Robust Numerical Path Tracking for Polynomial Homotopies

Jan Verschelde
joint work with Simon Telen and Marc Van Barel

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

Structured Matrix Days, 23-24 May 2019, Limoges, France

Outline

1. Problem Statement
   - adaptive step control
   - Padé approximants and power series
2. Detecting Nearby Singularities
   - applying the ratio theorem of Fabry
   - an illustrative example
3. Distance to the Nearest Path
   - an estimate based on the Jacobian and Hessian matrices
   - an illustrative example
4. Algorithm and Computational Experiments
   - an apriori step control algorithm
   - two generic polynomials
   - the 184,756 paths to all cyclic 11-roots

Robust Numerical Path Tracking

1. Problem Statement
   - adaptive step control
   - Padé approximants and power series
2. Detecting Nearby Singularities
   - applying the ratio theorem of Fabry
   - an illustrative example
3. Distance to the Nearest Path
   - an estimate based on the Jacobian and Hessian matrices
   - an illustrative example
4. Algorithm and Computational Experiments
   - an apriori step control algorithm
   - two generic polynomials
   - the 184,756 paths to all cyclic 11-roots

problem statement

To solve a polynomial system $f(x) = 0$, the homotopy

$$h(x, t) = \gamma(1 - t)g(x) + t\,f(x) = 0,$$

defines solution paths $x(t)$ satisfying $h(x(t), t) = 0$, where

- $\gamma \in \mathbb{C}$ is a random constant,
- $t \in [0, 1]$ is the continuation parameter,
- $g(x) = 0$ is the start system, with known solutions $x(0)$,
- $f(x) = 0$ is the target system, with solutions $x(1)$.

If all solutions of $g(x) = 0$ are regular, then, except for a finite number of bad choices of γ, all solution paths $x(t)$ are regular, for all $t < 1$.

Variable step size Δt control in predictor-corrector methods:

- Δt is too large: divergence and/or path jumping.
- Δt is too small: inefficient, we care only about $x(1)$.

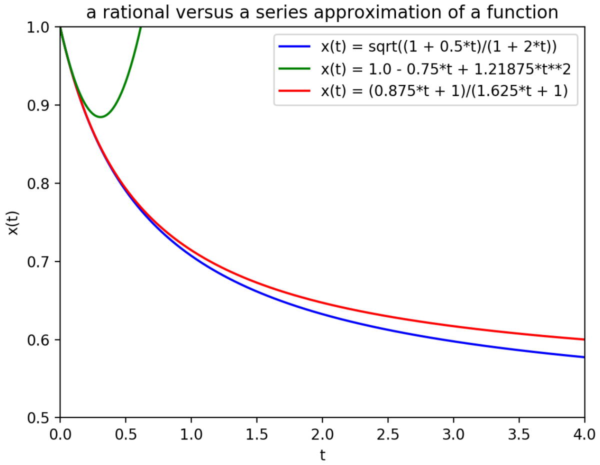
five relevant papers, in chronological order

1. E. Fabry. Sur les points singuliers d'une fonction donnée par son développement en série et l'impossibilité du prolongement analytique dans des cas très généraux. *Annales scientifiques de l'École Normale Supérieure*, 13:367–399, 1896.
2. H. Schwetlick and J. Cleve. Higher order predictors and adaptive steplength control in path following algorithms. *SIAM Journal on Numerical Analysis*, 24(6):1382–1393, 1987.
3. D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Adaptive multiprecision path tracking. *SIAM Journal on Numerical Analysis*, 46(2):722–746, 2008.
4. P. Gonnet, S. Güttel, and L. N. Trefethen. **Robust** Padé approximation via SVD. *SIAM review*, 55(1):101–117, 2013.
5. N. Bliss and J. Verschelde. The method of Gauss–Newton to compute power series solutions of polynomial homotopies. *Linear Algebra and its Applications*, 542:569–588, 2018.

Robust Numerical Path Tracking

Padé Approximants

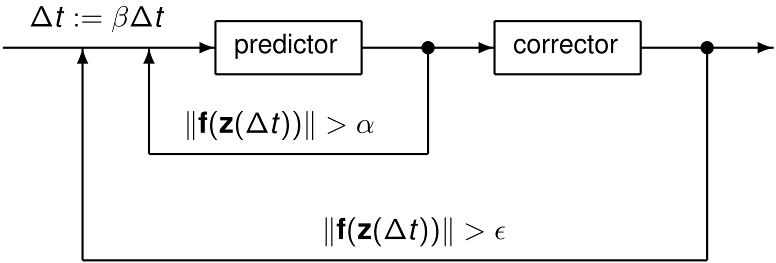
Consider the homotopy: $(1-t)(x^2-1)+t(3x^2-3/2)=0$.



G. A. Baker and P. Graves-Morris. *Padé Approximants*. Cambridge UP, 1996.

a posteriori and a priori step control

An *a posteriori* step control uses feedback loops.



Extreme choices for α and ϵ (not recommended):

- If $\alpha \leq \epsilon$, then the corrector is not needed.
- If $\alpha = \infty$, then the first feedback loop does never happen.

Setting 0.5 for β cuts the step size Δ in half.

Our goal: develop an *a priori* step control algorithm.

linearization

Working with truncated power series, computing modulo $O(t^d)$, is doing arithmetic over the field of formal series $C[[t]]$.

Linearization: consider $\mathbb{C}^n[[t]]$ instead of $\mathbb{C}[[t]]^n$. Instead of a vector of power series, we consider a power series with vectors as coefficients.

Solve $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{C}^{n \times n}[[t]]$, $\mathbf{b}, \mathbf{x} \in \mathbb{C}^n[[t]]$.

$$\begin{aligned} \mathbf{A} &= A_0 t^a + A_1 t^{a+1} + \cdots , \\ \mathbf{b} &= \mathbf{b}_0 t^b + \mathbf{b}_1 t^{b+1} + \cdots \\ \mathbf{x} &= \mathbf{x}_0 t^{b-a} + \mathbf{x}_1 t^{b-a+1} + \cdots \end{aligned}$$

where $A_i \in \mathbb{C}^{n \times n}$ and $\mathbf{b}_i, \mathbf{x}_i \in \mathbb{C}^n$.

block linear algebra

Computing the first d terms of the solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$:

$$\begin{array}{l} \left(A_0 t^a + A_1 t^{a+1} + A_2 t^{a+2} + \cdots + A_d t^{a+d}\right) \\ \cdot \ \left(\mathbf{x}_0 t^{b-a} + \mathbf{x}_1 t^{b-a+1} + \mathbf{x}_2 t^{b-a+2} + \cdots + \mathbf{x}_d t^{b-a+d}\right) \\ = \mathbf{b}_0 t^b + \mathbf{b}_1 t^{b+1} + \mathbf{b}_2 t^{b+2} + \cdots + \mathbf{b}_d t^{b+d}. \end{array}$$

Written in matrix format:

$$\left[\begin{array}{ccccc} A_0 & & & & \\ A_1 & A_0 & & & \\ A_2 & A_1 & A_0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ A_d & A_{d-1} & A_{d-2} & \cdots & A_0 \end{array} \right] \left[\begin{array}{c} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_d \end{array} \right] = \left[\begin{array}{c} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_d \end{array} \right].$$

If A_0 is regular, then solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ is straightforward.

Robust Numerical Path Tracking

detecting nearby singularities

Applying the ratio theorem of Fabry, we can detect singular points based on the coefficients of the Taylor series.

Theorem (the ratio theorem, Fabry 1896)

If for the series $x(t) = c_0 + c_1 t + c_2 t^2 + \cdots + c_n t^n + c_{n+1} t^{n+1} + \cdots,$

we have $\lim_{n \to \infty} c_n / c_{n+1} = z$, *then*

- *z is a singular point of the series, and*
- *it lies on the boundary of the circle of convergence of the series.*

Then the radius of this circle is less than $|z|$.

L. Leau in 1899 comments on the proof (Bieberbach, 1955, page 51): “*d'habiles calculs malheureusement assez complexes.*"

A proof is in the book
The Taylor Series, an introduction to the theory of functions of a complex variable, by Paul Dienes, Dover Publications, 1957.

the ratio theorem of Fabry and Padé approximants

Consider $n = 3$, $x(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4$.

$$[3/1]_x = \frac{a_0 + a_1 t + a_2 t^2 + a_3 t^3}{1 + b_1 t}$$

$$\begin{array}{rcl}
(c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4)(1 + b_1 t) & = & a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\
c_0 + \quad c_1 t + \quad c_2 t^2 + \quad c_3 t^3 + \quad c_4 t^4 & & \\
+ b_1 c_0 t + b_1 c_1 t^2 + b_1 c_2 t^3 + b_1 c_3 t^4 & = & a_0 + a_1 t + a_2 t^2 + a_3 t^3
\end{array}$$

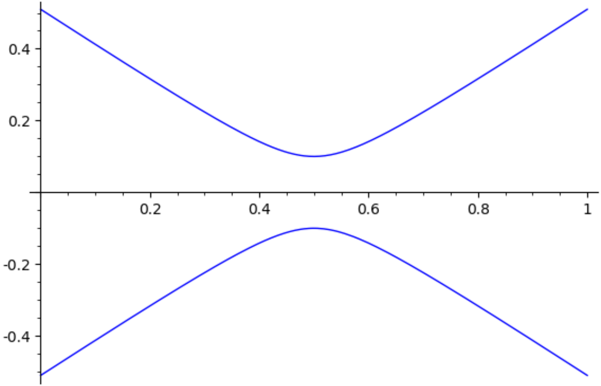
We solve for b_1 in the term for t^4: $c_4 + b_1 c_3 = 0 \Rightarrow b_1 = -c_4/c_3$.

The denominator of $[3/1]_x$ is $1 - c_4/c_3 t$. The pole of $[3/1]_x$ is c_3/c_4.

Robust Numerical Path Tracking

an illustrative example

Consider the homotopy $x^2 - (t - 1/2)^2 - p^2 = 0$, for $p > 0$.

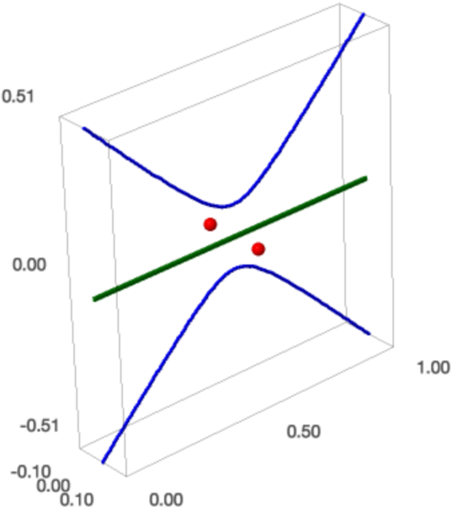


$$x(t) = \pm\sqrt{4p^2 + 4t^2 - 4t + 1}$$

R. B. Kearfott and Z. Xing. An interval step control for continuation methods. *SIAM Journal on Numerical Analysis*, 31(3):892–914, 1994.

two nearby singularities for complex values of t

$$4p^2 + 4t^2 - 4t + 1 = 0 \Rightarrow t = 1/2 \pm p\sqrt{-1}$$



poles of a [6/2]-Padé approximant

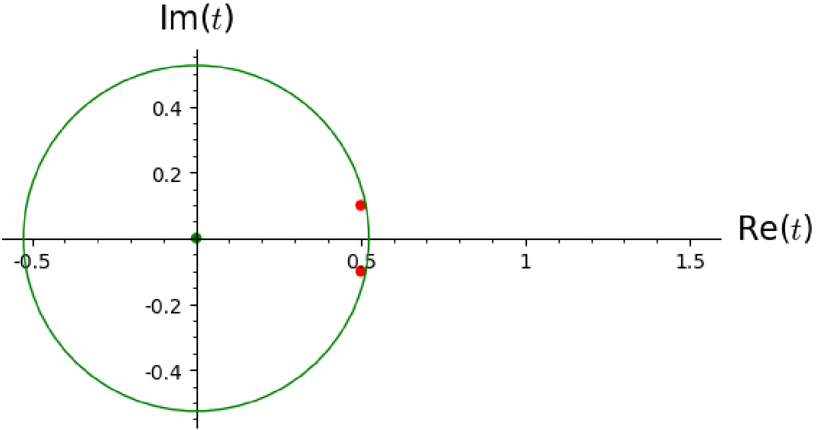
$x(t) = \sqrt{4p^2 + 4t^2 - 4t + 1}$ for $p = 0.1$
has singularities at $t = 0.5 \pm 0.1 I$.

t	poles	radius
0.0	$0.522 \pm 0.054 I$	0.525
0.1	$0.428 \pm 0.056 I$	0.431
0.2	$0.336 \pm 0.061 I$	0.341
0.3	$0.250 \pm 0.077 I$	0.261
0.4	$0.139 \pm 0.126 I$	0.188
0.5	$\pm 0.126 I$	0.126

The poles of the Padé approximant predict the distance to the nearest singularity.

approaching a nearby singularity – step 0

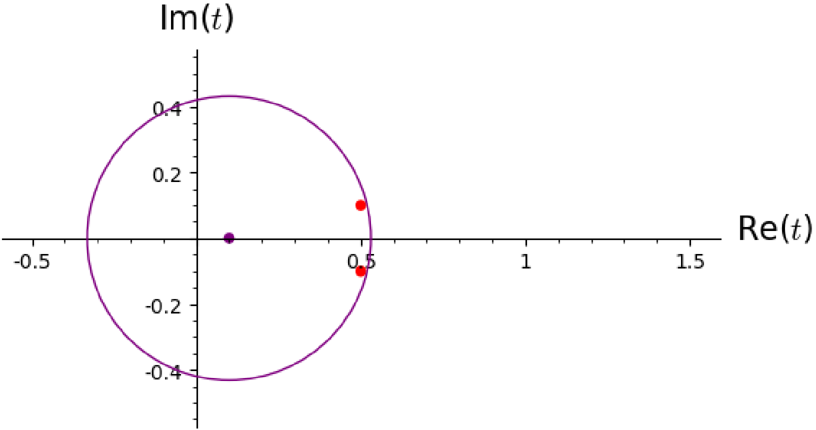
The circle centered at $t = 0.0$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

approaching a nearby singularity – step 1

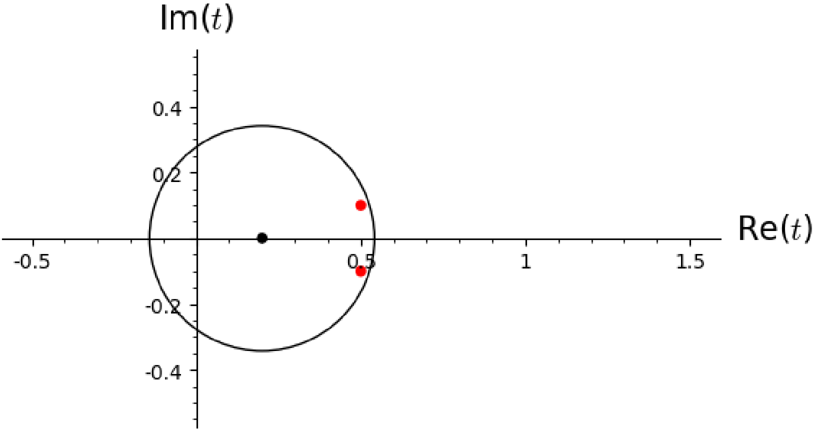
The circle centered at $t = 0.1$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

approaching a nearby singularity – step 2

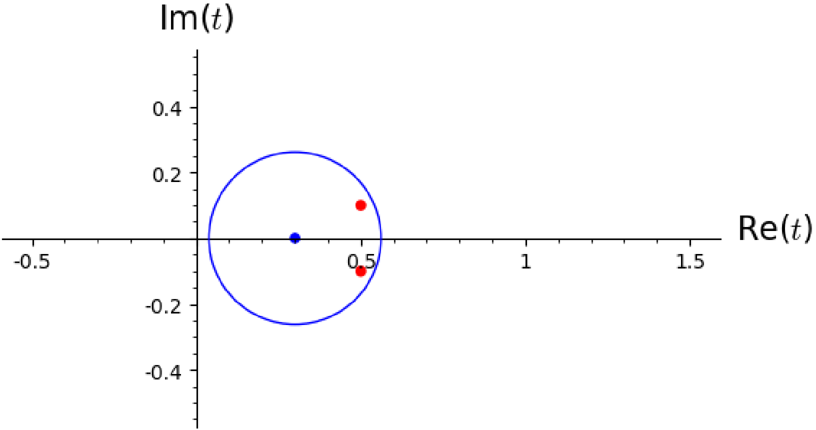
The circle centered at $t = 0.2$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

approaching a nearby singularity – step 3

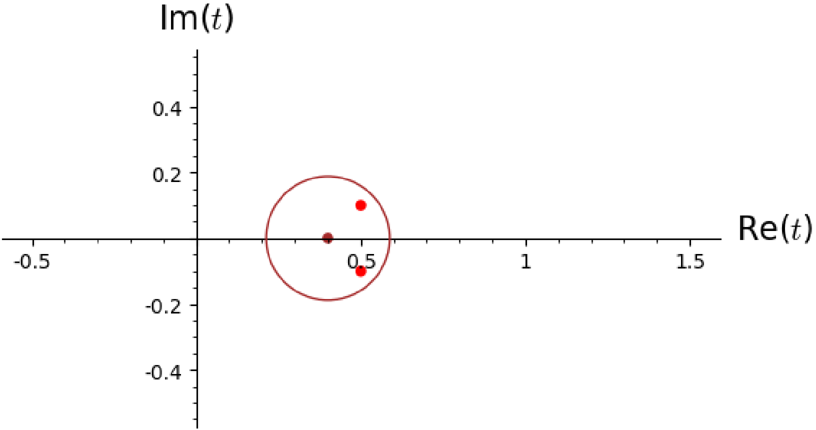
The circle centered at $t = 0.3$ has as radius the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

approaching a nearby singularity – step 4

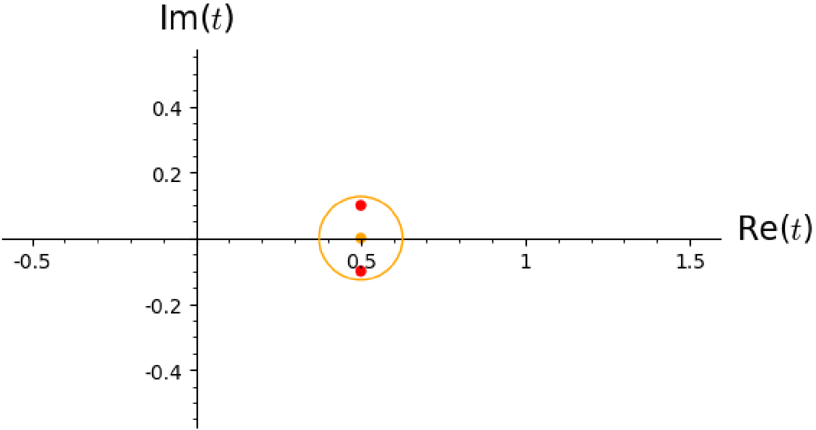
The circle centered at $t = 0.4$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

close to a nearby singularity – step 5

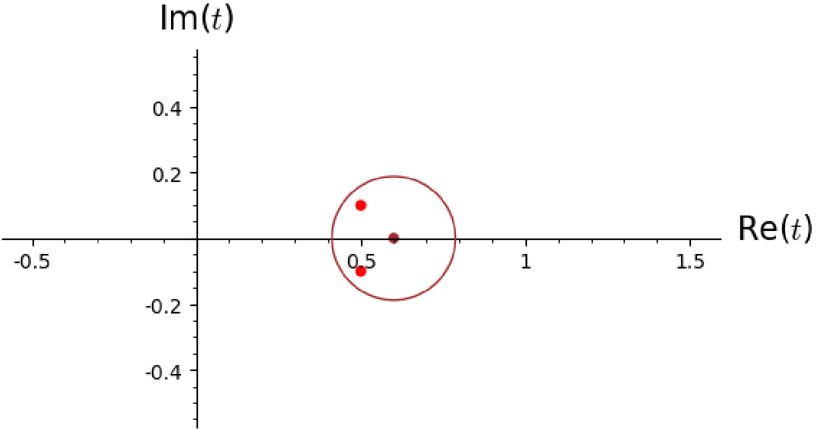
The circle centered at $t = 0.5$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

leaving a nearby singularity – step 6

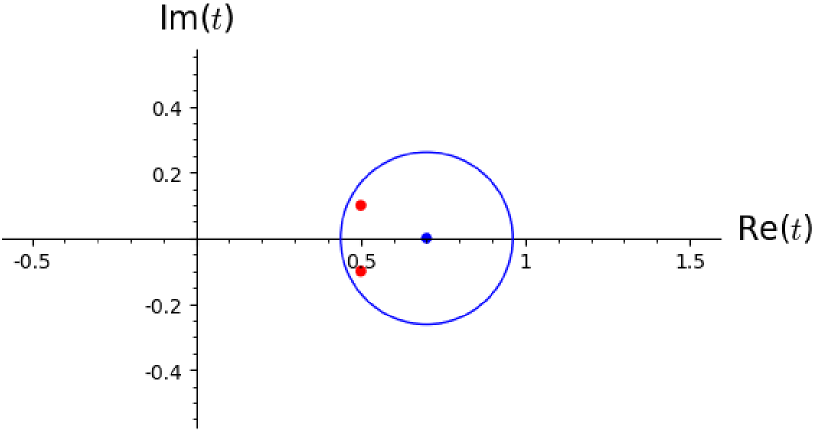
The circle centered at $t = 0.6$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

leaving a nearby singularity – step 7

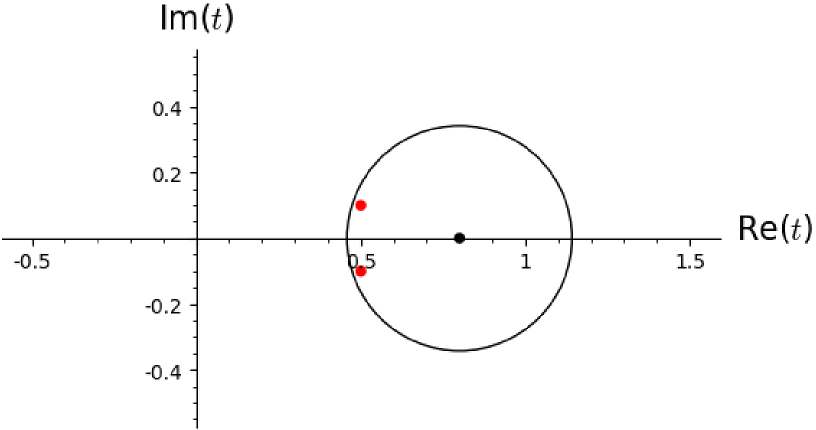
The circle centered at $t = 0.7$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

leaving a nearby singularity – step 8

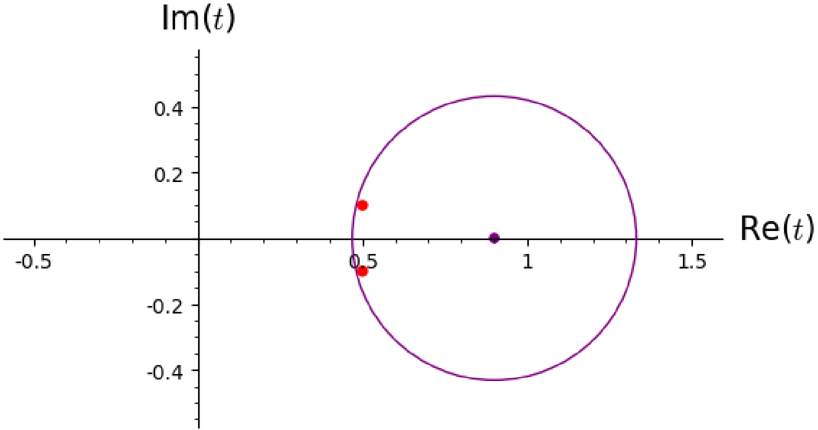
The circle centered at $t = 0.8$ has as radius the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

leaving a nearby singularity – step 9

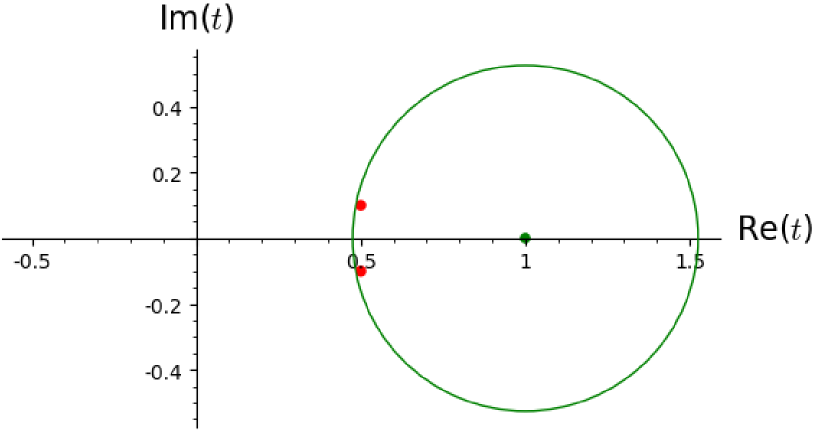
The circle centered at $t = 0.9$ has as radius the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

leaving a nearby singularity – step 10

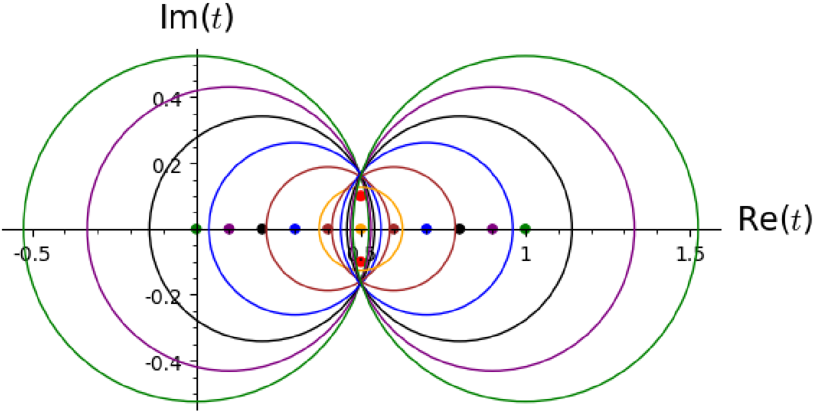
The circle centered at $t = 1.0$ has as radius
the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

approaching and leaving a nearby singularity

Circles centered at $t = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$ have as radius the computed distance to the closest pole.



The singular points at $0.5 \pm 0.1I$ are represented by the red dots.

a phcpy session with a [6/2]-Padé predictor

phcpy is the Python interface to PHCpack

```
t :  4.93828714960806E-01   0.00000000000000E+00
m : 1
the solution for t :
 x :  1.00190242833497E-01   0.00000000000000E+00
== err :  4.795E-09 = rco :  1.000E+00 = res :  1.210E-17 =

t : 4.938e-01, step : 9.383e-02, frp : 1.877e-01

closest pole :  (0.13875598086124408, -0.126341160276525б)

poles: [[(0.13875598086124408-0.1263411602765256j),
         (0.1387559808612441+0.1263411602765256j)]]
```

- The poles are computed at $t = 0.4$.
- The tracker slows down before 0.5.
- The imaginary parts of the poles are close in magnitude to $0.1 = p$.

the approximation error of the Padé approximant

Consider the Padé approximant $[L/M]_k = p_k(t)/q_k(t)$, with degrees $\deg(p_k) = L$, $\deg(q_k) = M$, $k = 1, 2, \ldots, n$, for the kth coordinate $x_k(t)$ of the solution around $t = 0$.

For small step size Δt, we estimate the error, for $\ell = L + M + 2$,

$$e_k(\Delta t) = \frac{p_k(\Delta t)}{q_k(\Delta t)} - x_k(\Delta t) = e_{0,k}(\Delta t)^\ell + O((\Delta t)^{\ell+1}).$$

Using $|e_k(\Delta t)| \approx |e_{0,k}(\Delta t)^\ell|$ leads to

$$\left\| x(\Delta t) - \left(\frac{p_1(\Delta t)}{q_1(\Delta t)}, \frac{p_2(\Delta t)}{q_2(\Delta t)}, \ldots, \frac{p_n(\Delta t)}{q_n(\Delta t)} \right) \right\| \approx \|e_0\| \, |\Delta t|^\ell,$$

with $e_0 = (e_{0,1}, e_{0,2}, \ldots, e_{0,n})$.

Robust Numerical Path Tracking

1. Problem Statement
   - adaptive step control
   - Padé approximants and power series
2. Detecting Nearby Singularities
   - applying the ratio theorem of Fabry
   - an illustrative example
3. Distance to the Nearest Path
   - an estimate based on the Jacobian and Hessian matrices
   - an illustrative example
4. Algorithm and Computational Experiments
   - an apriori step control algorithm
   - two generic polynomials
   - the 184,756 paths to all cyclic 11-roots

distance to the nearest path

Let $y(t)$ and $z(t)$ be two distinct solution paths of $h(x,t) = 0$.
We assume $z(t)$ is close to $y(t)$ and denote $\Delta z = y(t) - z(t) \in \mathbb{C}^n$.
Our goal is to estimate $\|\Delta z\|$.

$$h(y(t),t) \approx h(z(t),t) + J_h(z(t),t)\Delta z + \frac{v}{2},$$

where J_h is the Jacobian matrix,

$$v = \left[\begin{array}{c} \langle \mathcal{H}_1(z(t),t)\Delta z, \Delta z \rangle \\ \langle \mathcal{H}_2(z(t),t)\Delta z, \Delta z \rangle \\ \vdots \\ \langle \mathcal{H}_n(z(t),t)\Delta z, \Delta z \rangle \end{array} \right] \qquad \mathcal{H}_k(x,t) = \frac{\partial^2 h_k}{\partial x_\ell \partial x_m}, \quad 1 \leq \ell, m \leq n,$$

$\mathcal{H}_k$ is the Hessian of the homotopy h_k, and $\langle \cdot, \cdot \rangle$ is the inner product.

the largest singular values of the Hessians

Abbreviate $J_h(z(t), t)$ by J_h and $\mathcal{H}_k(z(t), t)$ by $\mathcal{H}_k$:

$$0 \approx 0 + J_h \Delta z + \frac{1}{2} \left[\begin{array}{c} \langle \mathcal{H}_1 \Delta z, \Delta z \rangle \\ \langle \mathcal{H}_2 \Delta z, \Delta z \rangle \\ \vdots \\ \langle \mathcal{H}_n \Delta z, \Delta z \rangle \end{array} \right].$$

The Hessian matrices are Hermitian and have a unitary diagonalization $\mathcal{H}_k = V_k \Lambda_k V_k^H$, for the Hermitian transpose $\cdot^H$, $V_k^H V_k = I$.
There is some vector w_k such that $\|w_k\| = \|\Delta z\|$ and $\Delta z = V_k w_k$.

$$\begin{array}{rcl} \langle \mathcal{H}_k \Delta z, \Delta z \rangle & = & \langle \Lambda_k w_k, w_k \rangle \\ \Rightarrow |\langle \mathcal{H}_k \Delta z, \Delta z \rangle| & \leq & \sigma_{k,1} \|w_k\|^2 = \sigma_{k,1} \|\Delta z\|^2, \end{array}$$

for the largest singular value $\sigma_{k,1}$ of $\mathcal{H}_k$.

the smallest singular value of the Jacobian matrix

Applying $|\langle \mathcal{H}_k \Delta z, \Delta z \rangle| \leq \sigma_{k,1} \|\Delta z\|^2$ leads to

$$\left\| \begin{array}{c} \langle \mathcal{H}_1 \Delta z, \Delta z \rangle \\ \langle \mathcal{H}_2 \Delta z, \Delta z \rangle \\ \vdots \\ \langle \mathcal{H}_n \Delta z, \Delta z \rangle \end{array} \right\| \leq \left\| \begin{array}{c} \sigma_{1,1} \|\Delta z\|^2 \\ \sigma_{2,1} \|\Delta z\|^2 \\ \vdots \\ \sigma_{n,1} \|\Delta z\|^2 \end{array} \right\| = \|\Delta z\|^2 \sqrt{\sigma_{1,1}^2 + \sigma_{2,1}^2 + \cdots + \sigma_{n,1}^2}.$$

As we have an upper bound for the right hand side of

$$J_h \Delta z \approx -\frac{1}{2} \left[\begin{array}{c} \langle \mathcal{H}_1 \Delta z, \Delta z \rangle \\ \langle \mathcal{H}_2 \Delta z, \Delta z \rangle \\ \vdots \\ \langle \mathcal{H}_n \Delta z, \Delta z \rangle \end{array} \right]$$

we use a lower bound for the left hand side: $\|J_h \Delta z\| \geq \sigma_n(J_h) \|\Delta z\|$,
where $\sigma_n(J_h)$ is the smallest singular value of the Jacobian matrix J_h.

an estimate for the distance to the nearest path

$$
\begin{array}{rcl}
\sigma_n(J_h)\|\Delta z\| & \leq & \|J_h \Delta z\| \\
 & \approx & \dfrac{1}{2} \left\| \begin{array}{c} \langle \mathcal{H}_1 \Delta z, \Delta z \rangle \\ \langle \mathcal{H}_2 \Delta z, \Delta z \rangle \\ \vdots \\ \langle \mathcal{H}_n \Delta z, \Delta z \rangle \end{array} \right\| \\
 & \leq & \dfrac{1}{2} \|\Delta z\|^2 \sqrt{\sigma_{1,1}^2 + \sigma_{2,1}^2 + \cdots + \sigma_{n,1}^2} \\
 & \Downarrow & \\
\dfrac{2\sigma_n(J_h)}{\sqrt{\sigma_{1,1}^2 + \sigma_{2,1}^2 + \cdots + \sigma_{n,1}^2}} & \lesssim & \|\Delta z\|
\end{array}
$$

The lower bound for $\|\Delta z\|$ decreases

- as J_h becomes close to a singular matrix, and/or
- as the curvature of the path increases.

Robust Numerical Path Tracking

1. Problem Statement
   - adaptive step control
   - Padé approximants and power series
2. Detecting Nearby Singularities
   - applying the ratio theorem of Fabry
   - an illustrative example
3. Distance to the Nearest Path
   - an estimate based on the Jacobian and Hessian matrices
   - an illustrative example
4. Algorithm and Computational Experiments
   - an apriori step control algorithm
   - two generic polynomials
   - the 184,756 paths to all cyclic 11-roots

an illustrative example

Consider the homotopy $h(x,t) = x^{p+1} - q(x - (t - 1/2)) = 0$, where

- $p \in \mathbb{Z}^+$ controls the curvature.
- $q \in \mathbb{R}^+$ controls the distance to the nearest solution.

At $t = 1/2$: $h(x, t = 1/2) = (x^p - q)x = 0$. Consider $z = q^{1/p}$.

- $J_h = (p+1)x^p - q$, $J_h(x = z) = (p+1)q - q = pq$,
  for fixed p, small q, the Jacobian J_h is close to singular.
- $\mathcal{H} = (p+1)px^{p-1}$, $\mathcal{H}(x = z) = (p+1)pq^{(p-1)/p}$,
  for fixed q, the Hessian $\mathcal{H}$ grows quadratically in p.

$$|\Delta z| \geq \frac{J_h(x = z)}{\mathcal{H}(x = z)} = \frac{q}{(p+1)q^{(p-1)/p}}$$

$z = q^{1/p}$: small $q \ll 1$: the lower bound for $|\Delta z|$ decreases for all p,
large $q \gg 1$: the lower bound for $|\Delta z|$ decreases for large p.

Robust Numerical Path Tracking

1. Problem Statement
   - adaptive step control
   - Padé approximants and power series
2. Detecting Nearby Singularities
   - applying the ratio theorem of Fabry
   - an illustrative example
3. Distance to the Nearest Path
   - an estimate based on the Jacobian and Hessian matrices
   - an illustrative example
4. Algorithm and Computational Experiments
   - an apriori step control algorithm
   - two generic polynomials
   - the 184,756 paths to all cyclic 11-roots

an apriori step control algorithm

To determine the step size Δt, do as follows:

1. Compute the distance R to the closest pole.
2. Compute the lower bound $L = \dfrac{2\sigma_n(J_h)}{\sqrt{\sigma_{1,1}^2 + \sigma_{2,1}^2 + \cdots + \sigma_{n,1}^2}}$
   on the distance to the nearest path.
3. Use the approximation error of the Padé approximant:
   $$\left\| x(\Delta t) - \left(\frac{p_1(\Delta t)}{q_1(\Delta t)}, \frac{p_2(\Delta t)}{q_2(\Delta t)}, \ldots, \frac{p_n(\Delta t)}{q_n(\Delta t)} \right) \right\| \approx \|e_0\| \, |\Delta t|^\ell,$$
   compute $\|e_0\| |\Delta t|^\ell \leq L \Rightarrow \Delta t \leq \left(\dfrac{L}{\|e_0\|} \right)^{1/\ell} = D$.
4. $\Delta t = \min(\beta_1 R, \beta_2 D)$, for some constants β_1, β_2, $0 < \beta_1, \beta_2 < 1$.

Robust Numerical Path Tracking

1. Problem Statement
   - adaptive step control
   - Padé approximants and power series
2. Detecting Nearby Singularities
   - applying the ratio theorem of Fabry
   - an illustrative example
3. Distance to the Nearest Path
   - an estimate based on the Jacobian and Hessian matrices
   - an illustrative example
4. Algorithm and Computational Experiments
   - an apriori step control algorithm
   - two generic polynomials
   - the 184,756 paths to all cyclic 11-roots

two generic polynomials

Two polynomials of degree d lead to a homotopy with d^2 paths.

`phc -p` : a posteriori step control with corrector feedback loop
`phc -u` : a priori step control based on poles and Hessians

		phc -p		phc -u	
d	d^2	#fail	user cpu time	#fail	user cpu time
10	100	0	124ms	0	5s 130ms
20	400	2	2s 59ms	0	1m 11s 610ms
30	900	8	10s 462ms	0	6m 10s 521ms
40	1600	23	33s 558ms	0	20m 19s 950ms
50	2500	39	1m 13s 489ms	0	44m 0s 807ms

Ran on one core of 3.1 GHz Intel Core i7, 16 GB 1867 MHz DDR3, in double precision, early 2015 MacBook Pro, macOS Sierra 10.12.6.

Done with version 2.4.67, `phc -u` is still under development.

Robust Numerical Path Tracking

1. Problem Statement
   - adaptive step control
   - Padé approximants and power series
2. Detecting Nearby Singularities
   - applying the ratio theorem of Fabry
   - an illustrative example
3. Distance to the Nearest Path
   - an estimate based on the Jacobian and Hessian matrices
   - an illustrative example
4. Algorithm and Computational Experiments
   - an apriori step control algorithm
   - two generic polynomials
   - the 184,756 paths to all cyclic 11-roots

the 184,756 paths to all cyclic 11-roots

The cyclic 11-roots problem is a sparse polynomial system

- in 11 variables with 181,756 isolated solutions;
- the mixed volume of the Newton polytopes equals 181,756.

The start system is a system with the same Newton polytopes, but with randomly generated complex coefficients.

A run with `phc` on some difficult path shows:

- Around $t = 0.5$, the coordinates take extreme values, suggesting a diverging path.
- But there is no nearby pole at $t = 0.5$ and `phc -u` can complete without the bound involving the Jacobian and Hessians.

These computations are confirmed with the program `Padé.jl`, Julia code written by Simon Telen and Marc Van Barel.

work still in progress...