

Mathematische Beschreibung des Apfelmännchens nach Mandelbrodt

Maximale Iterationsanzahl $b \in \mathbb{N} \setminus 0$

Anzahl der Bildpunkte der x -Achse $m \in \mathbb{N} \setminus (0,1)$

Minimal des Realanteils der Zahlenebene $rmin \in \mathbb{Q}$ (gewöhnlich -1)

Maximal des Realanteils der Zahlenebene $rmax \in \mathbb{Q}$ (gewöhnlich 2)

Anzahl der Bildpunkte der y -Achse $n \in \mathbb{N} \setminus (0,1)$

Minimal des Imaginäranteils der Zahlenebene $imin \in \mathbb{Q}$ (gewöhnlich -1)

Maximal des Imaginäranteils der Zahlenebene $imax \in \mathbb{Q}$ (gewöhnlich 1)

$X := \{\text{Bildpunkte der } x\text{-Achse}\} \subset \mathbb{Q}$

$x_0 \in X \wedge x_0 = rmin$

$x_m \in X \wedge x_m = f_x(x_{m-1}) : \rightarrow x_m = x_{m-1} + (rmax - rmin)(m-1)^{-1}$

$Y := \{\text{Bildpunkte der } y\text{-Achse}\} \subset \mathbb{Q}$

$y_0 \in Y \wedge y_0 = imin$

$y_n \in Y \wedge y_n = f_y(y_{n-1}) : \rightarrow y_n = y_{n-1} + (imax - imin)(n-1)^{-1}$

$C \subset \mathbb{C} \wedge C := X \times Y$

$c_{mn} \in C \rightarrow c_{mn} := (x_m | y_n)$

$(i_0, r_0) \in \mathbb{R} \wedge i_0, r_0 := 0$

$v_0 \in \mathbb{C} \wedge v_0 := (r_0 | i_0)$

$v_z \in \mathbb{C} \wedge v_z := f_v(v_{z-1}) \rightarrow v_z = (v_{z-1})^2 - c_{mn}$

(berechnungshinweis)

$v_z = (r_z | i_z) = (r_{z-1} | i_{z-1})^2 - c_{mn}$

$r_z = (r_{z-1})^2 - (i_{z-1})^2 - x_m$

$i_z = r_{z-1} i_{z-1} + r_{z-1} i_{z-1} - y_n$

$k_z \in \mathbb{Q} \wedge k_z := (|v_z|)^2 = (r_z)^2 + (i_z)^2$

$a_{mn} \in \mathbb{N}^0 \wedge a_{mn} := f_k(k_z) \rightarrow \begin{cases} z : k_z \geq 4 \\ 0 : k_z < 4 \wedge z < b \end{cases}$

a_{mn} ist die Anzahl der Iterationen bis zur Abbruchbedingung,

nach welcher die Helligkeit oder Farbe des Bildpunktes festgelegt wird

Die Funktion in der Programmiersprache c

```
int amn(double xm, double yn, int b){
    double r = 0;
    double i = 0
    double array[] = {0, 0, xm, yn};
    int count = 0;
    while(count < b){
        count++;
        r = array[0]*array[0] - array[1]*array[1] - array[2];
        i = array[0]*array[1] + array[0]*array[1] - array[3];
        if ((r*r + i*i) >= 4){
            return count;
        }
        array[0] = r;
        array[1] = i;
    }
    return 0;
}
```