A dynamic F4 algorithm

John Perry

University of Southern Mississippi

ACA 2019

Outline

Motivation, technical background

Motivation, technical background:

Basics

Important questions

For a system of polynomial equations, determine:

- existence of solutions
- dimension of solutions
- explicit description of solutions
- is f in subspace

Analogies

	linear
structure	vector space
multipliers	field elements
workspace	subspace
presentation	spanning set
***good* presentation**	basis
transformation	Gauss-Jordan

Analogies

	linear	**non-linear**
structure	vector space	polynomial ring
multipliers	field elements	monomials
workspace	subspace	ideal
presentation	spanning set	basis
***good* presentation**	basis	Gröbner basis
transformation	Gauss-Jordan	Buchberger

Buchberger Example

$$x^2 + y^2 - 4$$
$$xy - 1$$

Buchberger Example

$$\begin{array}{r} x^2 + y^2 - 4 \\ xy - 1 \end{array} \implies \begin{array}{rrr} & x^2 y + y^3 - 4y & \\ - & (x^2 y \qquad\qquad - x) & \\ \hline & y^3 - 4y + x & \end{array}$$

S-poly reduction

Motivation, technical background:

A "linear algebra" point of view

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction ⇜⇝ Buchberger's algorithm

$$\begin{array}{c c}
\begin{array}{cccc} x^2 & xy & y^2 & 1 \end{array} & \\
\left(\begin{array}{cccc} 1 & & 1 & -4 \\ & 1 & & -1 \end{array}\right) & \begin{array}{c} g \\ f \end{array}
\end{array}$$

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction ⇝ Buchberger's algorithm

$$\begin{array}{c}
\begin{array}{cccccc} xy^2 & x^2 & xy & y^2 & y & 1 \end{array} \\
\left(\begin{array}{cccccc}
 & 1 & & 1 & & -4 \\
1 & & & & -1 & \\
 & & \color{red}{1} & & & \color{red}{-1}
\end{array}\right)
\begin{array}{c} g \\ yf \\ \color{red}{f} \end{array}
\end{array}$$

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction $\rightsquigarrow$ Buchberger's algorithm

$$
\begin{array}{c}
\begin{array}{cccccccc} x^2y & xy^2 & x^2 & xy & y^2 & x & y & 1 \end{array} \\
\left(\begin{array}{cccccccc}
 & & 1 & & 1 & & & -4 \\
1 & & & & & -1 & & \\
 & 1 & & & & & -1 & \\
 & & & {\color{red}1} & & & & {\color{red}-1}
\end{array}\right)
\end{array}
\begin{array}{c} \\ g \\ xf \\ yf \\ {\color{red}f} \end{array}
$$

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction $\rightsquigarrow$ Buchberger's algorithm

$$\begin{array}{l}
\begin{array}{cccccccccc} \cdots & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 \end{array} \\
\left(\begin{array}{cccccccccc}
 & 1 & & 1 & & & & & -4 & \\
 & & & & \color{red}{1} & & \color{red}{1} & & & \color{red}{-4} \\
\ddots & & & & & & \ddots & & & \\
 & 1 & & & & & & -1 & & \\
 & & 1 & & & & & & -1 & \\
 & & & & & 1 & & & & -1
\end{array}\right)
\begin{array}{c} y\mathbf{g} \\ \color{red}{\mathbf{g}} \\ \vdots \\ x\mathbf{f} \\ y\mathbf{f} \\ \mathbf{f} \end{array}
\end{array}$$

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction $\leftrightsquigarrow$ Buchberger’s algorithm

$$\begin{array}{cccccccccccc}
\cdots & x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 & \\
 & 1 & & 1 & & & & & -4 & & & xg \\
 & & 1 & & 1 & & & & & -4 & & yg \\
 & & & & & \color{red}{1} & & \color{red}{1} & & & \color{red}{-4} & \color{red}{g} \\
 & \ddots & & & & & & \ddots & & & & \vdots \\
 & & 1 & & & & & & -1 & & & xf \\
 & & & 1 & & & & & & -1 & & yf \\
 & & & & & & 1 & & & & -1 & f
\end{array}$$

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction $\leftrightsquigarrow$ Buchberger's algorithm

$$
\begin{array}{cc}
\begin{array}{ccccccccccc} \cdots & x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 \end{array} & \\
\left(\begin{array}{ccccccccccc}
\ddots & & \ddots & & & & & \ddots & & & \\
 & 1 & & 1 & & & & & -4 & & \\
 & & 1 & & 1 & & & & & -4 & \\
 & & & & & 1 & & 1 & & & -4 \\
 & \ddots & & & & & & \ddots & & & \\
 & & 1 & & & & & & -1 & & \\
 & & & 1 & & & & & & -1 & \\
 & & & & & & 1 & & & & -1
\end{array}\right) &
\begin{array}{c} \vdots \\ x\mathbf{g} \\ y\mathbf{g} \\ \mathbf{g} \\ \vdots \\ x\mathbf{f} \\ y\mathbf{f} \\ \mathbf{f} \end{array}
\end{array}
$$

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction $\leftrightsquigarrow$ Buchberger's algorithm

$$
\begin{array}{cccccccccccc}
\cdots & x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 & \\
\ddots & & \ddots & & & & & \ddots & & & & \vdots \\
 & 1 & & 1 & & & & & -4 & & & xg \\
 & & 0 & & 1 & & & & 1 & -4 & & yg \\
 & & & & & 1 & & 1 & & & -4 & g \\
 & \ddots & & & & & & \ddots & & & & \vdots \\
 & & 1 & & & & & & -1 & & & xf \\
 & & & 1 & & & & & & -1 & & yf \\
 & & & & & & 1 & & & & -1 & f
\end{array}
$$

F4 [Lazard, 1983], [Faugère, 1999]

Gaussian reduction $\leftrightsquigarrow$ Buchberger's algorithm

$$\begin{array}{cc}
\begin{array}{ccccccccccc} \cdots & x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 \end{array} & \\
\left(\begin{array}{ccccccccccc}
\ddots & & \ddots & & & & & \ddots & & & \\
 & 1 & & 1 & & & & & -4 & & \\
 & & & & 1 & & & & 1 & -4 & \\
 & & & & & 1 & & 1 & & & -4 \\
 & \ddots & & & & & & \ddots & & & \\
 & & 1 & & & & & & -1 & & \\
 & & & 1 & & & & & & -1 & \\
 & & & & & & 1 & & & & -1
\end{array}\right) &
\begin{array}{c} \vdots \\ xg \\ yg \\ g \\ \vdots \\ xf \\ yf \\ f \end{array}
\end{array}$$

GB: $\{ \quad xy - 1, \quad x^2 + y^2 - 4, \quad y^3 + x - 4 \quad \}$

Advantages

1. Locality of data

Advantages

1. Locality of data
2. Easy to parallelize

Advantages

1. Locality of data
2. Easy to parallelize
3. Sparse matrix data structures, algorithms

F4 Algorithm (highly simplified)

inputs $F, >$

repeat

- build “important submatrix”
- perform Gauss-Jordan

until all important submatrices triangular

$G \leftarrow$ important submatrices’ rows

return G

Dynamic F4

Dynamic F4:

Idea

Back to Linear Algebra

Triangularize?

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & & \\ 1 & 1 & & & \\ 1 & & & & \end{pmatrix}$$

Back to Linear Algebra

Triangularize?

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & & \\ 1 & 1 & & & \\ 1 & & & & \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 \\ & & 1 & 1 & 1 \\ & & & 1 & 1 \\ & & & & 1 \end{pmatrix}$$

No reduction needed!

Back to Linear Algebra

Triangularize?

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & & \\ 1 & 1 & & & \\ 1 & & & & \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 \\ & & 1 & 1 & 1 \\ & & & 1 & 1 \\ & & & & 1 \end{pmatrix}$$

No reduction needed!

Can we do this with Gröbner bases?

- change ordering *during* algorithm
- *dynamic* variant of *static* algorithm
- *adapt*, not *abolish*

Why?

Why?

1. Why not?

Why?

1. Why not?
2. Might be "practical"
   - faster?
     - maybe...
     - ...but not if overhead is too large

Why?

1. Why not?
2. Might be “practical”
    - faster?
        - maybe…
        - …but not if overhead is too large
    - smaller basis?
        - faster application
        - tradeoff could be worth it!

Why not implemented?

- monomials $\leftrightsquigarrow$ columns
- swap columns? $\rightsquigarrow$ out of order!
- out of order? $\rightsquigarrow$ not Gröbner!

hard to implement w/out breaking other things

Why not implemented?

- monomials $\leftrightsquigarrow$ columns
- swap columns? $\rightsquigarrow$ out of order!
- out of order? $\rightsquigarrow$ not Gröbner!

hard to implement w/out breaking other things

Non-trivial problem

Example

$$\begin{array}{c}
\begin{array}{ccccccccccc}
\cdots & x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1
\end{array} \\
\left(\begin{array}{ccccccccccc}
\ddots & & \ddots & & & & & \ddots & & & \\
 & 1 & & 1 & & & & & -4 & & \\
 & & 1 & & 1 & & & & & -4 & \\
 & & & & & 1 & & 1 & & & -4 \\
 & \ddots & & & & & & \ddots & & & \\
 & & 1 & & & & & & -1 & & \\
 & & & 1 & & & & & & -1 & \\
 & & & & & & 1 & & & & -1
\end{array}\right)
\begin{array}{c}
\vdots \\ xg \\ yg \\ g \\ \vdots \\ xf \\ yf \\ f
\end{array}
\end{array}$$

Example

$$
\begin{array}{cc}
\begin{matrix} \cdots & x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 \end{matrix} & \\
\begin{pmatrix}
\ddots & & \ddots & & & & & \ddots & & & \\
 & 1 & & 1 & & & & & -4 & & \\
 & & \color{red}{1} & & \color{red}{1} & & & & & \color{red}{-4} & \\
 & & & & & 1 & & 1 & & & -4 \\
 & \ddots & & & & & & \ddots & & & \\
 & & \color{red}{1} & & & & & & \color{red}{-1} & & \\
 & & & 1 & & & & & & -1 & \\
 & & & & & & 1 & & & & -1
\end{pmatrix}
&
\begin{matrix} \vdots \\ xg \\ yg \\ g \\ \vdots \\ xf \\ yf \\ f \end{matrix}
\end{array}
$$

Example

$$
\begin{array}{c@{}c}
\begin{array}{ccccccccccc}
\cdots & x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1
\end{array} & \\
\begin{pmatrix}
\ddots & & \ddots & & & & & \ddots & & & \\
 & 1 & & 1 & & & & & -4 & & \\
 & & 1 & & 1 & & & & & -4 & \\
 & & & & & 1 & & 1 & & & -4 \\
 & \ddots & & & & & & \ddots & & & \\
 & & 1 & & & & & & -1 & & \\
 & & & 1 & & & & & & -1 & \\
 & & & & & & 1 & & & & -1
\end{pmatrix} &
\begin{array}{c}
\vdots \\ xg \\ yg \\ g \\ \vdots \\ xf \\ yf \\ f
\end{array}
\end{array}
$$

Example

$$
\begin{array}{cc}
\begin{array}{ccccccccccc}
\cdots & x^3 & {\color{red}y^2} & xy^2 & y^3 & x^2 & xy & {\color{red}x^2y} & x & y & 1
\end{array} & \\
\begin{pmatrix}
\ddots & & \ddots & & & & & \ddots & & & \\
 & 1 & & 1 & & & & & -4 & & \\
 & & & & 1 & & & {\color{red}1} & & -4 & \\
 & & {\color{red}1} & & & 1 & & & & & -4 \\
 & \ddots & & & & & & \ddots & & & \\
 & & & & & & & {\color{red}1} & -1 & & \\
 & & & 1 & & & & & & -1 & \\
 & & & & & & 1 & & & & -1
\end{pmatrix} &
\begin{array}{c}
\vdots \\ xg \\ yg \\ g \\ \vdots \\ xf \\ yf \\ f
\end{array}
\end{array}
$$

Example

$$\begin{array}{cc}
\begin{array}{ccccccccccc} \cdots & x^3 & {\color{red}y^2} & xy^2 & y^3 & x^2 & xy & {\color{red}x^2y} & x & y & 1 \end{array} & \\
\left(\begin{array}{ccccccccccc}
\ddots & & \ddots & & & & & \ddots & & & \\
 & 1 & & 1 & & & & & -4 & & \\
 & & & & 1 & & & {\color{red}1} & & -4 & \\
 & & {\color{red}1} & & & 1 & & & & & -4 \\
 & \ddots & & & & & & \ddots & & & \\
 & & & & & & & {\color{red}1} & -1 & & \\
 & & & 1 & & & & & & -1 & \\
 & & & & & & 1 & & & & -1
\end{array}\right) &
\begin{array}{c} \vdots \\ x\mathrm{g} \\ y\mathrm{g} \\ \mathrm{g} \\ \vdots \\ x\mathrm{f} \\ y\mathrm{f} \\ \mathrm{f} \end{array}
\end{array}$$

$\{x^2y + x - 4y, y^2 + x^2 - 4, x^2y - 1\}$ not GB under any order

Past work

Dynamic Buchberger implemented, improved, explored

[Caboara, 1993]	AlPI, CoCoA (lost)
[Golubitsky, unpublished]	Maple
[Caboara & Perry, 2014]	Sage
[Hashemi & Talaashrafi, 2016]	Sage
[Perry, 2017]	C++
[Langeloh, 2019]	Sage

*[Gritzmann & Sturmfels, 1993] theory, not implementation

Dynamic F4:

Algorithm

Question #1

Can we implement a Dynamic F4 algorithm?

(1) Can it be done?

Question #1

Can we implement a Dynamic F4 algorithm?

(1) Can it be done?
(2) portably?

Question #1

Can we implement a Dynamic F4 algorithm?

(1) Can it be done?
(2) portably?
(3) efficiently?

Question #1

Can we implement a Dynamic F4 algorithm?

(1) Can it be done?
(2) portably?
(3) efficiently?
(4) in an existing computer algebra system?

Question #1

Can we implement a Dynamic F4 algorithm?

(1) Can it be done?
(2) portably?
(3) efficiently?
(4) in an existing computer algebra system?

(1, 2) Yes

Question #1

Can we implement a Dynamic F4 algorithm?

(1) Can it be done?
(2) portably?
(3) efficiently?
(4) in an existing computer algebra system?

(1, 2) Yes (3) Define "efficiently"

Question #1

Can we implement a Dynamic F4 algorithm?

(1) Can it be done?
(2) portably?
(3) efficiently?
(4) in an existing computer algebra system?

(1, 2) Yes (3) Define "efficiently"

(4) Yes, but not without a lot of work

F4 Algorithm (highly simplified)

inputs $F, >$

repeat

- build "important submatrix"
- perform Gauss-Jordan

until all important submatrices triangular

$G \leftarrow$ important submatrices' rows

return G

Dynamic F4 Algorithm (highly simplified)

inputs F

choose initial ordering

repeat

- build "important submatrix"
- perform Gauss-Jordan
- non-zero rows?
    - reconsider ordering
    - adjust matrix

until all important submatrices triangular

$G \leftarrow$ important submatrices' rows

return G

“Reconsider ordering”?

while some rows unprocessed

- identify each row’s possible lm’s *(parallel)*
- select best row, best ordering
- adjust matrix, reduce by new row *(parallel)*

“Reconsider ordering”?

while some rows unprocessed

- identify each row’s possible lm’s *(parallel)*
- select best row, best ordering
- adjust matrix, reduce by new row *(parallel)*

bottleneck: identifying possible lm’s over and over and …

Implementation?

DynGB

Work-in-progress (Summer 2019)

- “restricted” algorithm
  (once chosen, lm’s cannot change)
- portability: {C++11} ∪ {GMP, GLPK, PPL}
  - parallelism via STL `thread` / `async`

Implementation?

DynGB

Work-in-progress (Summer 2019)

- "restricted" algorithm
  (once chosen, lm's cannot change)
- portability: {C++11} ∪ {GMP, GLPK, PPL}
  - parallelism via STL `thread` / `async`
- works, but not a speed demon
  - slight disadvantage from weighted term ordering
  - Dynamic Buchberger is remarkably slow
  - Dynamic F4 is not bad

Implementation?

DynGB

Work-in-progress (Summer 2019)

- "restricted" algorithm
  (once chosen, lm's cannot change)
- portability: $\{$C++11$\} \cup \{$GMP, GLPK, PPL$\}$
  - parallelism via STL `thread` / `async`
- works, but not a speed demon
  - slight disadvantage from weighted term ordering
  - Dynamic Buchberger is remarkably slow
  - Dynamic F4 is not bad
- eventual plan is to fold dynamic code into existing GB implementation (e.g., Sage, Eder's F4 code)

Sample performance

This is where I demonstrate the implementation.

Hopefully it works just as well as the last time I tried it.

If it doesn't, I will cry.

Summary for offline readers

Computed homogeneous Cyclic-8 GB

- DynF4 computes a basis of 404 polynomials in 30 sec
- Sage / SINGULAR computes a basis of >1000 polynomials in 60 sec

A larger example

Homogeneous Cyclic-9, $\mathbb{Z}_{43}$

processor	2.5 GHz
basis size	1996
time	2246 sec (37 min)
static overhead	732 sec
reducing	1046 sec
dynamic overhead	450 sec
memory used	5.6 GB

Comparison to static

SINGULAR (Buchberger)

processor	2.5 GHz	
time	4h 20m	$(\times 7)$
basis size	5601	$(\times 2.8)$

Mathic GB (static F4)

processor	3.6 GHz	
time	11m	$(\times 2/5)$
memory used	11GB	$(\times 2)$
basis size	5602	$(\times 2.8)$

Eder's GB (static F4)

processor	3.1 GHz	
time	5m	$(\times 1/6)$
memory used	1GB	$(\times 1/5)$
basis size	5601	$(\times 2.8)$

Identifying incompatible terms

Question

How can we identify possible leading terms?

- $\mathbf{x}^2 + x$
- $\mathbf{x}^2 + xy + \mathbf{y}^2$

Equivalent question

How can we identify *incompatible* terms?

- $x^2 + \mathbf{x}$
- $x^2 + \mathbf{xy} + y^2$

Identifying incompatible terms:

No constraints

Binomials

Theorem ([Caboara, 1993])

$u \mid t$? $u \neq \mathrm{lm}(t+u)$

Binomials

Theorem ([Caboara, 1993])

$u \mid t$? $u \neq \mathrm{lm}(t+u)$

Necessary and sufficient for binomials:

- $u \nmid t \quad \Longrightarrow \quad u = \mathrm{lm}_{\text{“lex”}}(t+u)$

Trinomials

Question

Under what condition(s) is $u \neq \mathrm{lm}(t+u+v)$?

Trinomials

Question

Under what condition(s) is $u \neq \mathrm{lm}(t+u+v)$?

New criterion (w/Mitchell, 2019)

$$u^2 \mid tv$$

Trinomials

Question

Under what condition(s) is $u \neq \mathrm{lm}(t+u+v)$?

New criterion (w/Mitchell, 2019)

$$u^2 \mid tv$$

Sufficient, but not necessary

Trinomials (again)

Question

Under what condition(s) is $u \neq \mathrm{lm}(t+u+v)$?

Trinomials (again)

Question

Under what condition(s) is $u \neq \operatorname{lm}(t+u+v)$?

New criterion (w/Mitchell, 2019)

$t=\mathbf{x}^{\mathbf{a}}$, $u=\mathbf{x}^{\mathbf{b}}$, $v=\mathbf{x}^{\mathbf{c}}$

$$b_i b_j \leq (a_i - b_i)\left(c_j - b_j\right) \quad \forall i \neq j$$

Trinomials (again)

Question

Under what condition(s) is $u \neq \operatorname{lm}(t+u+v)$?

New criterion (w/Mitchell, 2019)

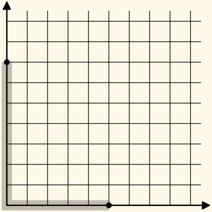
$t = \mathbf{x}^{\mathbf{a}}$, $u = \mathbf{x}^{\mathbf{b}}$, $v = \mathbf{x}^{\mathbf{c}}$

$$b_i b_j \leq (a_i - b_i)\left(c_j - b_j\right) \quad \forall i \neq j$$

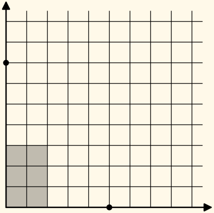
Necessary *and* sufficient
(under reasonable restrictions)

Geometry

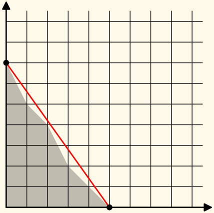
Detected incompatible monomials: x^5, y^7



$u \mid t, v$



$u^2 \mid tv$



$b_1 b_2 \leq (a - b_1)(c - b_2)$

Identifying incompatible terms:

Known constraints

Weighted degree ordering

$$x^3 \quad + \quad y^4 \quad + \quad x^2z$$

Weighted degree ordering

$$\begin{array}{ccccc} \color{red}{x^3} & + & y^4 & + & x^2z \\ 3 & & 0 & & 0 \end{array}$$

$$\color{red}{(1,0,0)}$$

Weighted degree ordering

$$\begin{array}{ccccc} x^3 & + & \color{red}{y^4} & + & x^2 z \\ 3 & & 4 & & 3 \\ & & \color{red}{(1,1,1)} & & \end{array}$$

Weighted degree ordering

$$\begin{array}{ccccc} x^3 & + & y^4 & + & x^2z \\ 3 & & 4 & & 5 \\ & & (1,1,3) & & \end{array}$$

Ordering subject to constraints

$$x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad > \quad x_1^{\beta_1} \cdots x_n^{\beta_n}$$

Ordering subject to constraints

$$x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad > \quad x_1^{\beta_1} \cdots x_n^{\beta_n}$$

$$\alpha \cdot \omega \quad > \quad \beta \cdot \omega$$

Ordering subject to constraints

$$x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad > \quad x_1^{\beta_1} \cdots x_n^{\beta_n}$$

$$\alpha \cdot \omega \quad > \quad \beta \cdot \omega$$

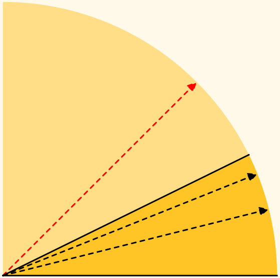
$$\sum (\alpha_i - \beta_i)\omega_i \quad > 0$$

inexact: simplex (GLPK)

exact: double description method (PPL, Skeleton)

Geometric view

linear constraints $\longrightarrow$ polyhedral cone (**skeleton**)



From Corner Point Theorem

Skeleton identifies minimal set of potential lm's
consistent with previous choices

Example

$$f = x + y$$
$$g = x^2 + y^2$$

Example

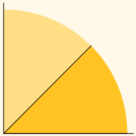
$$f = x + y$$
$$g = x^2 + y^2$$

- Choose $x = \mathrm{lm}(x + y)$

Example

$$f = x + y$$
$$g = x^2 + y^2$$

- Choose $x = \mathrm{lm}(x + y)$
- $x > y$? skeleton $\{(1,0),(1,1)\}$

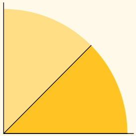


Example

$$f = x + y$$
$$g = x^2 + y^2$$

- Choose $x = \mathrm{lm}(x + y)$
- $x > y$? skeleton $\{(1,0),(1,1)\}$

$$\therefore \begin{array}{ccc} x^2 & > & y^2 \\ 2 & & 0 \text{ or } 2 \end{array}$$



Limitations

- Two terms only
- $u = \mathrm{lm}(t+u+v)$ iff some corner vector allows it? *NO*

Example ($t = x^5$, $u = x^2y^6$, $v = y^7$)

- u *can* be $\mathrm{lm}(t+u+v)$
- but initial skeleton will not determine this $\{(1,0),(0,1)\}$

Limitations

- Two terms only
- $u = \mathrm{lm}(t+u+v)$ iff some corner vector allows it? *NO*

Example ($t = x^5$, $u = x^2y^6$, $v = y^7$)

- u *can* be $\mathrm{lm}(t+u+v)$
- but initial skeleton will not determine this $\{(1,0),(0,1)\}$

Open question

Find a sufficient and necessary criterion for $u \neq \mathrm{lm}(f)$?

Conclusions, future directions

Summary

- Dynamic algorithms seek out "good" orderings
  - can be faster, smaller... but no guarantee
  - metrics need further investigation
- DynF4 shows promise (IMHO)
- Criteria to identify incompatible leading terms

Thank you!

Massimo Caboara

Università di Pisa

Christian Eder

Centre for Computer Algebra

William Stein

Николай Юрьевич Золотых

Казанский Федеральний Университет

University of Southern Mississippi

Галина Валеревна Заморий

Candice Mitchell

NASA Space Grant Consortium

Bibliography: General

- Buchberger, *An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*. PhD Thesis (1965), University of Innsbruck. English translation in *JSC* 41 (2006) 475–511.
- Lazard, "Gröbner bases, Gaussian elimination, and resolution of systems of algebraic equations." *EUROCAL '83*, Springer LNCS 162, 146–156.
- Robbiano, "On the theory of graded structures," *JSC* 2 (1986) 139–170.
- Mora and Robbiano, "The Gröbner fan of an ideal." *JSC* 6 (1988) 183–208.
- Faugère, "A new efficient algorithm for computing Gröbner bases (F_4)." *Journal of Pure and Applied Algebra* 139 (1999) 61–88.
- Faugère, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)." *ISSAC '02*, ACM Press, 75–82.

Bibliography: Dynamic

- Caboara, "A Dynamic Algorithm for Gröbner basis computation." *ISSAC '93*, ACM Press, 275–283.
- Caboara and Perry, "Reducing the size and number of linear programs in a dynamic Gröbner basis algorithm." *AAECC* 25 (2014) 99–117.
- O. Golubitsky, "Converging term order sequences and the dynamic Buchberger algorithm." Unpublished preprint received via private communication.
- Gritzmann and Sturmfels, "Minkowski Addition of Polytopes: Computational Complexity and Applications to Gröbner Bases." *SIAM J. Disc. Math* 6 (1993) 246–269.
- Hashemi and Talaashrafi, "A Note on Dynamic Gröbner Bases Computation." *CASC 2016*, Springer, 276–288.
- Langeloh, "Unrestricted dynamic Gröbner Basis algorithms." Master's Thesis, 2019.
- Perry, "Exploring the Dynamic Buchberger Algorithm." ISSAC 2017, ACM Press, 365–372.