

Mulțimea lui Mandelbrot

Mulțimile și funcțiile cu caracter excepțional (mulțimea lui Cantor, insula lui Koch, funcțiile lui Weierstrass și Takagi, curbele lui Peano, mulțimile Julia, ș.a.) au fost studiate intens de matematicieni încă de la apariția lor, în analiza matematică mai ales, în a doua jumătate a secolului al XIX-lea. Cel care a reușit să facă cunoscute aceste obiecte matematice publicului larg a fost Benoît Mandelbrot (1924 – 2010) care, prin două cărți fundamentale: *Les objets fractals, forme, hasard et dimension* (apărută în 1975) și *The Fractal Geometry of Nature* (1982), a introdus denumirea de *fractal* și a argumentat în mod convingător utilitatea lor în diverse domenii. O bună parte din succesul popularizării acestor noțiuni atât de abstracte a fost însă asigurat de reprezentările lor grafice computerizate, care au dezvăluit o adevărată lume ascunsă, plină de forme și culori încântătoare.

Frumusețea vizuală a fractalilor este indiscutabilă, și dintre ei cel mai faimos este, de departe, *mulțimea lui Mandelbrot*, despre care vom vorbi în continuare.

1. Mulțimi Julia pentru funcții pătratice. Plecăm de la studiul mulțimilor Julia J_c asociate funcțiilor pătratice

$$f_c : \overline{\mathbb{C}} \rightarrow \overline{\mathbb{C}}, \quad f_c(z) = z^2 + c,$$

unde c joacă rolul unui *parametru* care parcurge mulțimea numerelor complexe. Deoarece f_c este un polinom, știm că J_c este frontiera bazinului de atracție a punctului de la infinit (care este un atractor) sau, în mod echivalent, frontiera *mulțimii Julia plină*,

$$P_c = \{z_0 \in \mathbb{C} \mid \text{șirul } (f_c^{on}(z_0)) \text{ este mărginit}\}.$$

Pentru implemetarea unui algoritm de trasare a mulțimilor P_c , este util următorul rezultat: dacă există $k \in \mathbb{N}$ astfel încât $|(f^{ok}(z_0))| > 2$ atunci $f^{on}(z_0) \rightarrow \infty$ pentru $n \rightarrow \infty$ și, prin urmare, $z_0 \notin P_c$.

Prin intermediul funcțiilor f_c , asociem fiecărui număr complex $c \in \mathbb{C}$ mulțimea Julia J_c , și suntem interesați să determinăm valorile parametrului c pentru care mulțimea asociată J_c este *conexă*.

Amintim că o mulțime $A \subset \mathbb{C}$ este conexă dacă *nu există* submulțimile deschise D_1 și D_2 cu $A \cap D_1 \neq \emptyset$, $A \cap D_2 \neq \emptyset$, $A \cap D_1 \cap D_2 = \emptyset$ și $A \subset D_1 \cup D_2$.

Definiția 1. Mulțimea lui Mandelbrot, \mathcal{M} , este mulțimea valorilor lui c pentru care J_c este conexă:

$$\mathcal{M} = \{c \in \mathbb{C} \mid J_c \text{ este conexă} \}.$$

De exemplu, pentru $c = 0$ avem $f_0(z) = z^2$ și, prin urmare, $f_0^{cn}(z) = z^{2^n}$. Obținem,

$$P_0 = \{z_0 \in \mathbb{C} \mid \text{șirul } (z_0^{2^n}) \text{ este mărginit}\} = \{z_0 \in \mathbb{C} \mid |z_0| \leq 1\},$$

de unde urmează că J_0 este cercul de rază unu centrat în origine, care este în mod evident mulțime conexă. În concluzie, $0 \in \mathcal{M}$.

Programul următor trasează mulțimea J_c pentru $c = -1.0 + 0.35i$.

```
import ComplexPygame as C
import Color

def Julia():
    def f(z, c):
        return z * z + c

    C.setXminXmaxYminYmax(-2, 2, -2, 2)
    c = complex(-1.0, 0.35)
    nrIter = 1000
    for coloana in C.screenColumns():
        for zeta in coloana:
            z = zeta
            for k in range(nrIter):
                z = f(z, c)
                if C.rho(z) > 4: break
            C.setPixel(zeta, Color.Index(3 * k))
    if C.mustClose():
        return

if __name__ == '__main__':
    C.initPygame()
    C.run(Julia)
```

Examinând rezultatul arătat în Figura 1, observăm că mulțimea P_c nu este conexă și, cu atât mai mult, nici frontiera sa J_c nu este conexă, deci $c = -1.0 + 0.35i \notin \mathcal{M}$.

2. Trasarea mulțimii lui Mandelbrot. Stabilirea conectivității unei mulțimi pe cale algoritmică este extrem de dificilă și din acest motiv Definiția 1 nu este potrivită reprezentării grafice a mulțimii \mathcal{M} (dar este utilă în stabilirea, pe cale riguroasă, a proprietăților ei).

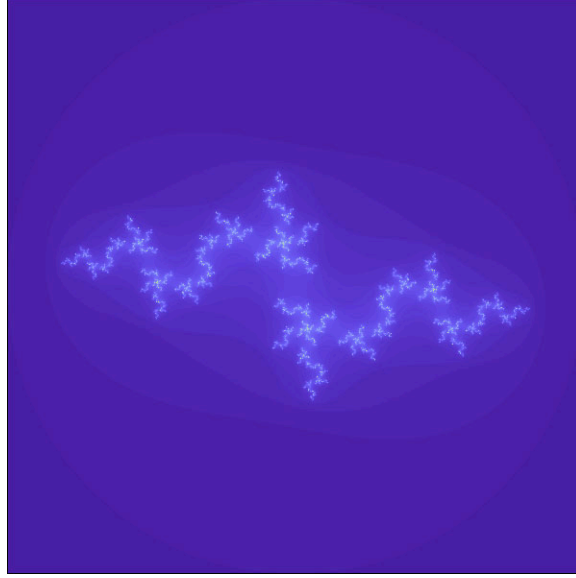


FIGURA 1. Mulțimea J_c pentru $c = -1.0 + 0.35i$.

O metodă foarte simplă de trasare a mulțimii \mathcal{M} se obține pe baza următorului rezultat¹ stabilit de Hans Brolin în 1965: mulțimea Julia asociată unui polinom de grad mai mare decât unu este conexă dacă și numai dacă nici unul dintre punctele sale critice (zerourile derivatei) nu se află în bazinul de atracție al punctului de la infinit.

În cazul nostru $f_c(z) = z^2 + c$ are derivata $f'(z) = 2z$ și deci $z = 0$ este singurul punct critic al lui f_c . Mulțimea J_c este conexă dacă și numai dacă $z = 0$ nu se află în bazinul de atracție al punctului de la infinit. Este ușor de văzut că șirul $(f_c^{on}(0))$ tinde la infinit dacă și numai dacă este nemărginit, și am justificat astfel următoarea caracterizare:

Teorema 1. *Mulțimea lui Mandelbrot este mulțimea valorilor lui c pentru care șirul $(f_c^{on}(0))$ este mărginit:*

$$\mathcal{M} = \{c \in \mathbb{C} \mid \text{șirul } (f_c^{on}(0)) \text{ este mărginit}\}.$$

Pentru a testa dacă parametrul $c \in \mathbb{C}$ este sau nu în mulțimea lui Mandelbrot calculăm termenii șirului $z_0 = 0$, $z_1 = z_0^2 + c$, \dots , $z_N = z_{N-1}^2 + c$, unde N este un număr suficient de mare. Dacă toți z_n sunt în modul mai mici decât doi, considerăm că șirul (z_n) este mărginit și, prin urmare, $c \in \mathcal{M}$. Pixelul corespunzător lui c va fi colorat cu negru (de exemplu) dacă $c \in \mathcal{M}$ și cu alb în

¹H. Brolin, Invariant sets under iteration of rational functions, *Arkiv för Matematik*, **6**(1965) p. 103–144.

caz contrar. Se obține o imagine ca în Figura 2, unde este reprezentată regiunea $[-2.5, 1.5] \times [-2.0, 2.0]$ din planul complex.

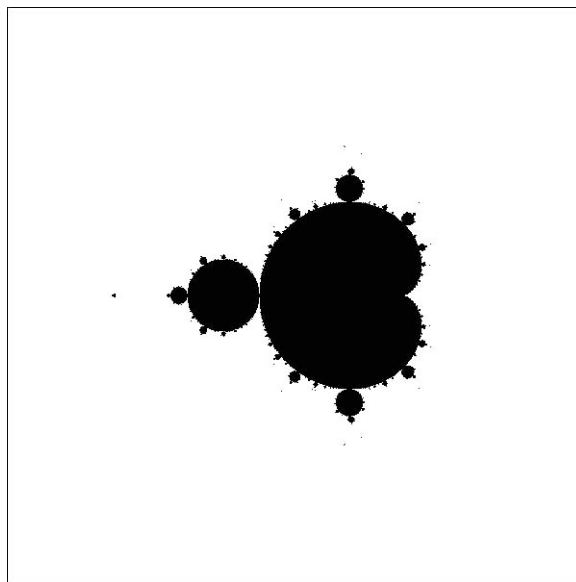
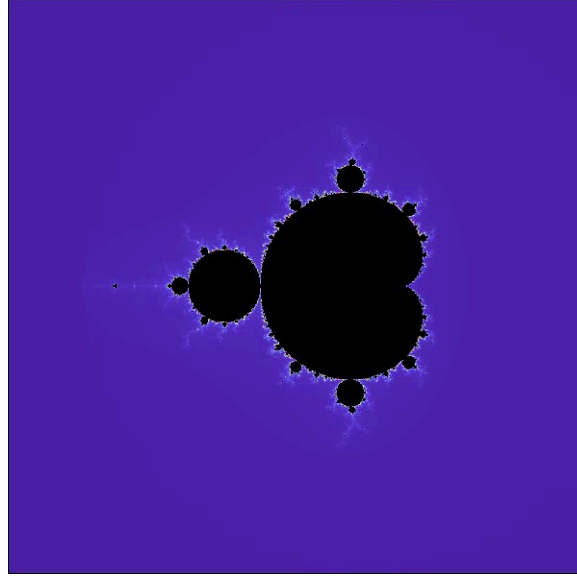


FIGURA 2. Mulțimea lui Mandelbrot

Imagini mult mai interesante se obțin prin metoda de colorare ETA, când fixăm culoarea pixelului corespunzător lui c în funcție de numărul de termeni ai șirului $(f_c^n(0))$ care au modulul mai mic decât doi.

```
def Mandelbrot():
    x0 = 0.27314055
    y0 = -0.486
    r0 = 2.0 / 10 ** 3
    nrIter = 1425
    C.setXminXmaxYminYmax(x0 - r0, x0 + r0, y0 - r0, y0 + r0)
    C.fillScreen()
    for coloana in C.screenColumns():
        for c in coloana:
            z = 0
            col = Color.Black
            for k in range(nrIter):
                z = z * z + c
                if abs(z) > 2:
                    col = Color.Index(700 + k)
                    break
            C.setPixel(c, col)
    if C.mustClose(): return
```

FIGURA 3. `class Mandelbrot`

Mulțimea lui Mandelbrot este o mulțime compactă și conexă, cuprinsă în pătratul $[-2, 2] \times [-2, 2]$, și este formată, în linii mari, dintr-un *corp principal* dat de cardioida

$$C_0 = \{c \in \mathbb{C} \mid c = \frac{1 - (\omega - 1)^2}{4} \text{ cu } |\omega| \leq 1\},$$

un *cap* circular

$$C_1 = \{c \in \mathbb{C} \mid |c + 1| \leq 1/4\}$$

și dintr-un șir infinit de *bulbi*.

Corpul C_0 este format din valorile lui c pentru care funcția f_c are un punct fix atractor. Într-adevăr, cu parametrizarea $c = (1 - (\omega - 1)^2)/4$, soluțiile ecuației $f_c(z) = z$, adică

$$z^2 - z + c = 0. \quad (1)$$

sunt $z_1 = \omega/2$ și $z_2 = 1 - \omega/2$. Impunând ca $f'_c(z) = 2z$ să fie subunitară în modul obținem că aceste două puncte fixe nu pot fi în același timp de tip atractor, iar valorile lui c pentru care cel puțin unul dintre ele este de tip atractor sunt date de apartenența $c \in C_0$.

Punctele $c \in C_1$ sunt caracterizate de existența unei orbite periodice atractive de perioadă doi. Să justificăm: punctele z_3 și z_4 formează o orbită periodică dacă sunt distincte și $f_c(z_3) = z_4$ iar $f_c(z_4) = z_3$. Rezultă imediat că ele sunt soluții pentru ecuația $(f_c \circ f_c)(z) = z$, care are forma

$$z^4 + 2cz^2 - z + c^2 + c = 0. \quad (2)$$

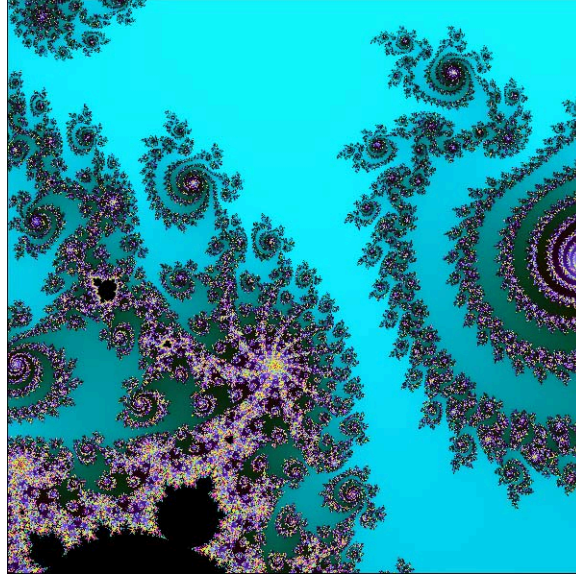


FIGURA 4. Mulțimea lui Mandelbrot. Detaliu;

Ecuația polinomială (2) are exact patru rădăcini în \mathbb{C} , și anume punctele z_3 și z_4 ale orbitei căutate și cele două puncte fixe z_1 și z_2 ale lui f_c determinate mai înainte. Deci rădăcinile ecuației (1) sunt și rădăcini pentru (2), rezultă de aici o relație de divizibilitate între polinoame care se verifică printr-o simplă împărțire:

$$z^4 + 2cz^2 - z + c^2 + c = (z^2 - z + c)(z^2 + z + c + 1).$$

Obținem că z_3 și z_4 sunt rădăcinile ecuației

$$z^2 + z + c + 1 = 0. \quad (3)$$

Să calculăm acum valoarea derivatei $(f_c \circ f_c)'$ în punctele z_3 și z_4 . Avem

$$(f_c \circ f_c)'(z_3) = f'_c(f_c(z_3))f'_c(z_3) = f'_c(z_4)f'_c(z_3) = 4z_3z_4.$$

Din ecuația (3) rezultă că $z_3z_4 = c + 1$. Deci

$$(f_c \circ f_c)'(z_3) = (f_c \circ f_c)'(z_4) = 4(c + 1),$$

și, prin urmare, orbita $\{z_3, z_4\}$ este atrătoare dacă și numai dacă $|c + 1| < 1/4$, adică $c \in C_1$.

Caracterizări asemănătoare relative la existența orbitelor periodice au loc și pentru ceilalți bulbi.

Mărind la scară regiunile aflate în preajma frontierei, se observă că aceasta este compusă, pe lângă componentele deja enumerate, și din numeroase *filamente* în care se află prinse copii mai mici ale mulțimii \mathcal{M} .

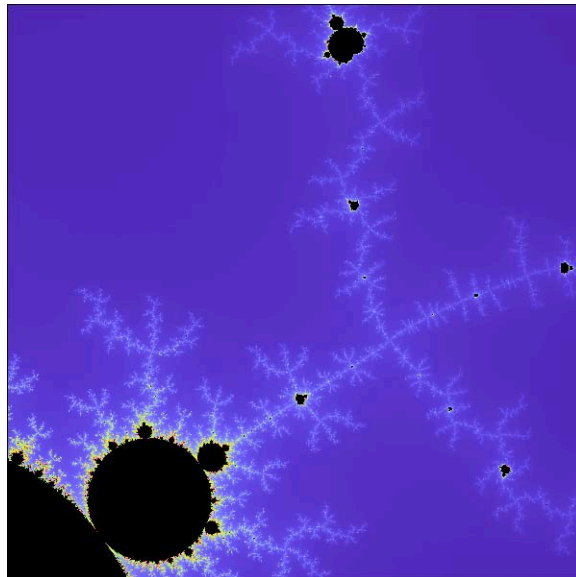


FIGURA 5. Filamente

Imagini foarte frumoase se obțin în regiuni care conțin puncte de frontieră, vezi de exemplu Figura 4 în care este redată vecinătatea punctului $c_0 = 0.27314 + 0.486i$.

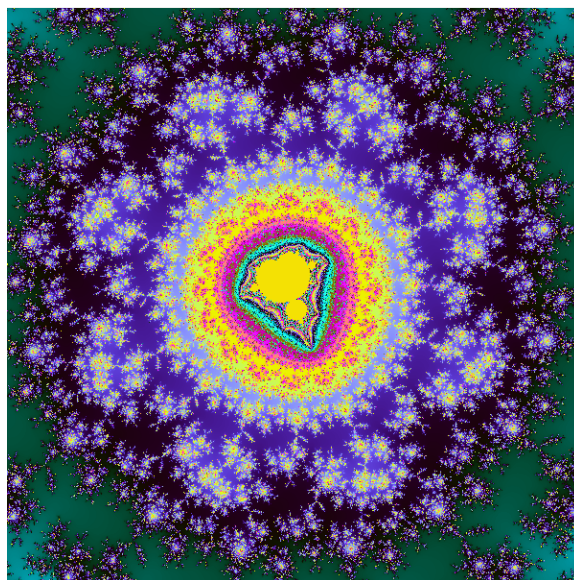


FIGURA 6. Un mandelbrot mic!

3. Cazul real. In cele ce urmează vom analiza intersecția mulțimii lui Mandelbrot cu axa reală, adică mulțimea valorilor lui $c \in \mathbb{R}$ pentru care șirul

$$\begin{cases} x_{n+1} = x_n^2 + c, & \text{pentru } n \geq 1, \\ x_0 = 0, \end{cases} \quad (4)$$

este mărginit.

Vom justifica egalitatea $\mathcal{M} \cap \mathbb{R} = [-2.0, 0.25]$, utilizând pentru ilustrarea comportamentului șirului (x_n) următoarea funcție de desenare:

```
def MandelbrotReal():

    def f(x, c):
        return x * x + c

    a = 2
    C.setXminXmaxYminYmax(-a, a, -a, a)
    C.fillScreen()
    C.setAxis()
    C.drawLineXY(-a, -a, a, a, Color.Black)

    c = -1.812
    # GRAFICUL y=f(x,c)
    xv = -a
    yv = f(xv, c)
    for k in range(C.dim):
        x, _ = C.getXY(k, 0)
        y = f(x, c)
        C.drawLineXY(xv, yv, x, y, Color.Black)
        xv = x
        yv = y

    # SIRUL x_n
    xv = 0
    xp = f(xv, c)
    for k in range(10 ** 5):
        col = Color.Index(k)
        C.drawLineXY(xv, xp, xp, xp, col)
        xv = xp
        xp = f(xv, c)
        C.drawLineXY(xv, xv, xv, xp, col)
        if C.mustClose() or xp < -2 or xp > 2:
            break
```


1. Cazul $c > 0.25$. Parabola $y = x^2 + c$ nu intersectează prima bisectoare, șirul (x_n) este monoton crescător și nemărgit, $c \notin \mathcal{M}$.

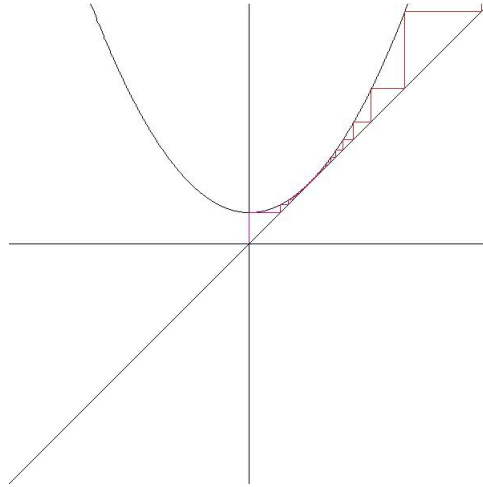


FIGURA 7. $c=0.26$

2. Cazul $c = 0.25$ Ecuația $x = f_c(x)$ are radacina dublă $x^* = 0.5$, șirul (x_n) tine la x^* , deci este mărginit.

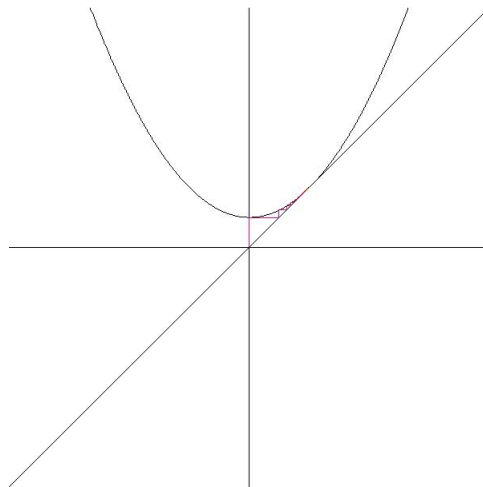


FIGURA 8. $c=0.25$

3. Cazul $-0.75 < c < 0.25$. Funcția f_c are două puncte fixe reale, $x_1^* = (1 + \sqrt{1 - 4c})/2$ și $x_2^* = (1 - \sqrt{1 - 4c})/2$, dintre care x_1^* este repulsor iar x_2^* este atractor. Sirul (x_n) converge la x_2^* în mod oscilant.

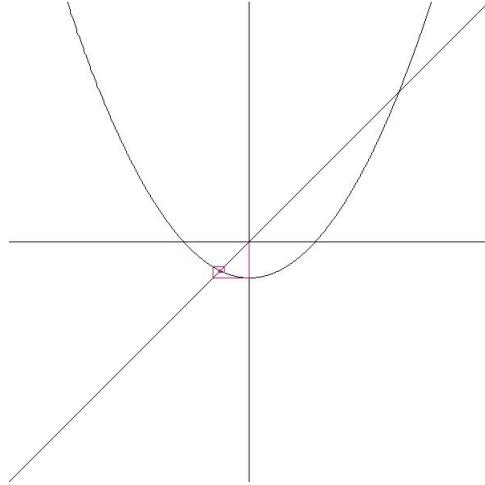


FIGURA 9. $c=-0.30$

4. Cazul $c = -0.75$ Atractorul x_2^* devine punct fix neutru, $x_n \rightarrow x_2^*$.

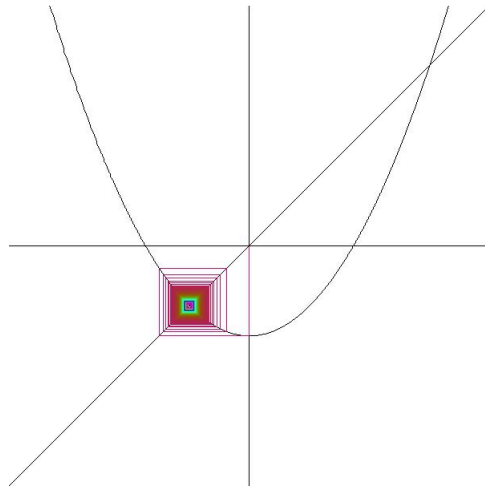


FIGURA 10. $c=-0.75$

5. Cazul $-1.25 < c < -0.75$ Punctul fix x_2^* este repulsor, apare o orbită periodică de perioadă doi de tip atractor.

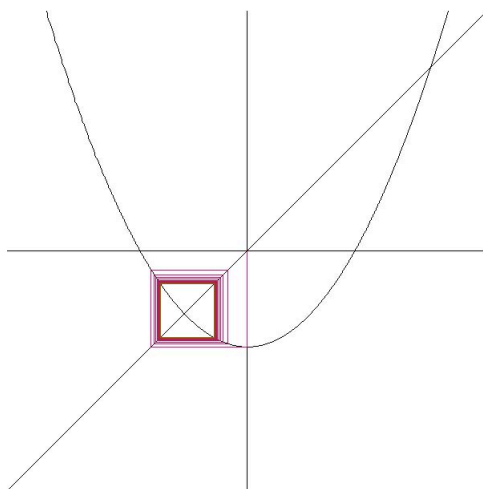


FIGURA 11. $c=-0.80$

6. Cazul $c = -1.25$. Orbita de ordin doi devine neutră, pentru $c < -1.25$ apare o orbită atrătoare periodică cu perioada patru.

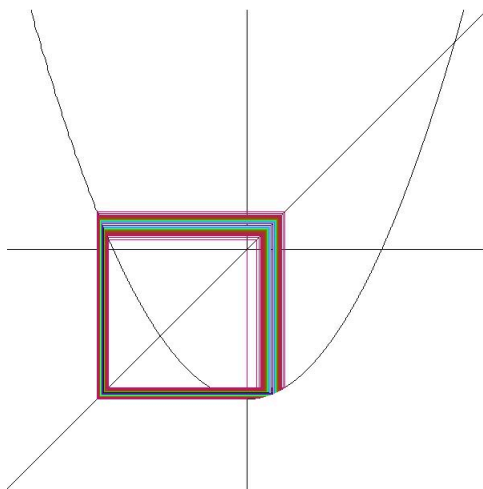


FIGURA 12. $c=-1.25$

7. Cazul $-2 < c < -1.25$. Sirul (x_n) rămâne în intervalul $[-2.0, 2.0]$ având în general un comportament oscilant neregulat, haotic. Apar un șir de *ferestre de stabilitate*, formate din valorile lui c pentru care x_n tinde către un ciclu limită periodic.

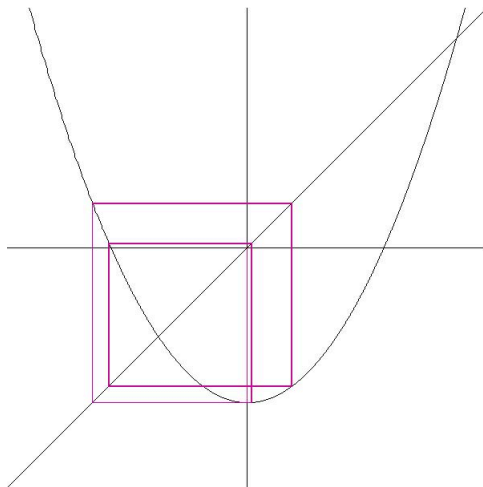


FIGURA 13. $c=-1.29$

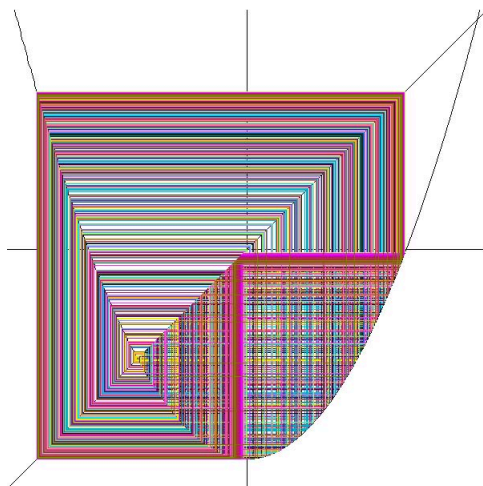
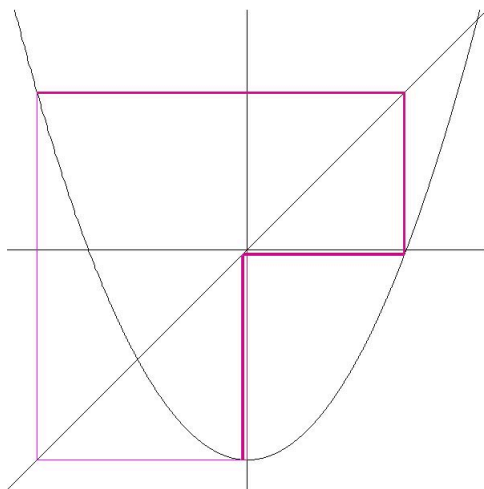
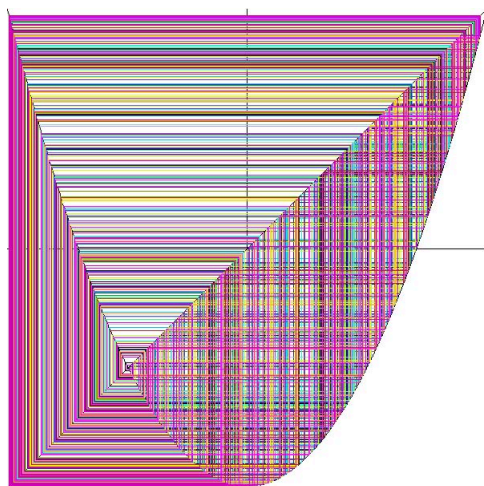


FIGURA 14. $c= -1.7499$

FIGURA 15. $c=-1.7501$ FIGURA 16. $c= -1.98$

8. Cazul $c = -2$. Avem $z_0 = 0$, $z_1 = -2$, $z_2 = 2$, $z_3 = 2$, \dots . Șirul (x_n) este constant egal cu 2 pentru $n \geq 2$, deci $c = -2 \in \mathcal{M}$.

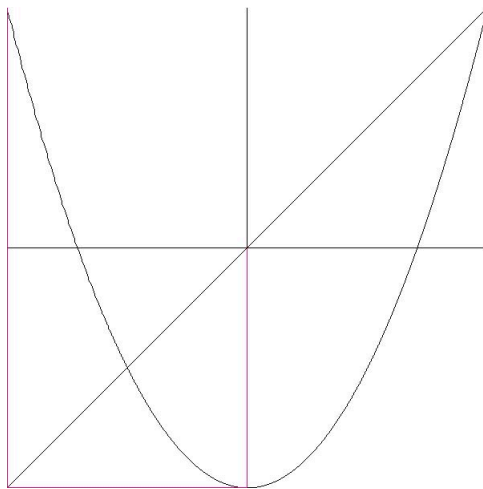


FIGURA 17. $c=-2.0$

9. Cazul $c < -2$. Avem $x_0 = 0$, $x_1 = c < -2$, $x_2 > 2$, \dots , $x_n \rightarrow +\infty$, deci $c \notin \mathcal{M}$.

4. Diagrama Feigenbaum. Analiza comportării șirului $x_n = f_c^{on}(0)$ în funcție de parametrul real c efectuată în paragraful precedent poate fi sintetizată în așa numita *diagramă Feigenbaum*, o reprezentare grafică în care pe verticală sunt puse valorile lui c iar pe orizontală punctele limită ale șirului (x_n) corespunzător. Metoda practică este următoarea: în dreptul fiecărui $c \in [-2.0, 2.0]$ reprezentăm (pe orizontală) toți termenii x_n din domeniul $n \in \{10000, 10001, \dots, 11000\}$, de exemplu. Valorile lor aproximează suficient de bine punctele limită ale șirului.

```
def Feigenbaum():
    def f(x, c):
        return x * x + c

    a = 2
    nrGradatii = 16
    delta = 2 * a / nrGradatii
    C.setXminXmaxYminYmax(-a, a, -a, a / 4)
    C.setAxis()
    for k in range(C.dim, 0, -1):
        _, c = C.getXY(0, k)
        x = 0
        for n in range(11650):
```

```

x = f(x, c)
if abs(x) > 2:
    break
if n < 10000:
    continue
h, _ = C.getHK(x)
C.setPixelHK(h, k, Color.Index(n))
if C.mustClose():
    break

for n in range(nrGradatii):
    c = -a + n * delta
    C.drawLineXY(a - 0.5, c, a, c, Color.Red)
    C.setText("c = {0:.2f}".format(c), complex(a - 0.25, c))

C.refreshScreen()

```

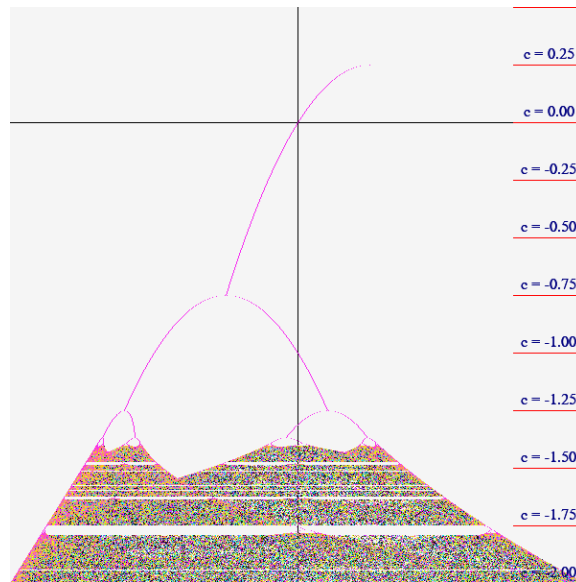


FIGURA 18. Diagrama Feigenbaum