

# A Historic Introduction to Gröbner Bases

Bruno Buchberger

RISC (Research Institute for Symbolic Computation)  
Johannes Kepler University, Linz, Austria

Lecture at School on Gröbner Bases and Applications  
IASBS (Institute for Advanced Studies in Basic Sciences) , Zanzan, Iran,  
July 9-22, 2005

Copyright Bruno Buchberger 2005.

Copying and storing is allowed under the following conditions:

- The file including the copyright note is kept unchanged
- You send an e-mail to [bruno.buchberger@jku.at](mailto:bruno.buchberger@jku.at)
- If you use material from this file, please, cite this lecture appropriately.

# A Quick (Non-historic) Introduction to Gröbner Bases

1965

1965 - 1976

1976 - 2005

Before 1965

# A Quick Introduction to Gröbner Bases

1965

1965 - 1976

1976 - 2005

Before 1965

## Motivation

- Groebner bases have many "nice" properties.
- Based on these nice properties dozens of [\(difficult\) problems](#) turned out to be [reducible](#) to the construction of Gröbner bases. (~ 600 papers, 10 textbooks, entry 13P10 in AMS index).
- For the construction of Gröbner bases we have an [algorithm](#).
- The notion of Gröbner basis and the algorithm is [easy to explain](#).
- However, the proof of the theorem on which the algorithm is based is [non-trivial](#).
- A very [active research](#) area: more efficient algorithms based on more theory, and more applications.

## The Nice Properties of Gröbner Bases

Two **easy-to-read summaries** of the nice properties of Gröbner bases and their applications:

B. Buchberger. Introduction to Gröbner Bases. In: B. Buchberger, F. Winkler (eds.). Gröbner Bases and Applications, London Mathematical Society Lecture Note Series, Vol. 251, Cambridge University Press, 1998, pp.3-31. **This paper will be copied and distributed.**

B. Buchberger. Gröbner-Bases: An Algorithmic Method in Polynomial Ideal Theory. Chapter 6 in: N.K. Bose (ed.), Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems, Reidel Publishing Company, Dordrecht - Boston - Lancaster, 1985, pp. 184-232. (**Second edition:** N.K.Bose (ed.): Multidimensional Systems Theory and Application, Kluwer Academic Publisher, 2003, pp. 89-128.)

### ■ Three Fundamental Properties to Which All Other Properties Can be Reduced Easily

Elimination Property

Canonical Simplification Property

Syzygy Property

### ■ Elimination property: if $F$ is a Groebner basis (w.r.t. lexicographic ordering) then

$$\mathbb{K}[x_1, \dots, x_i] \cap \text{Ideal}[F] = \text{Ideal}[\mathbb{K}[x_1, \dots, x_i] \cap F]$$

Applications:

- Complete solution of algebraic systems.
- Determination of the dimension of an ideal.
- Intersection and quotients of ideals.
- Non-linear syzygies: Given  $f_1, \dots, f_m$  find basis for the ideal  $\{g \mid g(f_1, \dots, f_m) = 0\}$ .
- Implicitization: Given  $f_1, \dots, f_m$  find basis for the ideal whose manifold is  $\{\langle f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n) \rangle \mid x_1, \dots, x_n \in \mathbb{C}\}$
- Invertibility of polynomial mappings.
- etc.

### ■ Canonical simplification property: if $F$ is Groebner basis then

$\text{remainder}_F$  is a canonical simplifier for the equivalence relation  $\equiv_F$ :

Definition:

$$f \equiv_F g \Leftrightarrow f - g \in \text{Ideal}[F]$$

Canonical simplification property:

$$\forall_f \text{ remainder}_F[f] \equiv_F f$$

$$\forall_{f,g} (f \equiv_F g \Leftrightarrow \text{remainder}_F[f] = \text{remainder}_F[g])$$

Corollary (solution of the "main problem of polynomial ideal theory"):

$$\forall_f (f \in \text{Ideal}[F] \Leftrightarrow \text{remainder}_F[f] = 0)$$

Applications:

- Test of  $\text{Ideal}[F] \subseteq \text{Ideal}[G]$  ?
- Test of  $\text{Ideal}[F] = \text{Ideal}[G]$  ?
- Radical membership:  $f \in \text{Radical}[F] \Leftrightarrow 1 \in \text{GB}[F \cup \{y, f-1\}]$
- Computation in  $K[\dots] / \text{Ideal}[F]$  ("Gröbner's" problem)
- Solvability of poly equations:  $F$  unsolvable  $\Leftrightarrow 1 \in \text{GB}[F]$
- Finiteness of manifold of  $F$
- Number of solutions of  $F$  (in case manifold of  $F$  is finite)
- Hilbert function: Value  $H[F,d]$  can be immediately read off the leading power products of the polynomials in  $\text{GB}[F]$ , see my PhD thesis 1965.

## ■ Syzygy property: if $F$ is Groebner basis then

Gail Zacharias 1978, David Spear

then a finite basis of the module of all solutions of the diophantine equation

$$f_1 z_1 + \dots + f_m z_m = 0$$

can be obtained from the finitely many  $S$ -polynomials  $[f_i, f_j]$ :

## The Linear Combination of Polynomials

$$f_1 = -2y + xy$$

$$f_2 = -x^2 + y^2$$

$$-2y + xy$$

$$-x^2 + y^2$$

**Leading power products:** w.r.t. an ordering of the power products (e.g. lexicographically, by total degree or ...)

Consider now the following linear combination of  $f_1$  and  $f_2$ :

$$g = (y) f_1 + (-x + 2) f_2$$

$$y(-2y + xy) + (2 - x)(-x^2 + y^2)$$

$$g = (y) f_1 + (-x + 2) f_2 \quad // \text{ Expand}$$

$$-2x^2 + x^3$$

Observation: The leading power product  $x^3$  of  $g$  is

neither a multiple of the leading power product  $xy$  of  $f_1$

nor a multiple of the leading power product  $y^2$  of  $f_2$ .

⏪

⏩

⏪

⏩

7 of 57

## Definition of Groebner Bases

A set  $F$  of **polynomials** is called a **Groebner basis** (w.r.t. the chosen ordering of power products) iff the above phenomenon cannot happen, i.e. iff

for all  $f_1, \dots, f_m \in F$  and polynomials  $h_1, \dots, h_m$ ,

the leading power product of  $h_1 f_1 + \dots + h_m f_m$

is a multiple of the leading power product of

at least one of the polynomials in  $F$ .

Counterexample: The Set  $F = \{f_1, f_2\}$  of the Above Example is **not** a Groebner basis.

⏪

⏩

⏪

⏩

8 of 57

## Example of a Groebner Basis

The following set G (results from F by adding  $-2x^2 + x^3$ ) is a Groebner basis:

$$G = \{-2x^2 + x^3, -2y + xy, -x^2 + y^2\}$$

$$\{-2x^2 + x^3, -2y + xy, -x^2 + y^2\}$$

For example,

$$(1 + 3y)(-2x^2 + x^3) + (8x + 3xy)(-2y + xy) + (2 - x - y^2)(-x^2 + y^2) \text{ // Expand}$$

$$-4x^2 + 2x^3 - 16xy + 2x^2y + 3x^3y + 2y^2 - 7xy^2 + 4x^2y^2 - y^4$$

$$(1)(-2x^2 + x^3) + (8x)(-2y + xy) + (y)(-x^2 + y^2) \text{ // Expand}$$

$$-2x^2 + x^3 - 16xy + 7x^2y + y^3$$

Why is it [difficult to check](#) whether a given F is a Groebner basis?

How can we check whether a given F is a Groebner basis?

How can we get an "equivalent" Groebner basis G for a given F (which may not be a Groebner basis)?

⏪

⏩

⏪

⏩

9 of 57

## The "Main Theorem" of Gröbner Bases Theory (BB 1965):

$$F \text{ is a Gröbner basis} \iff \forall_{f_1, f_2 \in F} \text{remainder}[F, S\text{-polynomial}[f_1, f_2]] = 0.$$

**Proof:** Nontrivial. Combinatorial. Some details  $\rightarrow$  my lectures on theorem proving.

The theorem [reduces](#) an [infinite](#) check to a [finite](#) check: Recall definition of "F is a Gröbner basis":

for all  $f_1, \dots, f_m \in F$  and polynomials  $h_1, \dots, h_m$ ,

the leading power product of  $h_1 f_1 + \dots + h_m f_m$

is a multiple of the leading power product of at least one of the polynomials in F.

The power of the Gröbner bases method is contained in this theorem and its proof.

⏪ ⏩ ⏴ ⏵

10 of 57

## S-Polynomials

$$f_1 = -2y + xy$$

$$f_2 = -x^2 + y^2$$

$$-2y + xy$$

$$-x^2 + y^2$$

$$\text{s-polynomial}[f_1, f_2] = y f_1 - x f_2$$

$$y(-2y + xy) - x(-x^2 + y^2)$$

$$\text{s-polynomial}[f_1, f_2] = y f_1 - x f_2 \quad // \text{Expand}$$

$$x^3 - 2y^2$$

⏪ ⏩ ⏴ ⏵

11 of 57

## The Algorithm 'remainder'

Roughly,  $\text{remainder}[F, g]$  results from replacing power products in  $g$  by a lower products using the polynomials in  $F$  until no more replacements are possible.

**Example:**  $x < y$

Consider, again,

$$f_1 = -2x^2 + x^3;$$

$$f_2 = -2y + xy;$$

$$f_3 = -x^2 + y^2;$$

$$F = \{f_1, f_2, f_3\};$$

and



$$g = x y - 3 x y^2;$$

A "reduction" ("division") step on g w.r.t. F:

$$g_1 = g + (3 x) f_3$$

$$x y - 3 x y^2 + 3 x (-x^2 + y^2)$$

$$g_1 = g + (3 x) f_3 \text{ // Expand}$$

$$-3 x^3 + x y$$

A next division step w.r.t. F:

$$F = \{f_1, f_2, f_3\}$$

$$\{-2 x^2 + x^3, -2 y + x y, -x^2 + y^2\}$$

$$g_2 = g_1 + (-1) f_2 \text{ // Expand}$$

$$-3 x^3 + 2 y$$

A next division step w.r.t. F:

$$F = \{f_1, f_2, f_3\}$$

$$\{-2 x^2 + x^3, -2 y + x y, -x^2 + y^2\}$$

$$g_3 = g_2 + (3) f_1 \text{ // Expand}$$

$$-6 x^2 + 2 y$$

$$-6 x^2 + 2 y$$

This is the remainder of the division of g w.r.t. F because ...

## Remainder Algorithms are Available in all Math Systems

```
PolynomialReduce[g, F, {y, x}]
```

```
{{0, 1 - 3 y, -6}, -6 x^2 + 2 y}
```

⏮

⏪

⏩

⏭

13 of 57

## Now We Can *Check* Groebnerianity

Let's again look to the above example:

```
F = {f1, f2, f3}
```

```
{-2 x^2 + x^3, -2 y + x y, -x^2 + y^2}
```

```
PolynomialReduce[f1 y - f2 x^2, F, {y, x}]
```

```
{{0, 0, 0}, 0}
```

```
PolynomialReduce[f1 y^2 - f3 x^3, F, {y, x}]
```

```
{{4 + 2 x + x^2, -4 y - 2 x y, -8}, 0}
```

```
PolynomialReduce[f2 y - f3 x, F, {y, x}]
```

```
{{1, 0, -2}, 0}
```

⏮

⏪

⏩

⏭

14 of 57

## The Problem of *Constructing* Gröbner Bases

Given  $F$ , find  $G$  s.t.  $\text{Ideal}(F) = \text{Ideal}(G)$  and  $G$  is a Gröbner basis.

(Ideal(F) := the set of all linear combinations  $h_1 f_1 + \dots + h_m f_m$

with  $f_1, \dots, f_m \in F$  and  $h_1, \dots, h_m$  arbitrary polynomials.)

⏪ ⏩ ⏴ ⏵

15 of 57

## An Algorithm for *Constructing* Gröbner Bases (BB 1965)

Recall the main theorem:

$$F \text{ is a Gröbner basis} \iff \forall_{f_1, f_2 \in F} \text{remainder}[F, \text{S-polynomial}[f_1, f_2]] = 0.$$

Based on the main theorem, the problem can be solved by the following algorithm:

Start with  $G := F$ .

For any pair of polynomials  $f_1, f_2 \in G$ :

$h := \text{remainder}[G, \text{S-polynomial}[f_1, f_2]]$

If  $h = 0$ , consider the next pair.

If  $h \neq 0$ , add  $h$  to  $G$  and iterate.

⏪ ⏩ ⏴ ⏵

16 of 57

## Correctness and Termination of the Algorithm

Correctness: Easy as soon as we know the main theorem.

Termination: by Dickson's Lemma (Dickson 1913, BB 1970).

⏪ ⏩ ⏴ ⏵

17 of 57

## Specializations

The [Gröbner bases algorithm](#),

for linear polynomials, specializes to [Gauss' algorithm](#), and

for univariate polynomials, specializes to [Euclid's algorithm](#).

⏪

⏩

⏴

⏵

18 of 57

## Example

Let's again look at

$$\begin{aligned} f_1 &= -2y + xy \\ f_2 &= -x^2 + y^2 \end{aligned}$$

$$-2y + xy$$

$$-x^2 + y^2$$

$$F = \{f_1, f_2\}$$

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

$F$  is not a Groebner basis.

The S-polynomial of  $f_1, f_2$ :

$$\text{s-polynomial}[f_1, f_2] = y f_1 - x f_2 \quad // \text{Expand}$$

$$x^3 - 2y^2$$

Its remainder w.r.t.  $F$  is:

$$-2x^2 + x^3.$$

All the other S-polynomials have remainder 0. Hence, we arrived at a Groebner basis.

The Groebner basis algorithm is now available in all math software systems, e.g. in *Mathematica*:

```
G = GroebnerBasis[F, {y, x}]
```

```
{-2 x^2 + x^3, -2 y + x y, -x^2 + y^2}
```

Navigation icons: back, forward, search, etc.

19 of 57

## Application: Solving Polynomial Systems

Is based on the Theorem that, w.r.t. lexicographic ordering, Gröbner bases are "triangular" (BB 1970, Trinks 1978).

Navigation icons: back, forward, search, etc.

20 of 57

## A Simple System of Equations

$$f_1 = -2y + xy$$

$$f_2 = -x^2 + y^2$$

$$-2y + xy$$

$$-x^2 + y^2$$

Find x, y such that

$$-2y + xy = 0$$

$$-x^2 + y^2 = 0$$

We compute

```
G = GroebnerBasis[F, {y, x}]
```

```
{-2 x^2 + x^3, -2 y + x y, -x^2 + y^2}
```

```
Solve[-2 x^2 + x^3 == 0, x]
```

```
{{x -> 0}, {x -> 0}, {x -> 2}}
```

```
{G[[2]], G[[3]]} /. {x -> 2}
```

```
{0, -4 + y^2}
```

```
Solve[-4 + y^2 == 0, y]
```

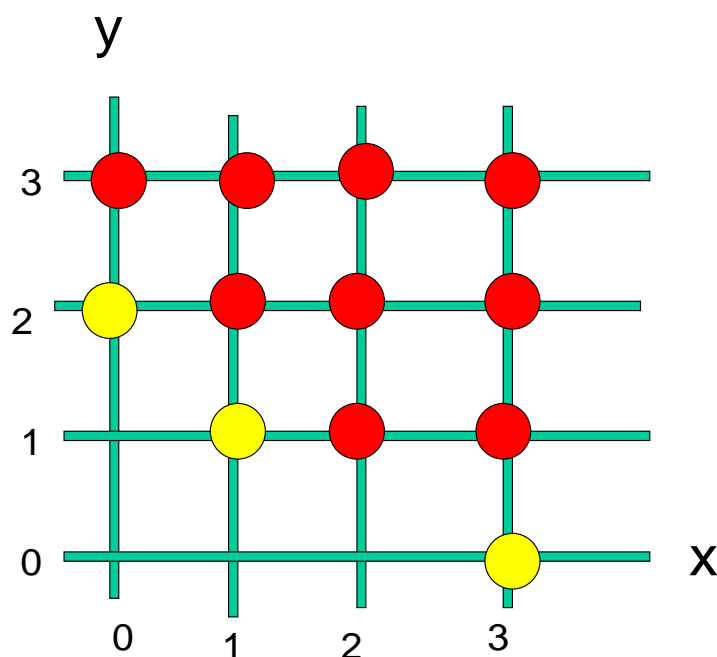
```
{{y -> -2}, {y -> 2}}
```

All this is already implemented in the *Mathematica* general Solve function:

```
Solve[{f1 == 0, f2 == 0}, {x, y}]
```

```
{{x -> 0, y -> 0}, {x -> 0, y -> 0}, {x -> 2, y -> -2}, {x -> 2, y -> 2}}
```

**Theorem (BB):** Solvability and the number of solutions can be predicted from the form of the Groebner basis.



## A More Complicated System of Equations

$$f_1 = xy - 2yz - z;$$

$$f_2 = y^2 - x^2z + xz;$$

$$f_3 = z^2 - y^2x + x;$$

$$F = \{f_1, f_2, f_3\};$$

```
{time, G} = GroebnerBasis[F] // Timing
```

```
{0. Second,
 {z + 4 z^3 - 17 z^4 + 3 z^5 - 45 z^6 + 60 z^7 - 29 z^8 + 124 z^9 - 48 z^10 + 64 z^11 - 64 z^12,
 -22001 z + 14361 y z + 16681 z^2 + 26380 z^3 + 226657 z^4 + 11085 z^5 -
 90346 z^6 - 472018 z^7 - 520424 z^8 - 139296 z^9 - 150784 z^10 + 490368 z^11,
 43083 y^2 - 11821 z + 267025 z^2 - 583085 z^3 + 663460 z^4 - 2288350 z^5 +
 2466820 z^6 - 3008257 z^7 + 4611948 z^8 - 2592304 z^9 + 2672704 z^10 - 1686848 z^11,
 43083 x - 118717 z + 69484 z^2 + 402334 z^3 + 409939 z^4 + 1202033 z^5 -
 2475608 z^6 + 354746 z^7 - 6049080 z^8 + 2269472 z^9 - 3106688 z^10 + 3442816 z^11}}
```

```
zsolexact1 = Solve[G[[1]] == 0, z]
```

```
{{z -> 0},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 - 124 #1^8 + 48 #1^9 -
 64 #1^10 + 64 #1^11 &, 1]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 2]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 3]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 4]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 5]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 6]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 7]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 8]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 9]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 10]},
 {z -> Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 11]}}
```

One can compute with these "abstract roots" like with "square roots". For example

```
G0 = G /. zsolexact1[[2]] // Simplify
```

```
{Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 -
 60 #1^6 + 29 #1^7 - 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
 4 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^3 -
 17 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^4 +
 3 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^5 -
```

```

45 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^6 +
60 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^7 -
29 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^8 +
124 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^9 -
48 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^10 +
64 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^11 -
64 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^12,
-22001 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
14361 y Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
16681 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^2 +
26380 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^3 +
226657 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^4 +
11085 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^5 -
90346 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^6 -
472018 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^7 -
520424 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^8 -
139296 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^9 -
150784 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^10 +
490368 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^11,
43083 y^2 - 11821 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 +
      29 #1^7 - 124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1] +
267025 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^2 -
583085 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^3 +
663460 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^4 -
2288350 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -
      124 #1^8 + 48 #1^9 - 64 #1^10 + 64 #1^11 &, 1]^5 +
2466820 Root[-1 - 4 #1^2 + 17 #1^3 - 3 #1^4 + 45 #1^5 - 60 #1^6 + 29 #1^7 -

```



```

124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]6 -
3008257 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]7 +
4611948 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]8 -
2592304 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]9 +
2672704 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]10 -
1686848 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]11,
43083 x - 118717 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 +
29 #17 - 124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1] +
69484 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]2 +
402334 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]3 +
409939 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]4 +
1202033 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]5 -
2475608 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]6 +
354746 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]7 -
6049080 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]8 +
2269472 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]9 -
3106688 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]10 +
3442816 Root[-1 - 4 #12 + 17 #13 - 3 #14 + 45 #15 - 60 #16 + 29 #17 -
124 #18 + 48 #19 - 64 #110 + 64 #111 &, 1]11}

```

G0 /. {Root[-1 - 4 #1<sup>2</sup> + 17 #1<sup>3</sup> - 3 #1<sup>4</sup> + 45 #1<sup>5</sup> -  
60 #1<sup>6</sup> + 29 #1<sup>7</sup> - 124 #1<sup>8</sup> + 48 #1<sup>9</sup> - 64 #1<sup>10</sup> + 64 #1<sup>11</sup> &, 1] → α}

```

{α + 4 α3 - 17 α4 + 3 α5 - 45 α6 + 60 α7 - 29 α8 + 124 α9 - 48 α10 + 64 α11 - 64 α12,
-22001 α + 14361 y α + 16681 α2 + 26380 α3 + 226657 α4 + 11085 α5 -
90346 α6 - 472018 α7 - 520424 α8 - 139296 α9 - 150784 α10 + 490368 α11,
43083 y2 - 11821 α + 267025 α2 - 583085 α3 + 663460 α4 - 2288350 α5 +
2466820 α6 - 3008257 α7 + 4611948 α8 - 2592304 α9 + 2672704 α10 - 1686848 α11,
43083 x - 118717 α + 69484 α2 + 402334 α3 + 409939 α4 + 1202033 α5 -
2475608 α6 + 354746 α7 - 6049080 α8 + 2269472 α9 - 3106688 α10 + 3442816 α11}

```

No more details in this talk!

Alternatively, compute numerically:

```
zsol = NSolve[G[[1]] == 0, z]
```

```
{ {z → -0.331304 - 0.586934 i}, {z → -0.331304 + 0.586934 i},
  {z → -0.296413 - 0.705329 i}, {z → -0.296413 + 0.705329 i},
  {z → -0.163124 - 0.37694 i}, {z → -0.163124 + 0.37694 i},
  {z → 0.}, {z → 0.0248919 - 0.89178 i}, {z → 0.0248919 + 0.89178 i},
  {z → 0.468852}, {z → 0.670231}, {z → 1.39282} }
```

```
Gsubnum = G /. zsol[[1]]
```

```
{1.33227 × 10-15 + 9.71445 × 10-17 i,
 (-523.519 - 4967.65 i) - (4757.86 + 8428.97 i) y,
 (-7846.9 - 8372.06 i) + 43083 y2, (-16311.7 + 16611. i) + 43083 x}
```

```
PolynomialGCD[Gsubnum[[2]], Gsubnum[[3]]]
```

```
1
```

```
ysol = NSolve[Gsubnum[[2]] == 0, y]
```

```
{{y → -0.473535 - 0.205184 i}}
```

```
ysol = NSolve[Gsubnum[[3]] == 0, y]
```

```
{{y → -0.473535 - 0.205184 i}, {y → 0.473535 + 0.205184 i}}
```

**Theorem** (Roider, Kalkbrener et al. 1990): It suffices to consider the poly in  $y$  with lowest degree.

```
xsol = NSolve[Gsubnum[[4]] == 0, x]
```

```
{{x → 0.378611 - 0.385558 i}}
```

```
F /. zsol[[1]] /. ysol[[1]] /. xsol[[1]]
```

```
{-1.88738 × 10-15 + 2.35922 × 10-16 i,
 1.02696 × 10-15 + 1.52656 × 10-16 i, -1.94289 × 10-15 - 1.11022 × 10-16 i}
```

# A Quick Introduction to Gröbner Bases

1965

1965 - 1976

1976 - 2005

Before 1965

---

## My PhD Thesis Work 1964/65

My PhD thesis work in which I introduced Gröbner bases theory.

Professor Wolfgang Gröbner was my thesis supervisor.

## The Mathematical Personality of Wolfgang Groebner (1899-1980)

Professor of mathematics at the University of Innsbruck, Austria.

One of "old masters" who managed to integrate, in their mathematical personality,

- \* algebra
- \* analysis
- \* geometry.

Always with a view towards "how one can compute things in examples".



25 of 57

## Gröbner's Seminar

1994, I "randomly" took the following problem from his seminar as my PhD topic:

Given:  $F$ .

Find: A linearly independent vector space basis for  $R := K[x_1, \dots, x_n] / \text{Ideal}(F)$

and the "structur constants" for multiplication in the associative algebra  $R$ .

Gröbner wanted a solution that can be ["programmed on a computing machine"](#).



26 of 57

## Gröbner's Proposal for a Procedure

Groebner proposed the following ["algorithm"](#) (for extracting a linearly independent basis from the basis of all power products):

- Store the polys in the basis as linear representation of the leading power products in terms of the others.

- Consider successively all power products starting from 1:

Decompose them into factors of power products in the present vector basis in all possible ways.

If this is not possible, keep the power product in the vector basis.

If this is possible, use the available representations for multiplication.

If there is more than one factorization and the resulting representations are different,  
add the difference of the results as a new representation.

- Iterate.



27 of 57

## Gröbner's Question to Me

The main question: When (if ever) can we [terminate](#) this procedure?

He said that there was ["no literature on this subject"](#).



28 of 57

## The Contents of My Thesis

I soon started to program Gröbner's for obtaining intuition from experiments. (I earned my living as a full time programmer at the Computing Center of the U of Innsbruck.)

At the same time I tried to prove certain first observations about the procedure, e.g. the correctness of a forerunner of the "product criterion".

After some time (of struggle and suffering), I got the intuition about what I called ["S-polynomials"](#) and how a complete algorithmic solution can be based on them.

I proved the essential [theorem](#) on the S-polynomials and formulated a real [algorithm](#) for the solution of Gröbner's problem in terms of the S-polynomials. Also, I implemented the algorithm.

Finally, in Dec.1995, I submitted my [PhD thesis](#):

B.B.

Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal).

PhD Thesis, Mathematical Institute, University of Innsbruck, Austria, 1965. (English translation by M. Abramson and R. Lumbert, to appear in Journal of Symbolic Computation, 2005)

⏪ ⏩ ⏴ ⏵

29 of 57

The contents of my PhD thesis1965:

- \* the [concept](#) of Groebner bases and reduced Groebner bases  
(but with "no name", rather characterized by the property that all polys in the  $\text{Ideal}(F)$  can be reduced to 0 w.r.t.  $F$ )
- \* the [S-polynomials](#)
- \* [the main theorem \(characterization of GB by S-polynomials\) with proof](#)
- \* the [algorithm](#) for computing GB based on S-polynomials
- \* first [applications](#) (computing in residue rings, a general closed form formula for the Hilbert function, solution of Gröbner's original problem )
- \* a complete running [implementation](#) on the ZUSE Z 23 V
- \* first [complexity](#) considerations (in the bivariate case).

The thesis did not yet contain a termination proof (which I gave only in the published version of my PhD thesis in aequ. math. 1970, see later) but a [termination](#) criterion.

In my thesis, I did not dare to say that, in fact, I finally proposed an algorithm different from the "procedure" proposed by Groebner but just called my algorithm a "variant" of Groebner's proposal.

## ■ Solution to Groebner's problem

Given:  $F$ . Find: a linearly independent vector space basis for  $K[\dots]/\text{Ideal}[F]$ .

Answer: If  $F$  is a Groebner basis then the set  $\{p \mid \exists_{f \in F} \text{lpp}[f] \text{ divides } p\}$  is a linearly independent vector space basis for  $K[\dots]/\text{Ideal}[F]$ .

⏪ ⏩ ⏴ ⏵

30 of 57

## Details About the First Implementation

One implementation in machine code and one in the "Formeluebersetzer" for the Zuse Z 23V.

256 memory cells (40 bits) in core memory and 8192 memory cells on a drum.

1 integer addition needed approx. 0.3 msec.

The complete flow charts and the programs are in my thesis.

31 of 57

## An Example Computed in 1965

The example Groebner gave me:

$$\begin{aligned} f_1 &= x_3^2 - \frac{1}{2} x_1^2 - \frac{1}{2} x_2^2 \\ f_2 &= x_1 x_3 - 2 x_3 + x_1 x_2 \\ f_3 &= x_1^2 - x_2 \end{aligned}$$

$$-\frac{x_1^2}{2} - \frac{x_2^2}{2} + x_3^2$$

$$x_1 x_2 - 2 x_3 + x_1 x_3$$

$$x_1^2 - x_2$$

$$\mathbf{F} = \{f_1, f_2, f_3\}$$

$$\left\{ -\frac{x_1^2}{2} - \frac{x_2^2}{2} + x_3^2, x_1 x_2 - 2 x_3 + x_1 x_3, x_1^2 - x_2 \right\}$$

It took 2.5 minutes on the Zuse Z 23V.

```
GroebnerBasis[F, {x3, x2, x1}, MonomialOrder ->
DegreeReverseLexicographic] // Timing
```

```
{0.01 Second,
 {x1^2 - x2, x1 x2 - 2 x3 + x1 x3, -2 x1 x2 - x2^2 + 4 x3 - x2 x3, -x2 - x2^2 + 2 x3^2,
 2 x2 + 7 x1 x2 + 6 x2^2 + x1 x2^2 - 16 x3, 12 x2 + 24 x1 x2 + 29 x2^2 - x2^3 - 64 x3}}
```

On the laptop I currently use (IBM Thinkpad T42), the *Mathematica* computation (which is actually C code) is approx. 1000 times faster.

The laptop is approx. 5000 times faster than the Zuse Z 23V.

If you code the GB algorithm using *Mathematica* (or any high-level language) as a programming language, the computation of the above example, takes again in the range of 30 seconds.

N. Wirth: "The software becomes faster slower than the hardware becomes faster."

If you are interested in this discussion, see

B. Buchberger. 1991. Groebner Bases in Mathematica: Enthusiams and Frustration. Proc. of IFIP Working Conference on Programming Environments for High-Level Scientific Problem Solving, 23-27 September, Karlsruhe, pp. 80-91.

⏮ ⏪ ⏩ ⏭

32 of 57

---

## Groebner's Reaction

Indirectly, I concluded that my thesis must have been accepted because I was admitted to the final exam (the PhD "Rigorosum").

However, Groebner never told me whether he was satisfied with my work. In particular, he did not tell me to publish the results. And I have good reasons to believe that he never read my thesis.

Hence, I was very frustrated and shifted my interest to recursive funtion theory.

⏮ ⏪ ⏩ ⏭

33 of 57



1965

1965 - 1976

1976 - 2005

Before 1965



34 of 57

## The Reaction of the Community

[Mathematicians](#), at that time, were not at all interested in algorithmic problems. (They even did not know what an algorithm is.)

The reason for this: Some mathematicians do not understand that non-trivial algorithms are based on non-trivial theorems.

[Computer science](#) did not yet exist. Rather, there were "programmers" coming from various background. They were, in general, not interested in mathematics or only in numerical mathematics.

(Frankly, I guess the situation did not change significantly.)



35 of 57

---

## My Aequationes Mathematicae Paper

Finally, Gerhard Wanner, a fellow assistant working in numerics (now professor at Geneve), encouraged me to publish my thesis. (He was mainly impressed by the fact that my algorithm generalized both Gauss' and Euclid's algorithm.)

Thus, I wrote a paper on the contents of my PhD thesis and sent it to *aequationes mathematicae*.

First attempt (in English): 1969. Editor Ostrowski: "Already many mathematicians stumbled over seemingly simple problems involving polynomials. Publish in German! Revise!"

Second attempt (in German): 1969. Anonymous referee: "A valuable contribution to our knowledge about polynomial ideals. Accept. And: Cite Hermann 1926."

My paper appeared in 1970:

B. B.

Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems (An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations). *Aequationes mathematicae* 4/3, 1970, pp. 374-383. (English translation by M. Abramson in: Buchberger, Winkler eds., *Gröbner Bases and Applications*, London Math. Society Lecture Notes Series, Vol. 251, Cambridge University Presse, Cambridge, 1998, pp. 535 -545.)

---

## Contents of My Aequationes Mathematicae Paper

In addition to what I had in the thesis, the *aequ.math.* paper contained:

- \* concentration on the notion of Groebner basis (still not with this name),
- \* the main theorem (on S-polys) and the algorithm based on the main theorem,
- \* termination proof in the general case,
- \* solvability, solution cardinality, and complete solutions of algebraic equations, (by "base change").

⏮ ⏪ ⏩ ⏭

37 of 57

## My Work 1970 - 1974

Various re-implementations for testing out new machines at the U of Innsbruck.

⏮ ⏪ ⏩ ⏭

38 of 57

## The First Reaction 1974

Wolfgang Trinks (Univ.of Karlsruhe) was the first who found interest in my paper and sent his student, R. Schrader, to me for learning all details. Re-implementation in ALGOL.

From this, W.Trink's paper (1978) resulted, which introduced

- \* the observation that the results remain valid if we use arbitrary "admissible" orderings,
- \* the "elimination property" of Groebner bases in case we use lexical orderings.

W. Trinks

On Buchberger's Method for Solving Systems of Algebraic Equations (German). J. Number Theory, Vol. 10, 1978, pp. 475-488.

⏮ ⏪ ⏩ ⏭

39 of 57

## Loos' Reaction 1976

Rüdiger Loos - a physicist who had started to work in computer algebra under the influence of George Collins - had conjectured, in a SYMSAC paper 1974, that the poly ideal congruence problem is undecidable.

When I met him 1976 at the U of Kaiserslautern, he was electrified by my results (the real story must be explained orally), immediately announced them at the next SYMSAC 76, and [this started an explosion of interest](#) in Groebner bases both among the symbolic computation community and among algebraists.

This motivated me very much to resume work on the subject and I wrote a paper in the SIGSAM Bull., 1976, in which I gave a new proof and I also introduced the name "[Groebner bases](#)" for the main object of study.

My reason for this: I now understood that Groebner had given me more in initiating my thesis [by a problem and a first idea](#) for the solution than somebody who supervises students by day-by-day cultivation. (However, Groebner did not read the new papers either.)

1965

1965 - 1976

1976 - 2005

Before 1965

---

## Research Effort 1976-1987

Many people started to do research on Groebner bases theory resulting in approximately 300 research papers, see the other lectures at the school.

I did my share in this effort mainly with the following contributions:

- \* the notion and usage of criteria
- \* the abstract notion of reduction rings
- \* complexity for the bivariate case
- \* relation to rewriting
- \* stream-lining of the proof for the main theorem

---

## Research Effort 1987-2005

Resulting in several hundred papers:

- Sophisticated research on high-level mathematical [problems that may be attacked using Groebner bases](#) as black-box tools for subproblems in many different fields of mathematics (not only commutative algebra).
- [Generalization](#) of the Gröbner bases approach to more and more algebraic domains.
- [Implementation](#) of the method routinely in all [general](#) computer algebra software systems (like *Mathematica*, Maple, Derive, etc.)
- [Special systems](#) for algorithms in algebraic geometry etc., which are heavily based on the Gröbner bases methodology (like Macaulay, CoCoA, Singular, RISA-ASIR, etc.)
- [Speeding up](#) the algorithm by various techniques: Gröbner "walk", linear algebra for reduction, numerics. (However, so far, all algorithms ultimately are based on S-polynomials and the corresponding theorem!)
- [Textbooks](#): by now, approx. 10 textbooks on Gröbner bases.

See the other lectures in the school.

I did not contribute to Gröbner bases research in recent years mainly for two reasons:

- \* I put a lot of energy into building up an international research facility for computer algebra (symbolic computation) research ([RISC](#)), the [Journal of Symbolic Computation](#), and the [Softwarepark Hagenberg](#).

- \* I became more and more interested in new algorithms for [automating the process of mathematical theory exploration](#).

- \* In particular, I became interested in [the intellectual process of algorithm invention](#). (See my subsequent lectures in this school.)

⏮ ⏪ ⏩ ⏭

44 of 57

1965

1965 - 1976

1976 - 2005

Before 1965

⏮ ⏪ ⏩ ⏭

45 of 57

## Euclid

Non-linear univariate systems.

Euclid's algorithm turns out to be a special case of the Gröbner bases algorithm.

⏮ ⏪ ⏩ ⏭

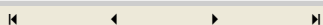
46 of 57

---

## C. F. Gauss

Linear multivariate systems.

Gauss's algorithm turns out to be a special case of the Gröbner bases algorithm.



47 of 57

---

## Division Process in 19th Century

A kind of division process for multivariate polynomials.



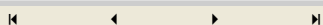
48 of 57

---

## P. Gordan 1899

Introduced (in the language of that time) the notion of Gröbner bases (based on the division process) and showed their existence (finiteness?) but did not give an algorithm for constructing them.

(A non-constructive proof of Gröbner bases is easy! Even of finite Gröbner bases. How?)



49 of 57

---

## L. E. Dickson 1913

A general lemma on the termination of certain sequences of tuples of natural numbers.

From this lemma, the termination of the Gröbner bases algorithm follows easily. And, then, Hilbert's basis theorem is an easy corollary of the existence of Gröbner bases.



50 of 57

---

## **M. Janet 1920**

Similar notions and methods for differential equations (viewed symbolically).

---

## **G. Hermann 1926**

A solution of the membership problem for polynomial ideals based on linear algebra.

Why did Gröbner not mention this paper to me? !



51 of 57

---


## **W. Gröbner 1954 (minus 17)**

The method he showed me in his seminar 1964 was already published in one of his papers (1954): "I am using this method already for 17 years..."

---

## **Ritt**

Characteristic sets. Wu 1978.



52 of 57

---

## **A.I. Shirshov 1962**

Similar notion and method for Lie algebras. M. Abramson.



53 of 57



---

## H. Hironaka 1964

In the frame of his famous paper on the resolution of singularities: the notion of "standard bases" in the domain of formal power series, but no algorithm for their construction.

⏮ ⏪ ⏩ ⏭

54 of 57

---

## .... others

⏮ ⏪ ⏩ ⏭

55 of 57

## Announcement:

---

## PhD Study at RISC

⏮ ⏪ ⏩ ⏭

56 of 57



## RISC (Research Institute for Symbolic Computation)

an institute of the Johannes Kepler University (JKU) in Linz, Austria

JKU has 15.000 students (3000 in computer science and math)

Linz: 200 km from Vienna, 100 km to Salzburg, industrial town, 200.000 inhabitants

3 full professors: B. B., Franz Winkler, Peter Paule

5 associate professors, 5 assistant professors

a couple of postdocs

approx. 20 PhD students from many different countries

working language: English

we concentrate on basic research in SC (computer algebra, computational logic, ...)

in education we concentrate on PhD study, curriculum: ~ 30 special courses in SC,

Thinking, Speaking, Writing course

we also emphasize application: RISC software company, Softwarepark Hagenberg (700 employees)

close interaction with Radon Institute for Computational and Applied Mathematics (Austrian Academy of Science)

## Application for PhD Study

send to [bruno.buchberger@jku.at](mailto:bruno.buchberger@jku.at)

detailed CV

list of all courses since undergraduate: title of course, credit hours, grade

2-3 recommendation letters from professors who know you

English (TOEFL)

interview over phone

prerequisite: master's degree

only for students who want to stay the entire period of PhD work (3-5 years)

1000 Euro per month

in coordination with your home institution

## Softwarepark Hagenberg

