

```

%Berechnungsmethoden der Energie- und Verfahrenstechnik
%           FS 2010
% Serie 1
% Marco Weber
% _____

close all
clear all
clc
%Aufgabe 1
%a)

x_0=0;
delta_x(1)=0;

for k=2:31
    delta_x(k)=0.5^k;
end

%funktion 1
delta_f_1=(sin(sqrt(2).*(x_0+delta_x))-sin(sqrt(2)*x_0))./(delta_x);

%funktion 2
delta_f_2=(exp(sqrt(2).*(x_0+delta_x))-exp(sqrt(2)*x_0))./(delta_x);

% _____
%b)

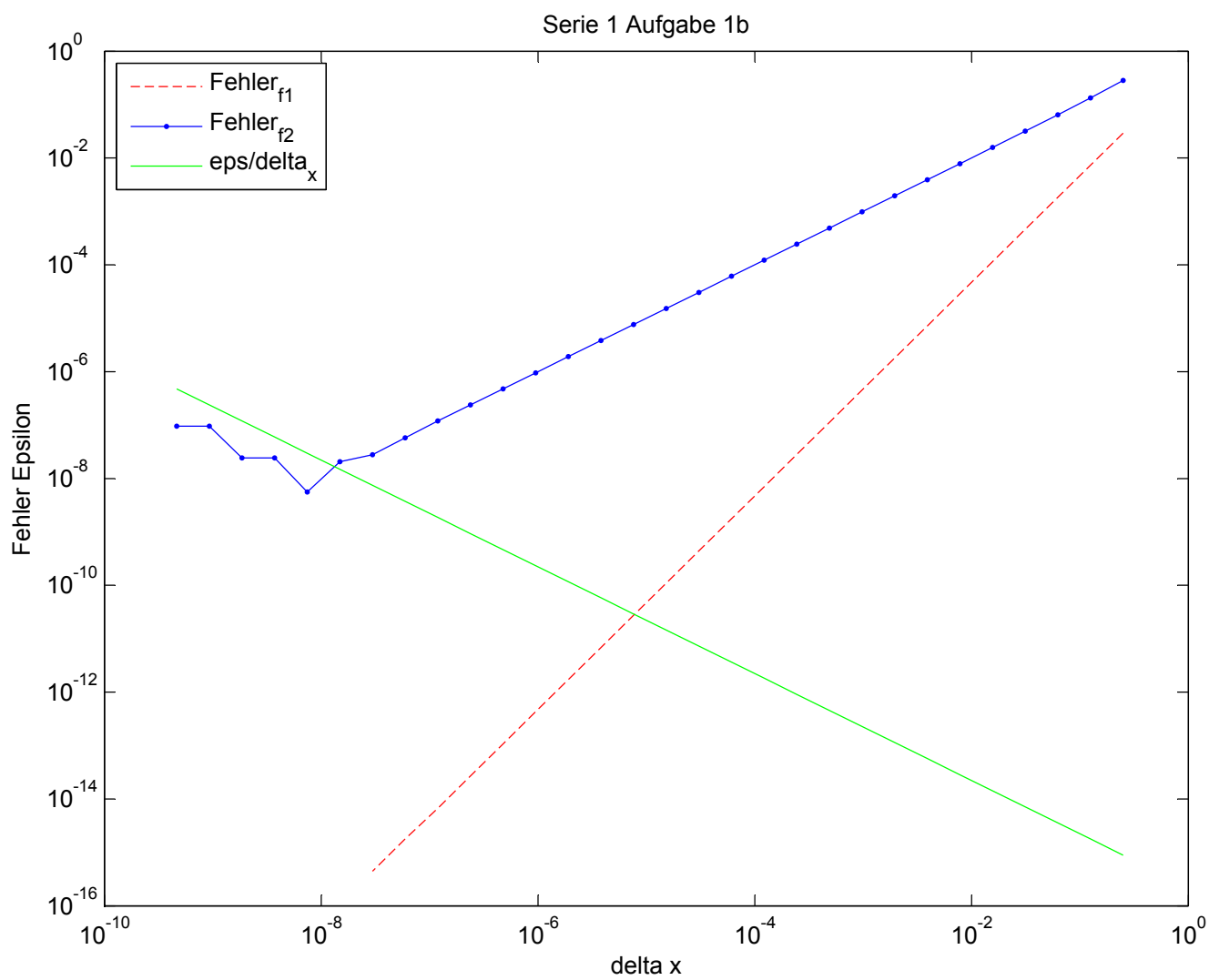
%exakte Ableitungen:
exa_f_1=cos(sqrt(2)*x_0)*sqrt(2);
exa_f_2=exp(sqrt(2)*x_0)*sqrt(2);

%Fehlerfunktionen
epsi_1=abs(delta_f_1-exa_f_1);
epsi_2=abs(delta_f_2-exa_f_2);

%Maschinengenauigkeit
XI=eps./delta_x;

figure(1)
loglog(delta_x,epsi_1,'--r')
hold on
loglog(delta_x,epsi_2,'.-b')
loglog(delta_x,XI,'g')
hold off
legend('Fehler_f_1','Fehler_f_2','eps/delta_x','Location','Northwest')
title('Serie 1 Aufgabe 1b')
xlabel('delta x')
ylabel('Fehler Epsilon')
% _____
%c)

```



```
%Berechnungsmethoden der Energie- und Verfahrenstechnik
```

```
% FS 2010
```

```
% Serie 1
```

```
% Marco Weber
```

```
%
```

```
close all
```

```
clear all
```

```
clc
```

```
%Aufgabe 2
```

```
N=20;
```

```
K=1:N/2;
```

```
delta_x=2*pi/N;
```

```
j=1:N;
```

```
x=delta_x.*j;
```

```
%a)
```

```
%funktion f
```

```
f=3+4*cos(2.*x-3)';
```

```
%Matrix F
```

```
for z=1:20
```

```
for s=1:N
```

```
F_m(z,s)= 1/N*exp(-i*(z-10)*x(s));
```

```
end
```

```
end
```

```
%fourier:
```

```
four=F_m*f;
```

```
%LÖSUNG:
```

```
%
```

```
%
```

```
% four =
```

```
%
```

```
% 0.0000 + 0.0000i
```

```
% 0 - 0.0000i
```

```
% -0.0000 + 0.0000i
```

```
% 0.0000 - 0.0000i
```

```
% -0.0000 + 0.0000i
```

```
% 0.0000 - 0.0000i
```

```
% -0.0000 + 0.0000i
```

```
% -1.9800 + 0.2822i
```

```
% 0.0000 + 0.0000i
```

```
% 3.0000
```

```
% 0.0000 - 0.0000i
```

```
% -1.9800 - 0.2822i
```

```
% -0.0000 - 0.0000i
```

```
% 0.0000 + 0.0000i
```

```
% -0.0000 - 0.0000i
```

```
%      0.0000 + 0.0000i
%     -0.0000 - 0.0000i
%           0 + 0.0000i
%      0.0000 - 0.0000i
%     -0.0000 + 0.0000i
```

```
% _____
%b)
```

```
clear s
clear z
```

```
%F_k vektoren gestapelt = G
```

```
for z=1:20
```

```
for s=1:N
```

```
G(z,s)= exp(i*(z-10)*x(s));
```

```
end
```

```
end
```

```
%um g zu finden müsse man  $G^{-1} \cdot f = g$  machen... die Inverse ist sehr
```

```
%rechenintensiv
```

```
g=inv(G)*f
```

```
%LÖSUNG:
```

```
% _____
```

```
% g =
```

```
%
%      0.0000 + 0.0000i
%     -1.9800 - 0.2822i
%      0.0000 + 0.0000i
%     -0.0000 + 0.0000i
%      0.0000 + 0.0000i
%      0.0000 - 0.0000i
%      0.0000 - 0.0000i
%     -0.0000 + 0.0000i
%      0.0000 + 0.0000i
%     -0.0000 + 0.0000i
%     -0.0000 - 0.0000i
%     -0.0000 - 0.0000i
%     -0.0000 + 0.0000i
%      0.0000 + 0.0000i
%     -0.0000 + 0.0000i
%      0.0000 - 0.0000i
%      0.0000 - 0.0000i
%     -1.9800 + 0.2822i
%     -0.0000 + 0.0000i
%      3.0000 - 0.0000i
```

```
% _____
%c)
```

```
k=-9:10;
```

```
figure(1)
title('Serie1 Aufgabe 2 c')
hold on
plot(k,abs(four),'b') %Betrag
plot(k,real(four),'r')
plot(k,imag(four),'m')
plot(k,abs(g),'c')
plot(k,real(g),'k')
plot(k,imag(g),'g')
legend('Betrag(f)', 'Re(f)', 'Im(f)', 'Betrag(g)', 'Re(g)', 'Im(g)', 'Location', 'SouthOutside')
xlabel('k-Achse')
```

```
%
%_____
%d)
```

```
P=F_m*G;
figure(2)
title('Vergleich G & F_m')
subplot(2,1,1)
plot(G)
title('G')
```

```
subplot(2,1,2)
plot(F_m)
title('F_m')
```

```
%Es sind sehr ähnliche Matrixen, da sie sich nur um einen Vorzeichenwechsel
%im Exponent und einem Vorfaktor unterscheiden. (Siehe plots)
%Das Produkt hat sehr viele NULL Einträge.
```

