Monodromy Solver: sequential and parallel

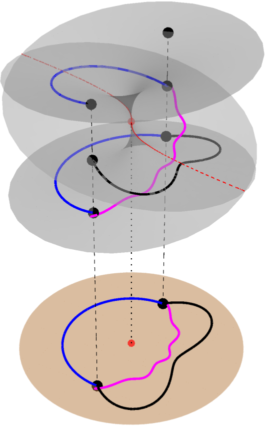
joint with
Nathan Bliss (UIC), Tim Duff (Georgia Tech), Jeff Sommars (UIC).

Anton Leykin

Georgia Tech

ISSAC 2018, New York, July 2018

Continuation in a nutshell: lifting paths from B to V.



- The covering map, $\pi : V \to B$, from total space, the set of pairs (problem, solution),

$$V = \{(a, x) \in B \times \mathbb{C} \mid x^3 - a = 0\}$$

to base space, a parameterized space of problems,

$$B = \{a \in \mathbb{C} \mid x^3 - a = 0 \text{ has 3 solutions}\}$$
$$= \mathbb{C} \setminus D, \text{ where } D = \{0\} \text{ is the branch locus.}$$

- Target system F and start system G are points in the base space B.

Example from yesterday

- Bilinear system (Bender, Faugère, Mantzaflaris, Tsigaridas)

$$\begin{cases} f_1 := 7x_0y_0 - 8x_0y_1 - x_1y_0 + 2x_1y_1 \\ f_2 := -5x_0y_0 + 7x_0y_1 - x_1y_0 - x_1y_1 \\ f_3 := -6x_0z_0 + 9x_0z_1 - x_1z_0 - 2x_1z_1 \end{cases}$$

- Parametric system that gives a branched covering (previous slide).

$$\begin{cases} g_1 := a_1x_0y_0 + a_2x_0y_1 + a_3x_1y_0 + a_4x_1y_1 \\ g_2 := a_5x_0y_0 + a_6x_0y_1 + a_7x_1y_0 + a_8x_1y_1 \\ g_3 := a_9x_0z_0 + a_{10}x_0z_1 + a_{11}x_1z_0 + a_{12}x_1z_1 \end{cases}$$

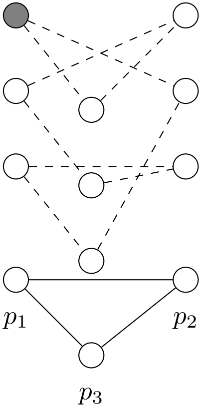
- Coefficient-parameter homotopy: to solve F, solve a generic instance of G and continue the solutions.

Books and software

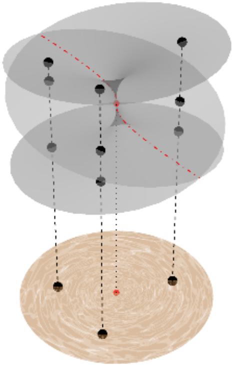
- Books:
  - Morgan, Solving polynomial systems using continuation for engineering and scientific problems (1987)
  - Allgower and Georg, Introduction to Numerical Continuation Methods (2003)
  - Sommese and Wampler, The numerical solution of systems of polynomials (2005).
- Software:
  - PHCpack (Verschelde);
  - Bertini (group of Sommese);
  - HOM4PS-3 (Tianran Chen);
  - HomotopyContinuation.jl (Breiding, Timme);
  - NumericalAlgebraicGeometry for Macaulay2 (L.).
    New: MonodromySolver (Duff, Hill, Jensen, Lee, L., Sommars).

Definitions

solution graph



homotopy graph
$G = (V, E)$

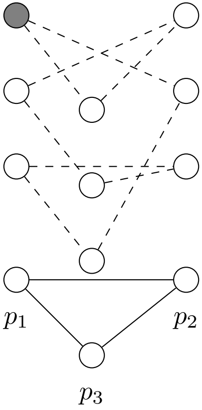


A state $x = (Q, C, A)$ of the algorithm:

- Q_v are known solutions at $v \in V$
- one-to-one correspondence C_e between subsets of Q_v and Q_w for an edge $e = \{v, w\} \in E$
- $A = \{t_1, \ldots, t_k\}$, atomic (homotopy path tracking) tasks being processed

Algorithm

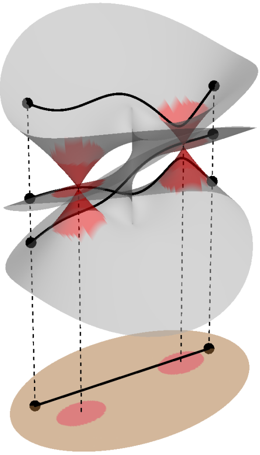
solution graph



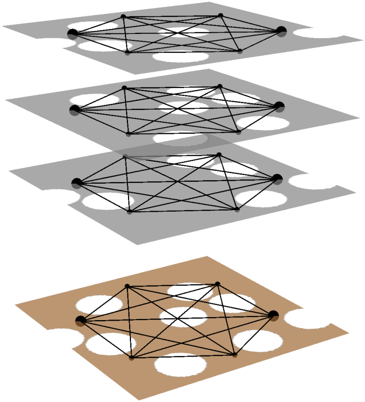
homotopy graph
$G = (V, E)$

- Create a random graph G along with a nontrivial initial state $x = (Q, C, A)$; e.g., compute a "seed".
- **while** a stopping criterion is not satisfied **do**
    - Pick an edge $\vec{e} = (w, v)$ according to an edge selection strategy.
    - $t \leftarrow (s, \vec{e})$
    - Update the state: $x \leftarrow (Q, C, A \cup \{t\})$.
    - Run homotopy continuation for task t.
    - **if** some task is finished **then**
        - Update Q, C, A accordingly.
    - **end if**
- **end while**

Segment homotopy and errors



Inflamed areas represent the "numerical hell".



- Total number of layers = degree d of the problem.
- Number of layers "perforated" by excluding one inflamed region is two (generically).

Probability model

- Main assumption: Edges induce uniformly random correspondences.
- For a task $t = (s, \vec{e}) \in A$, it translates to

$$\Pr(\mathrm{sol}_t \notin Q_{\mathrm{dest}(\vec{e})}) = \frac{d - |Q_{\mathrm{dest}(\vec{e})}|}{d - |C_e|},$$

  where sol_t is the solution obtained by following $\vec{e}$.
- Needed for a potential function in an edge selection strategy is

  $\mathbf{E}_v(Q, C, A)$ = *expected total number of known solutions at all vertices* $v \in V$ *after completing all tasks* $t \in A$.

- The basic update rule:

$$\mathbf{E}_v(Q, C, A \cup \{t\}) = \mathbf{E}_v(Q, C, A) + \Pr(\mathrm{sol}_t \notin \mathrm{sol}_A).$$

  where (assuming no failures)

$$\Pr(\mathrm{sol}_t \notin \mathrm{sol}_A) = \frac{d - \mathbf{E}_v(Q, C, A)}{d - |C_e| - \#\{t' \in A : \mathrm{edge}(t') = \vec{e}\}}.$$

...assuming failures

- Let $\alpha \in [0, 1]$ be the success rate.
- A state includes additional data F where $F_{\vec{e}}$ consists of $s \in Q_{\mathrm{src}(e)}$ such that the task $(s, \vec{e})$ has completed with a failure.
- Update formula:

$$\mathbf{E}_v(Q, C, A \cup \{t\}, F) = \mathbf{E}_v(Q, C, A, F) + \alpha \times \frac{\Big(d - \mathbf{E}_v(Q, C, A, F)\Big)\left(1 - \mathbb{E}\ \frac{\#F_{\mathrm{edge}(t)} + B}{d - \#C_e - \#\{t' \in A | \mathrm{edge}(t') = \vec{e}\} + B}\right)}{d - \#C_e - \#F_{\mathrm{edge}(t)} - \#\{t' \in A \mid \mathrm{edge}(t') = \vec{e}\}}.$$

where B a random variable with a binomial distribution:

$$B \sim \mathrm{Bin}(\#\{t' \in A : \mathrm{edge}(t') = \vec{e}\}, 1 - \alpha).$$

Experiments

- We built a simulator that given a timing for each edge of the solution graph, which can be used with
  - actual timings (e.g., from `PHCpack` and NumericalAG in `Macaulay2`)
  - fabricated timings (e.g., negative binomial distribution)

- Parallel performance:
  - The algorithm is not embarrassingly parallelizable, but...
  - ... asymptotically (as degree goes to ∞) the speedups are linear.
- Analysis of the α-threshold for the "global success/failure" (assuming a model with path failures):
  - theoretical analysis gives lower and upper bounds ...
  - ... the experiments approximate the actual threshold.
- Potential functions and edge selection: new update formulas $\mathbf{E}_v$ were used in comparison of several heuristic selection strategies.

Monodromy Solver summary

We have:

- A general framework for a randomized heuristic polynomial system solver.
- The cost (in terms of atomic tasks) is linear in the number of solutions.
- One implementation following this framework: `MonodromySolver.m2`

The present ISSAC article addressed

- parallelization,
- analysis of the modified algorithm in the presence errors,
- a simulator (built for extensive experimenting).