MMA 数学特論 I

Algorithms for polynomial systems: elimination & Gröbner bases

多項式系のアルゴリズム: グレブナー基底 & 消去法

April, 15th 2010: Introduction & Motivation
序文 & モチベーション

Presentation (ご紹介)

First Name: Xavier (グザヴィエ)

Last Name: Dahan (ダハン)

Position （職位）: Assistant Professor (助教)

Research interest （研究的興味): Algorithms for Algebra, Computer algebra

Short CV (履歴書)

2003: Master of Science (Maths& Computer Science) (理学修士）.

2003 - 2006: PhD student (博士程), École Polytechnique, France. (Computer science lab).

01/2007 - 10/2008: JSPS Post-Doc, 立教大学 (数理学府)

Since 11/2008: 助教.　九大数理学研究院　(G-COE "Maths-for-Industry").

About this course (この授業について)

Computational Mathematics
計算数学

Computational Algebra
計算代数

System of Polynomial equations

多項式系

→ Elimination and Gröbner bases ←

グレブナー基底と消去法

Computational Mathematics ?

TRADITIONAL MATHS
伝統的な数学 ⟶ COMPUTERS

Typical problems are: (ある典型的な問題):

1. How to represent into computers the mathematical objects ? (data structure)

   どのように数学的な対象をコンピューターで表す ？ (データ構造)

2. design fast and reliable algorithms to compute with these objects

   この数学的な対象を計算するために、早くて正確なアルゴリズムを作る。

3. solve new problems related to 1 and 2.

   １と２により、新たに起こった問題を解く.

Computational Mathematics?

Main problem: Solve equations (主な問題：方程式を解く)

Physics/Mechanic/Chemistry etc.

物理学／力学／化学、等

$\Big\downarrow$ model

Equations (linear, polynomial)

方程式 (線形の、多項式、等)

or (および)

PDE, ODE (偏／常微分程式)

Boundary conditions (境界条件) $\xleftarrow{\text{optional}}$ real life problem (現実の問題)

Physics/Mechanic/Chemistry etc.

物理学／力学／化学、等

model ↓

Equations (linear, polynomial)

方程式 (線形の、多項式、等)

or (および)

PDE, ODE (偏／常微分程式)

Boundary conditions (境界条件) ←optional— real life problem (現実の問題)

↓

Existence of solutions etc.

解の存在、など ←—— theoretical work (理論的な仕事)

Equations (linear, polynomial) 方程式 (線形の、多項式、等) or (および) PDE, ODE (偏／常微分程式)	model ⟵	Physics/Mechanic/Chemistry etc. 物理学／力学／化学、等
Boundary conditions (境界条件)	optional ⟵	real life problem (現実の問題)
↓		
Existence of solutions etc. 解の存在、など	⟵	theoretical work (理論的な仕事)
↓		
Method of resolution (discretization scheme, etc.) 解法の方法 (離散化スキーム)	additional ⟵ work	efficiency (convergence, discretization error etc.) 能率（収束、離散化誤差、等)

The two main methods (二つの主な方法)

1. **Numerical analysis**: uses approximation of a solution

(数値解析: 解近似を利用する)

- Extremely used in engineering science and in companies.
  (工学でも、会社でも、非常に使用されている)
- (i) Usually fast. 普通早い.
  (ii) Optimized implemetations for computer's architecture
  コンピュータのアーキテクチャのために最適実装である
  (iii) Approximation errors do occur. 近似誤差が実際に起こる。

2. **Symbolic Computation:** uses exact numbers or expressions...

(記号計算：正確な数字または式を計算する)

The two main methods (二つの主な方法)

2. Symbolic Computation: uses exact numbers or expressions...

- Very large expressions may appear ! Not always fast...
  非常に大きな式があるのは可能！　必ずしも早くない。。。
- No error . 誤差なし。
- The algorithms use more algebraic tools.
  このアルゴリズムが代数的なツールを利用する。

  $\Rightarrow$ new applications using algebraic techniques...
  代数的手法で、新たな応用できた。

About this course (この授業について)

Computational Mathematics
計算数学

Computational Algebra
計算代数

System of Polynomial equations
多項式系

$\longrightarrow$ Elimination and Gröbner bases $\longleftarrow$
グレブナー基底と消去法

Some applications of computer algebra (いくつか計算代数の応用)

- Cryptography: algebraic computations over finite fields (RSA, elliptic curves etc.)
  暗号学：有限体で代数的計算 (RSA、 離散対数、楕円曲線暗号、等)

- Error-correcting codes theory (Guruswami-Sudan decoding algorithm, Algebraic-geometry codes) 誤り訂正符号理論 (Guruswami-Sudan の復号アルゴリズム、代数幾何コード、等)

- Transform of the input equations into easier ones (for example, before applying a numerical scheme).
  入力方程式を簡単にする (例えば、数値表を使う前に)

- Polynomial systems computation: numerical methods are still quite inefficient.
  多項式系を計算：数値計算法がまだなかなか非効率のことがある。

Computer Algebra Systems

Not free: (無料ではない)

Mathematica: will be used in this class　(このクラスで使う予定)

Maple. Quite similar with Mathematica. Better for polynomial systems, but not available at Kyudai.
Mathematica と大体同じ。多項式について、もっと良いだが、九大では、なし。

Magma: very avanced algebraic functionalities. Efficient algorithms implemented. No graphical interface.

代数的高度機能がある。効率的アルゴリズムもある。グラフィカル・インターフェースなし。

Computer Algebra Systems

Freeware: (無料ソフト)

Risa/Asir: the Japanese computer algebra system. Good for polynomial systems. これが日本の計算代数ソフトだ。多項式のために良い。

Sage: open-source. Gather many free softwares into one.

Others: Singular – Cocoa – Mathemagix – Reduce – Axiom...

About this course (この授業について)

Computational Mathematics
計算数学

Computational Algebra
計算代数

System of polynomial equations
多項式系

$\longrightarrow$ Elimination and Gröbner bases $\longleftarrow$
グレブナー基底と消去法

Polynomial systems

Example: System of 2 equations with 3 unknowns

$$\begin{cases} f_1(x,y,z) = x^2yz - 3x^3y^1z^5 - x + y + 1 \\ f_2(x,y,z) = x^2 + y^2 + z^2 + xy + yz + xz \end{cases}$$

What for ? (i) over the **real** : robot motion planning, quantifier elimination etc. (実数体上；ロボット制御、消去法の限定)

(ii) **for the algebraists:** permits to do “experimental calulations” in commutative algebra. （代数学者にとって：可換環論の実験的計算ができる）

(iii) over small **finite fields**: important in cryptography. （有限体上：新しい暗号理論)

Polynomial systems

Learning to solve **polynomial systems** is good for:

- understanding computational mathematic in general: mix of typical algorithmic problems and non trivial mathematical background.
  一般的に計算数学をわかる　：典型的なアルゴリズムの問題と数学的背景の組み合わせだから。
- getting a concrete view of the underlying notions of algebraic geometry.
  代数幾何の基礎となることに具体的観点をもらう。
- the methods used can be generalized to the solve some differential equations (it is harder)
  この方法は、微分程式の一部を解くために、一般化できる。

Textbooks

- **Ideals, varieties and algorithms**. Cox, Little and O'Shea. Springer.
- **「グレブナー基底の計算基礎篇-計算代数入門 」**。横山 和弘、野呂　正行。

About this course (この授業について)

Computational Mathematics
計算数学

Computational Algebra
計算代数

System of polynomial equations
多項式系

→ Elimination and Gröbner bases ←

グレブナー基底と消去法

Case of linear polynomials (一次多項式)

Linear equation ⇔ polynomial of degree 1: (一次方程式 ⇔ 一次多項式).

$$\begin{cases} f_1(x,y,z) = 4x + 2y + 3z - 3 \\ f_2(x,y,z) = x + 2y + 3z - 3 \\ f_3(x,y,z) = -x - 2y - z - 3 \end{cases} \xrightarrow{matrix} \begin{pmatrix} 4 & 2 & 3 \\ 1 & 2 & 3 \\ -1 & -2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}$$

Each equation defines a plane in $\mathbb{R}^3$: (各方程式は平面を決定する)

$$H_1 := \{(a,b,c) \in \mathbb{R}^3 \mid f_1(a,b,c) = 0\}$$

$$H_2 := \{(a,b,c) \in \mathbb{R}^3 \mid f_2(a,b,c) = 0\}$$

$$H_3 := \{(a,b,c) \in \mathbb{R}^3 \mid f_3(a,b,c) = 0\}$$

Solution= $H_1 \cap H_2 \cap H_3$

Case of 1 variable (一元多項式の場合)

Polynomial with 1 unknown $\Leftrightarrow$ univariate polynomial

Solving 1 polynomial with 1 unknown: $f(X) = 0$.

Case 1: $\deg(f) = 0$ or 1 or 2 then it is easy.

Case 2: $\deg(f) = 3$ or 4, then CARDANO and FERRARI gave general formulas (XVI-th century) for the roots of f.

Case 3: $\deg(f) \geq 5$, then GALOIS showed that there is no general formula for the roots of f $\longrightarrow$ numerical approximation.

In this course, we always assume that we can solve univariate polynomials.

Case of 1 variable （一元多項式の場合)

System of 2 polynomials with 1 unknown: $\{f(X) = 0,\ g(X) = 0\}$.

Recall: $\alpha \in \mathbb{C}$ is a root of f and $g \Leftrightarrow \alpha$ is a root of $\boxed{\mathsf{gcd}(f, g)}$

Solving with 1 variable $\Leftrightarrow$ computing gcd.

Review: computing gcd with the Extended Euclidean Algorithm (EEA).

Ideals of commutative rings

Definition 1 *Let A be a commutative ring (example: $A = k[X_1, \ldots, X_n]$). A subset $I \subset A$ is called an* ideal *of A if the following three properties are verified:*

1. $0 \in I$
2. $\forall f, g \in I, \quad f + g \in I.$
3. $\forall f \in I, \quad \forall h \in A, \quad fh \in I.$

Example: Finitely generated ideals. The subset $\langle f_1, \ldots, f_s \rangle$ of $k[X_1, \ldots, X_n]$:

$$\langle f_1, \ldots, f_s \rangle := \left\{ \sum_{i=1}^{s} f_i g_i, \quad g_i \in [X_1, \ldots, X_n] \right\},$$

is an ideal of $k[X_1, \ldots, X_n]$. Its basis $f_1, \ldots, f_s$ is *finite* (it s a *finitely generated* ideal)

Polynomials in 1 variable

Definition 2 *Let* $P = a_0X^n + a_1X^{n-1} + \cdots + a_{n-1}X + a_n \in k[X]$ *be a polynomial in 1 variable over a field* k*, with* $a_0 \neq 0$ *(*$\Leftrightarrow \deg(P) = n$*). Define:*

The leading term *of* P*:* $\text{LT}(P) := a_0X^n$*.*

The leading coefficient *of* P*:* $\text{LC}(P) := a_0$*.*

The leading monomial *of* P*:* $\text{LM}(P) = X^n$*.*

P *is* monic *if* $\text{LC}(P) = 1$*.*

Example: If $P, Q \in k[X]$ then $\deg(P) \leq \deg(Q) \Leftrightarrow \text{LT}(P)|\text{LT}(Q)$ ($\text{LT}(P)$ "divides" $\text{LT}(Q)$).

$P = 3X^2+2X+1$, $Q = 2X^3+3 \Rightarrow \text{LT}(P) = 3X^2$, $\text{LT}(Q) = 2X^3$, $\text{LT}(P)|\text{LT}(Q)$, and $\frac{\text{LT}(Q)}{\text{LT}(P)} = \frac{2}{3}X$.

Euclidean division

Proposition 1 *Let* $a, b \in k[X]$*, with* $b \neq 0$ *and assume that* $\deg(a) \geq \deg(b)$*. There exists 2 polynomials* $q, r \in k[X]$*, such that*

$$a = bq + r, \qquad \textit{with either } r = 0 \textit{ or either } \deg(r) < \deg(b).$$

PROOF:Algorithmic. □

Remark: The Euclidean division holds if a and b are in any polynomial ring $A[X]$, where A is an *integral domain* (a commutative ring where for all elements x, y, holds: $xy = 0 \Rightarrow x = 0$ or $y = 0$) and if $\text{LC}(b)$ is *invertible* in A (is a *unit* in A).

The division algorithm in 1 variable

\# Inputs: $a, b \in k[X]$, $b \neq 0$, $\deg(a) \geq \deg(b)$

\# Outputs: (q, r) such that $a = bq + r$, with $r = 0$ or $\deg(r) < \deg(b)$

1: $r \leftarrow a$

2: $q \leftarrow 0$

3: **while** $(r \neq 0$ **and** $\mathrm{LT}(b) | \mathrm{LT}(r))$ **do** // *equivalent to* $\deg(b) \leq \deg(r)$

4: $\quad s \leftarrow \frac{\mathrm{LT}(r)}{\mathrm{LT}(b)}$

5: $\quad q \leftarrow q + s$

6: $\quad r \leftarrow r - sb$

7: **end while**

8: **return** (q, r)

Remark: usually, the symbol // after a line in an algorithm denotes just a comment.

Some well-known consequences

Corollary 1 *A polynomial over a field k of degree m has at most m roots in k.*

Corollary 2 *Let k be a field. For each ideal I of $k[X]$, there exists a polynomial f such that $I = \langle f \rangle$. If g is another polynomial such that $\langle g \rangle = I$, then $g = \lambda f$, for a $\lambda \in k$.*

In particular, there exists a unique monic *generator.*

Remark: Such generators have minimal degree among the non-zero polynomials in I.
Example: Let M be a square matrix with entries in k. The ideal I_M in $k[X]$ of the polynomials P such that $P(M)$ is the null matrix, contains a non-zero polynomial, (the characteristic polynomial for example, so $\{0\} \subsetneq I_M$). The generator of this ideal that is *monic*, is called the *minimal polynomial* of M.

Finding a generator of ideals in $k[X]$: GCD (1/3)

Problem: Given an ideal $I \subset k[X]$ generated by polynomials $f_1, \ldots, f_s$, how to find a generator g of I ?

Definition 3 *A GCD of $f, h \in k[X]$ is a polynomial g such that:*

(i) $g|f$ and $g|h$

(ii) if a polynomial $p|f$ and $p|g$, then $p|g$ as well.

Remark: In $k[X]$, a gcd of f and h is such that the ideals $\langle g \rangle = \langle f, h \rangle$. Hence, by Corollary 2, given two GCDs g_1 and g_2, there exists $\lambda \in k$ such that: $g_1 = \lambda g_2$. In particular, there exists one monic GCD.

The definition above with "divisibility" conditions, makes sense in more general rings than $\mathbb{Z}$ or $k[X]$, called *unique factorization domains* (*UFD* for short).

Proposition 2 *A* gcd *in $k[X]$ always exists and we can compute it.*

Euclidean algorithm in $k[X]$: finding GCD (2/3)

\# Inputs: $f, h \in k[X]$ with $f \neq 0$ and $\deg(f) \geq \deg(h)$

\# Outputs: a GCD of f and h

```
a ← f
b ← h
while (b ≠ 0) do
    (q, r) ← EuclideanDivision(a, b)        //so that: a = bq + r
    a ← b
    b ← r
end while
return a
```

Again, // means the beginning of a comment.

Finding a generator of ideals in $k[X]$: GCD (3/3)

Problem: Given an ideal $I \subset k[X]$ generated by polynomials $f_1, \ldots, f_s$, how to find a generator g of I ?

Property: $\mathsf{gcd}(f_1, \mathsf{gcd}(f_2, f_3)) = \mathsf{gcd}(\mathsf{gcd}(f_1, f_2), f_3)$

This permits to define $\mathsf{gcd}(f_1, f_2, f_3)$, and more generally *multiple GCDs* denoted $\mathsf{gcd}(f_1, \ldots, f_s)$.

Remark: As usual GCDs, multiple GCDs are *not* unique. Also, there is one *monic* multiple GCD.

Consequence: Solve the ideal membership problem in one variable.

1. Compute recursively a multiple GCD g of $f_1, \ldots, f_s$.
2. Compute the Euclidean division of f by g: $f = qg + r$.
3. $f \in \langle f_1, \ldots, f_s \rangle \Leftrightarrow r = 0$.