

Kurzlösungen lineare Algebra 01

Falls eine Aufgabe von einer Mehrheit nicht verstanden wird, so soll ie Aufgabe in der letzten Analysis-Lektion der Woche thematisiert werden.

Falls eine Aufgabe von einer Meinderheit nicht verstanden wird, so soll man dafür möglichst rasch für eine Sprechstunde anmelden.

Falls eine Aufgabe sowieso klar ist, bedeutet ein weiterer Zeitaufwand nur einen Zeitverlust. Falls niemand Verständnisprobleme vorbringt, so kann angenommen werden, dass die Aufgabe erledigt ist.

Uebungen 1 - 4

Organisatorisches, hier kein Resultat notwendig.

Uebung 5

a, b, c

Persönlich, Abgabe

Uebung 6 und 7

Siehe Wikipedia oder Literatur

MATLAB-Teil

Uebung 8

Sebststudium Skript

Siehe auch S 01 Selbststud. lin. Alg + Geom.

==> <http://rowicus.ch/Wir/ProblemsSolutBachelor/ProblemsSolutBachelor.html>

Uebung 9

```
>> sin(360)
ans = 0.95892
>> sin(pi)
ans = 1.2246e-16    (numerischer Rundungsrest, praktisch 0)
>> sin(pi/2)
ans = 1
```

Bogenmass

Uebung 10

15 Bit für Zahl und Bit für Vorzeichen, 0 nur einmal bei positiven Zahlen mitgezählt! Anzahl Zahlen von 1 bis 2^{15} :

$$2^{15}$$

Natürliche Zahlen von -32768 bis 32767
(Speicherbar in 2 Bytes zu je 8 Bits, stammt aus den 70-er Jahren)

Uebung 11

In Octave bisher keine kurze Befehlsvariante aufgefunden zur Lösung des Problems!

```
(* Befehl in Mathematica, 1.4 im 2-er-System *)
BaseForm[1.4,2]
```

```
1.01100110011001100112
```

```
(* Befehl in Mathematica, 1.4 im 3-er-System *)
BaseForm[1.4,3]
```

```
1.1012101210123
```

```
(* Befehl in Mathematica, 1.4 im 7-er-System *)
BaseForm[1.4,7]
```

```
1.2541267
```

```
(* Befehl in Mathematica, 1.4 im 16-er-System *)
BaseForm[1.4,16]
```

```
1.666616
```

Vorgehen in Octave: $1.4 = 1.0 + 0.4$, 0.4 entwickeln:

```
>> a=1/2^1
a = 0.50000
>> b=1/2^2
b = 0.25000
>> c=1/2^3
c = 0.12500
>> d=1/2^4
d = 0.062500
>> e=1/2^5
e = 0.031250
>> f=1/2^6
f = 0.015625
u.s.w.
```

```
>> 1*a
ans = 0.50000   Zu gross!
>> 0*a+1*b
ans = 0.25000   Zu klein!
>> 0*a+1*b+1*c
ans = 0.37500   Zu klein!
>> 0*a+1*b+1*c+1*d
ans = 0.43750   Zu gross!
>> 0*a+1*b+1*c+0*d
ans = 0.37500   Zu klein!
>> 0*a+1*b+1*c+0*d+1*e
ans = 0.40625   Zu gross!
>> 0*a+1*b+1*c+0*d+0*e
ans = 0.37500   Zu klein!
>> 0*a+1*b+1*c+0*d+0*e+1*f
ans = 0.39062   Zu klein!
>> u.s.w.
```

```
1.4 = 1.0+0*a+1*b+1*c+0*d+0*e+1*f+...
= 1*2^0+0*2^(-1)+1*2^(-2)+1*2^(-3)+0*2^(-4)+0*2^(-5)+1*2^(-6)+...
= binär 1.011001...
```

(Exakt von oben: 1.0110011001100110011_2)

Uebung 12

a, b

Erst ausprobieren, sonst ist der Spass schon vorher vorbei!

```
>> 1/3
ans = 0.33333
>> pi
pi = 3.1416
```

c

```
(* Befehl in Mathematica, 1001110.011_2 im 10-er-System *)
```

```
2^^1001110.011
```

```
(* "Von Hand" in Mathematica, 1001110.011_2 im 10-er-System *)
```

```
2^6+2^3+2^2+2^1+2^(-2)+2^(-3)
```

```
2^6+2^3+2^2+2^1+2^(-2)+2^(-3) // N
```

Octave:

```
>> 2^6+2^3+2^1+2^(-2)+2^(-3)
ans = 74.375
```

Uebung 13

```
>> sqrt(2)
ans = 1.4142
```

Uebung 14

Erst ausprobieren, sonst ist der Spass schon vorher vorbei!

a

```
>> realmax
realmax = 1.7977e+308
>> 10^308
ans = 1.0000e+308
```

b

```
>> realmax+100000
ans = 1.7977e+308    Nichts passiert!
```

c

```
>> realmax-(realmax-1)
ans = 0
>> realmax*1/(realmax-1)
ans = 1
>> (realmax/5)*5
ans = 1.7977e+308
>> 5*(realmax/5)
ans = 1.7977e+308
>> (realmax*5)/5
ans = Inf
```

==> Beachte die Umstände, in denen Fehler entstehen!!!!

**d Aufgabe für sehr weit Fortgeschrittene!
(Einsteiger sollten versuchen, die MATLAB-Lösung
nachzuvollziehen!)**

Vorübung 1:

```
>> sum(1:5)    Summe der natürlichen Zahlen von 1 bis 5
ans = 15
>> 1+2+3+4+5  Summe der natürlichen Zahlen von 1 bis 5
ans = 15
```

Vorübung 2: Summe der natürlichen Zahlen von 1 bis 5

```
>> a=1:5      Natürliche Zahlen von 1 bis 5 als Elemente eines "Vektors"
a =
```

1 2 3 4 5

```
>> b=5:-1:1   Natürliche Zahlen von 5 bis 1 als Elemente eines "Vektors"
b =
```

5 4 3 2 1

```
>> c=[a,6,b]  Vektoren zusammenfügen, dazwischen das Element 6
c =
```

Columns 1 through 8:

1 2 3 4 5 6 5 4

Columns 9 through 11:

3 2 1

```
>> -c        Negativer Vektor
ans =
```

Columns 1 through 7:

-1 -2 -3 -4 -5 -6 -5

Columns 8 through 11:

-4 -3 -2 -1

```
>> d=2.^(-c)  Vektor der Potenzen "2 hoch Element von d"
d =
```

Columns 1 through 8:

```
0.50000 0.25000 0.12500 0.06250 0.03125 0.01562 0.03125 0.06250
```

Columns 9 through 11:

```
0.12500 0.25000 0.50000
```

```
>> sum(d)           Elemente von d aufsummieren  
ans = 1.9531
```

Lösung der Aufgabe (Output der Zwischenschritte unterdrückt):

```
>> a=1:10000;  
>> b=a.^(-2);  
>> c=sum(b)  
c = 1.6448  
>>  
>> d=10000:-1:1;  
>> e=2.^(-2);  
>> f=sum(e)  
f = 0.25000  
>> e-f  
ans = 0  
>> f-e  
ans = 0  
  
(* Vergleich: Befehl in Mathematica *)  
c=Sum[n^(-2),{n,1,10000}] // N  
  
1.64483  
  
(* Vergleich: Befehl in Mathematica *)  
d=Sum[n^(-2),{n,10000,1,-1}] // N  
  
1.64483  
  
(* Vergleich: Befehl in Mathematica *)  
  
c-d  
  
0.
```

Uebung 15

Erst ausprobieren, sonst ist der Spass schon vorher vorbei!

```
>> a=sqrt(987654)  
a = 993.81  
>> b=sqrt(987653)  
b = 993.81
```

```
>> c=a-b
c = 0.00050312
>> d=(a^2-b^2)/(a+b)
d = 0.00050312
>> c-d
ans = -5.6156e-14
>> d-c
ans = 5.6156e-14  Differenz ergibt einen numerischen Rundungsrest!
```