
Table of Contents

.....	1
Das lineare System	1
Tikhonov-Regularisierung	10
Landweber-Verfahren	14
nu-Methode nach Brakhage, nu=1	17
Verfahren der konjugierten Gradienten	21
Plotte exakte Lsg. und Rekonstruktion der Verfahren	24
Zum Abschluss nun noch ein LEHRREICHES Beispiel,	26

```
clear;
echo on;

%% 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;
```

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);
pause; echo off;
f=sqrt(dim)*f; g=-sqrt(dim)*g; A=-A;
xt=linspace(0,1,200);
echo on;

clf;
set(gca,'fontsize',16);

plot(xt,f,'LineWidth',2.5); legend('exakte Lsg. f');
pause;

plot(xt,g,'LineWidth',2.5); legend('exakte Daten g');
pause;

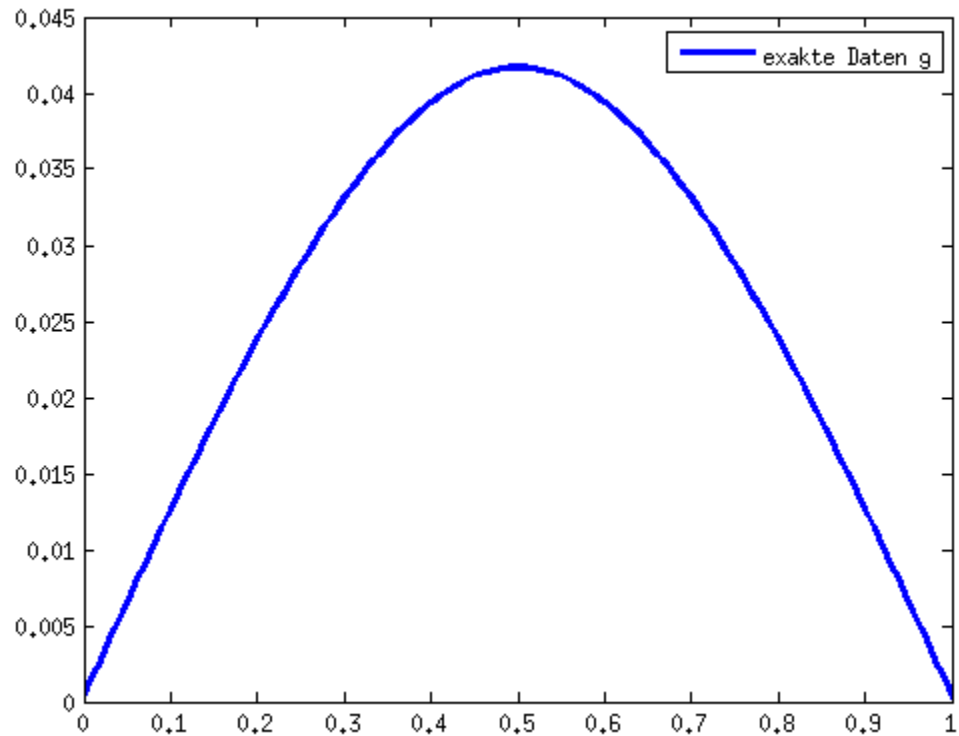
[A,g,f]=deriv2(dim,3);
pause; echo off;

clf;
set(gca,'fontsize',16);

plot(xt,f,'LineWidth',2.5); legend('exakte Lsg. f');
```

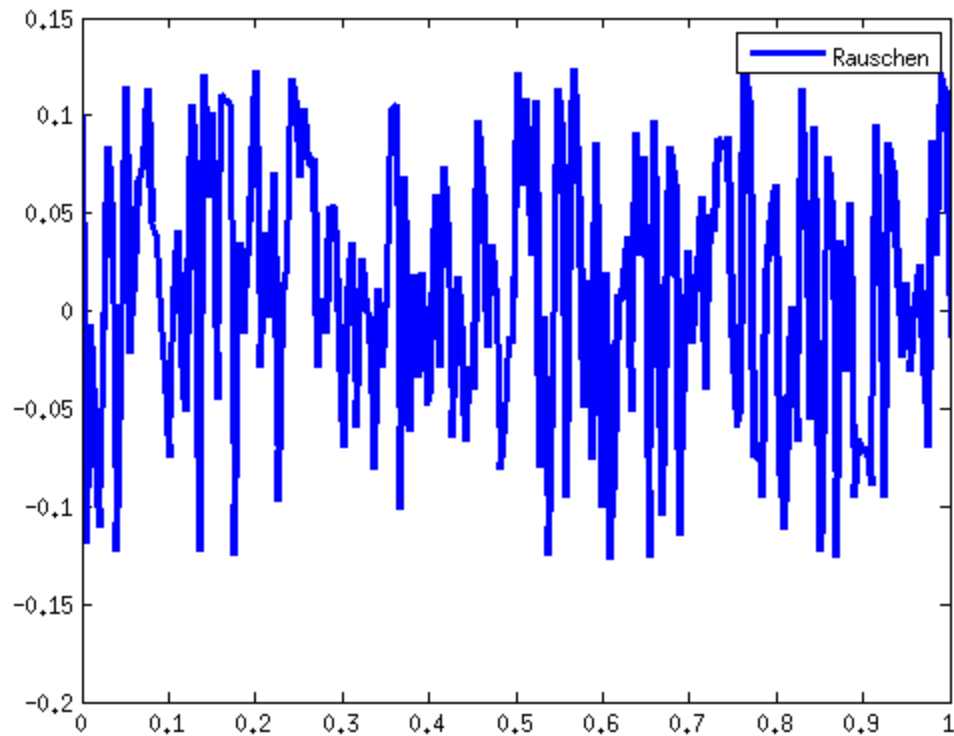
```
pause;
```

```
plot(xt,g, 'LineWidth',2.5); legend('exakte Daten g');  
pause;  
%%  
% Erzeuge Rauschen  
rand('seed',41997);
```



Erzeuge Rauschen

```
rand('seed',41997);  
rausch = (2*rand(dim,1)-ones(dim,1));  
rausch =rausch/norm(rausch);  
plot(xt,rausch, 'LineWidth',2.5); legend('Rauschen');  
pause;  
  
rausch = (2*rand(dim,1)-ones(dim,1));  
rausch =rausch/norm(rausch);  
plot(xt,rausch, 'LineWidth',2.5); legend('Rauschen');  
pause;  
%%  
% Addiere 1% rel. Rauschen zu den Daten,  
% d.h.  $\epsilon = 0.01 * ||g||_2$   
 $\epsilon = 0.01 * \text{norm}(g);$ 
```



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

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

% Plotte Ausschnitt von g und g_eps

```
plot(xt(60:140),g_eps(60:140), 'LineWidth',2.5); hold on;
plot(xt(60:140),g(60:140), 'Color','r', 'LineWidth',2.5); axis tight;
legend('verrauschte Daten  $g^\epsilon$ ', 'exakte Daten g');
hold off;
pause;
```

```
g_eps = g + eps*rausch;
```

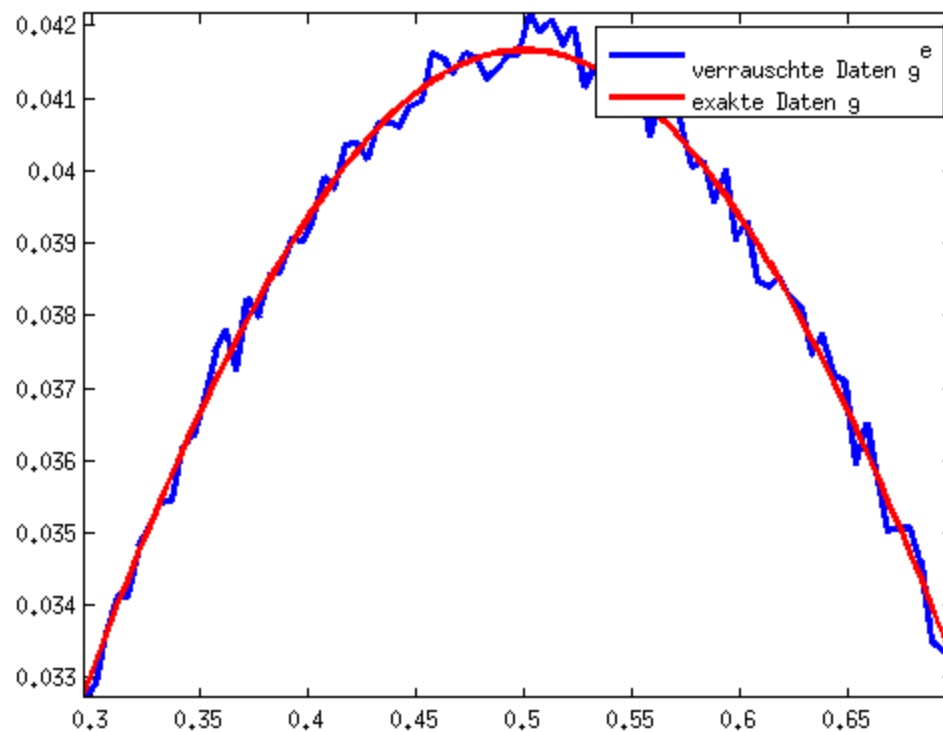
% Plotte Ausschnitt von g und g_eps

```
plot(xt(60:140),g_eps(60:140), 'LineWidth',2.5); hold on;
plot(xt(60:140),g(60:140), 'Color','r', 'LineWidth',2.5); axis tight;
legend('verrauschte Daten  $g^\epsilon$ ', 'exakte Daten g');
hold off;
pause;
```

```
%%
```

% Berechne SWZ von A und plote SW

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



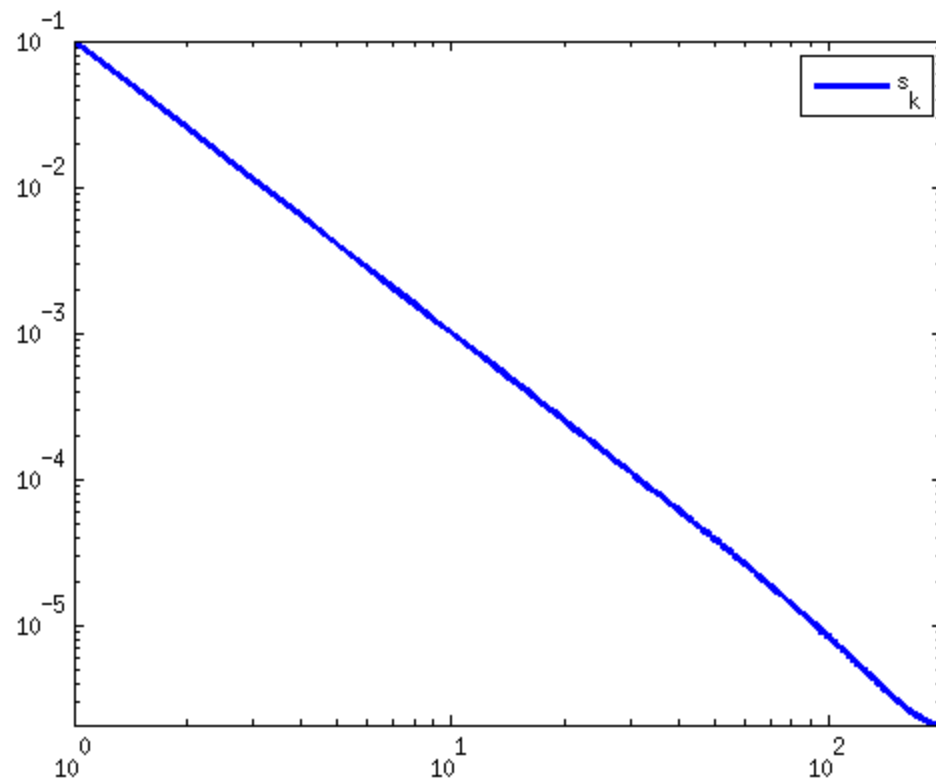
Berechne SWZ von A und plote SW

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

%semilogy(s,'LineWidth',2.5); legend('\sigma_k');
loglog(s,'LineWidth',2.5); legend('\sigma_k'); axis tight;
pause;

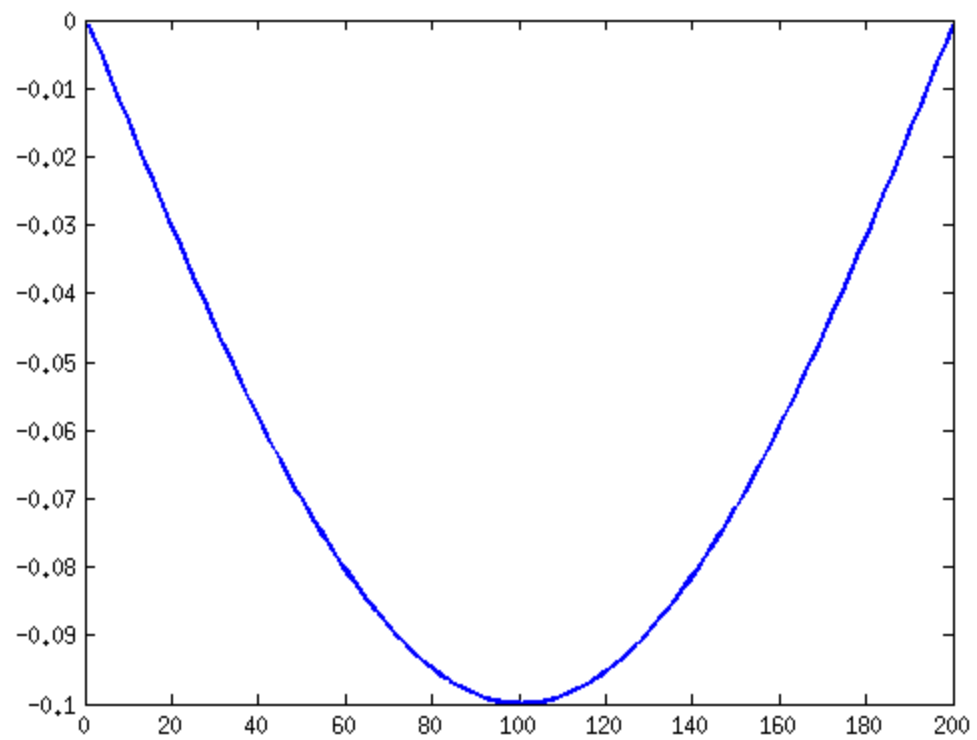
    pause;

    %semilogy(s,'LineWidth',2.5); legend('\sigma_k');
    loglog(s,'LineWidth',2.5); legend('\sigma_k'); axis tight;
    pause;
    %%
    % Plote singulaere Vektoren nach fallenden SW
    %
    plot(U(:,1), 'LineWidth',2);
```

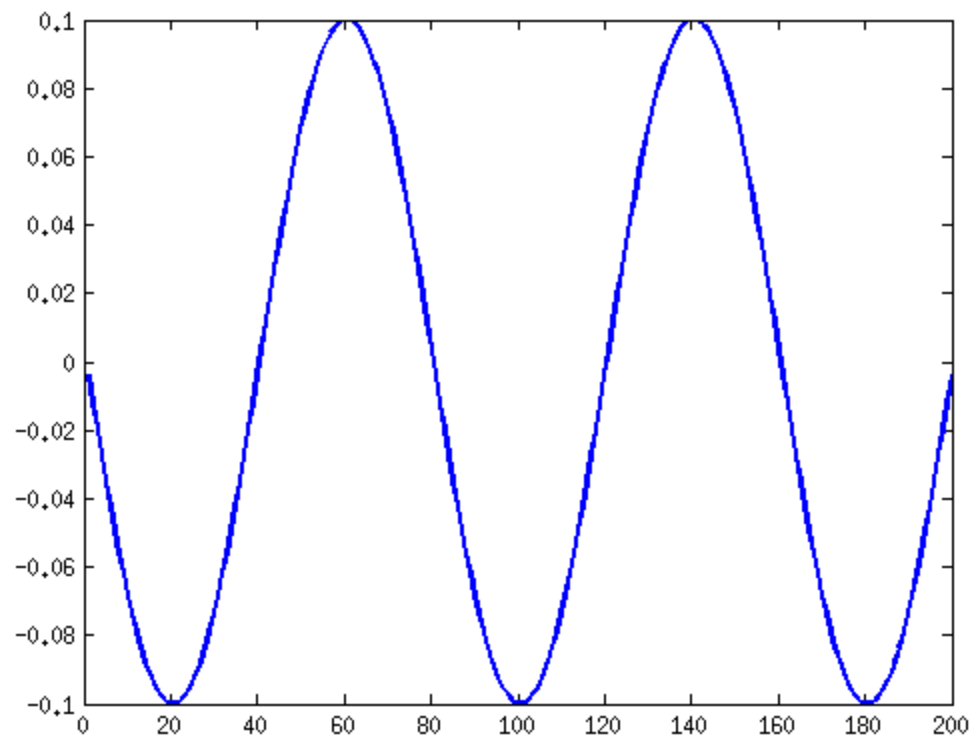


Plotte singuläre Vektoren nach fallenden SW

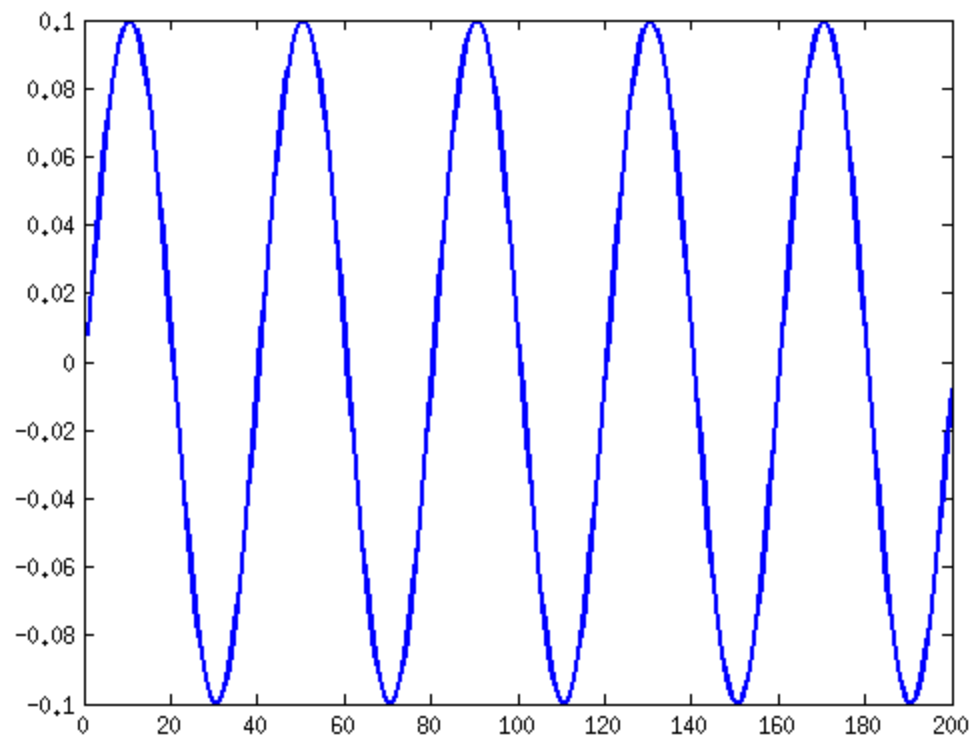
```
plot(U(:,1), 'LineWidth',2);  
pause;  
  
    pause;  
    %%  
    plot(U(:,5), 'LineWidth',2);
```



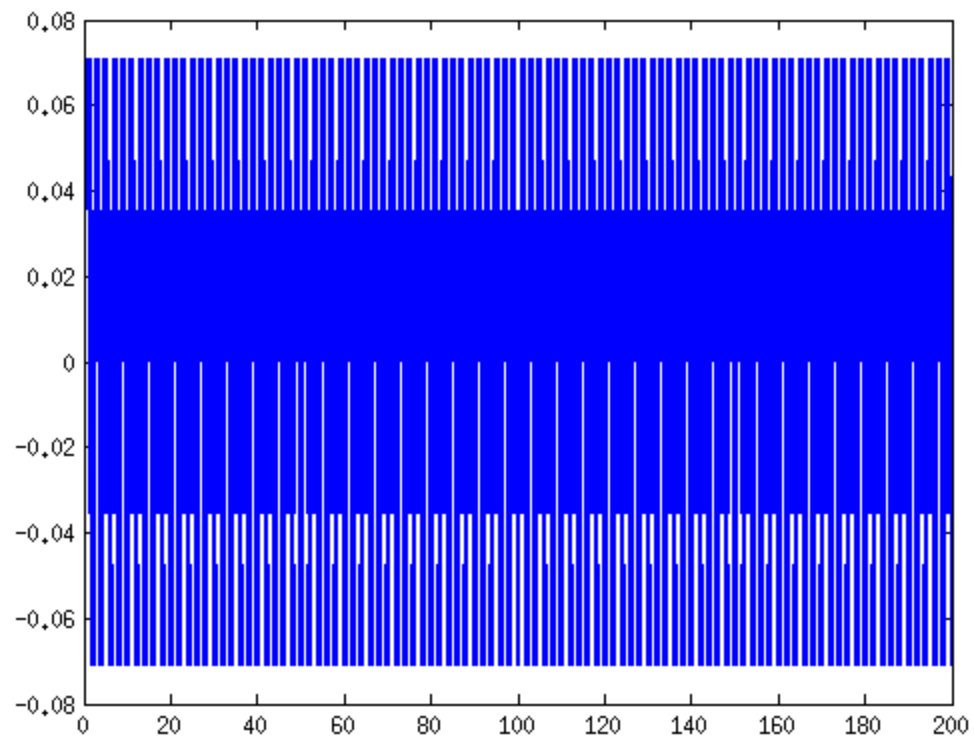
```
plot(U(:,5), 'LineWidth', 2);  
pause;  
  
    pause;  
    %%  
    plot(U(:,10), 'LineWidth', 2);
```



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

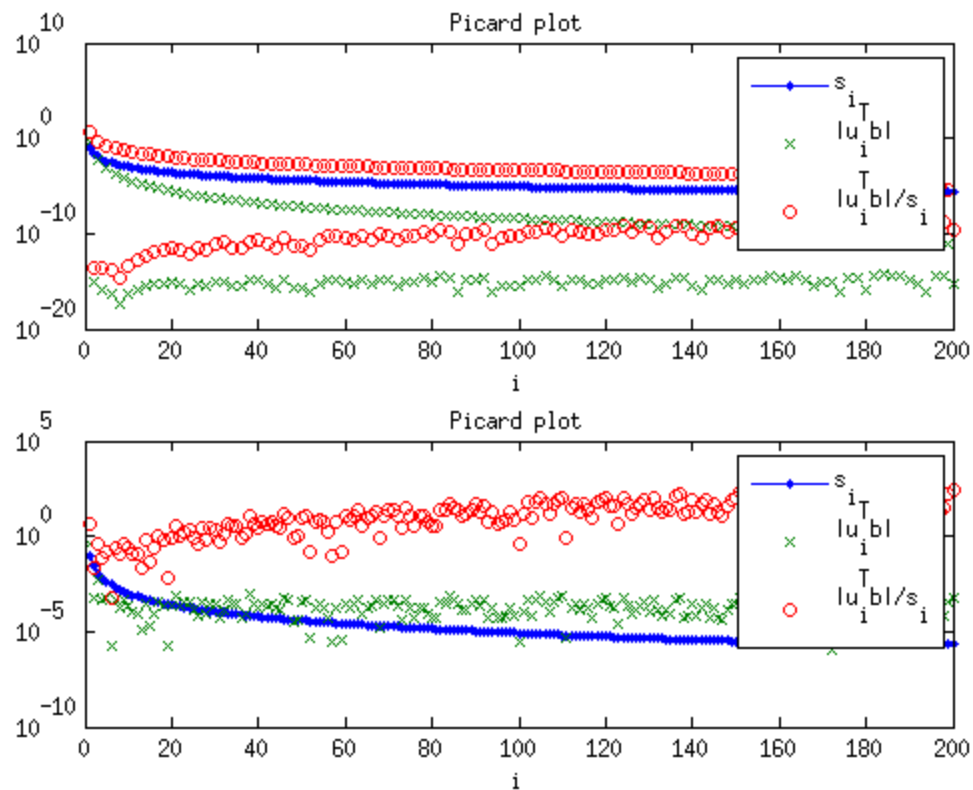


```
plot(U(:,dim), 'LineWidth',2);  
pause;  
  
    pause;  
  
    %%  
    % Ueberpruefe Picard-Bedingung  
    subplot(2,1,1); set(gca, 'fontsize',15); picard(U,s,g);
```

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);  
pause;  
  
    subplot(2,1,2); set(gca, 'fontsize',15); picard(U,s,g_eps);  
    pause;  
  
%% Tikhonov-Regularisierung  
t=[1/4 1/8 1/16 1/32 1/64 1/128 1/256 1/512 1/1024 1/2048];
```



Tikhonov-Regularisierung

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

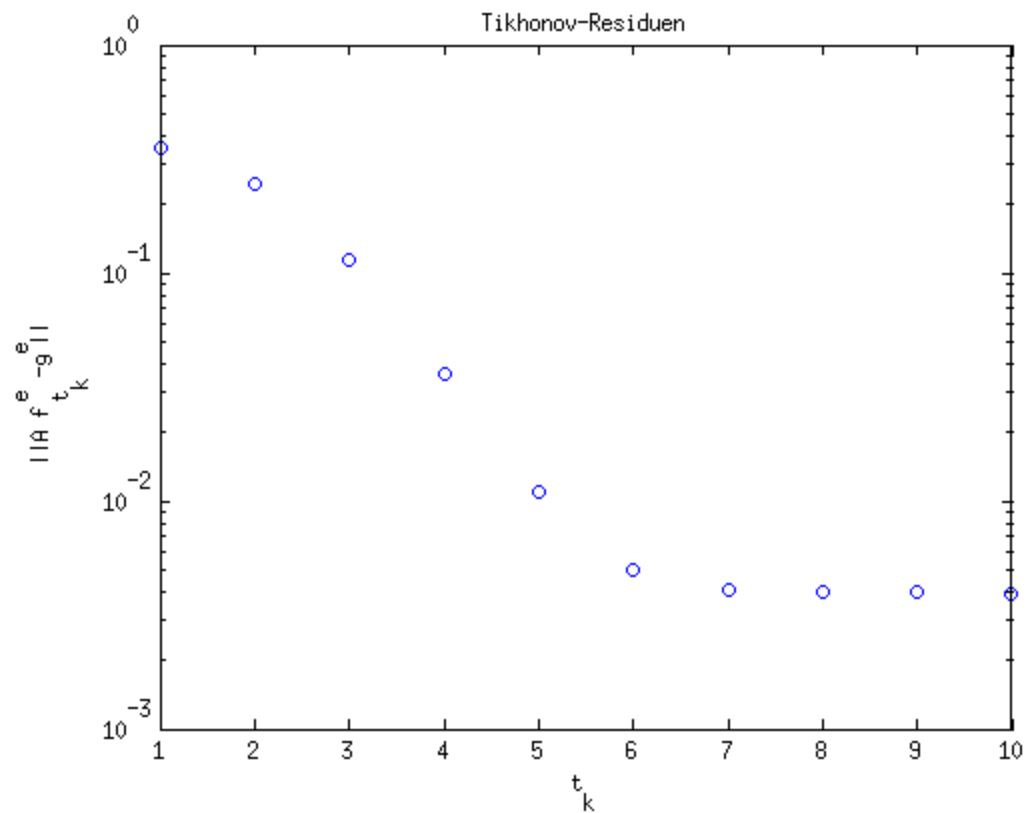
[f_t,res]=tikhonov(U,s,V,g_eps,t);
pause;

% Plotte Residuen zu t_i;

clf; set(gca,'fontsize',16);
semilogy(res,'o'); title('Tikhonov-Residuen'); xlabel('t_k');
ylabel('||A f_{t_k}^\epsilon - g^\epsilon||');
pause;

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

% Plotte Residuen zu t_i;
clf; set(gca,'fontsize',16);
semilogy(res,'o'); title('Tikhonov-Residuen'); xlabel('t_k');
ylabel('||A f_{t_k}^\epsilon - g^\epsilon||');
pause;
%%
% Plotte Abbruchschwelle tau*eps mit tau=1.1
tau = 1.1;
```

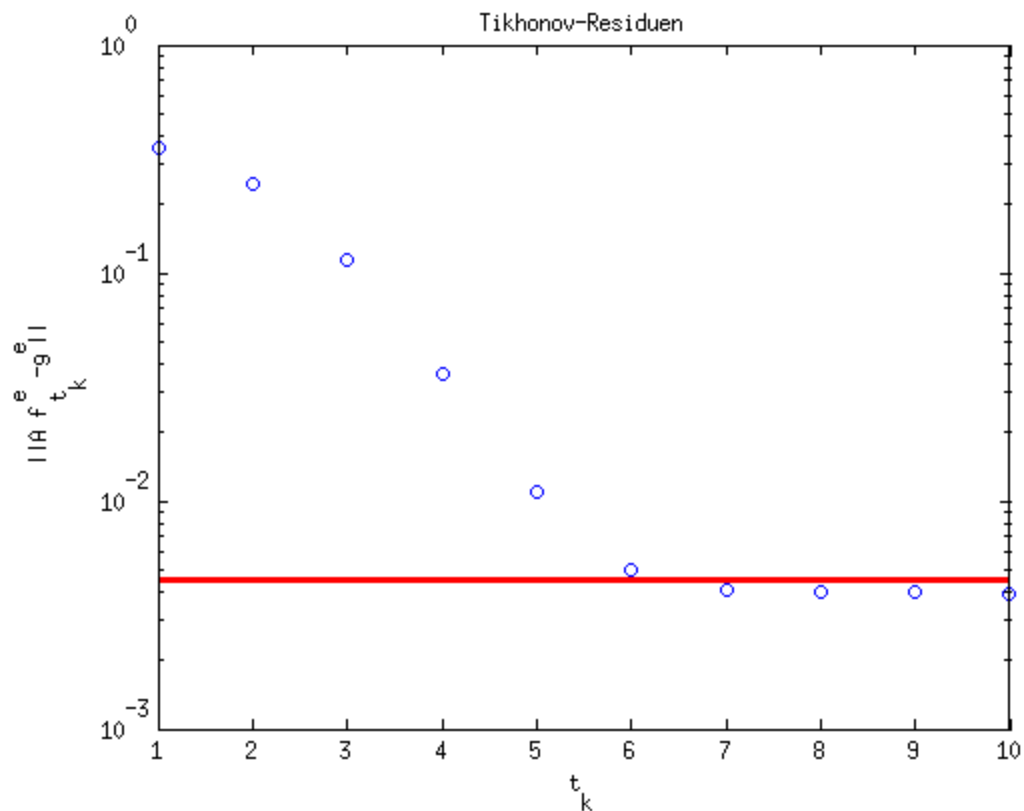


Plotte Abbruchschwelle $\tau \cdot \epsilon$ mit $\tau=1.1$

```
tau = 1.1;
```

```
E = tau*eps*ones(1,100);
hold on; semilogy(E(1:10), 'Color', 'r', 'LineWidth', 2.5); hold off;
pause;
```

```
E = tau*eps*ones(1,100);
hold on; semilogy(E(1:10), 'Color', 'r', 'LineWidth', 2.5); hold off;
pause;
%%
% Plotte ein paar Rekonstruktionen
plot(xt,f, 'LineWidth', 2); hold on;
```



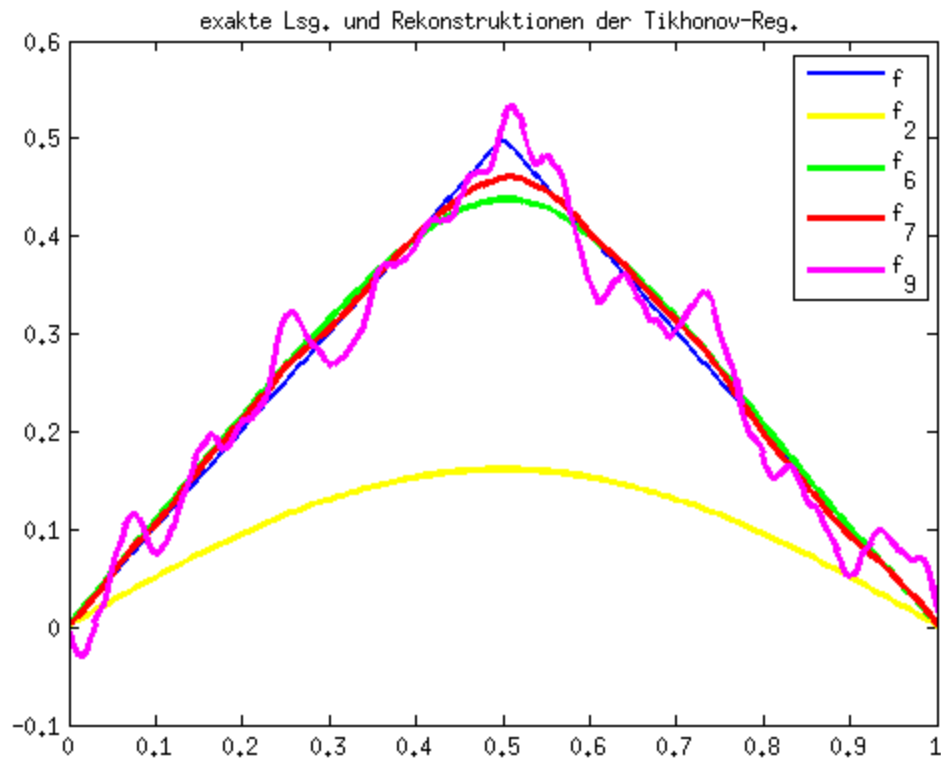
Plotte ein paar Rekonstruktionen

```

plot(xt,f,'LineWidth',2); hold on;
title('exakte Lsg. und Rekonstruktionen der Tikhonov-Reg. ');
plot(xt,f_t(:,2),'Color','y','LineWidth',2.5);
plot(xt,f_t(:,6),'Color','g','LineWidth',2.5);
plot(xt,f_t(:,7),'Color','r','LineWidth',2.5);
plot(xt,f_t(:,9),'Color','m','LineWidth',2.5);
legend('f','f_2','f_6','f_7','f_9'); hold off;
pause;

title('exakte Lsg. und Rekonstruktionen der Tikhonov-Reg. ');
plot(xt,f_t(:,2),'Color','y','LineWidth',2.5);
plot(xt,f_t(:,6),'Color','g','LineWidth',2.5);
plot(xt,f_t(:,7),'Color','r','LineWidth',2.5);
plot(xt,f_t(:,9),'Color','m','LineWidth',2.5);
legend('f','f_2','f_6','f_7','f_9'); hold off;
pause;
%%
% Berechne Fehler der Rekonstruktionen
echo 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',2.5);hold on;
plot(7,error(7),'.','Color','r','MarkerSize',30);
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
pause;

plot(error,'LineWidth',2.5);hold on;
plot(7,error(7),'.','Color','r','MarkerSize',30);
title('absoluter Fehler derTikhonov-Reg. '); xlabel('t_k');
ylabel('||f-f_{t_k}^\epsilon||'); hold off;
```

```
% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip
```

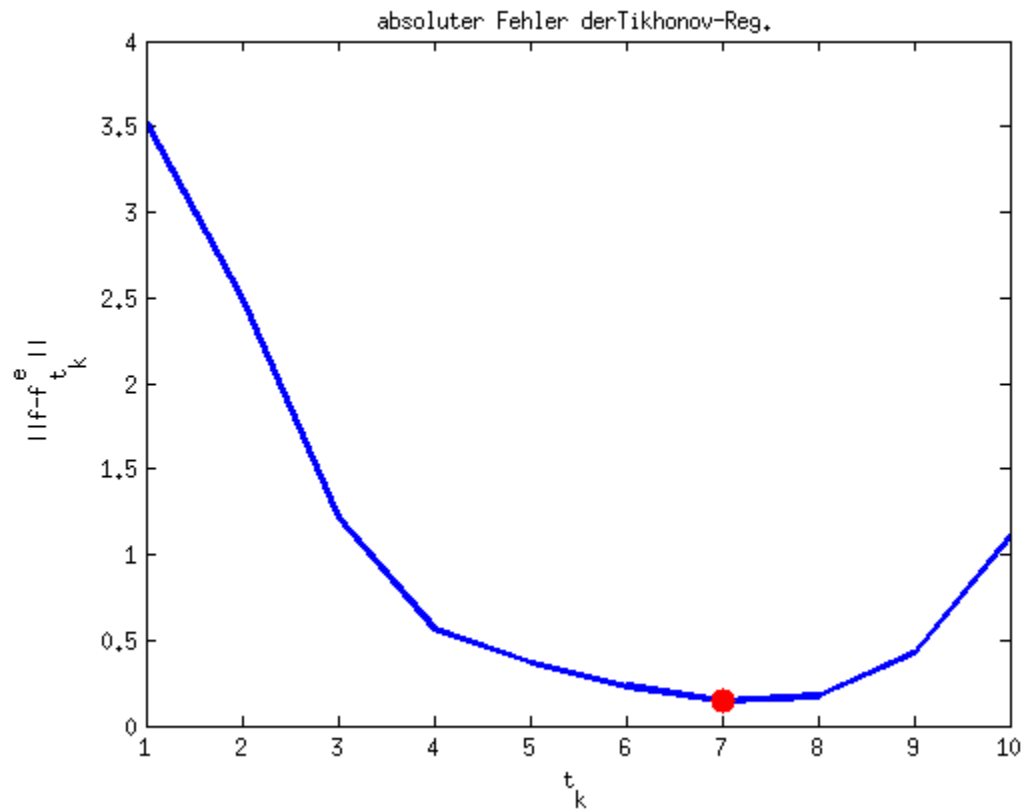
```
f_tik=f_t(:,7);  
err_tik=error(:,7);
```

```
% Ende Tikhonov-Regularisierung
```

```
pause;
```

```
%% Landweber-Verfahren
```

```
clear f_t res error;
```



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',16);  
semilogy(res(1:100),'o'); title('Residuen des Landweber-Verf.');
```

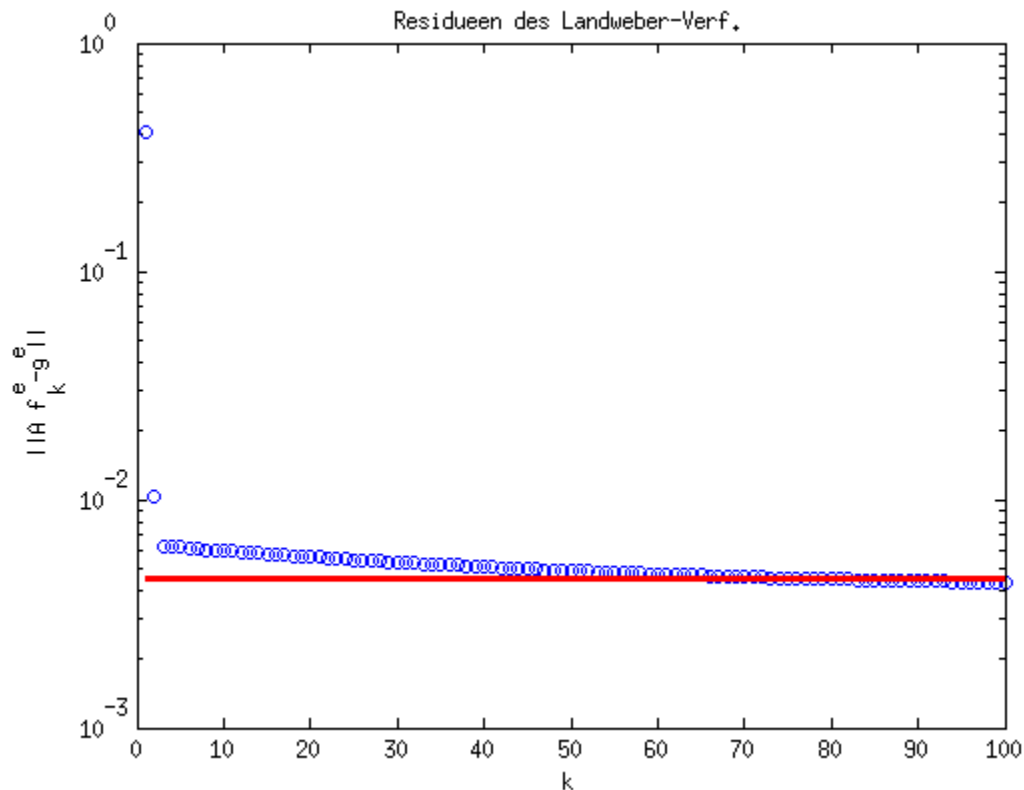
```
xlabel('k'); ylabel('||A f_k^\epsilon - g^\epsilon||');
```

```
hold on; semilogy(E(1:100),'Color','r','LineWidth',2.5); hold off;  
pause;
```

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

pause

```
% Plote Residuen des Landweber-Verfahrens
clf; set(gca, 'fontsize', 16);
semilogy(res(1:100), 'o'); title('Residueen des Landweber-Verf. ');
xlabel('k'); ylabel('||A f_k^\epsilon - g^\epsilon||');
hold on; semilogy(E(1:100), 'Color', 'r', 'LineWidth', 2.5); hold off;
pause;
%%
% Berechne Fehler der Rekonstruktionen des Landweber-Verfahrens
echo off;
```



Berechne Fehler der Rekonstruktionen des Landweber-Verfahrens

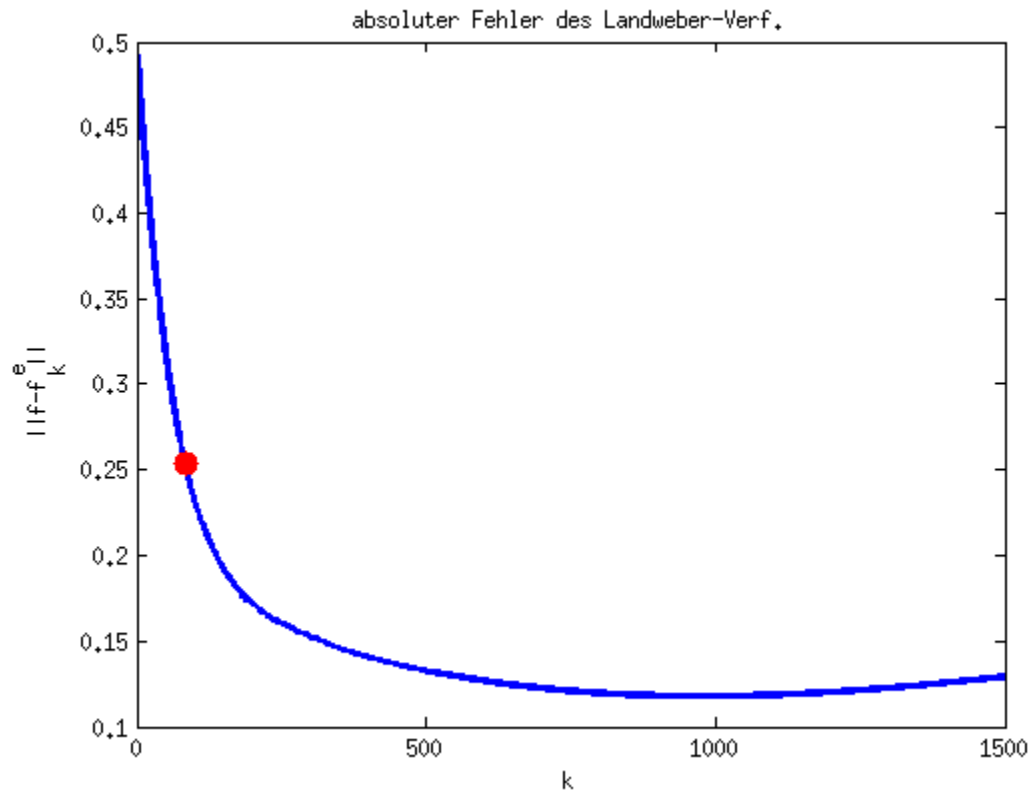
```
echo off;
for i=1:1500;
    error(i)=norm(f-f_t(:,i));
end;
clf; set(gca, 'fontsize', 16);
echo on;

plot(error, 'LineWidth', 2.5); hold on;
plot(83, error(83), '.', 'Color', 'r', 'MarkerSize', 30);
title('absoluter Fehler des Landweber-Verf. '); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;
pause;
```

```

plot(error, 'LineWidth', 2.5); hold on;
plot(83, error(83), '.', 'Color', 'r', 'MarkerSize', 30);
title('absoluter Fehler des Landweber-Verf. '); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;
pause;
%%
% Plote ein paar Rekonstruktionen des Landweber-Verfahrens
plot(xt, f, 'LineWidth', 2.5); hold on;

```



Plote ein paar Rekonstruktionen des Landweber-Verfahrens

```

plot(xt, f, 'LineWidth', 2.5); hold on;
title('exakte Lsg. und Rekonstruktionen des Landweber-Verf. ');
plot(xt, f_t(:, 10), 'Color', 'y', 'LineWidth', 2.5);
plot(xt, f_t(:, 60), 'Color', 'g', 'LineWidth', 2.5);
plot(xt, f_t(:, 83), 'Color', 'r', 'LineWidth', 2.5);
plot(xt, f_t(:, 1000), 'Color', 'm', 'LineWidth', 2.5);
legend('f', 'f_{10}', 'f_{60}', 'f_{83}', 'f_{1000}'); hold off;
pause;

% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip

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

% Ende Landweber-Verfahren

```

```
pause;
```

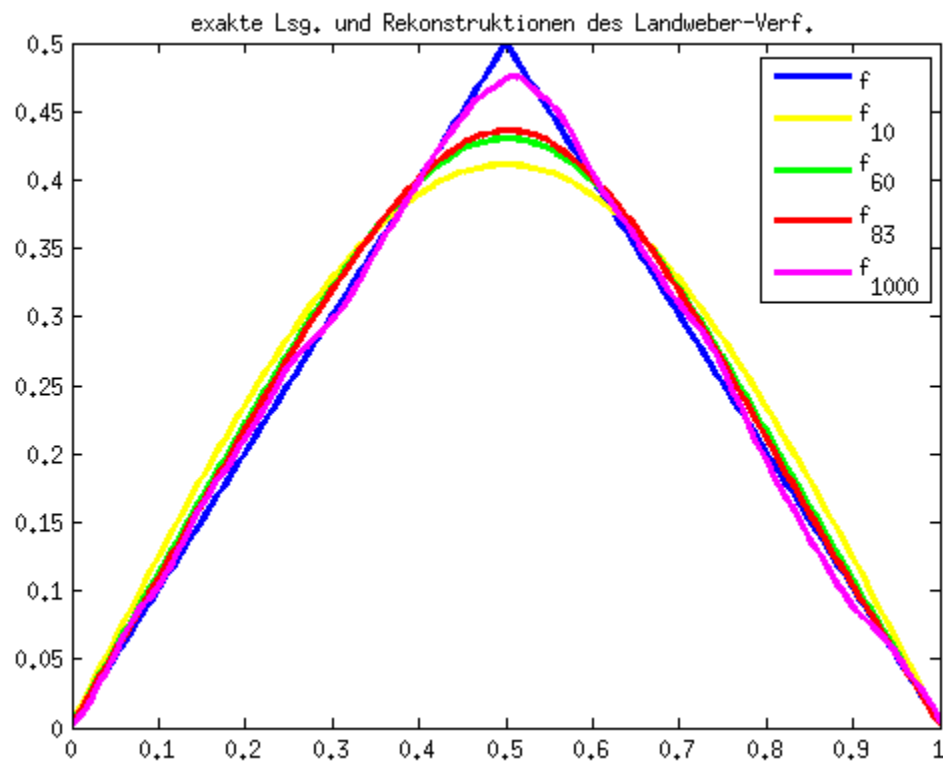
```
title('exakte Lsg. und Rekonstruktionen des Landweber-Verf.');
```

```
plot(xt,f_t(:,10), 'Color', 'y', 'LineWidth',2.5);  
plot(xt,f_t(:,60), 'Color', 'g', 'LineWidth',2.5);  
plot(xt,f_t(:,83), 'Color', 'r', 'LineWidth',2.5);  
plot(xt,f_t(:,1000), 'Color', 'm', 'LineWidth',2.5);  
legend('f', 'f_{10}', 'f_{60}', 'f_{83}', 'f_{1000}'); hold off;  
pause;
```

```
% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip  
f_land=f_t(:,83);  
err_land=error(:,83);
```

```
% Ende Landweber-Verfahren  
pause;
```

```
%% nu-Methode nach Brakhage, nu=1  
clear f_t res error;
```



nu-Methode nach Brakhage, nu=1

```
clear f_t res error;
```

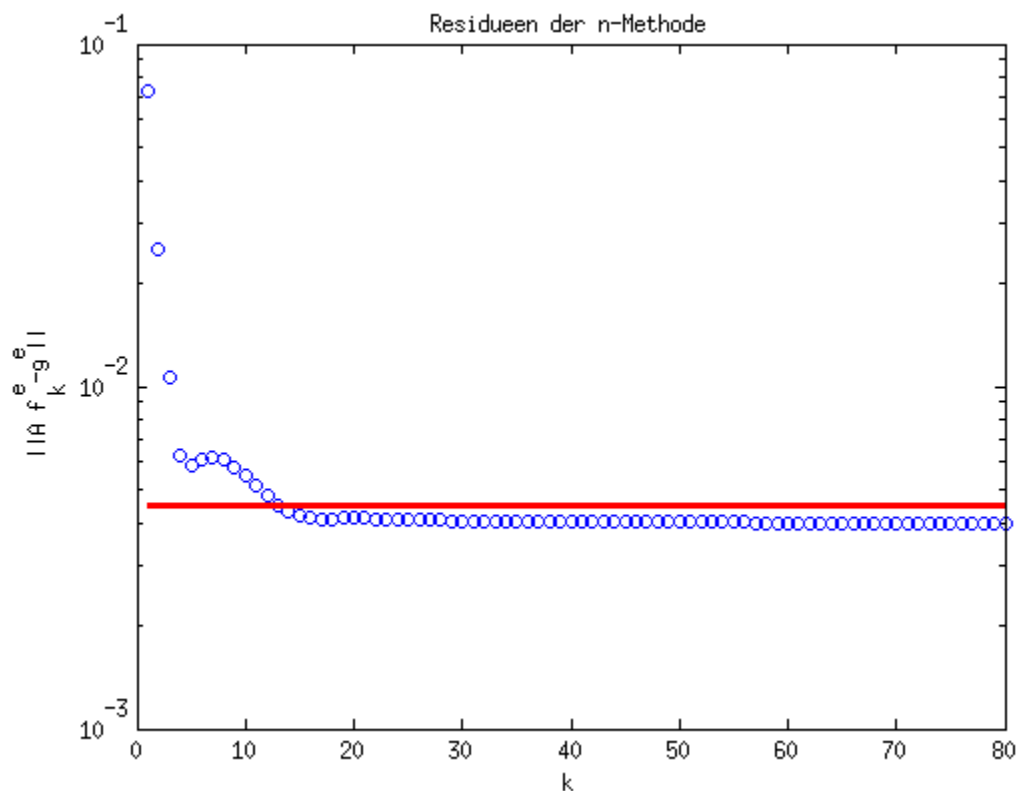
```
[f_t,res]=nu(A,g_eps,80,1); % 80 Schritte  
pause
```

```
% Plote Residuen der Iterierten der nu-Methode ;
```

```
clf; set(gca,'fontsize',16);
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',2.5); hold off;
pause;
```

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

```
% Plote Residuen der Iterierten der nu-Methode ;
clf; set(gca,'fontsize',16);
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',2.5); hold off;
pause;
%%
% Berechne Fehler der Rekonstruktionen der nu-Methode
echo off;
```



```
Berechne Fehler der Rekonstruktionen der nu-Methode
```

```
echo off;
for i=1:80
```

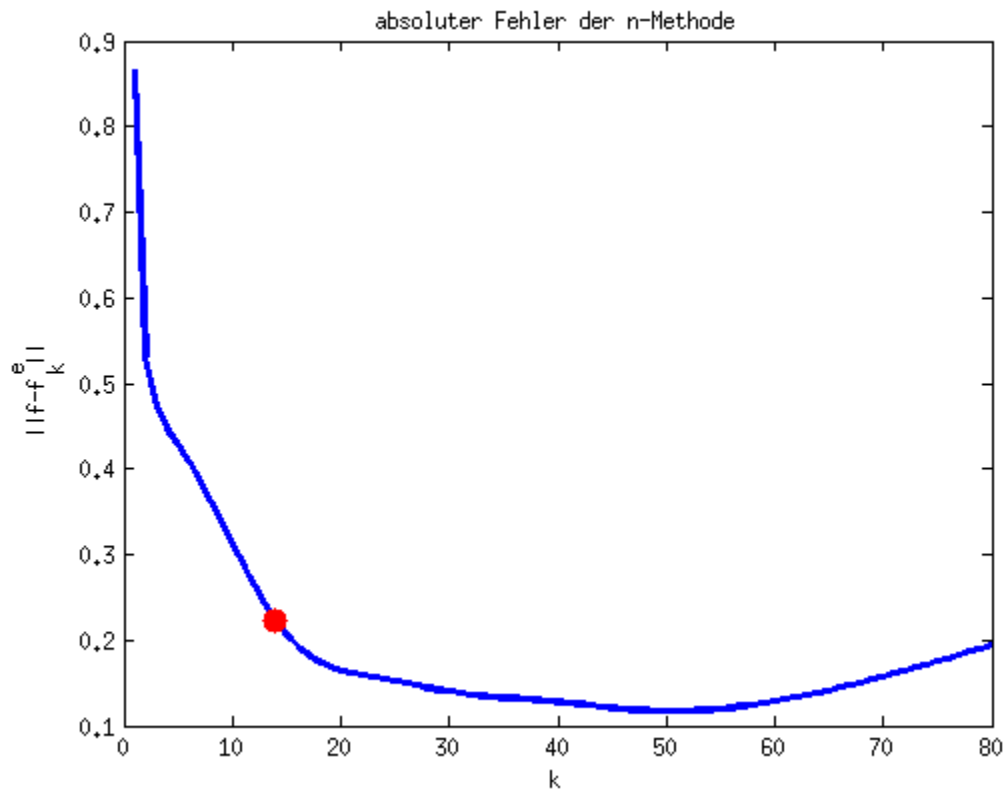
```

    error(i)=norm(f-f_t(:,i));
end;
clf; set(gca, 'fontsize',16);
echo on;

plot(error, 'LineWidth',2.5);hold on;
plot(14,error(14),'.','Color','r','MarkerSize',30);
title('absoluter Fehler der \nu-Methode'); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;
pause;

plot(error, 'LineWidth',2.5);hold on;
plot(14,error(14),'.','Color','r','MarkerSize',30);
title('absoluter Fehler der \nu-Methode'); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;
pause;
%%
% Plote ein paar Rekonstruktionen der nu-Methode
plot(xt,f, 'LineWidth',2.5); hold on;

```



Plote ein paar Rekonstruktionen der nu-Methode

```

plot(xt,f, 'LineWidth',2.5); hold on;
title('exakte Lsg. und Rekonstruktionen der \nu-Methode');
plot(xt,f_t(:,4), 'Color','y', 'LineWidth',2.5);
plot(xt,f_t(:,12), 'Color','g', 'LineWidth',2.5);

```

```
plot(xt,f_t(:,14), 'Color','r', 'LineWidth',2.5);
plot(xt,f_t(:,80), 'Color','m', 'LineWidth',2.5); axis tight;
legend('f', 'f_4', 'f_{12}', 'f_{14}', 'f_{80}'); hold off;
pause;
```

```
% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip
```

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

```
% Ende nu-Methode nach Brakhage
pause;
```

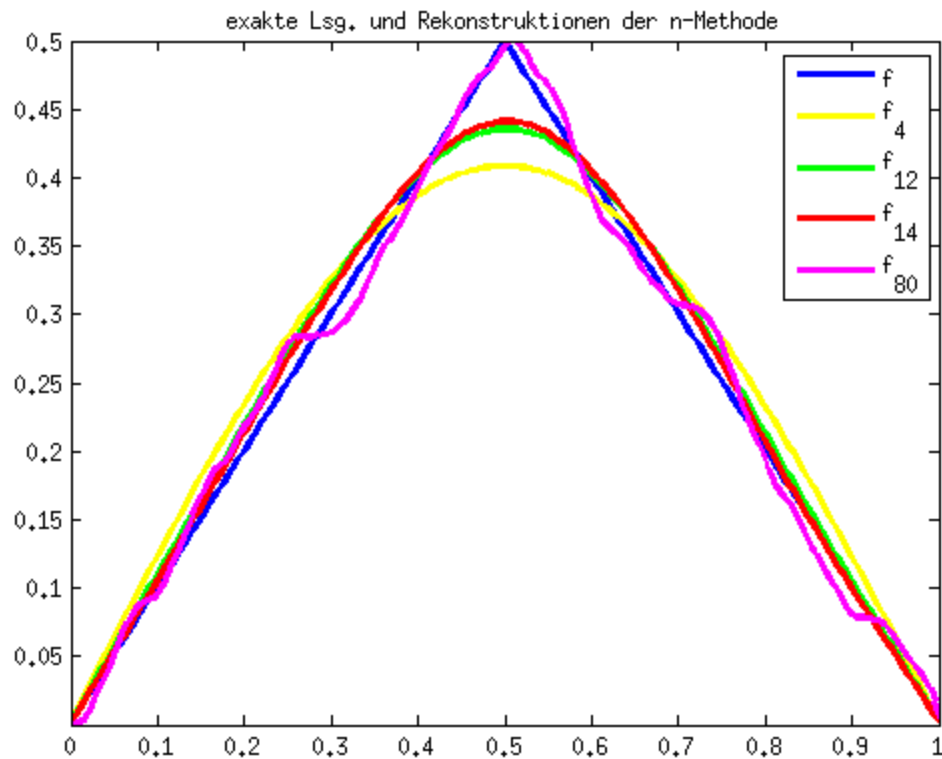
```
clear f_t res error;
```

```
    title('exakte Lsg. und Rekonstruktionen der \nu-Methode');
    plot(xt,f_t(:,4), 'Color','y', 'LineWidth',2.5);
    plot(xt,f_t(:,12), 'Color','g', 'LineWidth',2.5);
    plot(xt,f_t(:,14), 'Color','r', 'LineWidth',2.5);
    plot(xt,f_t(:,80), 'Color','m', 'LineWidth',2.5); axis tight;
    legend('f', 'f_4', 'f_{12}', 'f_{14}', 'f_{80}'); hold off;
    pause;
```

```
% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip
f_nu=f_t(:,14);
err_nu=error(:,14);
```

```
% Ende nu-Methode nach Brakhage
pause;
```

```
clear f_t res error;
%% Verfahren der konjugierten Gradienten
[f_t,res]=cgls(A,g_eps,10); % 10 Schritte
```



Verfahren der konjugierten Gradienten

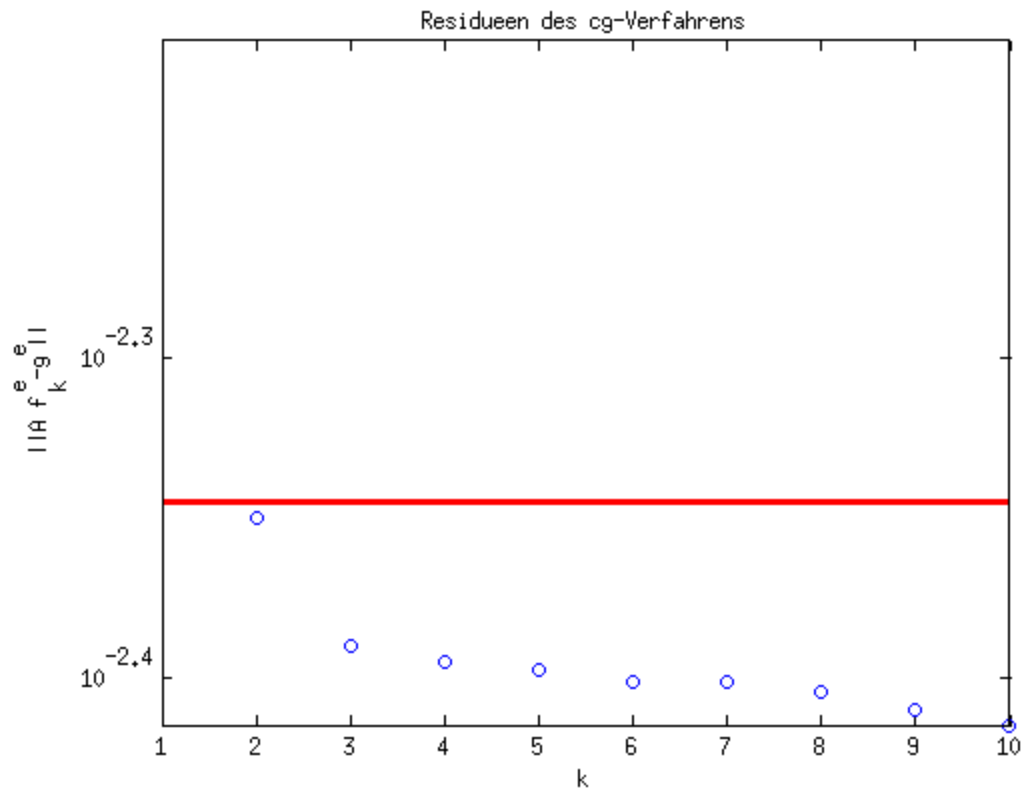
```
[f_t,res]=cgls(A,g_eps,10); % 10 Schritte
pause
```

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

```
clf; set(gca,'fontsize',16);
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',2.5); hold off;
pause;
```

```
pause
```

```
% Plotte Residuen der cg-Iterierten;
clf; set(gca,'fontsize',16);
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',2.5); hold off;
pause;
%%
% Berechne Fehler der cg-Rekonstruktionen
echo off;
```

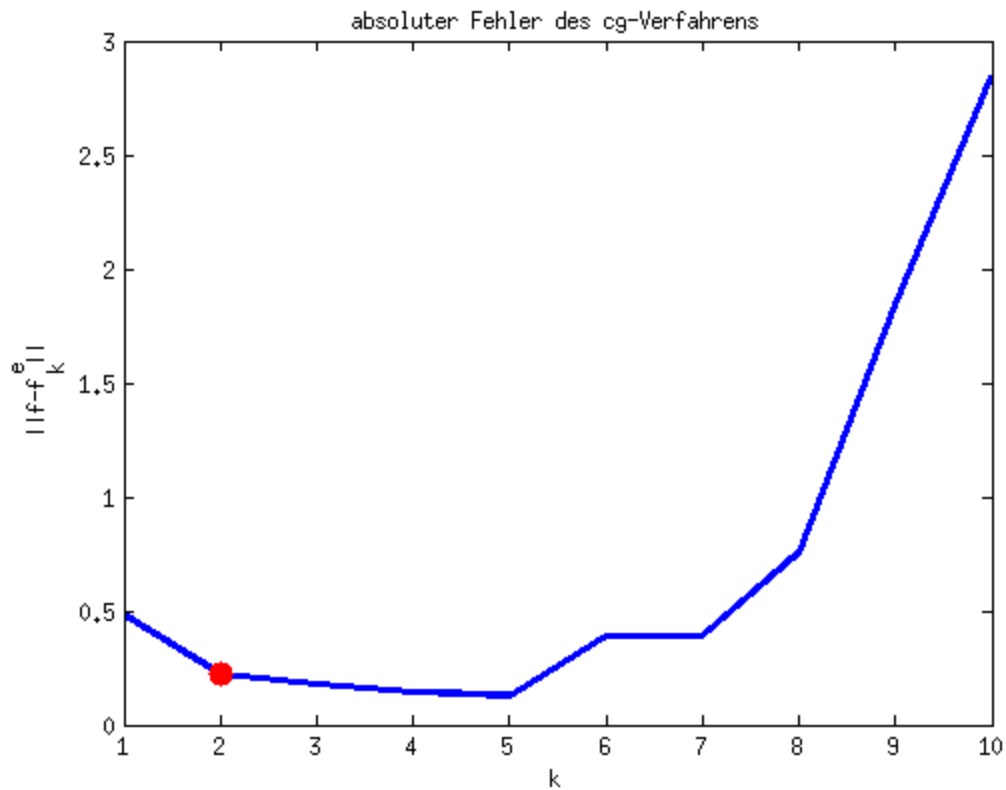


Berechne Fehler der cg-Rekonstruktionen

```
echo off;
for i=1:10
    error(i)=norm(f-f_t(:,i));
end;
clf; set(gca, 'fontsize',16);
echo on;

plot(error, 'LineWidth',2.5);hold on;
plot(2,error(2), '.', 'Color','r', 'MarkerSize',30);
title('absoluter Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;
pause;

plot(error, 'LineWidth',2.5);hold on;
plot(2,error(2), '.', 'Color','r', 'MarkerSize',30);
title('absoluter Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||'); hold off;
pause;
%%
% Plote ein paar cg-Rekonstruktionen
plot(xt,f, 'LineWidth',2.5); hold on;
```



Plotte ein paar cg-Rekonstruktionen

```

plot(xt,f,'LineWidth',2.5); hold on;
title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
plot(xt,f_t(:,1),'Color','y','LineWidth',2.5);
plot(xt,f_t(:,2),'Color','r','LineWidth',2.5);
plot(xt,f_t(:,4),'Color','g','LineWidth',2.5);
plot(xt,f_t(:,8),'Color','m','LineWidth',2.5);
legend('f','f_1','f_{2}','f_4','f_{8}'); hold off;
pause;

% Speichere Rekonstruktion und Fehler nach dem Diskrepanzprinzip

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

% Ende des cg-Verfahrens
pause

    title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
    plot(xt,f_t(:,1),'Color','y','LineWidth',2.5);
    plot(xt,f_t(:,2),'Color','r','LineWidth',2.5);
    plot(xt,f_t(:,4),'Color','g','LineWidth',2.5);
    plot(xt,f_t(:,8),'Color','m','LineWidth',2.5);
    legend('f','f_1','f_{2}','f_4','f_{8}'); hold off;
    pause;

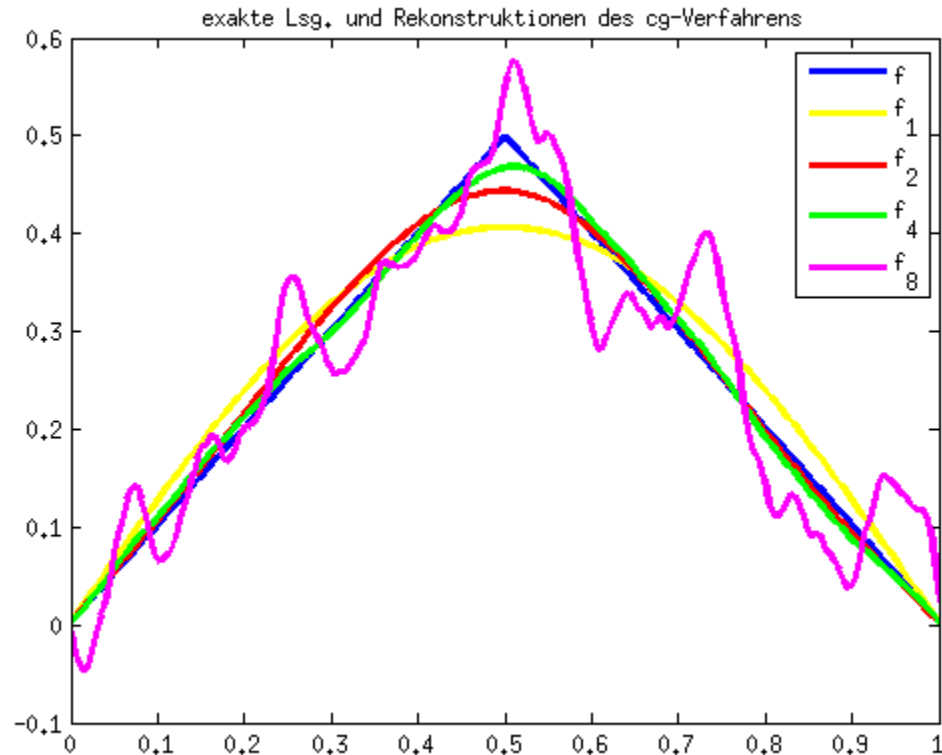
```

```
% 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',16);
```



Plote exakte Lsg. und Rekonstruktion der Verfahren

```
clf; set(gca,'fontsize',16);  
plot(xt,f,'LineWidth',2.5); hold on;  
title('exakte Lsg. und Rekonstruktionen nach Diskrepanzprinzip');  
plot(xt,f_tik,'Color','r','LineWidth',2.5);  
plot(xt,f_land,'Color','y','LineWidth',2.5);  
plot(xt,f_nu,'Color','g','LineWidth',2.5);  
plot(xt,f_cg,'Color','m','LineWidth',2.5);  
legend('f','f_{tik}','f_{land}','f_{nu}','f_{cg}'); hold off;  
pause;
```

```
% Relativer Fehler der Verfahren
```

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

```
pause
```

```
plot(xt,f,'LineWidth',2.5); hold on;  
title('exakte Lsg. und Rekonstruktionen nach Diskrepanzprinzip');  
plot(xt,f_tik,'Color','r','LineWidth',2.5);  
plot(xt,f_land,'Color','y','LineWidth',2.5);  
plot(xt,f_nu,'Color','g','LineWidth',2.5);  
plot(xt,f_cg,'Color','m','LineWidth',2.5);  
legend('f','f_{tik}','f_{land}','f_{nu}','f_{cg}'); hold off;  
pause;
```

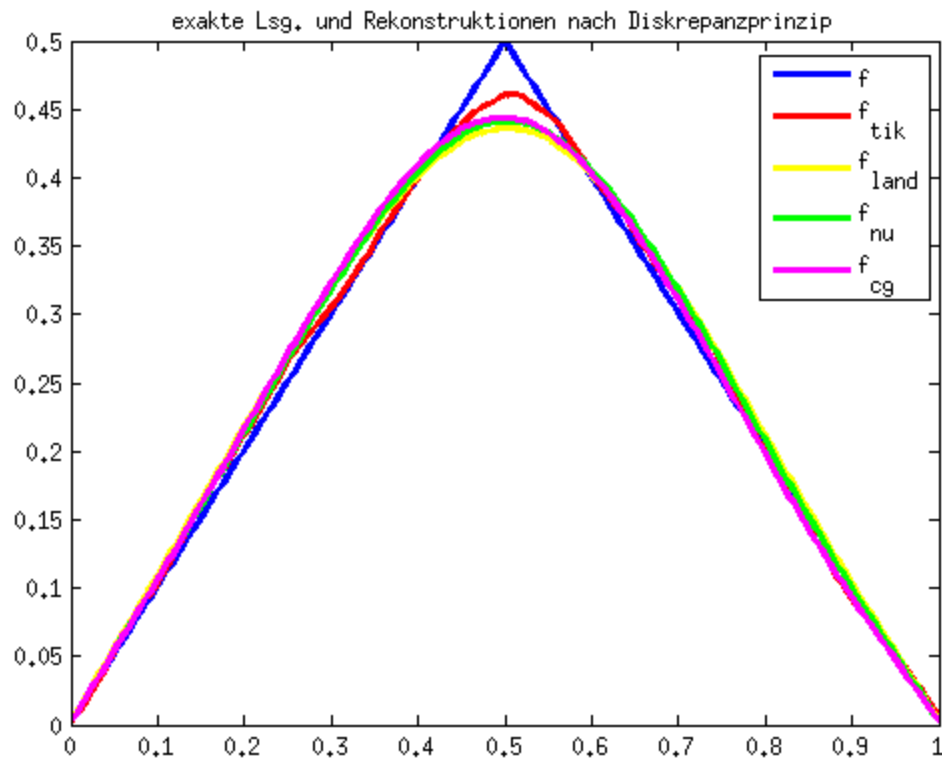
```
% Relativer Fehler der Verfahren  
err=[err_tik err_land err_nu err_cg]./norm(f)
```

```
err =
```

```
0.0354    0.0620    0.0546    0.0543
```

```
pause
```

```
%% 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;
```



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);
pause; echo off;
f=sqrt(dim)*f; g=-sqrt(dim)*g; A=-A;
echo on;

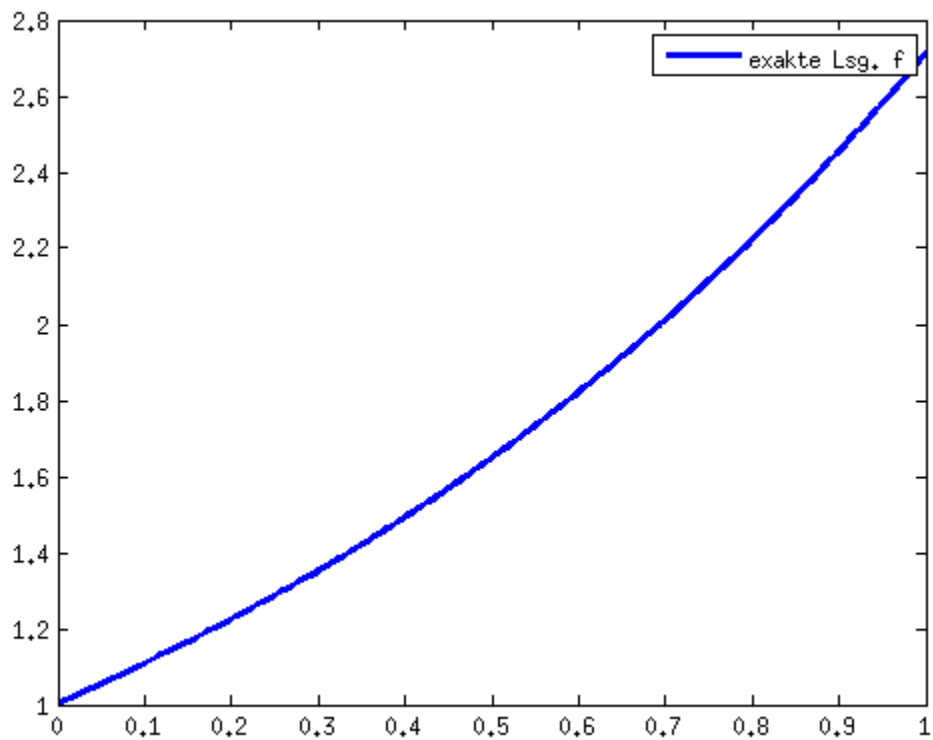
    [A,g,f]=deriv2(dim,2);
    pause; echo off;

    %%
    clf;

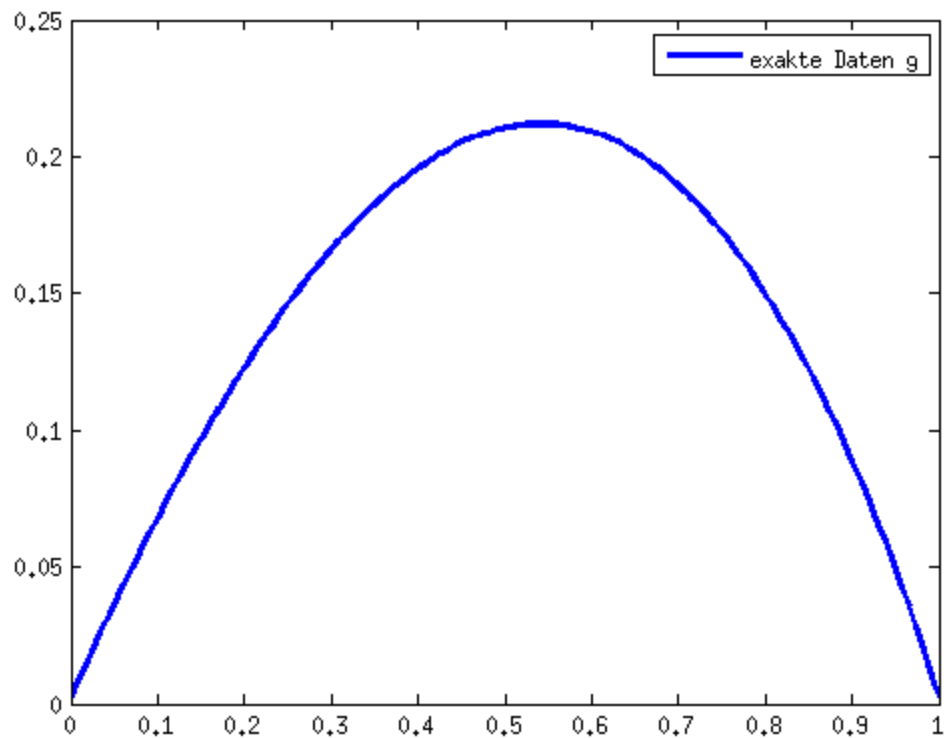
clf;
set(gca,'fontsize',16);
plot(xt,f,'LineWidth',2.5); legend('exakte Lsg. f');
pause;

    set(gca,'fontsize',16);
```

```
plot(xt,f,'LineWidth',2.5); legend('exakte Lsg. f');  
  
pause;  
%%  
plot(xt,g,'LineWidth',2.5); legend('exakte Daten g');
```



```
plot(xt,g,'LineWidth',2.5); legend('exakte Daten g');  
pause;  
  
pause;  
%%  
% Erzeuge Rauschen  
rand('seed',41997);
```



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;

% Plote Ausschnitt von g und g_eps

plot(xt(60:140),g_eps(60:140), 'LineWidth',2.5); hold on;
plot(xt(60:140),g(60:140), 'Color','r', 'LineWidth',2.5); axis tight;
legend('verrauschte Daten g^\epsilon','exakte Daten g')
hold off; pause

% Initialisiere Diskrepanzprinzip

tau = 1.01;

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

```

% Verfahren der konjugierten Gradienten

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

    rausch = (2*rand(dim,1)-ones(dim,1));
    rausch =rausch/norm(rausch);

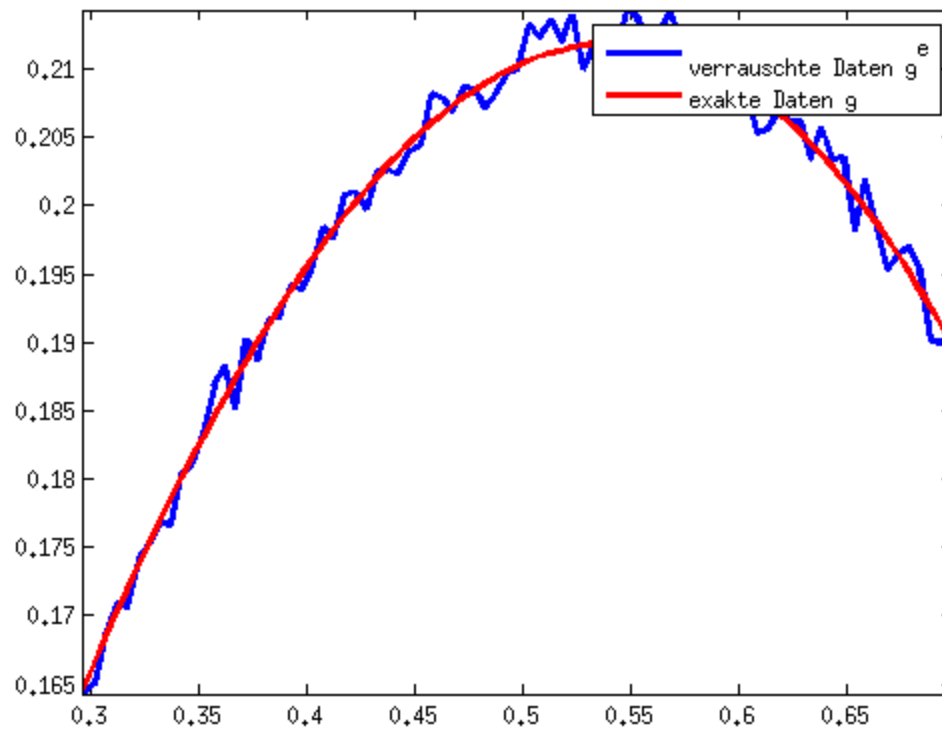
    % Addiere 1% rel. Rauschen zu den Daten,
    % d.h.  eps =0.01*||g||_2
    eps = 0.01*norm(g);
    g_eps = g + eps*rausch;

    % Plotte Ausschnitt von g und g_eps
    plot(xt(60:140),g_eps(60:140),'LineWidth',2.5); hold on;
    plot(xt(60:140),g(60:140),'Color','r','LineWidth',2.5); axis tight;
    legend('verrauschte Daten  $g^{\epsilon}$ ','exakte Daten g')
    hold off; pause

    % Initialisiere Diskrepanzprinzip
    tau = 1.01;

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

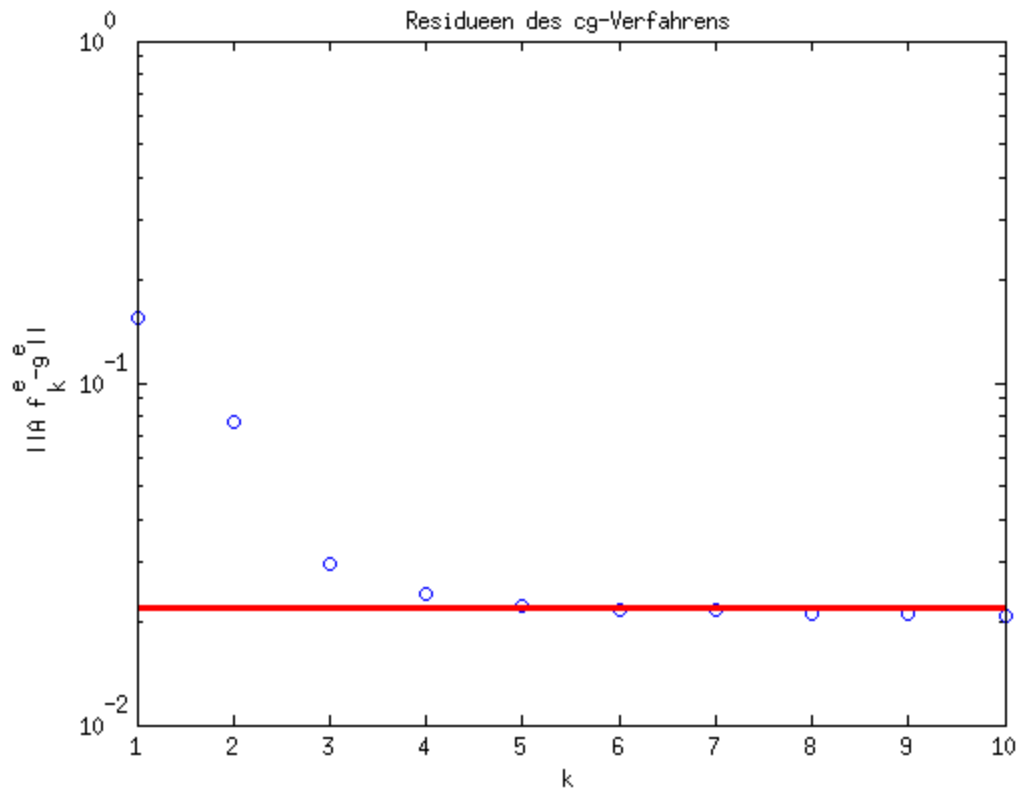
    % Verfahren der konjugierten Gradienten
    [f_t,res]=cgls(A,g_eps,10); % 10 Schritte
    pause
    %%
    % Plotte Residuen der cg-Iterierten;
    clf; set(gca,'fontsize',16);
```



Plotte Residuen der cg-Iterierten;

```
clf; set(gca, 'fontsize', 16);
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', 2.5);
hold off;
pause;

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', 2.5);
hold off;
pause;
%%
% Berechne rel. Fehler der cg-Rekonstruktionen
echo off;
```

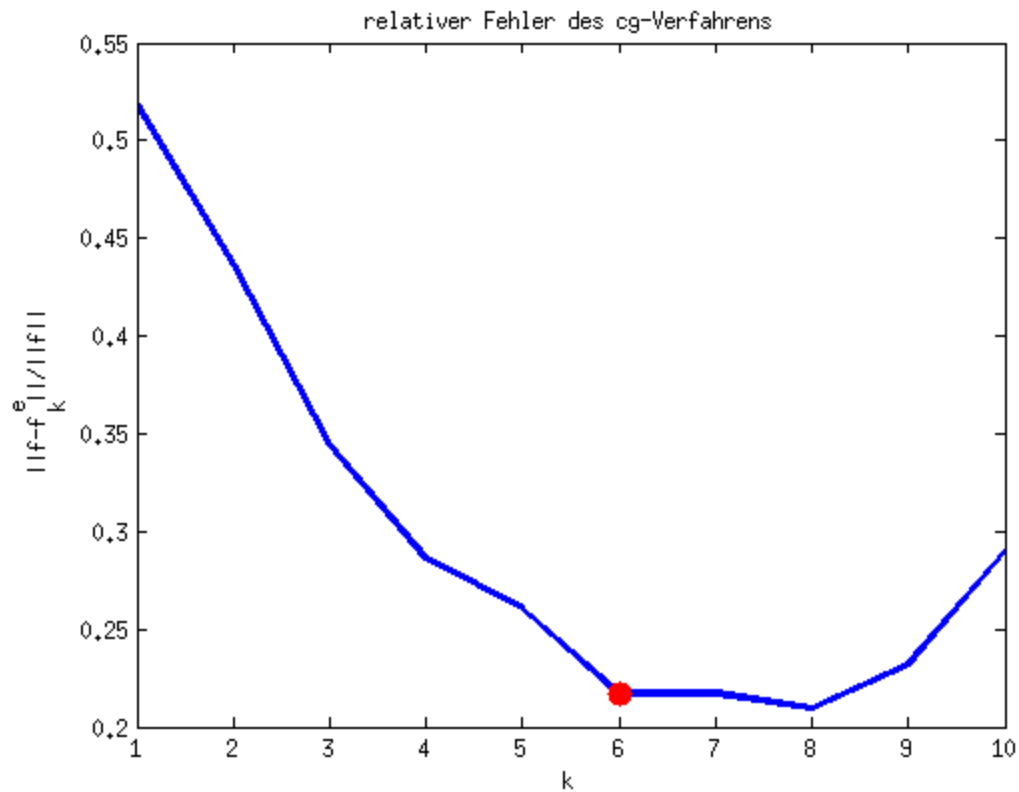


Berechne rel. Fehler der cg-Rekonstruktionen

```
echo off;
for i=1:10
    error(i)=norm(f-f_t(:,i))/norm(f);
end;
clf; set(gca, 'fontsize',16);
echo on;

plot(error, 'LineWidth',2.5);hold on;
plot(6,error(6), '.', 'Color','r', 'MarkerSize',30);
title('relativer Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||/||f||'); hold off;
pause;

plot(error, 'LineWidth',2.5);hold on;
plot(6,error(6), '.', 'Color','r', 'MarkerSize',30);
title('relativer Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||/||f||'); hold off;
pause;
%%
% Plote ein paar cg-Rekonstruktionen
plot(xt,f, 'LineWidth',2.5); hold on;
```



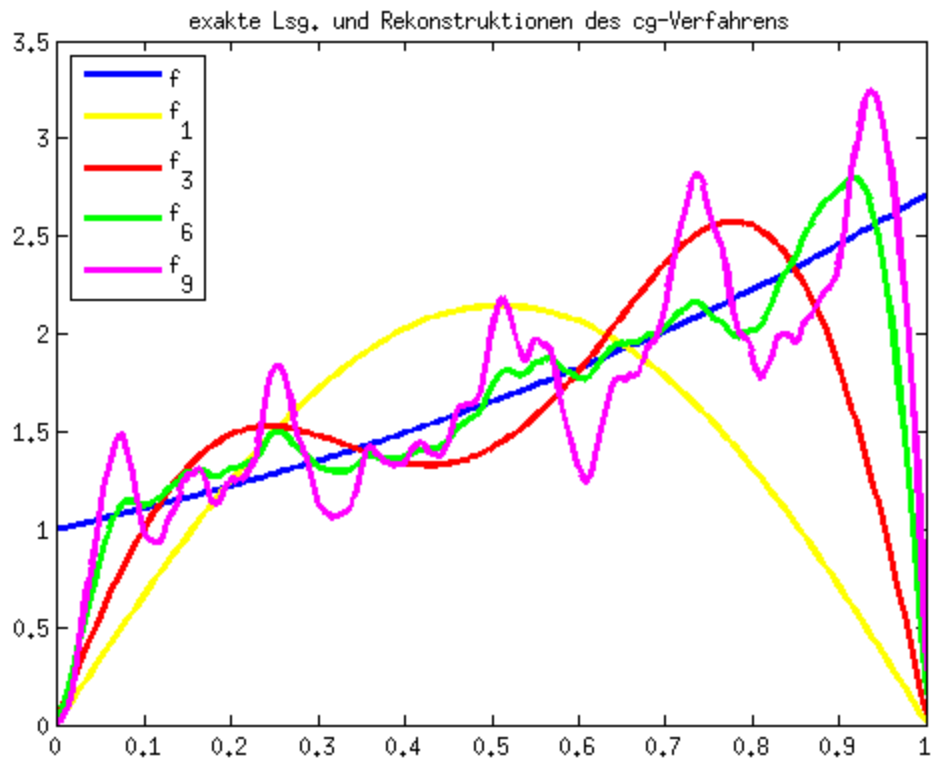
Plotte ein paar cg-Rekonstruktionen

```

plot(xt,f, 'LineWidth',2.5); hold on;
title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
plot(xt,f_t(:,1), 'Color', 'y', 'LineWidth',2.5);
plot(xt,f_t(:,3), 'Color', 'r', 'LineWidth',2.5);
plot(xt,f_t(:,6), 'Color', 'g', 'LineWidth',2.5);
plot(xt,f_t(:,9), 'Color', 'm', 'LineWidth',2.5);
legend('f', 'f_1', 'f_{3}', 'f_6', 'f_{9}',2); hold off;

    title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
    plot(xt,f_t(:,1), 'Color', 'y', 'LineWidth',2.5);
    plot(xt,f_t(:,3), 'Color', 'r', 'LineWidth',2.5);
    plot(xt,f_t(:,6), 'Color', 'g', 'LineWidth',2.5);
    plot(xt,f_t(:,9), 'Color', 'm', 'LineWidth',2.5);
    legend('f', 'f_1', 'f_{3}', 'f_6', 'f_{9}',2); hold off;
    %%
    % WAS IST DA PASSIERT ?????????????
    pause;

```

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

```
pause;
```

```
% Wie kann Abhilfe geschaffen werden ????
```

```
pause
```

```
% Wähle Anfangswert, der die Randwerte der  
% exakten Lsg. annimmt
```

```
pause;
```

```
clear f_t res error;
```

```
f_0 = (exp(1)-1)*linspace(0,1,200)+ones(1,200);
```

```
g_eps = g_eps - A*f_0';
```

```
pause;
```

```
% Verfahren der konjugierten Gradienten
```

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

```
pause
```

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

```
clf; set(gca,'fontsize',16);
```

```

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', 2.5);
hold off;
pause;

% Wie kann Abhilfe geschaffen werden ???
pause

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

clear f_t res error;

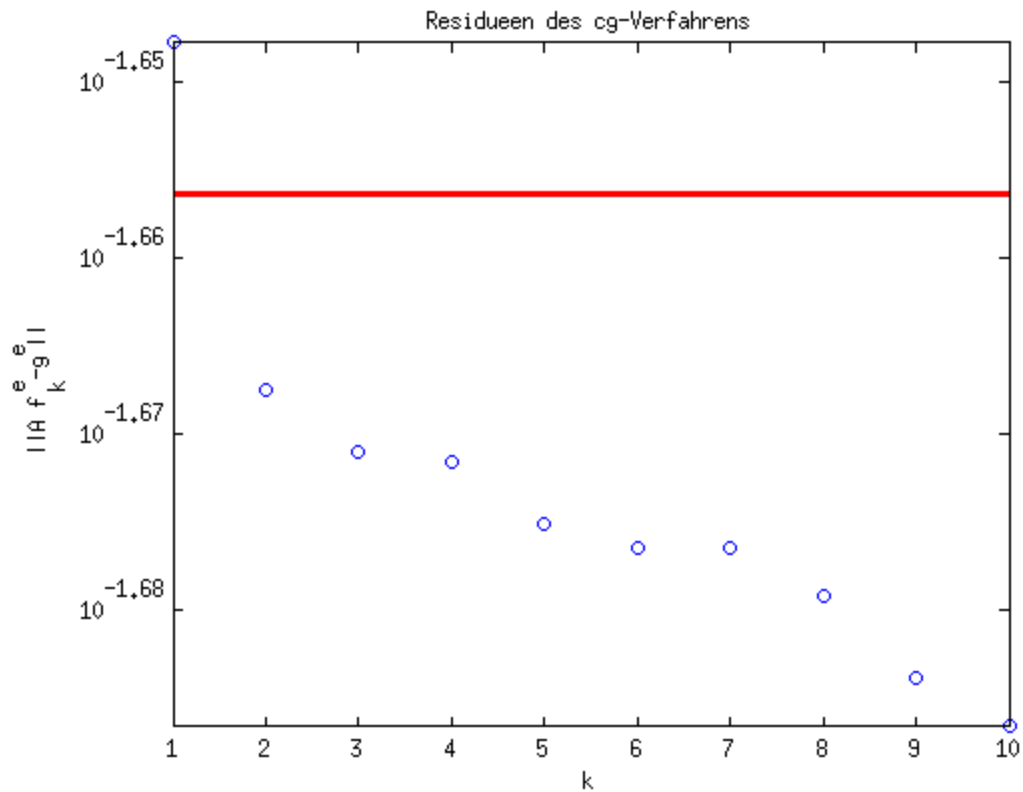
f_0 = (exp(1)-1)*linspace(0,1,200)+ones(1,200);

g_eps = g_eps - A*f_0';
pause;

% Verfahren der konjugierten Gradienten
[f_t,res]=cgls(A,g_eps,10); % 10 Schritte
pause

% Plotte Residuen der cg-Iterierten;
clf; set(gca, 'fontsize', 16);
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', 2.5);
hold off;
pause;
%%
% Berechne rel. Fehler der cg-Rekonstruktionen
echo off;

```

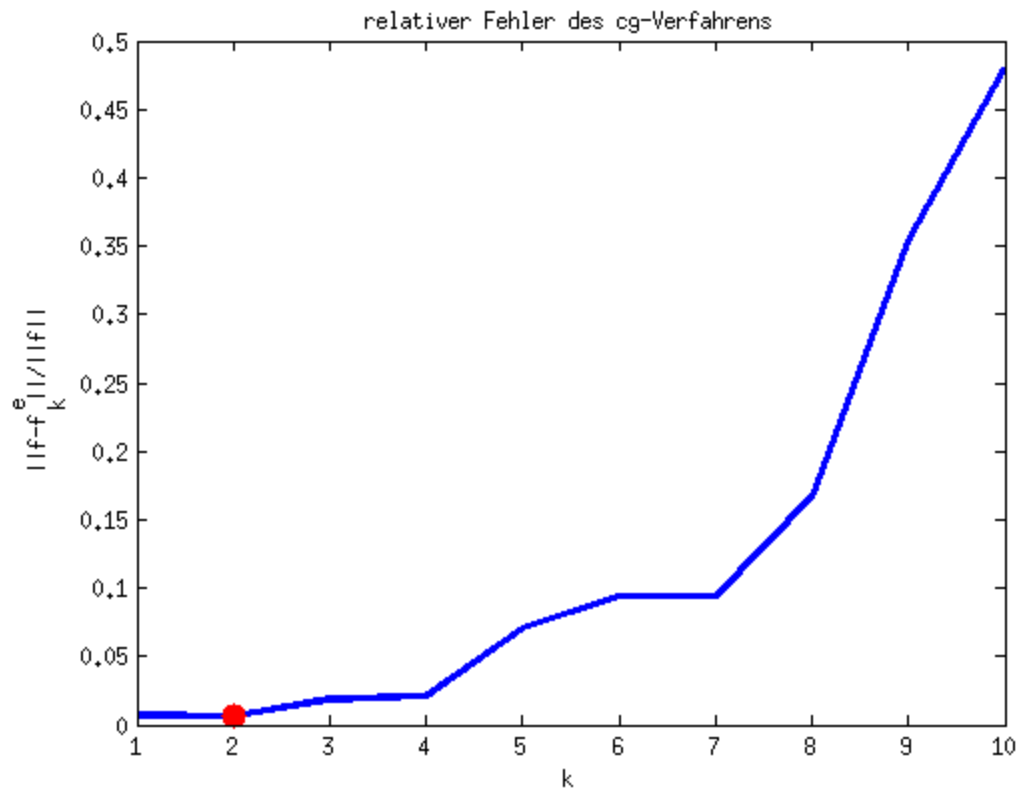


Berechne rel. Fehler der cg-Rekonstruktionen

```
echo off;
for i=1:10
    error(i)=norm(f-f_t(:,i)-f_0')/norm(f);
end;
clf; set(gca, 'fontsize',16);
echo on;

plot(error, 'LineWidth',2.5);hold on;
plot(2,error(2), '.', 'Color', 'r', 'MarkerSize',30);
title('relativer Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||/||f||'); hold off;
pause;

plot(error, 'LineWidth',2.5);hold on;
plot(2,error(2), '.', 'Color', 'r', 'MarkerSize',30);
title('relativer Fehler des cg-Verfahrens'); xlabel('k');
ylabel('||f-f_k^\epsilon||/||f||'); hold off;
pause;
%%
% Plote ein paar cg-Rekonstruktionen
plot(xt,f, 'LineWidth',2.5); hold on;
```



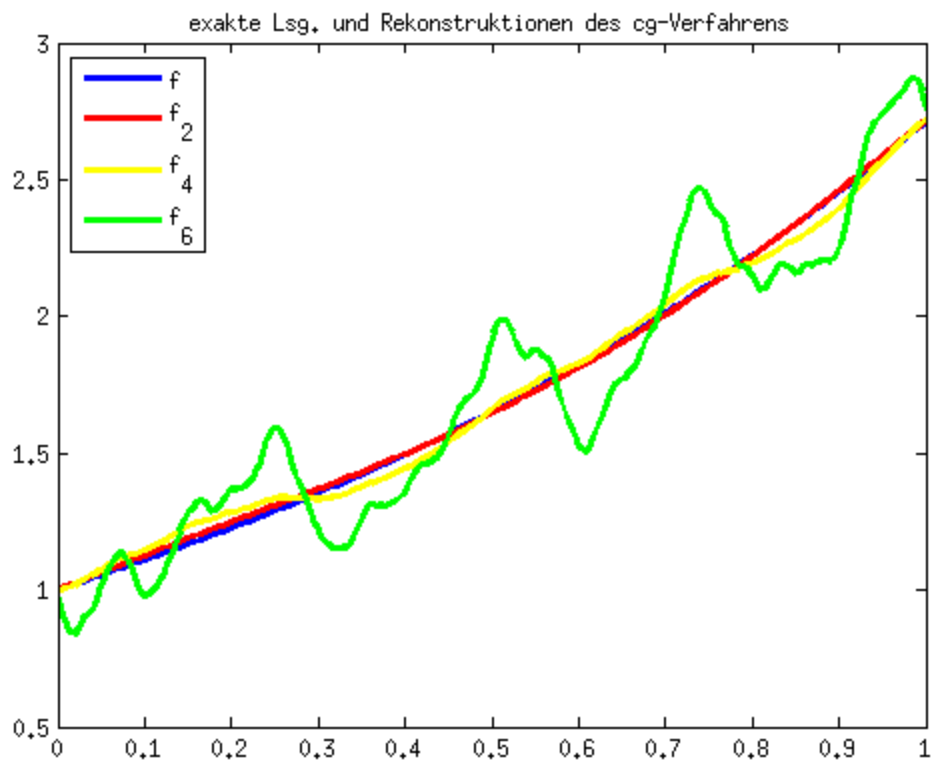
Plotte ein paar cg-Rekonstruktionen

```

plot(xt,f, 'LineWidth',2.5); hold on;
title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
plot(xt,f_t(:,2)+f_0, 'Color','r', 'LineWidth',2.5);
plot(xt,f_t(:,4)+f_0, 'Color','y', 'LineWidth',2.5);
plot(xt,f_t(:,6)+f_0, 'Color','g', 'LineWidth',2.5);
legend('f', 'f_2', 'f_{4}', 'f_6',2); hold off;

title('exakte Lsg. und Rekonstruktionen des cg-Verfahrens');
plot(xt,f_t(:,2)+f_0, 'Color','r', 'LineWidth',2.5);
plot(xt,f_t(:,4)+f_0, 'Color','y', 'LineWidth',2.5);
plot(xt,f_t(:,6)+f_0, 'Color','g', 'LineWidth',2.5);
legend('f', 'f_2', 'f_{4}', 'f_6',2); hold off;

```



Published with MATLAB® 7.14