PARDI !

François Boulier, François Lemaire, Marc Moreno Maza (LIFL)

25 Avril 2001

PARDI, PODI, PALGIE

entrée : Dans $\mathbf{k}[X]$

- deux ordres $\mathcal{R}, \overline{\mathcal{R}}$ sur X,
- un $\mathcal{R}$-ensemble caractéristique de $\mathbf{Sat}(C)$ avec $\mathbf{Sat}(C)$ premier.

sortie : un $\overline{\mathcal{R}}$-ensemble caractéristique de $\mathbf{Sat}(C)$.

algo : trois cas :

PALGIE : *Prime ALGebraic IdEal*, implanté en Aldor, C et Maple,

PODI : *Prime Ordinary Differential Ideal*, implanté en C,

PARDI : *Prime pARtial Differential Ideal*, implanté en Maple.

Exemples

- Pour $\mathcal{R} = x > y > z > s > t$ et $\overline{\mathcal{R}} = t > s > z > y > x$ on a :

$$\mathsf{palgie}(\left\{\begin{array}{l} x - t^3 \\ y - s^2 - 1 \\ z - s\,t \end{array}\right., \mathcal{R}, \overline{\mathcal{R}}) \quad = \quad \left\{\begin{array}{l} s\,t - z \\ (x\,y + x)s - z^3 \\ z^6 - x^2y^3 - 3x^2y^2 - 3x^2y - x^2 \end{array}\right.$$

- Pour

$\mathcal{R} = \cdots > v_{xx} > v_{xy} > \cdots > u_{xy} > u_{yy} > v_x > v_y > u_x > u_y > v > u$ et
$\overline{\mathcal{R}} = \cdots u_x > u_y > u > \cdots > v_{xx} > v_{xy} > v_{yy} > v_x > v_y > v$ on a :

$$\mathsf{pardi}(\left\{\begin{array}{l} v_{xx} - u_x \\ 4\,u\,v_y - (u_x\,u_y + u_x\,u_y\,u) \\ u_x^2 - 4\,u \\ u_y^2 - 2\,u \end{array}\right. \mathcal{R}, \overline{\mathcal{R}}) \quad = \quad \left\{\begin{array}{l} u - v_{yy}^2 \\ v_{xx} - 2\,v_{yy} \\ v_y\,v_{xy} - v_{yy}^3 + v_{yy} \\ v_{yy}^4 - 2\,v_{yy}^2 - 2\,v_y^2 + 1 \end{array}\right.$$

Historique :

- [Ollivier, 1990]
- [Boulier, 1996]
- [Boulier, 2000]

Applications :

- Résolution de systèmes d'EDP via des ordres *orderly*, souvent plus faciles que les ordres d'*élimination*.
- Changement (inversible) de variables.
- Comparaison de deux décompositions triangulaires (avec [Boulier et Lemaire, 2000]).
- *Paramétrique* $\longmapsto$ *Implicite*.
- Élimination des inconnues auxiliaires : $\mathbf{k}[Y < X] \longmapsto \mathbf{k}[X < Y]$ [Foursov et Moreno Maza, 2001].

Décompositions de Wu

Pour $F \subseteq \mathbf{k}[X]$ l'opération $\mathsf{charSet}(F)$ calcule un ensemble triangulaire C tel que $\mathbf{V}(F) = \mathbf{W}(C) \cup \bigcup_{p \in C} \mathbf{V}(F \cup \{\mathsf{init}(p)\})$.

$\mathsf{charSet}(F) ==$
    $P := F \setminus \{0\}$
    $Q := \emptyset$
    $C := \emptyset$
    while $(P \neq \emptyset)$ **and** $(P \cap \mathbf{k} = \emptyset)$ **repeat**
        $Q := Q \cup P \cup C$
        $C := \mathsf{basicSet}(Q)$
        $Q := Q \setminus C$
        $P := \mathsf{red}(Q, C) \setminus \{0\}$
    $P \cap \mathbf{k} \neq \emptyset$ => **return**$(\{1\})$
    return C

Si $\langle F \rangle$ est premier ...

$$\boxed{\text{Si } \langle F \rangle \text{ est premier et si } \mathbf{W}(C) \neq \emptyset \text{ alors on a } \mathbf{V}(F) = \overline{\mathbf{W}(C)}.}$$

Prenons $F = \begin{cases} u\,(u-1) \\ u\,v-1 \\ (u-1)\,w^2+w+1 \end{cases}$ On a $\langle F \rangle = \begin{cases} u-1 \\ v-1 \\ w+1 \end{cases}$

$\rightarrow$ Avec $u < v < w$ il vient $\mathsf{charSet}(F) = F$ mais $\mathbf{W}(F) = \emptyset$.

$\rightarrow$ Si on avait su $u-1 \in \langle F \rangle$ on aurait eu $\mathsf{charSet}(F) = \begin{cases} u\,(u-1) \\ u\,v-1 \\ w+1 \end{cases}$

$\rightarrow$ Si on savait pu calculer

$$\mathsf{regular?}(u, u\,(u-1)) \;=\; \{(\mathtt{true}, u-1),\ (\mathtt{false}, u)\}$$

on aurait eu $\mathsf{charSet}(F) = \langle F \rangle$.

Changement d'ordre par charSet

input: $\mathcal{R}, \overline{\mathcal{R}}$ two rankings and C a $\mathcal{R}$-char set of the prime (and not unit) ideal $\mathbf{Sat}(C)$.

output: $\overline{C}$ a $\overline{\mathcal{R}}$-char set of $\mathbf{Sat}(C)$.

$P := C$
$H := \{\mathsf{init}(p, \mathcal{R})$ **for** $p \in C\}$
$(Q, \overline{C}) := (\emptyset, \emptyset)$
while $(P \neq \emptyset)$ **repeat**
  $Q := Q \cup P \cup \overline{C}$
  $\overline{C} := \mathsf{basicSet}(Q, \overline{\mathcal{R}})$
  $Q := Q \setminus \overline{C}$
  $P := \mathsf{red}(Q, \overline{C}) \setminus \{0\}$
  $(P', H') := \mathsf{ensureRank}(P, \overline{\mathcal{R}}, C)$
  $(P, H) := (P \cup P', H \cup H')$
return $\mathsf{saturate}(\overline{C}, H)$

```
ensureRank(P, R̄, C) ==
    (P′, H′) := (∅, ∅)
    for p ∈ P repeat
        while p ≠ 0 and init(p, R̄) ∈ Sat(C) repeat
            P′ := P′ ∪ {init(p, R̄)}
            p := tail(p, R̄)
        if p ≠ 0 then H′ := H′ ∪ {init(p, R̄)}
    return (P′, H′)
```

```
saturate(C̄, H) ==
    for h ∈ H repeat
        for (bool, D) ∈ regular?(h, C̄) repeat
            if bool = true then
                C̄ := D
                break
    return C̄
```

PALGIE

input: $\mathcal{R}, \overline{\mathcal{R}}$ two rankings and C a $\mathcal{R}$-char set of the prime (and not unit) ideal $\mathbf{Sat}(C)$.

output: $\overline{C}$ a $\overline{\mathcal{R}}$-char set of $\mathbf{Sat}(C)$.

Principe : similaire au changement d'ordre par charSet.

Point-clef : la construction de $\overline{C}$ se fait au moyen d'une fonction de *pgcd modulo une chaîne régulière et modulo un idéal.*

$P := C;\ \overline{C} := \emptyset$
$H := \{\mathsf{init}(p, \mathcal{R})\ \textbf{for}\ p \in C\}$
$\textbf{while}\ (P \neq \emptyset)\ \textbf{repeat}$
$\quad p := \textbf{first}\ P;\ P := \textbf{rest}\ P$
$\quad p := \mathsf{red}(p, \overline{C})$
$\quad (p, P', H') := \mathsf{ensureRank}(p, \overline{\mathcal{R}}, C)$
$\quad (P, H) := (P \cup P', H \cup H')$
$\quad p = 0 \Longrightarrow \textbf{iterate}$
$\quad v := \mathsf{mvar}(p)$
$\quad \textbf{if}\ (\forall q \in \overline{C})\ \mathsf{mvar}(q) \neq v\ \textbf{then}$
$\qquad \overline{C} := \overline{C} \cup \{p\}$
$\quad \textbf{else}$
$\qquad (g, P', H') := \mathsf{lsr}(p, \overline{C}_v, \overline{C}_v^-, C)$
$\qquad (P, H) := (P \cup P', H \cup H')$
$\qquad \overline{C} := \overline{C} \setminus \{\overline{C}_v\} \cup \{g\}$
$\quad \overline{C} := \mathsf{saturate}(\overline{C}, H)$
$\textbf{return}\ \overline{C}$

$\mathsf{lsr}(p, \overline{C}_v, \overline{C}_v^-, C)$

- Posons $R = \mathbf{k}[X]$ et $R_-^v = \{q \in R \mid \mathsf{mvar}(q) < v\}$
- Puis $I = \mathbf{Sat}(C)$ et $I_-^v = I \cap R_-^v$.
- $\mathsf{lsr}(p, \overline{C}_v, \overline{C}_v^-, C)$ calcule le *last-subresultant* g de p et $\overline{C}_v$ dans $R_-^v[v]$ modulo I_-^v.
- Si le last-subresultant de p et $\overline{C}_v$ dans $R_-^v[v]$ appartient à R_-^v alors il est nul modulo I_-^v. Donc $g \notin \mathbf{k}$ et $\mathsf{mvar}(g) = v$.
- Toute quantité de R_-^v nulle modulo I_-^v mais non détectée comme nulle molulo $\overline{C}_v^-$ fournit une nouvelle équation à traiter.
- En fait, on effectue les calculs molulo $\overline{C}_v^-$ et on vérifie modulo I.
- Les coefficients dominants des pseudo-restes successifs fournissent de nouvelles inéquations.

Changement d'ordre par Rosenfeld–Gröbner

On connaît un ens. car. C de $I = [C] : H_C^\infty$ pour $\mathcal{R}$ (où H_C est l'ensemble des initiaux et séparants de C).

On cherche un ens. car. $\overline{C}$ de I pour $\overline{\mathcal{R}}$.

Initialement $\langle A, D, P, H \rangle = \langle \emptyset, \emptyset, C, H_C \rangle$.
À la fin du calcul $\langle A, D, P, H \rangle = \langle \overline{C}, \emptyset, \emptyset, \emptyset \rangle$.

Variables

A	ensemble d'équations déjà traitées
D	ensemble de paires critiques (Δ–polynômes) en attente
P	ensemble d'équations en attente
H	inéquations

Invariant
$I = [A \cup \Delta(D) \cup P] : H^\infty$

while $D \neq \emptyset$ **or** $P \neq \emptyset$ **repeat**

chose and remove a polynomial p from D or P

$p := \mathsf{red}(p, A)$

$(p, P', H') := \mathsf{ensureRank}(p, \overline{\mathcal{R}}, C)$

$(P, H) := (P \cup P', H \cup H')$

$p = 0 \Longrightarrow$ **iterate**

$A := A \cup \{p\}$

Add to D the new critical pairs

$H := H \cup \{\mathsf{init}(p, \overline{\mathcal{R}}), \mathsf{separant}(p, \overline{\mathcal{R}})\}$

return $\mathsf{saturate}(A, H)$

PARDI

1. Est plus simple conceptuellement que Rosenfeld–Gröbner, grâce à l'utilisation de pgcd.

2. Identifie et traite comme tels les sous–problèmes purement algébriques qui apparaissent lors du traitement différentiel,

3. Résout des exemples nouveaux. Exemple : l'équation d'Euler pour un fluide parfait.

$$\Sigma \left\{ \begin{array}{l} v^1_t + v^1 v^1_x + v^2 v^1_y + p_x = 0, \\ v^2_t + v^1 v^2_x + v^2 v^2_y + p_y = 0, \\ v^1_x + v^2_y = 0. \end{array} \right.$$

Pour un ranking orderly :

$$C \left\{ \begin{array}{l} p_{xx} = -2\, v^2_x\, v^1_y - 2\,(v^2_y)^2 - p_{yy}, \\ v^1_t = -v^2\, v^1_y - p_x + v^2_y\, v^1, \\ v^1_x = -v^2_y, \\ v^2_t = -v^1\, v^2_x - v^2\, v^2_y - p_y. \end{array} \right.$$

Pour un ranking d'élimination, l'ensemble caractéristique fait 600 Ko.

Plus de 50 dérivées différentes.

Calculs intermédiaires > 500 Mo sur MEDICIS.

while $D \neq \emptyset$ **or** $P \neq \emptyset$ **repeat**

  chose and remove a polynomial p from D or P

  $p := \mathsf{red}(p, A)$

  $(p, P', H') := \mathsf{ensureRank}(p, \overline{\mathcal{R}}, C)$

  $(P, H) := (P \cup P', H \cup H')$

  $p = 0 \Longrightarrow$ **iterate**

  $v := \mathsf{mvar}(p)$

  if $(\forall q \in A)\ \mathsf{mvar}(q) \neq v$ **then**

    $A := A \cup \{p\}$

    remove from A every q whith rank reducible by p

  else

    $(g, P', H') := \mathsf{lsr}(p, A_v, A_v^-, C)$

    $(P, H) := (P \cup P', H \cup H')$

    $A := A \setminus \{A_v\} \cup \{g\}$

  Add to D the new critical pairs

  $H := H \cup \{\mathsf{init}(p, \overline{\mathcal{R}}), \mathsf{separant}(p, \overline{\mathcal{R}})\}$

return $\mathsf{saturate}(A, H)$

PARDI

À la recherche des solitons

Nous recherchons les systèmes couplés d'équations d'évolution qui possèdent des solitons

$$
\left\{
\begin{array}{lcl}
u_t & = & u_{xxxxx} - 10uu_{xxx} + 30vv_{xxx} - 25u_x u_{xx} \\
 & & +45v_x v_{xx} + 20u^2 u_x - 30v^2 u_x - 60uvv_x \\
v_t & = & -9v_{xxxxx} + 10vu_{xxx} + 30uv_{xxx} + 35v_x u_{xx} \\
 & & +45u_x v_{xx} - 20uvu_x - 20u^2 v_x - 30v^2 v_x
\end{array}
\right.
$$

- Ces systèmes sont très rares et il n'y a pas d'algorithmes pour les fabriquer.
- On peut cependant sélectionner des systèmes-candidats dans de larges classes.
- Voici la classe où a été trouvé le système ci-dessus.

$$\left\{\begin{array}{rcl} u_t & = & a_1 u_{xxxxx} + a_2 v_{xxxxx} \\ & & +a_3 u u_{xxx} + a_4 v u_{xxx} + a_5 u v_{xxx} + a_6 v v_{xxx} \\ & & +a_7 u_x u_{xx} + a_8 v_x u_{xx} + a_9 u_x v_{xx} + a_{10} v_x v_{xx} \\ & & +a_{11} u^2 u_x + a_{12} u v u_x + a_{13} v^2 u_x \\ & & +a_{14} u^2 v_x + a_{15} u v v_x + a_{16} v^2 v_x \\ v_t & = & b_1 u_{xxxxx} + b_2 v_{xxxxx} \\ & & +b_3 u u_{xxx} + b_4 v u_{xxx} + b_5 u v_{xxx} + b_6 v v_{xxx} \\ & & +b_7 u_x u_{xx} + b_8 v_x u_{xx} + b_9 u_x v_{xx} + b_{10} v_x v_{xx} \\ & & +b_{11} u^2 u_x + b_{12} u v u_x + b_{13} v^2 u_x \\ & & +b_{14} u^2 v_x + b_{15} u v v_x + b_{16} v^2 v_x \end{array}\right.$$

- Pour étudier une classe on y cherche les systèmes couplés ayant une infinité de symétries généralisées.

⇒ Ceci fournit un système d'équations algébriques.

- Celui-ci ne peut être résolu que symboliquement.
- Cependant il compte souvent 120 inconnues et 500 équations !

⇒ Nous avons dû développer des techniques particulières pour ce type de systèmes.

- Nos calculs ont permis de réaliser la première classification complète (des systèmes généralisant ceux de Korteweg-de Vries, Sawada-Kotera et Kaup-Kupershmidt).
- Nous avons retrouvé les systèmes déjà connus et <u>découvert</u> 2 nouveaux systèmes possèdant des solitons.