Musterlösung Serie 1

```
1. function [epsM,it]= s1_1(eps0)
    y = eps0;
    while ((1+y)~=1)
        y = y/2;
    end
    % Jetzt liegt epsM im Intervall [y,2*y] (denn 1+y=1, aber 1+2*y>1).
    % Um epsM genauer zu bestimmen halbieren wir sukzessiv das Intervall,
    % indem wir fuer den jeweiligen Mittelpunkt z untersuchen,
    % ob 1+z==1 gilt oder nicht.
    x(2) = 2*y;
    x(1) = y;
    z=mean(x);
    c=1e-15;
    it=0;
    while (1+(1-c)*z==1+(1+c)*z) && (it<2000)
        it=it+1;
        if (1+z==1)
            x(1)=z;
            z=(z+x(2))/2;
        else
            x(2)=z;
            z=(z+x(1))/2;
        end
    end
    epsM=z;
```

Resultat mit Startwert 1.0: $\text{eps}_M = 1.110223024625158e-16$. Der MATLAB-Befehl `eps` liefert $\text{eps} = 2.220446049250313e-16$, die Definition von `eps` ist aber 'die Distanz zwischen 1.0 und der nächsten grösseren Zahl (double precision)'. Es gilt: $1+\text{eps}/2=1$, aber $1+1.0000000000000001*\text{eps}/2 \neq 1$. Laut dem Kriterium $\rho(1+\alpha) = 1, \forall \alpha \in \mathbb{R}, 0 < \alpha < \text{eps}_M$, muss also eps_M nahe bei $\text{eps}/2 = 1.110223024625157e-16$ sein. Bemerkung: $\text{eps}/2 = \frac{1}{2}B^{1-\ell} = 2^{-53}$.

```
2. function[sum_a,sum_b]= s1_2()
    format long
    sum_a=100;
    sum_b=0;
```

```

signum=-1;
for i=2:10000
    sum_a=sum_a+signum*1/i;
    sum_b=sum_b+signum*1/(10002-i);
    signum = -signum;
end
sum_b=sum_b+100;
end

```

a) 99.693097183060189

b) 99.693097183059947

Die Mantissenlänge ist auf fast 16 Stellen beschränkt. Im Fall a) können die kleinen Summanden nicht in voller Genauigkeit berücksichtigt werden.

3. Direkte Berechnungen mit MATLAB:

a) $x_M = -3.468647899005362e - 04$

b) $x_M = 9.999500033330833e - 05$

c) $x_M = -5.000000413701855e - 11$

Umformen der Terme:

a) $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$, $a = \sqrt[3]{29790}$, $b = 31$
 $\Rightarrow x_U = \frac{a^3 - b^3}{a^2 + ab + b^2} = -3.468647898960239e - 04$

b) $\ln(a) - \ln(b) = \ln\left(\frac{a}{b}\right)$
 $\Rightarrow x_U = \ln(1.0001) = 9.999500033329732e - 05$

c) $\cos(a) - \cos(b) = -2 \sin\left(\frac{a+b}{2}\right) \sin\left(\frac{a-b}{2}\right)$, $a = 0.00001$, $b = 0$
 $\Rightarrow x_U = -2 \sin(0.000005)^2 = -4.99999999958334e - 11$

‘Relativer’ Fehler, d.h. $|x_M - x_U|/|x_U|$:

a) 1.300877731899914e-11

b) 1.100520227416153e-13

c) 8.274870416178436e-08

4. a) Die maximale Konditionszahl von $f(x) = 1 - e^x$ für $x \in (0, 1)$ ist kleiner als

$$M := \max_{x \in (0,1]} \left| \frac{f'(x)x}{f(x)} \right| = \max_{x \in (0,1]} \left| \frac{-xe^x}{1 - e^x} \right|.$$

Siehe nächstes Blatt!

Wenn man $g(x) := (f'(x)x)/f(x)$ für x von 0 nach 1 plottet, sieht man dass $g(x)$ monoton wachsend ist und damit weiss man, dass das Maximum bei $x = 1$ ist (man kann es auch analytisch zeigen, indem man zeigt, dass $g'(x) > 0 \forall x \in (0, 1)$ (g monoton wachsend) sowie $\lim_{x \rightarrow 0} g(x) = 1$ (keine Singularität) gelten). Somit folgt, dass

$$M = g(1) = \frac{-e}{1-e} \approx 1.58198.$$

Der Fehler wird also maximal um 1.58198 verstärkt. Damit der relative Fehler von y bestimmt nicht grösser als 1% ist, darf der relative Fehler in x maximal ungefähr 0.00632. betragen. Die Genauigkeit von 0.1% ist also ausreichend.

b) Mit

```
function [Kond_eig, Kond_cond]=s1_4b()
format long
H=hilb(10);
d=eig(H);
Kond_eig=max(abs(d))/min(abs(d));
Kond_cond=cond(H);
```

erhalten wir

```
Kond_eig=1.602834846325956e+13,
Kond_cond=1.602539175007862e+13.
```

Bemerkung: cond benützt die Singulärwertzerlegung von H um die Konditionszahl von H zu berechnen.

```
5. A=[ 0.03482 0.07355 -0.1679
      0.07355 0.1554 -0.3548
      -0.1679 -0.3548 0.8098];
b=[ -1.8656;-3.9423; 8.9988 ] ;
db=[ 1.1172;0.68494 ;0.53171 ] *1e-4
x=A\b;
dx=A\db;
rel_err_b = norm(db)/norm(b)
rel_err_x =norm(dx)/norm(x)
rel_err_x/rel_err_b
cond(A)
```

Output

```
rel_err_b =1.414207299516318e-05
rel_err_x = 0.67961055929385
rel_err_x/rel_err_b = 4.805593632038886e+04
cond(A) = 9.194880161778109e+04
```