Prof. Dr. E. Kostina
M. Sc. H. Stibbe

Ruprecht-Karls-Universität
Institut für Angewandte Mathematik

Linear Optimization - WS 2017/2018

Solution 1

MATLAB-Excercises

M1

a) $\mathtt{c} = \mathtt{u} * \mathtt{B} = \begin{pmatrix} 106 & 76 & 232 \end{pmatrix}$,

b) $\mathtt{d} = \mathtt{B} * \mathtt{u}' = \begin{pmatrix} 116 \\ 152 \\ 170 \end{pmatrix}$,

c) $\mathtt{e} = \mathtt{c} * \mathtt{d} = 63288$,

d) $\mathtt{F} = \mathtt{d} * \mathtt{c} = \begin{pmatrix} 12296 & 8816 & 26912 \\ 16112 & 11552 & 35264 \\ 18020 & 12920 & 39440 \end{pmatrix}$,

e) $\mathtt{G} = (\mathtt{c}' * \mathtt{d}')' = F$.

M2

$$A = \begin{pmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{pmatrix}, \qquad y = \begin{pmatrix} 2 \\ 8 \\ 10 \end{pmatrix}$$

Matlab-Command: `lookfor 'matrix inverse'` (with apostrophes)

$$\mathtt{x} = \mathtt{inv(A)} * \mathtt{y} = \begin{pmatrix} -1 \\ 2 \\ 2 \end{pmatrix}$$

M3

$$2x^2 + 4x - 6 = 0 \quad \Leftrightarrow \quad x^2 + 2x - 3 = 0 \qquad \Rightarrow \qquad x_{1,2} = -1 \pm 2$$

Matlab-Input

```
a=[2;4;-6];
roots(a)
```

yields to the zeros -3 and 1

M4

a)
```
e = 0;
for n = 0:4
   e = e + 1/factorial(n)
end
```
yields to

$$e_0 = 1, \quad e_1 = 2, \quad e_3 = 2.5, \quad e_4 = 2.6667, \quad e_5 = 2.7083$$

b) `eM = exp(1)` yields to $e_{Mat} = 2.7183$ and
`eta = abs((e-eM)/eM)*100` the relative error $\eta_{rel} = 0.3660$ (in %)

c)
```
e=0;
n=0;
eta=1;
while eta > 10^(-6),
   e = e + 1/factorial(n)
   eta = abs((e-eM)/eM)*100
   n=n+1
end
```
yields to the iteration series

$$
\begin{aligned}
n = 0 :&\ e = 1, && \eta_{rel} = 63.2121,\\
n = 1 :&\ e = 2, && \eta_{rel} = 26.4241,\\
n = 2 :&\ e = 2.5, && \eta_{rel} = 8.0301,\\
n = 3 :&\ e = 2.6667, && \eta_{rel} = 1.8988,\\
n = 4 :&\ e = 2.7083, && \eta_{rel} = 0.3660,\\
n = 5 :&\ e = 2.7167, && \eta_{rel} = 0.0594,\\
n = 6 :&\ e = 2.7181, && \eta_{rel} = 0.0083,\\
n = 7 :&\ e = 2.7183, && \eta_{rel} = 0.0010,\\
n = 8 :&\ e = 2.7183, && \eta_{rel} = 1.1252e-04,\\
n = 9 :&\ e = 2.7183, && \eta_{rel} = 1.1143e-05,\\
n = 10 :&\ e = 2.7183, && \eta_{rel} = 1.0048e-06,\\
n = 11 :&\ e = 2.7183, && \eta_{rel} = 8.3161e-08
\end{aligned}
$$

d)
```
x = ...;
e=0;
eM = exp(x); n=0;
eta=1;
while eta > 10^(-6),
   e = e + (x^n)/factorial(n)
   eta = abs((e-eM)/eM)*100
   n=n+1
end
```

Aufgabe M5

```
function [x1,x2] = nullst(a)
if a(1) ~= 0
  p = a(2)/a(1)
  q = a(3)/a(1)
  D = (p^2)/4 - q
  if D < 0
    disp('D < 0: No real zero')
  elseif D == 0
    disp('D = 0: One real zero')
```

```
    x1 = -p/2
    x2 = x1
  else
    disp('D > 0: Two real zeros')
    x1 = -p/2 + sqrt(D)
    x2 = -p/2 - sqrt(D)
  end
elseif (a(1) == 0) & (a(2) ~= 0)
  disp('The unique zero is')
  x1 = -a(3)/a(2)
  x2 = x1
else
  disp('The polynom is constant and does not have a zero!')
end
```

d) (i) $x_1 = 1, \quad x_2 = -3$

(ii) $x_1 = -2 = x_2$

(iii) $x_1 = -1 = x_2$

(iv) no real zero

M6

```
function c = vectorprod(a,b)
c(1) = a(2)*b(3) - a(3)*b(2);
c(2) = a(3)*b(1) - a(1)*b(3);
c(3) = a(1)*b(2) - a(2)*b(1);
```

The input

```
a = [4;1;0];
b = [-2;-2;0];
c = vektorprod(a,b)
```

yields to the result $c = \begin{pmatrix} 0 & 0 & -6 \end{pmatrix}$

M7

```
B = [11 2; 3 5];
u = [4;6;8];
v = [8;12;16];
A = zeros(3);
A(2:3, 1:2) = B;
A(1,:) = u;
A(:,3) = v;
```

M8

a)
```
format long;
x0 = 0;
f = inline('x^2+2*x-3');
df = inline('2*x+2');
for i=1:3
  x0 = x0 - f(x0)/df(x0)
```

```
end
```

yields to the iterates (here rounded to 6 decimal places)

$$x_1 = 1.500000, \qquad x_2 = 1.050000, \qquad x_3 = 1.000610$$

b) - d)

```
function x = newton(f,df,x0,TOL,NMAX)
%newton Newton method for calculating a root of the function f starting from
the inital value x0. newton() iterates until |x_(n+1)-x_n| < TOL or n > NMAX.
The program can be started with 3, 4 or 5 input arguments.
tic;
tstart = tic;
if nargin < 3
  disp('not enough input parameters!')
elseif nargin == 3
  TOL = 10^(-6)
  NMAX = 20
elseif nargin == 4
  NMAX = 20
end
xn = x0+100;
xn1 = x0;
n = 0;
while (abs(xn1-xn) >= TOL) & (n <= NMAX),
  xn = xn1;
  xn1 = xn - f(xn)/df(xn);
  n = n+1; end
x = xn;
t = toc(tstart);
disp('used iterations:'), n-1
disp('Running time:'), t
```

e)

TOL	Iterations	Running time	Zero
10^{-3}	3	0.002636s	1.000609756097561
10^{-4}	4	0.003316s	1.000000092922295
10^{-5}	4	0.003267s	1.000000092922295
10^{-6}	4	0.002220s	1.000000092922295
10^{-7}	4	0.003253s	1.000000092922295
10^{-8}	5	0.003897s	1.000000000000002
10^{-9}	5	0.003898s	1.000000000000002

The runnings times are different in every run and here they are just supposed to show the dimension.

f) Simple Example:

```
xaxis = x-5:x+5;
yaxis = xaxis.^2 + 2*xaxis - 3;
plot(xaxis,yaxis,'b-',x,f(x),'rx')
```

plots an approximation of the function as a blue line with a red cross at the calculated zero

Aufgabe M9

a) `a+c` $= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

b) `c'*c` $= \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

c) `E(2,:)+c` $= \begin{pmatrix} 1 & 2 & 1 & 1 & 1 \end{pmatrix}$

d) `a.*c` $= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

e) `c*c'` $= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 5$

f) `E(:,3)+c` $\rightarrow$ Error, columnvector+rowvector

g) `a'*c` $= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

h) `E*c` $\rightarrow$ Error, matrix*rowvector

i) `D+F` $= \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$

j) `a'.*c` $\rightarrow$ Error, component-by-component multiplication in a product of a columnvector with a rowvector

k) `c*E` $= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

l) `D.*F` $= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ (component-by-component matrixmultiplication)

m) `D*F` $= \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

n) `B.*D` $= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

o) `B+D(1,1)` $= \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$