

Chapitre 4

Sur les liens entre quelques modèles

Dans ce chapitre, nous présentons quelques-uns de nos résultats de comparaisons entre différents modèles de calculs à temps continu. Nous commençons par nous intéresser à la puissance du General Purpose Analog Computer de Shannon [Shannon, 1941], et aux problèmes de Cauchy polynomiaux¹. Par la suite, nous nous intéressons à des sous-classes de fonctions \mathbb{R} -récurives. Les fonctions \mathbb{R} -récurives ont été introduites par [Moore, 1998]. Nous les relierons aussi aux fonctions calculables en analyse récursive.

Les résultats de ce chapitre ont tous été obtenus en collaborations. Les résultats sur le GPAC sont le fruit d’une collaboration avec Manuel Campagnolo, Daniel Graça et Emmanuel Hainry. Les résultats sur les fonctions \mathbb{R} -récurives relèvent de la thèse d’Emmanuel Hainry, que nous avons eu le plaisir d’encadrer.

4.1 Une thèse de Church-Turing pour les modèles à temps continu ?

Selon la thèse de Church-Turing² tous les modèles *raisonnables* et suffisamment puissants de calculs digitaux sont équivalents au modèle de la machine de Turing.

Aucun résultat similaire n’est connu lorsqu’on considère les calculs analogiques à temps continu³. Alors que de nombreux modèles ont été étudiés, comme le modèle de Blum Shub et Smale [Blum et al., 1989], les fonctions \mathbb{R} -récurives de Moore [Moore, 1996], les réseaux de neurones [Siegelmann, 1999], ou l’analyse récursive [Pour-El and Richards, 1989], [Ko, 1991], [Weihrauch, 2000], aucun d’entre eux ne peut se prétendre “universel”.

Cela est dû en partie au fait que peu de relations entre les modèles sont connues. En outre, certains des résultats affirment que les modèles ne sont pas équivalents, rendant l’espoir d’une thèse de Church-Turing pour les modèles analogiques inaccessible. Par exemple, le modèle de Blum Shub et Smale autorise les fonctions discontinues, alors que seules les fonctions continues sont calculables en analyse récursive [Weihrauch, 2000].

Cependant, nous allons montrer que cela n’est peut-être pas si utopique.

4.2 GPAC, problèmes de Cauchy polynomiaux et analyse récursive : trois paradigmes équivalents

Nous avons prouvé l’équivalence de deux modèles de calculs qui étaient jusque-là considérés comme non équivalents : l’analyse récursive et le *General Purpose Analog Computer* (GPAC) de Claude Shannon.

¹Observons que, de façon informelle, nous montrons quelque part que les machines universelles du début du XXI^{ème} siècle valent bien celles du début du XXI^{ème} : GPAC, et machines de Turing, même combat. 1 partout

²Voir aussi l’annexe A et ses discussions sur les variantes de cette thèse.

³Une thèse de Church-Turing a été proposée par Siegelmann et Sontag pour les modèles à espace continu et temps discret [Siegelmann and Sontag, 1994], [Siegelmann, 1995].

4.2.1 Point de vue historique

Le GPAC a été introduit en 1941 par Shannon [Shannon, 1941] comme un modèle mathématique d'un dispositif analogique : l'Analyseur Différentiel. Le tout premier analyseur différentiel a été construit au MIT en 1931 sous la direction de Vannevar Bush [Bush, 1931]. Les analyseurs différentiels ont été utilisés des années 1930 aux débuts des années 1960 pour résoudre des problèmes numériques, comme par exemple des problèmes balistiques. Les toutes premières versions étaient mécaniques. Plus tard, les analyseurs différentiels sont devenus électroniques. Nous renvoyons au chapitre 3 pour plus de détails.

Un GPAC peut être vu comme un circuit constitué d'éléments de base interconnectés. Ces éléments réalisent chacun un des types d'opérations de la figure 4.1, où les entrées sont des fonctions d'une variable indépendante, appelée le *temps* (dans un analyseur différentiel électronique, les entrées correspondent à des tensions). Ces éléments *boîtes noires* réalisent des additions, des multiplications, génèrent des constantes, ou intègrent.

Alors que beaucoup des fonctions sont connues pour être générables par un GPAC, une exception notable est la fonction

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt.$$

Ce fait est déjà observé par Shannon dans son article original [Shannon, 1941].

Si nous avons à l'esprit que cette fonction est connue pour être calculable dans le cadre de l'analyse récursive [Pour-El and Richards, 1989], ce résultat a souvent été interprété comme la preuve que le GPAC est un modèle trop faible, et en tout cas plus faible que l'analyse récursive.

Pendant, cela est plus dû à la notion de GPAC-calculabilité, plutôt qu'au modèle. En effet, on considère en général qu'un GPAC doit calculer une fonction en *temps réel*, une notion très restrictive de calcul. Si nous changeons la notion de calcul en une notion de *calcul convergent*, comme celle utilisée en analyse récursive, il a été montré récemment par Daniel Graça que la fonction Γ devient calculable [Graça, 2004].

Dans la suite de cette section, nous allons renforcer fortement ce résultat en montrant qu'en fait *toute fonction calculable* sur un domaine compact peut être calculée par un GPAC dans ce sens⁴. Réciproquement, nous montrons qu'avec des hypothèses raisonnables, la réciproque est aussi vraie.

En d'autres termes, nous prouvons que la puissance de calcul des GPAC coïncide avec celle de l'analyse récursive.

Notons, que ce résultat est nettement plus fort que le résultat de [Shannon, 1941] sur l'approximation des fonctions continues réelles par le GPAC, basé lui sur l'approximation de fonctions par le théorème de Weierstrass (toute fonction continue sur un compact peut être approximée à ε près par un polynôme de degré suffisamment élevé).

Observons que la notion de "calcul convergent" correspond à une classe particulière de fonctions \mathbb{R} -récursives [Moore, 1998], [Mycka and Costa, 2004], [Bournez and Hainry, 2006].

Observons en outre qu'il a été montré dans [Graça et al., 2005] que les machines de Turing peuvent être simulées par des GPACs. Nous montrons quelque part que cela peut être étendu aux machines de Turing de type 2, ou avec oracles, utilisées en analyse récursive : voir [Weihrach, 2000].

⁴Sauf explicitement précisé, dans ce chapitre, "fonction calculable" signifie "fonction calculable dans le sens de l'analyse récursive."

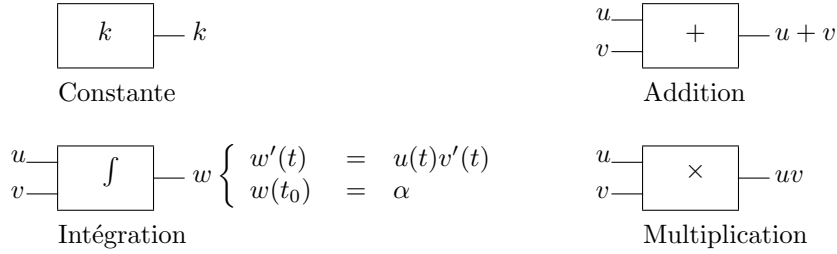


FIG. 4.1 – Les éléments de base d'un GPAC.

4.2.2 Le GPAC

Le GPAC a été initialement introduit par Shannon dans [Shannon, 1941], et le modèle a été raffiné dans les articles [Pour-El, 1974], [Lipshitz and Rubel, 1987], [Graça and Costa, 2003], et [Graça, 2004].

Le modèle consiste simplement en les familles de circuits que l'on peut construire avec les unités de base présentées dans la figure 4.1. Les connexions avec des rétroactions (feedback) sont autorisées, si ce n'est que certaines restrictions existent, puisque sinon cela peut mener soit à des systèmes sans évolution, ou dont la dynamique n'est pas définie (par exemple si l'on branche la sortie d'un additionneur sur son entrée), soit avec des comportements indésirables (comme la possibilité de plusieurs sorties pour une entrée fixée). Nous renvoyons pour les détails à [Graça and Costa, 2003].

Shannon, dans son article original, mentionne déjà que le GPAC génère tous les polynômes, la fonction exponentielle, les fonctions trigonométriques, et leurs inverses. Plus généralement, Shannon affirme que les fonctions générées par un GPAC sont différentiellement algébriques, c'est-à-dire satisfont les conditions de la définition suivante.

Définition 1 *La fonction unaire y est différentiellement algébrique (d.a.) sur l'intervalle I s'il existe un polynôme non nul p à coefficients réels tel que*

$$p(t, y, y', \dots, y^{(n)}) = 0, \quad \text{on } I. \quad (4.1)$$

Comme corollaire, puisque l'on sait que la fonction

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$$

n'est pas différentiellement algébrique [Hölder, 1887], nous obtenons le résultat de non-calculabilité⁵ suivant, présent dans [Shannon, 1941].

Proposition 1 *La fonction Γ ne peut pas être générée par un GPAC.*

Cependant, la preuve de Shannon, qui relie les fonctions générées par un GPAC avec les fonctions différentiellement algébriques, est incomplète (cela a été montré et partiellement corrigé dans [Pour-El, 1974], [Lipshitz and Rubel, 1987]). En fait, comme montré dans [Graça and Costa, 2003], le modèle initial du GPAC de Shannon n'est pas bien (complètement) défini.

⁵Il est très instructif de comparer le raisonnement utilisé, par rapport aux arguments usuels pour prouver la non calculabilité d'un problème ou d'une fonction pour les machines de Turing.

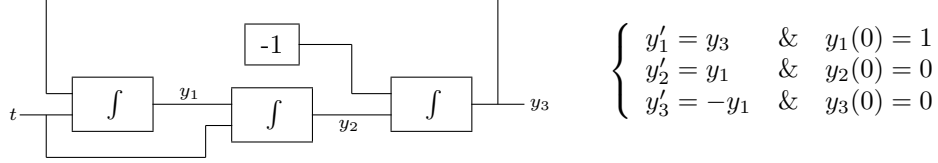


FIG. 4.2 – Génération de cos et sin avec un GPAC : version circuit sur la gauche et version équation différentielle sur la droite. On a $y_1 = \cos$, $y_2 = \sin$, $y_3 = -\sin$.

4.2.3 Problèmes de Cauchy polynomiaux

Pour la classe mieux définie de GPAC dans [Graça and Costa, 2003], la propriété suivante est vraie :

Proposition 2 *Une fonction scalaire $f : \mathbb{R} \rightarrow \mathbb{R}$ est générée par un GPAC si et seulement si elle est une projection d'une solution d'un problème de Cauchy polynomial, c'est-à-dire d'un système d'équations différentielles*

$$y' = p(y, t), \quad (4.2)$$

où p est un vecteur de polynômes. Une fonction $f : \mathbb{R} \rightarrow \mathbb{R}^k$ est générée par un GPAC si et seulement si toutes ses projections le sont.

A partir de maintenant, nous considérerons principalement qu'un GPAC correspond à une équation différentielle du type de (4.2), c'est-à-dire un problème de Cauchy polynomial selon la terminologie du chapitre 1. Pour un exemple d'utilisation de la proposition précédente, voir la figure 4.2.

Les fonctions générables par GPAC dans ce sens sont évidemment différentiellement algébriques.

Si on se rappelle que les solutions d'une équation différentielle ordinaire analytique sont toujours analytiques (voir par exemple [Arnold, 1992]), une autre conséquence est le résultat suivant :

Corollaire 1 *Si f est une fonction générée par un GPAC, alors elle est analytique.*

Comme nous l'avons vu dans la proposition 1, la fonction Γ n'est pas générable par un GPAC.

Cependant, il a été récemment prouvé par Daniel Graça qu'elle peut être calculée par un GPAC si nous modifions la notion de GPAC calculabilité en une notion de "calcul convergent" [Graça, 2004].

Dans les définitions originales de Shannon de GPAC, rien n'est supposé sur les constantes et sur les conditions initiales de l'équation différentielle (4.2). En particulier, il peut y avoir des réels non-calculables. Ce type de GPAC mène de façon triviale à des calculs non simulables par machines de Turing. Pour éviter cela, nous restreignons le modèle de [Graça, 2004] comme suit.

Définition 2 *Une fonction $f : [a, b] \rightarrow \mathbb{R}$ est GPAC-calculable⁶ et si et seulement s'il existe un polynôme calculable $p : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ et $n - 1$ valeurs calculables $\alpha_1, \dots, \alpha_{n-1}$, et des indices $i, j \in \{1, \dots, n\}$, tels que, pour tout $x \in [a, b]$,*

- si (y_1, \dots, y_n) est la solution⁷ du problème de Cauchy $y' = p(y, t)$ avec $y(0) = (\alpha_1, \dots, \alpha_{n-1}, x)$ au temps $t_0 = 0$,
- alors $\lim_{t \rightarrow \infty} y_j(t) = 0$ et $|f(x) - y_i(t)| \leq y_j(t)$.

Nous remarquerons que $\alpha_1, \dots, \alpha_{n-1}$ sont des paramètres auxiliaires nécessaires pour calculer f .

Le résultat de [Graça, 2004] peut alors se formuler en.

Proposition 3 ([Graça, 2004]) *La fonction Γ est GPAC-calculable.*

⁶Notons que dans ce chapitre, la terminologie GPAC calculable réfère à cette notion particulière. L'expression "générée par un GPAC" correspond à la notion de calcul de Shannon en temps réel.

⁷Nous supposons que chacun des $y_k(t)$ est défini pour tout $t \geq 0$.

4.2.4 Analyse récursive

Le principe sous-jacent à l'analyse récursive est de définir les fonctions calculables sur les réels en considérant des fonctionnelles sur des suites de rationnels rapidement convergentes [Weihrauch, 2000].

Soit $\nu_{\mathbb{Q}} : \mathbb{N} \rightarrow \mathbb{Q}$ la représentation suivante⁸ des nombres rationnels par les entiers :

$$\nu_{\mathbb{Q}}(\langle p, r, q \rangle) \mapsto \frac{p - r}{q + 1},$$

où $\langle \cdot, \cdot, \cdot \rangle : \mathbb{N}^3 \rightarrow \mathbb{N}$ est une bijection calculable.

Une suite d'entiers $(x_i)_{i \in \mathbb{N}}$ représente un réel x si $(\nu_{\mathbb{Q}}(x_i))$ converge rapidement vers x , noté par $(x_i) \rightsquigarrow x$, dans le sens suivant :

$$\forall i, |\nu_{\mathbb{Q}}(x_i) - x| < \exp(-i).$$

Pour une suite de k -tuples $(\mathbf{x}_i)_{i \in \mathbb{N}}$, nous écrivons $(\mathbf{x}_i) \rightsquigarrow \mathbf{x}$ lorsque cela est vrai composante par composante.

Définition 3 (Analyse récursive [Weihrauch, 2000]) *On dit qu'une fonction $f : \mathbb{R}^k \rightarrow \mathbb{R}$ est calculable s'il existe une fonctionnelle récursive $\Phi : (\mathbb{N}^k)^{\mathbb{N}} \times \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout $\mathbf{x} \in \mathbb{R}^k$, pour toute suite $X = (\mathbf{x}_n) \in (\mathbb{N}^k)^{\mathbb{N}}$, nous avons $(\Phi(X, j))_j \rightsquigarrow f(\mathbf{x})$ lorsque $X \rightsquigarrow \mathbf{x}$.*

Une fonction $f : \mathbb{R}^k \rightarrow \mathbb{R}^l$, avec $l > 1$, sera dite calculable, si ses projections le sont.

4.2.5 Notre résultat principal

Dans notre article [Bournez et al., 2006], nous prouvons que cela est vrai en fait pour toutes les fonctions calculables sur un compact.

En effet, nous montrons que si une fonction réelle f , définie sur un compact, est calculable, alors elle est GPAC-calculable. Réciproquement, nous montrons que si f , définie sur un compact, est GPAC-calculable, alors elle est calculable.

Théorème 1 (GPAC=Analyse Récursive) *Une fonction $f : [a, b] \rightarrow \mathbb{R}$ est calculable si et seulement si elle est GPAC calculable.*

Autrement dit, GPAC, problèmes de Cauchy polynomiaux, et analyse récursive sont trois paradigmes de calculs équivalents.

Corollaire 2 (Problèmes de Cauchy polynomiaux=Analyse Récursive) *Une fonction $f : [a, b] \rightarrow \mathbb{R}$ est calculable si et seulement si elle est correspond à la limite d'une fonction définissable par un problème de Cauchy polynomial dans le sens de la définition 2.*

Il peut être utile de mettre en rapport ce résultat d'une part sur nos remarques dans les chapitres 1 et 2, sur la puissance de modélisation des problèmes de Cauchy polynomiaux, et d'autre part, sur les liens avec une éventuelle thèse de Church-Turing pour les modèles analogiques.

D'un point de vue des calculs, nos résultats suggèrent que les problèmes de Cauchy polynomiaux et les GPACs sont de vraies contreparties aux machines de Turing.

Nous passons maintenant à une autre série de résultats, reliant fonctions \mathbb{R} -récursives et analyse récursive.

4.3 Fonctions \mathbb{R} -récursives et analyse récursive

4.3.1 Fonctions \mathbb{R} -récursives

Dans [Moore, 1996], Moore a introduit une classe de fonctions sur les réels inspirée de la définition algébrique des fonctions calculables sur les entiers : en observant que l'analogie continu d'une récursion

⁸D'autres représentations naturelles des nombres rationnels peuvent être utilisées, et donnent la même classe de fonctions calculées : voir [Weihrauch, 2000].

primitive est une équation différentielle (une intégration), Moore propose de considérer la classe des fonctions \mathbb{R} -récurives, définie comme la plus petite classe de fonctions qui contient certaines fonctions de base, et qui est close par composition, intégration, et par minimisation.

Cette classe de fonctions, aussi étudiée par des articles comme [Mycka, 2003b], [Mycka and Costa, 2004], [Mycka and Costa, 2006a], [Mycka and Costa, 2006b], [Mycka and Costa, 2005], peut se relier aux fonctions calculées par GPAC (aux fonctions générées par GPAC, selon la terminologie de la section précédente) : voir [Moore, 1996], corrigé par [Graça and Costa, 2003].

Si l'on met de côté les objections à propos de la réalité physique de l'opérateur de minimisation considéré dans l'article initial [Moore, 1996], les définitions originales des classes de [Moore, 1996] souffrent de plusieurs problèmes techniques⁹. Au moins une partie de ces problèmes permet d'utiliser un "compression trick" (une autre incarnation du paradoxe de Zénon), pour simuler en un temps borné un nombre non borné de transitions discrètes, et permet de reconnaître des langages arithmétiques.

Dans la série d'articles [Campagnolo et al., 2000], [Campagnolo et al., 2002], [Campagnolo, 2001], Campagnolo, Costa et Moore proposent de considérer la classe \mathcal{L} (bien définie) de fonctions qui correspond à la plus petite classe de fonctions qui ne contient certaines fonctions de base, et qui est close par composition et intégration *linéaire*.

La classe \mathcal{L} est reliée aux fonctions élémentaires sur les entiers en théorie de la calculabilité classique, et aux fonctions sur les réels élémentairement calculables en analyse récursive (les fonctions élémentairement calculables en analyse récursive sont aussi discutées dans [Zhou, 1997]) : toute fonction de la classe \mathcal{L} est élémentairement calculable au sens de l'analyse récursive, et réciproquement, toute fonction sur les entiers élémentaire est la restriction aux entiers d'une fonction qui appartient à la classe \mathcal{L} [Campagnolo et al., 2002], [Campagnolo, 2001].

Ces résultats établissent donc des liens entre les fonctions \mathbb{R} -récurives, le GPAC et la calculabilité classique sur les entiers.

4.3.2 Présentation de nos résultats

Les résultats précédents ne fournissent pas une caractérisation de *toutes* les fonctions sur les réels qui sont élémentairement calculables, mais seulement de celles qui préservent les entiers.

Nous avons prouvé qu'il était possible d'aller beaucoup plus loin.

Théorème 2 *Pour les fonctions sur les réels de classe \mathcal{C}^2 définies sur un produit d'intervalles compacts à extrémités rationnelles, f est élémentairement calculable dans le sens de l'analyse récursive si et seulement si f appartient à la plus petite classe de fonctions qui contient certaines fonctions de base, et qui est close par composition, intégration linéaire et un schéma limite simple.*

La force de ce résultat réside dans la caractérisation : le "si et seulement si". Observons que le sens indirect sans le schéma limite (f appartient à la plus petite classe de fonctions qui contient certaines fonctions de base, et qui est close par composition, intégration linéaire implique f élémentairement calculable) avait été établi dans [Campagnolo et al., 2000], [Campagnolo et al., 2002], [Campagnolo, 2001].

Nous avons étendu ce résultat à une caractérisation de tous les niveaux de la hiérarchie de Grzegorzczuk (observons que le théorème précédent peut se voir comme un corollaire du théorème qui suit).

Théorème 3 *Pour les fonctions sur les réels de classe \mathcal{C}^2 définies sur un produit d'intervalles compacts à extrémités rationnelles, f est calculable dans le sens de l'analyse récursive dans le niveau $n \geq 3$ de la hiérarchie de Grzegorzczuk si et seulement si f appartient à la plus petite classe de fonctions qui contient certaines (autres) fonctions de base, et qui est close par composition, intégration linéaire et un schéma limite simple.*

⁹Par exemple, des fonctions pas bien définies sont considérées, $\infty \times 0$ est toujours considéré comme 0, etc. . . Certains de ces problèmes sont discutés dans [Campagnolo et al., 2000], [Campagnolo et al., 2002], [Campagnolo, 2001], et en fait même par son auteur dans [Moore, 1996].

Là encore, la force est dans la caractérisation. Le sens indirect sans le schéma limite est prouvé dans [Campagnolo et al., 2000], [Campagnolo et al., 2002], [Campagnolo, 2001].

D'autre part, nous avons étendu cela aux fonctions calculables au sens de l'analyse récursive, et pas seulement aux fonctions *élémentairement calculables*.

Nous avons montré qu'il était possible de définir un schéma de minimisation, assez élégant, pour que :

Théorème 4 *Pour les fonctions sur les réels de classe C^2 définies sur un produit d'intervalles compacts à extrémités rationnelles, f est calculable dans le sens de l'analyse récursive si et seulement si f appartient à la plus petite classe de fonctions qui contient certaines fonctions de base, et qui est close par composition, intégration linéaire, minimisation, et un schéma limite simple.*

Nous présentons maintenant concrètement ces résultats.

4.3.3 Préliminaires mathématiques

Les résultats mathématiques suivants, bien connus, peuvent aider à motiver certains de nos schémas : voir par exemple [Ramis et al., 1995] pour la preuve du premier. Le second, très simple à établir, est prouvé explicitement dans [Bournez and Hainry, 2005].

Lemme 1 (Théorème des fonctions implicites) *Soit $f : \mathcal{D} \times \mathcal{I} \subset \mathbb{R}^{k+1} \rightarrow \mathbb{R}$ une fonction, où $\mathcal{D} \times \mathcal{I}$ est un produit d'intervalles fermés, de classe C^k , for $k \geq 1$. Supposons que pour tout $\mathbf{x} \in \mathcal{D}$, l'équation $f(\mathbf{x}, y) = 0$ possède exactement une solution y_0 et que cet y_0 appartient à l'intérieur de l'intervalle \mathcal{I} . Supposons, que pour tout \mathbf{x} ,*

$$\frac{\partial f}{\partial y}(\mathbf{x}, y_0) \neq 0$$

en y_0 .

Alors la fonction $g : \mathbb{R}^k \rightarrow \mathbb{R}$ qui envoie \mathbf{x} sur la racine correspondante y_0 est définie sur \mathcal{D} et est aussi de classe C^k .

Lemme 2 (Résultat de convergence simple) *Soit $F : \mathbb{R} \times \mathcal{V} \subset \mathbb{R}^{k+1} \rightarrow \mathbb{R}^l$ une fonction de classe C^1 , et $\beta(\mathbf{x}) : \mathcal{V} \rightarrow \mathbb{R}$, $K(\mathbf{x}) : \mathcal{V} \rightarrow \mathbb{R}$ des fonctions continues. Supposons que pour tout t et \mathbf{x} ,*

$$\left\| \frac{\partial F}{\partial t}(t, \mathbf{x}) \right\| \leq K(\mathbf{x}) \exp(-t\beta(\mathbf{x})).$$

Soit \mathcal{D} le sous ensemble des $\mathbf{x} \in \mathcal{V}$ tels que $\beta(\mathbf{x}) > 0$.

Alors,

- pour tout $\mathbf{x} \in \mathcal{D}$, $F(t, \mathbf{x})$ possède une limite $L(\mathbf{x})$ en $t = +\infty$.
- La fonction $L(\mathbf{x})$ est une fonction continue.
- En outre

$$\|F(t, \mathbf{x}) - L(\mathbf{x})\| \leq \frac{K(\mathbf{x}) \exp(-t\beta(\mathbf{x}))}{\beta(\mathbf{x})}.$$

4.3.4 Théorie classique de la calculabilité

La théorie de la récursion classique parle de fonctions sur les entiers. La plupart des classes de cette théorie peuvent se caractériser comme la clôture d'un ensemble de fonctions de base, par un nombre fini de règles pour former de nouvelles fonctions [Clote, 1998], [Cutland, 1980], [Rose, 1984], [Odifreddi, 1992] : étant donné un ensemble \mathcal{F} de fonctions, et un ensemble \mathcal{O} d'opérateurs sur les fonctions (un opérateur est une fonction qui envoie une ou plusieurs fonctions sur une nouvelle fonction), $[\mathcal{F}; \mathcal{O}]$ désignera la clôture de l'ensemble F par applications des opérations de \mathcal{O} .

Proposition 4 (Cas classique [Cutland, 1980]) *Soit f une fonction de \mathbb{N}^k vers \mathbb{N} pour $k \in \mathbb{N}$.*

La fonction f est

– élémentaire si et seulement si elle appartient à

$$\mathcal{E} = [0, S, U, +, \ominus, \times; \text{COMP}, \text{BSUM}, \text{BPROD}];$$

– dans la classe \mathcal{E}_n de la hiérarchie de Grzegorzcyk ($n \geq 3$) si et seulement si elle appartient à

$$\mathcal{E}_n = [0, S, U, +, \ominus, \times, E_{n-1}; \text{COMP}, \text{BSUM}, \text{BPROD}];$$

– primitive récursive si et seulement si elle appartient à

$$\mathcal{PR} = [0, S, U; \text{COMP}, \text{REC}];$$

– récursive¹⁰ si et seulement si elle appartient à

$$\mathcal{Rec} = [0, S, U; \text{COMP}, \text{REC}, \text{MU}].$$

Une fonction $f : \mathbb{N}^k \rightarrow \mathbb{N}^l$ est élémentaire (resp : primitive récursive, récursive, \mathcal{E}_n) si et seulement si ses projections le sont.

Les fonctions de base $0, (U_i^m)_{i,m \in \mathbb{N}}, S, +, \ominus$ et les opérateurs $\text{BSUM}, \text{BPROD}, \text{COMP}, \text{REC}, \text{MU}$ sont donnés par

1. 0 est la constante 0 ;
2. $U_i^m : \mathbb{N}^m \rightarrow \mathbb{N}, U_i^m : (n_1, \dots, n_m) \mapsto n_i$;
3. $S : \mathbb{N} \rightarrow \mathbb{N}, S : n \mapsto n + 1$;
4. $+$: $\mathbb{N}^2 \rightarrow \mathbb{N}, + : (n_1, n_2) \mapsto n_1 + n_2$;
5. \times : $\mathbb{N}^2 \rightarrow \mathbb{N}, \times : (n_1, n_2) \mapsto n_1 n_2$;
6. \ominus : $\mathbb{N}^2 \rightarrow \mathbb{N}, \ominus : (n_1, n_2) \mapsto \max(0, n_1 - n_2)$;
7. BSUM : somme bornée. Étant donnée une fonction f , $h = \text{BSUM}(f)$ est défini par

$$h : (\mathbf{x}, y) \mapsto \sum_{z < y} f(\mathbf{x}, z);$$

8. BPROD : produit borné. Étant donnée une fonction f , $h = \text{BPROD}(f)$ est défini par

$$h : (\mathbf{x}, y) \mapsto \prod_{z < y} f(\mathbf{x}, z);$$

9. COMP : composition. Étant données les fonctions f_1, \dots, f_p et g , $h = \text{COMP}(f_1, \dots, f_p, g)$ est définie comme la fonction qui satisfait

$$h(\mathbf{x}) = g(f_1(\mathbf{x}), \dots, f_p(\mathbf{x}));$$

10. REC : récursion primitive. Étant données les fonctions f et g , $h = \text{REC}(f, g)$ est défini comme la fonction qui satisfait

$$\begin{cases} h(\mathbf{x}, 0) & = f(\mathbf{x}) \\ h(\mathbf{x}, n + 1) & = g(\mathbf{x}, n, h(\mathbf{x}, n)); \end{cases}$$

11. MU : minimisation. Étant donnée une fonction f , la fonction μf est définie pour tout les \mathbf{x} pour lesquels il y a un y tel que $\forall z \leq y, f(\mathbf{x}, z)$ est défini et $f(\mathbf{x}, y) = 0$. Pour un tel \mathbf{x} , la minimisation de f est donnée par

$$\mu f : \mathbf{x} \mapsto \inf\{y; f(\mathbf{x}, y) = 0\}.$$

¹⁰Cette classe est souvent appelée la classe des fonctions *partielles récursives* car elle contient des fonctions partielles, par oppositions aux fonctions totales récursives.

12. Les fonctions E_n , impliquées dans la définition des classes \mathcal{E}_n de la hiérarchie de Grzegorzcyk, sont définies par récurrence comme suit (lorsque f est une fonction, $f^{[d]}$ dénote sa d ème itérée : $f^{[0]}(\mathbf{x}) = x$, $f^{[d+1]}(\mathbf{x}) = f(f^{[d]}(\mathbf{x}))$) :
- (a) $E_0(x, y) = x + y$,
 - (b) $E_1(x) = (x + 1) \times (y + 1)$,
 - (c) $E_2(x) = 2^x$;
 - (d) $E_{n+1}(x) = E_n^{[x]}(1)$ pour $n \geq 2$.

On a (voir [Cutland, 1980])

$$\mathcal{E} \subseteq \mathcal{PR} \subseteq \mathcal{Rec},$$

et les inclusions sont strictes. On a aussi

$$\mathcal{E}_3 = \mathcal{E}$$

et

$$\mathcal{PR} = \cup_i \mathcal{E}_i.$$

Si $\text{TIME}(t)$ et $\text{SPACE}(t)$ dénotent les classes de fonctions qui sont calculables en temps et espace t , alors pour tout $n \geq 3$,

$$\mathcal{E}_n = \text{TIME}(\mathcal{E}_n) = \text{SPACE}(\mathcal{E}_n),$$

et

$$\mathcal{E} = \text{TIME}(\mathcal{E}),$$

$$\mathcal{E} = \text{SPACE}(\mathcal{E})$$

et

$$\mathcal{PR} = \text{TIME}(\mathcal{PR}) = \text{SPACE}(\mathcal{PR}).$$

La classe \mathcal{PR} correspond aux fonctions calculables par des programmes qui utilisent des boucles *For-Next*. La classe \mathcal{E} correspond aux fonctions calculables bornées par une itérée de l'exponentielle. Au plus deux boucles *For-Next* imbriquées sont nécessaires pour une fonction de la classe \mathcal{E} , alors que les fonctions générales de la classe \mathcal{PR} peuvent nécessiter un nombre arbitrairement grand de boucles imbriquées [Cutland, 1980].

Terminons cette présentation en observant que l'opérateur de minimisation peut se renforcer en un opérateur de minimisation unique, comme suit.

Proposition 5 Une fonction f de \mathbb{N}^k vers \mathbb{N}^l , pour $k, l \in \mathbb{N}$, est récursive si et seulement si ses projections appartiennent à $[0, U, S; \text{COMP}, \text{REC}, \text{UMU}]$ où l'opérateur UMU est défini comme suit :

1. UMU : minimisation unique. Étant donnée une fonction f , telle que pour tout \mathbf{x} , il y a au plus un y avec $f(\mathbf{x}, y)$ défini et égal à 0, la minimisation unique de f , notée par $!\mu(f)(\mathbf{x})$, est définie pour tous les \mathbf{x} pour lesquels il y a un (unique) y avec $f(\mathbf{x}, y) = 0$. Pour un tel \mathbf{x} , $!\mu(f)(\mathbf{x})$ est défini comme cet y .

En calculabilité classique, des objets plus généraux que les fonctions sur les entiers peuvent être considérés, en particulier des fonctionnelles, c'est-à-dire des fonctions $\Phi : (\mathbb{N}^m)^{\mathbb{N}} \times \mathbb{N}^k \rightarrow \mathbb{N}^l$. Une fonctionnelle sera dite *élémentaire* (ou \mathcal{E}_n , *primitive récursivement*, *récursivement*) calculable lorsqu'elle appartient¹¹ à la classe correspondante.

En analyse récursive, une fonction f sur les réels sera dite *élémentaire* (respectivement \mathcal{E}_n) calculable si la fonctionnelle correspondante Φ dans la définition 3 l'est. La classe des fonctions calculables (respectivement élémentairement calculables, \mathcal{E}_n) sur les réels sera notée $\mathcal{Rec}(\mathbb{R})$ (resp. $\mathcal{E}(\mathbb{R})$, $\mathcal{E}_n(\mathbb{R})$).

¹¹Formellement, une fonction f sur les entiers peut être considérée comme une fonctionnelle $\bar{f} : (V, \bar{n}) \mapsto f(\bar{n})$. De façon similaire, un opérateur Op sur les fonctions f_1, \dots, f_m sur les entiers peut être étendu à un opérateur sur les fonctionnelles en fixant le premier argument $\overline{Op}(F_1, \dots, F_m) : (V, \bar{n}) \mapsto Op(f_1(V, \cdot), \dots, f_m(V, \cdot))(\bar{n})$.

Dans cet esprit, étant donné un ensemble \mathcal{F} de fonctions de base $\mathbb{N}^k \rightarrow \mathbb{N}^l$, et un ensemble \mathcal{O} d'opérateurs sur les fonctions sur les entiers, nous dénoterons toujours, de façon abusive, par $[f_1, \dots, f_p; O_1, \dots, O_q]$, la plus petite classe de fonctionnelles qui contient les fonctions de base $\bar{f}_1, \dots, \bar{f}_p$, plus la fonctionnelle $Map : (V, n) \rightarrow V_n$, qui donne le n ème élément d'une suite V , et qui est close par les opérateurs $\bar{O}_1, \dots, \bar{O}_q$. Par exemple, une fonctionnelle sera dite élémentaire si et seulement si elle appartient à $\mathcal{E} = [Map, \bar{0}, S, \bar{U}, \bar{+}, \bar{\ominus}; \text{COMP}, \text{BSUM}, \text{BPROD}]$.

4.3.5 Résultats de Campagnolo, Costa, Moore

En suivant les idées de [Moore, 1996], mais en observant que le schéma de minimisation de cet article est problématique, Campagnolo, Costa et Moore ont proposé dans la série [Campagnolo et al., 2000], [Campagnolo et al., 2002], [Campagnolo, 2001] de ne pas considérer des classes de fonctions sur les réels définies en analogie avec les fonctions récursives, mais seulement en analogie avec des sous-classes. Les fonctions considérées par Campagnolo et al. sont ainsi construites en analogie avec les fonctions élémentaires ou les classes de la hiérarchie de Grzegorzcyk.

En outre, ils proposent de restreindre le schéma d'intégration général de [Moore, 1996] en un schéma d'intégration *linéaire*.

Nous appelons *extension réelle d'une fonction* $f : \mathbb{N}^k \rightarrow \mathbb{N}^l$ une fonction \tilde{f} de \mathbb{R}^k vers \mathbb{R}^l dont la restriction à \mathbb{N}^k est f .

Définition 4 ([Campagnolo, 2001], [Campagnolo et al., 2002]) *Soient \mathcal{L} et \mathcal{L}_n les classes de fonctions $f : \mathbb{R}^k \rightarrow \mathbb{R}^l$, pour $k, l \in \mathbb{N}$, définies par*

$$\mathcal{L} = [0, 1, -1, \pi, U, \theta_3; \text{COMP}, \text{LI}]$$

et¹²

$$\mathcal{L}_n = [0, 1, -1, \pi, U, \theta_3, \overline{E}_{n-1}; \text{COMP}, \text{LI}]$$

où les fonctions de base $0, 1, -1, \pi, (U_i^m)_{i,m \in \mathbb{N}}, \theta_3, \overline{E}_n$ et les schémas COMP and LI sont définis comme suit :

1. $0, 1, -1, \pi$ sont les fonctions constantes correspondantes ;
2. $U_i^m : \mathbb{R}^m \rightarrow \mathbb{R}$ sont, comme dans le cas classique, les projections, $U_i^m : (x_1, \dots, x_m) \mapsto x_i$;
3. $\theta_3 : \mathbb{R} \rightarrow \mathbb{R}$ est défini comme $\theta_3 : x \mapsto x^3$ si $x \geq 0, 0$ sinon.
4. \overline{E}_n : pour $n \geq 3$, \overline{E}_n est une extension réelle monotone fixée et quelconque de la fonction \exp_n sur les entiers définie par induction par $\exp_2(x) = 2^x$, $\exp_{i+1}(x) = \exp_i^{[x]}(1)$.
5. COMP : composition. Elle est définie comme dans le cas classique : étant données f et $g, h = \text{COMP}(f, g)$ est la fonction qui satisfait $h(\mathbf{x}) = g(f(\mathbf{x}))$;
6. LI : intégration linéaire. A partir des fonctions g et h , $\text{LI}(g, h)$ est la solution maximale de l'équation différentielle linéaire

$$\frac{\partial f}{\partial y}(\mathbf{x}, y) = h(\mathbf{x}, y)f(\mathbf{x}, y)$$

avec $f(\mathbf{x}, 0) = g(\mathbf{x})$.

Dans ce schéma, si g va vers \mathbb{R}^n , $f = \text{LI}(g, h)$ aussi et $h(\mathbf{x}, y)$ est une matrice $n \times n$ dont les éléments sont dans \mathcal{L} .

Les classes \mathcal{L} et \mathcal{L}_n contiennent les fonctions usuelles comme $+$, \sin , \cos , $-$, \times , \exp , et les fonctions constantes r pour tout $r \in \mathbb{Q}$, mais seulement des fonctions totales, et de classe \mathcal{C}^2 [Campagnolo, 2001], [Campagnolo et al., 2002].

Une contribution majeure de [Campagnolo et al., 2002], [Campagnolo, 2001] est de relier ces classes de fonctions sur les réels aux classes de fonctions précédentes sur les entiers. Pour comparer des classes de fonctions sur les réels à des classes de fonctions sur les entiers, nous introduisons la notation suivante : étant donnée une classe \mathcal{C} de fonctions de \mathbb{R}^k vers \mathbb{R}^l , nous écrivons $\text{DP}(\mathcal{C})$ (DP pour partie discrète) pour la classe des fonctions de \mathbb{N}^k vers \mathbb{N}^l qui possèdent une extension réelle dans \mathcal{C} .

Proposition 6 ([Campagnolo et al., 2002], [Campagnolo, 2001]) *On a*

- $\text{DP}(\mathcal{L}) = \mathcal{E}$;
- $\text{DP}(\mathcal{L}_n) = \mathcal{E}_n$.

¹²Les classes \mathcal{L}_n ne sont pas définies ainsi dans [Campagnolo, 2001], [Campagnolo et al., 2002], mais les résultats dans ces articles montrent l'équivalence de leur définition avec celle-ci.

En fait, ces articles prouvent des inclusions plus fortes.

Proposition 7 ([Campagnolo et al., 2002], [Campagnolo, 2001]) *On a*

- $\mathcal{L} \subset \mathcal{E}(\mathbb{R})$.
- $\mathcal{L}_n \subset \mathcal{E}_n(\mathbb{R})$.

Cependant, il n'y a pas d'espoir d'avoir les inclusions inverses, puisque $\mathcal{E}(\mathbb{R})$ et $\mathcal{E}_n(\mathbb{R})$ contiennent des fonctions partielles, alors que les classes \mathcal{L} et \mathcal{L}_n sont des classes de fonctions totales.

4.3.6 Caractérisation des fonctions élémentairement calculables

Nous avons proposé de considérer de nouvelles classes de fonctions qui s'avèrent correspondre précisément aux classes $\mathcal{E}(\mathbb{R})$ et $\mathcal{E}_n(\mathbb{R})$.

Pour cela, nous nous restreignons aux fonctions définies sur un domaine compact. Une motivation est que les fonctions élémentairement calculables sur un domaine quelconque ne sont pas stables par composition.

Pour des raisons techniques, nous avons besoin (en fait plutôt pour la section suivante) de remplacer le schéma LI précédent, par le schéma CLI suivant.

Définition 5 (Schéma CLI) *A partir des fonctions g , h , et c , avec*

- *la norme de chacune des dérivées partielles de h bornée par c , sauf éventuellement sa dérivée partielle en t ,*

CLI(g, h, c) est solution¹³ de l'équation différentielle linéaire

$$\frac{\partial f}{\partial y}(\mathbf{x}, y) = h(\mathbf{x}, y)f(\mathbf{x}, y)$$

avec $f(\mathbf{x}, 0) = g(\mathbf{x})$.

Dans ce schéma, si g va vers \mathbb{R}^n , $f = \text{CLI}(g, h, c)$ va vers \mathbb{R}^{n+1} et $h(\mathbf{x}, y)$ est une matrice $(n+1) \times (n+1)$ dont les éléments sont dans \mathcal{L} .

On peut montrer que remplacer le schéma LI par le schéma CLI ne change rien à la discussion précédente.

Proposition 8

$$\begin{aligned} \mathcal{L} &= [0, 1, -1, \pi, U, \theta_3; \text{COMP}, \text{LI}] \\ &= [0, 1, -1, \pi, U, \theta_3; \text{COMP}, \text{CLI}] \end{aligned}$$

et

$$\begin{aligned} \mathcal{L}_n &= [0, 1, -1, \pi, U, \theta_3, \bar{E}_{n-1}; \text{COMP}, \text{LI}] \\ &= [0, 1, -1, \pi, U, \theta_3, \bar{E}_{n-1}; \text{COMP}, \text{CLI}] \end{aligned}$$

D'autre part, nous introduisons un schéma limite. L'idée de considérer un schéma limite a déjà été envisagée dans des articles comme [Mycka and Costa, 2004], [Mycka, 2003b]. Cependant, puisque nous sommes intéressés par les fonctions \mathbb{R} -sous-récurrentes, et non pas à construire une hiérarchie au-dessus des fonctions \mathbb{R} -récurrentes, notre schéma limite est plus restreint que celui de ces articles.

Les conditions que nous imposons sur notre schéma LIM_w sont inspirées par le lemme 2.

Définition 6 (Schéma LIM) *Soient $f : \mathbb{R} \times \mathcal{D} \subset \mathbb{R}^{k+1} \rightarrow \mathbb{R}^l$ et $K : \mathcal{D} \rightarrow \mathbb{R}$ deux fonctions, et $\beta : \mathcal{D} \rightarrow \mathbb{R}$ une fonction polynôme, avec l'hypothèse suivante : pour tout \mathbf{x} , $t \geq \|\mathbf{x}\|$,*

$$\left\| \frac{\partial f}{\partial t}(t, \mathbf{x}) \right\| \leq K(\mathbf{x}) \exp(-t\beta(\mathbf{x})).$$

Alors, sur tout produit d'intervalles fermés $I \subset \mathbb{R}^k$ sur lequel $\beta(\mathbf{x}) > 0$,

$$F(\mathbf{x}) = \lim_{t \rightarrow +\infty} f(t, \mathbf{x})$$

existe par le lemme 2.

Si F est de classe \mathcal{C}^2 , alors nous définissons $\text{LIM}_w(f, K, \beta)$ comme cette fonction $F : I \rightarrow \mathbb{R}$.

¹³En fait n'importe quelle restriction à un produit d'intervalles fermés de la solution maximale

Nous pouvons définir nos classes.

Définition 7 (Classes \mathcal{L}^* , \mathcal{L}_n^*) Les classes \mathcal{L}^* , et \mathcal{L}_n^* , pour $n \geq 3$, de fonctions de \mathbb{R}^k vers \mathbb{R}^l , pour $k, l \in \mathbb{N}$, sont définies comme les classes suivantes.

$$\mathcal{L}^* = [0, 1, -1, U, \theta_3; \text{COMP}, \text{CLI}, \text{LIM}],$$

et

$$\mathcal{L}_n^* = [0, 1, -1, U, \theta_3, \overline{E}_{n-1}; \text{COMP}, \text{CLI}, \text{LIM}].$$

Ces classes peuvent maintenant contenir des fonctions partielles, et étendent les classes précédentes.

Proposition 9

$$\mathcal{L} \subsetneq \mathcal{L}^*,$$

et

$$\mathcal{L}_n \subsetneq \mathcal{L}_n^*$$

pour tout $n \geq 3$.

D'autre part, un de nos résultats majeurs, prouvé dans nos publications [Bournez and Hainry, 2005] et [Bournez and Hainry, 2004a], est que ces classes caractérisent les fonctions élémentairement calculables en analyse récursive. Les théorèmes 2 et 3 précédents sont formellement.

Théorème 2 (Caractérisation de $\mathcal{E}(\mathbb{R})$) Soit $f : \mathcal{D} \subset \mathbb{R}^k \rightarrow \mathbb{R}^l$ une fonction sur les réels de classe \mathcal{C}^2 , avec \mathcal{D} un produit d'intervalles compacts à extrémités rationnelles.

$$f \text{ est dans } \mathcal{E}(\mathbb{R}) \text{ si et seulement si } f \in \mathcal{L}^*.$$

En fait cela se généralise à tous les niveaux de la hiérarchie de Grzegorzcyk (observons que le théorème 2 est le cas particulier $n = 3$ du théorème 3).

Théorème 3 (Caractérisation de $\mathcal{E}_n(\mathbb{R})$) Soit $f : \mathcal{D} \subset \mathbb{R}^k \rightarrow \mathbb{R}^l$ une fonction sur les réels de classe \mathcal{C}^2 , avec \mathcal{D} un produit d'intervalles compacts à extrémités rationnelles. Soit $n \geq 3$.

$$f \text{ est dans } \mathcal{E}_n(\mathbb{R}) \text{ si et seulement si } f \in \mathcal{L}_n^*.$$

Quelques extensions de ces résultats seront trouvées dans ces articles, en particulier avec un théorème de forme normale pour les fonctions de nos classes, et quelques schémas alternatifs à notre schéma limite, comme des schémas de recherche de minimum de fonctions convexes.

4.3.7 Caractérisation des fonctions calculables

Nous avons d'autre part ultérieurement montré qu'il était possible de caractériser les fonctions calculables, et pas seulement les fonctions élémentairement calculables.

Pour cela, nous devons introduire un opérateur de minimisation qui permet de simuler les minimisations discrètes sur les entiers.

Cependant, cet opérateur doit être plus restreint que la simple idée de retourner "le plus petit zéro", puisque cette idée, source des investigations dans [Moore, 1996], a montré être la cause de nombreux problèmes, discutés dans [Campagnolo, 2001], [Campagnolo et al., 2002], [Mycka, 2003b], [Mycka, 2003a]. Les articles [Mycka, 2003b], [Mycka and Costa, 2004] présentent une alternative bien fondée, en remplaçant la minimisation par une opération de passage à la limite. Nous proposons ici de nous en tenir à un schéma de recherche de zéro, mais restreint par rapport à celui de [Moore, 1996].

Notre idée est d'utiliser le schéma alternatif UMU schéma qui est équivalent au schéma MU en calculabilité classique (voir la proposition 5), mais qui a une contrepartie qui s'avère préserver les fonctions calculables en analyse récursive.

En effet, motivés par la proposition 5, par le lemme 1 (le théorème des fonctions implicites), et par les résultats de l'analyse récursive sur la calculabilité des zéros d'une fonction (voir par exemple [Weihrach, 2000] où les théorèmes 6.3.5 et 6.3.8 disent que la recherche d'un zéro unique est calculable), nous définissons notre opérateur de recherche de zéro UMU comme suit.

Définition 8 (Schéma UMU) *Étant donnée une fonction différentiable f de $\mathcal{D} \times \mathcal{I} \subset \mathbb{R}^{k+1}$ vers \mathbb{R} où $\mathcal{D} \times \mathcal{I}$ est un produit d'intervalles fermés,*

si pour tout $\mathbf{x} \in \mathcal{D}$,

- $y \mapsto f(\mathbf{x}, y)$ est une fonction non-décroissante,
- avec une unique racine y_0 sur \mathcal{I} ,
- et telle que y_0 soit à l'intérieur de l'intervalle \mathcal{I} ,
- avec

$$\frac{\partial f}{\partial y}(\mathbf{x}, y_0) > 0,$$

alors $\text{UMU}(f)$ est défini sur \mathcal{D} comme suit :

$$\text{UMU}(f) : \begin{cases} \mathcal{D} & \longrightarrow \mathbb{R} \\ \mathbf{x} & \longmapsto y_0 \text{ tel que } f(\mathbf{x}, y_0) = 0. \end{cases}$$

On définit alors.

Définition 9 (Classe $\mathcal{L}+!\mu$) *Soit $\mathcal{L}+!\mu$ l'ensemble de fonctions sur les réels défini par*

$$\mathcal{L}+!\mu = [0, 1, U, \theta_3; \text{COMP}, \text{CLI}, \text{UMU}].$$

On peut montrer que

$$\mathcal{L} \subset \mathcal{L}+!\mu,$$

et que $\mathcal{L}+!\mu$ ne contient que des fonctions de classe \mathcal{C}^2 , définies sur des produits d'intervalles fermés.

Notre résultat principal dans [Bournez and Hainry, 2004b], constitue la formalisation du théorème 4 informel précédent en les théorèmes 4 et 5 qui suivent.

Théorème 4 *Pour les fonctions totales*

$$\mathcal{R}ec = DP(\mathcal{L}+!\mu).$$

I.e.

- *Si une fonction totale de $\mathcal{L}+!\mu$ étend une fonction sur les entiers, alors cette fonction est totale récursive.*
- *Toute fonction totale récursive sur les entiers, possède une extension réelle dans $\mathcal{L}+!\mu$.*

En ajoutant un opérateur limite, et en combinant avec les constructions de [Bournez and Hainry, 2004a], [Bournez and Hainry, 2005], nous avons obtenu notre résultat majeur à propos des fonctions \mathbb{R} -récursives.

Définition 10 (Classe $\mathcal{L}+!\mu + \text{LIM}_w$)

$$\mathcal{L}^*_{!\mu} = [0, 1, U, \theta_3; \text{COMP}, \text{CLI}, \text{UMU}, \text{LIM}_w]$$

Nous avons le théorème suivant, qui constitue notre résultat majeur dans [Bournez and Hainry, 2005].

Théorème 5 *Pour les fonctions de classe \mathcal{C}^2 définies sur un produit d'intervalles compacts à extrémités rationnelles,*

$$\mathcal{L}^*_{!\mu} = \mathcal{R}ec(\mathbb{R}).$$

Nous avons proposé quelques extensions de ce résultat dans cet article.

En d'autres termes, nous savons relier les fonctions calculables en analyse récursive aux fonctions \mathbb{R} -récursives.

4.4 Discussions

En ce qui concerne les *modèles analogiques*, tous ces résultats ont plusieurs impacts. Premièrement, ils contribuent à comprendre les modèles analogiques, en particulier les relations entre les fonctions GPAC calculables, les fonctions \mathbb{R} -récurives, et les fonctions calculables en analyse réursive. En outre, ils prouvent qu'aucun phénomène d'hypercalcul ne peut se produire pour ces classes de fonctions. En particulier, on a affaire à des classes de fonctions qui sont robustes dans un certain sens. Voir la discussion du chapitre 3, à propos des notions de robustesse.

En ce qui concerne *l'analyse réursive*, nos résultats fournissent une caractérisation *purement algébrique* des fonctions élémentairement calculables sur les réels. Observons les bénéfices potentiels offerts par ces caractérisations par rapport aux définitions classiques en analyse réursive, qui requièrent des discussions à propos de machines de Turing d'ordre supérieur (ou de type 2) (voir par exemple [Weihrauch, 2000]), ou par rapport aux caractérisations dans l'esprit de [Brattka, 2003], [Kawamura, 2005].

D'autre part, nous observerons que nous montrons que les fonctions sur les réels calculables peuvent être définies par des schémas *continus*, c'est-à-dire sur des fonctions continues. Il est nettement plus naturel d'utiliser des schémas continus pour définir des fonctions continues, que d'utiliser des schémas limites sur des fonctions discrètes [Brattka, 2003], [Kawamura, 2005].

4.5 Une théorie de la complexité ?

Plus généralement, dans ce chapitre, nous avons relié plusieurs modèles, à priori distincts, de calculs à temps continu.

Tout d'abord, nous avons rappelé que les fonctions générées par le GPAC correspondent aux problèmes de Cauchy polynomiaux. Ensuite, nous avons montré que si l'on prend une notion de calcul par calculs convergents, alors le GPAC et les problèmes de Cauchy polynomiaux ont exactement la puissance de l'analyse réursive : les fonctions calculables sont les mêmes. D'autre part, nous avons aussi relié cette puissance à celle de classes de fonctions \mathbb{R} -récurives. Nous avons ainsi obtenu la première caractérisation algébrique, par l'analyse, des fonctions calculables en analyse réursive.

L'enjeu pour comprendre une éventuelle thèse de Church-Turing pour les modèles analogiques serait d'arriver à relier ces modèles aux autres modèles évoqués dans les autres chapitres de ce document.

Mais nous croyons qu'une question très importante est d'arriver à comprendre si on peut passer de la calculabilité à la complexité. Peut-on définir une notion de complexité, valide, et universellement acceptée pour les modèles analogiques ?

Nous avons présenté les résultats existants à ce jour dans le chapitre 3, et déjà discuté de nombreux problèmes ouverts. Mais les résultats de ce chapitre, laissent entrevoir des pistes.

Par exemple, l'ouvrage [Ko, 1991] présente une définition robuste de ce que sont les fonctions calculables en temps polynomial en analyse réursive. Pourrait-on caractériser les classes de complexité de l'analyse réursive algébriquement avec des schémas comme ceux évoqués ici ?

Est-il possible de relier les fonctions calculables en temps polynomial à une classe de fonctions GPAC calculables où l'erreur ε serait donnée par une fonction polynomiale en t ?

Et si de tels résultats étaient possibles, est-ce qu'ils pourraient se généraliser à d'autres classes de complexité ?

Des pistes pour cela sont données par les caractérisations algébriques en complexité classique des classes de complexité. Par exemple, la caractérisation de Bellantoni et Cook du temps polynomial en complexité classique, en termes de fonctions récurives sûres dans [Bellantoni and Cook, 1992]. Peut-on adapter ces caractérisations pour définir des classes de fonctions \mathbb{R} -récurives similaires au temps polynomial ? à l'espace polynomial ?

Le chapitre suivant est précisément relatif aux caractérisations dans l'esprit de Bellantoni et Cook, mais pour une autre classes de modèles de calculs : le modèle de Blum Shub et Smale, qui lui est à espace continu mais à temps discret.

Bibliographie

- [Arnold, 1992] Arnold, V. I. (1992). *Ordinary differential equations*. Springer-Verlag, Berlin.
- [Bellantoni and Cook, 1992] Bellantoni, S. and Cook, S. (1992). A new recursion-theoretic characterization of the poly-time functions. *Computational Complexity*, 2 :97–110.
- [Blum et al., 1989] Blum, L., Shub, M., and Smale, S. (1989). On a theory of computation and complexity over the real numbers ; NP completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1) :1–46.
- [Bournez et al., 2006] Bournez, O., Campagnolo, M. L., Graça, D. S., and Hainry, E. (2006). The general purpose analog computer and computable analysis are two equivalent paradigms of analog computation. In Cai, J., Cooper, S. B., and Li, A., editors, *Theory and Applications of Models of Computation, Third International Conference, TAMC 2006, Beijing, China, May 15-20, 2006, Proceedings*, volume 3959 of *Lecture Notes in Computer Science*, pages 631–643. Springer.
- [Bournez and Hainry, 2004a] Bournez, O. and Hainry, E. (2004a). An analog characterization of elementarily computable functions over the real numbers. In Diaz, J., Karhumäki, J., Lepisto, A., and Sannella, D. T., editors, *31th International Colloquium on Automata Languages and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 269–280, Turku, Finland. Springer.
- [Bournez and Hainry, 2004b] Bournez, O. and Hainry, E. (2004b). Real recursive functions and real extensions of recursive functions. In Margenstern, M., editor, *Machines, Computations and Universality (MCU'2004)*, volume 3354 of *Lecture Notes in Computer Science*, Saint-Petersburg, Russia.
- [Bournez and Hainry, 2005] Bournez, O. and Hainry, E. (2005). Elementarily computable functions over the real numbers and \mathbb{R} -sub-recursive functions. *Theoretical Computer Science*, 348(2–3) :130–147.
- [Bournez and Hainry, 2006] Bournez, O. and Hainry, E. (2006). Recursive analysis characterized as a class of real recursive functions. *Fundamenta Informaticae*, 74(4) :409–433.
- [Brattka, 2003] Brattka, V. (2003). Computability over topological structures. In Cooper, S. B. and Goncharov, S. S., editors, *Computability and Models*, pages 93–136. Kluwer Academic Publishers, New York.
- [Bush, 1931] Bush, V. (1931). The differential analyser. *Journal of the Franklin Institute*, 212(4) :447–488.
- [Campagnolo et al., 2000] Campagnolo, M., Moore, C., and Costa, J. F. (2000). An analog characterization of the subrecursive functions. In Kornerup, P., editor, *Proc. 4th Conference on Real Numbers and Computers*, pages 91–109. Odense University Press.
- [Campagnolo et al., 2002] Campagnolo, M., Moore, C., and Costa, J. F. (2002). An analog characterization of the Grzegorzcyk hierarchy. *Journal of Complexity*, 18(4) :977–1000.
- [Campagnolo, 2001] Campagnolo, M. L. (2001). *Computational complexity of real valued recursive functions and analog circuits*. PhD thesis, IST, Universidade Técnica de Lisboa.
- [Clote, 1998] Clote, P. (1998). Computational models and function algebras. In Griffor, E. R., editor, *Handbook of Computability Theory*, pages 589–681. North-Holland, Amsterdam.
- [Cutland, 1980] Cutland, N. J. (1980). *Computability : An introduction to recursive function theory*. Cambridge Univ. Press, Cambridge.

- [Graça et al., 2005] Graça, D., Campagnolo, M., and Buescu, J. (2005). Robust simulations of Turing machines with analytic maps and flows. In Cooper, B., Loewe, B., and Torenvliet, L., editors, *Proceedings of CiE'05, New Computational Paradigms*, volume 3526 of *Lecture Notes in Computer Science*, pages 169–179. Springer-Verlag.
- [Graça, 2004] Graça, D. S. (2004). Some recent developments on Shannon’s general purpose analog computer. *Mathematical Logic Quarterly*, 50(4–5) :473–485.
- [Graça and Costa, 2003] Graça, D. S. and Costa, J. F. (2003). Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5) :644–664.
- [Hölder, 1887] Hölder, O. (1887). Über die eigenschaft der gamma funktion keiner algebraische differentialgleichung zu genügen. *Math. Ann.*, 28 :1–13.
- [Kawamura, 2005] Kawamura, A. (2005). Type-2 computability and Moore’s recursive functions. In Brattka, V., Staiger, L., and Weihrauch, K., editors, *Proceedings of the 6th Workshop on Computability and Complexity in Analysis*, volume 120 of *Electronic Notes in Theoretical Computer Science*, pages 83–95, Amsterdam. Elsevier. 6th International Workshop, CCA 2004, Wittenberg, Germany, August 16–20, 2004.
- [Ko, 1991] Ko, K.-I. (1991). *Complexity Theory of Real Functions*. Progress in Theoretical Computer Science. Birkhäuser, Boston.
- [Lipshitz and Rubel, 1987] Lipshitz, L. and Rubel, L. A. (1987). A differentially algebraic replacement theorem, and analog computability. *Proceedings of the American Mathematical Society*, 99(2) :367–372.
- [Moore, 1996] Moore, C. (1996). Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162(1) :23–44.
- [Moore, 1998] Moore, C. (1998). Dynamical recognizers : real-time language recognition by analog computers. *Theoretical Computer Science*, 201(1–2) :99–136.
- [Mycka, 2003a] Mycka, J. (2003a). Infinite limits and r-recursive functions. *Acta Cybernetica*, 16 :83–91.
- [Mycka, 2003b] Mycka, J. (2003b). μ -recursion and infinite limits. *Theoretical Computer Science*, 302 :123–133.
- [Mycka and Costa, 2004] Mycka, J. and Costa, J. F. (2004). Real recursive functions and their hierarchy. *Journal of Complexity*, 20(6) :835–857.
- [Mycka and Costa, 2005] Mycka, J. and Costa, J. F. (2005). What lies beyond the mountains? computational systems beyond the Turing limit. *European Association for Theoretical Computer Science Bulletin*, 85 :181–189.
- [Mycka and Costa, 2006a] Mycka, J. and Costa, J. F. (2006a). Analog computation and beyond. Submitted.
- [Mycka and Costa, 2006b] Mycka, J. and Costa, J. F. (2006b). The $P \neq NP$ conjecture. Submitted.
- [Odifreddi, 1992] Odifreddi, P. (1992). *Classical Recursion Theory*, volume 125 of *Studies in Logic and the foundations of mathematics*. North-Holland.
- [Pour-El, 1974] Pour-El, M. B. (1974). Abstract computability and its relation to the general purpose analog computer (some connections between logic, differential equations and analog computers). *Transactions of the American Mathematical Society*, 199 :1–28.
- [Pour-El and Richards, 1989] Pour-El, M. B. and Richards, J. I. (1989). *Computability in Analysis and Physics*. Springer-Verlag.
- [Ramis et al., 1995] Ramis, E., Deschamps, C., and Odoux, J. (1995). *Cours de Mathématiques Spéciales, Tome 3, Topologie et éléments d’analyse*. Masson.
- [Rose, 1984] Rose, H. (1984). *Subrecursion*. Oxford university press.
- [Shannon, 1941] Shannon, C. E. (1941). Mathematical theory of the differential analyser. *Journal of Mathematics and Physics MIT*, 20 :337–354.
- [Siegelmann, 1995] Siegelmann, H. T. (1995). Computation beyond the Turing limit. *Science*, 268 :545–548.
- [Siegelmann, 1999] Siegelmann, H. T. (1999). *Neural Networks and Analog Computation - Beyond the Turing Limit*. Birkhäuser.

- [Siegelmann and Sontag, 1994] Siegelmann, H. T. and Sontag, E. D. (1994). Analog computation via neural networks. *Theoretical Computer Science*, 131(2) :331–360.
- [Weihrauch, 2000] Weihrauch, K. (2000). *Computable Analysis*. Springer.
- [Zhou, 1997] Zhou, Q. (1997). Subclasses of computable real valued functions. *Lecture Notes in Computer Science*, 1276 :156–165.