Présentation du cours. Simplification.

Bruno Salvy

INRIA

I. Introduction

1. Tour d'horizon des fonctionnalités
2. Problèmes principaux pour l'utilisateur
3. Objectifs du cours
4. Bases mathématiques

II. Simplification et calcul formel

1. La simplification
2. Théorème de Richardson
3. Conséquences
4. Théorème de Risch
5. Les constantes
6. La "résolution"

III. Calcul

1. Algorithmes de base
2. Succès algorithmiques
3. Calcul vs preuve
4. Lien avec le calcul numérique
5. Plan du cours

I. Introduction

Histoire abrégée du calcul formel

1953 1er programme de dérivation.

1958 Lisp (John McCarthy au MIT).

Années 60

- Sac-1 (G. Collins) manipulations de polynômes en Fortran.
- Alpak aux Bell Labs, polynômes et fractions rationnelles.
- Formac à IBM (structure en arbre).
- 1er programme d'intégration (en Lisp).

Fin des années 60:

- Mathlab au MIT (utilisation interactive, affichage bidimensionnel).
- Reduce (en Lisp) à Stanford

Années 70

MACSYMA (en LISP, au MIT)

SCRATCHPAD chez IBM.

Années 80

MAPLE à Waterloo (Canada) premier système écrit en C.

SMP par S. Wolfram.

SCRATCHPAD II chez IBM.

Systèmes spécialisés MACAULAY, GAP, CAYLEY, PARI,...

Années 90

MATHEMATICA par S. Wolfram (premier système avec une publicité grand public).

AXIOM (version commercialisée de SCRATCHPAD II).

Problèmes principaux (pour l'utilisateur)

simplification

$$\log(\tan(x/2+\pi/4)) - \operatorname{arcsinh}(\tan x) = ?$$

robustesse, sûreté des résultats ;

efficacité, taille des expressions intermédiaires ;

le bon usage.

Objectifs du cours

Tranformer la “boîte noire” en boîte à outils.

Montrer

- des outils importants ;
- leur usage ;
- leur fonctionnement ;
- leurs limitations.

Faire comprendre ce qu’est un algorithme dans ce domaine, le genre de calcul qui s’automatise, ce qui est routinier et ce qui est coûteux.

Bases mathématiques

algèbre commutative ;

algèbre différentielle ;

combinatoire.

II. Simplification et calcul formel

La simplification

Comment reconnaître facilement l'égalité, la nullité des expressions ?

Problèmes de forme normale.

$(x+1)^{100} = \cdots$ faut-il développer les polynômes ?

$\sin(x+y) = \sin x \cos y + \sin y \cos$, $\sin p \cos q = \cdots$.
Règles de récriture ?

$\log fg = \log f + \log g$, $\exp(f+g) = \exp(f)\exp(g)$,
$\exp\log f = \log\exp f = f$. Y a-t-il d'autres relations ?

Résultat négatif

Théoreme. [Richardson] Dans la classe obtenue à partir de $\mathbb{Q}(x)$, π, $\log 2$ par les opérations $+, -, \times$ et composition avec exp, sin et $|\cdot|$, le test à zéro est indécidable.

Conséquences

Les problèmes de simplification requièrent une approche différente de `simplify` ;

Le calcul formel ne peut pas fournir de boîtes noires comme `solve` ;

Les systèmes ne peuvent pas faire “comme à la main”.

Classes d'expressions

$\mathbb{A}$ anneau

entiers ;

rationnels ;

matrices ($\mathcal{M}_{p,q}(\mathbb{A})$) ;

polynômes $\mathbb{A} \to \mathbb{A}[x] \to \mathbb{A}[x_1, \ldots, x_n]$;

fractions rationnelles $\mathbb{A}(x_1, \ldots, x_n)$;

nombres algébriques $\mathbb{A}[x]/(P(x))$;

transcendence.

On sait reconnaître 0
$\to$ on peut calculer

Théorème de Risch

F fonction élémentaire :

$\mathbb{Q}$, x, $+$, $\times$, $-$, $\div$, exp, log

Problème : $F = 0$?

Réponse : on peut écrire F comme fraction rationnelle dans $\mathbb{Q}(x, \phi_1, \ldots, \phi_n)$ où les ϕ_i sont des fonctions élémentaires, ϕ_{i+1} transcendente sur $\mathbb{Q}(x, \phi_1, \ldots, \phi_i)$.

Fondement de l'algorithme

On construit itérativement $\mathbb{K}_j = \mathbb{Q}(x, \phi_1, \ldots, \phi_j)$ avec $\phi_{j+1} = \exp(v_j)$ ou $\phi_{j+1} = \log(v_j)$, $v_j \in \mathbb{K}_j$.

Pour assurer la transcendance, on utilise le fait que :

1. $\exp(v_j)$ est transcendant sur $\mathbb{K}_j$ ssi v_j ne peut être exprimé comme $c + \sum_{\ell=1}^{j-1} r_\ell v_\ell$, c constant, $r_\ell \in \mathbb{Q}$.

2. $\log(u_j)$ est transcendant sur $\mathbb{K}_j$ ssi u_j ne peut être exprimé comme $c \prod_{\ell=1}^{j-1} u_\ell^{r_\ell}$, c constant, $r_\ell \in \mathbb{Q}$.

Les constantes

e et π sont-ils algébriquement indépendants ?

Conjecture[Schanuel] Si $z_1, \ldots, z_n \in \mathbb{C}$ sont linéairement indépendants sur $\mathbb{Q}$, alors $\mathbb{Q}(z_1, \ldots, z_n, e^{z_1}, \ldots, e^{z_n})$ a degré de transcendance au moins n.

Approches heuristiques (arithmétique d'intervalles, méthodes probabilistes).

Restriction des classes de “constantes” considérées ($\mathbb{Q}$, $\mathbb{Q}[\alpha]$, $\mathbb{Q}[\alpha](x_1, \ldots, x_n)$, ...).

La “résolution”

Fait souvent sortir d’une classe simple (celle où s’exprime l’équation) ;

$\to$ zéro devient plus difficile à reconnaître ;

inutile si l’on veut faire du numérique ;

mais utile pour accéder à des informations *globales*.

III. Calcul

Algorithmes de base

- Euclide sur $\mathbb{A}[x]$;
- Gauss sur $\mathcal{M}_{p,q}(\mathbb{A})$;
- Buchberger sur $\mathbb{A}[x_1, \ldots, x_n]$ (bases de Gröbner) ;
- “Coefficients indéterminés”.

Savoir

- que ça existe ;
- à quoi ça sert ;
- comment on s’en sert.

Penser à s’y ramener.

Succès algorithmiques

- Algèbre linéaire exacte.
- Factorisation de polynômes.
- Manipulation de nombres algébriques.
- Sommation formelle.
- Intégration formelle.
- Équations différentielles linéaires.
- Résolution de systèmes polynomiaux.

Manipulations formelles vs. calculs

Le calcul formel sert

- à calculer dans certaines classes d'expressions ;
- à expérimenter, développer l'intuition ;
- à effectuer des gros calculs parfaitement maîtrisés.

Il ne sert pas

- à deviner des solutions ;
- à prouver des résultats (en général).

Lien avec le calcul numérique

Points de comparaison :

- vitesse ;
- précision ;
- stabilité.

Interaction bénéfique pour l'étude des comportements singuliers.

Plan du cours

1. Simplification. Structures de base.
2. Équations polynomiales.
3. Formes closes.
4. Au-delà des formes closes.
5. Calcul asymptotique.