Kinematics and Numerical Algebraic Geometry

Charles Wampler

General Motors R&D Center

Warren, Michigan, USA

Including collaborations with:

Andrew Sommese	Jon Hauenstein	Dan Bates
Notre Dame	Texas A&M	Colorado St.

Outline

- Motivation:
  - Brief introduction to kinematics
- Basic polynomial continuation
  - Finding isolated roots
- Numerical algebraic geometry
  - Dealing with positive-dimensional sets
- Examples from kinematics
- Some recent work
- Software

Part I

- Motivation
- Kinematics in a nutshell

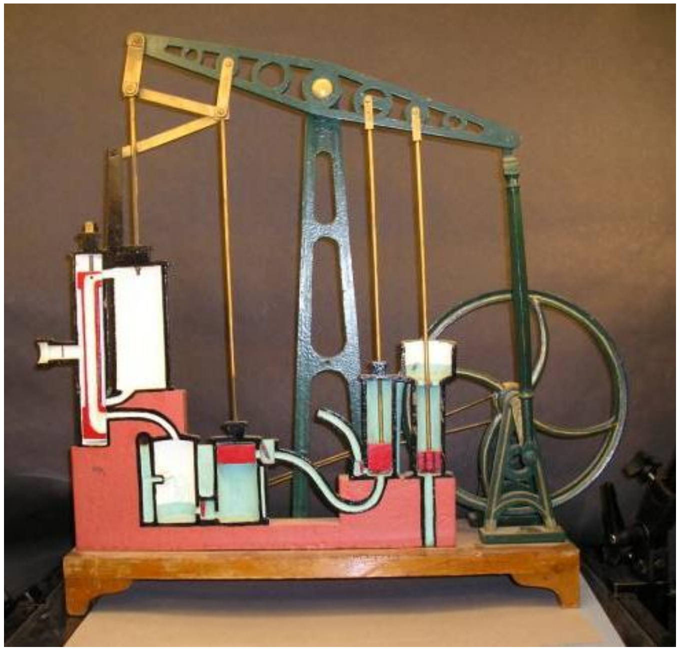
Why study polynomial systems?

- Mathematics
  - Intrinsically interesting
    - Algebra, algebraic geometry
    - Nonlinear, but with lots of structure
- Application areas
  - Economics & finance
    - Nash Equilibria
  - Chemical equilibrium
  - Computer-aided Geometric Design (CAGD)
    - Polynomial surface patches (B-splines, etc.)
  - Control theory
    - Pole placement, Optimal control
  - Kinematics...

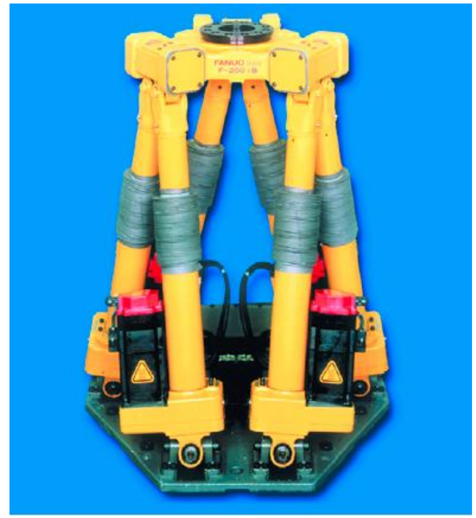
$$
\begin{array}{ll}
O_2 \rightleftharpoons 2O & k_1 X_{O_2} = X_O^2 \\
H_2 \rightleftharpoons 2H & k_2 X_{H_2} = X_H^2 \\
N_2 \rightleftharpoons 2N & k_3 X_{N_2} = X_N^2 \\
CO_2 \rightleftharpoons O + CO & k_4 X_{CO_2} = X_O X_{CO} \\
OH \rightleftharpoons O + H & k_5 X_{OH} = X_O X_H \\
H_2O \rightleftharpoons O + 2H & k_6 X_{H_2O} = X_O X_H^2 \\
NO \rightleftharpoons O + N & k_7 X_{NO} = X_O X_N .
\end{array}
$$

$$
\begin{aligned}
T_H &= X_H + 2X_{H_2} + X_{OH} + 2X_{H_2O} \\
T_C &= X_{CO} + X_{CO_2} \\
T_O &= X_O + X_{CO} + 2X_{O_2} + 2X_{CO_2} + X_{OH} + X_{H_2O} + X_{NO} \\
T_N &= X_N + 2X_{N_2} + X_{NO}
\end{aligned}
$$

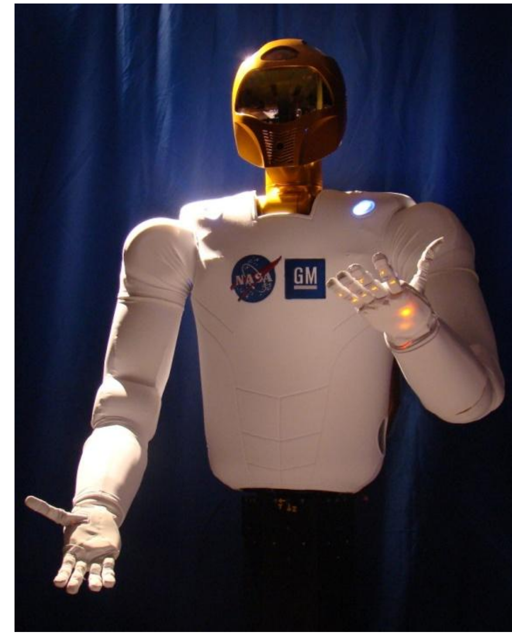
Kinematics: Then & Now



Model of Watt Engine
1784



FANUC F200
Robot



NASA-GM
Robonaut2

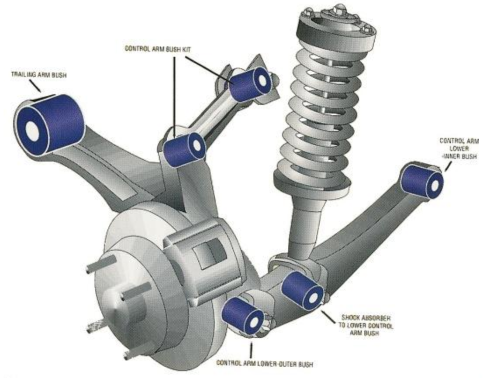
Closed-chain planar

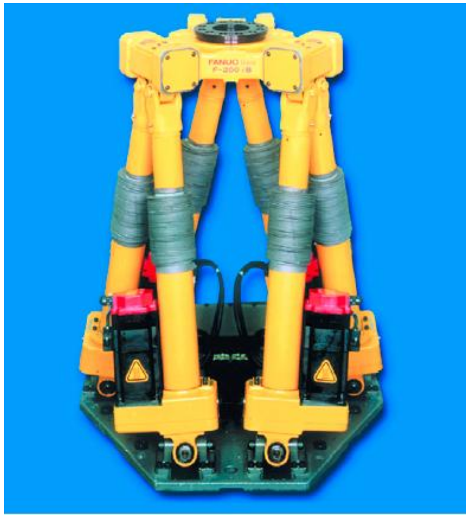
Closed-chain spatial

Open-chain spatial

Application: Kinematics

- Constrained mechanical motion
- Two major classes:
  - Linkages for motion constraint & transformation
    - Suspensions, engines, swing panels, etc.
  - Computer-controlled motion devices
    - Robots, human-assist devices, etc.
- Rigid links + common joints = polynomial equations
  - *Algebraic kinematics*

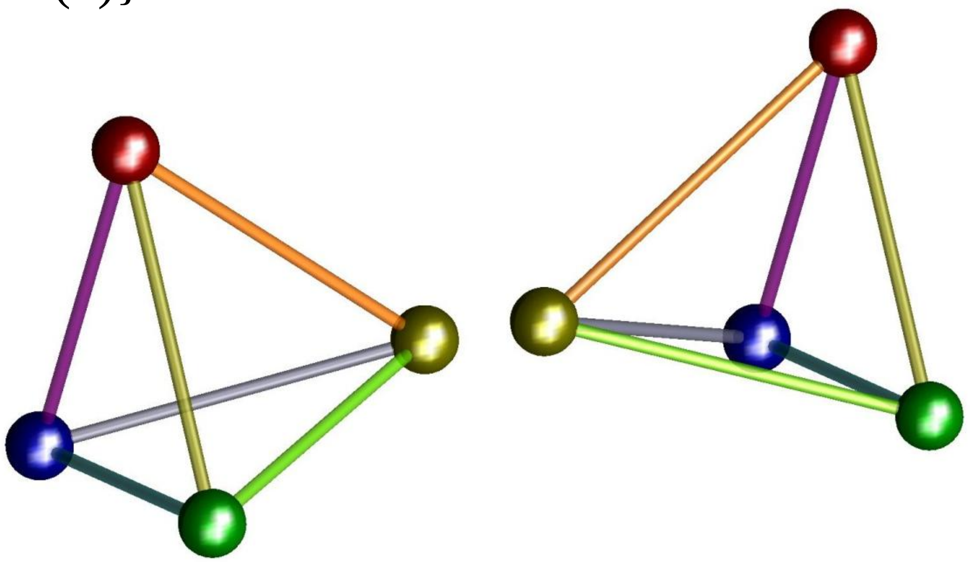




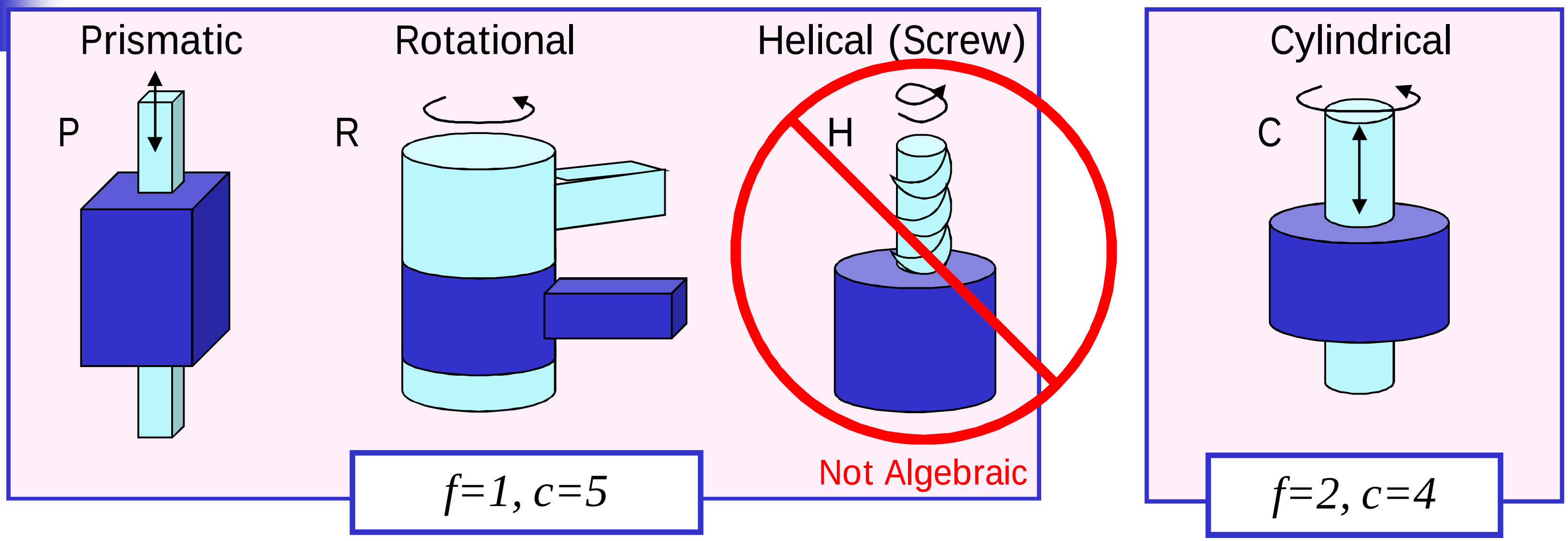
GM

Rigid-Body Motion

- A rigid body has two defining properties:
  - Preservation of distance
  - Preservation of handedness
- Accordingly, the pose of a rigid body lies in SE(3)
  - $SE(3) = \{(p, A) : p \in \mathrm{R}^3,\ A \in SO(3)\}$
- $SE(3)$ is *algebraic*, subject to the defining eqns. for $SO(3)$:
  - $A^{\mathrm{T}}A = \mathrm{I},\ \det A = 1$
- $dim\ SE(3) = 6$

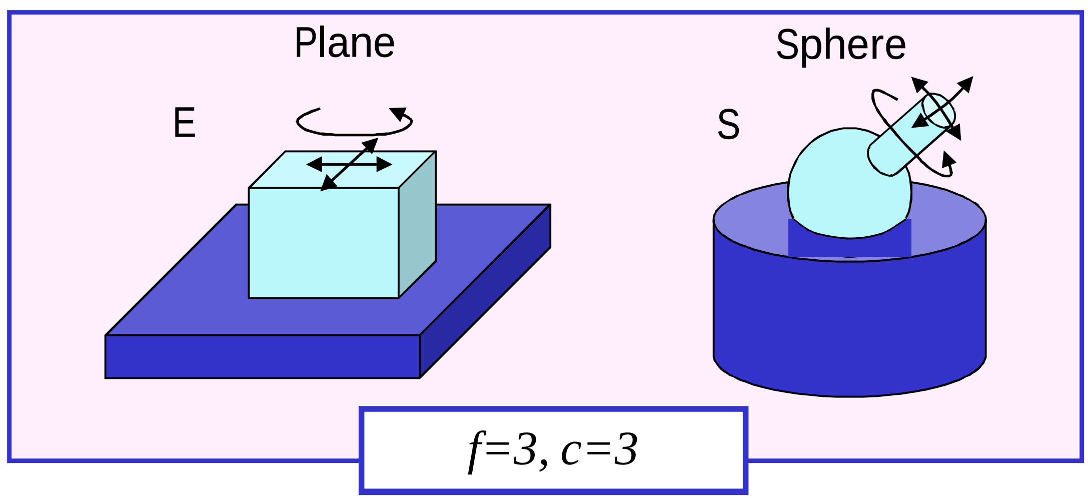


Joints: Lower-order pairs

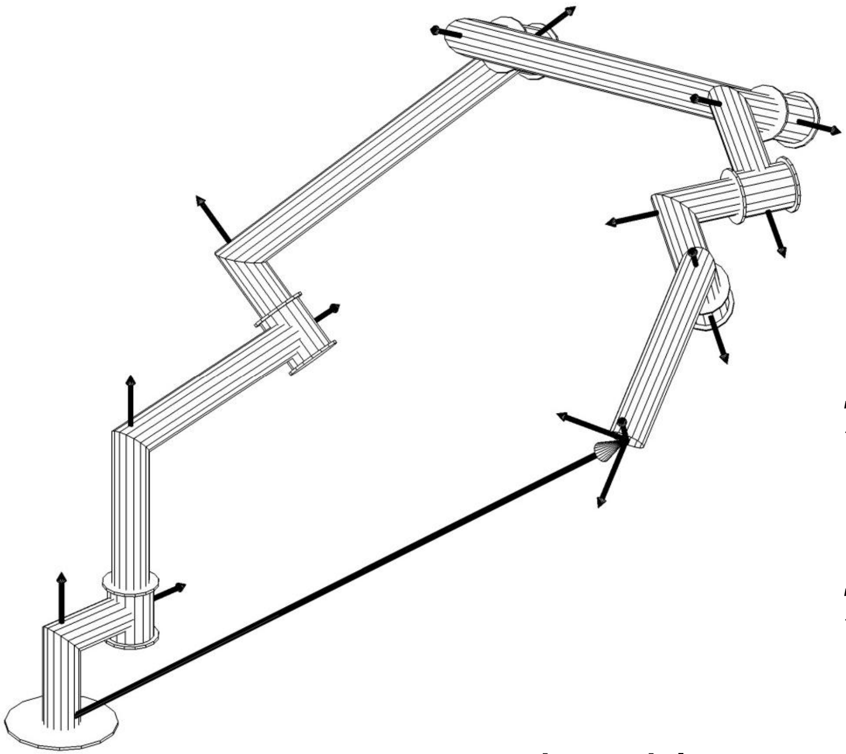


f = freedom

c = constraint in SE(3)



Example: Serial 6R Robot



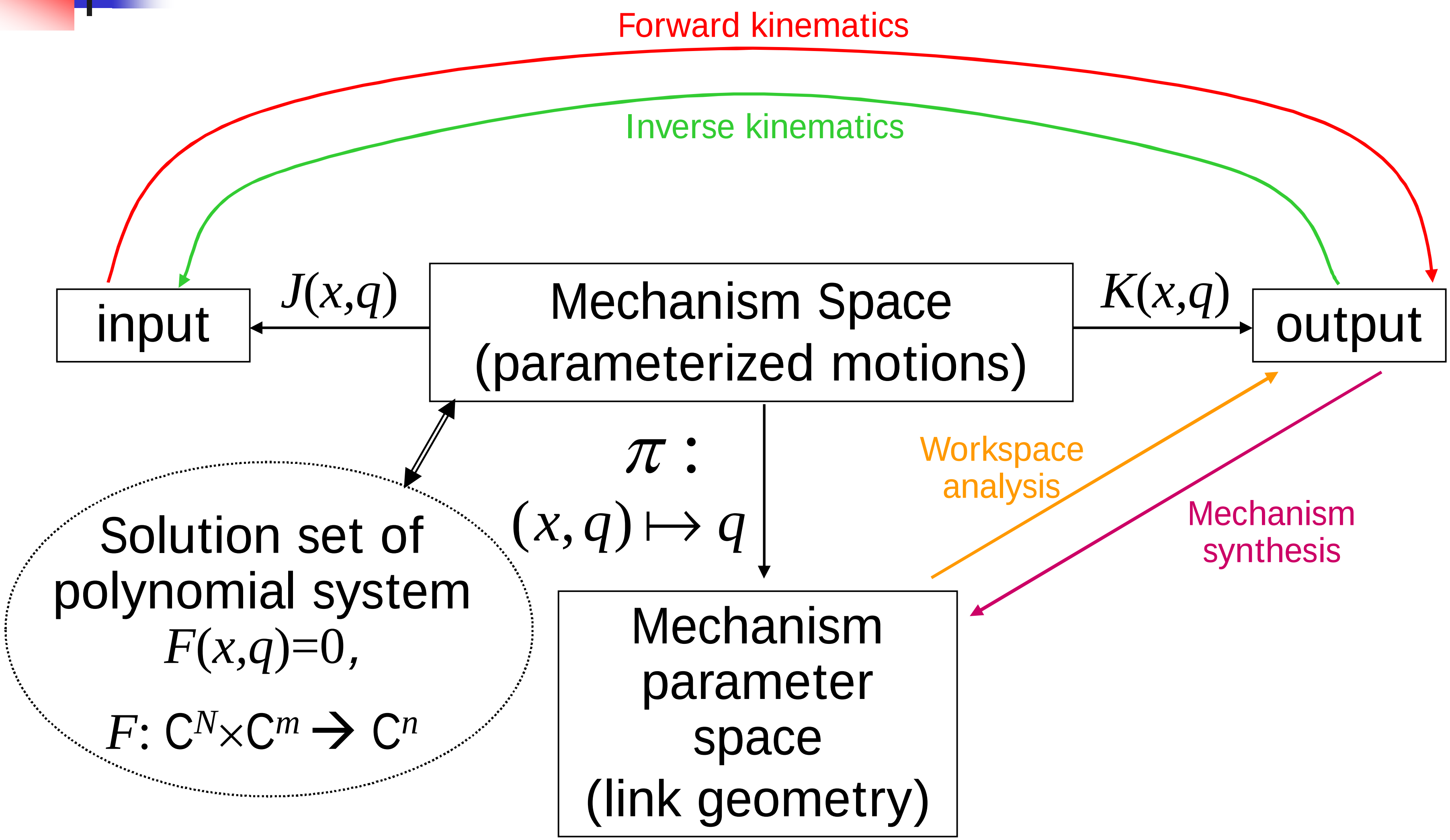
- Parameters given:
  - Length d_i, offset a_i, twist α_i
- Input:
  - Rotation angle at each joint, θ_i
- Output:
  - Position & orientation of end of arm, T_{end}

$$T_{end} = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5 \cdot T_6$$

$$T_i = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Forward problem:
  - Unique answer
- Inverse problem:
  - Up to 16 solutions

Big Picture


Forward kinematics
Inverse kinematics
input
$J(x,q)$
Mechanism Space (parameterized motions)
$K(x,q)$
output
Solution set of polynomial system $F(x,q)=0$,
F: $\mathbb{C}^N \times \mathbb{C}^m \rightarrow \mathbb{C}^n$
$\pi :$
$(x,q) \mapsto q$
Workspace analysis
Mechanism synthesis
Mechanism parameter space (link geometry)

Part II

- Basic polynomial continuation

What is Continuation?

- A method to solve N equations in N unknowns

$$F(x)=0, \quad F:\mathbf{C}^N \to \mathbf{C}^N$$

- Step 1: Define a homotopy

  $$H(x,t)=0, \quad H:\mathbf{C}^{N+1} \to \mathbf{C}^N$$

  such that

  $H(x,0)=F(x)$, and $H(x,1)=0$ is easily solved.

- Step 2: From each solution point of $H(x,1)=0$, follow the solution paths of $H(x,t)=0$ as t goes to 0.

- For polynomial systems, we can choose H to ensure that
  - All the paths go all the way to $t=0$.
  - Every isolated solution of $F(x)=0$ has a path leading to it.

Basic Total-degree Homotopy

To find all isolated solutions to the polynomial system

$$\begin{bmatrix} \mathrm{f}_1(x_1,\ldots,x_N) \\ \vdots \\ \mathrm{f}_N(x_1,\ldots,x_N) \end{bmatrix} = 0, \quad \deg(\mathrm{f}_i) = \mathbf{d_i}$$

form the linear homotopy

$$H(x,t) = (1\text{-}t)F(x) + tG(x) = 0,$$

where

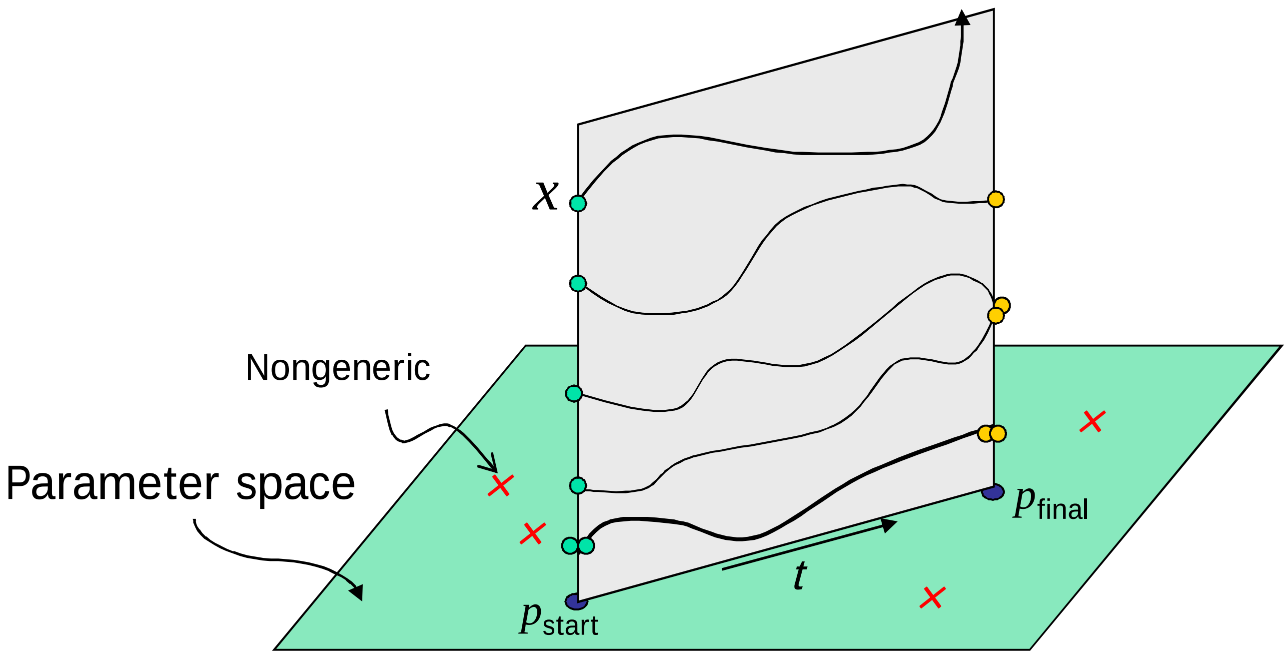
$$g_i(x) = a_i x_i^{\mathbf{d_i}} + b_i, \quad a_i, b_i \text{ random, complex.}$$

Number of paths to track = $d_1 \cdot d_2 \cdots d_N$

Solution paths

- Paths $x(t)$ implicitly defined by homotopy

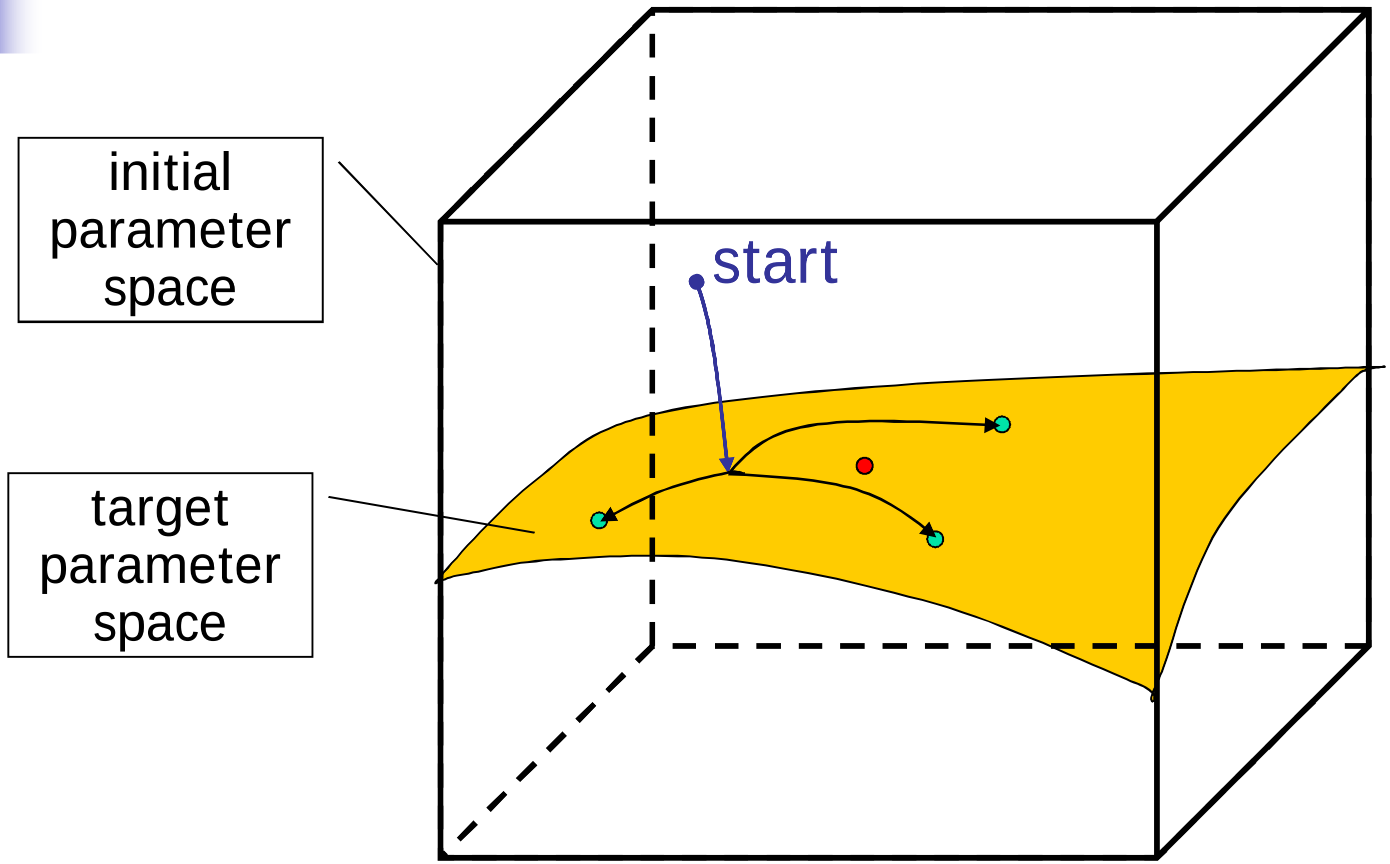
$$H\,(x;\,p(t)) = 0$$



Why it works: Generic Root Count

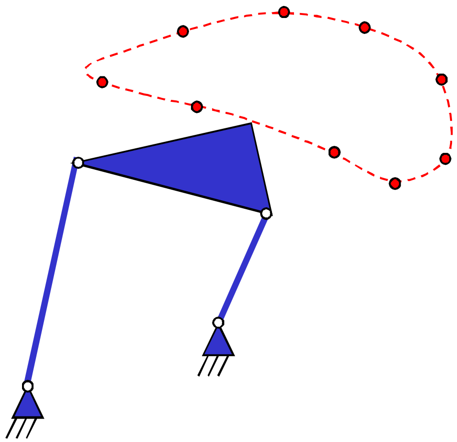
- A parameterized family of polynomial systems $F(x,q)=0$ has a *generic root count* :
  - Assume F: $\mathbb{C}^N \times Q \rightarrow \mathbb{C}^n$, Q an irreducible algebraic set
  - For almost all $q \in Q$, $F(x,q)=0$ has the same number of nonsingular, isolated roots. This is the generic root count.
  - The exceptions in Q are a proper algebraic subset.
  - So, a random 1-real-dimensional path in Q misses exceptions with *probability one.*
- For a nested parameter space, the generic root count can only go down. (“Upper semi-continuity”)

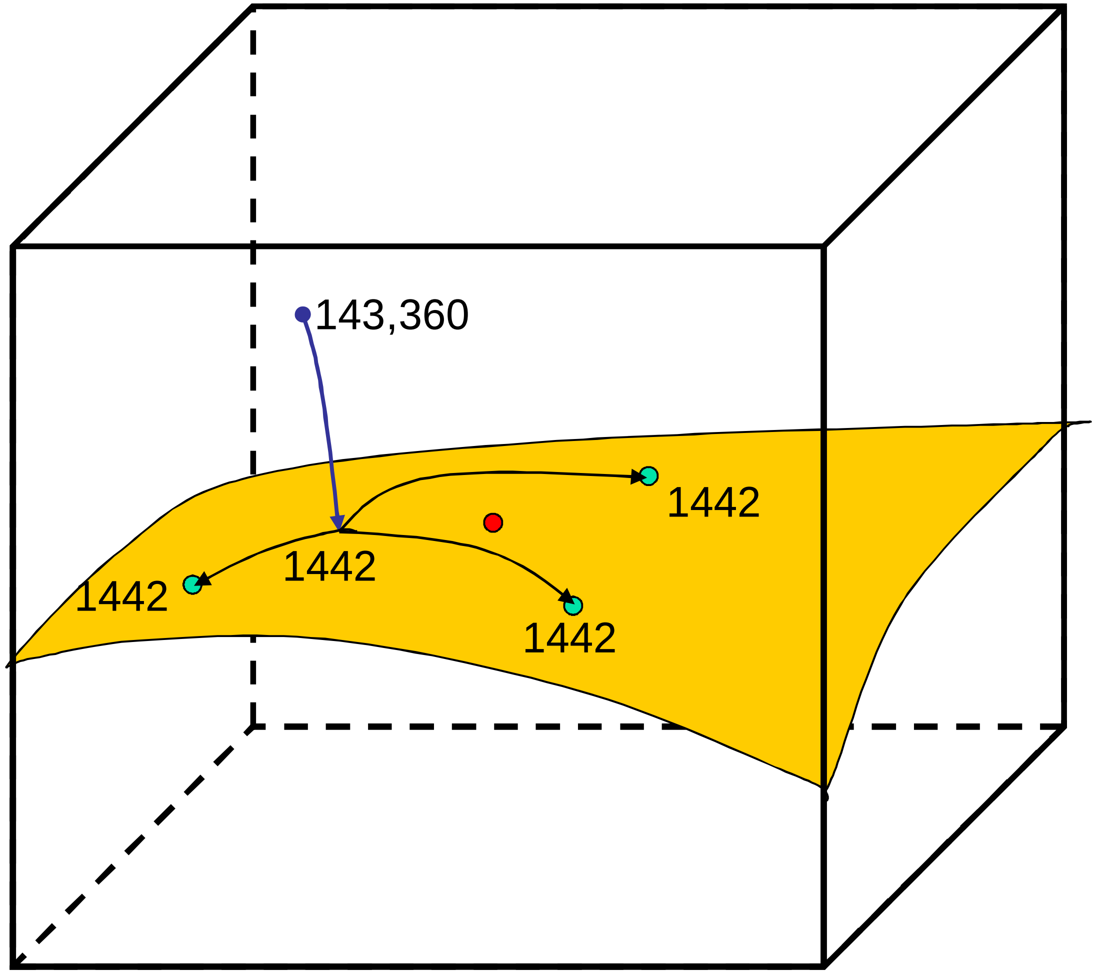
Parameter Continuation



- Start system easy in initial parameter space
- Root count may be much lower in target parameter space
- Initial run is 1-time investment for cheaper target runs

Parameter Continuation: 9-pt path synthesis





- Total degree
  - 7^8=5,764,801
- Multihomogeneous
  - 286,720
- Symmetry
  - 143,360
- Parameter homotopy
  - 1442 paths

Part III

- Numerical Algebraic Geometry

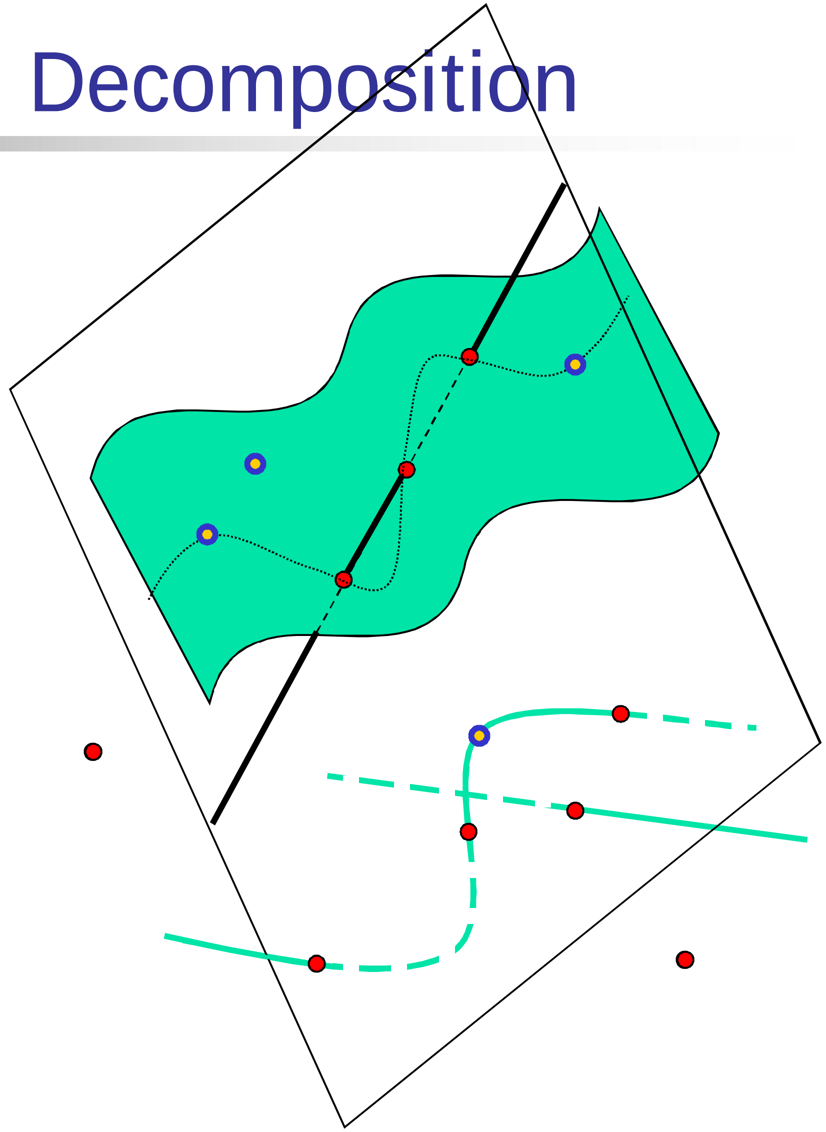
Irreducible Decomposition

Univariate	**Multivariate System**
1 Equation, 1 Variable	n Equations, N Variables
solution points	sol'n points, curves, surfaces, etc.
double roots, etc.	sets with multiplicity
Factorization, $\prod_i (x - a_i)^{\mu_i}$	**Irreducible decomposition**

Numerical Representation	
list of points	list of witness point sets

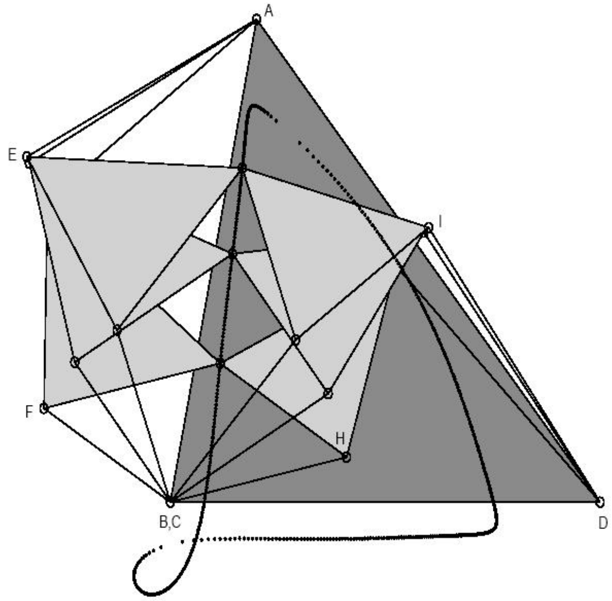
Numerical Irreducible Decomposition

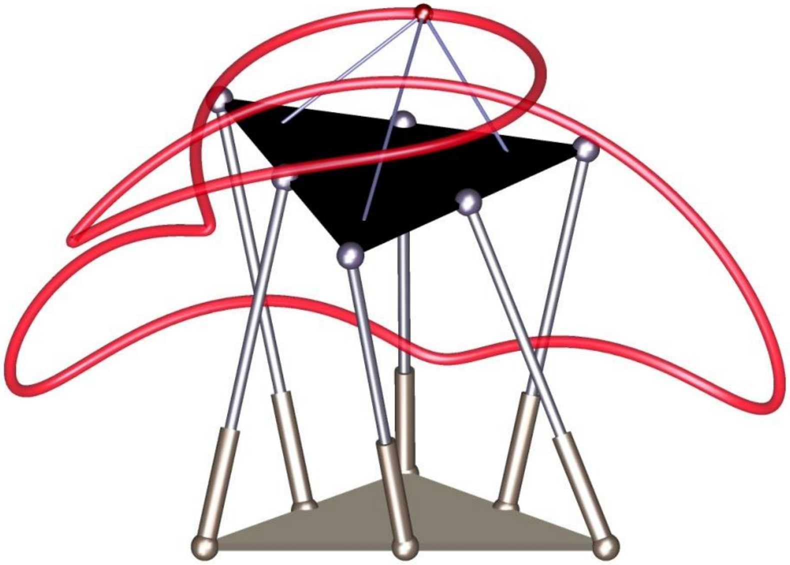
- Witness Set
  - Intersection of an algebraic set with a linear space of complementary dimension
    - Get *d* points on each degree *d* component
  - Defined dimension-by-dimension
- Witness set generation
  - Slice for every dimension
  - Homotopy finds all isolated solutions at each dimension
- Decomposition
  - Remove “junk” points
  - At each dimension, sort witness set into irreducible components



Part IV: Examples

- Let's see Numerical Algebraic Geometry at work in kinematics

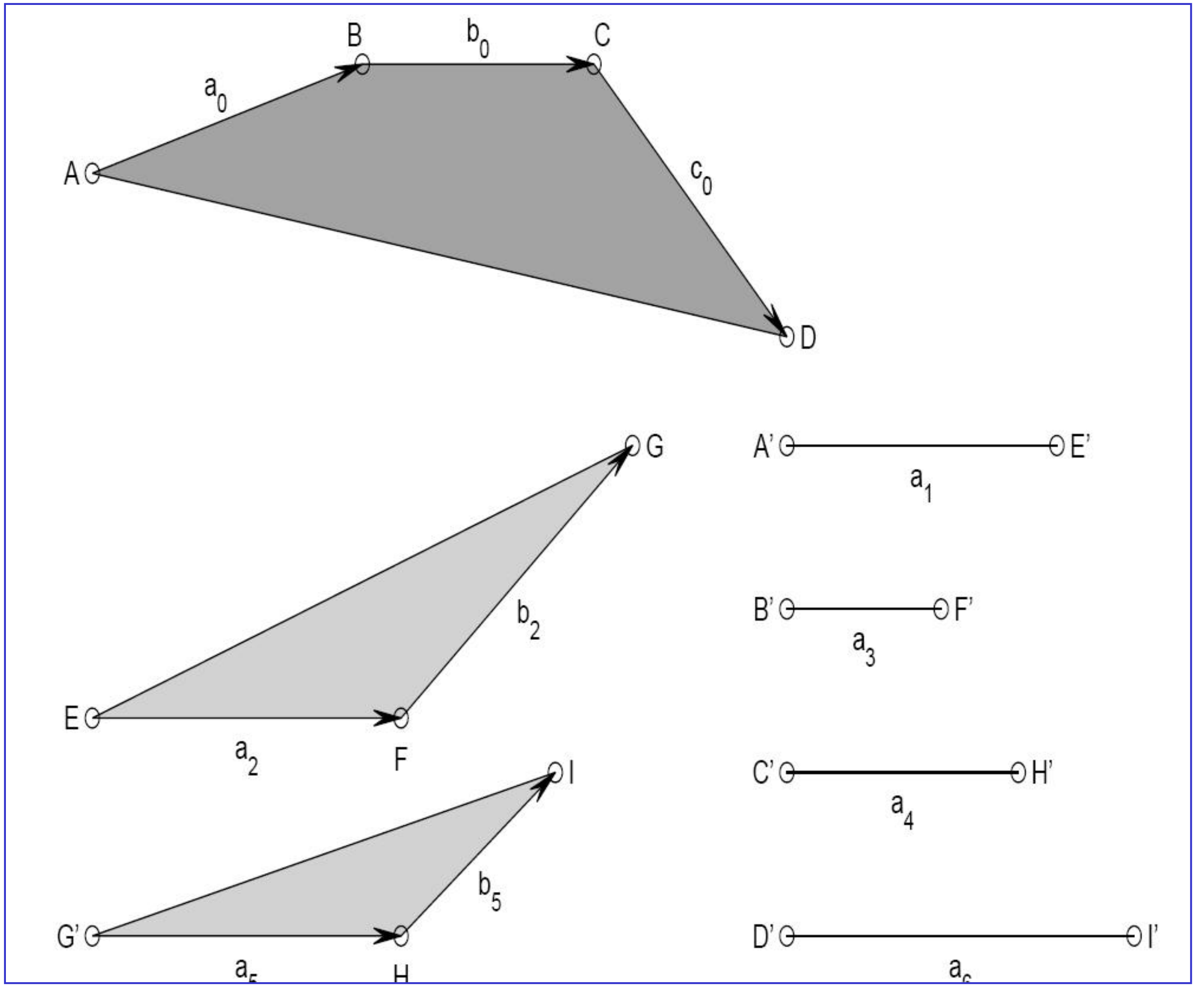




Example: 7-bar Structure

Problem:

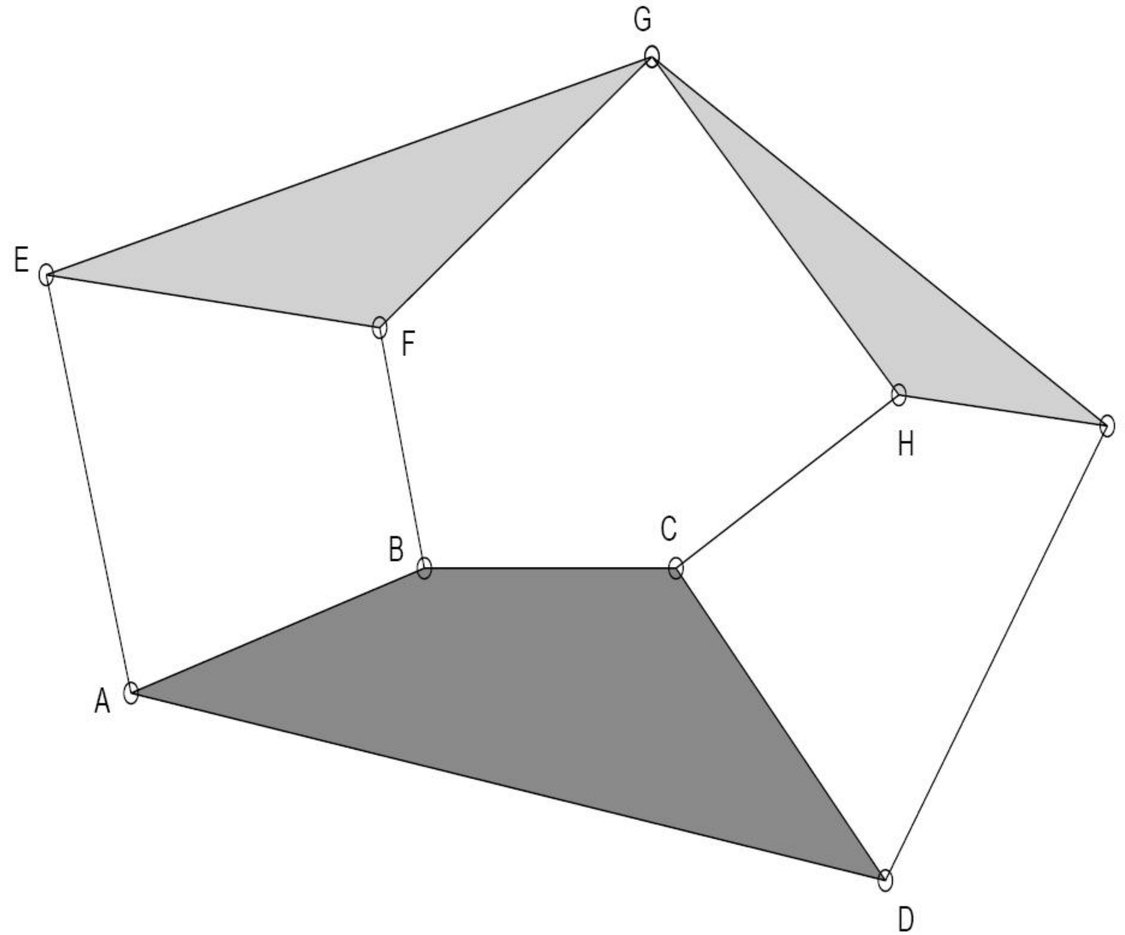
Assemble these 7 pieces, as labeled.



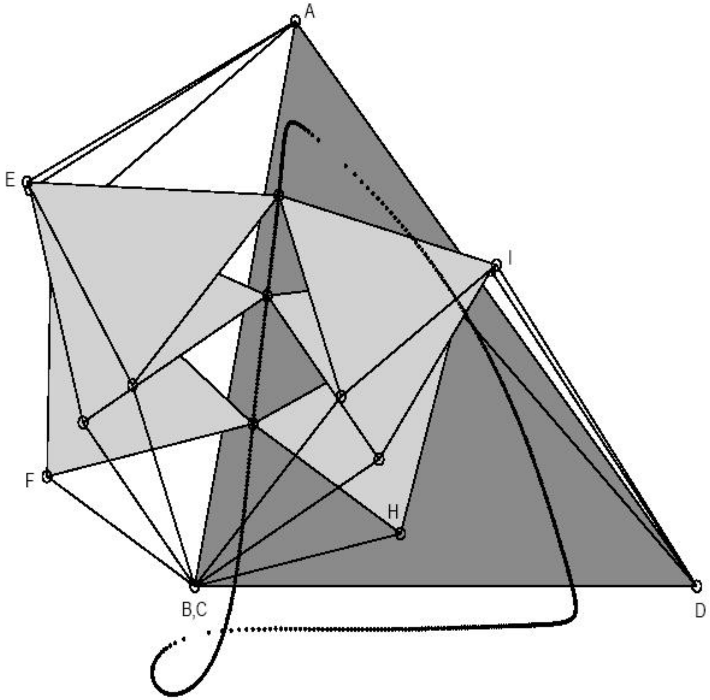
Result for Generic Links

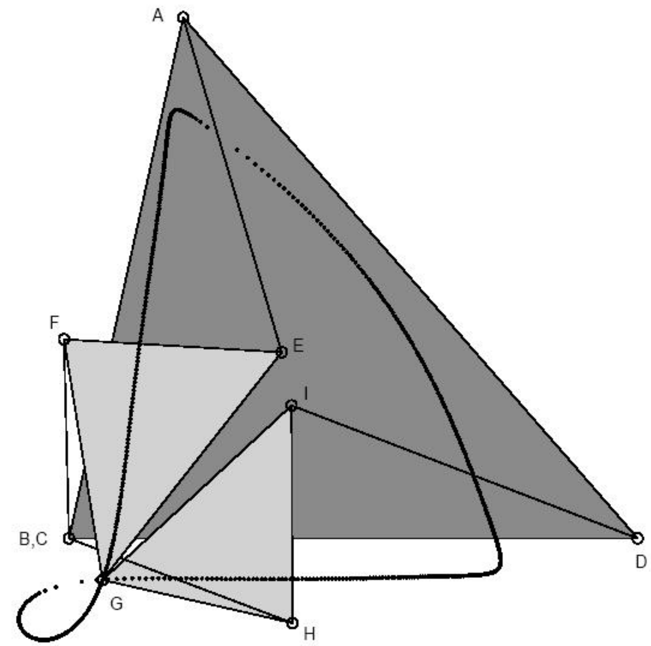
18 rigid structures

- 8 real, 10 complex for this set of links.
- All isolated – can be found with traditional homotopy



Special Links (Roberts Cognates)





Dimension 1:

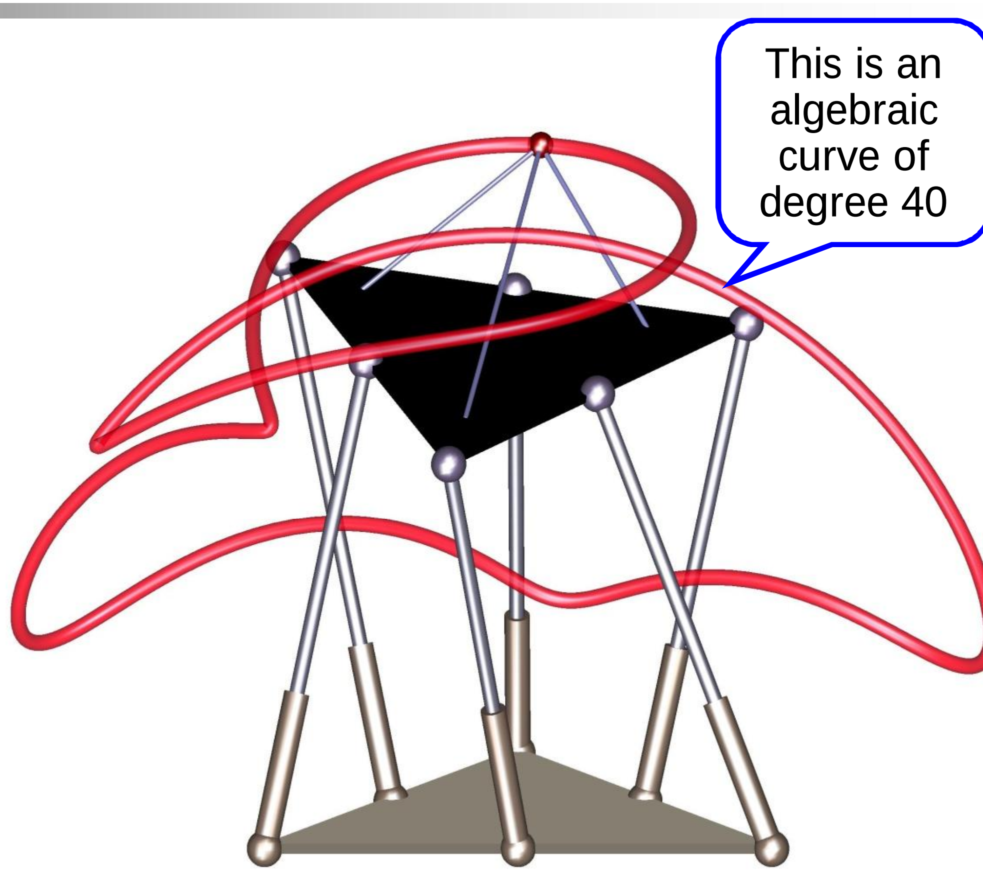
6^{th} degree four-bar motion

Dimension 0:

1 of 6 isolated (rigid) assemblies

Exceptional Stewart-Gough Platform

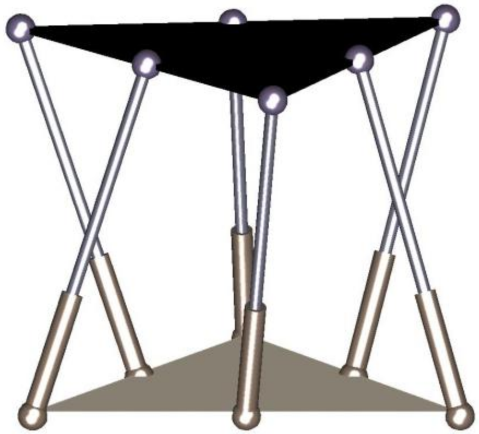
- Case 1: Top & bottom plates are equilateral triangles
  - Degree of top platform motion in Study (dual quaternion) coordinates is 28
  - Degree of path of a tracing point is 40.



- Case 2: In addition, leg lengths equal & plates congruent
  - Factors as 6+(6+6+6)+4=28

Even More Exceptional Stewart-Gough Platform

- As before, but with
  - leg lengths = altitude of base triangle
  - “Foldable Griffis-Duffy Platform”
- Degree 28 component now factors as
  - $3\times[2\times1]+3\times2+4+(4+4+4)$
  - We have extracted the *real* parts of these complex components
    - 3 double lines, 3 quadrics, 1 quartic



Part V

- Some recent work
  - Equation-by-equation Regeneration

Working Equation-by-Equation

- Basic step

$$V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ \boxed{L_k(x)} \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right) \Rightarrow V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ \boxed{f_k(x)} \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right)$$

Regeneration: Step 1

$$V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ \boxed{L_k(x)} \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right)$$

move linear fcn d_k times
(use linear homotopy)

$$V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ L_{k,1}(x) \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right) \quad \cdots \quad V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ L_{k,d_k}(x) \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right)$$

Union of sets

$$V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ \boxed{L_{k,1}(x)\cdots L_{k,d_k}(x)} \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right)$$

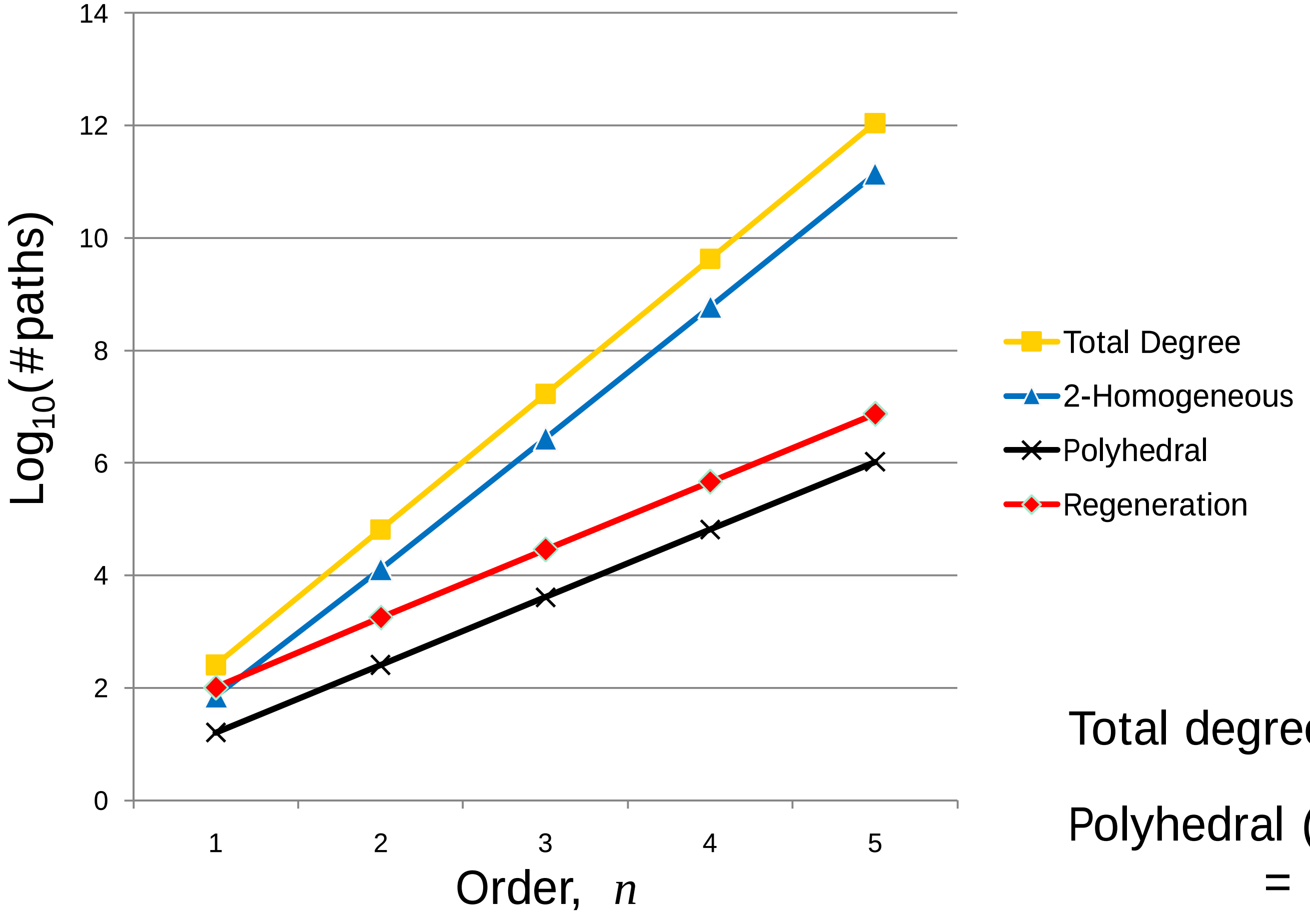
Regeneration: Step 2

$$V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ \boxed{L_{k,1}(x)\cdots L_{k,d_k}(x)} \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right) \xrightarrow{\text{Linear homotopy}} V_0\left(\begin{bmatrix} f_1(x) \\ \vdots \\ f_{k-1}(x) \\ \boxed{f_k(x)} \\ L_{k+1}(x) \\ \vdots \\ L_N(x) \end{bmatrix}\right)$$

Repeat for $k+1, k+2, \ldots, N$

Test Run: Lotka-Volterra Systems

- Discretized PDE (finite differences) population model
  - Order n system has $8n$ sparse bilinear equations

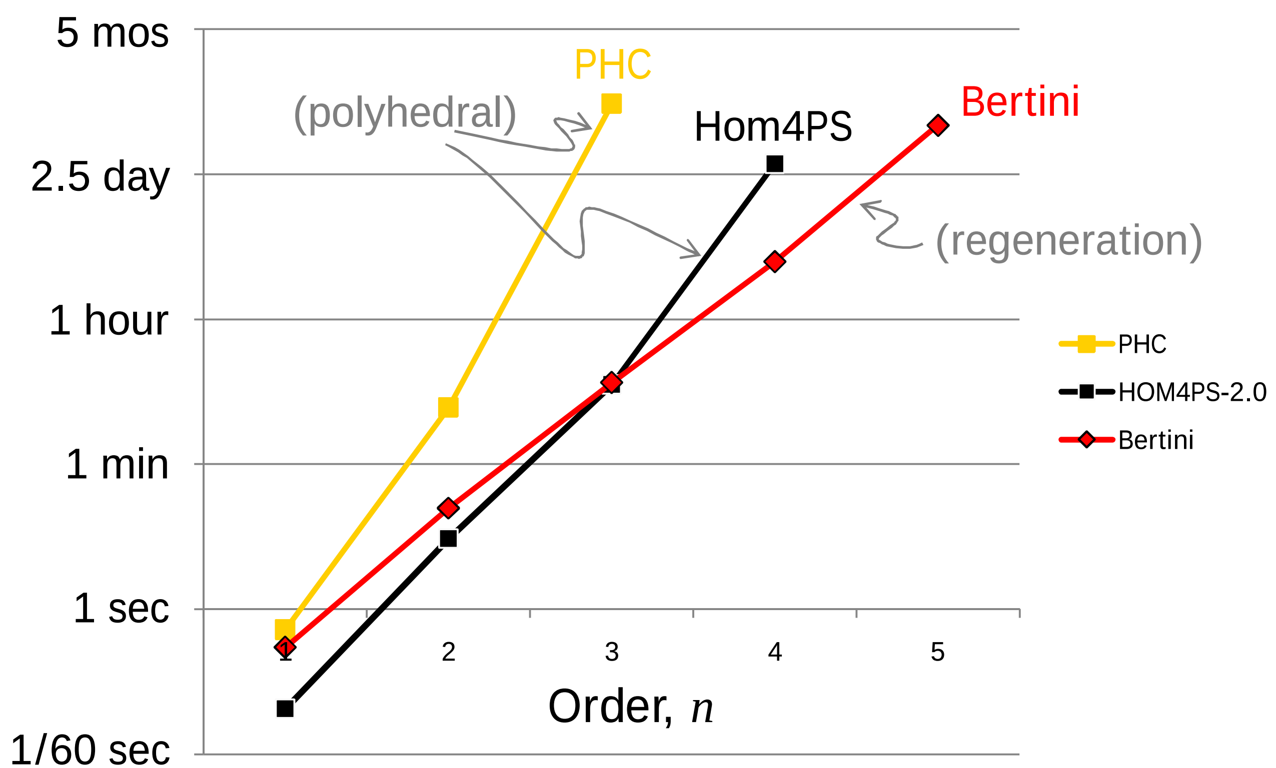


Total degree = 2^{8n}

Polyhedral (mixed volume) = 2^{4n} is exact

Lotka-Volterra Systems (cont.)

- Time Summary -- Single Processor



- Regeneration parallelizes easily (polyhedral does not)

Software

- Hom4PS (v2.0)
  - Isolated solutions only
  - Fast polyhedral
  - Author: T.-Y. Li (MSU)

- PHC
  - Numerical algebraic geometry
  - Polyhedral method
  - Author: Jan Verschelde (UIC)

- Bertini (v1.2)
  - Numerical algebraic geometry
    - Parallel computing option
    - Robust & efficient adaptive multiprecision
    - Regeneration
  - Authors: Bates, Hauenstein, Sommese & Wampler
  - Free download at
    - www.nd.edu/~sommese/bertini/

Wrap-up

- Much of kinematics is applied algebraic geometry
- Numerical polynomial continuation solves for isolated points
- Numerical algebraic geometry extends this to positive-dimensional sets
- Regeneration is the newest technique
- Bertini v1.2 offers all this & more
  - Parallel computing, in particular