

Wykład 8. Fraktale

1 Żuk Mandelbrota

```

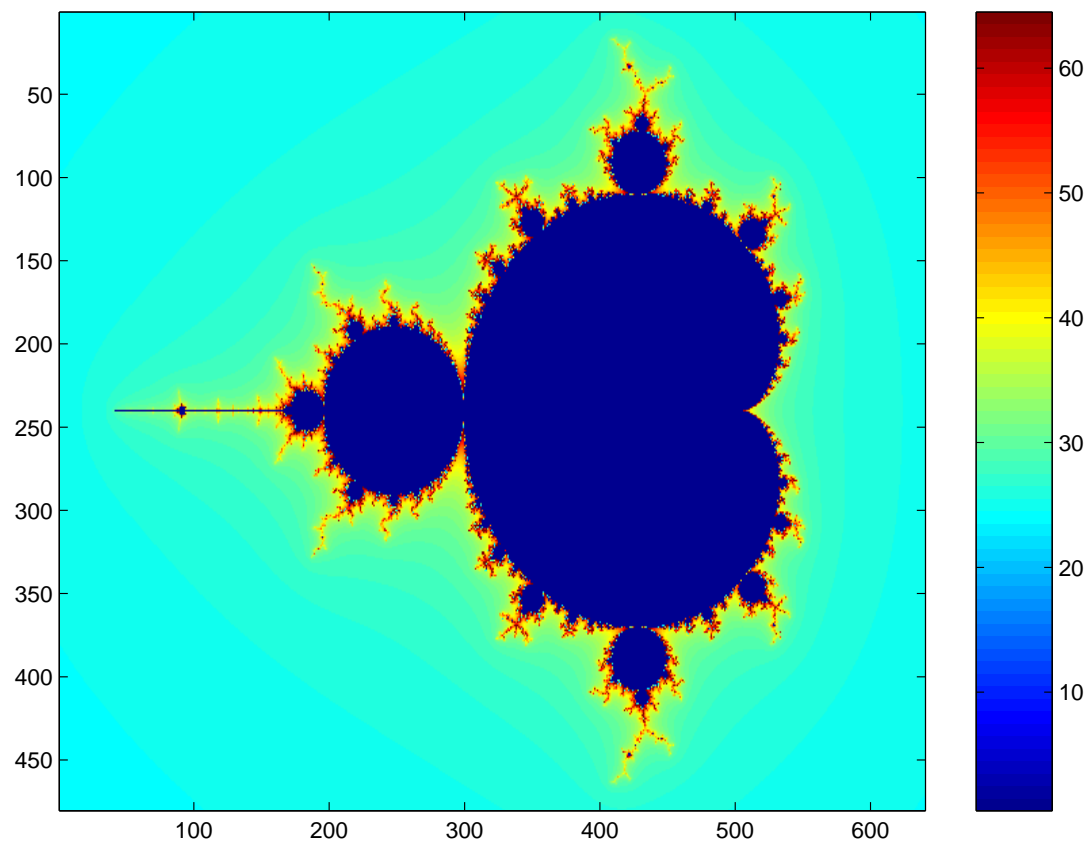
% plik zukkol.m      asz 2004.05.17
% Generowanie kolorowego Zuka Mandelbrota.
% W petli wybierane sa rozne stale zespolone C, wykorzystywane
% w generowaniu ciagu   Z[n+1]:=Z[n]^2 + C.
% okno graficzne otwierane za pomoca 'image' -> os OY opisana w dol.
%-----
clc, clear, close all
xwym=640; ywym=480; %1024;          % rozmiar tablicy danych
ap=-2.2; ak=0.9; da=(ak-ap)/xwym; % zakres   C = a + bi
bp=-1.2; bk=1.2; db=(bk-bp)/ywym;
maksiter=160; maxz=100;          % kryterium ograniczonosci ciagu

mkolor=colormap;   nkolor=size(mkolor,1);
tab=zeros(ywym,xwym);          % 2457600
if 0
    tab=logical(zeros(ywym,xwym)); % 307200
    colormap(flag)              % red, white, blue, black
    image(tab+2)
end
image(tab)
set(gcf,'color','white')
hold on
drawnow

for j=1:ywym
    b=bp+j*db;
    for i=1:xwym
        a=ap+i*da;
        x=0; y=0;
        iter=0;
        while iter<maksiter
            iter=iter+1;
            if x*x+y*y > maxz
                tab(j,i)=mod(iter,nkolor)+20;
                break
            end % if
            y1=2*x*y+b;
            x=x*x-y*y+a;
            y=y1;
        end % while
        if iter==maksiter
            tab(j,i)=1;
        end
    end % i
end % j

image(tab)
%axis off
colorbar
disp('Koniec')
%-----

```



2 Zbiory Julii

```
% plik juliakol.m      asz 2004.05.17
% Generowanie kolorowych zb. Julii
% W petli wybierane sa rozne stale zespolone C, wykorzystywane
% w generowaniu ciagu Z[n+1]:=Z[n]^r+C.
% Mozna wybrac sobie wykladnik potegi r.
% okno graficzne otwierane za pomoca 'image' -> os OY opisana w dol.
%-----
clc, clear, close all
xwym=640;  ywym=480; %1024;      % rozmiar tablicy danych
ap=-1.6; ak=1.6; da=(ak-ap)/xwym; % zakres punktow startowych
bp=-1.2; bk=1.2; db=(bk-bp)/ywym;
maksiter=60;  maxz=10;          % kryterium ograniczonosci ciagu
r=2;          % wykladnik potegi

mkolor=colormap;  nkolor=size(mkolor,1);
close
if 0
  aktkolor=white(1);
  mkolor(1,:)=white(1);

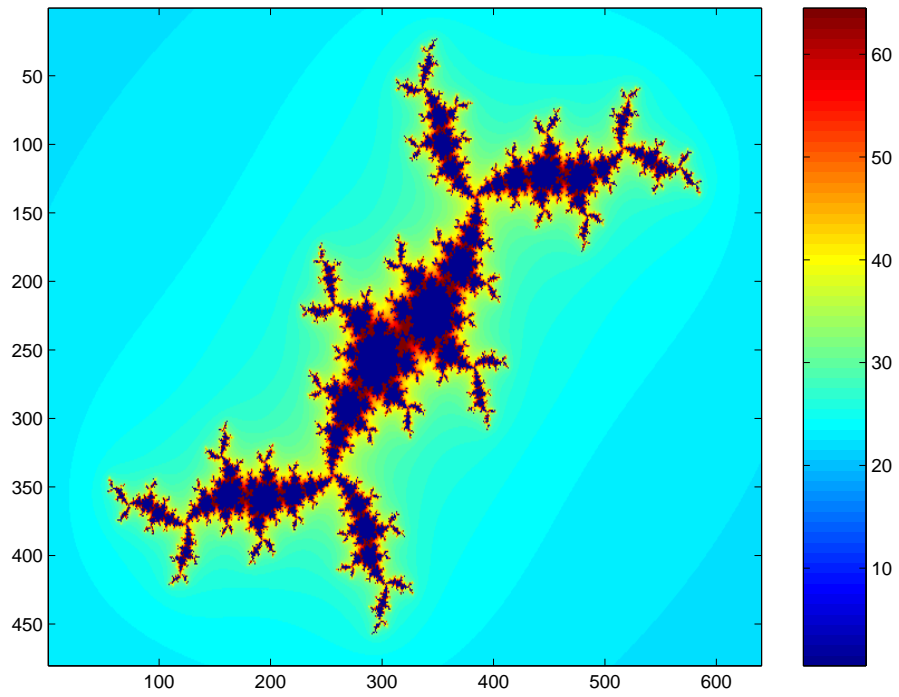
  tab=ones(ywym,xwym)*24;
  image(tab)
```

```
    set(gcf,'color','white')
    hold on
end
tab=ones(ywym,xwym)*24;

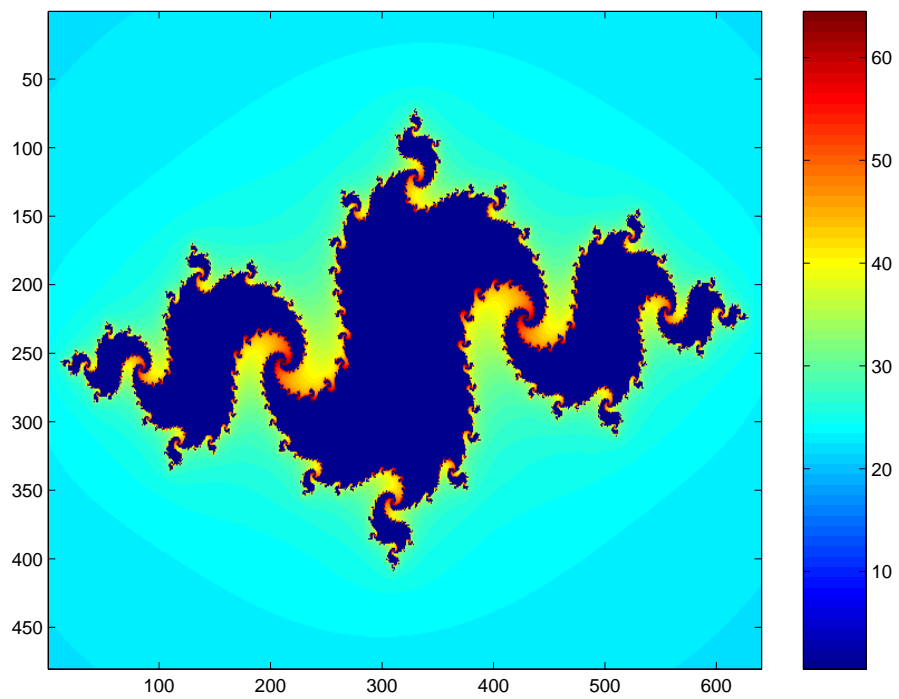
liczba=6; % 32          % liczba + 1  obrazkow zb. Julii
for ka=0:liczba
    a=0-ka/liczba; b=1-ka/liczba;
    for jj=1:ywym
        bb=bp+jj*db;
        for ii=1:xwym
            aa=ap+ii*da;
            x=aa; y=bb;
            iter=0;
            while iter<maksiter
                iter=iter+1;
                if x*x+y*y > maxz
                    tab(jj,ii)=mod(iter,nkolor)+20;
                    break
                end % if
                xr=x; yr=y;
                for n=2:r
                    y1=xr*y+x*yr;
                    xr=x*xr-y*yr;
                    yr=y1;
                end
                if 0
                    z=(x+y*i)^r; xr=real(z); yr=imag(z);
                end
                x=xr+a; y=yr+b;
            end % while
            if iter==maksiter
                tab(jj,ii)=1;
            end
        end % ii
    end % jj
    image(tab)
    %axis off
    colorbar
    disp(['obrazek' num2str(ka) ', Nacisnij klawisz'])
    pause(1)
end % for ka
disp('Koniec')
%-----
```

3 Zadania na pracownię

1. Proszę przygotować program rysujący “na bieżąco” obrazek fraktala.



otrzymany dla $ka=1$



otrzymany dla $ka=5$

4 Krzywa Kocha – rekurencyjna grafika żółwia

```

function Wkoch(n,lw)
% n - stopien rekurencji
% lw - liczba wierzchołkow wielokata
% wywołanie: Wkoch(3,5)
% asz 2004.05.24
%-----
clc, close all
global A
A=(sqrt(3)/6)*[0 1; -1 0]; % macierz dla wielokata
%      0      0.2887
% -0.2887     0

fig1=figure(1);
set(fig1,'DoubleBuffer','on');
set(gcf, 'color','w');
set(gca, 'FontSize', 14, 'Box', 'on', 'Xcolor', 'k', 'Ycolor', 'k');
axis([-1.5 1.5 -1.5 1.5])
plot([-1.4, 1.4],[-1.4, 1.4],'w')
axis equal
title(['platek Kocha(',num2str(n),',',',',num2str(lw),')'],...
      'fontsize', 14)
hold on
for k=0:lw-1
    x=[cos(2*k*pi/lw-pi/2); sin(2*k*pi/lw-pi/2)];
    y=[cos(2*(k+1)*pi/lw-pi/2); sin(2*(k+1)*pi/lw-pi/2)];
    fkoch(x,y,n)
end
%-----

function fkoch(x,y,n)
% x, y - kolumny = konce odcinka,
% n - stopien funkcji
global A
if n==0
    plot([x(1),y(1)], [x(2),y(2)],'k')
    drawnow
else
    x1=(2*x+y)/3;
    fkoch(x,x1,n-1)
    xy=(x+y)/2+A*(y-x);
    fkoch(x1,xy,n-1)
    x2=(x+2*y)/3;
    fkoch(xy,x2,n-1)
    fkoch(x2,y,n-1)
end

```

5 Paprotka Barnsley'a - plik z internetu

```
function fern
%FERN  MATLAB implementation of the Fractal Fern
%  Michael Barnsley, Fractals Everywhere, Academic Press, 1993.
%  This version runs forever, or until stop is toggled.
%  See also: FINITEFERN.

shg
clf reset
set(gcf,'color','white',...  %'menubar','none', ...
    'numbertitle','off','name','Fractal Fern')
x = [.5; .5];
h = plot(x(1),x(2),'.');
darkgreen = [0 2/3 0];
set(h,'markersize',1,'color','black','erasemode','none');
axis([-3 3 0 10])
drawnow
axis off
stop = uicontrol('style','toggle','string','stop', ...
    'background','white');
drawnow

p = [ .85  .92  .99  1.00];
A1 = [ .85  .04; -.04  .85];  b1 = [0; 1.6];
A2 = [ .20 -.26;  .23  .22];  b2 = [0; 1.6];
A3 = [-.15  .28;  .26  .24];  b3 = [0; .44];
A4 = [ 0    0 ;  0    .16];

hold on
cnt = 1;
tic
while ~get(stop,'value')
    r = rand;
    if r < p(1)
        x = A1*x + b1;
    elseif r < p(2)
        x = A2*x + b2;
    elseif r < p(3)
        x = A3*x + b3;
    else
        x = A4*x;
    end
    set(h,'xdata',x(1),'ydata',x(2));
    drawnow
    cnt = cnt + 1;
end
t = toc;
s = sprintf('%8.0f points in %6.3f seconds',cnt,t);
hold on
text(-1.5,-0.5,s,'fontweight','bold');
set(stop,'style','pushbutton','string','close','callback','close(gcf)')
```

* * *