Computing Power Series Solutions of Polynomial Systems

Jan Verschelde
joint work with Nathan Bliss

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science
http://www.math.uic.edu/~jan
emails: janv@uic.edu and nbliss2@uic.edu

Graduate Computational Algebraic Geometry Seminar, 25 August 2016.

Outline

1. Polynomial Homotopy Continuation
   - problem statement
   - truncated power series
2. Gauss-Newton on Truncated Power Series
   - Householder QR decomposition on series
   - series developments for cyclic 8-roots
3. Symbolic-Numeric Path Tracking
   - Gauss-Newton power series predictor
   - stages in one step of the path tracker
4. Linearization applied to Matrix Series
   - series with matrix coefficients
   - solving a linear matrix series system

polynomial homotopy continuation

$f(\mathbf{x}) = \mathbf{0}$ is a polynomial system we want to solve,
$g(\mathbf{x}) = \mathbf{0}$ is a start system (g is similar to f) with known solutions.

A homotopy $h(\mathbf{x}, t) = (1 - t)g(\mathbf{x}) + t\,f(\mathbf{x}) = \mathbf{0}$, $t \in [0, 1]$,
to solve $f(\mathbf{x}) = \mathbf{0}$ defines solution paths $\mathbf{x}(t)$: $h(\mathbf{x}(t), t) \equiv \mathbf{0}$.

Numerical continuation methods track the paths $\mathbf{x}(t)$, from $t = 0$ to 1.

Predictor-corrector method operate in two stages:

1. The predictor sets the new value for t and predicts $\mathbf{x}(t)$.
2. The corrector applies Newton's method to $h(\mathbf{x}, t) = \mathbf{0}$.

problem statement

Current predictor methods apply higher-order extrapolation,

- which may cause path crossing: the predicted point lies so close to another path that it gets corrected to that other path;
- which may not be sufficient to reach convergence in the corrector.

Our solution: apply Newton's method on truncated power series.

numerical analysis and symbolic computation

- E. L. Allgower and K. Georg: Introduction to Numerical Continuation Methods. Volume 45 of *Classics in Applied Mathematics*, SIAM, 2003.
- A. Morgan: Solving polynomial systems using continuation for engineering and scientific problems. Volume 57 of *Classics in Applied Mathematics*, SIAM, 2009.

Newton-Hensel iteration is discussed in the following:

- J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein: Deformation techniques for efficient polynomial equation solving. *Journal of Complexity* 16(1):70-109, 2000.
- D. Castro, L.M. Pardo, K. Hägele, and J.E. Morais, Kronecker's and Newton's Approaches to Solving: A First Comparison. *Journal of Complexity* 17(1):212-303 2001.
- A. Bompadre, G. Matera, R. Wachenchauzer, and A. Waissbein: Polynomial equation solving by lifting procedures for ramified fibers. *Theoretical Computer Science* 315(2-3):335-369, 2004.

truncated power series

A series $s(t)$ in t with coefficients $s_k \in \mathbb{C}$:

$$s(t) = s_0 + s_1 t + s_2 t^2 + \cdots + s_n t^n + O(t^{n+1}),$$

is truncated to a polynomial of degree n, after dropping $O(t^{n+1})$.

The inverse $x(t)$ of $s(t)$ is defined via $x(t) \times s(t) = 1 + O(t^{n+1})$.

The coefficients x_k of the inverse $x(t)$ are computed as

$$\begin{array}{rcl} x_0 & = & 1/s_0 \\ x_1 & = & -(s_1 x_0)/s_0 \\ x_2 & = & -(s_1 x_1 + s_2 x_0)/s_0 \\ & \vdots & \\ x_n & = & -(s_1 x_{n-1} + s_2 x_{n-2} + \cdots + s_n x_0)/s_0 \end{array}$$

Newton's method on truncated power series

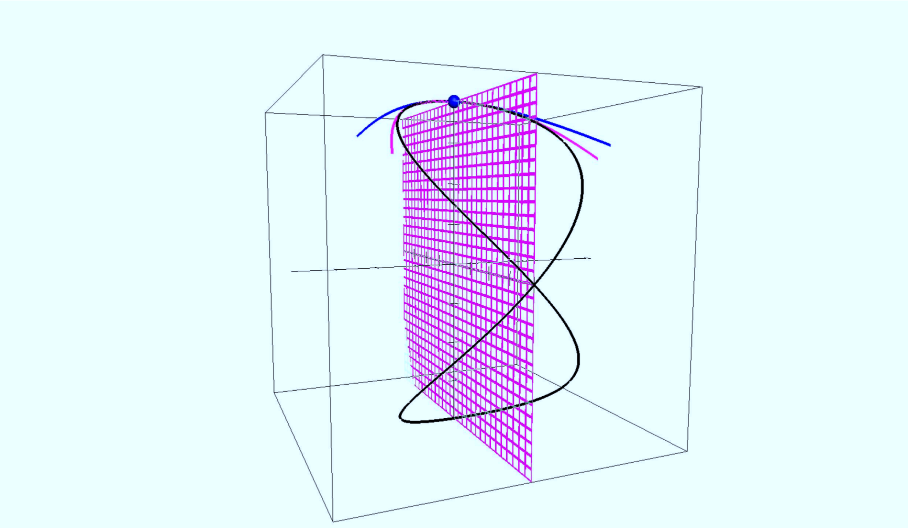
Given $c = c_0 + c_1 t + c_2 t^2 + \cdots + c_n t^n$, compute $\sqrt{c}$.

Apply Newton's method on the equation $x^2 - c = 0$,
doubling the degrees of the truncated power series in each step:

$$
\begin{array}{l}
x := \sqrt{c_0} + x_1 t \\
c := c_0 + c_1 t \\
k := 1 \\
\text{while } (k \leq n) \text{ do} \\
\quad \Delta x := (x^2 - c)/(2x) \\
\quad x := x - \Delta x \\
\quad x := x + x_{k+1} t^{k+1} + \cdots + x_{2k} t^{2k} \\
\quad c := x + c_{k+1} t^{k+1} + \cdots + c_{2k} t^{2k} \\
\quad k := 2 \times k
\end{array}
$$

Quadratic convergence: the order of Δx doubles in each step.

the Viviani curve



computing a power series solution

$$h(x(t), y(t), z(t), t) = \left\{ \begin{array}{r} (1-t)y + t(y-1) = 0 \\ x^2 + y^2 + z^2 - 4 = 0 \\ (x-1)^2 + y^2 - 1 = 0 \end{array} \right.$$

After 3 steps with Newton's method:

$$\begin{array}{rcl} y & = & t \\ x & = & 0.5t^2 \\ z & = & 2 - 0.25t^2 \end{array}$$

After 4 steps with Newton's method:

$$\begin{array}{rcl} y & = & t \\ x & = & 0.5t^2 + 0.125t^4 + 0.0625t^6 + 0.03125t^8 \\ z & = & 2 - 0.25t^2 - 0.078125t^4 - 0.041015625t^6 - 0.020751953125t^8 \end{array}$$

Gauss-Newton on truncated power series

Orthogonality is defined via an inner product on vectors:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \bar{u}_1 v_1 + \bar{u}_2 v_2 + \cdots + \bar{u}_n v_n, \quad ||\mathbf{u}||_2^2 = \langle \mathbf{u}, \mathbf{u} \rangle.$$

To make a vector $\mathbf{x}$ parallel to $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$:

$$\mathbf{v} = \mathbf{x} + ||\mathbf{x}||_2 \mathbf{e}_1, \quad H\mathbf{x} = \mathbf{x} - \frac{2\langle \mathbf{v}, \mathbf{x} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{v},$$

where H is a Householder transformation.

To transform a matrix A into an upper triangular matrix R, apply a sequence of Householder transformations:

$$H_n H_{n-1} \cdots H_1 A = R, \quad Q = H_1 \cdots H_{n-1} H_n, \quad A = QR.$$

This is well defined for matrices of truncated power series.

biunimodular vectors and cyclic n-roots

$$\left\{ \begin{array}{c} x_0 + x_1 + \cdots + x_{n-1} = 0 \\ i = 2, 3, 4, \ldots, n-1 : \displaystyle\sum_{j=0}^{n-1} \prod_{k=j}^{j+i-1} x_{k \bmod n} = 0 \\ x_0 x_1 x_2 \cdots x_{n-1} - 1 = 0. \end{array} \right.$$

The system arises in the study of biunimodular vectors.
A vector $\mathbf{u} \in \mathbb{C}^n$ of a unitary matrix A is biunimodular if for $k = 1, 2, \ldots, n$: $|u_k| = 1$ and $|v_k| = 1$ for $\mathbf{v} = A\mathbf{u}$.

- J. Backelin: *Square multiples n give infinitely many cyclic n-roots.* Technical Report, 1989.
- H. Führ and Z. Rzeszotnik. On biunimodular vectors for unitary matrices. *Linear Algebra and its Applications* 484:86–129, 2015.

series developments for cyclic 8-roots

Cyclic 8-roots has solution curves not reported by Backelin.

With Danko Adrovic (ISSAC 2012, CASC 2013): a tropism is $\mathbf{v} = (1, -1, 0, 1, 0, 0, -1, 0)$, the leading exponents of the series.

The corresponding unimodular coordinate transformation $\mathbf{x} = \mathbf{z}^M$ is

$$M = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{aligned} x_0 &= z_0 \\ x_1 &= z_1 z_0^{-1} \\ x_2 &= z_2 \\ x_3 &= z_3 z_0 \\ x_4 &= z_4 \\ x_5 &= z_5 \\ x_6 &= z_6 z_0^{-1} \\ x_7 &= z_7. \end{aligned}$$

Solving $\mathrm{in}_{\mathbf{v}}(\mathbf{f})(\mathbf{x} = \mathbf{z}^M) = \mathbf{0}$ gives the leading term of the series.

version 2.4.21 of PHCpack and 0.5.0 of phcpy

The source code (GNU GPL License) is availale at `github`.

After 2 Newton steps with `phc -u`, the series for z_1:

```
(-1.25000000000000E+00 + 1.25000000000000E+00*i)*z0^2
+( 5.00000000000000E-01 - 2.37676980513323E-17*i)*z0
+(-5.00000000000000E-01 - 5.00000000000000E-01*i);
```

After 3 Newton steps with `phc -u`, the series for z_1:

```
( 7.12500000000000E+00 + 7.12500000000000E+00*i)*z0^4
+(-1.52745512076048E-16 - 4.25000000000000E+00*i)*z0^3
+(-1.25000000000000E+00 + 1.25000000000000E+00*i)*z0^2
+( 5.00000000000000E-01 - 1.45255178343636E-17*i)*z0
+(-5.00000000000000E-01 - 5.00000000000000E-01*i);
```

Gauss-Newton power series predictor

To correct a solution, apply Gauss-Newton in complex arithmetic, on vectors of complex numbers.

The predictor is symbolic-numeric:

- Gauss-Newton on truncated power series: $\mathbf{x}(t)$, where $\mathbf{x}(t)$ is a vector of series, each series is of degree n.
- Step control via evaluation of the series, $\mathbf{y}(t) = f(\mathbf{x}(t))$. Let k be the order of $\mathbf{y}(t)$, $k < n$.

Let $\epsilon > 0$ be the tolerance on the residual $||f(\mathbf{x}(t))||$.

To compute the step size τ, solve $\epsilon = |y_k|\tau^k$:

$$\tau = \left(\frac{\epsilon}{|y_k|} \right)^{1/k}.$$

one method to predict *and* correct

Polynomials in the homotopy, with support A, have the form

$$h(\mathbf{x}, t) = \sum_{\mathbf{a} \in A} c_{\mathbf{a}}(t) \mathbf{x}^{\mathbf{a}}, \quad \mathbf{x}^{\mathbf{a}} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}.$$

where the coefficients $c_{\mathbf{a}}(t)$ are truncated power series.

Three stages in one step with the path tracker:

1. Given a tolerance $\epsilon > 0$, set the step size τ: $||h(\mathbf{x}(\tau), \tau)||$ is $O(\epsilon)$.
2. Shift the coefficient series $c_{\mathbf{a}}(t)$ into $c_{\mathbf{a}}(t - \tau)$ in the homotopy.
3. Correct the solution with Newton's method on series of degree 0. Continue with truncated power series of increasing degrees to compute solution series $\mathbf{x}(t)$, accurate up to a prescribed order.

Linearization

Consider a ***vector of power series*** $\mathbf{v}(t)$, truncated to degree two:

$$\mathbf{v}(t) = \begin{bmatrix} \mathbf{v}_1(t) \\ \mathbf{v}_2(t) \\ \mathbf{v}_3(t) \end{bmatrix} = \begin{bmatrix} v_{1,0} + v_{1,1}t + v_{1,2}t^2 \\ v_{2,0} + v_{2,1}t + v_{2,2}t^2 \\ v_{3,0} + v_{3,1}t + v_{3,2}t^2 \end{bmatrix}.$$

We can rewrite $\mathbf{v}(t)$ as

$$\mathbf{v}(t) = \underbrace{\begin{bmatrix} v_{1,0} \\ v_{2,0} \\ v_{3,0} \end{bmatrix}}_{\mathbf{v}_0} + \underbrace{\begin{bmatrix} v_{1,1} \\ v_{2,1} \\ v_{3,1} \end{bmatrix}}_{\mathbf{v}_1} t + \underbrace{\begin{bmatrix} v_{1,2} \\ v_{2,2} \\ v_{3,2} \end{bmatrix}}_{\mathbf{v}_2} t^2.$$

Then we have a ***power series vector*** $\mathbf{v}(t)$, truncated to degree two:

$$\mathbf{v}(t) = \mathbf{v}_0 + \mathbf{v}_1 t + \mathbf{v}_2 t^2, \quad \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{C}^3.$$

A truncated power series vector is a *vector polynomial*.

Vector Series and Matrix Series

Instead of working with a vector of power series,
consider a power series with vectors as coefficients.

A vector series $\mathbf{v}(t)$ is a series with vectors as coefficients:

$$\mathbf{v}(t) = \mathbf{v}_0 + \mathbf{v}_1 t + \mathbf{v}_2 t^2 + \cdots, \quad \mathbf{v}_k \in \mathbb{C}^{n \times n}.$$

A vector series, truncated to degree d,
is represented by a column vector $\mathbf{v} = [\mathbf{v}_0 \ \mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_d]^T$, $\mathbf{v}_k \in \mathbb{C}^n$.

A matrix series $\mathbf{A}(t)$ is a series with matrices as coefficients:

$$\mathbf{A}(t) = A_0 + A_1 t + A_2 t^2 + \cdots, \quad A_k \in \mathbb{C}^{n \times n}.$$

A matrix series, truncated to degree d,
is represented by a row vector $\mathbf{A} = [A_0 \ A_1 \ A_2 \ \cdots \ A_d], A_k \in \mathbb{C}^{n \times n}$.

A truncated matrix series vector is a *matrix polynomial*.

Matrix Series and Linear Systems

Given a row vector $\mathbf{A} = [A_0 \ A_1 \ A_2 \ \cdots \ A_d], A_k \in \mathbb{C}^{n \times n}$,
which represents a matrix series, truncated to degree d.

Given a column vector $\mathbf{b} = [\mathbf{b}_0 \ \mathbf{b}_1 \ \mathbf{b}_2 \ \cdots \ \mathbf{b}_d]^T$, $\mathbf{b}_k \in \mathbb{C}^n$,
which represents a vector series, truncated to degree d.

Denote $\mathbf{A}(t)\mathbf{x}(t) = \mathbf{b}(t)$, with $\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{x}_1 t + \mathbf{x}_2 t^2 + \cdots + \mathbf{x}_d t^d$:

$$\left[A_0 \ A_1 t \ A_2 t^2 \ \cdots \ A_d t^d \right] \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 t \\ \mathbf{x}_2 t^2 \\ \vdots \\ \mathbf{x}_d t^d \end{bmatrix} = \mathbf{b}_0 + \mathbf{b}_1 t + \mathbf{b}_2 t^2 + \cdots + \mathbf{b}_d t^d$$

as a linear system of vector and matrix series.

Solving a Linear Matrix Series System

Consider $\mathbf{A}(t)\mathbf{x}(t) = \mathbf{b}(t)$, for $d = 2$:

$$\begin{bmatrix} A_0 & A_1 t & A_2 t^2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 t \\ \mathbf{x}_2 t^2 \end{bmatrix} = \mathbf{b}_0 + \mathbf{b}_1 t + \mathbf{b}_2 t^2$$

Expanding along powers of t, ignoring powers higher than two:

$$\begin{aligned} t^0 &: \quad A_0 \mathbf{x}_0 = \mathbf{b}_0 \\ t^1 &: \quad A_0 \mathbf{x}_1 + A_1 \mathbf{x}_0 = \mathbf{b}_1 \\ t^2 &: \quad A_0 \mathbf{x}_2 + A_1 \mathbf{x}_1 + A_2 \mathbf{x}_0 = \mathbf{b}_2 \end{aligned}$$

which gives a triangular system. Suppose A_0 is invertible:

$$\begin{aligned} \mathbf{x}_0 &= A_0^{-1} \mathbf{b}_0 \\ \mathbf{x}_1 &= A_0^{-1} (\mathbf{b}_1 - A_1 \mathbf{x}_0) \\ \mathbf{x}_2 &= A_0^{-1} (\mathbf{b}_2 - A_1 \mathbf{x}_1 - A_2 \mathbf{x}_0) \end{aligned}$$

Numerical Linear Algebra

A power series is invertible if the leading coefficient is invertible.

If the leading coefficient A_0 of a matrix series $\mathbf{A}(t)$ is invertible, then we solve $\mathbf{A}(t)\mathbf{x}(t) = \mathbf{b}(t)$ via

$$\begin{aligned}
\mathbf{x}_0 &= A_0^{-1}\mathbf{b}_0 \\
\mathbf{x}_1 &= A_0^{-1}(\mathbf{b}_1 - A_1\mathbf{x}_0) \\
\mathbf{x}_2 &= A_0^{-1}(\mathbf{b}_2 - A_1\mathbf{x}_1 - A_2\mathbf{x}_0) \\
&\vdots \\
\mathbf{x}_d &= A_0^{-1}(\mathbf{b}_d - A_1\mathbf{x}_{d-1} - A_2\mathbf{x}_{d-2} - \cdots - A_d\mathbf{x}_0)
\end{aligned}$$

The solving of $\mathbf{A}(t)\mathbf{x}(t) = \mathbf{b}(t)$ is in that case reduced to the solving of $d + 1$ regular linear systems, using LU, QR, or SVD.

Numerical Conditioning

We apply the following numerical algorithm:

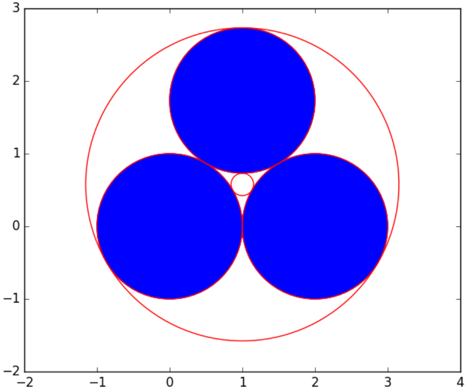
$$
\begin{aligned}
\text{solve } A_0 \mathbf{x}_0 &= \mathbf{b}_0 \\
\text{solve } A_0 \mathbf{x}_1 &= \mathbf{b}_1 - A_1 \widetilde{\mathbf{x}_0} \\
\text{solve } A_0 \mathbf{x}_2 &= \mathbf{b}_2 - A_1 \widetilde{\mathbf{x}_1} - A_2 \widetilde{\mathbf{x}_0} \\
&\vdots \\
\text{solve } A_0 \mathbf{x}_d &= \mathbf{b}_d - A_1 \widetilde{\mathbf{x}_{d-1}} - A_2 \widetilde{\mathbf{x}_{d-2}} - \cdots - A_d \widetilde{\mathbf{x}_0}
\end{aligned}
$$

Because of roundoff, solving $A_0 \mathbf{x}_0 = \mathbf{b}_0$ does not give the exact $\mathbf{x}_0$ but an approximate $\widetilde{\mathbf{x}_0}$. The approximation errors are propagated to the right hand sides of the other linear systems, so we compute not the exact $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d$ but approximate $\widetilde{\mathbf{x}_1}, \widetilde{\mathbf{x}_2}, \ldots, \widetilde{\mathbf{x}_d}$.

The condition number of A_0 predicts the size of the error.

What if A_0 is singular? Who cares?

A special position of the circles of Appolonius problem:



The given circles are double solutions to the problem.
With Puiseux power series solutions of the special position,
we can predict the solutions to perturbed instances of the problem.

conclusions

Solving polynomial systems with power series is inspired by tropical algebraic geometry. The leading exponents of series are *tropisms*.

Predicting the solution on a path defined by a homotopy with Gauss-Newton on truncated power series is promising.

One future research direction:

- shared memory parallel implementations,
- acceleration with Graphics Processing Units (GPUs),
- quality up: compensate extra cost with parallel computations.