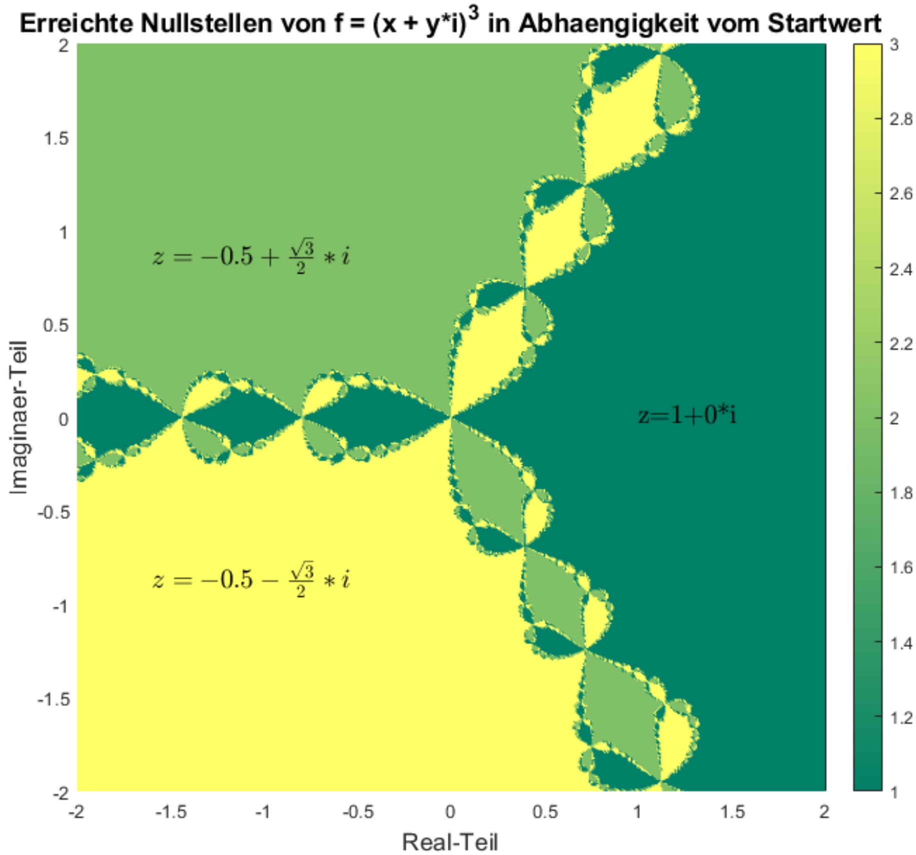
```
% Angewandte Numerik 1, SoSe 2020
% Uebungsblatt 10, Aufgabe 46: Newton-Verfahren fuer Systeme
%
% Naeherungsweise Berechnung der Nullstelle eines Systems nichtlinearer
% Gleichungen f mit Hilfe des Newton-Verfahrens fuer Systeme
%
% Betrachtet wird die Gleichung z^3 = 1
%
% Letzte Aenderung: 30.06.2020

% Cleanup
clearvars;
close all;
clc;

% Initialisierung -------------------------------------------------------
% Toleranz der Funktionswerte
toly  = 1e-10;

% Maximale Anzahl Iterationen
maxIt = 1000;

% Anzahl der Gitterpunkte in x- und y-Richtung
np    = 500;

% Bereich der Startwerte definieren
xmin  = -2;
xmax  =  2;
ymin  = -2;
ymax  =  2;

% Initialisieren der Arrays ---------------------------------------------
%
% Matrix fuer die Nummern der gefundenen Nullstellen initialisieren
% Nullstelle 1: [1;0], 2: [-0.5;sqrt(3)/2], oder 3: [-0.5;-sqrt(3)/2]
Z     = zeros(np,np);

% Matrix initialiseren: Anzahl Iterationen um eine Nullstelle zu erreichen
Iter  = zeros(np,np);

% Erzeugung aequidistanter Punkte
xp    = linspace(xmin, xmax, np);
yp    = linspace(ymin, ymax, np);

% Erzeugung eines Gitternetzes
[X,Y] = meshgrid(xp, yp);
```

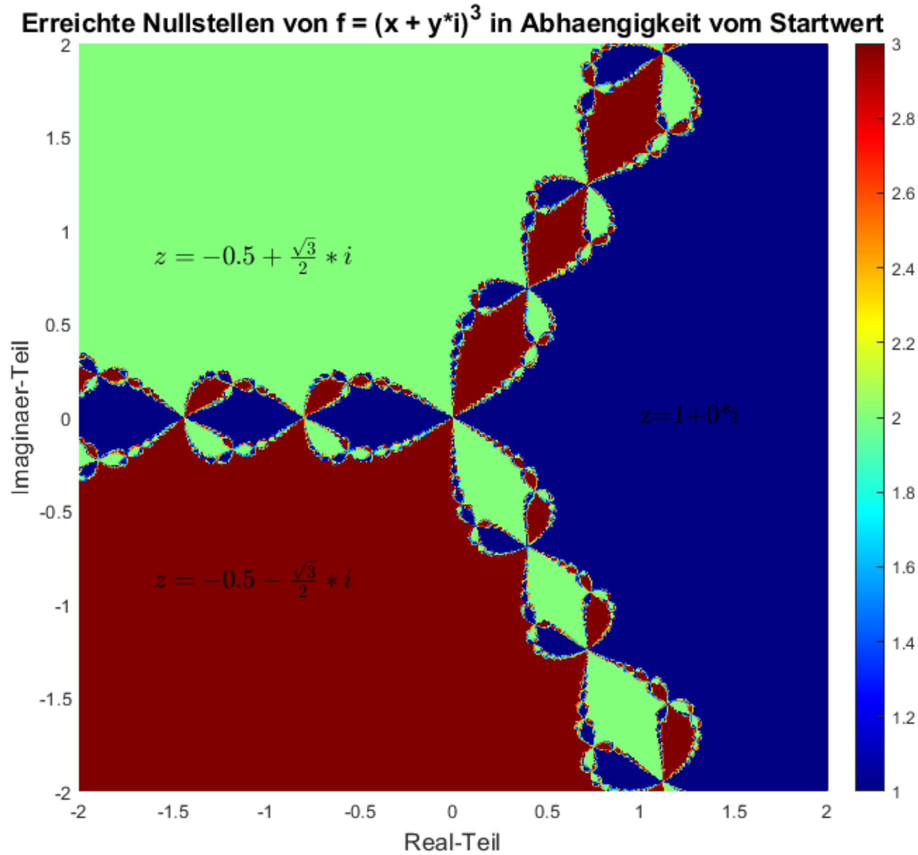
```
% Definition der Funktion, der Jacoby Matrix und der Nullstellen ----------
% Definiere die Nullstellen von z^3=1
root = round([1, -0.5, -0.5; 0, sqrt(3)/2, -sqrt(3)/2],10);

% Definiere die Funktion
% F = z^3 - 1 = (x + iy)^3 - 1 = x^3 + 3*i*x^2*y - 3*x*y^2 - i*y^3 - 1
% Reale Komponente:      u(x,y) = x^3 - 3*x*y^2 - 1
% Imaginaere Komponente: v(x,y) = 3*x^2*y - y^3
% Definiere F in R^2 als F = z^3 - 1 = (x(1) + i*x(2))^3 - 1
F  = @(x)[real( (x(1) + 1i*x(2) ).^3 - 1); ...
          imag( (x(1) + 1i*x(2) ).^3 - 1)];

% Jacoby Matrix erzeugen
% dF = @(x) [real(3*(x(1)+1i*x(2))^2), -imag(3*(x(1)+1i*x(2))^2); ...
%            imag(3*(x(1)+1i*x(2))^2),  real(3*(x(1)+1i*x(2))^2)];
% dF = @(x) [3*x(1)^2 - 3*x(2)^2,  -6*x(1)*x(2)        ; ...
%            6*x(1)*x(2),          3*x(1)^2 - 3*x(2)^2];

% Erzeuge einen symbolischen Spaltenvektor
syms x1 x2;
x   = [x1;x2];
% Erzeuge eine symbolische Variable aus einem function handle
Fs  = eval( extractAfter( func2str(F), '@(x)' ) );
% Erzeuge die Jacoby Matrix mit der Symbolic Toolbox
dFs = jacobian(Fs, x);
% Erzeuge die Jacoby Matrix als function handle mit einem
% Spaltenvektor als Eingabe
dF  = matlabFunction(dFs,'Vars',{x});
clear x x1 x2;

% Bestimmung der Nullstellen aller Punkte mit der Methode -----------------
for j = 1:np
  for k = 1:np
    x0        = [X(j,k); Y(j,k)];
    [xk]      = newtonSys(F, dF, x0, toly, maxIt);
    xend      = round(xk(:,end),10);
    Iter(j,k) = size(xk,2);
    temp      = find( all( root==xend ) );
    if ~isempty(temp)
      Z(j,k) = temp;
    end
  end
end

% Plotten der gefundenen Nullstellen in Abhaengigkeit ---------------------
```

```
% von den Startwerten mit Colormap summer
hf1 = figure( 'Name', 'Angewandte Numerik 1, Blatt 10, Aufgabe 46', ...
  'NumberTitle', 'off', 'Units', 'normalized', 'MenuBar', 'None',      ...
  'Position', [0.2, 0.07, 0.6, 0.85] );

% Checkerboard Plot der gefundenen Nullstellen in Abhaengigkeit von
% den Startwerten
pcolor(X,Y,Z);
colormap(summer);
shading interp;
colorbar;

% Textlabel
text(1,0,'z=1+0*i', 'FontSize',14, 'Interpreter','latex');
text(-1.6,  sqrt(3)/2, '$z=-0.5+\frac{\sqrt{3}}{2}*i$', 'FontSize',14, ...
  'Interpreter','latex');
text(-1.6, -sqrt(3)/2, '$z=-0.5-\frac{\sqrt{3}}{2}*i$', 'FontSize',14, ...
  'Interpreter','latex');

% Achsen, Label und Titel
axis square;
xlabel('Real-Teil', 'FontSize',13);
ylabel('Imaginaer-Teil', 'FontSize',13);
title(['Erreichte Nullstellen von f = (x + y*i)^3 in Abhaengigkeit ', ...
  'vom Startwert'], 'FontSize',14);

% Plotten der gefundenen Nullstellen in Abhaengigkeit --------------------
% von den Startwerten mit Colormap jet
hf2 = figure( 'Name', 'Angewandte Numerik 1, Blatt 10, Aufgabe 46', ...
  'NumberTitle', 'off', 'Units', 'normalized', 'MenuBar', 'None',      ...
  'Position', [0.2, 0.07, 0.6, 0.85] );

% Checkerboard Plot der gefundenen Nullstellen in Abhaengigkeit von
% den Startwerten
pcolor(X,Y,Z);
colormap(jet);
shading interp;
colorbar;

% Textlabel
text(1,0,'z=1+0*i', 'FontSize',14, 'Interpreter','latex');
text(-1.6,  sqrt(3)/2, '$z=-0.5+\frac{\sqrt{3}}{2}*i$', 'FontSize',14, ...
  'Interpreter','latex');
text(-1.6, -sqrt(3)/2, '$z=-0.5-\frac{\sqrt{3}}{2}*i$', 'FontSize',14, ...
  'Interpreter','latex');
```

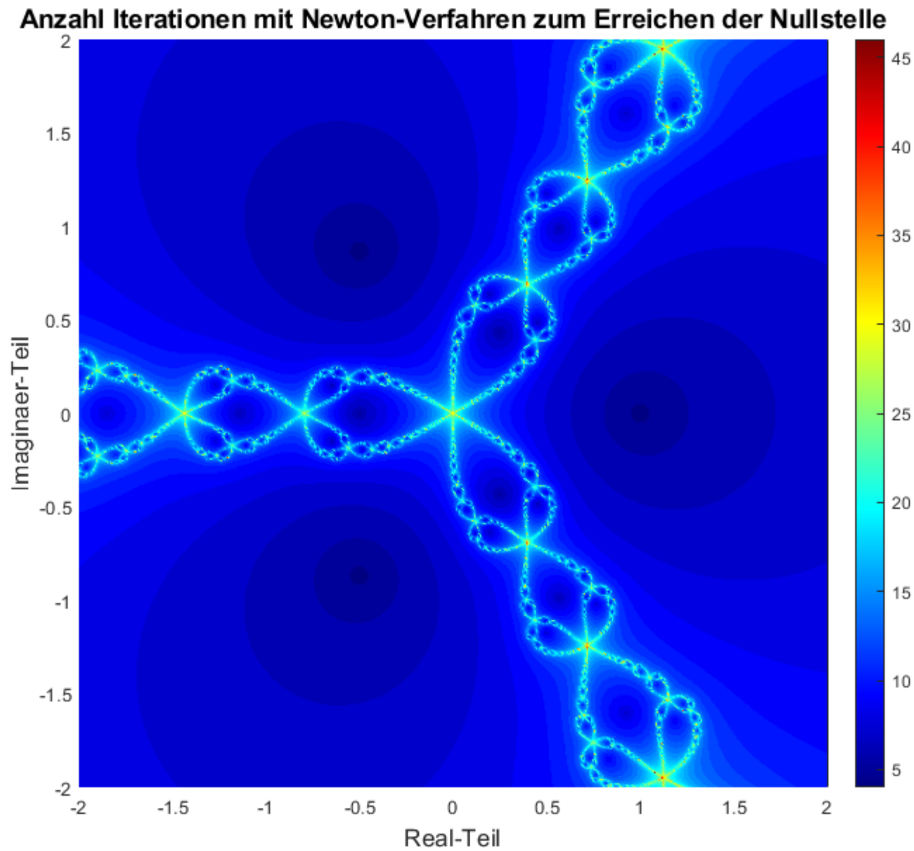
```
% Achsen, Label und Titel
axis square;
xlabel('Real-Teil', 'FontSize',13);
ylabel('Imaginaer-Teil', 'FontSize',13);
title(['Erreichte Nullstellen von f = (x + y*i)^3 in Abhaengigkeit ', ...
  'vom Startwert'], 'FontSize',14);

% Plot: Anzahl der Iterationen um die Nullstellen -------------------------
% zu finden in Abhaengigkeit von den Startwerten
hf3 = figure( 'Name', 'Angewandte Numerik 1, Blatt 10, Aufgabe 46', ...
  'NumberTitle', 'off', 'Units', 'normalized', 'MenuBar', 'None',      ...
  'Position', [0.2, 0.07, 0.6, 0.85] );

% Checkerboard Plot: Anzahl der Iterationen um die Nullstellen zu finden
pcolor(X,Y,Iter);
colormap(jet);
shading interp;
colorbar;

% Achsen, Label und Titel
axis square;
xlabel('Real-Teil', 'FontSize',13);
ylabel('Imaginaer-Teil', 'FontSize',13);
title(['Anzahl Iterationen mit Newton-Verfahren zum Erreichen ', ...
  'der Nullstelle'], 'FontSize',14);
```



Erreichte Nullstellen von f = (x + y*i)^3 in Abhaengigkeit vom Startwert
$z = -0.5 + \frac{\sqrt{3}}{2} * i$
$z = -0.5 - \frac{\sqrt{3}}{2} * i$
z=1+0*i
Imaginaer-Teil
Real-Teil
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1



Erreichte Nullstellen von f = (x + y*i)^3 in Abhaengigkeit vom Startwert
$z = -0.5 + \frac{\sqrt{3}}{2} * i$
$z = -0.5 - \frac{\sqrt{3}}{2} * i$
z=1+0*i
Imaginaer-Teil
Real-Teil


Anzahl Iterationen mit Newton-Verfahren zum Erreichen der Nullstelle
Imaginaer-Teil
Real-Teil
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
45
40
35
30
25
20
15
10
5