Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON n° 5668

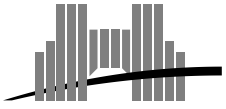
Pseudozéros et PGCD de polynômes en précision finie

Rapport du stage de
3 e année ENSIMAG – DEA MA

Stef GRAILLAT

sous la direction de
Philippe LANGLOIS

Version du 22 juin 2001 à 15h57



École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : lip@ens-lyon.fr

Résumé

Résumé

L' *algèbre numérique polynomiale* ou *calcul algébrique approché* a pour objectif de résoudre les problèmes algébriques pour des données (coefficients des polynômes par exemple) non plus exactes mais seulement connues avec une précision finie. Dans ce travail, nous étudions les zéros d'équations polynomiales, le PGCD de deux polynômes et leur primalité.

Les *pseudozéros* d'un polynôme sont les zéros des polynômes qui lui sont "proches". Nous prouvons un théorème pour calculer ces pseudozéros en norme quelconque et présentons quelques simulations numériques.

Nous donnons une formule explicite pour le problème du calcul du polynôme le plus proche d'un polynôme admettant une racine donnée.

Nous analysons et comparons plusieurs algorithmes permettant de calculer le PGCD approché de deux polynômes.

Enfin, nous montrons l'apport des pseudozéros dans l'étude de la primalité de deux polynômes approchés.

Mots-clés: pseudozéro, PGCD approché, précision finie, perturbation.

Abstract

The *numerical polynomial algebra* or *approximate algebraic computation* aims to solve algebraic problems with data not exact but given with a finite precision. In this work, we study the zeros of polynomial equations, the GCD of two univariate polynomials and the coprimeness.

The pseudozeros of a polynomial are the zero of polynomials near from it. We proof a theorem to compute the pseudozero set for each norm and we add some numerical simulations.

We give an explicit formula for the problem that is to compute the nearest polynomial with a given root.

We analyse and compare several algorithms allowing the computation of an approximate GCD of two univariate polynomials.

At last, we highlight how that the pseudozero set enable us to decide of the coprimeness of two approximate polynomials.

Keywords: pseudozero, approximate gcd, finite precision, perturbation.

Remerciements

Je tiens à remercier Jean-Michel MULLER de m'avoir accueilli au sein de son équipe.

Je remercie tout particulièrement Philippe LANGLOIS pour m'avoir fait découvrir un sujet passionnant et pour m'avoir guidé sur ce sujet à multiples facettes.

Merci aussi à Gilles VILLARD pour m'avoir conseillé à de nombreuses reprises et pour les éclairages qu'il m'a donnés sur le sujet.

Je remercie bien sûr tous les membres de l'équipe Arenaire pour l'accueil chaleureux et la bonne humeur qui y règne. Merci à Arnaud, Claire, David, Florent, Marc, Nathalie, Nicolas et Sylvie.

Pour finir, j'exprime aussi ma gratitude envers les personnes du secrétariat pour leur gentillesse et leur dévouement.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Notations

$:=$	définition.
$\mathbb{C}$	le corps des nombres complexes.
$\mathbb{R}$	le corps des nombres réels.
$\mathbb{K}$	un corps (habituellement $\mathbb{R}$ ou $\mathbb{C}$).
$\mathbb{K}[x]$	l'anneau de polynômes d'indeterminé x à coefficients dans $\mathbb{K}$.
$\mathbb{K}(x)$	le corps des fractions rationnelles d'indeterminé x à coefficients dans $\mathbb{K}$.
$\mathbb{K}[x, y]$	l'anneau de polynômes d'indeterminés x et y à coefficients dans $\mathbb{K}$.
$\mathbb{K}(x, y)$	le corps des fractions rationnelles d'indeterminés x et y à coefficients dans $\mathbb{K}$.
$\mathcal{M}_{n,m}(\mathbb{K})$	matrice de taille (n, m) sur le corps $\mathbb{K}$.
$\mathbb{P}_n$	polynôme à coefficients complexes de degré au plus n.
$\overline{z}$	le conjugé du nombre complexe z.
$\underline{z}$	le vecteur $(1, z, z^2, \ldots, z^n)$.
x^T	la transposé du vecteur x.
x^*	la transconjugé du vecteur x.
$\|\cdot\|_*$	la norme duale de $\|\cdot\|$: $\|y\|_* = \sup_{\|x\|=1} \|y^* x\|$.
$\|\cdot\|_p$	la norme p de Hölder : $\|x\|_p = (\sum_{i=1}^n \|x_i\|^p)^{1/p},\ 1 \leq p \leq \infty$.
$\mathbf{u}$	la précision machine.

Introduction

Les résultats d'expériences sont connus avec une incertitude comme par exemple celle due aux appareils de mesure ou bien à des approximations telles que celles induites par le codage des réels en flottants ou des calculs antérieurs. Ces valeurs incertaines apparaissent comme coefficients de fractions rationnelles en traitement du signal, CAO, etc. Le calcul de zéros et de PGCD de polynômes permet de les simplifier.

La plupart des problèmes algébriques sont mal posés : de petites perturbations sur les données du problème peuvent entraîner des résultats très différents. Il est donc nécessaire de redéfinir le problème de façon à le rendre bien posé. Nous verrons que c'est le cas pour le PGCD. Quant au calcul de la solution, il sera souvent nécessaire d'adapter l'algorithme dans le cas non perturbé.

Nous considérons ici trois problèmes liés : le calcul du polynôme le plus proche d'un polynôme donné ayant une racine donnée, du PGCD de deux polynômes et de leur primalité. Dans ce rapport, nous étudions comment la notion de pseudozéros peut permettre de répondre à ces trois problèmes.

Nous nous intéressons tout d'abord à la notion de *pseudozéros*. Il s'agit de l'ensemble des zéros des polynômes " proches " d'un polynôme donné. Nous regardons les méthodes de calcul de cet l'ensemble. Peu de travaux avaient été effectués sur le sujet [10, 13, 19]. Récemment, Stetter a remis au goût du jour les pseudozéros de polynômes à plusieurs variables pour la résolution de systèmes d'équations polynomiales [18].

L'étude des pseudozéros nous amène à chercher le polynôme le plus proche d'un autre polynôme ayant une racine donnée. En nous inspirant des travaux de Mosier [13] et de Trefethen et Toh [19], nous simplifions ceux de Kaltofen et Hitz [31, 32] et nous résolvons un problème posé par Hitz dans sa thèse.

Nous nous intéressons ensuite au calcul d'un PGCD de deux polynômes aux coefficients approchés. Cela nécessite une redéfinition du PGCD en un ϵ-PGCD. Nous proposons un état de l'art sur cette notion et nous analysons l'apport des pseudozéros. On s'intéresse ensuite à la primalité de deux polynômes. La primalité peut être déduite facilement si l'on possède un PGCD. Or le calcul d'un ϵ-PGCD est coûteux. C'est pourquoi on étudie des algorithmes plus simples pour tester la primalité.

Ces différents résultats sont organisés de la façon suivante.

1) Dans un premier chapitre, nous formalisons succinctement les problèmes algébriques approchés qui regroupent entre autres la notion de ϵ-pseudozéros et de ϵ-PGCD.

2) Dans un deuxième chapitre, nous introduisons la notion de *pseudozéros* d'un polynôme p. Nous proposons un algorithme de calcul de l'ensemble des pseudozéros d'un polynôme pour presque toutes les normes. Ce résultat est une contribution nouvelle au sujet. Nous présentons quelques optimisations de cet algorithme et une série de théorèmes caractérisant les pseudozéros. Enfin, des simulations numériques permettent d'expliquer les problèmes que l'on peut rencontrer lors de la recherche de zéros de polynômes par des méthodes numériques.

3) L'étude des pseudozéros nous amène à considérer le problème suivant : étant donnés un polynôme p et un complexe u, trouver le polynôme p_u le plus proche de p ayant u comme racine. En utilisant une technique introduite par Mosier [13] reprise ensuite par Trefethen et Toh [19], nous simplifions les travaux de Kaltofen et Hitz [31, 32, 33] : nous avons obtenu une formule explicite pour ce polynôme pour presque toutes les normes. Il s'agit là aussi d'une contribution nouvelle au sujet.

4) Nous introduisons ensuite le problème du calcul du PGCD approché de deux polynômes lorsque les coefficients sont des nombres complexes ou réels approchés. Un PGCD approché (ou ϵ-PGCD) est défini comme suit.
   Étant donnés deux polynômes p et q de degré respectif n et m, et ϵ un réel strictement positif, on appelle *ϵ-diviseur* (ou *diviseur approché*) de p et q tout diviseur des polynômes perturbés $\widehat{p}$ et $\widehat{q}$ vérifiant $\|p - \widehat{p}\| \leq \epsilon$, $\|q - \widehat{q}\| \leq \epsilon$ et $\deg(p - \widehat{p}) \leq n$, $\deg(q - \widehat{q}) \leq m$. Un *$\epsilon$-PGCD* de p et q est un *ϵ-diviseur* de degré maximum.
   Nous proposons un état de l'art sur les différentes méthodes existantes pour calculer un PGCD approché. Nous distinguons les méthodes à base d'optimisation, de divisions euclidiennes et de méthodes matricielles (SVD). Nous effectuons ensuite une étude géométrique du PGCD approché grâce à l'utilisation des pseudozéros.

5) Pour finir, nous regardons la primalité de deux polynômes à coefficients approchés. Il est clair qu'un calcul de PGCD approché permet de répondre à la question. Néanmoins, d'autres méthodes semblent plus rapides. Nous étudions d'abord un algorithme proposé à Beckermann et Labahn [23, 24], puis nous montrons que les tracés de pseudozéros permettent aussi de répondre simplement à cette question.

6) Enfin, nous concluons en récapitulant les principaux résultats obtenus et nous proposons des perspectives de travail futur.

Pour la recherche bibliographique, nous avons utilisé MathSciNet, la version électronique des Mathematical Reviews et le Zentralblatt-MATH, version électronique du Zentralblatt fur mathematik.

Pour la programmation, nous utilisons les logiciels MATLAB version 6 et MAPLE 6. Les simulations numériques ont été effectuées sur un Sun Ultra 5 (processeur UltraSparc cadencé à 333 MHz) muni de 256 Mo de RAM.

Chapitre 1

Calcul algébrique approché

Round numbers are always false.

— SAMUEL JOHNSON, *Boswell's Life of Johnson (1791)*

Nous formalisons dans ce chapitre ce que nous entendons par *calcul algébrique approché*. En effet, dans la plupart des problèmes de calcul scientifique, les données ne sont connues qu'avec une certaine précision. Les données provenant souvent d'observations de phénomènes physiques et mesurées par des appareils physiques, elles sont entachées d'erreurs liées à des phénomènes tels que la présence de bruit, l'erreurs de l'appareil de mesure ou bien l'approximation des nombres en flottant, etc.

En algèbre linéaire, de tels problèmes ont mené à la création d'une nouvelle branche de l'analyse numérique intitulée *algèbre linéaire numérique*. Pour ce qui est des polynômes, très peu d'études ont été entreprises. Il conviendrait donc de créer une nouvelle discipline : *l'algèbre polynomiale numérique*.

1.1 Définition du problème

Considérons un problème $\mathcal{P}$ avec comme données d'entrée des éléments d'un ensemble $\mathcal{A} \subset \mathbb{C}^n$ et pour résultats des éléments d'un ensemble $\mathcal{Z} \subset \mathbb{C}^m$. Le problème est décrit par l'application $F : \mathcal{A} \to \mathcal{Z}$ qui associe à une donnée d'entrée $a \in \mathcal{A}$ le résultat $z \in \mathcal{Z}$ du problème $\mathcal{P}$.

Pour $a = (\alpha_1, \ldots, \alpha_n) \in \mathcal{A}$ et une tolérance ϵ, nous définissons un *ϵ-voisinage* de a noté $N_\epsilon(a)$ par

$$N_\epsilon(a) := \{\widehat{a} \in \mathcal{A} : \|\widehat{a} - a\| \leq \epsilon\}.$$

Vu les incertitudes sur les entrées, le résultat ne peut être un seul point de $\mathcal{Z}$ mais l'ensemble des résultats pour les éléments du voisinage de l'entrée (voir la figure 1.1). Ainsi, nous avons la définition suivante.

Définition 1.1.1. Une quantité $z \in \mathcal{Z}$ est une *ϵ-pseudosolution* du problème $\mathcal{P}$ si elle est le résultat $F(\widehat{a})$ de $\mathcal{P}$ pour des $\widehat{a} \in N_\epsilon(a)$. L'ensemble des pseudosolutions est alors noté $Z_\epsilon(a)$. On a donc

$$Z_\epsilon(a) := \{\widehat{z} \in \mathcal{Z} : \widehat{z} = F(\widehat{a}) \quad \text{pour} \quad \widehat{a} \in N_\epsilon(a)\}.$$

Définition 1.1.2. Pour un problème $\mathcal{P}$, l'application du problème $F : A \subset \mathcal{A} \to \mathcal{Z}$ et $\widehat{z} \in \mathcal{Z}$, on définit la variété suivante :

$$\mathcal{M}(\widehat{z}) := \{\widehat{a} \in A : F(\widehat{a}) = \widehat{z}\} \subset A.$$

On vérifie facilement le critère suivant.

Théorème 1.1.1.

$$\widehat{z} \in Z_\epsilon(a) \Leftrightarrow \mathcal{M}(\widehat{z}) \cap N_\epsilon(a) \neq \emptyset$$

Si la fonction F est différentiable, on peut alors calculer assez facilement le conditionnement du problème. Nous ne le ferons pas ici car dans la suite du rapport les fonctions F que nous aurons ne seront pas différentiables.

Ce formalisme général regroupe la notion de *pseudozéros* et de ϵ-*PGCD* que nous allons étudier ci-après.

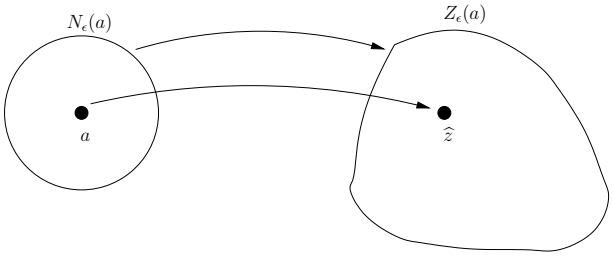


FIG. 1.1: *Analyse approchée*

1.2 Exemples

Soit p un polynôme de $\mathbb{P}_n = \mathbb{C}_n[z]$. On peut alors définir un ϵ-voisinage de ce polynôme par

$$N_\epsilon(p) = \{\widehat{p} \in \mathbb{P}_n : \|p - \widehat{p}\| \leq \epsilon\}.$$

On définit alors l'ensemble des ϵ-pseudozéros comme les pseudosolutions du problème de la recherche de zéro dans le cas de polynômes exacts :

$$Z_\epsilon(p) = \{z \in \mathbb{C} : \widehat{p}(z) = 0 \text{ pour } \widehat{p} \in N_\epsilon(p)\}.$$

De la même façon, on peut définir la notion de

- ϵ-diviseur (ou diviseur approché) : tout diviseur des polynômes proches de p et q notés $\widehat{p}$ et $\widehat{q}$ vérifiant $\deg \widehat{p} \leq n$, $\deg \widehat{q} \leq m$ et $\max(\|p - \widehat{p}\|, \|q - \widehat{q}\|) \leq \epsilon$.
- ϵ-PGCD (PGCD approché) : un ϵ-diviseur de degré maximum.

Nous détaillerons tout cela dans les chapitres 2 et 4.

1.3 Analyse qualitative

Nous verrons dans le cas des pseudozéros que l'on obtient une image de l'ensemble des zéros comme le montre la figure 1.2. Il s'agit de l'ensemble des pseudozéros du polynôme de Wilkinson, *i.e.* le polynôme unitaire ayant pour racines $1, 2, \ldots, 20$. Nous verrons par la suite que l'ensemble des pseudozéros de la figure 1.2 nous permet de dire qu'une incertitude de 2^{-23} nous empêche de distinguer les zéros de 10 à 20. On ne peut pas dès lors faire une analyse précise des résultats obtenus. Nous faisons plutôt une analyse qualitative du problème.

Notre objectif est de montrer que les pseudozéros permettent néanmoins de résoudre des problèmes comme par exemple la primalité de deux polynômes approchés.

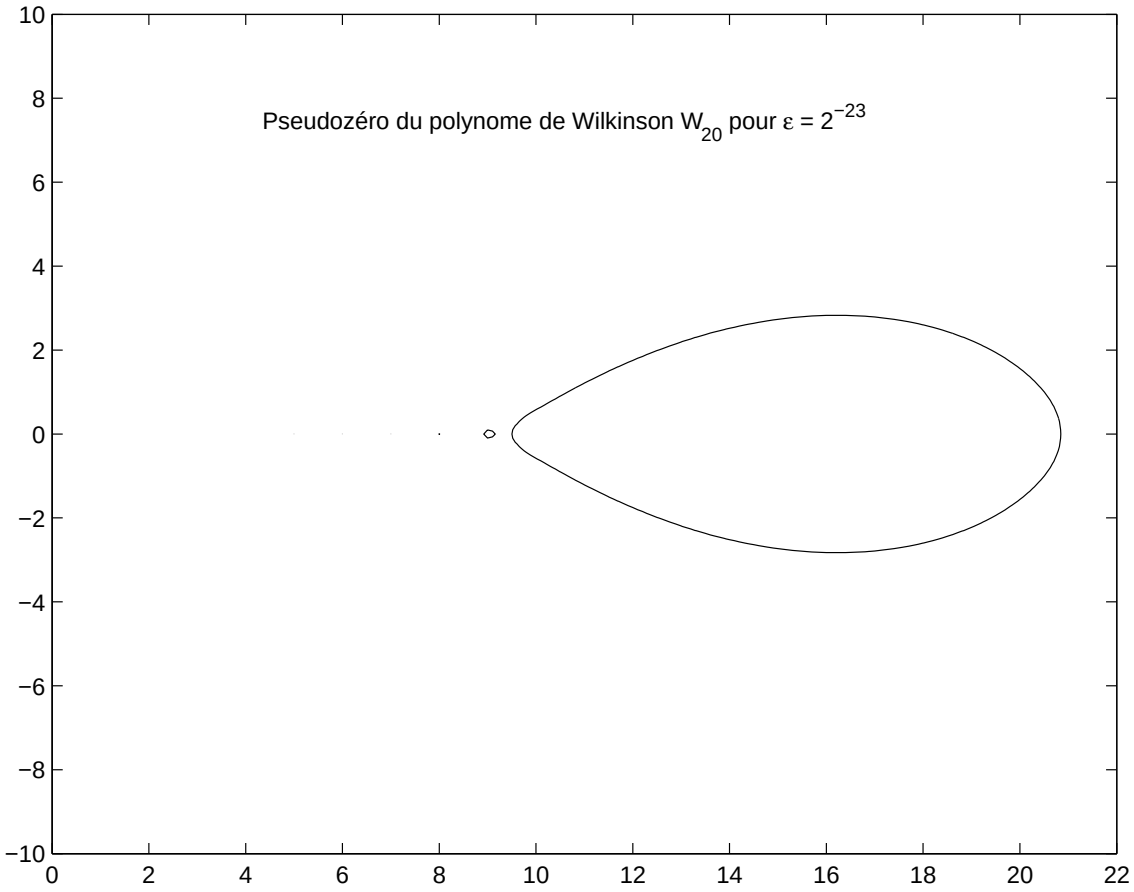


FIG. 1.2: *Exemple de tracé de pseudozéros .*

Chapitre 2

Étude des pseudozéros

The shortest path between two truths in the real domain passes through the complex domain.

— JACQUES HADAMARD, *Quoted in The Mathematical Intelligencer, volume 13, no. 1, Winter 1991*

2.1 Introduction

L'étude des racines d'un polynôme est un problème récurrent en mathématiques et qui trouve son origine dans l'antiquité (voir [15] pour un historique). En effet, la connaissance des racines permet entre autre la factorisation du polynôme ou bien le calcul du PGCD de deux polynômes.

Or Galois ayant montré que pour des polynômes de degré supérieur ou égal à 5, on ne peut trouver les racines par radicaux, un calcul numérique s'impose alors afin de déterminer une approximation de ces racines.

Le problème de la précision finie en arithmétique des ordinateurs intervient alors. En effet, les coefficients étant codés comme des nombres flottants, il y a une perturbation des coefficients en passant en arithmétique flottante. Celle-ci peut influer de manière importante sur les racines du polynôme.

Trois notions permettent d'étudier l'influence de la précision finie sur les racines :

- le nombre de conditionnement et la théorie des perturbations linéaires [5, 10] ;
- les pseudozéros [10, 11, 13, 18, 19, 22] ;
- le pseudospectre de la matrice compagnon [19, 20], notion issue plus particulièrement de l'algèbre linéaire.

L'intérêt des pseudozéros vient du fait qu'ils permettent d'étudier de manière géométrique la sensibilité des racines du polynôme par rapport aux perturbations de ses coefficients et donc se prêtent bien à une représentation graphique.

Dans l'étude qui suit, nous nous sommes principalement appuyés sur les articles [10, 11, 13, 18, 19, 22]. Les autres articles et ouvrages de la bibliographie sur les pseudozéros ont été consultés plus succinctement.

Dans la section suivante, nous introduisons les premières notations et étudions les différentes mesures sur les perturbations. Puis, nous définissons de façon rigoureuse la notion de pseudozéros pour énoncer quelques propriétés. Ensuite, nous exposons l'algorithme que nous avons utilisé pour calculer ces ensembles et énonçons quelques théorèmes relatifs au nombres de racines dans les composantes connexes d'un ensemble de pseudozéros. Enfin, nous nous intéressons au problème classique de la recherche des multiplicités de racines.

2.2 Les perturbations

2.2.1 Choix du type de perturbations

Soit à trouver les racines du polynôme $p \in \mathbb{P}_n = \mathbb{C}_n[z]$ (ensemble des polynômes à coefficients dans $\mathbb{C}$ de degré au plus n)

$$p(z) = p_n z^n + \cdots + p_1 z + p_0. \tag{2.1}$$

Soit $\|\cdot\|$ une norme sur $\mathbb{P}_n$, on peut alors définir sur $\mathbb{P}_n$ une topologie en définissant le ϵ-voisinage de p par

$$N_\epsilon(p) = \{\widehat{p} \in \mathbb{P}_n : \|p - \widehat{p}\| \leq \epsilon\}.$$

On parle alors de perturbations non structurées. Nous discuterons plus loin les différents choix de norme pour $\|\cdot\|$.

Remarque. On peut affaiblir cette définition en prenant non plus une norme sur $\mathbb{P}_n$ mais une distance. □

Notons $Z(p)$ l'ensemble des racines de p et pour tout $\epsilon > 0$, définissons l'ensemble des ϵ-pseudozéros par

$$Z_\epsilon(p) = \{z \in \mathbb{C} : z \in Z(\widehat{p}) \text{ pour } \widehat{p}\},$$

où $\widehat{p} \in N_\epsilon(p)$. De tels ensembles ont été étudiés par Mosier [13] puis par Trefethen et Toh [19].

On considère en fait que le polynôme $\widehat{p}$ est le polynôme dont les coefficients sont des perturbations de ceux de p. Par conséquent, pour définir $\widehat{p}$, il nous faut définir les perturbations des coefficients. On distingue classiquement deux types de perturbations :

- les perturbations non structurées,
- les perturbations structurées.

Les perturbations structurées permettent de choisir la forme du polynôme perturbé. Un type générique est

$$\widehat{p}(z) = \sum_{i=0}^{n} (p_i + \epsilon f_i(\epsilon)) z^i,$$

où les $(p_i)_{i=0,\ldots,n}$ sont les coefficients de p et où les fonctions f_i appartiennent à $\mathbb{C}[z]$.

Un type plus simple de perturbations structurées semblent être les perturbations dites *linéaires* de la forme

$$\widehat{p}(z) = p(z) + \epsilon g(z),$$

où g est un polynôme de degré inférieur ou égal à celui de p donnant la structure de la perturbation.

Nous n'avons pas trouvé de référence bibliographique traitant des perturbations structurées de polynômes. Il nous semble intéressant d'étudier ce type de perturbations en vue de son utilisation pour les PGCD approchés.

2.2.2 Choix des normes sur les perturbations

En suivant [10], on distingue surtout deux types de perturbations non structurées :

- les perturbations *normwise*,
- les perturbations *componentwise*.

Les perturbations *normwise*

Soit p défini par la relation (2.1) et $\widehat{p}$ un polynôme perturbé de p. On définit la norme *normwise* par

$$\|p - \widehat{p}\|^{\mathcal{N}} = \frac{\|p - \widehat{p}\|}{\beta},$$

où $\|\cdot\|$ est une norme sur les polynômes et β est un réel quelconque. On prendra souvent $\beta = \|p\|$ afin d'avoir une norme relative.

Remarque. Un *scaling* (cf [10, p. 51]) sur la norme est utilisé dans [19]. Cela consiste à définir la norme $\|x\|_d = \|Dx\|_2$ ou D est une matrice diagonale $(d_0, \ldots, d_n)$. Dans ce cas $\|p\|_d = \left(\sum_{i=0}^{n} |d_i|^2 |a_i|^2\right)^{1/2}$. Cela revient à prendre la norme *normwise* avec $\beta = 1$ et $\|\cdot\| = \|\cdot\|_d$. □

Les perturbations *componentwise*

En utilisant les mêmes notations que précédemment, on définit la norme *componentwise* par

$$\|p - \widehat{p}\|^{\mathcal{C}} = \max_i \frac{|p_i - \widehat{p}_i|}{f_i},$$

où les $(f_i)_{i=0,\ldots,n}$ sont des réels positifs. En général, on prend $f_i = |p_i|$ afin d'avoir une norme relative.

Remarque. Il s'agit de la norme utilisée par Mosier [13]. □

En conclusion, les perturbations *componentwise* permettent de "contrôler" les perturbations des coefficients alors que les perturbations *normwise* permettent un "contrôle" global sur le vecteur des coefficients.

2.3 Étude et calcul des ϵ-pseudozéros

Après avoir défini les types de perturbations utilisées et les normes associées, nous pouvons définir rigoureusement la notion de ϵ-pseudozéros. On rappelle que l'on définit le ϵ-voisinage de p par

$$N_\epsilon^{\mathcal{T}}(p) = \left\{\widehat{p} \in \mathbb{P}_n : \|p - \widehat{p}\|^{\mathcal{T}} \leq \epsilon\right\}, \tag{2.2}$$

où $\mathcal{T}$ désigne soit $\mathcal{C}$ pour la norme *componentwise*, soit $\mathcal{N}$ pour la norme *normwise*. On peut alors définir l'ensemble des ϵ-pseudozéros par

$$Z_\epsilon^{\mathcal{T}}(p) = \left\{z \in \mathbb{C} : \widehat{p}(z) = 0 \text{ pour } \widehat{p} \in N_\epsilon^{\mathcal{T}}(p)\right\}. \tag{2.3}$$

Cette définition ne se prête pas au calcul car elle n'est pas constructive. Néanmoins, selon le type de perturbations (*normwise* ou *componentwise*), nous identifions une forme calculable de l'ensemble des ϵ-pseudozéros.

Dans le cas de perturbations *normwise*, on a le théorème suivant (voir [10]).

Théorème 2.3.1. *L'ensemble des pseudozéros en norme* normwise *vérifie*

$$Z_\epsilon^{\mathcal{N}}(p) = \left\{z \in \mathbb{C} : \frac{|p(z)|}{\|\underline{z}\|_* \beta} \leq \epsilon\right\}, \tag{2.4}$$

où $\underline{z} = (1, z, \ldots, z^n)$ *et* $\|\cdot\|_*$ *est la norme duale de* $\|\cdot\|$.

Mosier [13] utilise déjà cet ensemble avec la norme ∞. De la même façon, Trefethen et Toh [19] réutilisent cet ensemble avec la norme 2. Ici, nous généralisons leur résultat pour une norme quelconque moyennant le fait qu'elle vérifie $\|\overline{z}\| = \|z\|$. Cette hypothèse n'est tout de même pas très restrictive car vérifiée par toutes les normes usuelles sur $\mathbb{C}^{n+1}$.

Remarque. Nous identifierons par la suite $\mathbb{P}_n$ et $\mathbb{C}^{n+1}$. On peut même, si besoin est, identifier $\mathbb{P}_n$ et $\mathbb{C}^n$ en considérant les polynômes unitaires (coefficient dominant p_n égal à 1). Un polynôme sera donc vu soit comme un polynôme au sens classique du terme soit comme le vecteur de ses coefficients. □

Preuve : On rappelle que la norme duale $\|\cdot\|_*$ sur $\mathbb{C}^{n+1}$ est définie par

$$\|x\|_* = \max_{z\neq 0} \frac{|z^*x|}{\|z\|} = \max_{\|z\|=1} |z^*x|.$$

Si $z \in Z_\epsilon(p)$ alors il existe $\widehat{p} \in \mathbb{P}_n$ tels que $\widehat{p}(z) = 0$ et $\|p - \widehat{p}\|^{\mathcal{N}} = \|p - \widehat{p}\|/\beta \leq \epsilon$. En utilisant l'inégalité de Hölder généralisée (*i.e.* $|x^*y| \leq \|x\|\|y\|_*$), on a

$$|p(z)| = |p(z) - \widehat{p}(z)| = |\sum_{i=0}^{n}(p_i - \widehat{p}_i)z^i| \leq \|p - \widehat{p}\|\|\underline{z}\|_*.$$

Et par conséquent, on a bien $|p(z)| \leq \epsilon\|\underline{z}\|_*\beta$.

Pour montrer la réciproque, soit $u \in \mathbb{C}$ tel que $|p(u)| \leq \epsilon\|\underline{u}\|_*\beta$. Un résultat classique nous permet d'affirmer l'existence d'un vecteur $d = (d_i) \in \mathbb{C}^{n+1}$ de norme 1 vérifiant $d^*\underline{u} = \|\underline{u}\|_*$ ([5, p. 119] ou [7, p. 278]). Ce vecteur d est appelé le vecteur dual de $\underline{u}$. Définissons le polynôme r par

$$r(z) = \sum_{k=0}^{n} r_k z^k \text{ avec } r_k = \overline{d}_k,$$

et le polynôme p_u par

$$p_u(z) = p(z) - \frac{p(u)}{r(u)}r(z). \tag{2.5}$$

On peut vérifier que p_u est le polynôme le plus proche de p (au sens de la norme $\|\cdot\|$) ayant u comme racine (voir le chapitre 3).
Il est clair que $r(u) = d^*\underline{u} = \|\underline{u}\|_*$ et $p_u(u) = 0$. Donc

$$\|p - p_u\| = \frac{|p(u)|}{|r(u)|}\|r\| \leq \|\overline{d}\|\epsilon\beta.$$

Moyennant l'hypothèse que $\|\overline{d}\| = \|d\|$ (ceci étant vrai pour toutes les normes usuelles), et comme $\|d\| = 1$, on a

$$\|p - p_u\| \leq \epsilon\beta.$$

Ainsi $u \in Z_\epsilon(p)$. ■

De la même façon, pour les perturbations de type *componentwise*, nous avons le théorème suivant.

Théorème 2.3.2. *L'ensemble des pseudozéros en norme* componentwise *vérifie*

$$Z_\epsilon^{\mathcal{C}}(p) = \left\{ z \in \mathbb{C} : \frac{|p(z)|}{\sum_{i=0}^{n}|f_i||z|^i} \leq \epsilon \right\}. \tag{2.6}$$

Preuve : Il s'agit d'une démonstration analogue à celle faite précédemment. Néanmoins on pourra se reporter à [13] pour la preuve complète. ■

Remarque. Dans les deux théorèmes précédents, la forme générale de $Z_\epsilon(p)$ est $Z_\epsilon(p) = \{z \in \mathbb{C} : |g(z)| \leq \epsilon\}$ avec g fonction calculable. Dans la suite nous utiliserons g afin d'encapsuler les deux cas correspondants aux perturbations *normwise* et *componentwise*. □

A partir de ces théorèmes, nous pouvons alors calculer les pseudozéros. Pour cela, nous avons utilisé le logiciel MATLAB. L'algorithme utilisé est l'algorithme 2.1.

L'arrêt et la correction de l'algorithme est immédiat.
Étudions maintenant la complexité de cet algorithme. Notons L la longueur du carré et h le pas de discrétisation. L'évaluation de $g(u)$ nécessite l'évaluation d'un polynôme, ce qui se fait en $\mathcal{O}(n)$, le calcul de la norme d'un vecteur, dont la complexité dépend de la norme. Notons $\mathcal{O}(\|\cdot\|_*)$ cette complexité. La

Algorithme 2.1 Calcul de pseudozéros

Entrée : le polynôme p et la perturbation ϵ
Sortie : le pseudozéro tracé dans le plan complexe

1: On maille un carré contenant toutes les racines de p à l'aide de la commande MATLAB `meshgrid`.
2: On calcule $g(z)$ pour tous les points z de la grille.
3: On affiche la ligne de niveau $|g(z)| = \epsilon$ à l'aide de la commande MATLAB `contour`.

complexité de l'algorithme précédent est donc en $\mathcal{O}((L/h)^2(n + \|\cdot\|_*))$.

Nous avons testé notre programme sur les polynômes suivants qui sont utilisés dans les articles [13, 19, 22].

- le polynôme de Wilkinson $W_{20}(z) = \Pi_{k=1}^{20}(z-k)$;
- le polynôme de Wilkinson tronqué $W_{10}(z) = \Pi_{k=1}^{10}(z-k)$;
- le polynôme $E(z) = \sum_{k=0}^{20} z^k/k!$;
- le polynôme $U(z) = z^{20} + z^{19} + \cdots + z + 1$;
- le polynôme unitaire ayant comme zéros $2^{-10}, 2^{-9}, \ldots, 2^9$.

La figure 2.1 représente l'ensemble des pseudozéros du polynôme de Wilkinson W_{20} en norme *componentwise* en ne perturbant que le coefficient de z^{19} avec une perturbation $\epsilon = 2^{-23}$.

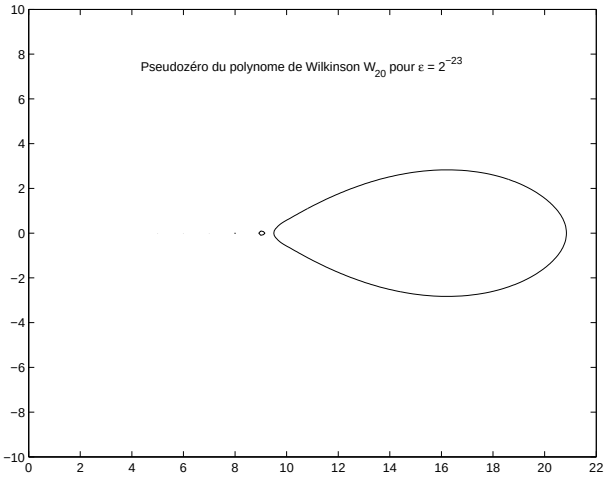


FIG. 2.1: *Pseudozéros du polynôme de Wilkinson W_{20} pour $\epsilon = 2^{-23}$*

Remarque. Nous effectuons ici quelques remarques concernant l'effet de la *précision finie* sur les pseudozéros. En effet, deux questions se posent principalement. D'une part, comment peut-on définir la grille *a priori* de façon à ce qu'elle contienne toutes les racines et que sa finesse permette une bonne précision des pseudozéros ? D'autre part, le calcul des pseudozéros revient à évaluer un polynôme. Or naturellement, cette évaluation est entachée d'une erreur. Quelles sont alors les conséquences de cette erreur sur les pseudozéros ?

Pour ce qui est du choix de la grille, nous n'avons rien trouvé dans la littérature. Il semble que le choix de la grille soit fait de manière non automatique par tâtonnements successifs. Il nous semble que l'on peut automatiser ce choix par exemple en estimant les racines du polynôme par la méthode de Newton. Néanmoins, cette méthode est coûteuse. Il semble intéressant donc de diminuer la taille de la

grille en localisant de manière plus simple les racines. Soit p un polynôme unitaire de degré n et $\{z_i\}$ l'ensemble de ses n racines. Notons $r = \max_{i=1,\dots,n} |z_i|$. On peut alors montrer [9, p. 154] que

$$r \leq \max\{1, \sum_{k=1}^{n} |p_k|\}.$$

Cela va nous permettre de définir un carré contenant l'ensemble des pseudozéros. En effet, soit $z \in Z_\epsilon(p)$. On sait alors qu'il existe $\widehat{p} \in N_\epsilon(p)$ tel que $\widehat{p}(z) = 0$. Le complexe z étant une racine de $\widehat{p}$, il est donc inférieur à la plus grande racine de $\widehat{p}$. Par conséquent, on a l'inégalité

$$|z| \leq \max\{1, \sum_{k=1}^{n} |\widehat{p}_k|\}.$$

Supposons que l'on soit dans le cas d'une perturbation en norme p de Hölder $\|\cdot\|_p$. On sait alors que $\|p - \widehat{p}\|_p \leq \epsilon$. Or $\|\cdot\|_\infty \leq \|\cdot\|_p$. Donc $\|p - \widehat{p}\|_\infty \leq \epsilon$ ce qui se réécrit $|p_i - \widehat{p}_i| \leq \epsilon$ et ensuite $|\widehat{p}| \leq |p_i| + \epsilon$ pour tout $i = 1, \dots, n$. Ainsi

$$|z| \leq \max\{1, \sum_{i=1}^{n} |p_i| + n\epsilon\} =: R.$$

On déduit de ce qui précède que

$$Z_\epsilon(p) \subset B(0, R) \text{ boule fermée de centre } 0 \text{ et de rayon } R.$$

Nous pouvons alors définir la grille par $[-R, R] \times [-R, R]$.
Le problème avec cette méthode, c'est que si les coefficients du polynôme sont grands alors la grille est très grande même si les racines sont petites. Une solution pourrait être d'utiliser d'autres bornes sur les racines, voir de choisir les bornes en fonction de la taille des coefficients. Mais cela n'a pas encore été étudié.

Discutons maintenant la finesse de la grille. Si l'on veut un ensemble de pseudozéros, il faut que le choix de la grille permette d'isoler les racines de p. Le pas de la grille doit donc être choisi en conséquence. Il existe une grande quantité de résultats concernant la séparation des racines. Nous ne nous y sommes pas encore intéressés mais cela devrait faire partie d'un travail futur.

Intéressons nous maintenant au problème de précision finie. Nous avons vu précédemment que le calcul de pseudozéros revenait à l'évaluation en les points d'une grille d'une certaine fonction g définie par $g(z) = p(z)/f(z)$ ou p est un polynôme et f est une norme. Il se trouve que pour les normes usuelles, on a $f(z) \geq 1$ et que l'erreur numérique associée est négligeable. Seule l'erreur sur l'évaluation du polynôme p semble être à prendre en compte. Notons y l'évaluation en machine de $p(z)$, on peut alors montrer [5, p. 105]

$$|y - p(z)| \leq \mathbf{u} \sum_{i=0}^{n} |p_i||z|^i =: \eta,$$

ou $\mathbf{u}$ est la précision machine. La borne plus précise suivante est proposée par Kahan

$$|y - p(z)| \leq 8\mathbf{u} \sum_{i=0}^{n} |s_i z^i| \quad \text{avec} \quad s_i = \sum_{j=i}^{n} p_j z^{j-i}.$$

Il est donc clair que si on choisit une perturbation de taille ϵ avec $\epsilon < \eta$ alors les résultats obtenus n'ont plus aucun sens. □

2.4 Étude des perturbations linéaires

Nous reprenons ici l'étude des pseudozéros dans le cadre de perturbations dite *linéaires*. On perturbe le polynôme de la façon suivante

$$\widehat{p}(z) = p(z) + \epsilon q(z),$$

où g est un polynôme vérifiant $\deg q \leq \deg p$. On définit alors l'ensemble des pseudozéros $Z_\epsilon(p)$ par

$$Z_\epsilon(p) = \{z \in \mathbb{C} : \exists \eta \text{ avec } |\eta| \leq \epsilon \text{ vérifiant } p(z) + \eta q(z) = 0\}.$$

On peut calculer cet ensemble grâce à la proposition suivante.

Proposition 2.4.1. *L'ensemble des pseudozéros pour une perturbation linéaire vérifie*

$$Z_\epsilon(p) = \{z \in \mathbb{C} : \Big|\frac{p(z)}{q(z)}\Big| \leq \epsilon\}.$$

Preuve : Il s'agit juste de réordonner les termes dans la définition ci-dessus. ■

Remarque. Si $q(z) = 0$ alors deux cas sont possibles. Si z est une racine de p alors $p(z)/q(z) = 0$ et z appartient à l'ensemble $Z_\epsilon(p)$. Si maintenant $p(z) \neq 0$ alors alors $p(z)/q(z) = \infty$ et donc z n'est pas un pseudozéro. Il n'y a donc pas de problèmes dans le calcul moyennant le fait de considérer les égalités suivantes : $0/0 = 0$ et $1/0 = \infty$. □

2.5 Quelques théorèmes relatifs aux pseudozéros

Moyennant quelques hypothèses sur ϵ et sur le coefficient dominant de p (que l'on peut supprimer si on prend p sous forme unitaire) on peut affirmer que l'ensemble des pseudozéros est un ensemble borné. Nous supposerons dans la suite que ceci est réalisé.

Les théorèmes suivants sont dûs à Mosier [13]. Ils établissent certaines relations entre les racines du polynôme p et les racines des polynômes "proches" de lui (*i.e.* qui appartiennent à $N_\epsilon(p)$).

Théorème 2.5.1. *En se plaçant dans les hypothèses précédentes* (i.e. *l'ensemble des pseudozéros est borné), si $q \in N_\epsilon(p)$ alors q et p ont le même nombre de racines (en comptant les multiplicités) dans chaque composante connexe de $Z_\epsilon(p)$. De plus, il y a au moins une racine du polynôme p dans chaque composante connexe de $Z_\epsilon(p)$.*

Preuve : Nous retranscrivons la preuve se trouvant dans [13, p. 269].
Soit $z \in Z_\mu$ ou Z_μ est une composante connexe de $Z_\epsilon(p)$. Par définition, il existe $\widehat{p} \in N_\epsilon(p)$ tel que $\widehat{p}(z) = 0$. Notons maintenant m_μ le nombre de zéros de p se situant dans Z_μ (en comptant les multiplicités). Notons aussi $p_t(z) := (1-t)p(z) + t\widehat{p}(z)$ pour $t \in [0,1]$. On vérifie alors que

$$\|p_t - p\| = t\|p - \widehat{p}\| \leq t\epsilon \leq \epsilon \quad \text{car} \quad t \in [0,1].$$

Par conséquent $p_t \in N_\epsilon(p)$ pour tout $t \in [0,1]$. Or les coefficients de p_t sont des fonctions de t et il est bien connu que les racines d'un polynôme sont des fonctions continues dans leurs coefficients. Ainsi comme t varie de 0 à 1, les racines de p_t forment des chemins continus des racines de p_0 à celles de p_1. En effet, comme les composantes connexes de $Z_\epsilon(p)$ sont bornées et disjointes, les zéros ne peuvent "sauter" dans une autre composante connexe ou bien partir à l'infini. Les racines restent donc dans Z_μ. Donc $\widehat{p} = p_1$ admet lui aussi m_μ racines. ■

Le théorème suivant permet de donner une condition nécessaire et suffisante pour savoir si le voisinage d'une racine contient deux racines de p.

Théorème 2.5.2. *Un voisinage d'une racine de p contient deux racines de p si et seulement si il contient un $u \in \mathbb{C}$ comme racine double de p_u (défini en* (2.5) *dans la preuve du théorème* (2.3.1)*).*

Preuve : La preuve (assez technique) se trouve dans [13, p. 271]. ■

Nous allons maintenant introduire une notion de multiplicité pour une composante connexe de $Z_\epsilon(p)$ qui nous permettra par la suite de simplifier des définitions et des énoncés de théorèmes.

Définition 2.5.1. La *multiplicité* d'une composante connexe Z_μ de $Z_\epsilon(p)$ est le nombre commun de zéros dans Z_μ de tous les polynômes de $N_\epsilon(p)$.

2.6 Pseudozéros dans le cas de polynômes réels

Dans le cas ou le polynôme p est réel, *i.e.* $p \in \mathbb{R}_n[x]$, les polynômes perturbés sont eux aussi réels. On définit alors le voisinage de p par $N_\epsilon(p) := \{q \in \mathbb{R}[x] : \|p - q\| \leq \epsilon\}$.
Deux cas sont alors envisageables. Ou bien on cherche les pseudozéros réels, auxquels cas c'est quasiment identique à l'étude précédente, ou alors on cherche tous les pseudozéros complexes non réels.

Nous allons nous attacher au cas complexe. On définit alors l'ensemble des pseudozéros par $Z_\epsilon(p) := \{z \in \mathbb{C} : \widehat{p}(z) = 0 \text{ pour } \widehat{p} \in N_\epsilon(p)\}$.

Il est clair que $Z_\epsilon(p)$ est symétrique par rapport à l'axe des réels. En effet si $z \in Z_\epsilon(p)$ alors $\overline{z} \in Z_\epsilon(p)$ car si $\widehat{p}(z) = 0$ et $\widehat{p} \in \mathbb{R}_n[x]$ alors $\widehat{p}(\overline{z}) = \overline{\widehat{p}(z)} = 0$. On peut alors démontrer le théorème suivant.

Théorème 2.6.1. *Soit Z_μ une composante connexe de $Z_\epsilon(p)$ de multiplicité 1. Alors ou bien $Z_\mu \subset \mathbb{R}$ ou bien $Z_\mu \cap \mathbb{R} = \emptyset$.*

Preuve : Si Z_μ contient des réels et des complexes alors Z_μ doit être symétrique par rapport à l'axe des réels. Soit $z \in \mathbb{C}\backslash\mathbb{R}$ tel que $z \in Z_\mu$. Il existe donc $\widehat{p} \in N_\epsilon(p) \subset \mathbb{R}_n[x]$ tel que $\widehat{p}(z) = 0$. Par conséquent $\widehat{p}(\overline{z}) = 0$ et donc $\widehat{p}$ admet au moins deux racines dans Z_μ ce qui contredit le fait que Z_μ soit de multiplicité 1. ■

Corollaire 2.6.2. *Une composante connexe Z_μ de $Z_\epsilon(p)$ vérifiant $\emptyset \neq Z_\mu \cap \mathbb{R} \neq Z_\mu$ est de multiplicité au moins 2.*

Nous avons donc quelques résultats vis à vis des perturbations réelles. Néanmoins, nous n'avons pas réussi à obtenir une formule explicite permettant de calculer l'ensemble des pseudozéros.
Cependant, dans le cas des perturbations linéaires (section 2.4), le résultat obtenu dans le cas de perturbations complexes s'applique aussi pour les perturbations réelles.

2.7 Pseudozéros simultanés

Nous reprenons ici la terminologie de Stetter [18]. Soient $\{z_k\}_{k=1,\ldots,m}$ un ensemble de pseudozéros appartenant à des composantes connexes différentes. On peut se demander s'il existe un polynôme perturbé $\widehat{p}$ appartenant à $Z_\epsilon(p)$ vérifiant $p(z_k) = 0$ pour $k = 1, \ldots, m$.

Définition 2.7.1. Un ensemble $\{z_k\}$, $k = 1, \ldots, m$ est un ensemble de *ϵ-pseudozéros simultanés* si il existe $\widehat{p} \in N_\epsilon(p)$ tel que $\widehat{p}(z_k) = 0$ pour $k = 1, \ldots, m$.

Soit $\mathcal{E}$ la variété en $\alpha = (\alpha_j)$ définie par

$$\sum_{j=0}^{n} \alpha_j z_k^j + p(z_k) = 0 \quad \text{pour} \quad k = 1, \ldots, m. \tag{2.7}$$

Le théorème suivant énonce une condition nécessaire et suffisante pour qu'un ensemble $\{z_k\}_{k=1,\ldots,m}$ soit un ensemble de *ϵ-pseudozéros simultanés*.

Théorème 2.7.1. *Un ensemble $\{z_k\}_{k=1,\ldots,m}$ est un ensemble de* ϵ-pseudozéros simultanés *si et seulement si*

$$\delta := \min_{\alpha \in \mathcal{E}} \|\alpha\| \leq \epsilon. \tag{2.8}$$

Preuve : Montrons tout d'abord la condition suffisante. Soit $\beta = (\beta_j)_{j=1,\ldots,n} \in \mathcal{E}$ vérifiant $\|\beta\| \leq \epsilon$. On remarque alors que

$$\widehat{p}(z) = p(z) + \sum_{j=0}^{n} \beta_j z^j \in N_\epsilon(p).$$

En effet,

$$\|\widehat{p} - p\| = \|\sum_{j=0}^{n} \beta_j z^j\| = \|\beta\| \leq \epsilon.$$

Et , comme $\beta \in \mathcal{E}$, on a : $\widehat{p}(z_k) = 0$ pour $k = 1, \ldots, m$.

Pour la condition nécessaire, on suppose l'existence de $\widehat{p} \in N_\epsilon(p)$ vérifiant $\widehat{p}(z_k) = 0$ pour $k = 1, \ldots, m$.
Définissons $f(z) = \widehat{p}(z) - p(z) = \sum_{i=0}^{n} f_i z^i$. Comme $\widehat{p} \in N_\epsilon(p)$, on a $\|f\| \leq \epsilon$. De plus $\widehat{p}(z_k) = 0$ pour $k = 1, \ldots, m$. Par conséquent,

$$\sum_{j=0}^{n} f_j z^j + p(z_k) = 0 \quad \text{pour} \quad k = 1, \ldots, m.$$

Donc $f = (f_j) \in \mathcal{E}$ et $\delta \leq \epsilon$. ■

Remarque. Dans le cas particulier ou $n = m$, l'équation (2.7) admet une solution unique (sous réserve bien sûr que la matrice associée au système soit inversible). La variété étant réduite à un unique point, il suffit de calculer la distance de ce point à l'origine et de la comparer à ϵ. □

On peut aussi définir des ensembles de pseudozéros simultanés en restreignant les ensembles de pseudozéros. Supposons par exemple avoir fixé un pseudozéro z_1 d'un polynôme p. On peut restreindre l'ensemble des pseudozéros en disant qu'il s'agit des racines des polynômes perturbés $\widehat{p}$ vérifiant $\widehat{p}(z_1) = 0$.

Définition 2.7.2. Pour $z_1 \in Z_\epsilon(p)$, on définit l'ensemble

$$Z_\epsilon(p \mid z_1) := \{z \in \mathbb{C} : \exists \widehat{p} \in N_\epsilon(p) \text{ avec } \widehat{p}(z_1) = 0 \text{ et } \widehat{p}(z) = 0\}.$$

On peut alors énoncer le théorème suivant, dont la preuve est immédiate.

Théorème 2.7.2. *$z \in Z_\epsilon(p \mid z_1)$ si et seulement si z et z_1 sont des pseudozéros simultanés de p.*

2.8 Pseudozéros avec multiplicité

Définition 2.8.1. Un pseudozéro z de p est dit de *multiplicité* m si il existe un polynôme perturbé $\widehat{p}$ appartenant à $N_\epsilon(p)$ ayant z comme racine avec la multiplicité m.

On sait que z est une racine de multiplicité m de $\widehat{p}$ si et seulement si

$$\widehat{p}^{(k)}(z) = 0 \quad \text{pour} \quad k = 0, \ldots, m-1. \tag{2.9}$$

En notant que $\widehat{p}(x) = p(x) + \sum_{j=0}^{n} \alpha_j x^j$ et en utilisant (2.9), on obtient

$$\sum_{j=0}^{n-k} \binom{j}{k} \alpha_j z^{j-k} + \frac{1}{k!} p^{(k)}(z) = 0, \quad k = 0, \ldots, m-1. \tag{2.10}$$

L'équation (2.10) est l'équation d'une variété linéaire de $\mathbb{C}^n$ que nous noterons $\mathcal{E}^{(m)}(z)$.
Nous avons alors le corollaire suivant.

Corollaire 2.8.1. *$z \in \mathbb{C}$ est un ϵ-pseudozéro de multiplicité m si et seulement si $\mathcal{E}^{(m)}(z)$ vérifie* (2.8)*.*

2.9 Simulations numériques

Dans cette section, nous proposons quelques simulations numériques obtenues avec le package que nous avons écrit en MATLAB.

La figure 2.2 représente l'ensemble des pseudozéros du polynôme $p(z) = z^2 - (10.5 + 10.2i)z + (1.5 + i53.5)$ pour différentes valeurs de ϵ. Les perturbations du polynômes sont de type *componentwise* pondérées par $f_0 = 4,\ f_1 = 0.5,\ f_2 = 0.01$. On peut remarquer qu'avec une tolérance de 0.01, on ne peut

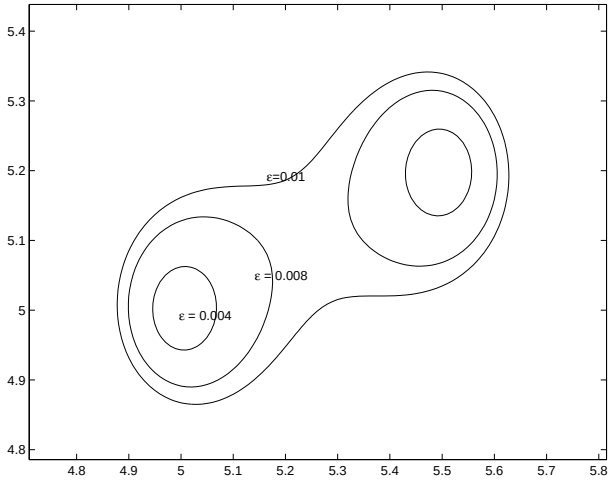


FIG. 2.2: *Ensembles des pseudozéros du polynôme* $p(z) = z^2 - (10.5 + 10.2i)z + (1.5 + i53.5)$ *pour trois valeurs différentes de* ϵ.

distinguer les deux racines de p qui sont $5 + i5$ et $5.5 + i5.2$. En diminuant la tolérance, on voit alors la séparation des deux racines.

La figure 2.3 montre l'évolution de l'ensemble des pseudozéros $Z_{\epsilon\|p\|_d}(p)$ du polynôme $p(z) = 1 + z + \cdots + z^{20}$ dont les racines sont les racines 20$^{\text{e}}$ de l'unité. Nous utilisons la norme $\|\cdot\|_d$ de Trefethen et Toh [19] (voir la remarque page 9) avec $d = \|p\|_2 p^{-1}$.

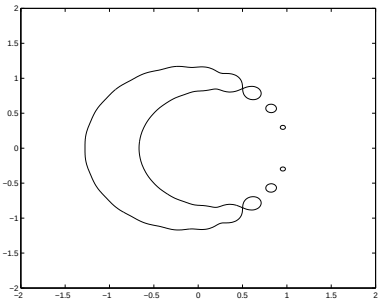
La figure 2.4 montre l'évolution de l'ensemble des pseudozéros $Z_{\epsilon\|p\|_d}(p)$ du polynôme p ayant pour racines $2^{-10}, 2^{-9}, \ldots, 2^9$ avec la même norme que précédemment. On voit très clairement la variation d'incertitude sur les racines en fonction de la tolérance ϵ.

La figure 2.5 montre d'une part les valeurs de $g(z)$ (voir remarque page 10) sur la grille que nous avons choisi et d'autre part l'ensemble des pseudozéros (en norme 2) $Z_\epsilon(p)$ pour $\epsilon = 0.1$ du polynôme $p(z) = (z - 1)(z - 2)$.

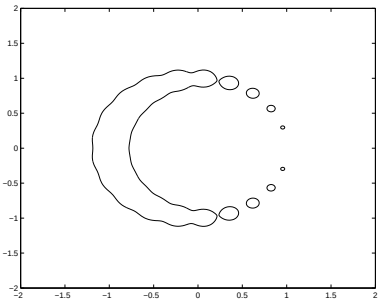
2.10 Conclusion sur les pseudozéros

Les pseudozéros semblent être un outil de choix dans l'étude des racines des polynômes en précision finie. Ils aident à la compréhension des problèmes numériques liés à la recherche de racines par l'algorithme de Newton par exemple.

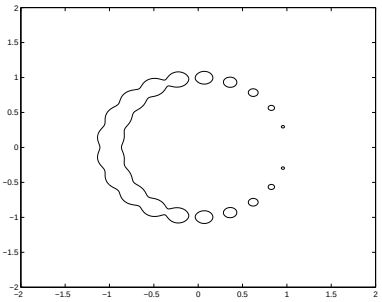
Mais l'intérêt principal des pseudozéros est qu'ils permettent de transposer les perturbations sur les coefficients des polynômes en des perturbations sur les racines.



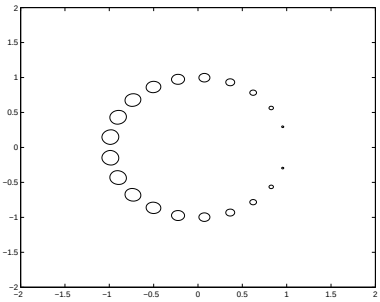
(a) $\epsilon = 10^{-1}$



(b) $\epsilon = 10^{-1.1}$

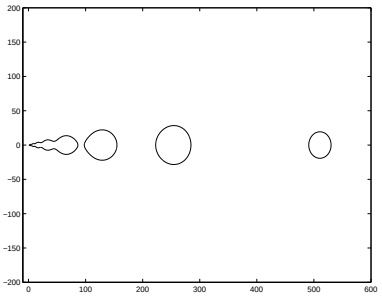


(c) $\epsilon = 10^{-1.2}$

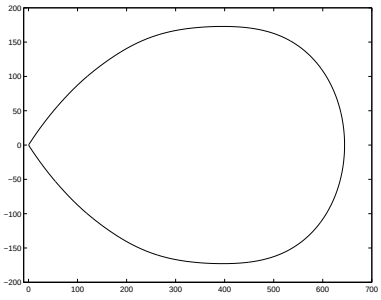


(d) $\epsilon = 10^{-1.35}$

FIG. 2.3: *Ensemble des pseudozéros pour diverses valeurs de ϵ du polynôme $p(z) = 1 + z + \cdots + z^{20}$.*

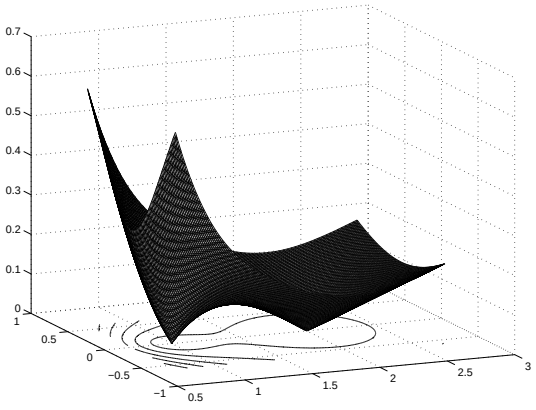


(a) $\epsilon = 10^{-3}$

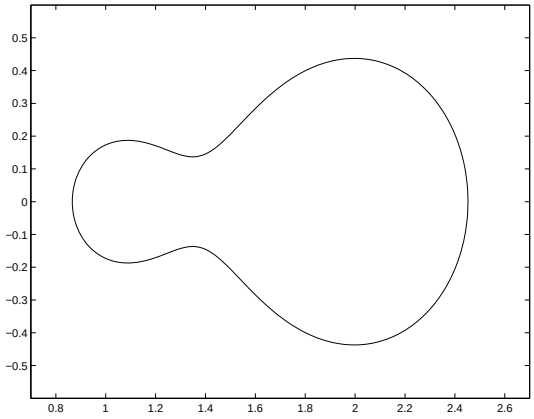


(b) $\epsilon = 10^{-2}$

FIG. 2.4: *Ensemble des pseudozéros pour diverses valeurs de ϵ du polynôme p ayant pour racine $2^{-10}, 2^{-9}, \ldots, 2^{9}$.*



(a) valeurs de $g(z)$ aux points de la grille



(b) coupe horizontale dans la figure (a)

FIG. 2.5: *Ensemble des pseudozéros pour* $\epsilon = 0.1$ *du polynôme* $p(z) = (z-1)(z-2)$.

Chapitre 3

Polynôme le plus proche ayant une racine donnée

I start by looking at a 2 by 2 matrix.
Sometimes I look at a 4 by 4 matrix.
That's when things get out of control and too hard.
Usually 2 by 2 or 3 by 3 is enough, and I look at them,
and I compute with them, and I try to guess the facts.

— PAUL R. HALMOS, *In Paul Halmos : Celebrating 50 Years of Mathematics (1991)*

Des travaux portant sur ce problème ont été effectués par Kaltofen et Hitz [31, 32, 33]. Ils utilisent pour cela une méthode de minimisation paramétrique introduite par Karmarkar et Lakshman [35, 36].

Ici, notre but est de résoudre le problème sans avoir besoin de techniques d'optimisations (souvent coûteuses). Pour cela, nous utilisons une technique introduite par Mosier [13] et réutilisée ensuite par Trefethen et Toh [19].

3.1 Les travaux de Kaltofen et Hitz

Dans leurs articles [31, 32, 33], Kaltofen et Hitz se sont intéressés au problème suivant :

Étant donné $p \in \mathbb{P}_n$ et $u \in \mathbb{C}$, trouver $p_u \in \mathbb{P}_n$ tel que $p_u(u) = 0$ et $\|p - p_u\|$ soit minimun.

Notons

$$p(z) = \sum_{k=0}^{n} p_k z^k.$$

Soit $\widetilde{p} \in \mathbb{P}_n$ tel que $\widetilde{p}(u) = 0$:

$$\begin{aligned}
\widetilde{p}(z) &= (z-u)\sum_{k=0}^{n-1} \widetilde{p}_k z^k, \\
&= \widetilde{p}_{n-1} z^n + (\widetilde{p}_{n-2} - u) z^{n-1} + \cdots + (\widetilde{p}_0 - u\widetilde{p}_1) z - u\widetilde{p}_0, \\
&= \widetilde{p}_{n-1} z^n + \sum_{k=1}^{n-1} (\widetilde{p}_{k-1} - u\widetilde{p}_k) z^k - u\widetilde{p}_0.
\end{aligned}$$

Notons $\delta = p - \widetilde{p}$ et

$$\begin{aligned} b &= [p_0, \ldots, p_{n-1}, p_n]^T \in \mathbb{C}^{n+1}, \\ w &= [\widetilde{p}_0, \ldots, \widetilde{p}_{n-1}]^T \in \mathbb{C}^n. \end{aligned}$$

Le problème revient donc à trouver δ vérifiant

$$\|\delta\| = \min_{w \in \mathbb{C}^n} \|Mw - b\|,$$

où

$$M = \begin{bmatrix} -u & & & \\ 1 & -u & & 0 \\ & \ddots & \ddots & \\ 0 & & 1 & -u \\ & & & 1 \end{bmatrix} \in \mathbb{C}^{(n+1)\times n}.$$

La méthode de résolution dépend alors de la norme utilisée. Dans le cas de la norme 2, il s'agit de minimiser une forme quadratique. Pour la norme ∞, il s'agit de programmation linéaire que l'on peut résoudre à l'aide de l'algorithme du *simplexe* (qui est de complexité exponentielle). Notre approche basée sur le vecteur dual nous permet aisément d'avoir une formule explicite pour presque toutes les normes, ce que ne permet pas l'approche de Kaltofen et Hitz.

3.2 Polynôme le plus proche ayant une racine donnée

Soit p de $\mathbb{P}_n$ et $u \in \mathbb{C}$ un complexe qui sera une racine du polynôme que l'on cherche. Le problème est alors le suivant :

> *Trouver un polynôme $p_u \in \mathbb{P}_n$ vérifiant $p_u(u) = 0$ et tel que s'il existe un polynôme $q \in \mathbb{P}_n$ avec $q(u) = 0$ alors on ait $\|p - p_u\| \leq \|p - q\|$.*

Notons $\underline{u} = (1, u, u^2, \ldots, u^n)$. On sait alors ([5, p. 119] ou [7, p. 278]) qu'il existe un vecteur $d \in \mathbb{C}^{n+1}$ vérifiant $d^* \underline{u} = \|\underline{u}\|$ et $\|d\| = 1$. Définissons alors les polynômes r et p_u par

$$\begin{aligned} r(z) &= \sum_{k=0}^{n} r_k z^k \quad \text{avec} \quad r_k = \overline{d}_k, \\ p_u(z) &= p(z) - \frac{p(u)}{r(u)} r(z). \end{aligned}$$

On peut alors montrer le théorème suivant.

Théorème 3.2.1. *Le polynôme p_u est le polynôme le plus proche de p au sens de la norme $\|\cdot\|$ ayant u pour racine.*

Remarque. On suppose dans le reste du chapitre que la norme $\|\cdot\|$ vérifie la condition suivante : pour tout vecteur $v \in \mathbb{C}^{n+1}$ on a $\|\overline{v}\| = \|v\|$. □

Preuve : Il est clair que $p_u(u) = 0$. Il nous reste à démontrer que p_u est le plus proche de p vérifiant cette égalité. Soit donc $q \in \mathbb{P}_n$ vérifiant $q(u) = 0$. En appliquant l'inégalité de Hölder, on a

$$|p(z) - q(z)| = |\sum (p_i - q_i) z^i| \leq \|p - q\| \|\underline{z}\|_*.$$

En appliquant l'inégalité précédente avec $z = u$, on obtient

$$|p(u)| \leq \|p - q\| \|\underline{u}\|_* \quad \text{et donc} \quad \frac{|p(u)|}{\|\underline{u}\|_*} \leq \|p - q\|. \tag{3.1}$$

On remarque de plus que

$$\|p - p_u\| = \frac{|p(u)|}{|r(u)|}\|r\|.$$

Or $\|r\| = \|\overline{d}\| = \|d\| = 1$ et $r(u) = d^*\underline{u} = \|\underline{u}\|_*$. Donc

$$\|p - p_u\| = \frac{|p(u)|}{\|\underline{u}\|_*}.$$

En reportant l'égalité précédente dans l'inéquation (3.1), on obtient $\|p - p_u\| \leq \|p - q\|$. ■

3.3 Calcul de p_u

Nous avons donc la forme générale de p_u. Il nous reste cependant à calculer ce polynôme, c'est-à-dire trouver le vecteur dual de $\underline{u}$. Le problème peut se formuler ainsi

Trouver $d \in \mathbb{C}^{n+1}$ *vérifiant* $d^*\underline{u} = \|\underline{u}\|_* \quad$ *et* $\quad \|d\| = 1.$

Ce problème n'ayant pas de solution explicite dans le cas d'une norme quelconque, nous allons faire le calcul pour les normes p de Hölder. Nous noterons par la suite $u = |u|e^{i\theta}$.

3.3.1 Cas de la norme $\|\cdot\|_\infty$

Le problème se réécrit sous la forme

Trouver $d \in \mathbb{C}^{n+1}$ *vérifiant*

$$\sum_{j=0}^{n} \overline{d}_j u^j = \sum_{j=0}^{n} |u|^j \quad \textit{et} \quad \max_{j=1,\dots,n} |d_j| = 1.$$

On remarque aisément que $d_j = e^{ij\theta}$ convient. Dans ce cas, p_u s'écrit

$$p_u(z) = p(z) - \frac{p(u)}{\sum_{j=0}^{n} |u|^j} \sum_{j=0}^{n} e^{-ij\theta} z^j.$$

Exemple 3.3.1. Prenons un exemple simple : $p(z) = z + 1$ et $u = 1/2$. Dans ce cas $\underline{u} = (1, 1/2)$ et $r(z) = z + 1$. En appliquent ce qui précède, on obtient $p_u(z) = 1 + z - ((1 + 1/2)/(1 + 1/2))(1 + z) = 0$. Justifions ce résultat. En effet, p_u est nécessairement de la forme $\alpha(z - 1/2)$, $\alpha \in \mathbb{C}$. Dans ce cas

$$\|p - \alpha(z - 1/2)\|_\infty = \max\{|1 - \alpha|, |1 + (1/2)\alpha|\} =: d.$$

Trois cas se présentent alors

- si $\alpha > 0$ alors $d = 1 + (1/2)\alpha > 1$;
- si $\alpha < 0$ alors $d = 1 - \alpha > 1$;
- si $\alpha = 0$ alors $d = 1$.

Le minimum est donc obtenu pour $\alpha = 0$.

3.3.2 Cas de la norme $\|\cdot\|_2$

Le problème se réécrit sous la forme

Trouver $d \in \mathbb{C}^{n+1}$ *vérifiant* $d^* \underline{u} = \|\underline{u}\|_2 \quad et \quad \|d\|_2 = 1.$

S'agissant de la norme euclidienne, on remarque aisément que $d = \underline{u}/\|\underline{u}\|_2$ convient. Les composantes d_j de d s'écrivent donc

$$d_j = \frac{|u|^j e^{ij\theta}}{\left[\sum_{j=0}^n |u^j|^2\right]^{1/2}}.$$

Le polynôme s'écrit alors

$$p_u(z) = p(z) - \frac{p(u)}{\left[\sum_{j=0}^n |u^j|^2\right]^{1/2}} \sum_{j=0}^n |u|^j e^{-ij\theta} z^j.$$

3.3.3 Cas de la norme $\|\cdot\|_1$

Le problème se réécrit sous la forme

Trouver $d \in \mathbb{C}^{n+1}$ *vérifiant*

$$\sum_{j=0}^n \overline{d}_j u^j = \max\{1, |u|, \ldots, |u|^n\} \quad et \quad \sum_{j=0}^n |d_j| = 1.$$

Il convient de distinguer deux cas. En effet, ou bien $|u| \leq 1$, auquel cas $\max\{1, |u|, \ldots, |u|^n\} = 1$ ou bien $|u| > 1$ et dans ce cas $\max\{1, |u|, \ldots, |u|^n\} = |u|^n$.

- $|u| \leq 1$
  On doit alors avoir $\sum_{j=0}^n \overline{d}_j u^j = 1$ et $\sum_{j=0}^n |d_j| = 1$. Prenons $d = (1, 0, \ldots, 0)$. On vérifie facilement que d convient. Le polynôme p_u s'écrit alors

  $$p_u(z) = p(z) - p(u).$$

- $|u| > 1$
  On doit alors avoir $\sum_{j=0}^n \overline{d}_j u^j = |u|^n$ et $\sum_{j=0}^n |d_j| = 1$. Prenons $d = (0, \ldots, 0, e^{in\theta})$. On vérifie facilement que d convient. Le polynôme p_u s'écrit alors

  $$p_u(z) = p(z) - \frac{p(u)}{|u|^n} e^{-in\theta} z^n.$$

3.3.4 Cas de la norme $\|\cdot\|_p$ ($2 < p < \infty$)

Le problème s'écrit alors sous la forme

Trouver $d \in \mathbb{C}^{n+1}$ *vérifiant*

$$\sum_{j=0}^n \overline{d}_j u^j = \|\underline{u}\|_p \quad et \quad \|\underline{u}\|_q = 1 \quad avec \quad \frac{1}{p} + \frac{1}{q} = 1.$$

Cela revient à résoudre en $d = (d_j)$ les équations suivantes :

$$\sum_{j=0}^n \overline{d}_j u^j = \left[\sum_{j=0}^n |u^j|^q\right]^{1/q}, \tag{3.2}$$

$$\left[\sum_{j=0}^n |d_j|^p\right]^{1/p} = 1. \tag{3.3}$$

Posons $d_j := \frac{|u^j|^{q-1}e^{ij\theta}}{\|\underline{u}\|_q^{q-1}}$. Vérifions alors que $d = (d_j)$ convient. Il est clair que d vérifie bien l'équation (3.2). Pour ce qui est de l'équation (3.3), on a

$$\begin{aligned}\Big[\sum_{j=0}^{n}|d_j|^p\Big]^{1/p} &= \frac{1}{\|\underline{u}\|_q^{q-1}}\Big[\sum_{j=0}^{n}|u^j|^{p(q-1)}\Big]^{1/p} = \frac{1}{\|\underline{u}\|_q^{q-1}}\Big[\sum_{j=0}^{n}|u^j|^q\Big]^{1/p},\\ &= \frac{1}{\|\underline{u}\|_q^{q-1}}\Big[\|\underline{u}\|_q^q\Big] = \frac{\|\underline{u}\|_q^{p/q}}{\|\underline{u}\|_q^{q-1}} = \|\underline{u}\|_q^{p/q-q+1} = \|\underline{u}\|_q^0 = 1.\end{aligned}$$

Donc le vecteur d est bien solution de notre problème. Le polynôme p_u s'écrit alors

$$p_u(z) = p(z) - \frac{p(u)}{\sum_{j=0}^{n}|u^j|^{q-1}}\sum_{j=0}^{n}|u^j|^{q-1}e^{-ij\theta}z^j.$$

3.4 Conclusion sur la méthode

La méthode de calcul du polynôme le plus proche ayant une racine donnée proposée ici est plus simple que la méthode proposée dans [31, 32, 33]. Elle donne des réponses explicites pour les normes p de Hölder. Il nous semble que le fait de posséder une formule explicite nous permettra de résoudre d'autres problèmes liés à la recherche de racines communes.

Chapitre 4

Étude de la notion de ϵ-PGCD

In teaching, writing and research,
there is no greater clarifier than
a well-chosen example.

— CHARLES F. VAN LOAN, *Using Examples to Build Computational Intuition (1995)*

4.1 Introduction

On va montrer dans cette introduction la limite de la définition du PGCD algébrique dans le domaine de la précision finie. Prenons par exemple deux polynômes p et q unitaires et tels que $\deg p > 1$. On suppose de plus que p divise q. Cela revient à dire que $\gcd(p, q) = p$. Or pour toute constante $\varepsilon > 0$, on a $\gcd(p + \varepsilon, q) = 1$. Par conséquent, une petite perturbation du polynôme p peut entraîner un « saut » important du PGCD. Le PGCD ne dépend donc pas continûment des perturbations de ses coefficients. Le calcul du PGCD est par conséquent un problème mal posé au sens d'Hadamard.

On a le même type de problèmes avec la notion de polynômes "premiers entre eux". L'exemple suivant issu de [24] est révélateur. Soient p et q les polynômes suivant

$$p(z) = (z - \frac{1}{3})(z - \frac{5}{3}) = z^2 - 2z + \frac{5}{9}, \quad q(z) = z - \frac{1}{3}.$$

Lorsque l'on transforme les coefficients de p et q en nombres flottants, les deux polynômes, qui ont une racine commune en arithmétique exacte, deviennent premiers entre eux. De la même façon, les polynômes

$$p(z) = 50z - 7, \quad q(z) = z - \frac{1}{7},$$

premiers entre eux en arithmétique exacte, ne le sont plus en considérant une précision de deux chiffres après la virgule ($1/7 = 0.14285714$ et $7/50 = 0.14$).

L'idée dans l'aspect approché est de ne plus considérer les polynômes comme des données exactes mais plutôt comme des boules dans l'espace des polynômes. On s'attend alors à ce que, après une légère perturbation des polynômes, on obtienne un PGCD approché proche du PGCD exact.

4.2 Historique

Le calcul du PGCD approché a été relancé en 1985 par A. Schönhage [42] sous le terme de "Quasi-GCD". Il suppose dans son étude qu'il peut avoir une approximation arbitrairement précise des réels. Autrement dit il travaille en précision infinie.

Pour $f \in \mathbb{P}_n$, on notera $\rho(f)$ le réel défini par $\rho(f) = \max_{z \text{ racine de } f} |z|$. Le problème qu'il résout est alors le suivant :

Étant donnés $\|\cdot\|_1$ la norme 1, $f \in \mathbb{P}_n$, $g \in \mathbb{P}_m$ avec $1 \leq m < n$, $\|f\|_1, \|g\|_1 \in [\frac{1}{2}, 1]$, $\rho(f) \leq \frac{1}{4}$ et étant donné $0 < \varepsilon \leq \frac{1}{2}$, trouver des polynômes h ,$u \in \mathbb{P}_{m-1}$, $v \in \mathbb{P}_{n-1}$ telle que

(i) $\|hf_1 - f\|_1 < \varepsilon$, $\|hg_1 - g\|_1 < \varepsilon$ *pour* f_1, g_1 *bien choisis*

(ii) $\|uf + vg - h\|_1 < \varepsilon \|h\|_1$

Le polynôme h est appelé un *Quasi-GCD* de p et q. Il résout ce problème en utilisant un algorithme d'Euclide modifié de façon à la rendre plus stable.

Mais le problème est que généralement, on ne peut avoir une précision infinie. En effet, dès que l'on travaille sur une machine, les réels sont approximés en nombres flottants et par conséquent les données que l'on code sur la machine sont entachées d'erreurs. La définition de PGCD approché a du évoluer. C'est ce que nous allons étudier dans les sections suivantes.

4.3 Définition du ϵ-PGCD

Il y a plusieurs façon de définir un ϵ-PGCD mais la plus couramment admise est la suivante.

Définition 4.3.1. Étant donnés deux polynômes p et q de degré respectif n et m, et ϵ un réel strictement positif, on appelle *ϵ-diviseur* (ou *diviseur approché*) de p et q tout diviseur des polynômes perturbés $\widehat{p}$ et $\widehat{q}$ vérifiant $\|p - \widehat{p}\| \leq \epsilon$, $\|q - \widehat{q}\| \leq \epsilon$ et $\deg(p - \widehat{p}) \leq n$, $\deg(q - \widehat{q}) \leq m$. Un *$\epsilon$-PGCD* de p et q est un ϵ-diviseur de degré maximum.

Remarque. Autrement dit, $\epsilon\text{-}\gcd(p, q) = \gcd(\widehat{p}, \widehat{q})$ pour $\widehat{p} \in \mathbb{P}_n$ et $\widehat{q} \in \mathbb{P}_m$ vérifiant $\|p - \widehat{p}\| \leq \epsilon$, $\|q - \widehat{q}\| \leq \epsilon$ et tel que $\deg \gcd(\widehat{p}, \widehat{q})$ soit maximal.
On remarquera aussi que la définition n'implique pas l'unicité du ϵ-PGCD. Il est clair que le degré du PGCD est unique mais en aucune manière le ϵ-PGCD. □

Il est classique de représenter un polynôme sous deux formes différentes :

- par ses racines, ou
- par ses coefficients.

4.4 Calcul d'un ϵ-PGCD : polynômes définis par ses racines

Commençons par le cas où le polynôme est représenté par ses racines. Nous reprenons ici l'étude de Pan [39]. Soient deux polynômes $p \in \mathbb{P}_n$ et $q \in \mathbb{P}_m$ unitaires ayant respectivement pour racines $(u_i)_{i=1\ldots n}$ et $(v_j)_{j=1\ldots m}$. Ils s'écrivent donc :

$$p(z) = \prod_{i=1}^{h} (z - u_i)^{\alpha_i},$$

et

$$q(z) = \prod_{j=1}^{k} (z - v_j)^{\beta_j}.$$

L'algorithme 4.1 nous permet de calculer ce PGCD.

L'arrêt et la correction de l'algorithme est immédiat.
Effectuons un calcul de complexité. L'étape 1 s'effectue en $\mathcal{O}(n^2)$. En effet, on fait au plus hk comparaisons. Or on a les inégalités $hk \leq nm \leq n^2$ d'où la complexité.
L'étape 2 revient à multiplier des polynômes entre eux. On vérifie alors que ceci peut être fait en $\mathcal{O}(n)$. En conclusion, la complexité totale de l'algorithme est en $\mathcal{O}(n^2)$.

Algorithme 4.1 Calcul de PGCD

Entrée : h et k, les racines $(u_i)_{i=1\ldots n}$ et $(v_j)_{j=1\ldots m}$, les multiplicités $(\alpha_i)_{i=1\ldots h}$ et $(\beta_j)_{j=1\ldots k}$
Sortie : $d = \gcd(p, q)$

1: **Pour** $j = 1 \ldots k$ **faire**
si il existe i tel que $u_i = v_j$ **alors** mémoriser v_j et $\mu_j = \min(\alpha_i, \beta_j)$.
2: On retourne en sortie le polynôme $d(z) = \prod_j (z - v_j)^{\mu_j}$, le produit se faisant sur les j dont les u_j et μ_j ont été stockés dans la phase précédente. Si aucune valeur n'a été stockée, on renvoie $d(z) = 1$.

On se place maintenant dans le cas plus intéressant où les polynômes sont représentés par des approximations de leurs racines avec une tolérance bornée par δ. On ne considère plus alors p et q mais $\widehat{p}$ et $\widehat{q}$ ayant leurs racines dans un δ-voisinage de u_i et v_j respectivement. Nous recherchons donc un ϵ-diviseur de $\widehat{p}$ et $\widehat{q}$ de degré maximal. Mais auparavant, nous devons regarder le lien entre la perturbation des racines d'un polynôme (perturbation définie localement) et la perturbation en norme sur le polynôme.

Lemme 4.4.1. *Soient p et $\widehat{p}$ deux polynômes unitaires de degré n ayant respectivement pour racines u_i et $\widehat{u}_i$ tel que*

$$|u_i - \widehat{u}_i| \leq \delta, \quad i = 1, \ldots, n.$$

Alors

$$\|p - \widehat{p}\|_1 \leq \|p\|_1((1+\delta)^n - 1).$$

Preuve : La valeur de $\|p-\widehat{p}\|_1$ atteint son maximum pour $p(z) = \Pi_{i=0}^n (z-u_i)$ et $\widehat{p} = \Pi_{i=0}^n (z-(\widehat{u}_i - \delta)) = p(x+\delta)$. On a alors

$$\begin{aligned}
\|p - \widehat{p}\|_1 &= \|\sum_{i=1}^{n} p^{(i)}(z)\delta^i/i!\|_1, \\
&\leq \sum_{i=1}^{n} \|p^{(i)}(z)/i!\|_1 \delta^i, \\
&\leq \|p\|_1 \sum_{i=1}^{n} \delta^i \binom{n}{i}, \\
&= \|p\|_1((2+\delta)^n - 1).
\end{aligned}$$

■

Par définition d'un ϵ-diviseur, il convient donc de choisir δ vérifiant

$$\|p\|_1((1+\delta)^n - 1) \leq \epsilon. \tag{4.1}$$

L'algorithme 4.2 permet de calculer un ϵ-diviseur.

4.5 Calcul d'un ϵ-PGCD : polynômes définis par ses coefficients

Dans les sections suivantes, nous supposerons les polynômes représentés par leurs coefficients et non plus par leurs racines.
Dans la littérature, on trouve trois approches différentes pour calculer un ϵ-PGCD de deux polynômes.

La première approche consiste à adapter l'algorithme classique d'Euclide pour le calcul du PGCD de deux polynômes en modifiant les tests et les conditions d'arrêt [34]. Cependant, on ne peut pas prouver en sortie de l'algorithme que l'on a bien un ϵ-PGCD (ϵ-diviseur de degré maximum).

La deuxième approche est une approche matricielle [27, 29, 30]. Elle consiste à se ramener à des problèmes matriciels (en général), et utiliser des outils numériques (par exemple la SVD (Décomposition

Algorithme 4.2 Calcul d'un ϵ-diviseur

Entrée : δ vérifiant (4.1), les racines $(u_i)_{i=1\ldots n}$ et $(v_j)_{j=1\ldots m}$
Sortie : $d = \epsilon\text{-}\gcd(p, q)$ avec

$$p(z) = \prod_{i=1}^{n}(z - u_i), \quad q(z) = \prod_{j=1}^{m}(z - v_j)$$

1: **Pour** toutes les paires (i, j), $i = 1, \ldots, n$ et $j = 1, \ldots, m$ **faire**
si $|u_i - v_j| \leq \delta$ **alors** mémoriser la paire (i, j).
2: Définir le graphe bipartite G ayant pour sommets les éléments des deux ensembles $U = \{u_1, \ldots, u_n\}$ et $V = \{v_1, \ldots, v_m\}$, connectés par les arêtes (u_i, v_j) ssi le couple (i, j) a été mémorisé dans l'étape précédente.
Rechercher un *matching* maximal $(u_{i_1}, v_{j_1}), \ldots, (u_{i_r}, v_{i_r})$ dans G (avec le plus de couples distincts possibles).
3: Calculer $\widehat{d}$ défini par

$$\widehat{d} = \prod_{q=1}^{r}(z - z_q), \quad z_q = (u_{i_q} + v_{j_q})/2, \quad q = 1, \ldots, r.$$

en Valeurs Singulières)). Cette méthode permet d'obtenir une borne supérieure sur le degré d'un ϵ-PGCD.

La troisième approche traite le problème comme un problème d'optimisation [26, 35]. L'idée générale est de poser les coefficients et le degré du futur PGCD comme inconnus.

Nous proposons maintenant une présentation succincte de ces différentes approches.

4.5.1 Résolution par algorithme d'Euclide adapté

La façon classique de calculer un PGCD de deux polynômes p et q exacts est d'appliquer l'algorithme classique d'Euclide. Il semble donc normal d'essayer d'adapter cet algorithme pour les PGCD approchés [29, 34]. L'inconvénient majeur de cette méthode est qu'elle ne permet pas de certifier que le résultat obtenu est bien un PGCD approché et non pas seulement un diviseur approché. En effet, l'algorithme ne permet pas de prendre en compte toutes les perturbations, mais seulement celles données par les divisions successives. L'algorithme repose sur ce qu'on appelle des PRS (Polynomial Remainder Sequences) généralisées.

Soient $f_1 = p$ et $f_2 = q$. On effectue alors la division euclidienne

$$f_{j-1} = q_j f_j + f_{j+1}.$$

On définit aussi deux autres suites de polynômes $r_j^{(1)}$ et $r_j^{(2)}$ par

$$\begin{cases} r_0^{(1)} = 0, \quad r_1^{(1)} = 1, \\ r_j^{(1)} = q_j r_{j-1}^{(1)} + r_{j-2}^{(1)}, \end{cases}$$

et

$$\begin{cases} r_1^{(2)} = 0, \quad r_2^{(2)} = 1, \\ r_j^{(2)} = q_j r_{j-1}^{(2)} + r_{j-2}^{(2)}. \end{cases}$$

On peut alors montrer [29] que l'on a

$$f_1 = r_j^{(1)} f_j + r_{j-1}^{(1)} f_{j+1}, \tag{4.2}$$

et

$$f_2 = r_j^{(2)} f_j + r_{j-1}^{(2)} f_{j+1}, \tag{4.3}$$

avec $\deg r_j^{(1)} f_j = \deg f_1$ et $\deg r_j^{(2)} f_j = \deg f_2$. Dès que les restes dans les équations (4.2) et (4.3) sont de normes inférieures à ϵ, on peut considérer que f_j est un ϵ-diviseur commun de p et q.

On a donc finalement un algorithme pour calculer un diviseur approché.

Algorithme 4.3 Calcul d'un diviseur approché par l'algorithme d'Euclide

Entrée : p, q deux polynômes et ϵ
Sortie : un diviseur approché g de p et q avec une tolérance ϵ
  Initialiser $f_1 = p$, $f_2 = q$ et $j = 1$
  répéter
    $j := j + 1$
    Calculer q_j et f_{j+1} par la division $f_{j+1} = q_j f_j + f_{j+1}$.
    Calculer $r_j^{(1)}$ et $r_j^{(2)}$ par $r_j^{(i)} = q_j r_{j-1}^{(i)} + r_{j-2}^{(i)}$.
  jusqu'à $\|r_{j-1}^{(1)} f_{j+1}\| \leq \epsilon$ et $\|r_{j-1}^{(2)} f_{j+1}\| \leq \epsilon$.
  Retourner $g = f_j$

La terminaison de l'algorithme est immédiate. Regardons de plus près l'algorithme de façon à prouver qu'il renvoie bien un ϵ-diviseur. Les équations (4.2) et (4.3) se réécrivent sous la forme

$$f_1 - r_j^{(1)} f_j = r_{j-1}^{(1)} f_{j+1}, \tag{4.4}$$
$$f_2 - r_j^{(2)} f_j = r_{j-1}^{(2)} f_{j+1}. \tag{4.5}$$

On suppose maintenant que l'on arrive à la fin de l'algorithme, c'est-à-dire que l'on a

$$\|r_{j-1}^{(1)} f_{j+1}\| \leq \epsilon \text{ et } \|r_{j-1}^{(2)} f_{j+1}\| \leq \epsilon.$$

En reportant cela dans les égalités (4.4) et (4.5) ci-dessus, on obtient $\|f_1 - r_j^{(1)} f_j\| \leq \epsilon$ et $\|f_2 - r_j^{(2)} f_j\| \leq \epsilon$ ce qui montre bien que f_j est un ϵ-diviseur de f_1 et f_2.

Remarque. Nous allons montrer ici un contre-exemple afin de corroborer le fait que l'algorithme précédent ne renvoie pas toujours un ϵ-PGCD.
Prenons en effet

$$\begin{aligned} f_1 &= x^5 + 5.503x^4 + 9.765x^3 + 7.647x^2 + 2.762x + 0.37725, \\ f_2 &= x^4 - 2.993x^3 - 07745x^2 + 2.007x + 0.7606. \end{aligned}$$

Le tableau 4.1 déroule l'algorithme présenté ci-dessus. La première colonne donne les restes, la seconde la valeur qui sert de terminaison à l'algorithme et la troisième la norme du reste correspondant (en norme 2). □

j	f_j	$\max(\|b_j r_{j-1}^{(1)} f_{j+1}\|, \|b_j r_{j-1}^{(2)} f_{j+1}\|)$	$\|f_j\|$
3	$35.968x^3 + 12.220x^2 - 15.050x - 6.0840$	41.310	41.310
4	$0.77623x^2 + 0.78164x + 0.19677$	10.226	1.1190
5	$-0.0018829x - 0.00051872$	0.00056718	0.0019530
6	0.040344	0.36110	0.040343
7	0	0	0

TAB. 4.1: *Pour* $\epsilon = 5.6 \cdot 10^{-4}$, *l'algorithme* 4.3 *donne* 0 *pour le degré du PGCD alors que pour* $\epsilon = 1.6 \cdot 10^{-4}$ *le degré du PGCD est* 2 *avec* $x^2 + 1.007x + 0.2534$.

4.5.2 Résolution par optimisation

Recherche d'une racine commune

Nous nous basons sur les articles de Karmarkar et Lakshman [35, 36]. Tout d'abord on s'intéresse au problème plus simple suivant qui est de trouver une perturbation minimale sur les polynômes afin qu'ils aient une racine commune α.

Soient donc p et q deux polynômes de $\mathbb{C}[z]$ de degré respectif n et m. Posons

$$\phi(z) = (z-\alpha)(\sum_{i=0}^{n-1} \phi_i z^i) = \sum_{i=0}^{n} (\phi_{i-1} - \alpha\phi_i) z^i \text{ avec } \phi_{-1} = \phi_n = 0 \text{ et } \phi_{n-1} = 1,$$

$$\gamma(z) = (z-\alpha)(\sum_{j=0}^{m-1} \gamma_j z^j) = \sum_{j=0}^{m} (\gamma_{j-1} - \alpha\gamma_j) z^j \text{ avec } \gamma_{-1} = \gamma_m = 0 \text{ et } \gamma_{m-1} = 1.$$

Le problème est de trouver α, ϕ, γ tels que $\mathcal{N} = \|p-\phi\|_2^2 + \|q-\gamma\|_2^2$ soit minimale. On peut alors montrer que $\mathcal{N}$ peut se mettre sous la forme

$$\mathcal{N} = y^* Q y - (y^* r + r^* y) + s,$$

où y est le vecteur

$$y = [\phi_{n-2}, \phi_{n-1}, \ldots, \phi_0, \gamma_{m-2}, \gamma_{m-3}, \ldots, \gamma_0]^T,$$

Q une matrice hermitienne définie positive de taille $(n+m-2)$, r un vecteur de taille $(n+m-2)$ et s un scalaire complexe.

Pour tout $\alpha \in \mathbb{C}$, $\mathcal{N}$ a un minimum au point noté y_m vérifiant $Qy_m = r$. La valeur en ce point de $\mathcal{N}$ que l'on notera $\mathcal{N}_m$ est $\mathcal{N}_m = -r^* Q^{-1} r + s$. $\mathcal{N}_m$ est une fonction réelle de la variable complexe α et nous voulons la minimiser par rapport à α. Notons $\alpha = a + ib$ où a et b sont des variables réelles. $\mathcal{N}_m$ est alors une fraction rationnelle de $\mathbb{R}(a, b)$. Alors les minima locaux de $\mathcal{N}_m$ sont donnés par

$$\frac{\partial \mathcal{N}_m}{\partial a} = \frac{\partial \mathcal{N}_m}{\partial b} = 0.$$

En fait, on peut montrer qu'il existe une constant B ne dépendant que de p et q telle que le minimum se situe dans le pavé $[-B, B] \times [-B, B]$. Il nous suffit donc de trouver les points d'intersection de

$$\frac{\partial \mathcal{N}_m}{\partial a} = \frac{\partial \mathcal{N}_m}{\partial b} = 0$$

dans cette zone et de chercher le minimum parmi ces points. L'algorithme 4.4 qui permet de calculer α, ϕ et γ.

Dans [36], les auteurs donnent une formulation légèrement différente de leur algorithme. Ils montrent que la recherche d'une racine commune se ramène à minimiser la fonction dépendant de α suivante :

$$\mathcal{N}(\alpha) = \frac{p(\alpha)\overline{p(\alpha)}}{\sum_{i=0}^{n-1} (\overline{\alpha}\alpha)^i} + \frac{q(\alpha)\overline{q(\alpha)}}{\sum_{i=0}^{m-1} (\overline{\alpha}\alpha)^i},$$

si p est unitaire de degré n et q unitaire de degré m. Cela donne de nouveau une fraction rationnelle en a et b (si $\alpha = a + ib$) à minimiser.

Calcul du PGCD approché

L'idée est ici la même que précédemment. Mais au lieu de chercher un facteur commun sous la forme $(z-\alpha)$, on cherche un facteur commun proche de degré k :

$$\alpha_k(z) \quad = \quad \sum_{j=0}^{k} \alpha_{k,j} z^j,$$

Algorithme 4.4 Calcul du plus proche diviseur commun

Entrée : Deux polynômes $p, q \in \mathbb{C}[z]$
Sortie : Deux polynômes $p + \widehat{p}, q + \widehat{q} \in \mathbb{C}[z]$ et α racine commune de $\widehat{p}$ et $\widehat{q}$

1: Calculer $\mathcal{N} = y^*Qy - (y^*r + r^*y) + s$.
2: Calculer $\mathcal{N}_m = -r^*Q^{-1}r + s$.
3: Trouver les solutions réelles $\mathcal{S}$ de $\partial\mathcal{N}_m/\partial a = \partial\mathcal{N}_m/\partial b = 0$ dans le pavé $[-B, B] \times [-B, B]$.
4: Chercher le couple (a, b) de $\mathcal{S}$ qui minimise $\mathcal{N}_m$ et calculer $y_m = Q^{-1}r$.
5: Retourner $\alpha = a + ib$ et

$$\widehat{p} = (z - \alpha) \sum_{i=0}^{n-1} \phi_i z^i,$$

$$\widehat{q} = (z - \alpha) \sum_{j=0}^{m-1} \gamma_j z^j.$$

et deux polynômes

$$\begin{aligned} \Phi_k(z) &= \sum_{j=0}^{n-k} \phi_{k,j} z^j, \\ \Gamma_k(z) &= \sum_{j=0}^{m-k} \gamma_{k,j} z^j, \end{aligned}$$

tels que

$$\|p - \alpha_k \Phi_k\| \leq \epsilon \text{ et } \|q - \alpha_k \Gamma_k\| \leq \epsilon.$$

Mais nous cherchons un facteur commun de plus grand degré. Le problème se réécrit donc sous la forme :

Trouver un entier d tel qu'il existe des polynômes α_k, Φ_k et Γ_k pour $k = 1, \ldots, d+1$ avec

$$\begin{aligned} \alpha_k(z) &= \sum_{j=0}^{k} \alpha_{k,j} z^j, \\ \Phi_k(z) &= \sum_{j=0}^{n-k} \phi_{k,j} z^j, \\ \Gamma_k(z) &= \sum_{j=0}^{m-k} \gamma_{k,j} z^j, \end{aligned}$$

tels que

$$\|p - \alpha_k \Phi_k\| \leq \epsilon \textit{ et } \|q - \alpha_k \Gamma_k\| \leq \epsilon$$

pour $k = 1, \ldots, d$ et

$$\|p - \alpha_{d+1} \Phi_{d+1}\| > \epsilon \textit{ et } \|q - \alpha_{d+1} \Gamma_{d+1}\| > \epsilon.$$

Supposons que nous soyons à l'étape k, *i.e.* le degré du diviseur commun que l'on cherche est k. En reprenant la méthode précédente, cela revient à minimiser les deux fonctions

$$\mathcal{N}^{(p,\Phi,\alpha)} = \|p - \alpha_k \Phi_k\|^2 = y^*_{(p,\Phi)} Q_{(p,\alpha)} y_{(p,\Phi)} - y^*_{(p,\Phi)} r_{(p,\alpha)} - r^*_{(p,\alpha)} y_{(p,\Phi)} + s_{(p,\alpha)},$$

$$\mathcal{N}^{(q,\Gamma,\alpha)} = \|q - \alpha_k \Gamma_k\|^2 = y^*_{(q,\Phi)} Q_{(q,\alpha)} y_{(q,\Phi)} - y^*_{(q,\Phi)} r_{(q,\alpha)} - r^*_{(q,\alpha)} y_{(q,\Phi)} + s_{(q,\alpha)}.$$

De la même façon que précédemment, on peut montrer que ces fonctions ont pour minimum en fonction de α (on minimise sur Φ et Γ)

$$\mathcal{N}_m^{(p,\alpha)} = r_{(p,\alpha)}^* Q_{(p,\alpha)}^{-1} r_{(p,\alpha)} + s_{(p,\alpha)},$$
$$\mathcal{N}_m^{(q,\alpha)} = r_{(q,\alpha)}^* Q_{(q,\alpha)}^{-1} r_{(q,\alpha)} + s_{(q,\alpha)}.$$

Il faut ensuite vérifier si on a bien $\mathcal{N}_m^{(p,\alpha)} < \epsilon^2$ et $\mathcal{N}_m^{(q,\alpha)} < \epsilon^2$. Si c'est le cas, il faut minimiser $\mathcal{N}_m^{(p,\alpha)} + \mathcal{N}_m^{(q,\alpha)}$ sur α et on passe à l'étape suivante. Sinon, on s'arête et le ϵ-PGCD est α_{k-1}.

Cet algorithme est très coûteux. Pour une étude des différentes façons d'implémenter la minimisation, on peut se référer à [26] qui effectue une comparaison selon trois algorithmes de minimisation différents.

Remarque. Le problème est difficile. En effet, si l'algorithme de minimisation que l'on utilise ne nous renvoie qu'un minimum local, alors on obtiendra en sortie qu'un ϵ-diviseur et non un ϵ-PGCD. □

4.5.3 Une approche matricielle : la SVD

La SVD est un outil couramment utilisé en analyse numérique. Sa principale qualité est qu'elle est stable par rapport aux perturbations. Faisons tout d'abord quelques rappels sur la Décomposition en Valeurs Singulières (SVD) [3, 7].

Théorème 4.5.1. *Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$ avec $\mathbb{K} = \mathbb{R}$ où $\mathbb{C}$ de rang k. Alors A peut s'écrire sous la forme*

$$A = UDV^*,$$

où $U \in \mathcal{M}_m(\mathbb{K})$ et $V \in \mathcal{M}_n(\mathbb{K})$ sont des matrices unitaires (on dit souvent orhogonale dans le cas réel). La matrice $D = (\sigma_{ij}) \in \mathcal{M}_{m,n}(\mathbb{K})$ vérifie $\sigma_{ij} = 0$ pour $i \neq j$ et $\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{kk} > \sigma_{k+1,k+1} = \cdots = \sigma_{qq} = 0$ où $q = \min(m, n)$.

Remarque. On voit aisément que si $n = m$ alors la matrice D est une matrice diagonale. □

Une des plus importantes propriétés de la SVD pour son utilisation dans le calcul de PGCD est que σ_k représente la distance en norme euclidienne à la matrice la plus proche de rang strictement inférieur à k.

Soient maintenant deux polynômes p et q deux polynômes réels de degré respectifs n et m. Notons S la matrice de Sylvester de p et q. On rappelle que la matrice de Sylvester de p et q est

$$S = \begin{bmatrix} p_0 & 0 & \cdots & 0 & q_0 & 0 & \cdots & 0 \\ p_1 & p_0 & \ddots & \vdots & q_1 & q_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ p_n & & \ddots & p_0 & q_m & & \ddots & q_0 \\ 0 & p_n & & p_1 & 0 & q_m & & q_1 \\ \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & p_n & 0 & \dots & 0 & q_m \end{bmatrix} \in \mathbb{C}^{(n+m)\times(n+m)}.$$

Un résultat très intéressant est le suivant :

Proposition 4.5.2. *Si r est le rang de la matrice de Sylvester alors le degré d'un PGCD de p et q est $n + m - r$.*

L'idée de l'article [27] est de généraliser cette méthode en définissant la notion de rang numérique. On obtient alors l'algorithme 4.5.

Nous allons maintenant justifier théoriquement cet algorithme par les deux lemmes suivants.

Lemme 4.5.3. *Si $E = S(\Delta p, \Delta q) = S(p + \Delta p, q + \Delta q) - S(p, q)$ alors $\|\Delta p\|_2 \leq \|E\|_2$, $\|\Delta p\|_2 \leq \|E\|_2$ et*

$$\|E\|_2^2 \leq \|E\|_F^2 = n\|\Delta p\|_2^2 + m\|\Delta q\|_2^2,$$

où $\|\cdot\|_F$ est la norme de Frobenius.

Algorithme 4.5 Calcul d'un diviseur approché par SVD

Entrée : Deux polynômes $p, q \in \mathbb{R}[z]$ et une tolérance ϵ
Sortie : Un polynôme d de degré n_d satisfaisant

1. le polynôme d est le PGCD exact de deux polynômes $\widehat{p}$ et $\widehat{q}$ vérifiant $\|p - \widehat{p}\| \leq \widehat{\epsilon}$ et $\|q - \widehat{q}\| \leq \widehat{\epsilon}$. Nous verrons plus loin à quoi correspond $\widehat{\epsilon}$.

2. Parmi les polynômes $(d^*, \widehat{p}, \widehat{q})$ vérifiant les propriétés 1 et 2, on choisit ceux tels que $\widehat{p}$ et $\widehat{q}$ soient les plus proches de p et q au sens des moindres carrés.

1: Former la matrice de Sylvester S de p et q .
2: Calculer la SVD de $S = UDV^T$.
3: Trouver le maximum k tel que $\sigma_k > \epsilon\sqrt{m+n}$ et $\sigma_{k+1} \leq \epsilon$ (si pour tout j $\sigma_j > \epsilon\sqrt{m+n}$ alors $d = 1$ et s'il n'y a pas de "saut" alors renvoyer une erreur. L'indice k définit alors le rang numérique de S (ou ϵ-rang) et le degré de d est $n_d = n + m - k$.
4: Calculer d par une des méthodes suivantes

(a) On calcule d par l'algorithme d'Euclide en s'arrêtant quand le degré du reste est n_d.

(b) On résoud un problème de minimisation.

Lemme 4.5.4. *Si les valeurs singulières de $S(p, q)$ sont σ_j, $j = 1, 2, \ldots, m + n$ et $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > \epsilon\sqrt{m+n} > \epsilon \geq \sigma_{k+1} \ldots \geq \sigma_{m+n}$ et si $\widetilde{d}$ est un diviseur commun de $p+\Delta p$ et $q+\Delta q$ avec $\deg \widetilde{d} \geq n+m-k+1$, alors $\|\Delta p\| > \epsilon$ ou $\|\Delta q\| > \epsilon$.*

Preuve : La matrice de rang $n + m - k + 1$ la plus proche $S + E$ vérifie $\|E\|_2 \geq \sigma_k > \epsilon\sqrt{n+m}$. Ainsi $S(p + \Delta p, q + \Delta q) - S(p, q) = S(\Delta p, \Delta q)$ vérifie $\|S(\Delta p, \Delta q)\| \geq \sigma_k > \epsilon\sqrt{n+m}$. Mais si l'on a $\|\Delta p\| \leq \epsilon$ et $\|\Delta q\| \leq \epsilon$ alors en appliquant le lemme (4.5.3) il vient $\|S(\Delta p, \Delta q)\|_2^2 \leq n\epsilon^2 + m\epsilon^2$ c'est à dire $\|S(\Delta p, \Delta q)\|_2^2 \leq \epsilon\sqrt{n+m}$. ■

Le lemme précédent nous permet juste de conclure que $n_d = n + m - k$ est une borne supérieure du degré d'un ϵ-PGCD.

Le problème avec l'algorithme précédent est le suivant : les perturbations des deux polynômes ne sont pas inférieures à ϵ mais à un $\widehat{\epsilon}$. On peut alors essayer de minimiser les perturbations. On compare ensuite la norme des perturbations à ϵ. Si elles sont toutes les deux inférieures à ϵ alors on a bien un ϵ-PGCD, sinon on ne peut garantir la maximalité du degré.

Prenons un exemple. Soit $p(z) = (z-1)(z-2) = z^2 - 3z + 2$, $q(z) = (z-1.08)(x-1.82) = z^2 - 2.9z + 1.9656$ et $\epsilon = 0.016$. Les valeurs singulières de la matrice de Sylvester de p et q sont $6.698774 > 3.110021 > 0.0333550 > 0.0156330$. En appliquant l'algorithme précédent, on trouve que le degré d'un ϵ-PGCD est 1. On calcule alors le ϵ-PGCD est utilisant des divisions euclidiennes. On trouve $0.1(z - 0.344) = q - p$. Si on cherche les perturbations minimales associées à ce PGCD on trouve que l'une d'elle a pour norme 1.020883. Or avec cette tolérance, le degré d'un ϵ-PGCD est au moins 2. En effet, $(z - 0.96)(z - 2.04)$ est un ϵ-diviseur pour $\epsilon < 1.020883$. Ainsi, en utilisant les divisions euclidiennes, on est pas certain de trouver un ϵ-PGCD avec la tolérance donnée.

4.6 PGCD approché et certification

Le problème de la certification est apparu dans les articles [29, 30]. Le principe est la suivant :

– on calcul une borne inférieure du degré d'un ϵ-PGCD en utilisant l'algorithme d'Euclide modifié ;

– on calcule une borne supérieure du degré d'un ϵ-PGCD en utilisant des SVD.

Pour une large classe de paire de polynômes, ces deux bornes coïncident. Cela permet donc de certifier le degré d'un ϵ-PGCD.
Nous n'avons pas étudié en détail cette approche mais elle nous semble intéressante et fera parti d'un travail futur.

4.7 Une étude géométrique

Nous introduisons dans cette section une nouvelle définition de ϵ-PGCD à l'aide des pseudozéros. Il s'agit d'une définition proche de celle de Pan [39]. Nous allons le définir sous la forme d'un algorithme. Tout d'abord on trace les pseudozéros de p et q que l'on superpose. Lorsqu'une composante connexe de l'ensemble des pseudozéros de p intersecte une composante connexe de l'ensemble des pseudozéros de q alors on choisit au hasard un complexe dans l'intersection. On fait cela pour toutes les intersections entre les différentes composantes connexes. On obtient alors une suite de nombre $(\alpha_i)_{i=1,\dots,l}$. On définit alors un ϵ-PGCD par $\Pi_{i=1}^{l}(z-\alpha_i)$.

Certes, cet algorithme ne nous permet pas de certifier que l'on a un PGCD de degré maximal car on ne tient pas compte des multiplicités des racines. Il s'agit simplement de regrouper les racines "presque communes".

4.8 Synthèse sur le calcul de ϵ-PGCD

Nous récapitulons dans le tableau 4.2 les différentes méthodes du calcul d'un ϵ-PGCD issues des nombreux articles que nous avons lu sur le sujet. En effet, notre travail sur les ϵ-PGCD a surtout consisté en un synthèse d'articles.

Méthode	Fonctionne	Coût	Remarques
Euclide	ϵ-diviseur	$\mathcal{O}((n+m)^2)$	borne inférieure du degré
Optimisation	ϵ-PGCD	exponentiel	$\|\cdot\|_2$
SVD	$\widehat{\epsilon}$-diviseur	$\mathcal{O}((n+m)^2)$	borne supérieure du degré
Racines	ϵ-diviseur	$\times$	$\times$
Certification	ϵ-PGCD	$\times$	$\times$
Pseudozéros	$\widehat{\epsilon}$-diviseur	$\times$	$\times$

TAB. 4.2: *Différentes méthodes pour calculer un ϵ-PGCD.*

Chapitre 5

Polynômes premiers entre eux

Dans ce chapitre, nous étudions la notion de polynômes *premiers entre eux*. On dit que deux polynômes sont *premiers entre eux* s'ils n'ont aucunes racines communes. Il est souvent intéressant, avant de commencer un calcul de ϵ-PGCD coûteux, de faire un test de coprimalité.

Nous étudions tout d'abord les articles [23, 24] qui donnent des bornes sur les perturbations afin que deux polynômes restent premiers entre eux. Ensuite, nous montrons que les tracés de pseudozéros permettent de tester graphiquement la coprimalité de deux polynômes.

5.1 Borne sur les perturbations

5.1.1 Définitions et notations

Soient p et q deux polynômes de $\mathbb{C}[z]$ définis par

$$p(z) = p_n z^n + p_{n-} z + \cdots + p_0, \quad q(z) = q_m z^m + q_{m-1} z^{m-1} + \cdots q_0, \quad p_n, q_m \neq 0.$$

Le PGCD de p et q est donné par

$$\gcd(p,q)(z) = \prod_{\gamma \in A \cap B} (z-\gamma), \text{ où } p(z) = p_n \cdot \prod_{\alpha \in A} (z-\alpha), \quad q(z) = q_m \cdot \prod_{\beta \in B} (z-\beta).$$

On définit de plus sur $\mathbb{C}[z]$ la norme $\|\cdot\|$ par

$$\|p\| = \sum_j |p_j|.$$

Par extension, on définit

$$\|(p,q)\| = \max\{\|p\|, \|q\|\} = \max\{\sum |p_i|, \sum |q_j|\}.$$

On pose alors la définition suivante :

Définition 5.1.1. Pour $p, q \in \mathbb{C}[z]$, on définit

$$\epsilon(p,q) = \inf\{\|(p-\widehat{p}, q-\widehat{q})\| : (\widehat{p},\widehat{q}) \text{ ont une racine commune et } \deg\widehat{p} \leq n, \deg\widehat{q} \leq m\},$$

Remarque. Ceci signifie que tous polynômes $\widetilde{p}, \widetilde{q}$ vérifiant $\|(p-\widetilde{p}, q-\widetilde{q})\| \leq \varepsilon < \epsilon(p,q)$ avec $\deg\widetilde{p} \leq n, \deg\widetilde{q} \leq m$ sont premiers entre eux. Les polynômes p et q sont dit ε-premiers. □

5.1.2 Calcul d'une borne inférieure pour $\epsilon(p,q)$

Il est bien connu que le calcul de PGCD peut être vu comme un problème d'algèbre linéaire. Soit $S(p,q)$ la matrice de Sylvester de (p,q),

$$S(p,q) = \begin{bmatrix} p_0 & 0 & \cdots & 0 & q_0 & 0 & \cdots & 0 \\ p_1 & p_0 & \ddots & \vdots & q_1 & q_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ p_n & & \ddots & p_0 & q_m & & \ddots & q_0 \\ 0 & p_n & & p_1 & 0 & q_m & & q_1 \\ \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & p_n & 0 & \dots & 0 & q_m \end{bmatrix} \in \mathbb{C}^{(n+m)\times(n+m)}.$$

Le critère de Sylvester affirme que deux polynômes sont *premiers entre eux* si et seulement si $S(p,q)$ est une matrice *inversible*. Berkermann et Labahn ont montré [24] que

Lemme 5.1.1. *Pour tout polynôme p et q, nous avons*

$$\epsilon(p,q) \geq \frac{1}{\|S(p,q)^{-1}\|}. \tag{5.1}$$

Preuve : Elle repose sur le théorème de Gastinel (voir [24]). ■

Dans [24], Berkermann et Labahn proposent de calculer une borne supérieure de $\|S(p,q)^{-1}\|$.

5.1.3 Calcul d'une borne inférieure pour $\|S(p,q)^{-1}\|$

Le théorème de Bezout assure que p et q sont premiers entre eux si et seulement si il existe $u, v \in \mathbb{C}[z]$, avec $\deg u < n, \deg v < m$, satisfaisant

$$p \cdot u + q \cdot v = 1. \tag{5.2}$$

L'équation (5.2) peut se réécrire matriciellement sous la forme

$$S(p,q) \cdot \begin{bmatrix} \vec{v} \\ \vec{u} \end{bmatrix} = (1,0,\dots,0)^T. \tag{5.3}$$

Ainsi, deux polynômes sont premiers entre eux si et seulement si on peut déterminer la première colonne de l'inverse de la matrice de Sylvester correspondante.

En explicitant l'inverse de la matrice de Sylvester, on peut montrer le

Théorème 5.1.2. *Soient u et v deux polynômes de degrés au plus $n-1$ et $m-1$ satisfaisant (5.2). Alors*

$$\left\| \begin{bmatrix} v \\ u \end{bmatrix} \right\| \leq \|S(p,q)^{-1}\| \leq \left\| \begin{bmatrix} v \\ u \end{bmatrix} \right\| + 2 \cdot \|f\| \cdot \|(p,q)\| \tag{5.4}$$

ou f est donnée par $f(z) = f_{-1}z^{-1} + \ldots + f_{1-n-m}z^{1-n-m} = \frac{u(z)}{p(z)} + \mathcal{O}(z^{-n-m})_{z\to\infty}$.

Preuve : Voir [24]. ■

Introduisons le problème dual de (5.2) qui est équivalent à l'existence de $\underline{u}, \underline{v} \in \mathbb{C}[z]$ avec $\deg \underline{u} < m, \deg \underline{v} < n$ vérifiant

$$p(z) \cdot \underline{v}(z) + q(z) \cdot \underline{u}(z) = z^{n+m+1}, \tag{5.5}$$

ce qui se réécrit matriciellement sous la forme

$$S(p,q) \begin{bmatrix} \vec{\underline{v}} \\ \vec{\underline{u}} \end{bmatrix} = (0,\dots,0,1)^T. \tag{5.6}$$

On peut alors montrer que

$$f(z) = z^{1-n-m}[\underline{v}(z) \cdot u(z) - \underline{u}(z) \cdot v(z)]. \tag{5.7}$$

En notant

$$\kappa = \left\| \begin{bmatrix} v & \underline{v} \\ u & \underline{u} \end{bmatrix} \right\| = \max\{ \left\| \begin{bmatrix} v \\ u \end{bmatrix} \right\|, \left\| \begin{bmatrix} \underline{v} \\ \underline{u} \end{bmatrix} \right\| \}$$

et en combinant le théorème 5.1.2 et la relation (5.6), on obtient le

Théorème 5.1.3. *Soient u, v et $\underline{u}, \underline{v}$ solution de (5.2) et (5.6). Nous avons*

$$\kappa \leq \|S(p,q)^{-1}\| \leq \kappa + 2 \cdot \|f\| \cdot \|(p,q)\|,$$

avec $\|f\| = \|\underline{v} \cdot u - \underline{u} \cdot v\|$. De plus, on a la majoration suivante, $\|f\| \leq \kappa^2$.

En reprenant l'inégalité (5.1) et le théorème précédent, on obtient une borne inférieure de $\epsilon(p,q)$,

$$\epsilon(p,q) \geq \frac{1}{\kappa + 2\kappa^2 \cdot \|(p,q)\|}. \tag{5.8}$$

Or numériquement, on trouve que $\|S(p,q)^{-1}\|$ est proportionnel à κ et non à κ^2. Il semble donc que la borne trouvée ne soit pas optimale. Afin d'obtenir une borne plus précise, nous devons effectuer une étude plus détaillée de $\epsilon(p,q)$.

On peut montrer le résultat suivant :

Théorème 5.1.4. *Nous avons*

$$\epsilon(p,q) = \inf_{z \in \overline{\mathbb{C}}} \left\| \frac{p(z)}{\|(1,z^n)\|}, \frac{q(z)}{\|(1,z^m)\|} \right\| \tag{5.9}$$

où $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$.

Si on note $h(p,q,z) = \inf_{z \in \overline{\mathbb{C}}} \| \frac{p(z)}{\|(1,z^n)\|}, \frac{q(z)}{\|(1,z^m)\|} \|$ et z^* le point où h atteint son minimum, on montre alors que z^* est la racine commune des polynômes perturbés $\widehat{p}$ et $\widehat{q}$. Il résulte de cela que si z^* appartient au disque unité alors

$$\epsilon(p,q) = \inf_{|z| \leq 1} \|(p(z), q(z))\| \geq \frac{1}{\|(v,u)^T\|}$$

sinon

$$\epsilon(p,q) = \inf_{|z| \geq 1} \|(p(z), q(z))\| \geq \frac{1}{\|(\underline{v},\underline{u})^T\|}.$$

Finalement, en regroupant les deux inégalités précédentes, on obtient

$$\epsilon(p,q) \geq \frac{1}{\kappa}. \tag{5.10}$$

5.2 Utilisation d'une SVD

Nous avons dit, dans le chapitre précédent, que l'utilisation d'une SVD permet d'avoir une borne supérieure du degré d'un ϵ-PGCD. Par conséquent, si l'on trouve que le degré d'un ϵ-PGCD est 0 alors on est sûr que les deux polynômes sont premiers entre eux. Sinon, on ne peut rien dire.

L'inconvénient de ses algorithmes est sous coût qui est en $\mathcal{O}((n+m)^3)$.

5.3 Aspect géométrique à l'aide des pseudozéros

Nous allons maintenant étudier ce que la notion de pseudozéros peut apporter à l'étude de la primalité. De part la définition des pseudozéros, nous pouvons affirmer deux choses :

- si l'intersection des pseudozéros est vide alors les deux polynômes sont premiers entre eux,
- si l'intersection est non vide alors ils ont un ϵ-PGCD non trivial.

Montrons ces affirmations :

Soient p et q deux polynômes à coefficients complexes. Si $Z_\epsilon(p) \cap Z_\epsilon(q) = \emptyset$ alors par définition de l'ensemble des pseudozéros, on ne peut pas trouver $\widehat{p} \in N_\epsilon(p)$ et $\widehat{q} \in N_\epsilon(q)$ ayant une racine commune. Cela signifie bien que p et q sont ϵ-premier entre eux. Si maintenant $Z_\epsilon(p) \cap Z_\epsilon(q) \neq \emptyset$ alors prenons $a \in Z_\epsilon(p) \cap Z_\epsilon(q)$. Cela signifie qu'il existe $\widehat{p} \in N_\epsilon(p)$ et $\widehat{q} \in N_\epsilon(q)$ tels que $\widehat{p}(a) = 0$ et $\widehat{q}(a) = 0$. Donc le polynôme $(z - a)$ divise $\widehat{p}$ et $\widehat{q}$. Par conséquent un ϵ-PGCD est au moins de degré 1 et donc p et q ne sont pas ϵ-premiers entre eux.

Néanmoins un problème se pose. Les composantes connexes d'un ensemble de pseudozéros sont convexes. Mais si la discrétisation de la grille n'est pas assez fine, il se peut que l'on trouve que l'intersection est vide alors qu'elle ne l'est pas en réalité, comme le montre la figure 5.1.

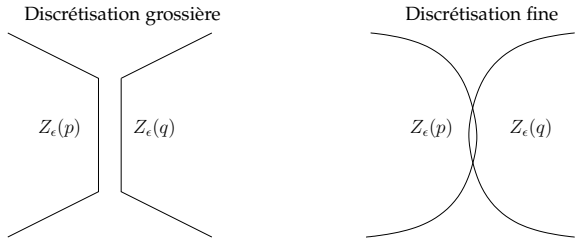


FIG. 5.1: *Influence de la discrétisation dans le choix de la primalité.*

5.4 Synthèse

On récapitule ici les différentes méthodes pour tester la ϵ-primalité. Les deux premières méthodes du tableau 5.1 sont issues de la littérature. La troisième est notre contribution au sujet.

Méthode	Coût	Certification
Borne de Beckermann et Labahn	$\mathcal{O}((n+m)^2)$	primalité
SVD	$\mathcal{O}((n+m)^3)$	primalité
pseudozéros	$\times$	non primalité

TAB. 5.1: *Différentes méthodes pour tester l'ϵ-primalité.*

Conclusion et perspectives

Travail effectué

Nous avons étudié la notion de *pseudozéros* et plus particulièrement la façon de les calculer numériquement. Nous avons pour cela écrit un package MATLAB qui permet le tracé de l'ensemble des pseudozéros pour divers types de perturbations.

Nous nous sommes ensuite intéressé au polynôme le plus proche d'un polynôme donné ayant une racine donnée. Nous avons obtenu des résultats simples qui semblent prometteurs dans la recherche de racines communes de deux polynômes. Nous avons aussi résolu un problème posé par Hitz dans sa thèse.

Nous avons proposé un état de l'art sur le notion de PGCD approché. Nous comparons les différentes approches, leur coût et les résultats qu'elles donnent. Puis nous montrons que les pseudozéros peuvent être une façon de calculer un diviseur approché.

Enfin, nous testons la ϵ-primalité de deux polynômes. Après une courte présentation des travaux de Beckermann et Labahn [23, 24] nous montrons que les pseudozéros permettent de répondre simplement à cette question.

En résumé, nous avons montré que les pseudozéros permettent de résoudre des problèmes sur des polynômes approchés.

Perspectives

Pour les pseudozéros, il s'agirait de mieux automatiser les tracés. En effet, actuellement il faut l'intervention de l'utilisateur pour raffiner le tracé. Pour cela, deux directions nous semblent intéressantes. D'une part, il nous faut trouver des bornes sur les racines les plus fines possible, d'autre part il nous faut quantifier le mieux possible les écarts entre les racines afin d'avoir une grille qui permette la séparation des racines.

Nous avons commencé à étudier le problème suivant :

trouver le polynôme réel le plus proche d'un polynôme réel donné ayant une racine complexe donnée.

Il s'agit d'un problème plus difficile que dans le cas de polynômes complexes.

Nous avons pu voir lors de l'étude des ϵ-PGCD que la plupart des algorithmes présentés sont en fait des heuristiques. Il nous semblerait donc intéressant d'étudier plus en détail les problèmes de certification [29, 30].

Les algorithmes présentés sont tous en précision infinie. Il semble donc que leur passage en précision finie est inévitable si l'on veut avoir des algorithmes efficaces. Par conséquent, l'étude du comportement

numérique en précision finie de ces algorithmes, comme par exemple celui d'Euclide, est nécessaire.

De même que nous nous sommes intéressé au calcul de pseudozéros et de PGCD approchés dans le cas d'une seule variable, il semble logique d'étudier ensuite le cas de plusieurs variables. De la même façon, on pourra aussi étudier la factorisation sans carré, etc.

Bibliographie

The library is the mathematician's laboratory.

— PAUL R. HALMOS, *I want to be a Mathematician (1958)*

1 Références des ouvrages généraux

[1] W. GAUTSCHI – "Questions of numerical condition related to polynomials", Studies in numerical analysis, Math. Assoc. America, Washington, DC, 1984, p. 140–177.

[2] K. O. GEDDES, S. R. CZAPOR et G. LABAHN – *Algorithms for computer algebra*, Kluwer Academic Publishers, Boston, MA, 1992.

[3] G. H. GOLUB et C. F. VAN LOAN – *Matrix computations*, third éd., Johns Hopkins University Press, Baltimore, MD, 1996.

[4] D. J. HIGHAM et N. J. HIGHAM – *MATLAB guide*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[5] N. J. HIGHAM – *Accuracy and stability of numerical algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[6] — , *Handbook of writing for the mathematical sciences*, second éd., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[7] R. A. HORN et C. R. JOHNSON – *Matrix analysis*, Cambridge University Press, Cambridge, 1990.

[8] D. E. KNUTH – *The art of computer programming. Vol. 2*, second éd., Addison-Wesley Publishing Co., Reading, Mass., 1981, Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

[9] M. MIGNOTTE – *Mathematiques pour le calcul formel*, Presses Universitaires de France, Paris, 1989.

2 Références sur les pseudozéros

[10] F. CHAITIN-CHATELIN et V. FRAYSSÉ – *Lectures on finite precision computations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[11] A. EDELMAN et H. MURAKAMI – "Polynomial roots from companion matrix eigenvalues", *Math. Comp.* **64** (1995), no. 210, p. 763–776.

[12] D. HINRICHSEN et B. KELB – "Spectral value sets: a graphical tool for robustness analysis", *Systems Control Lett.* **21** (1993), no. 2, p. 127–136.

[13] R. G. MOSIER – "Root neighborhoods of a polynomial", *Math. Comp.* **47** (1986), no. 175, p. 265–273.

[14] A. M. OSTROWSKI – *Solution of equations and systems of equations*, Academic Press, New York, 1966.

[15] V. Y. PAN – "Solving a polynomial equation: some history and recent progress", *SIAM Rev.* **39** (1997), no. 2, p. 187–220.

[16] R. SCHÄTZLE – "On the perturbation of the zeros of complex polynomials", *IMA J. Numer. Anal.* **20** (2000), no. 2, p. 185–202.

[17] H. J. STETTER – "The nearest polynomial with a given zero, and similar problems", *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)* **33** (1999), no. 4, p. 2–4.

[18] H. J. STETTER – "Polynomials with coefficients of limited accuracy", Computer algebra in scientific computing—CASC'99 (Munich), Springer, Berlin, 1999, p. 409–430.

[19] K.-C. TOH et L. N. TREFETHEN – "Pseudozeros of polynomials and pseudospectra of companion matrices", *Numer. Math.* **68** (1994), no. 3, p. 403–425.

[20] L. N. TREFETHEN – "Computation of pseudospectra", Acta numerica, 1999, Cambridge Univ. Press, Cambridge, 1999, p. 247–295.

[21] J. H. WILKINSON – *Rounding errors in algebraic processes*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1963.

[22] H. ZHANG – "Numerical condition of polynomials in different forms", *ETNA* **12** (2001), p. 66–87.

3 Références sur le PGCD approché

[23] B. BECKERMANN et G. LABAHN – "A fast and numerically stable Euclidean-like algorithm for detecting relatively prime numerical polynomials", *J. Symbolic Comput.* **26** (1998), no. 6, p. 691–714.

[24] — , "When are two numerical polynomials relatively prime?", *J. Symbolic Comput.* **26** (1998), no. 6, p. 677–689.

[25] S. CABAY, A. R. JONES et G. LABAHN – "Algorithm 766: Experiments with a weakly stable algorithm for computing Pade-Hermite and simultaneous Pade approximants.", *ACM Trans. Math. Softw.* **23** (1997), no. 1, p. 91–110.

[26] P. CHIN, R. M. CORLESS et G. F. CORLISS – "Optimization strategies for the approximate GCD problem.", *Proceedings of the 1998 international symposium on symbolic and algebraic computation, ISSAC '98* (O. Gloor, éd.), Août 1998, p. 228–235.

[27] R. M. CORLESS, P. M. GIANNI, B. M. TRAGER et S. M. WATT – "The singular value decomposition for polynomial systems.", *Proceedings of the 1995 international symposium on symbolic and algebraic computation, ISSAC '95* (A. H. M. Levelt, éd.), ACM Press, Juillet 1995, p. 195–207.

[28] I. Z. EMIRIS – "Symbolic-numeric algebra for polynomials", Rapport de recherche, Institut National de Recherche en Informatique et en Automatique (INRIA), 1997.

[29] I. Z. EMIRIS, A. GALLIGO et H. LOMBARDI – "Numerical univariate polynomial GCD", The mathematics of numerical analysis (Park City, UT, 1995), Amer. Math. Soc., Providence, RI, 1996, p. 323–343.

[30] — , "Certified approximate univariate GCDs", *J. Pure Appl. Algebra* **117/118** (1997), p. 229–251.

[31] M. A. HITZ – "Efficient algorithms for computing the nearest polynomial with constrained roots", Thèse, Rensselaer Polytechnic Institute, Avril 1998.

[32] M. A. HITZ et E. KALTOFEN – "Efficient algorithms for computing the nearest polynomial with constrained roots", *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation (Rostock)* (New York), ACM, 1998, p. 236–243 (electronic).

[33] M. A. HITZ, E. KALTOFEN et Y. N. LAKSHMAN – "Efficient algorithms for computing the nearest polynomial with a real root and related problems", *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (Vancouver, BC)* (New York), ACM, 1999, p. 205–212 (electronic).

[34] V. HRIBERNIG et H. J. STETTER – "Detection and validation of clusters of polynomial zeros", *J. Symbolic Comput.* **24** (1997), no. 6, p. 667–681.

[35] N. KARMARKAR et Y. LAKSHMAN – "Approximate polynomial greatest common divisors and nearest singular polynomials.", *Proceedings of the 1996 international symposium on symbolic and algebraic computation, ISSAC '96* (Y. N. Lakshman, éd.), Juillet 1996, p. 35–39.

[36] N. K. KARMARKAR et Y. N. LAKSHMAN – "On approximate GCDs of univariate polynomials", *J. Symbolic Comput.* **26** (1998), no. 6, p. 653–666, Symbolic numeric algebra for polynomials.

[37] M.-T. NODA et T. SASAKI – "Approximate GCD and its application to ill-conditioned algebraic equations", *Proceedings of the International Symposium on Computational Mathematics (Matsuyama, 1990)*, vol. 38, 1991, p. 335–351.

[38] M.-A. OCHI, M.-T. NODA et T. SASAKI – "Approximate greatest common divisor of multivariate polynomials and its application to ill-conditioned systems of algebraic equations", *J. Inform. Process.* **14** (1991), no. 3, p. 292–300.

[39] V. Y. PAN – "Numerical computation of a polynomial gcd and extensions", Rapport de Recherche 2969, Institut National de Recherche en Informatique et en Automatique (INRIA), Août 1996.

[40] D. RUPPRECHT – "Elément de géométrie approchée : Etude du pgcd et de la factorisation", Thèse, Université de Nice-Sophia Antipolis, Janvier 2000.

[41] T. SASAKI et M.-T. NODA – "Approximate square-free decomposition and root-finding of ill-conditioned algebraic equations", *J. Inform. Process.* **12** (1989), no. 2, p. 159–168.

[42] A. SCHÖNHAGE – "Quasi-gcd computations", *J. Complexity* **1** (1985), no. 1, p. 118–137.

Annexes

Annexe A

Présentation du laboratoire

God does arithmetic.

— KARL FRIEDRICH GAUSS (1777-1855)

Je fais actuellement mon stage de DEA de Mathématiques Appliquées au sein du **Laboratoire de l'Informatique du Parallélisme** (LIP). Le LIP est le laboratoire d'informatique de l'École Normale Supérieure (ENS) de Lyon. Il est associé au CNRS, à l'INRIA et est structuré en quatre projets :

- ReMap : Parallélisme, réseaux et systèmes ;
- PLUME : Utilisation et amélioration de la déduction automatique ;
- Arenaire : Synthèse d'architectures et arithmétique des ordinateurs ;
- MC2 : Modèles de calcul et complexité.

Pour mon travail, je me suis intégré dans l'équipe Arenaire sous la direction de Philippe LANGLOIS, maître de conférence. L'objectif du projet Arenaire est l'étude de l'arithmétique des ordinateurs. La recherche s'effectue selon deux axes : d'une part, obtenir des algorithmes fiables et prouvés en utilisant l'arithmétique existante et d'autre part améliorer l'arithmétique existante.

Dans le premier axe de recherche, les membres du projet Arenaire étudient la précision multiple et l'arithmétique exacte visant à étendre la précision des calculs en créant un niveau intermédiaire entre l'arithmétique implantée par la machine et le programme. Cette couche permet à l'utilisateur de faire des calculs intermédiaires exacts ou plus précis sans modifier son programme.
Ils étudient aussi des preuves de propriétés sur les environnements flottants. En effet, les opérateurs flottants disponibles sur la plupart des machines suivent la norme IEEE 754. Cela permet d'écrire des algorithmes très fins sur ces opérateurs, et de prouver des propriétés.

Dans le deuxième axe de recherche, ils étudient le calcul des fonctions élémentaires et les liens entre arithmétique et architecture. En effet la norme IEEE 754 spécifie que le résultat d'une opération arithmétique doit toujours être l'arrondi (au plus près, sauf si l'utilisateur a demandé un autre mode d'arrondi) du résultat exact. Ceci n'est pas exigé pour les fonctions élémentaires car on a longtemps crû qu'une telle requête était impossible à satisfaire. Ils veulent résoudre ce problème pour les fonctions les plus courantes (fonctions trigonométriques, exponentielle, logarithme, etc). Ils travaillent aussi sur la conceptions d'opérateurs arithmétiques. En particulier, ils s'intéressent aux circuits reconfigurables de type "FPGA" (Field Programmable Gate Array), et aux opérateurs asynchrones.

Dans ce cadre, Philippe LANGLOIS s'intéresse plus particulièrement à la validation et l'amélioration de l'effet de l'arithmétique flottante sur les algorithmes numériques du calcul scientifique. Le projet intègre donc des numériciens et des informaticiens.

Annexe B

Sujet et objectif du stage

I find a great part of the information I have was acquired by looking up something and finding something else on the way .

— FRANKLIN P. ADAMS

B.1 Présentation du sujet

Voici le sujet du stage tel qu'il m'a été proposé par Philippe LANGLOIS.

B.1.1 Motivation

Les calculs en arithmétique flottante sur ordinateur sont sujets aux erreurs d'arrondi qui dégradent la précision des résultats et peuvent modifier les propriétés numériques des algorithmes. Les travaux de recherche en précision finie ont pour objectifs de mieux comprendre ces effets, de proposer des modèles théoriques et des outils qui aident à valider la fiabilité des résultats et la robustesse des algorithmes. L'interaction entre calculs symboliques et numériques fournit un cadre algorithmique où cette fiabilité est essentielle.

B.1.2 Travail proposé

Dans ce stage, nous étudions le pgcd de deux polynômes. L'effet de la précision finie sur les coefficients des polynômes nécessite déjà de redéfinir la notion de pgcd. Pour les mêmes raisons, de nouveaux algorithmes de calcul de pgcd doivent être introduit : la version numérique de l'algorithme d'Euclide n'est pas satisfaisante.

Par ailleurs, la notion de pseudozéros de polynômes est à rapprocher de celle de pseudospectres en algèbre linéaire numérique dont l'intérêt pour l'étude des algorithmes en précision finie est prouvée par les travaux récents de Trefethen et Chaitin-Chatelin par exemple.

L'objectif du stage est d'étudier l'intérêt des pseudozéros pour la détermination du pgcd de deux polynômes numériques. On pourra par exemple étudier les limites des algorithmes de calcul de pgcd grâce aux tracés de pseudozéros. On essaiera aussi d'utiliser ces pseudozéros pour proposer un algorithme de calcul de pgcd.

B.2 Planning

B.2.1 Tache 1 : étude des pseudozéros

Les objectifs dans cette première partie sont les suivants.

- Comprendre les aspects théoriques des pseudozéros en comprenant bien la définition et en identifiant les paramètres pertinants.
- Développer et valider un package MATLAB pour tracer des pseudozéros.

Dans ce but, nous étudierons la notion de ϵ-pseudozéros avec une vision applicative. Nous commencerons par étudier la bibliographie [11, 13, 19, 22]. Nous regarderons aussi le lien entre la notion de pseudozéros et la notion de précision finie. Après un développement théorique, notre but est de programmer un package MATLAB pour tracer facilement des pseudozéros. Nous regarderons aussi la notion de pseudospectres (notion largement étudiée depuis quelques années) afin de voir si l'on peut appliquer certains des résultats aux pseudozéros.

B.2.2 Tache 2 : étude de la notion de ϵ-pgcd

Les objectifs dans cette seconde partie sont les suivants.

- Comprendre les effets de la précision finie sur le pgcd de polynômes; l'objectif étant d'avoir de bonnes définitions.
- Comprendre les effets de la précision finie sur l'algorithme d'Euclide; l'objectif étant d'avoir de bon "cas test".

Pour cela, nous étudierons tout d'abord la bibliographie, et en particulier [23, 24, 35, 39, 42]. Nous essaierons de voir en quoi l'algorithme d'Euclide est instable numériquement. Cela implique une redéfinition de la notion de pgcd en la notion de pgcd approché. Dans la littérature, on trouve plusieurs définitions de pgcd approchés. Nous allons les étudier et voir la pertinence par rapport à notre problème de ces différentes définitions.

B.2.3 Tache 3 : lien entre les deux notions

Notre but est ici de voir ce que peuvent apporter les pseudozéros dans l'étude l'algorithme d'Euclide en précision finie.

B.3 Travail effectué

B.3.1 Tache 1 : études des pseudozéros

Etude théorique

Nous avons tout d'abord commencé par étudier la notion de ϵ-pseudozéros en nous référant à [11, 13, 19, 22].

Nous avons ensuite étudié plus succinctement la notion de pseudospectres (notion généralisant celle de pseudozéros) pour voir si cela pouvait nous apporter des éclaircissements pour la notion de pseudozéros. Nous n'avons pas trouvé de résultats sur les pseudospectres pouvant s'appliquer facilement aux pseudozéros.

D'un point de vue plus théorique, nous avons généralisé des résultats issus de [13, 19, 22] pour une norme (presque) quelconque. Tout ceci est expliqué plus en détails dans le chapitre 2.

Simulations numériques

Nous avons programmé un package MATLAB nous permettant de tracer facilement des ϵ-pseudozéros. Nous avons comparé nos propres tracés à ceux de la bibliographie [13, 19].

B.3.2 Tache 2 : étude de la notion de ϵ-pgcd

Nous avons commencé à étudier les limites de l'algorithme d'Euclide en précision finie et à regarder quelles étaient les alternatives utilisées dans la littérature. Nous avons aussi commencé à étudier la notion de ϵ-pgcd puis les différentes méthodes utilisées pour le calculer.

B.4 Outils utilisés

Pour la recherche bibliographique, nous avons utilisé MathSciNet, la version électronique des Mathematical Reviews et le Zentralblatt-MATH, version électronique du Zentralblatt fur mathematik.

Pour la programmation, nous utilisons les logiciels MATLAB version 6 et MAPLE 6. Les simulations numériques ont été effectuées sur un Sun Ultra 5 (processeur UltraSparc cadencé à 333 MHz) muni de 256 Mo de RAM.

Annexe C

Séminaires et conférences

Il s'agit de la liste des conférences et séminaires auxquels j'ai assistés.

C.1 Séminaire ALEPH & GÉODE

Ce séminaire a eu lieu le mardi 6 février 2001 au laboratoire GAGE de l'École Polytechnique dans le cadre de l'UMS MEDICIS.

Perturbation Bounds for Polynomial Roots

Arnold SCHÖNHAGE

(Université de Bonn, Allemagne)

14h, Part I (tutorial for non-experts)

Theoretical worst case a priori bounds

Let f, g be two complex monic n-th degree polynomials with root radius $\rho(f) \leq 1$, roots $u_1, ..., u_n$ of f, roots $v_1, ..., v_n$ of g, and l_1-norm $|f-g| < \epsilon$. Then the symmetrized root distance $\delta = \delta(f,g)$ = min (over all numberings of the v's) of $max_j|u_j - v_j|$ is bounded (worst case) by $\delta \leq c(n).\,\epsilon^{1/n}$; we present a sketchy proof for Schätzle's results about the sharpest constants $c(n)$, converging to 2 for $n \to \infty$. Analogous bounds hold for general (nonmonic) f and g of (formal) n-th degree satisfying $|f-g| < \epsilon.|f|$.

14h45, Pause thé/café

15h15, Part II (main research talk)

Practical a posteriori bounds for local clusters

Here $g(x) = \prod_{j=1}^{n}(x-u_j)$ with known u_j shall satisfy $|f-g| < \epsilon$ — Where are the roots z_j of f? Studying the set $W = \{w \in \mathbf{C} : |g(w)| < \epsilon\}$ yields estimates for the diameters of its components, typically containing some multiplicity of $m << n$ roots of g, and f. Then similar local bounds $|u_j - v_j| < h(m).\,\epsilon^{1/m}$ are obtained, and the methods from Part I can be used to derive similar locally sharp bounds of this kind. – Numerical examples will illustrate our findings –

C.2 29e école de printemps d'informatique théorique

Les écoles de printemps d'informatique théorique ont pour objectif de présenter à de jeunes chercheurs et à des non spécialistes les bases de sujets ayant une application grandissante en informatique et de faciliter les échanges d'informations entre spécialistes.

Elles comportent donc tant des exposés d'introduction que des conférences plus spécialisées.

Ces écoles de printemps ont acquis au fil des ans une réputation internationale et leur audience auprès des universitaires et industriels français et étrangers n'a cessé de croître grâce à la grande qualité des cours et des conférences qui y sont proposés. Cette 29ème session s'adresse aux chercheurs interesses par les aspects mathématiques et algorithmiques rencontres en "Arithmétique des Ordinateurs", et qui proviennent de domaines assez éloignés comme l'informatique, la théorie des nombres, l'analyse numérique, le calcul formel et la logique. L'école comprendra des cours et des exposés de recherche. Les cours seront consacrés aux thèmes suivants :

- Arithmétique multi-précision
- Circuits arithmétiques
- Division et racine carrée
- Algorithmes de calcul des fonctions élémentaires
- Arithmétique rationnelle Modèles de calculabilité sur les réels
- Systèmes de numération
- Arithmétique en ligne
- Théorie des nombres
- Systèmes modulaires

Les exposés de recherche comprendront des exposés relatifs à l'arithmétique des ordinateurs en géometrie algorithmique et en calcul symbolique. Des résultats récents en arithmétique modulaire et en virgule flottante seront également présentés.

```
programme

Lundi 26 mars

11:00 - 12:00 : Accueil et introduction
12:00 - 14:00 Déjeuner
14:00 - 15:30 : Systèmes de Numération, Christiane Frougny, LIAFA, Université Paris 8
15:30 - 16:00 Pause
16:00 - 18:00 : Circuits arithmétiques élémentaires, Vojin Oklobdzija,
                University of California at Davis, USA

Mardi 27 mars

09:00 - 09:45 Arithmétique asynchrone, Arnaud Tisserand, INRIA Rhône-Alpes -
              LIP, projet Arénaire
09:45 - 10:00 Pause
10:00 - 12:00 Evaluation des Fonctions Elementaires, Jean-Michel Muller,
              CNRS-LIP, projet Arénaire
12:00 - 14:00 Déjeuner
14:00 - 15:00 Automates et Transcendances, Jean-Paul Allouche, CNRS-LRI
15:00 - 15:30 Pause
```

```
15:30 - 17:00 Modèles réels, Christian Michaux, Université de Mons, Belgique
17:00 - 17:45 Arithmétique virgule flottante , Marc Daumas, CNRS-LIP,
              projet Arénaire

Mercredi 28 mars

09:00 - 10:30 Arithmétique rationnelle ,Peter Kornerup, Odense University, Danemark
10:30 - 11:00 Pause
11:00 - 12:00 Systèmes modulaires de numération ,Laurent Stéphane Didier,
              Université de Brest
12:00 - 13:30 Déjeuner
13:30 Après-midi libre

Jeudi 29 mars

09:00 - 11:00 Algorithme de calcul multi-précision ,Paul Zimmermann, INRIA
              Lorraine, projet SPACES
11:00 - 11:15 Pause
11:15 - 12:00 Algorithme RMP ,Guillaume Hanrot, INRIA Lorraine, projet SPACES
12:00 - 14:00 Déjeuner
14:00 - 15:30 Algorithmes de Division ,Paolo Montuschi, Politecnico di Torino, Italie
15:30 - 16:00 Pause
16:00 - 16:45 Le Dilemme du Fabricant de Tables , Vincent Lefèvre, INRIA
              Lorraine, projet SPACES
16:45 - 17:30 Preuves arithmétiques à l'aide du système Coq , Laurent Théry,
              INRIA Sophia, projet LEMME

Vendredi 30 mars

09:00 - 10:00 Arithmétique en ligne ,Jean-Michel Muller, CNRS-LIP, projet Arénaire
10:00 - 10:30 Pause
10:30 - 11:15 Arithmétique d'Intervalles , Nathalie Revol, INRIA Rhône-Alpes -
              Université de Lille
11:15 - 12:00 Cours et démonstrateur sur Internet d'arithmétique des
              ordinateurs, Alain Guyot, INPG
12:00 Fin de l'école - Déjeuner
```

C.3 Séminaires du LIP

Architecture des systèmes de communication pour "grappes".
Loïc Prylli, CNRS LIP
L'exposé commencera par présenter les différents types de matériel réseau pour les architectures de type «grappes». A partir de là on montrera que l'interface entre les noeuds de calcul et le réseau est une donnée toujours cruciale dans les performances du système de communication. Nous présenterons les diverses techniques systèmes permettant d'obtenir faibles latences et haut débit (accès en mode utilisateur à la carte réseau, transfert à zéro copie-mémoire, élimination des interruptions). Et nous présenterons la problématique d'implémentation d'un protocole asynchrone entre les différents niveaux de l'architecture (code sur processeur principal, logiciel embarqué, implémentation matérielle), en expliquant les problèmes d'interaction entre ces niveaux.

Segmentation de séquences par partitionnement maximalement prédictif.
Application aux séquences génétiques.

Laurent Gueguen, Laboratoire de Biométrie et de Biologie Evolutive (UCLB)

Le partitionnement maximalement prédictif sous contrainte d'ordre total est une méthode de classification qui cherche à partager des séquences d'objets qualitatifs en segments homogènes. L'homogénéité est définie selon un critère basé sur la notion de prédiction. Pour un problème donné, on se munit d'un ensemble fini de prédicteurs possibles. A chaque prédicteur,on associe une fonction - la prédiction - à valeurs réelles sur les objets de la séquence. Un segment est évalué par la somme des prédictions de tous ses éléments par un même prédicteur ad hoc. L'évaluation est ainsi un critère d'homogénéité. Une partition de la séquence est évaluée par la somme des prédictions sur ses segments. Sur la base de cette évaluation on veut pouvoir, grâce au prédicteur de chaque segment d'une partition, disposer d'un résumé de la séquence qui mette en relief une éventuelle structure de cette séquence. Il faut alors estimer le nombre de segments en lequel il est le plus judicieux d'opérer cette partition.

Je présente un algorithme qui, sur la donnée d'une séquence, d'un ensemble de prédicteurs et d'un entier k, construit l'ensemble des partitions optimales en i segments de la séquence pour tout i entre 1 et k. C'est ce que j'appelle un partitionnement. Ceci permet aussi d'observer l'évolution des partitions en fonction de leurs nombres de classes.

L'algorithme présenté a une complexité en temps linéaire avec la longueur de la séquence, la taille de l'ensemble des prédicteurs et le nombre maximum de segments. On peut alors partitionner de très grandes séquences, et les séquences biologiques en constituent un domaine naturel d'expérimentation. Je présenterai ainsi entre autres une méthode de détection des origines de réplication de génomes bactériens.

Basse consommation d'énergie et opérateurs arithmétiques matériels
Arnaud TISSERAND (INRIA-LIP)

La basse consommation d'énergie est une contrainte de conception de plus en plus forte pour les circuits intégrés. Avec le développement actuel des appareils portables, les circuits intégrés doivent effectuer des tâches complexes tout en ne consommant que très peu d'énergie. Même dans les processeurs de bureau, la consommation d'énergie peut être critique (problème de dissipation, densité de courant élevée, écologie, bruits...).

Cet exposé sera décomposé en deux grandes parties. Dans la première, nous allons regarder pourquoi les circuits intégrés consomment et quelles sont les techniques classiques pour réduire la consommation. En particulier, nous allons voir que la basse consommation peut être abordée depuis le niveau algorithmique jusqu'au plus bas niveau électrique. Dans la seconde partie de l'exposé, nous regarderons le cas de quelques opérateurs arithmétiques matériels simples (addition et multiplication essentiellement).

CompoVis : Composants Logiciels Corba pour la Visualisation Scientifique sur Internet
Jean-Marc Pierson (LIL)

CompoVis est une architecture logicielle distribuee offrant a un utilisateur connecte par Internet un ensemble de services de calcul executes sur une grille de calcul. Les principales caracteristiques de l'architecture sont son extensibilite, sa portabilite, la mobilite des services, et son integration avec les technologies Internet, entre autres les bases de donnees. Le coeur de notre systeme repose sur la technologie Corba et des composants logiciels, facilement interoperables. Cree au depart dans le but de creer un service de visualisation sur Internet, un prototype aujourd'hui operationnel de l'architecture met en jeu des technologies comme les Applets Java, les Servlets, une base de donnee MySQL, et le bus Corba de Visibroker pour de simples services de calculs. Toutefois son extension vers les services de visualisation est en cours actuellement. Dans mon intervention, je m'attacherai plus particulierement a montrer la motivation de notre approche, la modelisation que nous proposons du probleme et les choix techniques adoptes.

Active Reliable Multicast Protocols for Efficient Data Distribution on the Internet

Pham CongDuc (RESAM)

In contrast to traditinal networking, active networking allows routers themselves to play an active role by executing application-dependent services on incoming packets. The talk with first present the current status of Internet networks and then the enhancements active networking could bring to this communication infrastructure. After describing what active networking is (and is not), the talk will move on describing one particular application: active reliable multicast. Recently, the use of active network concepts have been proposed in the multicast research community. Reliable multicast protocols have gained popularity with active services contribution where routers implement additional functionalities. Contributing mainly to feedback implosion problems, retransmission scoping and cache of data, these active protocols open new perspectives for achieving high throughput and low latencies on wide-area networks. The talk will present the general strategies behind active reliable multicast protocols, some performance results and some insights on dimensioning active networks.

C.4 Conférences d'intérêt général

Les origines de la vie, vues par un géologue
par Pierre Thomas, Professeur à l'ENS Lyon et planétologue.
Jeudi 26 avril 2001, 18h00, amphi DSM

Le rôle des erreurs dans le développement des mathématiques
par Etienne Ghys, directeur de l'unité de mathématiques pures et appliquées à l'ENS Lyon
Jeudi 03 mai 2001, 18h00, amphi DSM

La théorie de la relativité restreint et générale
par Julien Salomez, élève en physique à l'ENS Lyon
Jeudi 10 mai 2001, 18h00, amphi DSM

C.5 Séminaire "La mesure de Lebesgue à 100 ans !!!"

Le 29 avril 1901, Lebesgue publiait une note aux Comptes-Rendus de l'Académie des Sciences intitulée : *Sur une généralisation de l'intégrale définie*.

Il s'agit de l'acte de naissance de l'intégrale de Lebesgue. Cette note est remarquable. Il serait difficile de dire en moins de mots ce qu'est un ensemble mesurable et sa mesure, une fonction intégrable et son intégrale.

Pour commémorer ce centenaire important, l'Ecole Normale Supérieure de Lyon, en association avec la Société Mathématique de France, organise une rencontre mathématique.

PROGRAMME

```
Vendredi 27 avril 2001

11 h 00 : Gustave CHOQUET
La mesure et l'intégration, à la charnière de deux siècles.

14 h 00 : Gérard BEN AROUS
Lebesgue, Wiener, Wigner.

15 h 15 : Pierre de la HARPE
Mesures finiment additives et paradoxes.

17 h 00 : Bruno SEVENNEC
```

Mesure invariante et équirépartition dans les groupes compacts.

Samedi 28 avril 2001 :

9 h 30 : Jean DHOMBRES
Une autre origine de l'intégrale de Lebesgue : la théorie de l'approximation et le rôle du théorème de Weierstrass.

11 h 00 : Jean-Pierre KAHANE
Séries trigonométriques, séries et intégrales.