MATERIAŁY XIV KONFERENCJI SZKOLENIOWEJ
Z TEORII ZAGADNIEŃ EKSTREMALNYCH

1993 Łódź str. 39

WYKŁADNIK ŁOJASIEWICZA W NIESKOŃCZONOŚCI - Z KOMPUTERA

A. Miodek (Łódź)

1. Wstęp

Niech $H = (f, g) : \mathbb{C}^2 \to \mathbb{C}^2$ będzie odwzorowaniem wielomianowym. Oznaczmy $N(H) = \{\nu \in \mathbb{R} : \exists A > 0,\ \exists B > 0, \forall |z| > B,\ A|z|^\nu \leq |H(z)|\}$. Przez wykładnik Łojasiewicza w nieskończoności odwzorowania H będziemy rozumieć $\sup N(H)$, gdy $N(H) \neq \emptyset$ i $-\infty$, gdy $N(H) = \emptyset$. Wykładnik ten będziemy oznaczać przez $\mathcal{L}_\infty(H)$.

Jeżeli $h : \mathbb{C}^2 \to \mathbb{C}^2$ jest funkcją wielomianową i oznaczymy $H = (h'_x, h'_y)$, to $\mathcal{L}_\infty(H)$ nazywamy wykładnikiem Łojasiewicza w nieskończoności funkcji h i oznaczamy $\mathcal{L}_\infty(h)$.

Wykładnik Łojasiewicza był przedmiotem wielu badań w ostatnich latach. W przypadku dwuwymiarowym wyniki uzyskane przez Chądzyńskiego i Krasińskiego w pracy [CK], anonsowane w [CK_1], (częściowe wyniki były podane przez Płoskiego [P] oraz przez Ha [H]) dają dokładne wzory na wykładnik Łojasiewicza, w postaci pozwalającej ułożyć program komputerowy obliczający ten wykładnik dla dowolnego odwzorowania wielomianowego $H : \mathbb{C}^2 \to \mathbb{C}^2$ i dowolnej funkcji wielomianowej $h : \mathbb{C}^2 \to \mathbb{C}$.

W niniejszej pracy przedstawię algorytm obliczania wykładnika Łojasiewicza oraz program komputerowy napisany w proceduralnym języku pakietu

Mathematica. Używałem programu Mathematica (MS–DOS 38 6/7) 1.2.

2. Podstawowe twierdzenia

Niech $H : \mathbb{C}^2 \ni (x, y) \mapsto (f(x, y), g(x, y)) \in \mathbb{C}^2$ będzie odwzorowaniem wielomianowym. W dalszej części tego paragrafu będziemy zakładać, że H spełnia warunki:

$$0 < \deg f = \deg_y f, \qquad 0 < \deg g = \deg_y g. \tag{2.1}$$

Zauważmy, że warunki (2.1) nie stanowią istotnego ograniczenia w obliczaniu wykładnika Łojasiewicza. Wynika to z jednej strony z faktu, że dla $f = \text{const.}$ lub $g = \text{const.}$ obliczamy łatwo $\mathcal{L}_\infty(H)$ z definicji, z drugiej zaś strony z faktu, że dla dowolnego automorfizmu liniowego L w $\mathbb{C}^2$ mamy $\mathcal{L}_\infty(H) = \mathcal{L}_\infty(H \circ L)$.

Niech $Q(u, v, x) = \mathrm{Res}_y(f-u,\ g-v)$ będzie rugownikiem $f-u$ i $g-v$ względem y. Z własności rugownika nietrudno wynika, że Q nie znika tożsamościowo. Połóżmy

$$Q(u, v, x) = Q_0(u, v)x^N + \cdots + Q_N(u, v), \qquad Q_0 \neq 0. \tag{2.2}$$

(2.3) Twierdzenie (twierdzenie 3.1 w [CK]). *Jeżeli odwzorowanie wielomianowe $H : \mathbb{C}^2 \to \mathbb{C}^2$ spełnia (2.1), to*

(i) $Q_0 = \text{const.}$ *wtedy i tylko wtedy, gdy* $\mathcal{L}_\infty(H) > 0$,
(ii) $Q_0 \neq \text{const.}$ *i* $Q_0(0) \neq 0$ *wtedy i tylko wtedy, gdy* $\mathcal{L}_\infty(H) = 0$,
(iii) *istnieje r takie, że* $Q_0(0) = \cdots = Q_{r-1}(0) = 0$ *i* $Q_r(0) \neq 0$ *wtedy i tylko wtedy, gdy* $-\infty < \mathcal{L}_\infty(H) < 0$,
(iv) $Q_0(0) = \cdots = Q_N(0) = 0$ *wtedy i tylko wtedy, gdy* $\mathcal{L}_\infty(H) = -\infty$.

(2.4) Twierdzenie (twierdzenie 3.2 w [CK], porównaj też wniosek 2.6 w [P]). *Dla $\mathcal{L}_\infty(H) > 0$ mamy*

$$\mathcal{L}_\infty(H) = \Big[\max_{1\le i\le N} \frac{\deg Q_i}{i}\Big]^{-1}.$$

(2.5) Twierdzenie (twierdzenie 3.3 w [CK], porównaj też twierdzenie 1.45 w [H]). *Dla $-\infty < \mathcal{L}_\infty(H) < 0$, gdy $Q_0(0) = \cdots = Q_{r-1}(0) = 0$ i $Q_r(0) \neq 0$ mamy*

$$\mathcal{L}_\infty(H) = \Big[-\min_{0\le i\le r-1} \frac{\mathrm{ord}_0\, Q_i}{r-i}\Big]^{-1}.$$

3. Wykładnik Łojasiewicza w nieskończoności dla odwzorowania wielomianowego - algorytm

Niech $H : \mathbb{C}^2 \ni (x, y) \mapsto (f(x, y), g(x, y)) \in \mathbb{C}^2$ będzie dowolnym odwzorowaniem wielomianowym, niekoniecznie spełniającym (2.1). Podamy teraz algorytm obliczania $\mathcal{L}_\infty(H)$:

Procedura 1. Gdy $f \equiv 0$ lub $g \equiv 0$, to wprost z definicji dostajemy, że $\mathcal{L}_\infty(H) = -\infty$.

Dalej zakładamy, że $f \not\equiv 0$ i $g \not\equiv 0$.

Procedura 2. Gdy $f = \text{const.}$ lub $g = \text{const.}$, to również z definicji pokazujemy łatwo, że $\mathcal{L}_\infty(H) = 0$.

Niech $f \neq \text{const.}$ i $g \neq \text{const.}$ W kolejnym kroku dobieramy automorfizm liniowy L przestrzeni $\mathbb{C}^2$ tak, by odwzorowanie $H \circ L$ spełniało warunki (2.1).

Procedura 3. Kładziemy $L(x, y) = (x + ky,\, y)$, $k = 0$ oraz w miejsce H podstawiamy $H \circ L$. Sprawdzamy, czy spełnione są warunki (2.1). Jeżeli tak, to przechodzimy do procedury następnej. Jeżeli nie, to w miejsce k podstawiamy $k + 1$ i ponownie sprawdzamy (2.1).

Nietrudno pokazać, że procedura 3 zakończy się po skończonej ilości podstawień.

W dalszych obliczeniach przyjmujemy w miejsce f i g odpowiednio $f \circ L$ i $g \circ L$. Oczywiście, warunek (2.1) jest spełniony.

Procedura 4. Obliczamy $Q(u, v, x) = \mathrm{Res}_y(f - u,\, g - v)$ i tworzymy listę współczynników $Q_i(u, v)$.

Na podstawie twierdzenia (2.4) i (2.3) (i) mamy

Procedura 5. Jeżeli $\deg Q_0 = 0$, tzn. $Q_0 = \text{const.} \neq 0$, to

$$\mathcal{L}_\infty(H) = \Big[\max_{1 \le i \le N} \frac{\deg Q_i}{i}\Big]^{-1}.$$

Na podstawie twierdzenia (2.3) (ii) mamy

Procedura 6. Jeżeli $\deg Q_0 > 0$ i $Q_0(0) \neq 0$, to $\mathcal{L}_\infty(H) = 0$.

Pozostał do rozważenia przypadek, gdy $Q_0 \neq \text{const.}$ i $Q_0(0) = 0$.

Procedura 7. Przyjmujemy $r = 1$ i sprawdzamy, czy spełniony jest warunek

$$Q_r(0) = 0. \tag{3.1}$$

Jeżeli tak, to powiększamy r o 1 i ponownie sprawdzamy, czy zachodzi warunek (3.1). Procedurę tę wykonujemy tak długo, aż warunek (3.1) nie będzie spełniony lub otrzymamy $r + 1 > N$.

Jeżeli dla pewnego r otrzymaliśmy, że warunek (3.1) nie jest spełniony, to na podstawie twierdzenia (2.3) (iii) oraz (2.5) mamy

Procedura 8.

$$\mathcal{L}_\infty(H) = \Big[-\min_{0 \le i \le r-1} \frac{\mathrm{ord}_0\, Q_i}{r - i}\Big]^{-1}$$

.

Jeżeli $r + 1 > N$, tzn. $Q_0(0) = \cdots = Q_N(0) = 0$, wówczas na podstawie twierdzenia (2.3) (iv) mamy

Procedura 9.

$$\mathcal{L}_\infty(H) = -\infty.$$

Algorytm został zakończony po wyczerpaniu wszystkich możliwości.

4. Wykładnik Łojasiewicza w nieskończoności dla wielomianu

Niech $h : \mathbb{C}^2 \ni (x, y) \mapsto h(x, y) \in \mathbb{C}$ będzie dowolnym wielomianem. Zgodnie z definicją $\mathcal{L}_\infty(h)$ wystarczy rozpocząć algorytm od następującej procedury.

Procedura 0. $f = h'_x$, $g = h'_y$. Następnie przechodzimy do algorytmu opisanego w poprzednim punkcie.

5. Kilka przykładów

Przykład 1. Dla $h(x, y) = (x - (\sqrt{5} + 1)\, y^2)(2 + 3i - x)^2$ mamy $\mathcal{L}_\infty(h) = 2$.

Przykład 2. Dla $H = (f, g)$, gdzie $f(x, y) = y - x^3$, $g(x, y) = (xy - y)^3 + y$ mamy $\mathcal{L}_\infty(H) = 3$.

Przykład 3. Dla $h(x, y) = y^2 + (x + y^3)^2$ mamy $\mathcal{L}_\infty(h) = -\frac{1}{3}$.

6. Program

Aby uruchomić program dla obliczania wykładnika Łojasiewicza odwzorowania $H : (x, y) \mapsto (f(x, y), g(x, y))$ należy podać komendę: LExpMapp $[f, g, x, y]$. Natomiast program dla obliczania wykładnika Łojasiewicza dla wielomianu $h : (x, y) \to h(x, y)$ uruchamiamy komendą: LExpPoly $[h, x, y]$.

Poniżej podaję kolejne procedury programu. Zastosowałem w nim łatwe do zdefiniowania makropolecenia:

degpoly$[f, x, y]$ - oblicza stopień wielomianu f zmiennych x, y;

ord 20$[f, x, y]$ - oblicza rząd w zerze wielomianu f zmiennych x, y;

resultant$[f, g, y]$ - oblicza rugownik wielomianów f, g o współczynnikach zespolonych, względem zmiennej y. Makropolecenie to jest niezbędne, gdyż w programie Mathematica komendę Resultant $[f, g, y]$ stosuje się do wielomianów o współczynnikach wymiernych. W przeciwnym wypadku wynikiem działania tej komendy jest 1.

```
LExpPoly/: LExpPoly[h_,x_,y_]: = (f =.; f = D[h,x];
                                   g = .; g = D[h,y];
                                   LExpMapp[f,g,x,y])
LExpMapp/: LExpMapp [f_,g_,x_,y_]: = (wyklloj = .; fl = .; gl = .;
                                       fl = f; gl = g;
                                       If[fl = = = 0 || gl = = = 0, proc11, proc20];
                                       x = .; y = .;
                                       wyklloj)
proc11/: proc11: = (wyklloj = -Infinity)
proc20/: proc20: = (If[(Exponent[fl, x] = = 0 && Exponent[fl,y] = = 0)
                    || (Exponent[gl,x] = = 0 && Exponent[gl,y] = = 0),
                    proc21, proc30])
proc21/: proc21: = (wyklloj = 0)
proc30/: proc30: = Block[{tf,tg}, tf = .; tg = .; tf = fl; tg = gl;
                   df = .; dg = .; k = .; x.; y = .; x1 = .; y1 = .;
                   df = degpoly[tf,x,y];
                   dg = degpoly[tg,x,y];
                   x = x1 + k* y1; y = y1;
```

```
                              k = 0;
                              proc31]
proc31/: proc31: = Block[{mf,mg}, mf = .; mg = .;
                                tf = Expand[f];
                                tg = Expand[g];
                                mf = Exponent[tf,y1];
                                If[df = = mf,tf,proc32];
                                mg = Exponent[tg,y1];
                                If[dg = = mg,tg,proc32];
                                proc40]
proc32/: proc32: = (k = k+1; proc31)
proc40/: proc40: = (q = .; v = .; u = .; nq = .; s4 = .; ar = .; br = .; qc = .;
            q = resultant[tf-u, tg-v, y1];
            q = Expand[q];
            nq = Exponent[q, x1];
            s4 = Table[qc[i], {i,0,nq}];
            Do[qc[i] = Coefficient[q, x1, nq-i],{i,0,nq}];
            ar = Exponent[qc[0],u];
            br = Exponent[qc[0],v];
            If[ar = = 0 && br = = 0, proc50, proc60])
proc50/: proc50: = Block[{t5,s5},t5 = .;s5 = .;a50 = .;
                  Table[a50[i],{i,1,nq}];
                  Do[a50[i] = degpoly[qc[i],u,v],{i,1,nq}];
                  t5 = Table[a50[i]/i, {i,1,nq}];
                  s5 = Max[t5];
                  wyklloj = (1/s5)])
proc60/: proc60: = (u = 0; v = 0;
                      If[qc[0]! = 0, proc21, proc70])
proc70/: proc70: = (r7 = .; r7 = 0; i7 = .;
                         For[i7 = 0, (qc[i7] = = 0 && (i7¡ = nq),i7++, r7++];
                         r7 = r7-1;
                         If[r7 = = nq, proc11, proc80])
proc80/: proc80: = Block[{deg1q,s8,r8,i8},
                          i8 = .; deg1q = .; s8 = .; r8 = .; u = .; v = .; a8 = .;
                          deg1q = degpoly[q,u,v];
                          s8 = Table[a8[i8], {i8,0,r7}];
                          Do[a8[i8] = If[qc[i8] = = = 0, deg1q,
                          (ord20[qc[i8],u,v])/(r7 + 1 - i8)],{i8,0,r7}];
                          r8 = Min[s8];
                          wyklloj = 1/(-r8)]
```

References

[CK]. J. Chądzyński, T. Krasiński, *On the Łojasiewicz exponent at infinity for polynomial mappings of* $\mathbb{C}^2$ *into* $\mathbb{C}^2$ *and components of polynomial automorphisms of* $\mathbb{C}^2$, Ann. Pol. Math. (w druku).

[CK$_1$]. J. Chądzyński, T. Krasiński, *Sur l'exposant de Łojasiewicz à l'infini pour les applications polynômial* $\mathbb{C}^2$ *en* $\mathbb{C}^2$ *les composantes des automorphismes polynômes* $\mathbb{C}^2$, C.R. Acad. Sci. Paris, serie I (w druku).

[H]. H.V. Ha, *Nombres de Łojasiewicz et singularités à l'infini des polynômes de deux variables complexes*, ibid. **311** (1990), 429–432.

[P]. A. Płoski, *On the growth of proper polynomial mappings*, Ann. Pol. Math. **45** (1985), 297–309.

The Łojasiewicz exponent at infinity - by computer

Summary. In the paper an algorithm and a computer program in Mathematica for calculation of the Łojasiewicz exponent at infinity for polynomial mappings of $\mathbb{C}^2$ into $\mathbb{C}^2$ is given.

Bronisławów, 11–15 stycznia, 1993 r.