Cursul 12

Şiruri recurente în planul complex

Vom studia, în continuare, comportarea şirurilor definite prin relaţii de recurenţă de forma

$$\begin{cases} z_{n+1} = f(z_n), \quad n = 0, 1, 2, \ldots, \\ z_0 = a, \end{cases} \tag{1}$$

cu $f : D \to D$ o funcţie continuă pe domeniul $D \subset \mathbb{C}$.

Şirul (z_n) este definit, deci, de o recurenţă de ordinul întâi *autonomă*, în care z_{n+1} depinde numai de valoarea termenului precedent z_n, nu şi de indicile său n. În consecinţă, aceste şiruri sunt *fără memorie*: dacă la un moment dat n_0 şirul (z_n) ajunge într-un punct $\tilde{a} \in D$, atunci de la n_0 încolo termenii şirului dat coincid cu cei ai şirului cu data iniţială $z_0 = \tilde{a}$, păstrând mereu un decalaj de n_0 unităţi între ranguri.

Mai mult, aceste şiruri pot fi exprimate cu ajutorul *iteratelor* funcţiei f, adică cu funcţiile

$$f^{\circ n} = f \circ f \circ \cdots \circ f \text{ (de } n \text{ ori)}.$$

Într-adevăr, avem: $z_1 = f(a)$, $z_2 = f(f(a)) = (f \circ f)(a)$, $z_3 = f(f(f(a))) = (f \circ f \circ f)(a)$ ş.a.m.d. Prin urmare,

$$z_n = f^{\circ n}(a),$$

pentru orice $n \geq 1$, adică (z_n) este şirul valorilor în $z = a$ ale iteratelor funcţiei f.

Continuitatea funcţiei f implică o proprietate importantă a şirurilor (z_n) definite de (1): dacă $z_n \to z^*$ atunci, trecând la limită în relaţia de recurenţă, obţinem că $z^* = f(z^*)$. Altfel spus, limita unui astfel de şir (z_n), dacă există, este un *punct fix* al funcţiei f, adică o soluţie a ecuaţiei $z = f(z)$.

Deoarece existenţa punctelor fixe pentru o funcţie f oarecare este importantă atât pentru funcţii reale cât şi pentru funcţii complexe, şi nu numai, vom prezenta în contiunare acest studiu într-un cadru mai general, cel al spaţiilor metrice. Amintim că $\mathbb{R}$ şi $\mathbb{C}$, cu distanţa uzuală $d(u, v) = |u - v|$, sunt spaţii metrice complete.

§1. Teorema de punct fix a lui Banach

Teorema de punct fix a lui Banach, cunoscută şi sub denumirea de *principiul contracţiilor*, a fost formulată şi demonstrată în anul în 1922 de către Ştefan Banach (1892-1945), fondatorul analizei funcţionale.

Definiţie. Fie (X, d) un spaţiu metric. O funcţie $f : X \to X$ pentru care există $q \geq 0$ astfel încât

$$d(f(x), f(y)) \leq q d(x, y), \quad \forall\, x, y \in X,$$

este numită funcţie *lipschitziană*. În cazul în care $q < 1$ spunem că f este o *contracţie*.

Orice funcţie lipschitziană este continuă, reciproca nu are loc în general.

Teoremă (Banach)**.** *Dacă (X, d) este un spaţiu metric complet iar funcţia $f : X \to X$ este o contracţie, atunci f are un punct fix unic în X. Mai mult, pentru orice punct iniţial $x_0 \in X$, şirul aproximaţiilor succesive*

$$\begin{cases} x_0 \in X \\ x_{n+1} = f(x_n), \quad n \in \mathbb{N}, \end{cases} \tag{2}$$

este convergent la punctul fix $x^* \in X$ al lui f, viteza de convergenţă fiind dată de estimarea

$$d(x_n, x^*) \leq \frac{d(x_0, x_1)}{1-q} q^n, \tag{3}$$

unde $q \in [0, 1)$ este constanta Lipschitz a lui f.

Demonstraţie. Considerăm un punct $x_0 \in X$ fixat arbitrar şi definim şirul (x_n) prin relaţia (2). Vom arăta, pentru început, că (x_n) este un şir Cauchy.

Deoarece f este o contracţie, avem, pentru orice $n \in \mathbb{N}$,

$$d(x_{n+1}, x_{n+2}) = d(f(x_n), f(x_{n+1})) \leq q d(x_n, x_{n+1}),$$

de unde rezultă

$$d(x_n, x_{n+1}) \leq q^n d(x_0, x_1),$$

pentru orice $n \in \mathbb{N}$. De aici obţinem imediat, pentru orice $n, m \in \mathbb{N}$ cu $n \leq m$,

$$d(x_n, x_m) \leq \sum_{i=n}^{m-1} d(x_i, x_{i+1}) \leq d(x_0, x_1) \sum_{i=n}^{m-1} q^i \leq \frac{d(x_0, x_1)}{1-q} q^n. \tag{4}$$

Am folosit majorarea dată de suma seriei geometrice

$$\sum_{i=0}^{m-n-1} q^i \leq \sum_{i=0}^{\infty} q^i = \frac{1}{1-q},$$

care este convergentă deoarece constanta Lipschitz q este în intervalul $[0, 1)$.

Fie $\varepsilon > 0$ fixat arbitrar. Deoarece $\lim_{n\to\infty} \frac{d(x_0,x_1)}{1-q} q^n = 0$ rezultă că există un $n_\varepsilon \in \mathbb{N}$ astfel încât $n \geq n_\varepsilon$ implică $\frac{d(x_0,x_1)}{1-q} q^n < \varepsilon$. Din (4) urmează că, pentru orice $n \geq n_\varepsilon$ şi $m \geq n_\varepsilon$, avem $d(x_n, x_m) < \varepsilon$, şi deci (x_n) este şir Cauchy. Spaţiul metric X fiind complet, rezultă că (x_n) este convergent, adică există $x^* \in X$ astfel încât

$$\lim_{n\to\infty} x_n = x^*.$$

În sfârşit, deoarece f este o contracţie, este continuă şi, trecând la limită în relaţia de recurenţă (2), obţinem egalităţile

$$x^* = \lim_{n\to\infty} x_{n+1} = \lim_{n\to\infty} f(x_n) = f(\lim_{n\to\infty} x_n) = f(x^*),$$

care arată că x^* este punct fix pentru f.

Pentru a demonstra unicitatea punctului fix, fie $x^{**} \in X$, cu $x^* \neq x^{**}$, un alt punct fix al lui f. Atunci $d(x^*, x^{**}) > 0$ şi obţinem imediat că

$$d(x^*, x^{**}) = d(f(x^*), f(x^{**})) \leq qd(x^*, x^{**}) < d(x^*, x^{**}),$$

de unde rezultă o contradicţie.

Estimarea (3) se obţine din (4) prin trecere la limită cu $m \to \infty$.

□

Exemplu. Definim $f : \mathbb{C} \to \mathbb{C}$ prin

$$f(z) = az + i,$$

unde

$$a = \frac{9}{10}(\cos\frac{\pi}{7} + i\sin\frac{\pi}{7}).$$

Deoarece

$$|f(u) - f(v)| = |a||u - v| \leq \frac{9}{10}|u - v|,$$

pentru orice u şi v din $\mathbb{C}$, f este o contracţie. Unicul său punct fix este soluţia ecuaţiei $z = az + i$, adică $z^* = i/(1 - a)$.

Clasa $C^\#$ următoare reprezintă grafic, pentru funcţia f de mai sus, comportarea şirului aproximaţiilor succesive pentru diverse valori ale datei iniţiale:

```
public class Banach : ComplexForm
{
    static Complex i = new Complex(0, 1);
    static Complex a = Complex.setRoTheta(0.9, Math.PI / 7);
    Complex f(Complex z)
    {
        return a * z + i;
    }

    void transformaSiTraseaza(Complex[] fig)
    {
        for (int k = 0; k < fig.Length; k++)
        {
            Complex z = fig[k], zprim = f(z);
            setLine(z, zprim, getColor(100 * k));
            fig[k] = zprim;
        }
    }
    public override void makeImage()
    {
        setXminXmaxYminYmax(-5.1, 2.1, -2.1, 5.1);
        ScreenColor = Color.White;
        initScreen();
        setAxis();
        Complex[] fig = new Complex[] { -1 - i, -0.6 - i,
```

```
            -0.2 - i, 0.2 - i, 0.6 - i, 1 - i };

            int nrEtape = 100;
            for (int k = 0; k < nrEtape; k++)
            {
                transformaSiTraseaza(fig);
            }
            resetScreen();
        }
    }
```

Rezultatul este dat în Figura 1, în care vârfurile celor şase linii poligonale reprezintă punctele succesive ale şirului $z_{n+1} = f(z_n)$, pentru şase valori iniţiale diferite ale lui z_0.

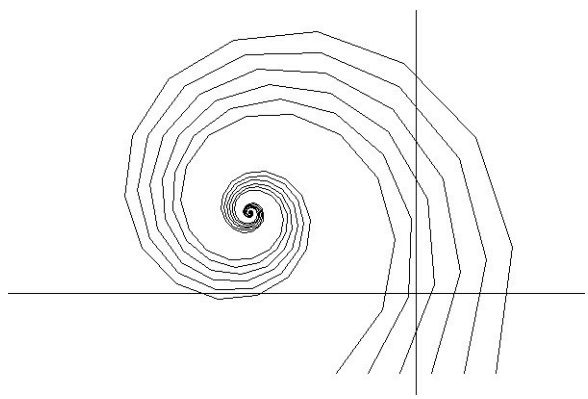

FIGURA 1. Iteraţii trasate de clasa `Banach`.

Găsirea unei contracţii potrivite pentru rezolvarea unei anumite ecuaţii este, în general, o chestiune dificilă, de multe ori următoarea *variantă locală* este salvatoare: dacă ştim că o aplicaţie $f : X \to X$ are un punct fix x^* undeva în spaţiul metric complet (X, d), şi dacă reuşim să arătăm că există $r_0 > 0$ şi $q \in [0, 1)$ astfel încât

$$d(f(x), f(y)) \leq q d(x, y)$$

pentru orice x şi y din $\overline{S}(x^*, r_0) = \{x \,|\, d(x, x^*) \leq r_0\}$, atunci rezultă mediat că f este o contracţie pe $X_0 = \overline{S}(x^*, r_0)$ şi, prin urmare, pentru orice x_0 suficient de aproape de x^* (adică $x_0 \in X_0$) şirul valorilor în x_0 ale iteratelor lui f este convergent la x^*.

În cazurile numerice $X = \mathbb{R}$ sau $X = \mathbb{C}$ avem chiar un rezultat mai precis:

Teoremă. *Fie x^* un punct fix al funcţiei $f : \mathbb{R} \to \mathbb{R}$ (sau $f : \mathbb{C} \to \mathbb{C}$) derivabilă cu derivata continuă (respectiv olomorfă). Dacă*

$$|f'(x^*)| < 1,$$

atunci există $r_0 > 0$ astfel încât, pentru orice x_0 cu $|x_0 - x^*| \leq r_0$ avem

$$\lim_{n \to \infty} f^{\circ n}(x_0) = x^*.$$

Demonstraţie. Fixăm o constantă q astfel încât $|f'(x^*)| < q < 1$. Din continuitatea derivatei în x^*, rezultă că există $r_0 > 0$ astfel încât $|f'(x)| \leq q$ pentru orice $x \in \overline{S}(x^*, r_0)$.

Pentru orice x şi y din $\overline{S}(x^*, r_0)$, calculând integrala derivatei pe segmentul cu capetele x şi y, obţinem

$$|f(x) - f(y)| = |\int_{[x,y]} f'(z)dz| \leq q|x - y|,$$

şi aplicăm în continuare varianta locală a principiului contracţiilor. $\square$

În cazul unei contracţii $f : X \to X$, spunem că punctul să fix, $x^* \in X$, este un *atractor global*, deoarece are proprietatea că pentru orice $x_0 \in X$ şirul iteratelor $(f^{\circ n}(x_0))$ converge la x^*. Se mai spune că x^* este *atractorul* contracţiei f.

In cazul ultimei teoreme, spunem că punctul fix x^* este un *atractor local*, deoarece convergenţa lui $(f^{\circ n}(x_0))$ la x^* este asigurată numai pentru x_0 dintr-o vecinătate a punctului fix. Tot în cazul funcţiilor numerice, un punct fix x^* este numit *repulsor* dacă $|f'(x^*)| > 1$ şi *neutru* dacă $|f'(x^*)| = 1$. In cazul când x^* este repulsor se poate arăta că pentru orice $x \neq x^*$ dar suficient de apropriat de acesta, $d(x, x^*) < d(f(x), x^*)$, şi, prin urmare, un şir de iterate $x_n = f^{\circ n}(x_0)$ poate să conveargă la x^* numai dacă $x_n = x^*$ de la un loc încolo.

§2. Recurenţe definite de metoda lui Newton

Vom prezenta în continuare o problemă formulată în anul 1879 de Arthur Cayley (1821 – 1895) privind convergenţa metodei lui Newton de aproximare a rădăcinilor polinoamelor în mulţimea numerelor complexe. Problema s-a dovedit a fi foarte dificilă, rezolvarea ei completă fiind dată 40 de ani mai târziu de către Gaston Julia (1893 – 1978) şi Pierre Fatou (1878 – 1929).

Incepem prin a prezenta *metoda lui Newon* (numită şi *metoda tangentei*). Fie $p : \mathbb{C} \to \mathbb{C}$ o funcţie polinomială (un polinom) de grad $n \geq 1$. Ne propunem să îi aflăm rădăcinile prin transformarea ecuaţiei

$$p(z) = 0, \tag{5}$$

într-o problemă de punct fix, adică într-o ecuaţie de forma

$$f(z) = z. \tag{6}$$

Fie $\alpha : \mathbb{C} \to \mathbb{C}$ o funcţie care nu ia niciodată valoarea zero. Folosind echivalenţa

$$p(z) = 0 \ \Leftrightarrow z + \alpha(z)p(z) = z,$$

vom considera funcţii f de forma $f(z) = z + \alpha(z)p(z)$. Rădăcinile $z = z^*$ ale lui p coincid deci cu punctele fixe ale lui f, mai trebuie să alegem funcţia α astfel încât aceste puncte fixe să fie atractori, măcar de tip local. Cerem aşadar ca $f'(z^*) = 0$, şi obţinem:

$$1 + \alpha'(z^*)p(z^*) + \alpha(z^*)p'(z^*) = 0 \ \Leftrightarrow \ 1 + \alpha(z^*)p'(z^*) = 0 \ \Leftrightarrow \ \alpha(z^*) = -\frac{1}{p'(z^*)},$$

dacă $p'(z^*) \neq 0$. Ne vom restrânge prin urmare la cazul rădăcinilor simple.

Reciproc, fie acum $z^* \in \mathbb{C}$ o rădăcină simplă a lui p, adică $p(z^*) = 0$ şi $p'(z^*) \neq 0$. Din continuitatea derivatei lui p rezultă că există o vecinătate V_0 a lui z^* pe care p' nu se anulează, şi unde putem defini $f : V_0 \to \mathbb{C}$ prin

$$f(z) = z - \frac{p(z)}{p'(z)}. \tag{7}$$

Deoarece

$$f'(z^*) = \frac{p(z^*)p''(z^*)}{p'^2(z^*)} = 0,$$

rezultă că f este o contracţie pe o vecinătate $\widetilde{V}_0 \subset V_0$ a lui z^*, şi aplicând Teorema de punct fix a lui Banach, obţinem următorul rezultat:

Teoremă (Metoda lui Newton)**.** *Fie $p : \mathbb{C} \to \mathbb{C}$ o funcţie polinomială şi z^* o rădăcină simplă a sa. Atunci există $r > 0$ astfel încât pentru orice $z_0 \in S(z^*, r)$ şirul definit de relaţia*

$$z_{n+1} = z_n - \frac{p(z_n)}{p'(z_n)}, \tag{8}$$

pentru orice $n \in \mathbb{N}$, este convergent la z^*.

Demonstraţie. Şirul definit de (8) este şirul aproximaţiilor succesive ataşat lui f

$$z_{n+1} = f(z_n), \tag{9}$$

adică şirul $(f^{\circ n}(z_0))$ al valorilor în z_0 ale iteratelor funcţiei f asociată lui p prin relaţia (7), care, după cum am argumentat deja, este o contracţie pe $S(z^*, r)$ al cărei atractor este chiar rădăcina z^* a lui p. $\square$

Să aplicăm metoda lui Newton în câteva cazuri simple.

Începem cu $p(z) = z - 1$ care are numai rădăcina $z^* = 1$. Funcţia f asociată are forma $f(z) = 1$ pentru orice $z \in \mathbb{C}$, şi deci şirul (9) este şirul constant $z_n = 1$, pentru orice $n \geq 1$. Rezultatul este corect dar banal.

Considerăm acum ecuaţia

$$z^2 - 1 = 0$$

care are două soluii, $z_1^* = 1$ şi $z_2^* = -1$. Funcţia f dată de (7) poate fi pusă sub forma

$$f(z) = \frac{1}{2}\left(z + \frac{1}{z}\right), \tag{10}$$

şi, prin urmare, şirul aproximaţiilor succesive este dat de relaţia

$$z_{n+1} = \frac{1}{2}\left(z_n + \frac{1}{z_n}\right), \tag{11}$$

pentru orice $n \in \mathbb{N}$.

Se observă imediat că dacă data iniţială este reală atunci tot şirul rămâne în $\mathbb{R}$ şi are următorul comortament: pentru orice $\alpha > 0$, dacă $z_0 = \alpha > 0$, şirul converge la punctul fix $z_1^* = 1$ iar dacă $z_0 = -\alpha > 0$ atunci (z_n) tinde la $z_2^* = -1$.

Această comportare simetrică este mai generală: deoarece $f(-z) = -f(z)$, pentru orice $z \in \mathbb{C} \setminus \{0\}$, rezultă că şirul $(\tilde{z}_n)$ care pleacă din $\tilde{z}_0 = -z_0$ va avea termenii $\tilde{z}_n = -z_n$, pentru orice $n \in \mathbb{N}$. Studiul convergenţei şirului (z_n) în funcţie de data sa iniţială poate fi redus astfel numai la cazul $\operatorname{Re} z_0 \geq 0$, de exemplu. Mai mult, deoarece

$$f(i\beta) = \frac{i}{2}\left(\beta - \frac{1}{\beta}\right),$$

rezultă că dacă data iniţială $z_0 = i\beta$, cu $\beta \in \mathbb{R}^*$, este de pe axa pur imaginară $\operatorname{Re} z = 0$, atunci întregul şir rămâne pe această axă şi deci este divergent (dacă ar fi convergent ar trebui să tindă la unul dintre cele două punctele fixe, z_1^* sau z_2^*).

Deoarece se poate întâmpla ca pentru un $n_0 \in \mathbb{N}$ să avem $z_{n_0} = 0$ şi atunci z_{n_0+1} să nu mai fie bine definit, pentru un studiu unitar se preferă extinderea mulţimii $\mathbb{C}$ la $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ şi prelungirea funcţiei f dată de (10) la $\overline{\mathbb{C}}$, adăugând la definiţie $f(0) = \infty$ şi $f(\infty) = \infty$.

Din varianta locală a principiului contracţiilor avem următorul rezultat parţial: dacă data iniţială z_0 este suficient de aproape de unul dintre punctele fixe, atunci şirul converge către acel punct fix. Studiul complet a fost dat de Cayley, care a stabilit următoarele:

Teoremă (Cayley)**.** *Fie z_n şirul dat de relaţia* (11)*. Dacă $\operatorname{Re} z_0 > 0$ atunci $z_n \to +1$, dacă $\operatorname{Re} z_0 < 0$, $z_n \to -1$ iar dacă $\operatorname{Re} z_0 = 0$ şirul rămâne pe axa imaginară reunită cu punctul de la infinit, având una din comportările următoare: sau $z_n = \infty$ de la un loc încolo, sau este periodic de la un loc încolo, sau "vagabondează" la nesfârşit pe axa imaginară fără să aibă vreo limită.*

Demonstraţie. Scriind relaţia (11) sub forma

$$z_{n+1} = \frac{z_n^2 + 1}{2z_n},$$

obţinem imediat că

$$\frac{z_{n+1} - 1}{z_{n+1} + 1} = \frac{(z_n - 1)^2}{(z_n + 1)^2},$$

de unde, cu substituţia

$$u_n = \frac{z_n - 1}{z_n + 1}, \tag{12}$$

obţinem relaţia de recurenţă

$$u_{n+1} = u_n^2,$$

pentru orice $n \in \mathbb{N}$.

Termenul general al şirului (u_n) este

$$u_n = u_0^{2^n},$$

pentru orice $n \in \mathbb{N}$, şi, inversând substituţia (12), obţinem

$$z_n = \frac{1 + u_n}{1 - u_n} = \frac{1 + u_0^{2^n}}{1 - u_0^{2^n}},$$

pentru orice n, de unde urmează că $z_n \to +1$ dacă $|u_0| < 1$ şi $z_n \to -1$ dacă $|u_0| > 1$. Dar $|u_0| < 1$ înseamnă $|z_0 - 1| < |z_0 + 1|$, echivalentă cu $\operatorname{Re} z_0 > 0$.

□

Să observăm că pentru funcţia extinsă $f : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$, punctul de la infinit este punt fix dar, spre deosebire de z_1^* şi z_2^* care sunt atractori locali, punctul de la infinit este repulsor, deoarece în orice vecinătate a lui există date iniţiale z_0 din care plecând, şirul (z_n) părăseşte definitiv la un moment dat acea vecinătate.

Următoarea clasă $C^{\#}$ colorează cu roşu *bazinul de atracţie* al atractorului z_1^*, format din punctele z_0 din care şirul converge la $z_1^* = +1$ şi colorează cu albastru bazinul lui $z_2^* = -1$. Celelate puncte sunt lăsate negre. Peste desen sunt trasate în final, cu alb, axele de coordonate.

```
public class Newton2 : ComplexForm
{
    Complex zStar1 = 1;
    Complex zStar2 = -1;
    Complex f(Complex z)    // f(z)=0.5*(z+1/z)
    {
        return 0.5 * (z + 1 / z);
    }
    public override void makeImage()
    {
        setXminXmaxYminYmax(-1.5, 1.5, -1.5, 1.5);
        int nrIter = 300;
        foreach (Pixel px in bmpPixels)
        {
            Complex z = getZ(px);
            Color col = Color.Black;
            for (int k = 0; k < nrIter; k++)
            {
                if ((z - zStar1).Ro2 < 0.01){
                    col = Color.Red;
                    break;
                }
                if ((z - zStar2).Ro2 < 0.01){
                    col = Color.Blue;
                    break;
                }
                z = f(z);
            }
            setPixel(px, col);
        }
        setAxis();
        resetScreen();
    }
}
```

Rezultatul este arătat în Figura 2 şi confirmă categoric studiul anterior.

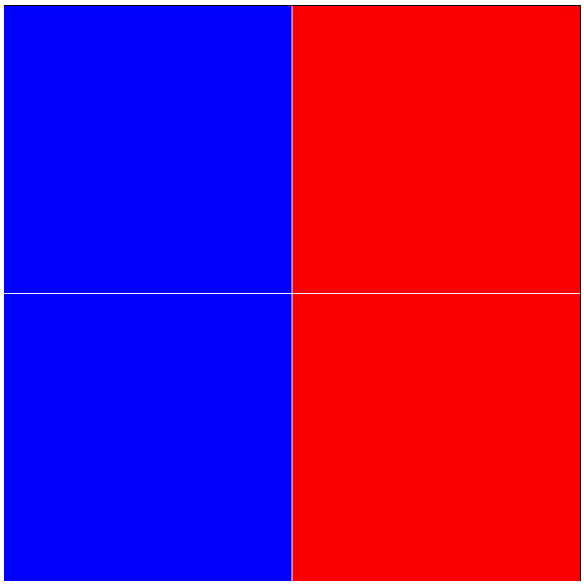

FIGURA 2. `Class Newton2`

Un desen mai diversificat se obţine când ilustrăm *viteza de convergenţă* a şirului (z_n) în raport cu data sa iniţială. În acest scop, clasa `Newton2ETA` listată mai jos colorează punctele z_0 pe baza "timpului de scăpare", utilizând aşa numitul *Escape Time Algorithm*: culoarea lui z_0 este stabilită în funcţie de numărul de iteraţii efectuate până la luarea unei decizii privind comportamentul şirului generat de el (adică până când z_n ajunge într-o vecinătate prestabilită a limitei sale, sau până este atins un număr maxim de iteraţii).

Benzile colorate din Figura 3 sunt formate din puncte care ajung în acelaşi timp la o distanţă faţă de limitele lor mai mică decât `0.01`.

```
public class Newton2ETA : ComplexForm
{
    Complex zStar1 = 1;
    Complex zStar2 = -1;

    Complex f(Complex z)    // f(z)=0.5*(z+1/z)
    {
        return 0.5 * (z + 1 / z);
    }
    public override void makeImage()
    {
        setXminXmaxYminYmax(-10.5, 10.5, -10.5, 10.5);

        int k, nrIter = 300;
        foreach (Pixel px in bmpPixels)
        {
            Complex z = getZ(px);
            for (k = 0; k < nrIter; k++)
```

```
            {
                if ((z - zStar1).Ro2 < 0.01
                  || (z - zStar2).Ro2 < 0.01) break;
                z = f(z);
            }
            setPixel(px, getColor(k * 50));
        }
        setAxis();
        resetScreen();
    }
}
```

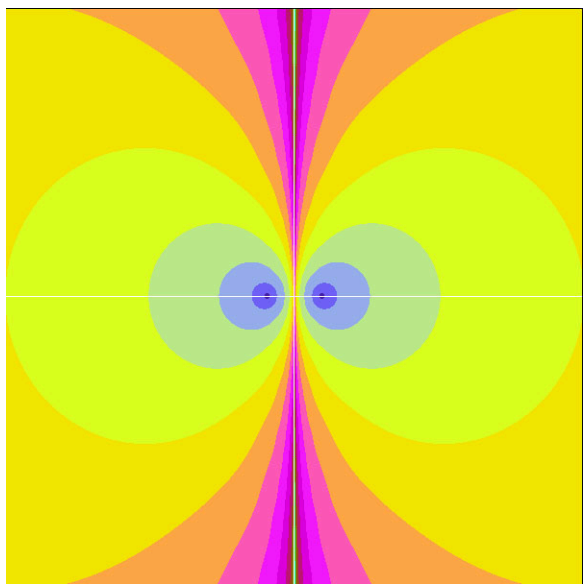

FIGURA 3. `Class Newton2ETA`

§3. Problema lui Cayley

Am ajuns, în sfârşit, la problema formulată de Arthur Cayley: să se stabilească bazinele celor trei atractori locali ai funcţiei f asociate polinomului $p(z) = z^3 - 1$.

După cum este bine ştiut, ecuaţia

$$z^3 - 1 = 0, \tag{13}$$

are ca soluţii rădăcinile de ordin trei ale unităţii:

$$\varepsilon_0 = \omega^0 = \cos 0 + i \sin 0 = 1,$$

$$\varepsilon_1 = \omega^1 = \cos \frac{2\pi}{3} + i \sin \frac{2\pi}{3} = -\frac{1}{2} + i\frac{\sqrt{3}}{2},$$

$$\varepsilon_2 = \omega^2 = \cos \frac{4\pi}{3} + i \sin \frac{4\pi}{3} = -\frac{1}{2} - i\frac{\sqrt{3}}{2},$$

care sunt rădăcini simple, polinomul având descompunerea

$$p(z) = (z - \varepsilon_0)(z - \varepsilon_1)(z - \varepsilon_2).$$

Aceste rădăcini sunt punctele fixe al funcţiei f dată de (7), pe care o scriem sub forma

$$f(z) = \frac{2z^3 + 1}{3z^2},$$

pentru orice $z \in \mathbb{C} \setminus \{0\}$. La fel ca în cazul precedent, extindem funcţia f la $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ adăugând cazurile $f(0) = \infty$ şi $f(\infty) = \infty$.

Să notăm cu B_0, B_1 şi B_2 cele trei bazine de atracţie ale punctelor fixe ε_0, ε_1 şi ε_2, adică

$$B_i = \{z_0 \in \overline{\mathbb{C}} \,|\, f^{\circ n}(z_0) \to \varepsilon_i\},$$

pentru $i = 1, 2, 3$.

Să stabilim mai întâi o proprietate geometrică: din $\omega^3 = 1$ rezultă imediat că $f(\omega z) = \omega f(z)$, pentru orice z, şi prin urmare, dacă şirul $(\tilde{z}_n)$ pleacă din $\tilde{z}_0 = \omega z_0$ atunci $\tilde{z}_1 = f(\omega z_0) = \omega f(z_0) = \omega z_1$, de unde urmează că $\tilde{z}_n = \omega z_n$, pentru orice $n \in \mathbb{N}$. Dacă $z_n \to \varepsilon_0$ atunci $\tilde{z}_n \to \omega\varepsilon_0 = \varepsilon_1$, adică $z_0 \in B_0$ implică $\omega z_0 \in B_1$. Analog se pot obţine toate celelalte implicaţii pentru a stabili că bazinul B_1 este B_0 rotit cu $120°$ în jurul originii, iar B_2 este B_1 rotit cu acelaşi unghi.

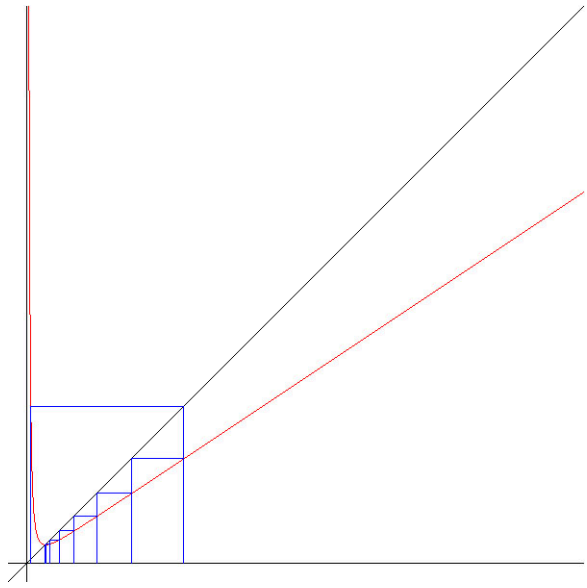

FIGURA 4. Şirul (z_n) pentru $z_0 = 0.2 \in \mathbb{R}_+^*$

Vom arăta acum că mulţimile B_0, B_1 şi B_2 sunt *deschise*. Deoarece ε_0 este un atractor local, ştim că există $r > 0$ astfel încât $z_0 \in S(\varepsilon_0, r)$ implică $f^{\circ n}(z_0) \to \varepsilon_0$. Notăm $S_0 = S(\varepsilon_0, r)$ şi avem, deci, $S_0 \subset B_0$. Definim

$$S_1 = f^{-1}(S_0) = \{z \,|\, f(z) \in S_0\},$$

şi avem imediat că $z_0 \in S_1$ implică $z_1 \in S_0$ şi deci $f^{\circ n}(z_0) \to \varepsilon_0$, adică $S_1 \subset B_0$. Să observăm că S_1 este mulţime dechisă, fiind contraimaginea printr-o funcţie continuă a unei mulţimi deschise.

Definim în mod recurent

$$S_{n+1} = f^{-1}(S_n),$$

pentru orice $n \geq 0$, şi obţinem astfel un şir de mulţimi deschise incluse în B_0. Avem deci

$$\bigcup_n S_n \subset B_0.$$

Pentru a stabili incluziunea inversă, fixăm arbitrar un z_0 în B_0 şi considerăm şirul $z_n = f^{\circ n}(z_0)$. Din $z_n \to \varepsilon_0$, rezultă că există $n_0 \in \mathbb{N}$ astfel încât $z_n \in S_0 = S(\varepsilon_0, r)$, pentru orice $n \geq n_0$. Dar $z_{n_0} = f(z_{n_0-1}) \in S_0$ înseamnă $z_{n_0-1} \in f^{-1}(S_0) = S_1$, şi, din aproape în aproape, obţinem $z_0 \in S_{n_0}$. Am arătat astfel că

$$\bigcup_n S_n = B_0,$$

de unde urmează că mulţimea B_0 este deschisă, ca reuniune de mulţimi deschise.

Să observăm că în raţionamentul precedent am utilizat numai continuitatea lui f, am arătat astfel că pentru orice funcţie continuă, bazinul de atracţie al unui atractor local este o mulţime deschisă.

Să analizăm acum cazul $z_0 = x_0 \in \mathbb{R}_+^*$. Este uşor de văzut că şirul (z_n) rămâne pe semiaxa $\mathbb{R}_+^*$ şi că tinde la 1 pentru orice $x_0 \in \mathbb{R}$. Vezi Figura 4 în care este trasat, cu roşu, graficul restricţiei lui f la $\mathbb{R}_+^*$.

Rezultă ca întreagă semiaxă $\mathbb{R}_+^*$ este inclusă în bazinul B_0 şi, prin urmare, şi celelalte două semiaxele obţinute prin rotaţii de 120° în jurul originii sunt incluse în B_1, respectiv în B_2. De aici deducem că originea se află pe frontiera fiecăruia dintre cele trei bazine.

Pe baza informaţiilor strânse până acum bazinele de atracţie, colorate în albastru, galben şi roşu, ar putea arăta ca în Figura 5

Dar rezolvarea nu este aşa de simplă. Să considerăm, de exemplu, *predecesorii lui zero*, adică acele puncte z_0 pentru care există $n_0 \geq 1$ astfel încât $z_{n_0} = f^{\circ n_0}(z_0) = 0$. Notăm $Z_0 = \{0\}$ şi definim $Z_1 = f^{-1}(Z_0)$, $Z_2 = f^{-1}(Z_1)$, şi aşa mai departe. Mulţimea predecesorilor lui zero este, prin urmare,

$$Z = \bigcup_n Z_n. \tag{14}$$

Să observăm că Z_1 are trei elemente, şi anume soluţiile ecuaţiei $f(z) = 0$, care sunt

$$\zeta_0 = \frac{1}{\sqrt[3]{2}}\left(\cos\frac{\pi}{3} + i\sin\frac{\pi}{3}\right),$$

$$\zeta_1 = \frac{1}{\sqrt[3]{2}}\left(\cos\frac{3\pi}{3} + i\sin\frac{3\pi}{3}\right),$$

$$\zeta_2 = \frac{1}{\sqrt[3]{2}}\left(\cos\frac{5\pi}{3} + i\sin\frac{5\pi}{3}\right).$$

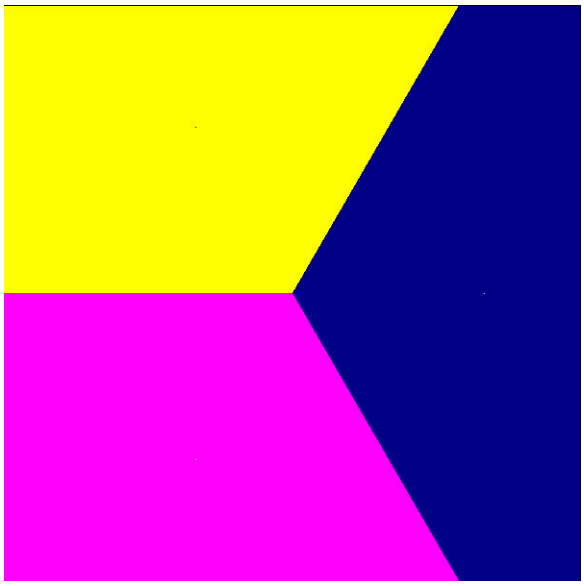

FIGURA 5. O încercare de trasare a bazinelor B_0, B_1 şi B_2.

Z_2 are nouă elemente, soluţiile ecuaţiilor $f(z) = \zeta_0$, $f(z) = \zeta_1$ şi $f(z) = \zeta_2$, şi aşa mai departe.

Ştim că punctul zero se află în acelaşi timp pe frontiera fiecăruia dintre cele trei bazine, vom arăta că fiecare predecesor al său are exact aceeaşi proprietate! Fie, de exemplu, $a_3 \in Z_3$ un predecesor de ordin 3 al lui zero. Notăm $a_2 = f(a_3)$ şi $a_1 = f(a_2)$. Din $a_3 \in Z_3$ rezultă $a_2 \in Z_2$ şi, în continuare, $a_1 \in Z_1$, de unde urmează că

$$(f \circ f \circ f)(a_3) = f(f(f(a_3))) = f(f(a_2)) = f(a_1) = 0$$

şi

$$\begin{aligned}(f \circ f \circ f)'(a_3) &= f'(f(f(a3)))f'(f(a_3))f'(a_3)\\ &= f'(a_1)f'(a_2)f'(a_3) \neq 0,\end{aligned}$$

deoarece derivata f' se anulează numai în punctele fixe ε_0, ε_1 şi ε_2, care evident nu sunt predecesori ai lui zero. Urmează că aplicaţia $f^{\circ 3} = f \circ f \circ f$ este local inversabilă cu inversa g derivabilă (este un difeomorfism local) şi deci păstrează frontierele. Bazinele B_i fiind invariante prin $f^{\circ 3}$, din

$$f^{\circ 3}(a_3) = 0 \in \partial B_0 \cap \partial B_1 \cap \partial B_2$$

obţinem că

$$a_3 = g(0) \in \partial g(B_0) \cap \partial g(B_1) \cap \partial g(B_2) = \partial B_0 \cap \partial B_1 \cap \partial B_2,$$

şi deci în orice vecinătate a lui a_3 se găsesc puncte şi din B_0, şi din B_1, şi din B_2.

Privind Figura 5, înţelegem că frontierele bazinelor nu sunt aşa simple cum le-am trasat noi, nişte semidrepte plecând din origine. Punctul ζ_0, de exemplu, se află pe semiaxa trasată de noi între B_0 şi B_1, şi în el trebuie să se întâlnească toate cele trei culori. De fapt situaţia este mult mai gravă: în orice punct în care

se întâlnesc două culori trebuie să se întâlnescă şi a treia, înţelegând prin aceasta că frontierele celor trei bazine coincid. Studiul complet este dat de următoarea teoremă, pe care o prezentăm fără demonstraţie:

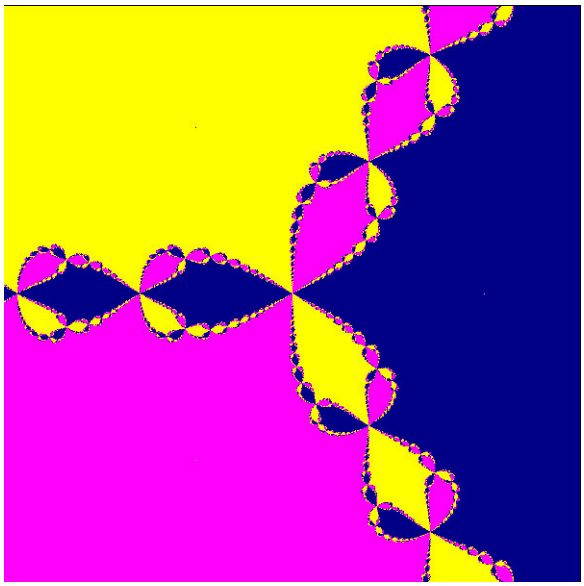

FIGURA 6. `Class Newton3`

Teoremă (Julia). *Fie* $J_f = \overline{\mathbb{C}} \setminus (B_0 \cup B_1 \cup B_2)$. *Mulţimea* J_f *este infinită, nemărginită, închisă, nu are puncte izolate şi nici puncte interioare.* J_f *este închiderea mulţimii predecesorilor lui zero şi, în acelaşi timp, este frontiera fiecăruia dintre cele trei bazine de atracţie, adică*

$$J_f = \partial B_0 = \partial B_1 = \partial B_2.$$

Următoarea clasă $C^{\#}$ colorează cele trei bazine cu albastru, galben şi roşu. Mulţimea J_f, pe care o vom numi *mulţimea Julia* asociată funcţiei f, ar trebui să apară colorată în negru, dar datorită discretizării planului în pixeli, punctele ei sunt cele aflate între doi pixeli vecini de culori diferite (lângă un asemenea punct mai trebuie să mai fie şi un al treilea pixel cu a treia culoare). Vezi Figura 6.

```
public class Newton3 : ComplexForm
{
    Complex eps0 = Complex.setRoTheta(1.0, 0.0 * Math.PI / 3.0);
    Complex eps1 = Complex.setRoTheta(1.0, 2.0 * Math.PI / 3.0);
    Complex eps2 = Complex.setRoTheta(1.0, 4.0 * Math.PI / 3.0);

    Complex f(Complex z)    // f(z)=(2*z*z*z+1)/(3*z*z)
    {
        return (2 * z * z * z + 1) / (3 * z * z);
    }
```

```
    public override void makeImage()
    {
        setXminXmaxYminYmax(-1.5, 1.5, -1.5, 1.5);
        int nrIter = 300;
        foreach (Pixel px in bmpPixels)
        {
            Complex z = getZ(px);
            Color col = Color.Black;
            for (int k = 0; k < nrIter; k++)
            {
                if ((z - eps0).Ro2 < 0.01)
                {
                    col = Color.DarkBlue;
                    break;
                }
                if ((z - eps1).Ro2 < 0.01)
                {
                    col = Color.Yellow;
                    break;
                }
                if ((z - eps2).Ro2 < 0.01)
                {
                    col = Color.Fuchsia;
                    break;
                }
                z = f(z);
            }
            setPixel(px, col);
        }
        setPixel(eps0, Color.White);
        setPixel(eps1, Color.Black);
        setPixel(eps2, Color.White);
        resetScreen();
    }
}
```

O reprezentare mai sugestivă a mulţimii J_f este obţinută în Figura 7, pe baza metodei timpului de scăpare, ETA.

În final, să remarcăm faptul că metoda lui Newton, care a fost stabilită pentru rădăcini simple, funcţionează şi în cazul rădăcinilor multiple. Să considerăm, spre exemplificare, un polinom p cu trei rădăcini distincte a_1, a_2 şi a_3, având multiplicităţile k_1, k_2 şi, respectiv, k_3, adică

$$p(z) = (z - a_1)^{k_1}(z - a_2)^{k_2}(z - a_3)^{k_3}. \tag{15}$$

Atunci

$$p'(z) = k_1(z - a_1)^{k_1-1}(z - a_2)^{k_2}(z - a_3)^{k_3}$$
$$+k_2(z - a_1)^{k_1}(z - a_2)^{k_2-1}(z - a_3)^{k_3} + k_3(z - a_1)^{k_1}(z - a_2)^{k_2}(z - a_3)^{k_3-1},$$

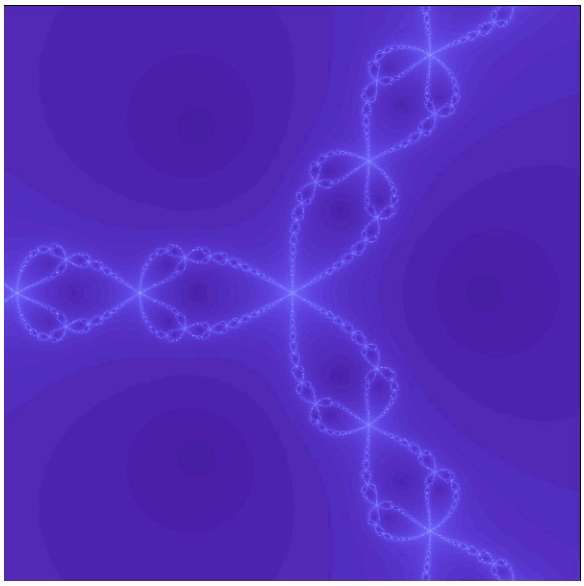

FIGURA 7. `Class Newton3`

deci

$$\frac{p'(z)}{p(z)} = \frac{k_1}{z-a_1} + \frac{k_2}{z-a_2} + \frac{k_3}{z-a_3}.$$

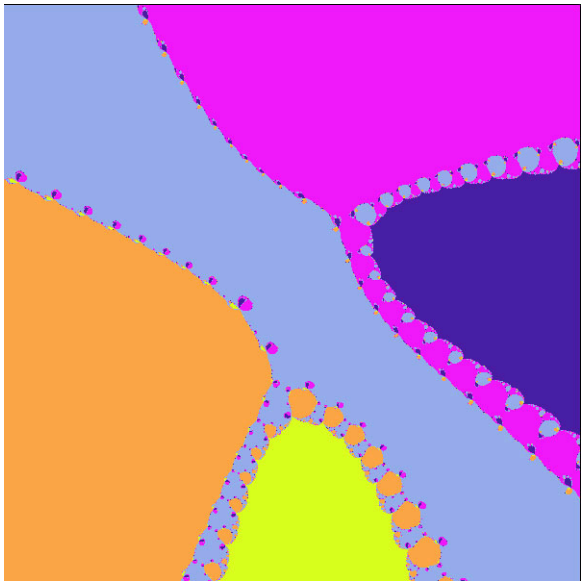

FIGURA 8. `Class Newtonk`

Obţinem pentru funcţia f dată de relaţia (7) următoarea formă

$$f(z) = z - \frac{1}{\frac{k_1}{z-a_1} + \frac{k_2}{z-a_2} + \frac{k_3}{z-a_3}}.$$

Derivăm

$$f'(z) = 1 - \frac{1}{\left(\frac{k_1}{z-a_1} + \frac{k_2}{z-a_2} + \frac{k_3}{z-a_3}\right)^2} \left(\frac{k_1}{(z-a_1)^2} + \frac{k_2}{(z-a_2)^2} + \frac{k_3}{(z-a_3)^2}\right)$$

$$= 1 - \frac{k_1(z-a_2)^2(z-a_3)^2 + k_2(z-a_1)^2(z-a_3)^2 + k_3(z-a_1)^2(z-a_2)^2}{\big(k_1(z-a_2)(z-a_3) + k_2(z-a_1)(z-a_3) + k_3(z-a_1)(z-a_2)\big)^2},$$

şi obţinem:

$$0 \le f'(a_1) = 1 - \frac{1}{k_1} < 1,$$

pentru orice $k_1 \in \mathbb{N}^*$. Punctul fix a_1 este un atractor local, dar cu $f'(a_1) \neq 0$ pentru $k_1 > 1$.

De fapt metoda este valabilă şi pentru funcţii p de forma (15) în care exponenţii k_i, $i = 1, 2, 3$, sunt chiar numere complexe, atât timp cât este îndeplinită condiţia:

$$|f'(a_i)| = |1 - \frac{1}{k_i}| < 1,$$

pentru orice $i \in \{1, 2, 3\}$.

Următorul program ilustrează un astfel de caz.

```
public class Newtonk : ComplexForm
 {
     Complex a1 = new Complex(1.1, 0.6);
     Complex a2 = new Complex(-0.1, 0.6);
     Complex a3 = new Complex(0.1, -1.6);
     Complex a4 = new Complex(-1.1, -0.6);
     Complex a5 = new Complex(0.3, 1.5);

     Complex k1 = Complex.setRoTheta(1.2, 0.1);
     Complex k2 = Complex.setRoTheta(1.3, 0.2);
     Complex k3 = Complex.setRoTheta(1.0, 0.1);
     Complex k4 = Complex.setRoTheta(2.1, 0.05);
     Complex k5 = Complex.setRoTheta(2.3, 0.1);

     Complex f(Complex z)
     {
         Complex w1 = k1 / (z - a1);
         Complex w2 = k2 / (z - a2);
         Complex w3 = k3 / (z - a3);
         Complex w4 = k4 / (z - a4);
         Complex w5 = k5 / (z - a5);
         return z - 1 / (w1 + w2 + w3 + w4 + w5);
     }

     public override void makeImage()
     {
         setXminXmaxYminYmax(-3, 3, -3, 3);
         int nrIter = 300;
```

```
            foreach (Pixel px in bmpPixels)
            {
                Complex z = getZ(px);
                int kol = 0;
                for (int k = 0; k < nrIter; k++)
                {
                    if ((z - a1).Ro2 < 0.01)
                    {
                        kol = 100;
                        break;
                    }
                    if ((z - a2).Ro2 < 0.01)
                    {
                        kol = 200;
                        break;
                    }
                    if ((z - a3).Ro2 < 0.01)
                    {
                        kol = 300;
                        break;
                    }
                    if ((z - a4).Ro2 < 0.01)
                    {
                        kol = 400;
                        break;
                    }
                    if ((z - a5).Ro2 < 0.01)
                    {
                        kol = 500;
                        break;
                    }
                    z = f(z);
                }
                setPixel(px, getColor(kol));
            }
        }
    }
```