Certifying Roots of Polynomial Systems on Macaulay2

Kisun Lee
Georgia Institute of Technology

International Symposium on Symbolic & Algebraic Computation
Beijing, China

Certifying (regular) Solutions

Given a compact region $I \subset \mathbb{C}^n$ (or $\mathbb{R}^n$), apply an algorithm to certify

(1) the existence

(2) the uniqueness

of a root of a system in I.

Certifying (regular) Solutions

Given a compact region $I \subset \mathbb{C}^n$ (or $\mathbb{R}^n$), apply an algorithm to certify

(1) the existence

(2) the uniqueness

of a root of a system in I.

Previous Implementations

`alphaCertified` : Hauenstein and Sottile (2012)
implement α-Theory

NumericalCertification.m2

Available at https://github.com/klee669/M2

Software Requirement

Macaulay2 version 1.14

Functionality

- M2 Internal certification package
- α-Theory / Interval arithmetic implementation
- Easier interface

Two Paradigms

Krawczyk method

combines interval arithmetic and Newton's method

Interval arithmetic
- For any arithmetic operator $\odot$,

$[a, b] \odot [c, d] = \{x \odot y \mid x \in [a, b], y \in [c, d]\}$

α-Theory

certify an approximation converges to a solution quadratically

Quadratic convergence
- For $N_F(x) := x - F'(x)^{-1}F(x)$

(Newton operator),

$$\left\| N_F^k(x) - x^* \right\| \leq \left(\frac{1}{2}\right)^{2^k-1} \left\| x - x^* \right\|$$

Two Paradigms – Krawczyk method

F : a square differentiable system on $U \subset \mathbb{C}^n$

I : an interval to certify

$\square F(I) := \{F(x) \mid x \in I\}$: an interval extension of F over an interval I

y : a point in I

Y : an invertible matrix

Define the Krawczyk operator

$$K_y(I) = y - YF(y) + (Id - Y\square F'(I))(I - y)$$

Two Paradigms – Krawczyk method

F : a square differentiable system on $U \subset \mathbb{C}^n$
I : an interval to certify
$\square F(I) := \{F(x) \mid x \in I\}$: an interval extension of F over an interval I
y : a point in I
Y : an invertible matrix
Define the Krawczyk operator

$$K_y(I) = y - YF(y) + (Id - Y\square F'(I))(I - y)$$

Theorem (Krawczyk 1969)**.** The following holds:

(1) If $x \in I$ is a root of F, then $x \in K_y(I)$

(2) If $K_y(I) \subset I$, then there is a root of F in I (existence)

(3) If I has a root and $\sqrt{2}\|Id - Y\square F'(I)\| < 1$, then there is a root of F in I and it is unique where $\|\cdot\|$ is the maximum operator norm (uniqueness)

Two Paradigms - α-Theory

Let $x = (x_1, \ldots, x_n)$ be a point in $\mathbb{C}^n$ and $N_F(x) = x - F'(x)^{-1}F(x)$.

$$\begin{aligned}
\alpha(F,x) &:= \beta(F,x)\gamma(F,x) \\
\beta(F,x) &:= \|x - N_F(x)\| = \|F'(x)^{-1}F(x)\| \\
\gamma(F,x) &:= \sup_{k\geq 2} \left\| \frac{F'(x)^{-1}F^{(k)}(x)}{k!} \right\|^{\frac{1}{k-1}}
\end{aligned}$$

Two Paradigms - α-Theory

Let $x = (x_1, \ldots, x_n)$ be a point in $\mathbb{C}^n$ and $N_F(x) = x - F'(x)^{-1}F(x)$.

$$
\begin{aligned}
\alpha(F,x) &:= \beta(F,x)\gamma(F,x) \\
\beta(F,x) &:= \|x - N_F(x)\| = \|F'(x)^{-1}F(x)\| \\
\gamma(F,x) &:= \sup_{k\geq 2} \left\| \frac{F'(x)^{-1}F^{(k)}(x)}{k!} \right\|^{\frac{1}{k-1}}
\end{aligned}
$$

If $\alpha(F,x) < \frac{13-3\sqrt{17}}{4}$, then x converges quadratically to x^*. Also, $\|x - x^*\| \leq 2\beta(F,x)$.

Two Paradigms - α-Theory

Let $x = (x_1, \ldots, x_n)$ be a point in $\mathbb{C}^n$ and $N_F(x) = x - F'(x)^{-1}F(x)$.

$$\begin{aligned}
\alpha(F,x) &:= \beta(F,x)\gamma(F,x) \\
\beta(F,x) &:= \|x - N_F(x)\| = \|F'(x)^{-1}F(x)\| \\
\gamma(F,x) &:= \sup_{k\geq 2}\left\|\frac{F'(x)^{-1}F^{(k)}(x)}{k!}\right\|^{\frac{1}{k-1}} \leq \frac{\mu(F,x)D^{\frac{3}{2}}}{2\|x\|_1} \text{ (Shub, Smale 1993)}
\end{aligned}$$

If $\alpha(F,x) < \frac{13-3\sqrt{17}}{4}$, then x converges quadratically to x^*. Also, $\|x - x^*\| \leq 2\beta(F,x)$.

Two Paradigms - α-Theory

Let $x =$ … $^{-1}F(x)$.

$\alpha(F,x)$

$\beta(F,x)$

$\gamma(F,x)$ … e 1993)

Tomorrow 10:30

"Effective certification of approximate solutions to systems of equations involving analytic functions"

More analysis and experiments for a root certification

If $\alpha(F,x) < \frac{13-3\sqrt{17}}{4}$, then x converges quadratically to x^*. Also, $\|x - x^*\| \leq 2\beta(F,x)$.

Demonstration - α-Theory

```
i1 : needsPackage "NumericalCertification";
i2 : R = RR[x1,x2,y1,y2];
```

Demonstration - α-Theory

```
i1 : needsPackage "NumericalCertification";
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
```

Demonstration - α-Theory

```
i1 : needsPackage "NumericalCertification";
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{.95, .32, -.30, .95}};
```

Demonstration - α-Theory

```
i1 : needsPackage "NumericalCertification";
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{.95, .32, -.30, .95}};

i5 : computeConstants(f,p1)
o5 = o5 = (.00621269, .0000277104, 224.2)
o5 : Sequence
```

Demonstration - α-Theory

```
i1 : needsPackage "NumericalCertification";
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{.95, .32, -.30, .95}};

i5 : computeConstants(f,p1)
o5 = o5 = (.00621269, .0000277104, 224.2)
o5 : Sequence

i6 : certifySolution(f,p1)
o6 = true
```

Demonstration - α-Theory

```
i1 : needsPackage "NumericalCertification";
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{.95, .32, -.30, .95}};

i5 : computeConstants(f,p1)
o5 = o5 = (.00621269, .0000277104, 224.2)
o5 : Sequence

i6 : certifySolution(f,p1)
o6 = true

i7 : p2 = point{{.9, .3, -.3, 1}}; -- poorly approximated solution
```

Demonstration - α-Theory

```
i1 : needsPackage "NumericalCertification";
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{.95, .32, -.30, .95}};

i5 : computeConstants(f,p1)
o5 = o5 = (.00621269, .0000277104, 224.2)
o5 : Sequence

i6 : certifySolution(f,p1)
o6 = true

i7 : p2 = point{{.9, .3, -.3, 1}}; -- poorly approximated solution

i8 : certifySolution(f,p2) -- not an approximate solution
o8 = false
```

Demonstration - α-Theory

```
i9 : p1 = point{{.95,.32,-.30,.95}};
i10 : p3 = point{{.65,.77,.76,-.64}}; -- two approximate solutions
```

Demonstration - α-Theory

```
i9 : p1 = point{{.95,.32,-.30,.95}};
i10 : p3 = point{{.65,.77,.76,-.64}}; -- two approximate solutions

i11 : certifyDistinctSoln(f,p1,p3)
o11 = true
```

Demonstration - α-Theory

```
i9 : p1 = point{{.95,.32,-.30,.95}};
i10 : p3 = point{{.65,.77,.76,-.64}}; -- two approximate solutions

i11 : certifyDistinctSoln(f,p1,p3)
o11 = true

i12 : p1 = point{{.954379+ii*.001043 , .318431, -.298633, .947949}};
-- a complex approximate root close to an actual root
```

Demonstration - α-Theory

```
i9 : p1 = point{{.95,.32,-.30,.95}};
i10 : p3 = point{{.65,.77,.76,-.64}}; -- two approximate solutions

i11 : certifyDistinctSoln(f,p1,p3)
o11 = true

i12 : p1 = point{{.954379+ii*.001043 , .318431, -.298633, .947949}};
-- a complex approximate root close to an actual root

i13 : certifyRealSoln(f,p1)
o13 = true
```

Demonstration - Krawczyk method

```
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
```

Demonstration - Krawczyk method

```
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : (I1, I2, I3, I4) = (interval(.94,.96), interval(.31,.33),
                         interval(-.31,-.29), interval(.94,.96));
```

Demonstration - Krawczyk method

```
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : (I1, I2, I3, I4) = (interval(.94,.96), interval(.31,.33),
                         interval(-.31,-.29), interval(.94,.96));
i5 : o = intervalOptionList apply({x1 => I1, x2 => I2, y1 => I3, y2 => I4},
                                  i -> intervalOption i);
```

Demonstration - Krawczyk method

```
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : (I1, I2, I3, I4) = (interval(.94,.96), interval(.31,.33),
                         interval(-.31,-.29), interval(.94,.96));
i5 : o = intervalOptionList apply({x1 => I1, x2 => I2, y1 => I3, y2 => I4},
                                  i -> intervalOption i);

i6 : krawczykOper(f,o)
-- warning: experimental computation over inexact field begun
--          results not reliable (one warning given per session)
o6 = {{[.954149, .954609]}, {[.318086, .318777]}, {[-.298824, -.298442]},
      {[.947663, .948236]}}
o6 : IntervalMatrix
```

Demonstration - Krawczyk method

```
i2 : R = RR[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -3.5, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : (I1, I2, I3, I4) = (interval(.94,.96), interval(.31,.33),
                         interval(-.31,-.29), interval(.94,.96));
i5 : o = intervalOptionList apply({x1 => I1, x2 => I2, y1 => I3, y2 => I4},
                                  i -> intervalOption i);

i6 : krawczykOper(f,o)
-- warning: experimental computation over inexact field begun
--          results not reliable (one warning given per session)
o6 = {{[.954149, .954609]}, {[.318086, .318777]}, {[-.298824, -.298442]},
      {[.947663, .948236]}}
o6 : IntervalMatrix

i7 : krawczykMethod(f,o)
given interval contains a unique solution
o7 = true
```

Demonstration – exact computation

```
i2 : R = QQ[x1,x2,y1,y2]; -- computation over Q
```

Demonstration – exact computation

```
i2 : R = QQ[x1,x2,y1,y2]; -- computation over Q
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{95/100, 32/100, -30/100, 95/100}}; -- exact input
```

Demonstration – exact computation

```
i2 : R = QQ[x1,x2,y1,y2]; -- computation over Q
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{95/100, 32/100, -30/100, 95/100}}; -- exact input

i5 : computeConstants(f,p1)
       21324026093882418049     17681521    120600632116900
o5 = (---------------------, -----------, ---------------)
      3432333340166716036800  638081440000    537914617947
```

Demonstration – exact computation

```
i2 : R = QQ[x1,x2,y1,y2]; -- computation over Q
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : p1 = point{{95/100, 32/100, -30/100, 95/100}}; -- exact input

i5 : computeConstants(f,p1)
       2132402609388241804 9     17681521    12060063211690 0
o5 = (---------------------, -----------, ---------------)
      34323333401667160368 00  638081440000    537914617947

i6 : certifySolution(f,p1)
o6 = true
```

Demonstration – exact computation

```
i2 : R = QQ[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : (I1, I2, I3, I4) = (interval(94/100,96/100), interval(31/100,33/100),
                         interval(-31/100,-29/100), interval(94/100,96/100));
i5 : o = intervalOptionList apply({x1 => I1, x2 => I2, y1 => I3, y2 => I4},
                                   i -> intervalOption i);
```

Demonstration – exact computation

```
i2 : R = QQ[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : (I1, I2, I3, I4) = (interval(94/100,96/100), interval(31/100,33/100),
                          interval(-31/100,-29/100), interval(94/100,96/100));
i5 : o = intervalOptionList apply({x1 => I1, x2 => I2, y1 => I3, y2 => I4},
                                    i -> intervalOption i);

i6 : krawczykOper(f,o)
        381087  381271      254087  254639          895127     893983
o6 = {{[------, ------]}, {[------, ------]}, {[- -------, - -------]},
        399400  399400      798800  798800         2995500    2995500
        1892483  1893627
      {[-------, -------]}}
        1997000  1997000
```

Demonstration – exact computation

```
i2 : R = QQ[x1,x2,y1,y2];
i3 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i4 : (I1, I2, I3, I4) = (interval(94/100,96/100), interval(31/100,33/100),
                         interval(-31/100,-29/100), interval(94/100,96/100));
i5 : o = intervalOptionList apply({x1 => I1, x2 => I2, y1 => I3, y2 => I4},
                                    i -> intervalOption i);

i6 : krawczykOper(f,o)
        381087  381271      254087  254639          895127     893983
o6 = {{[------, ------]}, {[------, ------]}, {[- -------, - -------]},
        399400  399400      798800  798800         2995500    2995500
        1892483  1893627
      {[-------, -------]}}
        1997000  1997000

i7 : krawczykMethod(f,o)
o7 = true
```

Demonstration – exact computation

```
i2 : FF = QQ[i]/ideal(i^2+1) -- Gaussian rational (extension of Q with imaginary numbers)
o2 = FF
o2 : QuotientRing
```

Demonstration – exact computation

```
i2 : FF = QQ[i]/ideal(i^2+1) -- Gaussian rational (extension of Q with imaginary numbers)
o2 = FF
o2 : QuotientRing

i3 : R = FF[x1,x2,y1,y2];
i4 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i5 : p1 = point{{95/100, 32/100, -30/100, 95/100}};
```

Demonstration – exact computation

```
i2 : FF = QQ[i]/ideal(i^2+1) -- Gaussian rational (extension of Q with imaginary numbers)
o2 = FF
o2 : QuotientRing

i3 : R = FF[x1,x2,y1,y2];
i4 : f = polySystem {3*y1 + 2*y2 -1, 3*x1 + 2*x2 -7/2, x1^2 + y1^2 -1, x2^2 + y2^2 -1};
i5 : p1 = point{{95/100, 32/100, -30/100, 95/100}};

i6 : certifySolution(f,p1)
Warning: invertibility check for Jacobian is skipped for Gaussian rational inputs
o6 = true
```

Thanks for your attention!