

```
% Demoprogram for bruk i FYS2130 januar 2009.
% Skrevet av Arnt Inge Vistnes
% Programmet er rotete fordi det inneholder et vell av variasjonsmuligheter
% som kan egne seg ved utforsking av FFT, IFFT og digital filtrering.
```

```
function lydfilavspilling09b
```

```
m = 'Sound (WAV) file';
c = 'lydfil2.wav'; % Velg i så fall nstart = 8000;
%c = 'lydfil4.wav'; % Velg i så fall nstart = 8000;
%c = 'lydfil6.wav'; % Velg i så fall nstart = 60000;
%c = 'solo2.wav'; % Velg i så fall nstart = 22000;
[m d] = wavfinfo(c);
% d % For utskrift om ønskelig
Fs = 44100; % Samplingsfrekvens (CD kvalitet)
type = 'int16'; % Ikke nødvendig?
N = 1024*16; % Velger et tall som er av typen 2^n for å få fft til
% å gå raskt!
nstart = 8000; % Velger i de tre neste linjene utsnitt i lydfil
nslutt = nstart+N-1;
[y, Fs, type] = wavread(c, [nstart nslutt]);

% Total tid lydutsnittet tar (i sek)
T = N/Fs % Sløyfer her semikolon for å få ut en testutskrift
% Diverse arrays for x-akse i plot m.m.
t = linspace(0, T*(N-1)/N, N);
g = zeros(N, 1);
delta_f = 1/T;
f_max = N/T
f = linspace(0, f_max*(N-1)/N, N);

% Plukker ut lydsignal som skal Fourier omvendes
g = y(:, 1); % MERK: y som vi får fra wavread kan være av typen y(N, 2)
% dersom lydfile er i stereo. Denne programlinjen plukker
% ut bare en av kanalene slik at vi får et mono-signal i
% fortsettelsen.

% Spiller av utsnittet som er brukt
wavplay(g, Fs); % NB: Velg sound(g, Fs); dersom du er på en Linux-maskin!

% Plotter opprinnelig tidsfunksjon
plot(t, g, 'k');
%xlim([-0.2, 6.5]); % Kan begrense plottet dersom man ønsker det, men
%ylim([-1.2, 1.2]); % pass på at du ikke mister oversikten!
title('Opprinnelig tids-funksjon');
xlabel('Tid (sek)');
ylabel('Relativt signal');
figure;

% Fourier transform gjøres her
G = fft(g, N)/N;
```

```

% Plot absoluttverdier og vinkler
m = abs(G);
M = N; % Øvre grense for plot (hvor stor del som skal tas med)
plot(f(1:M),2*m(1:M))
title('Fourier-spekteret (amplitude for hver frekvens)');
xlabel('Frekvens (Hz)');
ylabel('Relativt styrke');
%figure;
%p = unwrap(angle(G)); % Beregning av fase
%plot(f(1:M),p(1:M)); % Plotting av fase (problematisk, pga 2pi grense)
%title('Faseplot');
%xlabel('Frekvens (Hz)');
%ylabel('Relativt fase (radianer)');

%Eller plot reell og imaginær del av frekvensspekteret
an = zeros(N,1);
bn = zeros(N,1);
an = real(G);
bn = imag(G);
%plot(f,an, '-r');
%xlim([-0.2,6.5]);
%hold on;
%plot(f,bn, '-b');
%title('Fourier-komponenter (cos-ledd: røde, sin-ledd: blå)');
%xlabel('Frekvens (Hz)');
%ylabel('Relativt signal');

%*****

pause; % Gir tid for kommentarer før lavpassfilter anvendes og IFFT foretas

% Kan nå legge inn et filter i frekvensdomenet, f.eks. et lavpassfilter
% Høyeste frekvens som skal slippe gjennom:
f_max = 1500.0;
% Beregner tilsvarende posisjon i spekter
nl = round((f_max / Fs) * N)
nh = N - nl
for i=nl:nh
    G(i) = 0.0;
end;
figure; % Plotter filtrert spekter (fullt område!)
Gx = abs(G);
plot(f(1:M),2*Gx(1:M))
title('Fourier-spekteret (filtrert)');
xlabel('Frekvens (Hz)');
ylabel('Relativt styrke');

% Invers Fourier transform gjøres her (NB HELE frekvensspekteret!!!)
% Komplekse tall.
g2 = ifft(G,N);

```

```

g3 = real(g2)*N;
% Plotter opprinnelig tidsfunksjon
figure;
plot(t,g3,'-k');
%xlim([-0.2,6.5]); % Kan begrense plottet dersom man ønsker det, men
%ylim([-1.2,1.2]); % pass på at du ikke mister oversikten!
title('Tilbaketransformert tids-funksjon');
xlabel('Tid (sek)');
ylabel('Relativt signal');
wavplay(g3,Fs); % NB: Velg sound(g3,Fs); dersom du er på en Linux-maskin!

%*****

pause; % Gir tid for kommentarer før høypassfilter og IFFT foretas

% Kan nå legge inn et filter i frekvensdomenet, f.eks. et høypassfilter
% Laveste frekvens som skal slippe gjennom:
f_min = 1200.0;
% Beregner tilsvarende posisjon i spekter
nl = round((f_min / Fs ) * N)
nh = N - nl
for i=1:nl
    G(i) = 0.0; % Merk: Bruker frekvensspekteret etter lavpassfiltrering
end;
for i=nh:N
    G(i) = 0.0;
end;
figure; % Plotter filtrert spekter (for fullt område bytt M med N)
Gx = abs(G);
plot(f(1:M),2*Gx(1:M))
title('Fourier-spekteret (dobbelt filtrert)');
xlabel('Frekvens (Hz)');
ylabel('Relativt styrke');

% Ny invers Fourier transform gjøres her
g2 = ifft(G,N);

g3 = real(g2)*N;
% Plotter opprinnelig tidsfunksjon
figure;
plot(t,g3,'-k');
%xlim([-0.2,6.5]); % Kan begrense plottet dersom man ønsker det, men
%ylim([-1.2,1.2]); % pass på at du ikke mister oversikten!
title('Tilbaketransformert tids-funksjon etter dobbelt filtrering');
xlabel('Tid (sek)');
ylabel('Relativt signal');
wavplay(g3,Fs); % NB: Velg sound(g3,Fs); dersom du er på en Linux-maskin!

```