# A Blackbox Polynomial System Solver on Parallel Shared Memory Computers

Jan Verschelde

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science

The 20th Workshop on Computer Algebra in Scientific Computing
17-21 September 2018, Lille, France

# Outline

1. Introduction
   - problem statement
   - the cyclic *n*-roots systems

2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets

3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - solving the cyclic 12-roots system

# A Parallel Blackbox Polynomial System Solver

1. Introduction
   - problem statement
   - the cyclic $n$-roots systems

2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets

3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - solving the cyclic 12-roots system

# problem statement

## executive summary

The problem:
*Solve a polynomial system on a computer with multicore processors.*

1. What does *solve* mean?
    - compute a numerical irreducible decomposition,
    - in blackbox mode: fixed algorithms, tolerances, and parameters.
2. What type of *shared memory parallelism*?
    - pipelining to interlace root counting with path tracking,
    - multithreading: apply dynamic load balancing to path trackers.

Focus on one benchmark: the cyclic $n$-roots problems, $n = 12$.

Free and Open Source Software: QDlib, MixedVol, DEMiCs, PHCpack.

Extend blackbox solver in PHCpack: `phc -B -tp`, with `p` threads.

# A Parallel Blackbox Polynomial System Solver

1. Introduction
   - problem statement
   - the cyclic $n$-roots systems

2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets

3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - solving the cyclic 12-roots system

# the cyclic 4-roots system

The equations for the cyclic 4-roots problem are

$$\mathbf{f}(\mathbf{x}) = \begin{cases} x_1 + x_2 + x_3 + x_4 = 0 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 = 0 \\ x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_1 + x_4x_1x_2 = 0 \\ x_1x_2x_3x_4 - 1 = 0. \end{cases}$$

This system has two solution curves of degree two.
There are no isolated solutions.

**Lemma (Backelin's Lemma)**

*If n has a quadratic divisor, $n = \ell k^2$, $\ell < k$,*
*then there are $(k-1)$-dimensional cyclic n-roots.*

We focus on $n = 8, 9, 12$.

# the embedded cyclic 4-roots system

To compute generic points on the solution curves, the system is augmented by one linear equation and one slack variable $z_1$.

$$E_1(\mathbf{f}(\mathbf{x}), z_1) = \left\{ \begin{array}{rcl} x_1 + x_2 + x_3 + x_4 + \gamma_1 z_1 & = & 0 \\ x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_1 + \gamma_2 z_1 & = & 0 \\ x_1 x_2 x_3 + x_2 x_3 x_4 + x_3 x_4 x_1 + x_4 x_1 x_2 + \gamma_3 z_1 & = & 0 \\ x_1 x_2 x_3 x_4 - 1 + \gamma_4 z_1 & = & 0 \\ c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4 + z_1 & = & 0. \end{array} \right.$$

The constants $\gamma_1$, $\gamma_2$, $\gamma_3$, $\gamma_4$ and $c_0$, $c_1$, $c_2$, $c_3$, $c_4$ are randomly generated complex numbers.

The 4 solutions with $z_1 = 0$ are generic points on the solution curves.

# related work

- J.C. Faugère. **Finding all the solutions of Cyclic 9 using Gröbner basis techniques.** *Computer Mathematics - Proceedings of the Fifth Asian Symposium (ASCM 2001)*, pages 1–12. World Scientific, 2001.
- R. Sabeti. **Numerical-symbolic exact irreducible decomposition of cyclic-12.** *LMS Journal of Computation and Mathematics*, 14:155–172, 2011.
- T. Chen, T.-L. Lee, and T.-Y. Li. **Hom4PS-3: a parallel numerical solver for systems of polynomial equations based on polyhedral homotopy continuation methods.** In *Mathematical Software – ICMS 2014*, pages 183–190. Springer-Verlag, 2014.
- G. Malajovich. **Computing mixed volume and all mixed cells in quermassintegral time.** *Found. Comput. Math.*, 17(5):1293–1334, 2017.

# A Parallel Blackbox Polynomial System Solver

1. Introduction
   - problem statement
   - the cyclic $n$-roots systems

2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets

3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - solving the cyclic 12-roots system

# polyhedral homotopies in parallel

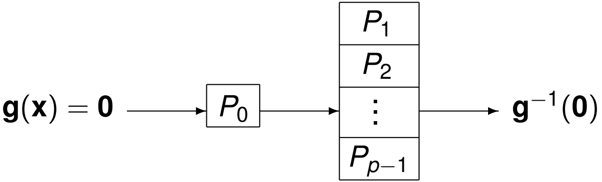
Polyhedral homotopies solve *sparse* polynomial systems.

1. The mixed volume provides a generically sharp root count.
2. Polyhedral homotopies track as many paths as the mixed volume.

The Bézout bound for the cyclic 12-roots problem is 479,001,600.
A linear-product bound lowers this to 342,875,319.
The mixed volume of the cyclic 12-roots system equals 500,352.

# a 2-stage pipeline

The input to the pipeline is a random coefficient system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ and the output are its solutions in the set $\mathbf{g}^{-1}(\mathbf{0})$.



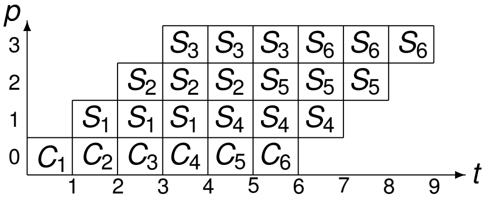

A 2-stage pipeline:

1. thread $P_0$ computes the cells which define paths; and
2. $p-1$ threads $P_1$, $P_2$, ..., $P_{p-1}$ track paths to solve $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.

# a space time diagram

Suppose the subdivision of the Newton polytopes has six cells.
For regularity, assume

1. producing one cell takes one time unit; and
2. solving a start system takes 3 time units.




A space time diagram for a 2-stage pipeline:

1. one thread produces 6 cells $C_1$, $C_2$, ..., $C_6$; and
2. 3 threads solve 6 start systems $S_1$, $S_2$, ..., $S_6$.

The total time equals 9 units. It takes 24 time units sequentially.
This pipeline with 4 threads gives a speedup of $24/9 \approx 2.67$.

# speedup

Consider a scenario with $p$ threads:

- the first thread produces $n$ cells; and
- the other $p - 1$ threads track all paths corresponding to the cells.

Assume that tracking all paths for one cell costs $F$ times the amount of time it takes to produce that one cell.
The sequential time $T_1$, parallel time $T_p$, and speedup $S_p$ are

$$T_1 = n + Fn, \quad T_p = p - 1 + \frac{Fn}{p-1}, \quad S_p = \frac{T_1}{T_p} = \frac{n(1+F)}{p - 1 + \frac{Fn}{p-1}}.$$

The *pipeline latency* is $p - 1$, the time to fill up the pipeline with jobs.

**Theorem**

*If $F = p - 1$, then $S_p = p$ for $n \rightarrow \infty$.*

# times with pipelined polyhedral homotopies

We run `phc` on a two 22-core 2.20 GHz processors.

Times of the pipelined polyhedral homotopies
on the embedded cyclic 12-roots system, 983,952 paths,
for increasing values 2, 4, 8, 16, 32, 64 of the tasks `p`:

| p | seconds | = | hms format | speedup |
|---|---|---|---|---|
| 2 | 62812.764 | = | 17h26m52s | 1.00 |
| 4 | 21181.058 | = | 5h53m01s | 2.97 |
| 8 | 8932.512 | = | 2h28m53s | 7.03 |
| 16 | 4656.478 | = | 1h17m36s | 13.49 |
| 32 | 4200.362 | = | 1h10m01s | 14.95 |
| 64 | 4422.220 | = | 1h13m42s | 14.20 |

The mixed volume computation is done by `MixedVol`,
ACM TOMS Algorithm 846 of T. Gao, T.Y. Li, and M. Wu.

# dynamic enumeration for mixed cells

`DEMiCs` of T. Mizutani and A. Takeda computes all mixed cells in a greedy manner, at a faster pace than `MixedVol`.

Times of the pipelined polyhedral homotopies with DEMiCs, on the embedded cyclic 12-roots system, 983,952 paths, for increasing values 2, 4, 8, 16, 32, 64 of tasks `p`.

| p | seconds | = | hms format | speedup |
|---:|---:|---|---:|---:|
| 2 | 56614 | = | 15h43m34s | 1.00 |
| 4 | 21224 | = | 5h53m44s | 2.67 |
| 8 | 9182 | = | 2h23m44s | 6.17 |
| 16 | 4627 | = | 1h17m07s | 12.24 |
| 32 | 2171 | = | 36m11s | 26.08 |
| 64 | 1989 | = | 33m09s | 28.46 |

The last time is an average over 13 runs. With 64 threads the times ranged between 23 minutes and 47 minutes.

# A Parallel Blackbox Polynomial System Solver

1. Introduction
   - problem statement
   - the cyclic $n$-roots systems

2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets

3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - solving the cyclic 12-roots system

# an example

Consider the following system:

$$\mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{l} (x_1 - 1)(x_1 - 2)(x_1 - 3)(x_1 - 4) = 0 \\ (x_1 - 1)(x_2 - 1)(x_2 - 2)(x_2 - 3) = 0 \\ (x_1 - 1)(x_1 - 2)(x_3 - 1)(x_3 - 2) = 0 \\ (x_1 - 1)(x_2 - 1)(x_3 - 1)(x_4 - 1) = 0. \end{array} \right.$$

In its factored form, the numerical irreducible decomposition shows:

1. The three dimensional solution set is defined by $x_1 = 1$.
2. For $x_1 = 2$, $x_2 = 1$ defines a two dimensional solution set.
3. There are twelve lines:
    1. $(2, 2, x_3, 1)$, $(2, 2, 1, x_4)$, $(2, 3, 1, x_4)$, $(2, 3, x_3, 1)$,
    2. $(3, 1, 1, x_4)$, $(3, 1, 2, x_4)$, $(3, 2, 1, x_4)$, $(3, 3, 1, x_4)$,
    3. $(4, 1, 1, x_4)$, $(4, 1, 2, x_4)$, $(4, 2, 1, x_4)$, $(4, 3, 1, x_4)$.
4. 4 isolated points: $(3, 2, 2, 1)$, $(3, 3, 2, 1)$, $(4, 3, 2, 1)$, $(4, 2, 2, 1)$.

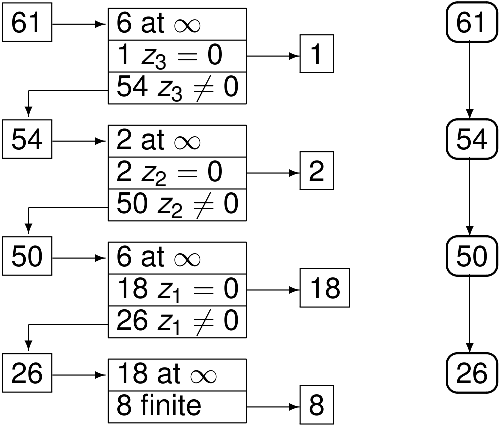
# the top dimensional system

Because the top dimension is three, three random hyperplanes and three slack variables $z_1$, $z_2$, $z_3$ are added:

$$\left\{ \begin{array}{r} (x_1 - 1)(x_1 - 2)(x_1 - 3)(x_1 - 4) + \gamma_{1,1} z_1 + \gamma_{1,2} z_2 + \gamma_{1,3} z_3 = 0 \\ (x_1 - 1)(x_2 - 1)(x_2 - 2)(x_2 - 3) + \gamma_{2,1} z_1 + \gamma_{2,2} z_2 + \gamma_{2,3} z_3 = 0 \\ (x_1 - 1)(x_1 - 2)(x_3 - 1)(x_3 - 2) + \gamma_{3,1} z_1 + \gamma_{3,2} z_2 + \gamma_{3,3} z_3 = 0 \\ (x_1 - 1)(x_2 - 1)(x_3 - 1)(x_4 - 1) + \gamma_{4,1} z_1 + \gamma_{4,2} z_2 + \gamma_{4,3} z_3 = 0 \\ c_{1,1} x_1 + c_{1,2} x_2 + c_{1,3} x_3 + c_{1,4} + z_1 = 0 \\ c_{2,1} x_1 + c_{2,2} x_2 + c_{2,3} x_3 + c_{2,4} + z_2 = 0 \\ c_{3,1} x_1 + c_{3,2} x_2 + c_{3,3} x_3 + c_{3,4} + z_3 = 0 \end{array} \right.$$

Solving this system will give one generic point on $x_1 = 1$ and start solutions for the lower dimensional solution sets.

# a cascade of homotopies



# speedup

Assume that every path requires the same amount of time.

## Theorem

*Let $T_p$ be the time it takes to track $n$ paths with $p$ threads. Then, the optimal speedup $S_p$ is*

$$S_p = p - \frac{p - r}{T_p}, \quad r = n \bmod p.$$

*If $n < p$, then $S_p = n$.*

## Corollary

*Let $T_p$ be the time it takes to track with $p$ threads a sequence of $n_0$, $n_1$, ..., $n_D$ paths. Then, the optimal speedup $S_p$ is*

$$S_p = p - \frac{dp - r_0 - r_1 - \cdots - r_D}{T_p}, \quad r_k = n_k \bmod p, k = 0, 1, \ldots D.$$

# A Parallel Blackbox Polynomial System Solver

1. Introduction
   - problem statement
   - the cyclic $n$-roots systems
2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets
3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - solving the cyclic 12-roots system

# homotopy membership tests

The solution line $x_1 = 1$ is represented by a generic point.
The isolated solution is the point $(2, 0)$.

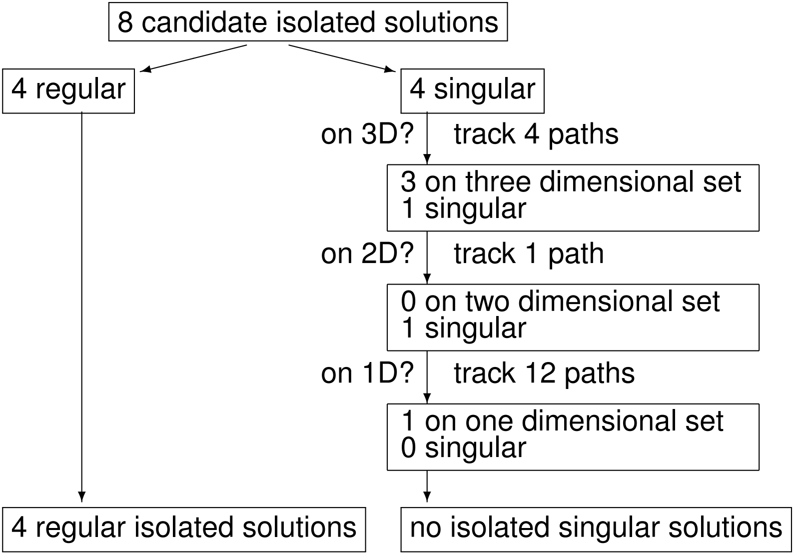
A homotopy membership verifies $(2, 0)$ does not belong to the line.

$$\mathbf{h}(\mathbf{x}, z_1, t) = \left\{ \begin{array}{rcl} (x_1 - 1)(x_1 - 2) + \gamma_1 z_1 & = & 0 \\ (x_1 - 1)x_2^2 + \gamma_2 z_1 & = & 0 \\ (1-t)c_0 + tc_3 + c_1 x_1 + c_2 x_2 + z_1 & = & 0. \end{array} \right.$$

For $t = 0$, the constant $c_0$ is so that $(2, 0)$ satisfies the last equation.

Tracking the path starting at $(2, 0, 0)$ computes another generic point on $x_1 = 1$ if $(2, 0)$ would lie on the solution line $x_1 = 1$.

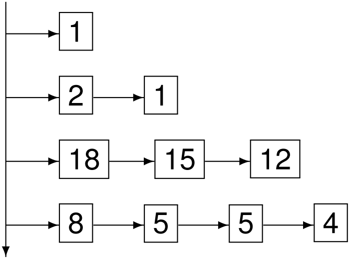
# testing singular candidate isolated points

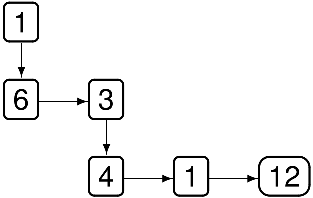


# cascades of homotopy membership filters

Points on higher dimensional solution sets are removed.

On the example, we have one linear 3-dimensional set,
one linear 2-dimensional set, 12 lines, and 4 isolated points.




The numbers at the right equal the number of paths in each stage.

# speedup

## Corollary

*Let $T_p$ be the time it takes to filter $n_D$, $n_{D-1}$, ..., $n_{\ell+1}$ singular points on components respectively of dimensions $D$, $D-1$, ..., $\ell+1$ and degrees $d_D$, $d_{D-1}$, ..., $d_{\ell+1}$. Then, the optimal speedup is*

$$S_p = p - \frac{(D-\ell)p - r_D - r_{D-1} - \cdots - r_{\ell+1}}{T_p}, \quad r_k = (n_k d_k) \bmod p,$$

*for* $k = \ell+1, \ldots, D-1, D$.

There are two reasons for a reduced parallelism:

1. The number of singular solutions and the degrees of the solution sets could be smaller than the number of available cores.
2. To filter the output of the cascade, there are $D(D+1)/2$ stages, so longer sequences of homotopies are considered.

# A Parallel Blackbox Polynomial System Solver

1. Introduction
   - problem statement
   - the cyclic *n*-roots systems
2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets
3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - solving the cyclic 12-roots system

# solving cyclic 8- and cyclic 9-roots systems

Both cyclic 8 and cyclic 9-roots are relatively small problems.

Wall clock times in seconds with `phc -B -tp` for `p` threads:

| | cyclic 8-roots | | cyclic 9-roots | |
|---|---|---|---|---|
| p | seconds | speedup | seconds | speedup |
| 1 | 181.765 | 1.00 | 2598.435 | 1.00 |
| 2 | 167.871 | 1.08 | 1779.939 | 1.46 |
| 4 | 89.713 | 2.03 | 901.424 | 2.88 |
| 8 | 47.644 | 3.82 | 427.800 | 6.07 |
| 16 | 32.215 | 5.65 | 267.838 | 9.70 |
| 32 | 22.182 | 8.19 | 153.353 | 16.94 |
| 64 | 20.103 | 9.04 | 150.734 | 17.24 |

With 64 threads, we can solve the cyclic 9-roots problem faster than solving the cyclic 8-roots problem on one thread.

# running in double double and quad double precision

Double double and quad double arithmetic are implemented in QDlib, a software library by Y. Hida, X. S. Li, and D. H. Bailey, 2001.

In double precision, with 64 threads, the time

- for cyclic 8-roots reduces from 3 minutes to 20 seconds and
- for cyclic 9-roots from 43 minutes to 2 minutes and 30 seconds.

The wall clock times below are with 64 threads in higher precision.

| | cyclic 8-roots | | | cyclic 9-roots | | |
|---|---|---|---|---|---|---|
| | seconds | = | hms format | seconds | = | hms format |
| dd | 53.042 | = | 53s | 498.805 | = | 8m19s |
| qd | 916.020 | = | 15m16s | 4761.258 | = | 1h19m21s |

With 64 threads, we can compensate for the computational overhead caused by double double precision and achieve *quality up*.

# A Parallel Blackbox Polynomial System Solver

1. Introduction
   - problem statement
   - the cyclic *n*-roots systems

2. A Parallel Numerical Irreducible Decomposition
   - pipelining to solve the top dimensional system
   - computing lower dimensional solution sets
   - filtering lower dimensional solution sets

3. Computational Experiments
   - solving the cyclic 8- and cyclic 9-roots systems
   - **solving the cyclic 12-roots system**

# solving the cyclic 12-roots system in parallel

The wall clock time on the blackbox solver on one thread is about 95 hours (almost 4 days), which includes the linear-product bound.
The time reduces from 4 days to less than 3 hours with 64 threads:

| | solving top system | | | cascade and filter | | | grand | |
|---|---|---|---|---|---|---|---|---|
| p | start | contin | total | cascade | filter | total | total | speedup |
| 2 | 62813 | 47667 | 110803 | 44383 | 2331 | 46714 | 157518 | 1.00 |
| 4 | 21181 | 25105 | 46617 | 24913 | 1558 | 26471 | 73089 | 2.16 |
| 8 | 8933 | 14632 | 23896 | 13542 | 946 | 14488 | 38384 | 4.10 |
| 16 | 4656 | 7178 | 12129 | 6853 | 676 | 7529 | 19657 | 8.01 |
| 32 | 4200 | 3663 | 8094 | 3415 | 645 | 4060 | 12154 | 12.96 |
| 64 | 4422 | 2240 | 7003 | 2228 | 557 | 2805 | 9808 | 16.06 |

The solving of the top dimensional system breaks up in two stages:

- the solving of a start system (start) and the
- continuation to the solutions of the top dimensional system (contin).

Speedups are good in the cascade, but the filtering contains the factorization in irreducible components, which does not run in parallel.

# running in double double precision

A run in double double precision with 64 threads ends after 7 hours and 37 minutes.

This time lies between the times in double precision

- with 8 threads, 10 hours and 39 minutes, and
- with 16 threads, 5 hours and 27 minutes.

Confusing quality with precision, from 8 to 64 threads,
the working precision can be doubled,
with a reduction in time by 3 hours, from 10.5 hours to 7.5 hours.

# conclusions

## and future directions

A numerical irreducible decomposition for the cyclic 12-roots system can be computed in less than 3 hours with 64 threads,

- using pipelining to solve the top dimensional system,
- with multitasking for the cascade and filtering stages.

In double double precision, it takes 7.5 hours with 64 threads.

Two future directions:

1. apply GPU acceleration in the blackbox solver,
2. apply pipelining in all stages of the solver.