

Algoritm

```

julia(cx, cy, a, N, nrR, nrI, f, c) :=
    (
        (
            maxR
            minR
            maxI
            minI
        )
        ←
        (
            cx + a
            cx - a
            cy + a
            cy - a
        )

        (
            pasR
            pasI
        )
        ←
        (
            round( (maxR - minR) / nrR, 6 )
            round( (maxI - minI) / nrI, 6 )
        )

        for i ∈ 0..nrR - 1
            for j ∈ 0..nrI - 1
                C ← c
                z ← minR + i·pasR + i·(minI + j·pasI)
                n ← 0
                while n < N ∧ |z| < 2
                    n ← n + 1
                    z ← f(z, C)
                    if n = N
                        Ri,j ← 255
                        Gi,j ← 255
                        Bi,j ← 255
                    otherwise
                        Ri,j ← 255 - (2n/N)·255
                        Gi,j ← 255 - (3n/N)·255
                        Bi,j ← log(255, n + 1)
                (
                    z
                    R
                    G
                    B
                )
    )

```

Explicatii

- minR, maxR, minI, maxI sunt parametri pentru domeniul lui z
- pasI si pasR sunt pasii determinati pentru axa reala si axa imaginara
- z este initializat si reprezinta toate punctele din planul initial inainte de iterare
- z este iterat punct cu punct cu ajutorul functiei f pana la o valoare N
- C este constanta functiei recursive
- prin for i si for j se parcurg punct cu punct planul complex iar prin while se itereaza fiecare punct din planul complex pana cand iese afara din domeniul sau ajunge la ultima iterata stabilita
- R,G,B sunt vectori pentru determinarea si retinerea culorilor