Factoring Solution Sets of Polynomial Systems in Parallel

Jan Verschelde

Department of Math, Stat & CS
University of Illinois at Chicago
Chicago, IL 60607-7045, USA

Email: jan@math.uic.edu

URL: http://www.math.uic.edu/~jan

Workshop on High Performance Scientific and Engineering Computing (HPSEC-05)

Oslo, Norway, 15 June 2005

Joint work with Anton Leykin.

Factoring Positive Dimensional Solution Sets

(Outline of the Talk)

1. Determine the number of irreducible factors and their degrees.
   e.g: $x^2 + y^2 - 1$ is irreducible, while $x^2 - y^2 = (x - y)(x + y)$.

2. Monodromy certified by Linear Traces:
   $\longrightarrow$ **homotopy algorithms scale well as degrees grow.**

3. Performance of a first Parallel Implementation
   $\longrightarrow$ **the algorithms are no longer embarrassingly parallel!**

4. A Probabilistic Complexity Study
   $\longrightarrow$ the study outlines future modifications of the algorithm.

Some Background Literature

D.C.S. Allison, A. Chakraborty, and L.T. Watson: **Granularity issues for solving polynomial systems via globally convergent algorithms on a hypercube.** ***J. of Supercomputing***, 3:5–20, 1989.

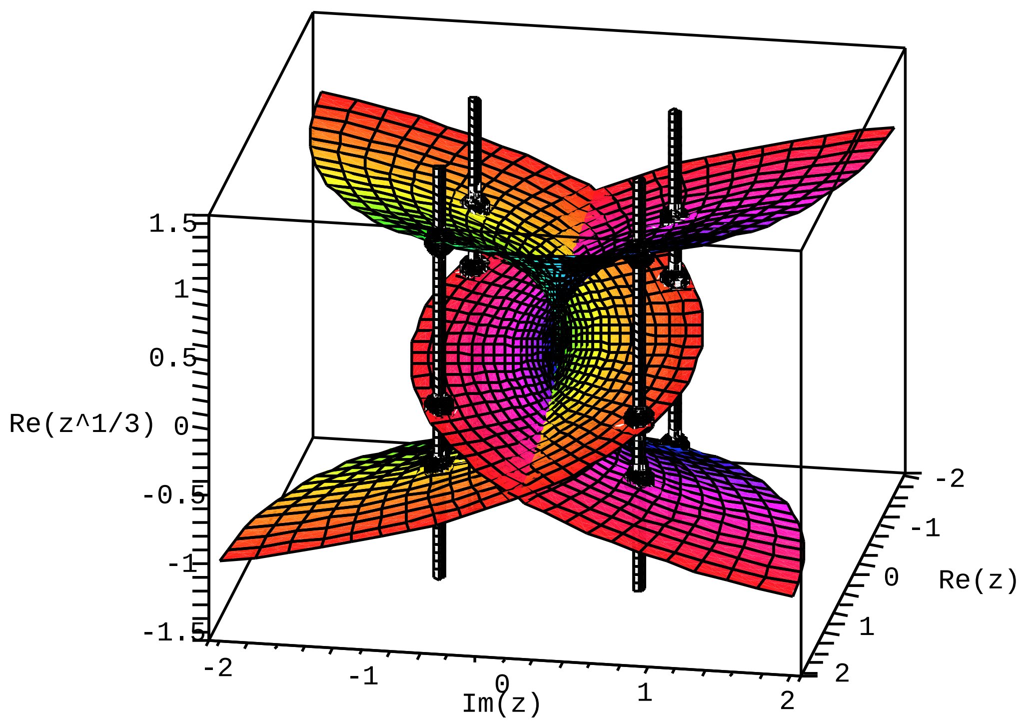
G. Chèze and A. Galligo: **Four lectures on polynomial absolute factorization.** In ***Solving Polynomial Equations: Foundations, Algorithms, and Applications***, pages 339–392. Springer–Verlag, 2005.

A.J. Sommese, J. Verschelde, and C.W. Wampler: **Using monodromy to decompose solution sets of polynomial systems into irreducible components.** In ***Application of Algebraic Geometry to Coding Theory, Physics and Computation***, pages 297–315. Kluwer, 2001.

A.J. Sommese, J. Verschelde, and C.W. Wampler: **Symmetric functions applied to decomposing solution sets of polynomial systems.** ***SIAM J. Numer. Anal.***, 40(6):2026–2046, 2002.

A.J. Sommese, J. Verschelde, and C.W. Wampler: **Numerical irreducible decomposition using PHCpack.** In ***Algebra, Geometry, and Software Systems***, pages 109–130. Springer–Verlag, 2003.

The Riemann Surface of $z^3 - w = 0$:



Loop around the singular point (0,0) permutes the points.

Generating Loops by Homotopies

W_L represents a k-dimensional solution set of $f(\mathbf{x}) = \mathbf{0}$, cut out by k random hyperplanes L. For k other hyperplanes K, we move W_L to W_K, using the **homotopy** $\boldsymbol{h}_{L,K,\boldsymbol{\alpha}}(\mathbf{x}, t) = 0$, from $t = 0$ to 1:

$$\boldsymbol{h}_{L,K,\boldsymbol{\alpha}}(\mathbf{x}, t) = \left(\begin{array}{c} f(\mathbf{x}) \\ \boldsymbol{\alpha}(1-t)L(\mathbf{x}) + tK(\mathbf{x}) \end{array} \right) = \mathbf{0}, \quad \boldsymbol{\alpha} \in \mathbb{C}.$$

The constant $\boldsymbol{\alpha}$ is chosen at random, to avoid singularities, as $t < 1$.

To turn back we generate another random constant $\boldsymbol{\beta}$, and use

$$\boldsymbol{h}_{K,L,\boldsymbol{\beta}}(\mathbf{x}, t) = \left(\begin{array}{c} f(\mathbf{x}) \\ \boldsymbol{\beta}(1-t)K(\mathbf{x}) + tL(\mathbf{x}) \end{array} \right) = \mathbf{0}, \quad \boldsymbol{\beta} \in \mathbb{C}.$$

A permutation of points in W_L occurs only among points on the same irreducible component.

Linear Traces as Stop Criterium

$$\text{Consider} \quad f(x, y(x)) = (y - y_1(x))(y - y_2(x))(y - y_3(x)) = y^3 - \boldsymbol{t_1(x)} y^2 + t_2(x) y - t_3(x)$$

We are interested in **the linear trace**: $\boldsymbol{t_1(x) = c_1 x + c_0}$.
Sample the cubic at $x = x_0$ and $x = x_1$. The samples are $\{(x_0, y_{00}), (x_0, y_{01}), (x_0, y_{02})\}$ and $\{(x_1, y_{10}), (x_1, y_{11}), (x_1, y_{12})\}$.

$$\text{Solve} \quad \begin{cases} y_{00} + y_{01} + y_{02} = c_1 x_0 + c_0 \\ y_{10} + y_{11} + y_{12} = c_1 x_1 + c_0 \end{cases} \quad \text{to find} \quad c_0, c_1.$$

With t_1 we can predict the sum of the y's for a fixed choice of x.
For example, samples at $x = x_2$ are $\{(x_2, y_{20}), (x_2, y_{21}), (x_2, y_{22})\}$.
Then, $t_1(x_2) = c_1 x_2 + c_0 = y_{20} + y_{21} + y_{22}$.
If $\neq$, then samples come from irreducible curve of degree > 3.

Monodromy Breakup certified by Linear Traces

Input: W_L, d, N *witness set, degree, #loops*

Output: $\mathcal{P}$ *partitioned witness set*

---

0. initialize $\mathcal{P}$ with d singletons; ***done by master node***
1. generate two slices L' and L'' parallel to L; *broadcast data to nodes*
2. **track d paths** for witness set with L'; *executed **in parallel***
3. **track d paths** for witness set with L''; *executed **in parallel***
4. **for** k **from** 1 **to** N **do**
   - 4.1 generate new slices K and a random α; *broadcast K and α*
   - 4.2 **track d paths** defined by $h_{L,K,\alpha}(\mathbf{x}, t) = \mathbf{0}$; *executed **in parallel***
   - 4.3 generate a random β; *broadcast β to nodes*
   - 4.4 **track d paths** defined by $h_{K,L,\beta}(\mathbf{x}, t) = \mathbf{0}$; *executed **in parallel***
   - 4.5 compute the permutation and update $\mathcal{P}$; ***done by master node***
   - 4.6 **exit when** linear trace test certifies $\mathcal{P}$.

A Benchmark Example: cyclic 8-roots

The system

$$f(\mathbf{x}) = \begin{cases} f_i = \displaystyle\sum_{j=0}^{7} \prod_{k=1}^{i} x_{(k+j) \bmod 8} = 0, \quad i = 1, 2, \ldots, 7 \\ f_8 = x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 - 1 = 0 \end{cases}$$

has 1152 isolated solutions and a solution curve of degree 144, which breaks up into 16 irreducible factors.

There are 8 factors of degree 16, and 8 quadratic factors.

Our equipment consists of one workstation with two dual 2.4Ghz processors, running Linux, and serving two Rocketcalc clusters: one with four and an other with eight 2.4Ghz processors. So we have a **total of 14 processors: a master node and 13 slave nodes.**

Computational Results

- Fluctuations in **work loads** and influence of **number of loops** needed:

#L	4	5	6	7	7	7	7	7	8	9
min	6.0	7.8	9.2	10.1	10.3	10.9	10.9	10.7	11.8	12.3
max	9.9	11.5	12.8	15.4	15.1	14.7	14.1	14.5	16.3	16.9
total	11.7	14.9	16.9	19.2	19.3	19.5	19.7	20.3	21.9	23.4

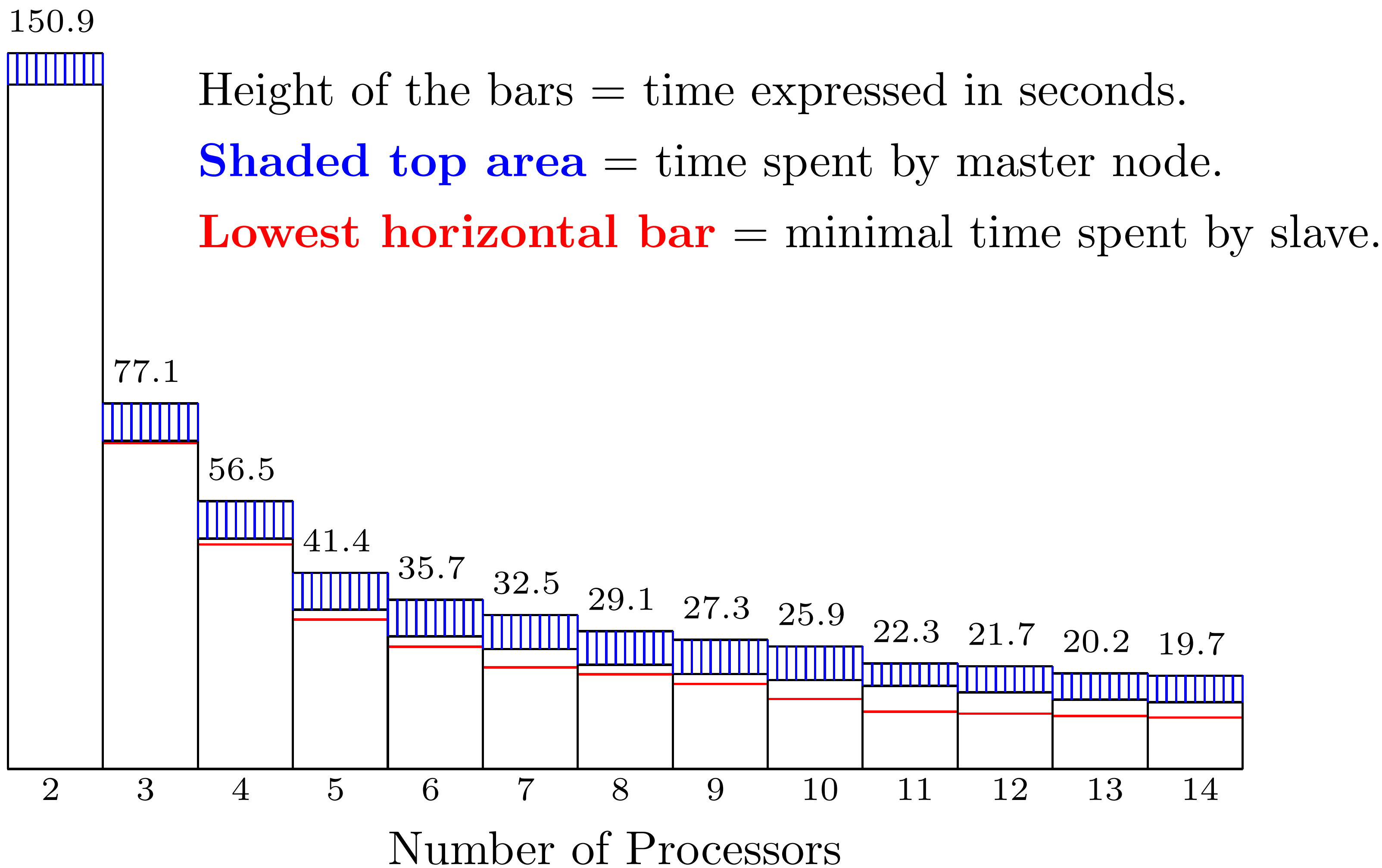
Results of 10 runs on 14 processors. **#L = number of loops**, **min** and **max** are the minimal and maximal time (in seconds) spent by the slave nodes.

- **Speedup**:

NP	2	3	4	5	6	7	8	9	10	11	12	13	14
min	—	68.7	47.4	31.5	25.8	21.5	20.0	18.0	14.8	12.1	11.7	11.2	10.9
max	144.3	69.2	48.6	33.6	28.0	25.3	22.0	20.1	18.8	17.6	16.2	14.7	14.1
total	150.9	77.1	56.5	41.4	35.7	32.5	29.1	27.3	25.9	22.3	21.7	20.2	19.7

Execution times for number of processors NP, from 2 to 14, **using 7 loops**.

Performance of a First Parallel Implementation

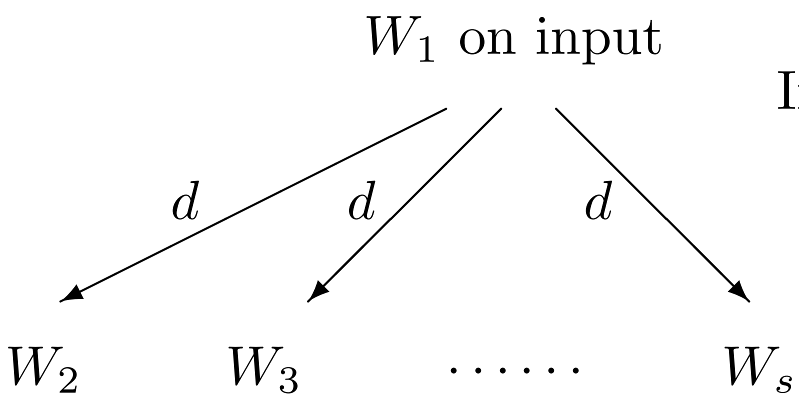


Probabilistic Complexity Study on an Irreducible Set

Denote s = number of witness sets used;

$W_i = d$ isolated solutions to $\left\{ f(\mathbf{x}) = \mathbf{0}, L^{(i)}(\mathbf{x}) = \mathbf{0} \right\}$, $i = 1, 2, \ldots, s$;

ν_{ij} = number of paths between W_i and W_j.



Initialization of graph with weights ν_{ij}

As long as not all points are connected, do

1) pick a point of a minimal set in partition of W_1;
2) pick a minimal edge ν_{ij};
3) compute the probability of getting connected;
4) update partition after a coin toss.

Results of a Simulation

The **probability** that p will get connected to q in tracking a path from W_i to W_j is modeled by the formula

$$prob_{(i,j)}(p,q) = \frac{1}{d(1 + r\ln(1+\nu_{ij}))}, \quad r > 0.$$

NP $\backslash s$	2	5	6	7	8	9
1	1.0	1.0	1.0	1.0	1.0	1.0
11	11.1	11.9	9.53	9.95	11.1	10.7
21	18.4	21.2	16.7	16.5	19.1	19.0
31	24.8	28.4	21.9	21.5	26.6	26.1
41	30.6	32.2	29.5	26.9	34.3	32.1
51	34.2	39.0	30.6	29.2	37.5	38.1

speedups for NP processors, s = #witness sets

Discussion and Conclusions

The performance of our first parallel implementation is okay for medium sized degrees on our cluster of 14 processors.

Our probabilistic model shows the tradeoff between using more witness sets (more work) and obtaining connections faster.

Hybrid approaches place greater emphasis on the application of linear traces to group witness points along irreducible factors.