i-1} \xi_i^{-v_i} \\ \theta_i &= \xi_i^{u_i} \mu_i^{m_i} \end{aligned}$$

On a donc

$$\xi_i = \theta_i^{q_i} \mu_{i-1}^{-m_i} \tag{2.3}$$

et

$$\mu_i = \theta_i^{-v_i} \mu_{i-1}^{u_i} \tag{2.4}$$

En effet, en utilisant les deux égalités précédentes et la relation de Bézout $u_i q_i - m_i v_i = 1$, on obtient :

$$\begin{aligned} \theta_i^{q_i} \mu_i^{-m_i} &= \xi_i^{u_i q_i} \mu_i^{q_i m_i} \mu_{i-1}^{-m_i} \\ &= \xi_i^{u_i q_i} \xi_i^{-m_i v_i} \\ &= \xi_i \end{aligned}$$

et de même,

$$\begin{aligned} \theta_i^{-v_i} \mu_{i-1}^{u_i} &= \xi_i^{-u_i v_i} \mu_i^{-m_i v_i} \mu_{i-1}^{u_i} \\ &= \xi_i^{-u_i v_i} \mu_i \mu_i^{-u_i q_i} \mu_{i-1}^{u_i} \\ &= \mu_i \end{aligned}$$

La récurrence (2.4) se traduit facilement en :

$$\mu_i = \theta_i^{-v_i} \theta_{i-1}^{-v_{i-1} u_i} \theta_{i-2}^{-v_{i-2} u_{i-1} u_i} \cdots \theta_1^{-v_1 u_2 u_3 \cdots u_i}$$

Et donc, en utilisant (2.3),

$$\xi_i = \theta_i^{q_i} \left(\theta_{i-1}^{v_{i-1}} \theta_{i-2}^{v_{i-2} u_{i-1}} \cdots \theta_1^{v_1 u_2 u_3 \cdots u_{i-1}} \right)^{m_i} \tag{2.5}$$

Finalement, la proposition est montrée en prouvant par récurrence l'énoncé $\mathcal{R}(i)$ suivant :
Il existe un ensemble d'entiers $\{f_{ij}\}_{1 \leq j < i \leq h}$ *tel que :*

$$\theta_i = \beta_i \prod_{j=1}^{i-1} \beta_j^{v_j f_{ij}} \tag{2.6}$$

et il existe un développement de Puiseux classique comme (2.1) et un ensemble d'entiers $\{e_{ij}\}_{1 \leq j < i \leq h}$ *tels que :*

$$\xi_i = \beta_i^{q_i} \prod_{j=1}^{i-1} \beta_j^{v_j e_{ij}}.$$

$\mathcal{R}(1)$ est trivialement vérifié. Soit $i > 1$ et supposons $\mathcal{R}(i-1)$ vraie. On a alors :

$$\theta_i = \beta_i \beta_{i-1}^{-1} \xi_{i-1}^{u_{i-1}} = \beta_i \beta_{i-1}^{-1} \beta_{i-1}^{q_{i-1} u_{i-1}} \prod_{j=1}^{i-2} \beta_j^{v_j e_{i-1,j} u_{i-1}}$$

En posant $f_{ij} = e_{i-1,j} u_{i-1}$ pour $1 \leq j \leq i-2$ et $f_{i,j-1} = m_{i-1}$, on obtient (2.6). L'expression de ξ_i se déduit alors de (2.5) et (2.6). □

Remarque 10. *En supposant que les v_i sont choisis dans $\mathbb{N}$, il est facile de voir que les e_{ij} et les f_{ij} sont alors eux aussi dans $\mathbb{N}$.*

Remarque 11. *En utilisant la formule (2.5) et la définition de θ_i, on obtient facilement une équation de récurrence donnant les ξ_i en fonction des β_i. Néanmoins, cela ne permet pas d'obtenir de formule simple. D'un autre côté, en remplaçant les u_j par leur définition dans (2.2), on obtient une expression des β_i en fonction des ξ_i : si pour $1 \leq i \leq j \leq h$, $s_{ij} = \sum_{k=j}^{i} \frac{m_k}{q_j \cdots q_k}$, alors :*

$$\beta_i = \xi_i^{u_i} \prod_{j=1}^{i} \xi_j^{-v_j s_{ji}}.$$

Pour conclure cette partie, nous donnons l'expression des coefficients des paramétrisations retournées par l'algorithme `RNPuiseux` en fonction des ξ_i. Pour simplifier les expressions, on introduit la notation suivante pour $0 \leq i \leq h-1$:

$$\xi_{(i)} = \xi_{i+1}^{v_{i+1}} \xi_{i+2}^{v_{i+2} q_{i+1}} \xi_{i+3}^{v_{i+3} q_{i+1} q_{i+2}} \cdots \xi_h^{v_h q_{i+1} \cdots q_{h-1}}.$$

On pose également $\xi_{(h)} = 1$. La sortie est alors donnée par :

$$\begin{aligned} P(x) &= \xi_{(0)} x^{q_1 q_2 \cdots q_h} = \xi_{(0)} x^e \\ Q(x,y) &= \xi_{(1)}^{m_1} x^{\frac{m_1}{q_1}} (\xi_1^{u_1} + \xi_{(2)}^{m_2} x^{\frac{m_2}{q_1 q_2}} (\xi_2^{u_2} + \cdots + \xi_{(h)}^{m_h} x^{\frac{m_h}{q_1 q_2 \cdots q_h}} (\xi_h^{u_h} + y) \cdots)). \end{aligned}$$

On en déduit la paramétrisation :

$$\begin{aligned} \tilde{x}(T) &= \xi_{(0)} T^e \\ \tilde{y}(T) &= \xi_1^{u_1} \xi_{(1)}^{m_1} T^{m_1 q_2 \cdots q_h} + \\ &\quad \xi_2^{u_2} \xi_{(1)}^{m_1} \xi_{(2)}^{m_2} T^{m_1 q_2 \cdots q_h + m_2 q_3 \cdots q_h} + \\ &\quad \cdots \\ &\quad \xi_h^{u_h} \xi_{(1)}^{m_1} \xi_{(2)}^{m_2} \cdots \xi_{(h)}^{m_h} T^{m_1 q_2 \cdots q_h + m_2 q_3 \cdots q_h + \cdots + m_h} + \cdots \end{aligned} \tag{2.7}$$

2.1.7 Croissance des coefficients des développements de Puiseux rationnels

P.G. Walsh mentionne dans [Wal99], sans néanmoins donner de preuve ou d'exemple, que contrairement aux développements de Puiseux classiques, les développements de Puiseux rationnels calculés par `RNPuiseux` peuvent avoir des coefficients ayant une taille binaire exponentielle. En pratique, nous utilisons l'algorithme `RNPuiseux` uniquement sur les corps finis, et les coefficients ont donc une taille essentiellement bornée par le logarithme du cardinal du corps. De ce fait, cette croissance potentiellement importante des coefficients n'a pas de réel impact sur notre stratégie. Néanmoins, il nous est apparu intéressant d'étudier le fonctionnement de l'algorithme rationnel.

Nous illustrons ici cette croissance des coefficients des développements de Puiseux rationnels, ainsi que l'impact que peut avoir le choix des entiers u et v dans l'algorithme `Bézout` avec l'exemple suivant. Ici, la croissance n'est pas exponentielle, mais conduit néanmoins à des coefficients ayant une taille pathologiquement élevée. Nous ferons quelques remarques à la fin de cette section sur des pistes qui pourraient être creusées pour éviter une telle croissance.

Exemple 18. *Soit a et h des entiers positifs et considérons la paramétrisation suivante, introduite dans un autre contexte [HM87] :*

$$\tilde{x}(T) = T^{2^h}, \tilde{y}(T) = \sum_{k=1}^{h} a\, T^{3\,2^h(1-1/2^k)} \tag{2.8}$$

On définit alors $d = 2^h$ et $F_d(x,y) = Résultant_T(x - \tilde{x}(T), y - \tilde{y}(T))$, de telle sorte que $\deg_y(F_d) = d$ et $\deg_x(F_d) = 3(d-1)$. Il y a une unique place au-dessus de 0 pour F_d, et de ce fait, un système de développements de Puiseux rationnels de F_d au-dessus de 0 contient une unique paramétrisation. On utilise les mêmes notations que dans la précédente sous-section. En appliquant l'algorithme `RNPuiseux` à F_d, on voit que h est le nombre d'appels récursifs, et l'on a $m_i = 3$ et $q_i = 2$ pour $1 \leq i \leq h$.

- ***Première stratégie.*** *On choisit les entiers v_i positifs et minimaux, de telle sorte que $u_i = 2$ et $v_i = 1$. On a $\theta_1 = a$ et, pour $i > 1$, $\theta_i = \alpha_i \alpha_{i-1}^{-1} \xi_{i-1}^2 = \xi_{i-1}^2$, et la relation (2.5) donne :*

$$\xi_i = \xi_{i-1}^4 (\xi_{i-1}^2 \xi_{i-3}^4 \cdots \xi_1^{2^{i-2}} a^{2^{i-2}})^3.$$

  Ainsi, ξ_i est un entier, puissance de a, et une estimation rapide montre que l'exposant est plus grand que 4^i pour $i > 3$. En substituant ceci dans

$\xi_{(0)}$, *on en déduit que* $\xi_{(0)}$ *est une puissance de* a *ayant un exposant supérieur à* $4^h 2^{h-1} > d^3/2$ *pour* $h > 3$. *Finalement, la taille binaire de* $\xi_{(0)}$ *est supérieure à* $d^3/2 \log(a)$ *pour* $h > 3$. *De plus, le comportement est clairement pire pour les autres coefficients de (2.7). Pour* F_{16}, *l'algorithme* `RNPuiseux` *(avec les choix de* v_i *mentionnés ci-dessus) retourne la paramétrisation :*

$$\begin{aligned} \tilde{x}(T) &= a^{3072} T^{16} \\ \tilde{y}(T) &= a^{4609} T^{24} + a^{6913} T^{36} + a^{8065} T^{42} + a^{8641} T^{45} \end{aligned}$$

Si l'on considère la paramétrisation (2.8), on voit que le résultat est loin d'être optimal !

- ***Seconde stratégie.*** *Un autre choix raisonnable est* $u_i = v_i = -1$. *Un tel choix donne des coefficients de taille bien inférieure, puisque les exposants dans (2.5) sont considérablement plus petits, et ont de plus des signes alternés. On obtient ainsi pour* F_{16} *:*

$$\begin{aligned} \tilde{x}(T) &= a^{528} T^{16} \\ \tilde{y}(T) &= a^{793} T^{24} + a^{1189} T^{36} + a^{1387} T^{42} + a^{1486} T^{45} \end{aligned}$$

Le résultat est ici plus satisfaisant que le précédent, mais reste néanmoins fortement éloigné de la solution optimale.

- ***La stratégie de Maple 11.*** *La commande* `algcurves[puiseux]` *retourne un résultat bien pire que les deux précédents :*

$$\begin{aligned} \tilde{x}(T) &= a^{24672} T^{16} \\ \tilde{y}(T) &= a^{37009} T^{24} + a^{55513} T^{36} + a^{64765} T^{42} + a^{69391} T^{45} \end{aligned}$$

Comme remarqué dans [Duv89], une sortie optimale n'est pas forcément atteignable par l'algorithme `RNPuiseux`, quel que soit le choix effectué pour u et v, et ce même pour des cas simples.

Exemple 19. *Soit* $F(x, y) = y^7 - 9x^5$ *un polynôme dans* $\mathbb{Q}[x, y]$. *L'algorithme* `RNPuiseux`$(F, \mathbb{Q})$ *rend une paramétrisation de la forme* $(\tilde{x}(T) = 9^v T^7, \tilde{y}(T) = 9^u T^5)$, *où le couple* $(u, v) \in \mathbb{Z}^2$ *vérifie la relation de Bézout* $7u - 5m = 1$. *Ainsi, quelque soit le choix de* u *et* v, *l'algorithme ne peut rendre la paramétrisation* $(\tilde{x}(T) = 3T^7, \tilde{y}(T) = 3T^5)$.

Il se peut donc que des transformations autres que celles de `RNPuiseux` permettent de mieux contrôler la taille des coefficients. Dans l'exemple 18, on peut ainsi réduire les puissances de a au fur et à mesure des calculs par des substitutions de la forme $T \leftarrow U/a^s$, obtenant ainsi des coefficients plus

petits. Mais nous ne savons pas précisément dans quelle mesure ce genre d'astuce est efficace en général.

Si L est un corps de nombres, P.G. Walsh a montré que l'on peut construire une paramétrisation avec $\tilde{x}(T) = \lambda T^e$, où λ est un coefficient de taille binaire en $O(d_y^{11} d_x^3)$, et dont les autres coefficients sont de taille polynomiale en l'entrée [Wal99]. Mais la construction de Walsh nécessite de connaître a priori les développements de Puiseux classiques. Elle ne fournit pas d'algorithme permettant de construire directement de tels développements rationnels de taille polynomiale. De plus, nous ne savons pas si cette construction est proche de la taille optimale.

Pour résumer, un algorithme qui calcule les développements de Puiseux rationnels avec des coefficients prouvés de petite taille est à l'heure actuelle inconnu.

2.2 Réduction modulaire des développements de Puiseux

Le but de cette partie est, étant donné un polynôme $F \in K[x, y]$ où K est un corps de nombres, de trouver un nombre premier p et un idéal premier $\mathfrak{P}$ divisant p, tel que, en calculant les développements de Puiseux rationnels modulo $\mathfrak{P}$, on ait suffisamment d'information pour calculer $\mathcal{T}(F)$. Nous montrerons dans la partie 2.4 que cette information est suffisante pour guider le calcul numérique des développements de Puiseux.

Pour ce faire, nous commencerons par rappeler la définition de la caractéristique d'un développement de Puiseux. Ceci nous permettra d'étudier la réduction modulaire de ces développements, et ainsi définir un critère de bonne réduction (local et global) du polynôme F. Nous montrerons ensuite qu'un nombre premier p vérifiant ce critère permet de calculer l'arbre $\mathcal{T}(F)$ à l'aide de calculs effectués dans une extension de $\mathbb{F}_p$. Nous donnerons alors différents algorithmes pour trouver un tel premier p, ainsi que des bornes sur la taille de ce premier.

2.2.1 Caractéristique d'un développement de Puiseux

Nous commençons par étudier les relations entre les coefficients d'un développement de Puiseux et le discriminant de son polynôme minimal. Dans cette sous-section, nous nous plaçons dans un corps L de caractéristique quel-

conque, que nous noterons car(L) .

Nous faisons tout d'abord quelques rappels sur les discriminants de polynômes :

Si U est un polynôme univarié, séparable de coefficient dominant u, de degré s, et si l'on note $u_1, \ldots, u_s$ ses racines, alors on a :

$$\Delta_U = \pm u^{2s-2} \prod_{\substack{1 \le i,j \le s \\ i \ne j}} (u_i - u_j) \tag{2.9}$$

On peut facilement en déduire que si U est un polynôme univarié qui admet une factorisation en un produit de polynômes $U = \prod_{i=1}^{r} U_i$, alors :

$$\Delta_U = \prod_{i=1}^{r} \Delta_{U_i} \prod_{\substack{1 \le i,j \le r \\ i \ne j}} \texttt{Resultant}(U_i, U_j). \tag{2.10}$$

Considérons maintenant $S(x) = \sum_{i=n}^{\infty} \beta_i x^{i/e}$, où $n \in \mathbb{Z}$ et $\beta_n \ne 0$, une série de Puiseux classique, d'indice de ramification $e > 1$. On définit alors la suite finie $(B_0, R_0), (B_1, R_1), \ldots, (B_g, R_g)$ de couples d'entiers comme étant :

- $B_0 = n$, $R_0 = e$.
- Si $R_{j-1} > 1$, on pose $B_j = \min\{i > B_{j-1} \mid \beta_i \ne 0 \text{ et } i \not\equiv 0 \mod R_{j-1}\}$ et $R_j = |\text{pgcd}(B_j, R_{j-1})|$.

  Si $R_{j-1} = 1$, on arête et on définit $g = j - 1$.

On peut noter que $g \ge 1$ et $R_g = 1$. On définit également $Q_j = R_{j-1}/R_j$, $M_j = B_j/R_j$ pour $1 \le j \le g$, et H_j le plus grand entier relatif tel que $B_j + H_j R_j < B_{j+1}$ pour $0 \le j \le g-1$.

On a alors $e = Q_1 Q_2 \ldots Q_g$, et on peut voir que M_j est un entier premier à Q_j.

Alors, avec ces notations, quitte à réindexer ses coefficients, S peut être écrit de la manière suivante :

$$\begin{array}{rcccc} S(x) & = & \sum_{j=n}^{H_0} \beta_{0,j} x^j & & \\ & + & \gamma_1 x^{\frac{M_1}{Q_1}} & + & \sum_{j=1}^{H_1} \beta_{1,j} x^{\frac{M_1+j}{Q_1}} \\ & + & \gamma_2 x^{\frac{M_2}{Q_1 Q_2}} & + & \sum_{j=1}^{H_2} \beta_{2,j} x^{\frac{M_2+j}{Q_1 Q_2}} \\ & + & \cdots & + & \cdots \\ & + & \gamma_g x^{\frac{M_g}{Q_1 Q_2 \cdots Q_g}} & + & \sum_{j=1}^{\infty} \beta_{g,j} x^{\frac{M_g+j}{Q_1 Q_2 \cdots Q_g}} \end{array} \tag{2.11}$$

Ici, les monômes de S sont ordonnés par degré (rationnel) croissant.

Définition 21 (voir [Zar81] ou [BK86]). *La* ***caractéristique*** *de S est le uplet d'entiers $(e; B_1, \ldots, B_g)$. Les* ***coefficients caractéristiques*** *sont les éléments de la suite $(\gamma_1, \ldots, \gamma_g)$ et les* ***monômes caractéristiques*** *sont les monômes de S correspondants.*

Exemple 20. *Soit*

$$\begin{aligned} S(x) &= 1 + 3x^{1/2} + 2x + x^{7/6} + 5x^{3/2} + 7x^{5/3} + 2x^{7/4} \\ &= 1 + x^{6/12} + 2x^{12/12} + x^{14/12} + 5x^{18/12} + 7x^{20/12} + 2x^{21/12} \end{aligned}$$

une série de Puiseux d'indice de ramification $e = 12$. On obtient alors :

$$\begin{array}{cc|cc|c} B_0 = 0 & R_0 = 12 & & & \\ B_1 = 6 & R_1 = 6 & Q_1 = 2 & M_1 = 1 & H_0 = 0 \\ B_2 = 14 & R_2 = 2 & Q_2 = 3 & M_2 = 7 & H_1 = 1 \\ B_3 = 21 & R_3 = 1 & Q_3 = 2 & M_3 = 21 & H_2 = 3 \end{array}$$

et donc $g = 3$. On vérifie bien que $Q_1Q_2Q_3 = 2 \cdot 3 \cdot 2 = 12 = e$. On peut alors écrire S de la manière suivante :

$$\begin{aligned} S(x) &= 1 \\ &+ 3x^{1/2} + 2x \\ &+ x^{7/6} + 0 \cdot x^{8/6} + 5x^{9/6} + 7x^{10/6} \\ &+ 2x^{21/12} \end{aligned}$$

La caractéristique de S est alors $(12; 6, 14, 21)$. Les coefficients caractéristiques sont $(3, 1, 2)$, et les monômes caractéristiques sont $3x^{1/2}, x^{7/6}, 2x^{21/12}$.

Autrement dit, les monômes caractéristiques sont les monômes qui font « apparaître » la ramification de la série, en considérant ces monômes par degrés rationnels croissants.

Notons v_x la valuation x-adique et $\mathrm{tc}(S)$ le coefficient de plus bas degré de la série S, de telle sorte que $S(x) = \mathrm{tc}(S)x^{v_x(S)} +$ *termes de valuation supérieure*. On a alors la proposition suivante :

Proposition 14. *Soit $G(x, y)$ le polynôme minimal unitaire sur $\overline{L}((x))$ d'un développement de Puiseux ramifié $S \in \overline{L}((x^{1/e}))$. On suppose que car$(L)$ ne divise pas e. Alors, si $\Delta_G(x)$ est le discriminant de G en la deuxième variable y, on a :*

$$\mathrm{tc}(\Delta_G) = \pm \left(\prod_{i=1}^{g} Q_i^{R_i} \prod_{i=1}^{g} \gamma_i^{R_{i-1} - R_i} \right)^e \tag{2.12}$$

$$v_x(\Delta_G) = \sum_{i=1}^{g} B_i(R_{i-1} - R_i) \tag{2.13}$$

Démonstration. Afin de simplifier les notations, nous noterons, tout au long de cette preuve, $v = v_x(\Delta_G)$ et $\theta = \mathrm{tc}(\Delta_G)$. Les conjugués de S au-dessus de $\overline{L}((x))$ sont $\{S^{[i]}\}_{0\leq i\leq e-1}$, de telle sorte que l'on a :

$$\Delta_G = \prod_{\substack{0\leq i,j\leq e-1\\ i\neq j}} (S^{[i]} - S^{[j]}).$$

À l'aide de cette relation, on peut remarquer que v dépend uniquement de la contribution des termes $x^{M_i/(Q_1\cdots Q_i)} = x^{B_i/e}$. Ainsi, symboliquement parlant (c'est-à-dire si l'on considère les γ_i comme des variables), v est déterminé par l'exposant de γ_i dans θ. Donc si (2.12) est vraie, alors (2.13) l'est aussi, puisque :

$$v = \sum_{i=1}^{g} e\,(R_{i-1} - R_i)\frac{B_i}{e} = \sum_{i=1}^{g} B_i(R_{i-1} - R_i).$$

Pour prouver (2.12), nous procédons par récurrence sur g. Pour tout entier positif r, notons δ_r le discriminant de $x^r - 1$, soit $\delta_r = \pm r^r$.

Si $g = 1$, le développement de Δ_G en puissances fractionnaires croissantes de x est :

$$\begin{aligned}\Delta_G &= \prod_{\substack{0\leq i,j\leq e-1\\ i\neq j}} \left(\gamma_g(\zeta_e^{M_g i} - \zeta_e^{M_g j})x^{M_g/Q_g} + \cdots\right)\\ &= \gamma_g^{e(e-1)} \left(\prod_{\substack{0\leq i,j\leq e-1\\ i\neq j}} (\zeta_e^{M_g i} - \zeta_e^{M_g j})\right) x^{e(e-1)M_g/Q_g} + \cdots\end{aligned}$$

Comme M_g est premier à Q_g et $Q_g = e$, $\zeta_e^{M_g}$ est une racine primitive e-ième de l'unité. On obtient donc $\theta = \delta_e\gamma_g^{e(e-1)} = \pm Q_g^{Q_g}\gamma_g^{e(R_{g-1}-R_g)}$, ce qui montre le cas $g = 1$.

Supposons maintenant que $g > 1$. Pour simplifier les notations, on pose $Q = Q_1$ et $R = R_1 = Q_2\cdots Q_g$. On définit également $H \in \overline{L}((x^{1/Q}))[y]$ comme suit :

$$H = \prod_{i=0}^{R-1} (y - S^{[i\,Q]}).$$

Puisque $[Q] = [Q, e]$ engendre le groupe de Galois de $\overline{L}((x^{1/e}))$ sur $\overline{L}((x^{1/Q}))$, H est le polynôme minimal de S au-dessus de $\overline{L}((x^{1/Q}))$. De plus, la factori-

sation de G sur $\overline{L}((x^{1/Q}))$ est donnée par :

$$G = \prod_{i=0}^{Q-1} H^{[i]}.$$

En utilisant la relation (2.10), on obtient $\Delta_G = \Pi_1 \Pi_2$ avec :

$$\Pi_1 = \prod_{i=0}^{Q-1} \Delta_{H^{[i]}} \qquad \Pi_2 = \prod_{\substack{0 \le i,j \le Q-1 \\ i \ne j}} \texttt{Resultant}(H^{[i]}, H^{[j]}).$$

On a donc besoin de connaître la contribution à θ de Π_1 et Π_2. Nous commençons par Π_1. Soit $U(x,y) = H(x^Q, y)$ le polynôme minimal de $S(x^Q)$ au-dessus de $\overline{L}((x))$. Comme U a pour éléments caractéristiques $(R; B_2, \cdots, B_g)$, notre hypothèse de récurrence nous permet de dire que :

$$\Delta_U = \left(\pm \prod_{i=2}^{g} Q_i^{R_i} \prod_{i=2}^{g} \gamma_i^{R_{i-1}-R_i} \right)^R x^u + \cdots$$

pour un certain entier positif u. On a donc :

$$\Delta_{H^{[j]}} = \zeta_e^{uRj} \left(\pm \prod_{i=2}^{g} Q_i^{R_i} \prod_{i=2}^{g} \gamma_i^{R_{i-1}-R_i} \right)^R x^{\frac{u}{Q}} + \cdots$$

Comme $QR = e$ et $\zeta_e^{uRj} = \zeta_Q^{uj}$, la contribution de Π_1 à θ est :

$$\pm \left(\prod_{i=2}^{g} Q_i^{R_i} \prod_{i=2}^{g} \gamma_i^{R_{i-1}-R_i} \right)^e \tag{2.14}$$

Estimons maintenant la contribution de $\texttt{Resultant}(H^{[i]}, H^{[j]})$. Chaque différence de racines dans le produit qui définit le résultant est de la forme :

$$\gamma_1(\zeta_Q^{M_1 i} - \zeta_Q^{M_1 j})(x^{M_1/Q_1} + \cdots)$$

et on a R^2 différences de ce type. Comme il y a $Q(Q-1)$ résultants dans le produit, et que $\zeta_Q^{M_1}$ est une racine primitive Q-ième de l'unité, on peut conclure que la contribution de Π_2 à θ est :

$$\gamma_1^{R^2Q(Q-1)} \prod_{\substack{0 \le i,j \le Q-1 \\ i \ne j}} \left(\zeta_Q^{M_1 i} - \zeta_Q^{M_1 j} \right)^{R^2} = \gamma_1^{R^2Q(Q-1)} \delta_Q^{R^2} = \pm Q_1^{eR_1} \gamma_1^{e(R_0 - R_1)}$$

En multipliant cette équation avec (2.14), on obtient finalement (2.12). □

La valeur de $v_x(\Delta_G)$ est un résultat bien connu (voir par exemple [Zar81]). Néanmoins, nous n'avons pas trouvé d'expression pour $\mathrm{tc}(\Delta_G)$ dans la littérature.

Exemple 21. *Considérons à nouveau la série $S(x)$ définie dans l'exemple 20. Si G est le polynôme minimal de la série S, alors le monôme de plus bas degré de Δ_G est $-6^{96}\,x^{113}$. On vérifie bien sur cette exemple que :*

$$\sum_{i=1}^{3} B_i(R_{i-1} - R_i) = 6(12-6) + 14(6-2) + 21(2-1) = 113$$

et

$$-\left(\prod_{i=1}^{3} Q_i^{R_i} \prod_{i=1}^{3} \gamma_i^{R_{i-1}-R_i}\right)^{12} = -\left((2^6\,3^2\,2^1)\,(3^{(12-6)}1^{(6-2)}2^{(2-1)})\right)^{12} = -6^{96}$$

2.2.2 Critère de bonne réduction

Dans cette partie, nous allons définir un critère de bonne réduction de F en un idéal premier $\mathfrak{p}$. On considère donc un polynôme $F = \sum_{k=0}^{d_y} a_k(x)y^k \in K[x,y]$, où K est corps de nombres algébriques. On utilise les notations et hypothèses habituelles concernant les degrés de F, et on note $R_F(x) = \texttt{Resultant}_y(F, F_y)$.

Soit $\mathfrak{o}$ l'anneau des entiers algébriques de K. Si $\mathfrak{p}$ est un idéal premier de $\mathfrak{o}$, on note $v_\mathfrak{p}$ la valuation correspondante de K. Cette valuation définit un anneau de valuation de K dont $\mathfrak{p}$ est la place :

$$\mathfrak{o}_\mathfrak{p} = \{\alpha \in K \,|\, v_\mathfrak{p}(\alpha) \geq 0\}.$$

Notons maintenant L l'extension finie de K engendrée par les coefficients des développements de Puiseux de F. La proposition 13 nous permet alors d'affirmer que L contient le corps des coefficients des développements de Puiseux rationnels calculés par `RNPuiseux`.

Si $\mathfrak{O}$ est l'anneau des entiers de L, un idéal premier $\mathfrak{P}$ de $\mathfrak{O}$ définit une valuation $v_\mathfrak{P}$, et donc un anneau de valuation :

$$\mathfrak{O}_\mathfrak{P} = \{\alpha \in L \,|\, v_\mathfrak{P}(\alpha) \geq 0\}.$$

Dans la suite, $\mathfrak{P}$ notera toujours un idéal premier de $\mathfrak{O}$ divisant $\mathfrak{p}$. Si $\alpha \in \mathfrak{O}_\mathfrak{P}$, nous noterons $\overline{\alpha}$ la réduction modulo $\mathfrak{P}$ de α, et nous étendrons cette notation aux polynômes et séries à puissances fractionnaires ayant des coefficients dans $\mathfrak{O}_\mathfrak{P}$. Si $\alpha \in \mathfrak{o}_\mathfrak{p}$, comme $\mathfrak{P}$ divise $\mathfrak{p}$, la réduction modulo $\mathfrak{P}$ et $\mathfrak{p}$ coïncident et nous utiliserons donc là aussi la notation $\overline{\alpha}$.

Définition 22. *Soit p un nombre premier et $\mathfrak{p}$ un idéal premier de $\mathfrak{o}$ divisant p. Si les conditions suivantes sont satisfaites :*
- $F \in \mathfrak{o}_{\mathfrak{p}}[x, y]$,
- $p > d_y$,
- $v_{\mathfrak{p}}(\mathrm{tc}(R_F)) = 0$,

alors on dit que F a une ***bonne $\mathfrak{p}$-réduction locale*** *en $x = 0$.*

On peut bien entendu généraliser cette définition en un point $x = x_0$ quelconque : il suffit de considérer le polynôme $R_F(x + x_0)$ à la place de R_F. Dans la suite, lorsque nous parlerons de bonne $\mathfrak{p}$-réduction locale, sauf précision contraire, nous supposerons que nous sommes en $x = 0$.

Remarque 12. *Si F a une bonne $\mathfrak{p}$-réduction locale, comme $\mathfrak{P}$ divise $\mathfrak{p}$, on a alors $v_{\mathfrak{P}}(\mathrm{tc}(R_F)) = 0$, et F a donc une bonne $\mathfrak{P}$-réduction locale. Nous utiliserons librement cette remarque par la suite.*

De plus, puisque a_{d_y} et Δ_F sont tous les deux à coefficients dans $\mathfrak{o}_{\mathfrak{p}}$, la dernière condition est équivalente aux deux égalités $v_{\mathfrak{p}}(\mathrm{tc}(a_{d_y})) = 0$ et $v_{\mathfrak{p}}(\mathrm{tc}(\Delta_F)) = 0$. On peut noter que $v_{\mathfrak{p}}(\mathrm{tc}(a_{d_y})) = 0$ implique $\deg_y(\overline{F}) = d_y$.

Nous donnons maintenant un résultat fondamental pour notre stratégie de réduction. Celui-ci est la conséquence de deux résultats d'analyse p-adique. Soit $\mathbb{C}_p$ le corps des nombres p-adiques, qui est le completé de la clôture algébrique de $\mathbb{Q}_p$, lui même complété de $\mathbb{Q}$ pour la valeur absolue p-adique (voir par exemple [Rob00] pour plus de précisions sur les corps p-adiques). Nous considèrerons le corps L comme un sous-corps de $\mathbb{C}_p$, par le biais de sa complétion $\mathfrak{P}$-adique. Nous notons $|.|_p$ la valeur absolue de $\mathbb{C}_p$, de telle sorte que :

$$\mathfrak{O}_{\mathfrak{P}} = \{\alpha \in L \,|\, |\alpha|_p \leq 1\}.$$

Enfin, pour tout $\rho \in \mathbb{R}^{+*}$, on définit $D(0, \rho^-) = \{x_0 \in \mathbb{C}_p \mid |x_0|_p < \rho\}$ et $\dot{D}(0, \rho^-) = \{x_0 \in \mathbb{C}_p \mid x_0 \neq 0 \text{ et } |x_0|_p < \rho\}$.

Théorème 9. *Les séries de Puiseux de F au-dessus de zéro convergent p-adiquement dans le disque pointé $\dot{D}(0, \rho^-)$, où ρ est la plus petite valeur absolue p-adique des racines non nulles de R_F.*

Démonstration. Cela vient du théorème 2.1 de [DR79]. □

Proposition 15. *Soit $S = \sum_{i=0}^{\infty} \beta_i x^{i/e} \in \mathbb{C}_p[[x]]$ une série convergente et bornée p-adiquement sur $D(0, 1^-)$. Alors on a :*

$$\sup_{x \in D(0,1^-)} |S(x)|_p = \sup_{i \geq 0} |\beta_i|_p.$$

Démonstration. Voir par exemple [Rob00, section 4.6]. □

Théorème 10. *Si F a une bonne $\mathfrak{p}$-réduction locale, alors les coefficients des développements de Puiseux de F au-dessus de 0 sont dans $\mathfrak{O}_{\mathfrak{P}}$.*

Nous commençons par montrer le lemme suivant :

Lemme 6. *Soit $P(x) = x^m(c_0 + \cdots + c_r x^r) \in \mathfrak{O}_{\mathfrak{P}}[x]$ un polynôme tel que $|c_0|_p = 1$. Alors P n'a pas de racine non nulle dans $D(0, 1^-)$.*

Démonstration. Supposons que $x_0 \in \mathbb{C}_p \backslash \{0\}$ vérifie $|x_0|_p < 1$ et $P(x_0) = 0$. Alors comme $P \in \mathfrak{O}_{\mathfrak{P}}[x]$, on a $|c_i x_0^i|_p < 1$ pour $1 \leq i \leq r$. Or, on a $|a+b|_p = \max\{|a|_p, |b|_p\}$ si $|a|_p \neq |b|_p$. Cela implique $|c_0 + \cdots + c_r x_0^r| = 1$, ce qui est en contradiction avec l'égalité $c_0 + \cdots + c_r x_0^r = 0$. □

Démonstration. (du théorème 10)
Soit $S(x) = \sum_{i=n}^{\infty} \beta_i x^{i/e}$, $n \in \mathbb{Z}$ l'une des séries solutions de F. Comme les coefficients de R_F appartiennent à $\mathfrak{O}_{\mathfrak{P}}$ et que $|\mathrm{tc}(R_F)|_p = 1$, alors, d'après le lemme 6, R_F n'a pas de racine non nulle dans $D(0, 1^-)$. Par conséquent, le théorème 9 nous assure que $S(x)$ converge dans $\dot{D}(0, 1^-)$.

Ensuite, notons $v = v_x(a_{d_y})$. Le polynôme $F_0(x, y) = x^{(d_y-1)v} F(x, y/x^v) \in \mathfrak{o}_{\mathfrak{P}}[x, y]$ a pour coefficient dominant $a(x) = a_{d_y}(x)/x^v$, qui vérifie $|a(0)|_p = 1$, et donc, en effectuant un raisonnement similaire à celui de la preuve du lemme 6 : pour tout $x_0 \in D(0, 1^-)$, $|a(x_0)|_p = 1$.

De plus, $S_0(x) = x^{d_y} S(x)$ est une série solution de F_0 qui converge sur $D(0, 1^-)$. De ce fait, pour tout $x_0 \in D(0, 1^-)$, comme $|a(x_0)|_p = 1$, si $|S_0(x_0)|_p > 1$, alors $S_0(x_0)$ ne peut vérifier l'équation $F_0(x_0, S_0(x_0)) = 0$. $S_0(x)$ est donc bornée par 1 sur $D(0, 1^-)$, et la proposition 15 implique $|\beta_i|_p \leq 1$, et donc $\beta_i \in \mathfrak{O}_{\mathfrak{P}}$. □

Il est important de noter que le théorème 10 est vrai pour n'importe quel $\mathfrak{P}$ divisant $\mathfrak{p}$.

Exemple 22. *Soit $F(x, y) = y^2 - x^3(p + x)$ avec $p > 2$. Les développements de Puiseux au-dessus de 0 sont :*

$$S_{1j}(x) = (-1)^j \sqrt{p} x^{3/2} \left(1 + \frac{x}{p}\right)^{1/2} = (-1)^j \sqrt{p} x^{3/2} \left(1 + \frac{x}{2\,p} - \frac{x^2}{8\,p^2} + \cdots\right).$$

On voit bien que ces séries ne sont pas réductibles modulo p. Néanmoins, notre critère sur le coefficient de plus bas degré du résultant détecte cela. En effet, ici on a $R_F = 4px^3 + 4x^4$, et on voit bien que p divise le coefficient

en x^3 de R_F. Néanmoins, il existe un système de développements de Puiseux rationnels qui est réductible modulo p : $\{\tilde{x} = p\,T^2, \tilde{y} = p^2T^3 + \frac{1}{2}p^2T^5 + \cdots\}$. Mais la réduction modulo p de cette paramétrisation $\{\tilde{x} = 0, \tilde{y} = 0\}$ est triviale et donc non utilisable. De plus, $\{\tilde{x} = T^2/p, \tilde{y} = T^3/p + \frac{1}{2}T^5/p^3 + \cdots\}$ est un autre système de développements de Puiseux rationnels qui n'est pas réductible.

Exemple 23. *Soit $F(x,y) = x\,(p+x)\,y^2 + y + x$. Les développements de Puiseux de F au-dessus de 0 sont donnés par :*

$$S_1(x) = -x - px^3 - x^4 + \cdots \text{ et } S_2(x) = -\frac{1}{p\,x} + \frac{1}{p^2} + \frac{p^3-1}{p^3}x + \cdots$$

La série S_2 n'est évidemment pas réductible modulo p. De plus, le discriminant de F en y est $\Delta_F = -4\,x^3 - 4p\,x^2 + 1$, donc le coefficient de plus bas degré ne s'annule pas modulo p. Néanmoins, le coefficient de plus bas degré du coefficient de tête de F, égal à p, s'annule modulo p. Notre critère détecte donc bien aussi cette non-réductibilité des séries. Cet exemple justifie le fait qu'on ait besoin du résultant de F et F_y, et pas seulement du discriminant de F, pour notre critère de bonne réduction.

Corollaire 4. *Si F a une bonne $\mathfrak{p}$-réduction locale, alors tout facteur unitaire G de $F(x,y)$ dans $\overline{K}((x))[y]$ satisfait $v_{\mathfrak{P}}(\mathrm{tc}(\Delta_G)) = 0$.*

Démonstration. Notons $F = GH$. La relation (2.10) montre que :

$$\mathrm{tc}(\Delta_F) = \pm\mathrm{tc}(\Delta_G)\mathrm{tc}(\Delta_H)\mathrm{tc}(\texttt{Resultant}(G,H))^2.$$

Le théorème 10 nous permet de dire que les coefficients de G et H sont dans $\mathfrak{O}_{\mathfrak{P}}$, et donc $\mathrm{tc}(\Delta_G)$, $\mathrm{tc}(\Delta_H)$ et $\mathrm{tc}(\texttt{Resultant}(G,H))$ aussi. Le résultat est alors une conséquence triviale de $v_{\mathfrak{p}}(\mathrm{tc}(\Delta_F)) = v_{\mathfrak{P}}(\mathrm{tc}(\Delta_F)) = 0$ (voir la remarque 12). □

Corollaire 5. *Si F a une bonne $\mathfrak{p}$-réduction locale, alors les coefficients caractéristiques de tous les développements de Puiseux ramifiés de F au-dessus de 0 ont une valuation $\mathfrak{P}$-adique nulle. En d'autres termes, la réduction modulo $\mathfrak{P}$ préserve les éléments caractéristiques des cycles ramifiés de F au-dessus de 0.*

Démonstration. En appliquant le corollaire 4 à chaque facteur irréductible de $F(x,y)$ dans $\overline{K}((x))[y]$, et en utilisant la proposition 14, on obtient le résultat. □

Il est néanmoins important de noter que l'annulation modulo $\mathfrak{P}$ des coefficients des développements de Puiseux n'est pas totalement contrôlée par notre critère de bonne réduction. En effet, si F est irréductible dans $\overline{K}[[x]][y]$, tous les coefficients qui ne sont pas caractéristiques peuvent s'annuler modulo $\mathfrak{P}$ (il suffit de considérer le polynôme minimal F sur $\mathbb{Q}(x)$ de $S(x) = px+x^{3/2}$, pour lequel on a $R_F = 4x^3$). Si F n'est pas irréductible, notre critère détecte néanmoins l'annulation des coefficients non caractéristiques qui « séparent » les cycles.

Exemple 24. *Soit* $F(x,y) = (y^2 - 2x + (p-1)x^2 + x^3)\,(y^2 - 2x - x^2)$. *Ici, la partie singulière des deux cycles est donnée par :*

$$(\tilde{x}_1(T) = 2\,T^2, \tilde{y}_1 = 2\,T + T^3) \text{ et } (\tilde{x}_2(T) = 2\,T^2, \tilde{y}_2 = 2\,T + (1-p)T^3)$$

Ici, on voit que ce sont les deux coefficients en T^3 *qui séparent les deux cycles. La différence est ici égale à* $\pm p$. *Ainsi, si l'on réduit ces cycles modulo* p, *la différence est annulée, et donc non préservée. Mais cette annulation est détectée par notre critère sur le résultant, puisque* $\mathrm{tc}(R_F) = 64p^4$.

Cette propriété sera formalisée dans le théorème 12.

Théorème 11. *Notons* $\{S_i\}_{1\le i\le s}$ *un ensemble de représentants pour les cycles de* F *au-dessus de* 0. *Si* F *a une bonne* $\mathfrak{p}$*-réduction, alors* $\{\overline{S_i}\}_{1\le i\le s}$ *est un ensemble de représentants pour les cycles de* $\overline{F}$ *au-dessus de* 0.

Démonstration. Les $\overline{S_i}$ satisfont $\overline{F}(x, \overline{S_i}) = 0$ et, puisque $p > d_y$ et $\deg_y(\overline{F}) = d_y$ (voir la remarque 12), $R_{\overline{F}} = \overline{R_F} \neq 0$. Les $\overline{S_i}$ sont donc des racines distinctes de $\overline{F}$. Le corollaire 5 nous assure que l'indice de ramification de $\overline{S_i}$ est égal à l'indice de ramification de S_i, que nous notons e_i. Comme $\sum_{i=1}^{s} e_i = d_y$, on a obtenu un ensemble complet de représentants pour les cycles de $\overline{F}$. □

Nous montrons maintenant que l'algorithme `RNPuiseux` retourne des paramétrisations qui sont utiles quand elles sont réduites modulo $\mathfrak{P}$ (voir l'exemple 22)

Corollaire 6. *Soit* $R(T) = (\lambda T^e, \sum_{i=0}^{r} \mu_i T^{a_i})$ *(avec* $\mu_i \neq 0$*) une paramétrisation retournée par* `RNPuiseux`. *Si* F *a une bonne* $\mathfrak{p}$*-réduction locale, alors les* μ_i *appartiennent à* $\mathfrak{O}_{\mathfrak{P}}$ *et* $v_{\mathfrak{P}}(\lambda) = 0$.

Démonstration. Nous utilisons ici les notations de la partie 2.1.6. Si β_j est un coefficient caractéristique, alors le corollaire 5 nous permet de dire que $v_{\mathfrak{P}}(\beta_j) = 0$. Sinon, β_j est la racine d'un polynôme caractéristique d'une arête

du polygone de Newton ayant une pente entière. On a donc $q_j = 1, v_j = 0$ et $u_j = 1$ (voir la procédure `Bézout`). La proposition 13 implique alors que $v_{\mathfrak{P}}(\xi_i) = q_i v_{\mathfrak{P}}(\beta_i)$ pour tout $1 \leq i \leq h$. De même, on a $v_{\mathfrak{P}}(\xi_{(i)}) = 0$ pour $0 \leq i \leq h$. En particulier, $\lambda = \xi_{(0)}$ vérifie $v_{\mathfrak{P}}(\lambda) = 0$. Enfin, (2.7) montre que $v_{\mathfrak{P}}(\mu_i) = u_i q_i v_{\mathfrak{P}}(\beta_i)$. Si β_i est un coefficient caractéristique, cette dernière valuation est alors nulle. Sinon, elle est égale à $v_{\mathfrak{P}}(\beta_i) \geq 0$ puisque dans ce cas $q_i = u_i = 1$. □

Enfin, il est possible d'appliquer notre critère de bonne réduction locale à chaque place de $K[x]$. Cela nous mène à définir un critère de bonne réduction globale :

Définition 23. *Soit p un nombre premier et $\mathfrak{p}$ un idéal premier de $\mathfrak{o}$ divisant p. Si les conditions suivantes sont satisfaites :*
- *$F \in \mathfrak{o}_{\mathfrak{p}}[x, y]$,*
- *$p > d_y$,*
- *$[R_F] = [R_{\overline{F}}]$ (la factorisation sans carré du résultant est préservée),*

*alors on dit que F a une **bonne $\mathfrak{p}$-réduction** (globale).*

Remarque 13. *Un tel critère est déjà utilisé par M. Rybowicz dans son implantation de l'algorithme de B. M. Trager pour l'intégration des fonctions algébriques [Tra84], disponible depuis Maple V.5. Cette condition était déduite de preuves dans [Eic66, section III.6]. Ce critère a également été porté à l'intention de la communauté du calcul formel par Trager (document non publié), comme une conséquence d'un théorème plus sophistiqué de Fulton [Ful69].*

Proposition 16. *Si F a une bonne $\mathfrak{p}$-réduction globale, alors pour tout point critique $x_0 \in \overline{K}$ de F, et pour toute place $\mathfrak{P}$ de $K(x_0)$ divisant $\mathfrak{p}$, $F(x+x_0, y)$ a une bonne $\mathfrak{P}$-réduction locale en $x = 0$.*

Démonstration. Soit $x_0 \in \mathbb{C}$ un point critique de F. Notons tout d'abord que puisque la forme de la factorisation sans carré du résultant est conservée par réduction modulo $\mathfrak{p}$, on a $\deg_x(R_F) = \deg_x(R_{\overline{F}})$, c'est-à-dire que si l'on note $\mathrm{lc}(R_F)$ le coefficient principal de R_F, alors on a $v_{\mathfrak{p}}(\mathrm{lc}(R_F)) = 0$. De ce fait, x_0 est entier sur $\mathfrak{o}_{\mathfrak{p}}$ et $v_{\mathfrak{P}}(x_0) \geq 0$. On a donc $F(x + x_0, y) \in \mathfrak{O}_{\mathfrak{P}}[x, y]$.

Notons ensuite $R_F = c \prod_i R_i^{k_i}$ la factorisation sans carré et unitaire de R_F. Puisque $v_{\mathfrak{p}}(\mathrm{lc}(R_F)) = 0$, on a $R_i \in \mathfrak{o}_{\mathfrak{p}}[x]$. Posons alors $S = \prod_i R_i$ la partie sans carré de R_F. Alors l'égalité $[R_F] = [R_{\overline{F}}]$ est équivalente à $v_{\mathfrak{p}}(\Delta_S) = 0$. Ainsi, $v_{\mathfrak{P}}(\Delta_S) = 0$, et donc $v_{\mathfrak{P}}(\Delta_{S(x+x_0)}) = 0$, puisque le discriminant d'un polynôme univarié n'est pas modifié par un changement de variable. Ces deux

dernières égalités étant équivalentes à $[R_{F(x+x_0,y)}] = [R_{F(x+x_0,y) \mod \mathfrak{p}}]$, on a en particulier $v_{\mathfrak{p}}(\mathrm{tc}(R_{F(x+x_0,y)})) = 0$. □

De plus, le critère de bonne réduction globale assure également la bonne réduction des développements au-dessus de l'infini, comme l'indique la proposition suivante :

Proposition 17. *Si F a bonne $\mathfrak{p}$-réduction globale, alors $x^{d_x}F(1/x,y)$ a bonne $\mathfrak{p}$-réduction locale en $x = 0$.*

Démonstration. Si F a une bonne $\mathfrak{p}$-réduction globale, alors comme remarqué au début de la démonstration précédente, on a $v_{\mathfrak{p}}(\mathrm{lc}(R_F)) = 0$. Or, $\mathrm{tc}(R_{x^{d_x}F(1/x,y)}) = \mathrm{lc}(R_F)$. □

2.2.3 Réduction de l'arbre des polygones

Si $F \in \mathfrak{o}_{\mathfrak{p}}[x,y]$ et $p > d_y$, les algorithmes de la section 2.1 peuvent être appliqués à la réduction $\overline{F}$ de F modulo $\mathfrak{p}$, de telle sorte que les notations $\mathcal{T}(\overline{F})$ et $\mathcal{RT}(\mathbb{F}_{p^t}, \overline{F})$ ont un sens. Les développements alors calculés ont des coefficients qui appartiennent à une extension finie de $\mathbb{F}_p$.

Le résultat suivant est crucial. Il nous permettra de calculer l'information exacte requise à l'aide de calculs modulaires :

Théorème 12. *Si F a une bonne $\mathfrak{p}$-réduction locale, alors $\mathcal{T}(F) = \mathcal{T}(\overline{F})$.*

Remarque 14. *Une telle correspondance entre $\mathcal{T}(F)$ et $\mathcal{T}(\overline{F})$ ne peut pas être établie aussi simplement si les polygones de Newton classiques sont utilisés à la place des polygones génériques : les coefficients non caractéristiques des développements de Puiseux peuvent s'annuler après réduction modulaire, engendrant ainsi des modifications sur les polygones de Newton classiques.*

Pour prouver le théorème 12, nous commençons par donner plusieurs lemmes :

Lemme 7. *Soit F un polynôme vérifiant :*

(i) $F \in \mathfrak{O}_{\mathfrak{p}}[x,y]$,

(ii) $\deg_y F = d_y > 0$, $F(0,0) = 0$, $F(0,y) \neq 0$,

(iii) *les racines de F sont dans* $\cup_{e>0}\mathfrak{O}_{\mathfrak{p}}((x^{1/e}))$,

(iv) *F n'a pas de racines multiples,*

(v) $v_{\mathfrak{p}}(\mathrm{tc}(R_F)) = 0$.

Notons (q, m, l) les entiers associés à une arête Δ de $\mathcal{GN}(F)$ et ξ une racine de ϕ_Δ. Alors $F_0(x, y) = F(x^q, x^m(\xi + y))/x^l$ vérifie aussi les conditions (i) à (v).

Démonstration. Les conditions $F_0(0, 0) = 0$ et $F_0(0, y) \neq 0$ viennent des propriétés de `CNPuiseux`. Si l'on note $\{Y_i(x)\}_{1\leq i\leq d_y}$ les racines de F, alors les racines de F_0 sont $\{Y_i(x)/x^m - \xi\}_{1\leq i\leq d_y}$. Il est clair que ces racines sont distinctes. Comme $\xi \in \mathfrak{O}_\mathfrak{P}$, les coefficients de F_0 et ses racines le sont aussi. Si l'on note $a(x)$ le coefficient dominant de F, le terme $a(x)y^{d_y}$ devient $a(x^q)x^{md-l}(y + \xi)^{d_y}$, dont le coefficient en y^{d_y} est $a(x^q)x^{md-l}$, qui a le même coefficient de plus bas degré que $a(x)$. Ainsi, puisque le résultant est, à une puissance de $a(x)$ près, un produit de différences de racines, le coefficient de plus bas degré du nouveau résultant est lui aussi inchangé. □

Lemme 8. *Supposons que F soit un polynôme qui vérifie les conditions du lemme 7. Alors :*

(i) $\mathcal{GN}(F) = \mathcal{GN}(\overline{F})$.

(ii) Si Δ est une arête de $\mathcal{GN}(F)$, alors le polynôme caractéristique ϕ_Δ (respectivement $\overline{\phi_\Delta}$) de Δ dans F (respectivement $\overline{F}$) satisfait $[\phi_\Delta] = [\overline{\phi_\Delta}]$ (égalité des multiplicités des racines).

Démonstration. Soient $\{S_i\}_{1\leq i\leq w}$ les cycles de F qui s'annulent en 0, et $\{F_i\}_{1\leq i\leq w}$ leurs polynômes minimaux sur $\overline{K}((x))$. L'hypothèse sur les racines de F implique que $F_i \in \mathfrak{O}_\mathfrak{P}[[x]][y]$. Soit $V = \prod_{i=1}^w F_i$. V est un polynôme unitaire à coefficients dans $\mathfrak{O}_\mathfrak{P}$. On peut alors définir $U \in \mathfrak{O}_\mathfrak{P}[[x]][y]$ tel que $F = UV$. La proposition 10 page 56 nous donne alors $\mathcal{I}(F) = \mathcal{I}(V)$, ce qui implique que $U(0, 0) \neq 0$. Nécessairement, on a $\mathcal{GN}(F) = \mathcal{GN}(V)$. Nous allons commencer par montrer que $\mathcal{GN}(\overline{F}) = \mathcal{GN}(\overline{V})$, ce qui est équivalent à $v_\mathfrak{P}(U(0, 0)) = 0$. La relation (2.10) page 69 montre :

$$\mathrm{tc}(\Delta_F) = \pm \mathrm{tc}(\Delta_V)\,\mathrm{tc}(\Delta_U)\,\mathrm{tc}(\texttt{Resultant}(U, V))^2.$$

V étant unitaire, le dernier résultant est $\pm\prod_i U(x, v_i)$, où v_i parcourt l'ensemble des racines de V. Comme $v_i(0) = 0$, il s'ensuit que $\mathrm{tc}(\Delta_F)$ est le produit d'un puissance de $U(0, 0)$ et d'un élément de $\mathfrak{O}_\mathfrak{P}$. L'hypothèse $v_\mathfrak{P}(\mathrm{tc}(R_F))$ implique $v_\mathfrak{P}(\mathrm{tc}(\Delta_F))$ (remarque 12 page 74), et on obtient ainsi $v_\mathfrak{P}(U(0, 0)) = 0$. Finalement, on a donc $\mathcal{GN}(\overline{F}) = \mathcal{GN}(\overline{V})$.

Maintenant, pour prouver (i), d'après le lemme 5 page 53, il nous reste à montrer que $\mathcal{GN}(F_i) = \mathcal{GN}(\overline{F_i})$. Si $\mathcal{I}(F_i) = 1$, alors $\mathcal{I}(\overline{F_i}) = 1$ puisque F_i est unitaire. De ce fait, $\mathcal{GN}(F_i)$ et $\mathcal{GN}(\overline{F_i})$ sont réduits à l'unique arête $[(0, 1), (1, 0)]$. Supposons maintenant que $\mathcal{I}(F_i) > 1$. On voit facilement que

les hypothèses impliquent $v_{\mathfrak{P}}(\mathrm{tc}(\Delta_{F_i})) = 0$. La proposition 14 page 70 montre alors que S_i et $\overline{S_i}$ ont la même caractéristique. En particulier, $\overline{F_i}$ est irréductible dans $\overline{\mathbb{F}}_p[[x]][y]$, et $\mathcal{GN}(\overline{F_i})$ a une unique arête, dont le polynôme caractéristique a une seule racine. Si l'unique arête de $\mathcal{GN}(F_i)$ a une pente -1, alors il en est de même pour l'unique arête de $\mathcal{GN}(\overline{F_i})$, puisque l'annulation éventuelle modulo $\mathfrak{P}$ de $\mathrm{tc}(F_i(x,0))$ définira la même arête (éventuellement fictive). Si l'unique arête a une pente strictement plus grande que -1, alors $\mathrm{tc}(F_i(x,0))$ est une puissance positive d'un coefficient caractéristique, et ne s'annule donc pas modulo $\mathfrak{P}$. Dans les deux cas, on a $\mathcal{GN}(F_i) = \mathcal{GN}(\overline{F_i})$.

En ce qui concerne (ii), notons Δ l'arête commune à $\mathcal{GN}(F)$ et $\mathcal{GN}(\overline{F})$. Si Δ correspond uniquement aux polynômes irréductibles F_i et $\overline{F_i}$, ϕ_Δ et $\overline{\phi_\Delta}$ ont une unique racine ayant la même multiplicité, puisqu'ils ont le même degré, ce qui nous permet de conclure. Supposons donc que Δ corresponde à au moins deux polynômes irréductibles F_1 et F_2, associés aux racines ξ_1 et ξ_2 de ϕ_Δ. Pour montrer (ii), il nous suffit de montrer que $\xi_1 \neq \xi_2 \Rightarrow \overline{\xi_1} \neq \overline{\xi_2}$. Notons m et q les entiers premiers entre eux tels que la pente de Δ soit égale à $-m/q$. On définit alors $\beta_i = \xi_i^{1/q}$ (n'importe quel choix de racine q-ième convient). Le cycle associé à F_i peut être représenté par la série $\beta_i x^{m/q} + \ldots$. Ainsi, à l'aide de la relation (2.9) page 69, on voit qu'il existe $\delta \in \mathfrak{O}_{\mathfrak{P}}$ tel que $\mathrm{tc}(\Delta_F) = (\beta_1 - \beta_2)\delta$. De ce fait, $v_{\mathfrak{P}}(\beta_1 - \beta_2) = 0$, c'est-à-dire $\overline{\beta_1} \neq \overline{\beta_2}$, et donc $\overline{\xi_1} \neq \overline{\xi_2}$. □

Lemme 9. *Si F a une bonne $\mathfrak{p}$-réduction locale, alors :*

(i) $\mathcal{EN}(F) = \mathcal{EN}(\overline{F})$.

(ii) Notons Δ une arête de $\mathcal{EN}(F)$. Le polynôme caractéristique ϕ_Δ (respectivement $\overline{\phi_\Delta}$) de Δ dans F (respectivement $\overline{F}$) satisfait $[\phi_\Delta] = [\overline{\phi_\Delta}]$.

Démonstration. Notons $\mathcal{EN}(F) = \Delta_0, \ldots, \Delta_{s-1}$ la suite d'arêtes du polygone de Newton exceptionnel. Si $\Delta_k = [(i_k, j_k), (i_{k+1}, j_{k+1})]$, alors par définition de $\mathcal{EN}(F)$ on a $(i_0, j_0) = (0, 0)$ et $(i_s, j_s) = (d_y, v)$, où $v = v_x(a_{d_y})$ est la valuation x-adique du coefficient dominant de F.

Notons maintenant $F(x,y) = \sum_{i,j} a_{ij} x^j y^i$. L'affirmation (i), par définition du polygone de Newton exceptionnel, est équivalente à :

$$v_{\mathfrak{p}}(a_{i_k j_k}) = 0 \text{ pour tout } 1 \leq k \leq s$$

Une première conséquence de la bonne $\mathfrak{p}$-réduction de F est : $v_{\mathfrak{p}}(\mathrm{tc}(a_{d_y})) = 0$ (voir la remarque 12 page 74), c'est-à-dire $v_{\mathfrak{p}}(a_{i_s j_s}) = 0$. Soit maintenant un entier k vérifiant $1 \leq k \leq s-1$, et supposons que $v_{\mathfrak{p}}(a_{i_{k+1} j_{k+1}}) = 0$. Nous allons montrer que cela implique $v_{\mathfrak{p}}(a_{i_k j_k}) = 0$, ce qui prouvera alors (i) par

récurrence sur k. Notons donc $-\frac{m_k}{q_k}$ la pente de l'arête Δ_k, de telle sorte que si ξ_k désigne une racine du polynôme caractéristique ϕ_{Δ_k}, alors il existe une série de Puiseux de F dont le monôme de tête est égal à $\xi_k^{1/q_k} x^{m_k/q_k}$. Nous noterons S_{k,ξ_k} une telle série. De plus, comme $k \geq 1$, on a $\frac{m_k}{q_k} < \frac{m_0}{k_0}$, de sorte que si S_0 désigne une série de Puiseux associée à l'arête Δ_0, on a : $\mathrm{tc}(S_{k,\xi_k} - S_0) = \xi_k^{1/q_k}$. Or, on peut déduire de $v_{\mathfrak{p}}(R_F) = 0$ et de la relation (2.9) page 69 que $v_{\mathfrak{p}}(\mathrm{tc}(S_{k,\xi_k} - S_0)) = 0$, et donc $v_{\mathfrak{p}}(\xi_k) = 0$. Comme $a_{i_k j_k}$ est au signe près le produit de $a_{i_{k+1} j_{k+1}}$ et des racines de ϕ_{Δ_k}, nous pouvons conclure que $v_{\mathfrak{p}}(a_{i_k j_k}) = 0$, ce qui achève la démonstration de (i).

En ce qui concerne (ii), il suffit d'utiliser les mêmes arguments que dans la preuve de l'assertion (ii) du lemme 8. □

Remarque 15. *L'affirmation (i) du lemme 9 n'est pas vraie si le polygone de Newton exceptionnel est remplacé par le polygone générique. Par exemple, si l'on considère le polynôme $F(x,y) = (y+p+x)(y+1+x)$, notre critère de bonne réduction ne détecte pas l'annulation de $F(0,0)$ modulo p. Néanmoins, celui-ci détecte les changements de multiplicités de racines. Cette remarque justifie l'introduction de $\mathcal{EN}(F)$.*

Démonstration. (du théorème 12)

Le lemme 9 nous permet de dire que la racine, les sommets de profondeur 1, ainsi que les arêtes allant jusqu'aux sommets de profondeur 2 de $\mathcal{T}(\overline{F})$ sont étiquetés correctement.

Soit Δ une arête de $\mathcal{EN}(F)$, $mi + qj = l$ la droite qui porte Δ, ξ une racine de ϕ_Δ et $H(x,y) = F(x^q, x^m(y+\xi))/x^l$. Les hypothèses du lemme 7 sont alors vérifiées par H, puisque $\xi \in \mathfrak{O}_{\mathfrak{P}}$ et $\mathrm{tc}(R_H) = \mathrm{tc}(R_F)$. Notons $\mathcal{T}_0(H)$ le sous-arbre de $\mathcal{T}(F)$ correspondant à l'appel récursif `CNPuiseux`(H).

Nous allons maintenant montrer que, pour tout polynôme H satisfaisant les hypothèses du lemme 7, $\mathcal{T}_0(H) = \mathcal{T}_0(\overline{H})$. Pour cela, nous allons procéder par récurrence sur le nombre c d'appels à la fonction `CNPuiseux` nécessaires pour calculer $\mathcal{T}_0(H)$.

Si $c = 1$, alors $\mathcal{I}(H) = 1$, et $\mathcal{T}_0(H)$ est réduit à l'unique sommet étiqueté avec $\mathcal{GN}(H)$, qui consiste en l'unique arête $[(0,1),(1,0)]$. Le lemme 8 nous donne ainsi $\mathcal{T}_0(H) = \mathcal{T}_0(\overline{H})$.

Supposons dorénavant que $c > 1$. Le lemme 8 nous permet de dire que la racine de l'arbre $\mathcal{T}_0(F)$, les sommets de profondeur 1, ainsi que toutes les arêtes allant de la racine aux sommets de profondeur 2 de $\mathcal{T}_0(H)$ et $\mathcal{T}_0(\overline{H})$ coïncident et sont étiquetés de la même manière. Si H_0 est le polynôme obtenu à partir de H dans `CNPuiseux`, alors le nombre d'appels de fonction

nécessaire pour calculer $\mathcal{T}_0(H_0)$ est strictement inférieur à c. De plus, le lemme 7 nous assure que les hypothèses de récurrence peuvent être appliquées à H_0. Ainsi, $\mathcal{T}_0(H_0) = \mathcal{T}_0(\overline{H_0})$, et par construction des arbres de polygones, $\mathcal{T}_0(H) = \mathcal{T}_0(\overline{H})$. □

2.2.4 Choix d'un bon nombre premier p

Cette partie est dédiée au choix d'un idéal premier $\mathfrak{p}$ tel que F ait une bonne $\mathfrak{p}$-réduction.

Nous supposons ici que $F \in K[x, y]$, où $K = \mathbb{Q}(\gamma)$ est un corps de nombres, et que $d_y > 1$. Nous noterons M_γ le polynôme minimal de γ sur $\mathbb{Q}$, et nous représenterons les éléments de K comme des polynômes en γ de degrés strictement inférieurs à $w = [K : \mathbb{Q}]$, et avec des coefficients dans $\mathbb{Q}$. Quitte à effectuer un changement de variable dans M_γ et les coefficients de F, nous supposons que $\gamma \in \mathfrak{o}$, c'est-à-dire $M_\gamma \in \mathbb{Z}[z]$. Nous rappelons que nous notons R_F le résultant de F et F_y en y.

Définition 24. *Soit P un polynôme multivarié de $K[\underline{x}]$. Il existe un unique couple $(H, c) \in \mathbb{Z}[z, \underline{x}] \times \mathbb{N}$ avec $\deg_z(H) < w$ et $P(\underline{x}) = H(\gamma, \underline{x})/c$, où c est minimal. Le polynôme H est appelé le* ***numérateur*** *de P et noté* $\mathrm{num}(P)$. *L'entier c est le* ***dénominateur*** *de H, noté* $\mathrm{denom}(P)$. *On définit ensuite $\|P\|_\infty = \|H\|_\infty/c$ et la* ***taille*** *de P comme étant* $\mathrm{ht}(P) = \max\{\log c, \log \|H\|_\infty\}$.

Si l'on pose $F_n = \mathrm{num}(F)$ et $b = \mathrm{denom}(F)$, on a $F(x, y) = F_n(\gamma, x, y)/b$, de sorte que $R_F(x) = R_{F_n(\gamma,x,y)}/b^{2d_y-1}$ avec $R_{F_n(\gamma,x,y)} \in \mathbb{Z}[\gamma, x]$.

Bonne réduction locale

On souhaite trouver un nombre premier p et un idéal premier $\mathfrak{p}$ de $\mathfrak{o}$ divisant p tels que F ait une bonne $\mathfrak{p}$-réduction locale, c'est-à-dire tels que :

- (C_1) $p > d_y$.
- (C_2) p ne divise pas b.
- (C_3) On peut déterminer une représentation explicite d'un idéal premier $\mathfrak{p}$ de $\mathfrak{o}$ divisant p, tel que l'isomorphisme $\mathfrak{o} \to \mathfrak{o}/\mathfrak{p} \cong \mathbb{F}_{p^t}$ puisse être effectivement calculé.
- (C_4) $\mathrm{tc}(R_F) \not\equiv 0$ modulo $\mathfrak{p}$, condition équivalente à $\mathrm{tc}(R_{F_n(\gamma,x,y)}) \not\equiv 0$ modulo $\mathfrak{p}$ si (C_2) est satisfaite.

Les conditions (C_1) et (C_2) sont facilement vérifiées. On peut traiter la condition (C_3) de manière standard : notons $\overline{M}$ un facteur irréductible quelconque de M_γ dans $\mathbb{F}_p[z]$, et M un relèvement de $\overline{M}$ dans $\mathbb{Z}[z]$. Alors il est bien connu que si p est un nombre premier ne divisant pas l'indice $e_\gamma = [\mathfrak{o} : \mathbb{Z}[\gamma]]$, alors l'idéal $\mathfrak{p} = (p, M(\gamma))$ de $\mathfrak{o}$ est premier [Coh93]. Ainsi, les éléments de $\mathfrak{o}$ peuvent être réduits à l'aide du morphisme

$$\mathfrak{o} \to \mathfrak{o}/\mathfrak{p} \cong \mathbb{F}_p[z]/(\overline{M}) \cong \mathbb{F}_{p^t}$$

où $t = \deg \overline{M}$. Néanmoins, le calcul de e_γ n'est pas trivial, et il en est de même pour le calcul des générateurs des idéaux premiers divisant p quand p divise e_γ. Mais si e_γ n'est pas connu, il suffit de choisir un premier p qui ne divise pas Δ_{M_γ}, puisque e_γ divise Δ_{M_γ} [Coh93].

Remarque 16. *En pratique, afin de travailler dans l'extension de $\mathbb{F}_p$ la plus petite possible, nous choisirons $\overline{M}$ parmi les facteurs de $\overline{M_\gamma}$ de plus petit degré. De plus, il peut être intéressant d'essayer plusieurs nombres premiers p afin de réduire la taille de l'extension t, le cas $t = 1$ étant bien entendu le cas le plus favorable.*

Enfin, en ce qui concerne la condition (C_4), nous allons étudier différentes méthodes, déterministes et probabilistes. Pour faciliter l'étude, nous remplaçons la condition (C_4) par la condition plus forte suivante :

(C_4') $\mathrm{Norm}_{K/\mathbb{Q}}(\mathrm{tc}(R_{F_n(\gamma,x,y)})) \not\equiv 0$ modulo p.

Ainsi, si les conditions (C_1) à (C_4') sont vérifiées, alors pour tout idéal premier $\mathfrak{p}$ divisant p, F a une bonne réduction locale en $\mathfrak{p}$. En pratique, néanmoins, il n'est pas recommandé d'utiliser (C_4'). Enfin, nous introduisons la notation suivante :

$$N_F = b|\mathrm{Norm}_{K/\mathbb{Q}}(\mathrm{tc}(R_{F_n(\gamma,x,y)}))\Delta_{M_\gamma}|.$$

Les conditions (C_1) à (C_4') sont alors induites par :

(C_5) $p > d_y$ et $N_F \not\equiv 0$ modulo p.

Stratégie déterministe :

Nous déterminons une borne B telle que, pour tout premier $p > B$, la condition (C_5) soit satisfaite. Nous commençons par prouver les deux lemmes suivants, qui seront aussi utiles pour les stratégies probabilistes.

Lemme 10. *Le résultant $R_{F_n} \in \mathbb{Z}[z, x]$ de F_n et F_{ny} en y satisfait :*

$$\|R_{F_n}\|_\infty \leq (2d_y - 1)!\, {d_y}^{d_y} \left[(w+1)(d_x+1)\right]^{2d_y-2} \|F_n\|_\infty^{2d_y-1}.$$

Démonstration. Notons $a_i(z,x)$ le coefficient de y^i dans F_n. En développant le déterminant de la matrice de Sylvester de F_n et F_{ny}, on voit qu'il existe des indices $\{i_j\}_{1\leq j\leq 2d_y-1}$ entre 0 et d_y tels que :

$$\begin{aligned}\|R_{F_n}\|_\infty &\leq (2d_y-1)!\,\|\prod_{j=1}^{d_y-1} a_{i_j}(z,x)\prod_{j=d_y}^{2d_y-1} i_j a_{i_j}(z,x)\|_\infty \\ &\leq (2d_y-1)!\,{d_y}^{d_y}\,\|\prod_{j=1}^{2d_y-1} a_{i_j}(x,z)\|_\infty.\end{aligned}$$

La borne vient alors récursivement de :

$$\|a_i c\|_\infty \leq (w+1)(d_x+1)\|a_i\|_\infty\|c\|_\infty$$

pour tout $c(x,z)\in\mathbb{Z}[z,x]$ et de l'inégalité $\|a_i\|_\infty\leq\|F_n\|_\infty$. □

Lemme 11. *Soit $c\in\mathbb{Z}[\gamma]$ un coefficient de $R_{F_n(\gamma,x,y)}$. Notons :*

$$B_0 = \|R_{F_n}\|_\infty(\|M_\gamma\|_\infty+1)^{(w-1)(2d_y-2)} \quad (2.15)$$
$$B_1 = (w+1)^{(2w-1)/2}\|M_\gamma\|_\infty^{w-1}B_0^w \quad (2.16)$$
$$B_2 = w^w(w+1)^{(2w-1)/2}\|M_\gamma\|_\infty^{2w-1}. \quad (2.17)$$

Alors on a $\|c\|_\infty\leq B_0$, $|\operatorname{Norm}_{K/\mathbb{Q}}(c)|\leq B_1$ et $|\Delta_{M_\gamma}|\leq B_2$. En particulier :

$$\|R_{F_n(\gamma,x,y)}\|_\infty\leq B_0.$$

Démonstration. Par construction, le coefficient principal de F_n ne s'annule pas par évaluation en γ. De ce fait, l'évaluation et le résultant commutent, et on a $R_{F_n(\gamma,x,y)}=R_{F_n}(\gamma,x)$. Notons $C(z)\in\mathbb{Z}[z]$ le coefficient de x^i dans $R_{F_n}(z,x)$ et $c(z)\in\mathbb{Z}[z]$ le numérateur du coefficient de x^i dans $R_{F_n}(\gamma,x)$. Il est clair que $c(\gamma)=C(\gamma)$. Comme M_γ est unitaire, la division euclidienne nous donne $Q\in\mathbb{Z}[z]$ tel que $C=QM_\gamma+c$. Or, comme $\deg_z C\leq(2d_y-1)(w-1)$, en déroulant le processus de la division euclidienne, on peut montrer :

$$\|c\|_\infty\leq\|C\|_\infty(\|M_\gamma\|_\infty+1)^{(w-1)(2d_y-2)}.$$

Cette dernière inégalité nous donne le premier résultat.

Comme $\operatorname{Norm}_{K/\mathbb{Q}}(c(\gamma))=\texttt{Resultant}_z(M_\gamma(z),c(z))$, l'inégalité d'Hadamard (en utilisant des comparaisons triviales des normes) implique la deuxième inégalité. En utilisant les mêmes arguments, on peut montrer la troisième inégalité. □

Enfin, on obtient le résultat suivant, pour lequel on ne prétend aucune optimalité :

Proposition 18. *Posons $B = \max\{b, B_1, B_2\}$ (voir (2.16) et (2.17)). Alors pour tout $p > B$, la condition (C_5) est vérifiée. De plus, B peut être calculé de manière effective, et il existe un premier $p > B$ de taille :*

$$\mathrm{ht}(p) \in O(wd_y\,[w\,\mathrm{ht}(M_\gamma) + \mathrm{ht}(F) + \log\,(wd_xd_y)]).$$

Démonstration. Pour $d_y > 1$, on a $B_1 > d_y$. Si p est un nombre premier plus grand que B, alors la condition (C_5) est trivialement vérifiée. On applique ensuite le lemme 11 avec $c = \mathrm{tc}(R_{F_n(\gamma,x,y)})$. En prenant les logarithmes, et en utilisant la formule de Stirling dans la définition de B_1 et B_2, on voit alors que B possède la taille annoncée. Et comme il y a toujours un nombre premier entre B et $2B$, la proposition s'ensuit. □

Stratégie probabiliste :

Nous présentons ici deux algorithmes probabilistes, un de type « Monte-Carlo » et l'autre de type « Las Vegas », pour trouver un nombre premier p tel que la condition (C_5) soit vérifiée. Nous commençons par décrire une fonction intermédiaire qui sera utilisée par les deux méthodes. Cette fonction nécessite l'utilisation de deux sous-fonctions :

- `RandomPrime`(A, C), qui retourne un nombre premier p aléatoire dans l'intervalle $[A, C]$. Nous supposerons que les nombres premiers ainsi retournés sont uniformément distribués, et renvoyons le lecteur à [Sho05, section 7.5] pour la construction d'un tel algorithme, implanté dans [Sho].
- `NextPrime`(n), qui retourne le plus petit nombre premier strictement plus grand que n.

`Tirer-p`(B, d, ϵ)
`Entrée :`
B : *Un nombre réel positif.*
d : *Un entier* > 1.
ϵ : *Un nombre réel vérifiant* $0 < \epsilon \leq 1$.
`Sortie :`
Un nombre premier p vérifiant :
- *$p > d$,*
- *Pour tout entier $d \leq N \leq B$, p divise N avec une probabilité inférieure à ϵ.*

`Début`
`Si` $B < 3$ `alors Retourner NextPrime`(d) `Fin`

$K \leftarrow 2\ln B/(\epsilon \ln\ln B) + 2d/\ln d$

$C \leftarrow \max\{2d, K(\ln K)^2\}$

`Retourner RandomPrime`$(d+1, C)$

`Fin.`

Proposition 19. *L'algorithme* `Tirer-p` *retourne la sortie escomptée. De plus, la taille du premier p retourné par* `Tirer-p`*(B, d, ϵ) vérifie :*

$$\mathrm{ht}(p) \in O(\log\log B + \log d + \log \epsilon^{-1}).$$

Démonstration. Nous commençons par remarquer que la condition $p > d$ est automatiquement vérifiée. De plus, si $B < 3$, l'algorithme retourne un résultat correct de taille $O(\log d)$ et avec une probabilité 1, puisque comme $d > 1$, il existe toujours un nombre premier compris entre d et $2d$. On supposera donc dans la suite que $B \geq 3$. Alors pour tout entier positif n, nous noterons classiquement $\omega(n)$ le nombre de premiers qui divisent n. Étant donné un réel positif x, nous noterons $\pi(x)$ le nombre de premiers inférieurs où égaux à x. Les estimations de [BS96, section 8.8] nous amènent à :

$$\frac{x}{\ln x} < \pi(x)\ (x \geq 17), \quad \pi(x) < \frac{2x}{\ln x}\ (x > 1), \quad \omega(n) < \frac{2\ln n}{\ln\ln n}\ (n \geq 3).$$

Posons $h(x) = \frac{2\ln x}{\ln\ln x}$ et montrons tout d'abord que pour tout entier N avec $d \leq N \leq B$, on a $\omega(N) \leq h(B)$. La fonction $h(x)$ admet un minimum sur $[3, +\infty]$ égal à $2e$ et atteint en $x = e^e < 16$. Ainsi, si $N < e^e$, $\omega(N) \leq 2 \leq 2e \leq h(B)$. Pour $x > e^e$, la fonction h est croissante, de sorte que l'on a aussi $\omega(N) \leq h(B)$ si $N > e^e$.

Soit maintenant C un nombre supérieur à $2d$ et τ la probabilité qu'un premier donné par `RandomPrime`$(d+1, C)$ divise N. Il nous suffit de déterminer un entier C suffisamment grand de telle sorte que $\tau \leq \epsilon$. Comme il existe toujours un nombre premier entre d et $2d$ si $d > 1$, on a $\pi(C) - \pi(d) \geq 1$. De plus, étant donné que la fonction `RandomPrime` a un comportement uniforme, on cherche un nombre C tel que, pour tout entier N compris entre d et B, on ait :

$$\tau = \frac{\omega(N)}{\pi(C) - \pi(d)} \leq \epsilon. \tag{2.18}$$

Or, $B \geq 3$ et $d > 1$, donc les estimations faites ci-dessus montrent qu'il suffit de trouver un nombre C tel que :

$$\pi(C) \geq K = \frac{2\ln B}{\epsilon \ln\ln B} + \frac{2d}{\ln d}.$$

En posant $C = K(\ln K)^2$, on trouve $C/\ln C = K(\ln K)^2/(\ln K + 2\ln\ln K)$. Pour $B \geq 3$ et $d \geq 2$, K est plus grand que $4e$, et $(\ln K)^2/(\ln K + 2\ln\ln K) \geq 1$ (fonction croissante sur $[4e, +\infty]$). De plus, $C \geq 17$, de sorte que :

$$\pi(C) \geq \frac{C}{\ln C} \geq K,$$

et la relation (2.18) est vérifiée. Enfin, l'algorithme retourne un premier p vérifiant $\mathrm{ht}(p) \leq \max\{\log C, \log 2d\}$. Comme $\log C = \log K + 2\log\log K \in O(\log\log B + \log d + \log \epsilon^{-1})$, le résultat s'ensuit. □

Nous pouvons maintenant définir nos algorithmes probabilistes. Nous commençons par l'algorithme de type Monte-Carlo.

`MCGoodPrime(`F, M_γ, ϵ`)`
`Entrée :`
F : *Un polynôme sans facteur carré dans* $K[x, y]$ *de degré* $d_y > 1$,
M_γ : *Un polynôme irréductible unitaire dans* $\mathbb{Z}[z]$,
ϵ : *Un nombre réel vérifiant* $0 < \epsilon \leq 1$.
`Sortie :`
Un nombre premier p *vérifiant* (C_5) *avec une probabilité au moins égale à* $1 - \epsilon$.
`Début`
$(d_x, d_y, w) \leftarrow (\deg_x(F), \deg_y(F), \deg_z(M_\gamma))$
$F_n \leftarrow \mathrm{num}(F)$
$R \leftarrow (2d_y - 1)!\, {d_y}^{d_y} \left[(w+1)(d_x+1)\right]^{2d_y-1} \|F_n\|_\infty^{2d_y-1}$
$B_0 \leftarrow R\,(\|M_\gamma\|_\infty + 1)^{(w-1)(2d_y-2)}$
$B_1 \leftarrow (w+1)^{(2w-1)/2} \|M_\gamma\|_\infty^{w-1} B_0^w$
$B_2 \leftarrow w^w (w+1)^{(2w-1)/2} \|M_\gamma\|_\infty^{2w-1}$
$B' \leftarrow \max\{\mathrm{denom}(F), B_1, B_2\}$
`Retourner Tirer-p(`$B', d_y, \epsilon/3$`)`
`Fin.`

Proposition 20. *L'algorithme* `MCGoodPrime` *retourne le résultat escompté. De plus, la taille du nombre premier* p *retourné vérifie :*

$$\mathrm{ht}(p) \in O(\log(d_y w \log d_x) + \log \mathrm{ht}(F) + \log \mathrm{ht}(M_\gamma) + \log \epsilon^{-1}).$$

Démonstration. D'après la proposition 19, l'entier p divise $\mathrm{denom}(F)$ avec une probabilité inférieure à $\epsilon/3$. Il en est de même pour les deux autres facteurs de la condition (C_5). Donc p divise le produit avec une probabilité

inférieure à ϵ. En ce qui concerne la taille du premier p, il suffit d'utiliser l'estimation de B' donnée par la proposition 18, et la proposition 19 conduit au résultat. □

Enfin, nous concluons avec une méthode de type Las Vegas :

`LVGoodPrime(`F`,`M_γ`)`
`Entrée :`
F : *Un polynôme sans facteur carré dans* $K[x,y]$ *de degré* $d_y > 1$.
M_γ : *Un polynôme irréductible unitaire dans* $\mathbb{Z}[z]$.
`Sortie :`
Un nombre premier p *vérifiant* (C_5).
`Début`
$d_y \leftarrow \deg_y(F)$
$R \leftarrow \mathrm{num}(\mathrm{tc}(\mathtt{Resultant}_y(F, F_y)))$
$N_1 \leftarrow |\mathrm{Norm}_{K/\mathbb{Q}}(R(\gamma))|$
$N_2 \leftarrow |\mathtt{Disc}_z(M_\gamma)|$
$L \leftarrow \{\mathrm{denom}(F), N_1, N_2\}$
$B' \leftarrow \max L$
`Répéter`
$p \leftarrow$ `Tirer-p`$(B', d_y, 1/6)$
`tant que` p `divise un élément de` L `Fin`
`Retourner` p
`Fin.`

Proposition 21. `LVGoodPrime`*(*F*,*M_γ*) retourne un premier* p *satisfaisant :*

$$\mathrm{ht}(p) \in O(\log(d_y w \log d_x) + \log \mathrm{ht}(F) + \log \mathrm{ht}(M_\gamma)).$$

et le nombre moyen d'itérations est inférieur à 2.

Démonstration. Le raisonnement est analogue à celui de la proposition 20. On remarquera que N_1 est un diviseur de $\mathrm{Norm}_{K/\mathbb{Q}}(\mathrm{tc}(R_{F_n(\gamma,x,y)}))$. De plus, l'entier p obtenu à chaque tirage satisfait (C_5) avec une probabilité supérieure à $1/2$. □

Le calcul de $\mathrm{tc}(R_F)$ peut être coûteux. Néanmoins, dans le cadre du calcul du groupe de monodromie, nous aurons de toute manière besoin de calculer R_F. De plus, en pratique, on ne calculera pas la norme de $\mathrm{tc}(R_F)$, mais nous utiliserons plutôt des calculs modulo $\mathfrak{p} = (p, \overline{M})$.

Bonne réduction globale

Nous étendons maintenant les bornes obtenues pour trouver un nombre premier p et un idéal premier $\mathfrak{p}$ divisant p tel que F ait une bonne $\mathfrak{p}$-réduction globale.

Il s'agit donc de trouver un p et $\mathfrak{p}$ tels que les conditions (C_1), (C_2) et (C_3) (voir le début de cette section) soient vérifiées, et tels que :

(GC_4) La factorisation sans carré de $R_F(x)$ est préservée par réduction modulo l'idéal $\mathfrak{p}$ défini par la condition (C_3).

Notons $S(x)$ la partie sans carré *unitaire* de $R_{F_n(\gamma,x,y)}$ (polynôme unitaire sans carré de plus grand degré divisant $R_{F_n(\gamma,x,y)}$), $S_n = \mathrm{num}(S) \in \mathbb{Z}[z,x]$ et $S_d = \mathrm{denom}(S)$, de sorte que $\mathrm{lc}(S_n) = S_d$. On pose ensuite $R_{S_n} = \mathtt{Resultant}_x(S_n, S_{nx}) \in \mathbb{Z}[z]$. On a alors $R_{S_n}(\gamma) = R_{S_n(\gamma,x)}$ puisque le coefficient principal en x de S_n ne s'annule pas en $z = \gamma$.

Pour faciliter nos estimations, nous posons comme précédemment

$$N_S = b|\mathrm{Norm}_{K/\mathbb{Q}}(\mathrm{lc}(R_{F_n(\gamma,x,y)}))\mathrm{Norm}_{K/\mathbb{Q}}(R_{S_n(\gamma,x)})\Delta_{M_\gamma}|.$$

Lemme 12. *La condition suivante induit (C_1), (C_2), (C_3) et (GC_4) :*

(GC_5) *$p > d_y$ et $N_S \not\equiv 0$ modulo p.*

Démonstration. D'après un résultat de Weinberger et Rothschild [WR76] (voir aussi le théorème 3.1 de [Enc95]), S_d, et donc le coefficient principal de S_n, divise $\mathrm{Norm}_{K/\mathbb{Q}}(\mathrm{lc}(R_{F_n(\gamma,x,y)}))\Delta_{M_\gamma}$ dans $\mathbb{Z}$. Si (GC_5) est satisfaite, le coefficient principal de S_n ne s'annule alors pas modulo $\mathfrak{p}$. On en déduit que $\overline{R_{S_n}} = 0$ si et seulement si $R_{\overline{S_n}} = 0$, la notation $\overline{S_n}$ désignant la réduction modulo $\mathfrak{p}$ de S_n. Ainsi, (GC_5) implique que $\overline{S_n}$ n'a pas de racine multiple et que son degré est égal à celui de S_n. De même, le degré et le nombre de racines disctintes de $R_{F_n(\gamma,x,y)}$ sont préservés par réduction modulo $\mathfrak{p}$; sa décomposition sans carré l'est donc aussi, ainsi que celle de R_F. □

Stratégie déterministe :

Nous cherchons une borne B_G telle que pour tout nombre premier $p > B_G$, la condition (GC_5) soit vérifiée. Les bornes $\mathrm{Norm}_{K/\mathbb{Q}}(\mathrm{lc}(R_{F_n(\gamma,x,y)})) \leq B_1$ et $|\Delta_{M_\gamma}| \leq B_2$ sont déjà données par le lemme 11. Il reste donc essentiellement à déterminer une borne pour $|\mathrm{Norm}_{K/\mathbb{Q}}(R_{S_n(\gamma,x)})|$. Soit $\delta = \deg_x(R_F)$. On utilisera aussi δ comme borne sur le degré de S.

Lemme 13.

$$\|S_n\|_\infty \leq 2^{w+\delta}(\delta+1)^{\frac{1}{2}}(w+1)^{\frac{7w}{2}}\|R_{F_n(\gamma,x,y)}\|_\infty^\delta \|M_\gamma\|_\infty^{4\delta}.$$

Démonstration. On a $\|S_n\|_\infty \leq |\mathrm{Norm}_{K/\mathbb{Q}}(\mathrm{lc}(R_{F_n(\gamma,x,y)}))\Delta_{M_\gamma}|\|S\|_\infty$ d'après [WR76]. La dernière majoration provient de [Enc95], lemme 4.1. □

Lemme 14. *Le résultant* $R_{S_n} = \mathtt{Resultant}_x(S_n, S_{nx}) \in \mathbb{Z}[z]$ *vérifie :*

$$\|R_{S_n}\|_\infty \leq (2\delta-1)!\,\delta^\delta (w+1)^{2\delta-2}\|S_n\|_\infty^{2\delta-1}.$$

Démonstration. Il suffit d'appliquer le lemme 10 au polynôme S_n en remplaçant d_y par δ et d_x par 0. □

On obtient les bornes suivantes :

Lemme 15. *Notons :*

$$B_3 = \|R_{S_n}\|_\infty(\|M_\gamma\|_\infty+1)^{(w-1)(2\delta-2)} \quad (2.19)$$
$$B_4 = (w+1)^{(2w-1)/2}\|M_\gamma\|_\infty^{w-1}B_3^w \quad (2.20)$$

Alors $\|R_{S_n(\gamma,x)}\|_\infty \leq B_3$ *et* $|\mathrm{Norm}_{K/\mathbb{Q}}(R_{S_n(\gamma,x)})| \leq B_4$.

Démonstration. Il suffit d'appliquer les mêmes arguments que dans la preuve du lemme 11. □

Ces bornes nous amènent au résultat suivant :

Proposition 22. *Si l'on pose* $B_G = \max\{b, B_4\}$ *(voir (2.17) et (2.20)), alors pour tout* $p > B_G$*, la condition* (GC_5) *est vérifiée. De plus,* B_G *peut être calculé de manière effective, et il existe un premier* $p > B_G$ *de taille*

$$\mathrm{ht}(p) \in O(w^2 d_x^2 d_y^3\,[\mathrm{ht}(M_\gamma)+\mathrm{ht}(F)+\log(wd_xd_y)]).$$

Démonstration. Tout d'abord, on a $B_1 > d_y$ si $d_y > 1$. Ensuite, remarquons que $B_0 \leq B_3$, ce qui donne $B_1 \leq B_4$. La borne B_1 n'est donc pas à prendre en compte. De même, on vérifie facilement que $B_2 \leq B_4$. Si p est un nombre premier plus grand que B_G, alors la condition (GC_5) est trivialement vérifiée. En prenant le logarithme de B_4, puis en reportant dans les bornes des lemmes précédents, on obtient :

$$\mathrm{ht}(p) \in O(w^2\delta\log(w\delta) + w\delta^2 d_y\,[w\mathrm{ht}(M_\gamma)+\mathrm{ht}(F)+\log(wd_xd_y)]).$$

Ensuite, on factorise w dans le second terme de la somme, puis on majore δ par $d_x(2d_y-1)$. Pour finir, comme il y a toujours un nombre premier entre B et $2B$, la proposition s'ensuit. □

Stratégie probabiliste :

Comme dans le cas de la réduction locale, la borne B_G donnée par la proposition 22 nous permet d'obtenir deux algorithmes probabilistes pour trouver un nombre premier p vérifiant (GC_5).

`GMCGoodPrime`(F, M_γ, ϵ)
`Entrées :`
F : *Un polynôme sans facteur carré dans* $K[x,y]$ *de degré* $d_y > 1$.
M_γ : *Un polynôme irréductible unitaire dans* $\mathbb{Z}[z]$.
ϵ : *Un nombre réel vérifiant* $0 < \epsilon \leq 1$.
`Sortie :`
Un nombre premier p *vérifiant* (GC_5) *avec une probabilité au moins égale à* $1 - \epsilon$.
`Début`
$(d_x, d_y, w) \leftarrow (\deg_x(F), \deg_y(F), \deg_z(M_\gamma))$
$\delta \leftarrow d_x(2d_y - 1)$
$F_n \leftarrow \mathrm{num}(F)$
$R_1 \leftarrow (2d_y - 1)!\, {d_y}^{d_y} \left[(w+1)(d_x+1)\right]^{2d_y-1} \|F_n\|_\infty^{2d_y-1}$
$B_0 \leftarrow R_1 \left(\|M_\gamma\|_\infty + 1\right)^{(w-1)(2d_y-2)}$
$R_2 \leftarrow 2^{w+\delta}(\delta+1)^{\frac{1}{2}}(w+1)^{\frac{7w}{2}} B_0^\delta \|M_\gamma\|_\infty^{4\delta}$
$R_3 \leftarrow (2\delta-1)!\, \delta^\delta (w+1)^{2\delta-1} R_2^{2\delta-1}$
$B_3 \leftarrow R_3(\|M_\gamma\|_\infty + 1)^{(w-1)(2\delta-2)}$
$B_4 \leftarrow (w+1)^{(2w-1)/2} \|M_\gamma\|_\infty^{w-1} B_3^w$
$B' \leftarrow \max\{\mathrm{denom}(F), B_4\}$
Retourner `Tirer-p`$(B', d_y, \epsilon/4)$
`Fin`.

Proposition 23. *L'algorithme* `GMCGoodPrime` *retourne le résultat escompté. De plus, la taille du nombre premier* p *retourné vérifie :*

$$\mathrm{ht}(p) \in O(\log(wd_xd_y) + \log \mathrm{ht}(F) + \log \mathrm{ht}(M_\gamma) + \log \epsilon^{-1}).$$

Démonstration. Il suffit d'appliquer le raisonnement de la preuve de la proposition 20 en remplaçant les bornes de la proposition 18 par celles de la proposition 22 et en remarquant que N_S est formé de 4 facteurs. □

Enfin, pour la stratégie de type Las Vegas, nous utilisons l'algorithme `SQRfree`, qui à un polynôme de $K[z]$ associe sa partie sans facteur carré unitaire.

GLVGoodPrime(F,M_γ)
Entrées :
F : *Un polynôme sans facteur carré dans* $K[x,y]$ *de degré* $d_y > 1$.
M_γ : *Un polynôme irréductible unitaire dans* $\mathbb{Z}[z]$.
Sortie :
Un nombre premier p *vérifiant* (GC_5)
Début
$d_y \leftarrow \deg_y(F)$
$F_n \leftarrow \mathrm{num}(F)$
$R_{F_n} \leftarrow \mathrm{num}(\mathtt{Resultant}_y(F_n, F_{ny}))$
$S \leftarrow \mathrm{num}(\mathtt{SQRfree}(R_{F_n}, x))$
$R_S \leftarrow \mathtt{Resultant}_x(S, S_x)$
$N_2 \leftarrow |\mathtt{Disc}_z(M_\gamma)|$
$N_3 \leftarrow |\mathrm{Norm}_{K/\mathbb{Q}}(R_S)|$
$L \leftarrow \{\mathrm{denom}(F), N_2, N_3\}$
$B' \leftarrow \max L$
Répéter
$p \leftarrow \mathtt{Tirer\text{-}p}(B', d_y, 1/6)$
tant que p divise un élément de L Fin
Retourner p
Fin.

Proposition 24. GLVGoodPrime*(F,M_γ) retourne un premier p satisfaisant :*

$$\mathrm{ht}(p) \in O(\log(wd_xd_y) + \log \mathrm{ht}(F) + \log \mathrm{ht}(M_\gamma)).$$

et le nombre moyen d'itérations est inférieur à 2.

Démonstration. Le raisonnement est analogue à celui de la preuve de la proposition 23. □

2.3 Complexité

Dans cette partie, nous étudions la complexité de la partie modulaire de notre algorithme. Nous supposerons ici que le polynôme étudié F est unitaire en y. En effet, dans le cas non-unitaire, nous pensons que les résultats sont identiques, mais se ramener au cas unitaire par un changement élémentaire

ne suffit pas à le prouver ; il faudra recourir à des arguments plus fins qui restent à trouver au moment de la rédaction de ce mémoire. Nous rappelons que dans ce cas, on a l'égalité $R_F = \Delta_F$.

2.3.1 Calcul des développements de Puiseux rationnels dans un corps fini

Ici, L désigne un corps fini et $F \in L[x, y]$. Nous gardons les notations et hypothèses habituelles sur les différents degrés de F. Nous noterons $p > d_y$ la caractéristique de L, et l'on définit $t_0 = [L : \mathbb{F}_p]$. De plus, L_t représentera une extension de degré t sur L. Enfin, dans cette section, en ce qui concerne la multiplication de polynômes, nous ne considérons que la multiplication naïve et celle basée sur la transformée de Fourier (voir l'introduction de ce chapitre pour plus de détails à ce sujet). Nous parlerons de **multiplication standard** dans le premier cas, et de **multiplication rapide** dans le second.

Le but de cette section est de compter le nombre d'opérations de corps dans L induit par l'algorithme `RNPuiseux`. Plus précisément, nous allons prouver les théorèmes 13 et 14.

Théorème 13. *Il existe un algorithme qui calcule les parties singulières d'un système de développements de Puiseux rationnels au-dessus de* 0 *en* $\tilde{O}(d_y^3 d_x^2 + d_y^2 d_x t_0 \log p)$ *opérations dans le corps* L *si l'on utilise une multiplication rapide, et en* $\tilde{O}(d_y^5 d_x^2 + d_y^3 d_x t_0 \log p)$ *opérations si l'on utilise une multiplication standard.*

Ce résultat améliore les bornes de [Duv89] ou [HM87], qui sont en $O(d_y^6 d_x^2)$ opérations de corps. De plus, notre estimation inclut les coûts engendrés par les factorisations, là où D. Duval utilise le système D5 [DDD85] pour éviter ces factorisations. Ce gain vient de plusieurs points :

- Nous montrons que l'on peut effectuer les calculs modulo une puissance de x bien choisie (voir la proposition 25),
- Nous réduisons les transformations à des décalages de polynômes univariés, pour lesquels on peut employer des méthodes rapides (voir la proposition 26),
- Nous obtenons une majoration de la taille des sorties en fonction de la valuation x-adique du discriminant (voir le corollaire 28).

Pour notre algorithme de calcul du groupe de monodromie (voir le chapitre 3), nous aurons besoin de calculer les développements au-dessus de toutes les classes de conjugaisons sur L des points critiques. Plus précisément, si $\Delta_F =$

$\prod_i \Delta_i^{k_i}$ est une factorisation de Δ_F en produits de polynômes irréductibles de $L[x]$, alors les développements de Puiseux rationnels au-dessus des racines de Δ_i sont conjugués sur L. De ce fait, il suffit de calculer un système de développements de Puiseux rationnels au-dessus d'une seule racine c_i de Δ_i pour chaque polynôme Δ_i. On obtient alors le résultat suivant :

Théorème 14. *Il existe un algorithme qui calcule les parties singulières des développements de Puiseux rationnels au-dessus de l'ensemble des classes de conjugaison sur L des points critiques de F en $\tilde{O}(d_y^3 d_x^2 t_0 \log p)$ opérations dans le corps L si l'on utilise une multiplication rapide.*

On peut remarquer que ce résultat est quasiment le même que celui du calcul d'un système de développements de Puiseux rationnels. Cela est dû à l'étude de complexité effectuée en fonction de la taille de la sortie (voir le théorème 15).

Nous commençons en introduisant les notations de cette partie :

- Un système de développements de Puiseux rationnels au-dessus de 0 sera noté $\{R_i\}_{1 \leq i \leq \rho}$ avec

$$R_i(T) = (\tilde{x}_i(T), \tilde{y}_i(T)) = (\lambda_i T^{e_i}, \sum_{k=n_i}^{\infty} \beta_{ik} T^k) \text{ , où } \beta_{in_i} \neq 0$$

- (G_i, P_i, Q_i) est la sortie de `RNPuiseux` correspondant à R_i,
- (r_i, e_i, f_i), $1 \leq i \leq \rho$ sont respectivement l'indice de regularité, l'indice de ramification et le degré du corps des coefficients de R_i sur L.
- Pour chaque développement de Puiseux rationnel R_i, on peut déduire $e_i f_i$ développements de Puiseux classiques notés $S_{ijk}(x), 1 \leq k \leq e_i, 1 \leq j \leq f_i$.
- La quantité suivante servira lors de nos estimations :

$$\delta_F = \sum_{i=1}^{\rho} f_i r_i.$$

On rappelle également qu'une opération dans L_t peut se faire en $O(M(t) \log t)$ opérations dans le corps L. Pour plus de détails sur la complexité, et notamment en ce qui concerne les opérations sur les corps finis, nous renvoyons le lecteur à [vzGG99].

Remarque 17. *Il est important de noter que la quantité δ_F est essentiellement le nombre d'éléments de L nécessaires pour représenter les Y_i. En effet,*

chaque Y_i a au plus $r_i + 1$ coefficients non-nuls, et chacun de ces coefficients peut être représenté par au plus f_i éléments de L. Si l'on suppose que les séries Y_i sont représentés de manière dense (par exemple à l'aide d'un vecteur de $r_i + 1$ éléments de L_{f_i}, et les coefficients de Y_i sont eux-même représentés à l'aide de vecteurs de f_i éléments de L), alors la taille de la sortie est égale à $\delta_F + \sum_i f_i$, qui est comprise entre δ_F et $\delta_F + d_y$.

Nous allons diviser les preuves en plusieurs résultats.

Troncation des puissances de x

Proposition 25. *Les systèmes de développements de Puiseux rationnels de F et $\widetilde{F}^{\delta_F}$ au-dessus de 0 ont les mêmes parties singulières. De plus, les parties singulières des développements de Puiseux rationnels de F peuvent être calculés en appliquant l'algorithme* `RNPuiseux` *à $\widetilde{F}^{\delta_F}$ et en tronquant les polynômes considérés à chaque étape de l'algorithme modulo x^{δ_F+1}.*

Démonstration. Pour obtenir les parties singulières des développements de Puiseux rationnels de F, il nous suffit de connaître G_i modulo x pour chaque i. En posant $N_0 = 0$ dans la borne du lemme 16, et en utilisant le lemme 17, on voit qu'il suffit d'effectuer les calculs modulo x^{δ_F+1}. □

Pour simplifier les notations dans les lemmes suivants, nous enlevons les indices : R est un développement de Puiseux rationnel, (r, e, f) et (G, P, Q) sont les quantités associées. On définit $m_k i + q_k j = l_k$ $(1 \leq k \leq h)$ la suite des droites portant les arêtes de polygone de Newton rencontrées au cours de l'algorithme `RNPuiseux` pour calculer le triplet (G, P, Q), et $F = H_0, H_1, \cdots, H_h = G$ la suite de polynômes d'entrée de l'algorithme.

Lemme 16. *Soit N_0 un entier positif. Si l'on souhaite calculer $\widetilde{G}^{N_0}$, il suffit, à chaque étape de l'algorithme* `RNPuiseux`*, de calculer H_k modulo x^{N+1}, où $N = \frac{N_0}{e} + \sum_{k=1}^{h} \frac{l_k}{q_1 \ldots q_k}$.*

Démonstration. L'algorithme effectue les substitutions successives :

$$H_{k+1}(x, y) = \frac{H_k(\xi^{v_k} x^{q_k}, x^{m_k}(\xi^{u_k} + y))}{x^{l_k}}.$$

Si $H_k(x, y) = \sum_{ij} \alpha_{ij} x^j y^i$, on définit $H_{kw}(x, y) = \sum_{m_k i + q_k j = w} \alpha_{ij} x^j y^i$, de telle manière que $H_k = \sum_w H_{kw}$. Les monômes de H_{kw} sont transformés en les monômes de H_{k+1} situés sur la droite $j = w - l_k$ (voir la figure 2.5).

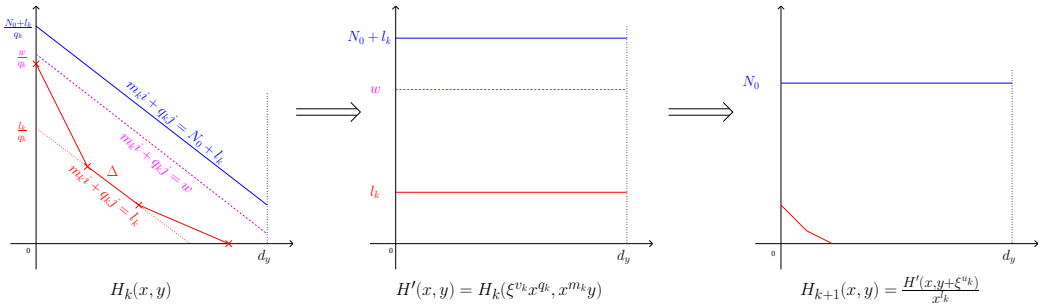

FIG. 2.5 – Intreprétation géometrique d'une substitution

Ainsi, pour déterminer $\widetilde{G}^{N_0} = \widetilde{H_h}^{N_0}$, il suffit de connaître $\widetilde{H_{h-1}}^{\frac{N_0+l_h}{q_h}}$. Par récurrence, on obtient la borne N annoncée, et on peut ainsi conclure qu'il suffit de calculer tous les H_k modulo x^{N+1}. □

Ce lemme peut aussi être utile si l'on souhaite calculer les développements au-delà de l'indice de régularité. Il est clair que l'on peut adopter une stratégie adaptative et tronquer encore plus les polynômes H_k en tenant compte des informations obtenues au fur et à mesure de l'algorithme. Il n'est néanmoins pas clair qu'une telle approche puisse améliorer la complexité asymptotique.

Lemme 17. *Avec les notations du lemme 16, on a :*

$$\sum_{k=1}^{h} \frac{l_k}{q_1 \cdots q_k} \leq \delta_F.$$

Démonstration. Notons Y_{ijk} la partie singulière de S_{ijk} et $\hat{F} = \prod_{ijk} y - Y_{ijk}$. Comme F et $\hat{F}$ ont des développements de Puiseux rationnels qui ont les mêmes parties singulières, en appliquant `RNPuiseux` à $\hat{F}$, on obtient la même suite $(m_k, q_k, l_k)_{1 \leq k \leq h}$ que pour le polynôme F, et une sortie $(\hat{G}, P, Q)$, où $P(x) = \lambda x^e$ et $Q(x,y) = Q_0(x) + x^r y$ sont les polynômes associés à F. Si l'on pose $c = l_h + l_{h-1} q_h + \cdots + l_1 q_2 \ldots q_h$, alors, de part la définition des substitutions, on a :

$$\hat{G}(x,y) = \frac{\hat{F}(P(x), Q(x,y))}{x^c}.$$

Comme $(1,0)$ appartient à $\mathcal{N}(\hat{G})$, on a $v_x(\hat{G}_y(x,0)) = 0$. Ainsi, en dérivant, on obtient $c = v_x(x^r \hat{F}_y(P(x), Q_0(x)))$, ce qui est équivalent à :

$$\frac{c}{e} = \frac{r + v_x(\hat{F}_y(P(x), Q_0(x)))}{e} = \frac{r}{e} + v_x(\hat{F}_y(x, Q_0((x/\lambda)^{1/e}))).$$

Supposons maintenant que la série de Puiseux tronquée $Q_0((x/\lambda)^{1/e})$ soit égale à $Y_{i'j'k'}$. On a alors :

$$\hat{F}_y(x, Q_0((x/\lambda)^{1/e})) = \hat{F}_y(x, Y_{i'j'k'}) = \prod_{\substack{(i,j,k) \\ (i,j,k)\neq(i',j',k')}} (Y_{i'j'k'} - Y_{ijk}).$$

Et comme pour tout triplet $(i,j,k) \neq (i',j',k')$, on a $v_x(Y_{i'j'k'} - Y_{ijk}) \leq \frac{r_i}{e_i}$, cela nous donne :

$$\frac{c}{e} = \sum_{k=1}^{h} \frac{l_k}{q_1 \dots q_k} \leq \frac{r_{i'}}{e_{i'}} + \sum_{\substack{(i,j,k) \\ (i,j,k)\neq(i',j',k')}} \frac{r_i}{e_i} = \sum_{k=1}^{\rho} \sum_{j=1}^{f_i} \sum_{k=1}^{e_i} \frac{r_i}{e_i} = \sum_{i=1}^{\rho} r_i f_i.$$

□

Coût des substitutions

Lemme 18. *Soit N un entier positif, $H \in L_t[x,y]$, $\xi \in L_t$, et posons $H'(x,y) = H(\xi^v x^q, x^m(\xi^u + y))/x^l$, où (m,q,l) correspond à une arête de $\mathcal{GN}(H)$. Alors on peut calculer les coefficients de $\widetilde{H'}^N$ en $O(NM(d_y))$ opérations dans le corps L_t.*

Démonstration. Étant donné que l'on souhaite calculer $\widetilde{H'}^N$, il nous suffit d'effectuer la transformation sur le polynôme :

$$\hat{H}(x,y) = \sum_{w=l}^{N+l} H_w(x,y), \text{ où } H_w(x,y) = \sum_{mi+qj=w} \alpha_{ij} x^j y^i.$$

On effectue alors le changement de variable pour chaque polynôme H_w :

$$\begin{aligned} H_w(\xi^v x^q, \xi^u x^m(1+y)) &= \sum_{mi+qj=w} \alpha_{ij} \left(\xi^v x^q\right)^j \left(x^m \left(\xi^u + y\right)\right)^i \\ &= x^w \sum_{mi+qj=w} \alpha_{ij} \xi^{vj} (\xi^u + y)^i \\ &= x^w R_w(y + \xi^u) \end{aligned}$$

où $R_w(z) = \sum_{mi+qj=w} \alpha_{ij} \xi^{vj} z^i$ est un polynôme univarié.

Pour calculer les puissances de ξ^v, on peut remarquer que l'exposant est borné par $(N+l)/q$. Or, l/q est borné par d_y, puisque les pentes du polygone générique sont au moins égales à -1. Ainsi, on peut calculer toutes les puissances nécessaires en $O(N+d_y)$ opérations dans L_t. Ensuite, on peut construire les polynômes R_w en au plus Nd_y multiplications dans L_t. Enfin, comme $p > d_y$, le décalage dans R_w peut être ramené à la multiplication de deux polynômes de degré d_y, et donc un coût en $O(M(d_y))$ [BP94]. Enfin, étant donné qu'il y a $N+1$ décalages à faire, le coût total est en $O(NM(d_y))$. □

Proposition 26. *L'ensemble des substitutions nécessaires pour calculer les parties singulières d'un système de développements de Puiseux rationnels de F au-dessus de 0 requiert $O\left(\delta_F(\delta_F+d_y)M(d_y)\tilde{M}(d_y)\log d_y\right)$ opérations dans le corps L.*

Démonstration. L'algorithme `RNPuiseux` effectue au plus r_i+1 substitutions dans le corps L_{f_i} à effectuer pour trouver le développement rationnels R_i. La proposition 25 nous assure que l'on peut effectuer toutes ces substitutions modulo x^{δ_F+1}. D'après le lemme 18, les substitutions nécessaires pour calculer R_i coûtent au plus $O(M(d_y)\delta_F(r_i+1))$ opérations dans L_{f_i}. En tenant compte de l'extension L_{f_i}/L, et en sommant sur l'ensemble des indices i, on obtient comme coût total :

$$\begin{aligned} & O\left(M(d_y)\delta_F\textstyle\sum_{i=1}^{\rho}(r_i+1)M(f_i)\log f_i\right) \\ = \; & O\left(M(d_y)\delta_F\textstyle\sum_{i=1}^{\rho}(r_i+1)f_i\tilde{M}(f_i)\log f_i\right) \\ = \; & O\left(\delta_F\left(\delta_F+d_y\right)M(d_y)\tilde{M}(d_y)\log d_y\right). \end{aligned}$$

En additionnant les deux bornes, on obtient le résultat. □

Coût des factorisations

Proposition 27. *L'ensemble des factorisations des polynômes caractéristiques nécessaires pendant l'algorithme* `RNPuiseux` *peuvent être calculées en*

$$O\left(\delta_F\log d_y\left[M(d_y^2)+M(d_y)\log d_y t_0\log p\right]\right)$$

opérations dans le corps L.

Démonstration. Nous commençons par rappeler qu'un polynôme de degré d à coefficients dans L_t peut se factoriser en $O(M(d^2)+M(d)t\,t_0\log p)$ opérations dans L_t [vzGG99, corollaire 14.30].

La première étape correspond au polygone exceptionnel. Comme le polynôme F est supposé unitaire, ce dernier est constituée d'une unique arête $[(0,0),(d_y,0)]$. Cette étape requiert donc la factorisation d'un polynôme de degré d_y et à coefficients dans L. Elle peut donc se faire en $O(M(d_y^2) + M(d_y)t_0 \log p)$ opérations dans L, ce qui est inclus dans notre borne.

Il reste ensuite au plus r_i factorisations à compter pour chaque développement de Puiseux rationnel R_i. Notons ϕ_Δ l'un de ces polynômes à factoriser, et L_t l'extension de L dans laquelle la factorisation de ϕ_Δ doit être déterminée. On peut facilement voir que le degré d_Δ de ϕ_Δ est au plus d_y/t : c'est trivialement vrai à la première étape de l'algorithme (puisqu'alors $t = 1$). Supposons maintenant que cette propriété soit vraie à une étape donnée de l'algorithme, et notons ξ une racine de ϕ_Δ, et k sa multiplicité. Alors si $\phi_{\Delta'}$ est un polynôme caractéristique de l'étape suivante induit par le choix de ξ, et $d_{\Delta'}$ son degré, la proposition 10 page 56 nous assure que :

$$d_{\Delta'} \leq k \text{ et } k[L_t(\xi) : L_t] \leq d_\Delta \leq d_y/t.$$

Ainsi, $d_{\Delta'} \leq d_y/[L_t(\xi) : L]$ et la propriété est vérifiée par récurrence.

De ce fait, la factorisation du polynôme caractéristique ϕ_Δ peut se faire en

$$O\left(M\left(\left(\frac{d_y}{t}\right)^2\right)\log\frac{d_y}{t} + t_0 t \log p M\left(\frac{d_y}{t}\right)\log\frac{d_y}{t}\right)$$

opérations dans le corps L_t. Comme chaque opération dans L_t peut être effectuée en $O(M(t)\log t)$ opérations dans L, et comme on peut vérifier que :

$$O\left(M\left(\left(\frac{d_y}{t}\right)^2\right)\log\frac{d_y}{t}M(t)\log t\right) \subset O(M(d_y^2)\log d_y)$$

et

$$O\left(M\left(\frac{d_y}{t}\right)\log\frac{d_y}{t}M(t)\log t\right) \subset O(M(d_y)\log^2 d_y)$$

(que l'on utilise la multiplication standard ou rapide), on obtient que le nombre d'opérations dans L nécessaire pour factoriser le polynôme caractéristique est en :

$$O\left(M(d_y^2)\log d_y + M(d_y)\log^2(d_y)t_0 t\log p\right).$$

Finalement, en multipliant par r_i (la première factorisation a déjà été prise en compte), en bornant t par f_i, et en additionnant sur i, on obtient le résultat. □

Borne pour δ_F

Proposition 28.

$$\delta_F = \sum_{i=1}^{\rho} r_i f_i \leq v_x(\Delta_F).$$

Démonstration. D'après la proposition 7 page 30, pour chaque série de Puiseux $S_{ijk}(x)$, il existe un autre série de Puiseux $S_{i_0j_0k_0}(x)$ avec $i_0 \in \{1 \dots \rho\}$, $j_0 \in \{1 \dots f_{i_0}\}$ et $k_0 \in \{1 \dots e_{i_0}\}$ tels que $\frac{r_i-1}{e_i} < v_x(S_{ijk}(x) - S_{i_0j_0k_0}(x)) \leq \frac{r_i}{e_i}$. De plus, si $v_x(S_{ijk}(x) - S_{i_0j_0k_0}(x)) \neq \frac{r_i}{e_i}$, e_i est un diviseur propre de e_{i_0} et $r_i \geq 1$. Donc, si l'on note $q = e_{i0}/e_i > 1$, il existe $m \in \mathbb{N}$ et $\alpha \neq 0 \in \overline{L}$ tel que $1 \leq m < q$ et :

$$S_{i_0j_0k_0}(x) = \widetilde{S_{ijk}}^{\frac{r_i-1}{e_i}}(x) + \alpha x^{\frac{r_i-1}{e_i}+\frac{m}{e_{i_0}}} + \cdots$$

Ainsi, pour $0 \leq l \leq q-1$, on a :

$$S_{i_0j_0k_0}^{[le_i,e_{i_0}]}(x) = \widetilde{S_{ijk}}^{\frac{r_i-1}{e_i}}(x) + \zeta_q^{ml}\alpha x^{\frac{r_i-1}{e_i}+\frac{m}{e_{i_0}}} + \cdots$$

On obtient donc :

$$\sum_{l=0}^{q-1} v_x(S_{ijk}(x) - S_{i_0j_0k_0}^{[le_i,e_{i_0}]}(x)) = q\frac{r_i}{e_i} + \frac{m-q}{e_i}.$$

Or, comme $q > 1$ et $r_i \geq 1$, on a $qr_i + m - q - r_i = (r_i - 1)(q-1) + m - 1 \geq 0$, ce qui implique :

$$\sum_{l=0}^{q-1} v_x(S_{ijk}(x) - S_{i_0j_0k_0}^{[le_i,e_{i_0}]}(x)) \geq \frac{r_i}{e_i}$$

Pour chaque série de Puiseux S_{ijk}, puisque $v_x(S_{ijk}(x) - S_{i'j'k'}(x)) \geq 0$ pour tout triplet $(i', j', k') \neq (i, j, k)$, on a :

$$v_x(F_y(x, S_{ijk}(x))) = \sum_{\substack{(i',j',k') \\ (i',j',k') \neq (i,j,k)}} v_x(S_{ijk}(x) - S_{i'j'k'}(x)) \geq \frac{r_i}{e_i}.$$

En sommant cette relation sur l'ensemble des (i, j, k), l'équation (2.9) nous donne :

$$v_x(\Delta_F) = \sum_{(i,j,k)} v_x(F_y(x, S_{ijk}(x))) \geq \sum_{i=1}^{\rho}\sum_{j=1}^{f_i}\sum_{k=1}^{e_i} \frac{r_i}{e_i} = \sum_{i=1}^{\rho} r_i f_i.$$

□

Preuves des théorèmes 13 et 14

Il est intéressant d'avoir une borne sur le nombre d'opérations dans L en fonction de la taille de la sortie, c'est-à-dire δ_F.

Théorème 15. *Le nombre d'opérations dans L nécessaire pour calculer la partie singulière d'un système de développements de Puiseux rationnels de F au-dessus de 0 est en :*

$$O\left(\delta_F M(d_y)\log d_y\left[\delta_F \tilde{M}(d_y) + M(d_y) + t_0 \log p \log d_y\right]\right)$$

$$\subset O\left(d_x\, M(d_y) d_y \log d_y\left[d_x M(d_y) + t_0 \log p \log d_y\right]\right).$$

Démonstration. La première affirmation est une conséquence des propositions 26 et 27, puisque $M(d_y^2) \in O(M(d_y)^2)$. L'inclusion est quant à elle une conséquence de la proposition 28, où $v_x(\Delta_F)$ est borné par $\deg_x(\Delta_F) \leq (2d_y - 1)d_x$. □

La preuve du théorème 13 est alors une conséquence triviale de ce résultat.

Démonstration. (du théorème 14)
Tout d'abord, étant donné que nous travaillons sur un corps fini, le calcul du discriminant Δ_F peut être effectué en

$$O(d_x M(d_x d_y)\log(d_x d_y))$$

opérations dans le corps L [vzGG99]. Cette étape est donc incluse dans notre borne de complexité.

De plus, Δ_F est un polynôme en x de degré $\delta \leq (2d_y - 1)d_x$. Par conséquent, ce polynôme peut être factorisé sur L en

$$O(M(\delta^2)\log\delta + t_0 \log p M(\delta)\log\delta) \subset \tilde{O}(d_x^2 d_y^2 + d_x d_y t_0 \log p)$$

opérations dans le corps L [vzGG99]. Cette étape est donc elle aussi incluse dans la borne de complexité.

Ensuite, si $\Delta_F = \prod_{i=1}^{m} R_i^{k_i}$ est la factorisation obtenue, on définit $t_i = \deg_x(R_i)$ et c_i une racine de R_i. Le calcul de $F_i = F(x + c_i, y)$ peut être effectué au coût de d_y décalages dans $L(c_i) = L_{t_i}$ des coefficients de F en y. Ceci peut donc être fait en $O(d_y M(d_x))$ opérations dans le corps L_{t_i} [BP94], et donc $O(d_y M(d_x) M(t_i)\log t_i) \subset \tilde{O}(d_y d_x t_i)$ opérations dans L. En sommant sur i et en bornant $\sum_i t_i$ par $(2d_y - 1)d_x$, on obtient à nouveau un résultat qui rentre dans notre estimation.

Enfin, on peut noter que la quantité δ_F associée au polynôme F_i est bornée par k_i (voir la proposition 28). D'après le théorème 15, l'appel de fonction `RNPuiseux`$(F(x+c_i,y), L(c_i))$ requiert

$$O^{\sim}\left(k_i M(d_y)\left[k_i \tilde{M}(d_y) + M(d_y) + t_0 t_i \log p\right]\right)$$

opérations dans le corps L_{t_i}, et donc

$$\begin{aligned} &O^{\sim}\left(k_i t_i M(d_y)\left[k_i \tilde{M}(d_y) + M(d_y) + t_0 t_i \log p\right]\right) \\ &\subset O^{\sim}\left(k_i t_i M(d_y)\left[d_x M(d_y) + d_x d_y t_0 \log p\right]\right) \end{aligned}$$

opérations dans le corps L en bornant le second k_i et le second t_i par $(2d_y - 1)d_x$. En sommant sur i et en bornant $\sum_i k_i t_i$ par $(2d_y - 1)d_x$, on obtient finalement une complexité de

$$O^{\sim}\left(d_x^2 d_y M(d_y)\left[M(d_y) + d_y t_0 \log p\right]\right)$$

opérations dans le corps L. Le théorème 14 se déduit alors de cette borne, puisqu'en utilisant une arithmétique rapide, on a $M(d_y) \in O^{\sim}(d_y)$. □

2.3.2 Complexité binaire du calcul de $\mathcal{T}(F)$

Soit $F \in K[x,y]$ un polynôme unitaire en y à coefficients dans un corps de nombres K représenté comme dans la section 2.2.4. On rappelle que $w = [K : \mathbb{Q}]$. Dans cette partie, nous étudions la complexité binaire du calcul de $\mathcal{T}(F)$. Nous n'estimons ici que les opérations binaires engendrées par les opérations arithmétiques dans les différents corps de coefficients. Si l'on suppose que l'implémentation est effectuée avec précaution (par exemple, l'accès aux coefficients des polynômes doit être effectué en temps constant), alors les résultats présentés ici donnent une borne supérieur réaliste du comportement d'un tel programme.

Les bornes que nous donnons pour les algorithmes probabilistes n'incluent pas le coût de génération des nombres premiers. De plus, pour simplifier les notations, nous nous plaçons dans le cadre de la multiplication rapide.

Nous supposons de plus que les éléments de $\mathbb{F}_p$ sont représentés par des entiers positifs. Pour simplifier les expressions, nous supposons également qu'une algorithmique rapide est utilisée pour l'arithmétique entière et polynomiale sur les corps finis.

Nous commençons par étudier la complexité des algorithmes probabilistes nous permettant de trouver un nombre premier p pour lequel F ait une bonne réduction.

Proposition 29. *L'algorithme* `MCGoodPrime`(F, M_γ, ϵ) *retourne le premier* p *en* $\tilde{O}(wd_y[\mathrm{ht}(F) + w\mathrm{ht}(M_\gamma)] + \log \epsilon^{-1})$ *opérations binaires.*

Démonstration. Si A est un entier, le calcul de A^N peut être effectué en $\tilde{O}(N\mathrm{ht}(A))$ opérations binaires en utilisant le scindage binaire (voir par exemple [BCS07, Mez07]). Comme celui de $d!$, à nouveau en utilisant le scindage binaire, peut être effectué en $\tilde{O}(d)$ opérations, il suffit de mettre bout à bout l'ensemble des opérations en tenant compte de la taille des nombres considérés pour obtenir le résultat. □

Théorème 16. *Étant donné un nombre réel* ϵ *vérifiant* $0 < \epsilon \leq 1$*, il existe un algorithme probabiliste de type Monte-Carlo qui calcule* $\mathcal{T}(F)$ *avec une probabilité d'erreur inférieure à* ϵ *et un nombre d'opérations binaires en :*

$$\tilde{O}(d_y^3 d_x^2 w^2 \log^2 \epsilon^{-1}[\mathrm{ht}(M_\gamma) + \mathrm{ht}(F)])$$

Démonstration. L'appel de fonction `MCGoodPrime`(F,M_γ,ϵ) nous donne un nombre premier p tel que F ait une bonne réduction locale avec une probabilité d'erreur inférieure à ϵ et une taille $\mathrm{ht}(p)$ donnée par la proposition 20. La complexité de cette étape, donnée par la proposition 29, est incluse dans notre borne.

Ensuite, on réduit les coefficients de M_γ et F_n modulo p, et on vérifie que p ne divise pas $b = \mathrm{denom}(F)$. On note $\overline{F}_n$ la réduction de F_n. Cette étape requiert $\tilde{O}(w\mathrm{ht}(M_\gamma) + wd_xd_y\mathrm{ht}(F_n) + \mathrm{ht}(b)\log p)$ opérations binaires.

On peut alors factoriser le polynôme $\overline{M_\gamma}$ sur $\mathbb{F}_p$ pour un coût de $\tilde{O}(w^2 + w\log p)$ opérations dans $\mathbb{F}_p$, et donc $\tilde{O}(w^2 \log p + w\log^2 p)$ opérations binaires. On choisit alors un facteur irréductible $\overline{M}$ de degré minimal. Les coefficients de $\overline{F}_n$ doivent être réduits modulo $\overline{M}$ et p. Il y a $(d_x+1)d_y$ coefficients et chaque division implique $\tilde{O}(w)$ opérations dans $\mathbb{F}_p$. Le pire des cas pour l'algorithme `RNPuiseux` apparaissant dans le cas où $\overline{M}$ est de degré w, on peut borner le coût de l'appel de fonction `RNPuiseux`$(\mathbb{F}_{p^w},\overline{F})$ par $\tilde{O}(d_y^3 d_x^2 + d_y^2 d_x w \log p)$ opérations dans $\mathbb{F}_{p^w}$ d'après le théorème 13. Le total est donc majoré par $\tilde{O}(d_y^2 d_x w[d_y d_x + w\log p + \mathrm{ht}(M_\gamma) + \mathrm{ht}(F_n)]\log p)$ opérations binaires. En prenant en compte la borne pour $\mathrm{ht}(p) = \log p$, et en appliquant quelques majorations grossières, on obtient le résultat. □

Proposition 30. *L'algorithme* `GMCGoodPrime`(F, M_γ, ϵ) *retourne le premier* p *en* $\tilde{O}(wd_x^2 d_y^3[\mathrm{ht}(F) + w\mathrm{ht}(M_\gamma)] + \log \epsilon^{-1})$ *opérations binaires.*

Démonstration. On applique le même raisonnement que dans la preuve de la proposition 29, en considérant les nombres engendrés par l'algorithme `GMCGoodPrime`. □

Théorème 17. *Étant donné un nombre réel ϵ vérifiant $0 < \epsilon \leq 1$, il existe un algorithme probabiliste de type Monte-Carlo qui calcule l'arbre de polygones de F au-dessus de chaque point critique fini de F avec une probabilité d'erreur inférieure à ϵ et un nombre d'opérations binaires en :*

$$O\tilde{}(d_y^3 d_x^2 w^2 \log^2 \epsilon^{-1}[\mathrm{ht}(M_\gamma) + \mathrm{ht}(F)])$$

Démonstration. Il suffit d'appliquer les mêmes arguments que ceux de la preuve du théorème 16 en utilisant les résultats des propositions 20 et 29, ainsi que le théorème 14. □

2.4 Calcul numérique des développements de Puiseux

À l'aide des calculs modulaires décrits dans la partie 2.2, nous obtenons l'arbre des polygones $\mathcal{T}(F)$. Cette section est dédiée au calcul numérique des séries de Puiseux à partir des données exactes contenues dans l'arbre $\mathcal{T}(F)$. Le travail présenté dans cette partie n'a pas prétention à être complet : nous ne prenons pas ici en compte la précision numérique nécessaire pour les calculs, ni la propagation des erreurs numériques. De ce fait, nous ne fournissons pas ici d'algorithmes certifiés. En pratique, nous nous contenterons d'ajouter un nombre de chiffres de précision égal au logarithme du nombre d'opérations nécessaires à l'algorithme pour garder les erreurs numériques sous contrôle. La certification du résultat reste donc un travail ouvert.

Une première esquisse de ce travail a été publié dans [Pot07, section 5]. Dans cet article, nous utilisons un algorithme basé sur l'algorithme classique `CNPuiseux` (voir la section 2.1.3), qui utilisait un filtre à deux étages : l'un sur les polygones de Newton [Pot07, section 5.3], et l'autre sur les partitions [Pot07, section 5.4]. Dans cette thèse, nous adoptons une stratégie similaire, en conservant notamment cette notion de filtre à deux étages. En ce qui concerne le tri utilisant les polygones de Newton, nous garderons la stratégie employée dans [Pot07]. Par contre, nous proposons une approche différente en ce qui concerne le tri utilisant les partitions. En effet, l'approche proposée dans [Pot07], basée sur un théorème de B. T. Smith [Smi70], ne tenait pas compte de la littérature plus récente. Ici, nous nous appuyerons sur les travaux effectués sur les pgcd approchés, et plus particulièrement ceux utilisant la décomposition en valeur singulière. Pour des raisons de rédaction, nous ferons référence aux travaux de Z. Zeng [Zen03, Zen04, Zen05]. Néanmoins, toute méthode de calcul de pgcd approché utilisant la décomposition

en valeur singulière peut être utilisée pour notre méthode. De plus, d'autres approches que celles utilisant la décomposition SVD peuvent être elles aussi utilisées, même si cela nécessite très certainement un peu de travail en ce qui concerne la partie permettant de trouver le degré du pgcd calculé. Nous tenons à remercier Mark van Hoeij pour avoir notifié l'idée d'utiliser la décomposition SVD au sein de cet algorithme, ainsi qu'Olivier Ruatta pour les explications fournies sur le calcul de pgcd approchés.

Enfin, pour simplifier les explications, nous supposons pour l'instant que nous calculons les développements de Puiseux au-dessus du point $x = 0$. Nous expliquerons comment traiter le cas général à la fin de cette section. De plus, nous utiliserons la notation suivante : un polynôme $\tilde{H}$ représentera une approximation du polynôme exact H.

Si l'on considère le point $x = 0$, on peut alors décrire la situation initiale de la manière suivante :

1. À la première étape de l'algorithme, nous avons une approximation numérique $\tilde{F}$ du polynôme F. La racine de l'arbre $\mathcal{T}(F)$ nous donne également son polygone exceptionnel $\mathcal{EN}(F)$. De ce fait, pour chaque arête Δ, nous obtenons une approximation $\tilde{\phi}_\Delta$ du polynôme caractéristique ϕ_Δ.
2. Ensuite, le sommet de profondeur 1 de $\mathcal{T}(F)$ associé à l'arête étiquetée par Δ nous donne la partition $[\phi_\Delta]$. Connaissant les multiplicités des racines du polynôme ϕ_Δ, il est possible de calculer une approximation des racines de ϕ_Δ. Ceci est par exemple traité dans la section 3 de [Zen03]. Nous décrirons ceci dans la section 2.4.2.
3. Finalement, il se peut que nous obtenions plusieurs approximations numériques correspondant à différentes racines de ϕ_Δ ayant la même multiplicité M. Chacune de ces racines définit un nouveau polynôme $\tilde{H}$. À cet ensemble de polynômes correspond l'ensemble des sous-arbres de $\mathcal{T}(F)$ dont les arêtes initiales sont étiquetées par (Δ, M). Néanmoins, comme il n'y a aucune correspondance canonique entre $\overline{K}$ et $\overline{\mathbb{F}_p}$, nous ne pouvons établir de bijection entre les sous-arbres de $\mathcal{T}(F)$ étiquetés par Δ et M et les polynômes $\tilde{H}$. De ce fait, nous traiterons les polynômes correspondant à la même arête Δ et à la même multiplicité M simultanément.

Ensuite, nous trierons les polynômes en fonction des sous-arbres dans la suite de l'algorithme. Plus précisément, afin de connecter les polynômes numériques obtenus aux feuilles de l'arbre $\mathcal{T}(F)$, nous utilisons un filtre à deux étages :

– Tout d'abord, nous séparons les polynômes numériques en fonction de

leur polygones de Newton. Cette partie sera effectué à l'aide de l'algorithme `Tri-Polygones` que nous décrirons dans la section 2.4.1.
- Puis nous séparons les polynômes approchés ayant le même polygone de Newton en fonction des multiplicités des racines de leurs polynômes caractéristiques. Cette partie, traitée par l'algorithme `Tri-Partitions`, est décrite dans la partie 2.4.2.

Il se peut que ce faisant, certains polynômes ne soient pas séparés lors de ce processus de filtrage. Néanmoins, nous avons alors obtenu l'ensemble des polygones de Newton et l'ensemble des partitions de leurs polynômes caractéristiques. De ce fait, nous avons eu toutes les informations exactes nécessaires pour suivre l'algorithme numériquement jusqu'à ce point. On se retrouve donc dans la même situation que dans le cas initial : il nous reste une liste de polynômes numériques que nous séparerons ultérieurement si nécessaire.

Définition 25. *Nous dirons que deux polynômes numériques $\tilde{H}_1$ et $\tilde{H}_2$ sont* ***frères*** *s'ils sont définis par l'algorithme* `Numerical-NPuiseux` *à partir de séries d'arêtes et sommets de $\mathcal{T}(F)$ étiquetées identiquement.*

Avant de décrire les algorithmes qui nous permettent de faire cela, nous allons commencer par traiter un exemple qui illustre le procédé de filtre à deux étages. Cet exemple est dérivé d'un exemple présenté sur un poster lors de la conférence ISSAC 2007. On peut noter que cet exemple ne présente aucune difficulté numérique : il a ici uniquement une fonction illustrative du processus de filtrage.

Nous considérons ici un polynôme $F \in K[x, y]$ de degré 68 en y et 60 en x. Après calculs modulaires, nous obtenons que les cycles de séries de Puiseux ont la forme suivante :

(1) $y = x^{1/2}(\alpha + x^{1/3}(\cdot + x^{1/12}(\cdot + \dots)))$
(2) $y = x^{1/2}(\alpha + x^{1/3}(\cdot + x^{1/12}(\cdot + \dots)))$
(3) $y = x^{1/2}(\beta + x^{1/2}(a + x^{1/2}(\cdot + \dots)))$
(4) $y = x^{1/2}(\beta + x^{1/2}(a + x^{1/2}(\cdot + \dots)))$
(5) $y = x^{1/2}(\beta + x^{1/2}(a + x^{1/2}(\cdot + \dots)))$
(6) $y = x^{1/2}(\beta + x^{1/4}(\cdot + x^{7/8}(\cdot + \dots)))$
(7) $y = x^{1/3}(\cdot + x^{4/15}(\cdot + \dots))$
(8) $y = x^{1/3}(\delta + x^{1/3}(b + x^{1/3}(\cdot + \dots)))$
(9) $y = x^{1/3}(\delta + x^{1/3}(\epsilon + x^{1/6}(\cdot + \dots)))$
(10) $y = x^{1/3}(\delta + x^{1/3}(\epsilon + x^{1/6}(\cdot + \dots)))$

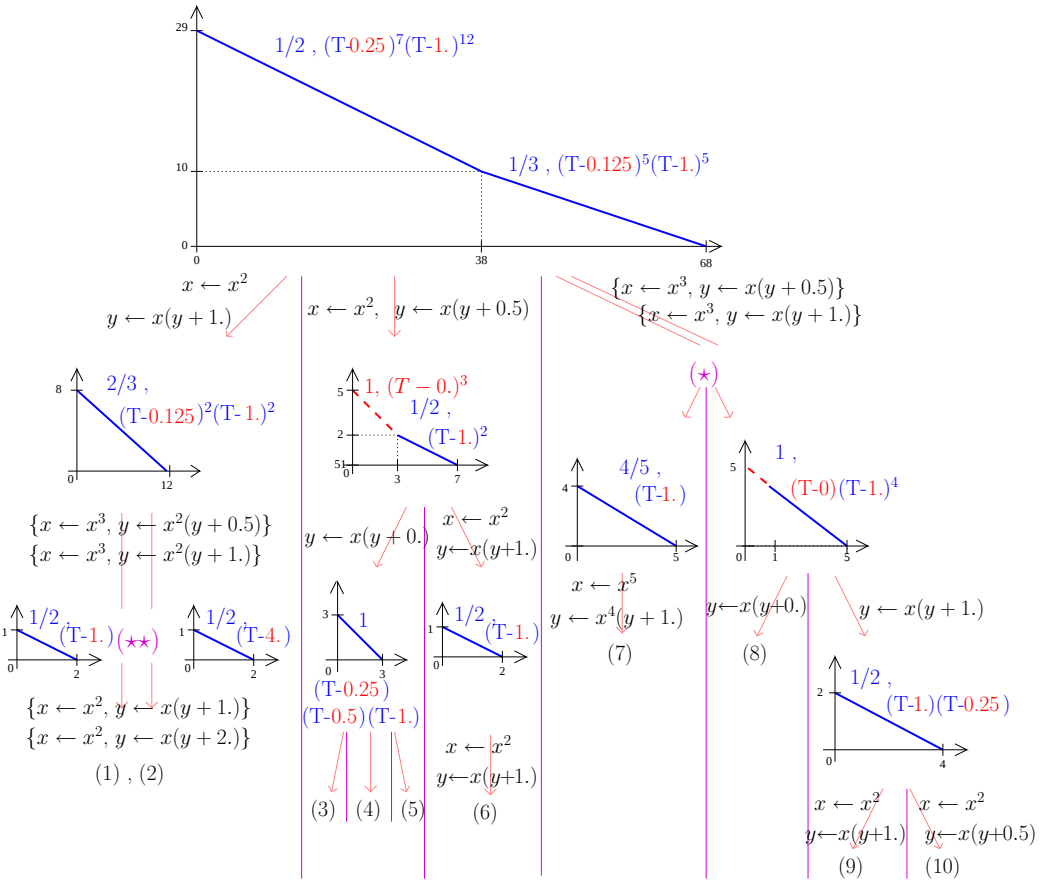

FIG. 2.6 – Suivre l'arbre des polygones

Ici, les coefficients représentés par des $\cdot$ sont des coefficients non nuls indéterminés. Les coefficients représentés par une même lettre grecque sont des coefficients non nuls et égaux. Enfin, les lettres a et b représentent des potentiels coefficients nuls, c'est-à-dire des coefficients nuls modulo p. Il nous faut donc calculer numériquement ces coefficients. S'ils sont effectivement nuls, on obtiendra des petites valeurs numériques, dont on peut espérer qu'elles n'influeront pas sur l'évaluation des séries. À l'aide de ces données calculées modulo p, on peut construire l'arbre des polygones $\mathcal{T}(F)$.

Sur la figure 2.6, nous avons représenté l'ensemble des polygones de Newton génériques considérés, ainsi que la factorisation sans facteur carré numérique des différents polynômes caractéristiques rencontrés au fur et à mesure de l'algorithme. Cette figure n'est pas donc une représentation précise de $\mathcal{T}(F)$, même si elle contient l'ensemble des informations contenues dans ce

dernier (excepté le polygone exceptionnel $\mathcal{EN}(F)$ qui ne présente pas ici d'intêret).

Nous allons détailler les deux situations qui apparaissent au niveau des symboles ($\star$) et ($\star\star$) de la figure 2.6.

Tout d'abord, si l'on considère le polynôme caractéristique associé à l'arête de pente 1/3 de $\mathcal{GN}(F)$, nous obtenons deux racines 0.25 et 1 de même multiplicité 5. Nous ne pouvons donc pas les séparer à ce stade de l'algorithme. À l'étape suivante, au niveau du symbole ($\star$), nous avons donc deux polynômes numériques, ainsi que deux sous-arbres de $\mathcal{T}(F)$. Ces sous-arbres possèdent une racine étiquetée par deux polygones de Newton différents. Cela va donc nous permettre de différencier nos deux polynômes numériques : en considérant le coefficient en x^4 de ces polynômes, on obtient deux coefficients 0. et 1. ; ainsi, le premier polynôme numérique correspond au polygone de Newton de droite, et le second à celui de gauche. Nous connaissons donc maintenant le polygone de Newton de chacun de nos polynômes numériques, ce qui nous permet de continuer nos calculs. Nous avons ici effectué le premier étage de notre filtre : il s'agit de l'utilisation de l'algorithme `Tri-Polygones`, décrit dans la section 2.4.1.

De la même façon, le polynôme caractéristique défini par l'arête de pente 2/3 du premier polygone de Newton de la colonne de gauche, situé au-dessus du symbole ($\star\star$), possède deux racines 1. et 0.125 ayant la même multiplicité 2. Par contre, à l'étape suivante, les deux polygones de Newton fournis par $\mathcal{T}(F)$ sont identiques, de même que la structure de multiplicité de l'unique arête de ce polygone. Il ne nous est donc pas possible de séparer les deux polynômes numériques. Mais nous possédons quand même leurs polygones de Newton, ainsi que leur structure de multiplicité. Cela nous permet donc de conduire nos calculs numériques. À la fin, nous finissons le calcul de la partie singulière sans avoir pu séparer ces deux cycles selon les branches de $\mathcal{T}(F)$. Mais cela ne nous est pas nécessaire, tant que nous connaissons les polygones de Newton et les structures de multiplicité des polynômes caractéristiques.

Nous allons maintenant donner la description de la partie numérique de notre algorithme, qui est donnée par la procédure `Numerical-NPuiseux`. Étant donné que nous faisons des calculs numériques, la notion de rationalité est ici inutile. Cet algorithme est donc basé sur l'algorithme classique `CNPuiseux`. De plus, pour pouvoir relier les calculs numériques aux informations exactes contenus dans $\mathcal{T}(F)$, nous utilisons deux algorithmes intermédiaires, `Tri-Polygones` et `Tri-Partitions`, décrits respectivement dans les parties 2.4.1 et 2.4.2. Enfin, comme l'algorithme traite plusieurs polynômes simultanément, il est nécessaire de connaître, pour chaque appel récursif, la

précédente racine qui a permis de calculer le polynôme $\tilde{H}$ considéré. En effet, dans la phase de "remontée" de l'algorithme (la dernière boucle `Pour` de l'algorithme), cette racine est nécessaire. Ainsi, nous utilisons également la table T pour établir ces connexions.

`Numerical-NPuiseux(`$\mathcal{H}, \mathcal{L}$`)`

`Entrée :`

$\mathcal{H}$: un ensemble de $(\tilde{H}, \tilde{\theta})$ où :
- *les $\tilde{H}$ sont des polynômes numériques ayant des ancêtres égaux dans $\mathcal{T}(F)$,*
- *$\tilde{\theta}$ est la dernière racine utilisée pour calculer $\tilde{H}$ ($\tilde{\theta} = 0$ pour l'appel initial).*

$\mathcal{L}$ l'ensemble des arbres de polygones possibles pour les éléments de $\mathcal{H}$.

`Sortie :`

ECHEC si la précision des approximations ne permet pas de conclure. Sinon, pour chaque polynôme $\tilde{H}$ de $\mathcal{H}$,
- *un ensemble de quadruplets $\{[\tilde{G}_i, P_i, \tilde{Q}_i, [\tilde{\xi}_i, \tilde{H}_i]]\}_i$, où :*
  - *$\tilde{H}_i$ est un polynôme numérique,*
  - *$\tilde{\xi}_i$ est l'approximation numérique de la racine du dernier polynôme caractéristique qui a permis de définir ce quadruplet*
  - *les triplets $\{[\tilde{G}_i, P_i, \tilde{Q}_i]\}_i$ forment une approximation numérique d'un ensemble de représentants pour :*
    - *les cycles de H au-dessus 0 pour l'appel initial,*
    - *les cycles de H qui s'annulent en $x = 0$ pour les appels récursifs.*

```
Début
  S_0 ← Tri-Polygones(H, L)
  Pour chaque (P, H_P, L_P) dans S_0 faire
    Si P = ((0,1),(1,0)) alors
      Retourner ∪_{(H̃,θ̃)∈H_P} {[H̃, x, y, [θ̃, H̃]]}
    Fin
    S_1, S_2 ← Tri-Partitions(P, H_P, L_P)
    Si S_1 = ECHEC alors Retourner ECHEC Fin
    Pour chaque arête Δ de P faire
      Calculer m, q, l
      Pour chaque (H', L') dans S_1 faire
      # Chaque racine du sous-arbre T ∈ L' est étiquetée par la même
      # partition M = (M_1^{α_1} ... M_s^{α_s}).
        Pour i allant de 1 à s faire
          H_i ← {}
          Pour chaque (H̃, θ̃) dans H' faire
```

```
                $\xi \leftarrow \mathcal{S}_2[\tilde{H}, \Delta]$
                Pour $j = 1 \dots \alpha_i$ faire
                  $\tilde{\alpha} \leftarrow \tilde{\xi}_{ij}^{1/q}$
                  # Les $\tilde{\xi}_{ij}$ sont les racines de multiplicité $M_i$ dans $\tilde{\xi}$
                  $\tilde{H}_0 \leftarrow \tilde{H}(x^q, x^m(\tilde{\alpha} + y))/x^l$
                  $\mathcal{H}_i \leftarrow \mathcal{H}_i \cup \{(\tilde{H}_0, \tilde{\alpha})\}$
                  $T[\tilde{\alpha}, \tilde{H}_0] \leftarrow [\tilde{\theta}, \tilde{H}]$
                Fin
              Fin
              $\mathcal{S}_3 \leftarrow$ Numerical-NPuiseux($\mathcal{H}_i, \mathcal{L}_i$)
              #$\mathcal{L}_i$ est l'ensemble des sous-arbres correspondant à $\mathcal{H}_i$
              Si $\mathcal{S}_3$ = ECHEC alors Retourner ECHEC
              Pour chaque $[\tilde{G}, P, \tilde{Q}, [\tilde{\beta}, \tilde{H}]]$ dans $\mathcal{S}_3$ faire
                $\mathcal{R} \leftarrow \mathcal{R} \cup \{[\tilde{G}, P^q, P^m(\tilde{\beta} + Q), T[\tilde{\beta}, \tilde{H}]]\}$
              Fin
            Fin
          Fin
        Fin
      Fin
    Retourner $\mathcal{R}$
Fin
```

2.4.1 Relier les polynômes aux polygones

Dans cette partie, nous décrivons, sans néanmoins le formaliser, l'algorithme `Tri-Polygones`. Nous avons donc un ensemble $\mathcal{H}$ de polynômes numériques $\tilde{H}$, ainsi que l'ensemble $\mathcal{L}$ des sous-arbres de $\mathcal{T}(F)$ correspondant à ces polynômes.

`Tri-Polygones`($\mathcal{H}, \mathcal{L}$)

`Entrée :`

$\mathcal{H}$ *: une liste de couples* $\{(\tilde{H}, \tilde{\theta})\}$ *où :*

- *les* $\tilde{H}$ *sont des polynômes numériques frères dans* $\mathcal{T}(F)$*,*
- $\tilde{\theta}$ *est la dernière racine utilisée pour calculer* $\tilde{H}$ *(*$\tilde{\theta} = 0$ *pour l'appel initial).*

$\mathcal{L}$ *: l'ensemble des arbres de polygones associés aux éléments de* $\mathcal{H}$

`Sortie :`

Pour chaque polygone de Newton $\mathcal{P}$ qui étiquette la racine d'un arbre de $\mathcal{L}$, un triplet $(\mathcal{P}, \mathcal{H}_\mathcal{P}, \mathcal{L}_\mathcal{P})$ où :
- *$\mathcal{H}_\mathcal{P}$ est la liste de polynômes numériques dont le polygone de Newton est $\mathcal{P}$.*
- *$\mathcal{L}_\mathcal{P}$ est l'ensemble des arbres de polygones de $\mathcal{L}$ ayant $\mathcal{P}$ pour racine.*

Pour décrire cet algorithme, nous introduisons :
- $\mathcal{D} = \{\mathcal{P}_1, \ldots, \mathcal{P}_r\}$, l'ensemble des polygones de Newton distincts qui étiquettent les racines des arbres de polygones dans $\mathcal{L}$,
- $n = [n_1, \ldots, n_r]$, un ensemble d'entiers tel que n_k est le nombre d'arbres de polygones dans $\mathcal{L}$ ayant pour racine $\mathcal{P}_k$.

Nous scindons $\mathcal{H}$ en fonction des polygones $\mathcal{P}_j$ de la façon suivante :

1. Si $r = 1$, alors retourner l'ensemble $(\mathcal{H}, \mathcal{L})$.
2. On calcule l'enveloppe convexe inférieure $\mathcal{P}$ de l'union des polygones de Newton dans $\mathcal{D}$.
3. Il existe un point $(i, j) \in \mathcal{P}$ qui appartient à l'un des $\mathcal{P}_k$, mais pas à l'ensemble des polygones. On pose alors $\mathcal{D}_1 = \{\mathcal{P}_k \mid (i, j) \in \mathcal{P}_k\}$ et $\mathcal{D}_2 = \mathcal{D} \backslash \mathcal{D}_1$. On notera ce dernier $\mathcal{D}_2 = \{\mathcal{P}_{k_1}, \ldots, \mathcal{P}_{k_r}\}$.
4. Ensuite, on trie les éléments de $\mathcal{H}$ par valeur décroissante de la valeur absolue de leur coefficient en $x^j y^i$, puis on scinde les éléments de $\mathcal{H}$ de la manière suivante : les $N_1 = n_{k_1} + \cdots + n_{k_r}$ premiers polynômes forment un ensemble $\mathcal{H}_1$, et les autres forment un ensemble $\mathcal{H}_2$.
5. Enfin, nous appliquons récursivement l'algorithme `Tri-Polygones` aux couples $(\mathcal{D}_1, \mathcal{H}_1)$ et $(\mathcal{D}_2, \mathcal{H}_2)$, obtenant ainsi deux ensembles $\mathcal{R}_1$ and $\mathcal{R}_2$. Nous retournons enfin l'ensemble $\mathcal{R}_1 \cup \mathcal{R}_2$.

2.4.2 Relier les polynômes aux partitions

Nous considérons maintenant l'algorithme `Tri-Partitions`. Nous disposons donc d'un ensemble $\mathcal{H}' = \{\tilde{H}_i\}_{1 \leq i \leq s}$ de polynômes numériques qui ont le même polygone de Newton $\mathcal{P}$, ainsi que l'ensemble des sous-arbres de $\mathcal{T}(F)$ associés à ces polynômes. Nous considérons ici une arête Δ de $\mathcal{P}$, et notre but est de calculer, pour chaque polynôme caractéristique $\tilde{\phi}_\Delta^{(i)}$ défini par $\tilde{H}_i$ et Δ, une approximation numérique de ses racines, en respectant la structure de multiplicité $[\phi_\Delta^{(i)}]$. Nous rappelons que $[\phi_\Delta^{(i)}]$ n'est pas connu, mais que l'on connaît uniquement un ensemble de candidats donnés par $\mathcal{T}(F)$.

Plus précisément, chaque polynôme exact H_i définit un polynôme caractéristique associé à l'arête Δ, que nous noterons dorénavant P_i. Ce polynôme

P_i possède une structure de multiplicité $[P_i]$. Nous ne disposons pas des polynômes exacts, mais uniquement d'une approximation $\tilde{H}_i$, et donc d'une approximation $\tilde{P}_i$ des P_i. Nous souhaitons donc trouver une approximation des racines multiples de polynômes connus de manière inexacte.

Pour ce faire, nous proposons une stratégie s'appuyant sur la décomposition en valeur singulière. Pour simplifier la rédaction, nous faisons uniquement référence aux travaux de Z. Zeng dans [Zen03, Zen05], dont le but est précisément de calculer les racines multiples de polynômes connus de manière inexacte. Outre le fait que ces travaux soient reconnus par la communauté du calcul formel pour leur qualité, nous verrons comment l'utilisation de la décomposition en valeur singulière (SVD) permet d'utiliser l'information exacte que nous possédons, à savoir les s partitions de $d = \deg(P_1) = \cdots = \deg(P_s)$, correspondant à l'ensemble des structures de multiplicité des polynômes $(P_1, \ldots, P_s)$.

Nous allons donc commencer par décrire les idées utilisées dans la stratégie développée dans [Zen03, Zen05], avant de montrer comment utiliser la liste des structures de multiplicité.

Stratégie développée par Z. Zeng

Dans [Zen03, Zen05], Z. Zeng présente deux algorithmes, qui utilisés ensemble, permettent de calculer les racines multiples d'un polynôme dont les coefficients sont connus de manière inexacte. Le premier algorithme, décrit dans la section 3 de ces articles, permet de calculer à une grande précision ces racines à partir des multiplicités des racines, ainsi qu'une première approximation des racines. Suivant [Kah72], l'auteur transforme le problème en un problème des moindres carrés sur une variété définie à l'aide de la structure de multiplicité.

Le second algorithme, décrit dans la section 4 de [Zen03, Zen05], calcule l'approximation initiale nécessaire au premier algorithme, ainsi que les multiplicités des racines du polynôme. Pour cela, cet algorithme utilise des calculs de plus grand diviseur commun (pgcd) de polynômes approchés et de leur dérivée, pour obtenir une factorisation sans facteur carré du polynôme considéré.

Avant de rentrer plus en détail dans cet algorithme, nous allons commencer par rappeler la définition du plus grand diviseur commun approché. En effet, le calcul exact du pgcd de deux polynômes est un problème mal conditionné : la plupart du temps, si l'on considère deux polynômes ayant un pgcd non trivial, et que l'on applique l'algorithme symbolique du calcul de pgcd à une

approximation de ces deux polynômes, on obtient un pgcd constant. Pour résoudre ce problème, de nombreux auteurs ont étudié le calcul d'un pgcd approché de deux polynômes connus de manière inexacte (voir par exemple [Zen04] pour une bibliographie détaillée).

Suivant [Zen03], nous rappelons les propriétés définissant le **plus grand diviseur commun approché** de deux polynômes P et Q, que nous noterons $u = agcd(P, Q)$:

1. *Proximité* : u est le pgcd exact de deux polynômes P_0 et Q_0 proche des deux polynômes P et Q,
2. *Degré maximal* : u est le polynôme de plus grand degré parmi les polynômes proches de P et Q,
3. *Distance minimale* : Le couple (P_0, Q_0) minimise la distance entre (P, Q) et l'ensemble des polynômes définissant un pgcd de degré $\deg(u)$.

Il existe différentes méthodes pour calculer un tel pgcd approché. Celle présentée dans les travaux de Z. Zeng est basée sur la décomposition en valeur singulière, et a été introduite dans [CGTW95]. L'algorithme II de [Zen03, Zen05] utilise cette décomposition : elle permet de trouver le degré du pgcd cherché, et ainsi, en faisant des appels successifs à l'algorithme de calcul de pgcd, la structure de multiplicité du polynôme considéré.

Plus précisément, étant donné deux polynômes P et Q, il existe r valeurs singulières nulles de la matrice de Sylvester de P et Q, où r est le degré de $pgcd(P, Q)$. L'intérêt de cette décomposition est qu'elle est numériquement stable. De ce fait, si les polynômes P et Q sont légèrement perturbés, il y aura r valeurs singulières "petites". Ainsi, l'idée est que si un seuil de tolérance est donné en entrée de l'algorithme, il est possible de déterminer le degré du pgcd cherché en comptant le nombre de valeurs singulières inférieures à un nombre dépendant de ce seuil (voir [Zen05, section 4.4] pour plus de détails sur ce seuil). De plus, pour économiser des calculs, il est possible de calculer les valeurs singulières par valeur décroissante.

Enfin, le vecteur singulier à droite de la décomposition SVD fournit une approximation initiale des polynômes v et w tels que, si $u = agcd(P, Q)$, $P = u \cdot v$ et $Q = u \cdot w$. À l'aide de cette donnée, on peut trouver une première approximation de u en résolvant un système linéaire. Ensuite, cette première approximation du triplet (u, v, w) est rafinée à l'aide d'une itération de Gauss-Newton qui permet de trouver le pgcd cherché.

En suivant ce procédé, étant donné un polynôme f, on peut ainsi calculer $u_0 = agcd(f, f')$, puis $u_1 = agcd(u_0, u_0')$ etc. En recommençant jusqu'à obte-

nir un pgcd constant, on obtient finalement une factorisation sans facteurs carrés de f, et donc la structure de multiplicité de f, ainsi qu'une approximation des racines multiples en utilisant un algorithme classique de résolution numérique sur la partie sans carré de f.

Utilisation de l'ensemble des partitions de $d_0 = \deg(P^{(k)})$

Comme vu dans l'introduction de cette section, nous disposons en plus des s polynômes $\tilde{P}^{(k)}$ de même degré d_0, de l'ensemble des s partitions de d_0 correspondant à l'ensemble des structures de multiplicité des $P^{(k)}$. Une première stratégie pourrait donc être la suivante :

1. Calculer les racines multiples des polynômes $\tilde{P}^{(k)}$ à l'aide des algorithmes de Z. Zeng,
2. Vérifier que l'ensemble des structures de multiplicité trouvées correspond à l'ensemble des partitions données par les calculs modulaires.

Néanmoins, il est possible d'utiliser la liste de partition à notre disposition au sein de l'algorithme de calcul de pgcd approché. En effet, comme nous possédons l'ensemble des structures de multiplicité des $P^{(k)}$, nous possédons des candidats pour les degrés de l'ensemble des pgcd approchés calculés lors de l'algorithme de Z. Zeng. Il est ainsi possible de séparer les polynômes $P^{(k)}$ à chaque calcul de pgcd effectué dans les algorithmes de Z. Zeng. De plus, cela nous permet d'obtenir un algorithme qui ne nécessite pas d'utiliser de seuil de tolérance pour les valeurs singulières, comme ce serait le cas avec une utilisation directe des algorithmes de Z. Zeng. Ce travail de tri à l'aide des pgcd successifs est présenté dans l'algorithme récursif `Tri-pgcd`. Pour chaque polynôme $\tilde{P}^{(k)}$, nous définissons maintenant la suite de triplets $\left(\tilde{u}_0^{(k)}, \tilde{v}_0^{(k)}, \tilde{w}_0^{(k)}\right), \ldots, \left(\tilde{u}_t^{(k)}, \tilde{v}_t^{(k)}, \tilde{w}_t^{(k)}\right)$ de la façon suivante :

- $\left(\tilde{u}_0^{(k)}, \tilde{v}_0^{(k)}, \tilde{w}_0^{(k)}\right) = (\tilde{P}^{(k)}, 1, 1)$,
- Pour $1 \leq j \leq t$, $\left(\tilde{u}_j^{(k)}, \tilde{v}_j^{(k)}, \tilde{w}_j^{(k)}\right)$ est le triplet associé au pgcd de $\tilde{u}_{j-1}^{(k)}$ et $\tilde{u}_{j-1}^{(k)\prime}$,
- $\deg(u_t^{(k)}) = 0$ et $\deg(u_{t-1}^{(k)}) > 0$

Le principe de l'algorithme `Tri-pgcd` est de calculer la suite de triplets ainsi définie, qui permet ensuite de calculer la décomposition sans facteurs carrés du polynôme $\tilde{P}$. Ainsi, on fera appel à cet algorithme avec une liste $\mathcal{U} = \{[(\tilde{P}^{(k)}, 1, 1)]\}_{1 \leq k \leq s}$, puis à l'appel récursif, on ajoutera le triplet associé au pgcd de $\tilde{P}^{(k)}$ et $\tilde{P}^{(k)\prime}$ à chaque élément de cette liste, etc.

Enfin, cet algorithme utilise la fonction intermédiaire `Diminuer`, qui à une liste d'entiers $[\alpha_1, \ldots, \alpha_s]$ associe la liste $[\alpha_i - 1 \mid \alpha_i > 1]$. De plus, si l est une partition, nous notons $\#l$ son cardinal. Par exemple, on a $\#[1^2 2^3 3^1] = 6$.

`Tri-pgcd(`$\mathcal{U}, \mathcal{L}, d_r$`)`

`Entrée :`

$\mathcal{U} = \{[\left(\tilde{u}_0^{(k)}, \tilde{v}_0^{(k)}, \tilde{w}_0^{(k)}\right), \ldots, \left(\tilde{u}_r^{(k)}, \tilde{v}_r^{(k)}, \tilde{w}_r^{(k)}\right)]\}_{1 \leq k \leq s}$ *où :*

- *la suite* $\left(\tilde{u}_0^{(k)}, \tilde{v}_0^{(k)}, \tilde{w}_0^{(k)}\right), \ldots, \left(\tilde{u}_r^{(k)}, \tilde{v}_r^{(k)}, \tilde{w}_r^{(k)}\right)$ *est constituée des* r *premiers triplets de la suite définie précédemment.*
- *Les polynômes* $\tilde{u}_r^{(k)}, 1 \leq k \leq s$ *ont le même degré.*

d_r *: le degré des polynômes* $\tilde{u}_r^{(k)}$ *considérés.*

$\mathcal{L}$ *: une liste de couples* $\{(l^{(k)}, \mathcal{L}^{(k)})\}_{1 \leq k \leq s}$ *où :*

- $l^{(k)}$ *est une partition de* d_r*,*
- $\mathcal{L}^{(k)}$ *est l'arbre de polygones d'où provient* $l^{(k)}$*.*

`Sortie :`

ECHEC si la précision des approximations ne permet pas de conclure.

Sinon, un ensemble de couples $\{(\mathcal{U}', \mathcal{L}')\}$ *où :*

- $\mathcal{U}'$ *est une liste d'éléments* $[(\tilde{u}_0, \tilde{v}_0, \tilde{w}_0), \ldots, (\tilde{u}_t, \tilde{v}_t, \tilde{w}_t)]$ *de suite de triplets comme définie précédemment.*
- $\mathcal{L}'$ *est l'ensemble des arbres de polygones associés aux éléments de* $\mathcal{U}'$*.*

`Début`

1. `Si` $d_r = 0$ `alors Retourner`
   $(\{[(\tilde{u}_0^{(k)}, \tilde{v}_0^{(k)}, \tilde{w}_0^{(k)}), \ldots, (\tilde{u}_{r-1}^{(k)}, \tilde{v}_{r-1}^{(k)}, \tilde{w}_{r-1}^{(k)})]\}_{1 \leq k \leq s}, \{\mathcal{L}^{(k)}\}_{1 \leq k \leq s})$
   `Fin`
2. $\mathcal{M} \leftarrow \{\#l^{(k)} \mid 1 \leq k \leq s\}$; $m \leftarrow \sum_{\alpha \in \mathcal{M}} \alpha$
3. `Calculer les` m `plus grandes valeurs singulières associées aux polynômes` $\tilde{u}_r^{(k)}$`. Pour chaque polynôme` $\tilde{u}_r^{(k)}$`, définir` n_k `le nombre de valeurs singulières associées au polynôme` $\tilde{u}_r^{(k)}$ `ainsi calculées.`
4. `Si` $\mathcal{M} \neq \{n_k\}_{1 \leq k \leq s}$ `alors Retourner ECHEC Fin`
5. `Pour` $1 \leq k \leq s$`, calculer le triplet` $(\tilde{u}_{r+1}^{(k)}, \tilde{v}_{r+1}^{(k)}, \tilde{w}_{r+1}^{(k)})$ `correspondant au pgcd approché de` $\tilde{u}_r^{(k)}$ `et` $\tilde{u}_r^{(k)\prime}$ `de degré` n_k`.`
6. $\mathcal{R} \leftarrow \emptyset$
   `Pour chaque valeur distincte` α `de` $\mathcal{M}$ `faire`
   $\mathcal{U}_\alpha \leftarrow \{[(\tilde{u}_0^{(k)}, \tilde{v}_0^{(k)}, \tilde{w}_0^{(k)}), \ldots, (\tilde{u}_{r+1}^{(k)}, \tilde{v}_{r+1}^{(k)}, \tilde{w}_{r+1}^{(k)})] \mid \deg(u_{r+1}^{(k)}) = \alpha\}$

$\mathcal{L}_\alpha \leftarrow \{(\texttt{Diminuer}(l^{(k)}), \mathcal{L}^{(k)}) \mid d - \#l^{(k)} = \alpha\}$
$\mathcal{R} \leftarrow \mathcal{R}, \texttt{Tri} - \texttt{pgcd}(\mathcal{U}_\alpha, \mathcal{L}_\alpha, \alpha)$
`Fin`

7. `Retourner` $\mathcal{R}$.

`Fin`

Pour le calcul des m plus grandes valeurs singulières (point 3 de l'algorithme `Tri-pgcd`), on peut procéder de la manière suivante :

- Calculer la plus grande valeur singulière associée à chaque polynôme $u_r^{(k)}$, $1 \leq k \leq s$.
- Tant que $m+s$ valeurs singulières n'ont pas été calculées, calculer la valeur singulière associée au polynôme dont la plus petite valeur singulière calculée est la plus grande parmi l'ensemble suivant :

$$\{\text{plus petite valeur singulière calculée pour le polynôme } u_r^{(k)} \mid 1 \leq k \leq s\}$$

En cas dégalité, n'importe quel choix parmi l'ensemble des plus petites valeurs convient, excepté s'il ne reste plus qu'une valeur singulière à calculer (dans ce cas, on retourne `ECHEC`). De plus, si l'on a déjà calculé les d_r valeurs singulières associées à l'un des polynômes, on ignore ce polynôme pour le choix des valeurs singulières à traiter.

En ce qui concerne le calcul des valeurs singulières par valeurs décroissantes, nous renvoyons le lecteur aux travaux de Z. Zeng.

Ainsi, l'un des intérêts de cet algorithme est qu'il ne nécessite pas de définir un seuil de tolérance. En effet, c'est ici la donnée des structures de multiplicité qui nous permet de savoir quand arrêter le calcul des valeurs singulières.

Enfin, nous pouvons maintenant décrire l'algorithme `Tri-Partitions` :

`Tri-Partitions`$(\mathcal{P}, \mathcal{H}, \mathcal{L})$

`Entrée :`
$\mathcal{P}$: un polygone de Newton,
$\mathcal{H}$: une liste de polynômes numériques $\{\tilde{H}\}$ ayant le même polygone de Newton $\mathcal{P}$,
$\mathcal{L}$: l'ensemble des sous-arbres de $\mathcal{T}(F)$ associés à ces polynômes $\tilde{H}$,
`Sortie :`

$\mathcal{R}$ *une liste contenant, pour chaque ensemble de partition* $\{[\phi_\Delta]\}_{\Delta\in\mathcal{P}}$*, où chaque* $[\phi_\Delta]$ *étiquette le fils d'arête* Δ *de* $\mathcal{P}$ *du même sous-arbre de* $\mathcal{L}$*, un couple* $(\mathcal{H}', \mathcal{L}')$ *tel que :*

- $\mathcal{H}'$ *est une liste de polynômes numériques dont chaque polynôme caractéristique* $\tilde{\phi}_\Delta$ *associé à l'arête* Δ *de* $\mathcal{P}$ *a pour partition* $[\phi_\Delta]$*.*
- $\mathcal{L}'$ *est l'ensemble des éléments de* $\mathcal{L}$ *tel que chaque fils de* $\mathcal{P}$ *d'arête étiquetée par* Δ *est étiqueté par la partition* $[\phi_\Delta]$*.*

$\mathcal{S}$ *une table à double entrée telle que* $\mathcal{S}[\tilde{H}, \Delta]$ *donne les racines multiples du polynôme caractéristique associé à* $\tilde{H}$ *et* Δ*.*

```
Début
  R ← {(H, L)}
  Pour chaque arête Δ de P faire
    Calculer d_Δ la longueur de Δ divisée par q
    R_1 ← {}
    Pour chaque couple (H_0, L_0) ∈ R faire
      H_1 ← {}
      Pour chaque polynôme H̃ ∈ H_0 faire
        Calculer φ̃_Δ le polynôme caractéristique associé à H̃
        H_1 ← H_1 ∪ {[H̃, φ̃_Δ]}
      Fin
      L_1 ← {}
      Pour chaque sous-arbre T ∈ L_0 faire
        Extraire la partition l_0 qui étiquette le fils d'arête
          Δ de P
        L_1 ← L_1 ∪ {[l_0, T]}
      Fin
      R_0 ← Tri-pgcd(H_1, L_1, d_Δ)
      Si R_0 =ECHEC alors Retourner ECHEC,ECHEC Fin
      Pour chaque couple (U, L_0) ∈ R_0 faire
        Définir H_0 l'ensemble des polynômes bivariés corres-
          pondant aux polynômes caractéristiques de U
        R_1 ← R_1 ∪ {(H_0, L_0)}
        Pour chaque [H̃, (ũ_0, ṽ_0, w̃_0), ..., (ũ_t, ṽ_t, w̃_t)] ∈ U faire
          Calculer l'ensemble ξ des racines multiples de P̃ à
            partir des triplets (ũ_i, ṽ_i, w̃_i)_{1≤i≤t} [Zen05, section 4.3]
          S[H̃, Δ] ← ξ
        Fin
      Fin
    Fin
    R ← R_1
```

```
  Fin
  Retourner (R, S)
Fin
```

2.4.3 Cas général

Dans cette section, nous avons pour l'instant considéré que nous étions au-dessus du point $x = 0$. Si l'on souhaite traiter les développements de Puiseux au-dessus de n'importe quel point critique, cela demande d'effectuer plus qu'un simple changement de variable $x \leftarrow x + x_0$. Une telle stratégie serait suffisante si l'on connaissait la factorisation symbolique du discriminant de F en y. En effet, on pourrait alors, pour chaque facteur irréductible :

1. Calculer une racine x_0 de ce facteur modulo p, effectuer un changement de variable $x \leftarrow x + x_0$ modulo p, puis calculer l'arbre des polygones $\mathcal{T}_0(F)$ du polynôme ainsi défini. Nous rappelons que, comme chaque racine engendre le même arbre de polygone, n'importe quel choix de racine convient.

2. Calculer l'ensemble des racines numériques du facteur irréductible considéré, puis pour chaque racine $\tilde{x}_1$, faire le changement de variable $x \leftarrow x+\tilde{x}_1$ et appliquer l'algorithme `Numerical-NPuiseux` en suivant l'arbre de polygones $\mathcal{T}_0(F)$.

Néanmoins, calculer la factorisation dans $K[x]$ du discriminant de F en y peut s'avérer coûteux. Pour éviter ce calcul, on peut se contenter de calculer la factorisation modulo p du discriminant, pour laquelle on dispose d'algorithmes rapides. Puis, pour chaque facteur irréductible de cette factorisation, on calcule l'arbre des polygones défini par le polynôme translaté de l'une des racines de ce facteur irréductible. Ensuite, on calcule l'ensemble des racines multiples numériques du discriminant à l'aide de la structure de multiplicité obtenue modulo p. À nouveau, comme il n'y a pas de correspondance entre $\overline{\mathbb{F}}_p$ et $\overline{K}$, il n'est pas possible de relier les polynômes numériques définis par translation à partir des racines numériques de multiplicité M aux arbres de polygones correspondants à ces polynômes (qui ont été calculés à partir des racines de multiplicité M modulo p).

Ainsi, nous traiterons ces racines simultanément, faisant appel à l'algorithme `Numerical-NPuiseux` avec ces deux ensembles. Nous séparerons ces ensembles dans la suite de l'algorithme à l'aide du processus de filtre à deux étages décrit précédemment.

2.5 Exemples

Durant cette thèse, nous avons programmé un prototype de l'algorithme `Numerical-NPuiseux`. Par manque de temps, ce dernier n'utilise pas à l'heure de la rédaction les algorithmes basés sur les travaux de Z. Zeng présentés dans la section 2.4.2, mais suit l'algorithme proposé dans [Pot07]. Néanmoins, ce prototype donne des résultats encourageant autant sur le point de la stabilité numérique que sur les temps de calcul. Ce prototype a été réalisé avec Maple, et les exemples présentés ici ont été calculés avec Maple 10.

Notons $M_{a,d}(x)$ le polynôme $x^d - 2(ax-1)^2$. Cette famille de polynômes vient de [Mig92, page 170]. La spécificité du polynôme $M_{a,d}(x)$ est qu'il possède deux racines réelles distantes de moins de $2a^{-\frac{d+2}{2}}$.

Le premier exemple illustre la stabilité numérique constatée. Nous considérons ici le polynôme $F_1(x,y) = y^3 - M_{10,5}(x)$. Dans notre algorithme de monodromie (voir le chapitre 3), nous avons besoin de calculer les séries de Puiseux au-dessus des racines du polynôme $M_{10,5}(x)$ à l'ordre $\frac{16}{3}$. Nous comparons ici le nombre de chiffres exacts des coefficients obtenus de deux façons différentes :

- en calculant les développements de Puiseux symboliquement (utilisation de la commande `algcurves[puiseux]` de Maple) puis en les évaluant numériquement (deuxième colonne),
- en utilisant notre prototype (troisième colonne).

La première colonne du tableau donne le nombre de chiffres de précision de la mantisse, c'est-à-dire la variable `Digits` de Maple.

Digits	évaluation numérique	algorithme `Numerical-NPuiseux`
10	0	7
40	0	36
50	6	47

On voit ici qu'à une précision donnée, notre algorithme donne de meilleurs résultats numériques que l'évaluation numérique des développements symboliques. De plus, on obtient un nombre de chiffres significatifs très proche de la variable `Digits` utilisée.

Nous donnons maintenant un deuxième exemple illustrant ce phénomène : nous considérons le polynôme $F_2(x,y) = (y^3 - M_{10,6}(x))(y^3 - M_{10,3}(x)) + y^2x^5$. Son discriminant en y possède un facteur irréductible $P(x) \in \mathbb{Z}[x]$ de degré 30, avec certains coefficients plus grands que 10^{13}. Nous nous intéressons

ici uniquement au premier coefficient non nul des séries de Puiseux définies au-dessus des racines de $P(x)$, celui de $x^{\frac{1}{2}}$. On obtient alors :

Digits	évaluation numérique	algorithme `Numerical-NPuiseux`
10	0	4
20	0	15
30	5	29

Enfin, nous considérons une famille de polynômes définis de la manière récursive suivante :

$$G_n(x,y) = \left(y^{\lceil \frac{n}{2} \rceil} - P_{\lceil \frac{n}{2} \rceil}(x)\right) G_{\lfloor \frac{n}{2} \rfloor}(x,y)$$

où

$$P_{n_0}(x) = \frac{1}{{n_0}^{3!}} x \left(x^{n_0} + (n_0 - 1)\, x - \frac{1}{n_0!} \right).$$

Les séries de Puiseux au-dessus de 0 des polynômes G_n ont leur coefficients dans $\mathbb{Q}$. En l'absence de nombre algébrique, les calculs symboliques ne sont donc pas handicapés par les extensions de corps.

Nous faisons ici les calculs avec 10 chiffres de précision en ce qui concerne les calculs numériques. Nous nous intéressons ici aux temps de calcul, comparant les temps mis par l'algorithme symbolique `algcurves[puiseux]` de Maple et par l'algorithme `Numerical-NPuiseux`(incluant le calcul de $\mathcal{T}(F)$) pour calculer la partie singulière des développements de Puiseux au-dessus de 0. Nous donnons également le nombre de chiffres exacts obtenus pour les coefficients des séries de Puiseux en utilisant l'algorithme `Numerical-NPuiseux`.

Polynôme considéré	algorithme symbolique temps en seconde	algorithme `Numerical-NPuiseux`	
		temps en secondes	précision
G_8	0.031	0.029	9
G_{12}	0.041	0.099	9
G_{16}	2.3	0.221	9
G_{20}	0.751	0.550	9
G_{24}	2.889	0.920	9
G_{28}	8.509	1.719	9
G_{32}	30.820	5.040	9

2.6 Conclusions et Perspectives

La méthode hybride que nous avons introduite nous permet de calculer une approximation numérique des séries de Puiseux au-dessus des points

critiques, avec une précision et une stabilité qui semble expérimentalement satisfaisante. À notre connaissance, cette façon de calculer en se servant de calculs modulaires pour guider des calculs numériques est nouvelle.

En ce qui concerne la partie modulaire de l'algorithme, le critère de bonne réduction nous permet de certifier l'exactitude des informations trouvées, et ainsi de calculer l'arbre des polygones $\mathcal{T}(F)$ à l'aide de calculs modulaires. De plus, la taille du premier p utilisé pour calculer l'arbre des polygones est petit, c'est-à-dire logarithmique en la taille des entrées, si l'on utilise l'un des deux algorithmes probabilistes `MCGoodPrime` ou `LVGoodPrime`. Le premier ne certifie pas l'exactitude du résultat, contrairement au deuxième. De plus, dans le cadre du calcul du groupe de monodromie, nous avons déjà calculé le discriminant de F en y, ce qui nous permet d'utiliser l'algorithme de type Las-Vegas sans coût supplémentaire. Enfin, dans le cas où F est unitaire en y, nous avons donné de nouvelles bornes pour la complexité de la partie modulaire de notre algorithme, améliorant les résultats existants.

Ensuite, nous avons montré comment utiliser cet arbre de polygones pour calculer numériquement les développements de Puiseux. Nous avons ainsi présenté un processus de filtre à deux étages pour pallier à la non correspondance entre $\overline{\mathbb{F}}_p$ et $\mathbb{C}$, ce qui nous permet de connecter les informations obtenues lors du calcul modulaire aux calculs numériques. Nous avons notamment montré comment utiliser les travaux de Z. Zeng pour calculer les racines multiples des polynômes caractéristiques en utilisant les structures de multiplicités potentielles fournies par l'arbre des polygones.

Néanmoins, ce travail reste ouvert. Tout d'abord, la complexité de la partie modulaire de l'algorithme ne traite pas le cas non unitaire. Il semblerait pourtant que celui-ci puisse se traiter dans la même complexité que le cas unitaire. Mais nous n'avons pas achevé à l'heure actuelle la preuve d'un tel résultat. Ensuite, la partie numérique de cet algorithme nécessite encore du travail : nous ne traitons pas ici la précision numérique nécessaire, et ne pouvons donc obtenir de bornes sur la complexité binaire de la partie numérique. De même, nous ne gérons pas non plus la propagation des erreurs numériques. Avec de telles informations, il serait alors possible de certifier l'algorithme, en prenant en compte les bornes sur la précision des approximations utilisées. De plus, il n'existe pas à l'heure actuelle d'implémentation de l'algorithme `Tri-Partitions`. Celui-ci pourra se faire à partir du paquetage développé par Z. Zeng pour Maple et Magma, où sont programmés les algorithmes décrits dans [Zen05], mais il n'est pas sûr qu'une utilisation de la stratégie décrite dans le paragraphe 2.4.2 ne nécessite pas de reprendre le code proposé dans ces paquetages.

Chapitre 3

Calcul du groupe de monodromie d'une courbe algébrique plane

Dans ce chapitre, on considère un corps de nombres algébriques K, $\overline{K}$ sa clôture algébrique, et $\mathcal{C} = \{(x_0, y_0) \in \mathbb{C}^2 \,|\, F(x_0, y_0) = 0\}$ une courbe algébrique plane, où $F = y^{d_y} + \sum_{k=0}^{d_y-1} a_k(x)y^k \in K[x, y]$ est un polynôme que nous supposerons sans facteur carré, unitaire et primitif en y. On notera d_x le degré de F en x, et l'on supposera que ce degré est strictement positif.

L'objet de ce chapitre est de décrire une méthode pour calculer le groupe de monodromie $\mathcal{M}$ de la courbe $\mathcal{C}$. Une première version de ce travail a été publiée dans [Pot07].

3.1 Introduction

Comme dans le chapitre 1, notons $\alpha_1, \ldots, \alpha_n$ l'ensemble des points critiques complexes de la courbe $\mathcal{C}$ (comme précisé dans la section 1.2.2 du chapitre 1, nous ne nous intéressons pas ici au point $x = \infty$). Le cas $n = 1$ étant facile à traiter, nous supposerons que $n > 1$ afin d'éviter les cas extrêmes dans notre raisonnement. Les points critiques étant en nombre fini, on peut choisir un point de base régulier $a \in \mathbb{C}\backslash\{\alpha_1, \ldots, \alpha_n\}$ tel qu'il n'y ait pas deux points critiques distincts alignés avec le point a. On peut ensuite ordonner les points critiques par valeur croissante de l'argument $\arg(\alpha_i - a)$, comme dans [TT84]. Pour calculer le groupe de monodromie $\mathcal{M}$, nous utiliserons des chemins homotopes à ceux de la figure 3.1, puisque ce sont ces chemins que Tretkoff et Tretkoff utilisent dans [TT84] pour calculer l'homologie de la surface de Riemann. Notre but est donc de trouver la permutation

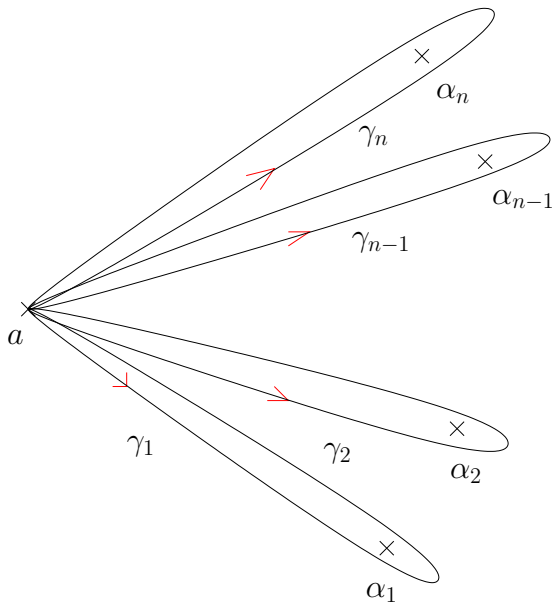

FIG. 3.1 – Chemins pour le monodromie

engendrée par chacun des lacets γ_i de la figure 3.1.

Pour ce faire, la méthode que nous proposons dans ce chapitre suit le schéma classique suivant :

1. Pour chaque lacet γ, on choisit une suite de points intermédiaires $a = x_0, x_1, \ldots, x_k = a$.
2. On calcule les fibres $\mathcal{F}(x_i)$ pour chaque point intermédiaire.
3. On apparie les éléments de $\mathcal{F}(x_i)$ à ceux de $\mathcal{F}(x_{i+1})$ bijectivement, de telle manière que deux éléments connectés correspondent au même prolongement analytique (voir la figure 3.2).

Il existe d'autres méthodes qui suivent ce schéma (voir la section 3.2.1). C'est le choix des chemins, ainsi que la façon dont les fibres sont calculées et connectées, qui caractérisent la méthode.

Dans ce chapitre, nous reprenons les principes de la méthode décrite dans [Pot07]. Les contributions apportées par ce travail sont les suivantes :

– Nous utilisons un arbre de recouvrement minimum pour la distance euclidienne de l'ensemble $\mathcal{V} = \{a, \alpha_1, \ldots, \alpha_n\}$, ceci afin de réduire la longueur totale des chemins que nous aurons à suivre. En effet, le prolongement analytique effectué le long de chaque portion de l'arbre commune

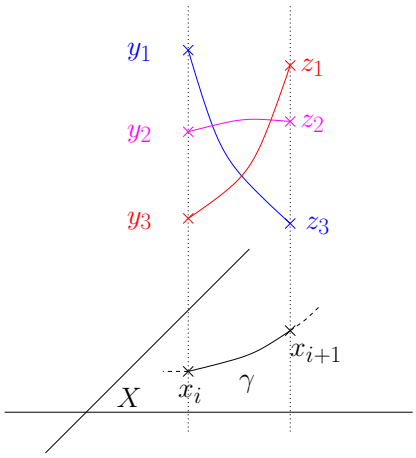

FIG. 3.2 – Méthode « relier les fibres »

à plusieurs lacets ne sera effectué qu'une seule fois. Nous proposons ensuite dans la partie 3.4 deux méthodes pour obtenir des chemins suivant cet arbre qui soient homotopes aux γ_i. Ce dernier point rectifie les chemins donnés dans [Pot07], qui n'étaient pas corrects : les contraintes d'espace nous ont amené à simplifier exagérément les explications, qui donnaient de ce fait des lacets pas forcément homotopes aux γ_i.

- Nous relions les fibres entre deux points intermédiaires à l'aide de développements en série tronqués à un ordre contrôlé. Nous donnons notamment des bornes sur les ordres de troncation afin d'avoir des connexions certifiées (proposition 34). De plus, nous calculons de tels développements au-dessus de points réguliers, mais aussi au-dessus des points critiques. Ces derniers décrivent les fonctions lorsqu'elles sont prolongées le long d'un arc de cercle autour d'un point critique (proposition 33), et nous donnent la monodromie locale (proposition 32).
- Enfin, dans la partie 3.5.2, nous étudions la complexité du prolongement analytique pour obtenir un compromis entre les ordres de troncation et le nombre de points intermédiaires à utiliser pour chaque arête de l'arbre. Nous donnons aussi une borne sur le nombre total de points intermédiaires à considérer en utilisant cette méthode. Une première stratégie de compromis entre le nombre de points intermédiaires et les ordres de troncation est présentée dans [Pot07]. Celle proposée ici est un raffinement de cette approche.

Pour finir cette introduction, nous rappelons qu'étant donné un point x_0 de $\mathbb{C}$, nous notons $\delta(x_0)$ la distance entre x_0 et son plus proche point critique (x_0

excepté s'il est lui même un point critique). De plus, nous noterons comme dans le chapitre 1 $X_0 = \mathbb{C}\backslash\{\alpha_1, \ldots, \alpha_n\}$ le plan complexe privé des points critiques de F.

3.2 État de l'art

Dans cette partie, nous décrivons brièvement les différentes approches existantes pour calculer le groupe de monodromie.

3.2.1 Méthode « relier les fibres »

Cette partie détaille les méthodes suivant la stratégie décrite dans la précédente section.

Dans [DvH01], Bernard Deconinck et Mark van Hoeij décrivent l'algorithme qu'ils ont implanté en Maple (commande `monodromy` du paquetage `algcurves`). Leur stratégie consiste à éviter les points critiques en choisissant des chemins qui contournent les points critiques à une distance au moins égale à $\frac{2}{5}\delta(\alpha_i)$ pour chaque point critique α_i. Ensuite, le prolongement analytique est effectué de la manière suivante : étant donnés deux points successifs d'un chemin x_1 et x_2, la fibre en x_2 est approchée à l'aide d'un développement en série à l'ordre 1 au dessus de x_1 (c'est-à-dire en utilisant uniquement la dérivée première) ; ils connectent chaque point de la fibre en x_2 à la plus proche estimation calculée à partir de $\mathcal{F}(x_1)$, à condition que les approximations soient suffisamment proches (dans le cas contraire, un point intermédiaire est ajouté et le procédé est alors réitéré). Dans la plupart des cas, cet algorithme rend un résultat correct dans un temps plutôt rapide. Néanmoins, le critère de connexion entre deux fibres n'est pas fiable, et peut amener à de mauvais résultats sur certains exemples. De plus, le processus de subdivision du chemin peut parfois être long (comme par exemple quand le chemin doit passer entre deux points critiques proches), ou amener à des messages d'erreur du fait d'un manque de précision utilisée pour les calculs, qui est dû à un contrôle empirique pour détecter les anomalies. On peut aussi noter que du fait de cet empirisme, un calcul de complexité de cet algorithme est difficile. Pour résumer, le code n'est pas complètement fiable et nécessite parfois une intervention humaine pour terminer.

Exemple 25. *Soit* $F(x, y) = y^3 - (x^{10} - 2\,(10\,x - 1)^2)$. *Ce polynôme a comme propriété que deux de ses points critiques, dont une approximation peut être donné par* 0.09999929292 *et* 0.1000007071, *sont à une distance inférieure à*

10^{-5}. *L'appel de fonction* `algcurves[monodromy](F,x,y)` *effectué avec une précision de 10 chiffres retourne :*

```
Error, (in algcurves/Permuted) Computation inaccurate, use
larger value for Digits
```

Le même appel de fonction effectué avec une précision de 12 chiffres retourne le groupe de monodromie.

Exemple 26. *Soit* $F(x,y) = y^4 - 200\,y^2 + 40\,y - 2 - x$. *Ce polynôme a pour discriminant* $\Delta_F = 2557952 - 25574403072\,x - 5121536\,x^2 - 256\,x^3$, *dont une approximation des racines est donnée par :* $\alpha_1 = -10402.99505, \alpha_2 = -9603.005051, \alpha_3 = 0.0001000200060$. *Ce polynôme n'est pas intrinsèquement compliqué : la seule difficulté éventuelle est la différence de taille entre les racines du discriminant, puisque* $\frac{|\alpha_1|}{\alpha_3} > 10^8$. *Relativement à cette différence, la différence entre les points critiques* α_1 *et* α_2 *est donc modérément petite, puisqu'inférieure à* 10^3. *Néanmoins, comme nous le verrons à la fin de ce chapitre, il est possible de calculer le groupe de monodromie de* F *en utilisant 10 chiffres de précision, et ce tout au long des calculs. Mais la fonction* `algcurves[monodromy](F,x,y)` *de Maple nécessite 60 chiffres de précision pour que le programme se termine sans erreur.*

Pour résoudre les problèmes de connexion, Mark van Hoeij et Marc Rybowicz (communication personnelle) ont programmé un nouvel algorithme qui retourne un résultat certifié. La certification vient d'un théorème de B. T. Smith [Smi70], qui leur permet de calculer à chaque étape une borne sur la longueur du pas que l'on peut effectuer pour avoir une connexion fiable. La majorité des calculs sont effectués numériquement, mais ils utilisent aussi de l'arithmétique d'intervalle et de l'arithmétique rationnelle exacte pour pouvoir certifier le résultat. Cela donne une méthode fiable, mais qui souffre de temps de calcul longs, notamment dans le cas où deux points critiques sont proches, où leur algorithme nécessite de trop nombreux pas pour passer entre. De plus, le calcul de la longueur d'un pas permettant d'obtenir la certification du résultat est coûteux

Dans ces deux méthodes, la présence de trop nombreux pas intermédiaires est dûe à la méthode de connexion, et en ce qui concerne la commande `algcurves[monodromy]` de Maple, au fait que les auteurs utilisent des développements à l'ordre 1. Dans cette thèse, nous nous utiliserons des développements tronqués à un ordre supérieur, et nous détaillerons un compromis entre le nombre de pas à effectuer et les ordres de troncation.

3.2.2 Utilisation d'une équation différentielle

Il existe de nombreux algorithmes permettant de trouver, à partir du polynôme F, une équation différentielle dont les séries $\{S_i(x)\}$ sont solutions [Com64, CC86, CC87b, CSTU02, BCL+07]. Une telle équation différentielle donne accès à des algorithmes efficaces pour calculer les développements en série des $S_i(x)$ [vdH99, vdH01, BCL+07, Mez07]. Plus précisément, à l'aide de l'équation de récurrence définie par l'équation différentielle, il est possible de calculer les N premiers termes d'une série solution en un temps linéaire en N. De plus, plusieurs travaux ont été effectués sur le prolongement analytique de séries solutions d'une équation différentielle [CC87a, CC90, vdH99, Mez07]. Ainsi, on pourrait résoudre notre problème en calculant l'équation différentielle associée au polynôme F, puis utiliser ces outils.

Mais ces études sont basées sur l'hypothèse que l'on cherche une grande précision pour le résultat. Dans notre cas, nous n'avons pas besoin d'une grande précision sur les séries étudiées, mais uniquement d'un nombre de termes suffisant pour pouvoir connecter l'évaluation des séries à la fibre. De ce fait, en pratique, nous avons besoin de développements en série à des ordres qui ne sont pas suffisamment grands pour profiter de la bonne complexité asymptotique de ces algorithmes. Or, les algorithmes utilisant l'équation différentielle ont une bonne complexité en fonction du nombre de termes souhaité N, mais pas forcément en fonction des autres paramètres tels que d_x, d_y ou la taille des coefficients. Même s'il n'y a aucune difficulté théorique pour construire l'équation différentielle associée au polynôme F, la taille de l'équation différentielle, ainsi que le temps de calcul de cette équation différentielle, peuvent être importants. Enfin, étant donné que l'on connaît précisément le polynôme F, on peut calculer les fibres, et on est donc certain de ne pas s'éloigner des feuillets. Nous n'avons donc pas choisi de suivre cette approche.

3.3 Prolongement analytique

Notre méthode pour calculer le groupe de monodromie $\mathcal{M}$ est une méthode utilisant le processus de prolongement analytique. Nous commençons donc par décrire ce procédé.

Nous rappelons tout d'abord quelques résultats sur les séries solutions du polynôme F vu comme un polynôme univarié en y, avant de s'intéresser au point de vue analytique des séries de Puiseux, puis de décrire comment

connecter les séries solutions au-dessus de x_0 à la fibre en un point x_1 du disque de convergence des séries.

3.3.1 Principe du prolongement analytique

Nous considérons tout d'abord un point du plan complexe x_0 régulier. On rappelle le résultat suivant :

Théorème 18 (des fonctions implicites)**.** *Soit x_0 un point régulier de F, et $\mathcal{F}(x_0) = \{y_1, \ldots, y_{d_y}\}$ la fibre de F en ce point. Alors il existe d_y séries entières, holomorphes $S_i(x) = \sum_{k=0}^{\infty} \beta_{ik}(x - x_0)^k$, telles que :*
- *$F(x, S_i(x)) = 0$ dans un voisinage de x_0,*
- *$S_i(0) = y_i$.*

De plus, le rayon de convergence de ces séries est au moins égal à $\delta(x_0)$, la distance entre x_0 et son plus proche point critique.

Nous rappelons que ces séries peuvent être calculées « rapidement » à partir de la fibre en utilisant l'algorithme `Newton-quadratique` [KT78, vzGG99] (voir la section 2.1.1 du chapitre 2)

Ce théorème nous permet de décrire le procédé de **prolongement analytique**. Notons

$$S(x) = \sum_{k=0}^{\infty} \beta_k \, (x - x_0)^k \tag{3.1}$$

l'une des d_y séries définies par le théorème 18, et soit $x_1 \in B_0 = D(x_0, \delta(x_0))$. Alors on peut exprimer $S(x)$ comme une série en $(x - x_1)$.

$$S(x) = \sum_{k=0}^{\infty} \beta'_k \, (x - x_1)^k \tag{3.2}$$

en développant les puissances $(x - x_0)^k = (x - x_1 + x_1 - x_0)^k \, , k \in \mathbb{N}$ et en réordonnant la série (3.1) en puissances de $(x - x_1)$. Le rayon de convergence de la série (3.2) vérifie trivialement $\delta(x_1) \geq \delta(x_0) - |x_1 - x_0| > 0$. Si cette inégalité est stricte, alors le disque de convergence $B_1 = D(x_1, \delta(x_1))$ s'étend en dehors de B_0, et l'on peut réitérer ce processus en un point x_2 qui appartient à B_2 mais pas à B_1 (voir la figure 3.3).

Ce processus s'appelle le prolongement analytique de la série (3.1).

Proposition 31. *Soit γ un chemin dans le plan complexe, de point de départ a et de point d'arrivée b, qui ne passe par aucun point critique de F. Alors les d_y séries $S_1, \ldots, S_{d_y}$ au dessus de a décrites dans le théorème 18 peuvent être prolongées analytiquement le long du chemin γ.*

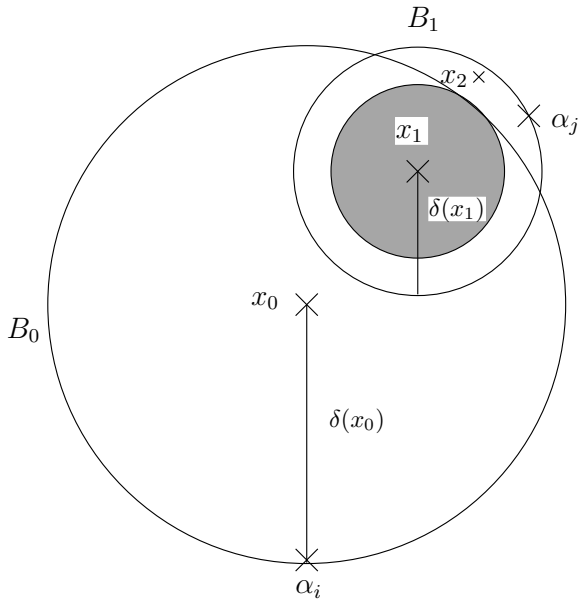

FIG. 3.3 – Une étape du prolongement analytique.

3.3.2 Séries de Puiseux : point de vue analytique

Si x_0 est un point critique, alors les séries solutions du polynôme F sont des séries de Puiseux

$$S_{ij}(x) = \sum_{k=0}^{\infty} \beta_{ik}\, \zeta_{e_i}^{jk}\, (x - x_0)^{\frac{k}{e_i}} \tag{3.3}$$

avec $0 \leq j \leq e_i - 1$ et $1 \leq i \leq s$ (voir le théorème 2 du chapitre 1). Nous allons maintenant rappeler quelques résultats d'analyse concernant ces séries. Pour cela, pour tout entier positif e, on choisit une détermination pour la fonction racine e-ième, que l'on note $\sqrt[e]{x}$. Plus précisément, cette détermination est caractérisée par un angle $\theta \in [-\pi, \pi[$ tel que $\sqrt[e]{z} = |z|^{1/e} \exp\left(I\, (\arg_\theta z)/e\right)$ avec $\arg_\theta z \in]\theta, \theta + 2\,\pi]$ et I le nombre complexe $I^2 = -1$. La ligne de discontinuité de la fonction est alors la demi-droite d'origine 0 et qui forme un angle θ avec l'axe réel positif. L'expression $x^{\frac{k}{e}}$ correspond alors à la fonction $\sqrt[e]{X^k}$ et les séries (3.3) définissent d_y fonctions dans un disque ouvert.

Lemme 19. *Le rayon de convergence des séries de Puiseux en un point x_0 est au moins égal à $\delta(x_0)$.*

Démonstration. voir [Mar67]. □

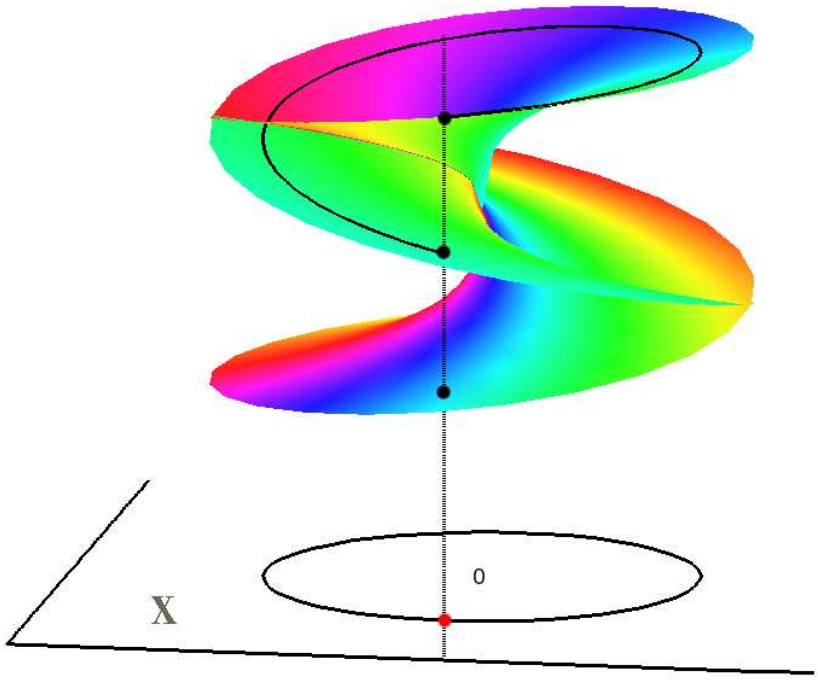

FIG. 3.4 – Prolongement analytique de F_1 autour du point critique 0

Soit maintenant B une demi-droite dans le plan complexe d'origine x_0 et caractérisée par l'angle $\theta \in [-\pi, \pi[$ formé avec l'axe réel positif, que l'on notera $B = (x_0, \theta)$. On choisit alors une détermination pour les fonctions racine e_i-ième de telle sorte que toutes les séries S_{ij} correspondantes admettent B comme ligne de discontinuité. Les fonctions S_{ij} sont alors analytiques dans le domaine simplement connexe $D(x_0, \delta(x_0))\backslash B$.

Exemple 27. *Considérons l'exemple 3, page 28 du chapitre 1, ainsi que la figure 3.4, qui donne une représentation de la partie réelle de la courbe algébrique plane associée à F_1 dans un voisinage de 0. Sur cette figure, la couleur représente la partie imaginaire, et les intersections illustrées ici n'en sont pas, puisque les parties imaginaires diffèrent en ces points. Soit $x_1 \neq 0$ un point dans ce voisinage 0 (le point rouge dans le plan complexe sur le cercle). Comme x_1 est un point régulier (le seul point critique fini de F_1*

est 0), la fibre en x_1 de F_1 est constituée de trois points distincts (les 3 points noirs sur la courbe). Comme l'illustre la figure 3.4, si l'on prolonge analytiquement une des séries $S_{1j}(x) \in \mathbb{Q}[[(x-x_1)]]$ solution de F_1 le long du cercle centré en 0 et passant par x_1, alors après un tour, la série prolongée $\tilde{S}_{1j}(x)$ vérifie $\tilde{S}_{1j}(x_1) \neq S_{1j}(x_1)$: en réitérant ce procédé pour chaque point de la fibre $\mathcal{F}(x_1)$, on s'aperçoit que faire un tour autour du point critique 0 engendre une permutation sur les points de la fibre.

Notre stratégie de prolongement analytique repose sur deux résultats que nous allons maintenant détailler.

Proposition 32. *Si r est un nombre réel positif plus petit que $\delta(x_0)$, alors le chemin $\gamma(t) = x_0 + r\exp((\theta + 2\pi t)I)$ décrit un cercle $C(c_0, r)$ autour du point x_0 quand t varie de 0 à 1. Pour tout $1 \leq i \leq s$, l'ensemble $\{S_{ij}(\gamma(0)\}_{0\leq j\leq e_i-1}$ est un sous-ensemble de $\mathcal{F}(\gamma(0))$. Le prolongement analytique le long du chemin γ de $S_{ij}(x)$, $0 \leq j \leq e_i - 1$ retourne $S_{i\overline{j+1}}(\gamma(0))$, où $\overline{l} = l$ si $1 \leq l \leq e_i - 1$ et $\overline{e_i} = 1$.*

En d'autres termes, le prolongement analytique le long de γ permute cycliquement les valeurs $\{S_{ij}(\gamma(0)\}_{0\leq j\leq e_i-1}$. Ainsi, les développements de Puiseux nous permettent de calculer l'action sur la fibre $\mathcal{F}(x_1)$ pour tout point $x_1 \in D(x_0, \delta(x_0))$. Cette action est la **monodromie locale** en x_0. De plus, le type de permutation engendrée par l'action locale est donnée par les indices de ramification des développements de Puiseux au-dessus de x_0.

Exemple 28. *Considérons à nouveau l'exemple 4, page 28 du chapitre 1 et regardons la permutation engendrée sur la fibre $\mathcal{F}(x_1)$, où x_1 est un point proche de 0, par le prolongement analytique le long du cercle centré en 0 et passant par x_0. La figure 3.5 illustre ce phénomène (la représentation de la courbe étant la même que pour la figure 3.4). On voit bien ici que les 3 séries conjuguées d'indice de ramification 3 engendrent un 3-cycle sur la fibre $\mathcal{F}(x_0)$. De même, les deux séries conjuguées d'indice de ramification 2 engendrent un 2-cycle, et la dernière série, d'indice de ramification 1, engendre un 1-cycle. Ainsi, la monodromie locale est une permutation de type $[3, 2, 1]$, correspondant aux indices de ramification $e_1 = 3, e_2 = 2$ et $e_3 = 1$.*

Proposition 33. *Tout prolongement analytique suivant un chemin inclut dans $D(x_0, \delta(x_0))\backslash B$ peut être calculé en évaluant les séries de Puiseux S_{ij} le long de ce chemin.*

Démonstration. Les séries S_{ij} sont analytiques dans $D(x_0, \delta(x_0))\backslash B$. □

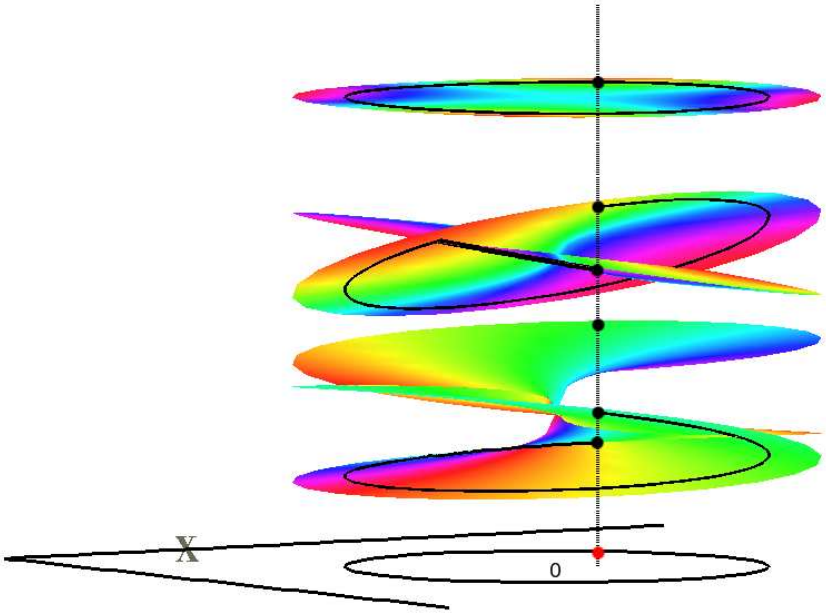

FIG. 3.5 – Prolongement analytique de F_2 autour du point critique 0

3.3.3 Ordres de troncation

Proposition 34. *Soit x_0 un point du plan complexe et*

- $S(x) = \sum_{k=0}^{\infty} \mu_k (x - x_0)^{\frac{k}{e}}$ *une série de Puiseux au-dessus de x_0,*
- $\widetilde{S}^N(x) = \sum_{k=0}^{N} \mu_k (x - x_0)^{\frac{k}{e}}$ *sa troncation à l'ordre N,*
- ρ *le rayon de convergence de $S(x)$,*
- $x_1 \in D(x_0, \rho)$,
- $r \in \mathbb{R}$ *tel que* $|x_1 - x_0| < r < \rho$,
- M *une borne supérieure pour* $\sup_{x \in D(x_0, r)} |S(x)|$,
- $\eta \in \mathbb{R}^{+*}$,
- $\beta = \left(\frac{|x_1 - x_0|}{r}\right)^{\frac{1}{e}}$.

Alors, $N \geq \dfrac{\ln\left(\frac{\eta}{M}\right) + \ln(1-\beta)}{\ln(\beta)} - 1 \Rightarrow \left|S(x_1) - \widetilde{S}^N(x_1)\right| \leq \eta.$

Démonstration. Traitons d'abord le cas non ramifié, c'est-à-dire $e = 1$.

$$\begin{aligned}\left|S(x_1) - \sum_{k=0}^{N} \mu_k (x_1 - x_0)^k\right| &= \left|\sum_{k=N+1}^{\infty} \mu_k (x_1 - x_0)^k\right| \\ &\leq \sum_{k=N+1}^{\infty} |\mu_k| \, |x_1 - x_0|^k\end{aligned}$$

En utilisant la formule de Cauchy, on a :

$$\begin{aligned}|\mu_k| &= \left|\frac{1}{2i\pi} \int_{\|\omega - x_0\| = r} \frac{S(\omega)}{(\omega - x_0)^{k+1}} \, \mathrm{d}\omega\right| \\ &= \frac{1}{2\pi} \left|\int_0^{2\pi} \frac{S(x_0 + re^{i\theta})}{r^{k+1} e^{i(k+1)\theta}} rie^{i\theta} \mathrm{d}\theta\right| \\ &\leq \frac{1}{2\pi} \int_0^{2\pi} \frac{|S(x_0 + re^{i\theta})|}{r^k} \, \mathrm{d}\theta \\ &\leq \frac{M}{r^k}\end{aligned}$$

Finalement, on a :

$$\begin{aligned}\left|S(x_1) - \sum_{k=0}^{N} \mu_k (x_1 - x_0)^k\right| &\leq \sum_{k=N+1}^{\infty} M\beta^k \\ &= \frac{M}{1-\beta} \, \beta^{N+1}\end{aligned}$$

$$\text{Or, } \frac{M}{1-\beta} \, \beta^{N+1} \leq \eta \quad \Longleftrightarrow \quad N \geq \frac{\ln\left(\frac{\eta(1-\beta)}{M}\right)}{\ln(\beta)} - 1 \quad \text{puisque } \beta < 1$$

Supposons maintenant que x_0 soit un point critique, et que la série $S(x) = \sum_{k=0}^{\infty} \mu_k (x - x_0)^{\frac{k}{e}}$ ait un indice de ramification $e > 1$. Si l'on pose $G(x, y) = F(x_0 + x^e, y)$, alors la série $S_1(x) = \sum_{k=0}^{\infty} \mu_k x^k$ est une série de Puiseux de la courbe algébrique plane définie par le polynôme G, et son indice de ramification est égal à 1. Comme $|x| < \rho$ si et seulement si $|x|^{1/e} < \rho^{1/e}$, le rayon de convergence de la série S_1 est au moins égal à $\rho^{1/e}$. De ce fait, en posant $x_2 = (x_1 - x_0)^{1/e} \in D(0, \rho^{1/e})$, on peut utiliser le résultat démontré dans le cas $e = 1$ pour la série S_1. On a donc :

$$N \geq \frac{\ln\left(\frac{\eta}{M}\right) + \ln(1-\beta)}{\ln(\beta)} - 1 \Rightarrow \left|S_1(x_2) - \widetilde{S_1}^{\,N}(x_2)\right| \leq \eta$$

et l'égalité $S_1(x_2) = S(x_1)$ nous permet de conclure. □

Remarque 18. *Pour trouver l'ordre de troncation nécessaire N, il nous faut estimer la borne M. On calculera une borne pour les valeurs des séries $|S(x)|$ dans le disque $D(x_0, \delta(x_0))$ en utilisant les bornes données dans [Mig92, page 170] sur les racines de polynômes univariés.*

3.3.4 Connecter les séries aux fibres

Soit x_0 un point du plan complexe et notons $\{S_1(x), \ldots, S_{d_y}(x)\}$ les d_y séries de Puiseux définies au-dessus de x_0. Dans le cas où x_0 est un point critique, notons B la ligne de discontinuité choisie pour ces séries. Soit alors $x_1 \in D(x_0, \delta(x_0))\backslash B$, et notons $\mathcal{F}(x_1) = \{y_1, \ldots, y_{d_y}\}$ la fibre au-dessus de x_1. Le but de cette section est, pour tout $1 \leq i \leq d_y$, de trouver l'indice j tel que $S_i(x_1) = y_j$. Cela définit une permutation σ telle que $S_i(x_1) = y_{\sigma(i)}$ pour tout $1 \leq i \leq d_y$. La stratégie présentée dans la section 3.5.1 utilisera ce moyen de connexion.

Pour obtenir la permutation σ, nous utiliserons un algorithme de résolution numérique d'équation qui, étant donné un polynôme univarié, rend des approximations numériques $\tilde{y_i}$ des racines y_i du polynôme, ainsi que des nombres $\rho_i \in \mathbb{R}^{+\star}$ tels que :

1. Les disques $D(\tilde{y_i}, \rho_i)$ ne s'intersectent pas, et chaque disque contient exactement une racine du polynôme.
2. Si ϵ est la plus petite distance entre deux $\tilde{y_i}$ distincts, alors $\epsilon/2 - r > 0$, où $r = max\{\rho_1, \ldots, \rho_{d_y}\}$.

Ce sujet a été fortement traité dans la littérature [Pan97]. Nous proposons maintenant une approche possible, basée sur un théorème de B. T. Smith [Smi70], que nous commençons par rappeler. Il est entendu que d'autres approches sont possibles et que nous n'avons pas étudié ce point de manière exhaustive.

Théorème 19. *Soit $P(z)$ un polynôme de degré N à coefficients dans le corps $\mathbb{C}$, et supposons que N racines complexes distinctes $z_1, \ldots, z_N$ soient données. On définit alors*

$$\rho_k = N\,|P(z_k)|\,/\prod_{\substack{i=1\\ i\neq k}}^{N} |z_i - z_k|.$$

Alors l'union des disques $D(z_k, \rho_k)$ contient l'ensemble des racines de $P(z)$. De plus, chaque composante connexe constituée de K cercles contient exactement K racines de $P(z)$.

Démonstration. voir [Smi70] □

L'utilisation que nous faisons de ces résultats sont basés sur l'hypothèse que nous avons à notre disposition un algorithme numérique qui calcule les racines d'un polynôme à une variable, et que cet algorithme converge quand on augmente la précision utilisée pour les calculs numériques (voir par exemple [Pan97] pour plus d'informations sur ce sujet). En effet, avec une telle hypothèse et le théorème 19, nous obtenons l'algorithme suivant :

1. Fixer arbitrairement une petite précision.
2. Calculer les approximations $z_1, \ldots, z_N$ des racines de P. Si certaines approximations coïncident, augmenter la précision, et répéter cette étape tant que les approximations ne sont pas toutes différentes.
3. Calculer les ρ_k du théorème 19. Si les disques ainsi construits s'intersectent, augmenter la précision et retourner au point 2.

Si l'on suppose que l'algorithme de résolution satisfait notre hypothèse, alors ce procédé converge. De plus, les ρ_i peuvent être aussi petits que l'on veut. De ce fait, la dernière étape peut être répétée jusqu'à ce que la quantité $\epsilon/2 - r$ soit positive.

On a alors le résultat suivant (illustré par la figure 3.6) :

Proposition 35. *Si N_i est un entier tel que $|S_i(x_1) - \widetilde{S}_i^{N_i}(x_1)| < \epsilon/2 - r$, alors σ est caractérisé par :*

$$|\widetilde{S}_i^{N_i}(x_1) - \tilde{y}_{\sigma(i)}| = \min_{1 \leq j \leq d_y} \{|\widetilde{S}_i^{N_i}(x_1) - \tilde{y}_j|\}$$

Démonstration. On a $S_i(x_1) = y_{\sigma(i)}$, et donc $|\widetilde{S}_i^{N_i}(x_1) - y_{\sigma(i)}| < \frac{\epsilon}{2} - r$. Ainsi, $|\widetilde{S}_i^{N_i}(x_1) - \tilde{y}_{\sigma(i)}| \leq |\widetilde{S}_i^{N_i}(x_1) - y_{\sigma(i)}| + |y_{\sigma(i)} - \tilde{y}_{\sigma(i)}| < \epsilon/2$. Comme ϵ est la distance minimum entre deux racines approchées $\tilde{y}_i$ et $\tilde{y}_j$, on a $|\widetilde{S}_i^{N_i}(x_1) - \tilde{y}_k| > \epsilon/2$ pour tout $k \neq \sigma(i)$, et la proposition suit. □

Ainsi, en supposant que l'on connaisse les indices de ramifications des séries définies au-dessus de x_0, on peut connecter les séries aux fibres de la façon suivante :

1. Calculer une approximation de la fibre $\mathcal{F}(x_1)$, ainsi qu'une borne r sur l'erreur de cette approximation, à l'aide d'un algorithme tel que décrit ci-dessus.

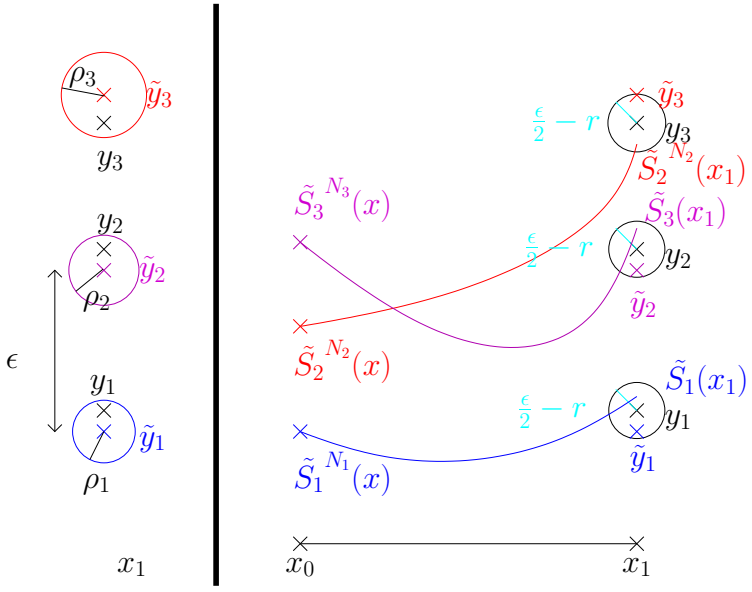

FIG. 3.6 – Relier les séries en x_0 à la fibre $\mathcal{F}(x_1)$

2. Calculer une borne M pour la valeur des séries $|S_i(x)|$ dans le disque $D(x_0, \delta(x_0))$ [Mig92, page 170]
3. Appliquer la proposition 34 avec une précision $\eta = \epsilon/2 - r$ pour obtenir des ordres de troncation N_i tels que $|S_i(x_1) - \widetilde{S}_i^{N_i}(x_1)| < \epsilon/2 - r$.
4. Évaluer les séries tronquées $\widetilde{S}_i^{N_i}(x)$ en x_1, puis connecter chaque série à l'approximation de la fibre la plus proche.

3.4 Choix de bons chemins pour le calcul de la monodromie

Nous commençons par considérer la figure 3.7. Ici, nous considérons donc une polynôme ayant deux points critiques finis α_1 et α_2. La partie gauche de la figure montre les chemins γ_i classiques : pour chacun de ces lacets γ_i, on part du point de base a, on va jusqu'au point x_i, on contourne le point critique α_i en suivant le cercle centré en α_i et de rayon $|\alpha_i - x_i|$, puis l'on retourne de x_i au point de base α_i. Supposons maintenant que nous ayons déjà calculé le prolongement analytique associé au chemin γ_2. Il parait alors plus intéressant d'utiliser le travail effectué de a au point x_0 lors de ce calcul pour

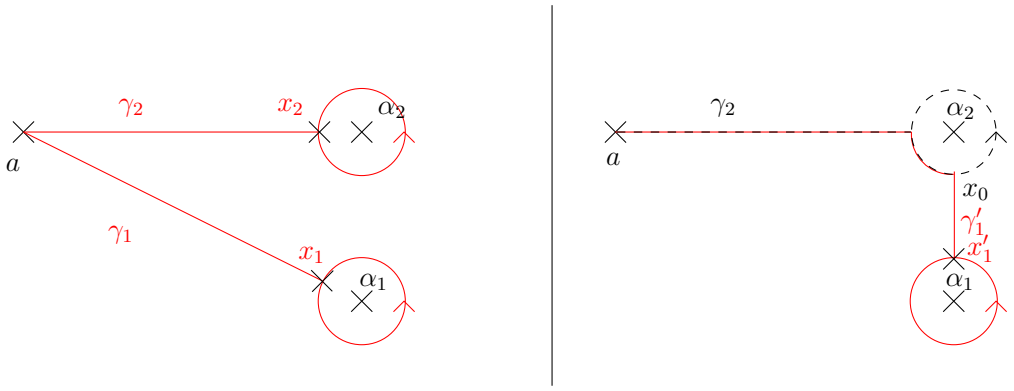

FIG. 3.7 – Choix possibles pour les chemins γ_i

traiter le point critique α_1. Nous n'avons ainsi qu'à calculer le prolongement analytique entre x_0 et x'_1 et le long du cercle centré en α_1. Avec un tel chemin γ'_1, on minimise donc la longueur totale du chemin à parcourir.

De manière générale, si l'on souhaite minimiser la longueur totale des chemins au-dessus desquels on doit effectuer le processus de prolongement analytique, il nous faut suivre des chemins passant le long d'un arbre de recouvrement minimum pour la distance euclidienne de l'ensemble de points $\mathcal{V} = \{a, \alpha_1, \ldots, \alpha_n\}$. Cette idée nous a été suggéré par Mark van Hoeij. Bien que naturelle, elle nécessite néanmoins un travail non négligeable pour trouver des chemins suivants cet arbre qui soit homotopes aux γ_i. Dans cette section, nous allons proposer deux méthodes pour trouver de tels chemins.

3.4.1 Arbre de recouvrement minimum pour la distance euclidienne

Nous commençons par décrire la façon de calculer cet arbre de recouvrement minimum. Nous décrivons également quelques propriétés intéressantes d'un arbre de recouvrement minimum pour la distance euclidienne, que nous utiliserons régulièrement par la suite.

Calcul d'un arbre de recouvrement minimal

Si l'on considère le graphe complet $\mathcal{G}$ ayant pour sommets $\mathcal{V} = \{\alpha_0 = a, \alpha_1, \ldots, \alpha_n\}$, alors les algorithmes classiques de théorie des graphes tels que l'algorithme de Prim ou l'algorithme de Kruskal [Gan06] nous permettent

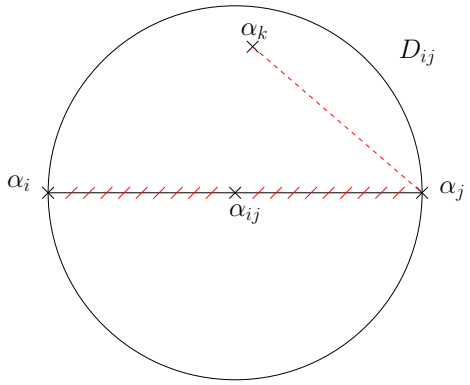

FIG. 3.8 – Propriété de l'arbre de recouvrement minimal pour la distance

de calculer un arbre de recouvrement minimal pour la distance euclidienne de l'ensemble $\mathcal{V}$, que nous noterons $\mathcal{T}$. La complexité de ces algorithmes dépendant du nombre d'arêtes du graphe $\mathcal{G}$, il est préférable de calculer préalablement une triangulation de Delaunay de l'ensemble $\mathcal{V}$, puis de trouver un arbre de recouvrement minimum pour l'ensemble ainsi obtenu. La triangulation de Delaunay peut être calculée avec une complexité en $O(n \log n)$. Ensuite, les algorithmes de Prim ou Kruskal ont une complexité elle aussi en $O(n \log n)$, puisque le nombre d'arêtes d'une triangulation complète est inférieur à $3n$.

Remarque 19. *Il se peut également que l'on obtienne des résultats intéressants en utilisant un arbre de Steiner [Gan06], mais ce dernier semble bien plus difficile à obtenir, étant donné qu'il n'existe pas à l'heure actuelle d'algorithme ayant une complexité polynomiale.*

Propriétés d'un arbre de recouvrement minimum pour la distance euclidienne

Dans la suite de ce chapitre, nous utiliserons les deux lemmes suivants :

Lemme 20. *Soit $[\alpha_i, \alpha_j]$ une arête de l'arbre $\mathcal{T}$, $r = |\alpha_i - \alpha_j|$ et $\alpha_{ij} = \frac{\alpha_i + \alpha_j}{2}$ le milieu de l'arête $[\alpha_i, \alpha_j]$. Alors il n'y a aucun point critique dans le disque ouvert $D_{ij} = D(\alpha_{ij}, \frac{r}{2})$.*

Démonstration. La figure 3.8 donne une illustration géométrique de la situation. Supposons que $\alpha_k \in D_{ij}$. Si l'on enlève l'arête $[\alpha_i, \alpha_j]$ à $\mathcal{T}$, on obtient deux composantes connexes. Le sommet α_k appartient à l'une d'entre elle,

par exemple celle de α_i. L'arbre $\mathcal{T}_2$, identique à $\mathcal{T}$ excepté l'arête $[\alpha_i, \alpha_j]$ remplacée par l'arête $[\alpha_k, \alpha_j]$, a un poids inférieur à celui de $\mathcal{T}$. □

Lemme 21. *L'angle entre deux arêtes de l'arbre $\mathcal{T}$ ayant un sommet commun est supérieur ou égal à $\frac{\pi}{3}$.*

Démonstration. Notons $[\alpha_i, \alpha_j]$ et $[\alpha_j, \alpha_k]$ les deux arêtes considérées. Alors, si l'on considère le triangle $\alpha_i\alpha_j\alpha_k$, comme $\mathcal{T}$ est un arbre de recouvrement minimum pour la distance, le côté $[\alpha_i, \alpha_k]$ est le côté le plus long des trois côtés du triangle. Cela revient à dire que l'angle entre les deux arêtes $[\alpha_i, \alpha_j]$ et $[\alpha_j, \alpha_k]$ est le plus grand des trois angles du triangle, et est donc supérieur ou égal à $\frac{\pi}{3}$. □

3.4.2 Chemins dans l'arbre $\mathcal{T}$

Le lemme 20 nous assure que le disque D_{ij} a une intersection non vide avec les disques $D(\alpha_i, \delta(\alpha_i))$ et $D(\alpha_j, \delta(\alpha_j))$. Ainsi, on peut choisir des points intermédiaires x_{ij} et x_{ji} tels que (voir la figure 3.9) :

- x_{ij} et x_{ji} appartiennent à l'arête $[\alpha_i, \alpha_j]$,
- $x_{ij} \in D_{ij} \cap D(\alpha_i, \delta(\alpha_i))$,
- $x_{ji} \in D_{ij} \cap D(\alpha_j, \delta(\alpha_j))$,

On peut supposer de plus que, pour le point critique α_i, tous les points $(x_{ij})_j$ ainsi créés sont à la même distance de α_i. Ainsi, pour chaque couple d'indices (k, j) tels que les arêtes $[\alpha_k, \alpha_i]$ et $[\alpha_i, \alpha_j]$ existent dans $\mathcal{T}$, il existe deux arcs de cercle $A(x_{ik}, x_{ij}, +)$ et $A(x_{ik}, x_{ij}, -)$ centrés en α_i, et allant de x_{ik} à x_{ij} respectivement dans le sens trigonométrique et dans le sens non trigonométrique.

Définition 26. *Étant donné deux points du plan complexe x_0 et x_1 appartenant à $\mathcal{T}$, on appellera* ***chemin dans l'arbre*** *tout chemin constitué d'une succession d'arcs de cercle $A(x_{ik}, x_{ij}, \pm)$ et d'arêtes $[x_{ij}, x_{ji}]$, ayant pour point de départ x_0 et pour point d'arrivée x_1.*

Exemple 29. *Si l'on considère la figure 3.9, la partie représentée par un trait plein est un chemin dans l'arbre égal à $A(x_{ik}, x_{ij}, +) \cdot [x_{ij}, x_{ji}] \cdot A(x_{ji}, x_{jl}, -)$.*

Il est ainsi possible, pour chaque point critique α_l, de construire un chemin dans l'arbre joignant a à l'un des points x_{lh} proche du point critique α_l. Notre but est de trouver un tel chemin dans l'arbre δ_l tel que, si β_l est le lacet ayant pour point de base x_{lh} et suivant le cercle centré en α_l dans le sens trigonométrique, alors le lacet $\gamma'_l = \delta_l^{-1} \cdot \beta_l \cdot \delta_l$ est homotope dans X_0 au chemin γ_l (voir la figure 3.1).

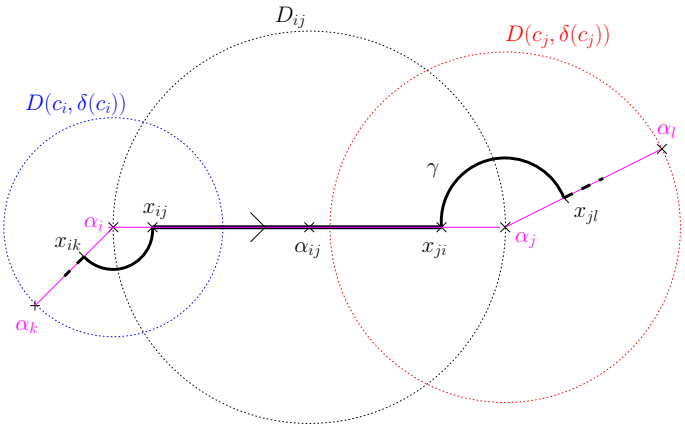

FIG. 3.9 – Disques de convergence et points intermédiaires

Exemple 30. *Considérons* $K = \mathbb{Q}$ *et* $F(x, y) = (x-1)\,y^3 - 2\,x^3\,y^2 + x\,y + 1$. *La figure 3.10 montre un tel chemin* γ_5' *(sur cette illustration, deux points critiques n'influant pas sur le chemin* γ_5 *ne figurent pas pour plus de clarté).*

3.4.3 Chemin dans l'arbre homotope à γ_l : méthode 1

Nous allons maintenant décrire une méthode permettant de construire des chemins dans l'arbre, comme définis précédemment, homotopes aux γ_l de la figure 3.1. Cette méthode 1 est un travail non publié en commun avec Marc Rybowicz. La deuxième méthode, présentée dans la section suivante est autonome.

Les principes sont les suivants :

1. Nous effectuons un parcours en profondeur de l'arbre (« depth-first search ») à partir du point de base a. Pour chaque sommet, ses voisins sont toujours traités dans un ordre trigonométrique, à partir de l'arête par laquelle on arrive au sommet. En conservant les arêtes traversées et en contournant les point critiques suivant des arcs toujours orientés dans le sens trigonométrique, nous pouvons construire des chemins dans l'arbre θ_l de sorte que les $\tau_l = \theta_l^{-1}\beta_l\theta_l$ engendrent le groupe $\Pi_1(X_0, a)$ (voir la figure 3.11), mais ne soient pas forcément homotopes aux γ_l dans X_0.
2. En nous inspirant d'une des preuves classiques du théorème 6, nous exprimons les τ_l en fonction des γ_l.

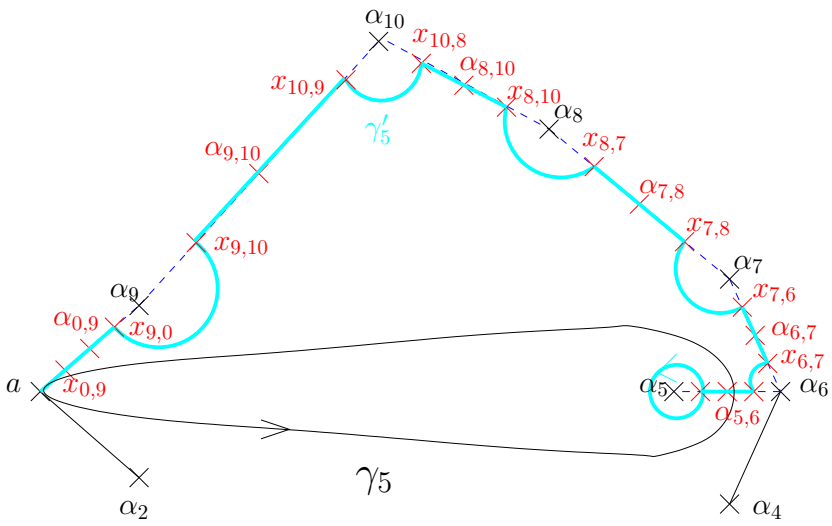

FIG. 3.10 – Chemin homotope à γ_5 suivant l'arbre $\mathcal{T}$

3. Enfin, si nécessaire, on peut exprimer les γ_l en fonction des τ_l en utilisant des méthodes purement algébriques. Pour cela, bien que des algorithmes sur les groupes libres permettant d'effectuer ce travail existent, nous proposerons une méthode ad hoc dont la complexité est plus facile à maîtriser.

D'autres solutions sont possibles, mais celle-ci présente les avantages suivants :

- Elle est relativement simple à expliquer, puisque scindable en trois parties complètement indépendantes.
- Les algorithmes de chacune des trois parties sont basés sur des techniques bien connues et ne nécessitent pas de preuves particulières.
- Il est clair que si l'on ne s'intéresse qu'à des générateurs quelconques de $\Pi_1(X_0, a)$, les deux dernières étapes ne sont pas nécessaires : un seul parcours de l'arbre et le tri des voisins de chaque sommet suffisent.
- Nous pouvons facilement obtenir certaines informations sur sa complexité.

Construction de générateurs de $\Pi_1(X_0, a)$

Nous formulons classiquement l'algorithme de parcours en profondeur de $\mathcal{T}$ à l'aide d'une pile qui contiendra des sommets de l'arbre. Une seconde pile va nous permettre de mémoriser les chemins θ_l en construction : à chaque

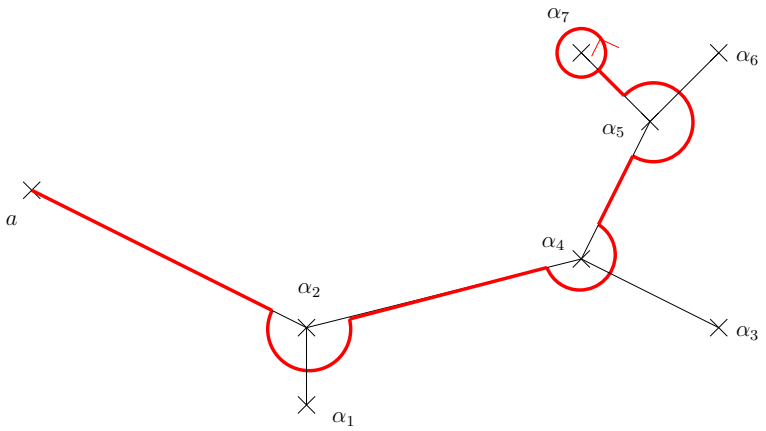

FIG. 3.11 – $\tau_7 = \theta_7^{-1}\beta_7\theta_7$

étape, elle contient la suite d'arêtes et d'arcs qui a permis d'arriver au point courant de l'arbre.

Les primitives sur les piles que nous utilisons sont :
- `PileVide`() : créé une nouvelle pile vide,
- `NonVide`(P) : renvoie $vrai$ si la pile est non vide, $faux$ sinon,
- `Sommet`(P) : renvoie l'élément au sommet de la pile P, sans le dépiler,
- `Empiler`(x,P) : empile l'objet x sur la pile P,
- `Dépiler`(P) : supprime l'objet se trouvant au sommet de la pile,
- `Copie`(P) : duplique le contenu de la pile P.

Nous considérons $\mathcal{T}$ comme un arbre enraciné en le point de base $a = \alpha_0$ et nous identifions chaque sommet α_i à son indice i. Nous supposons que l'arbre est donné classiquement par une table de voisins, c'est-à-dire par un tableau T indicé de 0 à n tel que $T[l]$ soit une pile (ou liste chaînée) contenant les voisins dans $\mathcal{T}$ du point critique l, ordonnés comme suit : le premier voisin est le père de l dans $\mathcal{T}$ et les autres voisins sont ordonnés dans le sens trigonométrique à partir de l'arête entre l et son père.

Le point critique 0 nécessite un traitement particulier. D'une part, il n'a pas de père : ses fils sont ordonnés à partir de n'importe quelle demi-droite se terminant en α_0. D'autre part, on n'a jamais à contourner ce sommet ni à l'inclure à l'intérieur d'un lacet.

Enfin, les arcs et portions d'arêtes à suivre sont exprimés en fonction des points x_{ij}. Mais aucun calcul n'est effectué sur ces points et l'on pourrait se contenter de conserver uniquement les indices.

```
Générateurs(T, n)
Entrées :
    T : Un tableau comme ci-dessus.
    n : Le nombre de points critiques.
Sortie : Un tableau C tel que C[l] contienne le chemin dans l'arbre θ_l^{-1}
         pour 1 ≤ l ≤ n, représenté sous forme d'une pile de segments et
         d'arcs.
Début
  PS ← PileVide()          // Pile des sommets (points critiques)
  PC ← PileVide()                                  // Pile des chemins
  Pour l de 1 à n faire
    Q[l] ← Sommet(T[l])      // Initialement, Q[l] est le père de l
    Dépiler(T[l])   // Le prochain arc autour de l débute en Q[l]
  Fin
  Empiler(0, PS)
  Tant que NonVide(PS) faire
    l ← Sommet(PS)
    Si NonVide(T[l]) alors      // Il reste des voisins à traiter
      v ← Sommet(T[l])
      Si l ≠ 0 alors
        Empiler(A(x_{lQ[l]}, x_{lv}, +), PC)      // On tourne autour de l
        Empiler([x_{lv}, x_{vl}], PC)             // puis on avance vers v
        Q[l] ← v                        // Le prochain arc débute en v
      Sinon
        Empiler([α_0, x_{v0}], PC)
      Fin
      Empiler(v, PS)
      Dépiler(T[l])              // On supprime v des voisins de l
    Sinon
      Tant que Sommet(PC) est un arc faire
        Dépiler(PC)              // Dépiler les arcs autour de l
      Fin
      Si l ≠ 0 alors
        C[l] ← Copie(PC)
        Dépiler(PC)        // Dépiler l'arête ayant conduit à l
      Fin
      Dépiler(PS)
    Fin
  Retourner(C)
Fin.
```

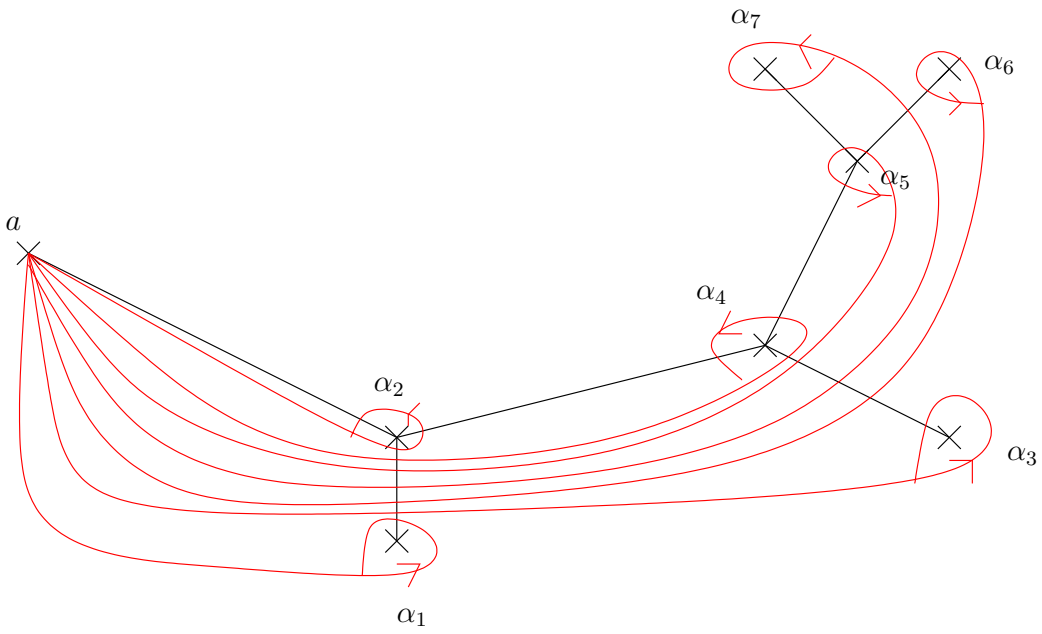

FIG. 3.12 – Un exemple des chemins τ_i

Théorème 20. *Cet algorithme renvoie le résultat attendu.*

Démonstration. Indication : Si nous supprimons les instructions relatives à la pile PC, nous obtenons l'algorithme classique de parcours en profondeur d'un arbre. Il se termine et tous les sommets sont traités. Les instructions relatives à PC ne servent qu'à mémoriser le suivi des arêtes et le contournement des points critiques par les arcs. □

Remarque 20. *On pourrait aussi combiner le tri des voisins avec ce parcours. Nous avons choisi de le découpler afin d'être plus clair.*

Théorème 21. *Les lacets $\tau_l = \theta_l^{-1}\beta_l\theta_l$, pour $1 \leq l \leq n$, engendrent $\Pi(X_0, a)$.*

Démonstration. Comme l'illustre la figure 3.12, les lacets τ_l sont homotopes à des lacets qui ne s'intersectent pas et chaque τ_l «entoure une fois» le point critique α_l. Le résultat est alors mentionné dans [LZ04]. Il peut se démontrer à l'aide du théorème de Seifert et Van Kampen. □

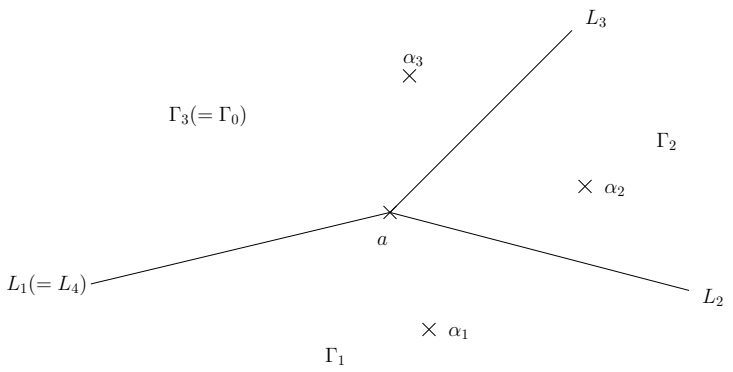

FIG. 3.13 – Découper le plan selon les points critiques

Expression des τ_l en fonction des γ_l

Nous proposons pour cette tâche d'utiliser une version effective de la preuve du théorème 6, telle que décrite par exemple dans [Vol97] (théorème 4.27).

D'après notre hypothèse sur le point a, il existe n demi-droites issues de a, notées $\{L_j\}_{1\leq j\leq n}$, telles que le cône Γ_j défini par a, L_j et L_{j+1} contienne un unique point critique α_j dans son intérieur (on pose $L_{n+1} = L_1$). La situation est illustrée par la figure 3.13. On rappelle que par hypothèse, il existe au moins deux points critiques.

Soit maintenant τ_l le lacet dans l'arbre de base $a = \alpha_0$ et entourant le point critique α_l. Lorsqu'on parcourt ce lacet en respectant son orientation, on rencontre les L_j ; nous notons $\{b_i\}_{1\leq i\leq h}$ les points d'intersection successifs (hormis a) et nous posons $b_0 = b_{h+1} = a$. On désignera par $L_{c(i)}$ l'unique demi-droite contenant b_i. Notons μ_i $(0 \leq i \leq h)$ la portion (orientée) du lacet τ_l entre b_i et b_{i+1}, de sorte que :

$$\tau_l = \mu_h \cdot \mu_{h-1} \cdots \mu_2 \cdot \mu_1 \cdot \mu_0.$$

Considérons ensuite (voir figure 3.14) :

$$\tau'_l = \mu_h \cdot [a, b_h] \cdot [b_h, a] \cdot \mu_{h-1} \cdots \mu_2 \cdot [a, b_2] \cdot [b_2, a] \cdot \mu_1 \cdot [a, b_1] \cdot [b_1, a] \cdot \mu_0.$$

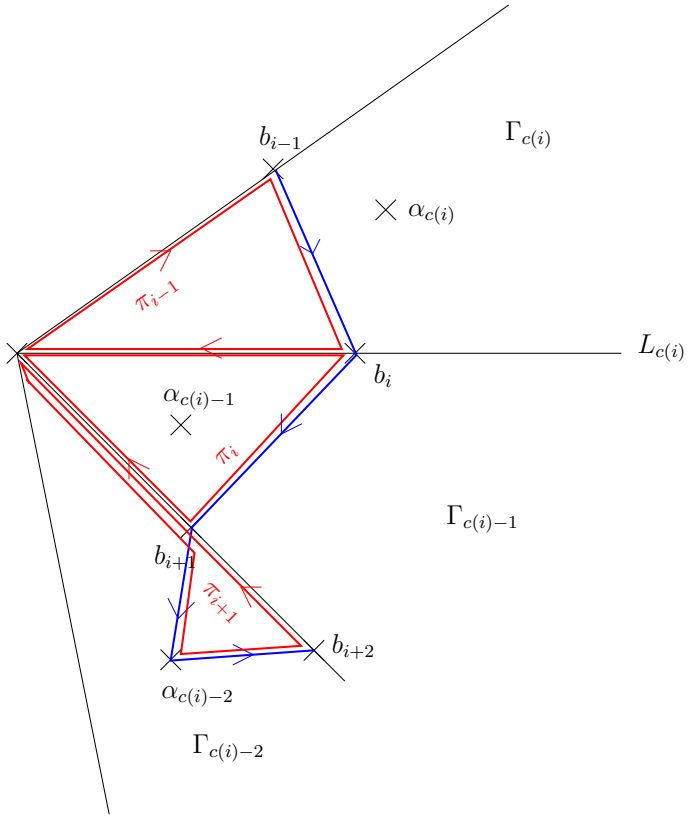

FIG. 3.14 – Exprimer τ_l en fonction des γ_i

Il est clair que τ'_l est homotope à τ_l dans X_0. Définissons les chemins :

$$\begin{aligned}
\pi_0 &= [b_1, a] \cdot \mu_0, \\
\pi_1 &= [b_2, a] \cdot \mu_1 \cdot [a, b_1], \\
\pi_2 &= [b_3, a] \cdot \mu_2 \cdot [a, b_2], \\
&\dots \\
\pi_{h-1} &= [b_h, a] \cdot \mu_{h-1} \cdot [a, b_{h-1}], \\
\pi_h &= \mu_h \cdot [a, b_h]
\end{aligned}$$

Les π_i sont des lacets de base a tels que chaque π_i soit inclus dans $\Gamma_{c(i)}$ ou $\Gamma_{c_{(i)-1}}$ (on pose $\Gamma_0 = \Gamma_n$). Pour exprimer τ_l en fonction des γ_j, il suffit donc d'exprimer à homotopie près chaque π_i en fonction des γ_j. Il est clair que π_i ne peut être homotope qu'à 5 lacets : $\gamma_{c(i)}^{\pm 1}$, $\gamma_{c(i)-1}^{\pm 1}$ (on pose $\gamma_0 = \gamma_n$) ou le lacet nul.

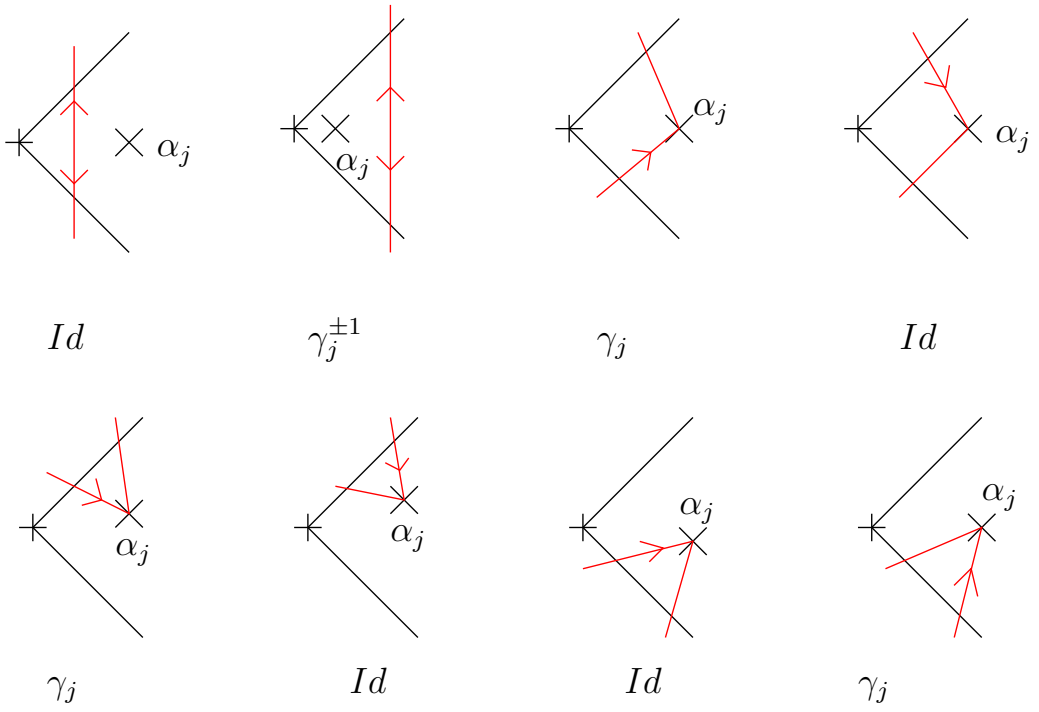

FIG. 3.15 – Différents cas possibles

Afin de simplifier les calculs et la présentation, nous faisons la remarque suivante : nous pouvons déformer continûment τ_l dans X_0 de sorte que les arcs de cercle de contournement des points critiques soient arbitrairement proches de ces derniers. Ainsi, au lieu d'avoir à déterminer les intersections de τ_l avec les L_j, il suffit de déterminer les intersections des L_j avec les arêtes sur le chemin dans $\mathcal{T}$ allant de a à α_l ; les arcs de cercle peuvent être ignorés. En tenant compte de cette remarque, les situations possibles sont données par la figure 3.15.

Les cas de π_0 et π_h se traitent comme des situations dégénérées des autres. Du point de vue des calculs, nous avons simplement besoin de pouvoir décider, lorsque le point critique pertinent n'appartient pas à μ_i, s'il se trouve du même côté de μ_i que a. S'il appartient μ_i, nous devons décider des positions relatives de b_i et b_{i+1} le long de $L_{c(i)}$.

Exemple 31. *Considérons l'exemple de la figure 3.12. En utilisant cette algorithme, nous obtenons immédiatement :*

$$\tau_1 = \gamma_1 \quad \tau_2 = \gamma_2 \quad \tau_3 = \gamma_2^{-1}\gamma_3\gamma_2 \quad \tau_4 = \gamma_2^{-1}\gamma_4\gamma_2 \quad \tau_5 = (\gamma_4\gamma_2)^{-1}\gamma_4(\gamma_4\gamma_2)$$
$$\tau_6 = (\gamma_5\gamma_4\gamma_2)^{-1}\gamma_6(\gamma_5\gamma_4\gamma_2) \quad \tau_7 = (\gamma_5\gamma_4\gamma_2)^{-1}\gamma_7(\gamma_5\gamma_4\gamma_2)$$

Les expressions sont relativement simples dans cet exemple, mais il en va autrement lorsque l'arbre possède une géométrie sinueuse, et en particulier lorsqu'il « s'enroule » sur lui-même.

Expression des γ_l en fonction des τ_l

Ce problème se ramène à de l'algorithmique des groupes libres : étant donné le groupe libre $G =< \gamma_l \mid 1 \leq l \leq n >$ à n générateurs nous pouvons définir le sous-groupe H de G par $H =< \tau_l \mid 1 \leq l \leq n >$, où les τ_l sont vus comme des mots en les γ_l. Par le théorème 21, on a $H = G$, de sorte que les γ_l s'expriment comme des mots en les τ_l. Les logiciels de calcul formel spécialisés pour les groupes tels que GAP (voir [GAP07]) ou MAGMA disposent de fonctions permettant, plus ou moins explicitement, de trouver ces expressions.

Exemple 32. *Nous reprenons le cas de l'exemple 3.12. La suite ci-dessous de commandes du paquetage FGA (Free Group Algorithms) de GAP 4 permet d'obtenir les expressions cherchées. Ce paquetage, dû à Christian Sievers [Sie03], fait partie de la distribution standard ; il est chargé par défaut.*

```
# Commandes pour réécrire un ensemble de générateurs
# (ti)_i = (tau_i)_i d'un groupe libre en fonction
# d'un autre ensemble de générateurs (gi)_i = (gamma_i)_i.

# Construit un groupe libre \'a 7 générateurs et
# affecte les générateurs aux variables g1,...,g7
#
G := FreeGroup("g1","g2","g3","g4","g5","g6","g7");
AssignGeneratorVariables(G);

# Expression des ti en fonction des gi.
# Le caractère ^ représente la conjugaison.
#
d1 := g4*g2;
d2 := g5*g4*g2;

t1 := g1;
t2 := g2;
t3 := g3^(g2^(-1));
t4 := g4^(g2^(-1));
t5 := g5^(d1^(-1));
```

```
t6 := g6^(d2^(-1));
t7 := g7^(d2^(-1));

# Construction du sous-groupe de H engendré par les ti
#
H := Group(t1,t2,t3,t4,t5,t6,t7);

# Construction d'un groupe libre \'a 7 générateurs
#
F := FreeGroup("z1","z2","z3","z4","z5","z6","z7");

# Construction de l'homomorphisme de F dans H tel que
# l'image de zi soit ti.
#
f := GroupHomomorphismByImages(F, H,
        GeneratorsOfGroup(F),
        GeneratorsOfGroup(H));

# Détermination des expressions des gi en fonction des zi.
#
gap> PreImagesRepresentative(f,g1);
z1
gap> PreImagesRepresentative(f,g2);
z2
gap> PreImagesRepresentative(f,g3);
z2*z3*z2^-1
gap> PreImagesRepresentative(f,g4);
z2*z4*z2^-1
gap> PreImagesRepresentative(f,g5);
z2*z4*z5*z4^-1*z2^-1
gap> PreImagesRepresentative(f,g6);
z2*z4*z5*z6*z5^-1*z4^-1*z2^-1
gap> PreImagesRepresentative(f,g7);
z2*z4*z5*z7*z5^-1*z4^-1*z2^-1
```

Il suffit maintenant de remplacer `zi` *par* τ_i *dans les expressions ci-dessus pour obtenir l'expression de* γ_i *et fonction des* τ_i*. A nouveau, on remarque que dans cet exemple, le calcul est simple, mais que c'est loin d'être le cas pour certains arbres.*

On pourra aussi accéder directement à des routines de plus bas niveau :

```
# Autre méthode: on ne passe pas par la construction
# de l'homomorphisme.
# [2, 3, -2] signifie: t2 * t3 * t2^(-1)
#
gap> AsWordLetterRepInGenerators(g1,H);
[ 1 ]
gap> AsWordLetterRepInGenerators(g2,H);
[ 2 ]
gap> AsWordLetterRepInGenerators(g3,H);
[ 2, 3, -2 ]
gap> AsWordLetterRepInGenerators(g4,H);
[ 2, 4, -2 ]
gap> AsWordLetterRepInGenerators(g5,H);
[ 2, 4, 5, -4, -2 ]
gap> AsWordLetterRepInGenerators(g6,H);
[ 2, 4, 5, 6, -5, -4, -2 ]
gap> AsWordLetterRepInGenerators(g7,H);
[ 2, 4, 5, 7, -5, -4, -2 ]
```

La méthode utilisée par le paquetage FGA de GAP 4 est basée sur la correspondance largement étudiée entre les sous-groupes d'un groupe libre et la notion d'« automate inverse » (nous remercions Laurent Bartholdi et Alexander Hulpke pour nous avoir éclairé sur le fonctionnement des routines GAP ci-dessus). Cette correspondance fournit notamment un algorithme pour déterminer une base d'un sous-groupe d'un groupe libre (ensemble minimal de générateurs). Un des avantages de cette approche est qu'elle permet de transférer des résultats de complexité obtenus dans le cadre de l'étude des automates.

Par exemple, on déduit de [BMMW00] (proposition 2.3) que le coût de la construction de l'automate correspondant au groupe $H =< \tau_l \mid 1 \leq l \leq n >$, vu comme sous-groupe du groupe $G =< \gamma_l \mid 1 \leq l \leq n >$ est en $O(N^2)$, où N désigne la longueur total des τ_l, considérés comme mots réduits en les γ_l. En conservant la trace des opérations effectuées durant la construction de l'automate, on peut obtenir l'expression des γ_l en fonction des τ_l, à un coût vraisemblablement similaire.

Néanmoins, nous avons étudié une approche plus directe basée sur des transformations de Nielsen. La situation que nous considérons est particulièrement simple du point de vue de la théorie des groupes et nous proposons ci-dessous un formalisme adapté.

Précisons nos définitions et notations :

- Un *mot* de G est définit comme étant une suite finie de symboles de l'alphabet $\mathcal{A} = \{\gamma_l, \gamma_l^{-1}\}_{1 \leq l \leq n}$. Il s'agit donc d'un objet de la forme $\gamma_{i_1}^{\epsilon_{i_1}} \gamma_{i_2}^{\epsilon_{i_2}} \cdots \gamma_{i_r}^{\epsilon_{i_r}}$ où les ϵ_{i_j} valent ± 1. Les mots représentent les éléments de G et le mot vide représente l'élément neutre de G, noté aussi 1.
- Un mot est dit *réduit* s'il ne contient aucune séquence de la forme $\gamma_{i_j}^{\epsilon_{i_j}} \gamma_{i_j}^{-\epsilon_{i_j}}$ (avec $\epsilon_{i_j} = \pm 1$). Tout mot w de G peut-être mis sous forme réduite $\overline{w}$ par suppression récursive des termes de la forme $\gamma_{i_j}^{\epsilon_{i_j}} \gamma_{i_j}^{-\epsilon_{i_j}}$. Deux mots représentent le même élément du groupe si et seulement si ils ont la même forme réduite.
- Un mot h est un *préfixe* d'un mot v s'il existe un mot w tel que $v = hw$.
- Si u et v sont deux mots, nous notons u^v le mot : $u^v = v^{-1}uv$. Ainsi, si w est un troisième mot, $(u^v)^w = u^{vw}$.
- Si m est un entier strictement positif et w un mot, nous noterons $w^m = ww \cdots w$ le mot formé de w repété m fois. Si m est strictement négatif, w^m est le mot w^{-1} répété $|m|$ fois.
- Dans la suite, nous considèrerons les τ_l comme des mots réduits en les γ_l. Pour tout l, il existe un unique mot réduit g_l de G tel que $\tau_l = \gamma_l^{g_l}$ et $\gamma_l^{g_l}$ soit réduit.
- Nous noterons $L(w)$ la *longueur* d'un mot w, c'est-à-dire le nombre de symboles $\gamma_{i_j}^{\epsilon_{i_j}}$ nécessaires à l'écriture de $\overline{w}$.

L'idée de l'algorithme est simple : si $\tau_l = \gamma_l^{g_l}$, on a $\gamma_l = \tau_l^{g_l^{-1}}$. Les g_l sont des mots en les γ_l, mais puisque $G = H$, ils peuvent aussi être vus comme des mots en les τ_l et τ_l^{-1}, maintenant considérés comme des symboles. Ainsi, en conjuguant τ_l par une suite de τ_i (et τ_i^{-1}) bien choisis, on doit obtenir l'expression de γ_l en fonctions des τ_i.

La proposition 36 ci-dessous indique la façon dont choisir les conjugués pour réduire la longueur des mots, tant que les τ_l sont différents des γ_l. En appliquant récursivement ces résultats, on obtiendra un mot de longueur 1, à savoir un γ_l ou son inverse.

Proposition 36. *Les énoncés suivants sont équivalents :*

(*a*) *Il existe un indice k ($1 \leq k \leq n$) tel que $\tau_k \neq \gamma_k$.*

(*b*) *Il existe deux indices k et l ($1 \leq k, l \leq n$), ainsi que $\epsilon_l \in \{\pm 1\}$, tels que $g_l^{-1}\gamma_l^{\epsilon_l}$ est un préfixe de g_k^{-1}.*

(*c*) *Il existe deux indices k et l ($1 \leq k, l \leq n$), ainsi que $\epsilon_l \in \{\pm 1\}$, tels que $L(\tau_k^{\tau_l^{\epsilon_l}}) < L(\tau_k)$.*

Démonstration. Montrons que (*a*) implique (*b*). Soit i_0 un indice tel $\tau_{i_0} \neq \gamma_{i_0}$. Il existe des entiers non nuls $u_1, \ldots, u_r$ ($r \geq 1$) et des indices $i_1, \ldots, i_r$ tels

que $i_j \neq i_{j+1}$ pour $1 \leq j \leq r-1$ et :

$$\gamma_{i_0} = \tau_{i_0}^{\tau_{i_1}^{u_1}\tau_{i_2}^{u_2}\ldots\tau_{i_r}^{u_r}}.$$

S'il n'existe aucun triplet (k, l, ϵ_l) vérifiant (b), alors pour tout j entre 0 et r, il existe un préfixe $\tilde{g}_{i_j}$ de g_{i_j} vérifiant :

$$\gamma_{i_0} = g_{i_r}^{-1}\gamma_{i_r}^{-u_r}\tilde{g}_{i_r}\cdots\tilde{g}_{i_1}^{-1}\gamma_{i_1}^{-u_1}\tilde{g}_{i_1}\tilde{g}_{i_0}^{-1}\gamma_{i_0}\tilde{g}_{i_0}\tilde{g}_{i_1}^{-1}\gamma_{i_1}^{u_1}\tilde{g}_{i_1}\cdots\tilde{g}_{i_r}^{-1}\gamma_{i_r}^{u_r}g_{i_r}$$

et tel que le membre de droite de l'égalité soit un mot réduit. C'est clairement impossible car les membres de gauche et de droite n'ont pas la même longueur.

Vérifions ensuite que (b) implique (c) : soit h un mot réduit tel que $g_k^{-1} = g_l^{-1}\gamma_l h$ (le cas $\epsilon_l = -1$ est analogue). Alors $\tau_k^{\tau_l} = \gamma_k^{h^{-1}g_l}$ et :

$$L(\tau_k^{\tau_l}) \leq 1 + 2[L(h) + L(g_l)] = 1 + 2[L(g_k) - 1] = L(\tau_k) - 2. \qquad (3.4)$$

En ce qui concerne $(c) \Rightarrow (a)$, il suffit de remarquer que si $\tau_k = \gamma_k$ alors $L(\tau_k^{\tau_l}) \geq L(\tau_k)$ car $L(\tau_k) = 1$ et $L(\tau_k^{\tau_l}) > 0$. La situation est identique si $\epsilon_l = -1$. □

On en déduit immédiatement l'algorithme ci-dessous, dans lequel on ne précise pas la représentation choisie pour les mots et pour lequel on ne cherche pas à optimiser la complexité. On suppose qu'on dispose d'une fonction `Longueur` qui donne la longueur d'un mot, d'une fonction `Reduction` pour la réduction d'un mot et enfin d'une fonction `Préfixe` qui a un couple (w_1, w_2) associe `Vrai` si le mot w_1 est un préfixe de w_2 et `Faux` sinon.

`Algorithme Conversion(`T`,`Γ`,`G`,`n`)`

`Entrée :`

T : *un tableau de symboles représentant les* τ_l,
Γ : *un tableau de symboles représentant les* γ_l,
G : *un tableau de mots réduits en les* $\Gamma[i]$ *représentant les* g_l,
n : *un entier, taille des tableaux précédents.*

`Sortie :`

Un tableau R *tel que* $R[l]$ *soit un mot réduit en les symboles* $T[i]$ *représentant* γ_l *en fonction des* τ_i.

`Début`

$N \leftarrow 0$
`Pour` i `de 1 à` n `faire`
$N \leftarrow N +$ `Longueur`$(G[i])$
$R[i] \leftarrow T[i]$

```
  Fin
  Tant que N ≠ 0 faire
    Pour i de 1 à n faire
      Pour j de 1 à n faire
        Si Préfixe(G[i]^-1 Γ[i]^±1, G[j]^-1) alors
          S ← Longueur(G[j])
          G[j] ← Reduction(G[j]G[i]^-1 Γ[i]^±1 G[i])
          R[j] ← R[i]^-1 R[j] R[i]
          N ← N − S + Longueur(G[j])
        Fin
      Fin
    Fin
  Fin
  Renvoyer R
Fin.
```

Sans avoir précisé la représentation des mots, il est délicat d'étudier la complexité de cet algorithme. Néanmoins nous proposons l'analyse grossière suivante : soit N la somme des longueurs des g_l. Par la proposition 36, la boucle `Tant que` sera exécutée au plus N fois puisqu'à chaque itération la longueur totale est réduite. Ainsi, les fonctions `Préfixe` et `Réduction` seront appelées au plus n^2N fois. Enfin, si M est le maximum des longueurs des g_l, il est raisonnable de penser que les appels de fonctions `Préfixe` et `Reduction` nécessiteront au plus $O(M)$ comparaisons et suppressions de symboles. Au final, on obtient $O(n^2NM)$ opérations élémentaires.

Cependant, si l'on cherche une façon astucieuse de représenter les mots permettant d'identifier plus rapidement les préfixes, on est amené à introduire un graphe orienté dont les arêtes sont étiqueté par des symboles de l'alphabet et à effectuer un traitement des mots τ_l analogue à celui de la construction de l'automate, telle que décrite dans [BMMW00]. Nous en tirons deux conclusions : notre approche spécifique n'est en fait pas fondamentalement différente de la méthode générale utilisée par le paquetage FGA de GAP, et, on peut déduire de la proposition 36 un algorithme de complexité $O(N^2)$.

Il serait intéressant d'étudier les majorations de N et M en fonction de n que l'on pourrait obtenir à partir de l'algorithme exprimant les τ_l en fonction des γ_l. Nous n'aborderons pas cette question en détail ici. Nous remarquerons simplement qu'il est facile de construire un exemple pour lequel $N \in O(n^2)$: prendre les points critiques alignés sur une droite horizontale et le point

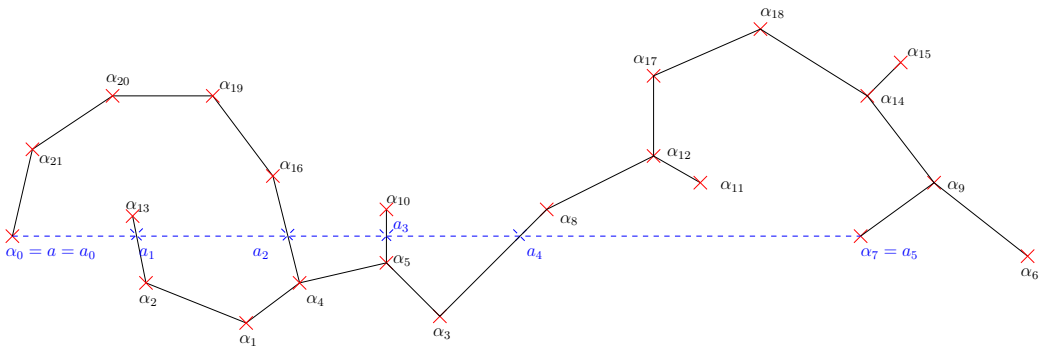

FIG. 3.16 – Trouver le chemin homotope à $[a, \alpha_7]$

de base à gauche des points critiques et au-dessus de cette droite. Dans ce cas, $\tau_l = (\gamma_{l-1} \dots \gamma_1)^{-1} \gamma_l (\gamma_{l-1} \dots \gamma_1)$ est de longueur $2l - 1$, de sorte qu'en sommant sur l, on obtient bien $O(n^2)$ (mais si le point de base est au-dessous de la droite, alors la longueur de τ_l est 1...).

3.4.4 Chemin dans l'arbre homotope à γ_l : méthode 2

Nous détaillons maintenant une autre approche possible pour trouver un chemin dans l'arbre homotope au segment $[a, \alpha_l]$. Nous avons choisi de détailler cette approche car elle présente l'intérêt de minimiser les calculs : elle ne nécessite que très peu de calculs autres que des comparaisons d'indices. On cherche donc quelle suite d'arêtes de $\mathcal{T}$ il faut parcourir, mais surtout dans quel sens on doit suivre l'arc de cercle $A(x_{ik}, x_{ij}, \pm)$ pour chaque couple $([\alpha_k, \alpha_i], [\alpha_i, \alpha_j])$ d'arêtes consécutives. Dans cette partie, *nous supposerons que le point de base a est choisi de telle sorte la partie réelle de a est inférieure à la partie réelle de chacun des points critiques α_i*. Cette hypothèse est effectuée afin de simplifier les explications. Nous expliquerons à la fin de cette section comment se passer d'une telle hypothèse. Enfin, pour illustrer nos propos tout au long de cette partie, nous traiterons les cas rencontrés sur des exemples, et notamment sur celui donné par la figure 3.16.

Découpage du segment $[a, \alpha_l]$

Remarque 21. *Si $[a = \alpha_0, \alpha_l]$ est une arête de $\mathcal{T}$, alors le chemin γ'_l cherché est tout simplement $[a, x_{l0}] \cdot A(x_{l0}, x_{l0}, +) \cdot [x_{l0}, a]$. Nous supposerons donc dans la suite que $[a, \alpha_l] \notin \mathcal{T}$.*

Dans ce cas, il est possible que le segment $[a, \alpha_l]$ traverse l'arbre $\mathcal{T}$:

Lemme 22. *Soit $[\alpha_i, \alpha_j]$ une arête de $\mathcal{T}$. Si cette arête intersecte le segment $[a, \alpha_l]$, alors on a $i < l < j$ ou $i > l > j$.*

Démonstration. Ceci est une conséquence directe de l'ordre choisi pour les points critiques (effectué avec la détermination principale) et de l'hypothèse sur le point de base a. □

Cette condition est nécessaire, mais pas suffisante. En effet, la figure 3.16 fournit un contre-exemple : l'arête $[\alpha_9, \alpha_6]$ vérifie $9 > 7 > 6$. Mais cette arête n'a pas d'intersection avec le segment $[a, \alpha_7]$; elle en a néanmoins une avec la droite $(a\,\alpha_7)$.

Nous noterons $a_0 = a, a_1, \ldots, a_{s-1}, a_s = \alpha_l$ les points d'intersection entre $\mathcal{T}$ et $[a, \alpha_l]$. Nous supposons que ces derniers sont ordonnés par ordre d'apparition sur le segment $[a, \alpha_l]$.

Remarque 22. *Compte tenu des hypothèses sur le choix du point de base, les points $(a_h)_{1 \leq h \leq s-1}$ sont réguliers.*

`Intersection(`$\mathcal{T}, l$`)`
`Entrée` :
$\mathcal{T}$: *Un arbre de recouvrement de $\mathcal{V}$.*
l : *un entier vérifiant $1 \leq l \leq n$.*
`Sortie` :
$\mathcal{L}$: *la liste ordonnée d'arêtes $[\alpha_i, \alpha_j]$ de $\mathcal{T}$ qui intersectent le segment $[a, \alpha_l]$. L'ordre est donné par l'ordre d'apparition des points d'intersection sur le segment $[a, \alpha_l]$.*
$\mathcal{L}'$: *la liste d'arêtes $[\alpha_i, \alpha_j]$ de $\mathcal{T}$ qui intersectent la demi-droite $\Delta_l \subset (a\alpha_l)$, d'origine α_l, et telle que $a \notin \Delta_l$.*

Cet algorithme utilise tout d'abord le lemme 22, puis trie la liste d'arêtes obtenues selon leur ordre d'apparition le long de la demi-droite $[a\,\alpha_l)$. Ceux ayant une intersection avec le segment $[a, \alpha_l]$ sont retournés ordonnés dans $\mathcal{L}$, et les autres sont retournés dans la liste $\mathcal{L}'$ (d'après l'hypothèse sur le point de base a, il n'y a pas d'intersection entre l'arbre $\mathcal{T}$ et la demi-droite Δ d'origine a, appartenant à la droite $(a\alpha_l)$, et telle que $\alpha_l \notin \Delta$. Seule la liste $\mathcal{L}$ nous intéresse en ce qui concerne les intersections entre $\mathcal{T}$ et le segment $[a, \alpha_l]$. Néanmoins, la liste $\mathcal{L}'$ sera utile pour l'algorithme `Sens-Parcours` décrit plus loin.

Enfin, nous ferons ici l'hypothèse que les points a_h ont été calculés, et nous utiliserons donc ces points dans la suite pour simplifier les explications.

D'un point de vue purement algorithmique, ce calcul n'est néanmoins pas nécessaire. Nous expliquerons ceci à la fin de cette section.

Exemple 33. *Si l'on considère la figure 3.16, la liste d'arêtes intersectant* (a, α_7) *est :*

$$\{[\alpha_2, \alpha_{13}], [\alpha_{16}, \alpha_4], [\alpha_5, \alpha_{10}], [\alpha_3, \alpha_8], [\alpha_9, \alpha_6]\}.$$

Après avoir ordonné ces arêtes par ordre d'apparition de a *à* α_7*, on élimine l'arête* $[\alpha_9, \alpha_6]$*, et l'on obtient* $\mathcal{L} = \{[\alpha_2, \alpha_{13}], [\alpha_{16}, \alpha_4], [\alpha_5, \alpha_{10}], [\alpha_3, \alpha_8]\}$*. On a de plus :*

$$a_0 = a, a_1 \in [\alpha_2, \alpha_{13}], a_2 \in [\alpha_{16}, \alpha_4], a_3 \in [\alpha_5, \alpha_{10}], a_4 \in [\alpha_3, \alpha_8] \text{ et } a_5 = \alpha_7$$

On a de plus $\mathcal{L}' = \{[\alpha_9, \alpha_6]\}$.

Le problème se ramène ensuite à trouver un chemin homotope au segment $[a_h, a_{h+1}]$ pour $0 \leq h \leq s-1$, puis à mettre bout à bout les chemins trouvés pour obtenir le chemin homotope à $[a, \alpha_l]$.

Sens de contournement des points critiques

A ce stade de l'algorithme, nous cherchons un chemin homotope au segment $[a_h, a_{h+1}]$, où a_h et a_{h+1} appartiennent à des arêtes de l'arbre $\mathcal{T}$, et tels que l'arbre $\mathcal{T}$ n'intersecte pas le segment ouvert $]a_h, a_{h+1}[$.

Nous commençons par poser les notations suivantes :
- π_h est le chemin dans l'arbre homotope à $[a_h, a_h{+}1]$ dans $\mathbb{C}\backslash\mathcal{V}$ recherché,
- τ_h est l'unique suite d'arêtes de $\mathcal{T}$ menant de a_h à a_{h+1}.

La suite d'arêtes τ_h sera donné par l'algorithme `Relier-arêtes`, que nous ne détaillons pas ici (il s'agit uniquement d'un parcours d'arbre). Le chemin π_h contient forcément les arêtes de τ_h. Néanmoins, il se peut que l'on ait à passer par d'autres arêtes intermédiaires.

Exemple 34. *Considérons le segment* $[a_4, a_5]$ *de la figure 3.16.*

On a ici $\tau_4 = [a_4, \alpha_8] \cdot [\alpha_8, \alpha_{12}] \cdot [\alpha_{12}, \alpha_{17}] \cdot [\alpha_{17}, \alpha_{18}] \cdot [\alpha_{18}, \alpha_{14}] \cdot [\alpha_{14}, \alpha_9] \cdot [\alpha_9, a_5]$. *Mais on voit bien sur la figure 3.17 que* π_3 *contient l'arête* $[\alpha_{12}, \alpha_{11}]$.

Supposons maintenant que nous connaissons la suite d'arêtes de $\mathcal{T}$ qu'il nous faut suivre pour obtenir le chemin π_h. Pour chaque couple d'arêtes consécutives $[\alpha_k, \alpha_i], [\alpha_i, \alpha_j]$ de cette suite, il nous faut trouver le sens ϵ_{kij} égal à $+$ ou $-$ que l'on doit utiliser pour contourner le point critique α_i,

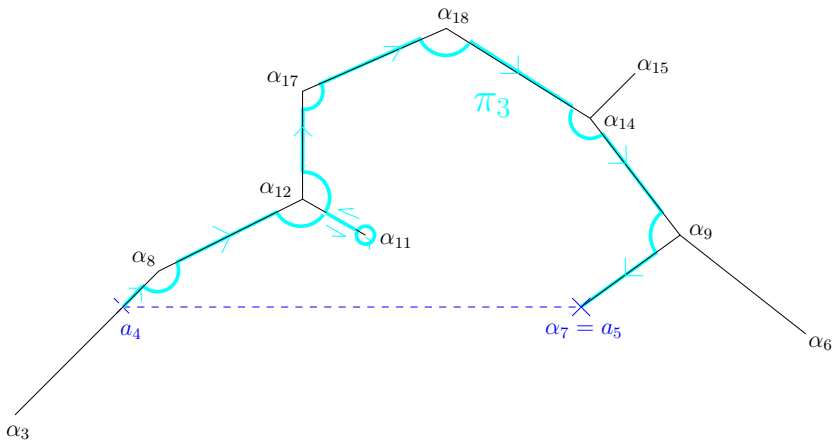

FIG. 3.17 – Le chemin π_4 homotope à $[a_4, a_5]$ dans $\mathbb{C}\backslash\mathcal{V}$

obtenant ainsi un arc de cercle $A(x_{ik}, x_{ij}, \epsilon_{kij})$. Il est à noter que l'on peut très bien avoir $k = j$, auquel cas $A(x_{ij}, x_{ij}, \epsilon_{jij})$ est le lacet $\beta_i^{\epsilon_{jij}}$ de point de base x_{ij}, et suivant le cercle de centre α_i et de rayon $|\alpha_i - x_{ij}|$ dans le sens ϵ_{jij}.

Lemme 23. *Soit π_h un chemin dans l'arbre homotope au segment $[a_h, a_{h+1}]$ dans $\mathbb{C}\backslash\mathcal{V}$. Alors si $A(x_{ik}, x_{ij}, \epsilon_{kij})$ et $A(x_{i'k'}, x_{i'j'}, \epsilon_{k'i'j'})$ sont des arcs de cercles appartenant à π_h, on a $\epsilon_{kij} = \epsilon_{k'i'j'}$.*

Démonstration. Supposons que ce ne soit pas le cas. Alors il existe une succession d'arêtes $[\alpha_k, \alpha_i]$, $[\alpha_i, \alpha_j]$, $[\alpha_j, \alpha_l]$ telle que la partie de π_h correspondant à ces arêtes soit $A(x_{ik}, x_{ij}, +) \cdot [x_{ij}, x_{ji}] \cdot A(x_{ji}, x_{jl}, -)$, comme c'est le cas sur la figure 3.9 ; le contournement des points critiques peut évidemment être inversé, mais cela donne un raisonnement similaire. Ainsi, le chemin construit traverse l'arbre $\mathcal{T}$. Or, pour être homotope à $[a_h, a_{h+1}]$, le chemin π_h ne doit pas intersecter l'arbre $\mathcal{T}$. □

Ainsi, chaque point critique contourné par le chemin π_h doit l'être dans le même sens. Il nous faut donc trouver ce sens de contournement.

Notons $X = \mathbb{C}\backslash[a_h, a_{h+1}]$ le plan complexe privé du segment $[a_h, a_{h+1}]$. Alors, dans X, tout chemin ayant pour point de départ a_h et pour point d'arrivée a_{h+1} est homotope à τ ou τ' (voir la figure 3.18).

Mais, comme l'on contourne tous les points critiques dans le même sens, toute suite d'arêtes que l'on aurait ainsi à ajouter à notre chemin forme en

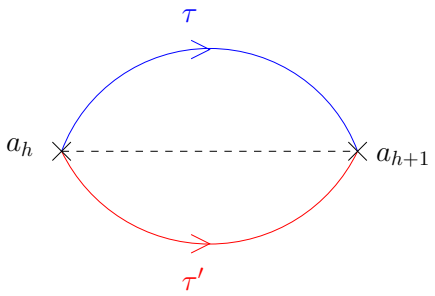

FIG. 3.18 – Classes d'homotopie dans $\mathbb{C}\backslash[a_h, a_{h+1}]$

fait un lacet qui ne traverse pas le segment $[a_h, a_{h+1}]$, et est donc homotopiquement nul dans X. De ce fait, le chemin π_h est homotope dans X à τ_h, et ce dernier est lui même homotope à l'un des deux chemins τ ou τ'.

La droite (a, α_l) divise naturellement le plan en deux parties, que nous appellerons respectivement le τ-plan pour le demi-plan contenant τ, et le τ'-plan pour celui contenant τ'.

Lemme 24. *Le point critique α_i appartient au τ-plan si et seulement si $i > l$.*

Démonstration. C'est une conséquence directe de l'ordre des points critiques et du choix du point de base a. □

La classe d'homotopie dans X de τ_h est alors donnée par le lemme 25. On note ici $\tau_h = [a_h, \alpha_{i_1}] \cdot [\alpha_{i_k}, \alpha_{i_{k+1}}]_{1 \leq k \leq s_h - 1} \cdot [\alpha_{i_{s_h}}, a_{h+1}]$. On rappelle que nous avons écarté le cas $[a_h, a_{h+1}] = [a, \alpha_l]$ (voir la remarque 21 et l'algorithme `Chemin`), ce qui implique $s_h \geq 1$.

Lemme 25. *Soit Δ la demi-droite $]a_h a_{h+1})$, et notons n_h l'entier égal au nombre de points d'intersection entre Δ et les s_h premières arêtes de τ_h, plus 1 si $i_{s_h} > l$. Alors on a :*

τ_h est homotope dans X à τ si et seulement si n_h est impair.

Démonstration. Soit $\Delta =]x_0 x_1)$ une demi-droite telle que :
- $x_0 \in]a_h, a_{h+1}[$,
- x_1 appartient au τ-plan,
- Δ ne passe par aucun point critique.

D'après le théorème de Jordan, le lacet $\tau_h \cdot [a_{h+1}, a_h]$ sépare le plan complexe en deux composantes connexes, l'une bornée que l'on notera $\mathcal{D}$, et l'autre non.

En parcourant la demi-droite Δ en partant de x_0, on obtient une succession d'entrées et de sorties dans $\mathcal{D}$. A la fin, comme $\mathcal{D}$ est borné, on se situe forcément dans la composante non bornée. Si le nombre d'intersections est impair, cela veut dire que l'on était dans $\mathcal{D}$ au départ, et donc que τ_h est homotope à τ. Sinon, on a commencé dans la composante non bornée, ce qui signifie que τ_h est homotope à τ'.

En considérant des demi-droites Δ formant un angle de plus en plus petit avec la droite $(a_h\, a_{h+1})$, on peut passer le raisonnement à la limite pour obtenir le lemme, avec l'argument supplémentaire suivant pour traiter la dernière arête : $i_{s_h} > l$ (ce qui signifie d'après le lemme 24 que $\alpha_{i_{s_h}}$ appartient au τ-plan) si et seulement si il existe $\eta > 0$ tel que pour toute droite Δ tel que décrite au début de cette preuve et formant un angle $0 < \epsilon \leq \eta$ avec la droite $(a\alpha_l)$, Δ intersecte $[\alpha_{i_{s_h}}, a_{h+1}[$. □

Ensuite, si τ_h est homotope à τ (voir figure 3.18), alors il nous faudra suivre l'arbre en contournant les points critiques dans le sens trigonométrique. Sinon, on les contournera dans le sens non trigonométrique.

Le lemme 25 nous conduit naturellement à l'algorithme suivant. Ici, on note $\#\mathcal{L}'$ le cardinal de la liste $\mathcal{L}'$.

`Sens-Parcours(`$\tau_h, \mathcal{L}', \mathcal{L}'', h, l$`)`
`Entrée :`
- $\tau_h = [a_h, \alpha_{i_1}] \cdot [\alpha_{i_k}, \alpha_{i_{k+1}}]_{1 \leq k \leq s_h - 1} \cdot [\alpha_{i_{s_h}}, a_{h+1}]$ *la liste d'arêtes de* $\mathcal{T}$ *menant de* a_h *à* a_{h+1} *avec* $s_h \geq 1$.
- $\mathcal{L}'$: *la liste ordonnée d'arêtes de* $\mathcal{T}$ *qui intersectent le segment* $[a, \alpha_l]$. *L'ordre est donné par l'ordre d'apparition des points d'intersection sur le segment* $[a, \alpha_l]$.
- $\mathcal{L}''$: *la liste d'arêtes* $[\alpha_i, \alpha_j]$ *de* $\mathcal{T}$ *qui intersectent la demi-droite* $\Delta_l \subset (a\alpha_l)$, *d'origine* α_l, *et telle que* $a \notin \Delta_l$.
- h : *l'indice du segment considérée.*
- l : *l'indice du point critique considéré.*

`Sortie :`
- ϵ : *Le sens* $+$ *ou* $-$ *dans lequel il faut contourner les points critiques pour suivre le chemin* π_h *homotope à* $[a_h, a_{h+1}]$ *dans* $\mathbb{C}\backslash\mathcal{V}$.

`Début`
  $\mathcal{L} \leftarrow \mathcal{L}'[h+2 \cdots \#\mathcal{L}'] \cup \mathcal{L}''$
  $n_h \leftarrow 0$
  `Pour` k `allant de 1 à` $s_h - 1$ `faire`
    `Si` $[\alpha_{i_k}, \alpha_{i_{k+1}}] \in \mathcal{L}$ `et` $i_k \neq l$ `alors` $n_h \leftarrow n_h + 1$ `Fin`
  `Fin`
  `Si` $i_{s_h} > l$ `alors` $n_h \leftarrow n_h + 1$ `Fin`

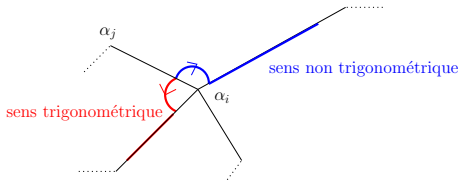

FIG. 3.19 – Contournement de α_i

```
  Si n_h est impair alors
    Retourner +
  Sinon
    Retourner −
  Fin
Fin.
```

Exemple 35. *Considérons le segment* $[a_0, a_1]$ *de la figure 3.16. On a ici* $\tau_0 = [a_0, \alpha_{21}] \cdot [\alpha_{21}, \alpha_{20}] \cdot [\alpha_{20}, \alpha_{19}] \cdot [\alpha_{19}, \alpha_{16}] \cdot [\alpha_{16}, \alpha_4] \cdot [\alpha_4, \alpha_1] \cdot [\alpha_1, \alpha_2] \cdot [\alpha_2, a_1]$.

Excepté la dernière arête, seule $[\alpha_{16}, \alpha_4]$ *a une intersection avec la droite* $[a, \alpha_7]$. *Or, si* $\mathcal{L}', \mathcal{L}''$ *est la sortie de* `Intersection`$(\mathcal{T}, 7)$, *on a* $[\alpha_{16}, \alpha_4] \in \mathcal{L}'$, *et c'est le deuxième élément de cette liste (voir l'exemple 33). Comme* $2 > 1$, $n_0 \leftarrow n_0 + 1$.

Pour la dernière arête, τ_0 *parcourt* $[\alpha_2, \alpha_{13}]$ *de* α_2 *à* α_{13}. *Puisque* $2 < 7$, *toute droite* Δ *comme dans la preuve du lemme 25 n'intersecte pas* $[\alpha_2, a_1]$. *Finalement, on obtient* $n_0 = 1$, *qui est impair. Le sens de contournement des points critiques entre* a_0 *et* a_1 *est donc le sens trigonométrique.*

Série d'arêtes à suivre

Nous allons maintenant décrire comment trouver le chemin π_h une fois que l'on connait le sens de contournement des points critiques que doivent suivre les arcs de cercle de ce chemin. Pour cela, il faut, à chaque point critique approché, contourner ce dernier jusqu'à ce que l'on rencontre une arête, qui est alors l'arête suivante de notre chemin. La figure 3.19 illustre ce point de vue.

Nous devons donc connaître un ordre local pour chaque point critique de l'arbre $\mathcal{T}$, donné par l'algorithme `Permutation-Locale`. Ce dernier retourne, pour chaque point critique α_i, l'ensemble $\mathcal{L}_i$ des indices de ses voisins, ainsi que la permutation σ_i, qui à un élément j de $\mathcal{L}_i$ retourne le voisin suivant k

de $\mathcal{L}_i$ dans l'ordre trigonométrique.

La suite d'arêtes suivie par le chemin π_h est alors donnée par l'algorithme suivant :

`Chemin-h(`$\mathcal{V}, \mathcal{T}, \mathcal{L}, \sigma, [a_h, a_{h+1}], h, l, \epsilon_h$`)`
`Entrées :`
$\mathcal{V}$: *l'ensemble des points critiques augmenté du point de base.*
$\mathcal{T}$: *un arbre de recouvrement minimum de* $\mathcal{V}$.
$\sigma = \sigma_0, \ldots, \sigma_n$: *les permutations locales en les* α_i.
$\mathcal{L} = \mathcal{L}_0, \ldots, \mathcal{L}_n$: *les ensembles voisins des* α_i.
$[a_h, a_{h+1}]$: *le segment considéré*
h : *l'indice du segment considéré.*
l : *l'indice du point critique considéré.*
$\epsilon_h = +$ ou $-$: *le sens de contournement des points critiques pour le chemin* π_h.
`Sortie :`
π_h : *Un chemin dans l'arbre* $\mathcal{T}$ *homotope à* $[a_h, a_{h+1}]$ *dans* $\mathbb{C}\backslash\mathcal{V}$.
`Début`
`Si` $h = 0$ `alors`
$j \leftarrow 0$
`Si` $\epsilon_h = +$ `alors`
`Si pour tout` $i \in \mathcal{L}_0$, $i < l$ `alors` $k \leftarrow \min \mathcal{L}_0$
`Sinon` $k \leftarrow \min\{i \in \mathcal{L}_0 \,|\, i > l\}$ `Fin`
`Sinon`
`Si pour tout` $i \in \mathcal{L}_0$, $i > l$ `alors` $k \leftarrow \max \mathcal{L}_0$
`Sinon` $k \leftarrow \max\{i \in \mathcal{L}_0 \,|\, i < l\}$ `Fin`
`Fin`
`Sinon`
`Si (`$\epsilon_h = +$ `et` $j_h < k_h$`) ou (`$\epsilon_h = -$ `et` $j_h > k_h$`) alors`
$(j, k) \leftarrow (j_h, k_h)$
`Sinon`
$(j, k) \leftarrow (k_h, j_h)$
`Fin`
`Fin`
$\pi_h \leftarrow [a_h, x_{kj}]$
$b \leftarrow vrai$
`Tant que` b `faire`
`Si` $\epsilon_h = +$ `alors` $w \leftarrow \sigma_k(j)$ `Sinon` $w \leftarrow {\sigma_k}^{-1}(j)$ `Fin`
$\pi_h \leftarrow \pi_h \cdot A(x_{kj}, x_{kw}, \epsilon_h)$
`Si` $a_{h+1} = \alpha_l$ `et` $w = l$ `alors`

```
      Si ϵ_h = + alors w_0 ← σ_w(k) Sinon w_0 ← σ_w^{-1}(k) Fin
      Si A(x_lk, x_lw0, ϵ_h) intersecte [α_0, α_l] alors
        π_h ← π_h · [x_kw, x_wk]
        (j, k) ← (k, w)
      Sinon
        b ← faux
        π_h ← π_h · [x_kw, a_{h+1}]
      Fin
    Sinon Si a_{h+1} ∈ [α_k, α_w] et
    (a_{h+1} = α_l ou (k < w et ϵ_h = −) ou (k > w et ϵ_h = +)) alors
      b ← faux
      π_h ← π_h · [x_kw, a_{h+1}]
    Sinon
      π_h ← π_h · [x_kw, x_wk]
      (j, k) ← (k, w)
    Fin
  Fin
  Retourner π_h
Fin.
```

À chaque boucle `Tant que` de l'algorithme `Chemin-h`, les indices j, k, w sont définis de telle façon que le chemin construit va de l'arête $[\alpha_j, \alpha_k]$ à l'arête $[\alpha_k, \alpha_w]$.

Si $a_h \in [\alpha_{j_h}, \alpha_{k_h}]$, la figure 3.20 illustre comment trouver par lequel des deux points intermédiaires $x_{j_h k_h}$ ou $x_{k_h j_h}$ le chemin π_h doit commencer.

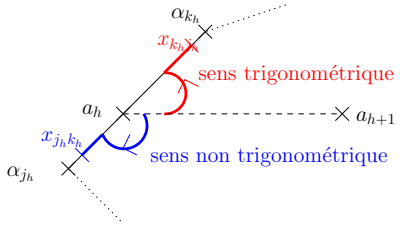

FIG. 3.20 – Point intermédiaire suivant a_h

Le cas $h = 0$ demande un traitement particulier. La figure 3.21 illustre alors les cas à considérer. On rappelle que nous avons écarté le cas où $[a, \alpha_l] \in \mathcal{T}$, ce qui implique que $l \notin \mathcal{L}_0$.

Enfin, le cas $a_{h+1} = \alpha_l$ demande de s'assurer que l'on oublie pas de contour-

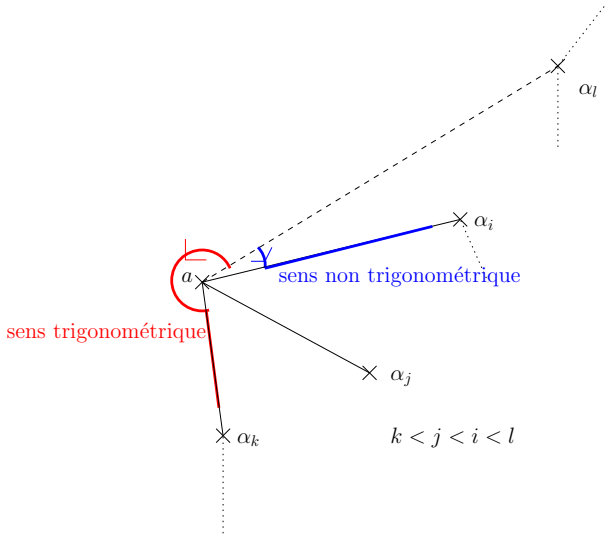

FIG. 3.21 – Choix de l'arête de départ

ner certains points critiques quand on arrive au point α_l. Le test d'intersection se résout en pratique par une simple comparaison d'arguments.

Exemple 36. *Considérons le chemin π_0 dans l'arbre $\mathcal{T}$ de la figure 3.16 homotope au segment $[a_0, a_1]$, illustré sur la figure 3.22. Ici, le seul point critique relié au point de base a est α_{21}. Le chemin π_0 commence donc par $[a, x_{21,0}]$. Ensuite, le sens de contournement des points critiques est le sens trigonométrique (voir l'exemple 35). L'ordre local en α_{21} nous donne $\sigma_{21}(0) = 20$. La suite du chemin est donc $A(x_{21,0}, x_{21,20}, +)$. A ce stade, nous n'avons pas encore atteint l'arête $[\alpha_2, \alpha_{13}]$ donc on continue, et on ajoute l'arête $[x_{20}, x_{19}]$ à π_0. Et on continue ainsi jusqu'à arriver au point $x_{2,1}$. On a alors $j = 1$ et $k = 2$. On a ainsi $w = \sigma_2(1) = 13$, et on ajoute donc au chemin l'arc de cercle $A(x_{2,1}, x_{2,13}, +)$. Ici, nous avons atteint l'arête $[\alpha_2, \alpha_{13}]$. Mais le chemin n'est pas fini, puisque pour avoir un chemin homotope au segment $[a_0, a_1]$ dans $\mathbb{C}\backslash\mathcal{V}$, il nous faut encore contourner le point critique α_{13}. Ceci est bien fait par notre algorithme : on a $k = 2 = \min\{2, 13\}$ et $\epsilon_h = +$ donc on continue la boucle* `Tant que`*. On ajoute donc le segment $[x_{2,13}, x_{13,2}]$ au chemin, puis on pose $j = 2$ et $k = 13$, et ensuite $w = \sigma_{13}(2) = 2$, ajoutant $A(x_{13,2}, x_{13,2}, +)$ au chemin. On a alors $k = 13$, $w = 2$ et $\epsilon_h = +$. Ceci est bien la fin de la boucle* `Tant que`*, et donc de l'algorithme. On obtient finalement :*

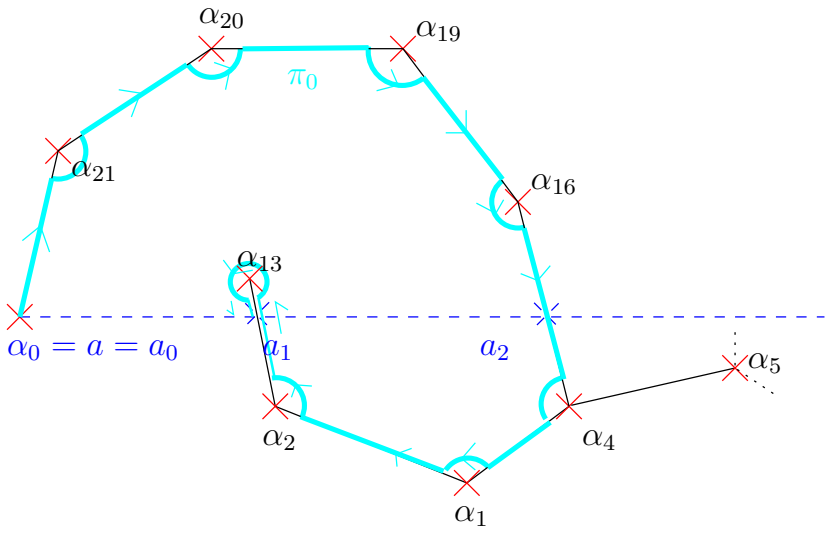

FIG. 3.22 – Le chemin π_0 homotope à $[a_0, a_1]$ dans $\mathbb{C}\backslash\mathcal{V}$

$\pi_0 = [a, x_{21,0}] \cdot A(x_{21,0}, x_{21,20}, +) \cdot [x_{21,20}, x_{20,21}] \cdot A(x_{20,21}, x_{20,19}, +) \cdot [x_{20,19}, x_{19,20}] \cdot A(x_{19,20}, x_{19,16}, +) \cdot [x_{19,16}, x_{16,19}] \cdot A(x_{16,19}, x_{16,4}, +) \cdot [x_{16,4}, x_{4,16}] \cdot A(x_{4,16}, x_{4,1}, +) \cdot [x_{4,1}, x_{1,4}] \cdot A(x_{1,4}, x_{1,2}, +) \cdot [x_{1,2}, x_{2,1}] \cdot A(x_{2,1}, x_{2,13}, +) \cdot [x_{2,13}, x_{13,2}] \cdot A(x_{13,2}, x_{13,2}, +) \cdot [x_{13,2}, a_1]$

Il se peut également que le début du chemin commence par contourner un point critique (ou plus généralement une branche de l'arbre), comme l'illustre la figure 3.23. Ici, le début du chemin π_2 est donné par $[a_2, x_{9,6}] \cdot A(x_{9,6}, x_{9,6}, +) \cdot [x_{9,6}, x_{6,9}] \cdot \ldots$

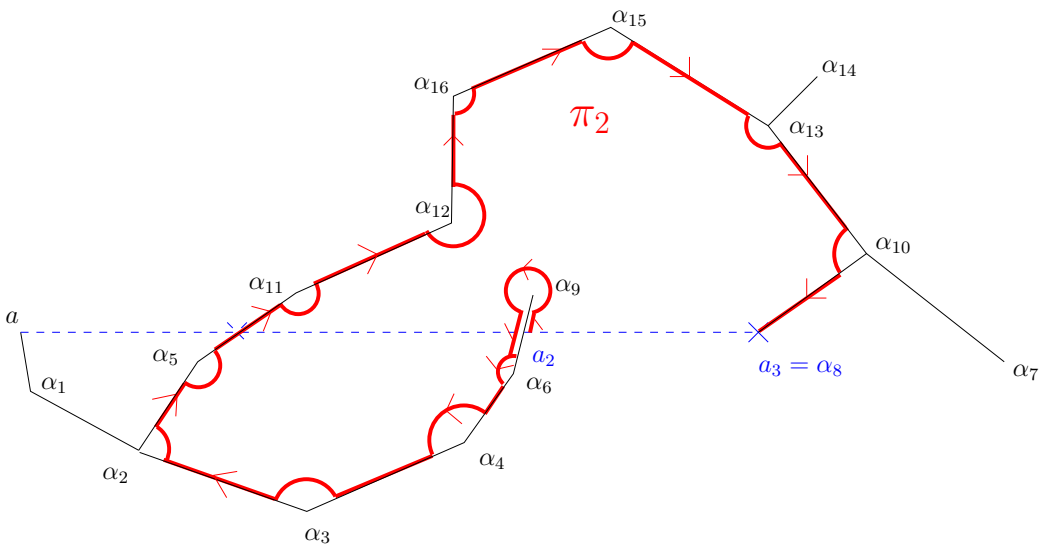

FIG. 3.23 – Le chemin π_2

Algorithme de construction des chemins

Enfin, l'algorithme `Chemin` donne, pour chaque point critique α_l, un chemin δ_l homotope à $[a, \sigma_l]$.

`Chemin`$(\mathcal{V}, \mathcal{T}, \sigma, \mathcal{L})$
`Entrées` :
$\mathcal{V}$: *l'ensemble des points critiques augmenté du point de base.*
$\mathcal{T}$: *un arbre de recouvrement minimum de* $\mathcal{V}$.
$\sigma = \sigma_0, \ldots, \sigma_n$: *les permutations locales en les* α_i.
$\mathcal{L} = \mathcal{L}_0, \ldots, \mathcal{L}_n$: *les ensembles voisins des* α_i.
`Sortie` : *Pour chaque point critique* α_l, *un chemin dans l'arbre* δ_l *homotope à* $[a, \alpha_l]$.
`Début`
$n \leftarrow \#\mathcal{V} - 1$
`Pour` l `allant de` 1 `à` n `faire`
`Si` $[a, \alpha_l] \in \mathcal{T}$ `alors` $\delta_l \leftarrow [a, \alpha_l]$ `Boucle suivante Fin`
$\delta_l \leftarrow \emptyset$
$(\mathcal{L}', \mathcal{L}'') \leftarrow$ `Intersection`$(\mathcal{T}, l)$
`Pour` h `allant de` 0 `à` $\#\mathcal{L}' - 1$ `faire`
$\tau \leftarrow$ `Relier-arêtes`$(\mathcal{T}, \mathcal{L}'[h], \mathcal{L}'[h+1])$
$\epsilon \leftarrow$ `Sens` $-$ `Parcours`$(\tau, \mathcal{L}', \mathcal{L}'', l)$
$\pi \leftarrow$ `Chemin` $-$ `h`$(\mathcal{V}, \mathcal{T}, \mathcal{L}, \sigma, [a_h, a_{h+1}], l, \epsilon)$

```
        δ_l ← δ_l · π
      Fin
    Fin
    Retourner δ
Fin.
```

Ainsi, nous avons décrit dans cette partie un algorithme qui construit un chemin suivant l'arbre homotope au chemin γ_l dans X_0. Nous concluons cette section en faisant quelques remarques, notamment sur le point de vue calculatoire de cet algorithme.

Tout d'abord, il n'est pas nécessaire de calculer précisément les valeurs des points a_h : la seule information dont nous avons besoin est de pouvoir ordonner les points a_h sur le segment $[a, \alpha_l]$. En effet, cet ordre mis à part, la seule information importante concernant a_h est l'arête de l'arbre à laquelle il appartient. Ainsi, on peut ôter à chacun des chemins π_h les arêtes $[a_h, x_{ij}]$ et $[x_{kw}, a_{h+1}]$, quitte à rajouter l'arête $[x_{kw}, x_{wk}]$ entre π_h et π_{h+1} pour l'obtention du chemin δ_l.

Cette remarque prise en compte, on peut voir que l'avantage de cet algorithme est qu'il a uniquement besoin de calculer deux choses :
- l'ordre des points d'intersection a_h entre $[a, \alpha_l]$ et $\mathcal{T}$.
- la permutation locale de chaque point critique rencontré.

Toutes les autres opérations se contentent de comparer les indices des points critiques.

Ensuite, pour simplifier la formulation, notre algorithme retourne des chemins δ_l homotope au segment $[a, \alpha_l]$. En pratique, on ne cherche pas à atteindre le point critique α_l, mais uniquement le point x_{lk}, si $[x_{kl}, \alpha_l]$ est la dernière arête du chemin δ_l retourné (en définissant $x_{0l} = a$ pour traiter le cas particulier où $[a, \alpha_l] \in \mathcal{T}$). Pour avoir un tel chemin, il suffit donc de remplacer l'arête $[x_{kl}, \alpha_l]$ par $[x_{kl}, x_{lk}]$ dans le chemin δ_l trouvé. Cette modification effectuée, le lacet $\gamma'_l = \delta_l^{-1} \cdot \beta_l \cdot \delta_l$ est homotope dans X_0 au chemin γ_l.

Enfin, l'hypothèse sur le point de base effectué au début de cette section peut facilement être supprimée : elle permet uniquement de mieux décrire la division du plan en les deux demi-plans que sont le τ-plan et le τ'-plan (voir le lemme 24). Ainsi, si l'on effectue les corrections suivantes :
- Remplacer « $i_{s_h} > l$ » par « $\alpha_{i_{s_h}}$ est dans le τ-plan »dans l'algorithme `Sens-Parcours`,
- Modifier les traitements des arêtes initiales et finales dans l'algorithme `Chemin-h` (il suffit de remplacer les comparaisons d'indices par la notion

d'appartenance au τ-plan ou au τ'-plan).
Alors l'hypothèse sur le point de base a n'a plus à être considérée dans nos algorithmes.

3.5 Connexion entre les fibres

Pour chaque point critique α_l, la section 3.4 nous donne un chemin dans l'arbre γ'_l homotope au chemin γ_l de la figure 3.1. Ce chemin γ'_l est constitué d'une succession de segments $[x_{ij}, x_{ji}]$ et d'arcs de cercles $A(x_{ji}, x_{jk}, \pm)$. Dans cette partie, nous allons décrire la façon dont on connecte les fibres successives au-dessus des points de ce chemin.

3.5.1 Principe

Nous commençons par expliquer le principe de la méthode en illustrant celle-ci à partir de la figure 3.9, page 141. Nous expliquerons dans la section 3.5.2 comment améliorer l'efficacité de ce procédé.

Nous commençons par calculer la fibre de F en le point de base a et au-dessus de tous les points x_{ij} par lesquels passe notre chemin. Ensuite, pour suivre le segment $[x_{ij}, x_{ji}]$, nous calculons les développements en série de F au-dessus du milieu α_{ij} de l'arête $[\alpha_i, \alpha_j]$. Puis l'on connecte ces développements en séries aux deux fibres $\mathcal{F}(x_{ij})$ et $\mathcal{F}(x_{ji})$, comme décrit dans la partie 3.3.4. Cela nous permet de connecter les éléments de ces fibres deux à deux.

Pour suivre l'arc de cercle $A(x_{ji}, x_{jk}, \pm)$, on calcule les développements en série de Puiseux au dessus du point critique α_j, puis on évalue ces développements en les points x_{ji} et x_{jk} pour les connecter aux fibres en ces points, et ainsi connecter les deux fibres entre elles (voir la proposition 33). Pour cela, il est recommandé de choisir une détermination pour la fonction racine e-ième qui soit telle que la ligne de discontinuité B de ces développements soit la plus éloignée possible des points x_{ji} et x_{jk}, afin d'éviter au maximum les problèmes numériques qui peuvent apparaître si les points de connexion étaient trop près de B. Le meilleur choix possible est la demi-droite $B = (\alpha_j, (\arg(c_k - c_j) + \arg(c_i - c_j))/2 + \mu\pi)$ où μ est égal à un si l'on va dans le sens trigonométrique et zéro sinon. De plus, d'après le lemme 21, l'angle formé entre chacune des arêtes considérées et la demi-droite B est au moins égal à $\frac{\pi}{6}$.

En combinant ces connexions, et si l'on note, comme à la fin de la section

3.4.1, δ_l le chemin qui va du point de base a à l'un des points de connexion x_{lh} proche du point critique α_l, alors on connecte ainsi $\mathcal{F}(a)$ à $\mathcal{F}(x_{lh})$. Le lacet effectué autour du point critique α_l est alors obtenu à l'aide de la proposition 32. Enfin, il n'est pas nécessaire d'effectuer le prolongement analytique le long du chemin δ_l^{-1}, puisque la connexion entre les fibres $\mathcal{F}(x_{lh})$ et $\mathcal{F}(a)$ le long du chemin δ_l^{-1} peut être obtenue en inversant celle calculée pour le chemin δ_l.

Ainsi, nous avons décrit une méthode qui permet de calculer le groupe de monodromie $\mathcal{M}$. Cette méthode nécessite les calculs suivants :

- Les développements de Puiseux au dessus de chacun des n points critiques α_i,
- Les développements en séries au dessus de $n+1$ points réguliers (les n milieux des arêtes de l'arbre, ainsi que le point de base a),
- une approximation numérique des fibres en les points intermédiaires, soient $2n$ points (on peut se contenter de 2 points intermédiaires par arête).

Néanmoins, jusqu'à présent, nous n'avons pas encore pris compte les ordres de troncation. Or, ceux-ci peuvent être élevés, et de ce fait, en se contentant de deux points d'évaluation par arête de $\mathcal{T}$, on obtiendrait une méthode inutilisable en pratique. En effet, si l'on considère la borne donnée par la proposition 34, on s'aperçoit que cette borne tend rapidement vers l'infini quand β tend vers 1, ce qui correspond au cas où le point d'évaluation x_1 est relativement proche du rayon de convergence de la série considérée. Cette situation apparaît quand on a une arête de l'arbre $\mathcal{T}$ constitué de deux points critiques α_j et α_k proches, et que l'un des voisins α_l de α_k dans $\mathcal{T}$ est à une distance plus importante. Dans ce cas, de part sa définition (voir la section 3.4.2), le point de connexion x_{kl} est nécessairement proche du rayon de convergence des séries calculées en α_{kl}, milieu de l'arête $[\alpha_k, \alpha_l]$.

Exemple 37. *Considérons le polynôme* $F(x,y) = y^3 - x^5 + 2\,(10x-1)^2 \in \mathbb{Q}[x,y]$. *Ce polynôme, provenant de [Mig92, page 170], a comme particularité que deux de ses points critiques sont très proches (à une distance inférieure à* $10^{-\frac{7}{2}}$*). En prenant par exemple* $a = -3 + 2I$ *comme point de base, on obtient des points critiques indexés de la façon suivante (on donne ici une approximation de ces points critiques) :*

- $\alpha_1 \simeq -2.990197594 - 5.065348139\,I$,
- $\alpha_2 \simeq 0.09977763419$,
- $\alpha_3 \simeq 0.1002248660$,
- $\alpha_4 \simeq 5.780392689$,
- $\alpha_5 \simeq -2.990197594 + 5.065348139\,I$.

L'arbre de recouvrement minimum est alors :

$$\mathcal{T} = \{[a, \alpha_5], [a, \alpha_2], [\alpha_2, \alpha_1], [\alpha_2, \alpha_3], [\alpha_3, \alpha_4]\}$$

Pour pouvoir calculer la permutation associée au point critique α_1, nous allons donc suivre l'arête $[\alpha_1, \alpha_2]$. Comme $\delta(\alpha_2) \simeq 0.00044723181$, le point $x_{2,1}$ le plus éloigné de α_2 que l'on puisse prendre vérifie $x_{2,1} \simeq 0.09954472791 - 0.0003817996285\,I$. De ce fait, pour évaluer les séries au-dessus de $\alpha_{1,2} \simeq -1.445209980 - 2.532674070\,I$ en $x_{2,1}$, on obtient (voir la proposition 34) $\beta = \frac{|\alpha_{1,2} - x_{2,1}|}{\delta(\alpha_{1,2})} \simeq 0.9998492506$. En utilisant les bornes de [Mig92], on trouve une borne $M = 26.85009798$ pour les valeurs des séries dans le disque de convergence. En calculant la fibre en $x_{2,1}$, on obtient une précision nécessaire $\eta \simeq 0.03565093653$. Au final, le critère de la proposition 34 nous dit qu'il faudrait avoir un ordre de troncation $N \geq 102309$ pour avoir une connexion fiable entre la fibre $\mathcal{F}(x_{2,1})$ et les séries en $\alpha_{1,2}$.

Pour raffiner notre stratégie, nous allons donc introduire des points intermédiaires supplémentaires, en nombre contrôlé, comme décrit dans la partie suivante.

3.5.2 Compromis entre le nombre de pas et les ordres de troncation

Comme nous l'avons vu dans l'exemple 37, les ordres de troncation donnés par la proposition 34 peuvent atteindre des tailles trop élevées en pratique. Nous allons ici étudier plus précisément cette borne, et allons expliquer comment la contrôler au mieux.

La proposition 34 nous donne donc une borne :

$$N \geq \frac{\ln\left(\frac{\eta}{M}\right) + \ln(1-\beta)}{\ln(\beta)} - 1, \text{ avec } \beta = \left(\frac{|x_1 - x_0|}{\rho}\right)^{\frac{1}{e}}$$

On peut séparer cette formule en deux parties. D'une part, on a le nombre $\ln\left(\frac{\eta}{M}\right)$, qui dépend des feuillets du revêtement $(\mathcal{C}, x)$: s'ils sont proches l'un de l'autre ou pas, s'ils prennent des valeurs importantes ou pas etc. Cette partie dépend donc intrinsèquement du problème, et est de ce fait difficile à contrôler. Par contre, l'autre partie de cette borne, $\frac{\ln(1-\beta)}{\ln(\beta)}$, dépend uniquement de β, et donc des points du plan complexe utilisés pour le prolongement analytique. De ce fait, cette partie est facilement contrôlable. Nous allons donc maintenant proposer une stratégie dans laquelle nous allons utiliser des

points intermédiaires supplémentaires, de telle sorte que chacun des nombres β considérés soient bornés par une constante. Il s'agit ici d'un raffinement du raisonnement proposé dans [Pot07]. En effet, l'argument utilisé dans cet article provenait principalement d'observations pratiques. Ici, nous considérons de plus la complexité du prolongement analytique pour affiner le choix du nombre β.

Optimiser le prolongement analytique

Dans notre contexte, le problème est donc de minimiser le coût (c'est-à-dire la complexité binaire) du prolongement analytique entre le milieu de l'arête α_{ij} et le point x_{ij}. Le problème est donc d'approcher le point critique α_i le long du chemin $[\alpha_{ij}, x_{ij}]$. De plus, on sait (voir le lemme 20), que pour tout point $x_0 \in [\alpha_{ij}, x_{ij}]$, on a $\delta(x_0) = |x_0 - \alpha_i|$. On peut donc schématiser notre problème de la façon suivante :

- Le point $x = 0$ est un point critique.
- x_0 est un point régulier dont le plus proche point critique est 0.
- $x_1 \in [0, x_0]$.
- On veut minimiser la complexité binaire du prolongement analytique entre x_0 et x_1.

Des études de ce cas ont été effectuées dans le cadre des séries solutions d'une équation différentielle [CC87a, CC90, vdH99, Mez07]. Ces études sont basées sur le fait qu'il existe des algorithmes calculant les séries solutions d'une équation différentielle ayant une complexité binaire quasi linéaire en le nombre de termes souhaités.

Dans notre cas, l'algorithme de Newton quadratique (voir la section 2.1.1), qui permet de calculer les séries solutions du polynôme F en un point régulier, a une complexité arithmétique quasi linéaire en le nombre de termes calculés (voir le théorème 8). Étant donné que cet algorithme peut être utilisé sur $\mathbb{C}$, on peut éviter les problèmes d'extensions de corps et de croissance des coefficients, et ainsi chaque opération arithmétique a un coût constant. Finalement, nous ferons donc l'hypothèse que l'on dispose d'un algorithme qui calcule une approximation numérique des séries solutions de F et qui possède une complexité binaire quasi linéaire en le nombre de termes calculés.

Notons $p_1, \ldots, p_{s+1}$ les points où l'on va calculer les séries solutions de F, et $x_1 = e_0, e_1, \ldots, e_s$ ($e_i \in [p_i, p_{i+1}]$) les points de connexion. Comme dans [CC87a, CC90, vdH99, Mez07], nous choisissons ces points de telle manière que le nombre $\beta \in]0, 1[$ intervenant dans les ordres de troncation soit constant, c'est-à-dire tel que :

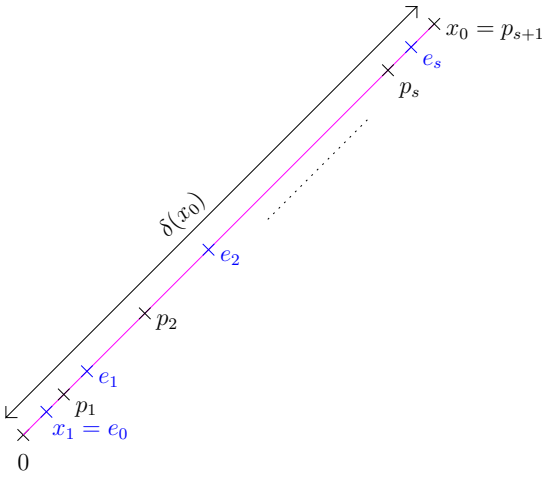

FIG. 3.24 – Points intermédiaires

- le point x_1 vérifie $|x_1| = \beta^e \delta(0)$ où e est le plus grand indice de ramification des séries de Puiseux au-dessus de 0,
- pour tout $1 \leq i \leq s$, on a $|p_i - e_i| = |p_i - e_{i-1}| = \beta |p_i|$.

On obtient donc $|p_{i+1}| = \frac{1}{1-\beta}|e_i|$, $|e_i| = (\beta+1)|p_i|$, et comme $|x_1| = \beta^e \delta(0)$:

$$|p_i| = \frac{(\beta+1)^{i-1}}{(1-\beta)^i}|x_1| = \frac{(\beta+1)^{i-1}\beta^e}{(1-\beta)^i}\delta(0)$$

$$\text{et } |e_i| = \left(\frac{\beta+1}{1-\beta}\right)^i |x_1| = \left(\frac{\beta+1}{1-\beta}\right)^i \beta^e \delta(0)$$

Finalement, on peut exprimer le nombre s en fonction de β : on veut que s vérifie $\frac{(\beta+1)^{s-1}}{(1-\beta)^s}|x_1| < |x_0| \leq \frac{(\beta+1)^s}{(1-\beta)^{s+1}}|x_1|$, ce qui nous conduit à :

$$s = \left\lceil \frac{\ln\left((1-\beta)\frac{|x_0|}{|x_1|}\right)}{\ln\left(\frac{\beta+1}{1-\beta}\right)} \right\rceil = \left\lceil \frac{\ln\left(\frac{1-\beta}{\beta^e}\frac{|x_0|}{\delta(0)}\right)}{\ln\left(\frac{\beta+1}{1-\beta}\right)} \right\rceil$$

Remarque 23. *La stratégie proposée ici consiste donc à « s'éloigner » du point critique* 0*. Avec un tel choix, on obtient généralement un point* p_{s+1} *tel que* $|p_{s+1}| > |x_0|$*. En pratique, on ajuste alors les points* e_s *et* p_{s+1} *de telle façon que* $p_{s+1} = x_0$ *et* $e_s \in [p_s, x_0]$*, comme c'est le cas sur la figure 3.24.*

Ensuite, si M est une borne supérieure pour $\sup_{x \in D(x_0, |x_0|)} S(x)$ et $\sup_{x \in D(0, \delta(0))} S(x)$, alors M majore aussi $\sup_{x \in D(p_i, |p_i|)} S(x), 1 \leq i \leq s+1$, puisque $D(p_i, |p_i|) \subset$

$D(x_0, |x_0|)$ pour $1 \leq i \leq s+1$. De même, on peut prendre pour valeur η la plus petite des précisions nécessaires sur l'ensemble des fibres $\mathcal{F}(e_i)$. On obtient ainsi deux valeurs M et η (voir la proposition 34) qui ont la même valeur pour tous les points intermédiaires considérés.

Finalement, sous l'hypothèse que l'on utilise un algorithme de calcul des développements en série de F qui a une complexité binaire quasi-linéaire, nous sommes ramenés à minimiser le nombre de pas multiplié par le coût de chaque pas (qui est constant dans notre stratégie), c'est-à-dire à minimiser la fonction suivante :

$$\begin{aligned} C(\beta) &= \left(\frac{\ln\left(\frac{\eta}{M}\right) + \ln(1-\beta)}{\ln(\beta)} - 1\right)\left(\frac{\ln\left(\frac{1-\beta}{\beta^e}\frac{|x_0|}{\delta(0)}\right)}{\ln\left(\frac{\beta+1}{1-\beta}\right)}\right) \\ &= f_1(\beta)\,(A + \ln(\beta) - \ln(1-\beta))\,(B + \ln(1-\beta) - e\ln(\beta)) \end{aligned}$$

avec $f_1(\beta) = -\frac{1}{\ln(\beta)\ln\left(\frac{\beta+1}{1-\beta}\right)}$, $A = -\ln\left(\frac{\eta}{M}\right) > 0$ et $B = \ln\left(\frac{|x_0|}{\delta(0)}\right) > 0$

Si A et B tendent vers l'infini, qui est le pire des cas (puisqu'il induit un coup important), la fonction $C(\beta)$ est équivalent à la fonction $f_1(\beta)$, qui prend son minimum en $\beta = \sqrt{2}-1$. Or, on a

$$C(\sqrt{2}-1) \simeq 1.2873004\ (A - 0.34657359)\,(B - 0.53479999 + 0.88137358\,e)\,.$$

La valeur $\beta = \sqrt{2}-1$ semble donc être la meilleure valeur à prendre. C'est cette valeur que nous utiliserons par la suite.

Nombre total de points intermédiaires

Nous revenons maintenant dans le cadre général de notre problème : on considère une demi-arête $[\alpha_{ij}, \alpha_i]$ de l'arbre $\mathcal{T}$. Si l'on remplace β par $\sqrt{2}-1$, $|x_0|$ par $\delta(\alpha_{ij})$ et 0 par α_i dans le nombre s de points intermédiaires trouvé dans la section précédente, on obtient un nombre de points intermédiaires à considérer pour la demi-arête $[\alpha_{ij}, \alpha_i]$ égal à :

$$\begin{aligned} s &= \left\lceil \frac{\ln(2-\sqrt{2}) - e\ln(\sqrt{2}-1) + \ln\left(\frac{\delta(\alpha_{ij})}{\delta(\alpha_i)}\right)}{\ln\left(\frac{\sqrt{2}}{2-\sqrt{2}}\right)} \right\rceil \\ &= e - 1 + \left\lceil \frac{\ln\left(\frac{2\delta(\alpha_{ij})}{\delta(\alpha_i)}\right)}{\ln(\sqrt{2}+1)} \right\rceil . \end{aligned}$$

On rappelle qu'avec un tel nombre de points intermédiaires, toutes les valeurs de β (voir la proposition 34) son inférieures ou égales à $\sqrt{2}-1$.

En additionnant le nombre de points intermédiaires obtenu pour chaque demi-arête, on obtient finalement, si n désigne le nombre de points critiques :

Proposition 37. *Le nombre de points de développement et d'évaluation nécessaire pour calculer le groupe de monodromie par notre méthode est en $O(n \ln \frac{L_{max}}{L_{min}} + g + d_y)$, où L_{max} et L_{min} sont respectivement les longueurs de la plus grande et la plus petite arête de $\mathcal{T}$, et g est le genre de la courbe $\mathcal{C}$.*

Démonstration. Pour chacune des demi-arêtes de l'arbre $\mathcal{T}$, on peut borner $\ln\left(\frac{2\delta(\alpha_{ij})}{\delta(\alpha_i)}\right)$ dans la valeur de s par $\ln \frac{L_{max}}{L_{min}}$. Étant donné qu'il y a n arêtes, on obtient le premier terme du résultat. D'après le lemme 21, il y a au plus 6 arêtes de $\mathcal{T}$ incidentes à α_i. Ainsi, lorsqu'on somme sur l'ensemble des demi-arêtes, on obtient une contribution bornée par $6\sum_{\mathfrak{P}}(e_{\mathfrak{P}}-1) = 12(g+d_y-1)$, où la somme est prise sur l'ensemble des places $\mathfrak{P}$ de $\mathcal{C}$ et la dernière égalité est la formule d'Hurwitz. Ceci nous donne le terme $(g+d_y)$. □

Corollaire 7. *Le nombre de points de développement et d'évaluation est en $O(D^2 \ln \frac{L_{max}}{L_{min}})$, où D est le degré total de la courbe $\mathcal{C}$.*

Démonstration. Le genre g et le degré du discriminant de F en y sont en $O(D^2)$ (voir respectivement [Mir95] et [vzGG99]). □

Ainsi, pour une famille de polynômes pour lesquels la quantité $\frac{L_{max}}{L_{min}}$ est bornée, le corollaire nous dit que le nombre total de points intermédiaires augmente linéairement en la taille de la sortie, qui est en $O(D^2)$. Finalement, on a :

Théorème 22. *Supposons que $F \in \mathbb{Z}[x,y]$ et notons $\|F\|_\infty$ la plus grande valeur absolue de ses coefficients. Alors le nombre de points de développements et d'évaluation est en $O(D^6 + D^5 \log \|F\|_\infty)$.*

Démonstration. Notons $R_F(x) \in \mathbb{Z}[x]$ le résultant en y de F et F_y. La quantité L_{max} peut être bornée par deux fois le rayon de n'importe quel disque contenant l'ensemble des points critiques. Choisissons $\|R_F\|_2$ pour une telle borne [MS99, exercice 132], où $\|\cdot\|_2$ est la norme euclidienne. Notons $S \in \mathbb{Z}[x]$ le produit des facteurs irréductibles de R_F dans $\mathbb{Z}[x]$. Puisque S est un diviseur de R_F dans $\mathbb{Z}[x]$ de degré n, on a :

$$\|S\|_2 \leq 2^n (n+1)^{\frac{1}{2}} \|R_F\|_2 \leq 2^q (q+1)^{\frac{1}{2}} \|R_F\|_2,$$

où $q = d_x(2d_y+1)$ majore le degré de S et de R_F. Pour L_{min}, on peut utiliser la borne inférieure $\mathrm{sep}(S)$ de [MS99]. Comme le polynôme S est supposé à coefficients dans $\mathbb{Z}$, on obtient :

$$\begin{aligned}\frac{L_{max}}{L_{min}} &\leq 2^{q(q-1)} q^{(q+2)/2} (q+1)^{\frac{q-1}{2}} \|R_F\|_2^q \\ &\leq 2^{q(q-1)} q^{(q+2)/2} (q+1)^{q+\frac{1}{2}} \|R_F\|_\infty^q.\end{aligned}$$

On applique ensuite le lemme 10 page 84 du chapitre 2, , puis on obtient le résultat à partir du corollaire 7 en majorant d_x et d_y par D. □

Dans la preuve de ce théorème, nous avons considéré S comme un facteur quelconque de R_F. Il serait intéressant d'avoir une borne plus fine pour $\|S\|_\infty$, qui ne dépendrait pas exponentiellement du degré n, car cela permettrait d'obtenir un résultat en $O(D^5)$ au lieu de $O(D^6)$.

3.6 Conclusion et Perspectives

Nous avons décrit dans ce chapitre un nouvel algorithme pour calculer le groupe de monodromie d'un revêtement $(\mathcal{C}, x)$. Celui-ci minimise la longueur totale des chemins parcourus dans le plan complexe en utilisant un arbre de recouvrement minimal pour la distance euclidienne de l'ensemble des points critiques. Nous avons notamment montré comment obtenir des chemins suivant cet arbre homotopes à ceux utilisés dans [TT84]. Notre stratégie utilise des développements de Puiseux au-dessus des points critiques, obtenant ainsi les monodromies locales, ainsi que le prolongement analytique le long de chemins proches des points critiques. L'utilisation de développements en séries tronquées à des ordres contrôlés permet de certifier la connexion entre deux fibres successives. De plus, ces développements en série permettent d'obtenir des bornes sur les intégrales utilisées lors du calcul de l'application d'Abel [DP07]. De plus, nous donnons un compromis entre le nombre de pas intermédiaires et les ordres de troncation considérés. On obtient ainsi des bornes sur le nombre de points intermédiaires utilisés. Finalement, en utilisant l'algorithme symbolique-numérique décrit dans le chapitre 2 pour calculer une approximation numérique des développements de Puiseux, on obtient un nouvel algorithme symbolique-numérique pour calculer le groupe de monodromie du revêtement $(\mathcal{C}, x)$.

Ce travail reste ouvert : si des bornes sur les erreurs engendrées pour les coefficients des séries de Puiseux sont données, alors il est possible, sous réserve d'avoir une algorithmique numérique donnant des bornes d'erreur, de

certifier le résultat obtenu, ce que nous ne fournissons pas à l'heure actuelle. De plus, nous ne fournissons pas de complexité pour le calcul du groupe de monodromie. Néanmoins, les bornes sur le nombre total de points intermédiaires sont une première étape vers un tel résultat.

Enfin, nous avons programmé un prototype de cet algorithme sous Maple, mais ce dernier nécessite d'être raffiné, et dépend également de l'algorithme symbolique-numérique de calcul de développements de Puiseux. Pour résumer, il reste un travail important de programmation à effectuer.

Conclusion

Nous avons présenté dans cette thèse une nouvelle approche pour calculer une approximation numérique des développements de Puiseux au-dessus d'un point critique. Cet algorithme modulaire-numérique nous permet d'obtenir un nouvel algorithme symbolique numérique pour calculer le groupe de monodromie du revêtement $(\mathcal{C}, x)$. Des prototypes de ces algorithmes ont de plus été implanté en Maple.

Cette approche consistant à guider des calculs numériques à l'aide d'informations exactes obtenues par calculs modulaires est à notre connaissance nouvelle. Les expérimentations menées à l'aide de cette nouvelle version de l'algorithme de Newton-Puiseux laissent à penser que cette méthode est efficace et numériquement stable.

De plus, nous avons donné de nouvelles bornes de complexité pour la partie modulaire de notre algorithme de calcul de développements de Puiseux, améliorant significativement les bornes existantes.

Enfin, si l'on a à disposition une algorithmique numérique permettant de contrôler les erreurs numériques tout au long des calculs, il est possible de certifier l'algorithme de calcul de développement de Puiseux, puis celui de calcul de groupe de monodromie.

Néanmoins, ce travail n'est pas clos. Outre la certification qui n'est pas effectuée dans cette thèse, il y a de nombreux points à étudier et améliorer. Tout d'abord, la partie numérique de l'algorithme de Newton-Puiseux n'a pas été programmée à ce jour, et n'est donc pas non plus utilisée dans le prototype d'algorithme développé pour le calcul de groupe de monodromie.

Ensuite, dans cette thèse, nous avons suivi le schéma classique de l'algorithme Newton-Puiseux pour calculer les développements de Puiseux. Il apparaît pourtant que cet algorithme semble améliorable, notamment en calculant l'ensemble des séries de Puiseux en même temps plutôt que séparément. Ceci est un travail que nous avons commencé à étudier avec Grégoire

Lecerf et Joris van der Hoeven.

Un autre point intéressant pour des travaux ultérieurs serait d'adapter la stratégie modulaire-numérique proposée dans cette thèse au cas des équations différentielles. En effet, les séries singulières irrégulières solutions d'une équation différentielle linéaire utilisent des polygones de Newton ainsi que des polynômes caractéristiques. Il semble donc naturel de réfléchir à une telle adaptation. Malheureusement, cela ne parait pas trivial : en effet, dans le cadre des équations différentielles, il ne parait pas clair que l'on puisse avoir un critère de réduction aussi simple que celui utilisé quand l'on connaît le polynôme bivarié qui définit la courbre algébrique.

Pour finir, il reste à traiter tout ce qui concerne le contrôle numérique des erreurs, ceci afin d'obtenir un algorithme certifié, mais également pour pouvoir obtenir la complexité binaire de la partie numérique de notre algorithme, nécessaire pour obtenir une complexité binaire du calcul du groupe de monodromie.

Bibliography

[BCL+07] A. Bostan, F. Chyzak, G. Lecerf, B. Salvy, and E. Schost. Differential equations for algebraic functions. In C. W. Brown, editor, *ISSAC'07: Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 25–32. ACM Press, 2007.

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer, 1997.

[BCS07] A. Bostan, F. Chyzak, and B. Salvy. *D-finitude : algorithmes et applications*. Cours de l'École Jeunes Chercheurs Informatique Mathématique, 2007. `url = http://algo.inria.fr/EJCIM07/EJCIM07-LN.pdf`.

[BD79] F. Baldassarri and B. Dwork. On Second Order Linear Differential Equations with Algebraic Solutions. *Amer. J. Math.*, 101(1):42–76, 1979.

[Ber95] L. Bertrand. Computing a Hyperelliptic Integral Using Arithmetic in the Jacobian of the Curve. *Applicable Algebra in Engineering, Communication and Computing*, 6:275–298, 1995.

[BK86] E. Brieskorn and H. Knörrer. *Plane Algbebraic Curves*. Birkhaüser, 1986.

[BMMW00] J.-C. Birget, S. Margolis, J. Meakin, and P. Weil. PSPACE-complete problems for subgroups of free groups and inverse finite automata. *Theoretical Computer Science*, 242:247–281, 2000.

[BP94] D. Bini and V. Y. Pan. *Polynomial and Matrix Computations*, volume 1 of *Progress in Theoretical Computer Science*. Birkhäuser, Saarbrücken, 1994.

[Bro90] M. Bronstein. Integration of Elementary Functions. *Journal of Symbolic Computation*, 9(2):117–173, 1990.

[BS96] E. Bach and J. Shallit. *Algorithmic Number Theory: Efficient Algorithms*, volume 1. The MIT Press, 1996.

[CC86] D. V. Chudnovsky and G. V. Chudnovsky. On Expansion of Algebraic Functions in Power and Puiseux Series. I. *Journal of Complexity*, 2(4):271–294, 1986.

[CC87a] D. V. Chudnovsky and G. V. Chudnovsky. Computer assisted number theory with applications. In *Number theory (New York, 1984–1985)*, volume 1240 of *Lecture Notes in Mathematics*, pages 1–68. Springer, Berlin, 1987.

[CC87b] D. V. Chudnovsky and G. V. Chudnovsky. On Expansion of Algebraic Functions in Power and Puiseux Series. II. *Journal of Complexity*, 3(1):1–25, 1987.

[CC90] D. V. Chudnovsky and G. V. Chudnovsky. Computer Algebra in the Service of Mathematical Physics and Number Theory. In David V. Chudnovsky and Richard D. Jenks, editors, *Computers in mathematics*, volume 125 of *Lecture Notes in Pure and Applied Mathematics*, pages 109–232, New York, 1990. Marcel Dekker.

[CGTW95] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The singular value decomposition for polynomial systems. In *ISSAC '95: Proceedings of the 1995 international symposium on Symbolic and algebraic computation*, pages 195–207. ACM Press, 1995.

[Che51] C. Chevalley. *Introduction to the Theory of Algebraic Functions of One Variable*, volume 6 of *Mathematical Surveys*. AMS, 1951.

[Coh93] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1993.

[Com64] L. Comtet. Calcul pratique des coefficients de Taylor d'une fonction algébrique. *L'Enseignement Mathématique*, 2(10):267–270, 1964.

[CS98] E. Compoint and M. Singer. Relations linéaires entre solutions d'une équation différentielle (Linear Relations Between the Solutions of a Differential Equation). *Ann. Fac. Sci. Toulouse*, Série 6, Vol. 7, no. 4:659–670, 1998.

[CSTU02] O. Cormier, M. F. Singer, B. M. Trager, and F. Ulmer. Linear Differential Operators for Polynomial Equations. *Journal of Symbolic Computation*, 34(5):355–398, 2002.

[Dav81] J. H. Davenport. *On the integration of algebraic functions.* Lecture Notes in Computer Science, Springer-Verlag, New York., 1981.

[DDD85] J. Della Dora, C. Dicrescenzo, and D. Duval. About a New Method for Computing in Algebraic Number Fields. In *EUROCAL 85*. Springer-Verlag LNCS 204, 1985.

[DP07] B. Deconinck and M. S. Patterson. Computing the Abel Map. In *Physica D: Nonlinear Phenomena*, 237:3214-3232, 2008.

[DR79] B. Dwork and P. Robba. On natural radii of p-adic convergence. *Trans. Amer. Math. Soc.*, 256:199–213, 1979.

[DRSS95] D. Duval, M. Rybowicz, A. Salinier, and P. Senechaud. *Calcul de primitives de fonctions algébriques.* Cours de DEA, Limoges, 1995.

[DS98] B. Deconinck and H. Segur. The KP Equation with Quasiperiodic Initial Data. *Phys. D*, 123(1-4):123–152, 1998.

[Duv87] D. Duval. *Diverses questions relatives au calcul formel avec des nombres algebriques.* PhD thesis, Université de Grenoble, 1987. Thèse d'État.

[Duv89] D. Duval. Rational Puiseux Expansions. *Compositio Mathematica*, 70:119–154, 1989.

[DvH01] B. Deconinck and M. van Hoeij. Computing Riemann Matrices of Algebraic Curves. *Phys. D*, 152/153:28–46, 2001. Advances in Nonlinear Mathematics and Science.

[Eic66] M. Eichler. *Introduction to the Theory of Algebraic Numbers and Functions.* Academic Press, 1966.

[Enc95] M. J. Encarnación. Computing gcds of Polynomials over Algebraic Number Fields. *Journal of Symbolic Computation*, 20:299–313, 1995.

[FÖ7] M. Fürer. Faster integer multiplication. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 57–66, New York, NY, USA, 2007. ACM.

[For81] O. Forster. *Lectures on Riemann Surfaces.* Graduate Text in Mathematics. Springer Verlag, New-York, Berlin, 1981.

[Ful69] W. Fulton. Hurwitz schemes and irreducibility of moduli of algebraic curves. *Annals of Mathematics, 90:542–575*, 1969.

[Gan06] J. L. Ganley. *"minimum spanning tree", in Dictionary of Algorithms and Data Structures [online].* Paul E. Black, ed., U.S. National Institute of Standards and Technology, 17 July 2006.

[GAP07] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4.10*, 2007. `url = http://www.gap-system.org`.

[Gri95] D. Griffits. Series expansions of algebraic functions. In *Computational Algebra and Number theory*, pages 267–277. Kluwer Academics, Netherlands, 1995.

[HM87] J.-P. Henry and M. Merle. Complexity of Computation of Embedded Resolution of Algebraic Curves. In *Proceedings Eurocal 87*, number 378 in Lecture Notes in Computer Science, pages 381–390. Springer-Verlag, 1987.

[Kah72] W. Kahan. Conserving confluence curbs ill-condition. Technical Report 6, University of California, Berkeley, 1972.

[KT78] H. T. Kung and J. F. Traub. All algebraic functions can be computed fast. *J. ACM*, 25(2):245–260, 1978.

[LZ04] S. K. Lando and A. K. Zvonkin. *Graphs on Surfaces and Their Applications.* Number 141 in Encyclopaedia of Mathematical Sciences. Springer-Verlag, 2004.

[Mar67] A. I. Markushevich. *Theory of functions of a complex variable. Vol. III.* Revised English edition, translated and edited by Richard A. Silverman. Prentice-Hall Inc., Englewood Cliffs, N.J., 1967.

[Mez07] M. Mezzarobba. Génération automatique de procédures numériques pour les fonctions d-finies. Master's thesis, Master parisien de recherche en informatique, octobre 2007. `url = http://www.eleves.ens.fr/home/mezzarob/`.

[Mig92] M. Mignotte. *Mathematics for computer algebra.* Springer-Verlag, New York, 1992.

[Mir95] R. Miranda. *Algebraic Curves and Riemann Surfaces.* Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 1995.

[MS99] M. Mignotte and D. Stefanescu. *Polynomials, an Algorithmic Approach.* Discrete Mathematics and Theoretical Computer Science. Springer, 1999.

[Pan97] V. Pan. Solving Polynomials: Some History and Recent Progress. *SIAM Review*, 39(2):187–220, 1997.

[Pat07] M. S. Patterson. *Algebro-geometric algorithms for integrable systems.* PhD thesis, University of Washington, 2007.

[Pot07] A. Poteaux. Computing monodromy groups defined by plane algebraic curves. In *Proceedings of the 2007 International Workshop on Symbolic-numeric Computation*, pages 36–45, New-York, 2007. ACM.

[PR08] A. Poteaux and M. Rybowicz. On the good reduction of puiseux series and the complexity of newton-puiseux algorithm over finite fields. In *ISSAC'08: Proceedings of the 2008 International Symposium on Symbolic and Algebraic Computation*, pages 239–246. ACM Press, 2008.

[Ris69] R. H. Risch. The Problem of Integration in Finite Terms. *Transactions of the American Mathematical Society*, 139:167–189, 1969.

[Rob00] A. M. Robert. *A course in p-adic analysis*, volume 198 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 2000.

[Sch77] A. Schönage. Schnelle multiplikation von polynomen über körpern der charakteristik 2. *Acta Informatica 7*, pages 395–398, 1977.

[Sho] V. Shoup. Ntl: A library for doing number theory. `url = http://www.shoup.net/ntl/`.

[Sho05] V. Shoup. *A Computational Introduction to Number Theory.* Cambridge University Press, 2005.

[Sie03] C. Sievers. Algorithmen fur freie Gruppen. Diplomarbeit, TU Braunschweig, 2003.

[Smi70] B. T. Smith. Error Bounds for Zeros of a Polynomial Based Upon Gerschgorin's Theorems. *J. ACM*, 17(4):661–674, 1970.

[SS71] A. Schönage and V. Strassen. Schnelle multiplikation großer zahlen. *Computing 7*, pages 281–292, 1971.

[Tra84] B. M. Trager. *Integration of Algebraic Functions.* PhD thesis, Department of EECS MIT, 1984.

[TT84] C.L. Tretkoff and M.D. Tretkoff. Combinatorial Group Theory, Riemann Surfaces and Differential Equations. *Contemp. Math.*, 33:467–517, 1984.

[vdH99] J. van der Hoeven. Fast Evaluation of Holonomic Functions. *Theoret. Comput. Sci.*, 210(1):199–215, 1999.

[vdH01] J. van der Hoeven. Fast Evaluation of Holonomic Functions Near and in Regular Singularities. *Journal of Symbolic Computation*, 31(6):717–743, 2001.

[vdH02] J. van der Hoeven. Relax, but don't be too lazy. *JSC*, 34:479–542, 2002.

[vH94] M. van Hoeij. An Algorithm for Computing an Integral Basis in an Algebraic Function Field. *Journal of Symbolic Computation*, 18:353–363, 1994.

[Vol97] H. Volklein. *Groups as Galois Groups.* Cambrige Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 1997.

[vzGG99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, Cambridge, 1999.

[Wal78] R. J. Walker. *Algebraic Curves.* Springer Verlag, Berlin-New York, 1978.

[Wal99] P. G. Walsh. On the Complexity of Rational Puiseux Expansions. *Pacific Journal of Mathematics*, 188:369–387, 1999.

[Wal00] P. G. Walsh. A Polynomial-time Complexity Bound for the Computation of the Singular Part of an Algebraic Function. *Mathematics of Computation*, 69:1167–1182, 2000.

[WR76] P. J. Weinberger and L. P. Rothschild. Factoring Polynomial over Algebraic Number Fields. *Transactions on Mathematical Software*, 2:335–350, 1976.

[Zar81] O. Zariski. *Le problème des modules pour les branches planes.* Hermann, Paris, 1981.

[Zen03] Z. Zeng. A method computing multiple roots of inexact polynomials. In *ISSAC '03: Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 266–272, New York, NY, USA, 2003. ACM.

[Zen04] Z. Zeng. The approximate gcd of inexact polynomials part 1: a univariate algorithm. In *ISSAC '04: Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 320–327. ACM Press, 2004.

[Zen05] Z. Zeng. Computing multiple roots of inexact polynomials. *Mathematics of Computation*, 74:869–903, 2005.

Calcul de développements de Puiseux et application au calcul du groupe de monodromie d'une courbe algébrique plane

Dans cette thèse, nous présentons un nouvel algorithme symbolique-numérique pour calculer une approximation numérique des développements de Puiseux au-dessus des points critiques, que nous utilisons pour calculer le groupe de monodromie d'une courbe algébrique plane. Essentiellement, l'algorithme de calcul de développements de Puiseux utilise des calculs modulo un nombre premier p bien choisi pour obtenir des informations exactes sur les séries de Puiseux. Ensuite, nous décrivons comment calculer une approximation numérique de ces séries de Puiseux à partir de ces informations exactes. Nous étudions également la complexité de la partie symbolique de notre algorithme. Enfin, nous proposons un algorithme symbolique-numérique pour calculer le groupe de monodromie d'une courbe algébrique plane qui utilise ces développements de Puiseux.

Mots clés : Développements de Puiseux, Courbes algébriques, Corps finis, Complexité, Calculs symboliques-numériques, Groupe de monodromie.

Computing Puiseux expansions and application to the computation of the monodromy group of a plane algebraic curve

We present a new symbolic-numeric algorithm to compute numerical approximations of Puiseux expansions above critical points. We use this new algorithm to compute monodromy groups of plane algebraic curves. In essence, we compute numerical approximations of Puiseux expansions in the following way: computations modulo a well chosen prime number p are used to obtain the exact information required to guide floating point computations. We also give complexity bounds for the symbolic part of our algorithm. Then, we propose a new strategy to compute monodromy groups of plane algebraic curves using this numerical approximations of Puiseux expansions.

Key words : Puiseux expansions, Alebraic Curves, Finite Fields, Complexity, Symbolic-Numeric computations, Monodromy groups.