Gröbner Bases

Martin R. Albrecht

DTU Crypto Group

22 October 2013

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

Why not …

This course involves implementing algorithms in multivariate polynomial rings for which we will be using the Sage mathematics software.

You already know Maple, so why not use it?

1. I do not know Maple so I could not help you with problems you might encounter.
2. Maple is an expensive piece of software that you might not have access to outside of this university, cutting you off the tool you know. Sage does not have this problem.
3. Maple is a closed source piece of software which means you cannot study, learn from and improve it. This goes against fundamental principles of reproducible science.

Blurb

Sage open-source mathematical software system

“Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.”

Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common Python-based interface.

First release 2005
> 300 releases
> 180 developers
> 200 papers cite Sage
> 100,000 web visitors/month

Latest version 5.11 released 2013-08-13
Shell, webbrowser (GUI), library
~ 100 components
> 2100 subscribers [sage-support]
> 6, 500 downloads/month

Python & Cython

python & Cython

Sage does **not** come with yet-another ad-hoc mathematical programming language, it uses **Python** instead.

- one of the most widely used programming languages (Google, IML, YouTube, NASA),
- easy for you to define your own data types and methods on it (bitstreams, ciphers, rings, whatever),
- very clean language that results in easy to read code,
- a **huge number of libraries**: statistics, networking, databases, bioinformatic, physics, video games, 3d graphics, numerical computation (scipy), and serious “pure” mathematics (via Sage)
- easy to use existing C/C++ libraries from Python (via **Cython**)

How to use Sage I

Use Sage via

- as a **command line** programme,
- as a **webapp** hosted on your local computer and
- as a webapp **on the Internet**

How to use Sage II

To install Sage on your computer head to

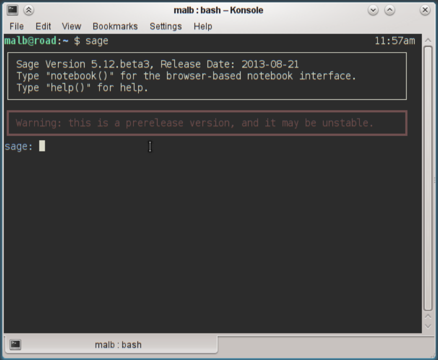
`http://sagemath.org`

and download the right version for your operating system
(Mac OS X, Windows, Linux …)

You can then either run Sage on the **command line** or start a **local webapp**.

How to use Sage III

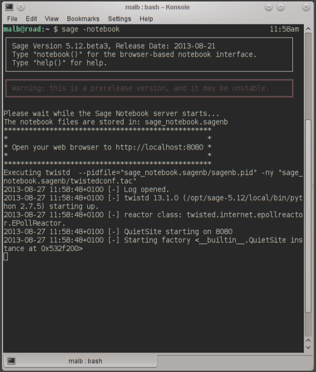
Calling sage



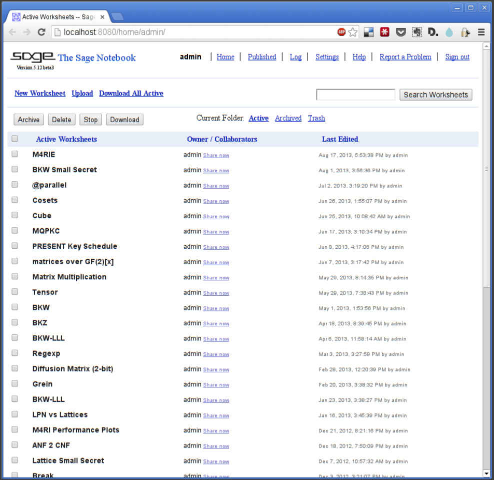
```
malb@road:~ $ sage                                                    11:57am
┌────────────────────────────────────────────────────────────────────┐
│ Sage Version 5.12.beta3, Release Date: 2013-08-21                  │
│ Type "notebook()" for the browser-based notebook interface.        │
│ Type "help()" for help.                                            │
└────────────────────────────────────────────────────────────────────┘
┌────────────────────────────────────────────────────────────────────┐
│ Warning: this is a prerelease version, and it may be unstable.     │
└────────────────────────────────────────────────────────────────────┘
sage:
```

How to use Sage IV

Calling `sage -notebook`



```
malb@road:~ $ sage -notebook                                        11:58am
  Sage Version 5.12.beta3, Release Date: 2013-08-21
  Type "notebook()" for the browser-based notebook interface.
  Type "help()" for help.

  Warning: this is a prerelease version, and it may be unstable.

Please wait while the Sage Notebook server starts...
The notebook files are stored in: sage_notebook.sagenb
**************************************************
*                                                *
* Open your web browser to http://localhost:8080 *
*                                                *
**************************************************
Executing twistd  --pidfile="sage_notebook.sagenb/sagenb.pid" -ny "sage_
notebook.sagenb/twistedconf.tac"
2013-08-27 11:58:48+0100 [-] Log opened.
2013-08-27 11:58:48+0100 [-] twistd 13.1.0 (/opt/sage-5.12/local/bin/pyt
hon 2.7.5) starting up.
2013-08-27 11:58:48+0100 [-] reactor class: twisted.internet.epollreacto
r.EPollReactor.
2013-08-27 11:58:48+0100 [-] QuietSite starting on 8080
2013-08-27 11:58:48+0100 [-] Starting factory <__builtin__.QuietSite ins
tance at 0x532f200>
```

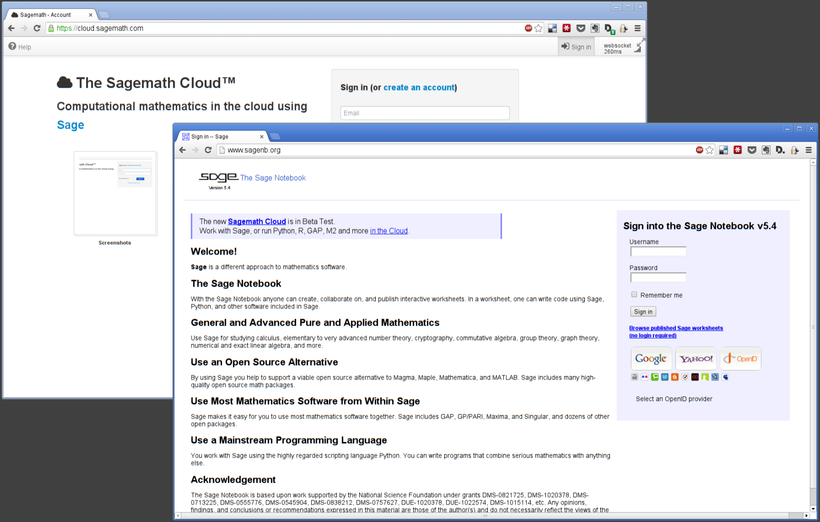


□	Active Worksheets	Owner / Collaborators	Last Edited
□	M4RIE	admin Share now	Aug 17, 2013, 5:53:38 PM by admin
□	BKW Small Secret	admin Share now	Aug 1, 2013, 3:56:36 PM by admin
□	@parallel	admin Share now	Jul 2, 2013, 3:19:20 PM by admin
□	Cosets	admin Share now	Jun 26, 2013, 1:55:07 PM by admin
□	Cube	admin Share now	Jun 25, 2013, 10:08:42 AM by admin
□	MQPKC	admin Share now	Jun 17, 2013, 3:10:34 PM by admin
□	PRESENT Key Schedule	admin Share now	Jun 8, 2013, 4:17:06 PM by admin
□	matrices over GF(2)[x]	admin Share now	Jun 7, 2013, 3:17:42 PM by admin
□	Matrix Multiplication	admin Share now	May 29, 2013, 8:14:35 PM by admin
□	Tensor	admin Share now	May 29, 2013, 7:38:43 PM by admin
□	BKW	admin Share now	May 1, 2013, 1:53:56 PM by admin
□	BKZ	admin Share now	Apr 18, 2013, 8:39:45 PM by admin
□	BKW-LLL	admin Share now	Apr 6, 2013, 11:58:14 AM by admin
□	Regexp	admin Share now	Mar 3, 2013, 3:27:59 PM by admin
□	Diffusion Matrix (2-bit)	admin Share now	Feb 28, 2013, 12:20:39 PM by admin
□	Grein	admin Share now	Feb 20, 2013, 3:38:32 PM by admin
□	BKW-LLL	admin Share now	Jan 23, 2013, 3:38:27 PM by admin
□	LPN vs Lattices	admin Share now	Jan 16, 2013, 3:45:39 PM by admin
□	M4RI Performance Plots	admin Share now	Dec 21, 2012, 8:21:16 PM by admin
□	ANF 2 CNF	admin Share now	Dec 18, 2012, 7:50:09 PM by admin
□	Lattice Small Secret	admin Share now	Dec 7, 2012, 10:57:32 AM by admin
□	Break	admin Share now	Dec 3, 2012, 3:21:07 PM by admin

How to use Sage V

You can also use Sage online:

`http://www.sagenb.org/` and `https://cloud.sagemath.com/`



The Sagemath Cloud™

Computational mathematics in the cloud using Sage

Sign in (or create an account)

Email

The Sage Notebook
Version 5.4

The new Sagemath Cloud is in Beta Test.
Work with Sage, or run Python, R, GAP, M2 and more in the Cloud.

Welcome!

Sage is a different approach to mathematics software.

The Sage Notebook

With the Sage Notebook anyone can create, collaborate on, and publish interactive worksheets. In a worksheet, one can write code using Sage, Python, and other software included in Sage.

General and Advanced Pure and Applied Mathematics

Use Sage for studying calculus, elementary to very advanced number theory, cryptography, commutative algebra, group theory, graph theory, numerical and exact linear algebra, and more.

Use an Open Source Alternative

By using Sage you help to support a viable open source alternative to Magma, Maple, Mathematica, and MATLAB. Sage includes many high-quality open source math packages.

Use Most Mathematics Software from Within Sage

Sage makes it easy for you to use most mathematics software together. Sage includes GAP, GP/PARI, Maxima, and Singular, and dozens of other open packages.

Use a Mainstream Programming Language

You work with Sage using the highly regarded scripting language Python. You can write programs that combine serious mathematics with anything else.

Acknowledgement

The Sage Notebook is based upon work supported by the National Science Foundation under grants DMS-0821725, DMS-1020378, DMS-0713225, DMS-0555776, DMS-0545904, DMS-0838212, DMS-0757627, DUE-1020378, DUE-1022574, DMS-1015114, etc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the

Sign into the Sage Notebook v5.4

Username

Password

□ Remember me

Sign in

Browse published Sage worksheets (no login required)

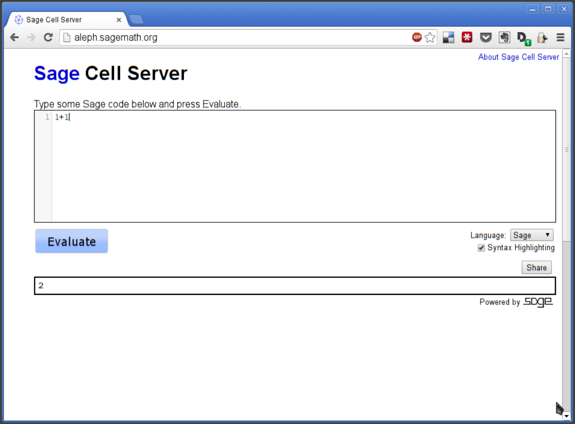
Google YAHOO! OpenID

Select an OpenID provider

How to use Sage VI

If you just want to do a quick calculation (and share it with others), try

`http://aleph.sagemath.org`



Sage Cell Server

aleph.sagemath.org

About Sage Cell Server

Sage Cell Server

Type some Sage code below and press Evaluate.

```
1+1
```

```
2
```

Powered by sage

How to use Sage VII

You can access a Sage 5.10 installed on IMM servers using your account credentials.

```
malb@road:~ $ ssh maroa@sunray4.imm.dtu.dk
Password:
Last login: Mon Oct  7 17:20:54 2013 from 10.16.166.157
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005

sunray4:~ $ gridterm
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LC_ALL = (unset),
        LC_CTYPE = "iso_8859_1",
        LANG = "en_US.utf8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
Last login: Mon Oct  7 17:30:24 CEST 2013 from sunray4.imm.dtu.dk on pts/18

grid03:~ $ sage
+--------------------------------------------------------------------+
| Sage Version 5.10, Release Date: 2013-06-17                        |
| Type "notebook()" for the browser-based notebook interface.        |
| Type "help()" for help.                                            |
+--------------------------------------------------------------------+
sage: 1+1
2
```

Intro to Programming in Sage

Go to Sage Worksheet.

See also

`http://sagemath.org/doc/thematic_tutorials/tutorial-programming-python.html`

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

Notation I

- By $\mathbb{F}$ we denote any field field in which we can effectively compute, e.g. $\mathbb{Q}$, or
- $\mathbb{F}_q$ is the finite field of order q where q is a prime power, e.g. $\mathbb{F}_{2^3}$ or $\mathbb{F}_7$
- $P = \mathbb{F}[x_1, \ldots, x_n]$ is the polynomial ring in $x_1, \ldots, x_n$ over $\mathbb{F}$.
- We assume n is finite in this lecture.

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(7))
sage: f = x*y + 3*y*z -3
sage: g = x^2 - 2*y^2
sage: f * g
x^3*y - 2*x*y^3 + 3*x^2*y*z + y^3*z - 3*x^2 - y^2
sage: f + g
x^2 + x*y - 2*y^2 + 3*y*z - 3
```

Notation II

Let $f = 3xy + 2$, we call

monomials $3\mathbf{xy} + 2 \cdot \mathbf{1}$

coefficients $\mathbf{3}xy + \mathbf{2} \cdot 1$

terms $\mathbf{3xy} + \mathbf{2}$

This follows [Cox et al., 2007] and is consistent with the usage in Sage.

```
sage: P.<x,y> = PolynomialRing(FiniteField(7))
sage: f = 2*x*y +3
sage: f.monomials()
[x*y, 1]
sage: f.coefficients()
[2, 3]
```

But [Becker and Weispfenning, 1991] swap the notion of term and monomial.

Monomial Orderings I

We will have to make decisions based on the "leading monomial" of a polynomial, which begs the question what that is.

- In $\mathbb{F}[x]$ this question is straight-forward: $x^i > x^j$ iff $i > j$.
- But what about $\mathbb{F}[x, y]$ is x^2 or xy bigger?
- Hence, we adjoin a **monomial ordering** to our ring $\mathbb{F}[x_1, \dots, x_n]$

Monomial Orderings II

Definition (Lexicographical)

Let

$$m_1 = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \text{ with } \alpha = (\alpha_1, \ldots, \alpha_n)$$

and

$$m_2 = x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n} \text{ with } \beta = (\beta_1, \ldots, \beta_n).$$

We say $m_1 >_{\text{lex}} m_2$ if, in the vector difference $\alpha - \beta \in \mathbb{Z}^n$, the leftmost nozero entry is positive.

Like in the dictionary: first you check x, then y ....

$$(0, 5, 3) - (0, 2, 9) = (0, 3, -6) \Rightarrow y^5 z^3 >_{\text{lex}} y^2 z^9 \in \mathbb{F}[x, y, z]$$

Monomial Orderings III

Definition (Degree Lexicographical)

Let

$$m_1 = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \text{ with } \alpha = (\alpha_1, \dots, \alpha_n)$$

and

$$m_2 = x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n} \text{ with } \beta = (\beta_1, \dots, \beta_n).$$

We say $m_1 >_{\text{deglex}} m_2$ if,

$$|\alpha| = \sum_{i=1}^{n} \alpha_i > |\beta| = \sum_{i=1}^{n} \beta_i, \text{ or } |\alpha| = |\beta| \text{ and } \alpha >_{\text{lex}} \beta.$$

We first check for degrees and only if those are equal we compare lexicographically.

$$y^2 z^9 >_{\text{deglex}} y^5 z^3 \in \mathbb{F}[x, y, z]$$

Monomial Orderings IV

Definition (Degree Reverse Lexicographical)

Let

$$m_1 = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \text{ with } \alpha = (\alpha_1, \ldots, \alpha_n)$$

and

$$m_2 = x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n} \text{ with } \beta = (\beta_1, \ldots, \beta_n).$$

We say $m_1 >_{\text{degrevlex}} m_2$ if,

$$|\alpha| = \sum_{i=1}^{n} \alpha_i > |\beta| = \sum_{i=1}^{n} \beta_i, \text{ or}$$

$|\alpha| = |\beta|$ and the rightmost nonzero entry of $\alpha - \beta \in \mathbb{Z}^n$ is negative.

$>_{\text{deglex}}$ is not $>_{\text{degrevlex}}$ with reversed variables:

$$x^2yz^2 > \not>_{\text{degrevlex}} xy^3z \text{ vs. } x^2yz^2 >_{\text{deglex}} xy^3z.$$

Monomial Orderings V

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(7), order="lex")
sage: x^2*y*z^2 > x*y^3*z
True

sage: P.<x,y,z> = PolynomialRing(FiniteField(7), order="deglex")
sage: x^2*y*z^2 > x*y^3*z
True

sage: P.<x,y,z> = PolynomialRing(FiniteField(7), order="degrevlex")
sage: x^2*y*z^2 > x*y^3*z
False
```

Monomial Orderings VI

- We write $>$ for any monomial ordering (such as $>_{\text{lex}}$) when it is clear from context which we mean
- $\mathrm{LM}(f)$: We call leading monomial of f that monomial in f which is largest wrt to $>$.
- $\mathrm{LC}(f)$, $\mathrm{LT}(f)$: We define the leading coefficient and leading term of f analogously.
- We extend the notion of $>$ to polynomials f, g in the natural way by comparing $\mathrm{LM}(f)$ and $\mathrm{LM}(g)$ first and move on to smaller monomials if they are equal.
- $\deg(f)$: We call the degree of f the maximal degree of any monomial in f. In particular, $\deg(f)$ can be $\neq \deg(\mathrm{LM}(f))$.

Exercises

- Construct $\mathbb{F}_7[x, y, z]$ with term order **deglex** in Sage
- Reorder the monomials of $f = 4x^5yz^4 + 2x^2y^8 - xy^4 + xyz^3 + 4yz + 3x$ wrt to the
  - **lexicographical**,
  - **degree lexicographical** and
  - **degree reverse lexicographical**

  ordering. Check your result with Sage.

Polynomial Division I

We will need to divide polynomials (with remainder) later:

Division of monomials is easy: $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ is divisible by $x_1^{\beta_1} \cdots x_n^{\beta_n}$ if $\alpha - \beta \geq (0, \ldots, 0)$ where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$.

But what about polynomials?

Polynomial Division II

```
1  begin
2      $a_1, \dots, a_s \leftarrow 0, \dots, 0$;  $r \leftarrow 0$;
3      while $p \neq 0$ do
4          division ← False;
5          for $1 \leq i \leq s$ do
6              if $LM(f_i)$ divides $LM(p)$ then
7                  $a_i \leftarrow a_i + \mathrm{LT}(p)/\mathrm{LT}(f_i)$;
8                  $p \leftarrow p - \mathrm{LT}(p)/\mathrm{LT}(f_i) \cdot f_i$;
9                  division ← True;
10                 break;
11         if $division = False$ then
12             $r \leftarrow r + \mathrm{LT}(p)$;
13             $p \leftarrow p - \mathrm{LT}(p)$;
```

Algorithm 1: Polynomial Division

Example

$p = 3x^2y + 2y^2 + x + 1$,
$f_1 = 2x^2 + 1$,
$f_2 = y + 1 \in \mathbb{F}_7[x, y]$ with deglex

1. $x^2 \mid x^2y$:
   $a_1 = -2y, p = 2y^2 + x + 2y + 1$
2. break and jump to line 3.
3. $y \mid y^2$:
   $a_2 = 2y, p = x + 1$
4. break and jump to line 3.
5. Neither y nor x^2 divide x:
   division = False
6. $r = x, p = 1$
7. Neither y nor x^2 divide x:
   division = False
8. $r = x + 1, p = 0$

Polynomial Division III

```
sage: P.<x,y> = PolynomialRing(FiniteField(7),order="deglex")
sage: f = 3*x^2*y + 2*y^2 + x + 1
sage: f1 = 2*x^2 + 1
sage: f2 = y + 1
sage: f == (-2*y)*f1 + (2*y)*f2 + (x+1)
True
```

Exercises

- Let $f = x^7y^2 + x^3y^2 - y + 1$, $f_1 = xy^2 - x$ and $f_2 = x - y^3 \in \mathbb{F}_7[x,y]$ with **deglex**.
  Divide f by (f_1, f_2). Then use **lex** and divde again.
- Let $f = xy^2 - x$, $f_1 = xy + 1$ and $f_2 = y^2 - 1 \in \mathbb{F}_7[x,y]$ with $>_{\text{lex}}$.
  Divide f by (f_1, f_2) and by (f_2, f_1) and compare the remainders.
- Let $f = xy^2z^2 + xy - yz$, $f_1 = x - y^2$, $f_2 = y - z^3$ and $f_3 = z^2 - 1 \in \mathbb{F}_7[x,y,z]$ with **deglex**.
  Divide f by (f_1, f_2, f_3), (f_2, f_3, f_1) and (f_3, f_1, f_2) and compare the remainders.

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

Definition

Let $\mathcal{I}$ be an ideal $\subset P$.

That is,

- $0 \in \mathcal{I}$,
- If $f, g \in \mathcal{I}$, then $f + g \in \mathcal{I}$, and
- if $f \in P, g \in \mathcal{I}$, then $f \cdot g \in \mathcal{I}$.

We denote by $\langle f_1, \dots, f_s \rangle$ is the ideal spanned by $f_1, \dots, f_s$, i.e. all

$$f = a_1 f_1 + \cdots + a_s f_s.$$

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(127),order='deglex')
sage: I = ideal(x*y + z, y^3 + 1, z^2 - x*5 - 1)
sage: (x*y + z) + (y^3 + 1) in I
True
sage: x*z*(z^2 - x*5 - 1) in I
True
```

Varieties I

Definition

Let $\mathbb{F}$ be a field, and let $f_1, \ldots, f_s$ be polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. Then we set

$$V(f_1, \ldots, f_s) = \{(a_1, \ldots, a_n) \in \mathbb{F}^n : f_i(a_1, \ldots, a_n) = 0 \text{ for all } 1 \leq i \leq s\}.$$

We call $V(f_1, \ldots, f_s)$ the affine variety defined by $f_1, \ldots, f_s$.

It's the set of all solutions to $f_i(x_1, \ldots, x_n) = 0$ for all $1 \leq i \leq s$.

For example, $f_1 = x^2 + y^2 - 1 \in \mathbb{R}[x, y]$ is the unit circle.

Varieties II

Lemma

$V(\mathcal{I})$ is an affine variety. In particular, if $\mathcal{I} = \langle f_1, \ldots, f_m \rangle$, then $V(\mathcal{I}) = V(f_1, \ldots, f_m)$.

Corollary

If $f_1, \ldots, f_s$ and $g_1, \ldots, g_t$ are bases of the same ideal in $\mathbb{F}[x_1, \ldots, x_n]$, so that $\langle f_1, \ldots, f_s \rangle = \langle g_1, \ldots, g_t \rangle$, then we have $V(f_1, \ldots, f_s) = V(g_1, \ldots, g_t)$.

…so finding a good basis $g_1, \ldots, g_t$ for $\langle f_1, \ldots, f_s \rangle$ such that finding $V(f_1, \ldots, f_s) = (g_1, \ldots, g_t)$ is easy would allow to solve non-linear systems of equations.

Varieties III

Example

Consider $\mathcal{I} = \langle x^2 + y^2 - 1, xy \rangle \in \mathbb{F}_7[x, y]$. The set of all points satisfying all polynomials in $\mathcal{I}$ are $(0, 1), (0, -1), (1, 0), (-1, 0)$.

The same in Sage:

```
sage: P.<x,y> = PolynomialRing(FiniteField(7))
sage: I = Ideal(x^2 + y^2 - 1, x*y)
sage: I.variety()
[{y: 0, x: 1}, {y: 0, x: 6}, {y: 1, x: 0}, {y: 6, x: 0}]
```

Exercises

1. Construct the Ideal $\mathcal{I} = \langle xy + 1, y^2 - 1 \rangle \in \mathbb{F}_7[x, y]$ in Sage with monomial ordering **deglex**.
2. Show that $x + y \in \mathcal{I}$.
3. Is $\{1\}$ an ideal? Is $\{x + 1, y - 2\}$ an ideal?
4. Show that $\langle 1 \rangle$ and $\langle x + 1, x - 1, y - 2 \rangle$ are the same.
5. What is $V(\{1\})$?
6. What is $V(\langle x + 1, y - 1 \rangle)$?
7. What is $V(x + y, x + 1)$?

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

Definition I

Definition (Gröbner Basis)

Let $\mathcal{I}$ be an ideal in $\mathbb{F}[x_1, \ldots, x_n]$ and fix a monomial ordering. A finite subset

$$G = \{g_1, \ldots, g_m\} \subset \mathcal{I}$$

is said to be a **Gröbner basis** of $\mathcal{I}$ if for any $f \in \mathcal{I}$ there exists $g_i \in G$ such that

$$\mathrm{LM}\left(g_i\right) \mid \mathrm{LM}\left(f\right).$$

Definition II

Gröbner bases generalise greatest common divisors over $\mathbb{F}[x]$: the GCD divides all its multiples.

```
sage: R.<x> = PolynomialRing(FiniteField(7))
sage: f = x^2 + 6
sage: I = Ideal(map(P, [R.random_element() * f for _ in range(5)]))
sage: I.groebner_basis()
[x^2 - 1]
```

Definition III

…and row echelon forms over $\mathbb{F}^n$: if $y + \mathit{tail}$ is in the vector space, the row echelon form has an element $y + \mathit{tail}'$.

```
sage: F = Sequence([-3*y, -2*x - y - 3*z + 2, x + y + 2*z - 1])
sage: F.groebner_basis()
[x - 1, y, z]
sage: A,v = F.coefficient_matrix()
sage: A.echelonize()
sage: (A*v).T
[x - 1     y     z]
```

Definition IV

You can use Sage to check if a list of polynomials is a Gröbner basis:

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(7),order="degrevlex")
sage: I = Ideal([x*y^2 - z, 2*x^2*y - 2, x*y - z + 1])
sage: I.basis_is_groebner()
False

sage: I.groebner_basis()
[y^2 - y - z + 1, y*z - y - z, z^2 - y - 2*z + 1, x - y + 1]
```

Reduced Gröbner Bases I

Definition (Reduced Gröbner Basis)

A **reduced Gröbner basis** for a polynomial ideal $\mathcal{I}$ is a Gröbner basis G such that:

$\mathrm{LC}(f) = 1$ for all $f \in G$;

$\forall f \in G, \nexists\ m \in$ the set of monomials of f such that m is divisible by any $\mathrm{LM}(g) \in G \setminus \{f\}$.

Reduced Gröbner bases generalise reduced row echelon forms:

$$\begin{pmatrix} 2 & 4 & 5 & 6 & 3 & 5 & 2 & 5 \\ & 5 & 2 & 2 & 2 & 1 & 0 & 2 \\ & & 6 & 5 & 6 & 2 & 1 & 6 \\ & & & 2 & 3 & 6 & 4 & 0 \\ & & & & 3 & 0 & 3 & 1 \\ & & & & & & 1 & 6 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ & 1 & 0 & 0 & 0 & 5 & 0 & 6 \\ & & 1 & 0 & 0 & 6 & 0 & 1 \\ & & & 1 & 0 & 3 & 0 & 0 \\ & & & & 1 & 0 & 0 & 6 \\ & & & & & & 1 & 6 \end{pmatrix}$$

Reduced Gröbner Bases II

By default, Sage will always computes the reduced Gröbner basis when computing a Gröbner basis. If a Gröbner basis was obtained by other means, the function

```
MPolynomialIdeal.interreduced_basis()
```

can be used to compute the reduced Gröbner basis.

```
sage: rgb = Ideal(gb).interreduced_basis()
```

Applications I

Representation Reduced Gröbner bases are a unique representation of an ideal with respect to a monomial ordering. So to check that two ideals are the same you only need to compute their reduced Gröbner bases.

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(7))
sage: I = Ideal(P.random_element() for _ in range(5))
sage: J = Ideal(P.random_element() for _ in range(5))
sage: I = Ideal(f - f((2,3,1)) for f in I.gens())
sage: J = Ideal(f - f((2,3,1)) for f in J.gens())
sage: I.groebner_basis() == J.groebner_basis()
True
sage: I == J
True
sage: I.gens() == J.gens()
False
sage: I.groebner_basis()
[x - 2, y - 3, z - 1]
```

Applications II

Membership The remainder r of the division of any $f \in P$ by G is unique. Hence, we can check whether $f \in \langle f_1, \dots, f_s \rangle$ by computing a Gröbner basis $G = (g_1, \dots, g_t)$ for $\langle f_1, \dots, f_s \rangle$ and check if $\overline{f}^G = 0$.

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(7))
sage: I = Ideal(P.random_element() for _ in range(5))
sage: I = Ideal(f - f((2,3,1)) for f in I.gens())

sage: f = P.random_element()
sage: f in I
False

sage: f = f - f((2,3,1))
sage: f in I
True

sage: f.reduce(I.gens())
2*z^2 + y - 3*z - 2
sage: f.reduce(I.groebner_basis())
0
```

Applications III

Solving Gröbner bases with respect to the **lex** ordering allow for finding $V(\mathcal{I})$ easily. We can “read off” the solution similarly to reading the solution from a row echelon form of a matrix.

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(7), order="lex")
sage: I = Ideal(P.random_element() for _ in range(5))
sage: I = Ideal(f - f((2,3,1)) for f in I.gens())
sage: I.groebner_basis()
[x - 2, y - 3, z - 1]
```

Exercises

- Let $f_1, \dots, f_m$ be polynomials in $\mathbb{F}[x_1, \dots, x_n]$. Assume that $V(f_1, \dots, f_m) = \{(s_1, \dots, s_n)\}$. Show that the reduced Gröbner basis of $\mathcal{I} = \langle f_1, \dots, f_m \rangle$ is $[x_1 - s_1, \dots, x_n - s_n]$.
- If we use **degrevlex** order with $x > y > z$, is

  $$\{x^4y^2 - z^5, x^3y^3 - 1, x^2y^4 - 2z\}$$

  a Gröbner basis or the ideal generated by these polynomials? Why or why not?
- $\{x + 1, y - 2, xy - 2x + y - 2\}$ is a Gröbner basis wrt the **degrevlex** ordering in $\mathbb{F}_7[x, y]$. Is it a reduced Gröbner basis? Is $\{2x + 2, 3y + 1\}$? If not, why not?

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

S-Polynomials I

Let $G = \{f_1, \ldots, f_s\} \subset \mathbb{F}[x_1, \ldots, x_n]$ and $\mathcal{I} = \langle G \rangle$. If there exists any $f \in \mathcal{I}$ with

$$\operatorname{LM}(f) \text{ is not divisible by any } \operatorname{LM}(f_1), \ldots, \operatorname{LM}(f_m),$$

then G is not a Gröbner basis for $\langle G \rangle$; by definition.

S-Polynomials II

To obtain a candidate for such f from $f_1, \dots, f_s$, we may choose two elements f_i and f_j of G and compute

$$s = ax^{\alpha} f_i - bx^{\beta} f_j \text{ with } a, b \in \mathbb{F}.$$

We know that $\mathrm{LM}\left(ax^{\alpha} f_i - bx^{\beta} f_j\right)$ is divisible by an element of the Gröbner basis of $\mathcal{I}$ because $ax^{\alpha} f_i - bx^{\beta} f_j \in \mathcal{I}$.

S-Polynomials III

Now assume that in s the terms $ax^{\alpha}\text{LT}(f_i)$ and $bx^{\beta}\text{LT}(f_j)$ cancel each other out.

If as a result $\text{LM}\left(ax^{\alpha}f_i - bx^{\beta}f_j\right)$ is not divisible by any $\text{LM}(f_1), \ldots, \text{LM}(f_t)$ we know that G cannot be a Gröbner basis.

S-Polynomials IV

S-polynomials are a (in fact: **the**) way to construct the required cancellations of leading terms:

Definition (S-Polynomial)

Let $f, g \in \mathbb{F}[x_1, \ldots, x_n]$ be non-zero polynomials.

Let x^γ be the least common multiple of $\mathrm{LM}(f)$ and $\mathrm{LM}(g)$, written as

$$x^\gamma = \mathrm{LCM}(\mathrm{LM}(f), \mathrm{LM}(g)).$$

The S-polynomial of f and g is defined as

$$S(f,g) = \frac{x^\gamma}{\mathrm{LT}(f)} \cdot f - \frac{x^\gamma}{\mathrm{LT}(g)} \cdot g.$$

We call f and g the **generators** of $S(f,g)$ and $\frac{x^\gamma}{\mathrm{LT}(f)} \cdot f$ and $\frac{x^\gamma}{\mathrm{LT}(g)} \cdot g$ the **components** of $S(f,g)$.

S-Polynomials V

Example

Let $f_1 = x^3 - 2xy$ and $f_2 = x^2y - 2y^2 + x$. The leading monomials with respect to **degrevlex** and $x > y$ are $\mathrm{LM}\left(f_1\right) = x^3$ and $\mathrm{LM}\left(f_2\right) = x^2y$ and thus $x^\gamma = x^3y$. The S-polynomial is:

$$\begin{aligned}
S(f_1,f_2) &= \frac{x^\gamma}{\mathrm{LT}\left(f_1\right)} \cdot f_1 - \frac{x^\gamma}{\mathrm{LT}\left(f_2\right)} \cdot f_2 \\
S(f_1,f_2) &= \frac{x^3y}{x^3} \cdot (x^3 - 2xy) - \frac{x^3y}{x^2y} \cdot (x^2y - 2y^2 + x) \\
S(f_1,f_2) &= y \cdot (x^3 - 2xy) - x \cdot (x^2y - 2y^2 + x) \\
S(f_1,f_2) &= x^3y - 2xy^2 - x^3y + 2xy^2 - x^2 \\
S(f_1,f_2) &= -x^2
\end{aligned}$$

S-Polynomials VI

The same example in Sage:

```
sage: P.<x,y> = PolynomialRing(QQ,order=’degrevlex’)
sage: f0 = x^3 - 2*x*y
sage: f1 = x^2*y -2*y^2 + x
sage: (x^3*y)//x^3 * f0 - (x^3*y)//(x^2*y) * f1
-x^2
```

S-Polynomials VII

It is **sufficient** to consider **only** S-polynomials since **any** reduction of leading terms can be attributed to S-polynomials.

[Buc65] Bruno Buchberger
Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal,
Phd Thesis at Universität Innsbruck, 1965.

[Buc06] Bruno Buchberger
Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal
Journal of Symbolic Computation, 41:3-4, p. 475-511, 2006.

Buchberger's Criterion I

Buchberger's Criterion

Let $\mathcal{I}$ be an ideal. $G = \{g_1, \dots, g_s\}$ is a Gröbner basis for $\mathcal{I}$, if and only if for all pairs $i \neq j$, the remainder r of the division of $S(g_i, g_j)$ by G (listed in some order) is zero, i.e. we have that $\overline{f}^G = 0$.

Proof.

See [Cox et al., 2007, p.85ff]

Buchberger's Criterion II

Example

Let $f_1 = x^3 - 2xy$ and $f_2 = x^2y - 2y^2 + x$. The S-polynomial is $-x^2$ which is not reducible by either $\mathrm{LM}(f_1) = x^3$ or $\mathrm{LM}(f_2) = x^2y$. Thus, (f_1, f_2) is not a Gröbner basis.

Buchberger’s Algorithm

```
Input: $F$ – a finite subset of $P$
Result: a Gröbner basis for the ideal $\mathcal{I}$
        spanned by $F$
begin
    $G \longleftarrow F$;
    $G_2 \longleftarrow \varnothing$;
    while $G_2 \neq G$ do
        $G_2 \longleftarrow G$;
        for $f, g \in G_2 \times G_2$ do
            if $LM(f) < LM(g)$ then
                $\tilde{s} \longleftarrow \overline{S(f,g)}^{G}$;
                if $\tilde{s} \neq 0$ then
                    add $\tilde{s}$ to $G$;
    return $G$;
```

Algorithm 2: Buchberger’s Algorithm

Correctness and termination:

1. At every stage of the algorithm, $G \subset \mathcal{I}$ and $\langle G \rangle = \mathcal{I}$ hold.
2. If $G_2 = G$ then $\overline{S(f,g)}^{G} = 0$ for all $f, g \in G$ and, by Buchberger’s criterion, G is a Gröbner basis.
3. The equality $G_2 = G$ occurs in finitely many steps since the ideals $\langle \mathrm{LM}(G) \rangle$, in iterations of the loop, form an ascending chain. This chain of ideals stabilizes after a finite number of iterations and at that moment $\langle \mathrm{LM}(G) \rangle = \langle \mathrm{LM}(G_2) \rangle$ holds, which implies $G_2 = G$.

Complexity

The running time of Buchberger's algorithm is not polynomial in the number of variables, as the intermediate bases G_2 grow exponentially during the calculations.

Theorem [Faugère and Ars, 2004]

Let $\mathcal{I}$ be an ideal in $\mathbb{F}_q[x_1, \ldots, x_n]$ generated by polynomials $f_1, \ldots, f_n$ of degrees $d_1, \ldots, d_n$ respectively. Assume $V(\mathcal{I})$ is finite.

A Gröbner basis computation for a **lex** monomial order reaches at most degree $D \leq \prod_{i=0}^{n-1} d_i$.

A Gröbner basis computation for a **degrevlex** monomial order reaches at most degree $D \leq 1 - n + \sum_{i=0}^{n-1} d_i$.

Optional Improvements I

Buchberger's algorithm leaves a lot of freedom to implement. The runtime can be reduced by applying a variety of improvements:

- The order in which the critical pairs f, g are selected.
- If $\overline{S(f,g)}^{G} = 0$ we learn nothing new, we can employ criteria to filter out such computations.
- Algorithms exist [Faugère et al., 1993] to convert a Gröbner basis (of an ideal with $V(\mathcal{I})$ finite) in one monomial order to a Gröbner basis in another monomial order, thus we may compute with respect to the **degrevlex** ordering first and then convert the result to **lex**.

Optional Improvements II

Buchberger himself gave two criteria to avoid useless reductions to zero. We only give the first one here:

Definition (Buchberger's First Criterion)

Let $f, g \in \mathbb{F}[x_1, \ldots, x_n]$ with disjoint leading terms, i.e. $\mathrm{LCM}(\mathrm{LM}(f), \mathrm{LM}(g)) = \mathrm{LM}(f) \cdot \mathrm{LM}(g)$. Then $\overline{S(f,g)}^{G} = 0$.

Proof.

See [Becker and Weispfenning, 1991, p.222].

Exercises

1. In $\mathbb{F}_7[x, y]$ with **deglex** compute a Gröbner basis for

   $$\langle x^2y + 6, xy^2 - x\rangle$$

   using Buchberger's algorithm. How many reductions to zero $\overline{S(f, g)}^G = 0$ did you observe?
2. Repeate Exercise 1 in $\mathbb{F}_7[x, y, z]$ with **deglex** for

   $$\langle x^2 + y, x^4 + 2x^2y + y^2 + 3\rangle$$

   what does the result tell you abour $V(\mathcal{I})$?
3. Repeate Exercise 2 with

   $$\langle x - z^4, y - z^5\rangle$$

   and use Buchberger's First Criterion to avoid useless reductions to zero. How many did you avoid?

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

Quotient Rings I

- Recall that we considered finite extension fields as $\mathbb{F}_p[x]/f(x)$ for some irreducible polynomial $f(x)$.
- By this we mean that all multiplies of $f(x)$ are zero, i.e., we quotient out by elements in $\langle f(x)\rangle$.
- In fact, we can quotient out by any ideal $\mathcal{I}$ and define the quotient ring $Q = R/\mathcal{I}$.

Theorem [Cox et al., 2007, p.226]

Let $\mathcal{I}$ be an ideal in $\mathbb{F}[x_1, \dots, x_n]$. The ideals in the quotient ring $\mathbb{F}[x_1, \dots, x_n]/\mathcal{I}$ are in one-to-one correspondence with the ideals in $\mathbb{F}[x_1, \dots, x_n]$ containing $\mathcal{I}$ (that is, the ideals $\mathcal{J}$ satisfying $\mathcal{I} \subset \mathcal{J} \subset P$).

In particular, we may identify

$$\mathcal{I} = \langle f_1, \dots, f_m, x_1^q - x_1, \dots, x_n^q - x_n\rangle \in \mathbb{F}_q[x_1, \dots, x_n]$$

with

$$\mathcal{J} = \langle f_1, \dots, f_m\rangle \in \mathbb{F}_q[x_1, \dots, x_n]/\langle x_1^q - x_1, \dots, x_n^q - x_n\rangle.$$

Sage

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(2))
sage: I = sage.rings.ideal.FieldIdeal(P)
sage: Q = P.quotient_ring(I); Q
Quotient of Multivariate Polynomial Ring in x, y, z \
      over Finite Field of size 2 by the ideal (x^2 + x, y^2 + y, z^2 + z)

sage: P.<x,y,z> = BooleanPolynomialRing() # much faster!
sage: P.defining_ideal()
Ideal (x^2 + x, y^2 + y, z^2 + z) of Multivariate Polynomial Ring \
      in x, y, z over Finite Field of size 2
```

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

Elimination Ideals I

Given an ideal $\mathcal{I}$ in a polynomial ring $P = \mathbb{F}[x_1, \ldots, x_n]$ over a field $\mathbb{F}$ and a number $j \in \{1, \ldots, n\}$, consider the set of all polynomials in $\mathcal{I}$ which involve only the variables $x_j, \ldots, x_n$. This set $I \cap \mathbb{F}[x_j, \ldots, x_n]$ is an ideal in $\mathbb{F}[x_j, \ldots, x_n]$.

Elimination Ideals II

Definition (Elimination Ideal)

Given $\mathcal{I} = \langle f_1, \dots, f_m \rangle \subset \mathbb{F}[x_1, \dots, x_n]$, the ℓ-th elimination ideal I_ℓ is the ideal of $\mathbb{F}_{x_l,\dots,x_n}$ defined by

$$\mathcal{I}_\ell = \mathcal{I} \cap \mathbb{F}[x_l, \dots, x_n].$$

Elimination Ideals III

If we could compute Gröbner bases for these elimination ideals, we could start with $n = \ell$ to get a univariate polynomial. We could then factor it and substitute, which again would produce a univariate polynomial but this time in x_{n-1} etc.

Preprocessing I

In what follows we augment our ideal in $\mathbb{F}_q[x_1, \dots, x_n]$ with polynomials $x_1^q - x_1, \dots, x_n^q - x_n$ which we call field polynomials.

That is $\mathcal{I} = \mathcal{I} \cup \{x_i^q - x_i \mid 1 \leq i \leq n\}$, which makes sure,

- that our ideal has only finitely many points in $V(\mathcal{I})$ and
- that our ideal is "radical".

Both of these conditions are needed for what follows.

Elimination Theorem I

Elimination Theorem

Let $I \subset \mathbb{F}[x_1, \ldots, x_n]$ be an ideal and let G be a Gröbner basis of $\mathcal{I}$ with respect to the **lex** monomial ordering where $x_1 > x_2 > \cdots > x_n$. Then for every $1 \leq \ell \leq n$, the set

$$G_\ell = G \cap \mathbb{F}[x_\ell, \ldots, x_n]$$

is a Gröbner basis fo the ℓ-th elimination ideal I_ℓ.

In other words, the Gröbner basis G has triangular shape, which we can use to solve.

Elimination Theorem II

Example

Let $P = \mathbb{F}_{127}[x, y, z]$, the monomial ordering **lex** and consider the ideal

$$I = \langle x + y + z, xy + xz + yz, xyz - 1\rangle$$

We add the field polynomials and compute the reduced Gröbner basis:

$$x + y + z, y^2 + yz + z^2, z^3 - 1,$$

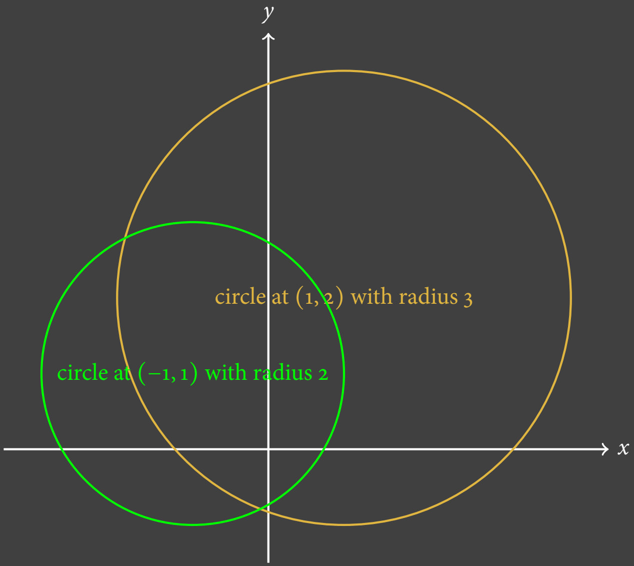
which has a triangular shape as predicted by the Elimination Theorem.

Elimination Theorem III

This result can be computed using Sage as follows:

```
sage: P.<x,y,z> = PolynomialRing(FiniteField(127),order='lex')
sage: I = sage.rings.ideal.Cyclic(P)
sage: I
Ideal (x + y + z, x*y + x*z + y*z, x*y*z - 1) of \
Multivariate Polynomial Ring in x, y, z over \
Finite Field of size 127
sage: J = I + sage.rings.ideal.FieldIdeal(P)
sage: g0,g1,g2 = J.groebner_basis(); g0,g1,g2
(x + y + z, y^2 + y*z + z^2, z^3 - 1)

sage: factor(g2)
(z - 19) * (z - 1) * (z + 20)

sage: factor(g1(x,y,19))
(y - 1) * (y + 20)

sage: factor(g0(x,1,19))
x + 20

sage: all(f(107,1,19)==0 for f in I.gens())
True
sage: J.variety()
[{y: 19, z: 1, x: 107}, {y: 107, z: 1, x: 19},
 {y: 1, z: 19, x: 107}, {y: 107, z: 19, x: 1},
 {y: 1, z: 107, x: 19}, {y: 19, z: 107, x: 1}]
```

An Example I


y
circle at $(1, 2)$ with radius 3
circle at $(-1, 1)$ with radius 2
x

An Example II

```
sage: x,y = var("x, y")
sage: p1 = (x-1)^2 + (y-2)^2 - 3^2 == 0
sage: p2 = (x+1)^2 + (y-1)^2 - 2^2 == 0
sage: sols = solve([p1,p2],x,y)
sage: sols
[x == -2/5*sqrt(5) - 1, y == 4/5*sqrt(5) + 1],
[x == 2/5*sqrt(5) - 1, y == -4/5*sqrt(5) + 1]]
```

An Example III

```
sage: P.<x,y> = PolynomialRing(QQ, order=’lex’)
sage: f1 = (x-1)^2 + (y-2)^2 - 3^2
sage: f2 = (x+1)^2 + (y-1)^2 - 2^2
sage: I = Ideal(f1,f2)
sage: I.groebner_basis()
[x + 1/2*y + 1/2, y^2 - 2*y - 11/5]

sage: I.variety()
[]

sage: I.variety(CC)
[{y: -0.788854381999832, x: -0.105572809000084},
  {y: 2.788854381999830, x: -1.89442719099992}]

sage: sage: P.<z> = QQ[]
sage: I.variety(NumberField(z^2-5,’a’))
[{y: -4/5*a + 1, x: 2/5*a - 1},
 {y: 4/5*a + 1, x: -2/5*a - 1}]
```

Contents

Sage

Polynomial Rings

Ideals

Gröbner Bases

Buchberger's Algorithm

Quotient Rings

Solving Polynomial Systems with Gröbner Bases

The F_4 Algorithm

Warm Up

As a warm-up, consider a linear system of equations over $\mathbb{F}_{127}[x, y, z]$.

$$\begin{aligned} f &= 26y + 52z + 62 = 0 \\ g &= 54y + 119z + 55 = 0 \\ h &= 41x + 91z + 13 = 0 \end{aligned} \qquad \begin{pmatrix} 0 & 26 & 52 & 62 \\ 0 & 54 & 119 & 55 \\ 41 & 0 & 91 & 13 \end{pmatrix}$$

After Gaussian elimination:

$$\begin{aligned} f' &= x + 29 = 0 \\ g' &= y + 38 = 0 \\ h' &= z + 75 = 0 \end{aligned} \qquad \begin{pmatrix} 1 & 0 & 0 & 29 \\ & 1 & 0 & 38 \\ & & 1 & 75 \end{pmatrix}$$

Thus, $x = -29$, $y = -38$ and $z = -75$ is a solution. We know this because Gaussian elimination produced small enough elements ($z + 75$) such that we can simply read of the solution.

Generalising to Non-Linear Polynomials

Now consider two polynomials in $\mathbb{F}_{127}[x, y, z]$ with term ordering **deglex**.

$$f = x^2 + 2xy - 2y^2 + 14z^2 + 22z \qquad \begin{pmatrix} 1 & 2 & -2 & 14 & 0 & 22 \\ 1 & 1 & 1 & 1 & 1 & 2 \end{pmatrix}$$
$$g = x^2 + xy + y^2 + z^2 + x + 2z$$

$$f = x^2 + 4y^2 - 12z^2 + 2x - 18z \qquad \begin{pmatrix} 1 & 0 & 4 & -12 & 2 & -18 \\ & 1 & -3 & 13 & -1 & 20 \end{pmatrix}$$
$$g' = xy + -3y^2 + 13z^2 - x + 20z$$

Gaussian elimination still “reduces” the system.

Limits of this Approach I

This approach fails for

$$\begin{aligned} f &= \mathbf{x}^2 - 2\mathbf{xy} - 2y^2 + 14z^2, \\ g &= \mathbf{x} + y + 2z. \end{aligned}$$

since **x** is not a monomial of f.

However, x divides two monomials of f: $\mathbf{x}^2$ and **xy**.

To account for those include multiples $m \cdot g$ of g such that

$$\mathrm{LM}(m \cdot g) = m \cdot \mathrm{LM}(g) \in \text{ the monomials of } f.$$

Limits of this Approach II

$$
\begin{aligned}
f &= x^2 - 2xy - 2y^2 + \dots \\
x \cdot g &= \mathbf{x^2} + xy \dots \\
y \cdot g &= \mathbf{xy} + y^2 + \dots \\
g &= x + y + 2z
\end{aligned}
$$

$$
\begin{pmatrix}
1 & -2 & -2 & 0 & 0 & 14 & 0 & \dots \\
1 & 1 & 0 & 2 & 0 & 0 & 0 & \dots \\
 & 1 & 1 & 0 & 2 & 0 & 0 & \dots \\
 & & & & & & 1 & \dots
\end{pmatrix}
$$

$$
\begin{aligned}
f' &= x^2 + 4yz + 14z^2, \\
h_1 &= \mathbf{xy} + 2xz + -4yz - \dots, \\
h_2 &= \mathbf{y^2} - 2xz + 6yz + \dots, \\
g &= x + y + 2z
\end{aligned}
$$

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & \dots & 0 & \dots \\
 & 1 & 0 & 2 & \dots & 0 & \dots \\
 & & 1 & -2 & \dots & 0 & \dots \\
 & & & & \dots & 1 & \dots
\end{pmatrix}
$$

Let’s call the preprocessing we performed “symbolic preprocessing” … but that alone is still not enough to solve the system.

Adding S-Polynomials

Finally, consider

$$\begin{aligned} f &= yx + 1, \\ g &= zx + 2. \end{aligned}$$

Neither LM (f) nor LM (g) divides any monomial in the other polynomial. However, we have

$$\begin{aligned} zf - yg &= z(yx + 1) - y(zx + 2), \\ &= \mathbf{xyz} + z - \mathbf{xyz} - 2y, \\ &= z - 2y. \end{aligned}$$

We constructed multiples of f and g such that when we subtract them their leading terms cancel out and something smaller is produced: we constructed an **S-polynomial**.

The F_4 Algorithm I

```
Input: $F = [f_1, \dots, f_m]$ – list of polynomials
Output: a Gröbner basis for $\langle f_1, \dots, f_{m-1} \rangle$
begin
    while True do
        $\overline{F} \leftarrow$ multiply all pairs $f_i, f_j \in F$ by $m_i, m_j$ such that $\mathrm{LM}(m_i f_i) = \mathrm{LM}(m_j f_j)$;
        $\overline{F} \leftarrow$ perform “symbolic preprocessing” on $\overline{F} \cup F$;
        $\tilde{F} \leftarrow$ peform Gaussian elimination on $\overline{F}$ ;
        $F \leftarrow F \cup \{f \in \tilde{F}$ with $\forall g \in F$ we have $\mathrm{LM}(g) \nmid \mathrm{LM}(f)\}$;
        if F didn’t change in the last iteration then
            return $F$;
```

Algorithm 3: simplified F_4

The F_4 Algorithm II

```
1 begin
2   while True do
3     $\overline{F} \leftarrow$ multiply all pairs $f_i, f_j \in F$ by $m_i, m_j$ such that $\mathrm{LM}(m_i f_i) = \mathrm{LM}(m_j f_j)$;
4     $\overline{F} \leftarrow$ perform “symbolic preprocessing” on $\overline{F} \cup F$;
5     $\tilde{F} \leftarrow$ peform Gaussian elimination on $\overline{F}$;
6     $F \leftarrow F \cup \{f \in \tilde{F}$ with $\forall g \in F$ we have $\mathrm{LM}(g) \nmid \mathrm{LM}(f)\}$;
7     if F didn’t change in the last iteration then
8       return $F$;
```

Buchberger select one pair in line 3 and use polynomial division instead of Gaussian elimination in line 5; implemented everywhere

F_4 use Buchberger’s criteria in line 3 to avoid useless pairs (= zero rows in the matrix); implemented in MAGMA, POLYBORI, FGB

F_5 use criteria in lines 3 and 4 such that all matrices have full rank under some assumption; implementation worked on in SINGULAR

The F_4 Algorithm III

```
1 begin
2     while True do
3         $\overline{F} \leftarrow$ multiply all pairs $f_i, f_j \in F$ by $m_i, m_j$ such that $\mathrm{LM}(m_i f_i) = \mathrm{LM}(m_j f_j)$;
4         $\overline{F} \leftarrow$ perform “symbolic preprocessing” on $\overline{F} \cup F$;
5         $\tilde{F} \leftarrow$ peform Gaussian elimination on $\overline{F}$;
6         $F \leftarrow F \cup \{f \in \tilde{F}$ with $\forall g \in F$ we have $\mathrm{LM}(g) \nmid \mathrm{LM}(f)\}$;
7         if F didn’t change in the last iteration then
8             return $F$;
```

Re-Linearisation Gaussian elimination is enough because we have so many equations

(Mutant)XL multiply by everything up to some degree in line 3 and skip line 4 (worse than Simple F_4 because of redundancies)

Fin

Questions?

Literature I

Becker, T. and Weispfenning, V. (1991).
Gröbner Bases – A Computational Approach to Commutative Algebra.
Springer Verlag, Berlin, Heidelberg, New York.

Cox, D., Little, J., and O'Shea, D. (2007).
Ideals, Varieties, and Algorithms.
Springer Verlag, Berlin, Heidelberg, New York.

Faugère, J.-C. and Ars, G. (2004).
Comparison of XL and Gröbner basis algorithms over finite fields.
Technical Report 5251, Institut National de Recherche en Informatique et en Automatique (INRIA).

Faugère, J.-C., Gianni, P. M., Lazard, D., and Mora, T. (1993).
Efficient computation of zero-dimensional Gröbner Bases by change of ordering.
Journal of Symbolic Computation, 16:329–344.