8ECM 2020 PORTOROŽ
8TH EUROPEAN CONGRESS OF MATHEMATICS
20 - 26 June 2021, Portorož, Slovenia

Software generation of images using mathematics

Bogdan Soban, BSc
retired
Freelance Generative Artist.
e-mail: bogdan@soban-art.com, www.soban-art.com

Algorithmic Art as a part of Digital Art

- The use of computer in the area if fine art
- Computer aided art (computer plays the role of tool)
- Computer generated art (computer plays the role of creative partner)
  - Object approach
  - Algorithmic approach
- We talk about algorithmic approach
- The essence of the approach - the author writes his own computer program
- The author must first be a programmer and only then an artist
- As a rule, the author cannot predict the outcome
- Use of different concepts and approaches (multi-level algorithmic, fractal, deformation, symmetric, spiral, wandering point, object, manipulation of previously created images and others)
- Everything happens on the object "picture"
- Drawing is performed in different coordinate systems - Cartesian, polar, complex plane - all in connection with the coordinate system of the programming language

Description of my creative approach

- I use my software developed in the Visual Basic programming language to generate images. An algorithmic approach is used, which enables a high level of diversity of results with relatively short but mathematically supported software solutions.
- At each point of the image, a programming algorithm is executed, consisting of mathematical formulas that enter pieces of information, which are constant for the current cycle (genetic code) and variables generated by the process itself. The result of the calculation is the color of the point or the position where to remove the color from the color palette, which plays the role of a key variable.
- The image coordinate system, the Cartesian and polar coordinate system, and the logic of the complex plane are used to control the image.
- The whole process is supported by a time-determined random number generator, which supplies the system with random values of various parameters and thus ensures unpredictability and non-repetition of the results.
- From the viewpoint of mathematics, the following notions appear in the algorithms: complex algebraic expressions, trigonometric and logarithmic functions, plane geometry, iterations and recursions, determination of random numbers in areas, conversion of coordinates between different coordinate systems, control of stochastic motion of a point in plane and space, circular and spiral movements, computation with complex numbers, original formulas for various calculations, number sequences (Fibonacci) and other mathematical operations.
- The resulting paintings are of the abstract type, which for the most part do not show their mathematical provenance and approach fine art.

Infrastructure and basic starting points

- Semi-object graphics programming language
- Everythings happens on the objec “picture”
- Coordinate systems
- Random numbers
- Gene code – program cycle constants
- Variables generated by process
- Color palettes palete – key variable
- Convert a numeric value to a color
- Draw by points (by rows and columns or randomly)
- A color calculation algorithm is performed for each point of the picture
- Some examples of mathematics formulas in algorithm
- Cycle constants and process variables enter in the calculation.
- The value ov calcualtion is converted to a color
- Coloring algorithm is used or removing color from palette
- The algorithm draws a point in selected color
- The number of different images goes towards infinity

Generating images in different coordinate systems

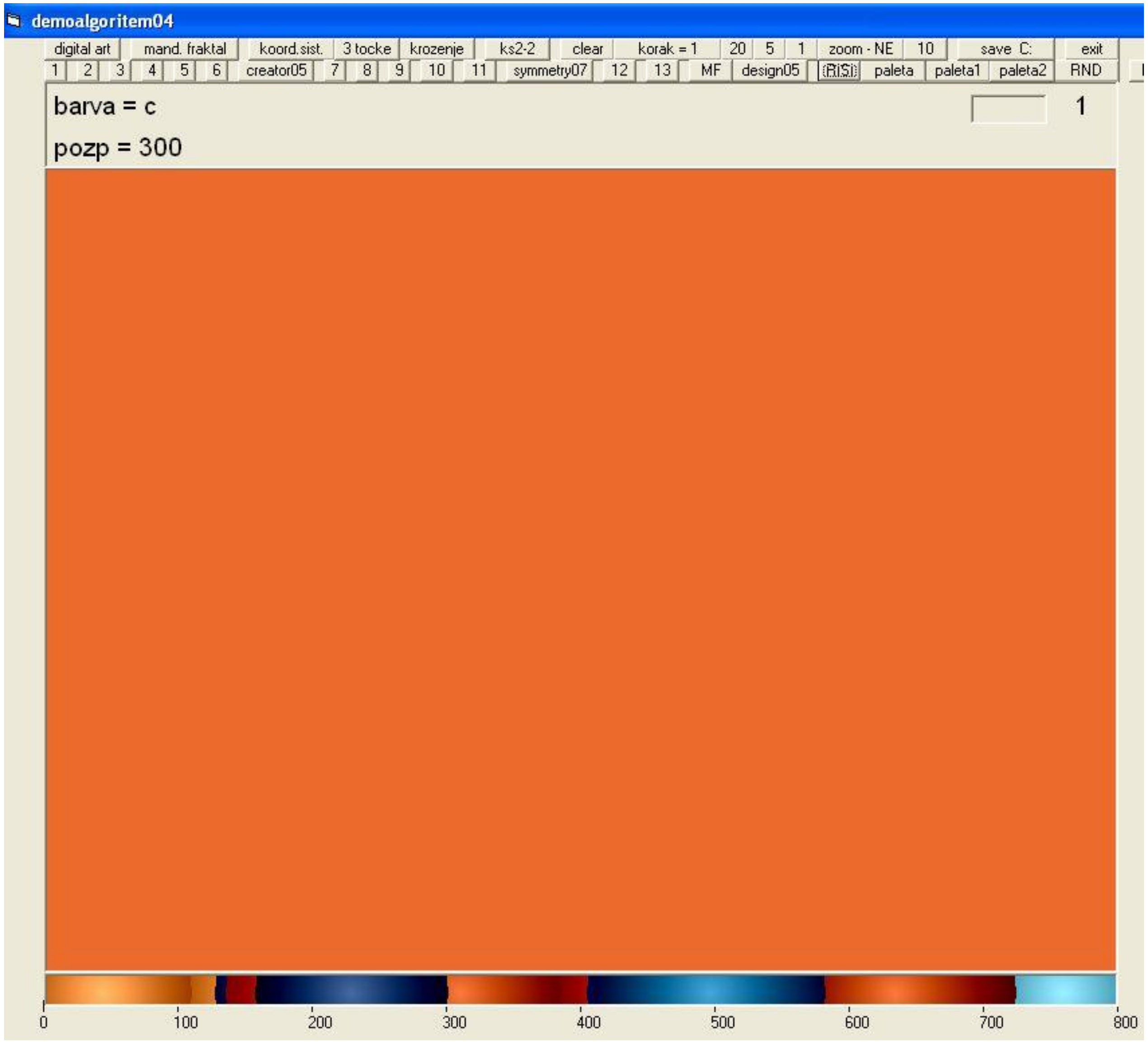
- Cartesian coordinate system
- Variables: i, j (VB6) x, y (Cartesian) rt1, rt2, rt3 (fixed points), rt (wandering point)
- Background processes
- Constants for the cycle: g1, g2, g3, g4, . . . (gene code)
- Unpredictable shapes

- Polar coordinate system
- Variables: r, fi (polar)
- Symmetrical shapes

- Drawing in the complex plane (complex numbers Z = x + yi)
- Perform iteration of Z(n) = Z(n-1)^2 + c on each pont of complex plane
- The result of iteration diverges or converges
- Variables: n (number of iteration)
- Fractal shapes (Mandelbrot, Julia)

Cartesian coordinate system

color = constant

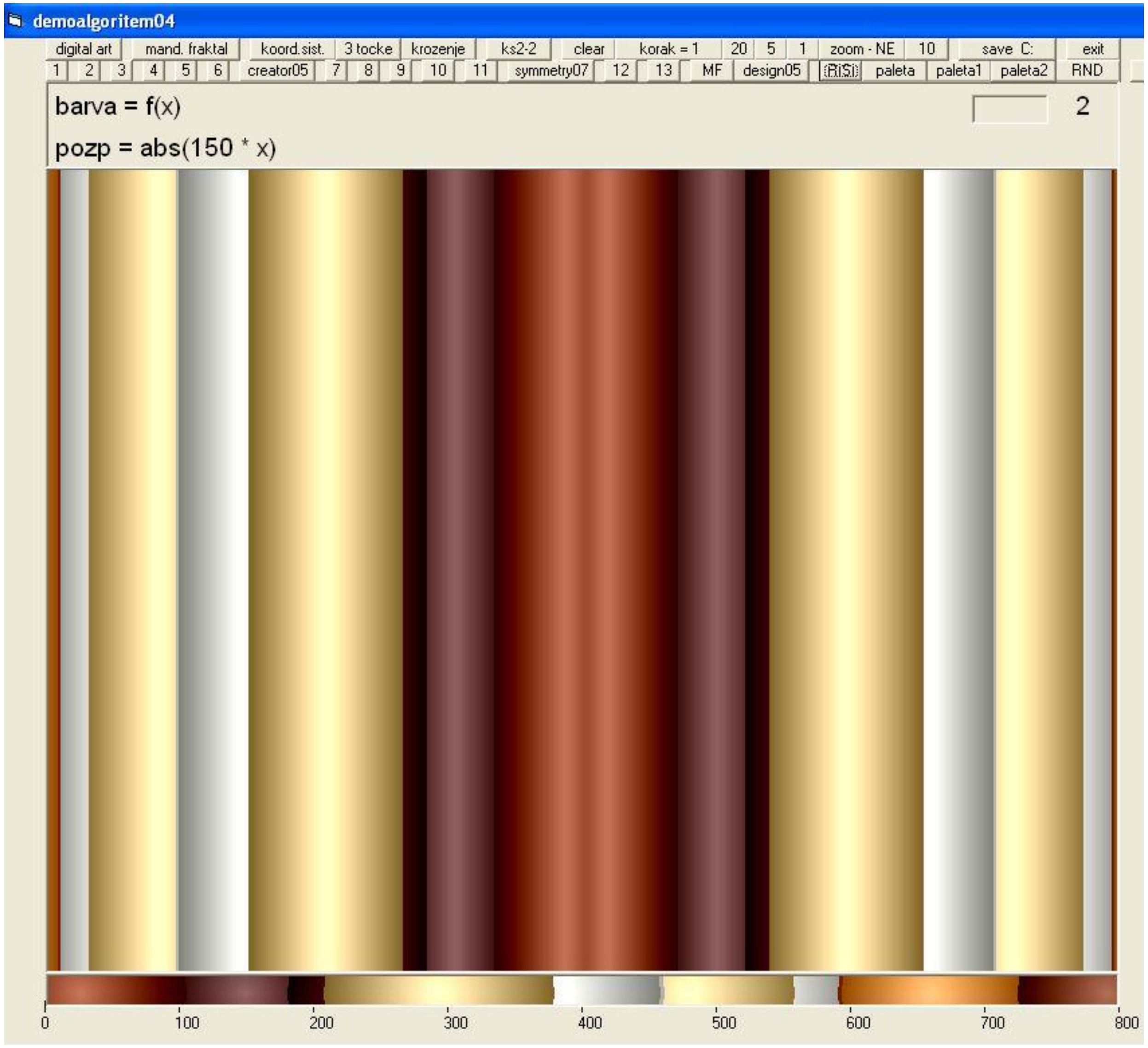


barva = c

pozp = 300

Cartesian coordinate system

color = ABS(150 * X)

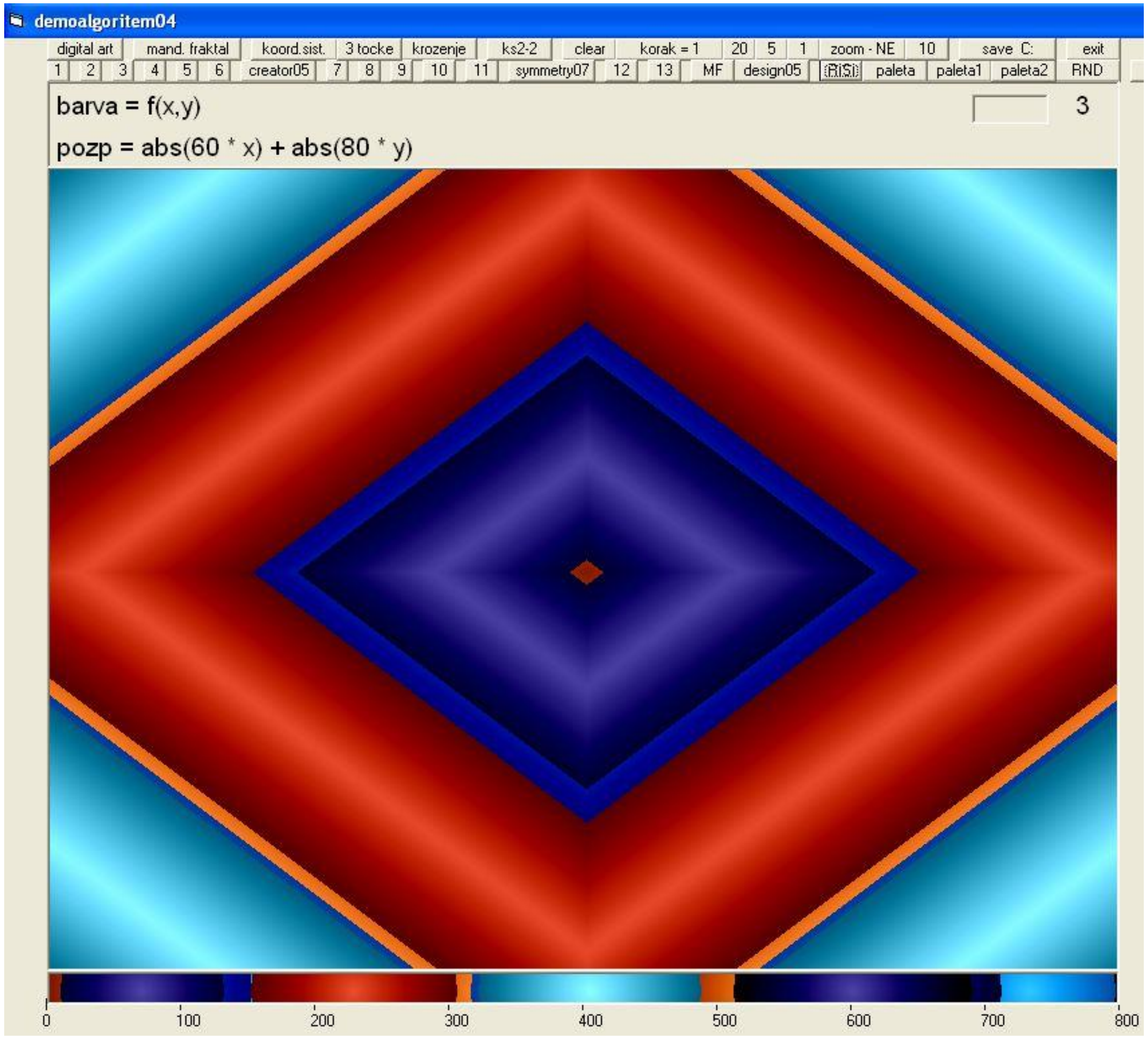


barva = f(x)

pozp = abs(150 * x)

Cartesian coordinate system

color = ABS(60 * X) + ABS(80 * y)

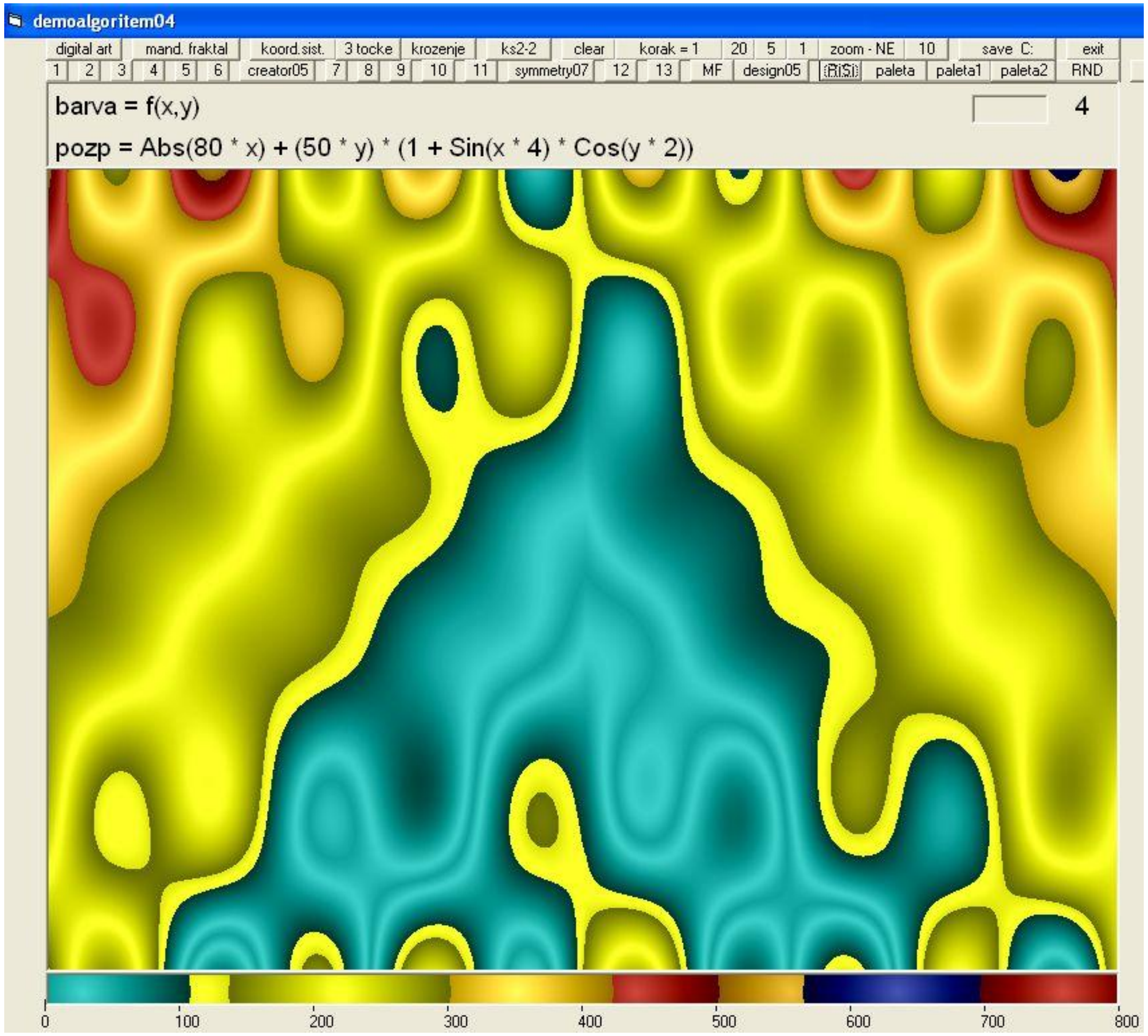


barva = f(x,y)

pozp = abs(60 * x) + abs(80 * y)

Cartesian coordinate system

color = ABS(80 * x) + (50 * y) * (1 + sin(x * 4) * cos(y * 2))

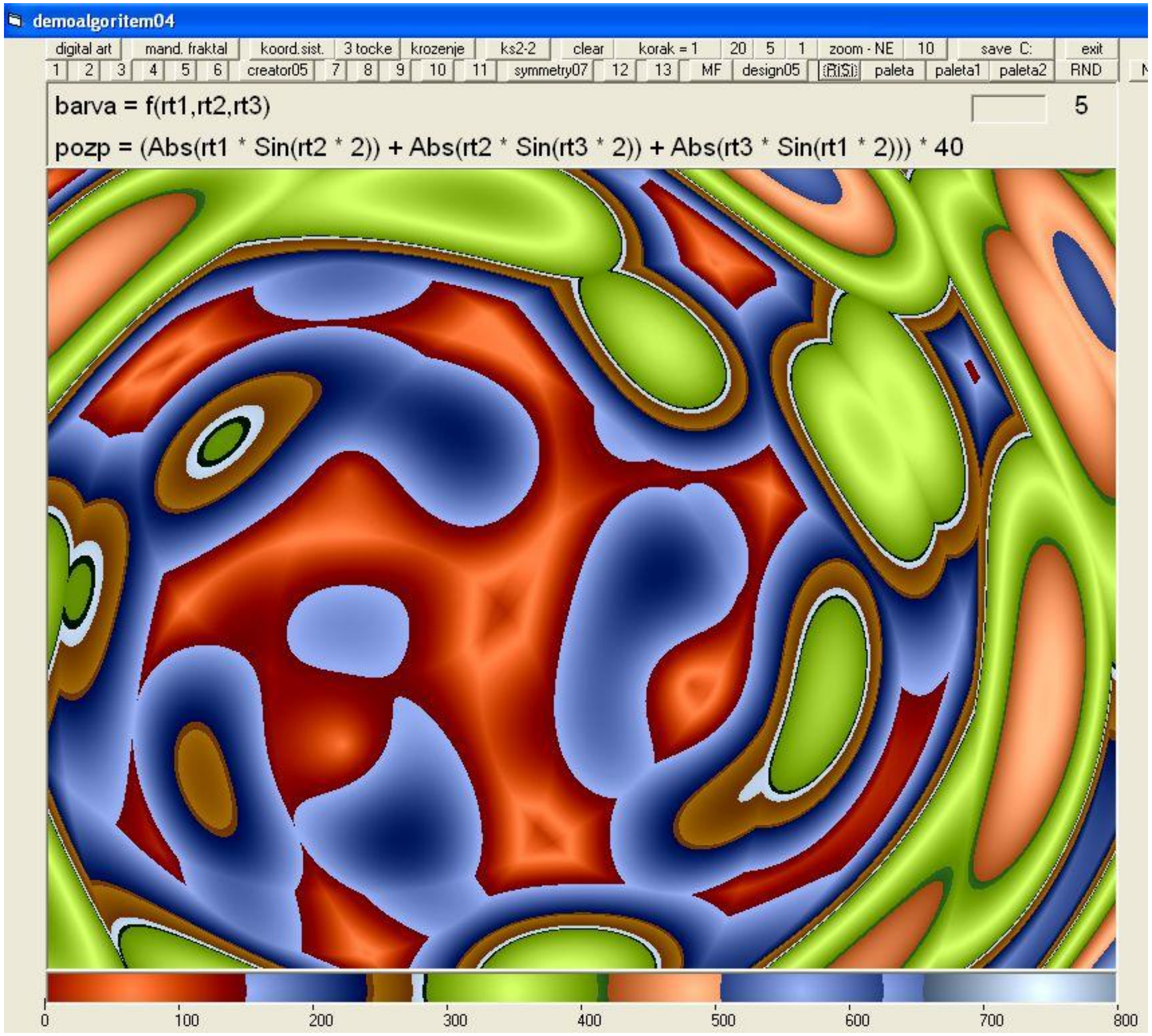


barva = f(x,y)

pozp = Abs(80 * x) + (50 * y) * (1 + Sin(x * 4) * Cos(y * 2))

Cartesian coordinate system

color = (ABS(rt1 * sin(rt2 * 2)) + ABS(rt2 * sin(rt3 * 2)) + ABS(rt3 * sin(rt1 * 2))) * 40

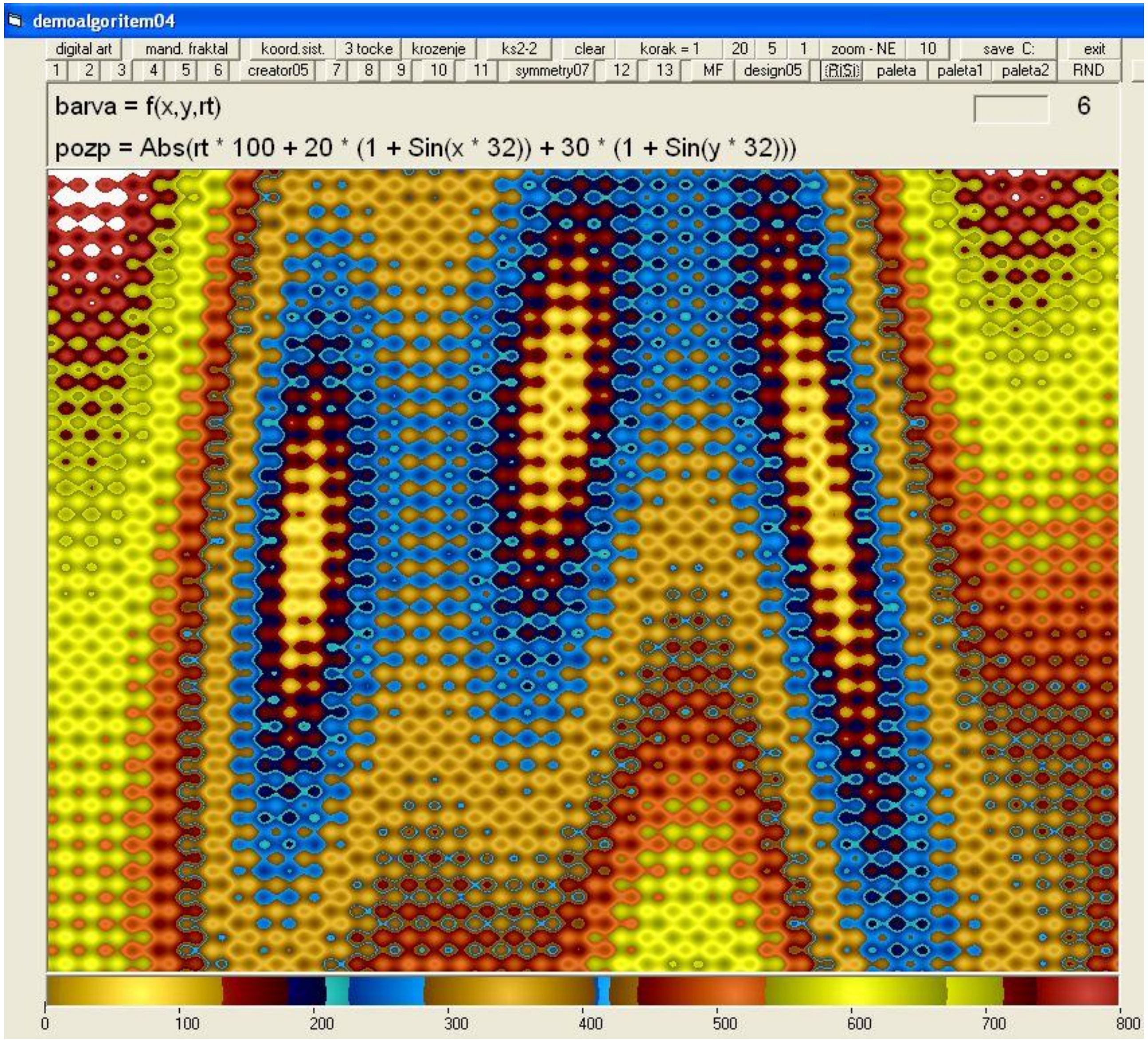


barva = f(rt1,rt2,rt3)   5

pozp = (Abs(rt1 * Sin(rt2 * 2)) + Abs(rt2 * Sin(rt3 * 2)) + Abs(rt3 * Sin(rt1 * 2))) * 40

Cartesian coordinate system

color = ABS(rt * 100 + 20 * (1 + sin(x *? 32)) + 30 * (1 + sin(y *32)))

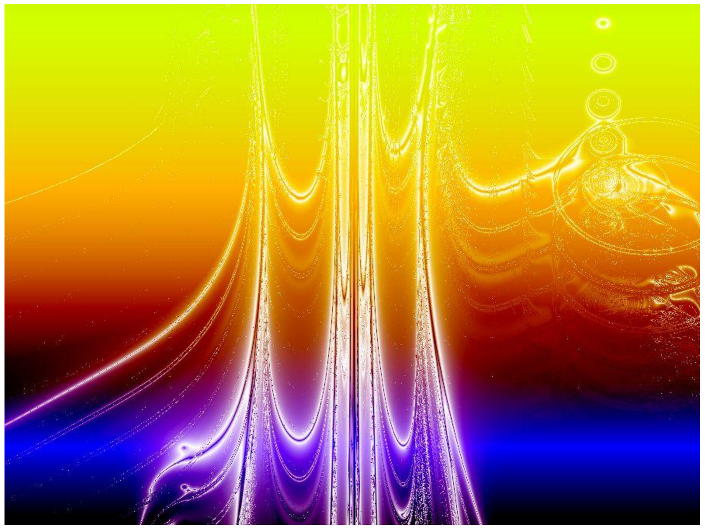
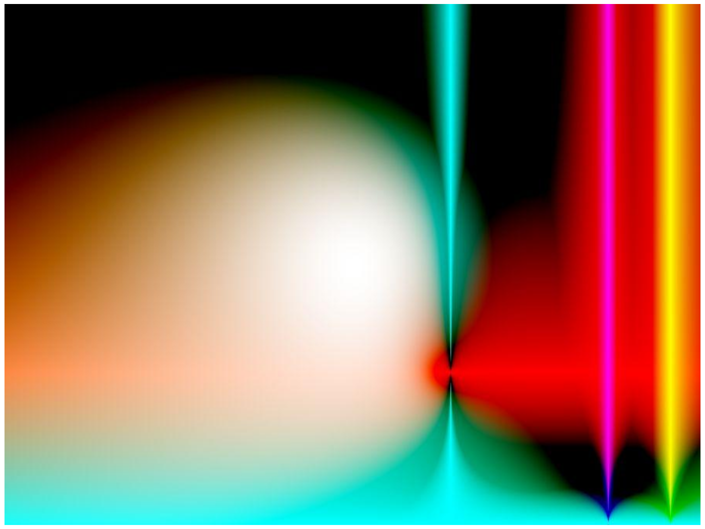
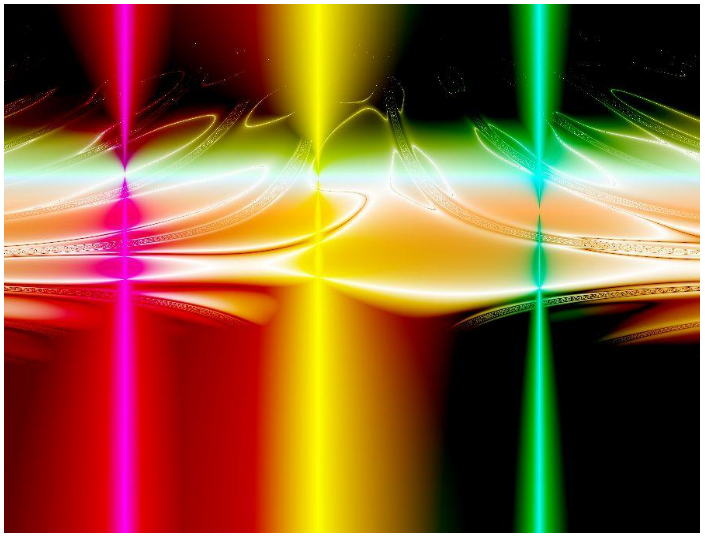
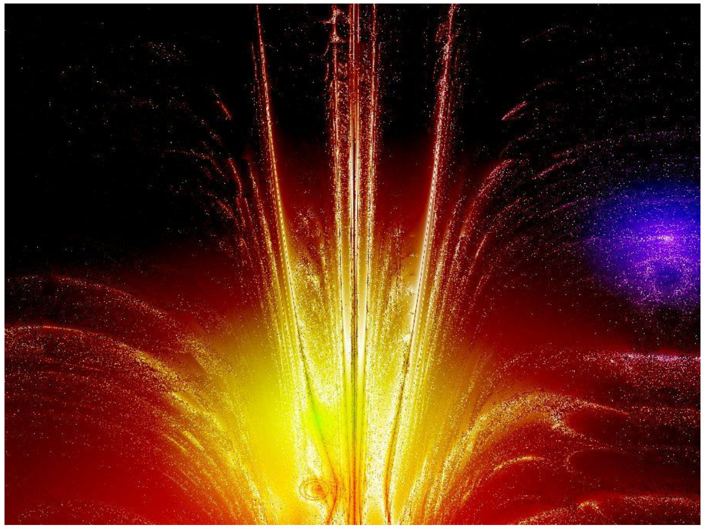


barva = f(x,y,rt)

pozp = Abs(rt * 100 + 20 * (1 + Sin(x * 32)) + 30 * (1 + Sin(y * 32)))

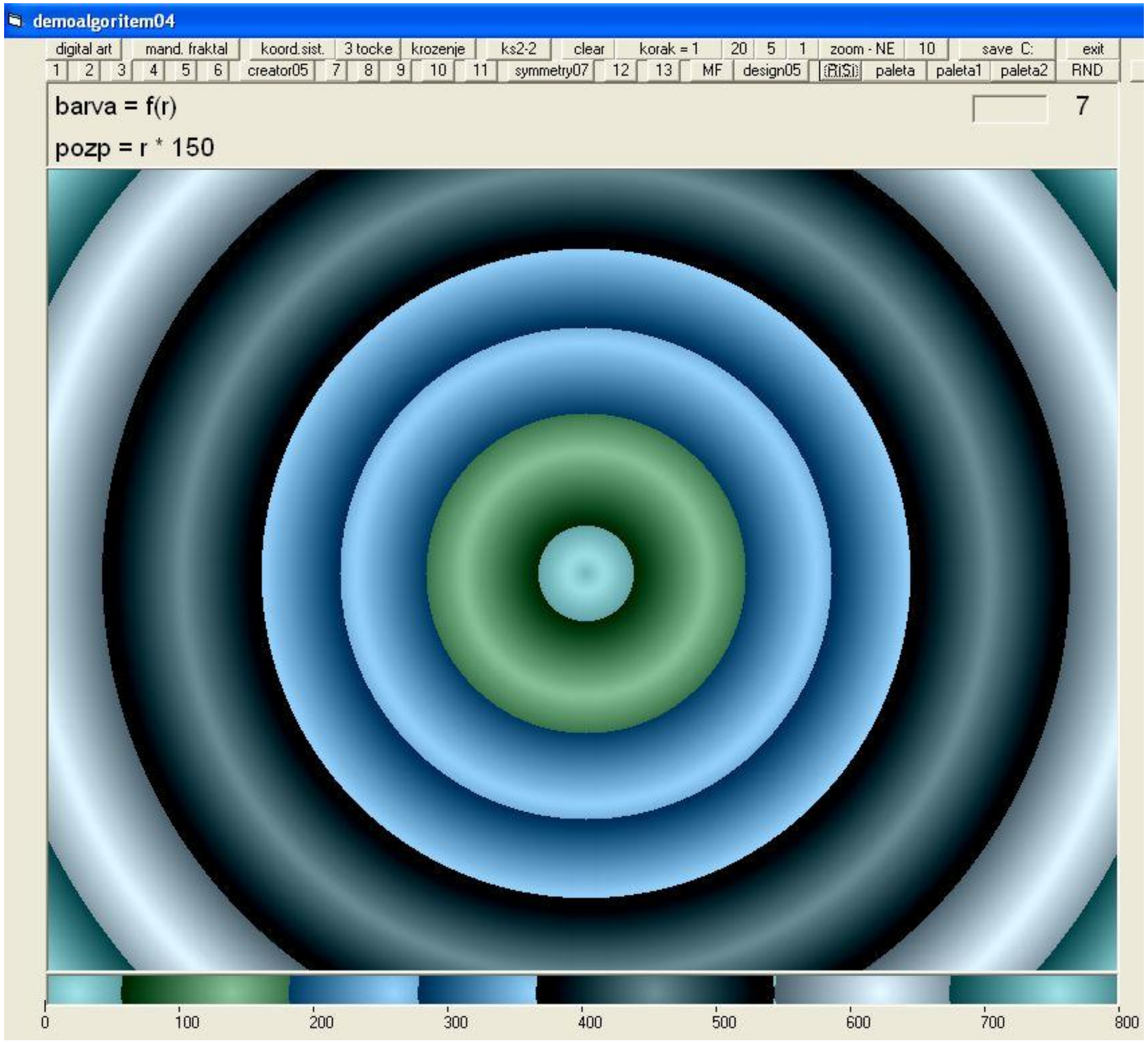
Cartesian coordinate system

examples from archive

Polar coordinate system

color = r * 150

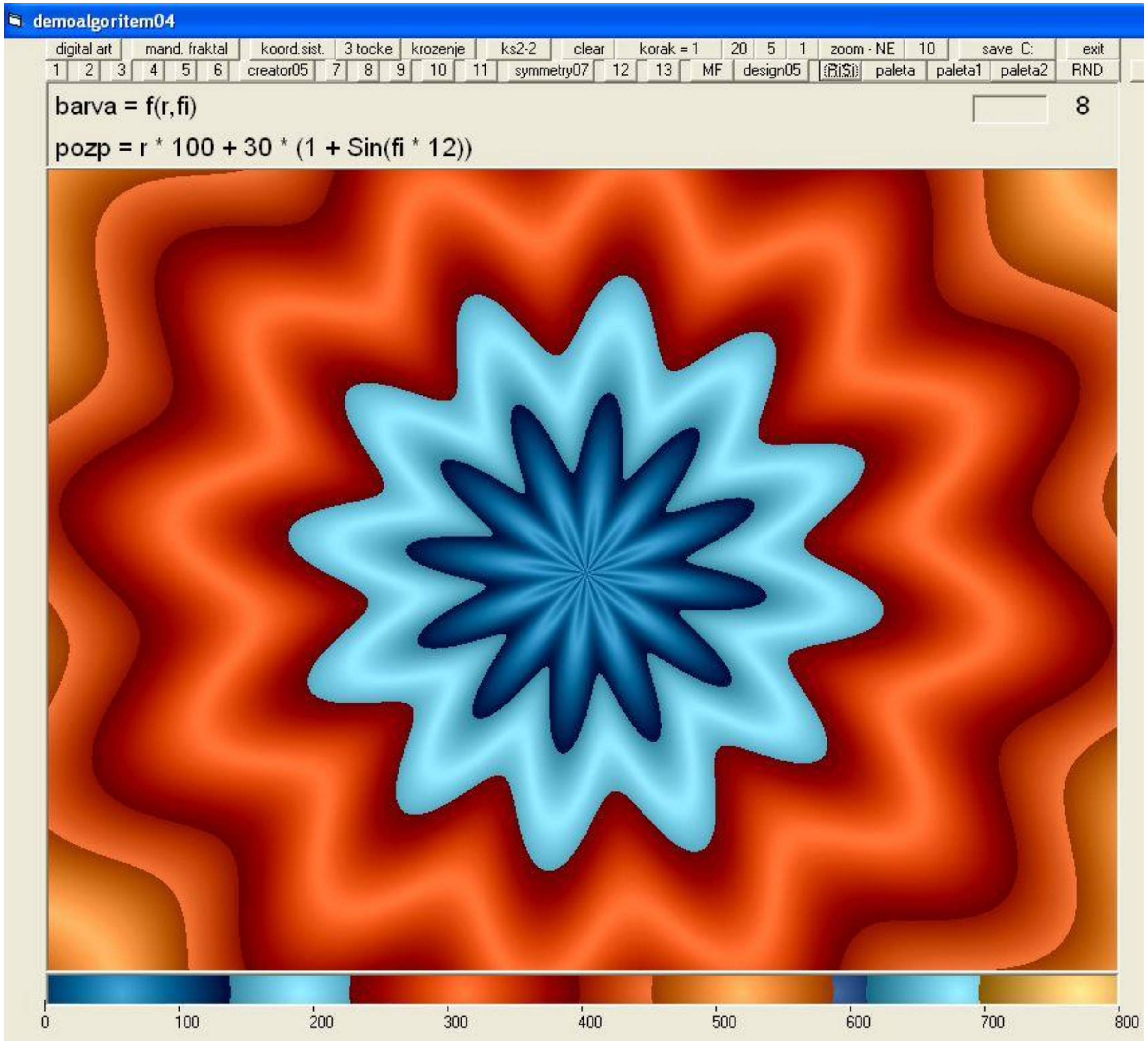


barva = f(r)

pozp = r * 150

Polar coordinate system

color = r * 100 + 30 * (1 + sin(fi * 12))

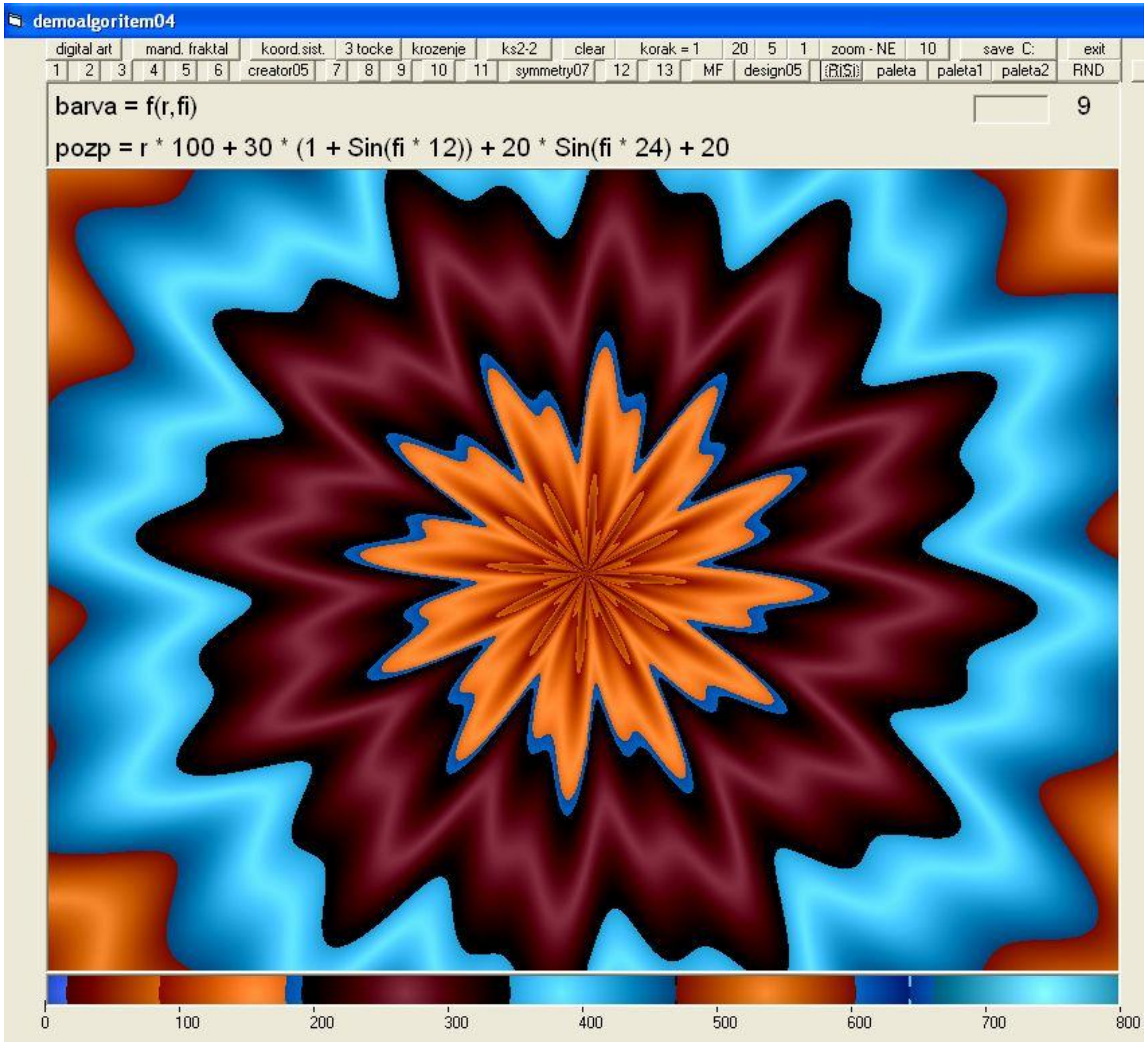


barva = f(r,fi)

pozp = r * 100 + 30 * (1 + Sin(fi * 12))

Polar coordinate system

color = r * 100 + 30 * (1 + sin(fi * 12)) + 20 * sin(fi * 24) + 20

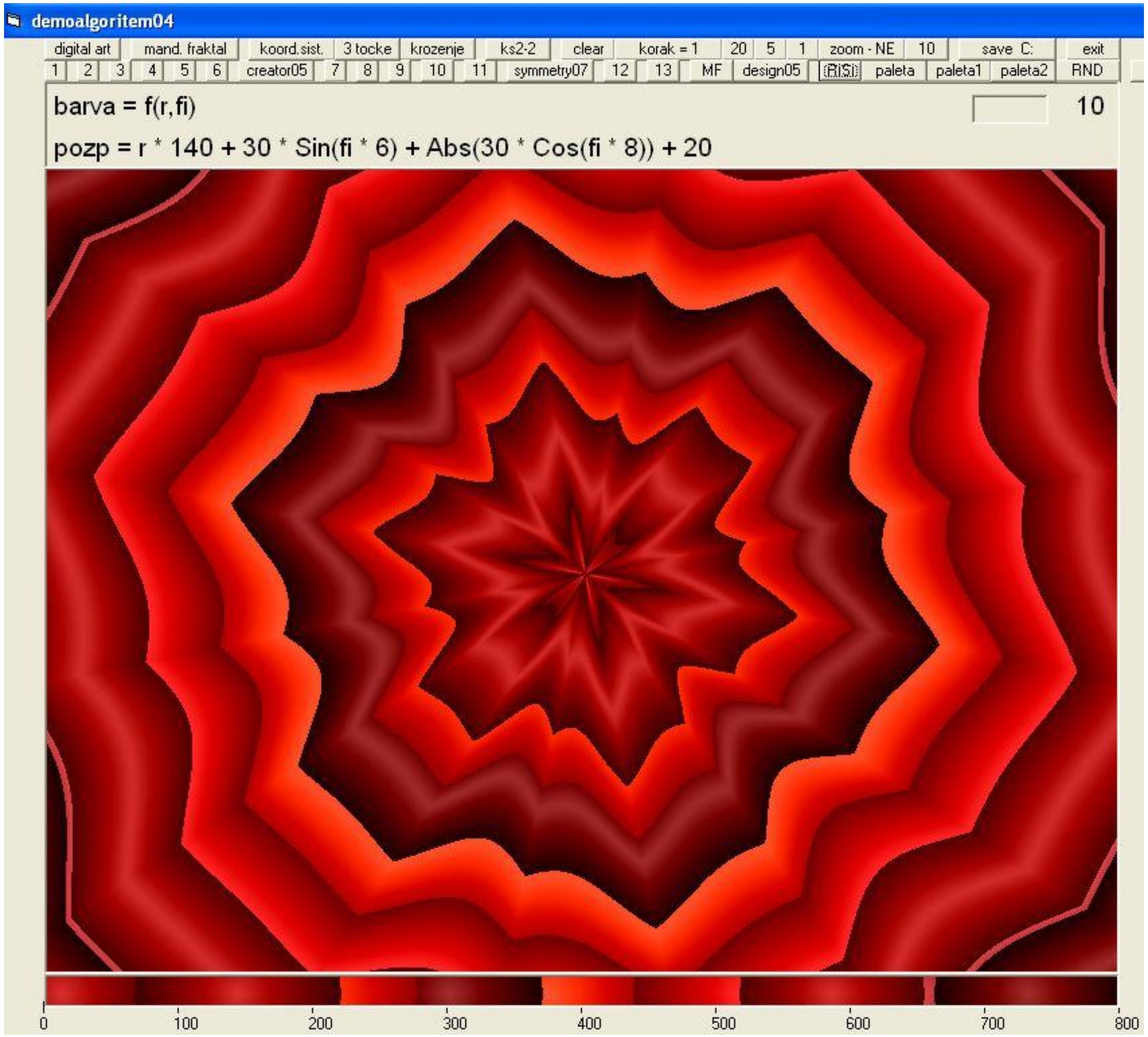


barva = f(r,fi)

pozp = r * 100 + 30 * (1 + Sin(fi * 12)) + 20 * Sin(fi * 24) + 20

Polar coordinate system

color = r * 140 + 30 * sin(fi * 6) + ABS(30 * cos(fi * 8)) + 20

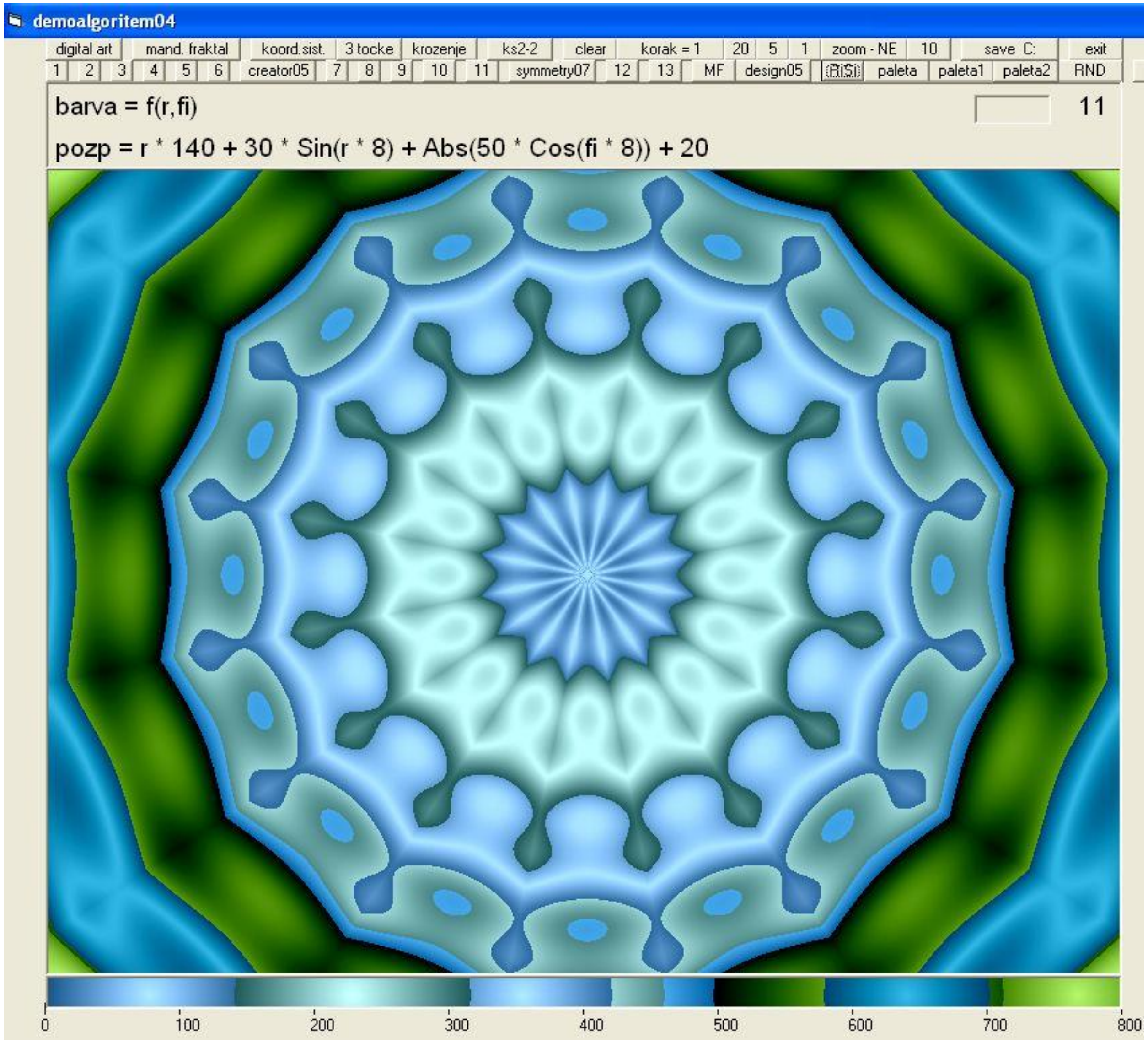


barva = f(r,fi)

pozp = r * 140 + 30 * Sin(fi * 6) + Abs(30 * Cos(fi * 8)) + 20

Polar coordinate system

color = r * 140 + 30 * sin(r * 8) + ABS(50 * cos(fi * 8)) + 20

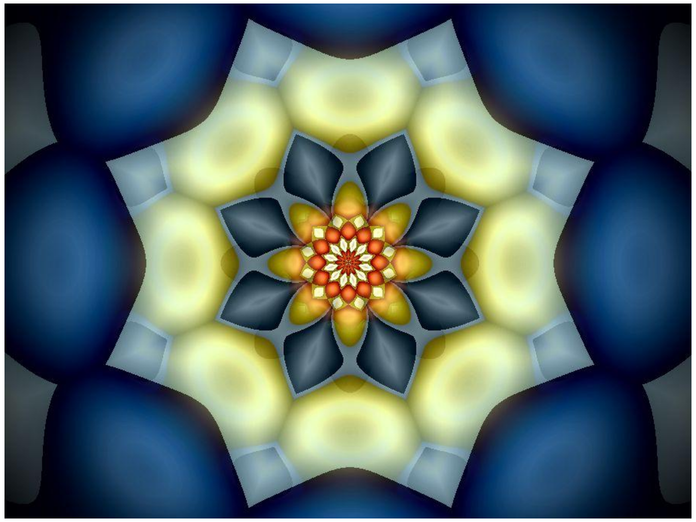
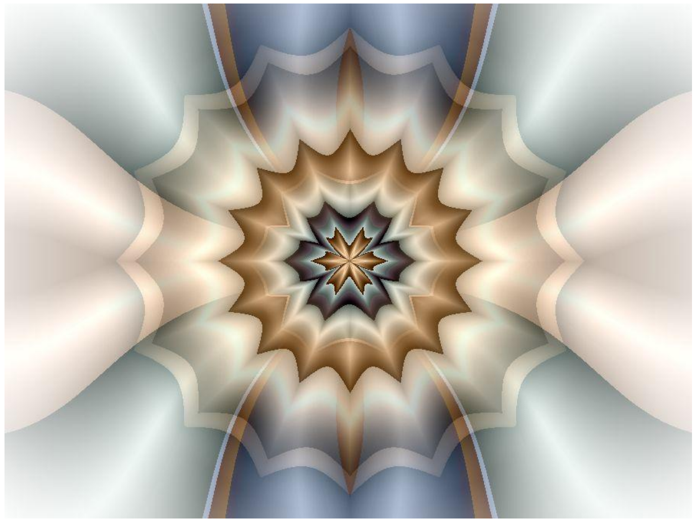


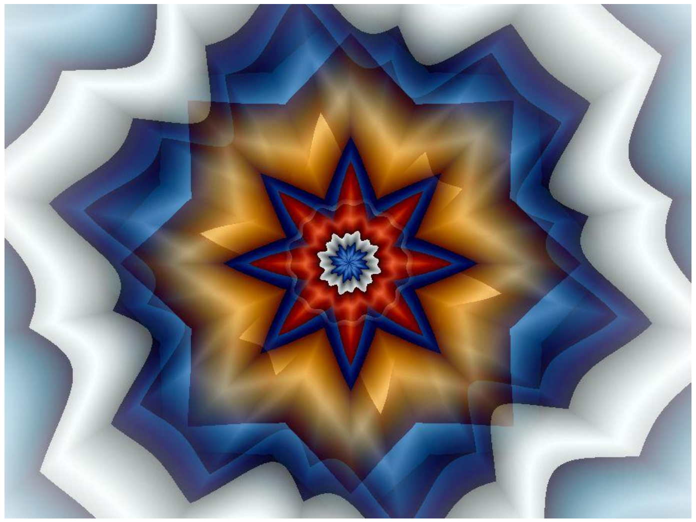
barva = f(r,fi)

pozp = r * 140 + 30 * Sin(r * 8) + Abs(50 * Cos(fi * 8)) + 20

Polar coordinate system

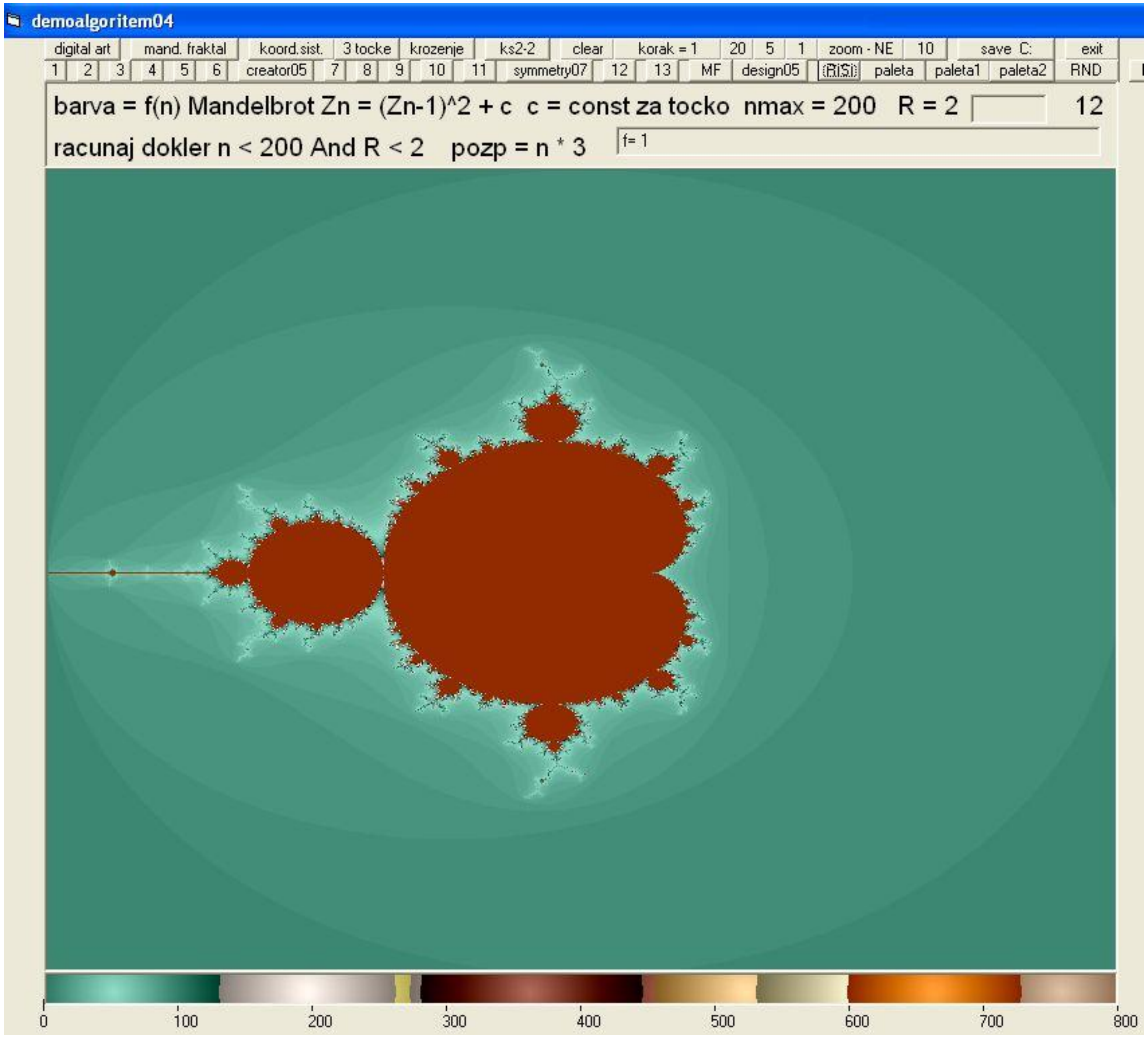
examples from archive



Complex plane

Mandelbrot fractal

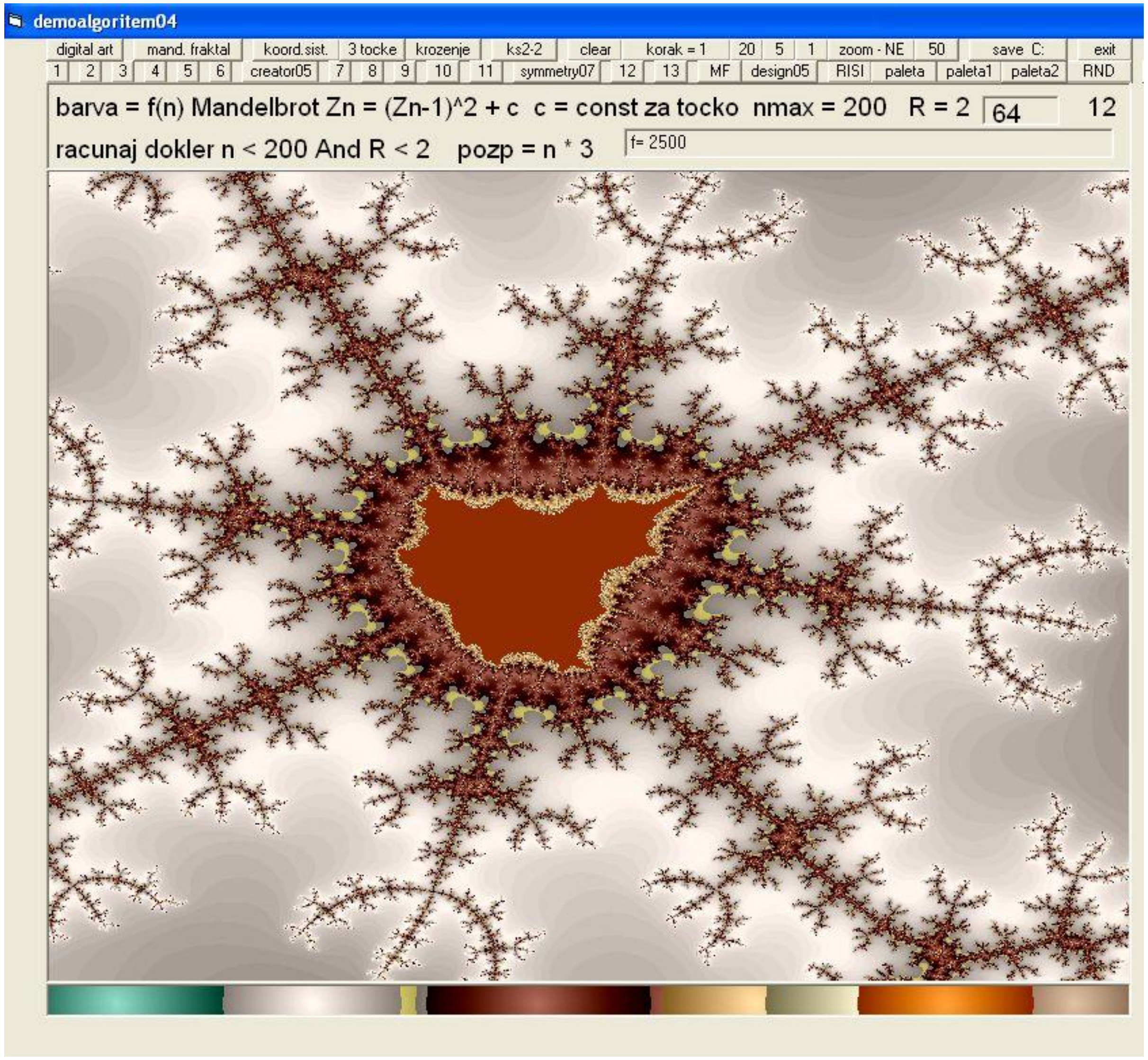


barva = f(n) Mandelbrot Zn = (Zn-1)^2 + c c = const za tocko nmax = 200 R = 2

racunaj dokler n < 200 And R < 2 pozp = n * 3

Complex plane

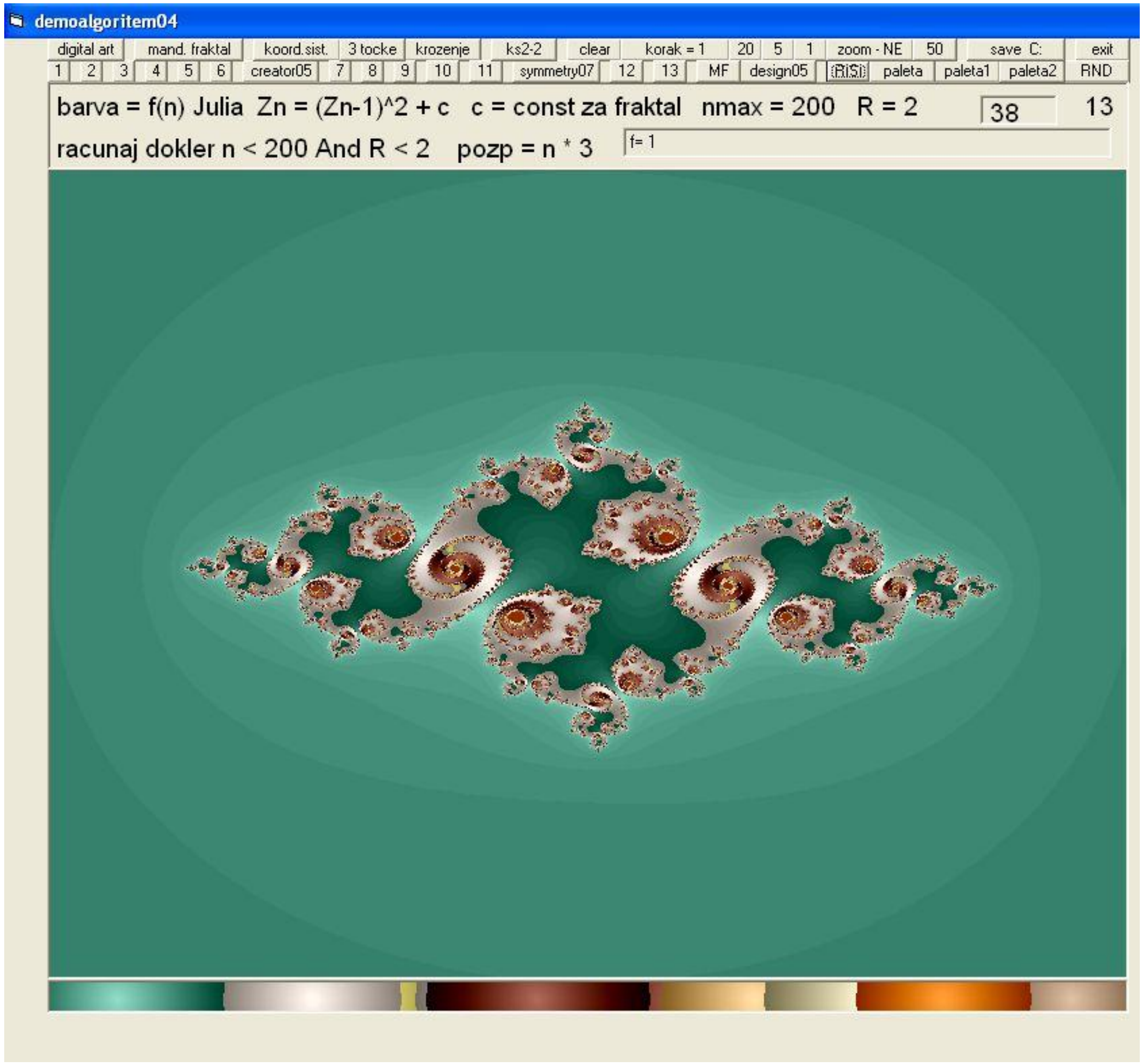
Mandelbrot fractal (magnification = 2500)



barva = f(n) Mandelbrot Zn = (Zn-1)^2 + c c = const za tocko nmax = 200 R = 2 64 12

racunaj dokler n < 200 And R < 2 pozp = n * 3 f= 2500

Complex plane
Julia fractal

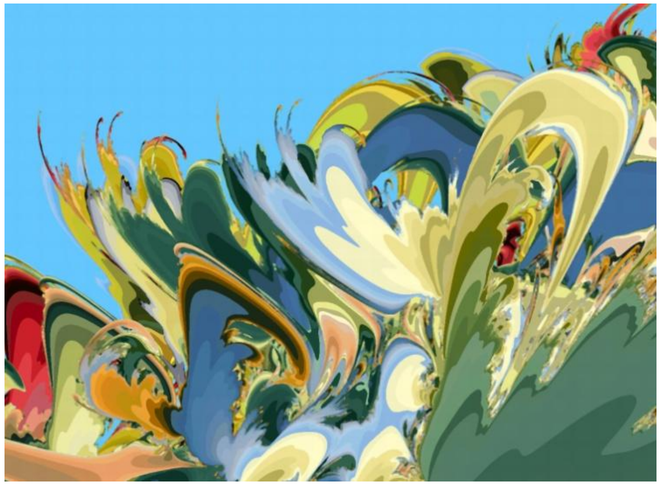


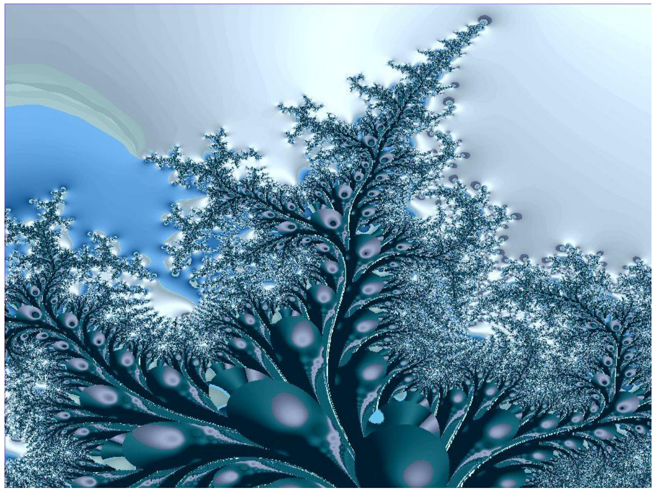
barva = f(n) Julia Zn = (Zn-1)^2 + c c = const za fraktal nmax = 200 R = 2 38 13

racunaj dokler n < 200 And R < 2 pozp = n * 3 f= 1

Complex plane

examples from archive

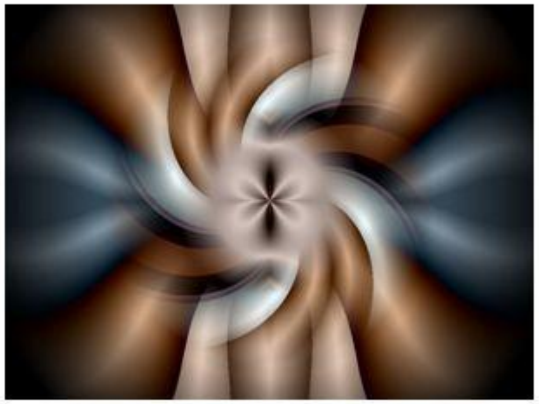


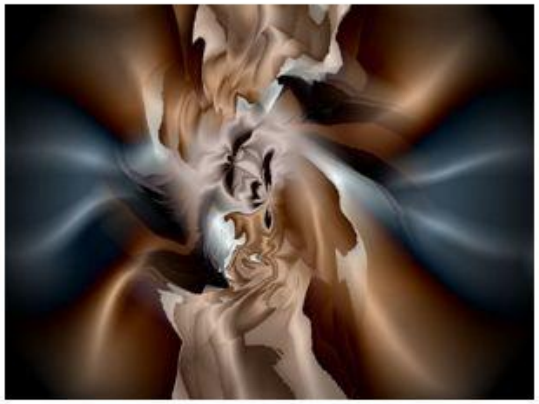
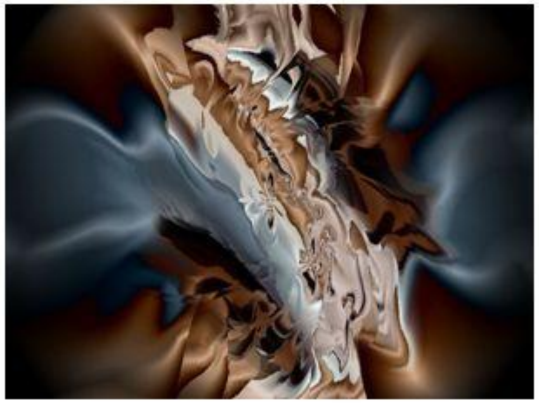


From mathematical order to shapeless forms DECOMPOSITION OF SYMMETRY - generation of symmetrical shapes and their transformation into shapeless images

- Generate symmetry forms
- Elaborate symmetry forms using deformation algorithm
- The process is similar to a two-dimensional cellular automaton
- Algorithm takes the color of one of the neighbor points
- After each cycle the picture is different and not similar to “image mother”
- The use of mutation process
- Deformate mutated images
- Run demostration program

example



Thank you for your attention