

Séance 1

```
%NOTE: Changement de maquette : 09 ; montage elec 12
clear all;
close all;
clc;
% Fichiers Machine Impédances
mes_050 = readmatrix('Seance1/MesuresMachine/50.CSV');
mes_055 = readmatrix('Seance1/MesuresMachine/55.CSV');
mes_060 = readmatrix('Seance1/MesuresMachine/60.CSV');
mes_065 = readmatrix('Seance1/MesuresMachine/65.CSV');
mes_070 = readmatrix('Seance1/MesuresMachine/70.CSV');
mes_075 = readmatrix('Seance1/MesuresMachine/75.CSV');
mes_080 = readmatrix('Seance1/MesuresMachine/80.CSV');
mes_085 = readmatrix('Seance1/MesuresMachine/85.CSV');
mes_090 = readmatrix('Seance1/MesuresMachine/90.CSV');

%Calcul des valeurs complexes ( $z = R + i\omega L$ ):
f = mes_050(:,1);
Z_050 = mes_050(:,3) + 2*pi*1i*f .* mes_050(:,2);
Z_055 = mes_055(:,3) + 2*pi*1i*f .* mes_055(:,2);
Z_060 = mes_060(:,3) + 2*pi*1i*f .* mes_060(:,2);
Z_065 = mes_065(:,3) + 2*pi*1i*f .* mes_065(:,2);
Z_070 = mes_070(:,3) + 2*pi*1i*f .* mes_070(:,2);
Z_075 = mes_075(:,3) + 2*pi*1i*f .* mes_075(:,2);
Z_080 = mes_080(:,3) + 2*pi*1i*f .* mes_080(:,2);
Z_085 = mes_085(:,3) + 2*pi*1i*f .* mes_085(:,2);
Z_090 = mes_090(:,3) + 2*pi*1i*f .* mes_090(:,2);
Z = [Z_050, Z_055, Z_060, Z_065, Z_070, Z_075, Z_080, Z_085, Z_090 ];

%Vecteur distance
x = 0.5:0.05:0.9;
figure;
```

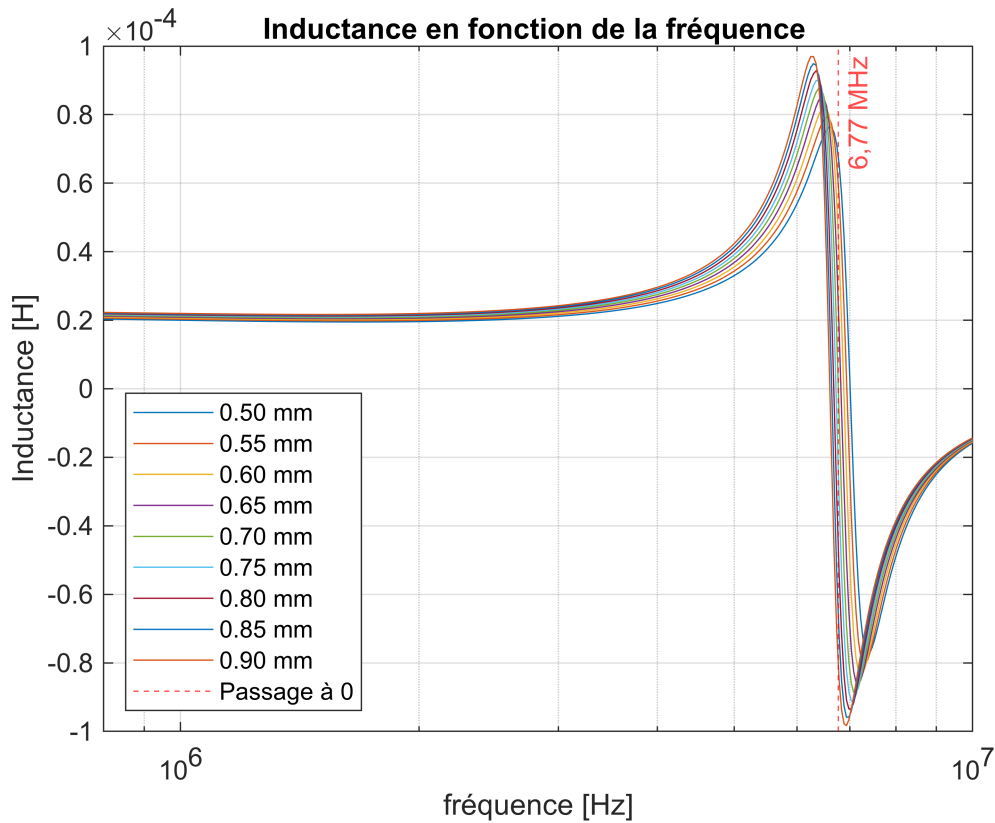
Inductance en fonction de la fréquence L(f)

```
% Inductances
Lb_050 = mes_050(:,2);
Lb_055 = mes_055(:,2);
Lb_060 = mes_060(:,2);
Lb_065 = mes_065(:,2);
Lb_070 = mes_070(:,2);
Lb_075 = mes_075(:,2);
Lb_080 = mes_080(:,2);
Lb_085 = mes_085(:,2);
Lb_090 = mes_090(:,2);
Lb = [Lb_050, Lb_055, Lb_060, Lb_065, Lb_070, Lb_075, Lb_080, Lb_085, Lb_090 ];
% Graphique
plot(f, Lb)
```

```

title('Inductance en fonction de la fréquence ');
ylabel("Inductance [H]")
xlabel("fréquence [Hz]")
xline(6.767*1000000,'--r', '6,77 MHz');
legend('0.50 mm', '0.55 mm', '0.60 mm', '0.65 mm', '0.70 mm', '0.75 mm', '0.80 mm', '0.85 mm', '0.90 mm', 'Passage à 0');
xlim([8e5 1e7]);
set(gca, 'XScale', 'log')
grid;

```



Inductance à 150 kHz en fonction de la distance L(x)

```

% Inductances à 150kHz
Lbx_050 = mes_050(26,2);
Lbx_055 = mes_055(26,2);
Lbx_060 = mes_060(26,2);
Lbx_065 = mes_065(26,2);
Lbx_070 = mes_070(26,2);
Lbx_075 = mes_075(26,2);
Lbx_080 = mes_080(26,2);
Lbx_085 = mes_085(26,2);
Lbx_090 = mes_090(26,2);
Lbx = [Lbx_050, Lbx_055, Lbx_060, Lbx_065, Lbx_070, Lbx_075, Lbx_080, Lbx_085, Lbx_090];

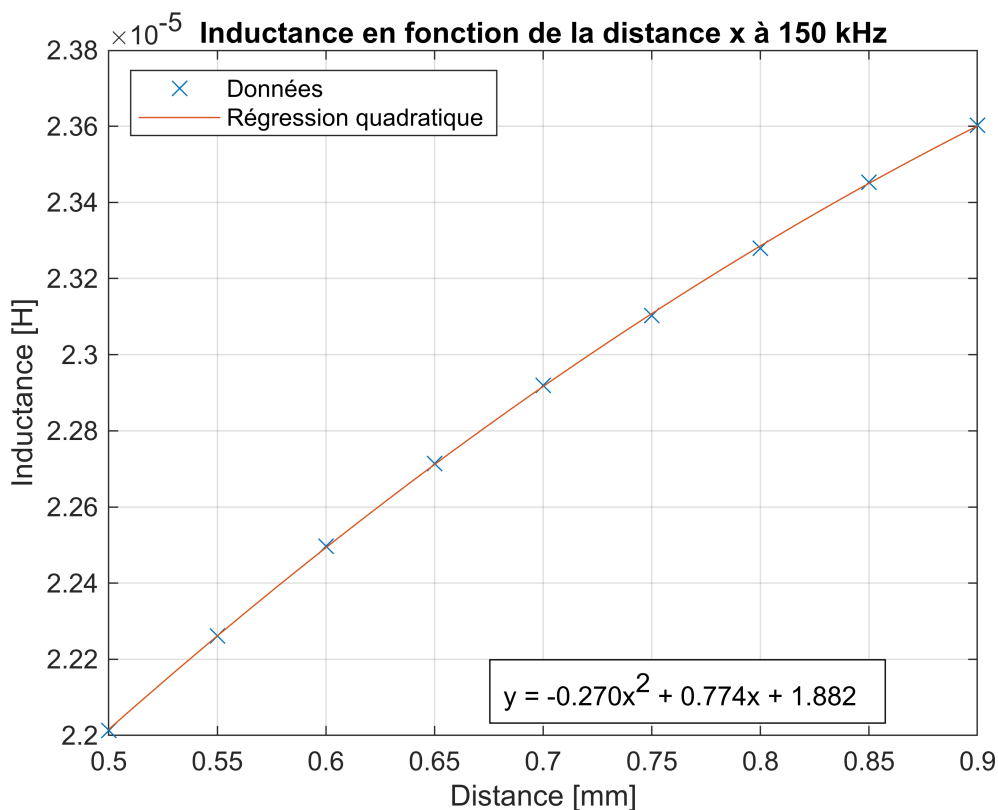
%Graphique

```

```

% Régression quadratique
coeffs1 = polyfit(x, Lbx, 2); % Ajustement d'une courbe d'ordre 2
x_fit = linspace(min(x), max(x), 100); % Points pour tracer la courbe lisse
Lbx_fit = polyval(coeffs1, x_fit); % Calcul des valeurs ajustées
% Tracé des données et de la régression
figure;
plot(x, Lbx, 'x', 'MarkerSize', 8, 'DisplayName', 'Données'); % Données mesurées
hold on;
plot(x_fit, Lbx_fit, '-', 'DisplayName', 'Régression quadratique'); % Courbe ajustée
grid on;
% Ajouter les labels et le titre
title('Inductance en fonction de la distance x à 150 kHz');
ylabel('Inductance [H]');
xlabel('Distance [mm]');
% Ajouter l'équation comme annotation
eqn1 = sprintf("y = %.3fx^2 + %.3fx + %.3f", coeffs1(1)*100000, coeffs1(2)*100000, coeffs1(3)*100000);
dim = [0.47 0.1 0.1 0.1]; % Position de l'annotation (valeurs normalisées entre 0 et 1)
annotation('textbox', dim, 'String', eqn1, 'FitBoxToText', 'on', 'BackgroundColor', 'w');
% Légende
legend('Location', 'northwest');
hold off;

```



Résistance en fonction de la fréquence R(f)

```

% Résistances

```

```

Rb_050 = mes_050(:,3);
Rb_055 = mes_055(:,3);
Rb_060 = mes_060(:,3);
Rb_065 = mes_065(:,3);
Rb_070 = mes_070(:,3);
Rb_075 = mes_075(:,3);
Rb_080 = mes_080(:,3);
Rb_085 = mes_085(:,3);
Rb_090 = mes_090(:,3);
Rb = [Rb_050, Rb_055, Rb_060, Rb_065, Rb_070, Rb_075, Rb_080, Rb_085, Rb_090 ];

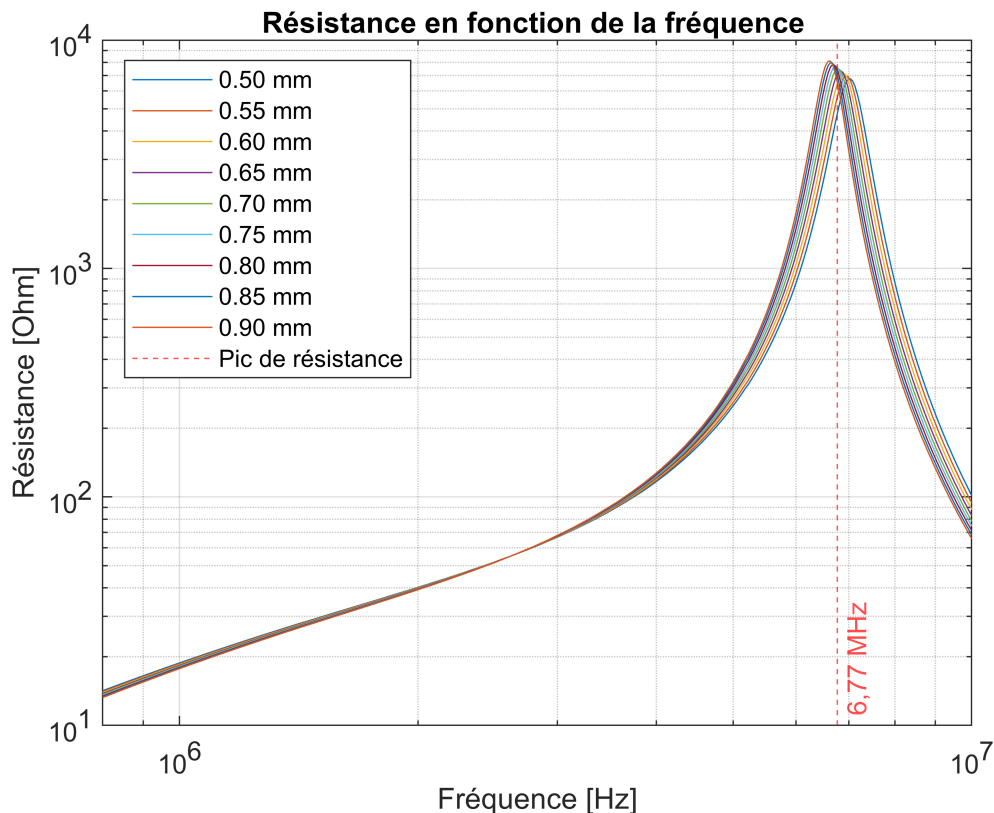
```

% Graphique échelle log/log

```

loglog(f, Rb);
grid;
title('Résistance en fonction de la fréquence ');
xl = xline(6.767*1000000,'--r', '6,77 MHz');
xl.LabelVerticalAlignment = 'bottom';
legend('0.50 mm', '0.55 mm', '0.60 mm', '0.65 mm', '0.70 mm', '0.75 mm', '0.80 mm','0.85 mm', '0.90 mm');
xlim([8e5 1e7]);
ylabel("Résistance [Ohm]")
xlabel("Fréquence [Hz]")

```



Résistance à 150 kHz en fonction de la distance R(x)

```

%Résistances à 150 kHz
Rbx_050 = mes_050(26,3);

```

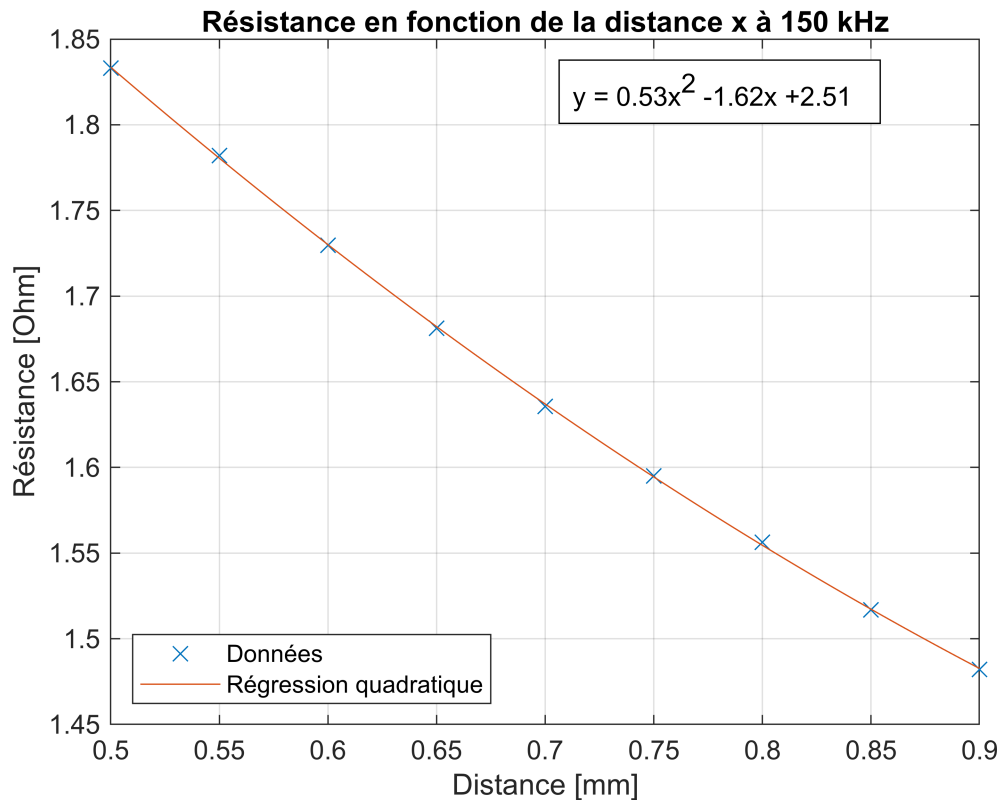
```

Rbx_055 = mes_055(26,3);
Rbx_060 = mes_060(26,3);
Rbx_065 = mes_065(26,3);
Rbx_070 = mes_070(26,3);
Rbx_075 = mes_075(26,3);
Rbx_080 = mes_080(26,3);
Rbx_085 = mes_085(26,3);
Rbx_090 = mes_090(26,3);
Rbx = [Rbx_050, Rbx_055, Rbx_060, Rbx_065, Rbx_070, Rbx_075, Rbx_080, Rbx_085, Rbx_090 ];

% Graphique

% Régression quadratique
coeffs2 = polyfit(x, Rbx, 2); % Ajustement d'une courbe d'ordre 2
x_fit = linspace(min(x), max(x), 100); % Points pour tracer la courbe lisse
Rbx_fit = polyval(coeffs2, x_fit); % Calcul des valeurs ajustées
% Tracé des données et de la régression
figure;
plot(x, Rbx, 'x', 'MarkerSize', 8, 'DisplayName', 'Données'); % Données mesurées
hold on;
plot(x_fit, Rbx_fit, '-', 'DisplayName', 'Régression quadratique'); % Courbe ajustée
grid on;
% Ajouter les labels et le titre
title('Résistance en fonction de la distance x à 150 kHz');
ylabel('Résistance [Ohm]');
xlabel('Distance [mm]');
% Ajouter l'équation comme annotation
eqn2 = sprintf("y = %.2fx^2 %.2fx %.2f", coeffs2(1), coeffs2(2), coeffs2(3)); % Formater l'équation
dim = [0.53 0.8 0.1 0.1]; % Position de l'annotation (valeurs normalisées entre 0 et 1)
annotation('textbox', dim, 'String', eqn2, 'FitBoxToText', 'on', 'BackgroundColor', 'w');
% Légende
legend('Location', 'southwest');
hold off;

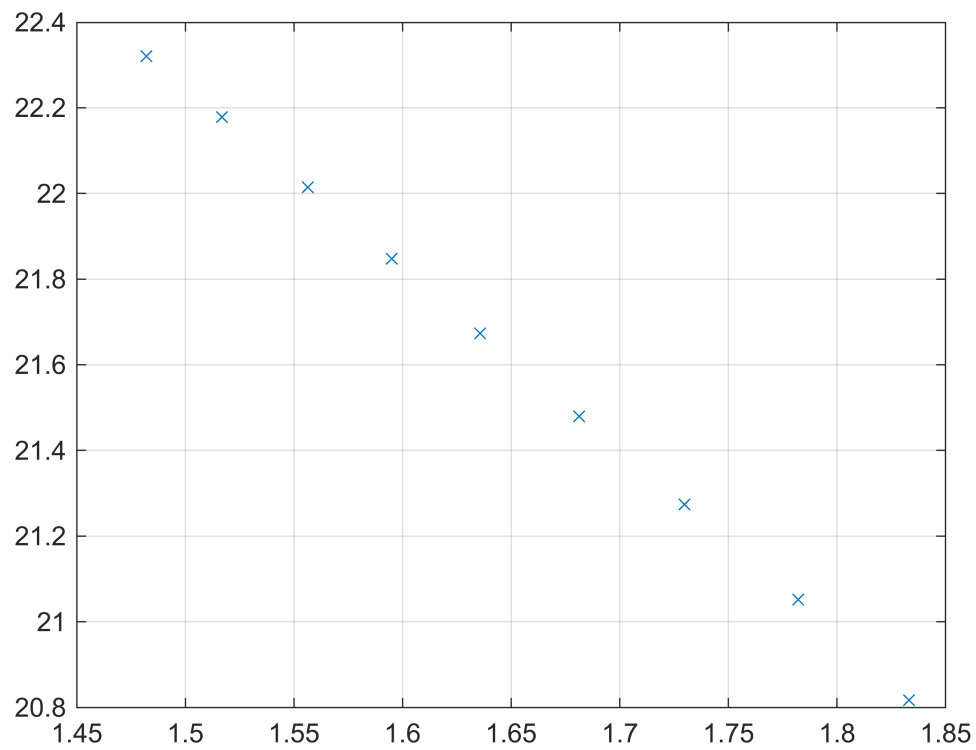
```



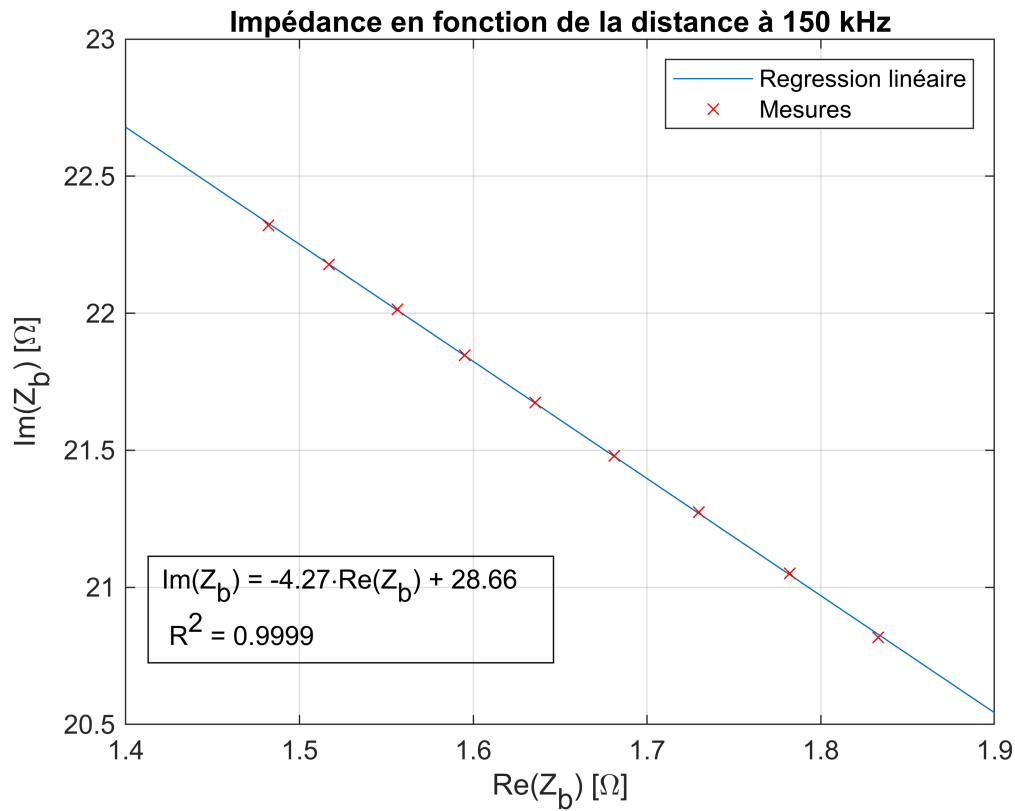
Impédance à 150 kHz

```
% Impédance
Z150 = [Z_050(26), Z_055(26), Z_060(26), Z_065(26), Z_070(26), Z_075(26), Z_080(26), Z_085(26)];

% Graphique
figure;
plot(real(Z150), imag(Z150), 'x');
%Regression lineaire
x_reg = linspace(1.4, 1.9, 1000);
reg = polyfit(real(Z150(1,:)), imag(Z150(1,:)), 1);
%Parametres
a = reg(1,1);
b = reg(1,2);
r = fitlm(real(Z150(1,:)), imag(Z150(1,:)));
R2 = r.Rsquared.Ordinary;
eqn3 = sprintf("Im(Z_{b}) = %.2f \cdot Re(Z_{b}) + %.2f \n R^{2} = %.4f", reg(1), reg(2), R2);
grid on;
```

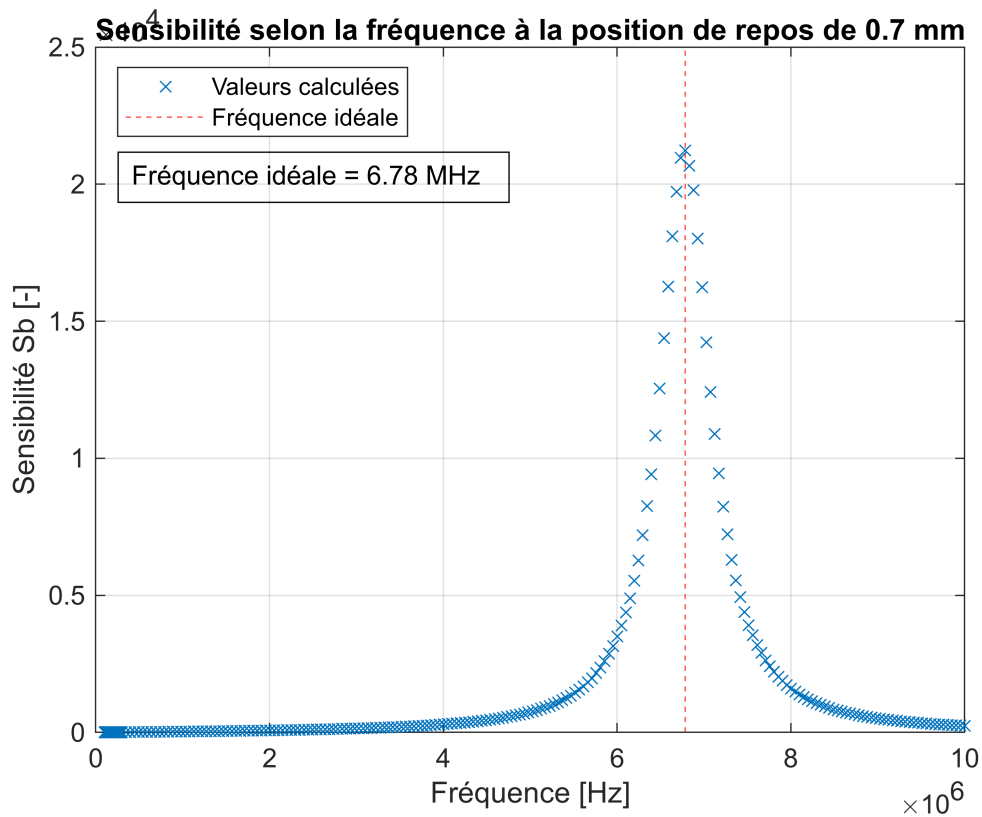


```
% Impédance avec regression lineaire
imZb = polyval(reg,x_reg);
plot(x_reg,imZb,"-",real(Z150(1,:)),imag(Z150(1:)), "rx")
title('Impédance en fonction de la distance à 150 kHz');
annotation("textbox",[0.15 0.01 0.3 0.3], 'string',eqn3, 'FitBoxToText', 'on');
legend('Regression linéaire', 'Mesures', 'location', 'northeast');
xlabel('Re( $Z_b$ ) [ $\Omega$ ]');
ylabel('Im( $Z_b$ ) [ $\Omega$ ]');
grid on;
```



Sensibilité

```
S = abs(Z(:, 4)-Z(:, 6))/0.1;
% Trouver le max
[vmax, indicefmax] = max(S);
fmax = f(indicefmax);
fmaxstr=sprintf("Fréquence idéale = %.2f MHz",fmax/(1e6));
% Graphique
plot(f,S,"x")
xlabel('Fréquence [Hz]');
ylabel('Sensibilité Sb [-]');
title('Sensibilité selon la fréquence à la position de repos de 0.7 mm');
xline(fmax, '--r');
annotation("textbox",[0.15 0.5 0.3 0.3],'string',fmaxstr,'FitBoxToText','on');
legend('Valeurs calculées','Fréquence idéale','Location','northwest');
grid on;
```

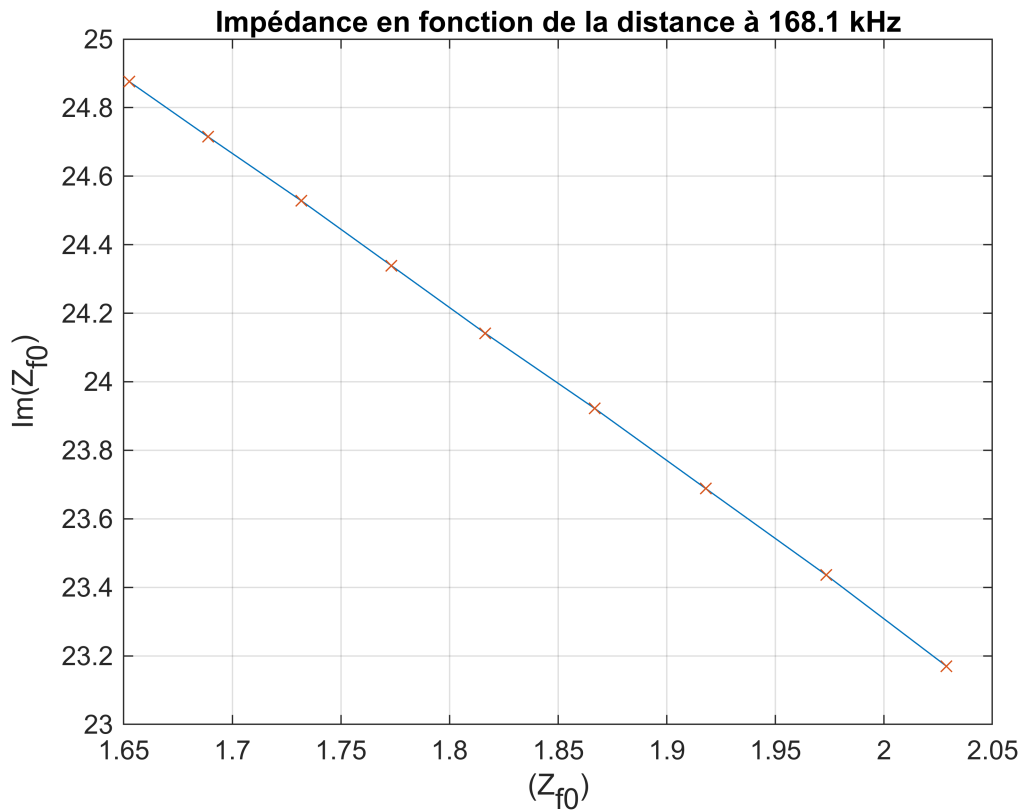
Seance 2

```
f0_mes = 166.5;      %170.8e3 ; %[Hz] (old)
Z(101, :) = [];
Z(300, :) = [];

% Calcul de la fréquence de résonance théorique
f0 = 150.3 * 1e3; % [kHz] fréquence résonnance mesurée
C = 39 * 1e-9;
f(101,:) = [];
f(300, :) = [];

for aaaa = 1 : 20
    z_f0_70 = interp1(f,Z(:,5), f0);
    Lb = imag(z_f0_70)/(f0*2*pi);
    Rb = real(z_f0_70);
    omega0 = sqrt(1/(Lb*C) - Rb^2/Lb^2);
    f0 = omega0/(2*pi);
    % U excitation
end
z_f0 = interp1(f,Z,f0);
plot(real(z_f0),imag(z_f0),real(z_f0),imag(z_f0),'x')
ylabel('Im(Z_{f0})')
xlabel('Z_{f0}')
title('Impédance en fonction de la distance à 168.1 kHz')
```

```
grid;
```



```
Q = omega0*Lb/Rb;  
Rp = Rb*(1+Q^2);  
  
Rmes = 324; %[Ohm]  
  
UmodRMS = 1.074;  
Umod = 1.519;  
  
H_z_f0 = z_f0./((1+1i.*C.*Rp.*omega0).*z_f0 + Rp);  
% Calcul des coordonnées réelles et imaginaires  
x = real(H_z_f0);  
y = imag(H_z_f0);  
  
% Calcul du cercle passant par ces points (utilisation des moindres carrés pour ajuster un cercle)  
A = [2*x', 2*y', ones(length(H_z_f0), 1)];  
B = x'.^2 + y'.^2;  
  
% Résolution du système linéaire pour obtenir les paramètres du cercle (méthode des moindres carrés)  
params = A \ B; % Résolution du système de manière optimale en termes de moindres carrés  
  
% Paramètres du cercle  
xc = params(1); % Centre du cercle en x  
yc = params(2); % Centre du cercle en y  
r = sqrt(params(3) + xc^2 + yc^2); % Rayon du cercle
```

```

% Affichage des points et du cercle
theta = linspace(0, 2*pi, 100); % Angles pour dessiner le cercle
x_circle = xc + r * cos(theta);
y_circle = yc + r * sin(theta);

labels = {'0.50', '0.55', '0.60', '0.65', '0.70', '0.75', '0.80', '0.85', '0.90'};

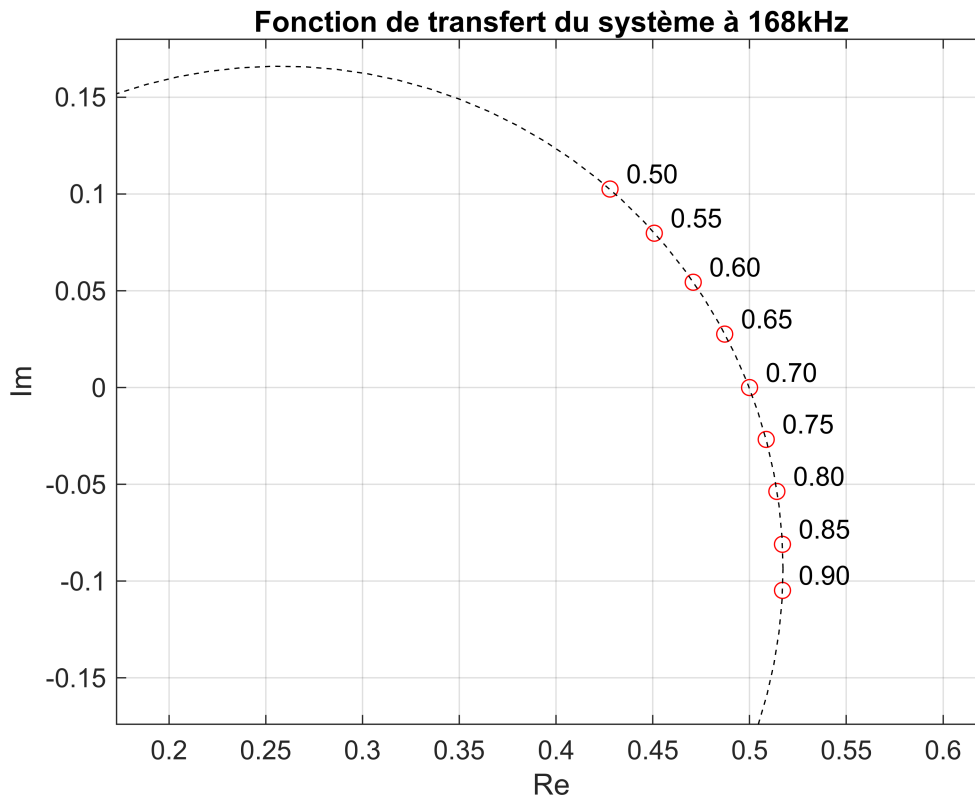
% Tracer
figure;
plot(x, y, 'ro'); % Points d'origine

for foo = 1:length(x)
    text(x(foo) + 0.035, y(foo), labels{foo}, 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right');
end

hold on;
plot(x_circle, y_circle, 'k--'); % Cercle
axis equal;
xlabel('Re');
ylabel('Im');
title('Fonction de transfert du système à 168kHz');
grid on;
hold off;
xlim([0.265 0.714])
ylim([-0.191 0.163])

xlim([0.173 0.622])
ylim([-0.174 0.180])

```



Seance 3

```
%Vecteur d'angle
angle = [0, 20, 40, 60, 80, 100, 120, 140, 160, 180];
%Tension de sortie mesurée
Umes = [317.4, 302.2, 250.3, 168.1, 65.28, -46.02, -152.2, -240.3, -299.6, -323];
% Tension de sortie mesurée
Uth = 0.5 * cos(angle*((2*pi)/360));
% Graphique
plot(angle, Uth, '-x', angle, Umes/1000, '-x')
xlabel('Déphasage \phi [deg]');
ylabel('U_{out} [V]');
legend('U_{out} calculé', 'U_{out} mesuré');
grid on;
title('Tension de sortie du filtre en fonction du déphasage');
```

