



# Solving Equations at the Naval Academy

**Will Traves**  
**MHPCC June 28, 2002**

# United States Naval Academy

Highly selective 4-year college whose mission is to develop young men and women morally, mentally and physically for a career in the Navy or the Marine Corps.

Faculty:  $\frac{1}{2}$  civilian faculty with Ph.D.'s and  $\frac{1}{2}$  military faculty with significant fleet experience. Faculty research efforts are well supported and collaboration with DoD labs is encouraged.

Rickover's legacy: Strong engineering and science majors.





## CSE at USNA

Center for Computational Science and Engineering (CSE):

- Multidisciplinary organization promoting the use of computation
- Enhances faculty and midshipmen research and supports allied undergraduate programs
- Course: Introduction to Computational Science and Engineering
  - Team-taught by Economics, Engineering, CS and Math
  - Topics: Neural Nets, Monte Carlo Methods, Target Tracking, CFD, Parameter Estimation, Non-Linear Dynamical Systems
- Other Research Interests: (Combat) Simulation, Wavelets, etc.

# Trident Scholar Program

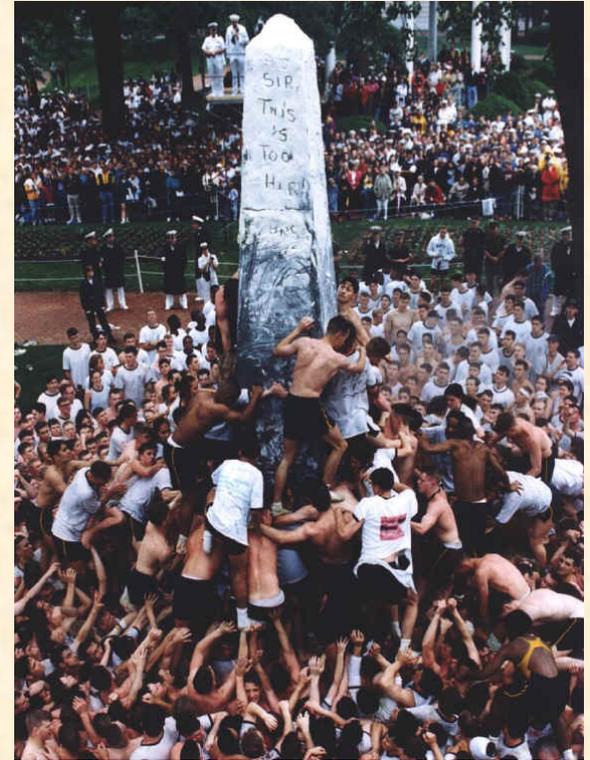
USNA's Trident Scholar Program provides an opportunity for select students to engage in independent study and research during their senior year.

Students submit research proposals in their junior year and collaborate with faculty and DoD scientists throughout the project.

Up to 24 credit-hours are assigned to the project.

Midn 1/c David Zane: Efficient Course and Exam Scheduling

Midn 1/c Matt Ahlert: Wavelets & the Galerkin Method for Acoustics



# Polynomial Equations

Polynomial equations arise in nearly every scientific activity.

Grobner Bases, a tool for dealing with such equations, find application in:

- CAGD (Describing Intersections)
- Computer Graphics (Implicitization)
- Computer-Aided Proof Systems (Database applications)
- Integer Programming (Oil extraction applications)
- Graph Coloring and Scheduling Problems
- Coding Theory
- Molecular Conformations
- Robotics

Pixar used CAGD to create Toy Story.



# Robotics



Canadarm2 will have 15 joints and a mechanical hand.

Robot arm has joints connected by fixed arms.

The hand configuration (location, attitude, etc.) is described by  $n$  parameters, each a function of the joint angles.

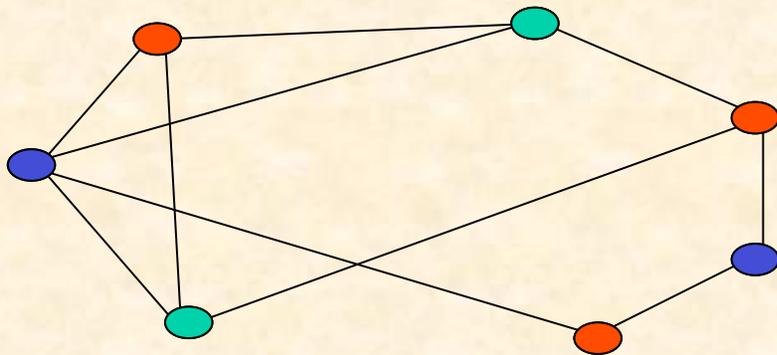
Inverse Kinematics Problem: Find angles of the joints to put the hand in a given position.

Letting  $s_i = \sin(\theta_i)$  and  $c_i = \cos(\theta_i)$ , we get a system of polynomial equations in the  $s_i$  and  $c_i$ .

If the robot has  $n$  joints there will be a finite number of joint arrangements solving the equations.

# Graph Coloring and Scheduling

Many types of scheduling problems can be modeled as both a graph coloring problem and a system of polynomial equations.



Example: Each dot represents a person, edges represent animosities.

Can we seat everyone at 3 tables so that no one sits with an enemy?  
(i.e. is the graph 3-colorable?)

$$x_i^3 - 1 = 0 \quad (1 \leq i \leq 7) \longleftarrow x_i \text{ is a 3}^{\text{rd}} \text{ root of unity.}$$

$$x_i^2 + x_i x_j + x_j^2 = 0 \quad (i \text{ and } j \text{ joined}) \longleftarrow x_i \text{ not equal to } x_j$$

System has a solution if and only if the Graph is 3-colorable.

# Computer Graphics (Implicitization)

There are two ways to represent the hyperbola:

- Parametric:  $x(t) = \frac{1+t^2}{1-t^2}$      $y(t) = \frac{2t}{1-t^2}$ .

- Good for curve tracing.
- Hard to tell if a point is on the curve.

- Implicit:  $x^2 - y^2 = 1$ .

- Poor for plotting but easy to tell whether a point  $(x,y)$  is on the curve.

Grobner Bases help us go from parametric to implicit form.

# Introduction to Grobner Bases

A Grobner Basis is a representation of a system of equations that may contain redundant equations, but that encapsulates a lot of information about the system.



Bruno Buchberger named Grobner Bases after his Ph.D. advisor, Wolfgang Grobner.

1. Ideals
2. Lex Order
3. Division
4. Leading Term Ideal
5. Buchberger Criterion
6. Elimination Theory
7. Solving Equations
8. Implicitization
9. Computational Issues

# Introduction to Grobner Bases

A Grobner Basis is a representation of a system of equations that may contain redundant equations, but that encapsulates a lot of information about the system.

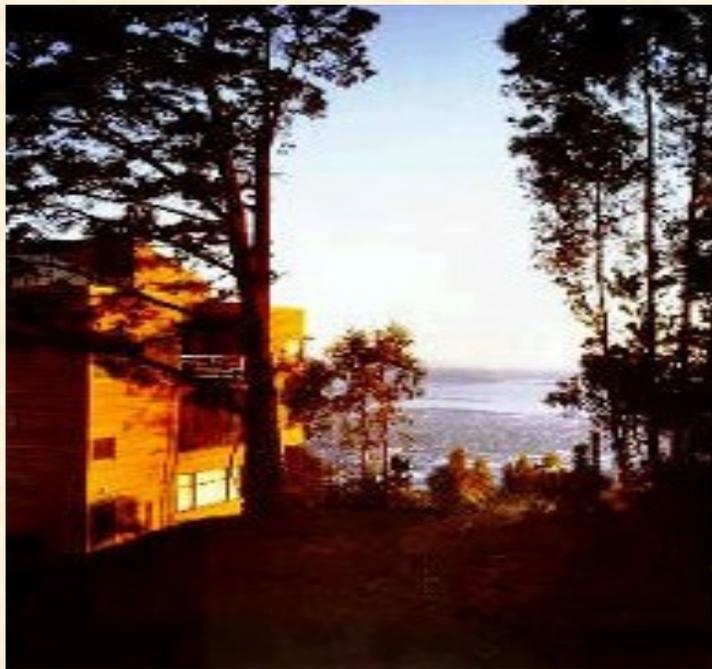


RISC is located in a castle in Austria. It was donated by the Austrian government to recognize industrial applications of mathematics.

1. Ideals
2. Lex Order
3. Division
4. Leading Term Ideal
5. Buchberger Criterion
6. Elimination Theory
7. Solving Equations
8. Implicitization
9. Computational Issues

# Introduction to Grobner Bases

A Grobner Basis is a representation of a system of equations that may contain redundant equations, but that encapsulates a lot of information about the system.



MSRI (Berkeley) is holding a year-long program in Commutative Algebra 2002-03.

1. Ideals
2. Lex Order
3. Division
4. Leading Term Ideal
5. Buchberger Criterion
6. Elimination Theory
7. Solving Equations
8. Implicitization
9. Computational Issues

# Ideals

Let  $C[x_1, \dots, x_n]$  be the set of complex polynomials in  $n$  variables.

For polynomials  $f_1, \dots, f_s$  we have the system of equations

$$f_1(x_1, \dots, x_n) = 0, \dots, f_s(x_1, \dots, x_n) = 0. \quad (*)$$

A **polynomial consequence** is a polynomial  $g_1 f_1 + g_2 f_2 + \dots + g_s f_s$ .

The **ideal**  $(f_1, \dots, f_s)$  is the set of all polynomial consequences of the  $f_i$ 's.

 The  $f_i$ 's are called basis elements for the ideal.

Solving  $(*)$  is equivalent to solving the system

$$\{g=0: g \text{ a polynomial consequence of } f_1, \dots, f_s\}.$$

Theorem:  $1$  is in  $(f_1, \dots, f_s)$  if and only if there are no solutions to  $(*)$ .

Question: When is  $f$  in  $(f_1, \dots, f_s)$ ?

# Lex Order

In order to answer this question and to construct Grobner Bases, we introduce an ordering on the monomials  $x^a y^b z^c$  in  $C[x,y,z]$ .

We say that  
if

$$x^a y^b z^c > x^d y^e z^f$$

$$a > d \text{ or}$$

$$a=d \text{ and } b > e \text{ or}$$

$$a=d \text{ and } b=e \text{ and } c > f.$$

This says that the bigger monomial is determined by degree in the left-most variables (eg.  $x^2 > xyz$ ).

The leading term of a polynomial is the largest monomial in the sum:

$$LT(5xy+3x^2) = 3x^2.$$

# Division Process

Can use Leading Terms to “divide”  $f = xy^2 - x$  by  $xy+1$  and  $y^2-1$ .

```
r=0
While f != 0
  if LT(fi) divides LT(f) then
    subtract a multiple of
    fi from f to kill LT(f).
  else
    r=r+LT(f)
    f=f-LT(f)
  endif
End
Return(r)
```

LT( $xy+1$ ) divides LT( $xy^2-x$ )

so change  $f = xy^2-x$  to

$$(xy^2-x) - y(xy+1) = -x-y.$$

Since LT( $-x-y$ ) is not divisible by  
LT( $xy+1$ ) or LT( $y^2-1$ ), add  $-x$  to  $r$ .

Now  $f = -y$ .

Since LT( $-y$ ) is not divisible by  
LT( $xy+1$ ) or LT( $y^2-1$ ), add  $-y$  to  $r$ .

Now  $f = 0$ . Output  $r = -x-y$ .

# Division Process

Can use Leading Terms to “divide”  $f = xy^2 - x$  by  $xy+1$  and  $y^2-1$ .

But  $f = xy^2 - x = x(y^2 - 1)$  is in the ideal  $(xy + 1, y^2 - 1)$ .

It would be nice if the remainder was zero when  $f$  is in the ideal.

We can correct this by using a “good” set of generators for our ideal – a Grobner Basis.

$LT(xy+1)$  divides  $LT(xy^2-x)$   
so change  $f = xy^2-x$  to  
 $(xy^2-x) - y(xy+1) = -x-y$ .

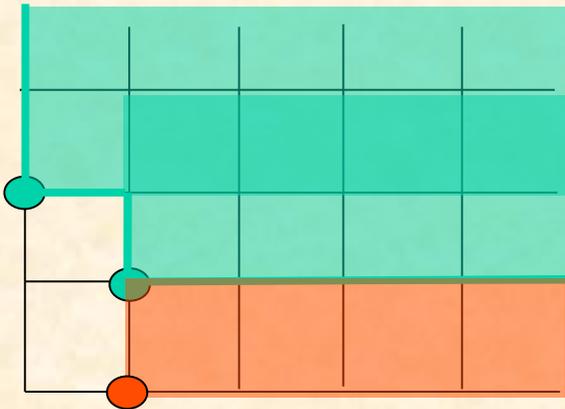
Since  $LT(-x-y)$  is not divisible by  $LT(xy+1)$  or  $LT(y^2-1)$ , add  $-x$  to  $r$ .  
Now  $f = -y$ .

Since  $LT(-y)$  is not divisible by  $LT(xy+1)$  or  $LT(y^2-1)$ , add  $-y$  to  $r$ .  
Now  $f = 0$ . Output  $r = -x-y$ .

# Leading Term Ideal

The leading term ideal of an ideal  $I$  is  $LT(I) = (LT(g) : g \text{ in } I)$ .

Example:  $LT((xy + 1, y^2 - 1))$  is not just  $(xy, y^2)$  because  $-x-y$  is in  $(xy+1, y^2-1)$  so  $x$  is in  $LT((xy + 1, y^2 - 1))$ .



Definition: a Grobner Basis for the ideal  $I=(h_1, h_2, \dots, h_s)$  is a collection of polynomials  $g_1, g_2, \dots, g_t$  such that  $(g_1, g_2, \dots, g_t) = I$  and  $(LT(g_1), LT(g_2), \dots, LT(g_t)) = LT(I)$ .

Theorem: If  $G=[g_1, g_2, \dots, g_t]$  is a Grobner Basis for  $I$  and  $f \in C[x_1, \dots, x_n]$  then the remainder  $Rem(f, G)$  of  $f$  upon division by  $G$  is well-defined and  $f \in I$  if and only if  $Rem(f, G) = 0$ .

# Interlude: Solving Equations

Grobner Bases tell us when a system has *a* solution ( $\text{Rem}(1, G) \neq 0$ )

Q1: How do we find a Grobner Basis? After all, it looks difficult to even find the leading term ideal.

Q2: How do we find the *solutions* to the system?

Buchberger answered Question 1 using S-polynomials.

# S-polynomials

The S-polynomial  $S(f_1, f_2)$  of  $f_1$  and  $f_2$  is the simplest combination of the two polynomials that cancels leading terms:

$$S(f_1, f_2) = \frac{x^c}{LT(f_1)} f_1 - \frac{x^c}{LT(f_2)} f_2,$$

where  $x^c$  is the least common multiple of  $LT(f_1)$  and  $LT(f_2)$ .

Ex:  $S(xy^2 - y + 1, x^2y + y^2 + x) = x(xy^2 - y + 1) - y(x^2y + y^2 + x)$   
 $= -2xy + x - y^3$

Since neither  $xy^2$  nor  $x^2y$  divide  $xy$ ,  $[xy^2 - y + 1, x^2y + y^2 + x]$  is not a Grobner Basis for the ideal it generates.

# Buchberger's Criterion

A set of polynomials  $g_1, g_2, \dots, g_t$  is a Grobner Basis for the ideal it generates if

$$\text{Rem}(S(g_i, g_j), G) = 0$$

for every pair  $(g_i, g_j)$ .

Suggests an [algorithm](#) to compute Grobner Bases:

If  $\text{Rem}(S(g_i, g_j), G) \neq 0$  then add  $\text{Rem}(S(g_i, g_j), G)$  to the list of polynomials and iterate.

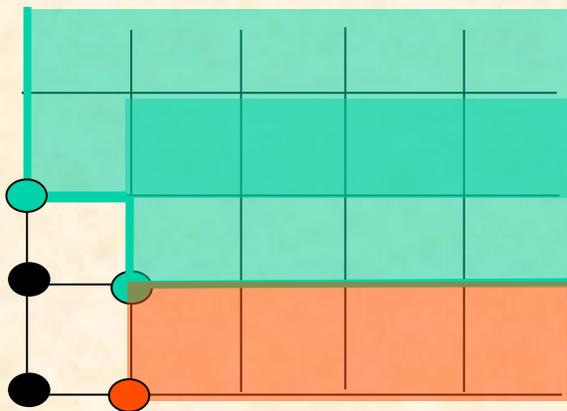
Example: A Grobner Basis for the ideal  $(xy+1, y^2-1)$  is given by  $[xy+1, y^2-1, -x-y]$ . In fact, this is redundant. Another Grobner Basis is given by  $[y^2-1, -x-y]$ .

# Solving Equations (again)

Finiteness Theorem: A system of equations has a finite number of solutions precisely when the Grobner Basis of the associated ideal contains a polynomial of the form

$$g_i = x_i^{m_i} + \text{lower order terms}$$

for each variable  $x_i$ .



In this case, the number of solutions (counted appropriately) is  $\dim_{\mathbb{C}} \mathbb{C}[x_1, \dots, x_n] / \text{LT}(I) =$  number of uncovered vertices in the staircase diagram.

# Elimination Theory

Solving the system 
$$\begin{cases} 2x - y = 0 \\ xy - 3x - 2 = 0 \end{cases}$$

Can be accomplished by eliminating  $y$ ,

$$x(y - 2x) + x(2x) - 3x - 2 = 2x^2 - 3x - 2 = 0$$

Solving for  $x$ ,

$$\in (2x - y, xy - 3x - 2) \cap C[x]$$

$$2x^2 - 3x - 2 = (2x + 1)(x - 2) = 0 \Rightarrow x = -\frac{1}{2}, 2.$$

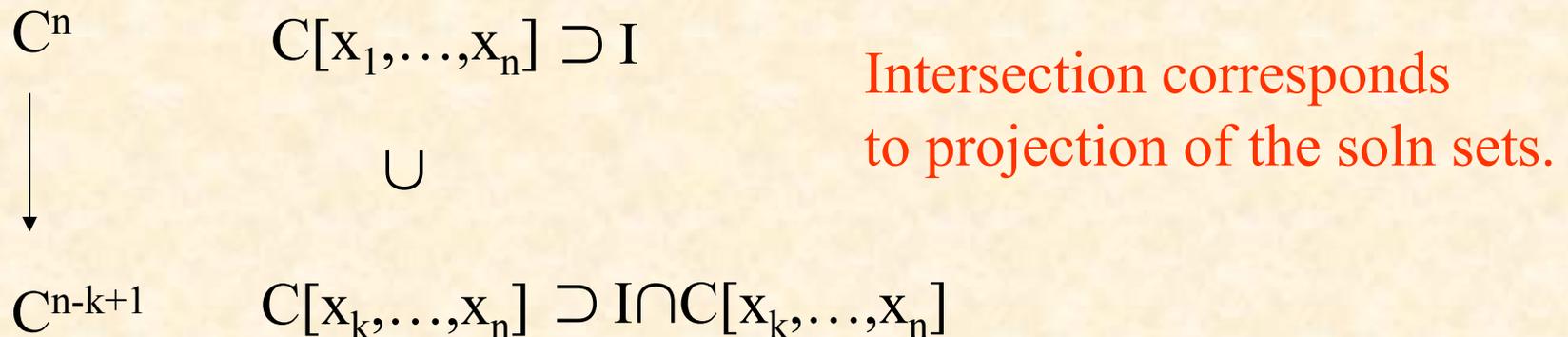
Now substitute to get the solutions:

$$\left(-\frac{1}{2}, 1\right) \quad \text{and} \quad (2, 4).$$

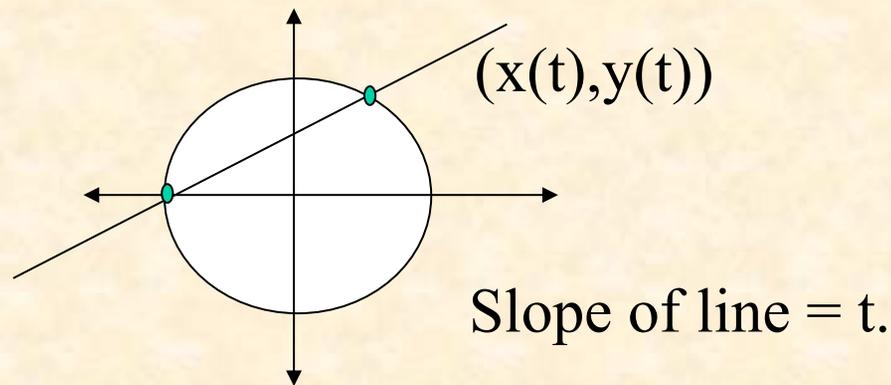
# Elimination with Grobner Bases

Elimination Theorem: If  $[g_1, \dots, g_t]$  is a Grobner Basis for  $I$  in Lex order then a Grobner Basis for  $I \cap C[x_k, x_{k+1}, \dots, x_n]$  is just given by those  $g_i$  that only involve the variables  $x_k, \dots, x_n$ .

When there are only finitely many solutions to the equations in  $I$ , elimination and back substitution produce them all.



# Implicitization



$$x(t) = \frac{1-t^2}{1+t^2}$$

$$y(t) = \frac{2t}{1+t^2}$$

Solution: Consider the ideal  $((1+t^2)x-(1-t^2), (1+t^2)y-2t)$  in  $C[t,x,y]$ . Eliminate  $t$  as follows. A Grobner Basis is

$$[-2tx - 2t + 2y, ty + x - 1, -x^2 - y^2 + 1].$$

The points on the curve must satisfy  $x^2 + y^2 = 1$ . The curve is a circle.

# Computational Issues

- In fact, best method to solve systems uses both Grobner Bases and some Linear Algebra.
- The Buchberger Algorithm is implemented in most symbolic computation packages (eg. [MAPLE](#), [MATHEMATICA](#)) and in many specialized packages ([MACAULAY2](#), [SINGULAR](#), [COCOA](#)).
- The polynomials in a Grobner Basis can be very large (both in degree and in the size of the coefficients)
- The number of polynomials in a Grobner Basis can be huge (doubly exponential in the degree and number of variables) and can differ radically depending on choice of monomial ordering
  - FGLM algorithm – Grobner Walk transforms GBs

# My work at MHPCC

My goal is to parallelize the Buchberger algorithm to compute Grobner Bases.

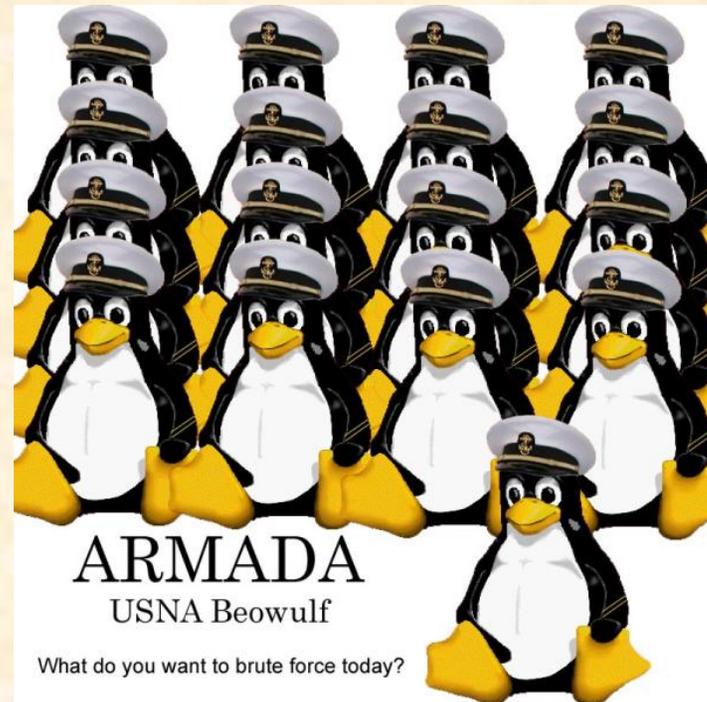
Size of polynomials

→ distributed across processors

Reduction of S-polynomials

→ requires communication

between processors



Algorithm has been implemented on shared memory machines. I would like to port the algorithm to a Cluster of Workstations (BEOWULF).