Synthèse d'une sinusoïde de fréquence 100 Hz

```
clc;
figure;
subplot(1, 1, 1);
```

Computation of the sinewave and display

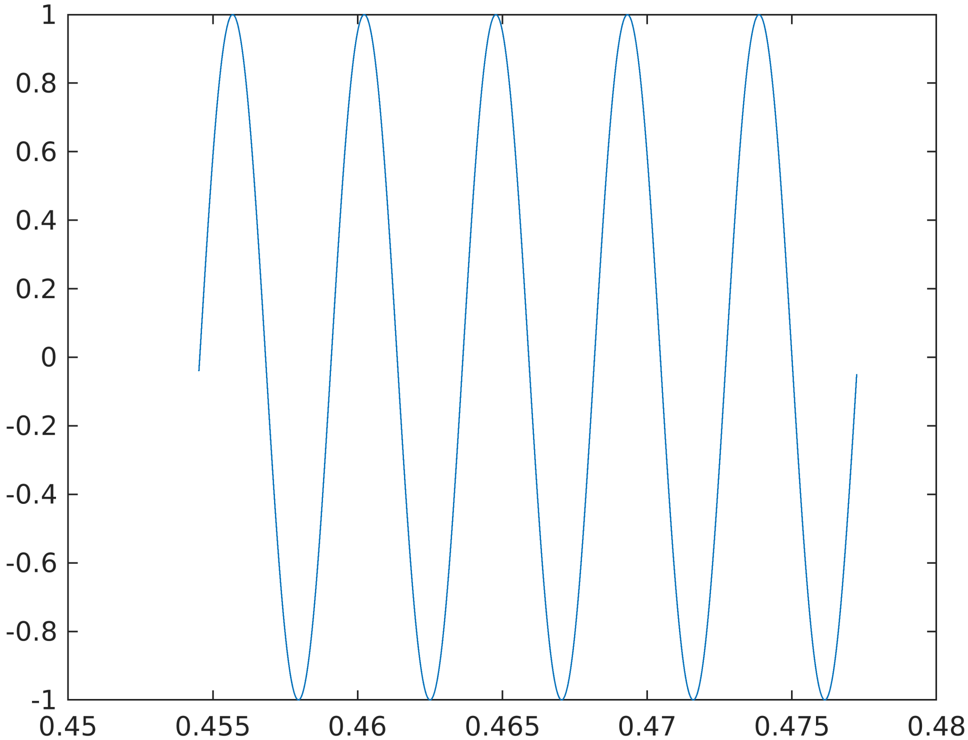
```
fs = 44100; % Sampling Freq.
dur = 3; % Sound duration in s.
t = linspace(0, dur, dur*fs);
freq = 220;
amp = 1;
phase = 0;

snd = amp .* sin(2*pi*freq.*t + phase);
t = t';
snd = snd';
% Ramp the sound extremities
snd = zeroends(snd, fs);

plot(t, snd);
```

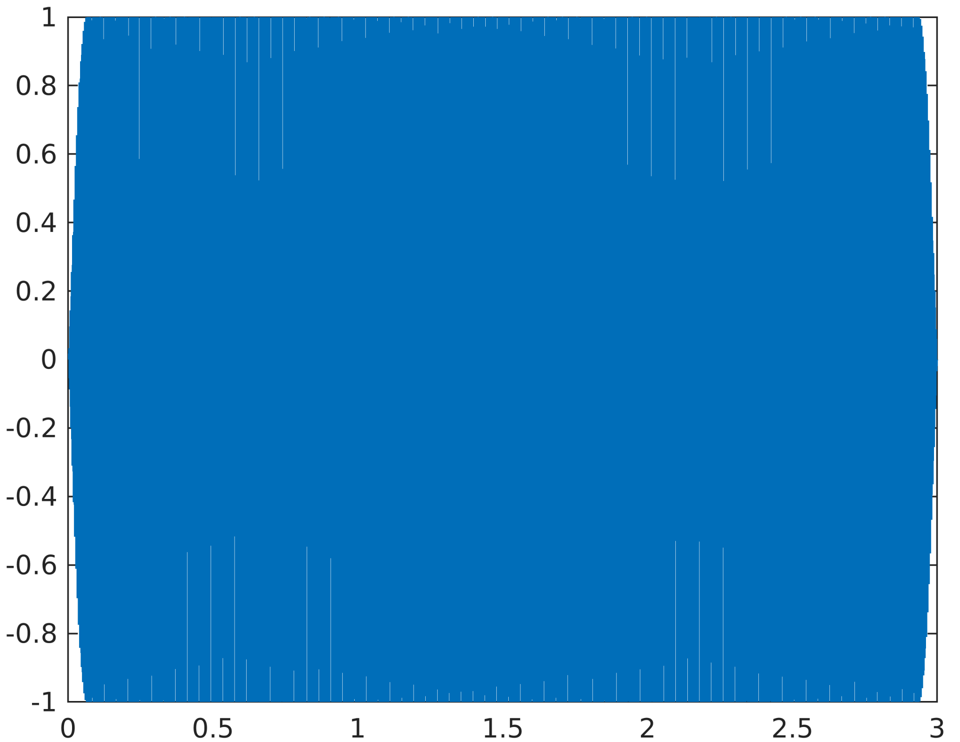
Display of a subset (N periods) that is outside the ramping parts

```
nperiods = 5;
start = 100;
select = round((start*((fs/freq))):((start+nperiods)*((fs/freq))));

figure;
subplot(1, 1, 1);
plot(t(select), snd(select));
```

Play and save the sound wave

```
%soundsc(snd, fs);
%pause(dur);

listen = audioplayer(snd, fs);
playblocking(listen);
```


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0.45
0.455
0.46
0.465
0.47
0.475
0.48

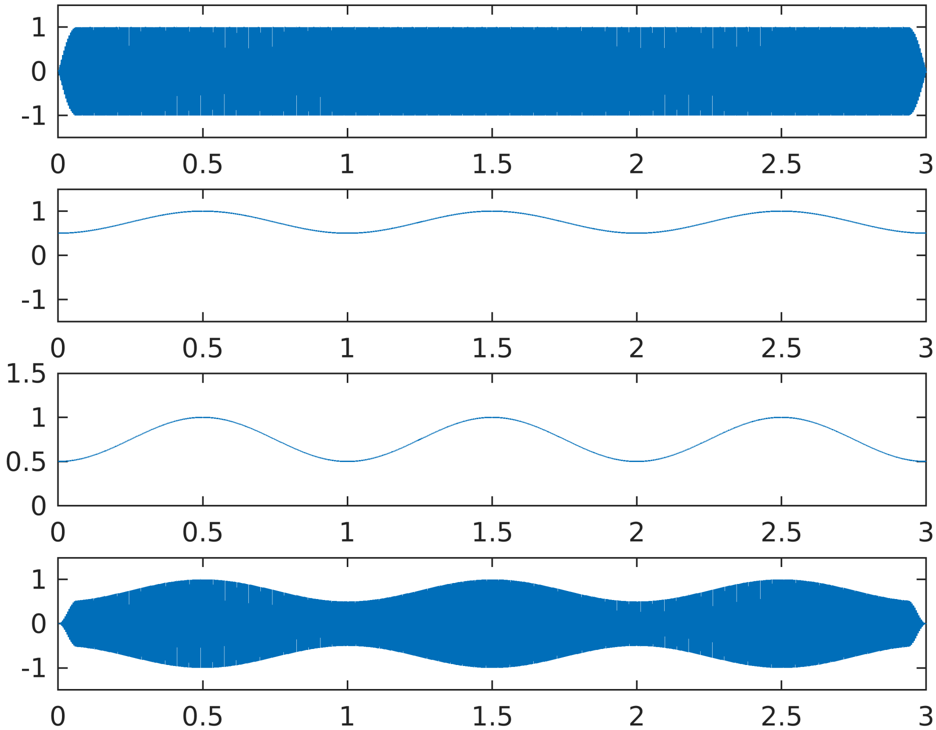

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.5
1
1.5
2
2.5
3

Generate an amplitude modulation: 1 Hertz (1 cycle par seconde, Sinewave AM)

```
AMf1 = 1; % Amplitude modulation frequency in Hertz
AMdepth1 = 0.5; % AM depth as a proportion of the carrier amplitude
AMphase1 = 1*pi;
mod = 1/2*AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1);

% Compute effective modulation depending on depth
mod = mod + (1 - max(mod));

final = mod .* snd;
final = zeroends(final, fs);
%final = modulate(snd, AMf, fs, 'am');
```

Affichage

```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, mod);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])

soundsc(final, fs);
pause(dur);
```



Restriction à une période de l'enveloppe

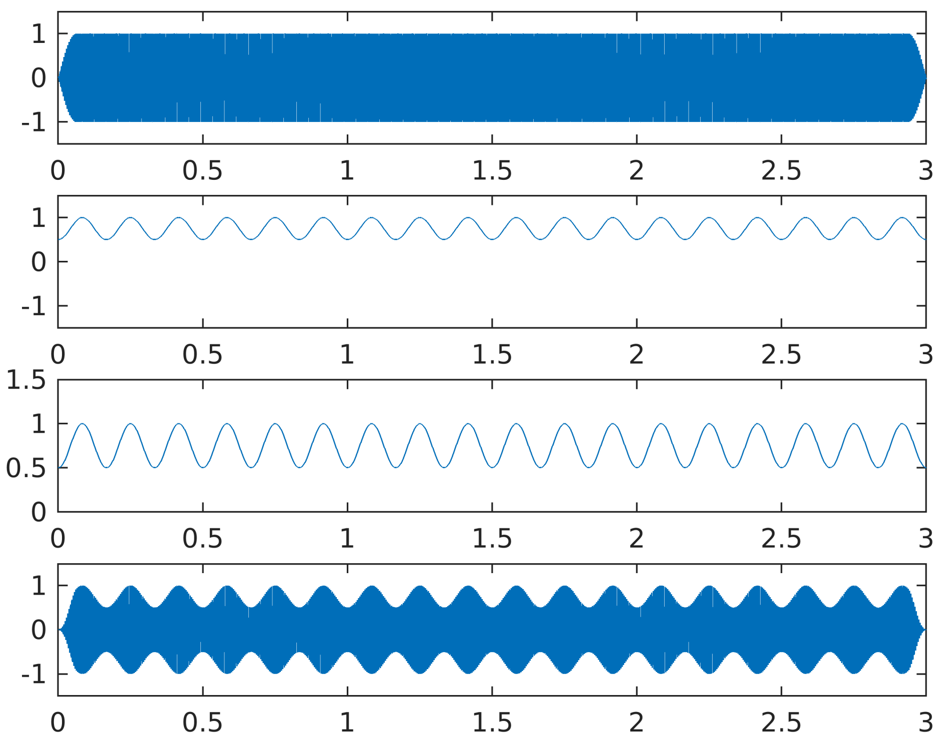
```
extract = final(1:(AMf1*fs));
soundsc(extract, fs);
pause(dur+0.5);
```

Generate an amplitude modulation: 6 Hertz (6 cycles par seconde, Sinewave AM)

```
AMf1 = 6; % Amplitude modulation frequency in Hertz
AMdepth1 = 0.5; % AM depth as a proportion of the carrier amplitude
AMphase1 = 1*pi;
mod = 1/2*AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1);

% Compute effective modulation depending on depth
mod = mod + (1 - max(mod));

final = mod .* snd;
final = zeroends(final, fs);
%final = modulate(snd, AMf, fs, 'am');
```

Affichage

```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, mod);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])

soundsc(final, fs);
pause(dur);
```



Restriction à une période de l'enveloppe

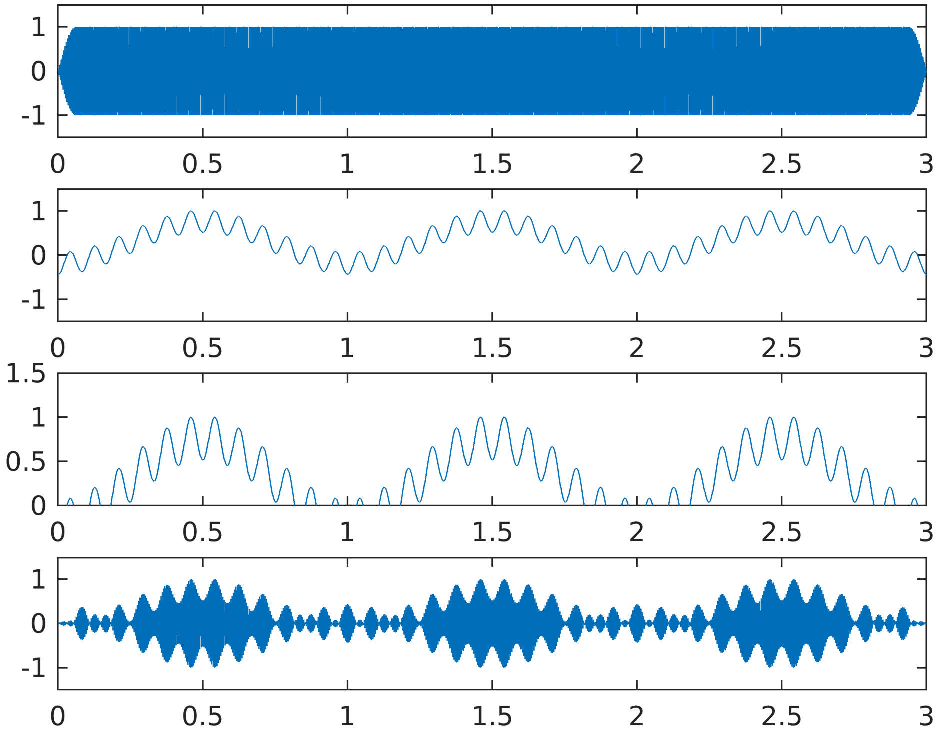
```
extract = final(1:(1/AMf1*fs));
soundsc(extract, fs);
pause(dur+0.5);
```

Generate a multiple frequency amplitude modulation: 1 and 12 Hertz (1 and 12 cycles par seconde, Sinewave AM)

```
AMf1 = 1; % Amplitude modulation frequency in Hertz
AMdepth1 = 0.95; % AM depth as a proportion of the carrier amplitude
AMphase1 = 1*pi;
AMf2 = 12; % Amplitude modulation frequency in Hertz
AMdepth2 = 0.5; % AM depth as a proportion of the carrier amplitude
AMphase2 = 1*pi;
mod = 1/2*AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1) + 1/2*AMdepth2 .* cos(2*pi*AMf2.*t +

% Compute effective modulation depending on depth
mod = mod + (1 - max(mod));

final = mod .* snd;
final = zeroends(final, fs);
%final = modulate(snd, AMf, fs, 'am');
```

Affichage

```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, mod);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])

soundsc(final, fs);
pause(dur);
```

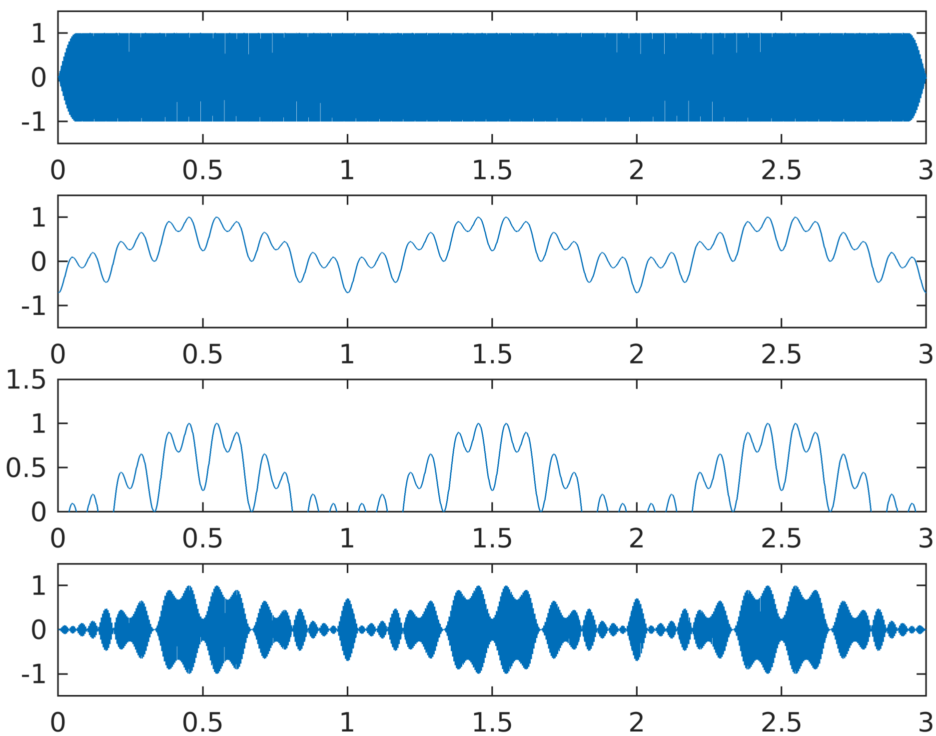


Generate a multiple frequency amplitude modulation: 1, 3 and 12 Hertz (1 and 12 cycles par seconde, Sinewave AM)

```
AMf1 = 1; % Amplitude modulation frequency in Hertz
AMdepth1 = 0.95; % AM depth as a proportion of the carrier amplitude
AMphase1 = 1*pi;
AMf2 = 12; % Amplitude modulation frequency in Hertz
AMdepth2 = 0.5; % AM depth as a proportion of the carrier amplitude
AMphase2 = 1*pi;
AMf3 = 6; % Amplitude modulation frequency in Hertz
AMdepth3 = 0.5; % AM depth as a proportion of the carrier amplitude
AMphase3 = 1*pi;
mod = 1/2*AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1) + 1/2*AMdepth2 .* cos(2*pi*AMf2.*t +

% Compute effective modulation depending on depth
mod = mod + (1 - max(mod));

final = mod .* snd;
final = zeroends(final, fs);
%final = modulate(snd, AMf, fs, 'am');
```

Affichage

```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, mod);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])

soundsc(final, fs);
pause(dur);
```



Generate an amplitude modulation: 1 Hertz (1 cycle par seconde), Half-Wave Rectified AM

```
AMf1 = 1; % Amplitude modulation frequency in Hertz
AMdepth1 = 0.5; % AM depth as a proportion of the carrier amplitude
AMphase1 = 1*pi;
mod = AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1);

% Half-Wave Rectification (1 cycle = 2 peaks)
modHWR = abs(mod)./max(mod).*AMdepth1;

% Compute effective modulation depending on depth
modHWR = modHWR + (1 - max(modHWR));

final = modHWR .* snd;
%final = modulate(snd, AMf, fs, 'am');

soundsc(final, fs);
pause(dur);
```

Affichage

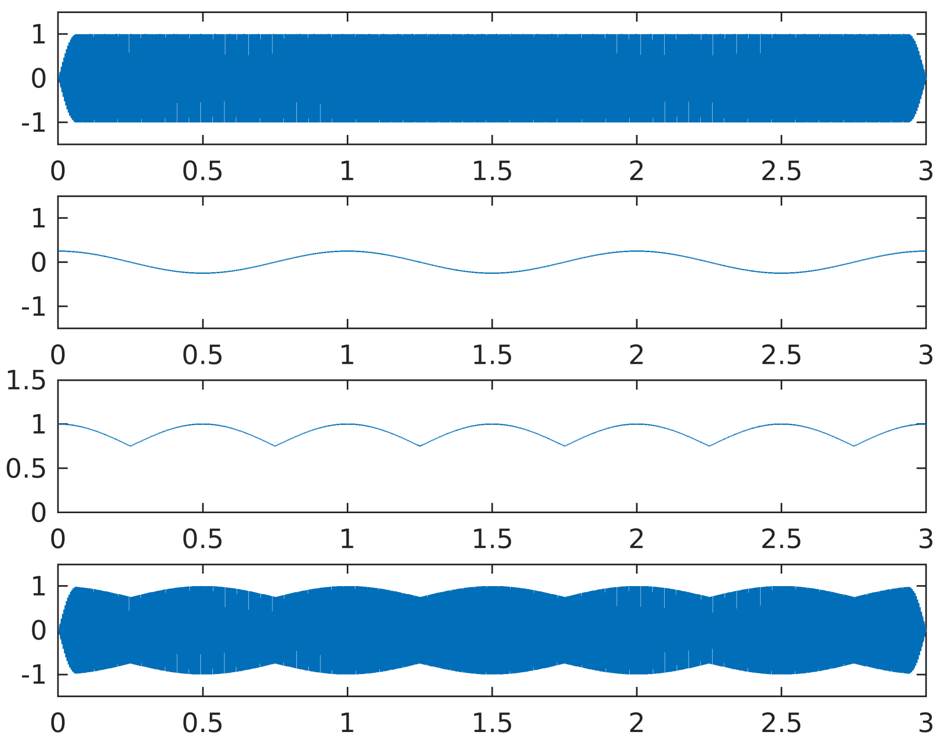
```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, modHWR);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])
```

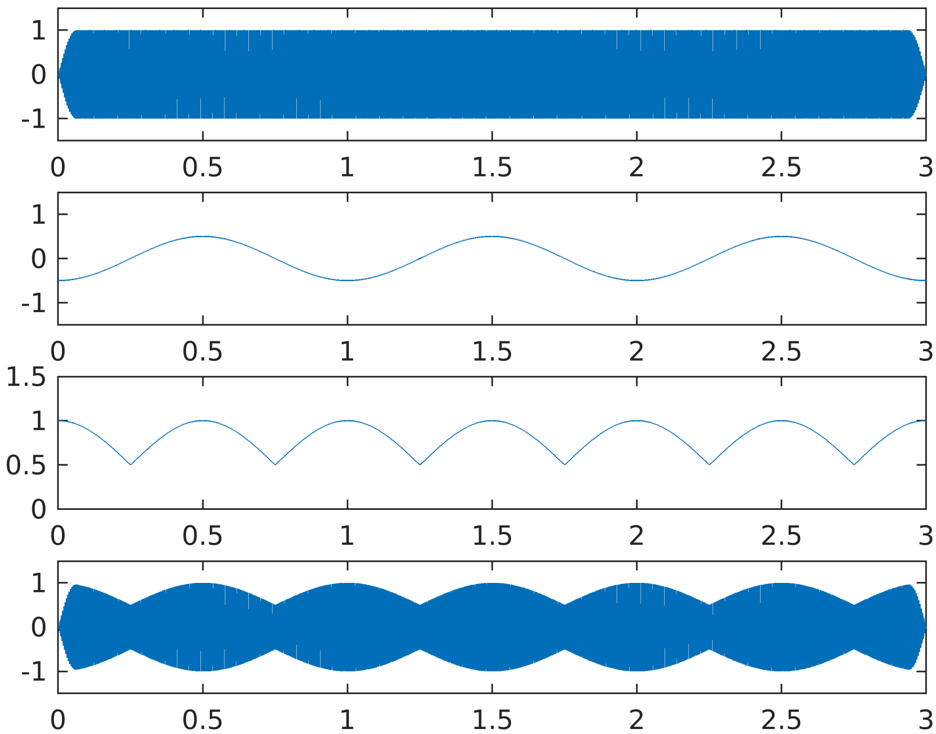
Generate an amplitude modulation: 1 Hertz (1 cycle par seconde), Profondeur de modulation 25%

```
AMf1 = 1; % Amplitude modulation frequency in Hertz
AMdepth1 = 0.25; % AM depth as a proportion of the carrier amplitude
AMphase1 = 0*pi;
mod = AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1);

% Half-Wave Rectification (1 cycle = 2 peaks)
modHWR = abs(mod)./max(mod).*AMdepth1;

% Compute effective modulation depending on depth
modHWR = modHWR + (1 - max(modHWR));

final = modHWR .* snd;
%final = modulate(snd, AMf, fs, 'am');
```

Affichage

```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, modHWR);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])

soundsc(final, fs);
pause(dur);
```





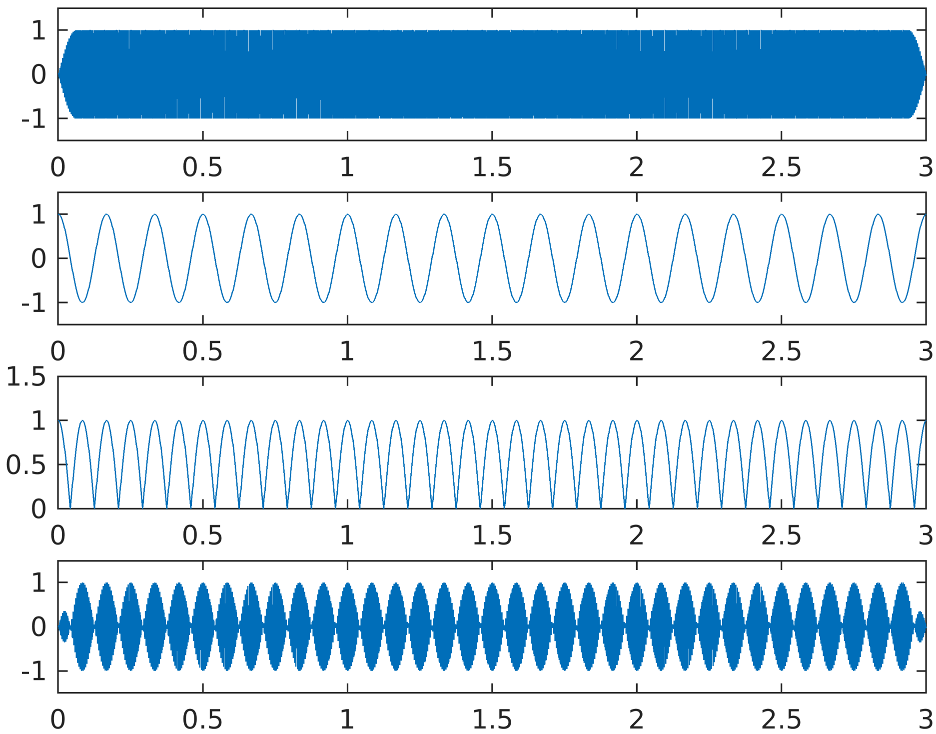
Generate an amplitude modulation: 6 Hertz (6 cycles par seconde)

```
AMf1 = 6; % Amplitude modulation frequency in Hertz
AMdepth1 = 1; % AM depth as a proportion of the carrier amplitude
AMphase1 = 0*pi;
mod = AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1);

% Half-Wave Rectification (1 cycle = 2 peaks)
modHWR = abs(mod)./max(mod).*AMdepth1;

% Compute effective modulation depending on depth
modHWR = modHWR + (1 - max(modHWR));

final = modHWR .* snd;
%final = modulate(snd, AMf, fs, 'am');
```

Affichage

```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
```

```
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, modHWR);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])

soundsc(final, fs);
pause(dur);
```



Generate an amplitude modulation: 0.1 Hertz (0.1 cycle par seconde)

```
AMf1 = 0.1; % Amplitude modulation frequency in Hertz
AMdepth1 = 0.5; % AM depth as a proportion of the carrier amplitude
AMphase1 = 0*pi;
AMamp = AMdepth1 * amp;
mod = AMdepth1 .* cos(2*pi*AMf1.*t + AMphase1);

% Half-Wave Rectification (1 cycle = 2 peaks)
modHWR = abs(mod)./max(mod).*AMdepth1;

% Compute effective modulation depending on depth
modHWR = modHWR + (1 - max(modHWR));

final = modHWR .* snd;
```

```
%final = modulate(snd, AMf, fs, 'am');
```

Affichage

```
figure();
subplot(4, 1, 1);
plot(t, snd);
ylim([-1.5, 1.5])
subplot(4, 1, 2);
plot(t, mod);
ylim([-1.5, 1.5])
subplot(4, 1, 3);
plot(t, modHWR);
ylim([0, 1.5])
subplot(4, 1, 4);
plot(t, final);
ylim([-1.5, 1.5])

soundsc(final, fs);
pause(dur);
```