Prof. W. Gander **Masaryk University Fall 2014**

Nonlinear Euqations

1. Bisection-Algorithm.

   (a) Improve the function `Bisekt`. Your `[x,y]=Bisection(f,a,b,tol)` should also compute a zero for functions with $f(a) > 0$ and $f(b) < 0$ to a given tolerance `tol`. Be careful to stop the iteration in case the user asks for a too small tolerance! If by the bisection process we arrive at an interval (a, b) which does not contain a machine number anymore then it is high time to stop the iteration.

   Solution:

```
function [x,y]=Bisection(f,a,b,tol)
% BISECTION computes a root of a scalar equation
%   [x,y]=Bisection(f,a,b,tol) finds a root x of the scalar function
%   f in the interval [a,b] up to a tolerance tol. y is the
%   function value at the solution

fa=f(a); v=1; if fa>0, v=-1; end;
if fa*f(b)>0
  error('f(a) and f(b) have the same sign')
end
if (nargin<4), tol=0; end;
x=(a+b)/2;
while (b-a>tol)  & ((a < x) & (x<b))
  if v*f(x)>0, b=x; else a=x; end;
  x=(a+b)/2;
end
if nargout==2, y=f(x); end;
```

   (b) Solve with bisection the equations

$$a) \quad x^x = 50 \qquad b) \quad \ln(x) = \cos(x) \qquad c) \quad x + e^x = 0.$$

   Hint: a starting interval is easy to find by sketching the functions involved.

   Solution:

   a) The function x^x is monotonically increasing. Since $1^1 = 1$ and $4^4 = 256$ the values $a = 1$ and $b = 4$ can be used for the bisection. The solution becomes

```
>> [x,f]=Bisection(@(x) x^x-50,1,4)
x =
   3.287262195355581
f =
     7.105427357601002e-15
```

b) Drawing the functions $\ln(x)$ and $\cos(x)$ we see that their cutting point is in the interval $(0, \pi/2)$, thus

```
>> [x,f]=Bisection(@(x) log(x)-cos(x),0,pi)
x =
   1.302964001216012
f =
    -2.220446049250313e-16
```

c) We write the equation $e^x = -x$ and from the graph of the two functions we get the interval $(-1, 0)$ for the solution, so

```
>> [x,f]=Bisection(@(x) exp(x)+x,-1,0)
x =
  -0.567143290409784
f =
    -1.110223024625157e-16
```

2. Find x such that

$$f(x) = \int_0^x e^{-t^2} dt - 0.5 = 0.$$

Hint: the integral cannot be evaluated analytically, so expand it in a series and integrate. Write a function `f(x)` to evaluate the series.

Since a function evaluation is expensive (summation of the Taylor series) but the derivatives are cheap to compute, a higher order method is appropriate. Solve this equation with Newton's or Halley's method.

Solution:

Take the series for e^x, substitute $x = -t^2$ and integrate to get the expansion

$$\int_0^x e^{-t^2} dt = x - \frac{x^3}{1!\,3} + \frac{x^5}{2!\,5} - \frac{x^7}{3!\,7} + \frac{x^9}{4!\,9} \mp \cdots \tag{1}$$

For evaluating the series we introduce the expressions

$$ta := (-1)^{i-1} \frac{x^{2i-1}}{(i-1)!} \qquad t := (-1)^i \frac{x^{2i+1}}{i!}$$

then $t = -ta * x^2/i$ and the partial sum is updated by $s_{\text{new}} = s_{\text{old}} + t/(2*i+1)$. We will stop the summation when $s_{\text{new}} = s_{\text{old}}$. Thus we get

```
function y = ff(x);
% is used in IntegralExp.m
t = x; snew = x; sold=0; i=0;
while sold ~= snew
  i = i+1;
  sold = snew;
```

```
  t = -t*x^2/i;
  snew = sold+t/(2*i+1);
end
y = snew;

% Solve \int_{0}^{x} e^{-t^2}dt - 0.5 = 0 with Newton and Halley
% use ff.m to compute Taylor series
%
format compact
format long

disp(' Newton')
x = 1; xa=2;
while abs(xa-x)>1e-10
  xa=x;
  y = ff(x)-0.5; ys = exp(-x^2);
  x = x - y/ys
end

disp('Halley')
x = 1; xa=2;
while abs(xa-x)>1e-10
  xa=x;
  y = ff(x)-0.5; ys = exp(-x^2); yss = -2*x*ys;
  t = y*yss/ys^2;
  x = x - y/ys/(1-0.5*t)
end

>> IntegralExp
 Newton
x =
   0.329062444950818
x =
   0.532365165339031
x =
   0.550852862865461
x =
   0.551039408434969
x =
   0.551039427609027
x =
   0.551039427609027
Halley
x =
   0.598466410057177
x =
   0.551087168834467
x =
   0.551039427609074
```

```
x =
   0.551039427609027
```

3. Compute the intersection points of an ellipsoid with a sphere and a plane. The ellipsoid has the equation

$$\left(\frac{x_1}{3}\right)^2 + \left(\frac{x_2}{4}\right)^2 + \left(\frac{x_3}{5}\right)^2 = 3.$$

The plane is given by $x_1 - 2x_2 + x_3 = 0$ and the sphere has the equation $x_1^2 + x_2^2 + x_3^2 = 49$

   (a) How many solutions do you expect for this problem?

   (b) Solve the problem with the `solve` and `fsolve` commands from MAPLE.

   (c) Write a MATLAB script to solve the three equations with Newton's method. Vary the initial values so that you get all the solutions.

Solution:

   (a) The intersection of the plane with the sphere is a circle and with the ellipsoid we get an ellipse. The intersection points are therefore the intersections of a circle with an ellipse. We expect thus in general 4 different solutions. Looking at the equations we notice that with a solution $\mathbf{x}$ also $-\mathbf{x}$ is a solution. Thus if we have found two different solutions then the two remaining solutions are given by changing the signs.

   (b) With the MAPLE commands

```
eqs:={(x1/3)^2+(x2/4)^2+(x3/5)^2=3,x1-2*x2+x3=0,x1^2+x2^2+x3^2=49};
solve(eqs,{x1,x2,x3});
```

   we obtain

$$\begin{aligned} x1 &= -\frac{31634}{247975}\left(RootOf\left(-3621220\,\_Z^2+63268\,\_Z^4+50197225\right)\right)^3 \\ &\quad +\frac{180746}{49595}\,RootOf\left(-3621220\,\_Z^2+63268\,\_Z^4+50197225\right), \\ x2 &= -\frac{15817}{247975}\left(RootOf\left(-3621220\,\_Z^2+63268\,\_Z^4+50197225\right)\right)^3 \\ &\quad +\frac{230341}{99190}\,RootOf\left(-3621220\,\_Z^2+63268\,\_Z^4+50197225\right), \\ x3 &= RootOf\left(-3621220\,\_Z^2+63268\,\_Z^4+50197225\right) \end{aligned}$$

   Using the numerical solver MAPLE delivers only one solution

```
fsolve(eqs,{x1,x2,x3});
```

$$\{x1 = -3.101446015,\, x2 = -3.977629342,\, x3 = -4.853812670\}$$

   (c) Programming in MATLAB Newton's algorithm we obtain

```
% Computing the intersection points
% of an ellipsoid, a sphere and a plane

k=0;
xold=[0,0,0]';
x=[-4 1 6]'     % x=-[-4 1 6]' % for neg solution
% x=[1 1 1]'    % x=-[1 1 1]'
while norm(x-xold)>1e-12*norm(x)
  xold=x; k=k+1;
  f=[(x(1)/3)^2+(x(2)/4)^2+(x(3)/5)^2-3
      x(1)^2+x(2)^2+x(3)^2-49
      x(1)-2*x(2)+x(3)];
  J=[2*x(1)/9 x(2)/8 2*x(3)/25
     2*x(1) 2*x(2)  2*x(3)
     1  -2 1];
  h=-J\f; x=x+h; norm(h)
 end
k,x
```

With the starting vector $\mathbf{x} = [1, 1, 1]$ we get convergence in 8 steps:

```
x =
     1
     1
     1
ans =
     1.414125519548956e+01
ans =
     6.353644140517445e+00
ans =
     2.186622967262920e+00
ans =
     3.581210720347850e-01
ans =
     1.070558121458718e-02
ans =
     9.718696508742296e-06
ans =
     8.014862832469677e-12
ans =
     1.199111836538749e-15
k =
     8
x =
     3.101446014850100e+00
     3.977629342271496e+00
     4.853812669692894e+00
```

The quadratic convergence is visible from the printout of the norm of the correction vector $\mathbf{h}$. Using the starting vector $\mathbf{x} = [-4, 1, 6]$ we converge to the second solution in 4 steps

```
x =
    -4
     1
     6
ans =
     2.872708652671924e-01
ans =
     7.067873458438775e-03
ans =
     5.582183986704225e-06
ans =
     4.059746603817636e-12
k =
     4
x =
    -3.781770586037488e+00
     1.010696349931241e+00
     5.803163285899970e+00
```

4. Modify the fractal program by replacing $f(z) = z^3 - 1$ with the function

$$f(z) = z^5 - 1.$$

   (a) Compute the 5 zeros of f using the command `roots`.

   (b) In order two distinguish the 5 different numbers, study the imaginary parts of the 5 zeros. Invent a transformation such that the zeros are replaced by 5 different positive integer numbers.

Solution: We first compute the zeros of $z^5 - 1$. The coefficients of the polynomial $z^5 - 1$ are

```
p=[1 0 0 0 0 -1]
```

With the function `roots` we can compute the zeros

```
>> W=roots(p)
W =
 -0.809016994374948 + 0.587785252292473i
 -0.809016994374948 - 0.587785252292473i
  0.309016994374947 + 0.951056516295152i
  0.309016994374947 - 0.951056516295152i
  1.000000000000000 + 0.000000000000000i
```

If we multiply the imaginary part by 2 we get

```
>> 2*imag(W)
ans =
   1.175570504584946
```

```
  -1.175570504584946
   1.902113032590305
  -1.902113032590305
                   0
```

Now we can add 3 and round the result to get

```
>> round(2*imag(W)+3)
ans =
     4
     2
     5
     1
     3
```

```
n=1000; m=30;
x=-1:2/n:1;
[X,Y]=meshgrid(x,x);
Z=X+1i*Y;                      % define grid for picture
for i=1:m                      % perform m iterations in parallel
  Z=Z-(Z.^5-1)./(5*Z.^4);      % for all million points
end;                           % each element of Z contains one root
a=10;
image(round(2*imag(Z)+3)*a);
```