INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Asymptotique automatique

Bruno Salvy

N ° 3707
Juin 1999

THÈME 2

Rapport de recherche

ISSN 0249-6399

Asymptotique automatique

Bruno Salvy

Thème 2 — Génie logiciel
et calcul symbolique
Projet Algo

Rapport de recherche n ° 3707 — Juin 1999 — 22 pages

Résumé : Ce rapport est une version préliminaire d'un chapitre à paraître dans un ouvrage collectif sur le calcul formel.

Mots-clé : calcul formel, développements asymptotiques

(Abstract on next page)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33) 01 39 63 55 11 – Télécopie : (33) 01 39 63 53 30

Symbolic Asymptotics

Abstract: This report is a preliminary version of a chapter to appear in a collective book on computer algebra.

Key-words: symbolic computation, asymptotic expansions

ASYMPTOTIQUE AUTOMATIQUE

BRUNO SALVY

... [les] connoissances de l'Infini dénuées de Calcul
n'ont produit que des Méthodes particulieres,
souvent embarassées et toujours confinées
à quelques cas assez simples,
la généralité étoit réservée au Calcul,
il embrasse tout, il donne tout...
Buffon (1740).

Le calcul formel consiste à manipuler des représentations finies d'objets mathématiques exacts. Ses méthodes sont originaires de l'algèbre qui fournit de telles représentations par des bases ou des systèmes de générateurs finis dans des contextes variés. La majorité des travaux en calcul formel a donc naturellement pour but de résoudre des problèmes de nature algébrique : manipulation de polynômes ou de leurs solutions en algèbre commutative, calcul de primitives ou recherche de solutions liouvilliennes d'équations différentielles linéaires en algèbre différentielle.

Cependant, peu de problèmes d'origine physique possèdent une solution représentable sous forme close. Les applications du calcul formel doivent donc largement provenir de sa capacité à produire et à manipuler des approximations *de précision arbitraire*. Pendant longtemps, cette utilisation du calcul formel pour l'analyse n'a pas bénéficié d'une algorithmisation aussi poussée ou aussi rigoureuse que les applications algébriques.

L'*asymptotique automatique* est un domaine récent du calcul formel qui fournit des approximations par des séries convergentes ou des développements asymptotiques. L'objectif est la mise au point d'outils efficaces permettant de produire puis de manipuler des approximations précises et donc souvent de très grande taille. Cet objectif nécessite le développement d'algorithmes exhaustifs déterminant les échelles asymptotiques les plus générales et les développements correspondants pour les solutions de grandes classes de problèmes.

L'asymptotique automatique a énormément progressé durant les dernières années. D'abord, des techniques développées au début des années 80 montrent comment calculer des développements en séries entières à de grands ordres pour les fonctions élémentaires ou pour les solutions d'équations différentielles non linéaires, avec une complexité du même ordre que celle du produit de séries, ce que nous décrivons en section 1. L'usage systématique des équations différentielles linéaires initié par Zeilberger à la fin des années 80 a conduit à une algorithmisation de la manipulation de la plupart des fonctions spéciales de physique mathématique. Il en découle en particulier un calcul très efficace des développements en séries ou asymptotiques de ces fonctions. Ces idées sont introduites en section 2. Enfin dans les années 90, des méthodes partielles et *ad hoc* de calculs de limites ont laissé la place à une nouvelle génération d'algorithmes qui garantissent la détermination des développements asymptotiques pour toutes les fonctions exp-log, leurs inverses pour la composition, les fonctions implicites et dans une certaine mesure les solutions d'équations différentielles algébriques. Ceci fait l'objet de la section 3.

1. Efficacité des calculs de séries

Les calculs de développements asymptotiques se ramènent en dernier ressort à des manipulations de séries tronquées en une variable de la forme

$$\sum_{k=0}^{n} a_k x^k + O(x^{n+1}). \tag{1}$$

Les systèmes de calcul formel emploient pour représenter les polynômes et les séries des structures de données similaires. Cependant, des algorithmes spécifiques tirent parti de la troncature pour obtenir efficacement un grand nombre de termes des développements [9, 1]. Cette efficacité est cruciale pour les applications en calcul des probabilités, en analyse combinatoire et en analyse numérique.

1.1. **Somme et produit.** L'addition de deux séries tronquées à l'ordre n de la forme (1) par la méthode naïve requiert $O(n)$ opérations sur les coefficients (nommées également opérations *arithmétiques*). Cette complexité est optimale : elle correspond au nombre de coefficients à calculer.

La méthode naïve pour le produit de séries tronquées à l'ordre n ou de polynômes de degré n nécessite $O(n^2)$ opérations arithmétiques. Un algorithme dû à Karatsuba permet le calcul du produit au prix de seulement $O(n^{\log_2 3})$ opérations ($\log_2 3 \simeq 1,58$). L'algorithme repose sur une méthode du type diviser-pour-régner. Il consiste à scinder chacun des deux polynômes en deux polynômes de degré $k \simeq n/2$:

$$P(x) = P_0(x) + x^k P_1(x), \qquad Q(x) = Q_0(x) + x^k Q_1(x).$$

Le produit s'obtient au prix de seulement trois produits de polynômes de degré k grâce aux identités

$$P(x)Q(x) = P_0(x)Q_0(x) + x^k R(x) + x^{2k} P_1(x)Q_1(x),$$
$$R = (P_0 + P_1)(Q_0 + Q_1) - P_0Q_0 - P_1Q_1.$$

L'application récursive de ce procédé mène à la complexité annoncée. Les additions supplémentaires requises par cette méthode la handicapent lorsque le degré est faible. Il vaut mieux alors recourir à la méthode naïve. Le seuil entre les deux méthodes dépend beaucoup de l'implantation des diverses opérations. La méthode s'étend en découpant en trois, puis quatre, ... avec des complexités qui s'amoindrissent, mais des seuils de plus en plus élevés. À la limite, il est possible d'atteindre une complexité en $O(n \log n)$ pour la multiplication en employant la transformée de Fourier rapide, mais ce résultat n'est applicable que pour des ordres extrêmement élevés. L'algorithme de Karatsuba ci-dessus donne satisfaction en pratique. Dans la suite, nous emploierons la notation $M(n)$ pour désigner le nombre d'opérations sur les coefficients effectuées lors du calcul du produit de deux séries à l'ordre n. On a donc $M(n) = O(n^2)$, $O(n^{1,58})$, $O(n^{1+\epsilon})$ ou $O(n \log n)$ selon la méthode choisie.

1.2. **Composition.** En général, la composition $F(G(x))$ de deux séries

(2)
$$F(x) = f_0 + \cdots + f_n x^n + O(x^{n+1}) \quad \text{et} \quad G(x) = g_1 x + \cdots + g_n x^n + O(x^{n+1})$$

est une opération plus coûteuse que le produit. Dans la méthode naïve, le nombre d'opérations arithmétiques requises est $O(nM(n))$. Une technique dite *pas de bébé, pas de géant* réduit ce coût à $O(\sqrt{n}M(n))$. La série F est d'abord récrite

$$F(x) = F_0(x) + x^k(F_1(x) + \cdots + x^k F_{\lceil n/k \rceil}(x)) + O(x^{n+1}),$$

où les séries F_i en nombre $1 + \lceil n/k \rceil$ comportent chacune k termes. Ensuite, il suffit de calculer les puissances $G^2, \ldots, G^k$ de G pour réduire à des additions le calcul des $F_i(G)$, puis le calcul de $F(G)$ par le schéma de Horner ci-dessus. Le choix $k = \lfloor \sqrt{n} \rfloor$ mène à la complexité $O(\sqrt{n}M(n))$. Si la méthode de Karatsuba est utilisée pour les produits, le nombre d'opérations sur les coefficients est donc réduit à $O(n^{2,08})$ au lieu de $O(n^3)$ pour la méthode naïve. Le gain apparaît déjà sur des ordres de quelques dizaines.

1.3. **Méthode de Newton.** Une information supplémentaire sur la série F permet de réduire significativement le coût du calcul de $F(G(x))$. Étant donnée la série G, le calcul des n premiers coefficients du développement de son exponentielle, son logarithme, son inverse et plus généralement le développement des solutions de nombreuses équations différentielles même non linéaires a une complexité en $O(M(n))$.

L'efficacité de ces calculs repose sur l'utilisation d'une variante symbolique de la méthode de Newton. Pour calculer une série solution d'une équation $\Phi(x, y(x)) = 0$, l'itération de Newton s'écrit

$$y_{k+1}(x) = y_k(x) - \frac{\Phi(x, y_k(x))}{\frac{\partial \Phi}{\partial y}(x, y_k(x))} \bmod x^{2^k+1},$$

où l'écriture mod x^{2^k+1} signifie que les termes d'ordre supérieur ne sont pas calculés. L'itération démarre avec $y_0(x) = 0$. Le nombre de coefficients corrects double à chaque itération. De nombreux résultats de complexité proviennent alors de l'identité $O(M(n)+M(n/2)+M(n/4)+\cdots)=O(M(n))$.

Inverse et division

Pour calculer l'inverse d'une série G de terme constant non nul, cette méthode est appliquée à $\Phi = G - 1/y$, ce qui conduit à l'itération

$$y_{k+1} = y_k(2 - Gy_k) \bmod x^{2^k+1}.$$

La complexité en $O(M(n))$ s'en déduit, ce qui montre que le coût de la division de séries est également de cet ordre.

Logarithme et exponentielle

Le calcul du logarithme découle de celui de l'inverse par le calcul terme à terme de l'intégrale de la série G'/G. Pour calculer l'exponentielle de la série G, la méthode de Newton est appliquée à $\Phi = \log y - G$, ce qui mène à l'itération

$$y_{k+1} = y_k(1 + G - \log y_k) \bmod x^{2^k+1},$$

dont la complexité d'évaluation est donc encore $O(M(n))$.

Équations différentielles linéaires

La solution de l'équation

$$y' - A(x)y = B(x), \tag{3}$$

où A et B sont des séries, est obtenue par variation de la constante. Notée symboliquement

$$y = e^{\int A}\int Be^{-\int A},$$

son calcul ne requiert que des calculs d'intégrales, de complexité $O(n)$, et d'exponentielles. Une version matricielle de ce calcul résout les équations différentielles linéaires de tout ordre à coefficients des séries en $O(M(n))$ opérations arithmétiques.

Équations différentielles non linéaires

La méthode de Newton s'étend aussi aux équations différentielles non linéaires du type $y' = F(x,y)$ où F est une série en x et y : en linéarisant l'opérateur, les calculs se réduisent à des équations du type (3).

Exemple 1. Pour calculer le développement en série de $y = \tan(G(x))$ où $G(x)$ est une série connue, l'opérateur $\Phi : y \mapsto y' - G'(1+y^2)$ est d'abord linéarisé par

$$\Phi(y+h) - \Phi(y) = h' - 2G'hy + O(h^2).$$

L'itération de Newton consiste à prendre $y_{k+1} = y_k + h_k$ de telle sorte que $\Phi(y_{k+1})$ s'annule modulo $O(h^2)$, ce qui revient à calculer

$$y_{k+1} = y_k + h_k \bmod x^{2^k+1}, \qquad h_k' - 2G'y_kh_k = -y_k' + G'(1+y_k^2).$$

L'équation de droite est du type (3). Cette itération fournit donc un calcul de n termes de y en $O(M(n))$ opérations sur les coefficients au lieu des $O(\sqrt{n}M(n))$ opérations requises pour le calcul de la composition des deux séries.

Inverses fonctionnels

Une autre application fructueuse de la méthode de Newton est l'inversion pour la composition : pour une série $G(x)$ de la forme (2) avec $g_1 \neq 0$, un tel inverse est une série F telle que $F(G(x)) = G(F(x)) = x$. La méthode de Newton appliquée à $\Phi = G(y) - x$ mène à l'itération

$$y_{k+1} = y_k - (G(y_k) - x)/G'(y_k) \bmod x^{2^k+1}.$$

Le coût du calcul est alors du même ordre de grandeur que celui de la composition par G, à savoir $O(\sqrt{n}M(n))$ si G est quelconque, et $O(M(n))$ pour de nombreuses fonctions par les méthodes qui viennent d'être décrites.

Exemple 2. L'équation $\tan x = x$ admet une solution réelle dans chacun des intervalles $I_k =](2k-1)\pi/2, (2k+1)\pi/2[$, pour $k \in \mathbb{Z}$. Le développement asymptotique de la racine appartenant à I_k lorsque k tend vers l'infini s'obtient par calcul d'inverse. La situation est d'abord ramenée en 0 par le changement de variables $x = (2k+1)\pi/2 - y$, qui conduit à l'équation

$$\frac{1}{\cot y + y} = \frac{1}{k\pi + \pi/2}.$$

Le membre gauche admet un développement en série qui se calcule en $O(M(n))$ opérations d'après l'exemple précédent. L'inversion donne ensuite la solution en puissances décroissantes de $k\pi + \pi/2$ en $O(M(n))$ opérations également.

Exemple 3. La méthode de Laplace permet de calculer le développement asymptotique d'intégrales de la forme

$$\int_a^b e^{-xp(t)} q(t)\, dt$$

lorsque $x \to \infty$. Pour cela, le changement de variable $p(t) = p(t_0) + u^2$ est effectué dans l'intégrale au voisinage d'un point t_0 où p est minimal. Ceci ramène l'intégrale à l'une des deux formes

$$\int_0^\infty e^{-xt} f(t)\, dt \quad \text{ou} \quad \int_{-\infty}^\infty e^{-xt^2} f(t)\, dt.$$

Le lemme de Watson (voir par exemple [21]) donne alors des conditions suffisantes sur f pour que le développement asymptotique s'obtienne en intégrant terme à terme le développement de Taylor de $f(t)$ à l'origine. Le calcul du changement de variable requiert une inversion de série qui peut être effectuée efficacement par la méthode décrite ci-dessus.

2. Le cas différentiel linéaire

Dans tous les exemples évoqués jusqu'ici, les coûts des calculs sont au mieux de l'ordre de $O(M(n))$ opérations arithmétiques. Les équations différentielles linéaires fournissent une classe importante de fonctions dont le développement en série se calcule plus rapidement. Si (f_n) désigne la suite des coefficients de la série $F(x)$, la correspondance définie par

$$F(x) \leftrightarrow f_n, \quad x^k F(x) \leftrightarrow f_{n-k}, \quad xF'(x) \leftrightarrow n f_n$$

met en bijection les équations différentielles linéaires à coefficients polynomiaux en x et les récurrences linéaires à coefficients polynomiaux en n. Une telle récurrence permet le calcul des n premiers éléments de la suite (f_n) en $O(n)$ opérations. Il s'ensuit également qu'il est possible d'évaluer ces fonctions numériquement en grande précision avec une faible complexité. De nombreuses constantes mathématiques comme π, $\ln 2$, γ, ... peuvent ainsi être approximées rapidement [2].

Exemple 4. Les nombres de Motzkin M_n comptent le nombre d'arbres unaires-binaires à n sommets. Ce sont les coefficients de la série entière satisfaisant

$$y = 1 + xy + x^2y^2.$$

De la solution $y = (1 - x - \sqrt{1 - 2x - 3x^2})/(2x^2)$, une double application de la formule du binôme de Newton fournit l'expression

$$M_n = -\frac{1}{2} \sum_{k=\lceil \frac{n+2}{2} \rceil}^{n+2} (-1)^k \binom{1/2}{k} \binom{k}{n+2-k} 2^{2k-n-2} 3^{n+2-k}.$$

La complexité élevée de l'évaluation de cette formule en fait un piètre moyen de calculer ces nombres. En revanche, il n'est pas difficile d'obtenir l'équation différentielle

$$x(x+1)(3x-1)y'(x) + (3x^2 + 3x - 2)y(x) + 2 = 0$$

dont découle la récurrence $(n+4)u_{n+2} = (2n+5)u_{n+1} + (3n+3)u_n$, et les conditions initiales $u_0 = u_1 = 1$, qui montrent comment calculer les n premiers nombres de Motzkin en $O(n)$ opérations sur des rationnels.

Plus généralement, il est possible de développer rapidement les fonctions algébriques, même lorsqu'aucune formule en termes de radicaux n'est disponible, grâce à une méthode décrite en §2.2.

Les solutions d'équations différentielles linéaires à coefficients polynomiaux sont omniprésentes (logarithme, exponentielle, fonctions algébriques, trigonométriques, trigonométriques hyperboliques, hypergéométriques, fonctions de Bessel, d'Airy, de Weber, de Struve, ...). Ainsi, pour toutes ces fonctions, le calcul du développement de Taylor en tout point où elle est analytique peut être effectué rapidement.

Parmi les propriétés remarquables de ces équations, nous ne détaillons ici que celles qui permettent d'exploiter cette efficacité particulière, ou tout au moins qui sont reliées aux développements en série. Dans toute cette section, les équations différentielles et les récurrences sont linéaires homogènes et à coefficients polynomiaux, même si cela n'est pas rappelé à chaque occasion.

2.1. **Propriétés de clôture.** Une équation différentielle linéaire homogène à coefficients polynomiaux impose la finitude de la dimension de l'espace vectoriel engendré par une solution et ses dérivées sur le corps des fractions rationnelles. Il en va de même pour l'espace engendré par une suite et ses décalées lorsque cette suite satisfait une récurrence linéaire homogène à coefficients polynomiaux. À l'inverse, cet argument de finitude ramène souvent à des calculs matriciels la détermination d'une équation différentielle ou d'une récurrence linéaire.

Par exemple, le produit $h = fg$ de deux solutions f et g d'équations différentielles d'ordre I et J vérifie une équation différentielle d'ordre au plus IJ obtenue en opérant dans l'espace vectoriel de dimension finie engendré par $\{f^{(i)}g^{(j)}, 0 \leq i < I, 0 \leq j < J\}$, où s'expriment toutes les dérivées de h.

Le même argument de finitude s'applique à la somme de deux solutions. On montre de la même manière que le produit et la somme de suites (u_n) et (v_n) solutions de récurrences vérifient des récurrences qui se calculent par élimination gaussienne. Il en va de même de la suite de terme général $\sum_{k=0}^{n} u_k v_{n-k}$, suite des coefficients du produit $\sum u_n x^n \sum v_n x^n$. La transformée de Laplace de $f = \sum f_n x^n$ est solution d'une équation différentielle lorsque f l'est : les coefficients du développement à l'infini sont donnés par $n! f_n$. Le même raisonnement s'applique à la transformée de Borel de f (une forme de Laplace inverse) et au produit de Hadamard (produit terme à terme) de deux séries. Dans tous ces cas, un calcul préliminaire de l'équation différentielle satisfaite par la fonction permet l'obtention des n premiers coefficients du développement en série en $O(n)$ opérations.

Exemple 5. Les développements locaux des solutions singulières d'équations différentielles linéaires font intervenir des séries qui satisfont elles aussi des équations différentielles. Le calcul de ces équations par les méthodes qui viennent d'être décrites donne un moyen de développer ces séries à de grands ordres à faible coût, ce qui est notamment utile pour les calculs de resommation de séries divergentes [12].

Exemple 6. Pour calculer le coefficient de x^{3000} du polynôme

$$(x+1)^{2000}(x^2+x+1)^{1000}(x^4+x^3+x^2+x+1)^{500},$$

il suffit de remarquer que les trois termes du produit satisfont des équations d'ordre 1 très simples. Si $p = y_1 y_2 y_3$ est le polynôme, alors

$$\begin{aligned} p'(x) &= y_1'(x)y_2(x)y_3(x) + y_1(x)y_2'(x)y_3(x) + y_1(x)y_2(x)y_3'(x), \\ &= \left(\frac{2000}{x+1} + \frac{1000}{x^2+x+1} + \frac{500}{x^4+x^3+x^2+x+1}\right) p(x). \end{aligned}$$

Après normalisation de cette dernière équation, l'extraction des coefficients fournit une récurrence linéaire (d'ordre 7) qui permet de calculer n coefficients de $p(x)$ en $O(n)$ opérations sur des entiers, et exige une très faible quantité de mémoire (7 entiers en multiprécision).

2.2. **Fonctions algébriques.** La technique suivante [4] remonte au dix-neuvième siècle. Elle est parfois connue sous le nom de résolvante de Cayley ou résolvante différentielle. Elle permet de calculer rapidement les séries de Taylor des fonctions algébriques. À partir d'une équation polynomiale $P(x, y(x)) = 0$, avec P un polynôme irréductible, les calculs sont linéaires en dimension finie grâce à l'inversion de la dérivée P_y modulo P : l'algorithme d'Euclide étendu fournit les polynômes A et B dans l'identité de Bézout

$$A(x,y)P_y(x,y) + B(x,y)P(x,y) = 1.$$

En dérivant $P(x, y(x)) = 0$ par rapport à x, il s'en déduit

$$y'(x) = -A(x,y)P_x(x,y(x)),$$

c'est-à-dire une expression de $y'(x)$ comme une combinaison linéaire des puissances de $y(x)$ à coefficients des fractions rationnelles en x. À partir de là il est facile d'exprimer toutes les dérivées de $y(x)$ de cette manière. L'équation $P(x, y) = 0$ montre que ces puissances engendrent un espace vectoriel de dimension finie. Le calcul s'achève par une élimination gaussienne qui donne l'équation différentielle cherchée.

Exemple 7. Partant de l'équation $P(x, y) = x^2y^2 + (x - 1)y + 1 = 0$ satisfaite par la série génératrice des nombres de Motzkin, l'algorithme d'Euclide étendu fournit d'abord

$$\frac{2x^2y + x - 1}{(1+x)(1-3x)} P_y' + B(x,y)P = 1.$$

La dérivée de $y(x)$ peut donc s'écrire

$$y' = -\frac{2x^2y + x - 1}{(1+x)(1-3x)}(2xy^2 + y) = \frac{(3x^2 + 3x - 2)y + 2}{x(1+x)(1-3x)},$$

ce qui est l'équation différentielle utilisée pour calculer la récurrence dans l'exemple 4.

Exemple 8. La fonction d'Airy $\mathrm{Ai}(z)$ est solution de l'équation différentielle $y''(z) - zy(z) = 0$. Elle admet également une représentation intégrale [21]

$$\mathrm{Ai}(z) = \frac{z^{1/2}e^{-\xi}}{\pi}\int_{-\infty}^{\infty} e^{-\xi((u-1)(4u^2+4u+1))}\,dv, \quad \xi = \frac{2}{3}z^{3/2}, \quad u = \sqrt{1+v^2/3}.$$

Pour étudier le comportement asymptotique de cette intégrale, la méthode de Laplace évoquée à l'exemple 3 suggère d'y effectuer le changement de variable $t^2 = (u-1)(4u^2+4u+1)$ qui amène à la forme

$$\mathrm{Ai}(z) = \frac{z^{1/2}e^{-\xi}}{\pi}\int_{-\infty}^{\infty} e^{-\xi t^2} f(t)\,dt, \qquad f = \frac{dv}{dt},$$

où il reste à calculer le développement de Taylor $f(t) = \sum_{n\geq 0} c_n t^n$.

L'élimination de u par un calcul de résultant donne une équation algébrique liant v à t à partir de laquelle la méthode qui vient d'être décrite fournit

$$(9t^2+18)v''(t) + 9tv'(t) - 4v(t) = 0.$$

Les coefficients c_n obéissent à une récurrence aisément calculée qui ne comporte que deux termes. Elle se résout donc explicitement. L'application de la méthode de Laplace renvoie alors

$$\mathrm{Ai}(x) \sim \frac{x^{-1/4}}{2\sqrt{\pi}} e^{-\frac{2}{3}x^{3/2}} \sum_{n\geq 0} (-1)^n \frac{(2n+1)(2n+3)\cdots(6n-1)}{12^{2n} n! x^{3n/2}}.$$

Le même raisonnement montre également l'existence et fournit un mode de calcul pour une équation différentielle satisfaite par $F(y(x))$ lorsque F satisfait une équation différentielle et y est algébrique.

Exemple 9. La moyenne arithmético-géométrique de deux réels $0 < b < a$ est définie comme la limite commune $M(a,b)$ des suites

$$a_{k+1} = \frac{a_k + b_k}{2}, \qquad b_{k+1} = \sqrt{a_k b_k}, \qquad a_0 = a, b_0 = b.$$

Les identités $M(a,b) = aM(1, b/a)$ et $M(a_0, b_0) = M(a_1, b_1)$, appliquées à $a_0 = 1+x$ et $b_0 = 1-x$, entraînent l'équation fonctionnelle suivante

$$(1+x)A(x^2) = A\left(\frac{4x}{(1+x)^2}\right) \tag{4}$$

satisfaite par $A(x) = 1/M(1, \sqrt{1-x})$. Il n'est pas difficile de montrer que A est analytique au voisinage de l'origine. L'équation ci-dessus permet de calculer un développement en série pour A sur lequel on observe que les coefficients de Taylor vérifient

$$u_{n+1} = \frac{(2n+1)^2}{4(n+1)^2} u_n. \tag{5}$$

Ceci mène naturellement au résultat suivant dû à Gauss :
$$A(x) = \sum_{n\geq 0} \frac{(1/2)(3/2)\cdots(n-1/2)}{n!^2} x^n.$$
Pour prouver cette égalité, il suffit de montrer que la somme infinie vérifie (4). Pour cela, une équation différentielle satisfaite par la somme est tirée de (5). Ensuite, les algorithmes décrits ci-dessus construisent des équations satisfaites par les membres droit et gauche de (4), puis une équation satisfaite par la différence des deux membres. Les solutions analytiques de cette équation ont une suite de coefficients qui vérifient une récurrence d'ordre 3. Comme la série vérifie (4) à l'ordre 3, elle est solution de cette équation différentielle, ce qui achève la preuve.

2.3. **Le cas multivarié.** Les méthodes évoquées précédemment s'étendent aux systèmes d'équations différentielles ou aux différences linéaires multivariés, ce qui permet de traiter diverses sommes et intégrales définies faisant intervenir des fonctions spéciales ou des polynômes orthogonaux. En particulier, il est souvent possible de calculer les développements en séries de telles sommes ou intégrales avec une complexité réduite. Nous nous limitons ici à un exemple d'application. Nous renvoyons le lecteur intéressé par plus de détails à [11, 3].

Exemple 10. La littérature regorge d'identités du type
$$\int_0^\infty xJ_1(ax)I_1(ax)Y_0(x)K_0(x)\,dx = -\frac{\ln(1-a^4)}{2\pi a^2},$$
où les fonctions J_1, I_1, Y_0 et K_0 sont des fonctions de Bessel qui vérifient des équations différentielles linéaires d'ordre 2. Les algorithmes de [3] partent de ces équations différentielles linéaires pour calculer d'abord un système d'équations satisfaites par l'intégrande, puis une équation différentielle satisfaite par l'intégrale. Cette équation est linéaire d'ordre 4, mais la récurrence linéaire satisfaite par les coefficients d'une solution ne fait intervenir que deux termes, ce qui permet de la résoudre et d'obtenir le membre droit.

Exemple 11. Les coefficients de Fourier, de Bessel, ou des développements de fonctions sur une base de polynômes orthogonaux classiques obéissent à une récurrence linéaire dès lors que la fonction développée satisfait une équation différentielle linéaire. Les méthodes évoquées ici calculent ces récurrences de changement de base.

3. Échelles multiples et multiséries

Les procédés de calcul rapide décrits jusqu'ici manipulent des développements en séries entières, mais ne font pas d'hypothèse sur l'échelle asymptotique utilisée. Elles s'appliquent naturellement aux points au voisinage desquels les fonctions à développer sont analytiques ; elles sont également effectives au voisinage de points singuliers, une fois déterminée l'échelle asymptotique dans laquelle les calculs se déroulent. Par exemple, il est manifeste que le calcul du développement asymptotique au voisinage de l'infini

$$\exp(e^{-x}) = 1 + e^{-x} + \frac{e^{-2x}}{2!} + \cdots$$

se ramène à un calcul sur des séries entières.

De nouveaux problèmes algorithmiques surgissent lorsque plusieurs modes de croissance asymptotique interviennent au sein d'un calcul. Ainsi, pour déterminer le comportement asymptotique de

$$\exp\left(\frac{1}{x+e^{-x}}\right) - \exp\left(\frac{1}{x}\right), \qquad x \to \infty$$

il ne faut pas développer en puissances de $1/x$, car cela mène à prolonger indéfiniment le calcul. Il faut commencer par effectuer un développement en puissances de $\exp(-x)$ avant d'en développer le coefficient dominant. Ce problème *d'annulation perpétuelle* peut rapidement devenir complexe. L'intérêt d'en automatiser la solution est patent sur des exemples comme

$$f(x) = \ln\ln(1 + x\exp(xe^x)) - \exp(e^{\ln\ln x + 1/x}), \qquad x \to \infty. \tag{6}$$

Loin d'être des artefacts apparaissant sur des exemples *ad hoc*, ces problèmes d'annulation perpétuelle se rencontrent fréquemment, par exemple dans certains calculs asymptotiques de variance en probabilités discrètes.

Une étude systématique de ces questions de détermination automatique des échelles asymptotiques a récemment permis l'algorithmisation complète des calculs asymptotiques pour de très larges classes de problèmes [13, 16, 17, 22]. Après avoir précisé le cadre de ces travaux en §3.1, nous détaillons les algorithmes qui en découlent pour les fonctions explicites au §3.3, pour leurs inverses fonctionnels au §3.4 et pour les fonctions implicites au §3.5. Le §3.6 donne des indications sur la façon dont les solutions d'équations différentielles algébriques se prêtent également à cette approche.

3.1. **Échelles et développements asymptotiques.** Les principales notions, définitions et notations dont nous ferons usage à propos de développements asymptotiques se trouvent dans [6, Chap. III]. Pour résumer, une échelle asymptotique au voisinage d'un point X est un ensemble de fonctions tel que le quotient de deux de ses éléments possède une limite, qui de surcroît est nulle ou infinie. Un développement asymptotique dans une telle échelle est une somme formelle

$$a_1\phi_1 + \cdots + a_k\phi_k + \cdots$$

où les ϕ_i sont des fonctions de l'échelle satisfaisant $\phi_{i+1} = o(\phi_i)$ au voisinage de X pour $i \geq 1$ et les coefficients a_i sont bornés et ne tendent pas vers 0 au voisinage de X.

Exemple 12. Les développements de Taylor des fonctions analytiques à l'origine s'expriment dans l'échelle $\{x^n, n \in \mathbb{N}\}$ avec des coefficients qui sont des constantes de $\mathbb{C}$.

Exemple 13. Le développement de la fonction J_0 de Bessel à l'infini est

$$J_0(x) \sim \sqrt{\frac{2}{\pi x}} \sum_{m \geq 0} \left(\cos(x - \pi/4)(-1)^m \frac{(4m)!^2}{(2m)!^3 2^{10m} x^{2m}} \right. \\ \left. - \sin(x - \pi/4)(-1)^m \frac{(4m+2)!^2}{(2m+1)!^3 2^{10m+5} x^{2m+1}} \right).$$

Ici, l'échelle asymptotique est $\{x^{-n-1/2}, n \in \mathbb{N}\}$ et les coefficients sont des fonctions trigonométriques.

Exemple 14. L'échelle asymptotique utilisée pour le développement asymptotique de la fonction d'Airy dans l'exemple 8 est $\{e^{-\frac{2}{3}x^{3/2}} x^{-3k/2-1/4}, k \in \mathbb{N}\}$ et les coefficients sont des nombres complexes.

Il est commode de se limiter à des échelles asymptotiques closes par produit et contenant les puissances réelles de tous leurs éléments. Les échelles asymptotiques sont alors spécifiées par la donnée d'une *base asymptotique*, c'est-à-dire de leurs générateurs pour ces deux opérations. En pratique, ces bases sont finies, ce qui en permet la manipulation par le calcul formel. Par convention, les éléments de ces bases sont choisis tendant vers l'infini et les bases sont ordonnées. Dans les exemples ci-dessus, des bases asymptotiques sont (x) et $(x, \exp(x^{3/2}))$. Si $(t_1, \ldots, t_n)$ est une telle base, $\mathbb{R}[[t_1; \ldots; t_n]]$ désigne l'ensemble des développements asymptotiques à coefficients réels dans l'échelle $\{t_1^{\alpha_1} \cdots t_n^{\alpha_n}, \alpha_i \in \mathbb{R}, 1 \leq i \leq n\}$.

L'objectif de la recherche actuelle en asymptotique automatique est de fournir pour de grandes classes de problèmes un calcul simultané des bases asymptotiques et des développements recherchés. Pour éviter les problèmes d'annulation perpétuelle, les algorithmes manipulent à la fois développements et représentations *exactes* des fonctions développées. Il est alors nécessaire de disposer d'algorithmes testant la nullité des fonctions.

3.2. **Les fonctions exp-log.** Les échelles asymptotiques les plus importantes en pratique sont constituées de fonctions exp-log [8]. Ces fonctions sont les fonctions réelles obtenues à partir d'une variable x et des constantes réelles par application répétée de l'exponentielle, du logarithme, des opérations de corps $(+, -, \times, \div)$ et de l'opération d'extension par les fonctions réelles solutions d'équations polynomiales à coefficients exp-log. Par exemple, la fonction f donnée dans l'équation (6) est une fonction exp-log, mais $\exp(\sqrt{-x^2}) + \exp(-\sqrt{-x^2})$ ne l'est pas, à cause de la contrainte de réalité.

L'importance des fonctions exp-log en asymptotique provient d'une propriété cruciale : les fonctions exp-log sont continues et ne s'annulent pas pour des valeurs suffisament grandes de leur argument ; elles sont donc asymptotiquement de signe fixe. Les dérivées successives d'une fonction exp-log étant elles-mêmes exp-log, il s'ensuit une très grande régularité du comportement asymptotique de ces fonctions. En particulier, toute fonction exp-log possède une limite, finie ou non, à l'infini. Comme les fonctions exp-log sont closes par opérations de corps, le quotient de deux fonctions exp-log possède également une limite, ce qui permet de comparer et d'ordonner ces fonctions.

Pour détecter la nullité d'une fonction exp-log, l'algorithme de Risch utilisé en intégration formelle réduit la question à la reconnaissance de la nullité des *constantes* exp-log. On dispose en théorie d'un tel algorithme [14], en admettant une conjecture de Schanuel qui affirme l'indépendance algébrique des exponentielles de complexes linéairement indépendants sur $\mathbb{Q}$.

3.3. **Multiséries pour les fonctions exp-log.** L'algorithme de calcul du comportement asymptotique des fonctions exp-log calcule simultanément le développement de la fonction donnée et la base asymptotique nécessaire aux calculs. Cette base $(t_1, \ldots, t_n)$ obéit aux contraintes suivantes :

1. $t_1 = \log_k x$ $(k \in \mathbb{N})$, le logarithme itéré k fois de x $(\log_0 x = x)$;
2. $\log t_i = o(\log t_{i+1})$, $1 \leq i < n$;
3. $\log t_{i+1}$ admet un développement asymptotique dans $\mathbb{R}[[t_1; \ldots; t_i]]$, pour $1 \leq i < n$.

En outre, l'algorithme assure que les éléments de la base sont eux-mêmes des fonctions exp-log.

Une telle base étant déterminée, toute expression exp-log en les t_i, préparée de telle sorte que les arguments des exponentielles ne tendent pas vers l'infini et ceux des logarithmes ne tendent ni vers 0 ni vers l'infini, peut être développée par rapport à t_n par des manipulations de séries en une variable. L'intérêt de cette préparation est que ces manipulations fournissent nécessairement au moins un terme non nul en un nombre fini d'étapes si l'on a pris soin de tester auparavant la non nullité de la fonction.

Exemple 15. Un procédé détaillé plus loin écrit la fonction f de l'équation (6) sous la forme

$$f = t_3 + t_2 + \log\left(1 + \frac{t_2 + \log(1 + t_3^{-1} t_5^{-1})}{t_3 t_4}\right) - t_3 \exp(t_2 e^{1/t_3} - t_2), \tag{7}$$

la base asymptotique étant $(t_1, \ldots, t_5) = (\log\log x, \log x, x, e^x, \exp(xe^x))$. Le développement par rapport à t_5 s'écrit alors

$$f = f_0 + f_1 t_5^{-1} + f_2 t_5^{-2} + \cdots .$$

Les coefficients f_i sont des fonctions exp-log; il est donc possible de tester la non nullité de

$$f_0 = t_3 + t_2 + \log\left(1 + \frac{t_2}{t_3 t_4}\right) - t_3 \exp(t_2 e^{1/t_3} - t_2).$$

Le calcul du développement de ce premier terme par rapport à t_4 puis du nouveau premier terme trouvé par rapport à t_3 et ainsi de suite jusqu'à obtenir des coefficients constants détermine que

$$f \sim -\frac{t_2^2}{2t_3} = -\frac{\log^2 x}{2x},$$

malgré les annulations qui se produisent dans des calculs intermédiaires, et qui reflètent des annulations perpétuelles dans les calculs en puissances de $1/x$. Un nombre arbitraire de coefficients du développement par rapport à n'importe quel élément de la base se détermine de manière similaire.

Le polygone de Newton

En admettant connue une base asymptotique $(t_1, \ldots, t_k)$ pour les coefficients de l'équation

$$a_0(x) + a_1(x)y + \cdots + a_d(x)y^d = 0, \tag{8}$$

une méthode mise au point par Newton pour le cas des séries formelles bivariées détermine le développement des solutions. Les coefficients $a_i(x)$, $0 \leq i \leq d$, admettent des développements en puissances décroissantes de t_k qui s'obtiennent par des manipulations de séries en une variable comme ci-dessus. En particulier, il est possible de calculer l'exposant α_i de t_k dans le premier terme non nul du développement de a_i, pour $0 \leq i \leq d$. Une solution y de (8) croît au plus vite comme une puissance de t_k. La méthode du polygone de Newton permet de déterminer les premiers termes du développement des solutions en puissances décroissantes de t_k, jusqu'à un stade où une itération de Newton s'applique pour calculer rapidement les termes suivants.

Si t_k^α est un mode de croissance possible pour y, au moins deux termes en $t_k^{i\alpha+\alpha_i}$ doivent dominer tous les autres dans le membre gauche de (8), et leurs coefficients doivent s'annuler. Ceci s'interprète graphiquement sur l'enveloppe convexe de l'ensemble de quarts de plans $\cup_{i=0}^d \left((i, -\alpha_i) + \mathbb{R}_+^2\right)$. Pour chaque segment de cette enveloppe, la pente du segment donne une valeur possible de l'exposant α — qui ne peut prendre que ces valeurs ; le coefficient correspondant est obtenu par une équation du type (8) de degré la différence des abscisses des extrémités du segment, où les coefficients se développent dans la base asymptotique plus petite $(t_1, \ldots, t_{k-1})$.

Calcul de la base

Le calcul de la base est effectué par induction sur l'expression de la fonction. Au commencement la base est réduite à (x). La question de son extension se pose chaque fois qu'un logarithme ou une exponentielle sont rencontrés. À ce stade est déjà connue la base $(t_1, \ldots, t_k)$ dans laquelle l'argument f de ce logarithme ou de cette exponentielle peut être développé. En particulier, des développements en série produisent un équivalent $A = ct_1^{\alpha_1} \cdots t_k^{\alpha_k}$ sur lequel la limite de f est apparente. Pour le logarithme, l'équation

$$\log(f) = \log c + \alpha_1 \log(t_1) + \cdots + \alpha_k \log(t_k) + \log(1 + (f - A)/A)$$

montre que seul le cas où $\alpha_1 \neq 0$ nécessite une adjonction à la base, en tête de laquelle il suffit de rajouter $\log(t_1)$. Par construction cet élément est de la forme $\log_k x$. Le cas de l'exponentielle est plus délicat. Une extension de la base n'est à envisager que lorsque f tend vers l'infini. Pour $1 < i \leq k$, il faut tester si $\alpha = \lim(f/\log t_i) \in \mathbb{R}^\star$, ce qui est réalisable par des calculs de séries. Si cela se produit, f est récrit $t_i^\alpha \exp(f - \alpha \log t_i)$. La question est reposée pour cette nouvelle exponentielle. Après un nombre fini d'étapes, l'algorithme atteint soit une situation où l'argument de l'exponentielle a une limite finie, soit $\log t_i = o(f)$ et $f = o(\log t_{i+1})$ pour un $i > 1$ (ou $\log t_n = o(f)$). Il suffit alors d'insérer la nouvelle exponentielle dans la base entre t_i et t_{i+1} (ou après t_n).

Exemple 16. Les premières étapes du déroulement de la construction pour la fonction f de (6) sont assez directs. Elles fournissent rapidement la base annoncée plus haut $(t_1, t_2, t_3, t_4, t_5) = (\log \log x, \log x, x, e^x, \exp(xe^x))$ en termes de laquelle le premier sommant de f s'écrit

$$\log \log(xe^{xe^x} + 1) = t_3 + t_2 + \log \left(1 + \frac{t_2 + \log(1 + t_3^{-1} t_5^{-1})}{t_3 t_4}\right)$$

tandis que l'argument des exponentielles du second sommant s'écrit $t_1 + 1/t_3$ et donc tend vers l'infini. La comparaison à $\log t_2$ montre alors qu'il faut récrire ce terme sous la forme $\exp(t_1 + 1/t_3) = t_2 \exp(1/t_3)$ où le nouvel argument de l'exponentielle tend vers 0. Ainsi la base est déterminée et la récriture (7) effectuée, ce qui permet tous les calculs ultérieurs.

3.4. **Inverses fonctionnels.** Le même type de traitement s'applique à tous les inverses des fonctions exp-log. Étant données les équations

$$y(f) = f(y) = x, \qquad x \to \infty, \tag{9}$$

où f est une fonction exp-log connue tendant vers l'infini à l'infini, il s'agit de déterminer une base asymptotique et le développement asymptotique de y dans cette base.

De telles questions sont fréquentes lors de l'application de la méthode du col à des développements de coefficients de fonctions génératrices en combinatoire [5]. Par exemple, la détermination du comportement asymptotique des nombres de Bell fait classiquement intervenir l'équation

$$Re^R - 1 = n, \qquad n \to \infty. \tag{10}$$

La détermination du nombre moyen de sous-ensembles dans une partition d'un ensemble de taille n demande en outre la connaissance de la solution R_1 de l'équation

$$\frac{R_1 e^{2R_1} - e^{R_1} + 1}{e^{R_1} - 1} = n, \qquad n \to \infty, \tag{11}$$

et en particulier une estimation asymptotique de $R - R_1$. L'utilisation de méthodes classiques mène à un même développement pour R et R_1 :

$$\log n - \log\log n + \frac{\log\log n}{\log n} + \cdots . \tag{12}$$

L'usage de multiséries produit un résultat infiniment plus précis, duquel un développement asymptotique de $R - R_1$ se déduit aisément, alors qu'il est inaccessible par la théorie classique.

Une fois déterminée une base asymptotique $(t_1, \ldots, t_k)$ pour la fonction f par l'algorithme de la section précédente, l'équation (9) se récrit

$$a_0(y)t_k^{\alpha_0}(y) + a_1(y)t_k^{\alpha_1}(y) + \cdots = x =: a_0(y)t_k^{\alpha_0}(y) + g(y),$$

avec $\alpha_0 > \alpha_1 > \cdots$. L'indéterminée x tendant vers l'infini, il en va de même pour le premier terme du membre gauche.

Le cas le plus propice est celui où le calcul peut se mener comme si le membre gauche était une série formelle, ce qui se produit lorsque $k = 1$, ou lorsque simultanément $\alpha_0 = 0$, $\log a_0(y) = O(\log y)$ et $\log y = o(\log g(y))$. Il est toujours possible de s'y ramener en un nombre fini d'étapes ; soit en calculant l'inverse de $f(e^y)$ lorsque $\alpha_0 = 0$ et $\log t_k(y) = O(\log y)$ (alors $\log g(y) = O(\log y)$), soit en calculant l'inverse de $\log(f)$ lorsque $\alpha_0 = 0$ et $\log y = o(\log a_0(y))$ ou lorsque $\alpha_0 > 0$ et $k \neq 1$. L'inverse de f lui-même s'en déduit par la composition inverse.

Une fois ces conditions remplies, en nommant y_0 la solution de

$$a_0(y_0)t_k^{\alpha_0}(y_0) = x$$

(dont un développement en multiséries est obtenu par une application récursive du même algorithme), la substitution de $y_0(x)$ à x dans l'équation (9) fournit $y(x + g(y_0(x))) = y_0(x)$. La solution de cette équation est $y(x) = y_0(x) + v(y_0(x))$, où $x + v(x)$ est solution de cette même équation avec x comme membre droit. La fonction v satisfait $v(x) = -g(y_0(x + v(x)))$. Elle se calcule par l'itération

$$v_{n+1}(x) = -g(y_0(x + v_n(x))),$$

initialisée avec $v_0 = 0$, qui est équivalente à l'itération de Newton dans le cas des séries formelles.

Pour mener le calcul de cette itération, il est utile d'éviter l'interférence de termes en x avec des termes en y_0. La réduction à des expressions ne faisant intervenir que y_0 s'opère en trois temps : tout d'abord y_0 étant l'inverse d'une fonction exp-log, ses dérivées s'expriment comme des fonctions exp-log en y_0 ; ensuite un développement de $y_0(x+g(y_0))$ en puissances de $t_k(y_0)$ à coefficients des fonctions exp-log en y_0 s'obtient par développement de Taylor ; enfin ce développement permet de calculer ceux des fonctions $t_i(y_0(x+g(y_0)))$ pour $1 \leq i \leq k$, à partir desquels l'itération peut être menée à bien, tous les tests à zéro opérant sur des fonctions exp-log en la variable y_0.

Exemple 17. L'équation (10) n'est pas sous la forme voulue pour effectuer l'itération, mais elle s'y ramène en une étape de l'algorithme par une application du logarithme, qui donne

$$R + \log R + \log(1 - e^{-R}/R) = \log n.$$

En nommant ζ la solution de $\zeta(n) + \log \zeta(n) = \log n$, le développement de R est donné par $R(n) = \zeta(n) + v(\zeta(n))$, où

$$v(x) = -\log(1 - e^{-\zeta(x+v(x))}/v(x)), \tag{13}$$

se calcule par itération, ce qui donne

$$R = \zeta + \frac{e^{-\zeta}}{\zeta+1} - \frac{1}{2}\frac{\zeta+2}{(\zeta+1)^3}e^{-2\zeta} + \cdots .$$

Une première approximation de R est $\zeta(n)$, dont le développement s'obtient par une application récursive de l'algorithme qui mène à écrire

$$\log \zeta(n) + \log\left(1 + \frac{\log \zeta(n)}{\zeta(n)}\right) = \log \log n,$$

et produit (12). Le calcul du développement de R_1 défini par l'équation (11) amène à l'introduction de la même fonction ζ mais avec une itération différente, qui donne

$$R_1 = \zeta - \frac{\zeta-1}{\zeta+1}e^{-\zeta} - \frac{1}{2}\frac{3\zeta^3+2\zeta^2-\zeta+4}{(\zeta+1)^3}e^{-2\zeta} + \cdots .$$

Le développement de la différence $R - R_1$ est alors facile à calculer, d'abord en ζ, puis de façon plus grossière en n.

3.5. **Fonctions implicites.** Le cas des fonctions implicites définies par une équation $f(x, y) = 0$ où f est une fonction exp-log en deux variables est beaucoup plus délicat. Grosso modo, l'algorithme repose sur une version bivariée de l'algorithme de développement des fonctions exp-log en une variable. La difficulté provient de ce que la base asymptotique fait alors intervenir à la fois des fonctions de x et des fonctions de y. Les extensions de la base mènent à une scission des calculs tenant compte de toutes les possibilités de comportements asymptotiques relatifs pour y et x. Ce calcul produit des estimations de $f(x, y)$ dans des domaines définis par des relations comme $\phi(x, y) = o(\psi(x, y))$ pour ϕ et ψ des fonctions exp-log. Les cas limites sont de la forme $\phi(x, y) \sim \psi(x, y)$. Ils contiennent les cas où f peut s'annuler. Un raffinement de l'estimation donnée par la condition dans ces cas aboutit au développement des fonctions implicites.

Exemple 18. Pour la fonction

$$H(x,y)=\frac{e^{x^2+2x\log^2 x+y}}{e^{x^2+x\log^2 x-1}},\qquad x\to\infty$$

la méthode évoquée ci-dessus fournit

$$H\sim\begin{cases}-e^{x^2+2x\log^2 x+y} & \text{si } y\to-\infty \text{ et } \log x=o(\log|y|),\\ ? & \text{si } \log|y|\sim 2\log x,\\ \exp(x\log^2 x) & \text{sinon.}\end{cases}$$

La résolution de l'équation $H+1=0$ conduit alors à raffiner le cas intermédiaire, ce qui mène en quelques étapes (laborieuses à la main!) aux premiers termes du développement de la solution

$$y(x)=-x^2-2x\log^2 x+\frac{e^{-x\log^2 x}}{x^2}-\frac{e^{-2x\log^2 x}}{2x^2}+\cdots.$$

3.6. **Équations différentielles algébriques.** Le développement des solutions d'équations différentielles algébriques de la forme $P(x,y,y',\dots,y^{(n)})=0$, où P est un polynôme est sensiblement plus délicat que les cas abordés jusqu'ici, qui n'en sont que des cas particuliers.

Une première difficulté provient du problème de la reconnaissance de zéro. Tous les tests à zéro dans les algorithmes précédents reposent sur le test à zéro des fonctions exp-log, pour lequel un algorithme existe en admettant la conjecture de Schanuell. Dans le cas des solutions d'équations différentielles algébriques, il existe des algorithmes de test à zéro [20, 10] qui réduisent la question à des tests à zéro sur des constantes, pour lesquels on ne dispose pas d'algorithme général à l'heure actuelle.

Rubel a exhibé [15] une équation différentielle d'ordre 4 admettant des solutions C^∞ sur $\mathbb{R}$ arbitrairement proches de toute fonction continue sur $\mathbb{R}$. Ceci montre que pour les questions asymptotiques, il est nécessaire de restreindre la classe des solutions considérées. Une restriction communément adoptée consiste à se concentrer sur les solutions possédant les mêmes propriétés que les fonctions exp-log, c'est-à-dire dont les germes à l'infini appartiennent à un corps clos par dérivation. Un tel corps de germes s'appelle un *corps de Hardy.* Ces fonctions sont asymptotiquement de signe constant, ainsi que leurs dérivées successives. Elles sont donc d'une très grande régularité.

Dans ces conditions, il est possible de classifier de manière finie les modes de croissance possibles des solutions pour tout ordre. Ceci donne un ensemble fini duquel on peut extraire et tester les ordres de grandeurs asymptotiques des solutions. Un changement de fonction inconnue permet alors de rechercher la suite du développement asymptotique [19] ; cependant l'équation différentielle satisfaite par cette suite est d'ordre plus élevé que l'équation de départ. Il est difficile d'obtenir beaucoup de termes de cette façon.

La difficulté d'une approche utilisant des multiséries dans ce problème est de déterminer une base asymptotique. Pour des indications sur ces questions, nous renvoyons le lecteur à [22].

Conclusion

L'automatisation de méthodes classiques en analyse asymptotique est contrainte par les nécessités du calcul formel. En particulier, il faut garantir la reconnaissance de zéro dans toutes les étapes du calcul, faute de quoi le phénomène d'annulation perpétuelle interdit ne serait-ce que la détermination garantie de limites. Cette contrainte conduit au développement d'une algorithmique spécifique conjuguant des méthodes classiques, dont beaucoup remontent à Newton, avec des idées récentes comme les multiséries, proches cousines des transséries d'Écalle [7]. Le développement de ce domaine — l'asymptotique automatique — est très rapide, et aucun des grands systèmes de calcul formel ne propose à l'heure actuelle une implantation des algorithmes décrits ici. Pourtant, par sa capacité à produire des approximations arbitrairement précises au voisinage de singularités, l'asymptotique automatique est appelée à faire le lien entre le calcul formel et le calcul numérique, ouvrant ainsi un vaste champ d'applications.

Références

1. R. P. Brent and H. T. Kung, *Fast algorithms for manipulating formal power series*, Journal of the ACM **25** (1978), no. 4, 581–595.
2. David V. Chudnovsky and Gregory V. Chudnovsky, *Computer algebra in the service of mathematical physics and number theory*, Computers in mathematics (Stanford, CA, 1986) (New York), Dekker, 1990, pp. 109–232.
3. Frédéric Chyzak, *Gröbner bases, symbolic summation and symbolic integration*, Gröbner Bases and Applications (B. Buchberger and F. Winkler, eds.), London Mathematical Society Lecture Notes Series, vol. 251, Cambridge University Press, 1998, pp. 32–60.
4. L. Comtet, *Calcul pratique des coefficients de Taylor d'une fonction algébrique*, L'Enseignement Mathématique **10** (1964), 267–270.
5. N. G. De Bruijn, *Asymptotic methods in analysis*, Dover, 1981.
6. Jean Dieudonné, *Calcul infinitésimal*, seconde ed., Hermann, Paris, 1980.
7. Jean Écalle, *Introduction aux fonctions analysables et preuve constructive de la conjecture de Dulac*, Actualités mathématiques, Hermann, Paris, 1992.
8. G. H. Hardy, *Orders of infinity*, Cambridge Tracts in Mathematics, vol. 12, Cambridge University Press, 1910.
9. Donald E. Knuth, *The art of computer programming*, 2nd ed., Computer Science and Information Processing, vol. 2: Seminumerical Algorithms, Addison-Wesley Publishing Co., Reading, Mass., 1981.
10. Ariane Péladan-Germa, *Testing identities of series defined by algebraic partial differential equations*, Applied algebra, algebraic algorithms and error-correcting codes (Paris, 1995), Springer, 1995, pp. 393–407.
11. Marko Petkovšek, Herbert S. Wilf, and Doron Zeilberger, $A = B$, A. K. Peters, Wellesley, MA, 1996.
12. Jean-Pierre Ramis, *Séries divergentes et théories asymptotiques*, Panoramas et Synthèses, vol. 121, Société Mathématique de France, 1993.

13. Dan Richardson, Bruno Salvy, John Shackell, and Joris Van der Hoeven, *Asymptotic expansions of exp-log functions*, ISSAC'96 (Y. N. Lakshman, ed.), ACM Press, 1996, pp. 309–313.
14. Daniel Richardson, *How to recognize zero*, Journal of Symbolic Computation **24** (1997), no. 6, 627–645.
15. L. A. Rubel, *A universal differential equation*, Bulletin of the American Mathematical Society **4** (1981), no. 3, 345–349.
16. Bruno Salvy and John Shackell, *Symbolic asymptotics: Functions of two variables, implicit functions*, Journal of Symbolic Computation **25** (1998), no. 3, 329–349.
17. ———, *Symbolic asymptotics: Multiseries of inverse functions*, Journal of Symbolic Computation (1999), À paraître. Disponible également comme rapport de recherche INRIA numéro 3264.
18. Bruno Salvy and Paul Zimmermann, *Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable*, ACM Transactions on Mathematical Software **20** (1994), no. 2, 163–177.
19. John Shackell, *Rosenlicht fields*, Transactions of the American Mathematical Society **335** (1993), no. 2, 579–595.
20. ———, *Zero-equivalence in function fields defined by algebraic differential equations*, Transactions of the American Mathematical Society **336** (1993), no. 1, 151–171.
21. Nico M. Temme, *Special functions*, John Wiley & Sons Inc., New York, 1996.
22. Joris Van der Hoeven, *Asymptotique automatique*, Ph.D. thesis, École polytechnique, Palaiseau, France, 1997.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105,
78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS
Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
(France)
http://www.inria.fr
ISSN 0249-6399