

# MATLAB-Demo aus der Vorlesung "Inverse Probleme" (WS2012/13) von Andreas Rieder

## Inhalt

- Das lineare System
- Tikhonov-Regularisierung
- Landweber-Verfahren
- nu-Methode nach Brakhage, nu=1
- Verfahren der konjugierten Gradienten
- Plote exakte Lsg. und Rekonstruktion der Verfahren
- Zum Abschluss nun noch ein LEHRREICHES Beispiel,

## Das lineare System

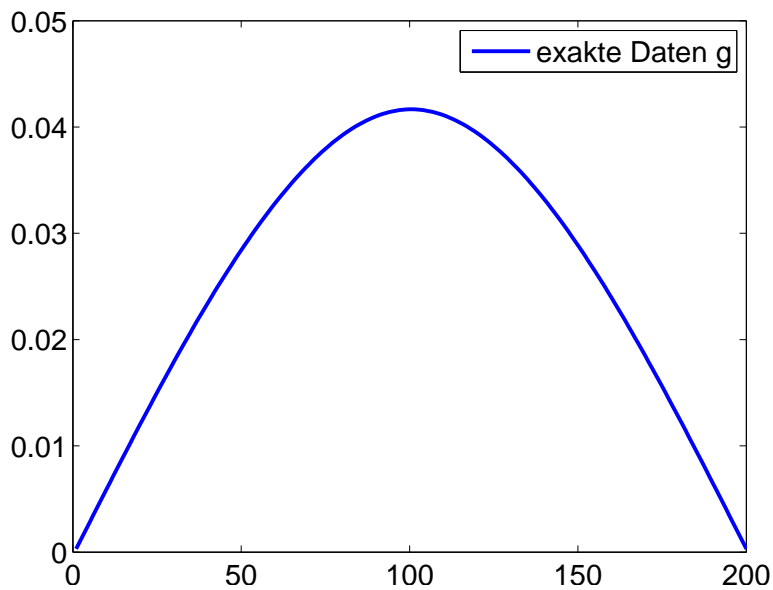
Zuerst wird das lineare System der Dimension 200x200 initialisiert.  
Es entsteht durch Diskretisierung einer maessig schlecht gestellten  
Integralgleichung 1. Art.

```
dim = 200;
[A,g,f]=deriv2(dim,3);
    echo off;
f=sqrt(dim)*f; g=-sqrt(dim)*g; A=-A;
echo on;

clf;
set(gca,'fontsize',15);

plot(f,'LineWidth',2); legend('exakte Lsg. f');

plot(g,'LineWidth',2); legend('exakte Daten g');
```

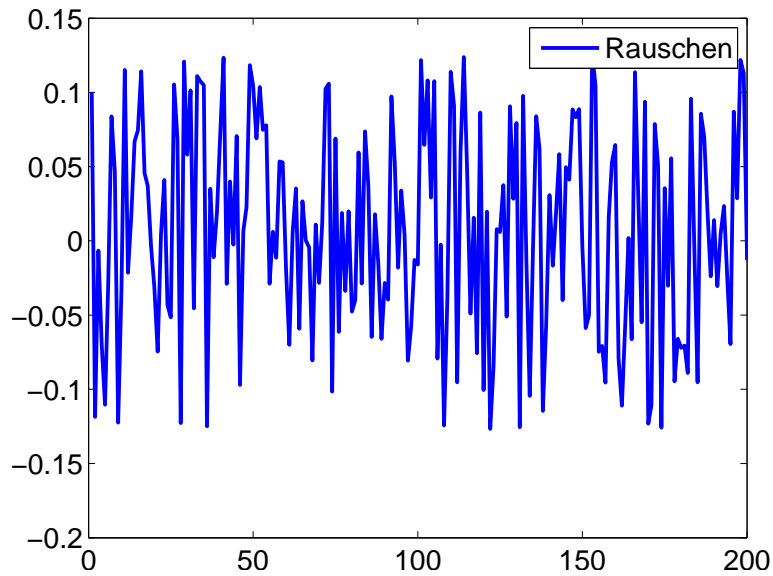


Erzeuge Rauschen

```

rand('seed',41997);
rausch = (2*rand(dim,1)-ones(dim,1));
rausch =rausch/norm(rausch);
plot(rausch,'LineWidth',2); legend('Rauschen');

```



Addiere 1% rel. Rauschen zu den Daten, d.h.  $\text{eps} = 0.01 * |g|_2$

```

eps = 0.01*norm(g);
g_eps = g + eps*rausch;

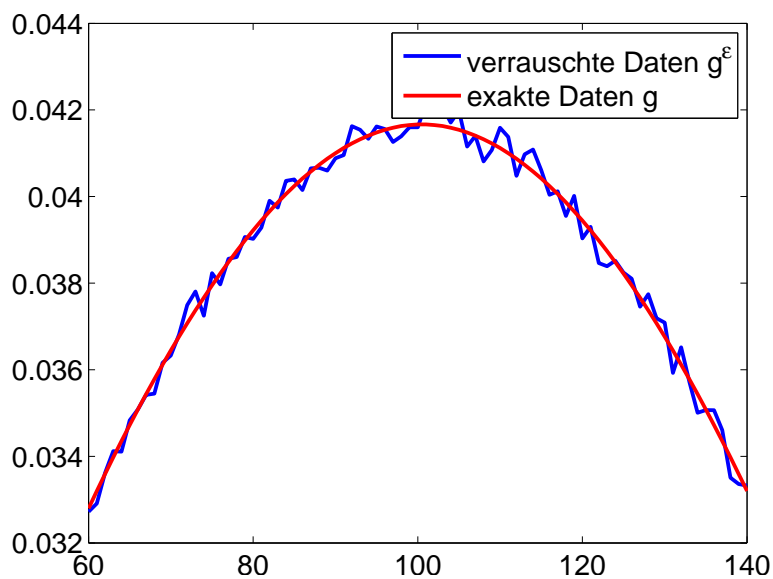
```

% Plotte Ausschnitt von g und g\_eps

```

plot(linspace(60,140,81),g_eps(60:140),'LineWidth',2); hold on;
plot(linspace(60,140,81),g(60:140),'Color','r','LineWidth',2);
legend('verrauschte Daten g^\epsilon','exakte Daten g');
hold off;

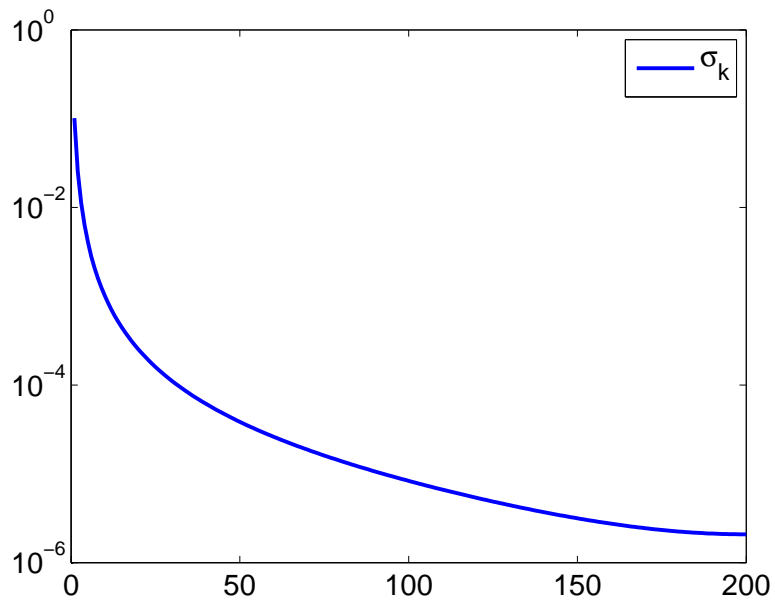
```



Berechne SWZ von A und plotte SW

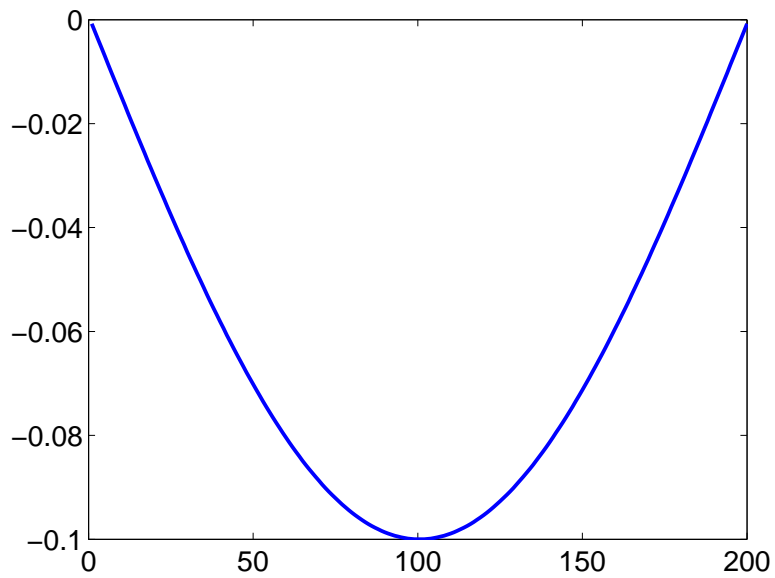
```
[U,s,V] = csvd(A);
```

```
semilogy(s,'LineWidth',2); legend('\sigma_k');
```

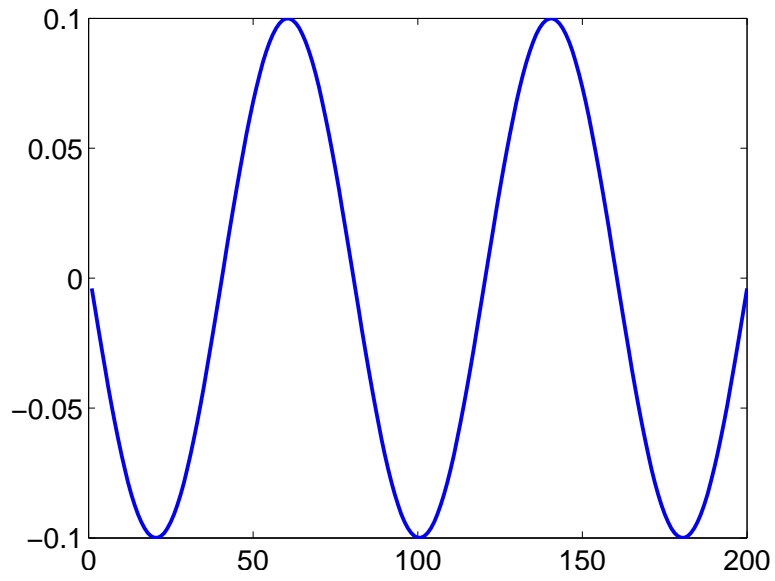


Plotte singuläre Vektoren nach fallenden SW

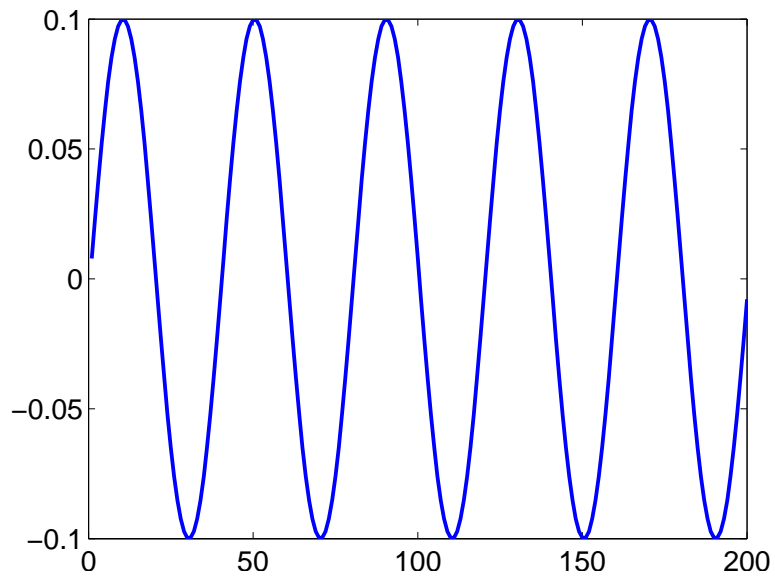
```
plot(U(:,1),'LineWidth',2);
```



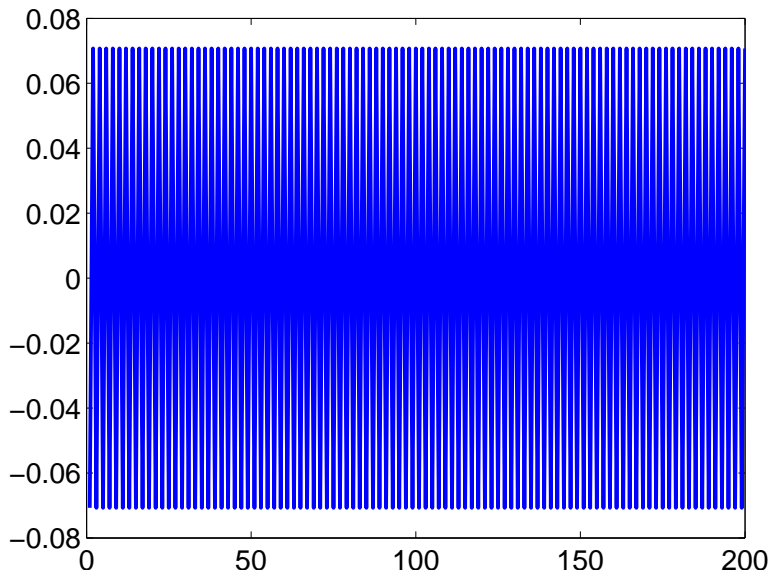
```
plot(U(:,5),'LineWidth',2);
```



```
plot(U(:,10),'LineWidth',2);
```

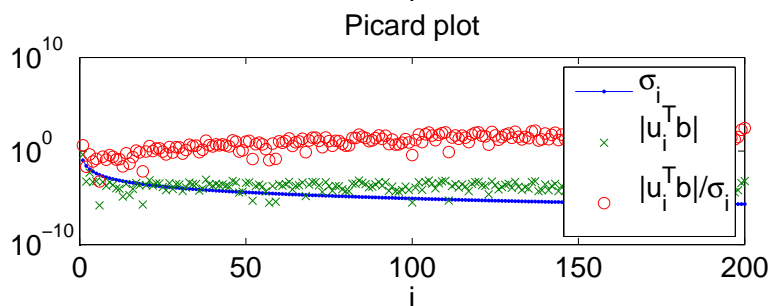
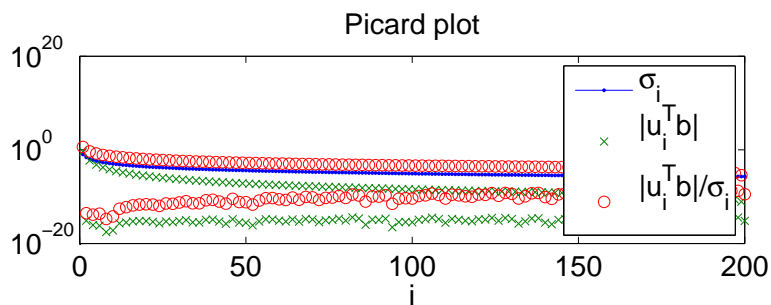


```
plot(U(:,dim),'LineWidth',2);
```



Ueberpruefe Picard-Bedingung

```
subplot(2,1,1); set(gca,'fontsize',15); picard(U,s,g);
subplot(2,1,2); set(gca,'fontsize',15); picard(U,s,g_eps);
```



## Tikhonov-Regularisierung

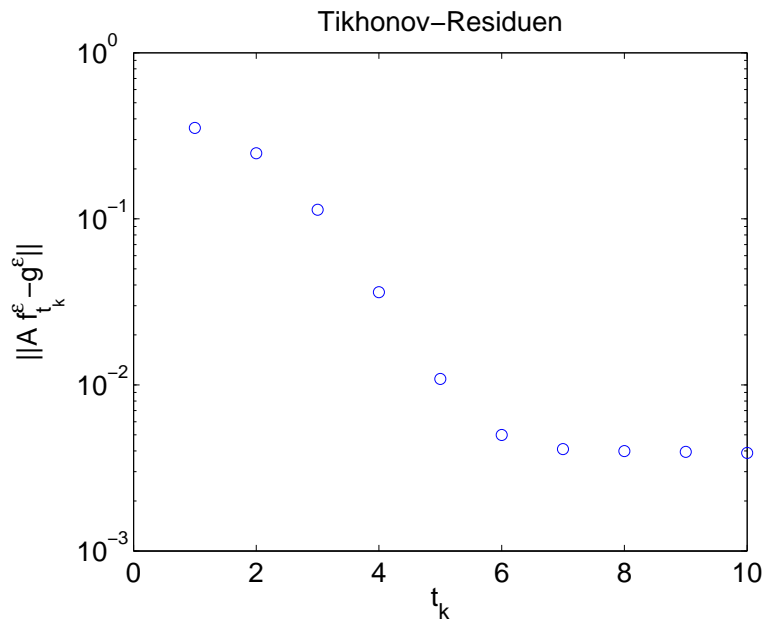
```
t=[1/4 1/8 1/16 1/32 1/64 1/128 1/256 1/512 1/1024 1/2048];
```

```
[f_t,res]=tikhonov(U,s,V,g_eps,t);
```

```
% Plotte Residuen zu t_i;
```

```
clf; set(gca,'fontsize',15);
semilogy(res,'o'); title('Tikhonov-Residuen'); xlabel('t_k');
```

```
ylabel('||A f_{t_k}^\epsilon - g^\epsilon||');
```

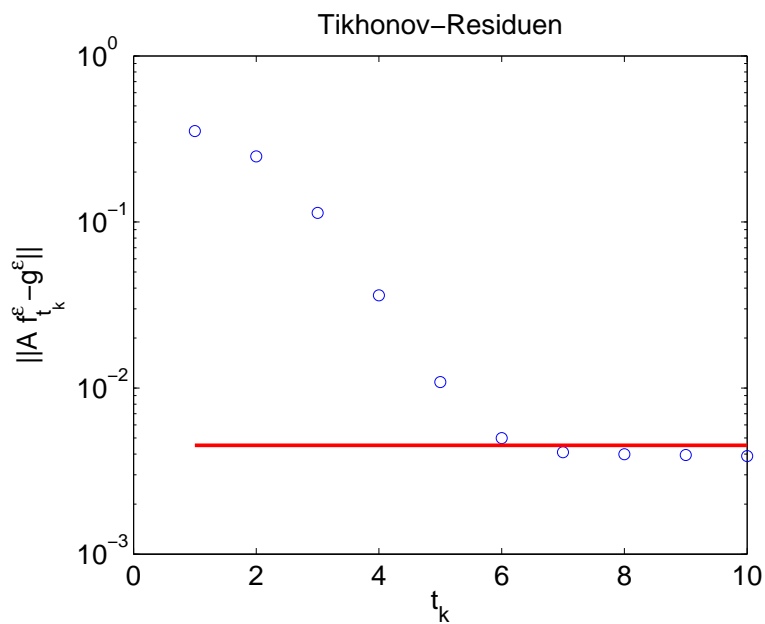


Plotte Abbruchschwelle  $\tau \cdot \epsilon$  mit  $\tau=1.1$

```
tau = 1.1;
```

```
E = tau*eps*ones(1,100);
```

```
hold on; semilogy(E(1:10),'Color','r','LineWidth',2); hold off;
```



Plotte ein paar Rekonstruktionen

```
plot(f,'LineWidth',2); hold on;
```

```
title('exakte Lsg. und Rekonstruktionen der Tikhonov-Reg.');
```

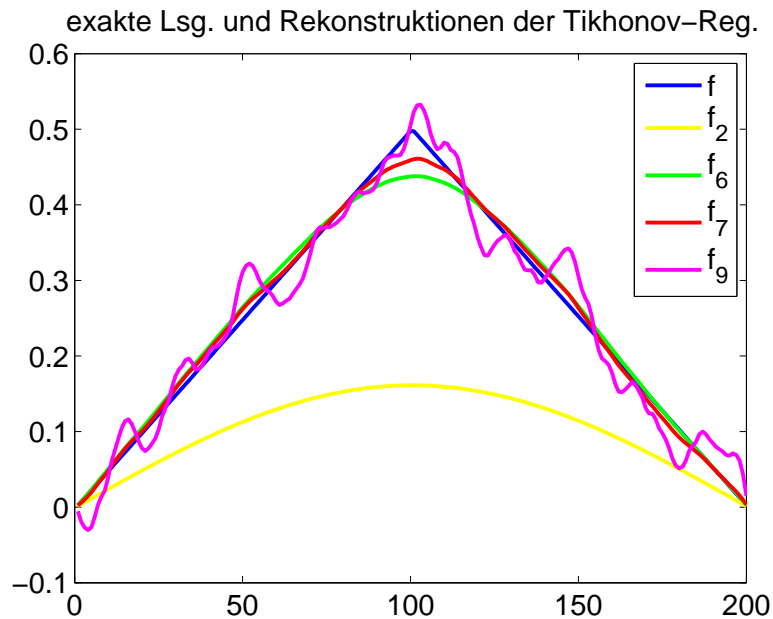
```
plot(f_t(:,2),'Color','y','LineWidth',2);
```

```
plot(f_t(:,6),'Color','g','LineWidth',2);
```

```

plot(f_t(:,7),'Color','r','LineWidth',2);
plot(f_t(:,9),'Color','m','LineWidth',2);
legend('f','f_2','f_6','f_7','f_9'); hold off;

```



Berechne Fehler der Rekonstruktionen

```

echo off;
for i=1:length(t)
    error(i)=norm(f-f_t(:,i));
end;
echo on;

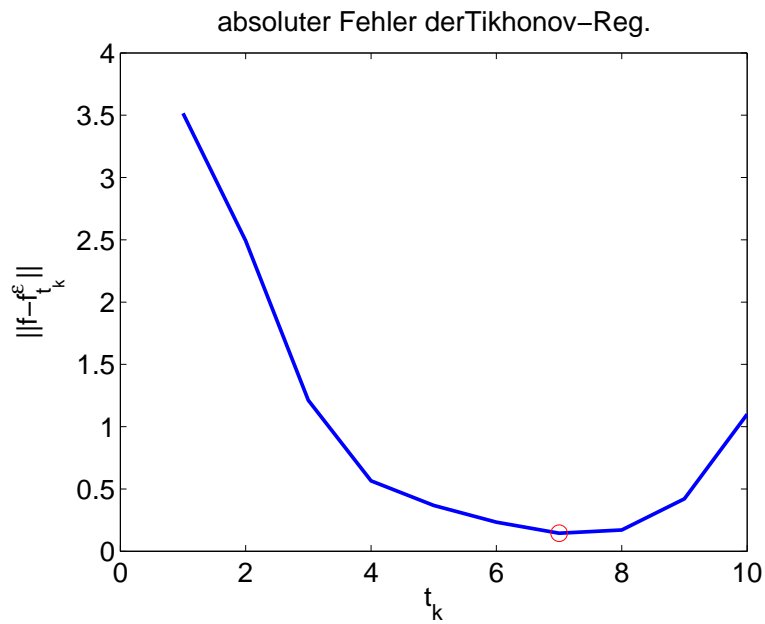
plot(error,'LineWidth',2);hold on;
plot(7,error(7),'o','Color','r','MarkerSize',9);
title('absoluter Fehler derTikhonov-Reg. '); xlabel('t_k');
ylabel('||f-f_{t_k}^\epsilon||'); hold off;

% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip

f_tik=f_t(:,7);
err_tik=error(:,7);

% Ende Tikhonov-Regularisierung

```



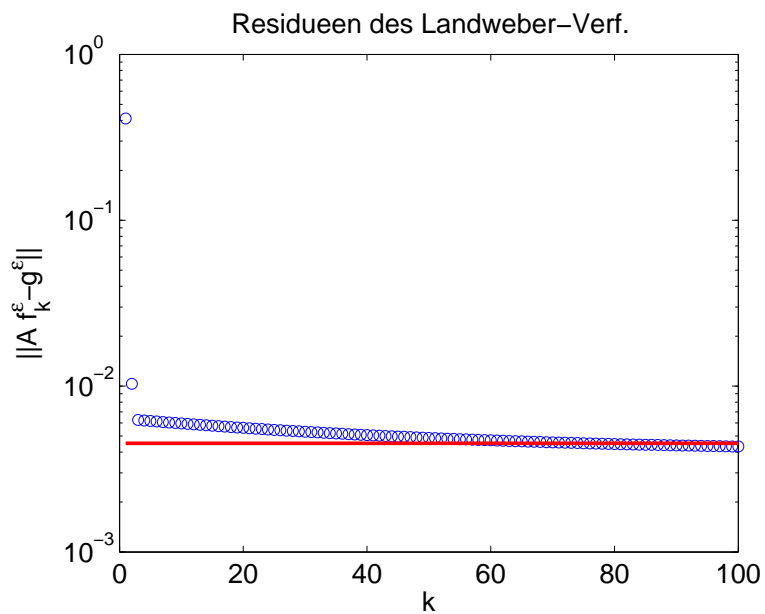
## Landweber-Verfahren

```
clear f_t res error;
```

```
[f_t,res]=landweber(A,g_eps,1500); % 1500 Schritte
pause
```

```
% Plotte Residuen des Landweber-Verfahrens
```

```
clf; set(gca,'fontsize',15);
semilogy(res(1:100),'o'); title('Residuen des Landweber-Verf. ');
xlabel('k'); ylabel('||A f_k^epsilon - g^epsilon||');
hold on; semilogy(E(1:100),'Color','r','LineWidth',2); hold off;
```



Berechne Fehler der Rekonstruktionen des Landweber-Verfahrens

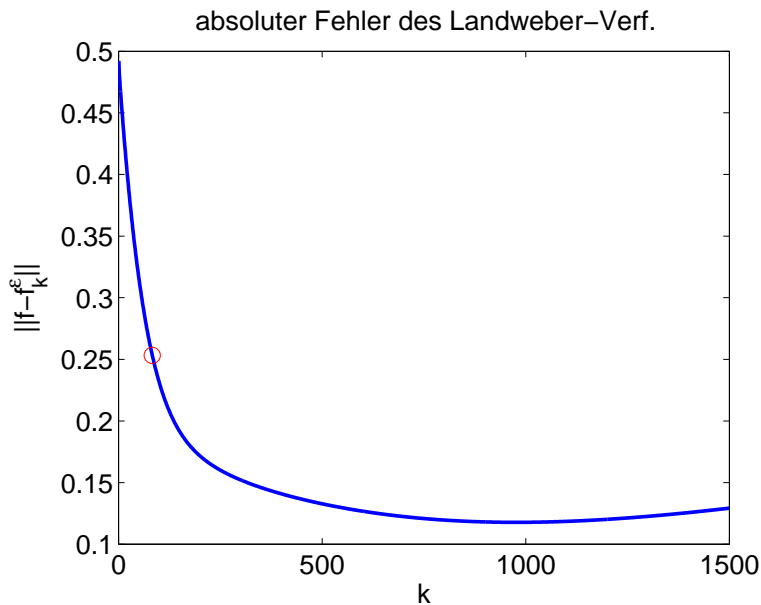


```

echo off;
for i=1:1500;
    error(i)=norm(f-f_t(:,i));
end;
clf; set(gca,'fontsize',15);
echo on;

plot(error,'LineWidth',2);hold on;
plot(83,error(83),'o','Color','r','MarkerSize',9);
title('absoluter Fehler des Landweber-Verf. '); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;

```



Plotte ein paar Rekonstruktionen des Landweber-Verfahrens

```

plot(f,'LineWidth',2); hold on;
title('exakte Lsg. und Rekonstruktionen des Landweber-Verf. ');
plot(f_t(:,10),'Color','y','LineWidth',2);
plot(f_t(:,60),'Color','g','LineWidth',2);
plot(f_t(:,83),'Color','r','LineWidth',2);
plot(f_t(:,1000),'Color','m','LineWidth',2);
legend('f','f_{10}','f_{60}','f_{83}','f_{1000}'); hold off;

```

% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip

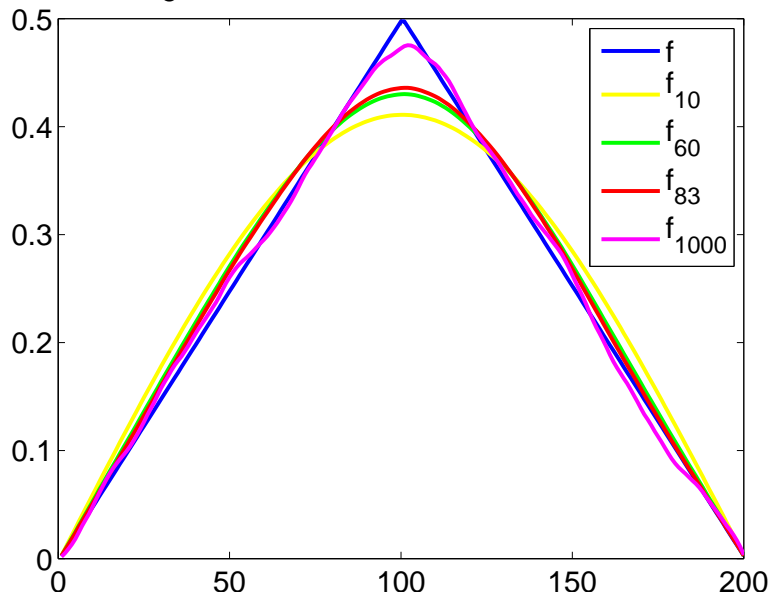
```

f_land=f_t(:,83);
err_land=error(:,83);

```

% Ende Landweber-Verfahren

exakte Lsg. und Rekonstruktionen des Landweber-Verf.



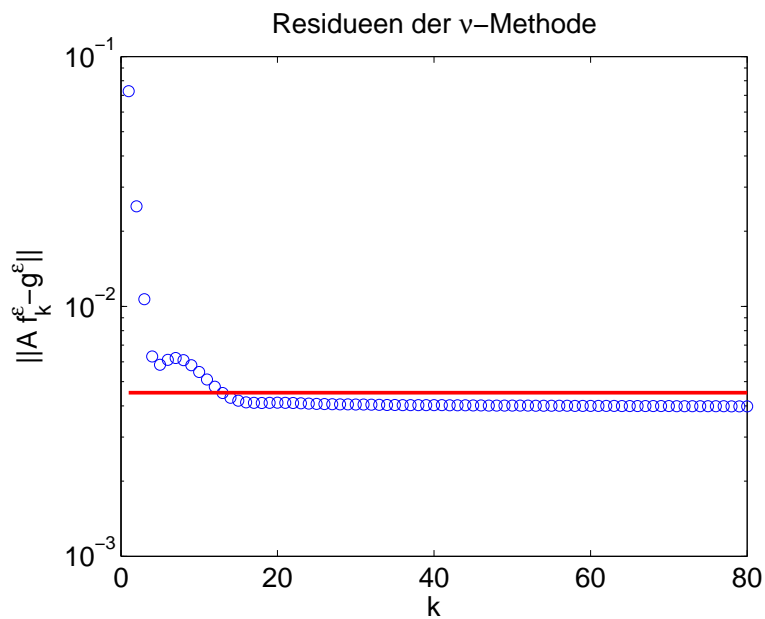
### nu-Methode nach Brakhage, nu=1

```
clear f_t res error;

[f_t,res]=nu(A,g_eps,80,1); % 80 Schritte
pause

% Plotte Residuen der Iterierten der nu-Methode ;

clf; set(gca,'fontsize',15);
semilogy(res,'o'); title('Residueen der \nu-Methode'); xlabel('k');
ylabel('||A f_k^\epsilon - g^\epsilon||');
hold on; semilogy(E(1:80),'Color','r','LineWidth',2); hold off;
```



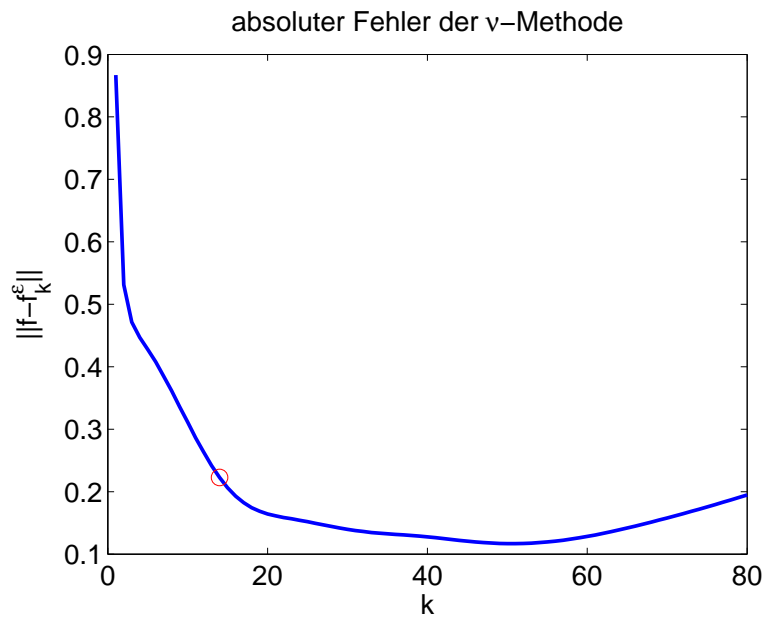
Berechne Fehler der Rekonstruktionen der nu-Methode

```

echo off;
for i=1:80
    error(i)=norm(f-f_t(:,i));
end;
clf; set(gca,'fontsize',15);
echo on;

plot(error,'LineWidth',2);hold on;
plot(14,error(14),'o','Color','r','MarkerSize',9);
title('absoluter Fehler der \nu-Methode'); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;

```



Plotte ein paar Rekonstruktionen der nu-Methode

```

plot(f,'LineWidth',2); hold on;
title('exakte Lsg. und Rekonstruktionen der \nu-Methode');
plot(f_t(:,4),'Color','y','LineWidth',2);
plot(f_t(:,12),'Color','g','LineWidth',2);
plot(f_t(:,14),'Color','r','LineWidth',2);
plot(f_t(:,80),'Color','m','LineWidth',2); axis tight;
legend('f','f_4','f_{12}','f_{14}','f_{80}'); hold off;

```

% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip

```

f_nu=f_t(:,14);
err_nu=error(:,14);

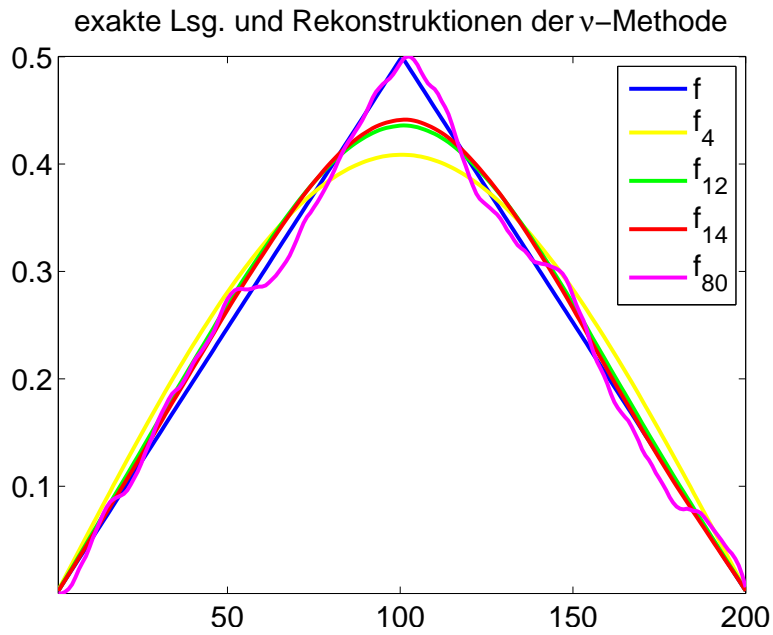
```

% Ende nu-Methode nach Brakhage

```

clear f_t res error;

```

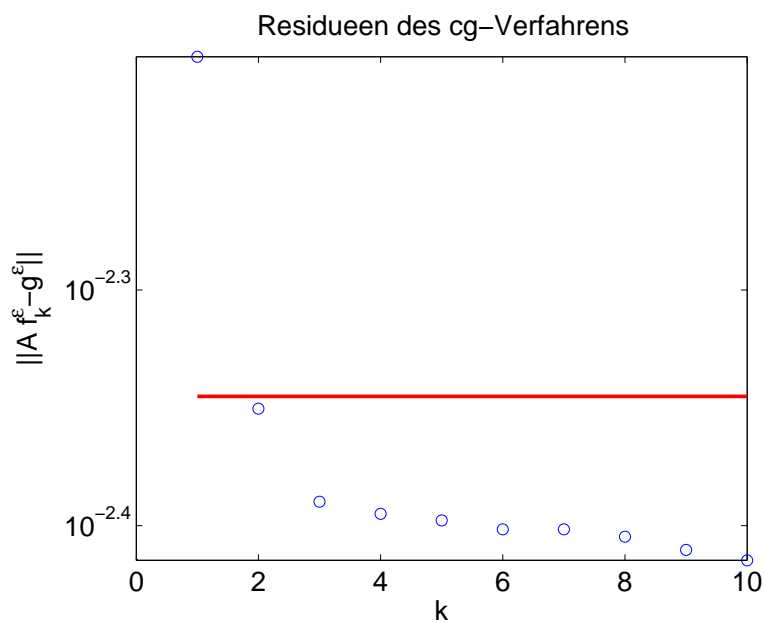


## Verfahren der konjugierten Gradienten

```
[f_t,res]=cgls(A,g_eps,10); % 10 Schritte
pause
```

```
% Plotte Residuen der cg-Iterierten;
```

```
clf; set(gca,'fontsize',15);
semilogy(res,'o'); title('Residuen des cg-Verfahrens'); xlabel('k');
ylabel('||A f_k^epsilon - g^epsilon||');
hold on; semilogy(E(1:10),'Color','r','LineWidth',2); hold off;
```



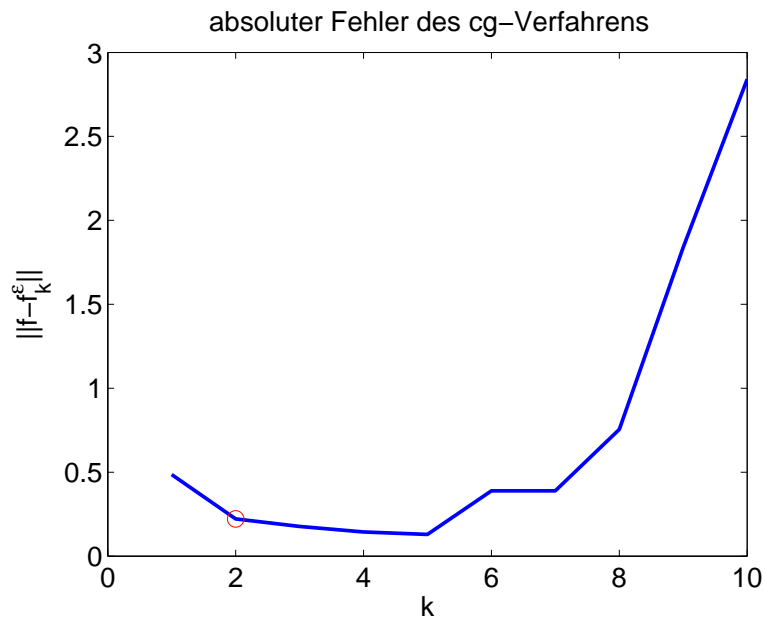
Berechne Fehler der cg-Rekonstruktionen

```

echo off;
for i=1:10
    error(i)=norm(f-f_t(:,i));
end;
clf; set(gca,'fontsize',15);
echo on;

plot(error,'LineWidth',2);hold on;
plot(2,error(2),'o','Color','r','MarkerSize',9);
title('absoluter Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;

```



Plotte ein paar cg-Rekonstruktionen

```

plot(f,'LineWidth',2); hold on;
title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
plot(f_t(:,1),'Color','y','LineWidth',2);
plot(f_t(:,2),'Color','r','LineWidth',2);
plot(f_t(:,4),'Color','g','LineWidth',2);
plot(f_t(:,8),'Color','m','LineWidth',2);
legend('f','f_1','f_{2}','f_4','f_{8}'); hold off;

```

% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip

```

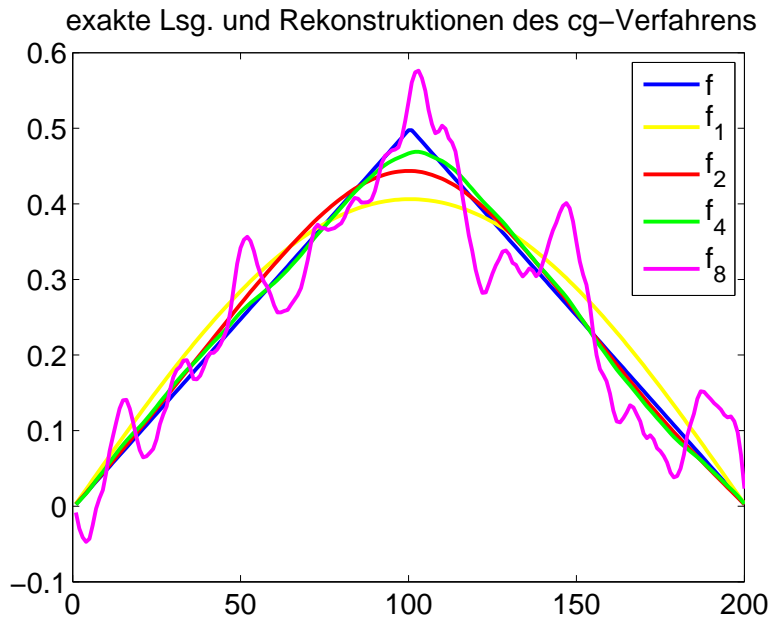
f_cg=f_t(:,2);
err_cg=error(:,2);

```

```

% Ende des cg-Verfahrens
pause

```



### Plote exakte Lsg. und Rekonstruktion der Verfahren

```

clf; set(gca,'fontsize',15);
plot(f,'LineWidth',2); hold on;
title('exakte Lsg. und Rekonstruktionen nach Diskrepanzprinzip');
plot(f_tik,'Color','r','LineWidth',2');
plot(f_land,'Color','y','LineWidth',2');
plot(f_nu,'Color','g','LineWidth',2');
plot(f_cg,'Color','m','LineWidth',2');
legend('f','f_{tik}','f_{land}','f_{nu}','f_{cg}'); hold off;

```

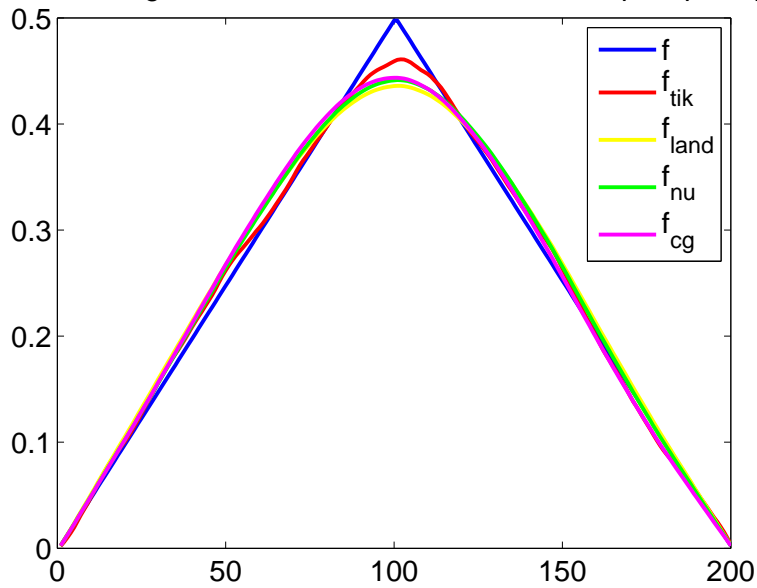
% Relativer Fehler der Verfahren

```
err=[err_tik err_land err_nu err_cg]./norm(f)
```

```
err =
```

```
0.0354    0.0620    0.0546    0.0543
```

exakte Lsg. und Rekonstruktionen nach Diskrepanzprinzip

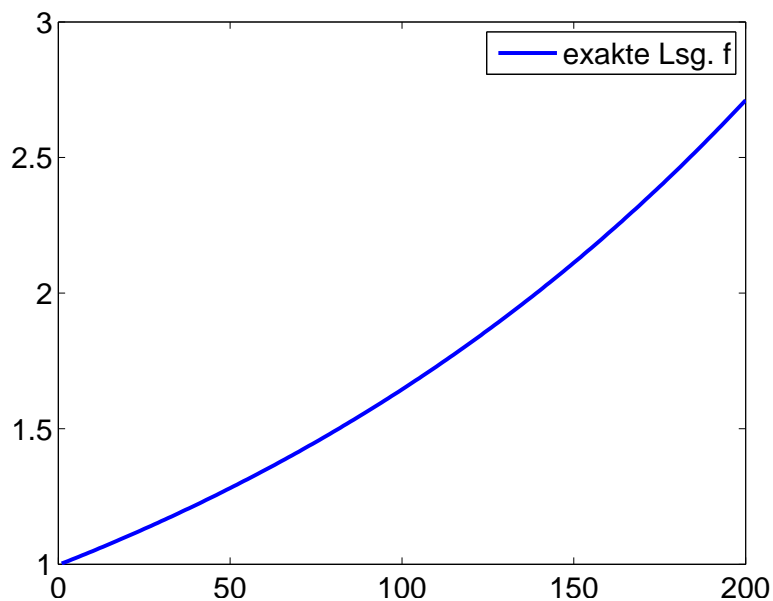


### Zum Abschluss nun noch ein LEHRREICHES Beispiel,

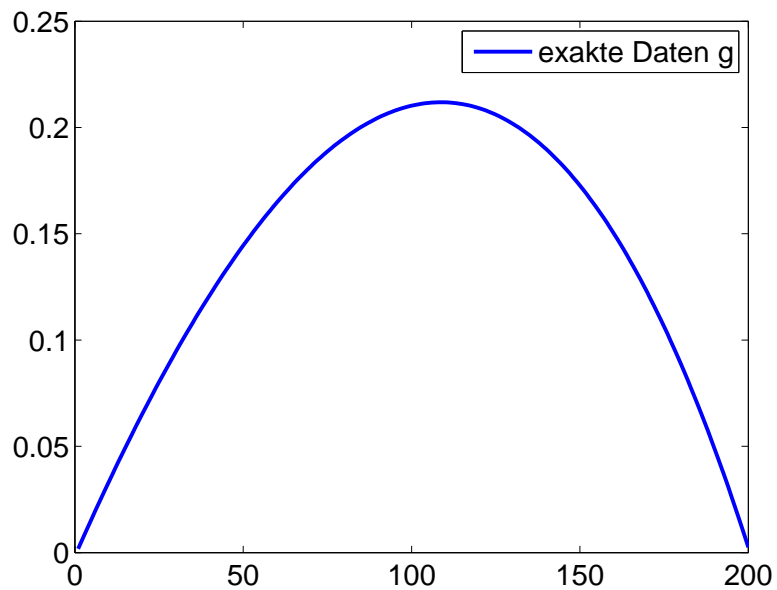
worin die obige Integralgleichung zugrunde liegt, allerdings mit einer anderen rechten Seite und einer anderen Loesung.

```
dim = 200;
[A,g,f]=deriv2(dim,2);
echo off;
f=sqrt(dim)*f; g=-sqrt(dim)*g; A=-A;
echo on;

clf;
set(gca,'fontsize',15);
plot(f,'LineWidth',2); legend('exakte Lsg. f');
```



```
plot(g,'LineWidth',2); legend('exakte Daten g');
```



Erzeuge Rauschen

```
rand('seed',41997);
rausch = (2*rand(dim,1)-ones(dim,1));
rausch =rausch/norm(rausch);

% Addiere 1% rel. Rauschen zu den Daten,
% d.h. eps =0.01*||g||_2

eps = 0.01*norm(g);
g_eps = g + eps*rausch;

% Plotte Ausschnitt von g und g_eps

plot(linspace(60,140,81),g_eps(60:140),'LineWidth',2); hold on;
plot(linspace(60,140,81),g(60:140),'Color','r','LineWidth',2);
legend('verrauschte Daten g^\epsilon','exakte Daten g')
hold off; pause

% Initialisiere Diskrepanzprinzip

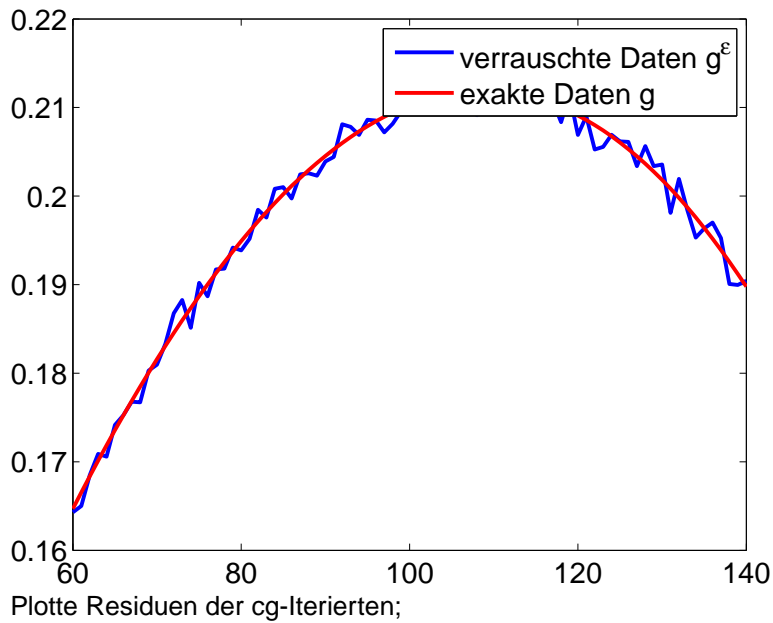
tau = 1.01;

E = tau*eps*ones(1,10);

% Verfahren der konjugierten Gradienten

[f_t,res]=cgls(A,g_eps,10); % 10 Schritte
pause
```

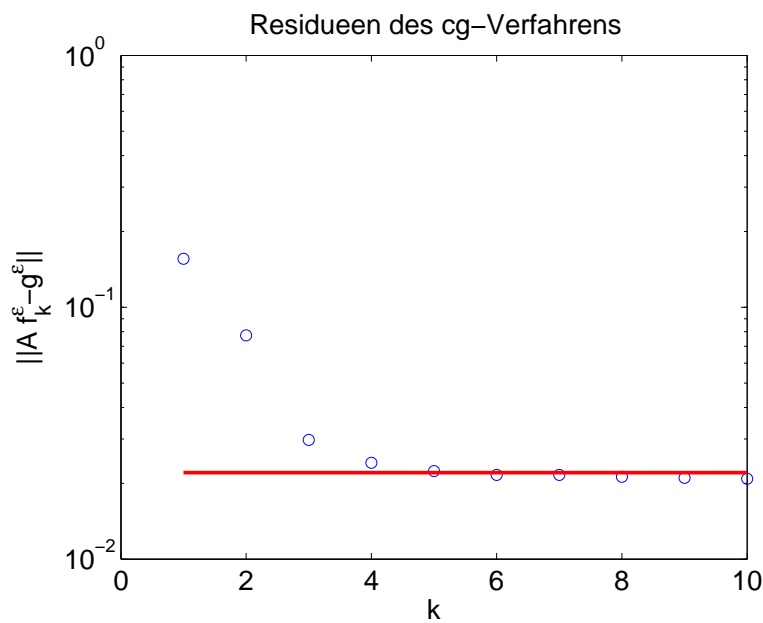




```

clf; set(gca,'fontsize',15);
semilogy(res,'o');
title('Residuen des cg-Verfahrens'); xlabel('k');
ylabel('||A f_k^epsilon - g^epsilon||');
hold on; semilogy(E(1:10),'Color','r','LineWidth',2);
hold off;

```



Berechne rel. Fehler der cg-Rekonstruktionen

```

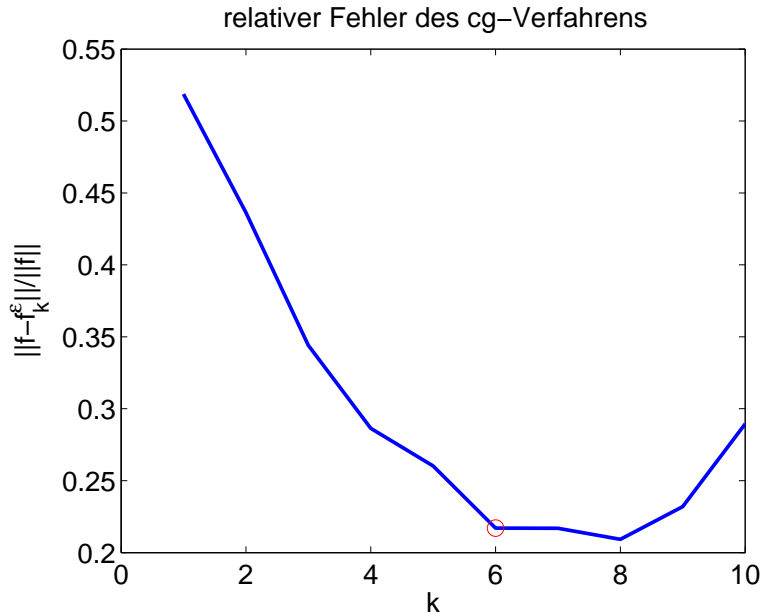
echo off;
for i=1:10
    error(i)=norm(f-f_t(:,i))/norm(f);
end;
clf; set(gca,'fontsize',15);
echo on;

```

```

plot(error,'LineWidth',2);hold on;
plot(6,error(6),'o','Color','r','MarkerSize',9);
title('relativer Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||/||f||'); hold off;

```

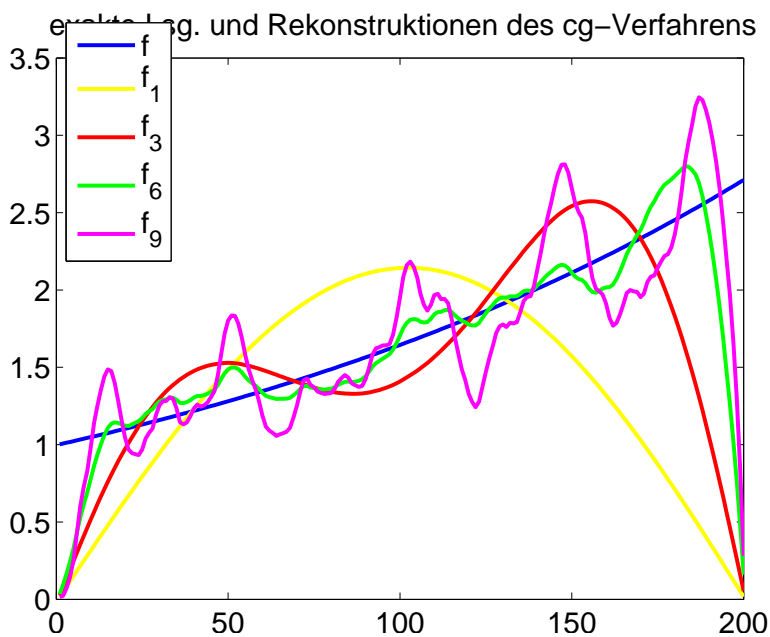


Plotte ein paar cg-Rekonstruktionen

```

plot(f,'LineWidth',2); hold on;
title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
plot(f_t(:,1),'Color','y','LineWidth',2);
plot(f_t(:,3),'Color','r','LineWidth',2);
plot(f_t(:,6),'Color','g','LineWidth',2);
plot(f_t(:,9),'Color','m','LineWidth',2);
legend('f','f_1','f_3','f_6','f_9',2); hold off;

```



WAS IST DA PASSIERT ??????????????

% Wie kann Abhilfe geschaffen werden ????

pause

% Waehle Anfangswert, der die Randwerte der  
% exakten Lsg. annimmt

clear f\_t res error;

f\_0 = (exp(1)-1)\*linspace(0,1,200)+ones(1,200);

g\_eps = g\_eps -A\*f\_0';

% Verfahren der konjugierten Gradienten

[f\_t,res]=cgls(A,g\_eps,10); % 10 Schritte

pause

% Plote Residuen der cg-Iterierten;

clf; set(gca,'fontsize',15);

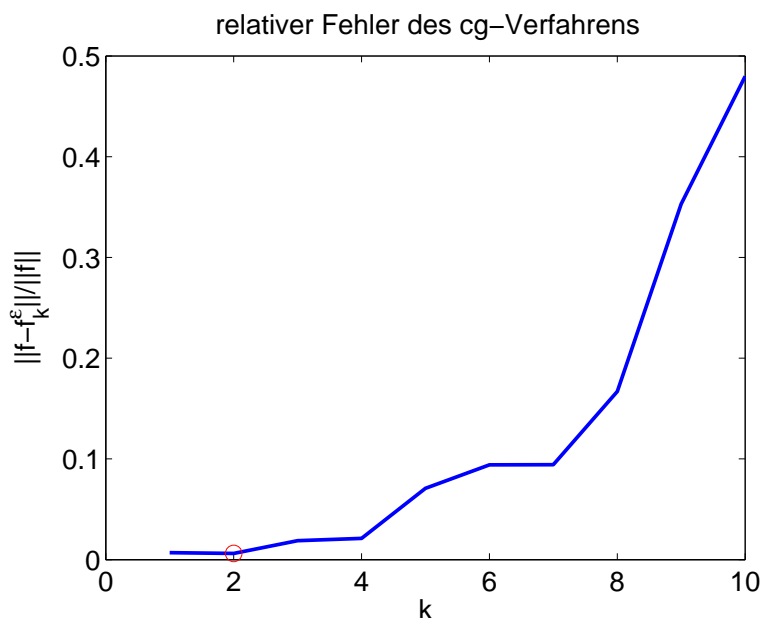
semilogy(res,'o');

title('Residueen des cg-Verfahrens'); xlabel('k');

ylabel('||A f\_k^\epsilon - g^\epsilon||');

hold on; semilogy(E(1:10),'Color','r','LineWidth',2);

hold off;



Plote ein paar cg-Rekonstruktionen

plot(f,'LineWidth',2); hold on;

title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');

```
plot(f_t(:,2)+f_0', 'Color', 'r', 'LineWidth', 2);  
plot(f_t(:,4)+f_0', 'Color', 'y', 'LineWidth', 2);  
plot(f_t(:,6)+f_0', 'Color', 'g', 'LineWidth', 2);  
legend('f', 'f_2', 'f_4', 'f_6', 2); hold off;
```

