Initialisation

On assume 3 signaux RF amp1, amp2 et amp3 entrant dans un amplificateur, tel que :

```
amp1 = -50; % dBm
amp2 = -30; % dBm
amp3 = -90:0.1:10; % dBm
```

Dans cet exemple, amp1 et amp2 sont des signaux d'amplitude fixe et bas niveau, tandis que amp3 varie d'un faible niveau à un fort niveau au-delà de la saturation de l'amplificateur.

L'amplificateur est décrit principalement par son gain petit signal et son point de compression à 1 dB :

```
gain_transducique = 20; % dB
p1dB = 15; % dBm
```

Son comportement en compression est décrit par un coefficient de transition entre le régime de gain linéaire et saturé, et un recul petit de la compression petit signal (compression des signaux de faible amplitude lorsqu'un signal de forte amplitude comprime l'amplificateur) est ajouté :

```
smo = sqrt(10); % coef de coude
recul_max = 10; % dB
```

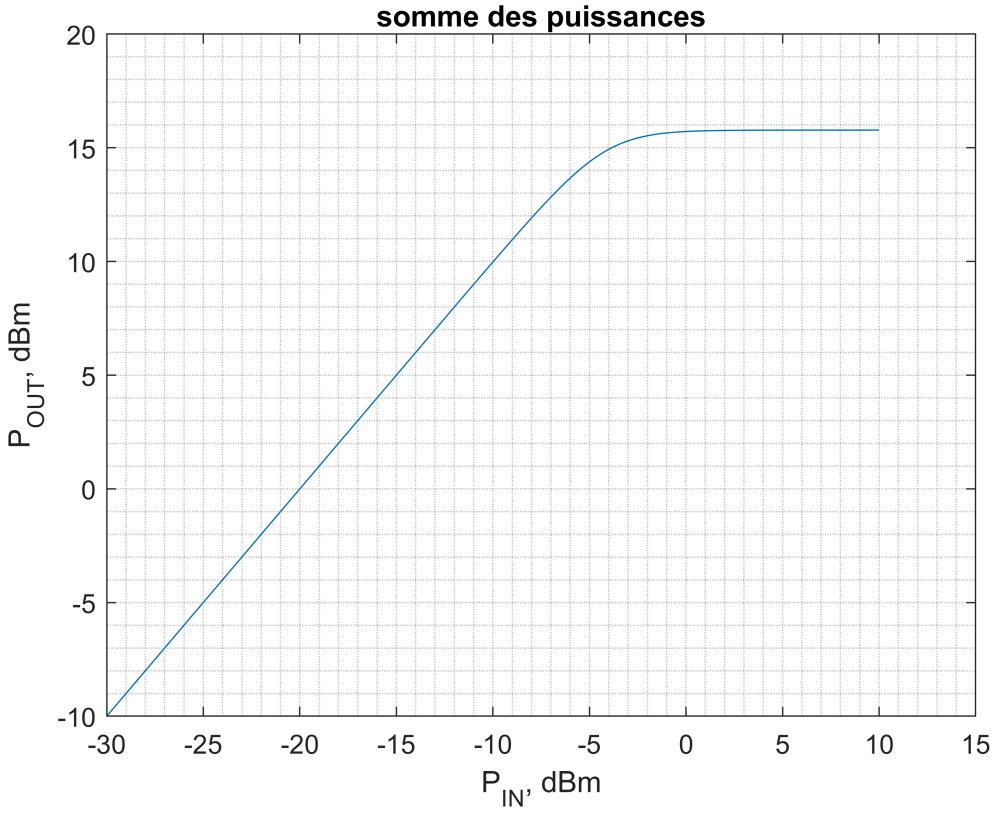
Calcul des états de compression des différents signaux

On commence par déterminer l'état de compression général de l'amplificateur en fonction de la somme des 3 signaux entrants :

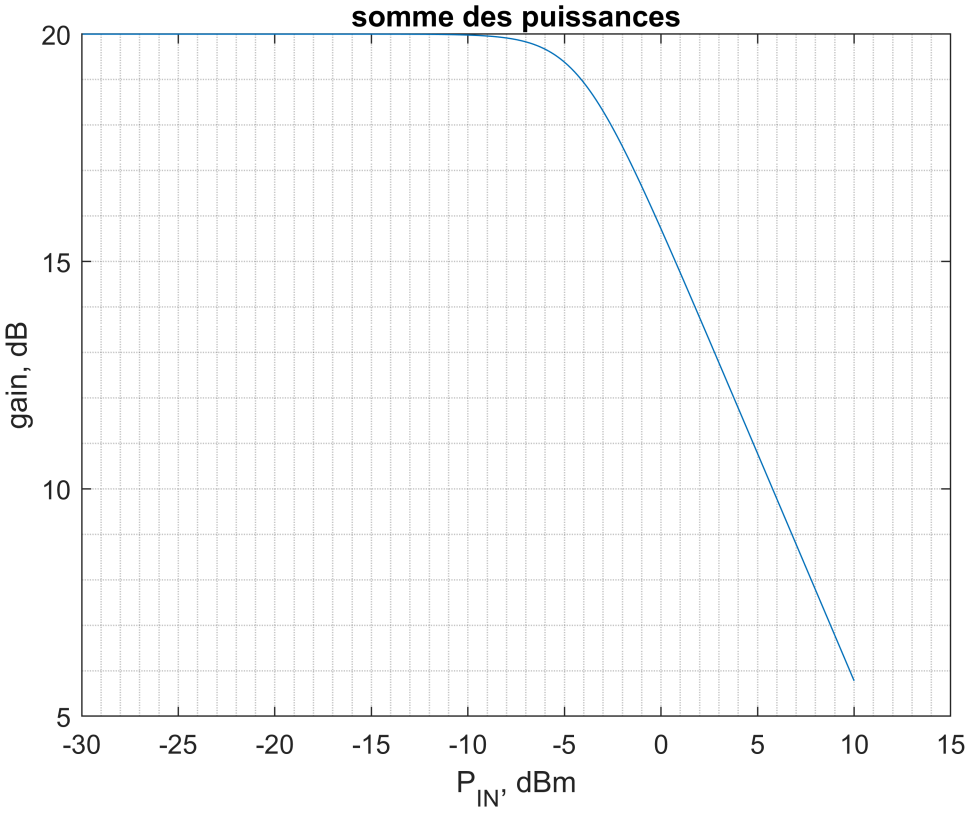
```
amplitude_in_total = 10*log10(10.^((amp1)/10) + ...
    10.^((amp2)/10) + ...
    10.^((amp3)/10));
amplitude_out_totale = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ...
    10.^(-(p1dB + sqrt(smo) - 1)/smo));
```

L'amplitude totale en sortie est la somme des opposés de la puissance de sortie linéaire (sans compression) et de la puissance de saturation (puissance maximale attendue en sortie de l'amplificateur). Le coefficient de coude smo permet d'adoucir la transition entre les deux régimes :

```
plot(amplitude_in_total, amplitude_out_totale)
grid minor
xlabel('P_I_N, dBm')
ylabel('P_O_U_T, dBm')
title('somme des puissances')
```

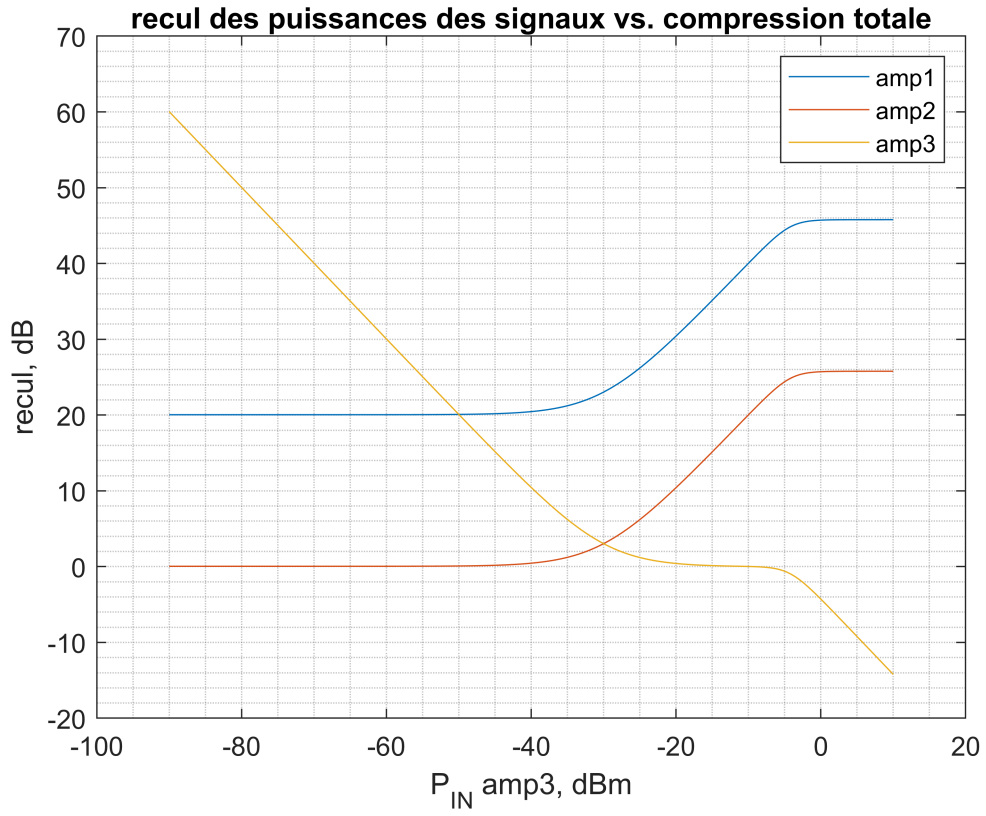



```
plot(amplitude_in_total, amplitude_out_totale-amplitude_in_total)
grid minor
xlabel('P_I_N, dBm')
ylabel('gain, dB')
title('somme des puissances')
```



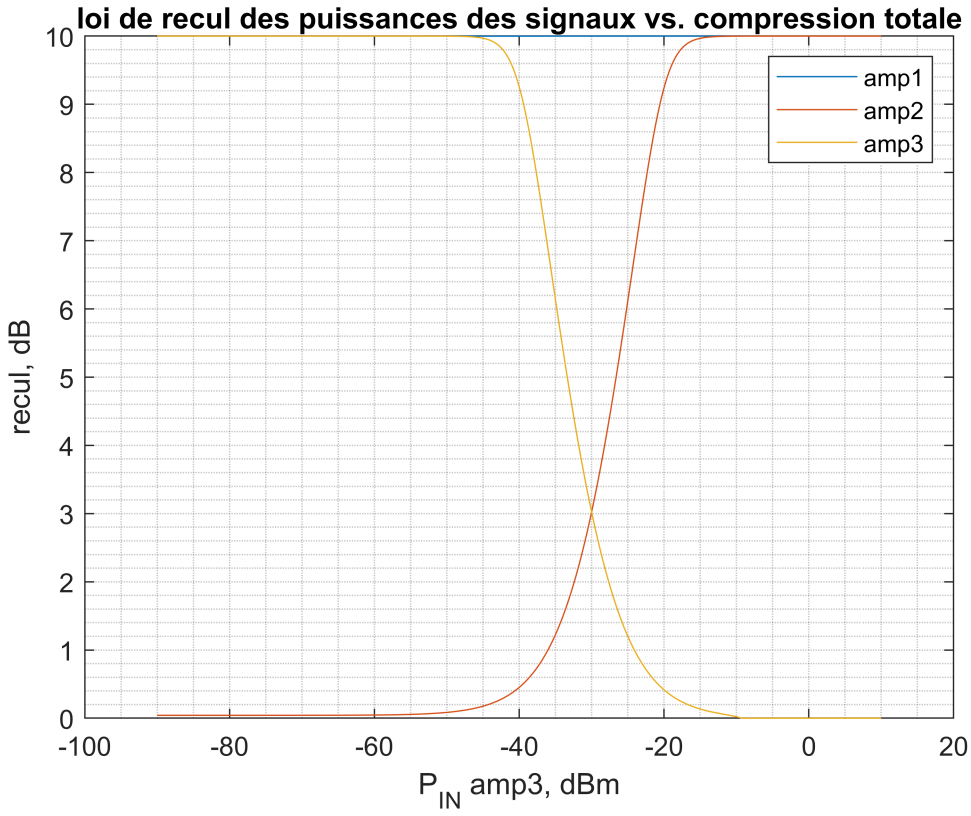
Par rapport à la puissance totale de l'amplificateur, chaque signal présente un recul de puissance dont on va se servir pour déterminer le comportement de l'amplification de chaque signal :

```
recul_amp1 = amplitude_out_totale - gain_transducique - amp1;
recul_amp2 = amplitude_out_totale - gain_transducique - amp2;
recul_amp3 = amplitude_out_totale - gain_transducique - amp3;

plot(amp3, recul_amp1, ...
    amp3, recul_amp2, ...
    amp3, recul_amp3)
grid minor
xlabel('P_I_N amp3, dBm')
ylabel('recul, dB')
title('recul des puissances des signaux vs. compression totale')
legend({'amp1', 'amp2', 'amp3'})
```



amp2 est le signal le plus fort dans l'amplificateur jusqu'à ce que amp3 le dépasse (-30 dBm dans cet exemple). On va limiter la dynamique de ces reculs entre [0 : recul_max] en y ajoutant également une transition douce dans dans la limite recul_max :

```
loi_de_compression_amp1 = max(0, -smo*log10(10.^(-recul_max/smo) + ...
    10.^(-recul_amp1/smo)));
loi_de_compression_amp2 = max(0, -smo*log10(10.^(-recul_max/smo) + ...
    10.^(-recul_amp2/smo)));
loi_de_compression_amp3 = max(0, -smo*log10(10.^(-recul_max/smo) + ...
    10.^(-recul_amp3/smo)));

plot(amp3, loi_de_compression_amp1, ...
    amp3, loi_de_compression_amp2, ...
    amp3, loi_de_compression_amp3)
grid minor
xlabel('P_I_N amp3, dBm')
ylabel('recul, dB')
title('loi de recul des puissances des signaux vs. compression totale')
legend({'amp1', 'amp2', 'amp3'})
```



Cette loi nous permet de recalculer le comportement de compression sur chaque signal afin notamment de provoquer une compression visible des petits signaux 10 dB (recul_max) avant la compression du signal fort, ce qui est aussi observé en pratique. Les gains et puissances pour les signaux sont alors les suivants :

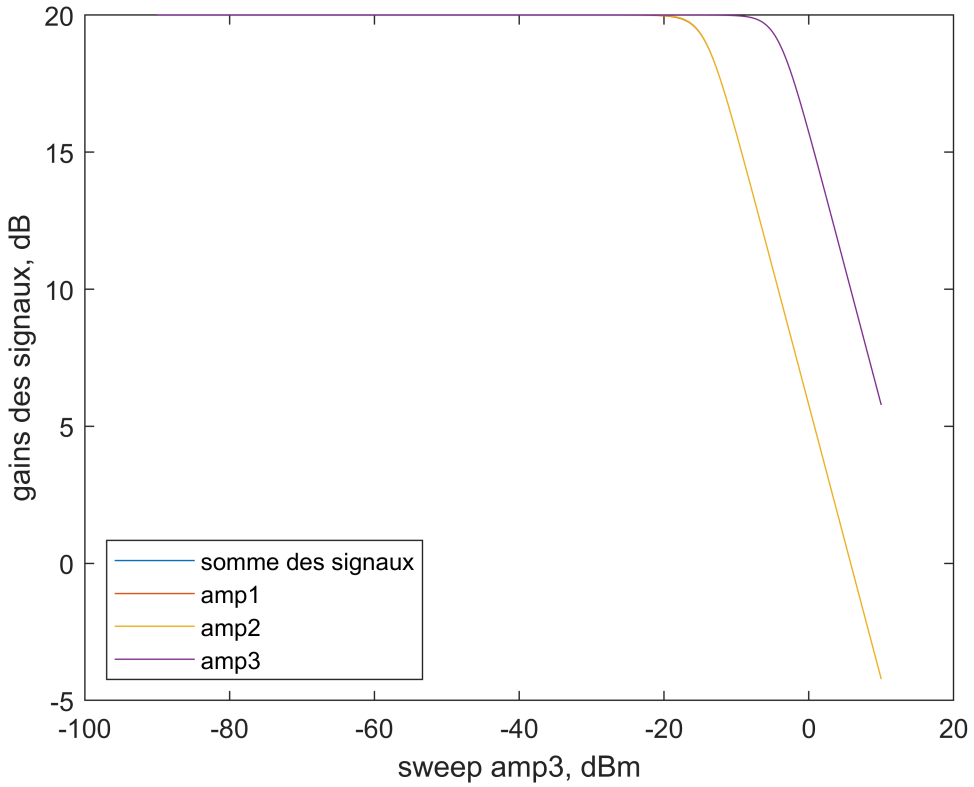
```
gain_non_lin_amp1 = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ...
    10.^(-(p1dB + sqrt(smo) - 1 - loi_de_compression_amp1)/smo)) - amplitude_in_total;
gain_non_lin_amp2 = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ...
    10.^(-(p1dB + sqrt(smo) - 1 - loi_de_compression_amp2)/smo)) - amplitude_in_total;
gain_non_lin_amp3 = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ...
    10.^(-(p1dB + sqrt(smo) - 1 - loi_de_compression_amp3)/smo)) - amplitude_in_total;

amplitude_out_non_lin_amp1 = amp1 + gain_non_lin_amp1;
amplitude_out_non_lin_amp2 = amp2 + gain_non_lin_amp2;
amplitude_out_non_lin_amp3 = amp3 + gain_non_lin_amp3;

somme = 10*log10(10.^(amplitude_out_non_lin_amp1/10) + ...
    10.^(amplitude_out_non_lin_amp2/10) + ...
    10.^(amplitude_out_non_lin_amp3/10));
```

Pour chaque signal, le gain non linéaire doit être recalculé et dépend de la somme de tous les signaux entrants ;

Le comportement des gains est le suivant :

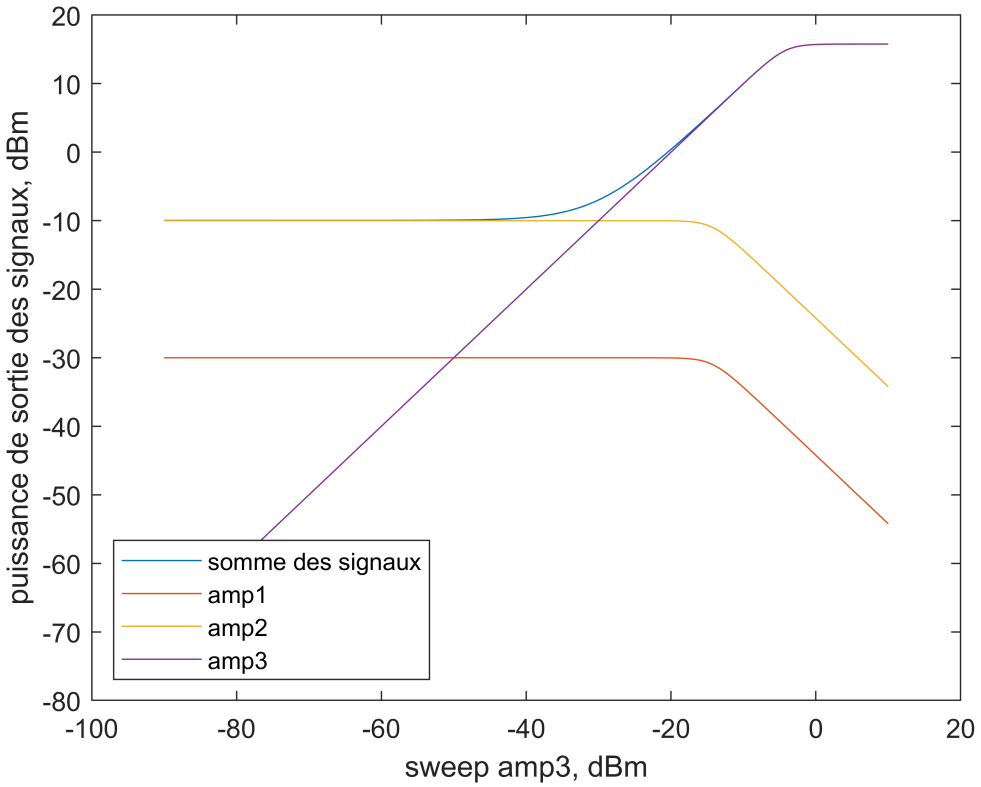
```
plot(amp3, amplitude_out_totale - amplitude_in_total, ...
    amp3, gain_non_lin_amp1, ...
    amp3, gain_non_lin_amp2, ...
```

```
    amp3, gain_non_lin_amp3);
legend({'somme des signaux', 'amp1', 'amp2', 'amp3'}, 'Location','southwest');
xlabel('sweep amp3, dBm');
ylabel('gains des signaux, dB');
```



Le comportement des puissances est le suivant :

```
plot(amp3, somme, ...
    amp3, amplitude_out_non_lin_amp1, ...
    amp3, amplitude_out_non_lin_amp2, ...
    amp3, amplitude_out_non_lin_amp3);
legend({'somme des signaux', 'amp1', 'amp2', 'amp3'}, 'Location','southwest');
xlabel('sweep amp3, dBm');
ylabel('puissance de sortie des signaux, dBm');
```

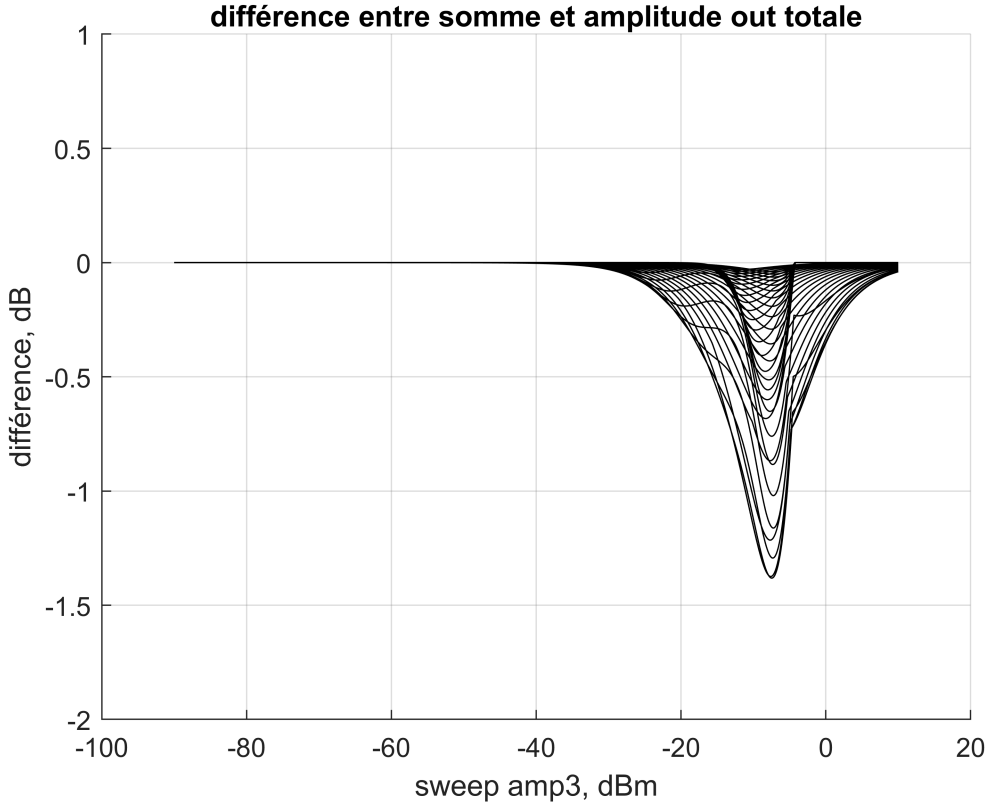


On observe la compression apparente de amp1 et amp2 10 dB avant amp3.

Dans ce modèle, des limitations apparaissent lors des calculs de plusieurs puissances proches de la compression ; le snipet suivant montre la différence entre la puissance de sortie calculée en additionnant toutes les puissances d'entrée (amplitude_out_totale) et la somme des amplitudes non linéaires (somme) en faisant varier amp2 de -30 à +10 dBm :

```
clf
for ii = -30:10
    amp1 = -50;
    amp2 = ii;
    amp3 = -90:0.1:10;

    gain_transducique = 20; % dB
    p1dB = 15; % dBm

    smo = sqrt(10); % coef de coude
    recul_max = 10; % dB

    % compression globale de l'ampli liée à la somme de tous les signaux
    amplitude_in_total = 10*log10(10.^((amp1)/10) + ...
        10.^((amp2)/10) + ...
        10.^((amp3)/10));
    amplitude_out_totale = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ..
        10.^(-(p1dB + sqrt(smo) - 1)/smo));

    % recul de chaque signal sur la compression totale
```

```
    recul_amp1 = amplitude_out_totale - gain_transducique - amp1;
    recul_amp2 = amplitude_out_totale - gain_transducique - amp2;
    recul_amp3 = amplitude_out_totale - gain_transducique - amp3;

    % loi de compression pour chaque signal
    loi_de_compression_amp1 = max(0, -smo*log10(10.^(-recul_max/smo) + ...
        10.^(-recul_amp1/smo)));
    loi_de_compression_amp2 = max(0, -smo*log10(10.^(-recul_max/smo) + ...
        10.^(-recul_amp2/smo)));
    loi_de_compression_amp3 = max(0, -smo*log10(10.^(-recul_max/smo) + ...
        10.^(-recul_amp3/smo)));

    % gain non linéaire de chaque signal
    gain_non_lin_amp1 = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ...
        10.^(-(p1dB + sqrt(smo) - 1 - loi_de_compression_amp1)/smo)) - amplitude_in_total;
    gain_non_lin_amp2 = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ...
        10.^(-(p1dB + sqrt(smo) - 1 - loi_de_compression_amp2)/smo)) - amplitude_in_total;
    gain_non_lin_amp3 = -smo*log10(10.^(-(amplitude_in_total + gain_transducique)/smo) + ...
        10.^(-(p1dB + sqrt(smo) - 1 - loi_de_compression_amp3)/smo)) - amplitude_in_total;

    % amplitude finale de chaque signal
    amplitude_out_non_lin_amp1 = amp1 + gain_non_lin_amp1;
    amplitude_out_non_lin_amp2 = amp2 + gain_non_lin_amp2;
    amplitude_out_non_lin_amp3 = amp3 + gain_non_lin_amp3;

    somme = 10*log10(10.^(amplitude_out_non_lin_amp1/10) + ...
        10.^(amplitude_out_non_lin_amp2/10) + ...
        10.^(amplitude_out_non_lin_amp3/10));

    hold on
    plot(amp3, somme-amplitude_out_totale, 'k')
    ylim([-2 1]);
    grid on
    hold off

end
xlabel('sweep amp3, dBm');
ylabel('différence, dB');
title('différence entre somme et amplitude out totale')
```


différence entre somme et amplitude out totale
différence, dB
sweep amp3, dBm
1
0.5
0
-0.5
-1
-1.5
-2
-100
-80
-60
-40
-20
0
20