Musterlösung Serie 2

1. **a)** Exakt:

$$\boxed{\begin{array}{rr} 0.005 & 1 \\ 1 & 1 \end{array}} \rightarrow \boxed{\begin{array}{rr} 0.005 & 1 \\ 200 & -199 \end{array}} \Rightarrow L = \begin{pmatrix} 1 & 0 \\ 200 & 1 \end{pmatrix}, R = \begin{pmatrix} 0.005 & 1 \\ 0 & -199 \end{pmatrix}$$

Rückwärts einsetzen

- $Lz = y: \ z_1 = 0.5, \ z_2 = 1 - 100 = -99$
- $Rx = z : x_2 = \frac{99}{199}, x_1 = \left(0.5 - \frac{99}{199}\right) 200 = \frac{100}{199}$

b) Mit Runden

$$\boxed{\begin{array}{rr} 0.005 & 1 \\ 1 & 1 \end{array}} \rightarrow \boxed{\begin{array}{rr} 0.005 & 1 \\ 200 & -199 \end{array}} \Rightarrow L = \begin{pmatrix} 1 & 0 \\ 200 & 1 \end{pmatrix}, R = \begin{pmatrix} 0.005 & 1 \\ 0 & -200 \end{pmatrix}$$

Rückwärts einsetzen

- $Lz = y: \ z_1 = 0.5, \ z_2 = 1 - 100 = -99$
- $Rx = z : x_2 = \frac{-99}{-200} \approx 0.50, x_1 = (0.5 - 0.5)\, 200 = 0$ (Auslöschung!)

c) Mit Runden und vertauschen der Zeilen (relative Spaltenmaximumstrategie)

$$\boxed{\begin{array}{rr} 1 & 1 \\ 0.005 & 1 \end{array}} \rightarrow \boxed{\begin{array}{rr} 1 & 1 \\ 0.005 & 1 \end{array}} \Rightarrow L = \begin{pmatrix} 1 & 0 \\ 0.005 & 1 \end{pmatrix}, R = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Rückwärts einsetzen

- $Lz = y: \ z_1 = 1, \ z_2 = 0.5 - 0.005 \approx 0.5$
- $Rx = z : x_2 = 0.5, x_1 = 1 - 0.5 = 0.5$

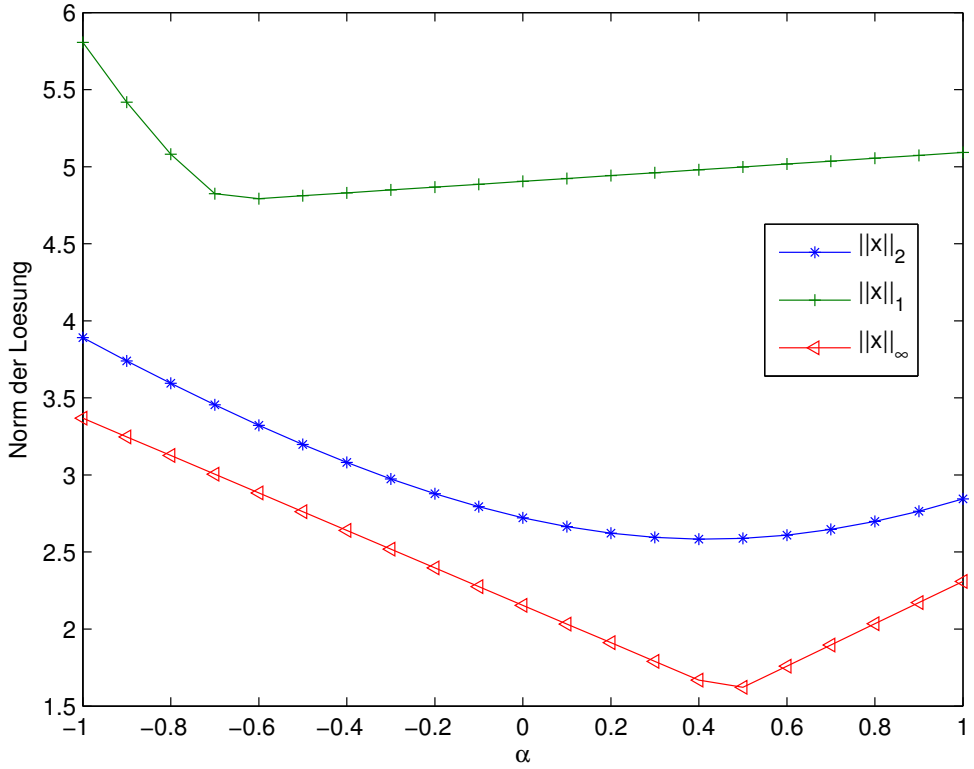
2.

```
clear all;
close all;
rand('state',0);
A=rand(4);
b=[1;1;1;0];
alpha=[-1:0.1:1];
[L,R,P]=lu(A);

for i=1:length(alpha)
 b(end)=alpha(i);
 x=R\(L\P*b);
 norm2_x(i)=norm(x);
 norm1_x(i)=norm(x,1);
```

Bitte wenden!

```
 norminf_x(i)=norm(x,inf);
end
plot(alpha,norm2_x,'-*',alpha,norm1_x,'-+',...
     alpha,norminf_x,'-<','MarkerSize',5)
xlabel('\alpha')
ylabel('Norm der Loesung')
legend('||x||_2','||x||_1','||x||_{\infty}')
```



3. Entscheidend für die Konvergenz eines Iterationsverfahrens $x^{k+1} = Tx^k + c$ ist der Spektralradius der Matrix T:

Es muss gelten: $\rho(T) < 1$.

Beim Jacobiverfahren ist $T = D^{-1}(-R - L)$, wobei $A = D + L + R$ (Skript S. 28)

Der Spektralradius einer Matrix kann z.B. wie folgt bestimmt werden.

```
D=diag(diag(M))
R=triu(M)-D;
L=tril(M)-D;
T=D\(-L-R);
max(abs(eig(T)))
```

Dies liefert für A: 0.7071, für B: 1.5811 und für C: 1.1241. Das Jacobiverfahren konvergiert also nur für die Matrix A.

Siehe nächstes Blatt!

4. Der folgende Code implementiert den im Skript auf Seite 24 beschriebenen Algorithmus

```
function [L,R,P] = lr_pivot(A)
if abs(det(A)) < 1e-10
    error('Die Matrix ist (fast) singulaer')
end
n=size(A,1);
L=eye(n); P=eye(n);
for k=1:n-1
  %Bestimmen der Zeilensskalierungsfaktoren q
  m=max(abs(A(k:n,k:n)),[],2);
  q=abs(A(k:n,k))./m;
  %waehle groesstes relatives Pivotelement (qmax)
  [qp,p_idx]=max(q);
  if(qp==0)
      error(['pivot element ist 0 in Zeile ',num2str(k)]);
  end
  p_idx=p_idx+k-1;
  if(p_idx~=k) % Vertausche Zeiten p_idx und k
   A([k,p_idx],:)=A([p_idx,k],:);
   P([k,p_idx],:)=P([p_idx,k],:);
   L([k,p_idx],1:k-1)=L([p_idx,k],1:k-1);
  end
  L(k+1:n,k)=A(k+1:n,k)./A(k,k);
  A(k+1:n,k:n)=A(k+1:n,k:n) - L(k+1:n,k)*A(k,k:n);
end
R=A;
```