numerical Schubert calculus

Jan Verschelde

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science
http://www.math.uic.edu/˜jan
jan@math.uic.edu

Graduate Computational Algebraic Geometry Seminar

Outline

1. Problem Statement
   - solving Schubert problems
   - homotopies for enumerative geometry
   - the module `schubert` of `phcpy`
2. Littlewood-Richardson Homotopies
   - checker games to count the roots

motivation

- Understanding a classical topic in algebraic geometry:

  Given four lines in general position in 3-space,
  find all lines that meet the given four lines in a point.

- Developing numerical homotopy continuation methods:
  1. The formal root count is *generically* sharp.
  2. Specific homotopies are *generically* optimal.

  Except of an algebraic subset of the input coefficients:
  1. the number of solutions equals the formal root count; and
  2. every solution lies at the end of one solution path as no paths diverge to infinity or converge to spurious components.

- Applications to linear systems control and matrix theory:
  1. output pole placement problem for linear systems,
  2. matrix completion with prescribed eigenvalues.

Schubert Varieties

A Schubert variety is defined by an n-dimensional flag F:

$$F = [\mathbf{f}_1 \mathbf{f}_2 \cdots \mathbf{f}_n] \in \mathbb{C}^{n \times n} \quad \langle \mathbf{f}_1 \rangle \subset \langle \mathbf{f}_1, \mathbf{f}_2 \rangle \subset \cdots \subset \langle \mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n \rangle$$

and a k-dimensional bracket $\omega \in \mathbb{N}^k$, $1 \leq \omega_1 < \omega_2 < \cdots < \omega_k \leq n$:

$$\Omega_\omega(F) = \left\{ X \in \mathbb{C}^{n \times k} \mid \dim(X \cap \langle \mathbf{f}_1, \ldots, \mathbf{f}_{\omega_i} \rangle) = i, i = 1, 2, \ldots, k \right\}.$$

For example: for $F \in \mathbb{C}^{6 \times 6}$, $\Omega_{[2\ 4\ 6]}(F)$ contains

$$X = \begin{bmatrix} 1 & 0 & 0 \\ x_{21} & 1 & 0 \\ x_{31} & x_{32} & 1 \\ x_{41} & x_{42} & x_{43} \\ 0 & x_{52} & x_{53} \\ 0 & 0 & x_{63} \end{bmatrix}$$

$$\dim(X \cap \langle \mathbf{f}_1, \mathbf{f}_2 \rangle) = 1$$
$$\dim(X \cap \langle \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4 \rangle) = 2$$
$$\dim(X \cap \langle \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5, \mathbf{f}_6 \rangle) = 3$$

expressed via conditions on minors $\longrightarrow$
system of 13 polynomials in 9 variables

Schubert Problems

The problem of four lines in 3-space can be expressed as

$$\Omega_{[2\ 4]}(F_1) \cap \Omega_{[2\ 4]}(F_2) \cap \Omega_{[2\ 4]}(F_3) \cap \Omega_{[2\ 4]}(F_4),$$

where the first two columns of the flags F_1, F_2, F_3, and F_4 contain the generators of the lines in $\mathbb{P}^3 \simeq \mathbb{C}^4$.

The Littlewood-Richardson rule computes the number of solutions:

$$\begin{aligned} [2\ 4]^4 &= (+1[2\ 3] + 1[1\ 4])[2\ 4][2\ 4] \\ &= (+1[1\ 3] + 1[1\ 3])[2\ 4] \\ &= +2[1\ 2] \end{aligned}$$

$\rightarrow$ there are two lines that meet the four general lines.

Outline

1. Problem Statement
   - solving Schubert problems
   - homotopies for enumerative geometry
   - the module `schubert` of `phcpy`

2. Littlewood-Richardson Homotopies
   - checker games to count the roots

a Geometric Littlewood-Richardson Rule

William Fulton: *Young Tableau. With Applications to Representation Theory and Geometry*. Cambridge University Press, 1997.
The first geometric proof and interpretation was given by
Ravi Vakil: *a geometric Littlewood-Richardson rule*. Ann of Math, 2006.

A combinatorial checker game for the Littlewood-Richardson coefficients gives homotopies to solve Schubert problems:
F. Sottile, R. Vakil, and J. Verschelde: *Solving Schubert problems with Littlewood-Richardson Homotopies.* In Proceedings of ISSAC 2010.

Motivation: experimental study of reality conjectures
`http://www.math.tamu.edu/~secant`
Christopher Hillar, Luis Garcia-Puente, Abraham Martin del Campo, James Ruffo, Zach Teitler, Stephen L. Johnson, Frank Sottile: *Experimentation at the Frontiers of Reality in Schubert Calculus.* Contemporary Math. AMS 2010.

Homotopies for Enumerative Geometry

- B. Huber, F. Sottile, and B. Sturmfels: Numerical Schubert calculus. *J. of Symbolic Computation*, 26(6):767–788, 1998.
- J. Verschelde: Numerical evidence for a conjecture in real algebraic geometry. *Experimental Mathematics* 9(2): 183–196, 2000.
- B. Huber and J. Verschelde: Pieri homotopies for problems in enumerative geometry applied to pole placement in linear systems control. *SIAM J. Control Optim.* 38(4):1265–1287, 2000.
- F. Sottile and B. Sturmfels: A sagbi basis for the quantum Grassmannian. *J. Pure and Appl. Algebra* 158(2-3): 347–366, 2001.
- T.Y. Li, X. Wang, and M. Wu: Numerical Schubert calculus by the Pieri homotopy algorithm. *SIAM J. Numer. Anal.* 20(2):578–600, 2002.
- J. Verschelde and Y. Wang: Computing dynamic output feedback laws. *IEEE Trans. Automat. Control.* 49(8):1393–1397, 2004.
- A. Leykin and F. Sottile: Galois group of Schubert problems via homotopy continuation. *Math. Comp.* 78(267): 1749–1765, 2009.

Outline

1. Problem Statement
   - solving Schubert problems
   - homotopies for enumerative geometry
   - the module `schubert` of `phcpy`

2. Littlewood-Richardson Homotopies
   - checker games to count the roots

the module schubert of phcpy

```
>>> from phcpy.schubert import pieri_root_count as prc
>>> help(prc)

>>> print prc.__doc__

    Computes the number of pdim-plane producing maps of
    degree qdeg that meet mdim-planes at mdim*pdim + qdeg*(mdim+pdim) points.

>>> print prc.__name__
pieri_root_count
>>> import inspect
>>> inspect.getargspec(prc)
ArgSpec(args=['mdim', 'pdim', 'qdeg'], \
varargs=None, keywords=None, defaults=None)
>>> prc(2,2,0)
Pieri root count for (2, 2, 0) is 2
the localization poset :
n = 0 : ([2 3],[2 3],1)([1 4],[1 4],1)
n = 1 :
n = 2 : ([1 3],[2 4],2)
n = 3 :
n = 4 : ([1 2],[3 4],2)
```

solving a generic instance of the problem of 4 lines

First we solve a generic instance:

```
>>> from phcpy.schubert import random_complex_matrix as rcm
>>> inputs = [rcm(4,2) for k in range(4)]
>>> from phcpy.schubert import run_pieri_homotopies as rph
>>> (pols, sols) = rph(2,2,0,inputs)
```

A basic verification consists in evaluation the `sols` at `pols`:

```
>>> from phcpy.schubert import verify
>>> verify(pols, sols)
(2.56947241262e-14-1.16712195464e-14j)
(2.70616862252e-16+1.8193779816e-14j)
the total check sum : (2.59653409884e-14+6.52256026967e-15j)
```

A solution:

```
>>> print sols[0]
t :  0.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x21 :  6.85482428065796E-02   9.24207758195128E-01
 x31 : -5.83884423158142E-02  -2.04839795784934E-02
 x32 : -1.07172856548658E+00  -1.87130208789554E+00
 x42 : -5.25781292125279E-01  -2.61949481714352E+00
== err :  3.831E-15 = rco :  2.275E-02 = res :  2.720E-15 =
```

solving a fully real instance

If the input planes osculate the rational normal curve, then the output planes will be real as well. The Shapiro conjecture was proven by Gabrielov and Eremenko.
We first make a Pieri problem with real inputs:

```
>>> from phcpy.schubert import real_osculating_planes as rop
>>> osculating = rop(2,2,0)
>>> from phcpy.schubert import make_pieri_system as mps
>>> target = mps(2,2,0,osculating)
```

Then we run a plain Cheater's homotopy:

```
>>> from phcpy.trackers import track
>>> targetsols = track(target, pols, sols)
```

Refining with Newton's method in double double precision:

```
>>> from phcpy.solver import newton_step
>>> newsols = newton_step(target, targetsols, precision='dd')
== err :  3.632E-15 = rco :  2.477E-02 = res :  4.094E-31 =
== err :  3.446E-15 = rco :  5.571E-03 = res :  1.556E-30 =
```

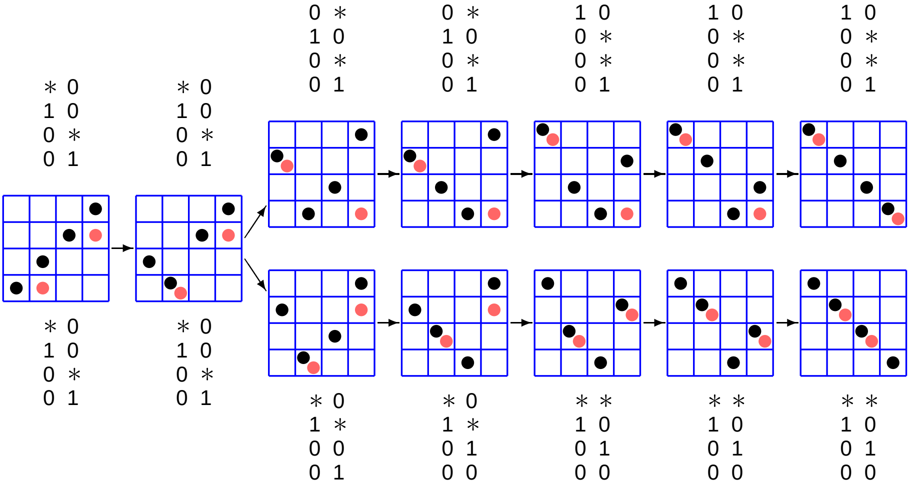
verifying that all solutions are real

```
>>> for sol in targetsols: print sol
...
t :  1.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x21 : -2.14842656584288E-02   1.49674823765163E-49
 x32 : -2.35413492464019E-01   5.55935059699177E-49
 x42 : -2.24678274447365E+00  -1.71056941445901E-49
 x31 : -5.66020448454766E-01  -4.27642353614751E-50
== err :  2.085E-16 = rco :  2.477E-02 = res :  1.370E-16 =
t :  1.00000000000000E+00   0.00000000000000E+00
m : 1
the solution for t :
 x21 : -8.66066335430695E-03  -3.06534039071054E-46
 x32 : -2.35413492464013E-01   3.94115193091355E-45
 x42 : -5.57353119320137E+00   1.40129846432482E-44
 x31 : -1.12349737224710E+00   3.50324616081204E-45
== err :  5.333E-15 = rco :  5.571E-03 = res :  1.457E-16 =
```

Outline

Checker Games

resolving [2 4][2 4][2 4][2 4]

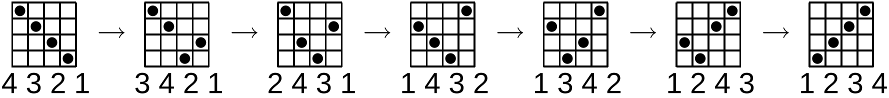


Encoding the Moves

bubble sort on $n\ n-1\ \cdots\ 2\ 1$

$$I \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ -1 & -1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$



Specialization in $\mathbb{P}^3$