Calcul effectif avec des séries formelles

PAR JORIS VAN DER HOEVEN

C.N.R.S., Université Paris-Sud

Luminy, 21/1/2003

Idée

- Calcul formel sur des fonctions transcendantes analytiques en les représentant par des séries formelles données en un point.
- Classes de fonctions : expressions explicites faisant intervenir $+, -, \times, \exp, \log, \mathrm{erf}$, etc. ou solutions à des équations différentielles.

Problèmes

- Calcul efficace des coefficients.
- Estimations d'erreur et prolongement analytique.
- Testes de nullité.
- Présence de singularités $\longrightarrow$ transséries.

Multiplication zélée de séries formelles

Entrée : $\begin{cases} f = f_0 + \cdots + f_{n-1}\, z^{n-1} \\ g = g_0 + \cdots + g_{n-1}\, z^{n-1} \end{cases}$

Sortie : $h = h_0 + \cdots + h_{n-1}\, z^{n-1} = f\, g + O(z^n)$.

Algorithmes classiques de multiplication

- Multiplication naïve en $O(n^2)$.
- Diviser pour régner en $O(n^{\log_2 3})$.
- Multiplication F.F.T. en $O(n \log n \log \log n)$.

Logarithme

$$\log f = \log f_0 + \int \frac{f'}{f}$$

Exponentiation

Pour n pair, supposons

$$\log g - f = O(z^{n/2}),$$

$$\text{avec} \quad \begin{cases} f = f_0 + \cdots + f_{n-1}\, z^{n-1}\,; \\ g = g_0 + \cdots + g_{n/2-1}\, z^{n/2-1} \end{cases}$$

Alors

$$\tilde{g} = g - (\log g - f)\, g$$

$$\Longrightarrow \log \tilde{g} - f = O(z^n).$$

$\longrightarrow$ algorithme d'exponentiation en $O(M(n))$, comme la multiplication

Calcul paresseux

Les coefficients sont calculés un par un et à chaque étape on n'effectue que les calculs strictement nécessaires. En particulier, on calcule $(f\,g)_n$ dès que $f_0, \ldots, f_n$ et $g_0, \ldots, g_n$ sont connus.

Conséquence

Avantage : f_n et g_n peuvent dépendre de $(f\,g)_0, \ldots, (f\,g)_{n-1}$
$\longrightarrow$ résolution d'équations implicites de point fixe.
Inconvénient : multiplication *a priori* en $O(n^2)$.

Exemple

Calcul de $g = e^f$ par

$$g = \int f'\,g$$

En effet :

$$g_n = \frac{1}{n}\,(f'\,g)_{n-1} = \frac{1}{n}\sum_{i=0}^{n-1}(i+1)\,f_{i+1}\,g_{n-1-i}.$$

Approche détendue [VdH]

Idée : anticipation $\longrightarrow$ accélération

Algorithme naïf

g_2	2		
g_1	1	2	
g_0	0	1	2
$\times$	f_0	f_1	f_2

0 $h_0 = f_0\, g_0$.

1 $h_1 = f_0\, g_1 + f_1\, g_0$.

2 $h_2 = f_0\, g_2 + f_1\, g_1 + f_2\, g_0$.

Algorithme détendu

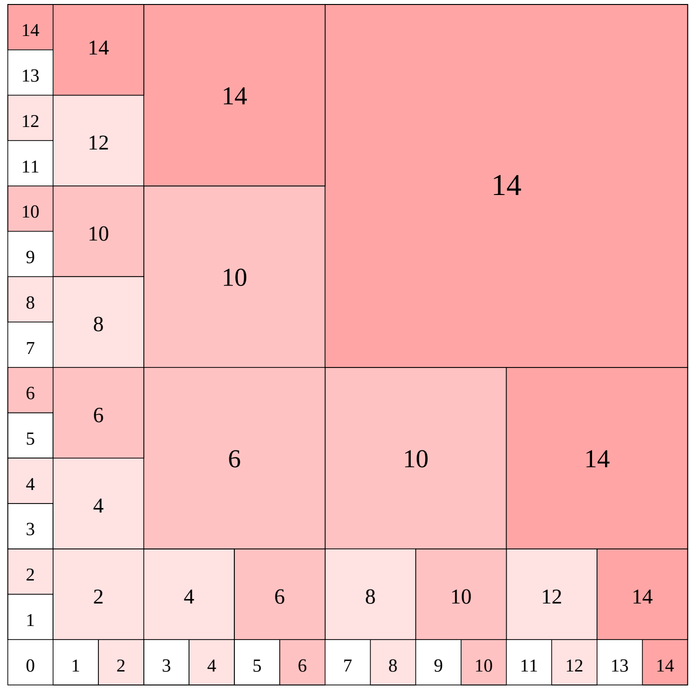
g_2	2		
g_1	1	1	
g_0	0	1	2
$\times$	f_0	f_1	f_2

0 $h_0 = f_0\, g_0$.

1 $h_1 = (f_0 + f_1)\,(g_0 + g_1) - f_0\, g_0 - f_1\, g_1$

2 $h_2 = f_0\, g_2 + f_1\, g_1 + f_2\, g_0$.

Multiplication détendue rapide



$\longrightarrow$ Algorithme détendu en $O(M(n) \log n)$.

Autres algorithmes détendus

Algorithme	Temps	Espace
Multiplication D.P.R.	$n^{\log_2 3}$	$n \log n$
Multiplication rapide	$M(n) \log n$	n
Division	$D(n)$	n
Équations différentielles	$D(n)$	n
Fonctions holonomes	n	n
Composition algébrique	$D(n) \log n$	n
Composition générale	$D(n)\sqrt{n \log n}$	$n^{3/2} \log n$
Composition char. fini	$D(n) \log n$	$n \log n$
Inversion $\rightarrow$ composition	$\uparrow$	$\uparrow$

$D(n)$: temps pour la multiplication détendue

Aussi : [LS]

Fonctions analytiques effectives

Classe abstraite avec méthodes pour

- Calculer le développement en série formel.
- Calculer une borne inférieur ρ pour le rayon de convergence.
- Caluler une borne pour $\sup_{|z| \leqslant r} |f(z)|$ pour tout $r < \rho$.
- Calcul du prolongement analytique vers z pour tout z avec $|z| < r$.

Pour assurer cohérence globale, *aussi* :

- Condition d'homotopie.
- Condition de continuité.

Un des buts : version fiable de COLUMBUS.

Examples de théorèmes

Théorème 1. $L_0, \ldots, L_{l-1}$ *: fonctions analytiques effectives. Alors l'équation*

$$f^{(l)} = L_{l-1} f^{(l-1)} + \cdots + L_0 f, \tag{1}$$

avec cond. init. eff. $f(0) = \nu_0, \ldots, f^{(l-1)}(0) = \nu_{l-1}$ *admet une solution analytique effective* f *avec* $\mathrm{Dom}\, f = \mathrm{Dom}\,(L_1, \ldots, L_l)$*.*

Théorème 2. *Soient* $P_1, \ldots, P_l$ *des polynômes à coeffs. eff. Alors le système*

$$\frac{\mathrm{d}}{\mathrm{d}z} \begin{pmatrix} f_1 \\ \vdots \\ f_l \end{pmatrix} = \begin{pmatrix} P_1(f_1, \ldots, f_l) \\ \vdots \\ P_l(f_1, \ldots, f_l) \end{pmatrix} \tag{2}$$

avec cond. init. eff. $f_1(0) = \nu_1, \ldots, f_l(0) = \nu_l$ *admet une solution analytique effective fidèle* $(f_1, \ldots, f_l)$*.*

Théorème 3. (DL89) *Supposons que* $f \in \mathbb{Q}[[z]]$ *soit la solution unique de*

$$P(z, f, \ldots, f^{(l)}) = 0,$$

avec cond. init. dans $\mathbb{Q}$*, et où* P *est polynomial avec coeffs. dans* $\mathbb{Q}$*. Alors, en général, on ne peut pas décider si* $\rho(f) < 1$ *ou* $\rho(f) \geqslant 1$*.*

Le problème des testes de nullité

Tester des identités functionnelles

- $\sin^2 x + \cos^2 x = 1$
- $\log(x^{x^x} + e^{x \log x}) - x^x \log x = \log\left(1 + x^{x(1 - x^{x-1})}\right)$

identités/fonctions = identités/constantes + identités/séries

Tester des identités pour des constantes

- $\sqrt[3]{\sqrt[5]{32/5} - \sqrt[5]{27/5}} = (1 + \sqrt[5]{3} - \sqrt[5]{9}) / \sqrt[5]{25}$
- $\int_{-\infty}^{\infty} e^{-x^2} \mathrm{d}x = \sqrt{\pi}$

Tester des identités pour des séries formelles

- $Q \in \mathcal{R}[F, F' \ldots, F^{(r)}] \subseteq \mathcal{R}\{F\}$, $\mathcal{R} = \mathcal{K}[z]$, $Q \notin \mathcal{R}$
- $f \in \mathcal{K}[[z]]$ avec $Q(f, f', \ldots, f^{(r)}) = 0$
- Étant donné $P \in \mathcal{R}\{F\}$, a-t-on $P(f) = 0$?
- Tours: replacer $\mathcal{R}$ par $\mathcal{R}[f, \ldots, f^{(r)}, S_Q(f)^{-1}]$ et continuer.

Algorithme $P\equiv 0$

Entrée : Un polynôme $P\in\mathcal{R}\{F\}=\mathcal{K}[\bar{F}]$

Sortie : **true** si et seulement si $P\equiv 0$

if $P(0)\neq 0$ **then return false**

$\mathcal{G}:=\{P\}$

while $\exists B\in\mathcal{G}\colon \mathrm{Red}(B',\mathcal{G})\neq 0$ **do**

  if $B'(0)\neq 0$ **then return false**

  $\mathcal{G}:=\mathrm{GroebnerBasis}(\mathcal{G}\cup\{B\})$

return true

Pourquoi ça marche

Si l'algorithme rend **true**, alors pour tout $B_i\in\mathcal{G}=\{B_1,\ldots,B_n\}$:

$$B_i'=\sum_{j=1}^{n}A_{i,j}B_j$$

En dérivant, on trouve $B_i^{(k)}(0)=0$ pour tout k.

Algorithme de saturation [Sh]

Algorithme $P \equiv 0$

Entrée : Un polynôme $P \in \mathcal{R}\{F\} = \mathcal{K}[\bar{F}]$

Sortie : **true** si et seulement si $P \equiv 0$

if $P(0) \neq 0$ **then return false**

$\mathcal{G} := \{P\}$

while $\exists i \colon \exists B \in G \colon \mathrm{Red}(\partial_i B, G) \neq 0$ **do**

  if $\partial_i B(0) \neq 0$ **then return false**

  $G := \mathrm{GroebnerBasis}(G \cup \{\partial_i B\})$ [P-G,VdH]

return true

Pourquoi ça marche

Si l'algorithme rend **true**, alors pour tout $B_i \in \mathcal{G} = \{B_1, \ldots, B_n\}$:

$$B_i' = \sum_{j=1}^{n} A_{i,j} B_j$$

En dérivant, on trouve $B_i^{(k)}(0) = 0$ pour tout k.

Le théorème de Puiseux

Soit $P \in \mathcal{K}[[z]][F]^*$. Alors

$$P(f) = 0$$

admet $\deg P$ solutions dans $\mathcal{K}^{\mathrm{alg}}[[z^{\mathbb{Q}}]]$.

Généralisation pour des équations algébriques asymptotiques [vdH]

Soient $P \in \mathcal{K}[[z]][F]^*$ et $\nu \in \mathbb{R} \cup \{-\infty\}$. Alors

$$P(f) = 0 \qquad (f \prec z^{\nu})$$

admet $\deg_{\prec z^{\nu}} P$ solutions dans $\mathcal{K}^{\mathrm{alg}}[[z^{\mathbb{Q}}]]$,

où $\deg_{\prec z^{\nu}} P$ désigne le degré de Newton de l'équation.

Le théorème de Puiseux

Soit $P \in \mathcal{K}[[z]][F]^*$. Alors

$$P(f) = 0$$

admet $\deg P$ solutions dans $\mathcal{K}^{\mathrm{alg}}[[z^{\mathbb{Q}}]]$.

Généralisation pour des éq. diff. alg. asymptotiques [vdH]

Soient $\mathfrak{L} = \cdots (\log \log z)^{\mathbb{Q}} (\log z)^{\mathbb{Q}} z^{\mathbb{Q}}$ et $\mathbb{L} = \mathcal{K}^{\mathrm{alg}}[[\mathfrak{L}]]$

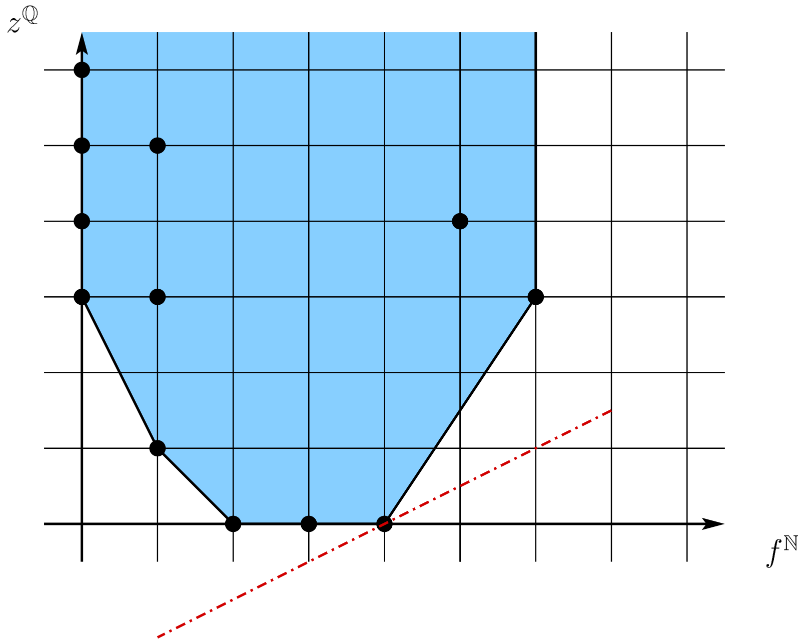
Soient $P \in \mathbb{L}\{F\}^*$ et $\mathfrak{m} \in \mathfrak{L} \cup \{z^{-\infty}\}$. Alors

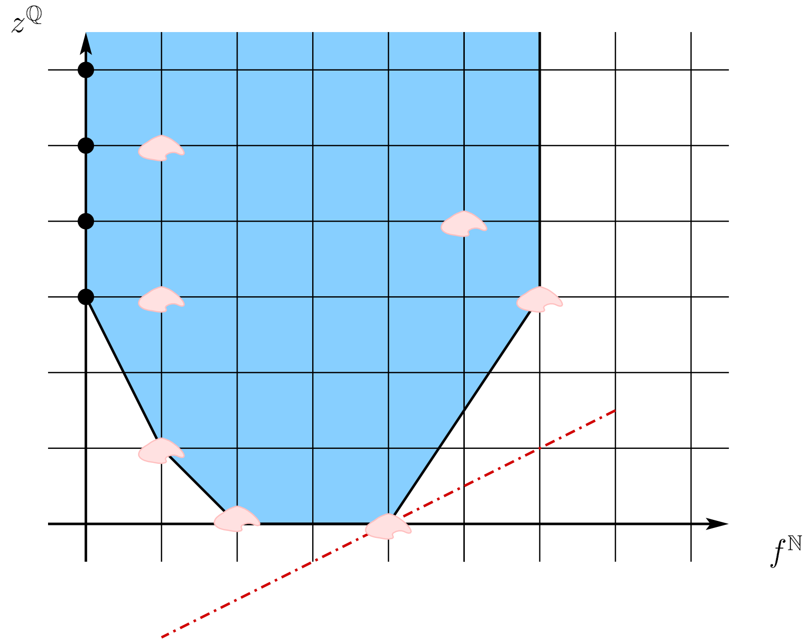
$$P(f) = 0 \qquad (f \prec \mathfrak{m})$$

admet $\deg_{\prec \mathfrak{m}} P$ solutions dans $\mathbb{L}$,

où $\deg_{\prec \mathfrak{m}} P$ désigne le degré de Newton différentiel de l'équation.

Degré de Newton




$z^{\mathbb{Q}}$
$f^{\mathbb{N}}$

Algorithme $P \equiv 0$

Entrée : un polynôme $P \in \mathcal{R}[F]$ où $F' = \frac{N(F)}{D(F)}$ et $D(F)(0) \neq 0$

Sortie : **true** si et seulement si $P \equiv 0$

Étape 1 [Cas trivial]

if $P = 0$ **then return true**

Étape 2 [Réduction]

$P := \mathrm{PseudoSqfree}(P)$

$R := \mathrm{PseudoGcd}(P, D\, P_z + N\, P_F) \approx \mathrm{RittRed}(D\, F' - N, P)$

[Teste final]

return $\deg_{\prec 1} R_{+f_0} \neq 0$ (ici $P_{+\varphi}(f) = P(f + \varphi)$)

Autre algorithme dans cas général [VdH]

Algorithme $P \equiv 0$

Entrée : un polynôme $P \in \mathcal{R}[F]$ où $F' = \frac{N(F)}{D(F)}$ et $D(F)(0) \neq 0$

Sortie : **true** si et seulement si $P \equiv 0$

Étape 1 [Cas trivial]

if $P = 0$ **then return true**

Étape 2 [Réduction]

$P := \mathrm{PseudoSqfree}(P)$

$R := \mathrm{PseudoGcd}(P, D\,P_z + N\,P_F) \approx \mathrm{RittRed}(D\,F' - N, P)$

[Teste final]

return $\deg_{\prec 1} R_{+f_0} \neq 0$ (ici $P_{+\varphi}(f) = P(f + \varphi)$)

Autre algorithme dans cas général [VdH]

Algorithme $P \equiv 0$

Entrée : un polynôme différentiel $P \in \mathcal{R}\{F\}$, préparation OK
Sortie : **true** si et seulement si $P \equiv 0$

Étape 1 [Cas trivial]

if $P = 0$ **then return true**

Étape 2 [Réduction]

$P := \mathrm{PseudoSqfree}(P)$

$R := \mathrm{PseudoGcd}(P, D\,P_z + N\,P_F) \approx \mathrm{RittRed}(D\,F' - N, P)$

[Teste final]

return $\deg_{\prec 1} R_{+f_0} \neq 0$ (ici $P_{+\varphi}(f) = P(f + \varphi)$)

Autre algorithme dans cas général [VdH]

Algorithme $P \equiv 0$

Entrée : un polynôme différentiel $P \in \mathcal{R}\{F\}$, préparation OK
Sortie : **true** si et seulement si $P \equiv 0$

Étape 1 [Cas trivial]

if $P = 0$ **then return true**

Étape 2 [Réduction]

$H := 1$, $R := P$, $\mathrm{reducing} :=$ **true**

while $\mathrm{reducing}$ [invariant: $H \not\equiv 0$ and $P \equiv 0 \Leftrightarrow R \equiv 0$]

if $R \in \mathcal{R}$ **then return** $R = 0$
else if $I_R \equiv 0$ **then** $R := R - I_R V_R$
else if $S_R \equiv 0$ **then** $H := I_R H, R := R \,\mathrm{rem}\, S_R$
else if $Q \,\mathrm{rem}\, R \neq 0$ **then** $H := I_R S_R H, R := Q \,\mathrm{rem}\, R$
else $H := I_R S_R H, \mathrm{reducing} :=$ **false**

[Teste final]

return $\deg_{\prec 1} R_{+f_0} \neq 0$ (ici $P_{+\varphi}(f) = P(f + \varphi)$)

Autre algorithme dans cas général [VdH]

Algorithme $P \equiv 0$

Entrée : un polynôme différentiel $P \in \mathcal{R}\{F\}$, préparation OK
Sortie : **true** si et seulement si $P \equiv 0$

Étape 1 [Cas trivial]

if $P = 0$ **then return true**

Étape 2 [Réduction]

$H := 1$, $R := P$, $\mathrm{reducing} := \mathbf{true}$

while $\mathrm{reducing}$ [invariant: $H \not\equiv 0$ and $P \equiv 0 \Leftrightarrow R \equiv 0$]

if	$R \in \mathcal{R}$	**then return**	$R = 0$
else if	$I_R \equiv 0$	**then**	$R := R - I_R V_R$
else if	$S_R \equiv 0$	**then**	$H := I_R H, R := R \,\mathrm{rem}\, S_R$
else if	$Q \,\mathrm{rem}\, R \neq 0$	**then**	$H := I_R S_R H, R := Q \,\mathrm{rem}\, R$
else			$H := I_R S_R H, \mathrm{reducing} := \mathbf{false}$

[Teste final]

Soit $R = R_q F^q + \cdots + R_0$

Soit $\alpha \in \mathbb{N}$ minimal avec $R_{i,\alpha} \neq 0$ pour un certain i

return $R_{q,\alpha} f_0^q + \cdots + R_{0,\alpha} = 0$

Autre algorithme dans cas général [VdH]

Algorithme $P \equiv 0$

Entrée : un polynôme différentiel $P \in \mathcal{R}\{F\}$, préparation OK

SORTIE : **true** si et seulement si $P \equiv 0$

Étape 1 [Cas trivial]

if $P = 0$ **then return true**

Étape 2 [Réduction]

$H := 1$, $R := P$, $\mathrm{reducing} := \textbf{true}$

while $\mathrm{reducing}$ [invariant: $H \not\equiv 0$ and $P \equiv 0 \Leftrightarrow R \equiv 0$]

$$\begin{array}{lll}
\textbf{if} & R \in \mathcal{R} & \textbf{then return } R = 0 \\
\textbf{else if} & I_R \equiv 0 & \textbf{then } R := R - I_R V_R \\
\textbf{else if} & S_R \equiv 0 & \textbf{then } H := I_R H, R := R \operatorname{rem} S_R \\
\textbf{else if} & Q \operatorname{rem} R \neq 0 & \textbf{then } H := I_R S_R H, R := Q \operatorname{rem} R \\
\textbf{else} & & \phantom{\textbf{then }} H := I_R S_R H, \mathrm{reducing} := \textbf{false}
\end{array}$$

[Teste final]

Soit k minimal avec $\deg_{\prec z^k} H_{+f_0+\cdots+f_k z^k} = 0$

return $\deg_{\prec z^k} R_{+f_0+\cdots+f_k z^k} \neq 0$

Démonstration de l'algorithme

- Supposons $\deg_{\prec 1} R_{+f_0} \neq 0$.
- Alors il existe un $\tilde{f} \in \mathcal{R}^{\mathrm{alg}}((z^{\mathbb{Q}}))$ avec $R(\tilde{f}) = 0$ et $\tilde{f} - f \prec 1$.
- f est la solution unique de $f' = \frac{N(f)}{D(f)}$ modulo $\prec 1$ dans $\mathcal{R}^{\mathrm{alg}}((z^{\mathbb{Q}}))$.
- Puisque $R := \mathrm{PseudoGcd}(P, D\, P_z + N\, P_F)$, on a une relation

$$H\,(D\,F' - N) = X\,R + Y R'.$$

- Puisque $R(\tilde{f}) = 0$ et $H(\tilde{f}) = H \neq 0$, on a $(D\,F' - N)(\tilde{f}) = 0$.
- Or f était la solution unique de $(D\,F' - N)(\tilde{f}) = 0$ modulo $\prec 1$.
- Donc $f = \tilde{f}$ et $R \equiv P \equiv 0$.

- Supposons que $\deg_{\prec z^k} R_{+f_0+\cdots+f_k z^k} \neq 0$.
- Alors il existe un $\tilde{f} \in \mathcal{R}^{\mathrm{alg}}((z^{\mathbb{Q}}))$ avec $R(\tilde{f}) = 0$ et $\tilde{f} - f \prec 1$.
- f est la solution unique de $f' = \frac{N(f)}{D(f)}$ modulo $\prec 1$ dans $\mathcal{R}^{\mathrm{alg}}((z^{\mathbb{Q}}))$.
- Puisque $R := \mathrm{PseudoGcd}(P, D\,P_z + N\,P_F)$, on a une relation

$$H\,(D\,F' - N) = X\,R + Y\,R'.$$

- Puisque $R(\tilde{f}) = 0$ et $H(\tilde{f}) = H \neq 0$, on a $(D\,F' - N)(\tilde{f}) = 0$.
- Or f était la solution unique de $(D\,F' - N)(\tilde{f}) = 0$ modulo $\prec 1$.
- Donc $f = \tilde{f}$ et $R \equiv P \equiv 0$.

Démonstration de l'algorithme

- Supposons que $\deg_{\prec z^k} R_{+f_0+\cdots+f_k z^k} \neq 0$.
- Il existe un $\tilde{f} \in \mathbb{L}$ avec $R(\tilde{f}) = 0$ et $\tilde{f} - f \prec z^k$.
- f est la solution unique de $Q(f) = 0$ modulo $\prec 1$ dans $\mathbb{L}$.
- k suffisamment grand pour que $H(\tilde{f}) \neq 0$ pour *tout* $\tilde{f}$ avec $\tilde{f} - f \prec z^k$.
- Puisque $R := \mathrm{PseudoGcd}(P, D\,P_z + N\,P_F)$, on a une relation

$$H\,(D\,F' - N) = X\,R + Y\,R'.$$

- Puisque $R(\tilde{f}) = 0$ et $H(\tilde{f}) = H \neq 0$, on a $(D\,F' - N)(\tilde{f}) = 0$.
- Or f était la solution unique de $(D\,F' - N)(\tilde{f}) = 0$ modulo $\prec 1$.
- Donc $f = \tilde{f}$ et $R \equiv P \equiv 0$.

Démonstration de l'algorithme

- Supposons que $\deg_{\prec z^k} R_{+f_0+\cdots+f_k z^k} \neq 0$.
- Il existe un $\tilde{f} \in \mathbb{L}$ avec $R(\tilde{f}) = 0$ et $\tilde{f} - f \prec z^k$.
- f est la solution unique de $Q(f) = 0$ modulo $\prec 1$ dans $\mathbb{L}$.
- k suffisamment grand pour que $H(\tilde{f}) \neq 0$ pour *tout* $\tilde{f}$ avec $\tilde{f} - f \prec z^k$.
- Puisque $Q \operatorname{rem} R = 0$ et $I_R S_R | H$, on a une relation du genre

$$H^\beta Q = X_0 R + \cdots + X_t R^{(t)}.$$

- Puisque $R(\tilde{f}) = 0$ et $H(\tilde{f}) \neq 0$, on a $Q(\tilde{f}) = 0$.
- Mais f était la solution unique de $Q(f) = 0$ modulo $\prec z^s$.
- Donc $f = \tilde{f}$ et $R \equiv P \equiv 0$.

Conjecture de témoin pour des constantes exp-log

- $\mathcal{E}^{\mathrm{expr}}$: expressions « valides » formées à partir de $\mathbb{Q}, +, \times, \exp, \log$.
- $\bar{\cdot} \colon \mathcal{E}^{\mathrm{expr}} \to \mathbb{R}$: valeur de l'expression.
- $s \colon \mathcal{E}^{\mathrm{expr}} \to \mathbb{N}$: taille de l'expression.
- $\mathcal{E}_1^{\mathrm{expr}}$: $f \in \mathcal{E}^{\mathrm{expr}}$ telles que $|\bar{g}| \leqslant 1$ pour toute sous-expression f de e^g.

Conjecture 4. *Il existe une fonction de témoin $\varpi(s) = K\, s$, telle que*

$$\bar{f} = 0 \iff |\bar{f}| < e^{-s(f)}$$

pour toute $f \in \mathcal{E}_1^{\mathrm{expr}}$.

Cas exp-log

- $K[[z]]$: anneaux des séries sur un corps de constantes.
- $\mathcal{E}^{\mathrm{expr}}$: expressions avec $K, z, +, \times, 1/(1+\cdot), \log(1+\cdot)$ and $\exp$.
- $\bar{\cdot} : \mathcal{E}^{\mathrm{expr}} \to \mathcal{E} \subseteq K[[z]]$: valeur d'une expression en tant que série.

Conjecture 5. *Il existe une fonction de témoin $\varpi(s) = K\,s$, telle que*

$$\bar{f} = 0 \iff v(\bar{f}) > \varpi(s(f))$$

pour toute $f \in \mathcal{E}^{\mathrm{expr}}$.

Cas algébrico-différentiel

Conjecture 6. *Il existe une fonction de témoin $\varpi(s) = K\,s$, avec*

$$P(f) = 0 \iff v(P(f)) > \varpi(s(\mathrm{Problem})),$$

où $s(\mathrm{Problem})$ désigne la « taille totale des entrées ».

Transséries réelles

Plus petit corps $\mathbb{R}[[[x]]]$ contenant $\mathbb{R}[[x^{-1}]]$ $(\text{avec } x \to \infty)$, stable sous

- $\sum$ infinie
- $\exp$ et $\log$

On peut munir $\mathbb{R}[[[x]]]$ de ∂, $\int$, $\circ$, $\circ^{-1}$.

Exemples

$$\begin{aligned}
\int e^{e^x} &= \frac{e^{e^x}}{e^x} + \frac{e^{e^x}}{e^{2x}} + \frac{2\,e^{e^x}}{e^{3x}} + \frac{6\,e^{e^x}}{e^{4x}} + \frac{24\,e^{e^x}}{e^{5x}} + \cdots \\
f(x) &= \log\log\left(x\,e^{x\,e^{e^x}} + 1\right) - \exp\exp\left(\log\log x + \frac{1}{x}\right) \\
&= -\frac{\log^2 x}{2x} - \frac{\log x}{2x} + \cdots
\end{aligned}$$

[SSRVdH]

Transséries complexes

$\longrightarrow$Choix du genre $e^z \prec 1$ ou $e^z \succ 1$, mais structure finale *unique*.

$\longrightarrow$Corps $\mathbb{C}[[[z]]]$

Théorème 7. *Soient* $P \in \mathbb{R}[[[x]]]\{F\}$ *et* $f < g \in \mathbb{R}[[[x]]]$ *avec* $P(f)\,Q(f) < 0$*. Alors il existe un* $h \in (f, g)$ *avec* $P(h) = 0$*.*

$$f^7\,(f'')^8 + \mathrm{e}^{\mathrm{e}^x}\,f'\,f''\,f''' + \Gamma(\Gamma(x)) = \mathrm{e}^{\mathrm{e}^x/(\log x - 1)}$$

Théorème 8. *Soit* $P \in \mathbb{C}[[[z]]]\{F\}$ *et soit* $\mathfrak{m}$ *un transmonôme. Alors*

$$P(f) = 0 \quad (f \prec \mathfrak{m}) \tag{3}$$

admet $\deg_{\prec \mathfrak{m}} P$ *solutions dans* $\mathbb{C}[[[z]]]$*. En outre : dans une échelle qui ne contient que des nouveaux logarithmes.*

Corollaire 9. *Le corps des transséries complexes est Picard-Vessiot clos (mais pas différentiellement clos).*

Théorème 10. *(Pour coeffs. « effectifs »). Il existe un algorithme pour calculer la solution générique de* (3) *dans* $\mathbb{C}[[[z]]]$ *(elle dépend de paramètres satisfaisants des contraintes réelles algébriques). En outre : borne uniforme pour la profondeur logarithmique.*