Roxana Drăgănoiu

Programare Matematică Grafică

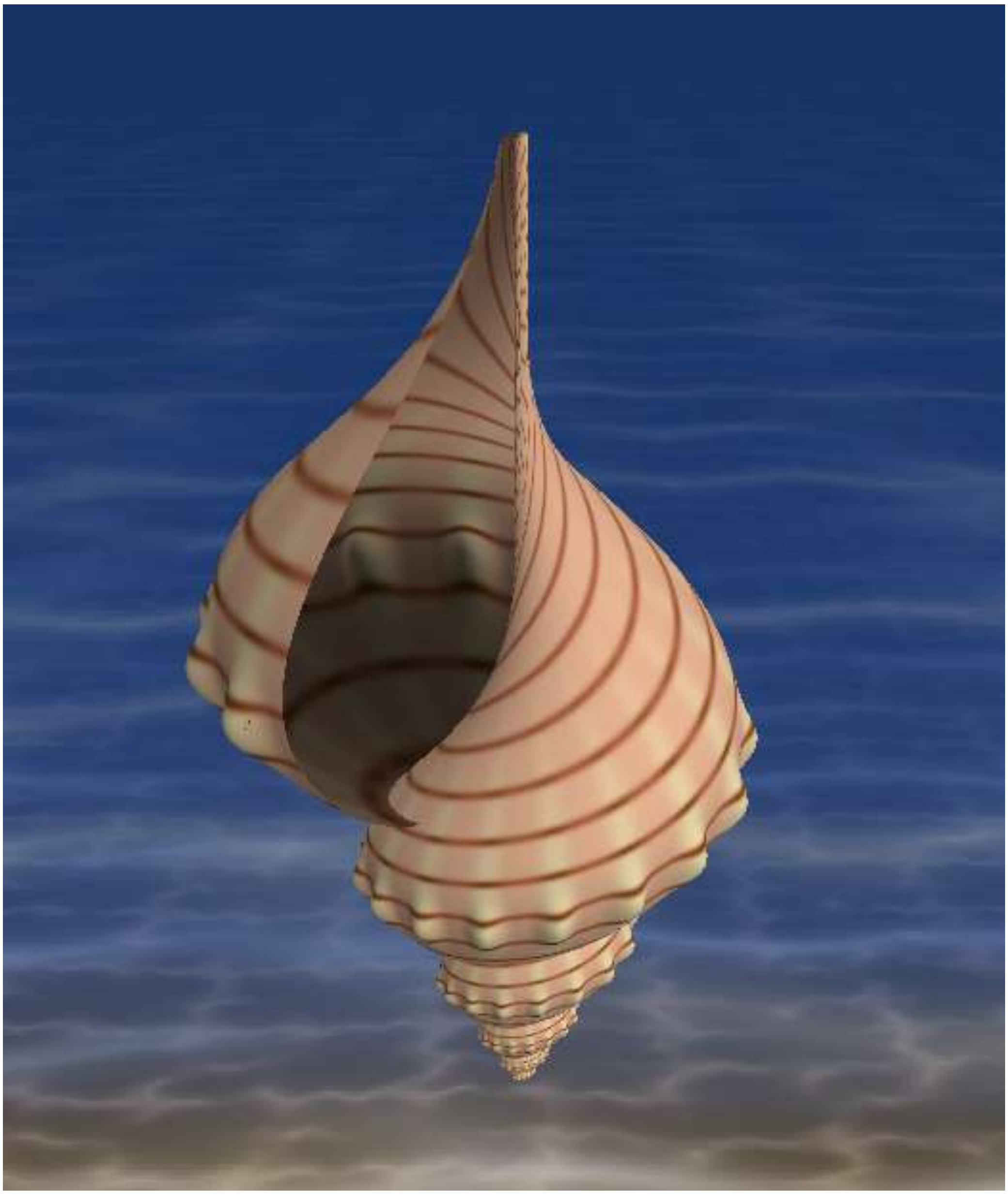
Modelare Matematică

2017
Bucureşti
Atelier Didactic

Roxana Draganoiu

Programare Matematica Grafica

Modelare Matematica



Introducere

DMA este o metoda de studiu/predare practica si atractiva a matematicii.
Este o adaptare a tehnicilor de programare pentru elevii de liceu, intrucat utilizeaza tehnici simple pe care le avem la dispozitie in programa de liceu atat de mate cat si programare. Chiar daca unele notiuni depasesc programa de liceu, ele sunt explicate intr-un mod simplu pe baza acesteia.

Cartea ofera un instrument de studiu al matematicii tinerilor pasionati de matematica si respectiv un instrument de predare profesorilor de matematica. Nu acopera teoria matematica ci doar strictul necesar modelarii matematice.

Programele din aceasta carte au fost realizate in Processing, datorita avantajelor pe care le are pentru grafica. Programarea se poate realiza insa in orice limbaj de programare si in orice IDE care sustine grafica.

Processing

Processing is an open source computer programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching the fundamentals of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks**.** Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts.
Limbajul utilizat este Java.

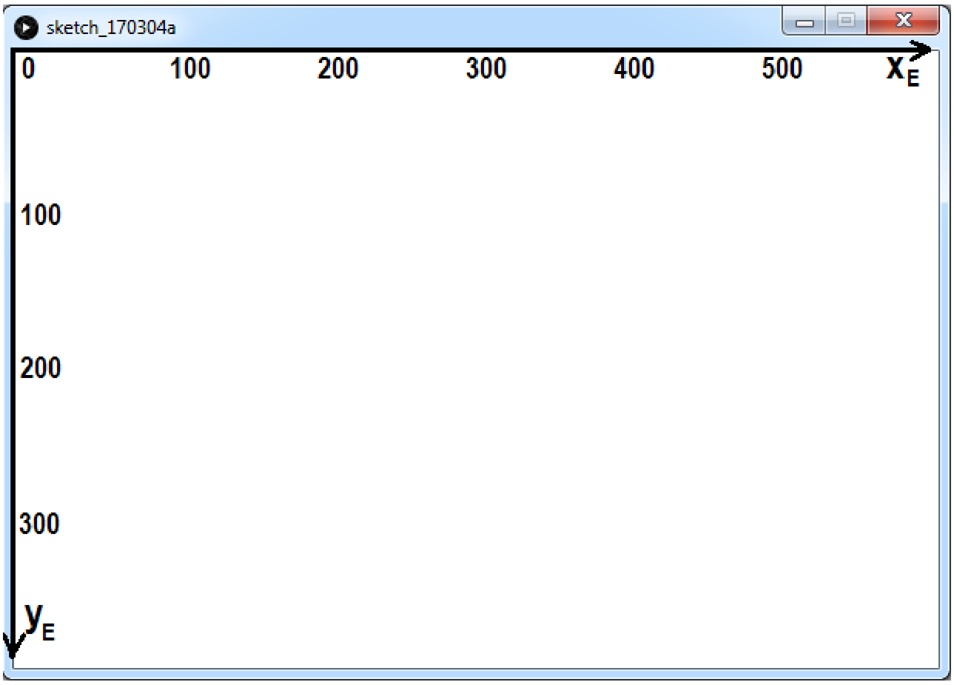
https://processing.org/download/?processing

O aplicatie simpla

Terorie programare

Coordonate ecran

The coordinate system for a window is based on the coordinate system of the display device. The basic unit of measure is the device unit (typically, the pixel). Points on the screen are described by x- and y-coordinate pairs. The x-coordinates increase to the right; y-coordinates increase from top to bottom. Originea O(0,0) este situata in coltul stanga-sus al ferestrei.



Aplicatie

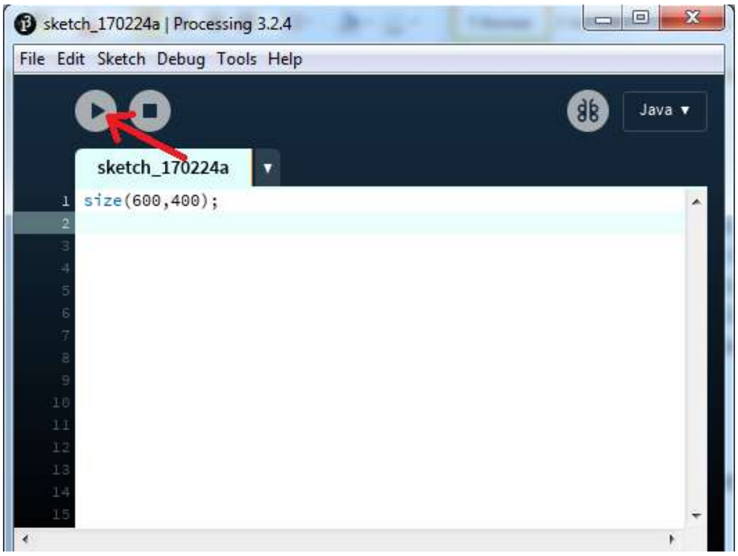
Vom redimensiona fereastra cu ajutorul functiei *size*. Implicit, fereastra este de 100x100 pixeli.

Teorie programare
size(Max_x,Max_y) – redimensioneaza fereastra cu valorile *Max_x* pentru latime si *Max_y* pentru inaltime

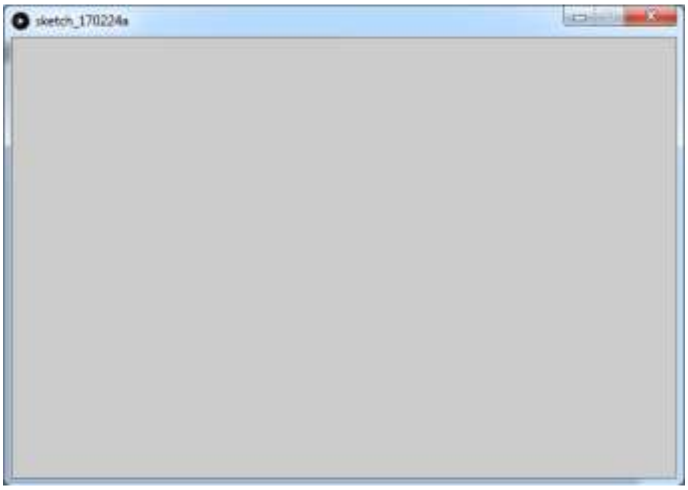
Aplicatie

Program

```
size(600,400);
```

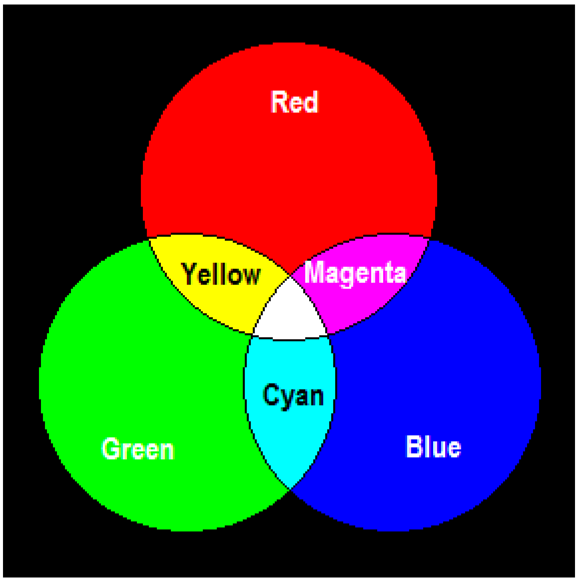


Rularea se face de la butonul stanga-sus, indicat cu sageata rosie. Se va deschide o fereastra cu latimea de 600 pixeli si inaltimea de 400 pieli.



Culori - Modelul RGB

Modelul cromatic RGB (**R**ed **G**reen **B**lue) este un model aditiv de culoare, în care culorile albastră, roșie și verde sunt amestecate în diferite moduri pentru a produce o gamă largă de culori.



R=Red(rosu)	G=Green(verde)	B=Blue(albastru)	culoare
0	0	0	
255	255	255	
255	0	0	
0	255	0	
0	0	255	
255	255	0	
255	0	255	
0	255	255	

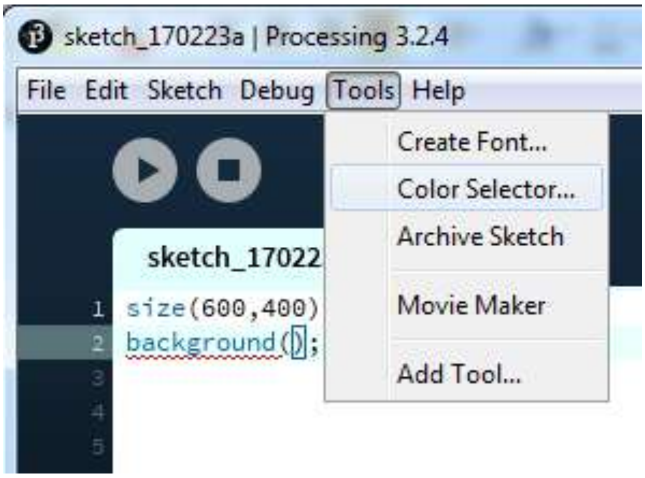
Teorie programare
background(Red,Green,Blue) – seteaza culoarea fundalului. *Red, Green* si *Blue* sunt numere.
background(value) – seteaza intensitatea de gri. 0 pentru negru, 255 pentru alb

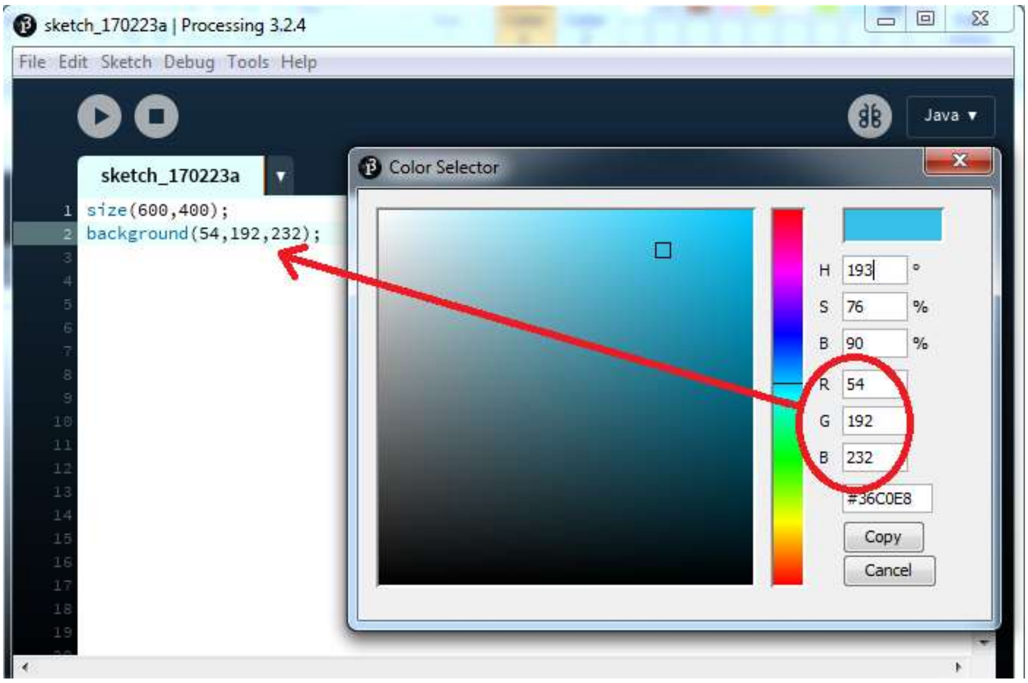
Aplicatie

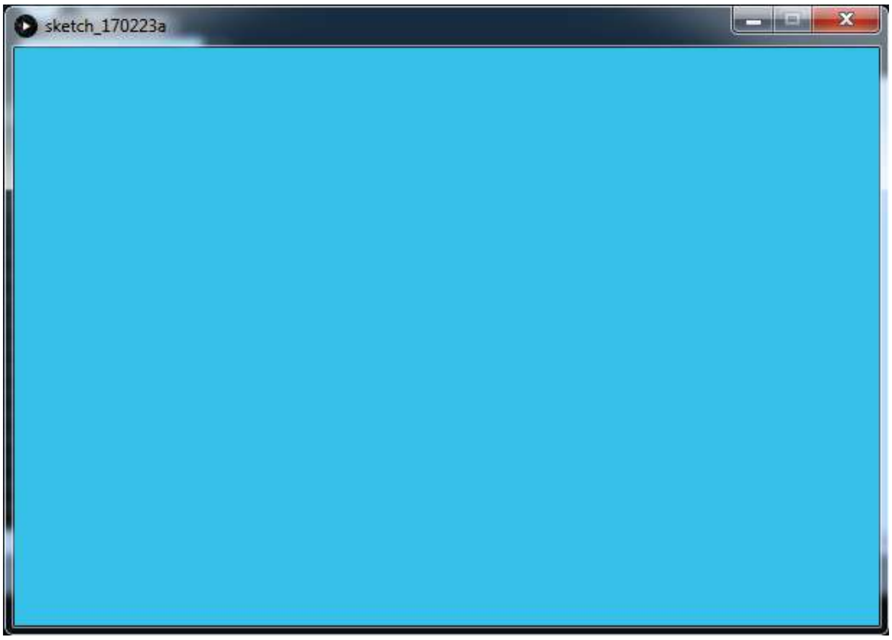
Vom schimba culoarea fundalului cu ajutorul functiei *background*. Valorile parametrilor functiei se pot alege din meniul *Tools->Color Selector*.

Program

```
size(600,400);
background(255);
```







Functii pentru grafica in Processing

<u>Teorie programare</u>

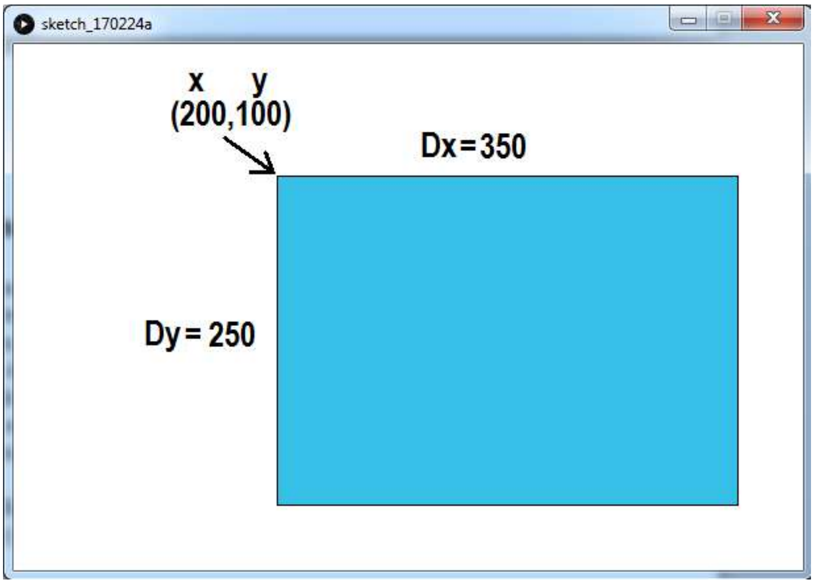
point(x,y) – deseneaza un punct de coordonate (x,y)
set(x,y,color(Red,Green,Blue)) – deseneaza un punct de coordonate (x,y) cu coloarea (Red,Green,Blue)
line(x1,y1,x2,y2) – deseneaza o linie de la (x1,y1) la (x2,y2)
rect(x,y,Dx,Dy) – deseneaza un dreptunghi cu coltul stanga-sus in (x,y) si dimensiunile *Dx* si *Dy*
ellipse(x,y,Dx,Dy) – deseneaza o elipsa cu centrul in (x,y) si dimensiunile *Dx* si *Dy*
stroke(Red,Green,Blue) – seteaza culoarea de desenare
fill(Red,Green,Blue) – seteaza culoarea de umplere a formelor

Aplicatie

In continuare vom desena un dreptunghi cu ajutorul functiei *rect*, pe care il vom colora cu culoarea aleasa in exemplul precedent, cu ajutorul functiei *fill*.

<u>Program</u>

```
size(600,400);
background(255);
fill(54,192,232);
rect(200,100,350,250);
```



Acoperirea suprafetelor in coordonate ecran

Teorie programare

Tipuri de variabile si declaratii de variabile:
int x; - variabila x este de tip intreg (N,Z)
float x; - variabila x este de tip real (Q,R)

Instructiunea repetitiva *for*, are rolul de a repeta o instructiune sau un set de instructiuni.

Structura

```
for(x=a;x<=b;x++)
  //instructiune
```

sau

```
for(x=a;x<=b;x++)
{
  //set instructiuni
}
```

Tipuri de incrementari/decrementari
x++ sau x=x+1 – creste din 1 in 1
x+=2 sau x=x+2 – creste din 2 in 2

x—sau x=x-1 – scade din 1 in 1
x-=2 sau x=x-2 – scade din 2 in 2

Traducerea limbajului matematic in programare

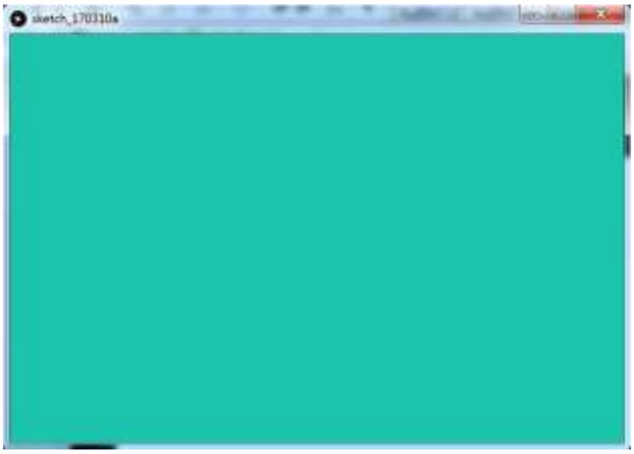
Math	↔	Programming
x=a	↔	x=a
if x=a	↔	if(x==a)
if x<a	↔	if(x<a)
if x≤a	↔	if(x<=a)
if x>a	↔	if(x>a)
if x≥a	↔	if(x>=a)
x div a	↔	x/a
x mod a	↔	x%a
x∈N,Z	↔	int x;
x∈Q,R	↔	float x;
x∈{a, a+1, ... ,b}	↔	for(x=a;x<=b;x++)
x∈[a,b]	↔	for(x=a;x<=b;x+=0.01)
(x,y)∈ [a,b]x[c,d]	↔	for(y=c;y<=d;y+=0.01) for(x=a;x<=b;x+=0.01)

f:[a,b]→R
$f(x)=\begin{cases} f1(x), & x\leq c \\ f2(x), & x>c \end{cases}$
	↔	for(x=a;x<=b;x+=0.01)
		if(x<=c)
		f=f1
		else
		f=f2

Aplicatie
Vom umple suprafata ferestrei din pixeli de aceeasi culoare.

Program

```
size(600,400);
background(255);
int x,y;
for(y=0;y<=400;y++)
  for(x=0;x<=600;x++)
    set(x,y,color(28,196,172));
```

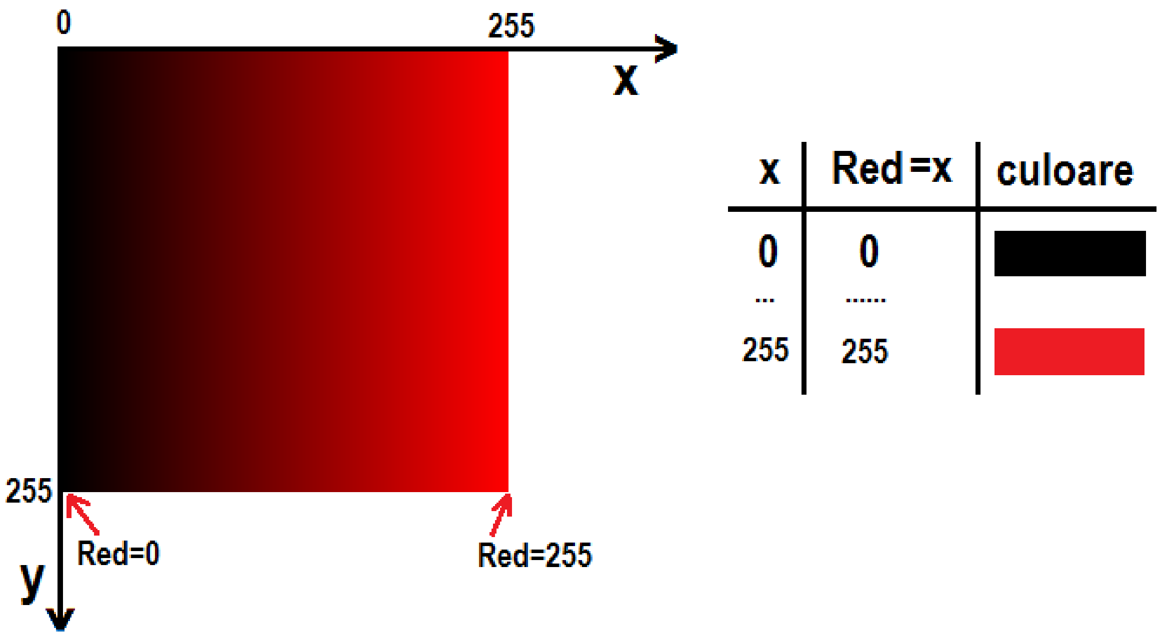


Colorarea suprafetelor cu degradeuri

Aplicatie

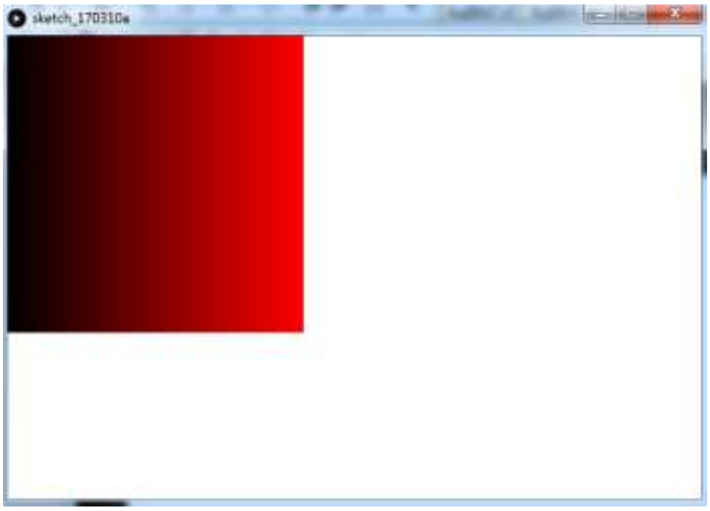
Vom desena un patrat colorat printr-un degrade de la negru la rosu, de-alungul axei OX. Pentru a face o asociere cat mai simpla intre coordonate si culori, vom alege latura patratului de 256 de pixeli, cu coordonate de la 0 la 255, adica exact aceleasi valori pe acre le va lua si parametrul Red.

Rezolvare matematica



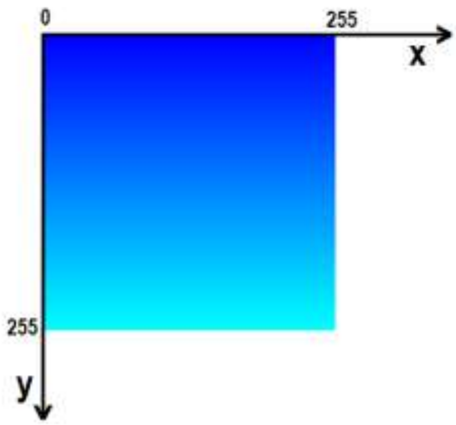
Program

```
size(600,400);
background(255);
int x,y,Red;
for(y=0;y<=255;y++)
  for(x=0;x<=255;x++)
  {
    Red=x;
    set(x,y,color(Red,0,0));
  }
```



In cazul in care degradeul se realizeaza pe verticala, atunci, parametrii culorii se calculeaza in functie de *y*.

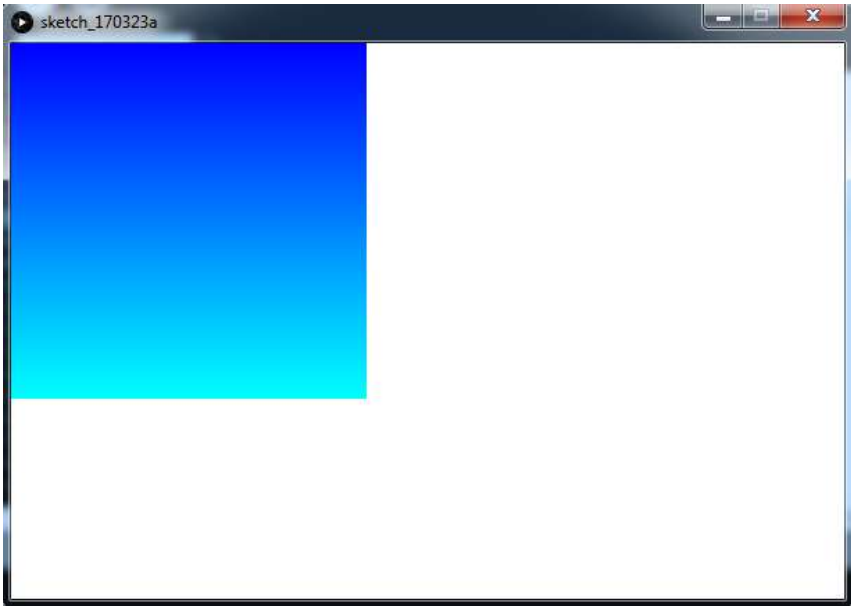
Aplicatie



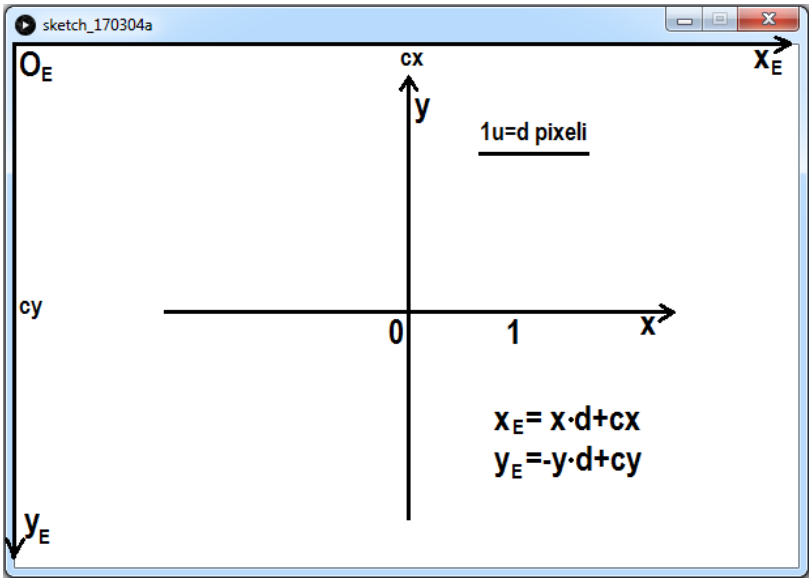
y	R=0	G=y	B=255	culoare
0	0	0	255	
......	......	......	......	
255	0	255	255	

<u>Program</u>

```
size(600,400);
background(255);
int x,y;
for(y=0;y<=255;y++)
  for(x=0;x<=255;x++)
  {
    set(x,y,color(0,y,255));
  }
```



Trecerea de la coordonate matematice la coordonate ecran



Trecerea de la coordonate matematice la coordonate ecran se face cu ajutorul formulelor:
xE=xd+cx
yE=-yd+cy

-unde xE si yE reprezinta coordonatele unui punct in coordonate ecran, d reprezinta numarul de pixeli ai unitatii reperului cartezian, iar cx si cy - coordonatele centrului reperului cartezian.

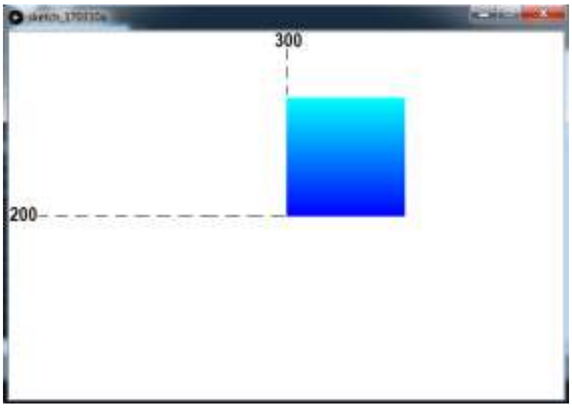
Aplicatie

Vom reprezenta in coordonate matematice patratul din aplicatia precedenta.

Intrucat in coordonate matematice nu mai lucram doar cu valori intregi, este necesar sa schimbam tipul variabilelor de la *int* la *float.* De asemeni, trebuie sa declaram variabilele d,cx si cy, pe care le si initializam. Intrucat variabilele x si y nu mai sunt in sistem ecran ci in sistem matematic, trebuie sa inlocuim parametrii functiei set, cu reprezentarile coordonatelor punctelor (x,y) in sistem ecran. Iar pasul de crestere al variabilelor iterative, trebuie sa fie proportional cu *d*, adica de forma *k/d*, unde *k* se alege convenabil, astfel incat sa nu ramana spatii neacoperite, cu conditia 0<*k*<1.

Program

```
size(600,400);
background(255);
float x,y,d=0.5,cx=300,cy=200;
for(y=0;y<=255;y+=0.8/d)
  for(x=0;x<=255;x+=0.8/d)
  {
    set(int(x*d+cx),int(-y*d+cy),color(0,y,255));
  }
```



Roxana Draganoiu

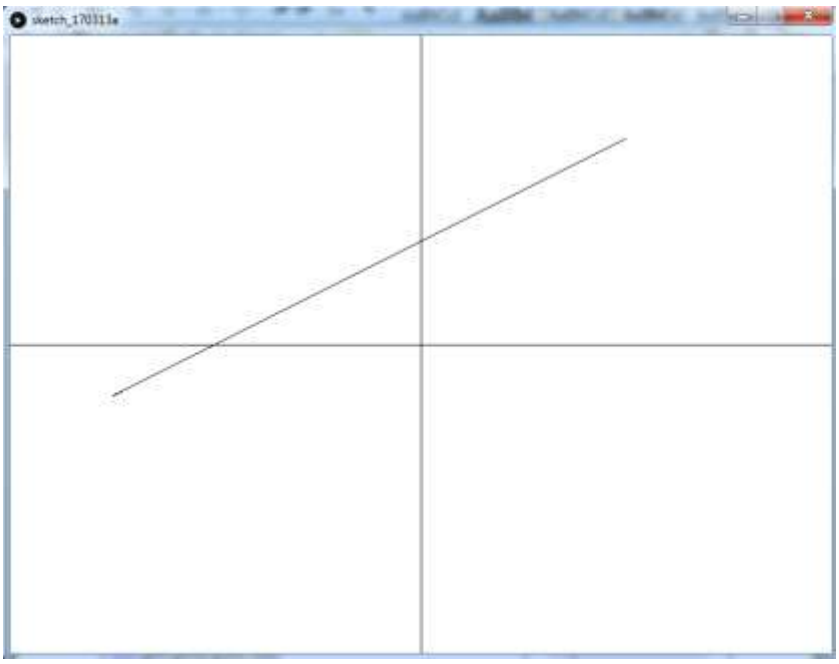
Reprezentarea grafica in programare a functiei de gradul I

<u>*Teorie programare*</u>

width – latime fereastra
height – inaltime fereastra

Aplicatie

```
size(800,600);
background(255);
float x,y,cx=400,cy=300,d=100;
//sistemul de axe
line(0,cy,width,cy);
line(cx,0,cx,height);
//domeniul functiei, D=[-2,3]
for(x=-3;x<=2;x+=0.3/d)
{
  //functia de gradul I
  y=0.5*x+1;
  //graficul functiei in reperul cartezian 2D
  point(x*d+cx,-y*d+cy);
}
```

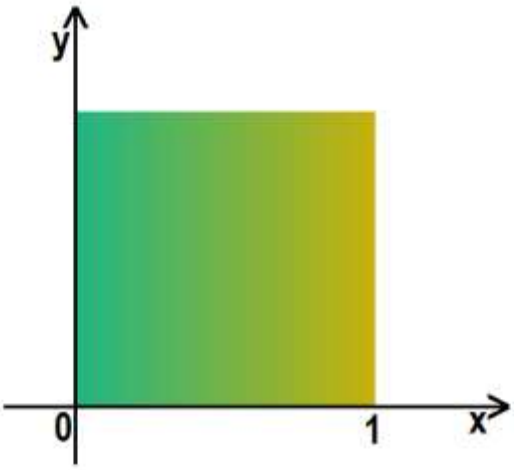


Degradeuri pe baza functiei de gradul I

Pentru obtinerea degradeurilor, este necesar sa gasim relatii intre coordonate si culori. Cea mai simpla metoda de a realiza un degrade intre oricare doua culori, este de a gasi expresii de forma $f(x)=ax+b$ pentru fiecare componenta a culorii, R,G si B. Astfel, trebuie sa determinam trei functii de gradul I, si anume $R(x)=a_1x+b_1$, $G(x)=a_2x+b_2$ si $B(x)=a_3x+b_3$.

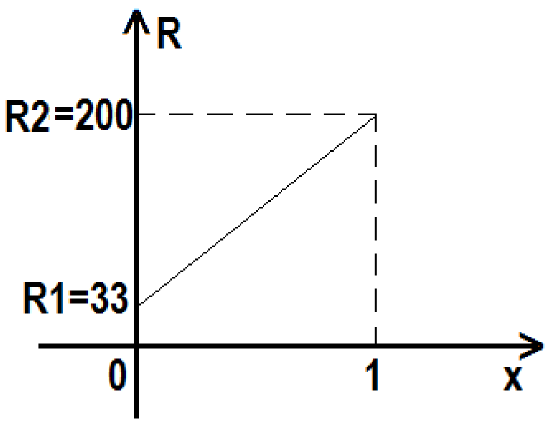
Aplicatie

Pentru inceput, vom studia degradeurile in functie de x. Si pentru a face lucrurile si mai simple, vom alege domeniul de valori D=[0,1].



x	R	G	B	culoare
0	33	182	132	
1	200	177	12	

Rezolvare matematica



x	R
0	33
1	200

$$R(x)=a\cdot x+b$$

$$\left.\begin{array}{l} R(0)=a\cdot 0+b=b \\ R(0)=R1 \end{array}\right| \Rightarrow b=R1$$

$$\left.\begin{array}{l} R(1)=a\cdot 1+b=a+b \\ R(1)=R2 \end{array}\right| \Rightarrow \begin{array}{l} a+b=R2 \\ a=R2-b \\ a=R2-R1 \end{array}$$

$$\boxed{R(x)=(R2-R1)\cdot x+R1}$$

$$\boxed{\begin{array}{l} \text{Red}=(R2-R1)\cdot x+R1 \\ \text{Green}=(G2-G1)\cdot x+G1 \\ \text{Blue}=(B2-B1)\cdot x+B1 \\ 0<x<1 \end{array}}$$

x	R= (R2-R1)·x+R1	G=(G2-G1)·x+G1	B=(B2-B1)·x+B1
0	R1=33	G1=182	B1=132
.....	.....	.....	.....
1	R2=200	G2=177	B2=12

Degradeuri prin interpolare liniara

$$Red = R1\cdot(1-x)+R2\cdot x$$
$$Green = G1\cdot(1-x)+G2\cdot x$$
$$Blue = B1\cdot(1-x)+B2\cdot x$$
$$0<x<1$$

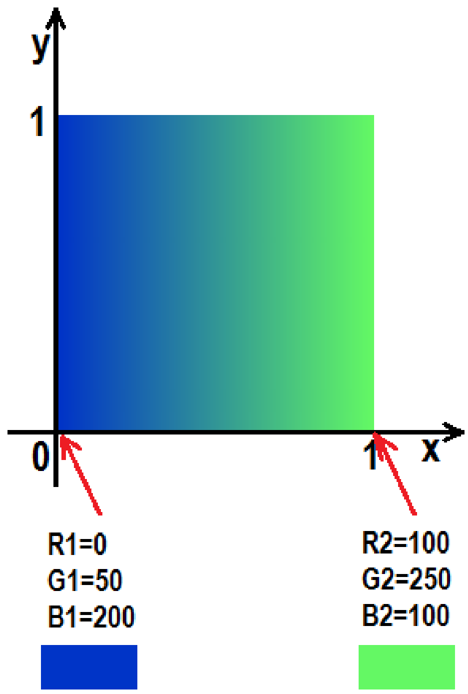
Degradeul prin interpolare liniara este o forma reorganizata a degradeului determinat pe baza functiei de gradul I. Iata mai jos demonstratia.

$$R= (R2-R1)\cdot x+R1 = R2\cdot x-R1\ x+R1 = R1\cdot(1-x)+R2\cdot x$$

$$G= (G2-G1)\cdot x+G1 = G2\cdot x-G1+G1 = G1\cdot(1-x)+G2\cdot x$$

$$B= (B2-B1)\cdot x+B1 = B2\cdot x-B1+B1 = B1\cdot(1-x)+B2\cdot x$$

Aplicatie



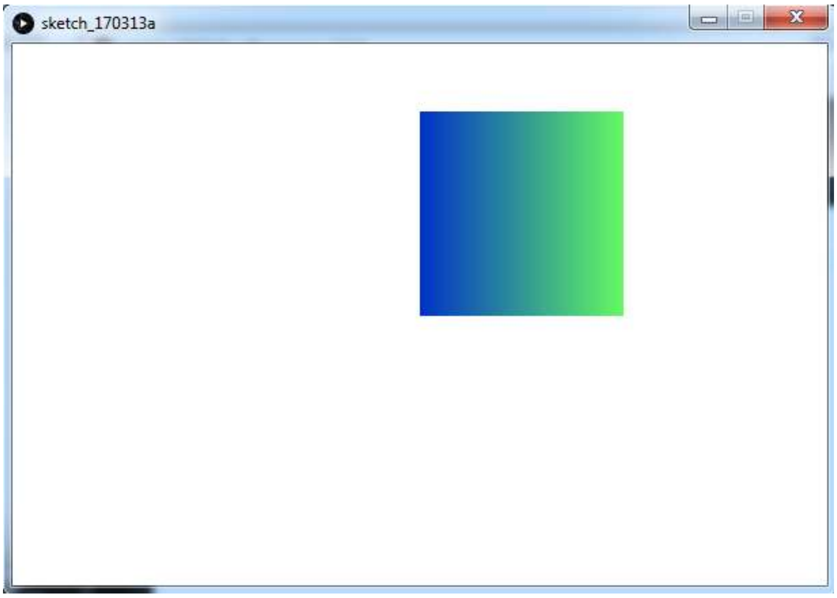
x=0

$$Red = R1\cdot(1-0)+R2\cdot 0 = R1 = 0$$
$$Green = G1\cdot(1-0)+G2\cdot 0 = G1 = 50$$
$$Blue = B1\cdot(1-0)+B2\cdot 0 = B1 = 200$$

x	Red	Green	Blue	culoare
0	0	50	200	
...	......	......	......	
0.5	50	150	150	
...	......	......	......	
1	100	250	100	

Program

```
size(600,400);
background(255);
float x,y,d=150,cx=300,cy=200,
Red,Green,Blue,
R1=0,G1=50,B1=200,R2=100,G2=250,B2=100;
for(x=0;x<=1;x+=0.1/d)
  for(y=0;y<=1;y+=0.1/d)
  {
    Red=R1*(1-x)+R2*x;
    Green=G1*(1-x)+G2*x;
    Blue=B1*(1-x)+B2*x;
    stroke(Red,Green,Blue);
    point(x*d+cx,-y*d+cy);
  }
```



Functii matematice in Processing

Teorie programare

PI – 3.14
abs(x) – modul
pow(x,y) – x la puterea y

sqrt(x) – radical
log(x) – logaritm natural (in baza e, unde e=2.71)
sin(x) – sinus
cos(x) – cosinus
tan(x) – tangenta
asin(x) – arcsinus
acos(x) – arccosinus
atan(x) – arctangenta

Pentru partea intreaga se foloseste conversia la intreg
int(x) – parte intreaga pentru x>0
int(x)+1 – parte intreaga pentru x<0

Pentru partea fractionara se foloseste formula {x}=x-[x]
x-int(x) – parte fractionara

Pentru radical de ordinul n se foloseste formula $x^{1/n}$
pow(x,1.0/n) – radical de ordin n

Pentru logaritm in orice baza se foloseste formula $log_a(b)=ln(b)/ln(a)$
log(b)/log(a) – $log_a(b)$

Pentru cotangenta, se foloseste formula ctg(x)=1/tg(x)
1/tan(x) – cotangenta

Pentru arcctg(x)=pi/2-arctg(x)
PI/2-atan(x)

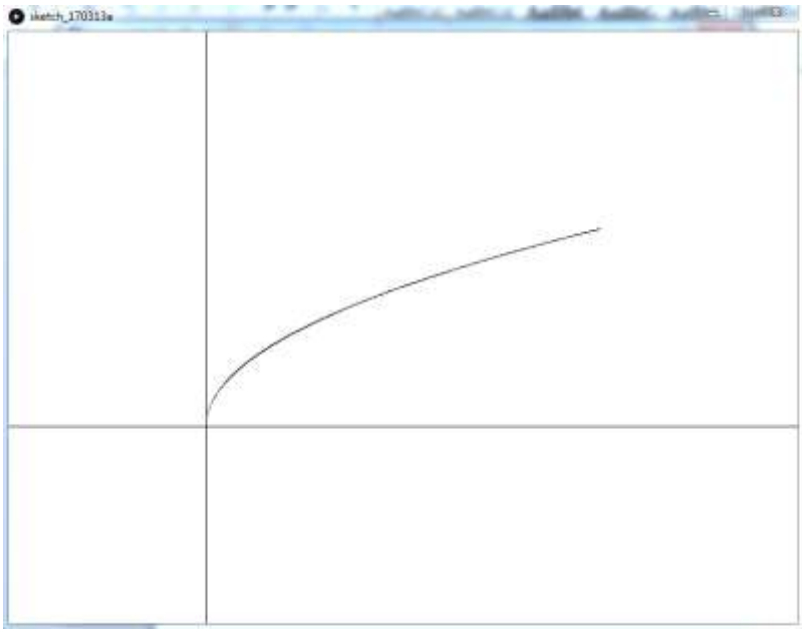
Reprezentarea grafica a functiilor in programare

Aplicatie

Program

```
size(800,600);
background(255);
float x,y,cx=200,cy=400,d=100;
//sistemul de axe
```

```
line(0,cy,width,cy);
line(cx,0,cx,height);
//domeniul functiei, D=[0,4]
for(x=0;x<=4;x+=0.005)
{
  //functia radical
  y=sqrt(x);
  //graficul functiei in reperul cartezian 2D
  point(x*d+cx,-y*d+cy);
}
```

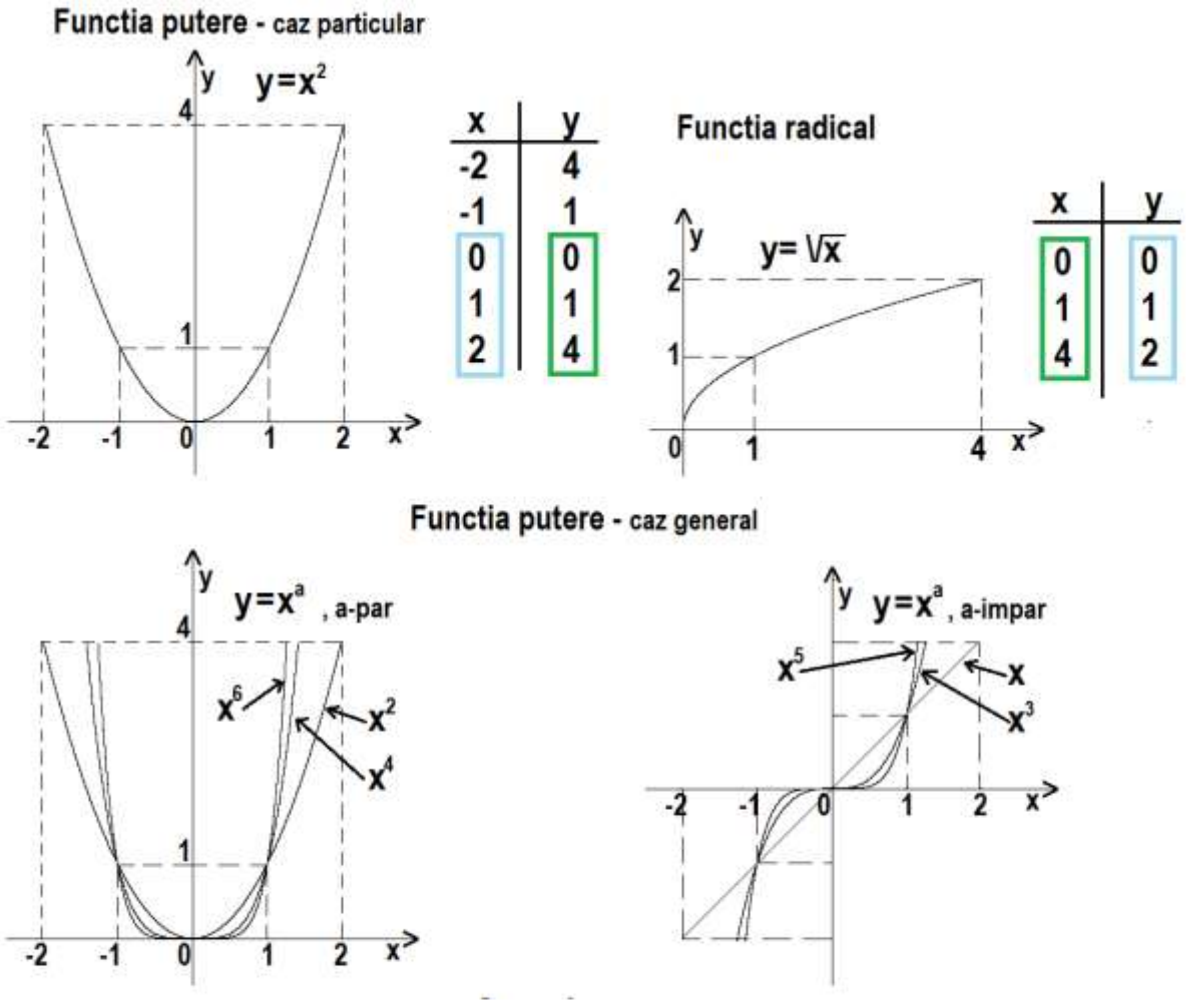


Degradeul prin interpolare neliniara

$$\begin{aligned}
Red &= R1\cdot(1-c)+R2\cdot c\\
Green &= G1\cdot(1-c)+G2\cdot c\\
Blue &= B1\cdot(1-c)+B2\cdot c\\
c&=c(x),\ \ c: D\rightarrow[0,1]
\end{aligned}$$

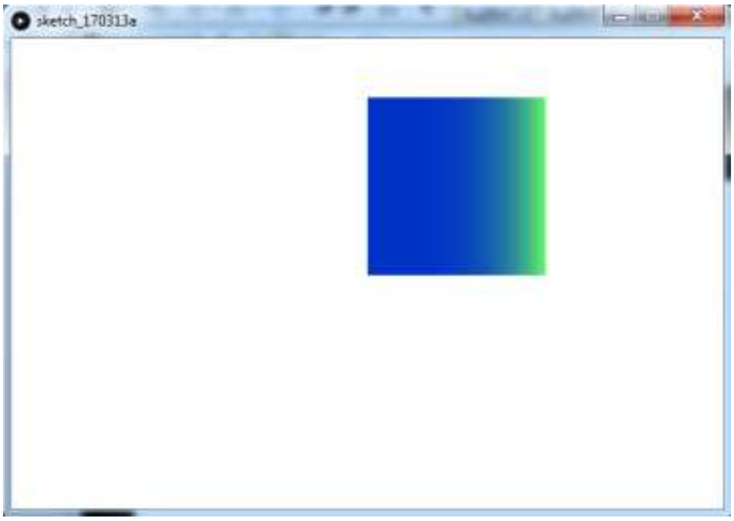
Vom numi:
- c - variabila de interpolare
- c(x) – functie de interpolare

Degradeul prin interpolare neliniara presupune ca variabila de interpolare *c* sa fie o functie neliniara, a carei codomeniu sa fie C=[0,1]. Un exemplu de astfel de functie, este functia putere, pe D=[0,1].

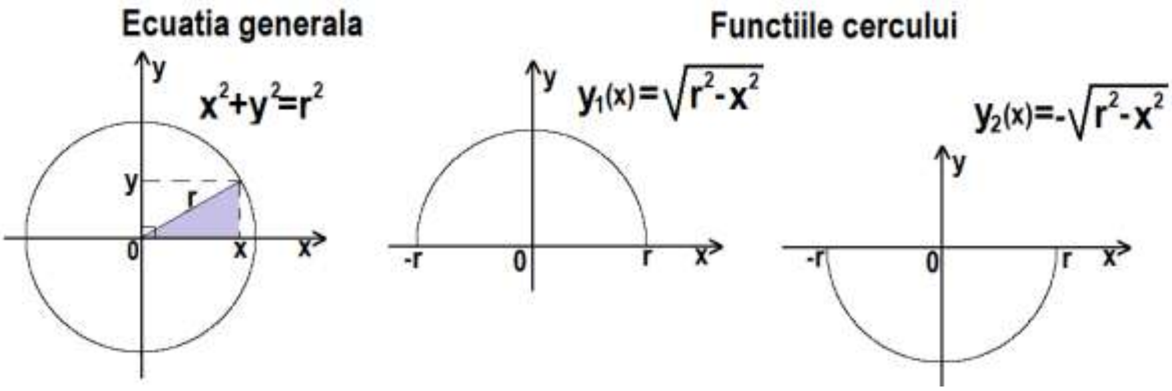


Aplicatie

```
size(600,400);
background(255);
float x,y,d=150,cx=300,cy=200,
Red,Green,Blue,c,
R1=0,G1=50,B1=200,R2=100,G2=250,B2=100;
for(x=0;x<=1;x+=0.1/d)
  for(y=0;y<=1;y+=0.1/d)
  {
    c=pow(x,4);
    Red=R1*(1-c)+R2*c;
    Green=G1*(1-c)+G2*c;
    Blue=B1*(1-c)+B2*c;
    stroke(Red,Green,Blue);
    point(x*d+cx,-y*d+cy);
  }
```



Cercul

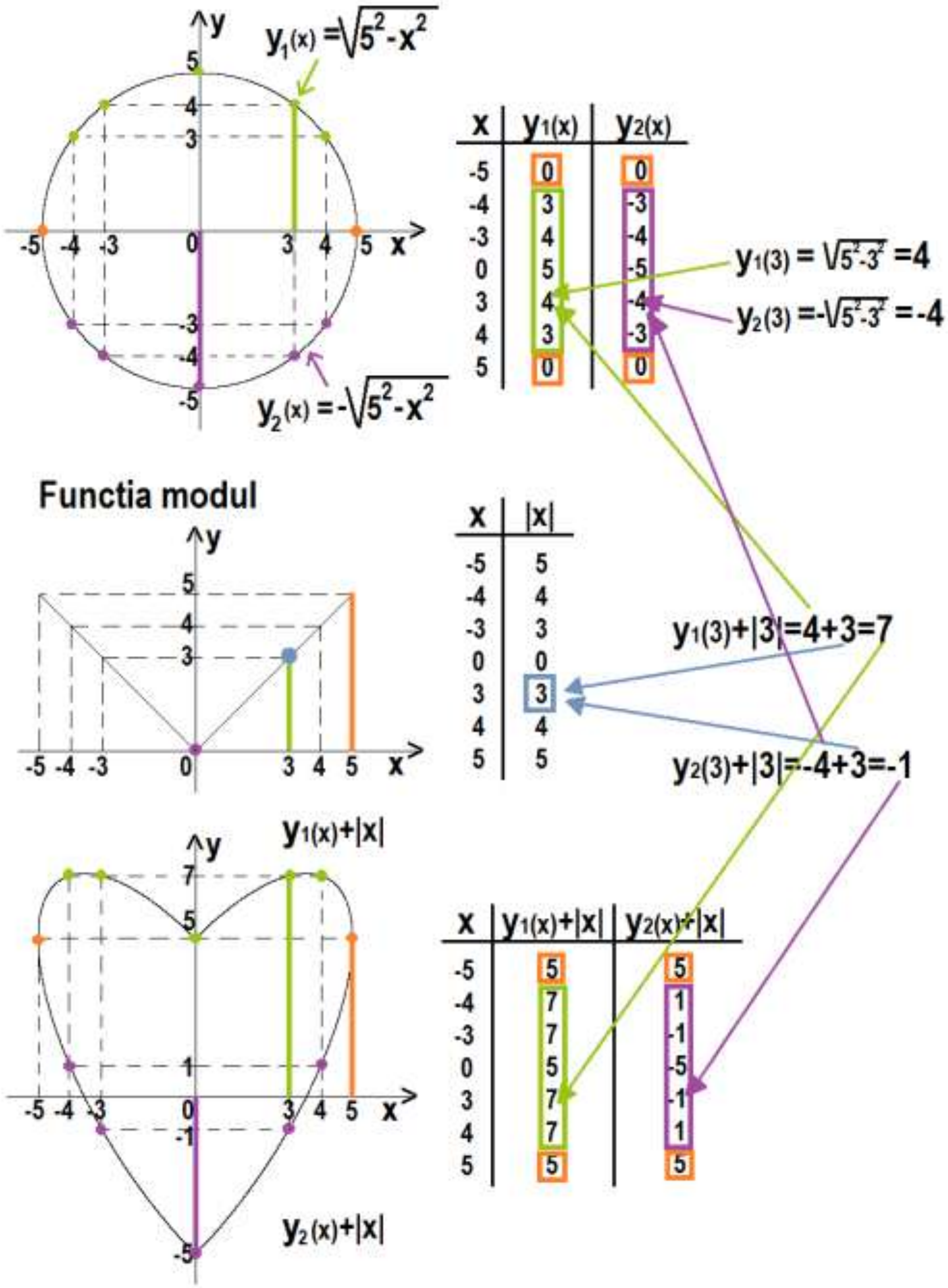


Modelare pe baza functiilor cercului

Aplicatie

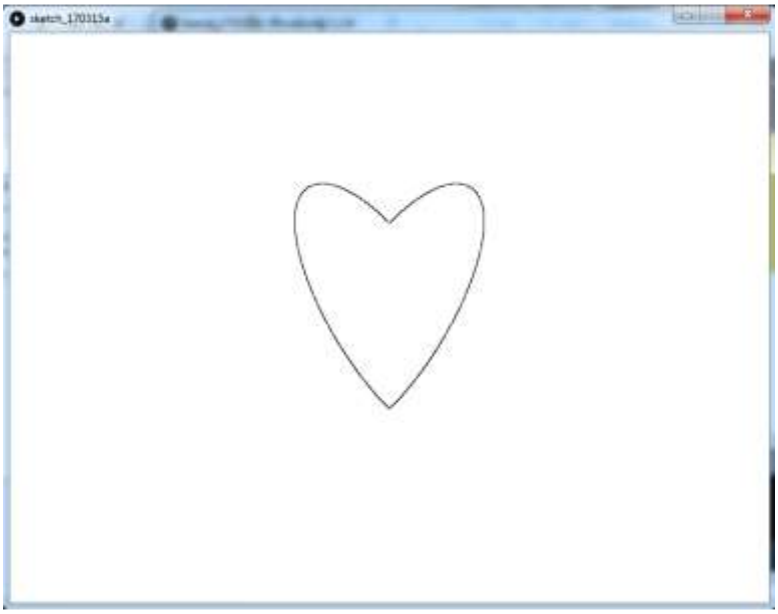
Din functiile cercului vom obtine o inima, adaugind functia modul la fiecare dintre acestea.

Rezolvare matematica



Program

```
size(800,600);
background(255);
float x,y,r=5,cx=400,cy=300,d=20;
for(x=-5;x<=5;x+=0.01/d)
{
  y=sqrt(r*r-x*x)+abs(x);
  point(x*d+cx,-y*d+cy);
  y=-sqrt(r*r-x*x)+abs(x)+0.2;
  point(x*d+cx,-y*d+cy);
}
```



Valoarea 0.2 s-a adaugat pentru a elimina spatiile albe.

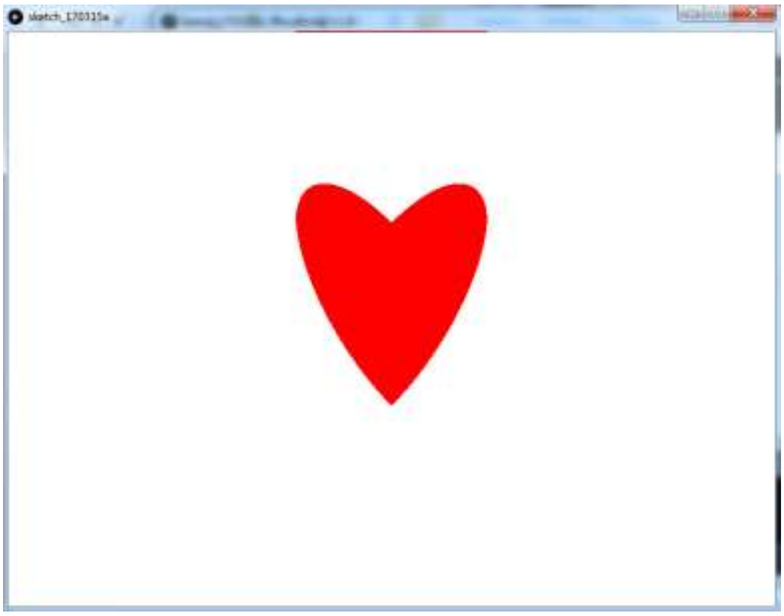
Colorarea suprafetelor delimitate de functii

Aplicatie

Vom colora inima din aplicatia precedenta. Pentru a colora inima, este necesar sa mai adaugam o bucla repetitiva, pentru a varia parametrul *r*. Intrucat inima sa obtinut din cerc, in care r este raza cercului, atunci si dimensiunea inimii se va determina cu ajutorul parametrului *r*.

Program

```
size(800,600);
background(255);
float x,y,r=5,cx=400,cy=300,d=20;
for(r=0;r<=5;r+=0.5/d)
  for(x=-5;x<=5;x+=0.1/d)
  {
     y=sqrt(r*r-x*x)+abs(x);
     set(int(x*d+cx),int(-y*d+cy),color(255,0,0));
     y=-sqrt(r*r-x*x)+abs(x)+0.5;
     set(int(x*d+cx),int(-y*d+cy),color(255,0,0));
  }
```



Colorarea suprafetelor delimitate de functii cu ajutorul degradeurilor prin interpolare liniara

Aplicatie

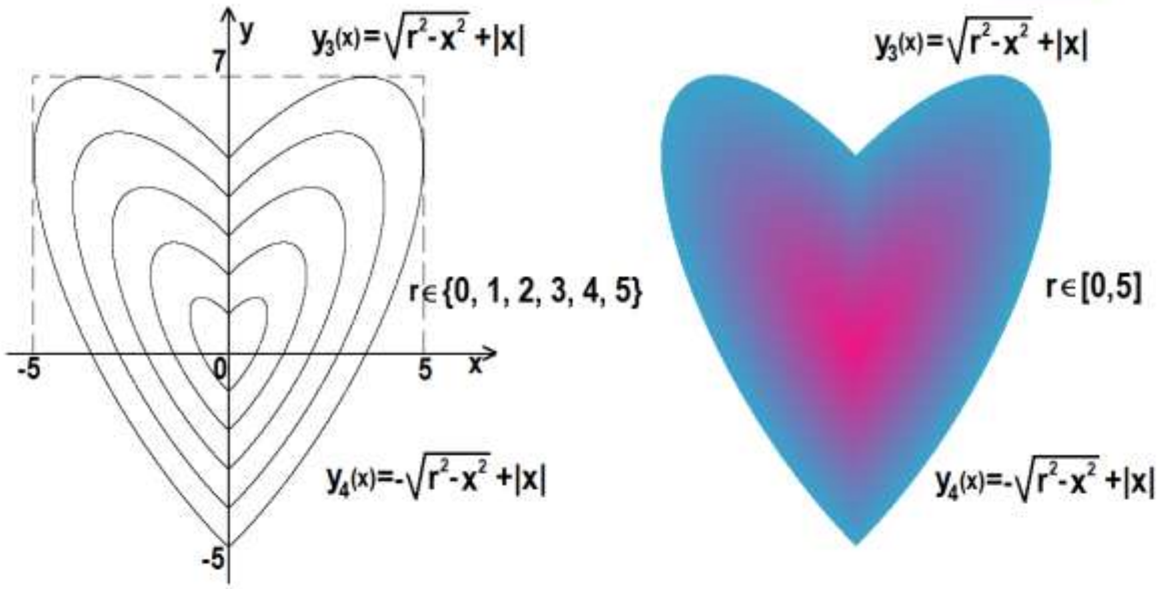
Vom colora inima cu ajutorul unui degrade prin interpolare liniara, astfel incat culoarea sa varieze in functie de dimensiunea inimii.

Rezolvare matematica

Intrucat culoarea variaza in functie de dimensiunea inimii, atunci variabila de interpolare va fi o functie liniara de *r*. Intrucat variabila de interpolare trebuie sa fie pe domeniul [0,1], iar domeniul de valori al lui r este D=[0,5], il vom imparti pe *r* la 5.

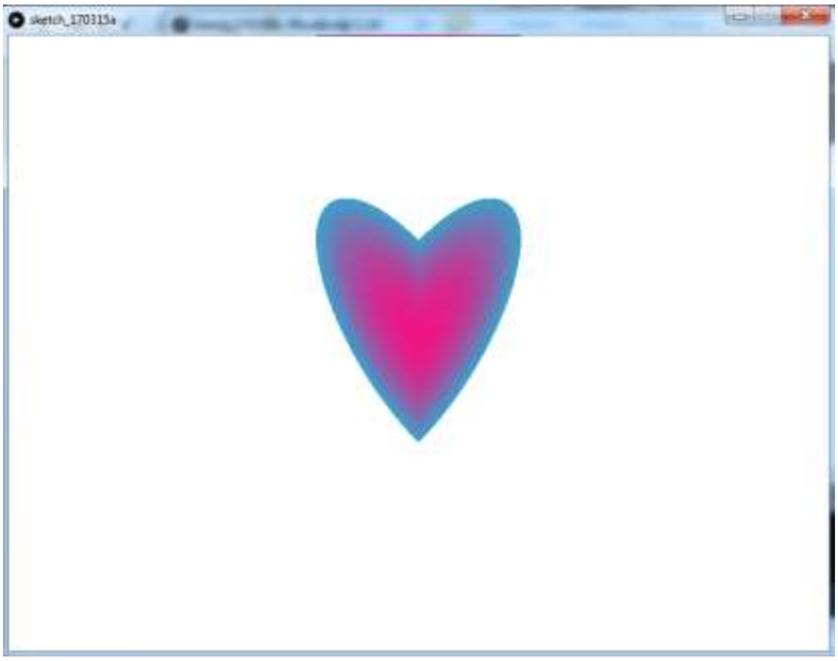
Inima

r	$\frac{r}{5}$	$R=R1\cdot(1-\frac{r}{5})+R2\cdot\frac{r}{5}$	$G=G1\cdot(1-\frac{r}{5})+G2\cdot\frac{r}{5}$	$B=B1\cdot(1-\frac{r}{5})+B2\cdot\frac{r}{5}$	culoare
0	0	R1=240	G1=21	B1=132	
.....	.....	.....	.....	.....	
5	1	R2=56	G2=165	B2=205	



Program

```
size(800,600);
background(255);
float x,y,r,cx=400,cy=300,d=20,
R1=240,G1=21,B1=132,R2=56,G2=165,B2=205,R,G,B,c;
for(r=0;r<=5;r+=0.6/d)
  for(x=-5;x<=5;x+=0.02/d)
  {
    c=pow(r/5,2);
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    y=pow(r*r-x*x,0.5)+abs(x);
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
    y=-pow(r*r-x*x,0.5)+abs(x)+0.2;
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
```



Colorarea suprafetelor delimitate de functii cu ajutorul degradeurilor prin interpolare neliniara

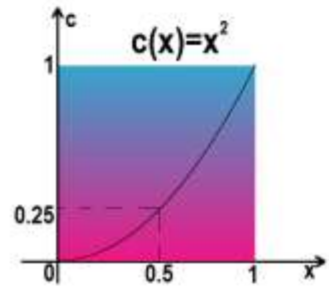
Aplicatie

Vom colora inima cu degradeuri prin interpolare neliniara, pe baza functiei putere. Vom face un studiu pentru diferite functii putere.

Rezolvare matematica

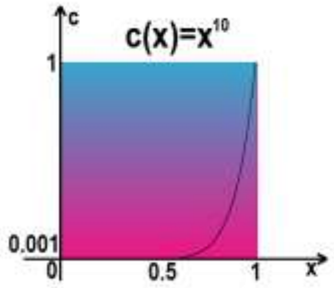
R	G	B	culoare
240	21	132	
56	165	205	

x	c(x)	R=R1·(1-c)+R2·c	G=G1·(1-c)+G2·c	B=B1·(1-c)+B2·c
0	0	R1=240	G1=21	B1=132
......	......	......	......	......
1	1	R2=56	G2=165	B2=205



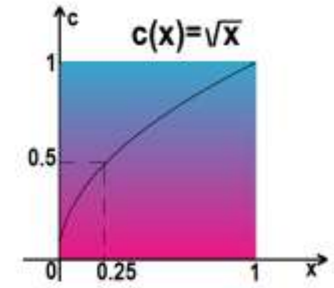
$$0.5^2=\left(\frac{1}{2}\right)^2=\frac{1}{2}=\frac{1}{4}=0.25$$

$$c\left(\frac{r}{5}\right)=\left(\frac{r}{5}\right)^2$$



$$0.5^{10}=\left(\frac{1}{2}\right)^{10}=\frac{1}{2^{10}}=\frac{1}{1024}\simeq 0.001$$

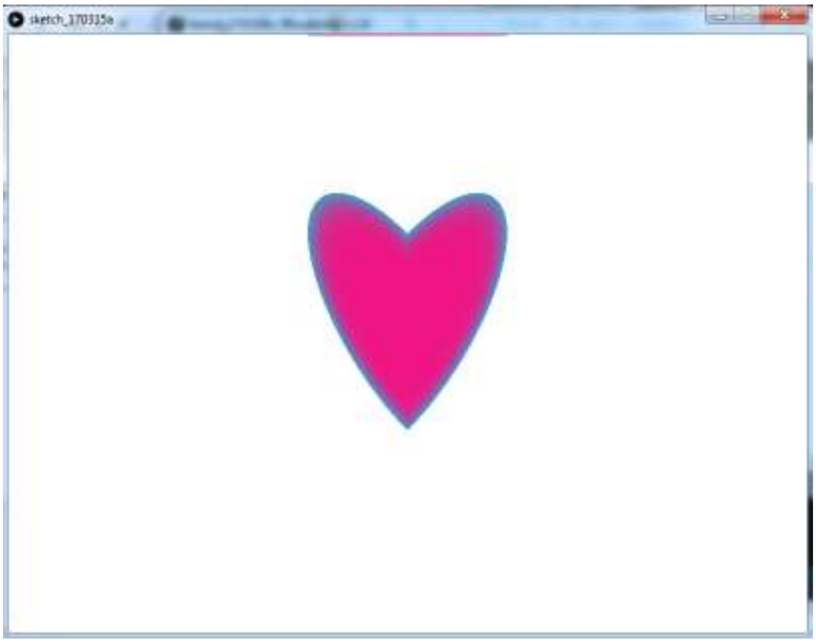
$$c\left(\frac{r}{5}\right)=\left(\frac{r}{5}\right)^{10}$$



$$c\left(\frac{r}{5}\right)=\sqrt{\frac{r}{5}}$$

Program

```
size(800,600);
background(255);
float x,y,r,cx=400,cy=300,d=20,
R1=240,G1=21,B1=132,R2=56,G2=165,B2=205,R,G,B,c;
for(r=0;r<=5;r+=0.6/d)
  for(x=-5;x<=5;x+=0.02/d)
  {
    c=pow(r/5,10);
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    y=pow(r*r-x*x,0.5)+abs(x);
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
    y=-pow(r*r-x*x,0.5)+abs(x)+0.2;
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
```



Delimitarea suprafetelor cu ajutorul inecuatiilor

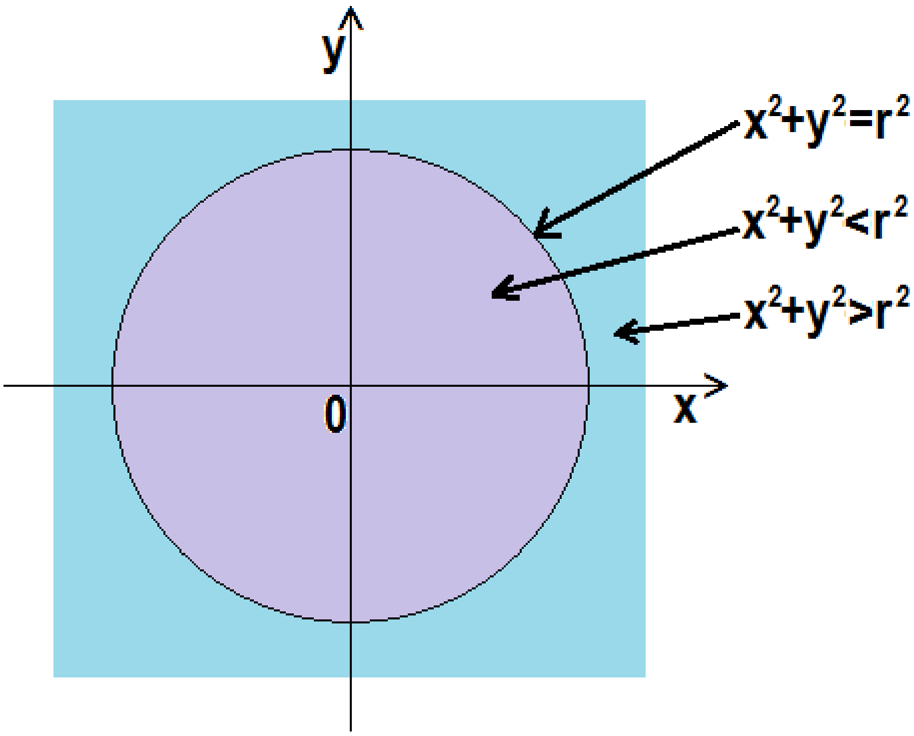
Teorie matematica

O ecuatie de 2 variabile poate fi scrisa ca o expresie egala cu 0, in felul acesta:

E(x,y)=0.

O ecuatie de 2 variabile este o curba inchisa sau deschisa care imparte planul XOY in 2 zone. Semnele expresiei din cele 2 zone sunt de multe ori diferite. Astfel, putem selecta una dintre zone impunind conditia:

E(x,y)<0 sau E(x,y)>0.



Exemple de expresii care au acelas semn in ambele zone sunt: -|E(x,y)|, $(E(x,y))^2$. Semnul expresiilor este un domeniu laborios si nu face subiectul cartii sa-l discutam in detaliu.

Teorie programare

Instructiunile conditionale determina programele sa testeze diferite conditii si in functie de acestea sa decida executia anumitor comenzi.

- **if()** - executa comenzile dorite cand o conditie scrisa in paranteze este adevarata.
- **if() ... else** - executa anumite comenzi cand o conditie este adevarata si alte comenzi cand aceasta este falsa.

Definitii:

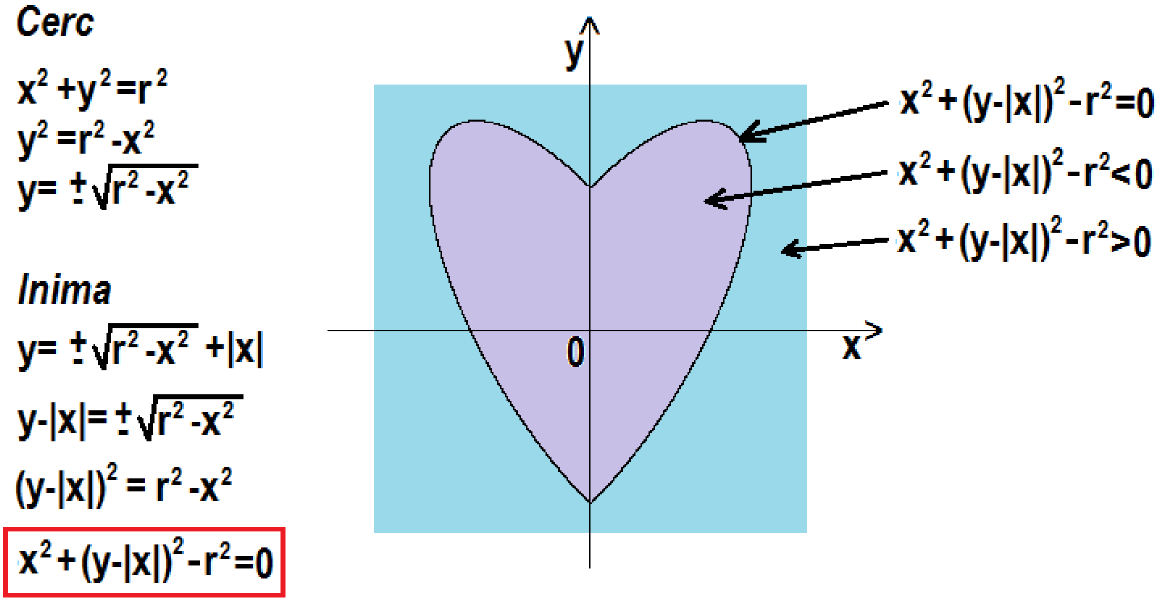
```
if (conditie) {
   // Codul care va fi executat daca este Adevarata conditia
}

if (conditie) {
   // codul care va fi executat daca este Adevarata conditia
}
else {
   // codul ce va fi executat daca conditia este falsa
}
```

Aplicatie

Vom colora interiorul unei inimi cu o culoare iar exteriorul cu alta culoare.

Rezolvare matematica

Vom determina ecuatia generala a inimii, pornind de la ecuatia cercului.



Intrucat ecuatia are cele mai multe solutii (x,y) pentru valori irationale ale lui x si y, iar in programare nu pot fi reprezentate numere irationale, conturul inimii nu poate fi desenat separat prin metoda de fata ci doar cu ajutorul functiilor, asa cum s-a aratat in capitolul „Modelarea pe baza functiilor cercului”. Din acest motiv, vom grupa ecuatia cu una dintre inecuatii.

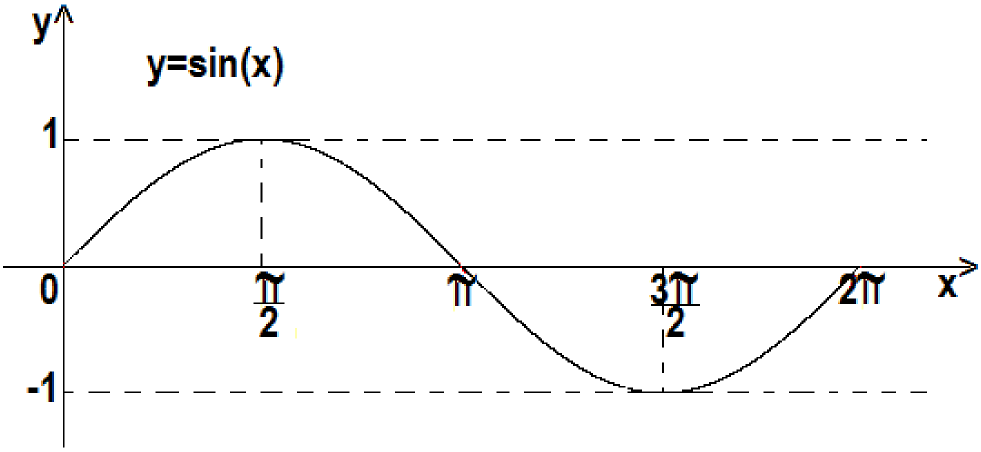
Program

```
size(600,400);
background(255);
float x,y,r=1,d=100,cx=300,cy=200;
for(x=-2;x<=2;x+=0.8/d)
  for(y=-2;y<=2;y+=0.8/d)
    if(x*x+pow(y-abs(x),2)-r*r<=0)
      set(int(x*d+cx),int(-y*d+cy),color(250,50,150));
    else
      set(int(x*d+cx),int(-y*d+cy),color(50,155,255));
```

Operatii si compuneri de functii utilizate in modelarea matematica

Vom alege o functie esantion. Vom studia care este efectul operatiilor si compunerilor cu anumite functii, asupra graficului functiei esantion.

Fie f(x)=sin(x) – funtia esantion

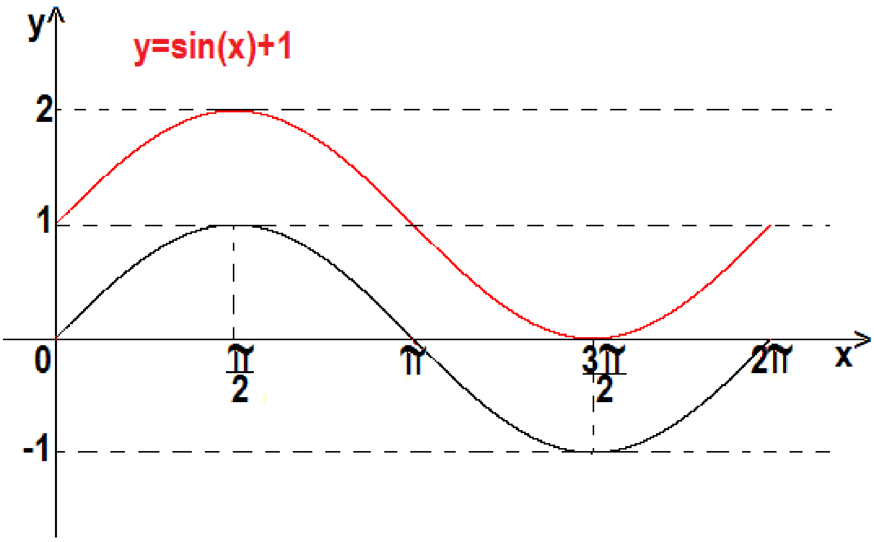


x	sin(x)
0	0
$\frac{\pi}{2}$	1
π	0
$\frac{3\pi}{2}$	-1
2π	0

Decalarea pe axa OY

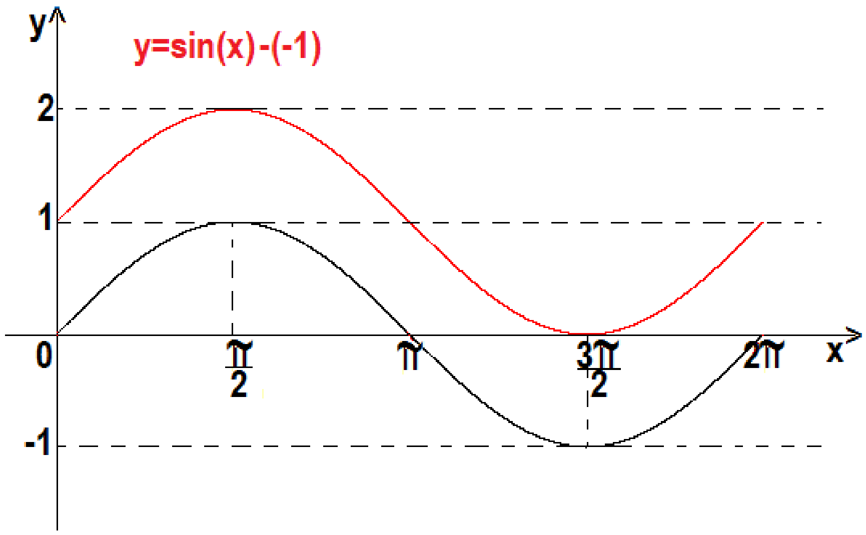
Se realizeaza cu ajutorul adunarii/scaderii functiei cu o constanta.

Adunarea functiei cu o constanta *a*-real are ca efect decalarea graficului functiei pe axa OY, cu valoarea constantei. (in sensul axei pt a>0 si in sens contrar pt a<0)



x	sin(x)	sin(x)+1
0	0	1
$\frac{\pi}{2}$	1	2
π	0	1
$\frac{3\pi}{2}$	-1	0
2π	0	1

Scaderea functiei cu o constanta *a*-real are ca efect decalarea graficului functiei pe axa OY, cu valoarea opusa constantei, -*a*. (in sensul axei pt a<0 si in sens contrar pt a>0)

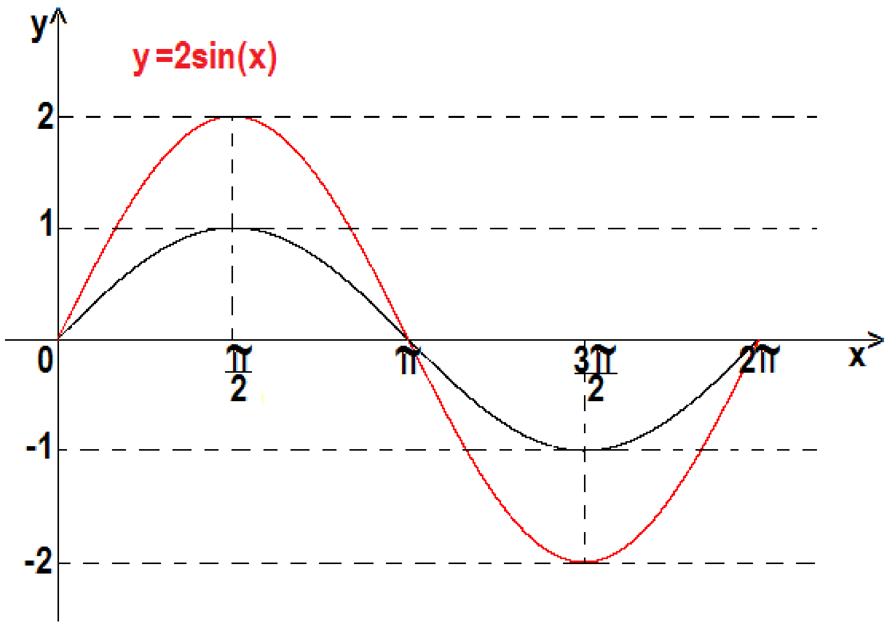


x	sin(x)	sin(x)-(-1)
0	0	1
$\frac{\pi}{2}$	1	2
π	0	1
$\frac{3\pi}{2}$	-1	0
2π	0	1

Decomprimarea/comprimarea pe axa OY

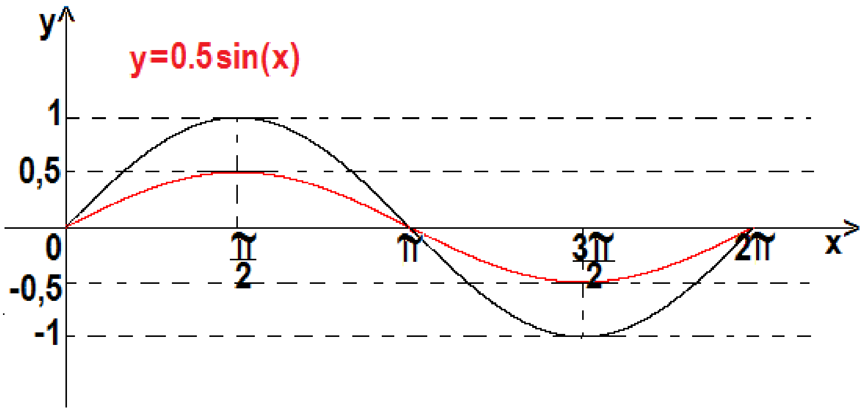
Se realizeaza cu ajutorul inmultirii/impartirii functiei cu o constanta.

Inmultirea functiei cu o constanta $a>1$ are ca efect decomprimarea graficului functiei pe axa OY, de atatea ori cat e valoarea constantei a.



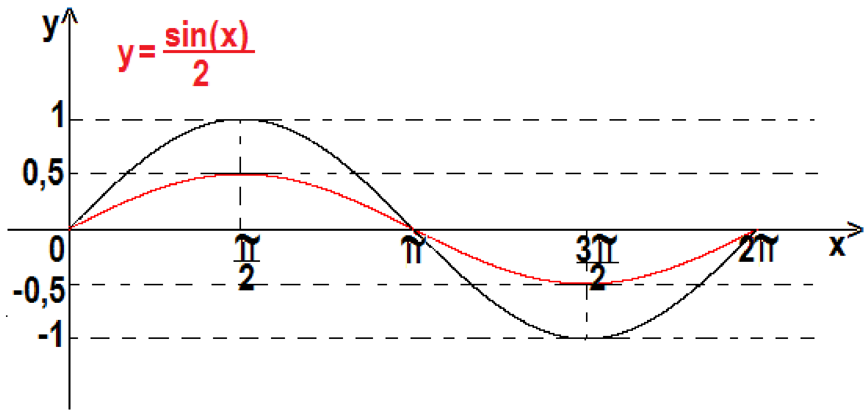
x	sin(x)	2sin(x)
0	0	0
$\frac{\pi}{2}$	1	2
π	0	0
$\frac{3\pi}{2}$	-1	-2
2π	0	0

Inmultirea functiei cu o constanta $0<a<1$ are ca efect comprimarea graficului functiei pe axa OY, de atatea ori cat e valoarea rasturnata a constantei, $1/a$.
Ex: 0.5=1/2



x	sin(x)	0,5sin(x)
0	0	0
$\frac{\pi}{2}$	1	0,5
π	0	0
$\frac{3\pi}{2}$	-1	-0,5
2π	0	0

Impartirea functiei la o constanta a>1 are ca efect comprimarea graficului functiei pe axa OY, de atatea ori cat e valoarea constantei a.

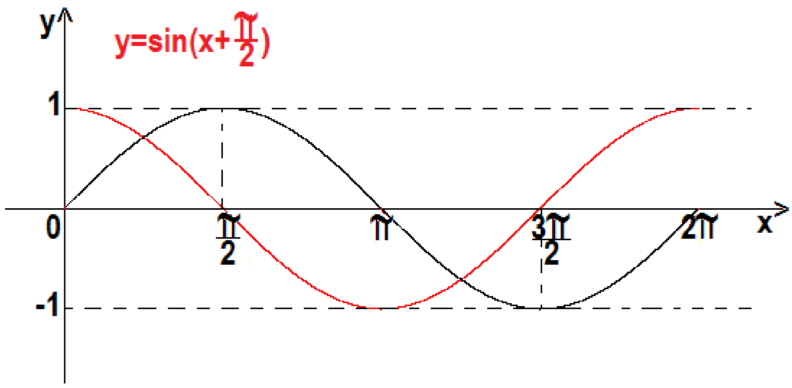


x	sin(x)	$\frac{\sin(x)}{2}$
0	0	0
$\frac{\pi}{2}$	1	0,5
π	0	0
$\frac{3\pi}{2}$	-1	-0,5
2π	0	0

Decalarea pe axa OX

Se realizeaza cu ajutorul adunarii/scaderii argumentului functiei cu o constanta.

Adunarea argumentului functiei cu o constanta *a*-real are ca efect decalarea graficului functiei pe axa OX, cu valoarea opusa constantei, -*a*. (in sensul axei pt a<0 si in sens contrar pt a>0)

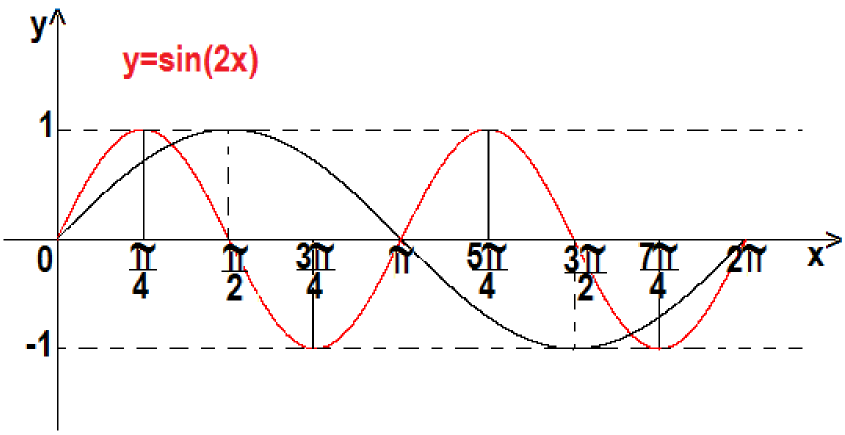


x	sin(x)	$x+\frac{\pi}{2}$	$\sin(x+\frac{\pi}{2})$
0	0	$\frac{\pi}{2}$	1
$\frac{\pi}{2}$	1	π	0
π	0	$\frac{3\pi}{2}$	-1
$\frac{3\pi}{2}$	-1	2π	0
2π	0	$\frac{5\pi}{2}$	1

Decomprimarea/comprimarea pe axa OX

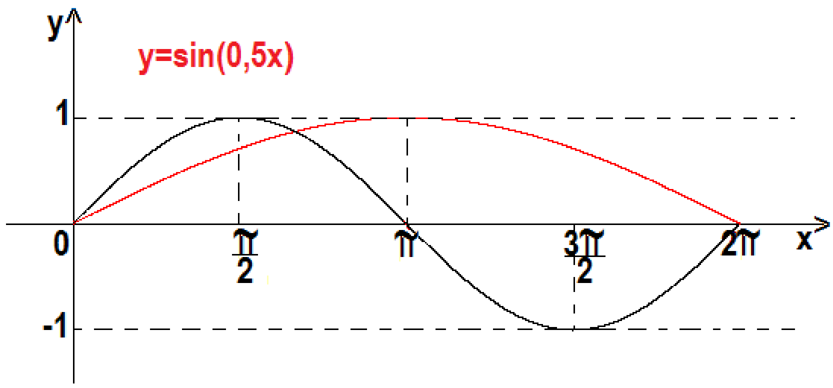
Se realizeaza prin inmultirea/impartirea argumentului functiei cu o constanta.

Inmultirea argumentului functiei cu o constanta $a>1$ are ca efect comprimarea graficului functiei pe axa OX, de atateaori cat e valoarea constantei a.



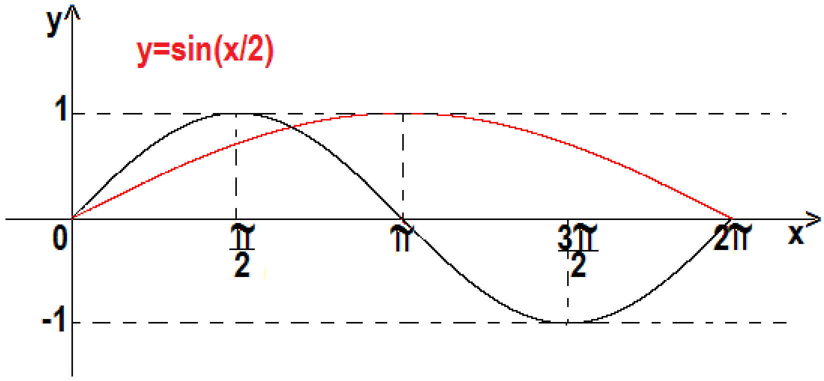
x	sin(x)	2x	sin(2x)
0	0	0	0
$\frac{\pi}{4}$		$\frac{\pi}{2}$	1
$\frac{\pi}{2}$	1	π	0
$\frac{3\pi}{4}$		$\frac{3\pi}{2}$	-1
π	0	2π	0
$\frac{5\pi}{4}$			
$\frac{3\pi}{2}$	-1		
$\frac{7\pi}{4}$			
2π	0		

Inmultirea argumentului functiei la o constanta $0<a<1$ are ca efect decomprimarea graficului functiei pe axa OX, de atateaori cat e valoarea rasturnata a constantei, $1/a$.



x	sin(x)	0,5x	sin(0,5x)
0	0	0	0
$\frac{\pi}{2}$	1		
π	0	$\frac{\pi}{2}$	1
$\frac{3\pi}{2}$	-1		
2π	0	π	0

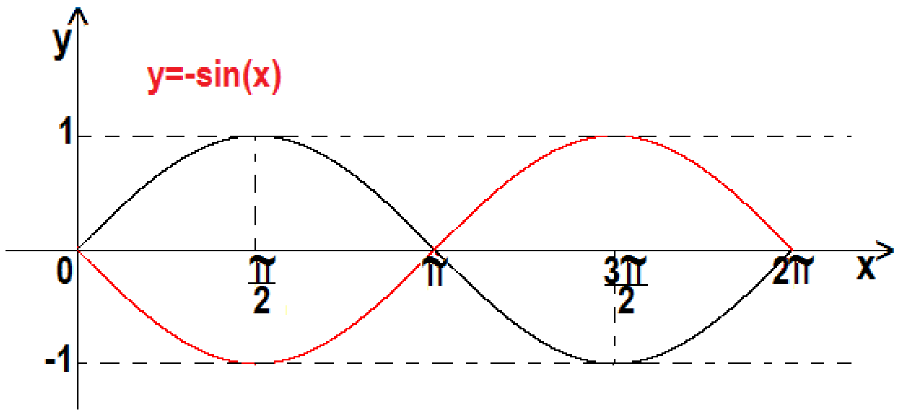
Impartirea argumentului functiei la o constanta $a>1$ are ca efect decomprimarea graficului functiei pe axa OX, de atateaori cat e valoarea constantei a.



x	sin(x)	x/2	sin(x/2)
0	0	0	0
$\frac{\pi}{2}$	1		
π	0	$\frac{\pi}{2}$	1
$\frac{3\pi}{2}$	-1		
2π	0	π	0

Inversarea pe axa OY

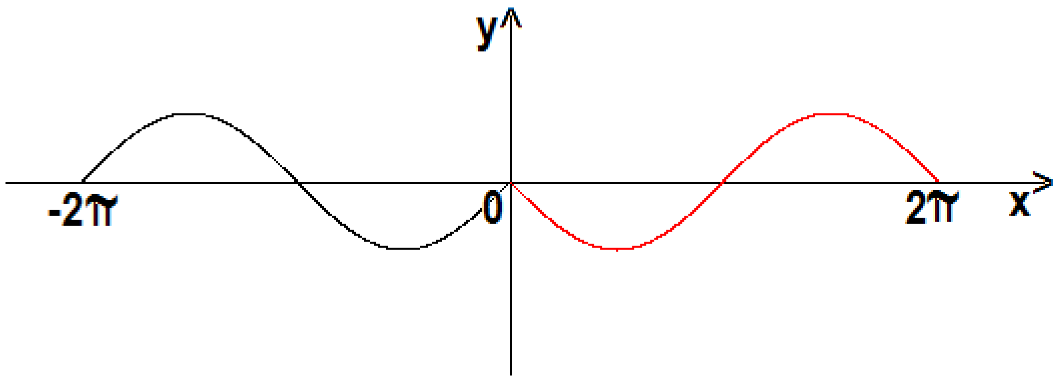
Se realizeaza cu ajutorul inmultirii functiei cu -1.



x	sin(x)	-sin(x)
0	0	0
$\frac{\pi}{2}$	1	-1
π	0	0
$\frac{3\pi}{2}$	-1	1
2π	0	0

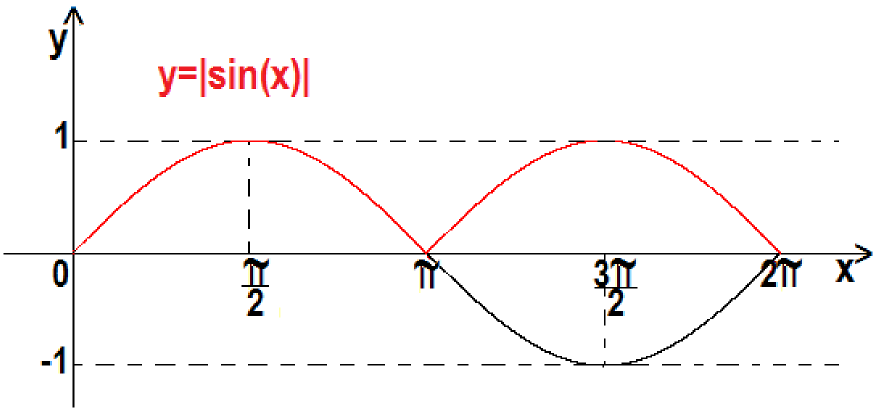
Inversarea pe axa OX

Se realizeaza cu ajutorul inmultirii argumentului functiei cu -1.



Plierea pe axa OY

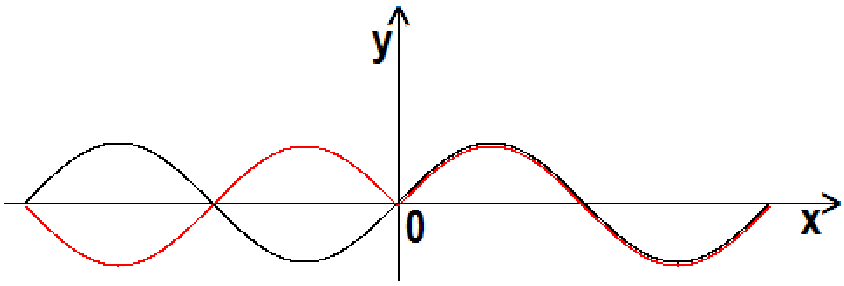
Se realizeaza cu ajutorul modulului, aplicat functiei.



x	sin(x)	|sin(x)|
0	0	0
$\frac{\pi}{2}$	1	1
π	0	0
$\frac{3\pi}{2}$	-1	1
2π	0	0

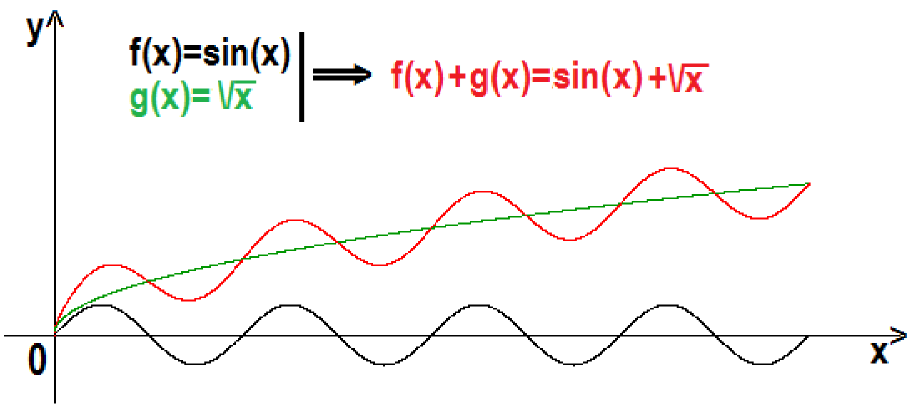
Deplierea pe axa OX

Se realizeaza cu ajutorul modulului, aplicat argumentului functiei.

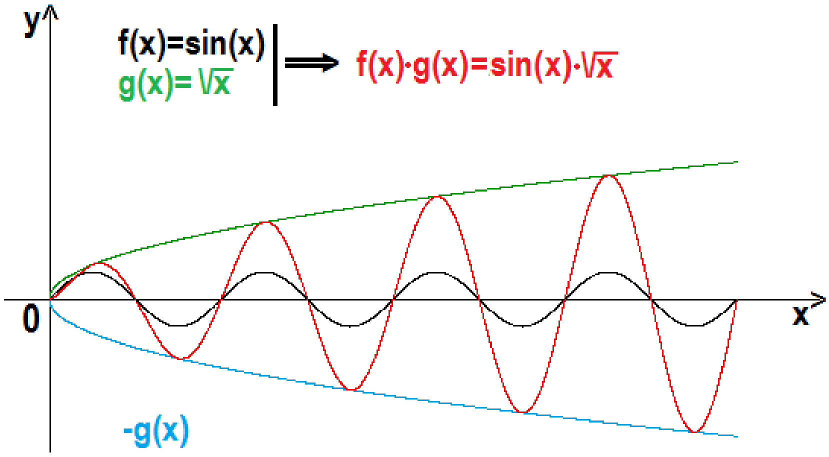


Alte metode de modelarea pe verticala (axa OY)

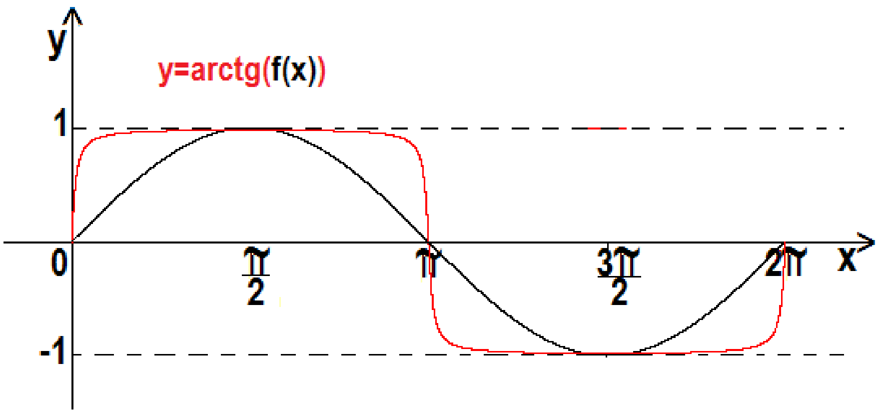
-prin adunarea functiilor.



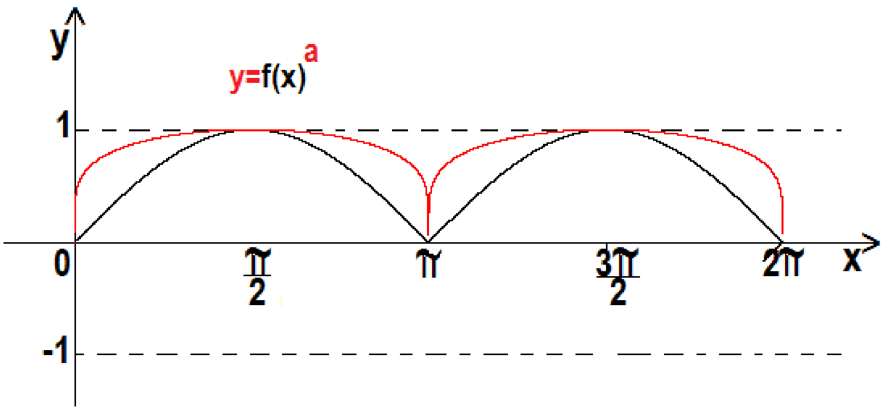
-prin inmultirea functiilor.



-cu ajutorul functiei arctangenta

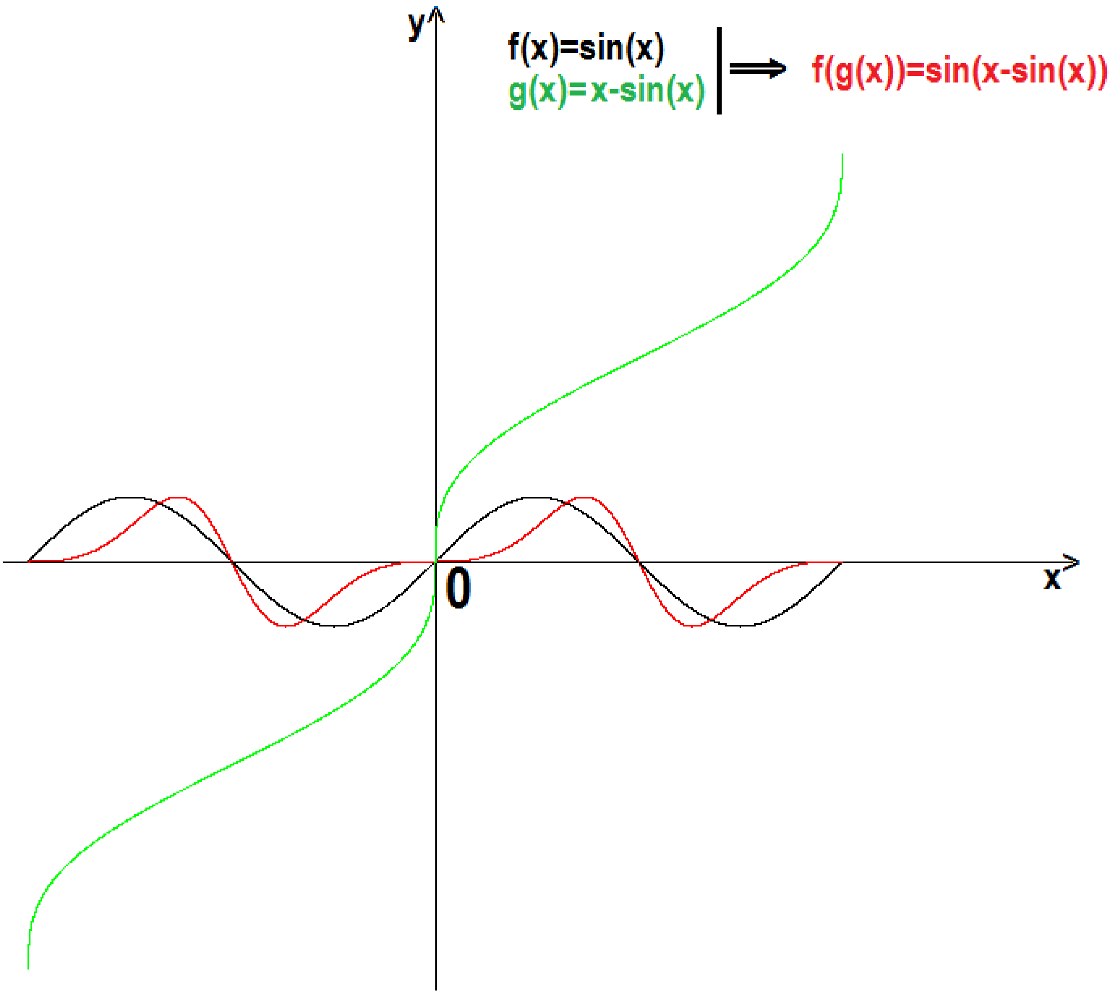


-prin ridicarea la putere subunitara (doar pentru f(x)>=0)



Alte metode de modelarea pe orizontala (axa OX)

-prin compunere de functii

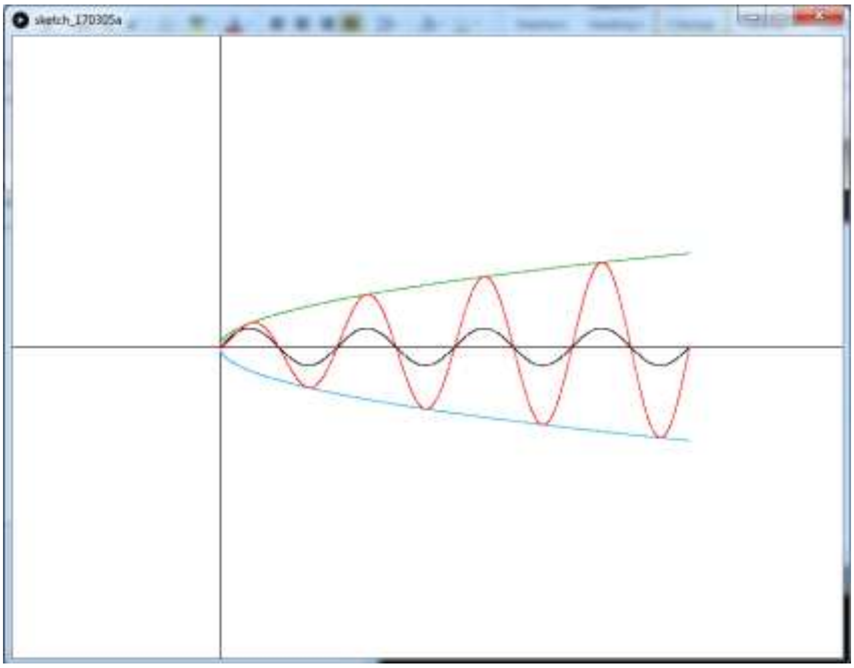


Aplicatie

Program

```
size(800,600);
background(255);
float x,y,cx=200,cy=300,d=18;
line(0,cy,800,cy);
line(cx,0,cx,600);
for(x=0;x<=8*PI;x+=0.005)
{
  y=sin(x);
  set(int(x*d+cx),int(-y*d+cy),color(0,0,0));
  y=sqrt(x);
  set(int(x*d+cx),int(-y*d+cy),color(0,150,0));
```

```
 y=-sqrt(x);
 set(int(x*d+cx),int(-y*d+cy),color(0,160,230));
 y=sqrt(x)*sin(x);
 set(int(x*d+cx),int(-y*d+cy),color(255,0,0));
}
```

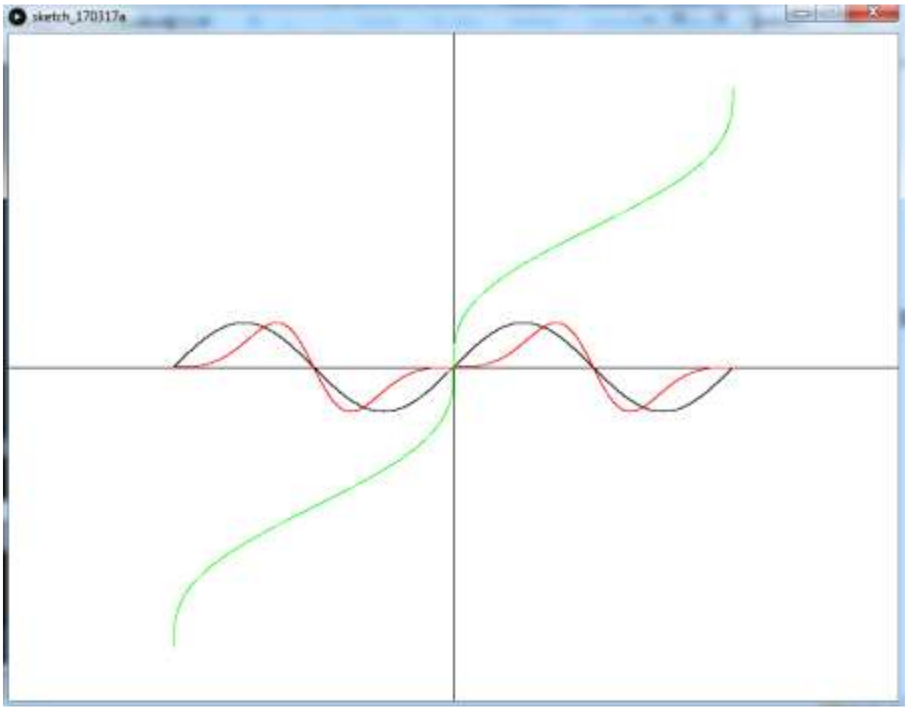


Aplicatie

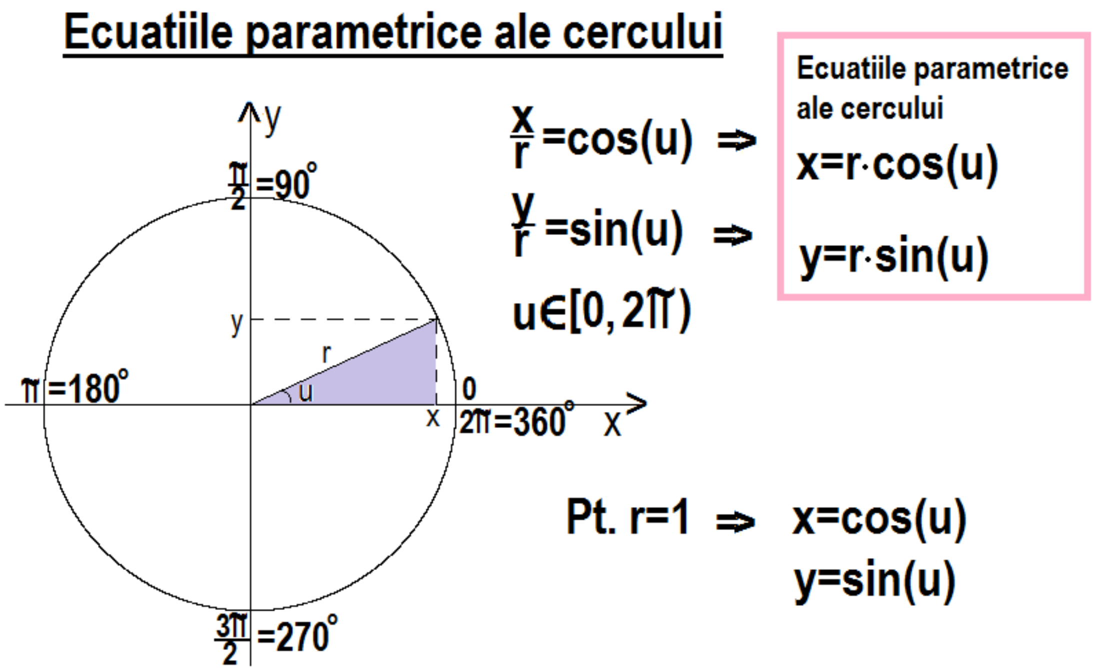
Program

```
size(800,600);
background(255);
float x,y,cx=400,cy=300,d=40;
line(0,cy,800,cy);
line(cx,0,cx,600);
for(x=-2*PI;x<=2*PI;x+=0.0005)
{
  y=sin(x);
  set(int(x*d+cx),int(-y*d+cy),color(0,0,0));
  y=sin(x-sin(x));
  set(int(x*d+cx),int(-y*d+cy),color(255,0,0));
}
for(y=-2*PI;y<=2*PI;y+=0.005)
{
  x=y-sin(y);
```

```
  set(int(x*d+cx),int(-y*d+cy),color(0,255,0));
}
```

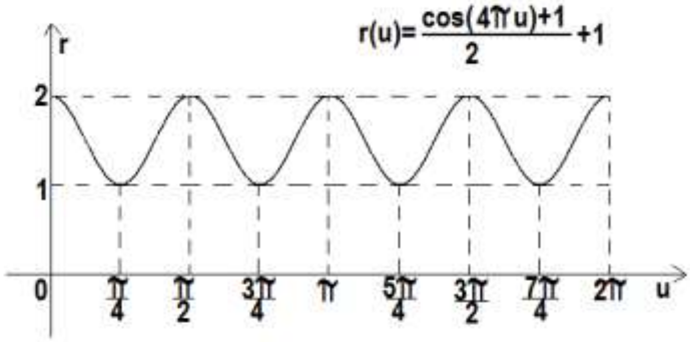


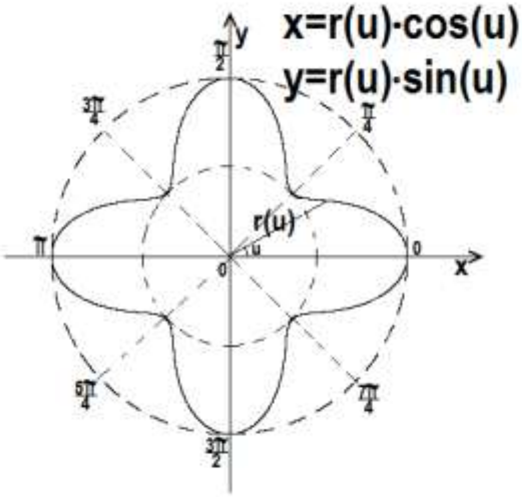
Cercul



Modelare pe baza ecuatiilor parametrice ale cercului

Modelarea formelor pe baza ecuatiilor parametrice ale cercului se face variind raza in functie de u.
Modelarea formelor simetrice se face cu ajutorul functiilor periodice.

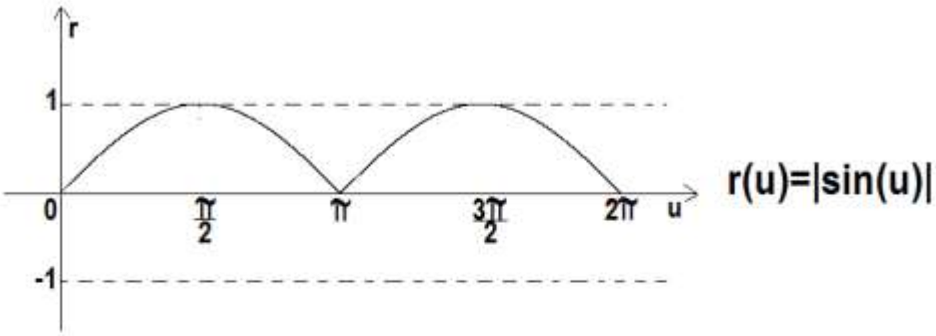


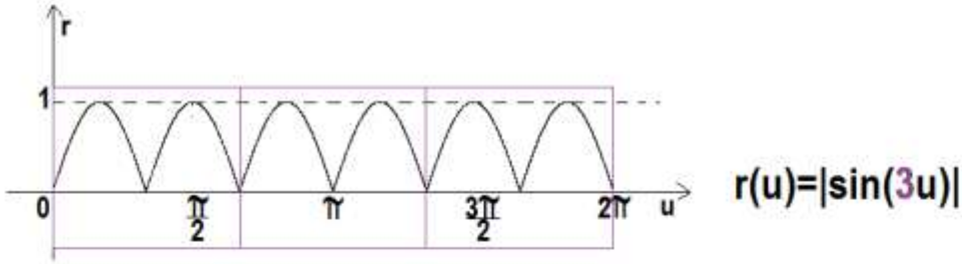


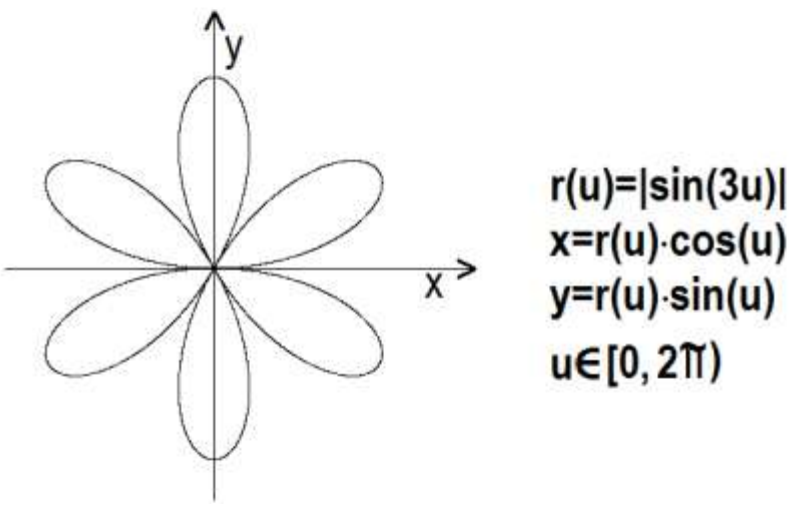
O categorie a formelor simetrice si usor de modelat sunt florile. Vom incepe prin a determina forma de baza a florii.

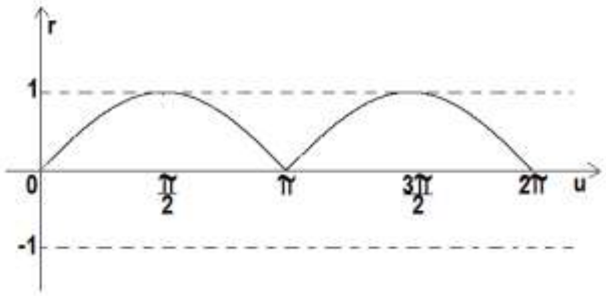
Floarea – forma de baza

O metoda simpla de a obtine forma unei flori, este sa pornim de la graficul functiei r(u)=|sin(u)|. Intrucat graficul functiei are 2 „petale” intr-o perioada, pentru a obtine un anumit numar de petale, vom comprima graficul functiei pe axa OX, prin inmultirea argumentului cu jumatatea numarului de petale.

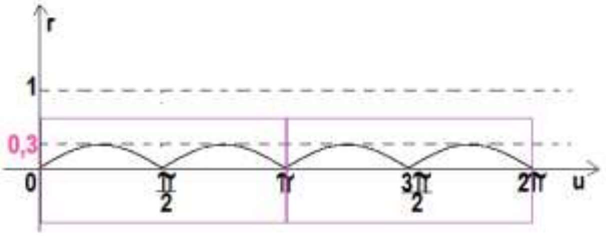

r
1
0
-1
π/2
π
3π/2
2π
u
r(u)=|sin(u)|


r
1
0
π/2
π
3π/2
2π
u
r(u)=|sin(3u)|

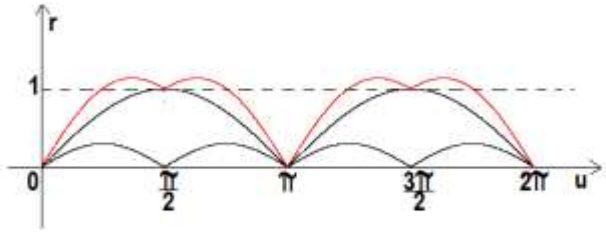

y
x
r(u)=|sin(3u)|
x=r(u)·cos(u)
y=r(u)·sin(u)
u∈[0, 2π)



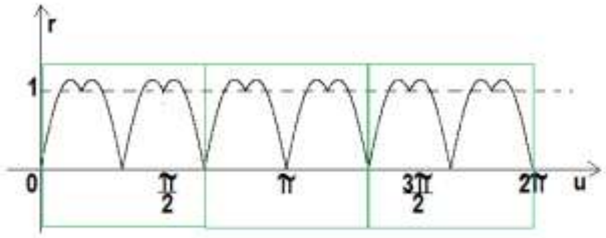
$$r(u)=|\sin(u)|$$



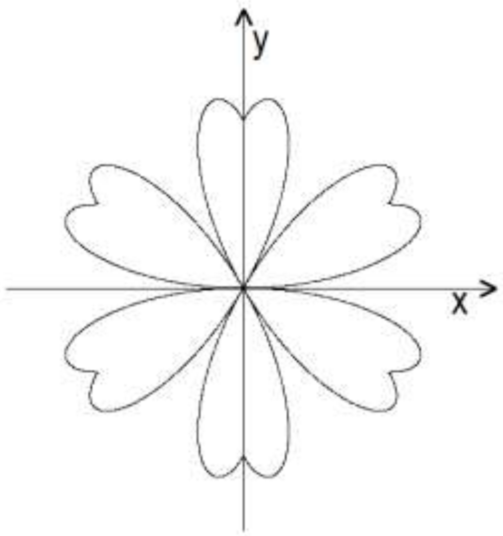
$$r(u)=0,3|\sin(2u)|$$



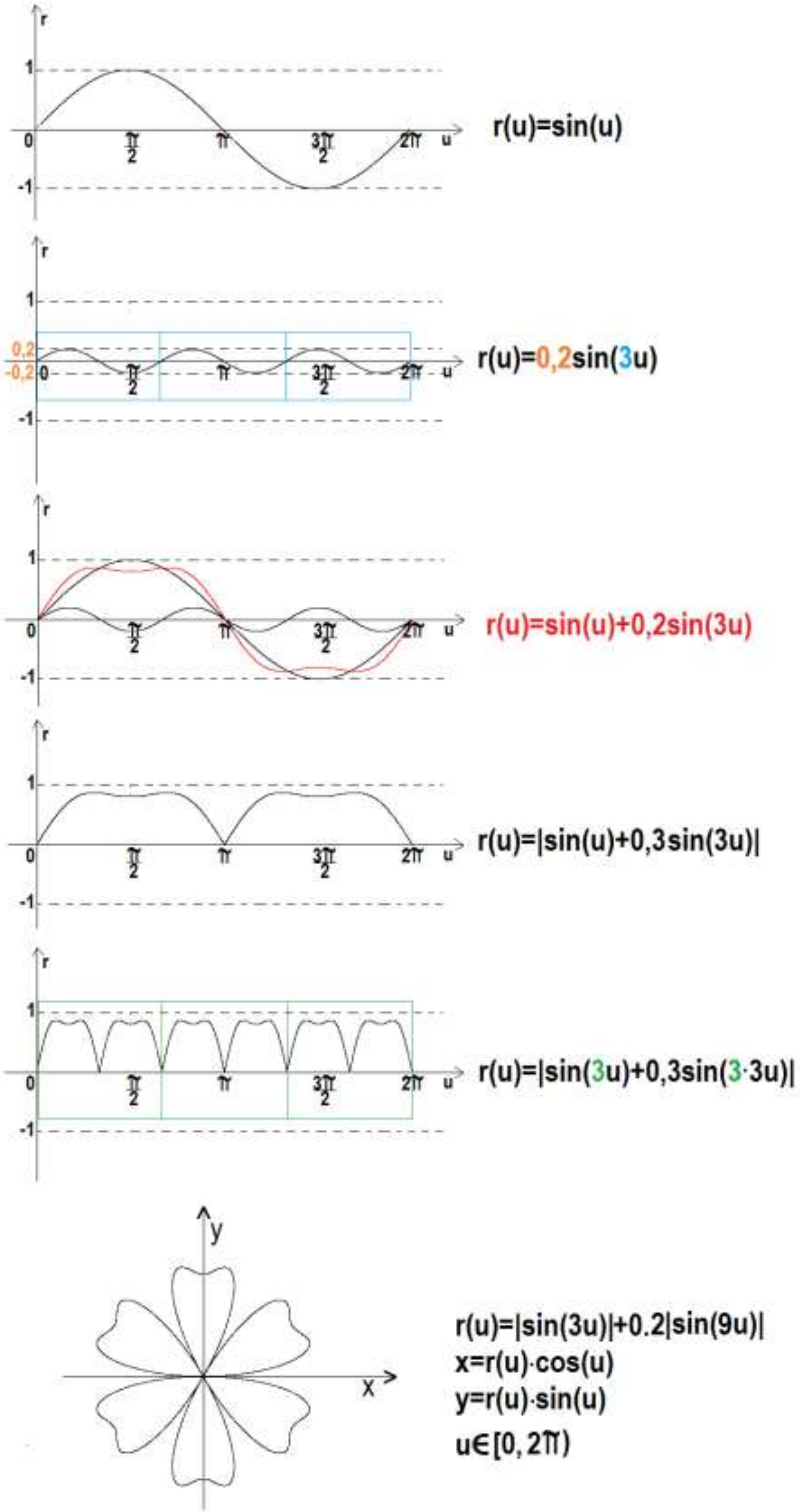
$$r(u)=|\sin(u)|+0,3|\sin(2u)|$$



$$r(u)=|\sin(3u)|+0,3|\sin(3\cdot 2u)|$$



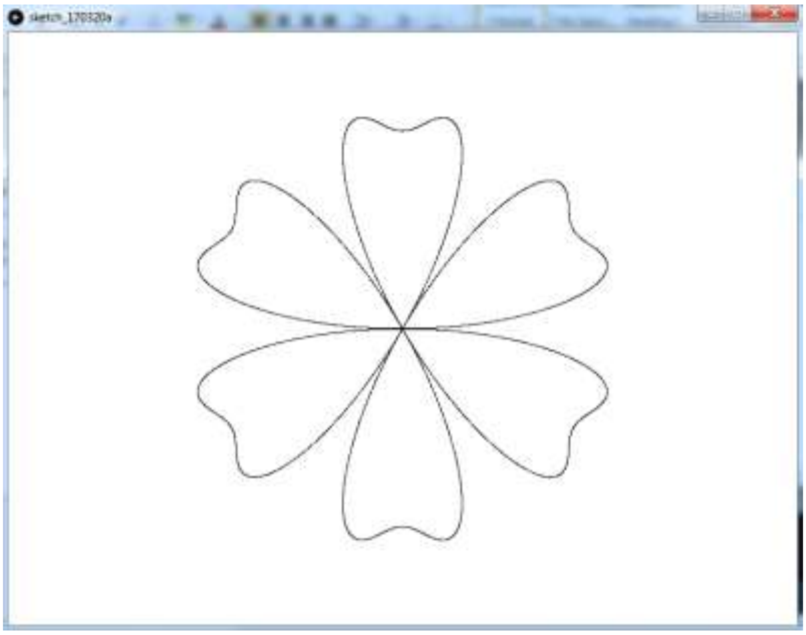
$$r(u)=|\sin(3u)|+0.3|\sin(6u)|$$
$$x=r(u)\cdot\cos(u)$$
$$y=r(u)\cdot\sin(u)$$
$$u\in[0,2\pi)$$


r(u)=sin(u)
r(u)=0,2sin(3u)
r(u)=sin(u)+0,2sin(3u)
r(u)=|sin(u)+0,3sin(3u)|
r(u)=|sin(3u)+0,3sin(3·3u)|
r(u)=|sin(3u)|+0.2|sin(9u)|
x=r(u)·cos(u)
y=r(u)·sin(u)
u∈[0, 2π)

Aplicatie

Program

```
size(800,600);
background(255);
float x,y,r,u,cx=400,cy=300,d=250;
for(u=0;u<=2*PI;u+=0.1/d)
{
  r=abs(sin(3*u)+0.2*sin(9*u));
  x=r*cos(u);
  y=r*sin(u);
  point(x*d+cx,-y*d+cy);
}
```



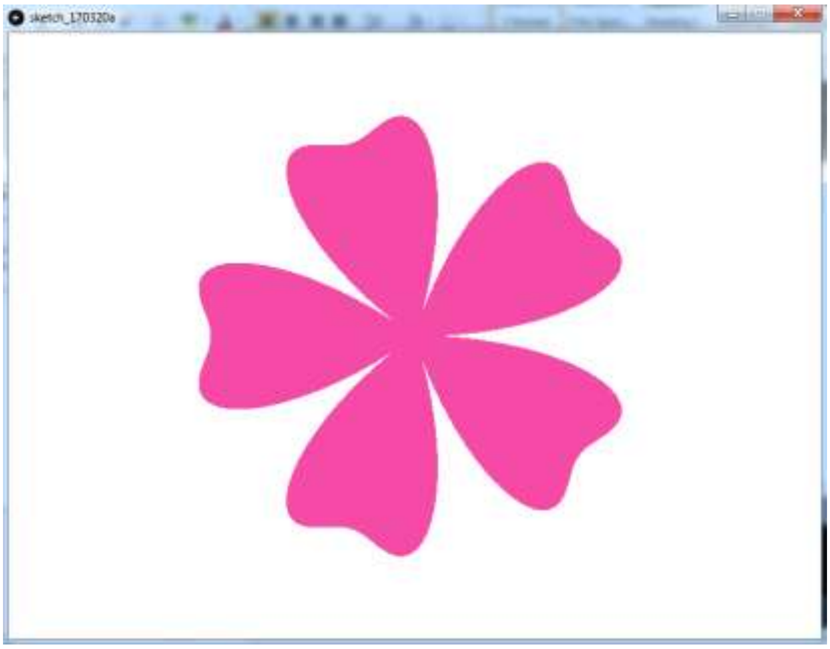
Aplicatie

Vom colora floarea cu o singura culoare. Pentru a umple floarea, vom mai adauga o bucla repetitiva *for* pentru a varia amplitudinea. Vom schimba totodata numarul de petale la 5.

Program

```
size(800,600);
background(255);
float x,y,r,a,u,cx=400,cy=300,d=250;
```

```
for(a=0;a<=1;a+=0.5/d)
  for(u=0;u<=2*PI;u+=0.5/d)
  {
    r=a*abs(sin(2.5*u)+0.2*sin(7.5*u));
    x=r*cos(u);
    y=r*sin(u);
    set(int(x*d+cx),int(-y*d+cy),color(245,74,165));
  }
```

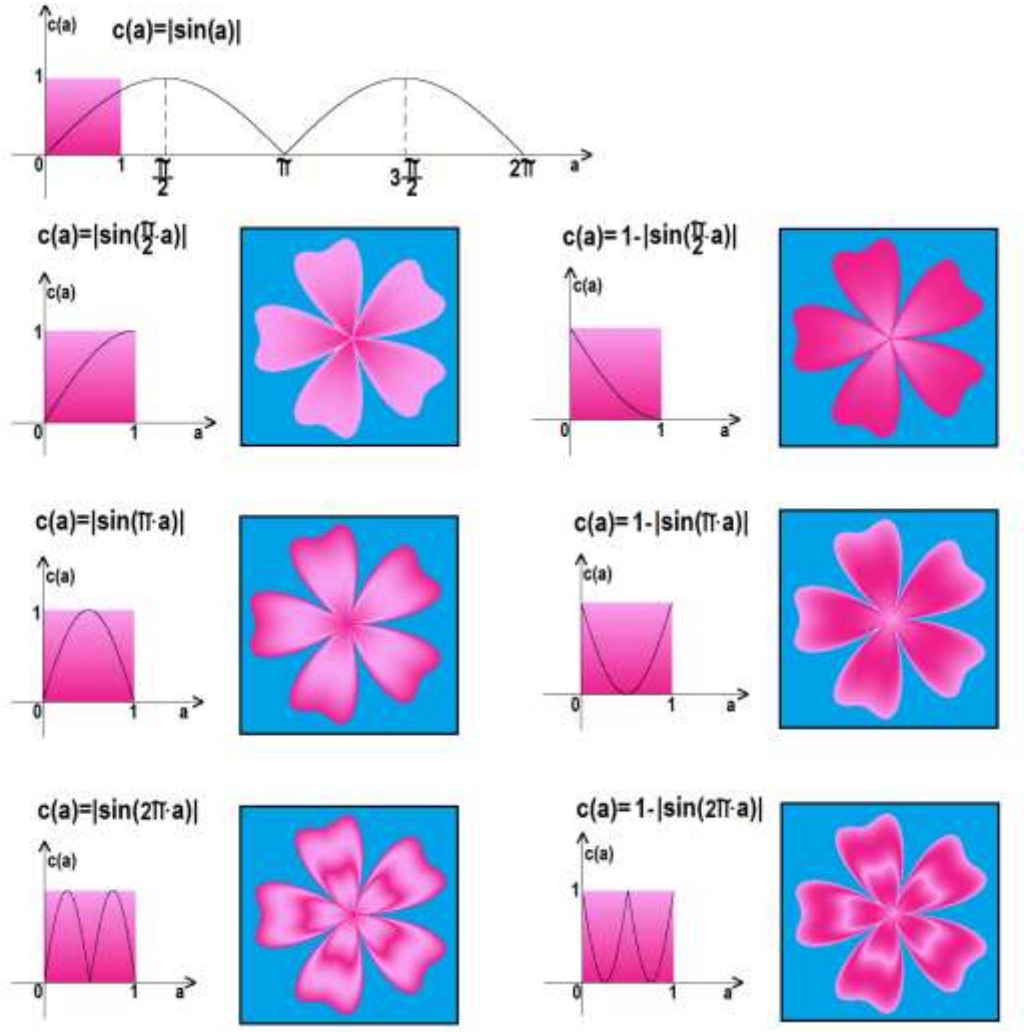


Determinarea functie de interpolare se face prin aceleasi metode ca si modelarea formelor.

Aplicatie

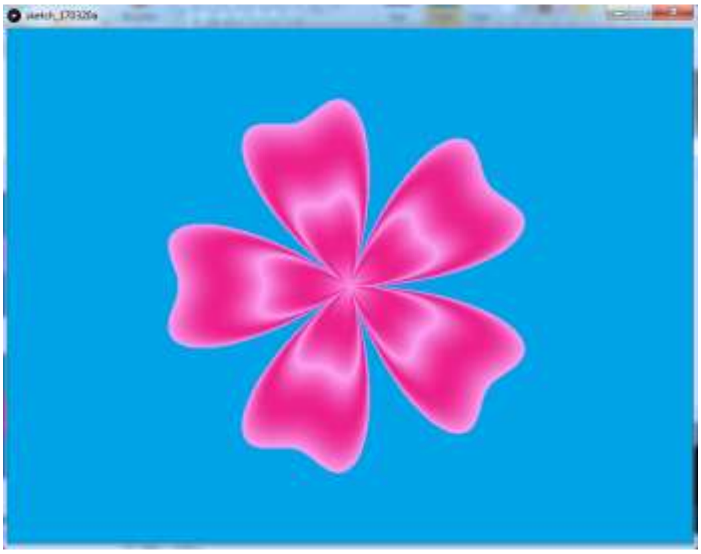
Vom colora floarea printr-un degrade neliniar, a carui variabila de interpolare sa fie in functie de amplitudine, adica de variabila *a*.
Variabila de interpolare trebuie sa fie o functie de forma c:D->[0,1], unde D este intervalul in care este cuprinsa variabila a, adica D=[0,1].
Pentru a intelege mai usor cum se detemina functia de interpolare, vom alege pentru inceput o functie a carei imagine este [0,1] si anume c=|sin(a)|. Intrucat functia isi atinge maximul in pi/2, pentru obtinerea unui singur degrade (o singura trecere de la 0 la 1), este suficient sa comprimam graficul functiei pe axa OX prin inmultirea argumentului cu pi/2. Daca dorim un degrade multiplu (mai multe oscilatii intre 0 si 1), atunci inmultim argumentul cu un multiplu de pi/2.
Pentru inversarea culorilor, se scade functia din 1.

.



Program

```
size(800,600);
background(0,162,232);
float x,y,r,a,u,cx=400,cy=300,d=250,
R2=250,G2=160,B2=240,R1=237,G1=33,B1=140,R,G,B,c;
for(a=0;a<=1;a+=0.5/d)
  for(u=0;u<=2*PI;u+=0.5/d)
  {
    r=a*abs(sin(2.5*u)+0.2*sin(7.5*u));
    x=r*cos(u);
    y=r*sin(u);
    c=1-abs(sin(2*PI*a));
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
```

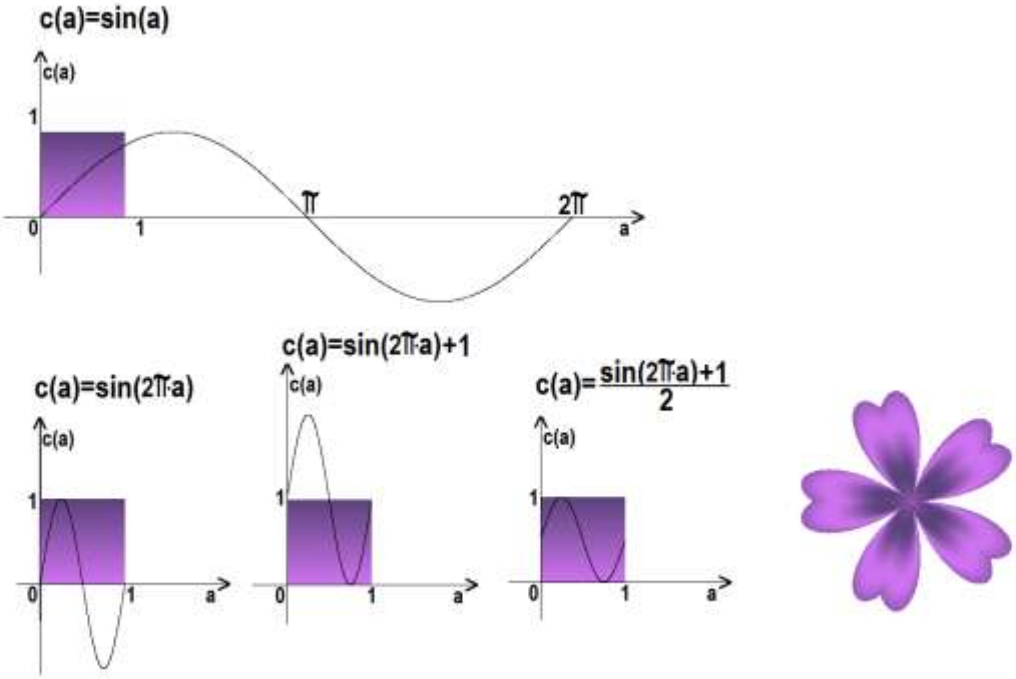


Aplicatie

Vom colora o floare cu 5 petale in forma de inima printr-un degrade similar cu cel din aplicatia precedenta.
De aceasta data, vom determina o functie de interpolare pornind de la o functie a carei imagine nu este [0,1]. Vom alege functia c=sin(a), c:[0,2pi]->[-1,1].

<u>Rezolvare matematica</u>

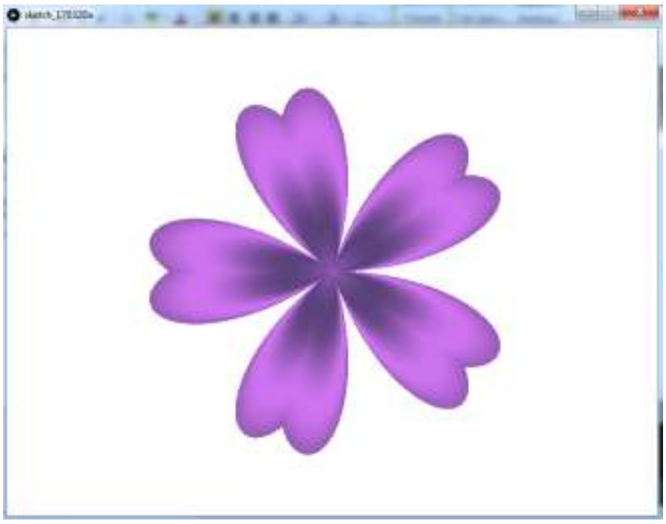
Intrucat variabila a este pe D=[0,1], comprimam mai intai functia pe OX prin inmultirea argumentului cu 2pi. Intrucat imaginea functiei este [-1,1], vom decala graficul in sus pe OY, adunind la functie 1. Imaginea functiei devine astfel [0,2]. Vom comprima graficul pe OY inmultind functia cu 2.



<u>Program</u>

```
size(800,600);
background(255);
float x,y,r,a,u,d=200,cx=400,cy=300,
R1=207,G1=116,B1=241,R2=92,G2=67,B2=124,R,G,B,c;
for(u=0;u<=2*PI;u+=0.5/d)
  for(a=0;a<=1;a+=0.5/d)
  {
    r=a*(abs(sin(2.5*u))+0.3*abs(sin(5*u)));
```

```
  x=r*cos(u);
  y=r*sin(u);
  c=(sin(2*PI*a)+1)/2;
  R=R1*(1-c)+R2*c;
  G=G1*(1-c)+G2*c;
  B=B1*(1-c)+B2*c;
  set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
 }
```

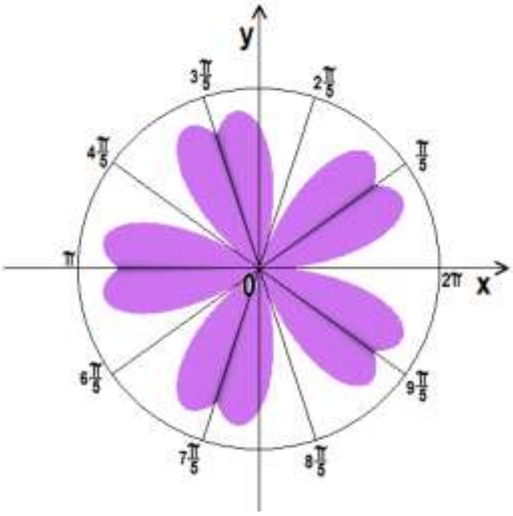


Aplicatie

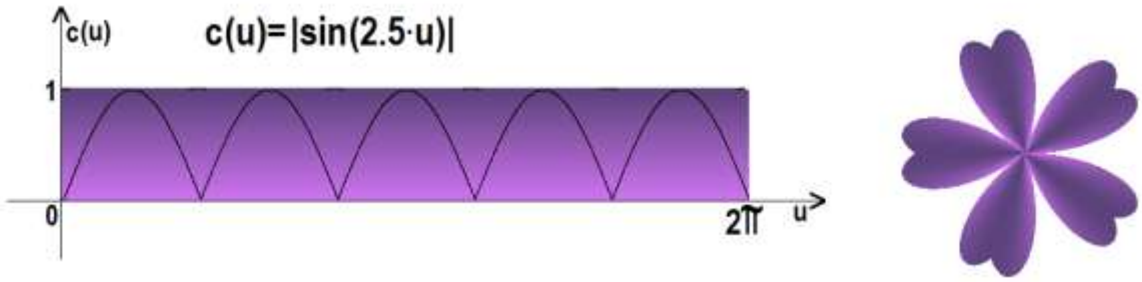
Vom desena dungi la jumatatea fiecarei petale a florii cu 5 petale in forma de inima printr-un degrade neliniar.

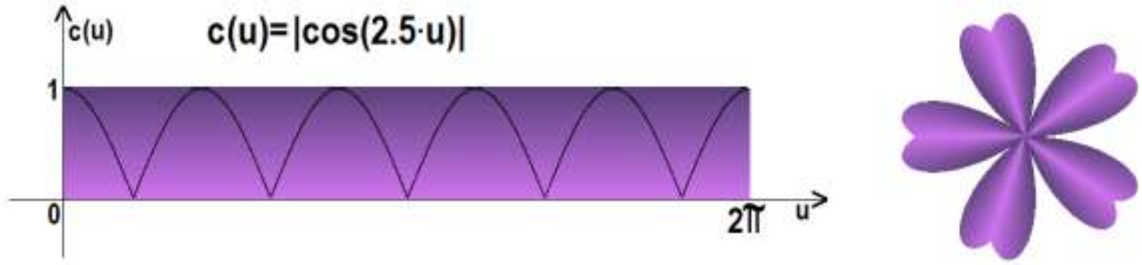
Rezolvare matematica

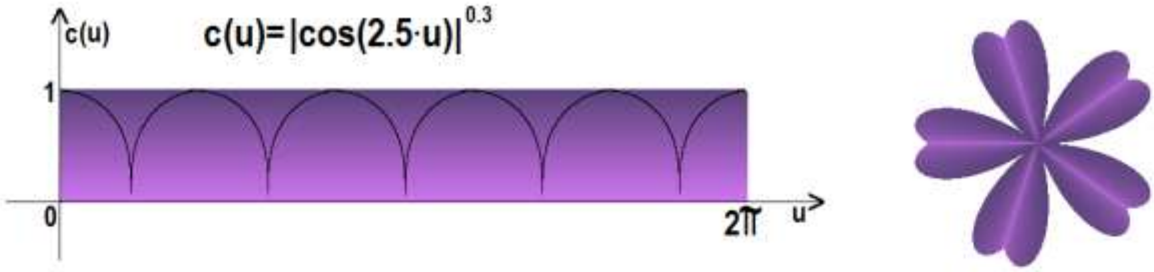
Pentru a obtine dungi la mijlocul fiecarei petale, trebuie ca variabila de interpolare sa varieze in functie de *u*. Astfel, c:[0,2PI]->[0,1].



Forma dungilor se poate obtine cu ajutorul functie c=|sin(u)|. Momentan nu ne preocupam de ordinea culorilor, intrucat se pot inversa la sfarsit. Comprimam graficul cu jumatatea numarului de petale si obtinem c=|sin(2.5u)|. In momentul de fata, dungile sunt intre petale. Intrucat floarea are 5 petale, jumatatea fiecareia este la un multiplu impar de pi/5. Decalam graficul pe axa OX spre stanga cu pi/5, adunind la argument pi/5. Se obtine c=|sin(2.5*(u+pi/5))|=|sin(2.5u+pi/2)| =|cos(2.5u)|. Pentru a le subtia aplicam functia putere, cu puterea subunitara. Spre exemplu $c=|\cos(2.5u)|^{0.3}$.

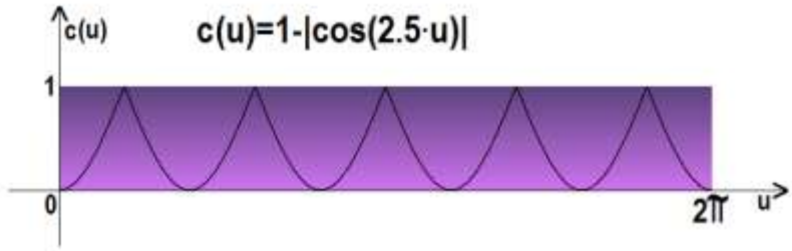


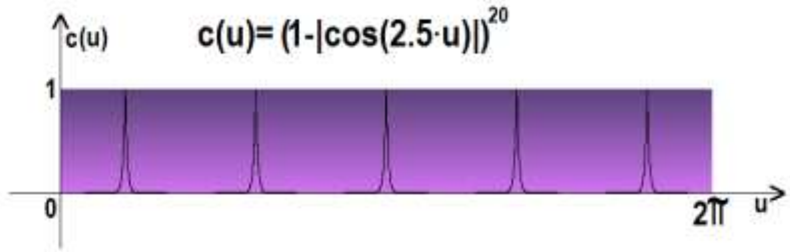




Testam diferite valori pentru putere, insa rezultatul nu este cel asteptat. Dungile astfel obtinute nu sunt clar conturate, asa ca aplicam o alta metoda.
Inversam graficul, scazind din 1 functia. Obtinem c=1-|cos(2.5u)|. Subtiem din nou dungile prin ridicarea la o putere supraunitara, cum ar fi $c=(1-|\cos(2.5u)|)^{20}$.
Testam iarasi diferite valori pentru putere, pana obtinem modelul dorit.

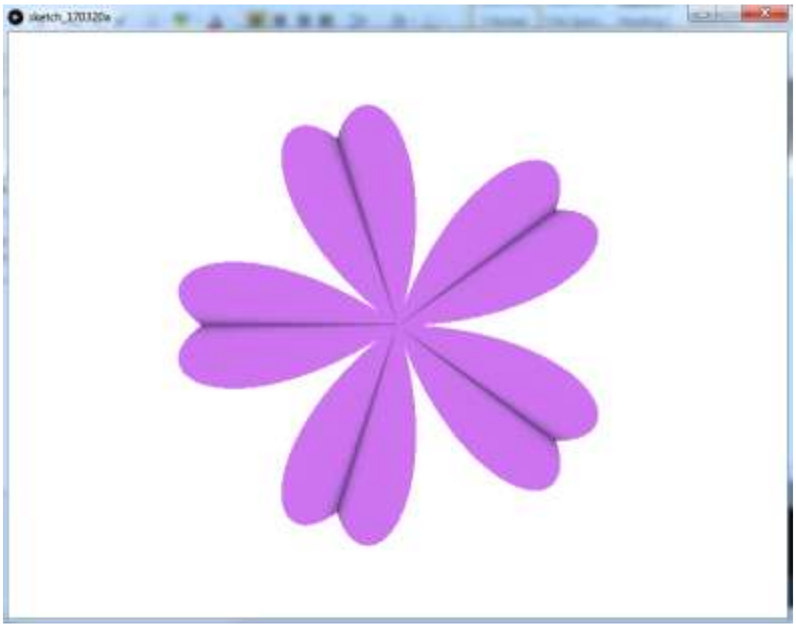
Desi aplicarea functiei putere este un exercitiu util pentru incepatori, nu este cea mai potrivita alegere in modelarea formelor cat si in modelarea in culoare.





<u>Program</u>

```
size(800,600);
background(255);
float x,y,r,a,u,d=200,cx=400,cy=300,
R1=207,G1=116,B1=241,R2=92,G2=67,B2=124,R,G,B,c;
for(u=0;u<=2*PI;u+=0.5/d)
  for(a=0;a<=1;a+=0.5/d)
  {
    r=a*(abs(sin(2.5*u))+0.3*abs(sin(5*u)));
    x=r*cos(u);
    y=r*sin(u);
    c=pow(1-abs(cos(2.5*u)),20);
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
```



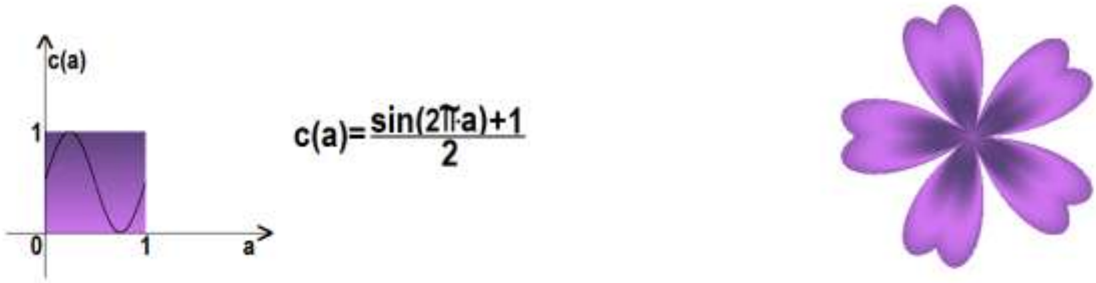
Compunerea degradeurilor

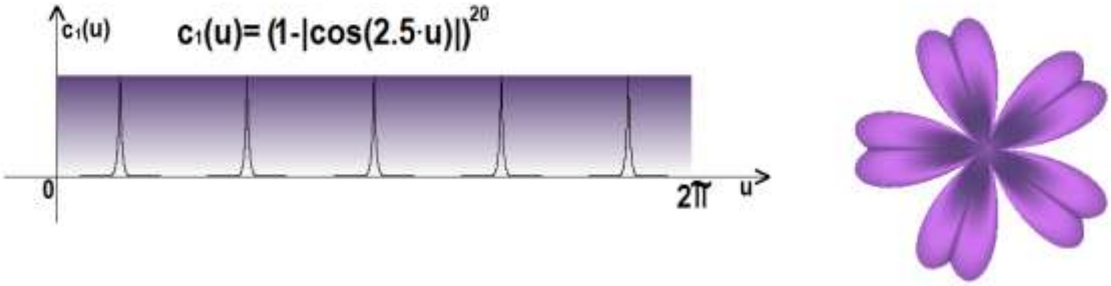
$$R=R_1\cdot(1-c)+R_2\cdot c$$
$$G=G_1\cdot(1-c)+G_2\cdot c$$
$$B=B_1\cdot(1-c)+B_2\cdot c$$
$$R=R\cdot(1-c_1)+R_3\cdot c_1$$
$$G=G\cdot(1-c_1)+G_3\cdot c_1$$
$$B=B\cdot(1-c_1)+B_3\cdot c_1$$
$$c=c(x)\,,\ c:D\rightarrow[0,1]$$
$$c_1=c_1(x)\,,\ c_1:D\rightarrow[0,1]$$

Compunerea degradeurilor, presupune ca output-ul unui degrade sa fie inputul altui degrade. Adica noile valori ale parametrilor R,G,B sa se calculeze in functie de vechile valori ale lor.

Aplicatie

Vom compune cele doua degradeuri din aplicatiile precedente.

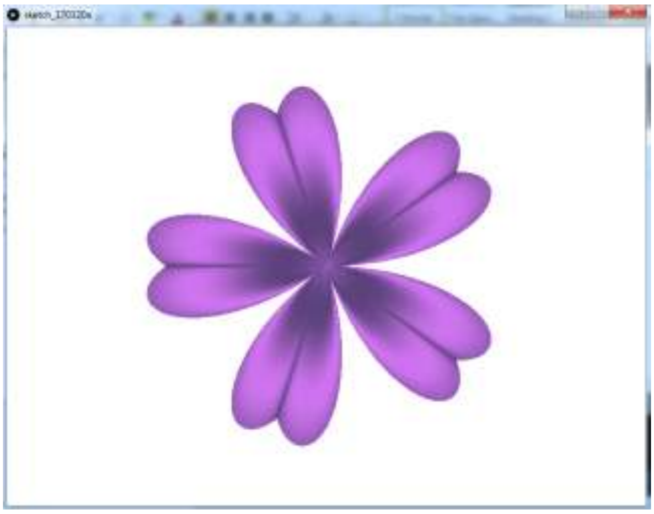




Program

```
size(800,600);
background(255);
float x,y,r,a,u,d=200,cx=400,cy=300,
R1=207,G1=116,B1=241,R2=92,G2=67,B2=124,R,G,B,c,c1;
for(u=0;u<=2*PI;u+=0.5/d)
  for(a=0;a<=1;a+=0.5/d)
  {
    r=a*(abs(sin(2.5*u))+0.3*abs(sin(5*u)));
    x=r*cos(u);
    y=r*sin(u);
    c=(sin(2*PI*a)+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    c1=pow(1-abs(cos(2.5*u)),20);
    R=R*(1-c1)+R2*c1;
    G=G*(1-c1)+G2*c1;
    B=B*(1-c1)+B2*c1;
```

```
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
```



La compunerea degradeurilor trebuie sa avem in vedere care degrade il suprapunem peste celalalt.

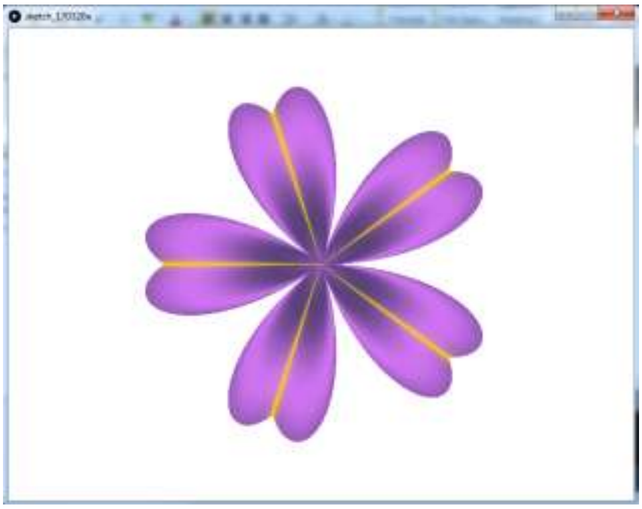
Aplicatie

Vom schimba culoarea de la cel de-al doilea degrade la galben pentru a iesi in evidenta.

Program

```
size(800,600);
background(255);
float x,y,r,a,u,d=200,cx=400,cy=300,
R1=207,G1=116,B1=241,R2=92,G2=67,B2=124,
R3=255,G3=226,B3=0,R,G,B,c,c1;
for(u=0;u<=2*PI;u+=0.5/d)
  for(a=0;a<=1;a+=0.5/d)
  {
    r=a*(abs(sin(2.5*u))+0.3*abs(sin(5*u)));
    x=r*cos(u);
    y=r*sin(u);
    c=(sin(2*PI*a)+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    c1=pow(1-abs(cos(2.5*u)),20);
```

```
    R=R*(1-c1)+R3*c1;
    G=G*(1-c1)+G3*c1;
    B=B*(1-c1)+B3*c1;
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
```



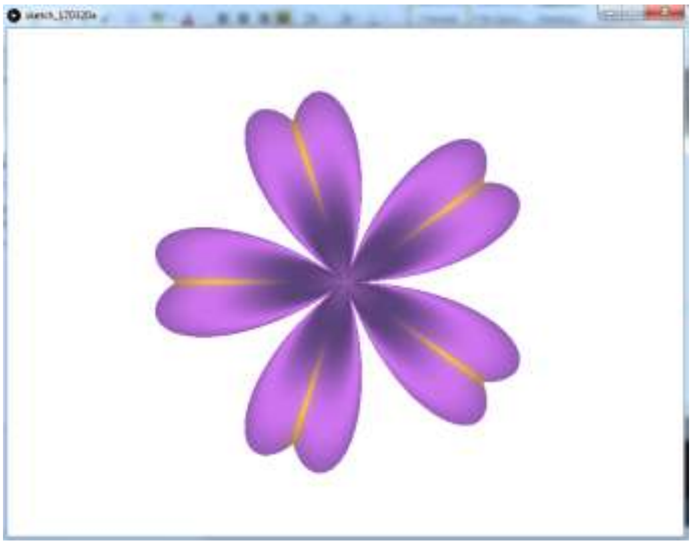
Aplicatie

Vom inversa ordinea degradeurilor.Consideram culoarea (R1,G1,B1) ca fiind cea a fondului.

Program

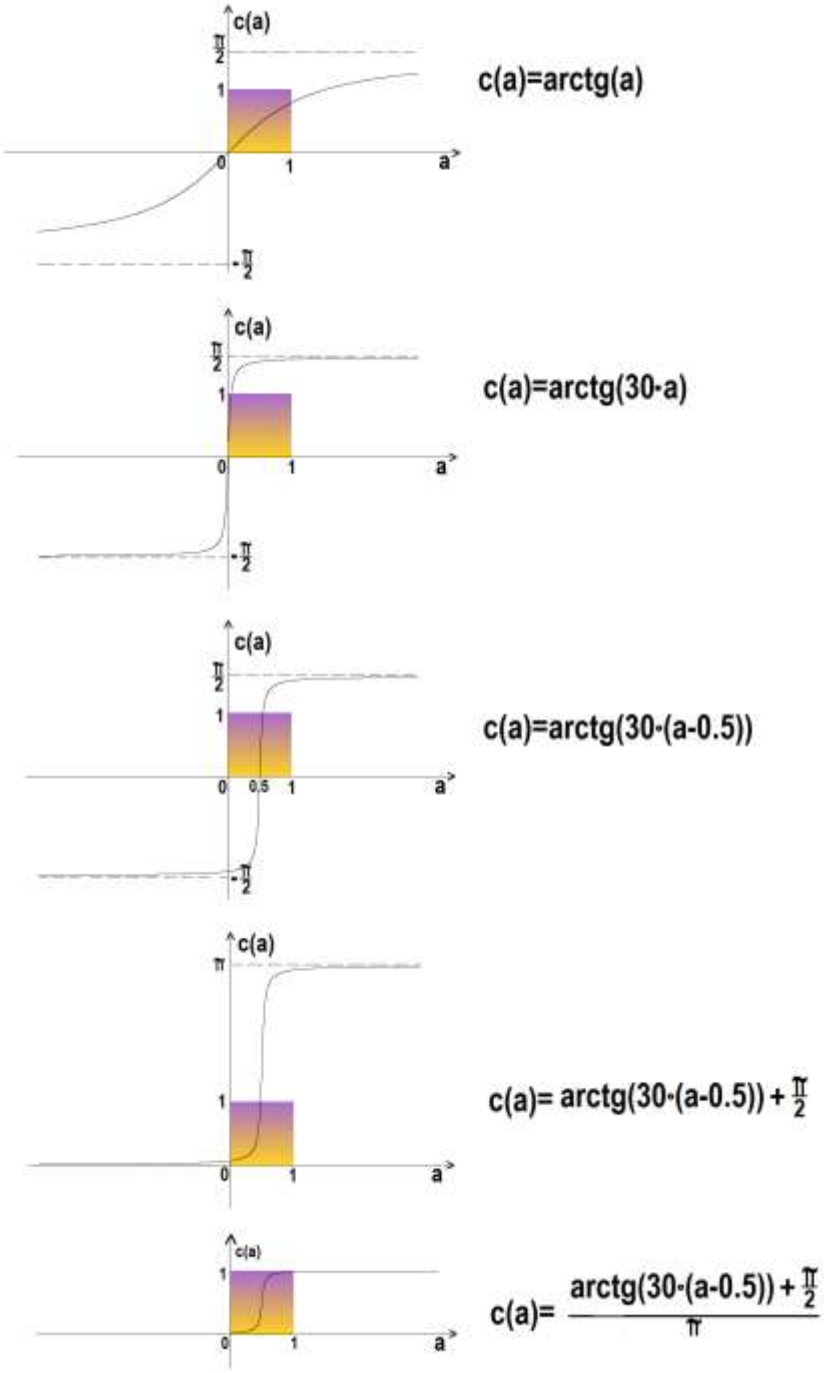
```
size(800,600);
background(255);
float x,y,r,a,u,d=200,cx=400,cy=300,
R1=207,G1=116,B1=241,R2=92,G2=67,B2=124,
R3=255,G3=226,B3=0,R,G,B,c,c1;
for(u=0;u<=2*PI;u+=0.5/d)
  for(a=0;a<=1;a+=0.5/d)
  {
    r=a*(abs(sin(2.5*u))+0.3*abs(sin(5*u)));
    x=r*cos(u);
    y=r*sin(u);
    c1=pow(1-abs(cos(2.5*u)),20);
    R=R1*(1-c1)+R3*c1;
    G=G1*(1-c1)+G3*c1;
    B=B1*(1-c1)+B3*c1;
    c=(sin(2*PI*a)+1)/2;
    R=R*(1-c)+R2*c;
```

```
  G=G*(1-c)+G2*c;
  B=B*(1-c)+B2*c;
  set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
}
```



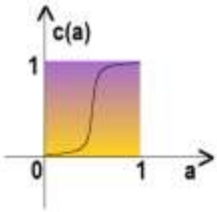
Accentuarea formelor si culorilor pe baza arctangentei

Functia arctangenta aduce multe avantaje in modelarea formelor si in modelarea in culoare, avantaje pe care le vom vedea mai tarziu, pe aplicatii.
Pentru moment dorim sa obtinem o functie de interpolare care sa arate ca o treapta, pornind de la arctangenta, astfel incat c:[0,1]->[0,1]. O vom numi „functia treapta”.
In imagine sunt prezentate etapele obtinerii functiei treapta.

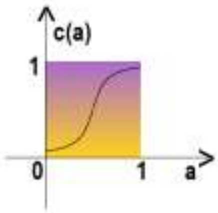


Aplicatie

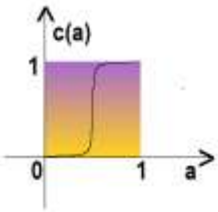
Vom studia avantajele aplicarii functiei treapta in modelarea in culoare si vom vedea totodata si de ce parametrii dispunem si care e rolul lor.



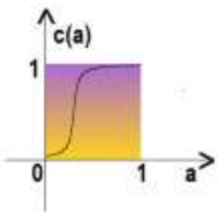
$$c(a)=\frac{\operatorname{arctg}(30\cdot(a-0.5))+\frac{\pi}{2}}{\pi}$$



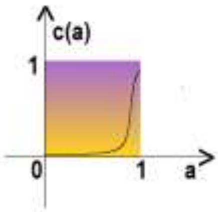
$$c(a)=\frac{\operatorname{arctg}(10\cdot(a-0.5))+\frac{\pi}{2}}{\pi}$$



$$c(a)=\frac{\operatorname{arctg}(100\cdot(a-0.5))+\frac{\pi}{2}}{\pi}$$



$$c(a)=\frac{\operatorname{arctg}(30\cdot(a-0.3))+\frac{\pi}{2}}{\pi}$$



$$c(a)=\frac{\operatorname{arctg}(30\cdot(a-0.9))+\frac{\pi}{2}}{\pi}$$

Vom denumi parametrii functiei treapta *alpha* si *beta*. Astfel functia treapta devine:

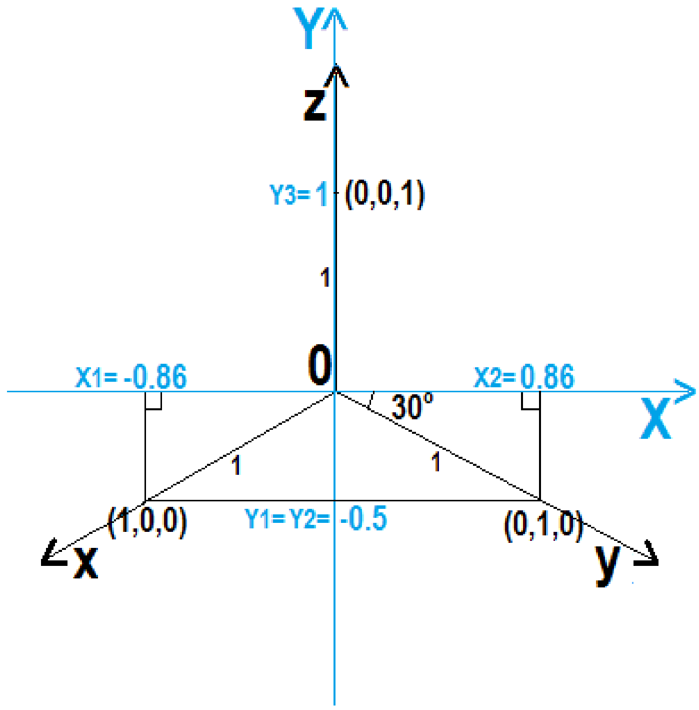
c(x)=arctg(*alpha*(x-*beta*))

Rolul parametrilor este:

-*alpha*: cu cat alpha este mai mare cu atat trecerea de la o culoare la alta este mai abrupta, iar cu cat alpha este mai mic cu atat trecerea este mai lenta

-*beta*: are rolul de decalaj, adica, in ce valoare a argumentului dorim sa se faca trecerea de la o culoare la alta.

Proiectia sistemului cartezian 3D in 2D



$X_1 = \cos(30^\circ) \simeq -0.86$ $\qquad$ $Y_1 = -\sin(30^\circ) \simeq -0.5$

$X_2 = -\cos(30^\circ) \simeq 0.86$ $\qquad$ $Y_2 = -\sin(30^\circ) \simeq -0.5$

$X_3 = 0$ $\qquad$ $Y_3 = 1$

$$(x,y,z) = x\cdot(1,0,0) + y\cdot(0,1,0) + z\cdot(0,0,1)$$

$$X = x\cdot X_1 + y\cdot X_2 + z\cdot X_3 = x\cdot 0.86 + y\cdot(-0.86) + z\cdot 0$$

$$Y = x\cdot Y_1 + y\cdot Y_2 + z\cdot Y_3 = x\cdot(-0.5) + y\cdot(-0.5) + z\cdot 1$$

$$\boxed{\begin{aligned} X &= -0.86\cdot x + 0.86\cdot y \\ Y &= z - 0.5\cdot x - 0.5\cdot y \end{aligned}}$$

Reprezentarea grafica a functiilor 3D

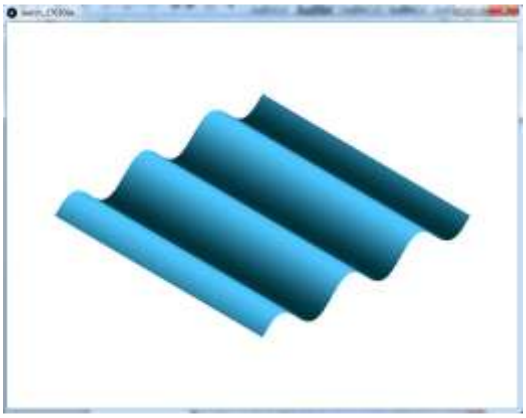
Vom studia pentru inceput cazurile particulare, in care unul dintre parametrii x sau y are valoare constanta.

Graficele functiilor care au y constant, sunt o etensie in 3D a graficelor functiilor 2D, prin translatie de-alungul axei OY. Similar, graficele functiilor cu x constant, sunt o extensie prin translatie de-a lungul axei OX. Acest lucru este relevant, pentru o mai buna intelegere a modelarii pe baza operatiilor si compunerilor de functii, pe care le vom studia mai tarziu.

Aplicatie 1

Program

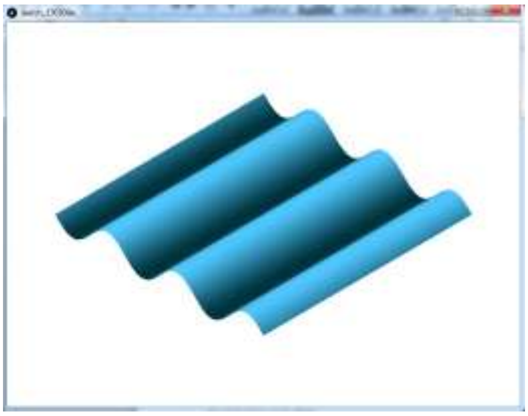
```
size(800,600);
background(255);
float x,y,z,X,Y,cx=400,cy=300,d=20,c,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B;
for(y=-3*PI;y<=3*PI;y+=0.5/d)
  for(x=-3*PI;x<=3*PI;x+=0.1/d)
  {
    z=sin(x);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=(z+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```



Aplicatie 2

Program

```
size(800,600);
background(255);
float x,y,z,X,Y,cx=400,cy=300,d=20,c,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B;
for(y=-3*PI;y<=3*PI;y+=0.5/d)
  for(x=-3*PI;x<=3*PI;x+=0.1/d)
  {
    z=sin(y);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=(z+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```

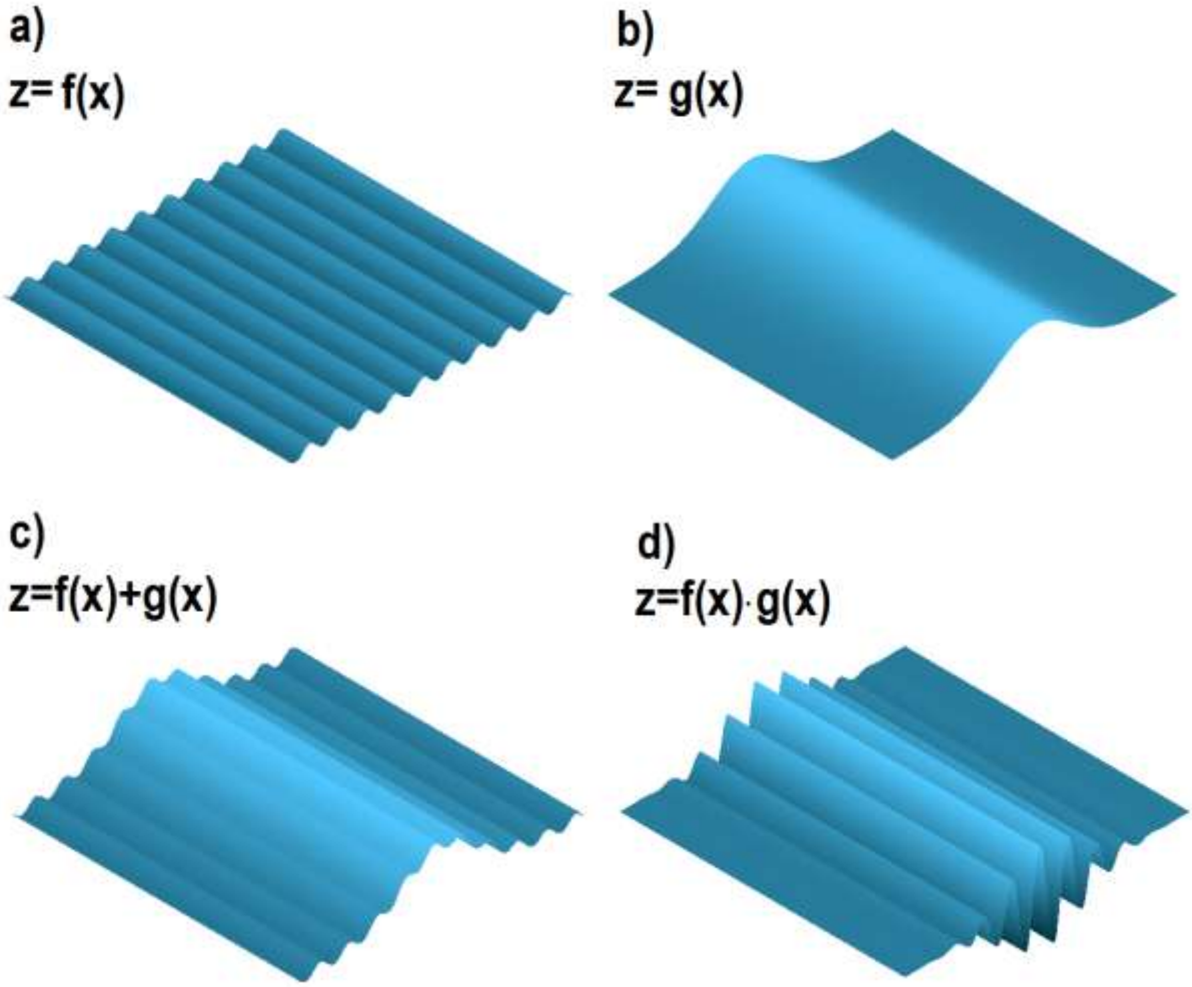


Operatii si compuneri de functii utilizate in modelarea 3D

Asa cum am vazut in capitolul precedent, functiile cu unul dintre parametrii x sau y constant sunt o extensie in 3D a functiilor de o variabila. Din acest motiv, si efectele operatiilor cu astfel de functii sunt aceleasi ca si cele studiate la 2D.

Convenim sa notam:
z=f(x), functiile care au y-cst
z=f(y), functiile care au x-cst

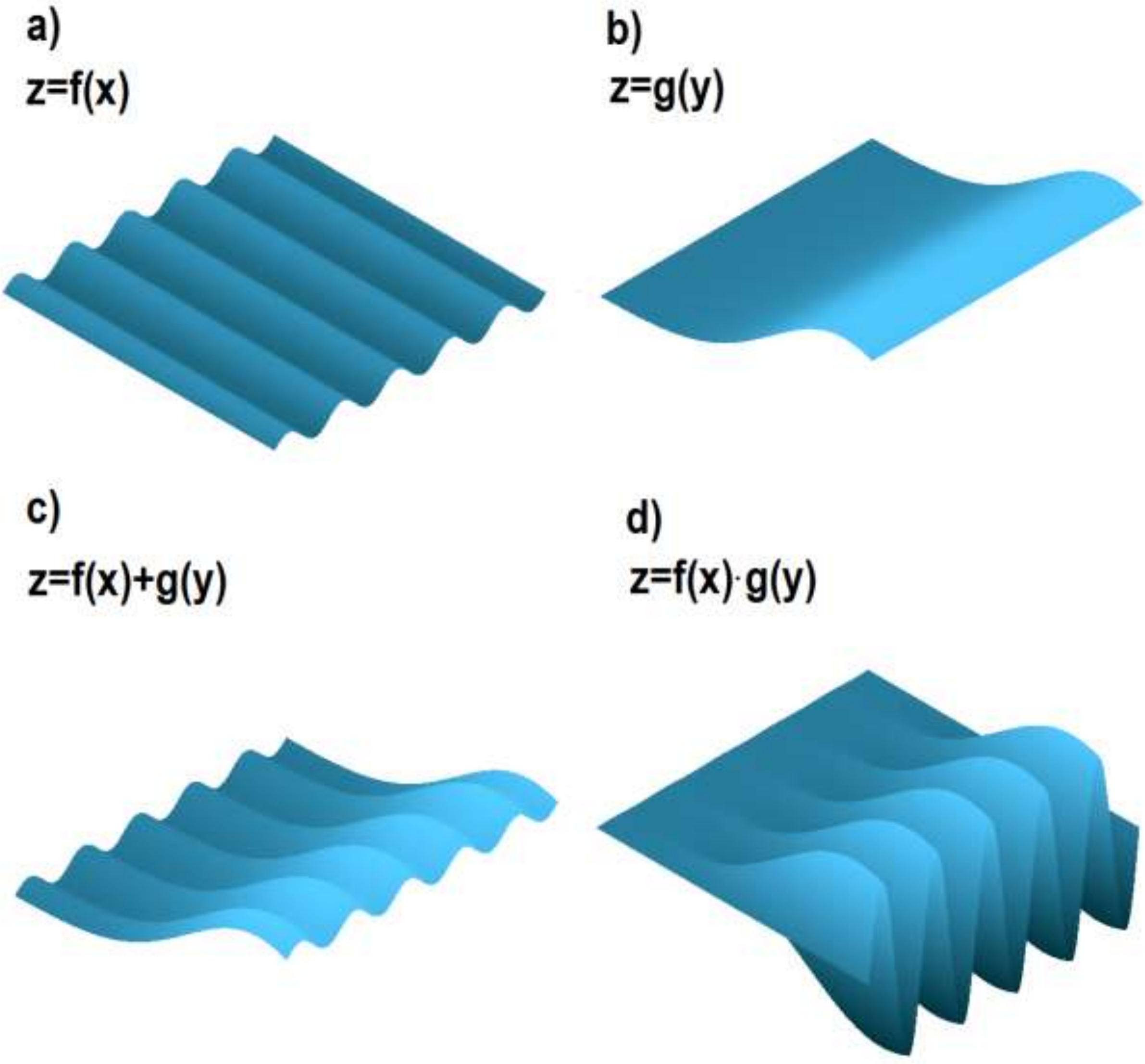
Operatii cu functii care depind de aceeasi variabila



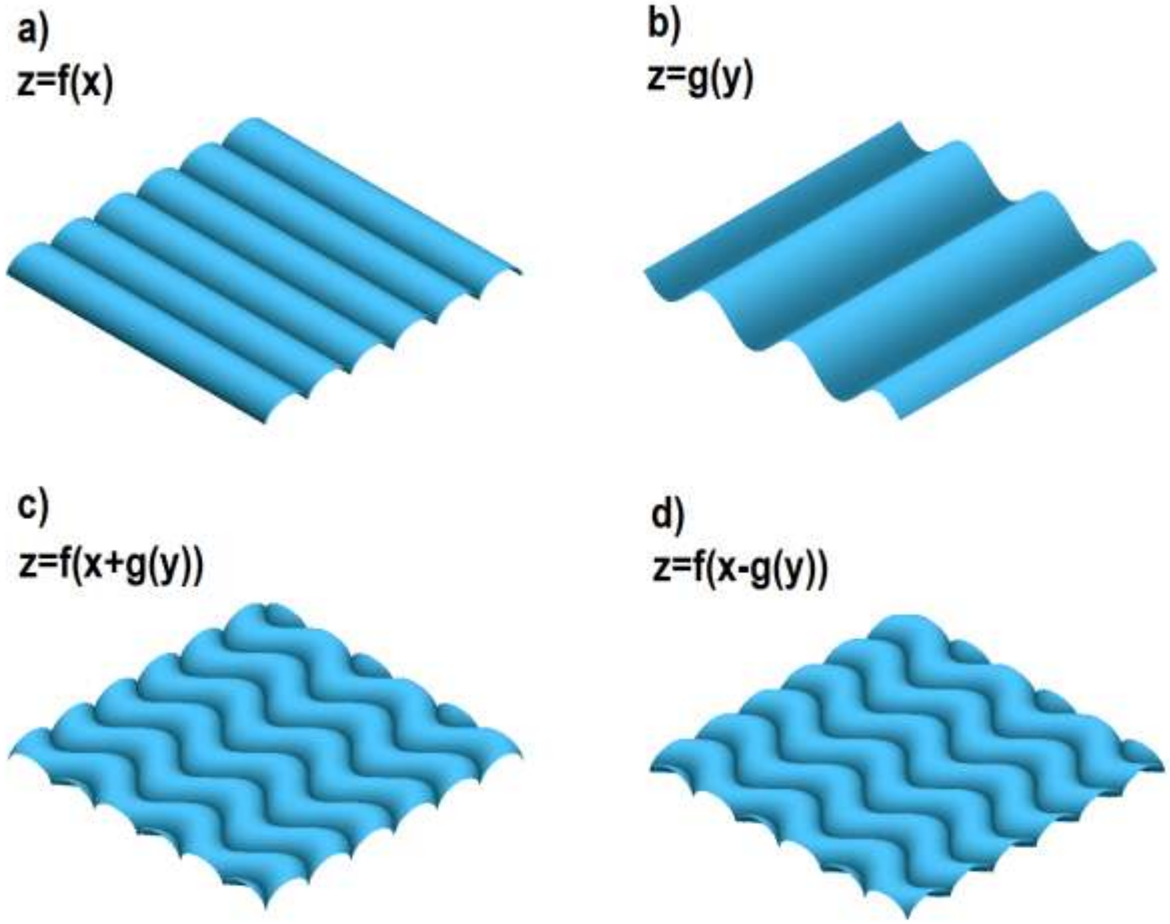
Operatii cu functii care depind de variabile diferite

Pentru a intelege efectul grafic al adunarii functiilor de tipul z=f(x) si z=g(y), nu putem imagina cum graficul uneia dintre functii, spre exemplu f(x), urca/coboara, in functie de cota data de cealalta functie - g(y).

In cazul inmultirii functiilor de tipul z=f(x) si z=g(y), amplitudinea uneia dintre functii se mareste/micsoreaza de atatea ori cat este cota celeilalte functii.



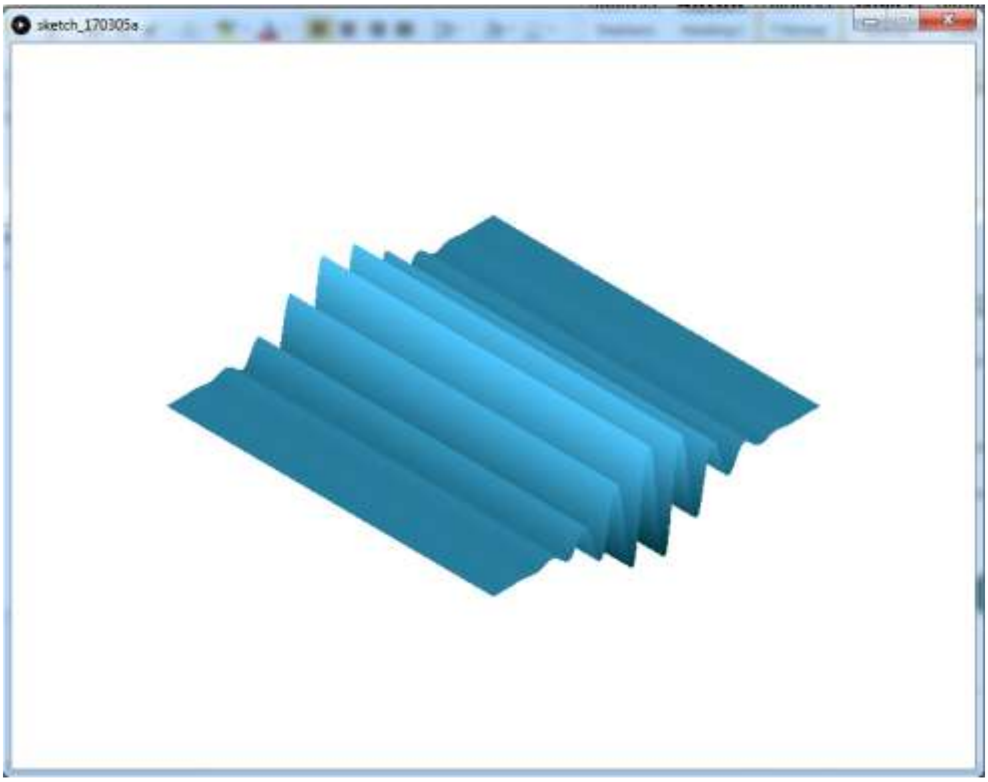
Compuneri de functii care depind de variabile diferite



Aplicatie 1

Program

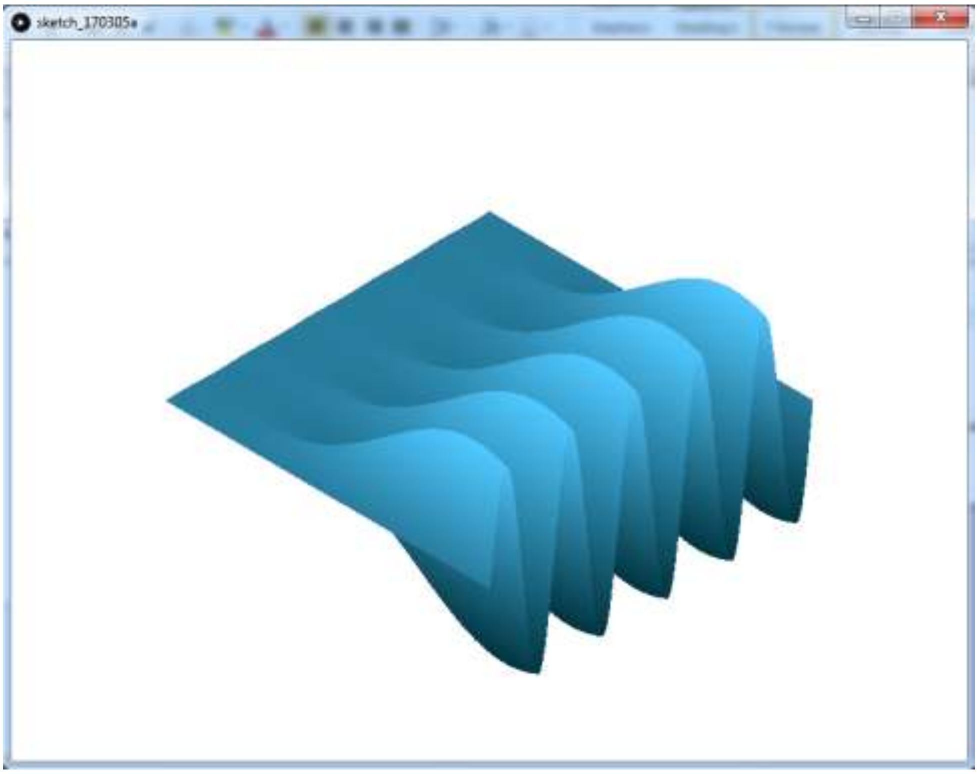
```
size(800,600);
background(255);
float x,y,z,X,Y,cx=400,cy=300,d=5,c,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B;
for(y=-10*PI;y<=10*PI;y+=0.5/d)
  for(x=-10*PI;x<=10*PI;x+=0.1/d)
  {
    z=sin(x)*10*pow(2,-pow(0.1*x,2));
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=(z/10+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```



Aplicatie 2

Program

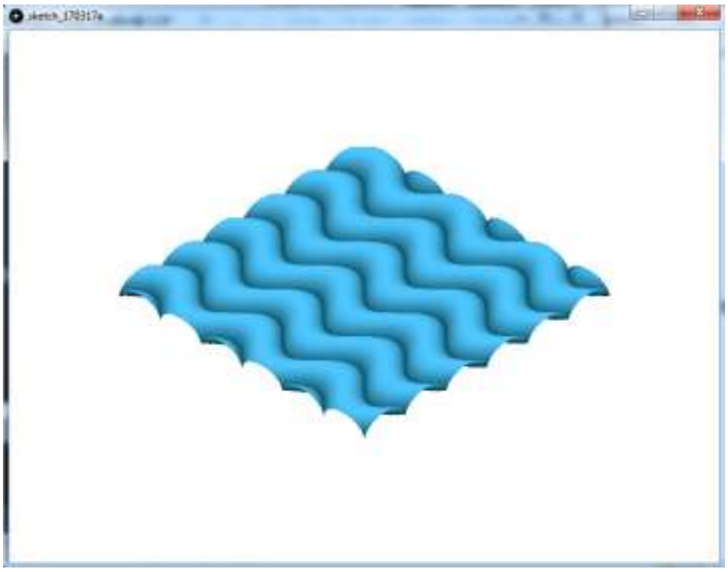
```
size(800,600);
background(255);
float x,y,z,X,Y,cx=400,cy=300,d=10,c,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B;
for(y=-5*PI;y<=5*PI;y+=0.5/d)
  for(x=-5*PI;x<=5*PI;x+=0.1/d)
  {
    z=sin(x)+10*pow(2,-pow(0.1*(y-4*PI),2));
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=(z/10+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```



Aplicatie 3

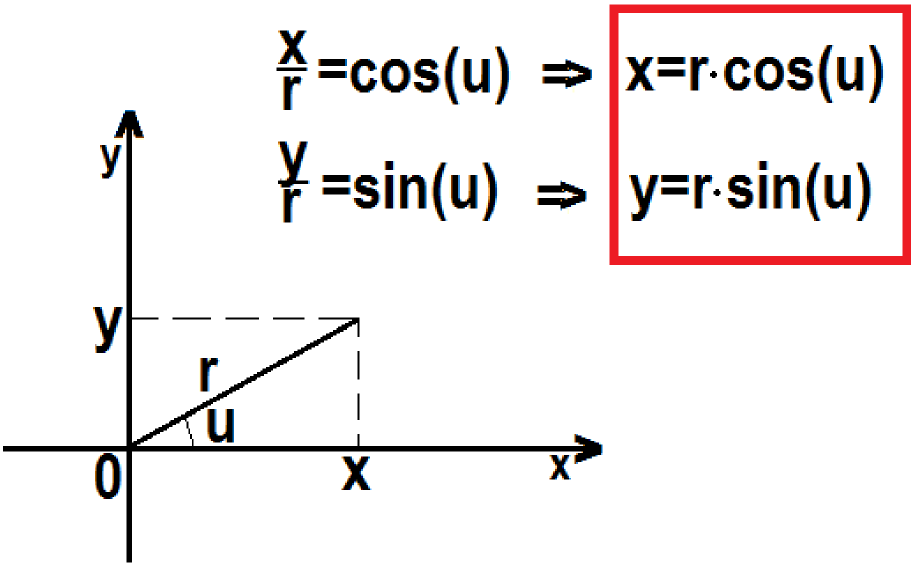
Program

```
size(800,600);
background(255);
float x,y,z,X,Y,cx=400,cy=300,d=17,c,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B;
for(y=-3*PI;y<=3*PI;y+=0.2/d)
  for(x=-3*PI;x<=3*PI;x+=0.2/d)
  {
    z=pow(abs(sin(x-sin(y))),0.5);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=z;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```



Coordonate polare

S**istemul de coordonate polare** este un sistem de coordonate bidimensional în care fiecărui punct din plan i se asociază un unghi și o distanță.



Trecerea in coordonate carteziene se face pe baza formulelor trigonometrice:

x=r cos(u)
y=r sin(u)

Reprezentarea grafica a functiilor in coordonate polare

Teorie programare

Tablouri

Un **tablou** este un sir de valori de acelasi tip, aflate in locatii consecutive de memorie.

Vectorii sunt tablouri unidimensionale

Declaratie:

```
tip[] nume=new tip[nr_max_elemente];
ex:
float v=new float[100];
```

Parcurgere:

```
for(int i=0;i<100;i++)
    ...v[i]...
```

Matricea este un tablou bidimensional

Declaratie:

```
tip[][] nume=new tip[nr_max_linii][nr_max_coloane];
Ex:
float[][] mat=new float[100][100];
```

Parcurgere:

```
int i,j,n,m;
for(i=0;i<n;i++)
   for(j=0;j<m;j++)
      ... m[i][j]...
```

Subprograme

Un subprogram este o secventa de instructiuni care poate fi apelata din programul principal sau dintr-un alt subprogram.

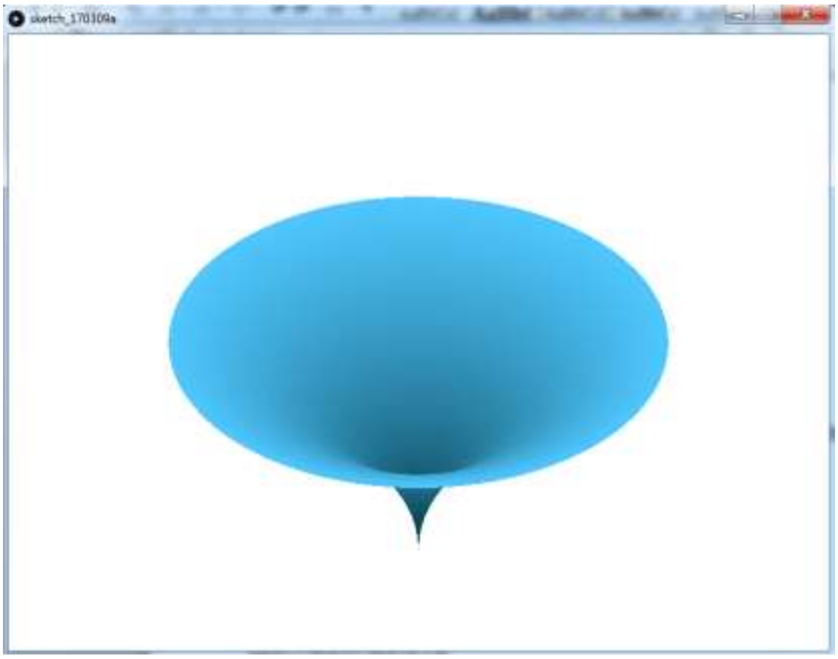
```
tip_returnat nume_funcţie (lista parametrilor formali)
{
 instrucţiune; // corpul funcţiei
}
```

Aplicatie 1

Cazul z=f(r)

Program

```
size(800,600);
background(255);
float x,y,z,r,u,X,Y,d=200,cx=400,cy=500,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B,c,c1;
for(r=0;r<=1;r+=0.1/d)
  for(u=0;u<=2*PI;u+=0.1/d)
  {
    z=sqrt(r);
    x=r*cos(u);
    y=r*sin(u);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=z;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```



In cazul reprezentarilor grafice ale functiilor in coordonate polare (si nu numai), apare de multe ori problema suprapunerii partii din spate a graficului peste partea din fata, cum e cazul urmator.

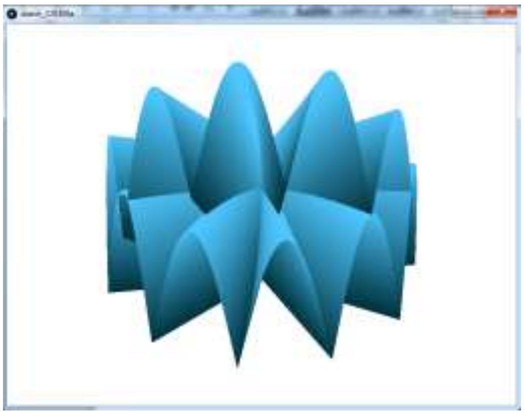
Aplicatie 2

Cazul z=f(u)

Program

```
size(800,600);
background(255);
float x,y,z,r,u,X,Y,d=200,cx=400,cy=400,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B,c,c1;
for(r=0;r<=1;r+=0.1/d)
  for(u=0;u<=2*PI;u+=0.1/d)
  {
    z=abs(sin(5*u));
    x=r*cos(u);
    y=r*sin(u);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=z;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
```

```
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```



O astfel de problema se rezolva cu ajutorul unei matrici. Vom asocia fiecarui element din matrice un pixel din fereastra. De cate ori se va desena un pixel, vom memora in matrice pe pozitia corespondenta pixelului respectiv, valoarea lui z. Punctele a caror proiectii in coordonate ecran corespund aceluiasi pixel, vor fi desenate sau nu, in functie de valoarea salvata in matrice, adica de valoarea lui z a punctului in coordonate carteziene 3D – (x,y,z). Daca valoarea lui z este mai mare decat cea din matrice, inseamna ca noul punct este in fata, deci il va desena. Daca valoarea lui z este mai mica, inseamna ca punctul e in spate, deci nu il va desena.

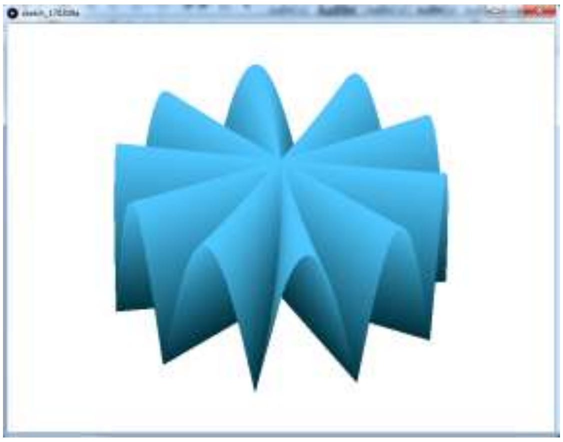
Aplicatie 2

Vom realiza aceeasi aplicatie cu ajutorul matricilor.

Program

```
size(800,600);
background(255);
float[][] mat=new float[800][600];
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
float x,y,z,r,u,X,Y,d=200,cx=400,cy=400,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B,c;
for(r=0;r<=1;r+=0.1/d)
  for(u=0;u<=2*PI;u+=0.1/d)
  {
    z=abs(sin(5*u));
```

```
   x=r*cos(u);
   y=r*sin(u);
   X=-0.86*x+0.86*y;
   Y=z-0.5*x-0.5*y;
   c=z;
   R=R1*(1-c)+R2*c;
   G=G1*(1-c)+G2*c;
   B=B1*(1-c)+B2*c;
   if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
   {
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
    mat[int(X*d+cx)][int(-Y*d+cy)]=z;
   }
  }
```



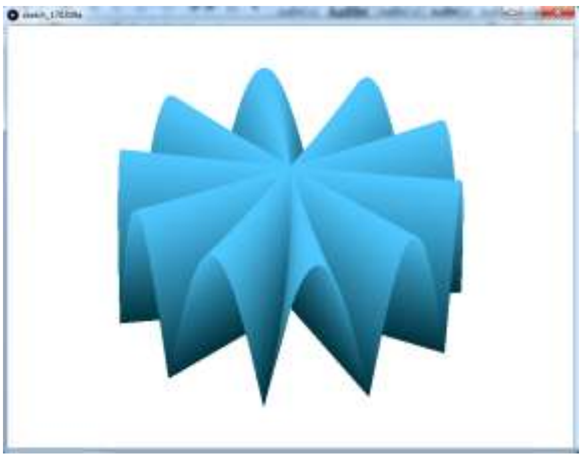
Aplicatie 3

Vom realiza aceeasi aplicatie, cu ajutorul subprogramelor.

Program

```
float[][] mat=new float[800][600];
void init_mat()
{
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
}
float f(float x)
{
  return abs(sin(5*x));
```

```
}
void setup()
{
size(800,600);
background(255);
init_mat();
float x,y,z,r,u,X,Y,d=200,cx=400,cy=400,
R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B,c;
for(r=0;r<=1;r+=0.1/d)
  for(u=0;u<=2*PI;u+=0.1/d)
  {
    z=f(u);
    x=r*cos(u);
    y=r*sin(u);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=z;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
    }
  }
}
```

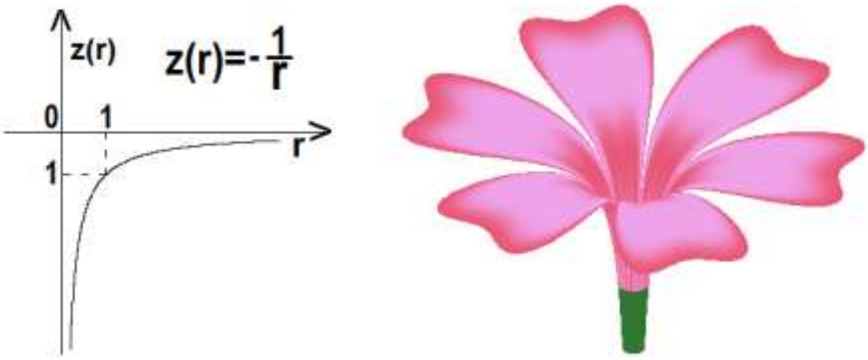


Roxana Draganoiu

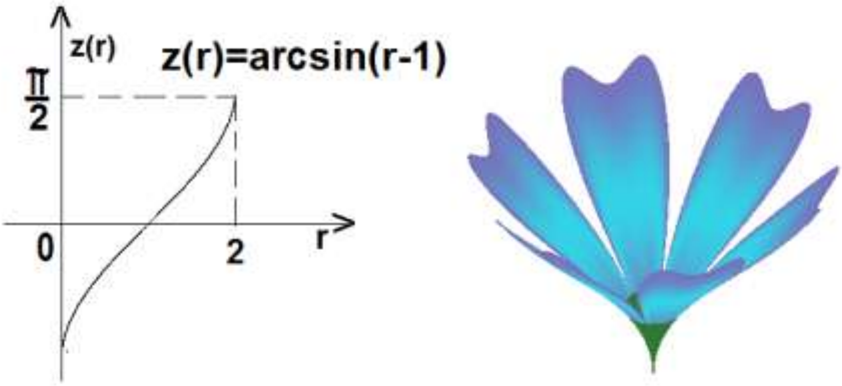
Modelarea florilor in 3D

Modelarea florilor in 3d se face cu functii in coordonate polare, datorita simetriei formei florii fata centru. Astfel, z=z(u,r).

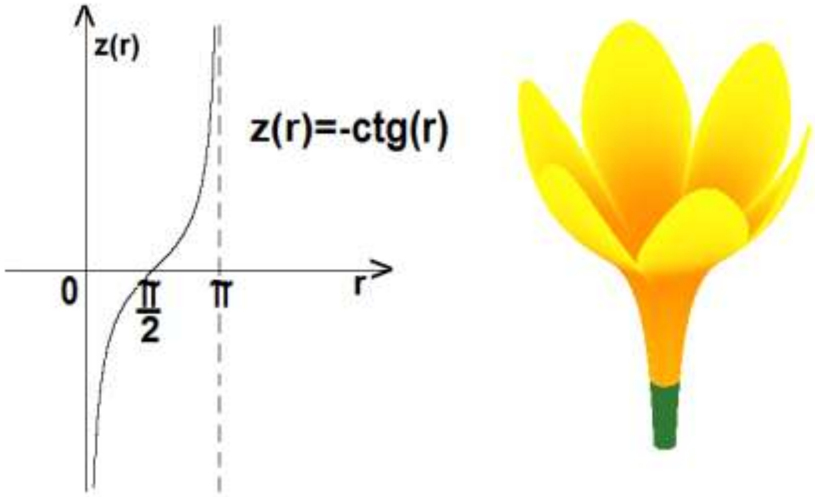
Vom studia mai intai cazul z=z(r). Pentru a modela o floare in 3D, trebuie in primul rand sa alegem o functie pentru forma dorita a florii vazuta din profil. Spre exemplu, daca dorim ca floarea sa aiba codita lunga si dreapta, trebuie sa ne gandim la o functie care are asimptota verticala in origine, cum este functia inversa, sau cotangenta.



$$r=r(u)$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$z=-\frac{1}{r}$$



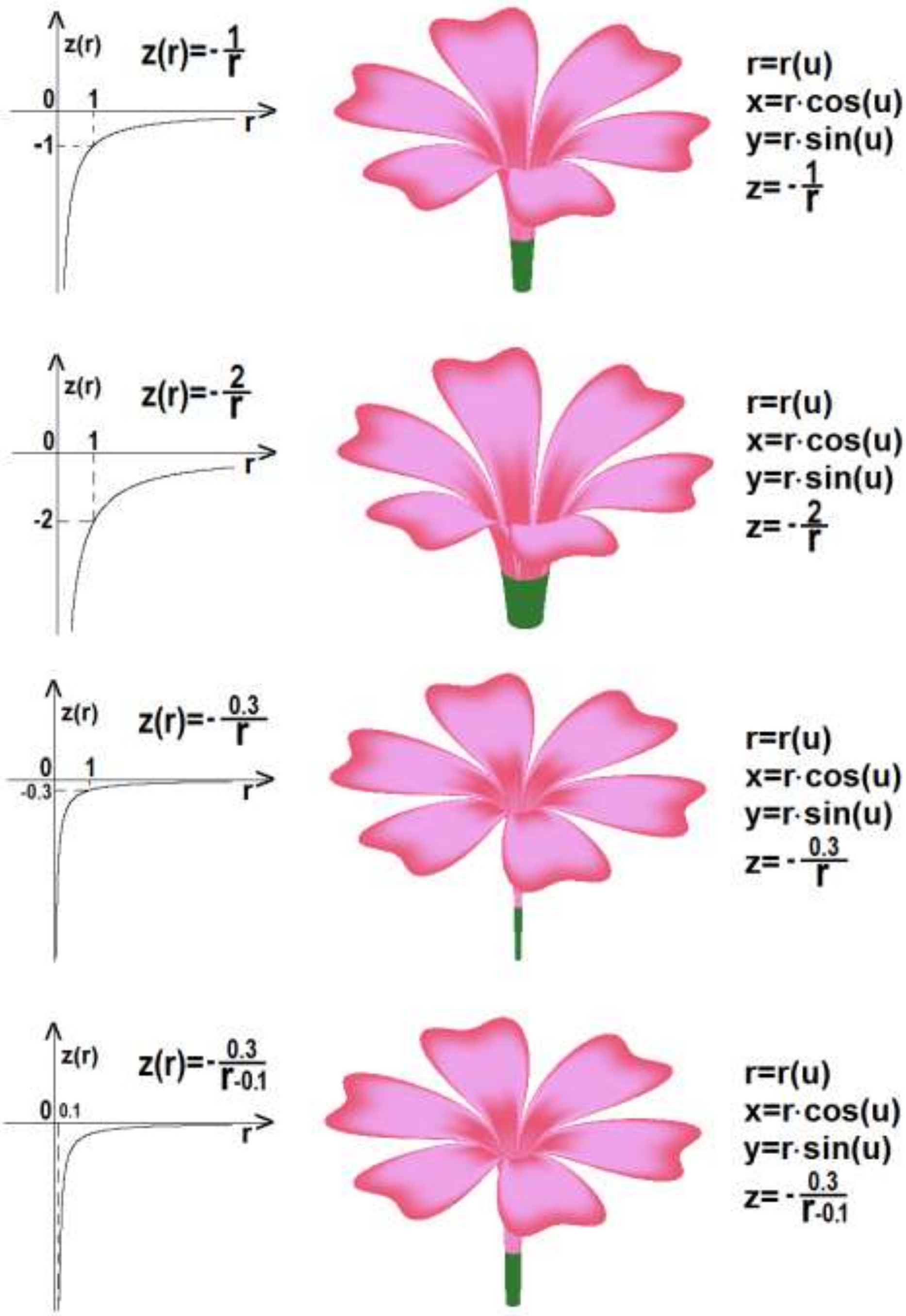
$$r=r(u)$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$z=\arcsin(r-1)$$



$$r=r(u)$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$z=-ctg(r)$$

O data forma aleasa, ea se poate modela cu o precizie mai mare prin modificarea diferitilor parametrii pe care ii adaugam in functie de ce anume dorim sa modelam. In cazul in care dorim sa facem modelare pa baza functiei inverse, spre exemplu, functia ar fi de forma z=a/(r-b). Rolul parametrilor fiind dupa cum urmeaza:

- parametrul a trebuie in primul rand sa fie un numar negativ, ca sa ne iasa codita in jos. In al doilea rand, putem regla cu el inclinatia petalelor, care e insa corelata de grosimea coditei.
- de parametrul b depinde doar grosimea coditei, ceea ce inseamna ca putem contracara efectul modificarii lui a.

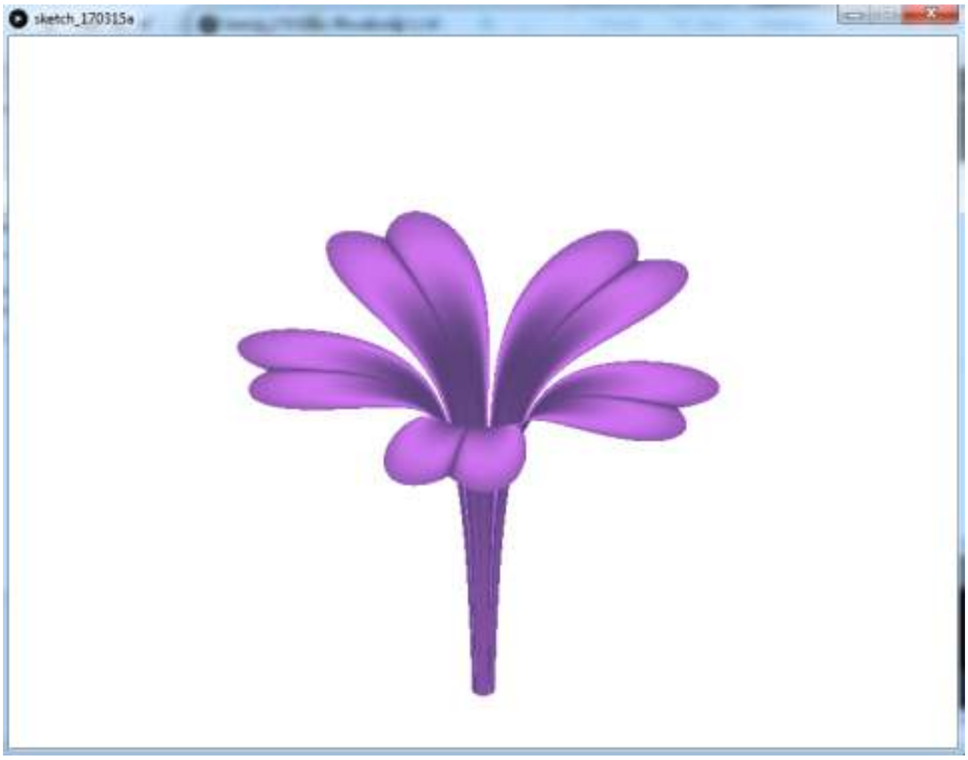


Aplicatie

Vom transpune in 3d floarea cu petalele in forma de inima. Lasam ca exercitiu testarea diferitelor functii pentru z.

<u>Program</u>

```
size(800,600);
background(255);
float[][] mat=new float[800][600];
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
float x,y,z,r,a,u,X,Y,d=150,cx=400,cy=250,pi=3.14,
R1=207,G1=116,B1=241,R2=92,G2=67,B2=124,R,G,B,c,c1;
for(u=0;u<=2*pi;u+=0.1/d)
  for(a=0;a<=1;a+=0.2/d)
  {
    r=a*(abs(sin(2.5*u))+0.3*abs(sin(5*u)));
    x=r*cos(u);
    y=r*sin(u);
    z=-0.1/(r);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=(sin(2*pi*a)+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    c1=pow(1-abs(cos(2.5*u)),20);
    R=R*(1-c1)+R2*c1;
    G=G*(1-c1)+G2*c1;
    B=B*(1-c1)+B2*c1;
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600&&z>-2)
    if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
    }
  }
```



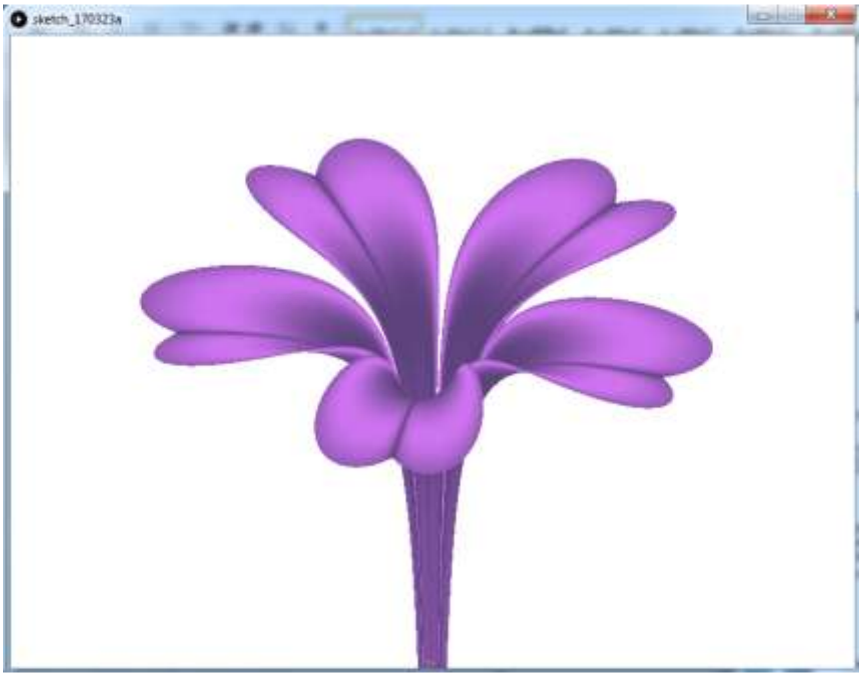
Aplicatie

Vom curba petalele adaugind la z si o componenta de forma z=f(u).

<u>*Program*</u>

```
size(800,600);
background(255);
float[][] mat=new float[800][600];
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
float x,y,z,r,a,u,X,Y,d=200,cx=400,cy=200,pi=3.14,
R1=207,G1=116,B1=241,R2=92,G2=67,B2=124,R,G,B,c,c1;
for(u=0;u<=2*pi;u+=0.1/d)
  for(a=0;a<=1;a+=0.2/d)
  {
    r=a*(abs(sin(2.5*u))+0.3*abs(sin(5*u)));
    x=r*cos(u);
```

```
    y=r*sin(u);
    z=-0.1/(r)-0.2*abs(sin(2.5*u));
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=(sin(2*pi*a)+1)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    c1=pow(1-abs(cos(2.5*u)),20);
    R=R*(1-c1)+R2*c1;
    G=G*(1-c1)+G2*c1;
    B=B*(1-c1)+B2*c1;
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600&&z>-2)
    if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
    }
  }
```

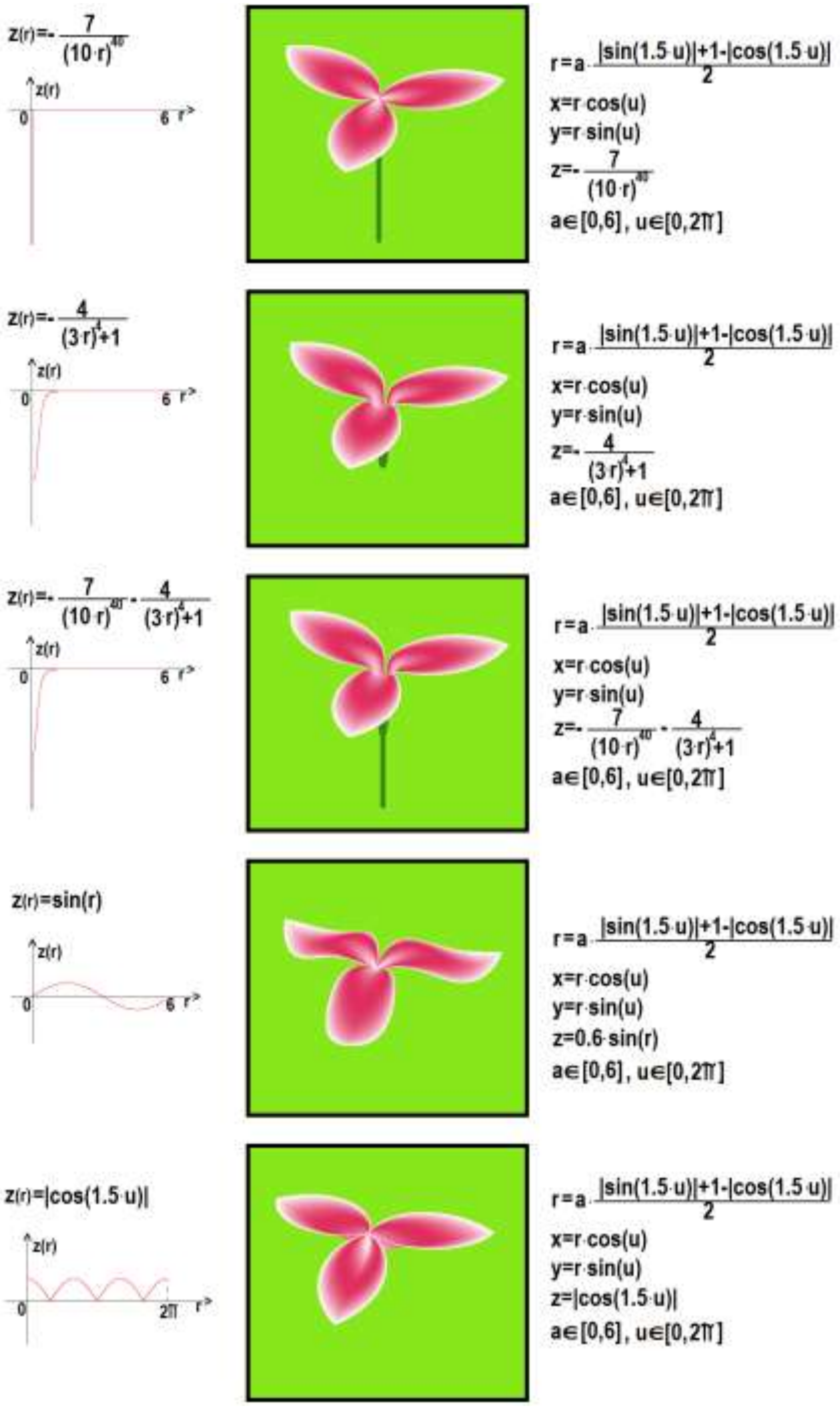


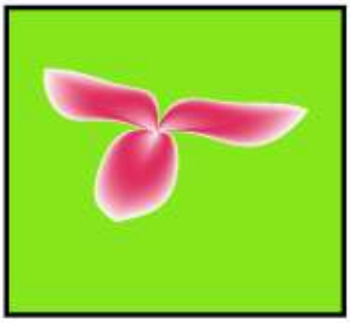
Crinul

Aplicatie

Vom realiza un crin pe baza metodelor studiate.

Rezolvare matematica



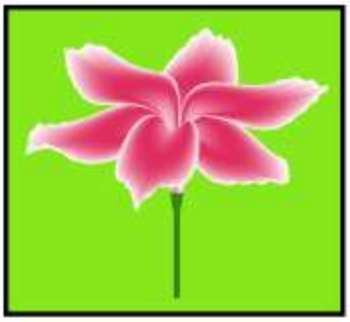


$$r = a \cdot \frac{|\sin(1.5 \cdot u)| + 1 - |\cos(1.5 \cdot u)|}{2}$$
$x = r \cdot \cos(u)$
$y = r \cdot \sin(u)$
$z = 0.6 \cdot \sin(r) \cdot |\cos(1.5 \cdot u)|$
$a \in [0,6]$, $u \in [0, 2\pi]$

$$r = a \cdot \frac{|\sin(1.5 \cdot u)| + 1 - |\cos(1.5 \cdot u)|}{2}$$
$x = r \cdot \cos(u)$
$y = r \cdot \sin(u)$
$z = 1.5 \cdot \left(\frac{r}{6}\right)^6 \cdot |\sin(6 \cdot u)|$
$a \in [0,6]$, $u \in [0, 2\pi]$

$$r = a \cdot \frac{|\sin(1.5 \cdot u)| + 1 - |\cos(1.5 \cdot u)|}{2}$$
$x = r \cdot \cos(u)$
$y = r \cdot \sin(u)$
$z = 0.2 \cdot \left(\frac{r}{6}\right)^6 \cdot \left(0.7 \cdot \sin(10 \cdot u) + \sin(23 \cdot u) + 7 \cdot |\sin(6 \cdot u)|\right)$
$a \in [0,6]$, $u \in [0, 2\pi]$

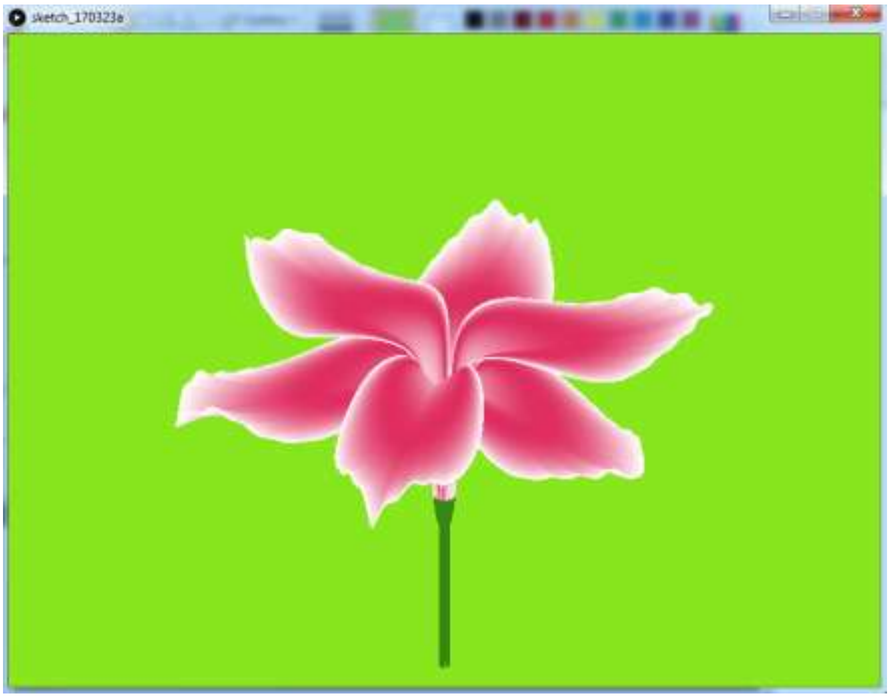
$$r = a \cdot \frac{|\sin(1.5 \cdot u)| + 1 - |\cos(1.5 \cdot u)|}{2}$$
$x = r \cdot \cos(u)$, $y = r \cdot \sin(u)$
$$z = -\frac{7}{(10 \cdot r)^{40}} - \frac{4}{(3 \cdot r)^4 + 1} + 0.6 \cdot \sin(r) \cdot |\cos(1.5 \cdot u)| +$$
$$0.2 \cdot \left(\frac{r}{6}\right)^6 \cdot \left(0.7 \cdot \sin(10 \cdot u) + \sin(23 \cdot u) + 7 \cdot |\sin(6 \cdot u)|\right)$$
$a \in [0,6]$, $u \in [0, 2\pi]$



$$r = a \cdot \frac{|\sin(1.5 \cdot u)| + 1 - |\cos(1.5 \cdot u)|}{2}$$
$x = r \cdot \cos\left(u + \frac{\pi}{3}\left[\frac{u}{2\pi}\right]\right)$, $y = r \cdot \sin\left(u + \frac{\pi}{3}\left[\frac{u}{2\pi}\right]\right)$
$$z = -\frac{7}{(10 \cdot r)^{40}} - \frac{4}{(3 \cdot r)^4 + 1} + 0.6 \cdot \sin(r) \cdot |\cos(1.5 \cdot u)| +$$
$$0.2 \cdot \left(\frac{r}{6}\right)^6 \cdot \left(0.7 \cdot \sin(10 \cdot u) + \sin(23 \cdot u) + 7 \cdot |\sin(6 \cdot u)|\right) -$$
$$0.35 \cdot \left[\frac{u}{2\pi}\right]$$
$a \in [0,6]$, $u \in [0, 2\pi]$

Program

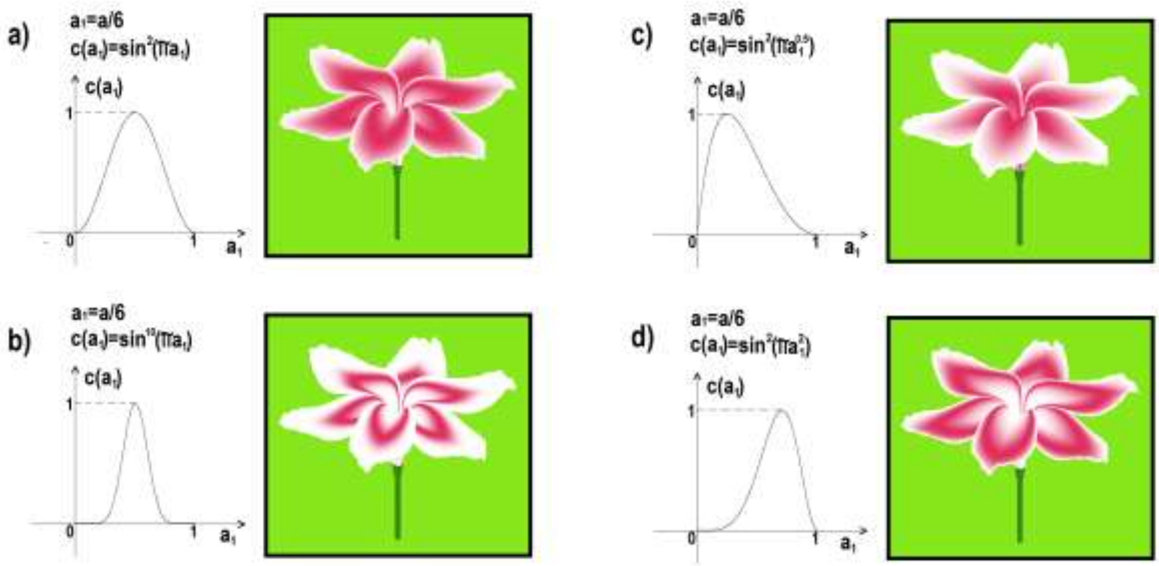
```
size(800,600);
background(135,230,30);
float[][] mat=new float[800][600];
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
float x,y,z,d=35,r,a,u,X,Y,cx=400,cy=300,
R,G,B,R1=255,G1=255,B1=255,R2=223,G2=46,B2=95,c;
for(u=0;u<4*PI;u+=0.05/d)
  for(a=0;a<=6;a+=0.5/d)
  {
    r=a*(abs(sin(u*1.5))+1-abs(cos(1.5*u)))/2;
    x=r*cos(u+PI/3*int(u/(2*PI)));
    y=r*sin(u+PI/3*int(u/(2*PI)));
    z=0.6*sin(r)*abs(cos(u*1.5))+0.2*pow(r/6,6)*(0.7*sin(u*70)
      +1*sin(23*u)+7*abs(sin(6*u)))-7/(pow(r*10,40))-4/(pow(r*3,4)+1)
      -0.35*int(u/(2*PI));
    X=0.86*x-0.86*y;
    Y=z-0.5*x-0.5*y;
    c=sin(PI*(a/6));
    R=c*R2+(1-c)*R1;
    G=c*G2+(1-c)*G1;
    B=c*B2+(1-c)*B1;
    if(z<-3.5)
      {R=57;G=134;B=21;}
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600&&z>-8)
    if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
    }
  }
```



Aplicatie

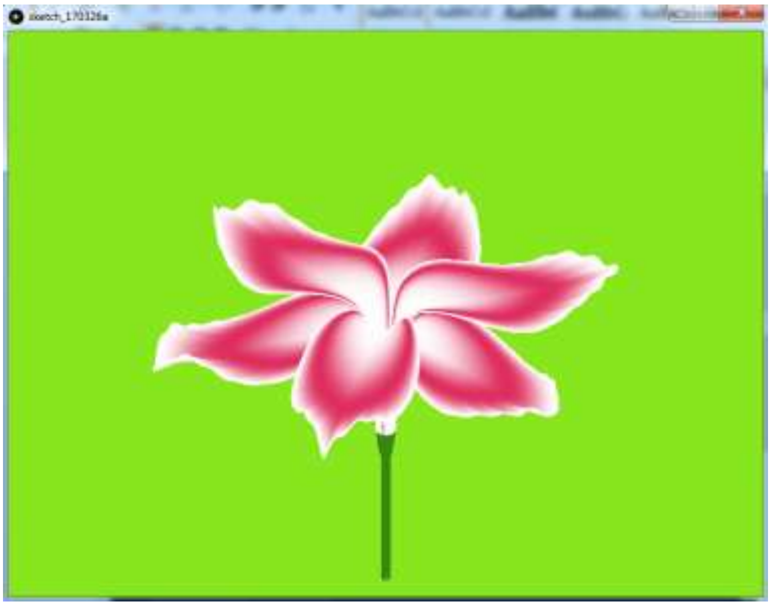
Vom studia efectul grafic pentru diferite functii de interpolare obtinute prin compunere de functii, aplicat degradeului utilizat la crinul din aplicatia precedenta. Mai exact, dorim sa vedem cum putem “strange” culoarea roz mai spre centrul petalei (fig a, b), iar apoi cum o putem decala mai spre mai spre interiorul florii (fig c) sau marginea petalelor (fig d).

Rezolvare matematica



Program

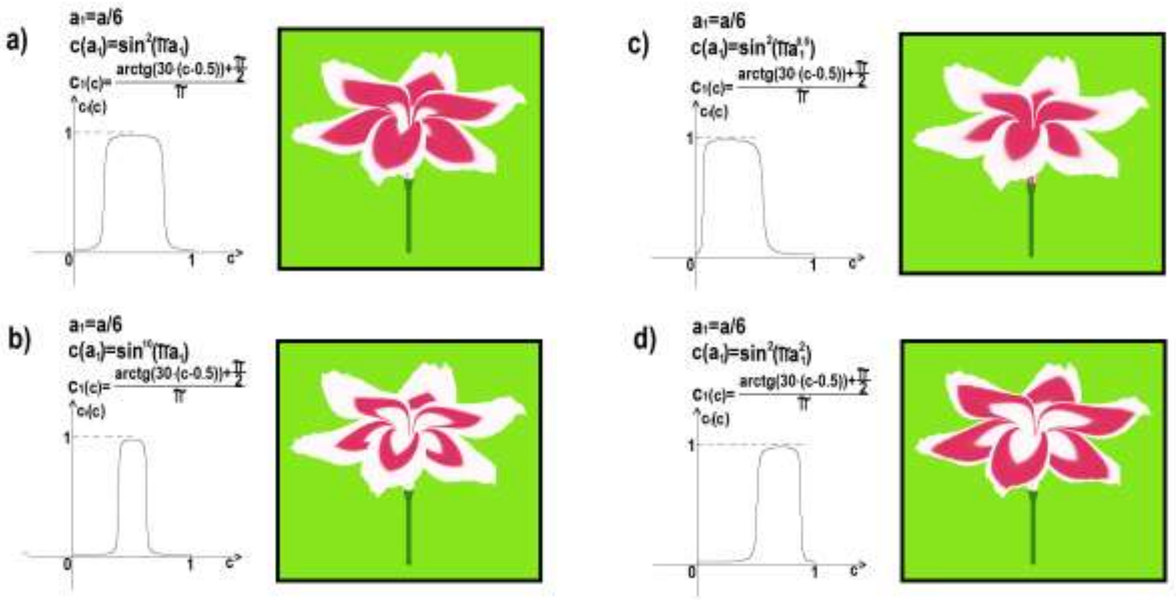
```
size(800,600);
background(135,230,30);
float[][] mat=new float[800][600];
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
float x,y,z,d=35,r,a,a1,u,X,Y,cx=400,cy=300,
R,G,B,R1=255,G1=255,B1=255,R2=223,G2=46,B2=95,c;
for(u=0;u<4*PI;u+=0.05/d)
  for(a=0;a<=6;a+=0.5/d)
  {
    r=a*(abs(sin(u*1.5))+1-abs(cos(1.5*u)))/2;
    x=r*cos(u+PI/3*int(u/(2*PI)));
    y=r*sin(u+PI/3*int(u/(2*PI)));
    z=0.6*sin(r)*abs(cos(u*1.5))+0.2*pow(r/6,6)*(0.7*sin(u*70)
      +1*sin(23*u)+7*abs(sin(6*u)))-7/(pow(r*10,40))-4/(pow(r*3,4)+1)
      -0.35*int(u/(2*PI));
    X=0.86*x-0.86*y;
    Y=z-0.5*x-0.5*y;
    a1=a/6;
    c=pow(sin(PI*pow(a1,2)),2);
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    if(z<-3.5)
      {R=57;G=134;B=21;}
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600&&z>-8)
    if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
    }
  }
```



Aplcatie

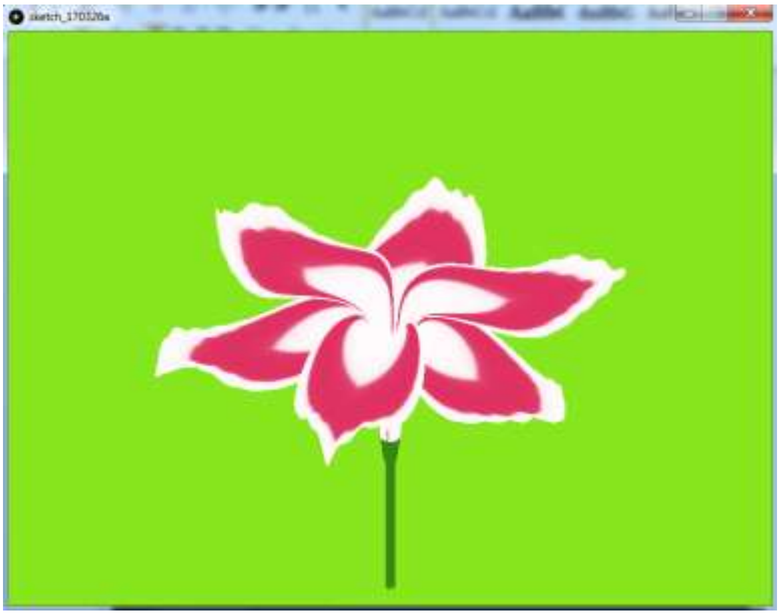
Vom face un studiu de conturare a culorilor cu ajutorul "functiei treapta" pentru degradeurile din aplicatia precedenta. Mai exact, vom compune functia treapta cu functiile de interpolare.

<u>*Rezolvare matematica*</u>



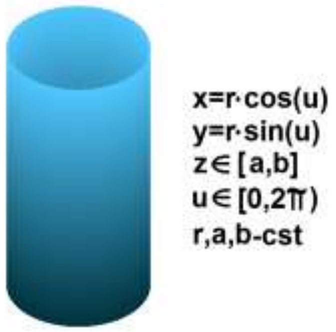
Program

```
size(800,600);
background(135,230,30);
float[][] mat=new float[800][600];
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
float x,y,z,d=35,r,a,a1,u,X,Y,cx=400,cy=300,
R,G,B,R1=255,G1=255,B1=255,R2=223,G2=46,B2=95,c,c1;
for(u=0;u<4*PI;u+=0.05/d)
  for(a=0;a<=6;a+=0.5/d)
  {
    r=a*(abs(sin(u*1.5))+1-abs(cos(1.5*u)))/2;
    x=r*cos(u+PI/3*int(u/(2*PI)));
    y=r*sin(u+PI/3*int(u/(2*PI)));
    z=0.6*sin(r)*abs(cos(u*1.5))+0.2*pow(r/6,6)*(0.7*sin(u*70)
      +1*sin(23*u)+7*abs(sin(6*u)))-7/(pow(r*10,40))-4/(pow(r*3,4)+1)
      -0.35*int(u/(2*PI));
    X=0.86*x-0.86*y;
    Y=z-0.5*x-0.5*y;
    a1=a/6;
    c=pow(sin(PI*pow(a1,2)),2);
    c1=(atan(30*(c-0.5))+PI/2)/PI;
    R=R1*(1-c1)+R2*c1;
    G=G1*(1-c1)+G2*c1;
    B=B1*(1-c1)+B2*c1;
    if(z<-3.5)
      {R=57;G=134;B=21;}
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-Y*d+cy)<600&&z>-
8)
    if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
    }
  }
```



Cilindrul

Ecuatiile parametrice



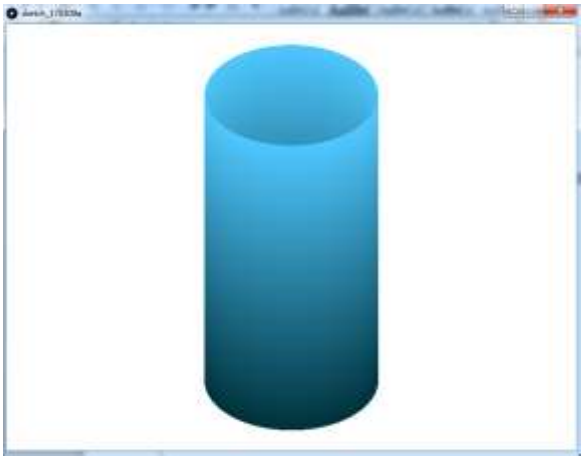
Reprezentarea grafica a cilindrului in programare cu ajutorul ecuatiilor parametrice este similara cu reprezentarea cercului, cu deosebire ca mai adaugam o bucla repetitiva *for*.

Aplicatie

Vom reprezenta grafic un cilindru.

Program

```
size(800,600);
background(255);
float x,y,z,u,r=1,X,Y,d=100,cx=400,cy=500,
c,R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B;
for(z=0;z<=4;z+=0.5/d)
  for(u=0;u<=2*PI;u+=0.1/d)
  {
    x=r*cos(u);
    y=r*sin(u);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=z/4;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
  }
```



Modelare pe baza ecuatiilor parametrice ale cilindrului

O metoda de modelare pa baza ecuatiilor parametrice ale cilindrului, presupune ca raza sa varieze in functie de *z* si *u*, adica r=r(z,u).
Pentru obtinerea formei dorite, trebuie sa ne imaginam suprafata cilindrului desfasurata.

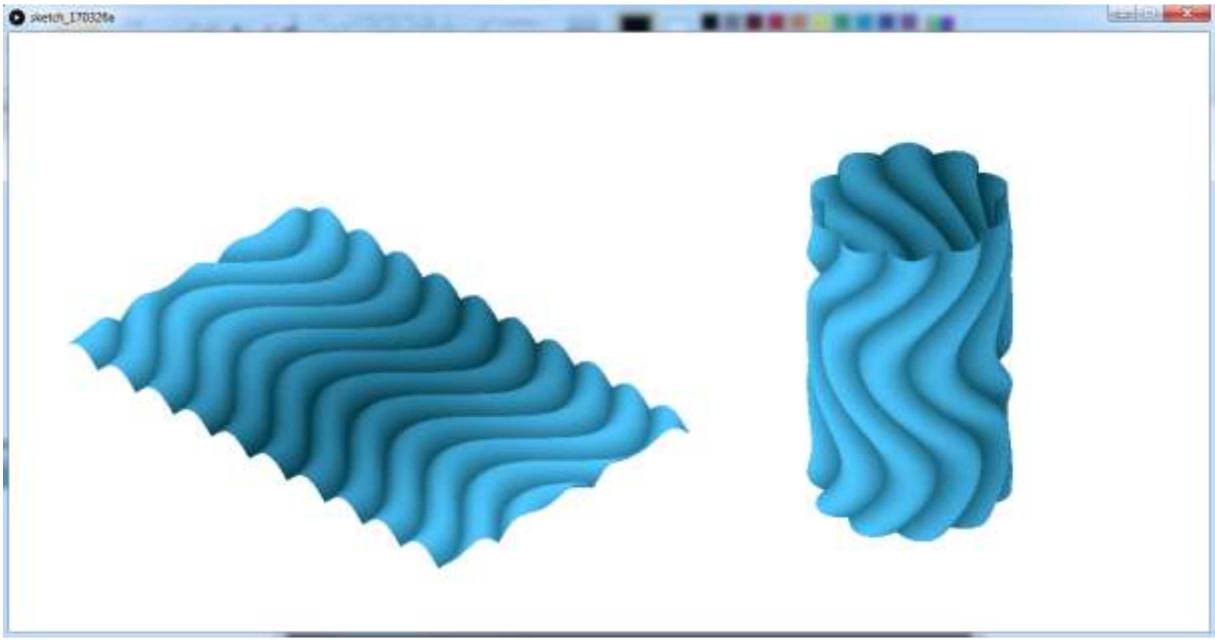
Aplicatie

Vom reprezenta grafic un cilindru modelat si desfasurarea acestuia, adica functia r(z,u).

Programare

```
size(1200,600);
background(255);
float x,y,z,u,r,X,Y,d=70,cx=600,cy=450,
c,R1=0,G1=50,B1=60,R2=80,G2=200,B2=255,R,G,B;
for(z=0;z<=4;z+=0.5/d)
  for(u=0;u<=2*PI;u+=0.1/d)
  {
    r=1+0.2*abs(sin(5*(u-0.5*sin(2*z))));
    x=r*cos(u);
    y=r*sin(u);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=((x+1.2)/2.4+(r-1)/0.2)/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
```

```
  set(int(X*d+cx+300),int(-Y*d+cy),color(R,G,B));
  //r(z,u)
  X=-0.86*z+0.86*u;
  Y=r-0.5*z-0.5*u;
  set(int(X*d+cx-300),int(-Y*d+cy-200),color(R,G,B));
}
```



Brandusa

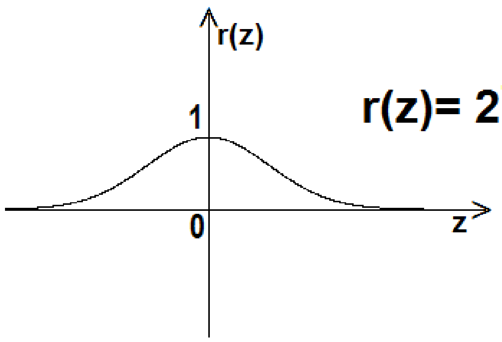
Aplicatie

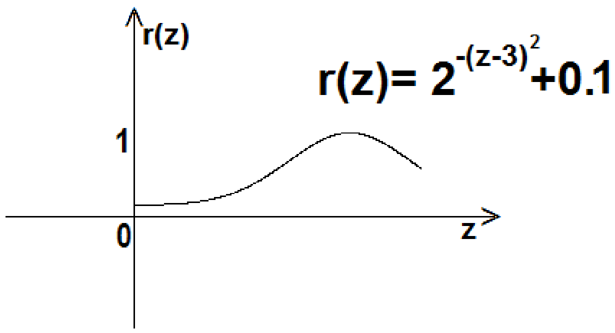
Vom realiza o brandusa.

<u>*Rezolvare matematica*</u>

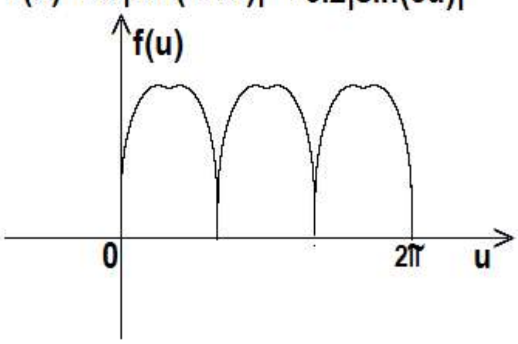
Pentru forma vazuta din profil, adica componenta r(z) a razei, vom alege o functie de tip Gauss. Se poate demonstra usor ca prin reorganizarea parametrilor, functia aleasa este o functie Gauss. Acest lucru nu este relevant insa, decat ca sa ne putem referi la functia aleasa intr-un fel. Asa ca o vom numi *functie Gauss*.

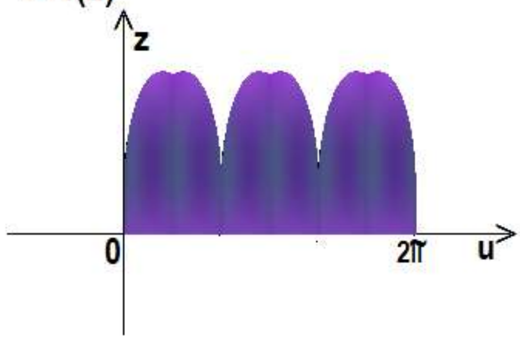
$$f(x)=a\cdot 2^{-(b\cdot(x-c))^2} \quad \overset{\text{Gauss}}{\longleftrightarrow} \quad f(x)=a_1\cdot e^{-b_1\cdot(x-c_1)^2}$$

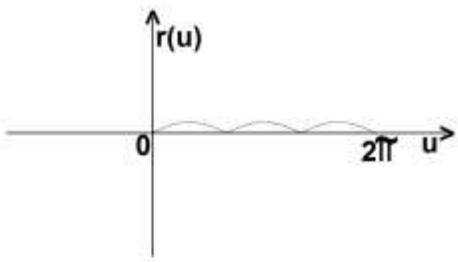




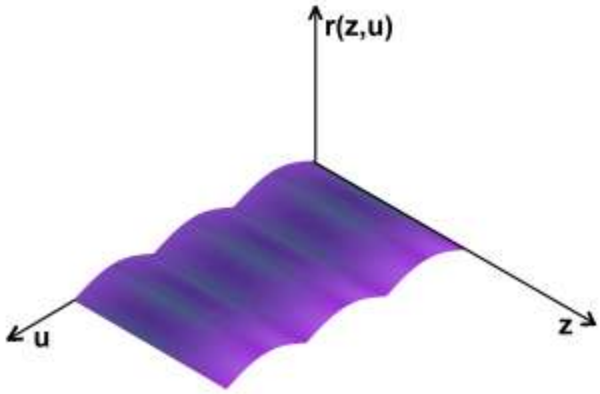
Pentru decuparea petalelor vom utiliza inecuatiile.



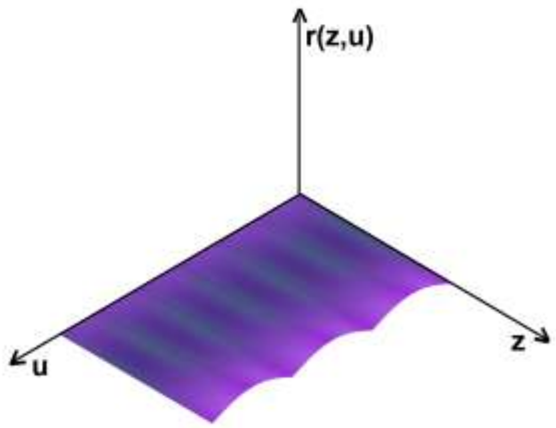




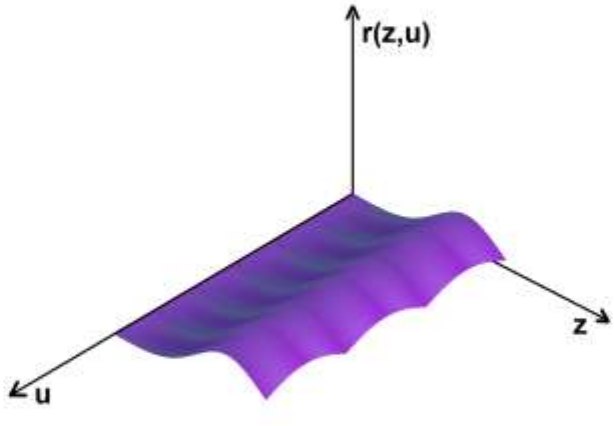
$$r(u)=0.3\cdot|\sin(1.5\cdot u)|$$



$$r(z,u)=0.3\cdot|\sin(1.5\cdot u)|$$



$$r(z,u)=0.3\cdot\left(\frac{z}{4}\right)^3\cdot|\sin(1.5\cdot u)|$$



$$r(z,u)=0.3\cdot\left(\frac{z}{4}\right)^3\cdot|\sin(1.5\cdot u)| + 2^{-(z-3)^2}+0.07$$

Ecuatiile parametrice ale cilindrului

$x=r\cdot\cos(u)$
$y=r\cdot\sin(u)$
$z\in[a,b]$
$u\in[0,2\pi)$
r,a,b-cst

$x=r(z)\cdot\cos(u)$
$y=r(z)\cdot\sin(u)$
$z\in[a,b]$
$u\in[0,2\pi)$
a,b-cst

$x=r(z,u)\cdot\cos(u)$
$y=r(z,u)\cdot\sin(u)$
$z\in[a,b]$
$u\in[0,2\pi)$
r,a,b-cst

$x=r(z,u)\cdot\cos(u)$
$y=r(z,u)\cdot\sin(u)$
$z<f(u)$
$u\in[0,2\pi)$
a,b-cst

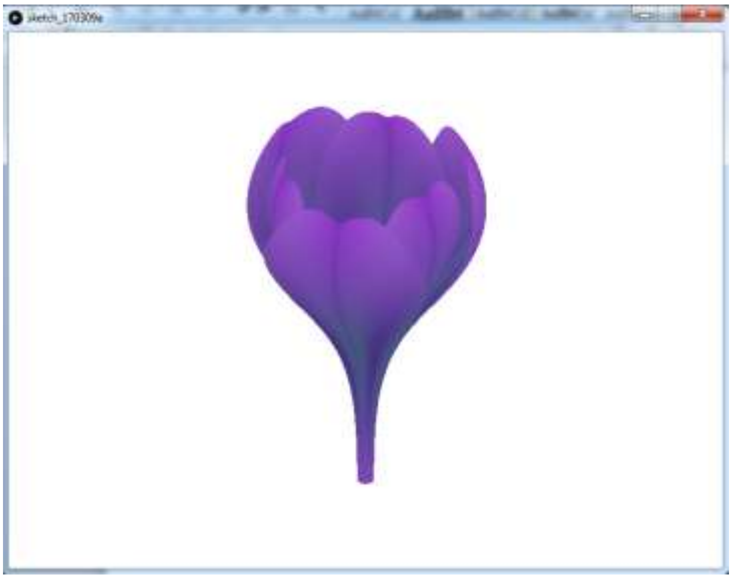
$x=r(z,u)\cdot\cos(u-\pi)$
$y=r(z,u)\cdot\sin(u-\pi)$
$z<f(u)$
$u\in[0,2\pi)$
a,b-cst

$x=r(z,u)\cdot\cos(u-\pi\cdot[\frac{u}{2\pi}])$
$y=r(z,u)\cdot\sin(u-\pi\cdot[\frac{u}{2\pi}])$
$z<f(u)$
$u\in[0,2\pi)$
a,b-cst

$x=(1+c\cdot[\frac{u}{2\pi}])\cdot r(z,u)\cdot\cos(u-\pi\cdot[\frac{u}{2\pi}])$
$y=(1+c\cdot[\frac{u}{2\pi}])\cdot r(z,u)\cdot\sin(u-\pi\cdot[\frac{u}{2\pi}])$
$z<f(u)$
$u\in[0,2\pi)$
a,b-cst

Program

```
size(800,600);
background(255);
float x,y,z,r,u,f,X,Y,pi=3.1415,d=100,cx=400,cy=500,
c,c1,c2,R1=140,G1=70,B1=215,R2=193,G2=145,B2=242,
R3=70,G3=10,B3=130,R,G,B;
for(z=0;z<=4;z+=0.5/d)
  for(u=0;u<4*pi;u+=0.4/d)
  {
    r=(pow(2,-pow(z-3,2)))+0.07+0.1*pow(z/4,3)*abs(sin(1.5*u));
    x=(1-0.2*int(u/(2*pi)))*r*cos(u-pi*int(u/(2*pi)));
    y=(1-0.2*int(u/(2*pi)))*r*sin(u-pi*int(u/(2*pi)));
    X=0.86*x-0.86*y;
    Y=z-0.5*x-0.5*y;
    f=3.5*pow(abs(sin(1.5*u)),0.25)+0.2*abs(sin(2*1.5*u));
    if(z<f)
    {
      c=pow(1-abs(sin(3*(u+PI/6))),0.1);
      R=R1*(1-c)+R2*c;
      G=G1*(1-c)+G2*c;
      B=B1*(1-c)+B2*c;
      c1=(sin(z)+1)/2;
      c2=(sin(z)*0.4*(sin(6*(u-PI/12)))+1)/2;
      R=R*(1-c1)+R3*c1;
      G=G*(1-c2)+G3*c2;
      B=B*(1-c1)+B3*c1;
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
    }
   }
```

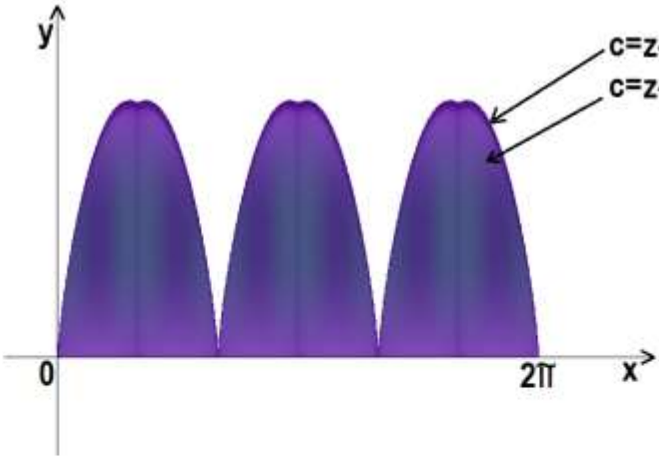


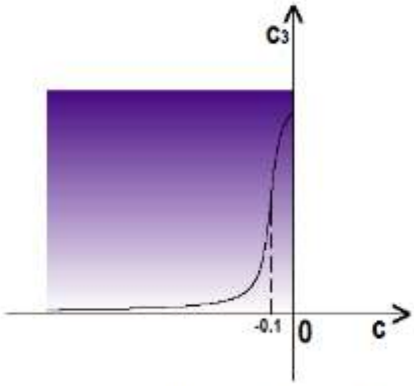
Aplicatie

Vom contura petalele.

Rezolvare matematica

Pentru a contura petalele vom folosi semnul expresiei z-f, pe care o vom nota cu c. Cum z=f pe conturul petalelor si z<f in interiorul petalelor, obtinem ca c=0 pe contur si c<0 in interior. Vom aplica functia treapta variabilei c cu parametrul *alpha*=30 si *beta*=-0.1.

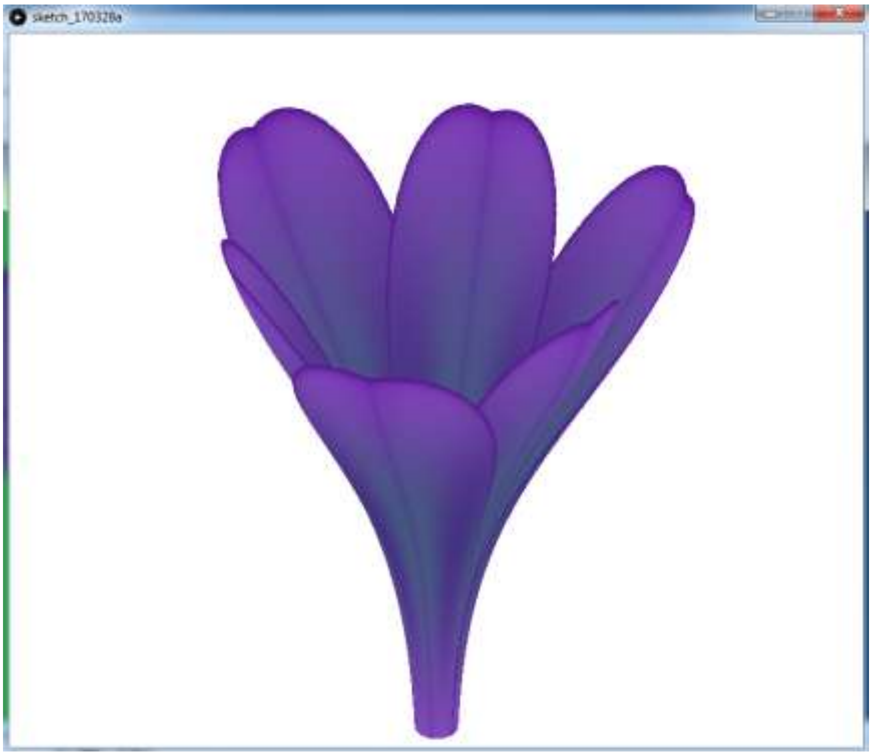




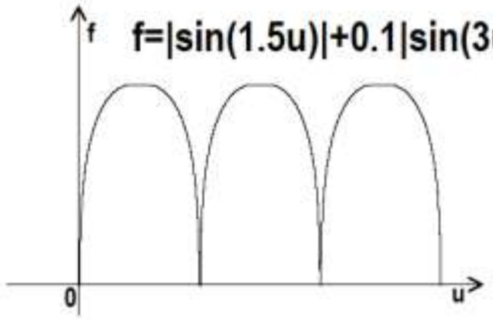
Program

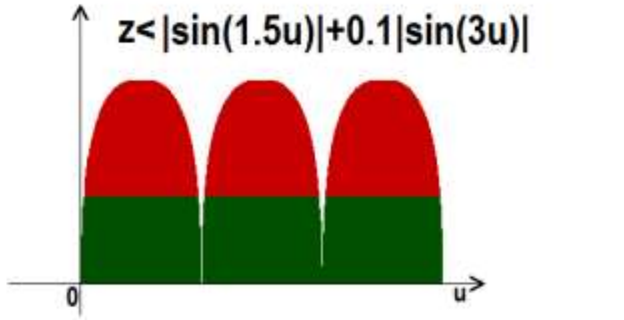
```
size(800,670);
background(255);
float x,y,z,r,u,X,Y,pi=3.1415,d=80,cx=100,cy=350,f,
c,c1,c2,c3,R1=140,G1=70,B1=215,R2=193,G2=145,B2=242,
R3=70,G3=10,B3=130,R,G,B;
for(z=0;z<=4;z+=0.5/d)
  for(u=0;u<4*pi;u+=0.4/d)
  {
    r=1.5*(pow(2,-pow(z-3.3,2)/2)*0.5)+0.01*pow(z,3)*abs(sin(1.5*u))+0.08;
    x=(1.2-0.2*int(u/(2*pi)))*r*cos(u-pi*int(u/(2*pi)));
    y=(1.2-0.2*int(u/(2*pi)))*r*sin(u-pi*int(u/(2*pi)));
    X=0.86*x-0.86*y;
    Y=z-0.5*x-0.5*y;
    f=3.3*pow(abs(sin(1.5*u)),0.7)+0.2*abs(sin(2*1.5*u));
    if(z<f)
    {
      c=pow(1-abs(sin(3*(u+PI/6))),0.1);
      R=R1*(1-c)+R2*c;
```

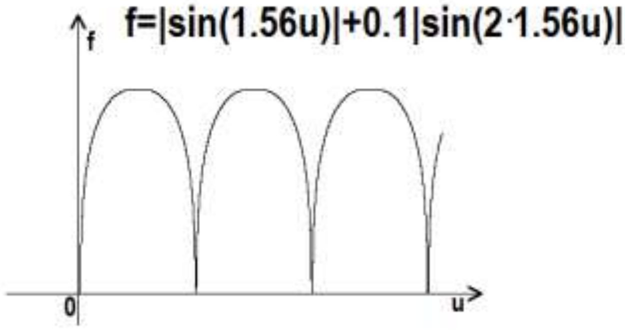
```
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    c1=(sin(z)+1)/2;
    c2=(sin(z)*0.4*(sin(6*(u-PI/12)))+1)/2;
    R=R*(1-c1)+R3*c1;
    G=G*(1-c2)+G3*c2;
    B=B*(1-c1)+B3*c1;
    c3=(atan(30*(z-f+0.1))+PI/2)/PI;
    R=R*(1-c3)+R3*c3;
    G=G*(1-c3)+G3*c3;
    B=B*(1-c3)+B3*c3;
    set(int(u*d+cx),int(-z*d+cy),color(R,G,B));
 }
}
```

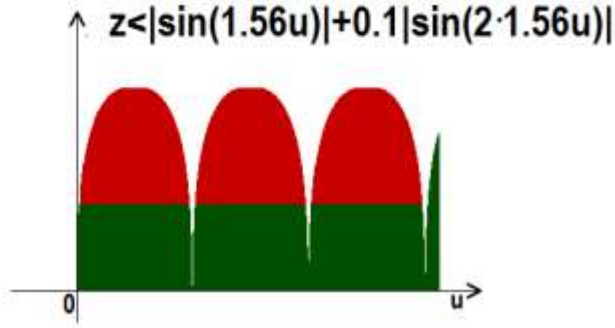


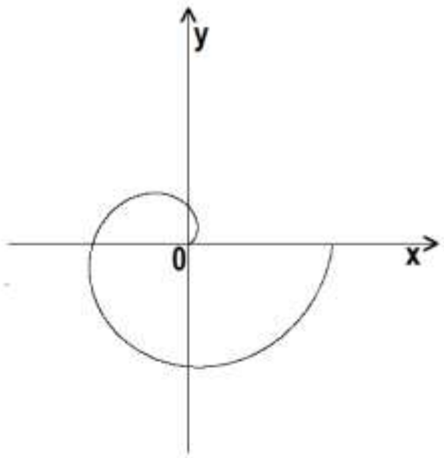
Trandafirul



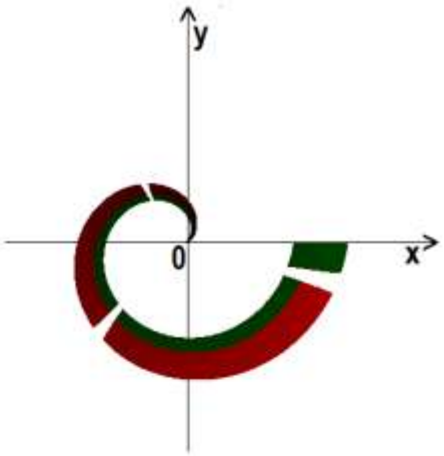


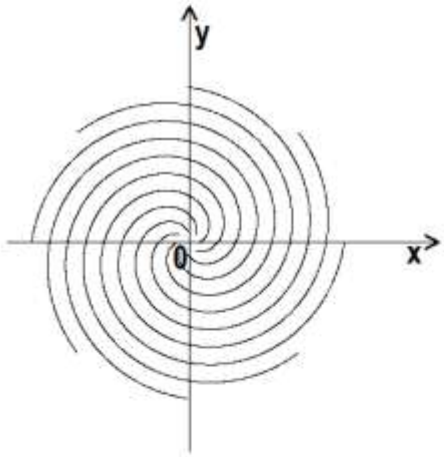




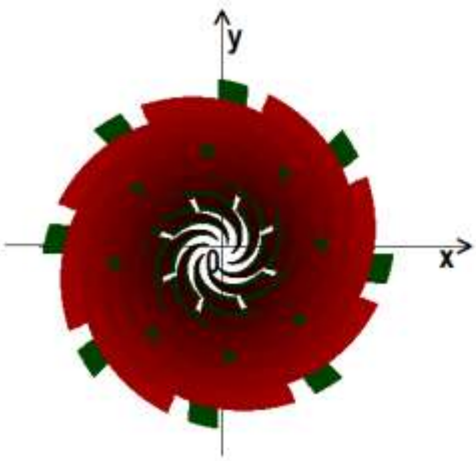


$r=0.2(u+0.1)+0.08$
$x=r\cdot\cos(u)$
$y=r\cdot\sin(u)$





$r=0.2(u+0.1)+0.08$
$x=r\cdot\cos(u+i\cdot\frac{2\pi}{8})$
$y=r\cdot\sin(u+i\cdot\frac{2\pi}{8})$



$$r=2^{-(z-8)^2}+0.08$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$u\in[0,\,2\pi)$$
$$z\in[6,10]$$

$$r=2^{-(z-8)^2\cdot 2}+2\cdot 2^{-(z-10)^2}+0.08$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$u\in[0,\,2\pi)$$
$$z\in[6,10]$$

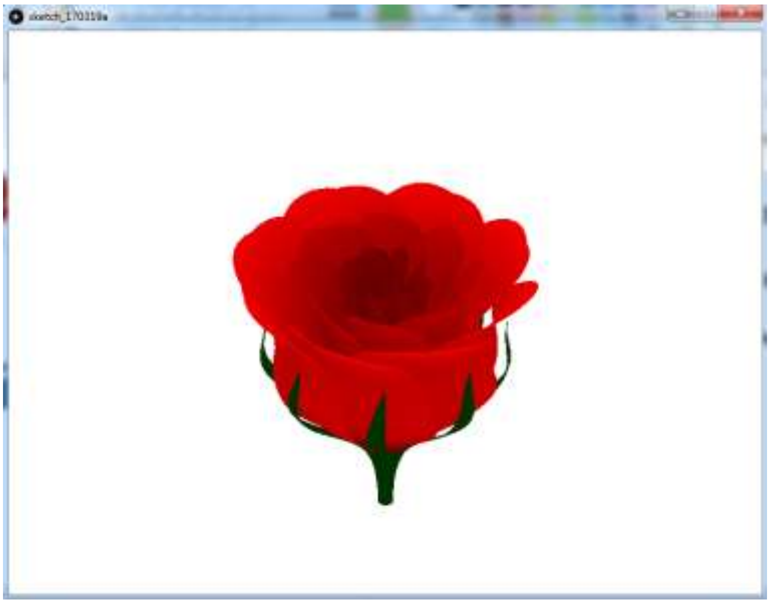
$$r=0.2(u+0.1)(2^{-(z-8)^2\cdot 2}+2\cdot 2^{-(z-10)^2})+0.08$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$u\in[0,\,2\pi)$$
$$z\in[6,10]$$

$$r=0.2(u+0.1)(2^{-(z-8)^2\cdot 2}+2\cdot 2^{-(z-10)^2})+0.08$$
$$x=r\cdot\cos(u) \qquad u\in[0,\,2\pi)$$
$$y=r\cdot\sin(u) \qquad z\in[6,10]$$
$$z<9.5\cdot(|\sin(1.56u)|+0.1|\sin(2\cdot 1.56u)|)^{0.1}$$

$$r=0.2(u+0.1)(2^{-(z-8)^2\cdot 2}+2\cdot 2^{-(z-10)^2})+0.08$$
$$x=r\cdot\cos\left(u+i\frac{2\pi}{8}\right) \qquad i\in\{0,1,\ldots,7\} \qquad u\in[0,\,2\pi)$$
$$y=r\cdot\sin\left(u+i\frac{2\pi}{8}\right) \qquad z\in[6,10]$$
$$z<9.5\cdot(|\sin(1.56u)|+0.1|\sin(2\cdot 1.56u)|)^{0.1}$$

Program

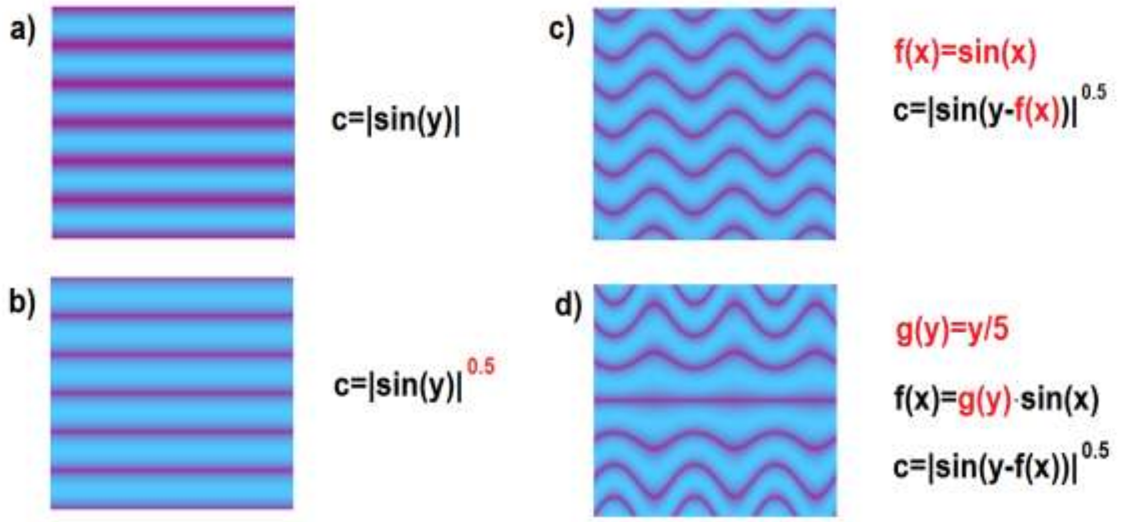
```
size(800,600);
background(255);
float x,y,z,r,u,X,Y,i,nr=8,f,d=70,cx=400,cy=500,
R1=80,G1=0,B1=0,R2=250,G2=0,B2=0,R3=0,G3=50,B3=0,
R4=185,G4=155,B4=0,R,G,B,c,c1;
for(z=0;z<=5;z+=0.5/d)
  for(u=0;u<=2*PI;u+=0.3/d)
    for(i=0;i<nr;i++)
    {
      r=0.2*(u+0.1)*(1*pow(2,-pow(z-2,2)*2)
        +2*pow(2,-pow(z-4,2)))+0.08;
      x=r*cos(u+i*2*PI/nr);
      y=r*sin(u+i*2*PI/nr);
      X=-0.86*x+0.86*y;
      Y=z-0.5*x-0.5*y;
      f=3.5*pow(abs(sin(1.56*u))+0.1*abs(sin(2*1.56*u)),0.3);
      if(z<f)
      {
        c=r/2;
        R=R1*(1-c)+R2*c;
        G=G1*(1-c)+G2*c;
        B=B1*(1-c)+B2*c;
        c=(atan(100*(u-3*PI/1.56))+PI/2)/PI;
        R=R*(1-c)+R3*c;
        G=G*(1-c)+G3*c;
        B=B*(1-c)+B3*c;
        c=(atan(100*(z-1))+PI/2)/PI;
        R=R3*(1-c)+R*c;
        G=G3*(1-c)+G*c;
        B=B3*(1-c)+B*c;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
    }
```



Texturi

$$\text{Red} = R1\cdot(1-c)+R2\cdot c$$
$$\text{Green} = G1\cdot(1-c)+G2\cdot c$$
$$\text{Blue} = B1\cdot(1-c)+B2\cdot c$$
$$c=c(x,y)\,, \quad c:\mathbb{R}^2\rightarrow[0,1]$$

Pentru realizarea texturilor, variabila de interpolare trebuie sa fie o functie de 2 variabile. Texturile se realizeaza prin aceleasi tehnici ca si modelarea functiilor de 2 variabile.

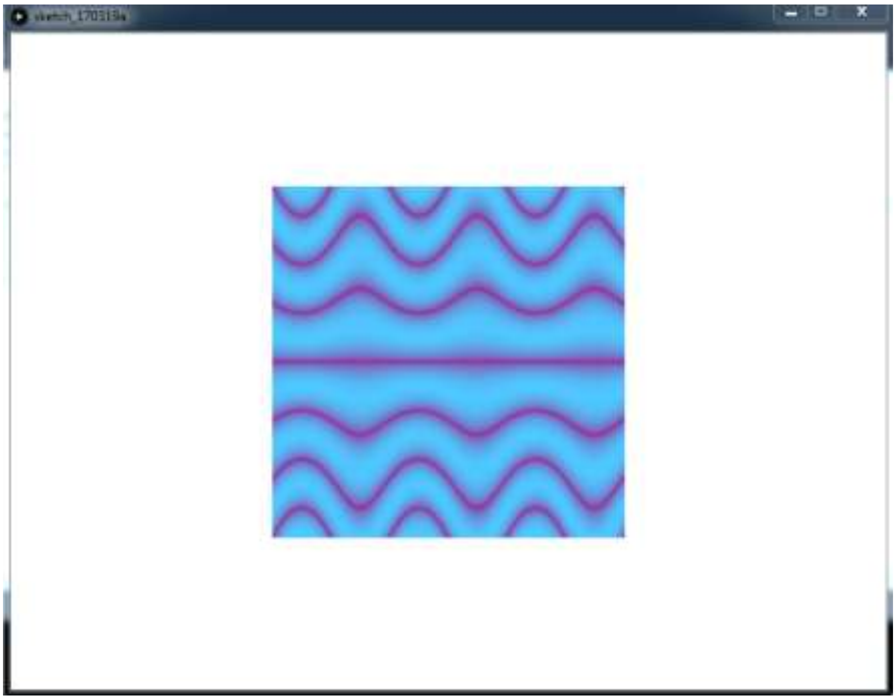


Aplicatie

Program

```
size(800,600);
background(255);
float x,y,cx=400,cy=300,d=17,c,
R1=160,G1=30,B1=140,R2=80,G2=200,B2=255,R,G,B;
for(y=-3*PI;y<=3*PI;y+=0.5/d)
  for(x=-3*PI;x<=3*PI;x+=0.1/d)
  {
    c=pow(abs(sin(y-(y/5)*sin(x))),0.5);
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
```

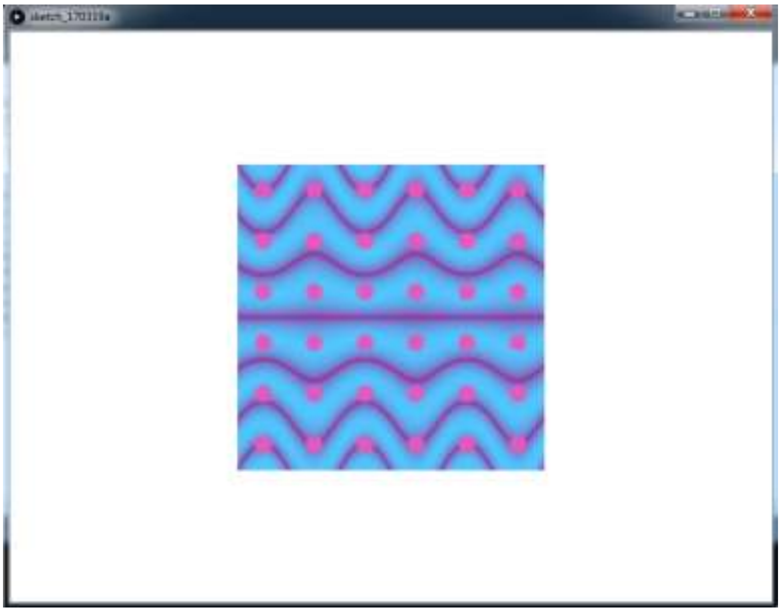
```
  set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
}
```



Compunerea texturilor

$$R=R_1\cdot(1-c)+R_2\cdot c$$
$$G=G_1\cdot(1-c)+G_2\cdot c$$
$$B=B_1\cdot(1-c)+B_2\cdot c$$
$$R=R\cdot(1-c_1)+R_3\cdot c_1$$
$$G=G\cdot(1-c_1)+G_3\cdot c_1$$
$$B=B\cdot(1-c_1)+B_3\cdot c_1$$
$$c=c(x,y)\text{ , } c:\mathbb{R}^2\rightarrow[0,1]$$
$$c_1=c_1(x,y)\text{ , } c_1:\mathbb{R}^2\rightarrow[0,1]$$

```
size(800,600);
background(255);
float x,y,cx=400,cy=300,d=17,
R1=160,G1=30,B1=140,R2=80,G2=200,B2=255,c,
R4=230,G4=90,B4=190,R,G,B,c0,c1;
for(y=-3*PI;y<=3*PI;y+=0.5/d)
  for(x=-3*PI;x<=3*PI;x+=0.1/d)
  {
    c=pow(abs(sin(y-(y/5)*sin(x))),0.5);
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    c0=pow(abs(sin(x)*sin(y)),5);
    c1=(atan(30*(c0-0.5))+PI/2)/PI;
    R=R*(1-c1)+R4*c1;
    G=G*(1-c1)+G4*c1;
    B=B*(1-c1)+B4*c1;
    set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
```



Panseluta

$r=a|\cos(2u)|,$

$u\in[-3\pi,\frac{\pi}{2}),\ a\in[0,1]$

$r=a|\cos(0.8(|2u|-0.4))|,$

$u\in[-3\pi,\frac{\pi}{2}),\ a\in[0,1]$

$r=a|\cos(0.8(|2u|-0.4))|$

$x=r\cdot\cos(u)$

$y=r\cdot\sin(u)$

$u\in[-3\pi,\frac{\pi}{2}),\ a\in[0,1]$

$r=a|\cos(0.8(|2u|-0.4))|$

$x=r\cdot\cos(u)$

$y=r\cdot\sin(u)$

$u\in[-3\pi,\frac{\pi}{2}),\ a\in[0,1]$

$z1=z^2 \quad ,z=x+y\cdot i$

$r=a|\cos(0.8(|2u|-0.4))|$

$x=r\cdot\cos(u-\frac{\pi}{4})$

$y=r\cdot\sin(u-\frac{\pi}{4})$

$u\in[-3\pi,\frac{\pi}{2}),\ a\in[0,1]$

$z1=z^2 \quad ,z=x+y\cdot i$

$r=a|\cos(0.8(|2u|-0.4))|$

$$x=\begin{cases} r\cdot\cos(u-\frac{\pi}{4}) & ,u\in[-3\pi,-3\pi+2.34)\cup[-3\pi+4.4,\frac{\pi}{2}) \\ r\cdot\cos(u+\frac{\pi}{4}+0.2) & ,u\in[-3\pi+2.34,-3\pi+4.4) \end{cases}$$

$$y=\begin{cases} r\cdot\sin(u-\frac{\pi}{4}) & ,u\in[-3\pi,-3\pi+2.34)\cup[-3\pi+4.4,\frac{\pi}{2}) \\ r\cdot\sin(u+\frac{\pi}{4}+0.2) & ,u\in[-3\pi+2.34,-3\pi+4.4) \end{cases}$$

$z1=z^2 \quad ,z=x+y\cdot i$

$r=a|\cos(0.8(|2u|-0.4))|\cdot 2\cdot\left[1+\frac{u+3\pi}{100}\right]^{0.1},a\in[0,1]$

$$x=\begin{cases} r\cdot\cos(u-\frac{\pi}{4}) & ,u\in[-3\pi,-3\pi+2.34)\cup[-3\pi+4.4,\frac{\pi}{2}) \\ r\cdot\cos(u+\frac{\pi}{4}+0.2) & ,u\in[-3\pi+2.34,-3\pi+4.4) \end{cases}$$

$$y=\begin{cases} r\cdot\sin(u-\frac{\pi}{4}) & ,u\in[-3\pi,-3\pi+2.34)\cup[-3\pi+4.4,\frac{\pi}{2}) \\ r\cdot\sin(u+\frac{\pi}{4}+0.2) & ,u\in[-3\pi+2.34,-3\pi+4.4) \end{cases}$$

$z1=z^2 \quad ,z=x+y\cdot i$

$$c1(y)=|\sin(y)|+1-\frac{|\cos(y)|}{2}$$

$$c2=\sin^{20}(x-c1(y))$$

$$c1(y)=|\sin(y)|+1-\frac{|\cos(y)|}{2}$$

$$c2=\sin^{20}(x-50\cdot c1^{0.2}(y))$$

$$c1(y)=|\sin(y)|+1-\frac{|\cos(y)|}{2}$$

$$c2=\sin^{20}(x-50\cdot c1^{0.2}(y))+\sin^{20}(x-50\cdot c1^{0.2}(y+\tfrac{\pi}{2}))$$

$$R=(1-c2)\cdot R1+c2\cdot R2$$
$$G=(1-c2)\cdot G1+c2\cdot G2$$
$$B=(1-c2)\cdot B1+c2\cdot B2$$

$$c3=1-\frac{\mathrm{arctg}(10(a-0.7))+\frac{\pi}{2}}{\pi}$$

$$R=(1-c3)\cdot R1+c3\cdot R3$$
$$G=(1-c3)\cdot G1+c3\cdot G3$$
$$B=(1-c3)\cdot B1+c3\cdot B3$$

$$c4=0.01(\sin(30u)+0.07\sin(55u)+0.2\sin(123u)-3|\sin(13u)|^{30})$$

$$c5=2^{-(a-c4-0.7)^2\cdot 30}$$

$$c6=\frac{\mathrm{arctg}(20(c5-0.7))+\frac{\pi}{2}}{\pi}$$

$$R=(1-c6)\cdot R+c6\cdot R4$$
$$G=(1-c6)\cdot G+c6\cdot G4$$
$$B=(1-c6)\cdot B+c6\cdot B4$$

$$c7=0.01(\sin(30u)+0.07\sin(55u)+0.2\sin(123u)-3|\sin(13u)|^{30})$$

$$c8=2^{-(a-c7-1)^2\cdot 10}$$

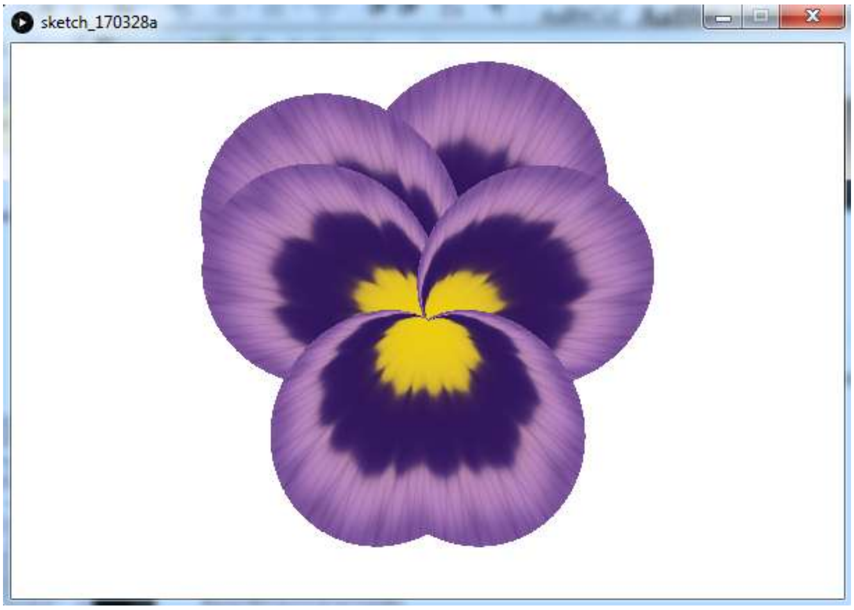
$$c9=\frac{\mathrm{arctg}(30(c8-1))+\frac{\pi}{2}}{\pi}$$

$$R=(1-c9)\cdot R+c9\cdot R4$$
$$G=(1-c9)\cdot G+c9\cdot G4$$
$$B=(1-c9)\cdot B+c9\cdot B4$$

Program

```
size(600,400);
background(255);
float x,y,r,r1,u,cx=300,cy=200,d=170,c,c1,c2,c3,rc,uc,x1,y1,
R1=210,G1=150,B1=250,R2=170,G2=110,B2=215,
R3=246,G3=218,B3=0,R4=40,G4=15,B4=90,R,G,B;
for(u=-3*PI+0.4;u<=PI/2-0.35;u+=0.3/d)
  for(r=0;r<=1;r+=0.3/d)
  {
    r1=pow(2-int(1+(u+3*PI-4.4)/100),0.1)*r*abs(cos((abs(2*u)-0.4)*0.80));
    x=r1*cos(u-PI/4);y=r1*sin(u-PI/4);
    if(u>=-3*PI+2.3&&u<=-3*PI+4.4)
    {
      x=r1*cos(u+PI/4+0.2);y=r1*sin(u+PI/4+0.2);
    }
    rc=r*7;uc=u*10;
    c=(pow(sin(1*(rc-50*pow((abs(sin(uc))+1-abs(cos(uc))/2),
      0.2))),20)+pow(sin(1*(rc-50*pow((abs(sin(uc+PI/2))
      +1-abs(cos(uc+PI/2))/2),0.2))),20))/2;
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    c1=(atan(10*(r-0.7))+PI/2)/PI;
    R=R3*(1-c1)+R*c1;
    G=G3*(1-c1)+G*c1;
    B=B3*(1-c1)+B*c1;
    c=0.01*(sin(30*u)+0.7*sin(55*u)+0.2*sin(123*u)
      -3*pow(abs(sin(13*u)),30));
    c1=pow(2,-30*pow(r-c-0.7,2));
    c2=(atan(20*(c1-0.7))+PI/2)/PI;
    R=R*(1-c2)+R4*c2;
    G=G*(1-c2)+G4*c2;
    B=B*(1-c2)+B4*c2;
    c=0.01*(sin(30*u)+0.7*sin(55*u)+0.2*sin(123*u)
      -3*pow(abs(sin(13*u)),30));
    c1=pow(2,-10*pow(r-c-1,2));
    c3=(atan(30*(c1-1))+PI/2)/PI;
    R=R*(1-c3)+R4*c3;
    G=G*(1-c3)+G4*c3;
    B=B*(1-c3)+B4*c3;
    x1=x*x-y*y;y1=2*x*y;
    set(int(x1*d+cx),int(-y1*d+cy),color(R,G,B));
```

```
}
```


sketch_170328a

Crinul cu nervuri

$$r=a(|\sin(1.5u)|+1-|\cos(1.5u)|)$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$z=0.6\sin(r)|\cos(1.5u)|$$

$$r=a(|\sin(1.5u)|+1-|\cos(1.5u)|)$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$z=0.6\sin(r)|\cos(1.5u)|-\frac{4}{(3r)^4+1}$$

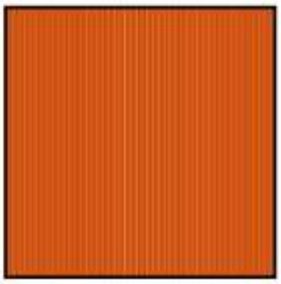
$$r=a(|\sin(1.5u)|+1-|\cos(1.5u)|)$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$z=0.6\sin(r)|\cos(1.5u)| -\frac{4}{(3r)^4+1}-\frac{7}{(10r)^{40}+1}$$

$$r=a(|\sin(1.5u)|+1-|\cos(1.5u)|)$$
$$x=r\cdot\cos\left(u+\pi+\left(1-\left[\frac{u}{2\pi}\right]\right)\frac{\pi}{3}\right)$$
$$y=r\cdot\sin\left(u+\pi+\left(1-\left[\frac{u}{2\pi}\right]\right)\frac{\pi}{3}\right)$$
$$z=0.6\sin(r)|\cos(1.5u)| -\frac{4}{(3r)^4+1}-0.35\left[\frac{u}{2\pi}\right]-\frac{7}{(10r)^{40}+1}$$

$$r=a(|\sin(1.5u)|+1-|\cos(1.5u)|)$$
$$x=r\cdot\cos\left(u+\pi+\left(1-\left[\frac{u}{2\pi}\right]\right)\frac{\pi}{3}\right)$$
$$y=r\cdot\sin\left(u+\pi+\left(1-\left[\frac{u}{2\pi}\right]\right)\frac{\pi}{3}\right)$$
$$z=0.6\sin(r)|\cos(1.5u)| +0.002\left(\frac{r}{10}\right)^6 10^a\cdot(0.7\sin(70u)+\sin(23u)+7|\sin(6u)|) -\frac{4}{(3r)^4+1}-0.35\left[\frac{u}{2\pi}\right]-\frac{7}{(10r)^{40}+1}$$

Textura

$$c=1-|\sin(30u)|$$



$$c=1-|\sin(150u)|^{0.2}$$

$$c=1-|\sin(150u+35\sin(3u)|\sin(3a)|)|^{0.2}$$

$$c=1-|\sin(150u+35\sin(3u)|\sin(3a)|)|^{0.2}$$

$$c1=2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}}$$

$$R=R1(1-c)+R2\cdot c$$
$$G=G1(1-c)+G2\cdot c$$
$$B=B1(1-c)+B2\cdot c$$
$$R=R(1-c1)+R3\cdot c1$$
$$G=G(1-c1)+G3\cdot c1$$
$$B=B(1-c1)+B3\cdot c1$$

$$c=1-|\sin(150u+35\sin(3u)|\sin(3a)|)|^{0.2}$$

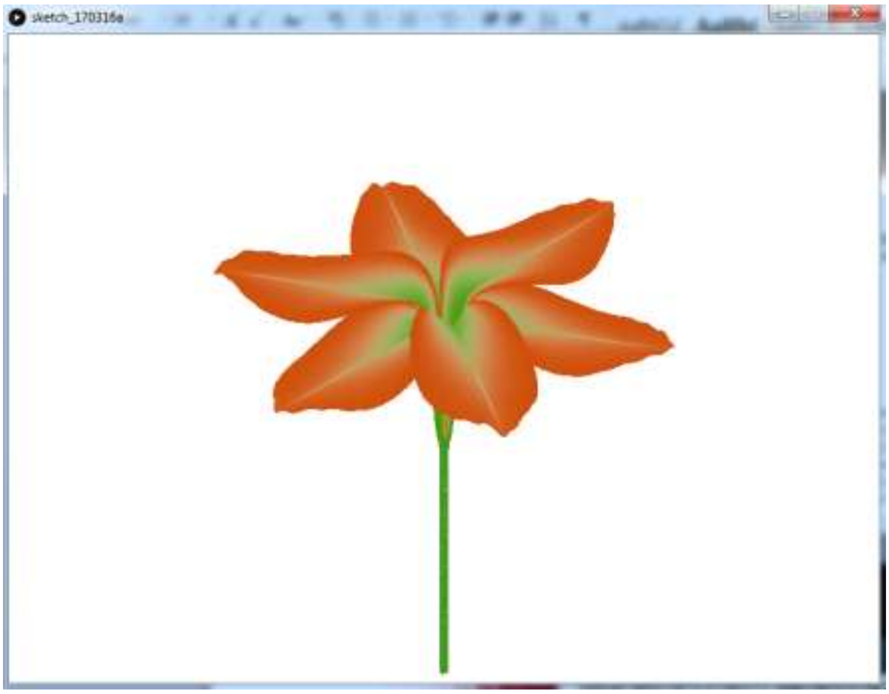
$$c1=2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}}$$

$$c2=2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}\cdot 2}$$

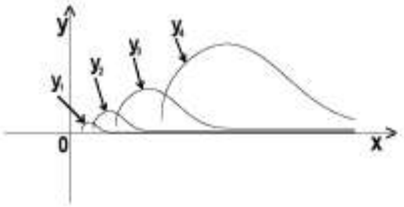
$$R=R1(1-c)+R2\cdot c$$
$$G=G1(1-c)+G2\cdot c$$
$$B=B1(1-c)+B2\cdot c$$
$$R=R(1-c1)+R3\cdot c1$$
$$G=G(1-c1)+G3\cdot c1$$
$$B=B(1-c1)+B3\cdot c1$$
$$R=R(1-c2)+R4\cdot c2$$
$$G=G(1-c2)+G4\cdot c2$$
$$B=B(1-c2)+B4\cdot c2$$

Program

```
size(800,600);
background(255);
float[][] mat=new float[800][600];
float x,y,z,d=30,a,r,u,X,Y,cx=400,cy=250,
R1=205,G1=76,B1=8,R2=225,G2=106,B2=38,R3=255,G3=255,B3=255,
R4=70,G4=150,B4=0,R,G,B,c,c1,c2;
int i,j;
for(i=0;i<8000;i++)
for(j=0;j<600;j++)
  mat[i][j]=-100;
for(u=0;u<4*PI;u+=0.1/d)
  for(a=0;a<=3;a+=0.3/d)
  {
    r=a*(abs(sin(u*1.5))+1-abs(cos(1.5*u)));
    x=r*cos(u+PI+PI/3*(1-int(u/(2*PI))));
    y=r*sin(u+PI+PI/3*(1-int(u/(2*PI))));
    z=0.6*sin(r)*abs(cos(u*1.5))+0.002*pow(r/10,6)*pow(10,a)
      *(0.7*sin(u*70)+sin(23*u)+7*abs(sin(6*u)))
      -4/(pow(r*3,4)+1)-0.35*int(u/(2*PI))-7/(pow(r*10,40)+1);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    c=1-pow(abs(sin(150*u+35*sin(3*u)*abs(sin(a)))),0.2);
    c1=pow(2,-a*a*pow(abs(sin(1.5*u+PI/2)),0.2));
    c2=pow(2,-a*a*pow(abs(sin(1.5*u+PI/2)),0.2)*2);
    R=R1*(1-c)+R2*c;
    G=G1*(1-c)+G2*c;
    B=B1*(1-c)+B2*c;
    R=R*(1-c1)+R3*c1;
    G=G*(1-c1)+G3*c1;
    B=B*(1-c1)+B3*c1;
    R=R*(1-c2)+R4*c2;
    G=G*(1-c2)+G4*c2;
    B=B*(1-c2)+B4*c2;
    if(int(-Y*d+cy)<600)
     if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
     {
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      if(z<-4)
        set(int(X*d+cx),int(-Y*d+cy),color(70,150,36));
     }
  }
```



Melcul de mare



$$y1=0.25\cdot 1\left(\frac{4x}{1}-0.8\right)^{0.4}\cdot 2^{-\left(\frac{4x}{1}-0.8\right)^{3}}+0.005\cdot 1$$

$$y2=0.25\cdot 2\left(\frac{4x}{2}-0.8\right)^{0.4}\cdot 2^{-\left(\frac{4x}{2}-0.8\right)^{3}}+0.005\cdot 2$$

$$y3=0.25\cdot 4\left(\frac{4x}{4}-0.8\right)^{0.4}\cdot 2^{-\left(\frac{4x}{4}-0.8\right)^{3}}+0.005\cdot 4$$

....

u	$\left[\frac{u}{2\pi}\right]$	$2^{\left[\frac{u}{2\pi}\right]}$	$\frac{u}{2\pi}$	$2^{\frac{u}{2\pi}}$
$[0,2\pi)$	0	1	$[0,1)$	$[1,2)$
$[2\pi,4\pi)$	1	2	$[1,2)$	$[2,4)$
$[4\pi,6\pi)$	2	4	$[2,3)$	$[4,8)$

$$p=2^{\frac{u}{2\pi}}$$

$$r=0.25p\left(\frac{4z}{p}-0.8\right)^{0.4}\cdot 2^{-\left(\frac{4z}{p}-0.8\right)^{3}}+0.005p$$

$$p=2^{\frac{u}{2\pi}}$$

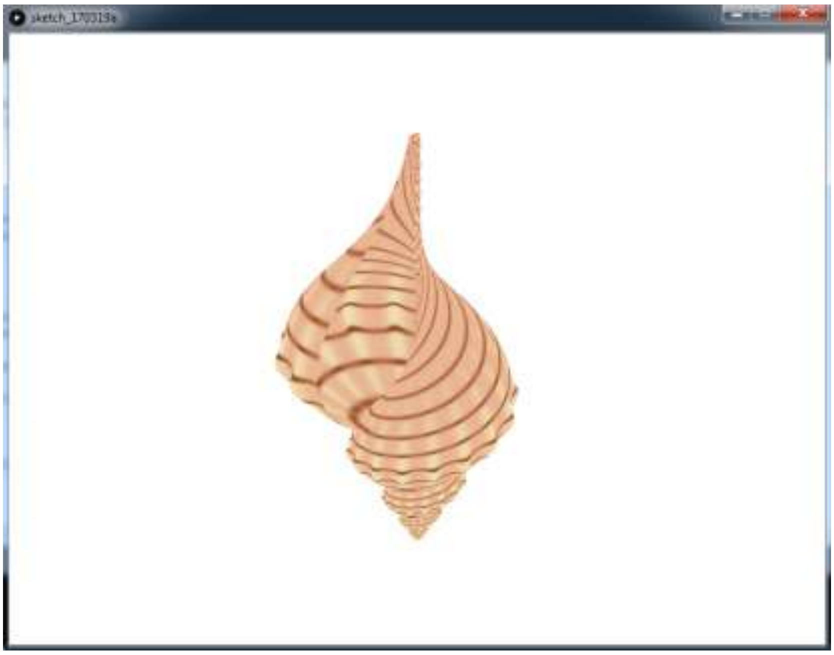
$$r=\left(0.25p\left(\frac{4z}{p}-0.8\right)^{0.4}\cdot 2^{-\left(\frac{4z}{p}-0.8\right)^{3}}+0.005p\right)\cdot\left(1+0.1\left(\sin\left(\frac{5z}{p}\right)\right)^{100}\cdot\left(\sin(8u)\right)^{4}\right)$$

Aplicatie

Program

```
float[][] mat=new float[800][600];
void init_mat()
{
  int i,j;
  for(i=0;i<800;i++)
    for(j=0;j<600;j++)
      mat[i][j]=-100;
}
void melc()
{
float x,y,z,r,u,p,X,Y,d=20,cx=400,cy=500,
R1=130,G1=64,B1=16,R2=240,G2=224,B2=176,
R3=227,G3=140,B3=121,R,G,B,c,c1;
for(z=0;z<=20;z+=0.25/d)
  for(u=0;u<=10*PI;u+=2.5/(p*d))
  {
    p=pow(2,(u/(2*PI)));
    r=(0.25*p*pow(4/p*z-0.8,0.4)*pow(2,-pow(4/p*z-0.8,3))+0.005*p)
    *(1+0.1*pow(sin(5*z/p),100)*pow(sin(8*u),4));
    x=r*cos(u);
    y=r*sin(u);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
      if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
      {
        mat[int(X*d+cx)][int(-Y*d+cy)]=z;
        c=1-pow(sin(80*z/pow(2,(u/(2*PI)))),20);
        R=(1-c)*R1+c*R2;
        G=(1-c)*G1+c*G2;
        B=(1-c)*B1+c*B2;
        c1=0.5+0.5*pow(sin(5*z/pow(2,(u/(2*PI)))),4)*pow(sin(8*u),4);
        R=(1-c1)*R3+c1*R;
        G=(1-c1)*G3+c1*G;
        B=(1-c1)*B3+c1*B;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
```

```
  }
}
void setup()
{
size(800,600);
background(255);
init_mat();
melc();
}
```



Translatia

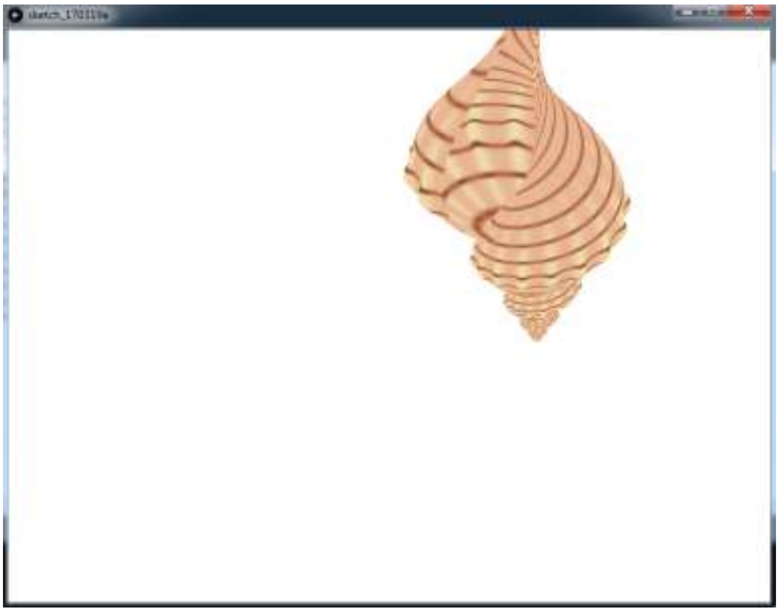
Aplicatie

<u>Program</u>

```
float[][] mat=new float[800][600];
void init_mat()
{
  int i,j;
  for(i=0;i<800;i++)
    for(j=0;j<600;j++)
      mat[i][j]=-100;
}
```

```
void melc(float x0,float y0,float z0)
{
float x,y,z,r,u,p,X,Y,d=20,cx=400,cy=500,
R1=130,G1=64,B1=16,R2=240,G2=224,B2=176,
R3=227,G3=140,B3=121,R,G,B,c,c1,
x1,y1,z1;
for(z=0;z<=20;z+=0.25/d)
  for(u=0;u<=10*PI;u+=2.5/(p*d))
  {
    p=pow(2,(u/(2*PI)));
    r=(0.25*p*pow(4/p*z-0.8,0.4)*pow(2,-pow(4/p*z-0.8,3))+0.005*p)
    *(1+0.1*pow(sin(5*z/p),100)*pow(sin(8*u),4));
    x=r*cos(u);
    y=r*sin(u);
    //translatie
    x1=x+x0;
    y1=y+y0;
    z1=z+z0;
    X=-0.86*x1+0.86*y1;
    Y=z1-0.5*x1-0.5*y1;
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
      if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
      {
        mat[int(X*d+cx)][int(-Y*d+cy)]=z;
        c=1-pow(sin(80*z/pow(2,(u/(2*PI)))),20);
        R=(1-c)*R1+c*R2;
        G=(1-c)*G1+c*G2;
        B=(1-c)*B1+c*B2;
        c1=0.5+0.5*pow(sin(5*z/pow(2,(u/(2*PI)))),4)*pow(sin(8*u),4);
        R=(1-c1)*R3+c1*R;
        G=(1-c1)*G3+c1*G;
        B=(1-c1)*B3+c1*B;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
  }
}
void setup()
{
size(800,600);
background(255);
init_mat();
melc(-9,0,4);
```

}



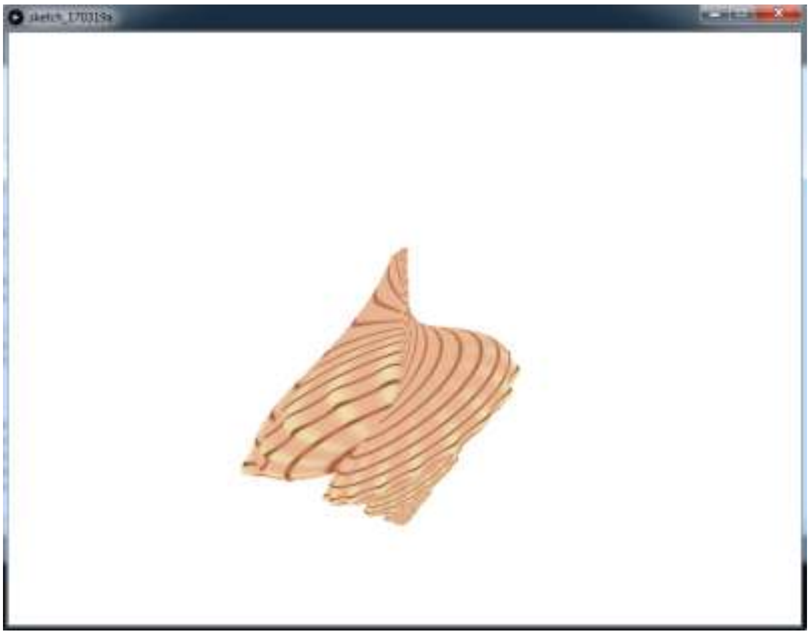
Scalarea

Aplicatie

Program

```
float[][] mat=new float[800][600];
void init_mat()
{
  int i,j;
  for(i=0;i<800;i++)
    for(j=0;j<600;j++)
      mat[i][j]=-100;
}
void melc(float scal_x,float scal_y,float scal_z)
{
float x,y,z,r,u,p,X,Y,d=20,cx=400,cy=500,
R1=130,G1=64,B1=16,R2=240,G2=224,B2=176,
R3=227,G3=140,B3=121,R,G,B,c,c1,
x1,y1,z1;
for(z=0;z<=20;z+=0.25/d)
  for(u=0;u<=10*PI;u+=2.5/(p*d))
  {
    p=pow(2,(u/(2*PI)));
    r=(0.25*p*pow(4/p*z-0.8,0.4)*pow(2,-pow(4/p*z-0.8,3))+0.005*p)
    *(1+0.1*pow(sin(5*z/p),100)*pow(sin(8*u),4));
    x=r*cos(u);
```

```
    y=r*sin(u);
    //scalare
    x1=scal_x*x;y1=scal_y*y;z1=scal_z*z;
    X=-0.86*x1+0.86*y1;
    Y=z1-0.5*x1-0.5*y1;
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
      if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
      {
        mat[int(X*d+cx)][int(-Y*d+cy)]=z;
        c=1-pow(sin(80*z/pow(2,(u/(2*PI)))),20);
        R=(1-c)*R1+c*R2;
        G=(1-c)*G1+c*G2;
        B=(1-c)*B1+c*B2;
        c1=0.5+0.5*pow(sin(5*z/pow(2,(u/(2*PI)))),4)*pow(sin(8*u),4);
        R=(1-c1)*R3+c1*R;
        G=(1-c1)*G3+c1*G;
        B=(1-c1)*B3+c1*B;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
  }
}
void setup()
{
size(800,600);
background(255);
init_mat();
melc(1.5,0.5,0.7);
}
```



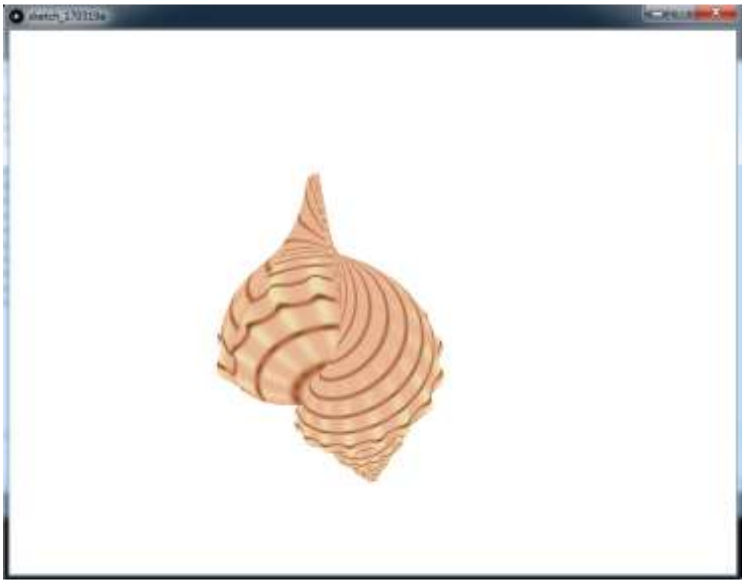
Rotatii

Aplicatie

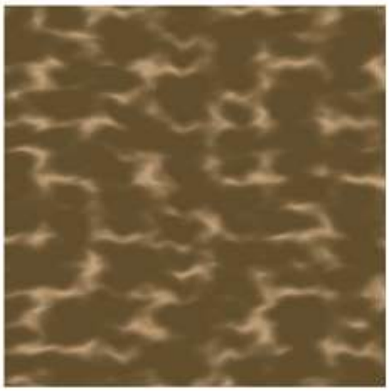
Program

```
float[][] mat=new float[800][600];
void init_mat()
{
  int i,j;
  for(i=0;i<800;i++)
    for(j=0;j<600;j++)
      mat[i][j]=-100;
}
void melc(float u_OX,float u_OY,float u_OZ)
{
float x,y,z,r,u,p,X,Y,d=20,cx=400,cy=500,
R1=130,G1=64,B1=16,R2=240,G2=224,B2=176,
R3=227,G3=140,B3=121,R,G,B,c,c1,
x1,y1,z1,x2,y2,z2,x3,y3,z3;
for(z=0;z<=20;z+=0.25/d)
  for(u=0;u<=10*PI;u+=2.5/(p*d))
  {
    p=pow(2,(u/(2*PI)));
    r=(0.25*p*pow(4/p*z-0.8,0.4)*pow(2,-pow(4/p*z-0.8,3))+0.005*p)
    *(1+0.1*pow(sin(5*z/p),100)*pow(sin(8*u),4));
```

```
    x=r*cos(u);
    y=r*sin(u);
     //rotatie OX
    x1=x;
    y1=y*cos(u_OX)-z*sin(u_OX);
    z1=y*sin(u_OX)+z*cos(u_OX);
    //rotatie OY
    x2=z1*sin(u_OY)+x1*cos(u_OY);
    y2=y1;
    z2=z1*cos(u_OY)-x1*sin(u_OY);
    //rotatie OZ
    x3=x2*cos(u_OZ)-y2*sin(u_OZ);
    y3=x2*sin(u_OZ)+y2*cos(u_OZ);
    z3=z2;
    X=-0.86*x3+0.86*y3;
    Y=z3-0.5*x3-0.5*y3;
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
      if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
      {
        mat[int(X*d+cx)][int(-Y*d+cy)]=z;
        c=1-pow(sin(80*z/pow(2,(u/(2*PI)))),20);
        R=(1-c)*R1+c*R2;
        G=(1-c)*G1+c*G2;
        B=(1-c)*B1+c*B2;
        c1=0.5+0.5*pow(sin(5*z/pow(2,(u/(2*PI)))),4)*pow(sin(8*u),4);
        R=(1-c1)*R3+c1*R;
        G=(1-c1)*G3+c1*G;
        B=(1-c1)*B3+c1*B;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
  }
}
void setup()
{
size(800,600);
background(255);
init_mat();
melc(-0.1,0.2,-0.3);
}
```



Textura

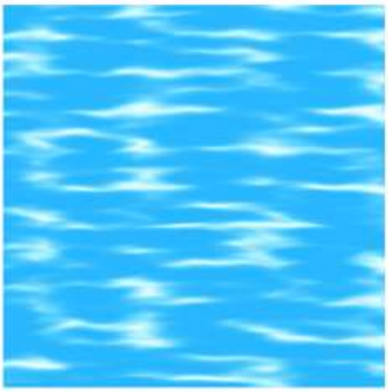


$$e1(x,y)=1-|\sin(1.2x+0.2\sin(x)+\sin(0.55y)+0.2\sin(3.7y)+0.1\sin(7.9y)+0.2\sin(1.8y))\cdot\sin(1.2y+0.5\sin(0.42x)+0.1\cos(5.3x)+0.2\sin(4.9x))|$$

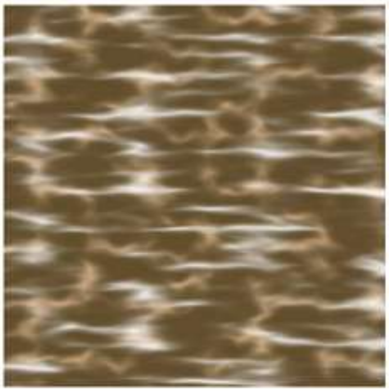
$$e2(x,y)=1-|\sin(2.3x+0.2\sin(1.7x)+0.2\sin(3.7y))\sin(2.3y+0.5\sin(x)+0.1\cos(8x)+0.2\sin(9x))|$$

$$e_{a,b}(x,y)=e1(a\cdot x,b\cdot y)\cdot e2(a\cdot x,b\cdot y)\cdot e1(x+y,y-x)$$

$$c=2^{-3\cdot e_{1,2}^2}$$



$$c1=2^{-3\cdot e_{0.5,3}^2}$$



$$c=2^{-3\cdot e_{1,2}^2}$$

$$R=R1(1-c)+R2\cdot c$$
$$G=G1(1-c)+G2\cdot c$$
$$B=B1(1-c)+B2\cdot c$$

$$c1=2^{-3\cdot e_{0.5,3}^2}$$

$$R=R(1-c1)+R4\cdot c1$$
$$G=G(1-c1)+G4\cdot c1$$
$$B=B(1-c1)+B4\cdot c1$$

$$c=2^{-3\cdot e_{1,2}^2}$$

$$R=R1(1-c)+R2\cdot c$$
$$G=G1(1-c)+G2\cdot c$$
$$B=B1(1-c)+B2\cdot c$$

$$c2=\frac{(\mathrm{arctg}(7-y)+\frac{\pi}{2})}{\pi}$$

$$R=R(1-c2)+R3\cdot c2$$
$$G=G(1-c2)+G3\cdot c2$$
$$B=B(1-c2)+B3\cdot c2$$

$$c1=2^{-3\cdot e_{0.5,3}^2}$$

$$R=R(1-c1)+R4\cdot c1$$
$$G=G(1-c1)+G4\cdot c1$$
$$B=B(1-c1)+B4\cdot c1$$

Aplicatie

Program

```
float e1(float x,float y)
{
  return 1-abs(sin(1.2*x+0.2*sin(x)+sin(0.55*y)
       +0.2*sin(3.7*y)+0.1*sin(7.9*y)+0.2*sin(1.8*y))
       *sin(1.2*y+0.5*sin(0.42*x)+0.1*cos(5.3*x)+0.2*sin(4.9*x)));
}
float e2(float x,float y)
{
  return 1-abs(sin(2.3*x+0.2*sin(1.7*x)+0.2*sin(3.7*y))
      *sin(2.3*y+0.5*sin(x)+0.1*cos(8*x)+0.2*sin(9*x)));
}
void setup()
{
size (800,600);
background (255);
float x,y,x1,y1,d=40,cx=400,cy=500,
R1=99,G1=79,B1=46,R2=211,G2=176,B2=135,
R3=41,G3=184,B3=255,R4=255,G4=255,B4=255,R,G,B,c,c1,c2,e;
for(y=0;y<=10;y+=0.5/d)
  for(x=-5;x<=5;x+=0.5/d)
  {
       x1=x;y1=2*y;
       e=e1(x1,y1)*e2(x1,y1)*e1(x1+y1,y1-x1);
       c=pow(2,-3*e*e);
       R=c*R1+(1-c)*R2;
       G=c*G1+(1-c)*G2;
       B=c*B1+(1-c)*B2;
       c2=(atan(7-y)+PI/2)/PI;
       R=c2*R+(1-c2)*R3;
       G=c2*G+(1-c2)*G3;
       B=c2*B+(1-c2)*B3;
       x1=0.5*x;y1=3*y;
       e=e1(x1,y1)*e2(x1,y1)*e1(x1+y1,y1-x1);
       c1=pow(2,-3*e*e);
       R=c1*R+(1-c1)*R4;
       G=c1*G+(1-c1)*G4;
       B=c1*B+(1-c1)*B4;
```

```
      set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
   }
}
```

Aplicatie

Program

```
float[][] mat=new float[800][600];
float[][][] mat1=new float[800][600][3];
float[][][] mat2=new float[800][600][3];
float d=40,cx=400,cy=500;
void init_mat()
{
  int i,j;
  for(i=0;i<800;i++)
    for(j=0;j<600;j++)
      mat[i][j]=-100;
}
void melc2()
{
float x,y,z,r,u,p,X,Y,d=10,cx=400,cy=450,
R2=250,G2=224,B2=176,R1=130,G1=64,B1=16,R,G,B,c;
for(u=0;u<=10*PI;u+=0.05/d)
  for(z=0;z<=pow(2,u/(2*PI));z+=0.25/d)
  {
```

```
    p=pow(2,(u/(2*PI)));
    r=0.25*p*pow(4/p*z-0.8,0.4)*pow(2,-pow(4/p*z-0.8,3))+0.005*p+0.1;
    r+=0.3*r*pow(abs(sin(1.5*u+PI/2)),500)*pow(abs(sin(30*z/p)),500);
    x=r*cos(u);
    y=r*sin(u);
    X=-0.86*x+0.86*y;
    Y=z-0.5*x-0.5*y;
    if(z>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      mat[int(X*d+cx)][int(-Y*d+cy)]=z;
      c=(1-pow(abs(sin(50*z/p+1.5))*abs(sin(6*u)),5));
      R=(1-c)*R1+c*R2;
      G=(1-c)*G1+c*G2;
      B=(1-c)*B1+c*B2;
      mat1[int(X*d+cx)][int(-Y*d+cy)][0]=R;
      mat1[int(X*d+cx)][int(-Y*d+cy)][1]=G;
      mat1[int(X*d+cx)][int(-Y*d+cy)][2]=B;
    }
  }
}
float e1(float x,float y)
{
  return 1-abs(sin(1.2*x+0.2*sin(x)+sin(0.55*y)
       +0.2*sin(3.7*y)+0.1*sin(7.9*y)+0.2*sin(1.8*y))
       *sin(1.2*y+0.5*sin(0.42*x)+0.1*cos(5.3*x)+0.2*sin(4.9*x)));
}
float e2(float x,float y)
{
  return 1-abs(sin(2.3*x+0.2*sin(1.7*x)+0.2*sin(3.7*y))
      *sin(2.3*y+0.5*sin(x)+0.1*cos(8*x)+0.2*sin(9*x)));
}
void nisip()
{
float x,y,x1,y1,
R1=99,G1=79,B1=46,R2=211,G2=176,B2=135,
R,G,B,c,e;
for(y=0;y<=10;y+=0.5/d)
  for(x=-5;x<=5;x+=0.5/d)
   {
     x1=x;y1=2*y;
     e=e1(x1,y1)*e2(x1,y1)*e1(x1+y1,y1-x1);
     c=pow(2,-3*e*e);
     R=c*R1+(1-c)*R2;
```

```
    G=c*G1+(1-c)*G2;
    B=c*B1+(1-c)*B2;
    mat1[int(x*d+cx)][int(-y*d+cy)][0]=R;
    mat1[int(x*d+cx)][int(-y*d+cy)][1]=G;
    mat1[int(x*d+cx)][int(-y*d+cy)][2]=B;
   }
}
void mare()
{
float x,y,x1,y1,
R3=41,G3=184,B3=255,R4=255,G4=255,B4=255,R,G,B,c1,c2,e;
for(y=0;y<=10;y+=0.5/d)
  for(x=-5;x<=5;x+=0.5/d)
  {
      R=mat1[int(x*d+cx)][int(-y*d+cy)][0];
      G=mat1[int(x*d+cx)][int(-y*d+cy)][1];
      B=mat1[int(x*d+cx)][int(-y*d+cy)][2];
      c2=(atan(7-y)+PI/2)/PI;
      R=c2*R+(1-c2)*R3;
      G=c2*G+(1-c2)*G3;
      B=c2*B+(1-c2)*B3;
      x1=0.5*x;y1=3*y;
      e=e1(x1,y1)*e2(x1,y1)*e1(x1+y1,y1-x1);
      c1=pow(2,-3*e*e);
      R=c1*R+(1-c1)*R4;
      G=c1*G+(1-c1)*G4;
      B=c1*B+(1-c1)*B4;
      set(int(x*d+cx),int(-y*d+cy),color(R,G,B));
  }
}
void setup()
{
size (800,600);
background (255);
nisip();
melc2();
mare();
}
```

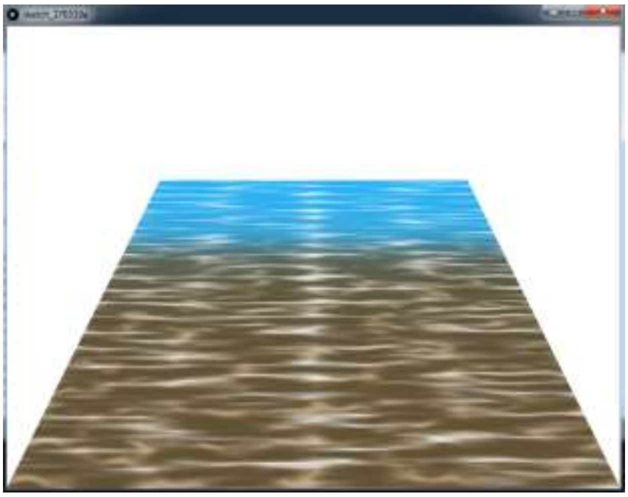
Perspectiva

Aplicatie

Program

```
float e1(float x,float y)
{
  return 1-abs(sin(1.2*x+0.2*sin(x)+sin(0.55*y)
       +0.2*sin(3.7*y)+0.1*sin(7.9*y)+0.2*sin(1.8*y))
       *sin(1.2*y+0.5*sin(0.42*x)+0.1*cos(5.3*x)+0.2*sin(4.9*x)));
}
float e2(float x,float y)
{
  return 1-abs(sin(2.3*x+0.2*sin(1.7*x)+0.2*sin(3.7*y))
      *sin(2.3*y+0.5*sin(x)+0.1*cos(8*x)+0.2*sin(9*x)));
}
void setup()
{
size (800,600);
background (255);
float x,y,z,x1,z1,X,Y,d=40,cx=400,cy=-200,
R1=99,G1=79,B1=46,R2=211,G2=176,B2=135,
R3=41,G3=184,B3=255,R4=255,G4=255,B4=255,R,G,B,c,c1,c2,e;
for(z=0;z<=20;z+=0.5/d)
  for(x=-5;x<=5;x+=0.5/d)
  {
       y=-10;
       //perspectiva
       X=d*x/(d-z);
```

```
        Y=d*y/(d-z);
        if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
        {
        x1=x;z1=2*z;
        e=e1(x1,z1)*e2(x1,z1)*e1(x1+z1,z1-x1);
        c=pow(2,-3*e*e);
        R=c*R1+(1-c)*R2;
        G=c*G1+(1-c)*G2;
        B=c*B1+(1-c)*B2;
        c2=(atan(z-7)+PI/2)/PI;
        R=c2*R+(1-c2)*R3;
        G=c2*G+(1-c2)*G3;
        B=c2*B+(1-c2)*B3;
        x1=0.5*x;z1=3*z;
        e=e1(x1,z1)*e2(x1,z1)*e1(x1+z1,z1-x1);
        c1=pow(2,-3*e*e);
        R=c1*R+(1-c1)*R4;
        G=c1*G+(1-c1)*G4;
        B=c1*B+(1-c1)*B4;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
        }
    }
}
```



Aplicatie

Program

```
float[][] mat=new float[800][600];
float[][][] mat1=new float[800][600][3];
```

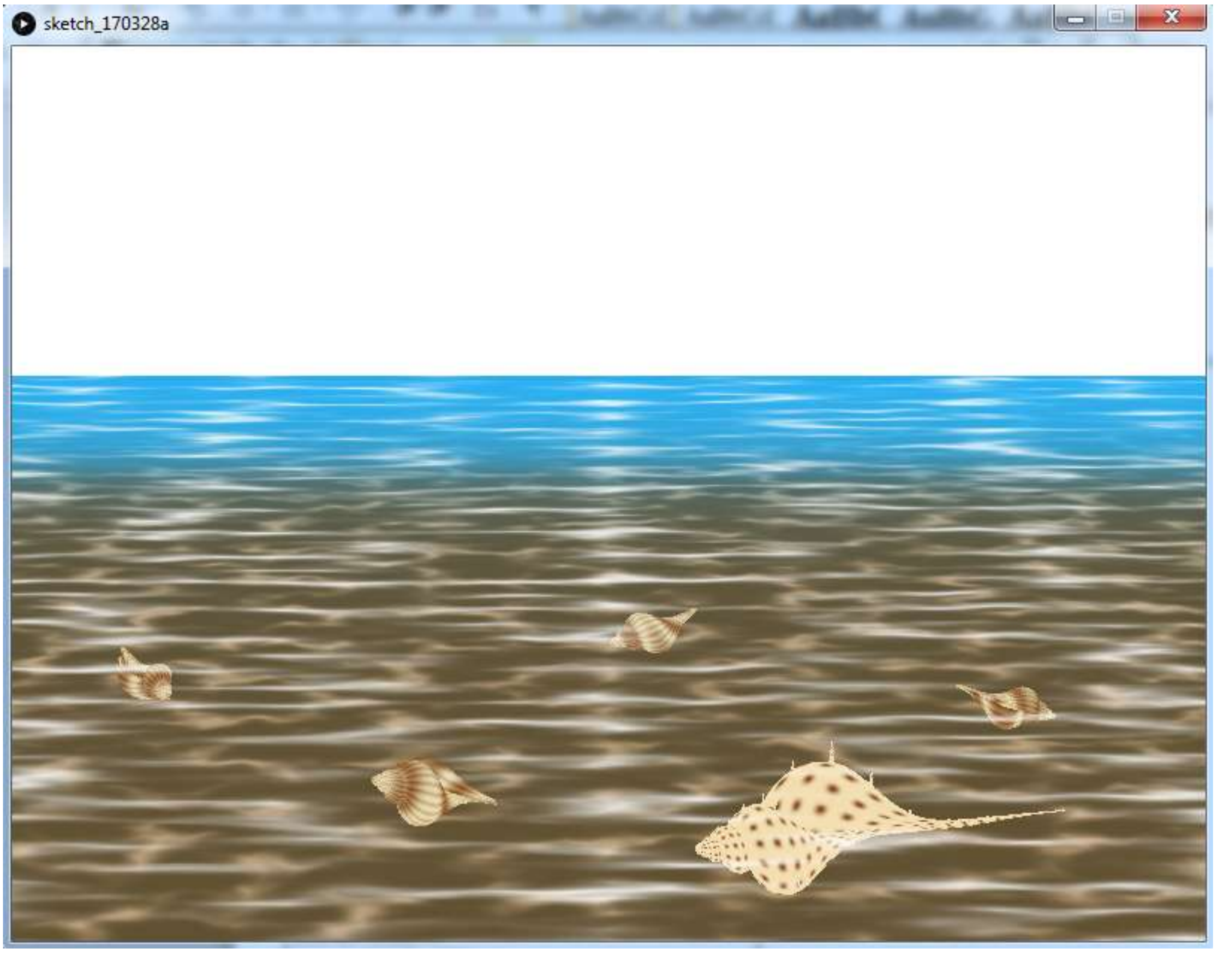
```
float[][][] mat2=new float[800][600][3];
float d=40,cx=400,cy=-200;
void init_mat()
{
  int i,j;
  for(i=0;i<800;i++)
    for(j=0;j<600;j++)
      mat[i][j]=-100;
}
void melc(float scal,float u_OX,float u_OY,float u_OZ,float x0,float y0,float
z0)
{
init_mat();
float x,y,z,r,u,p,X,Y,d=50,cx=400,cy=-200,
R1=130,G1=64,B1=16,R2=240,G2=224,B2=176,R,G,B,c,c1,
x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5;
for(z=0;z<=20;z+=0.25/(scal*d))
  for(u=0;u<=10*PI;u+=2.5/(p*scal*d))
  {
    p=pow(2,(u/(2*PI)));
    r=0.25*p*pow(4/p*z-0.8,0.4)*pow(2,-pow(4/p*z-0.8,3))+0.005*p;
    x=r*cos(u);
    y=r*sin(u);
    //scalare
    x1=scal*x;y1=scal*y;z1=scal*z;
    //rotatie OZ
    x2=x1*cos(u_OZ)-y1*sin(u_OZ);
    y2=x1*sin(u_OZ)+y1*cos(u_OZ);
    z2=z1;
    //rotatie OX
    x3=x2;
    y3=y2*cos(u_OX)-z2*sin(u_OX);
    z3=y2*sin(u_OX)+z2*cos(u_OX);
    //rotatie OY
    x4=z3*sin(u_OY)+x3*cos(u_OY);
    y4=y3;
    z4=z3*cos(u_OY)-x3*sin(u_OY);
    //translatie
    x5=x4+x0;
    y5=y4+y0;
    z5=z4+z0;
    //perspectiva
    X=d*x5/(d-z5);
```

```
    Y=d*y5/(d-z5);
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
      if(z5>mat[int(X*d+cx)][int(-Y*d+cy)])
      {
        mat[int(X*d+cx)][int(-Y*d+cy)]=z5;
        c=(abs(sin(10*u))+2-abs(sin(2*r)-1))/3;
        R=(1-c)*R1+c*R2;
        G=(1-c)*G1+c*G2;
        B=(1-c)*B1+c*B2;
        mat1[int(X*d+cx)][int(-Y*d+cy)][0]=R;
        mat1[int(X*d+cx)][int(-Y*d+cy)][1]=G;
        mat1[int(X*d+cx)][int(-Y*d+cy)][2]=B;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
  }
}
void melc2(float scal,float u_OX,float u_OY,float u_OZ,float x0,float y0,float
z0)
{
init_mat();
float x,y,z,r,u,p,X,Y,d=50,cx=400,cy=-200,
R2=250,G2=224,B2=176,R1=130,G1=64,B1=16,R,G,B,c,
x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5;
for(u=0;u<=10*PI;u+=0.05/(scal*d))
  for(z=0;z<=pow(2,u/(2*PI));z+=0.25/(scal*d))
  {
    p=pow(2,(u/(2*PI)));
    r=0.25*p*pow(4/p*z-0.8,0.4)*pow(2,-pow(4/p*z-0.8,3))+0.005*p+0.1;
    r+=0.3*r*pow(abs(sin(1.5*u+PI/2)),500)*pow(abs(sin(30*z/p)),500);
    x=r*cos(u);
    y=r*sin(u);
    //scalare
    x1=scal*x;y1=scal*y;z1=scal*z;
    //rotatie OZ
    x2=x1*cos(u_OZ)-y1*sin(u_OZ);
    y2=x1*sin(u_OZ)+y1*cos(u_OZ);
    z2=z1;
    //rotatie OX
    x3=x2;
    y3=y2*cos(u_OX)-z2*sin(u_OX);
    z3=y2*sin(u_OX)+z2*cos(u_OX);
    //rotatie OY
```

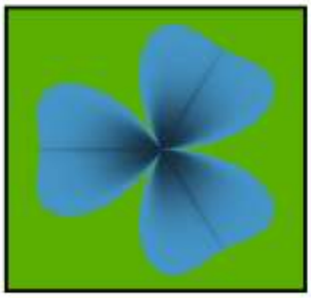
```
    x4=z3*sin(u_OY)+x3*cos(u_OY);
    y4=y3;
    z4=z3*cos(u_OY)-x3*sin(u_OY);
    //translatie
    x5=x4+x0;
    y5=y4+y0;
    z5=z4+z0;
    //perspectiva
    X=d*x5/(d-z5);
    Y=d*y5/(d-z5);
    if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
      if(z5>mat[int(X*d+cx)][int(-Y*d+cy)])
      {
        mat[int(X*d+cx)][int(-Y*d+cy)]=z5;
        c=(1-pow(abs(sin(50*z/p+1.5))*abs(sin(6*u)),5));
        R=(1-c)*R1+c*R2;
        G=(1-c)*G1+c*G2;
        B=(1-c)*B1+c*B2;
        mat1[int(X*d+cx)][int(-Y*d+cy)][0]=R;
        mat1[int(X*d+cx)][int(-Y*d+cy)][1]=G;
        mat1[int(X*d+cx)][int(-Y*d+cy)][2]=B;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
  }
}
float e1(float x,float y)
{
  return 1-abs(sin(1.2*x+0.2*sin(x)+sin(0.55*y)
       +0.2*sin(3.7*y)+0.1*sin(7.9*y)+0.2*sin(1.8*y))
       *sin(1.2*y+0.5*sin(0.42*x)+0.1*cos(5.3*x)+0.2*sin(4.9*x)));
}
float e2(float x,float y)
{
  return 1-abs(sin(2.3*x+0.2*sin(1.7*x)+0.2*sin(3.7*y))
      *sin(2.3*y+0.5*sin(x)+0.1*cos(8*x)+0.2*sin(9*x)));
}
void nisip()
{
float x,y,z,x1,z1,X,Y,
R1=99,G1=79,B1=46,R2=211,G2=176,B2=135,
R,G,B,c,e;
for(z=-9;z<=15;z+=0.5/d)
```

```
  for(x=-13;x<=13;x+=0.5/d)
   {
     y=-13;
     X=d*x/(d-z);
     Y=d*y/(d-z);
     x1=x;z1=2*z;
     e=e1(x1,z1)*e2(x1,z1)*e1(x1+z1,z1-x1);
     c=pow(2,-3*e*e);
     R=c*R1+(1-c)*R2;
     G=c*G1+(1-c)*G2;
     B=c*B1+(1-c)*B2;
     if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
     {
       mat1[int(X*d+cx)][int(-Y*d+cy)][0]=R;
       mat1[int(X*d+cx)][int(-Y*d+cy)][1]=G;
       mat1[int(X*d+cx)][int(-Y*d+cy)][2]=B;
       set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
     }
   }
}
void mare()
{
float x,y,z,x1,z1,X,Y,
R3=41,G3=184,B3=255,R4=255,G4=255,B4=255,R,G,B,c1,c2,e;
for(z=2;z<=20;z+=0.5/d)
  for(x=-10;x<=10;x+=0.5/d)
  {
     y=-10;
     X=d*x/(d-z);
     Y=d*y/(d-z);
     if(int(X*d+cx)>=0&&int(X*d+cx)<800&&int(-Y*d+cy)>=0&&int(-
Y*d+cy)<600)
     {
       R=mat1[int(X*d+cx)][int(-Y*d+cy)][0];
       G=mat1[int(X*d+cx)][int(-Y*d+cy)][1];
       B=mat1[int(X*d+cx)][int(-Y*d+cy)][2];
       c2=(atan(z-7)+PI/2)/PI;
       R=c2*R+(1-c2)*R3;
       G=c2*G+(1-c2)*G3;
       B=c2*B+(1-c2)*B3;
       x1=0.5*x;z1=3*z;
       e=e1(x1,z1)*e2(x1,z1)*e1(x1+z1,z1-x1);
```

```
        c1=pow(2,-3*e*e);
        R=c1*R+(1-c1)*R4;
        G=c1*G+(1-c1)*G4;
        B=c1*B+(1-c1)*B4;
        set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      }
    }
}
void setup()
{
size (800,600);
background (255);
nisip();
melc(0.07,-0.1*PI,0.6*PI,-0.4*PI,0,-13,-4);
melc(0.07,-0.1*PI,-0.5*PI,0,6,-13,0);
melc2(0.15,0,0.6*PI,0.5*PI,1,-13,6);
melc(0.07,-0.1*PI,1.2*PI,0.1*PI,-6,-13,-1);
melc(0.1,0.1*PI,0.7*PI,0.2*PI,-3,-13,3);
mare();
}
```

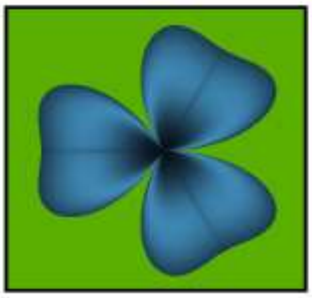

sketch_170328a

Stanjenelul



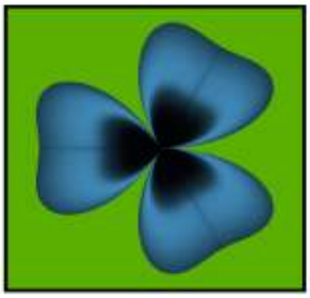
$c1=1-2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}}$

R=(1-c1) R1+c1 R2
G=(1-c1) G1+c1 G2
B=(1-c1) B1+c1 B2



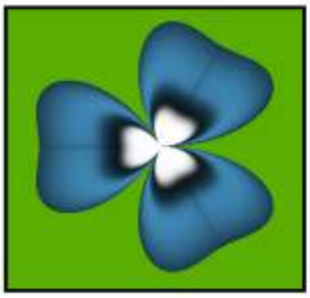
$c1=1-2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}}$
c2=sin(a)

R=(1-c2) R+c2 R2
G=(1-c2) G+c2 G2
B=(1-c2) B+c2 B2



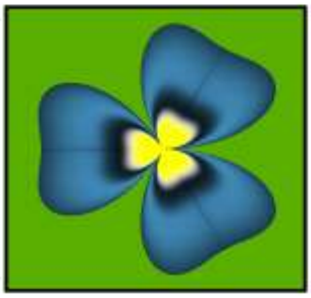
$c1=1-2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}}$
c2=sin(a)
$c3=1-2^{-a^{10}}$

R=(1-c3) R+c3 R2
G=(1-c3) G+c3 G2
B=(1-c3) B+c3 B2



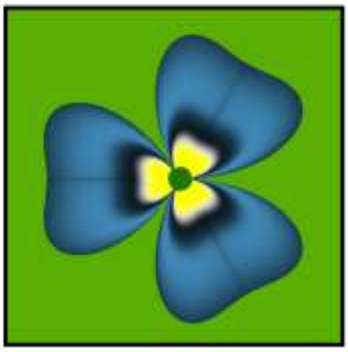
$c1=1-2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}}$
c2=sin(a)
$c3=1-2^{-a^{10}}$
$c4=1-2^{-a^{10}\cdot 3}$

R=(1-c4) R+c4 R3
G=(1-c4) G+c4 G3
B=(1-c4) B+c4 B3



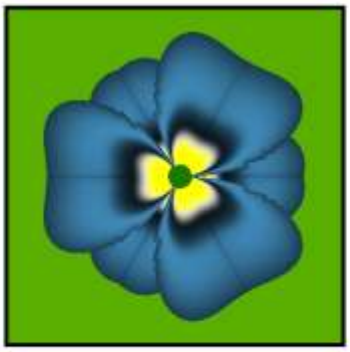
$c1=1-2^{-a^2|\sin(1.5u+\frac{\pi}{2})|^{0.2}}$
c2=sin(a)
$c3=1-2^{-a^{10}}$
$c4=1-2^{-a^{10}\cdot 3}$
$c5=1-2^{-a^{10}\cdot 30}$

R=(1-c5) R+c5 R4
G=(1-c5) G+c5 G4
B=(1-c5) B+c5 B4

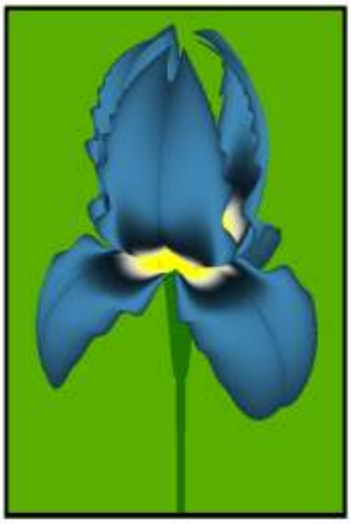


$$r=2.3a(|\sin(1.5u)|+0.2\sin(4.5u))$$
$$x=r\cdot\cos(u)$$
$$y=r\cdot\sin(u)$$
$$u\in[0,2\pi)$$
$$a\in[0,3]$$

$$r=2.3a(|\sin(1.5u)|+0.2\sin(4.5u))$$
$$x=r\cdot\cos(u)(1+0.5\sin((0.5r)^4))|\cos(1.5u)|^{30}$$
$$y=r\cdot\sin(u)(1+0.5\sin((0.5r)^4))|\cos(1.5u)|^{30}$$
$$u\in[0,2\pi)$$
$$a\in[0,3]$$



$$r=2.3a(|\sin(1.5u)|+0.2\sin(4.5u)) \quad ,a\in[0,3]$$
$$x=\begin{cases} r\cdot\cos(u)(1+0.5\sin((0.5r)^4))|\cos(1.5u)|^{30} & ,u\in[0,2\pi) \\ r\cdot\cos(u+\frac{\pi}{3}) & ,u\in[0,4\pi) \end{cases}$$
$$y=\begin{cases} r\cdot\sin(u)(1+0.5\sin((0.5r)^4))|\cos(1.5u)|^{30} & ,u\in[0,2\pi) \\ r\cdot\sin(u+\frac{\pi}{3}) & ,u\in[0,4\pi) \end{cases}$$



$$r=2.3a(|\sin(1.5u)|+0.2\sin(4.5u)) \quad ,a\in[0,3]$$
$$x=\begin{cases} r\cdot\cos(u)(1+0.5\sin((0.5r)^4))|\cos(1.5u)|^{30} & ,u\in[0,2\pi) \\ r\cdot\cos(u+\frac{\pi}{3}) & ,u\in[0,4\pi) \end{cases}$$
$$y=\begin{cases} r\cdot\sin(u)(1+0.5\sin((0.5r)^4))|\cos(1.5u)|^{30} & ,u\in[0,2\pi) \\ r\cdot\sin(u+\frac{\pi}{3}) & ,u\in[0,4\pi) \end{cases}$$
$$z=-0.6\sin(r)|\cos(1.5u)|-\frac{4}{(3r)^4+1}-\frac{7}{(10r)^{40}+1}+\left[\frac{u}{2\pi}\right](0.3\sin(r^2)|\cos(1.5u)|^{\frac{10}{r+1}}-0.35)$$

$$(x,y,z)\longrightarrow(x\cdot\cos(u)-z\cdot\sin(u),\ y\cdot\cos(u)-z\cdot\sin(u),\ d\cdot\sin(u)+z\cdot\cos(u)),\ d=\sqrt{x^2+y^2}$$

Program

```
float[][] mat=new float[800][600];
void init_mat()
{
int i,j;
for(i=0;i<800;i++)
  for(j=0;j<600;j++)
    mat[i][j]=-100;
}
void stanjenel()
{
float x1,y1,z1,d1,u1,x,y,z,a,r,u,X,Y,d=27,cx=400,cy=250,
R1=0,G1=20,B1=40,R2=70,G2=150,B2=200,
R3=255,G3=255,B3=255,R4=255,G4=255,B4=0,R,G,B,c;
for(u=0;u<4*PI;u+=0.1/d)
  for(a=0;a<=3;a+=0.1/d)
  {
    r=1.7*a*(abs(sin(1.5*u))+0.1*abs(sin(3*u)));
    if(u<2*PI)
      r=2.3*a*(abs(sin(1.5*u)+0.2*sin(4.5*u)));
    if(u<2*PI)
    {
      x=r*cos(u+PI/3*int(u/(2*PI)))*(1+0.5*sin(pow(r*0.5,4))*pow(abs(cos(1.5*u)),30));
      y=r*sin(u+PI/3*int(u/(2*PI)))*(1+0.5*sin(pow(r*0.5,4))*pow(abs(cos(1.5*u)),30));
    }
    else
    {
      x=r*cos(u+PI/3*int(u/(2*PI)));
      y=r*sin(u+PI/3*int(u/(2*PI)));
    }
    z=-0.6*sin(r)*abs(cos(u*1.5))
    -(4+0.35*(1-int(u/(2*PI))))/(pow(r*3,4)+1)-0.35*int(u/(2*PI))
    -7/(pow(r*10,40)+1);
    if(u>2*PI)
    {
      z=-0.6*sin(r)*abs(cos(u*1.5))-(4+0.35*(1-int(u/(2*PI))))
      /(pow(r*3,4)+1)-0.35*int(u/(2*PI))-7/(pow(r*10,40)+1)
      +0.3*sin(r*r)*pow(abs(cos(1.5*u)),10/(r+1));
    }
    if(u<2*PI)
      u1=0.25*r;
    else
      u1=-0.03*r*r;
    d1=sqrt(x*x+y*y);
    x1=x*cos(u1)-z*sin(u1);
    y1=y*cos(u1)-z*sin(u1);
    z1=d1*sin(u1)+z*cos(u1);
    X=-0.86*x1+0.86*y1;
    Y=z1-0.5*x1-0.5*y1;
```

```
    if(z1>mat[int(X*d+cx)][int(-Y*d+cy)])
    {
      mat[int(X*d+cx)][int(-Y*d+cy)]=z1;
      c=1-pow(2,-a*a*pow(abs(sin(1.5*u+PI/2)),0.2));
      c*=sin(a);
      c*=1-pow(2,-pow(a,10)*0.03);
      c*=1-pow(2,-pow(a,10)*1);
      R=R1*(1-c)+R2*c;
      G=G1*(1-c)+G2*c;
      B=B1*(1-c)+B2*c;
      c=pow(2,-pow(a,10)*3);
      R=R*(1-c)+c*R3;
      G=G*(1-c)+c*G3;
      B=B*(1-c)+c*B3;
      c=pow(2,-pow(a,10)*30);
      R=R*(1-c)+R4*c;
      G=G*(1-c)+G4*c;
      B=B*(1-c)+B4*c;
      set(int(X*d+cx),int(-Y*d+cy),color(R,G,B));
      if(r<0.5)
        set(int(X*d+cx),int(-Y*d+cy),color(30,120,0));
    }
  }
}
void setup()
{
size(800,600);
background(255);
init_mat();
stanjenel();
}
```