
Table of Contents

Aufgabe 8.36	1
Aufgabe 8.37	2
Aufgabe 8.40	4

Aufgabe 8.36

dbtype `aufgabe836.m`
aufgabe836

```
1      A = [1, 3, 0, 0;  
2          3, -6, 1, 0;  
3          0, 1, 3, 1;  
4          0, 0, 1, 2];  
5      A  
6      disp('Eigenwerte von A:')  
7      eig(A)  
8  
9      %f(x)=(-77 + 103 x - 36 x^2 + x^4)/((-3 + x)*(-2 + x)*(-1 +  
10         x)*(8 + x))  
10     c = [1,0,-36,103,-77];  
11     disp('Nullstellen von p')  
12     roots(c)
```

A =

```
1      3      0      0  
3     -6      1      0  
0      1      3      1  
0      0      1      2
```

Eigenwerte von A:

ans =

```
-7.1971  
1.3656  
2.0989  
3.7325
```

Nullstellen von p

ans =

```
-7.1971  
3.7325  
2.0989  
1.3656
```

Aufgabe 8.37

a

```
dbtype heron.m
```

```
% b
```

```
dbtype leastSquares.m
```

```
dbtype aufgabe837b.m
```

```
aufgabe837b
```

```
%c
```

```
dbtype aufgabe837c.m
```

```
aufgabe837c
```

```
1 function [X] = heron(v, a, kmax, x0)
2 % Output X=[x^1...x^kmax]^T
3 X = x0;
4 for k=1:kmax
5     x0=1/v * ((v-1)*x0 + a/(x0^(v-1)));
6     X=[X;x0];
7 end
8
9 end

1 function [C,p] = leastSquares(x,y)
2
3 A = [ones(length(x),1),x];
4
5 b = A'*y;
6 t = A'*A\b;
7
8 C = exp(t(1));
9 p = t(2);
10
11 end

1 kmax = 16;
2 x0 = 1;
3 v = 3;
4 a = 3375;
5 X = heron(v,a,kmax,x0);
6
7 xhat = a^(1/3);
8 d = abs(X-xhat);
9 length(d);
10
11 y = d(3:kmax+1);
12 x = d(2:kmax);
13
14 [C,p] = leastSquares(log(x),log(y));
15 fprintf('Konvergenzordnung p=%1.4f, Konstante C=%2.4f\n',p,C)
```

```

16
17     loglog(1:kmax-1,y,'rx',1:kmax-1,C*x.^p,'b-')
18
Konvergenzordnung p=1.7850, Konstante C=0.0149

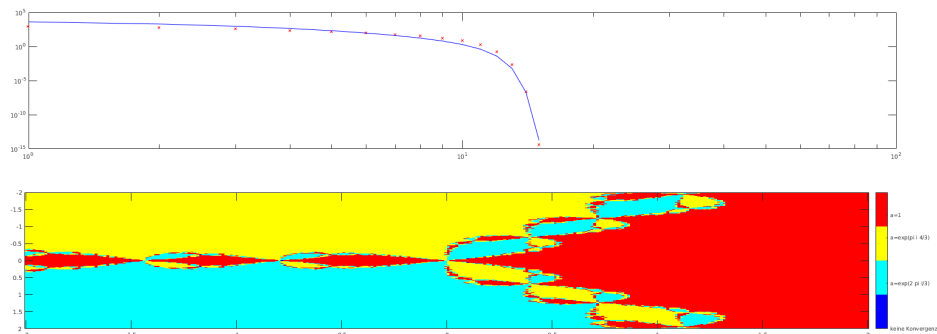
1     r=linspace(-2,2,300);
2     [U,V]=meshgrid(r,r);
3     Z=U+1i*V;
4
5     %exakte Loesungen
6     z1=1;
7     z2=exp(2*pi*1i/3);
8     z3=exp(pi*1i*4/3);
9
10    Z1=[]; %Startwerte die nach z1 konvergieren
11    Z2=[]; %Startwerte die nach z2 konvergieren
12    Z3=[]; %Startwerte die nach z3 konvergieren
13
14    kmax=30;
15    v=3;
16    a=1;
17    myeps=0.1;
18
19    C=zeros(size(Z));
20    for k=1:size(Z,1)
21        for j=1:size(Z,2)
22            z=Z(j,k);
23            X = heron(v,a,kmax,z);
24
25            t1=norm(X(end)-z1);
26            t2=norm(X(end)-z2);
27            t3=norm(X(end)-z3);
28
29
30            if(t1<myeps)
31                C(j,k)=4;
32                Z1=[Z1;z];
33            elseif(t2<myeps)
34                C(j,k)=2;
35                Z2=[Z2;z];
36            elseif(t3<myeps)
37                C(j,k)=3;
38                Z3=[Z3;z];
39            else
40                fprintf('Keine Konvergenz nach kmax=%d für z=%f  %f i\n',kmax,real(z),sign(imag(z))*abs(imag(z)))
41                C(j,k)=1;
42            end
43        end
44
45    end
46
47    figure(2)
48    colormap(jet(4))

```

```

49   imagesc(r,r,C)
50   colorbar
51   colorbar('YTickLabel',{'keine Konvergenz','','a=exp(2 pi
    i/3)','','a=exp(pi i 4/3)','a=1',''})
Keine Konvergenz nach kmax=30 für z=-1.505017 -0.046823 i
Keine Konvergenz nach kmax=30 für z=-1.505017 +0.046823 i
Keine Konvergenz nach kmax=30 für z=-1.170569 -0.180602 i
Keine Konvergenz nach kmax=30 für z=-1.170569 +0.180602 i
Keine Konvergenz nach kmax=30 für z=-0.782609 -0.006689 i
Keine Konvergenz nach kmax=30 für z=-0.782609 +0.006689 i
Keine Konvergenz nach kmax=30 für z=0.100334 -0.060201 i
Keine Konvergenz nach kmax=30 für z=0.100334 +0.060201 i
Keine Konvergenz nach kmax=30 für z=0.140468 -0.073579 i
Keine Konvergenz nach kmax=30 für z=0.140468 +0.073579 i
Keine Konvergenz nach kmax=30 für z=0.448161 -0.715719 i
Keine Konvergenz nach kmax=30 für z=0.448161 +0.715719 i
Keine Konvergenz nach kmax=30 für z=0.662207 -1.210702 i
Keine Konvergenz nach kmax=30 für z=0.662207 +1.210702 i
Keine Konvergenz nach kmax=30 für z=0.715719 -1.277592 i
Keine Konvergenz nach kmax=30 für z=0.715719 +1.277592 i

```



Aufgabe 8.40

```

dbtype aufgabe840.m
aufgabe840

```

```

1   f=inline('x^3-x-1','x');
2   f_abl=inline('3*x^2-1','x');
3
4   err=1e-4;
5   kmax=200;
6   %f(a)*f(b)<0
7   a=0;
8   b=2;
9   x0=0; %x0 in (a,b)
10
11
12  %Newton:
13  x_newt_0 = x0;
14  for k_newt = 1:200

```

```

15         x_newt = x_newt_0 - f(x_newt_0)/(f_abl(x_newt_0));
16         if( abs(x_newt - x_newt_0) <= err )
17             break;
18         end
19         x_newt_0 = x_newt;
20     end
21
22
23     %Bisektion:
24     a_bis = a;
25     b_bis = b;
26     %geg a<b mit f(a)f(b) < 0
27     for k_bis = 1:200
28         if(abs(b_bis-a_bis) <= err)
29             break;
30         end
31         x_bis = 1/2*(a_bis+b_bis);
32         if( f(x_bis)*f(a_bis) >0)
33             a_bis = x_bis;
34         else
35             b_bis = x_bis;
36         end
37     end
38
39
40     %Sekantenverfahren:
41
42     x_sek_1 = [a,b];
43     k_sek = 0;
44     x_sek = b;
45
46     for k_sek = 1:200
47         if( abs(x_sek_1(2) - x_sek_1(1)) > err)
48             break;
49         end
50         x_sek = x_sek_1(1) - f(x_sek_1(2))*(x_sek_1(2)-x_sek_1(1))/
(f(x_sek_1(2))-f(x_sek_1(1)));
51         x_sek_1 = [x_sek_1(2),x_sek];
52     end
53
54     %Regula Falsi:
55     a_regfal = a;
56     b_regfal = b;
57     for k_regfal = 1:200
58         x_regfal = (a_regfal * f(b_regfal) - b_regfal *
f(a_regfal) ) / (f(b_regfal) - f(a_regfal));
59
60
61         if( f(x_regfal) * f(a_regfal) > 0)
62             a_regfal = x_regfal;
63         else
64             b_regfal = x_regfal;
65         end
66

```

```

67         if(abs(x_regfal - a_regfal) <= err || abs(x_regfal -
        b_regfal) <= err)
68             break;
69         end
70     end
71
72     %%Fixpunktiteration
73     %%f(x)=0 <==>Loese Phi(x)=x nach x, mit Phi geeignet
74     %a)
75     Phi1 = inline('x^3-1','x');
76     x1_fix_0 = x0;
77     k1_fix = 0;
78
79     for k1_fix = 1:200
80         x1_fix = Phi1(x1_fix_0);
81         x1_fix_0 = x1_fix;
82         if(abs(x1_fix-x1_fix_0) <= err)
83             break;
84         end
85     end
86
87     %b)
88     %Phi_2
89     Phi2 = inline('(x+1)^(1/3)','x');
90     x2_fix_0 = x0;
91     for k2_fix = 1:200
92         x2_fix = Phi2(x2_fix_0);
93         x2_fix_0 = x2_fix;
94         if(abs(x2_fix - x2_fix_0) <= err)
95             break;
96         end
97     end
98
99
100
101     fprintf('-----
    \n')
102     fprintf('f=x^3-x-1  Nullstelle: x=1.32472\n')
103     fprintf('-----
    \n')
104     fprintf('Newton                ,   k=%2.0f   x=%2.4d \n',
        k_newt,x_newt)
105     fprintf('Sekantenverfahren    ,   k=%2.0f   x=%2.4d \n',
        k_sek,x_sek)
106     fprintf('Bisektion            ,   k=%2.0f   x=%2.4d \n',
        k_bis,x_bis)
107     fprintf('Regula Falsi          ,   k=%2.0f   x=%2.4d \n',
        k_regfal,x_regfal)
108     fprintf('Fixpunktiteration a),   k=%2.0f   x=%2.4d \n',
        k1_fix,x1_fix)
109     fprintf('Fixpunktiteration b),   k=%2.0f   x=%2.4d \n',
        k2_fix,x2_fix)

```

```
110
111
112
113
```

```
-----
f=x^3-x-1  Nullstelle: x=1.32472
-----
```

```
Newton           ,  k=21  x=1.3247e+00
Sekantenverfahren ,  k= 1  x=0002
Bisektion         ,  k=16  x=1.3248e+00
Regula Falsi      ,  k= 1  x=3.3333e-01
Fixpunktiteration a), k= 1  x=-0001
Fixpunktiteration b), k= 1  x=0001
```

Published with MATLAB® R2016a