# The F4 and F5 Algorithms for Computing Gröbner Bases

Roman Pearce Simon Fraser University

November 18th, 2005

## **Introduction: Gröbner Bases**

- Gröbner bases are a type of canonical basis for polynomial system
- They have a nice division property w.r.t. a *monomial order* 
  - *lexicographic* (dictionary) order: used for elimination
  - graded (total degree) orders: fast!

Example 
$$\{x^2 + y - z, 2xy - z, xz - 5\} \subset \mathbb{Q}[x, y, z]$$

• with graded lex order (x > y > z):  $\{z^2 - 10y, yz + 5x - 10y, 2y^2 + 10x - 20y + 5, xz - 5, 2xy - z, x^2 + y - z\}$ 

• with lex order 
$$(x > y > z)$$
:  
 $\{z^4 - 10z^3 + 250, 10y - z^2, 50x + z^3 - 10z^2\}$ 

# Timeline

- (1965) Buchberger's original algorithm
- (1979) Improved versions of Buchberger's algorithm
- (1988) Nearly optimal version of Buchberger's algorithm
- (1993) FGLM conversion method (f.d. systems only)
- (1997) Gröbner Walk conversion method
- (1999) F4 algorithm
- (2002) F5 algorithm

## **Buchberger's Algorithm**

• select pairs of polynomials and compute a *syzygy*:

$$(x^2 - 1, xy - 1) \longrightarrow y(x^2 - 1) - x(xy - 1) = x - y$$

- reduce each syzygy using the current basis
- if non-zero, add the result to the current basis ( $\rightarrow$  more syzygies)

#### Improvements:

- many syzygies are redundant (*criterion*)
- what syzygies should be reduced first? (*selection strategy*)
- some basis elements may become redundant (*minimality*)

#### **Reductions in the Buchberger Algorithm**

• like univariate division, but some terms may not reduce

Example Divide  $x^2y + y^3$  by  $G = [x^2 + y, xy^2 - xy, y^3 - 1]$  (grlex x > y)  $x^2y + y^3 \rightarrow [x^2y - yG_1] + y^3 = y^3 - y^2$  $\rightarrow [y^3 - G_3] - y^2 = -y^2 + 1$ 

- most time spent reducing syzygies to zero (wasted effort)
- equivalent to a matrix triangularization

	$ x^2y $	$y^{3}$	$y^2$	1			$x^2y$	$y^{3}$	$y^2$	1
S <sub>12</sub>	1	1	0	0	$\longrightarrow$	S <sub>12</sub>	1	1	0	0
$-yG_1$	1	0	1	0		$-yG_1$	0	1	-1	0
$-G_3$	0	1	0	-1		$-G_3$	0	0	1	-1

# The F4 Algorithm - 1

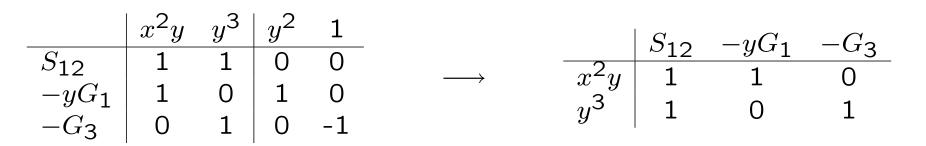
- put multiple syzygies into one matrix
- cost of all reductions decreases by two orders of magnitude
- exploit strategies for sparse linear algebra
- (!) modular algorithm: reduce mod p, extract only new rows

# The F4 Algorithm - 2

- put multiple syzygies into one matrix
- cost of all reductions decreases by two orders of magnitude
- exploit strategies for sparse linear algebra
- (!) modular algorithm: reduce mod p, extract only new rows
- matrices are big, with many more columns than rows
- must do (slower) multi-modular lifting
- unable to easily express Gröbner basis in terms of generators

## **More Efficient Reductions**

#### **Conversion to Nullspace Problem:**



**Conversion to Linear System:** 

- row reduce mod p to determine dependent columns
- stick those columns in the right hand side
- use p-adic lifting to recover solution
- solutions are syzygies: can express GB in terms of input

## **Reductions to Zero**

**Recall:** solution of  $AX = B \rightarrow$  nullspace elements  $\rightarrow$  new polynomials

**Problem:** what if the "new polynomial" is zero ?

• redundant rows in the original matrix  $(mg_i)$ 

## **Reductions to Zero**

**Recall:** solution of  $AX = B \rightarrow$  nullspace elements  $\rightarrow$  new polynomials

**Problem:** what if the "new polynomial" is zero ?

• redundant rows in the original matrix  $(mg_i)$ 

Faugère's insight:

•  $m \in \langle g_1, \ldots, g_{i-1} \rangle$ 

# F5 Algorithm

• compute Gröbner bases incrementally:

```
\{f_1\}, \{f_1, f_2\}, \{f_1, f_2, f_3\}, \ldots
```

- no Buchberger criterion, account only for:
  - 1)  $f_i f_j f_j f_i = 0$  (trivial syzygies)
  - 2)  $m \in \langle f_1, \ldots, f_{i-1} \rangle$
- assigns a *signature* to leading monomials to efficiently check 2)
- no reductions to zero if  $f_i \not\equiv 0 \mod \langle f_1, \dots, f_{i-1} \rangle$  (regular sequence)