

# 1 Introduction

A l'aide d'ondes ultrasonores, il est possible, grâce à la pression de radiation acoustique exercée par l'onde ultrasonore ,de maintenir en lévitation des objets de faible masse, comme de petites billes de polystyrène de quelques millimètres de rayon. Ce phénomène porte le nom de "lévitation acoustique".

La lévitation acoustique possède de nombreuses applications intéressantes telles que la création d' hologrammes, basée sur le déplacement rapide d'une bille de polystyrène, et la manipulation extrêmement stable et précise de petites particules [Bar14] [4]. Il est alors possible de manipuler et de confiner ,sans support matériel , des éléments radioactifs et des cellules vivantes par exemple.

On se propose donc d'étudier l'influence de la fréquence d'excitation sur les positions d'équilibre de billes de polystyrène en lévitation acoustique. Nous allons chercher à comprendre le phénomène physique sous-jacent, par le calcul d'un modèle théorique, la conception et la réalisation d'un modèle expérimental, puis la conception d'un modèle numérique. Nous allons ainsi aboutir par la mise en cohérence de ces trois modèles.

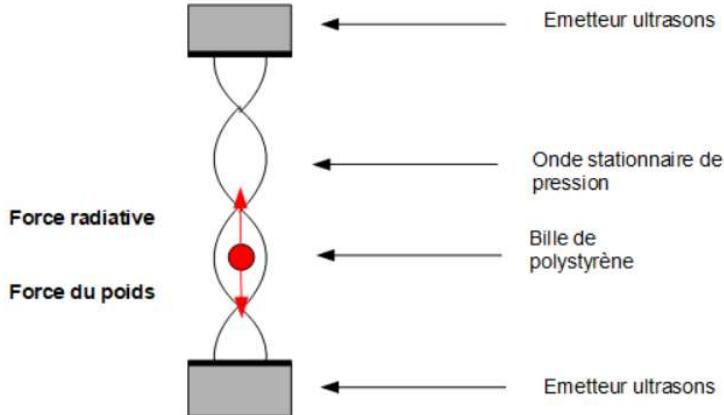


Figure 1: Principe des ondes stationnaires.

Le principe de notre expérience de lévitation acoustique est le suivant: deux émetteurs ultrasons émettent des ondes de pression qui ,par leurs interférences, créent une force de pression de radiation acoustique permettant de compenser localement dans les noeuds de pression la force de pesanteur.

## 2 Conception

La conception du modèle expérimental s'est déroulée en trois étapes:

1. Réalisation de petites expériences acoustiques avec la mise au point d'un premier prototype Arduino,
2. Réalisation d'un prototype analogique , qui n'a pas abouti,
3. Réalisation d'une version finale pilotée par Arduino.

### 2.1 Premier prototype Arduino

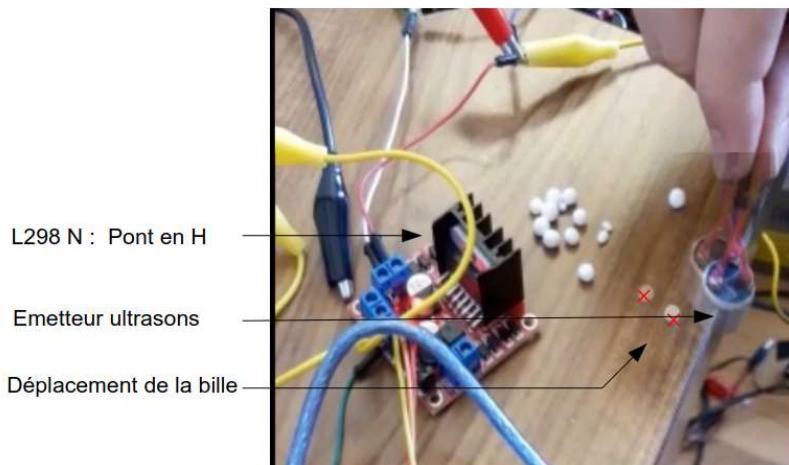


Figure 2: Déplacement d'un bille avec des ultrasons.

La première expérience consiste à vérifier l'influence des ondes ultrasons sur une bille de polystyrène. On cherche à faire rouler la bille, juste par action des ondes. On remarque que l'on y parvient à 40kHz pour une tension de 24V aux bornes de l'émetteur. La fréquence de 40 kHz a été choisie car il s'agit de la fréquence adaptée à ces émetteurs , récupérés sur des modules ultrasons Arduino.

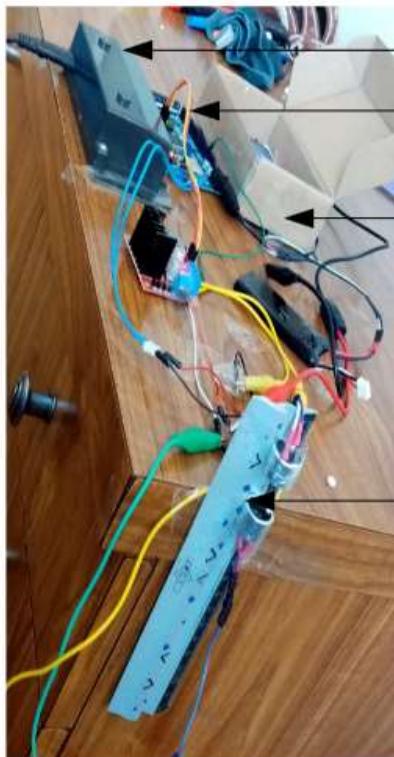


Figure 3: Déplacement d'une bille avec des ultrasons.

Alimentation 24V

Arduino mega

Oscilloscope

Deux émetteurs,  
même fréquence



Figure 4: Déplacement d'un bille avec des ultrasons.

La deuxième expérience consiste à positionner deux transducteurs ultrasons qui se font face , sur une règle graduée. On utilise une alimentation 24V et une carte moteur L298N pour alimenter les émetteurs. Un mini-oscilloscope permet de vérifier la fréquence générée par un Arduino Mega. Nous mettons ainsi en place des ondes stationnaires qui assurent la suspension des billes de polystyrène ,piégées au niveau des noeuds de pression.

## 2.2 Deuxième prototype analogique

Cependant ce prototype n'est pas satisfaisant car on ne peut régler de manière précise la fréquence d'excitation des émetteurs à ultrasons. C'est du au limites de l'Arduino, qui permet pas d'atteindre facilement le domaine de fréquence souhaité. En effet toute opération mathématique élémentaire prend au minimum 12 microsecondes à s'exécuter dans le calculateur de l'Arduino, ce qui est bien trop lent pour atteindre les 40 kHz. On cherche donc ensuite à créer un prototype analogique basé sur une horloge à amplificateur opérationnel en régime saturé.

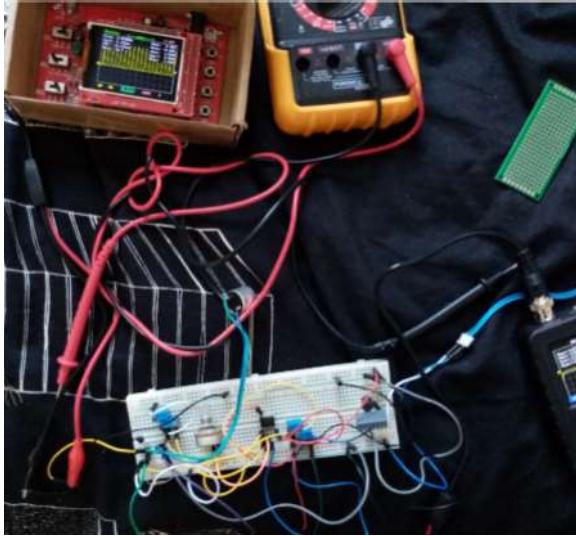


Figure 5: Première horloge sur platine d'essai.

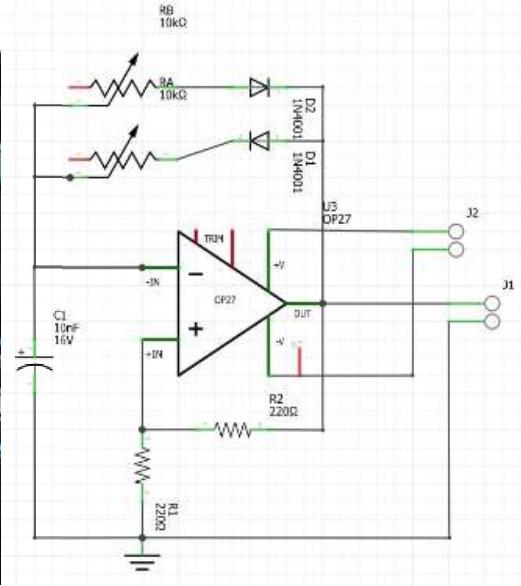


Figure 6: Schéma électrique de l'horloge à AOP en régime astable. Source: Fritz-ing

Ceci réalisé, on aboutit au dispositif suivant:

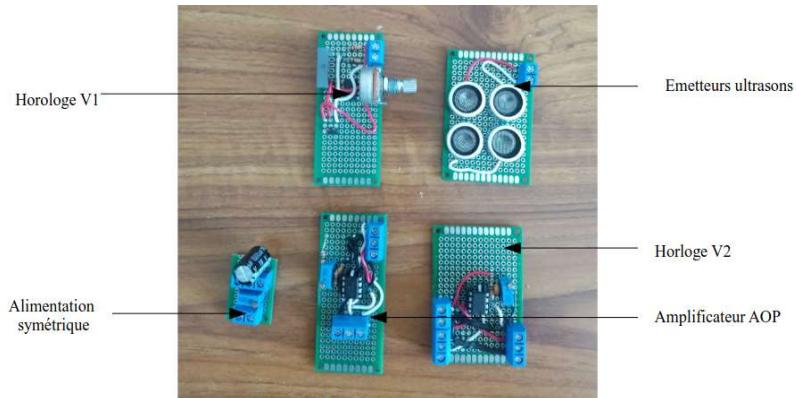


Figure 7: Prototype analogique.

Cependant, on rencontre alors le problème suivant: le signal en sortie d'un amplificateur linéaire intégré (ALI) de précision est trop faible pour pouvoir générer dans les émetteurs ultrasons un signal suffisant pour faire l'éviter des billes en polystyrène. En effet le signal en sortie d'un AOP est de +/- 7 V à 1 mA ce qui est insuffisant. Ensuite, on teste alors un mosfet et un pont en H, ce qui n'aboutit pas puis on revient à une version Arduino.

## 2.3 Version finale Arduino

Pour cela on utilise alors un code basé sur des opérateurs binaires et des fonctions élémentaires afin de permettre de modifier la fréquence. Ce code Arduino est basé sur un code de générateur créneau trouvé sur un forum [é121] [3] .

On utilise deux cartes moteur afin de manipuler les émetteurs indépendamment, et la même alimentation 24 V. En fixant les deux émetteurs à une règle on obtient le dispositif final.

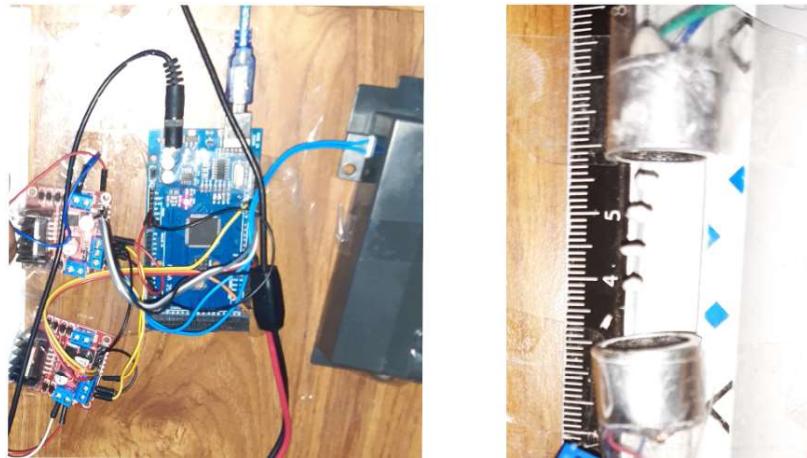


Figure 8: Version finale du dispositif pour mettre en lévitation des billes de polystyrène.

### 3 Modèle théorique

#### 3.1 Modélisation fluide: Approche perturbative

Hypothèse :

- Approximation acoustique

On se place dans le contexte de la propagation d'une onde dans l'air entraînant une perturbation par rapport à son état au repos. Ainsi on pose:

$$(1) = \begin{cases} P = P_0 + P_1 + P_2 \text{ où } P_0 \gg P_1 \gg P_2 \\ \vec{v} = \vec{v}_1 + \vec{v}_2 \text{ où } c_0 \gg \|\vec{v}_1\| \gg \|\vec{v}_2\| \\ \rho = \rho_0 + \rho_1 + \rho_2 \text{ où } \rho_0 \gg \rho_1 \gg \rho_2 \end{cases}$$

Et où  $(P_0, \vec{v}_0 = \vec{0}, \rho_0)$  correspond à l'état d'équilibre et  $(P_1, \vec{v}_1, \rho_1)$  perturbation au 1er ordre, et  $(P_2, \vec{v}_2, \rho_2)$  perturbation au 2nd ordre.

Ici  $c_0$  est la vitesse du son dans l'air.

- Evolution isentropique

$$(2) : \quad P_1 = c_0^2 \rho_1$$

- Navier-Stokes: Cas de la viscosité zéro

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho (\vec{v} \cdot \vec{\nabla}) \vec{v} + \eta (\vec{\nabla}^2 \vec{v} + \beta \vec{\nabla} (\vec{\nabla} \cdot \vec{v})) = -\vec{\nabla} P \text{ devient (pour } \eta = 0 \text{)} :$$

$$(3) : \quad \rho \frac{\partial \vec{v}}{\partial t} + \rho (\vec{v} \cdot \vec{\nabla}) \vec{v} = -\vec{\nabla} P$$

- Fluide irrotationnel

On suppose que notre fluide ne subit pas de rotation pure.

- Modèle de Gor'Kov

Hypothèse d'une petite sphère  $a \ll \lambda$  où  $a$  est le rayon de la sphère et  $\lambda$  la longueur d'onde.

#### 3.2 Champs de vitesse et de pression (Modèle théorique)

- On pose l'équation de conservation de la masse

$$(4) : \quad \vec{\nabla} \cdot (\rho \vec{v}) + \frac{\partial \rho}{\partial t} = 0$$

et au premier ordre on a

$$\rho_0 \vec{\nabla} \cdot \vec{v}_1 + \frac{\partial \rho_1}{\partial t} = 0$$

Puis on exploite (3) au 1er ordre :

$$\rho_0 \frac{\partial \vec{v}_1}{\partial t} = -\vec{\nabla} P_1 = -c_0^2 \vec{\nabla} P_1$$

Dérivons par rapport à t et

$$\frac{\partial^2 \vec{v}_1}{\partial t^2} = -\frac{c_0^2}{\rho_0} \vec{\nabla} (-\rho_0 \vec{\nabla} \cdot \vec{v}_1)$$

D'où nous obtenons l'équation d'Alembert :

$$(5) : \quad \frac{\partial^2 \vec{v}_1}{\partial t^2} = c_0^2 \Delta \vec{v}_1$$

- D'après (4) on a :

$$\frac{1}{c_0^2} \frac{\partial P_1}{\partial t} = -\rho_0 \vec{\nabla} \cdot \vec{v}_1$$

d'où, en définissant la compressibilité par:

$$\kappa_0 = \frac{1}{\rho_0 c_0^2}$$

on a

$$(6) : \quad \frac{\partial P_1}{\partial t} = -\frac{1}{\kappa_0} \vec{\nabla} \cdot \vec{v}_1$$

- Modèle simple:

Onde progressive monochromatique de vitesse selon  $\vec{e}_z$   
se propageant selon +z et -z d'après (5) :

$$\begin{aligned} \vec{v}_1 &= A \cos(wt - kz) \vec{e}_z - A \cos(wt + kz) \vec{e}_z \\ \vec{v}_1 &= 2A \sin(wt) \sin(kz) \vec{e}_z \end{aligned}$$

D'où d'après (6):

$$P_1 = 2A \rho_0 c_0 \cos(wt) \cos(kz)$$

On s'est ici appuyé sur le modèle proposé dans le sujet Mines [cMP20] [5]

### 3.3 Calcul du potentiel de la force de radiation

On s'appuie sur la thèse de Mikkel Settnes et Henrik Bruus [MS12] [2] pour essayer de comprendre l'origine de la formule de la force de radiation:

- Un calcul au 2nd ordre sur (3) en passant à la moyenne temporelle associé à l'équation de conservation de la masse (4) permet d'exprimer la perturbation de pression d'ordre 2 en fonction des perturbations d'ordre 1.

$$(7) : \quad \langle P_2 \rangle = \frac{1}{2} k_0 \langle P_1^2 \rangle + \frac{1}{2} \rho_0 \langle v_1^2 \rangle$$

- La force de radiation moyenne est donnée par :

$$\vec{F}_{rad} = \left\langle \int_{S(t)} P \vec{n} dS \right\rangle$$

D'après le thème "Pince acoustique : piégeage et manipulation d'un objet par pression de radiation d'une onde progressive." de Diego Baresh [Bar14] [4].

où  $S(t)$  surface de la sphère qui se déplace dans le temps. Pour ne pas avoir à prendre en compte ce déplacement, on l'exprime plutôt ainsi:

$$\vec{F}_{rad} = - \int \langle P_2 \rangle \vec{n} + \rho_0 \langle (\vec{n} \cdot \vec{v}_1) \vec{v}_1 \rangle dS$$

En appliquant les hypothèses de fluide irrotationnel et de petite sphère devant la longueur d'onde, on aboutit à, d'après la thèse de Mikkel Settnes et Henrik Bruus [MS12] [2] :

$$(8) : \quad U_{rad} = \frac{4}{3} \pi a^3 (f_1 * \kappa_0 / 2 \langle \rho_1^2 \rangle - f_2 \frac{3}{4} \rho_0 \langle v_1^2 \rangle)$$

avec  $f_1, f_2$  coefficients sans unité.

Avec  $a$  le rayon sphère, et  $\kappa_0$  le coefficient de compressibilité du fluide (Modèle de Gor'kov)

### 3.4 Application à notre modèle

Si on injecte notre modèle dans la partie 2)

$$\begin{cases} \vec{v}_1 = 2A \sin(\omega t) \sin(kz) \vec{e}_z \\ P_1 = 2A \rho_0 c_0 \cos(\omega t) \cos(kz) \end{cases}$$

dans l'énergie potentielle de la partie 3) et on trouve:

$$U_{rad} = B \cos(2kz) + C$$

où  $B, C$  deux constantes, ce qui nous donne

$$z_{eq,n} = constante + n \frac{\lambda}{2}$$

où on ne prends pas en compte la pesanteur.

### 3.5 Explication du phénomène de lévitation acoustique

On part alors des équations (5 et 6) pour résoudre notre problème de manière numérique. Pour cela on discrétise le domaine de temps et d'espace afin de résoudre l'équation d'Alembert (5) sur la vitesse.

Cela revient alors à résoudre la suite numérique

$$(V_{n,m})_{n,m \in [|0,Nt|]*[|0,Nz|]}$$

où n est l'indice temporel et m l'indice spatial.

Cela fait, on obtient alors la carte de vitesse suivante de la vitesse en fonction du temps et de l'espace, qui correspond bien avec le modèle d'onde stationnaire mais permet plus de flexibilité sur les conditions limites (par exemple ajouter un déphasage entre les émetteurs). On a alors le champs de vitesse pour des conditions initiales quelconques. On exploite la relation entre vitesse et pression (6) pour calculer le champs de pression et on peut ainsi calculer les moyennes temporelles de leurs carrés. On obtient alors par la formule (8) l'énergie potentielle de radiation acoustique que l'on superpose alors à l'énergie potentielle de pesanteur d'une bille pour obtenir l'énergie potentielle de notre système.

Les cartes de pression et de vitesse ci-dessous ont été générées par mon script Python nommé modélisation numérique, disponible en annexe.

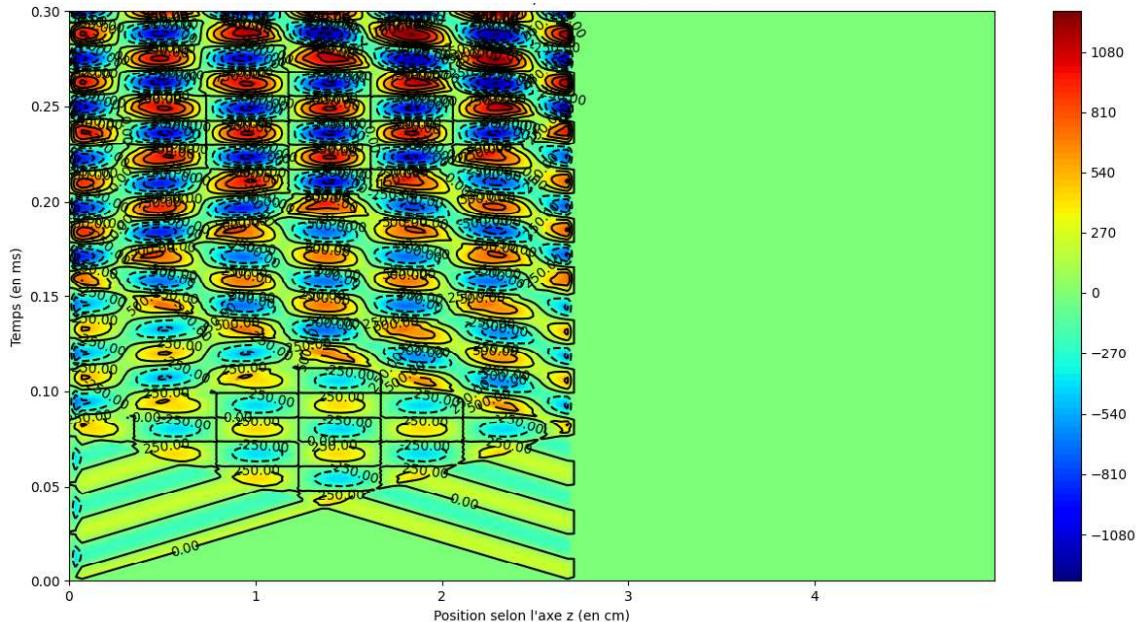


Figure 9: Carte Pression par rapport au temps et à la position à  $f=39.0$  kHz,  $h = 2.73$  cm

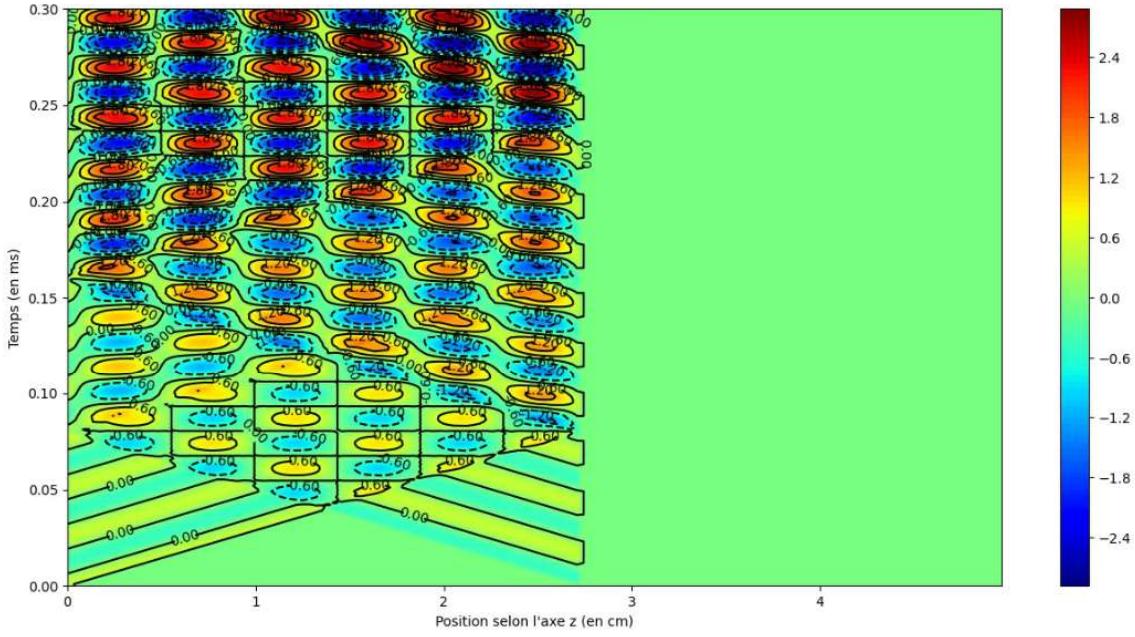


Figure 10: Carte Vitesse par rapport au temps et à la position à  $f=38.4$  kHz,  $h = 2.7$  cm

On obtient alors la courbe d'énergie potentielle suivante, et l'on remarque la présence de minimum locaux, qui correspondent à nos positions d'équilibre stables. L'allure de ce graphe est bien cohérent avec l'expérience: on présente un post-it entre les 2 émetteurs et on le promène entre eux. On remarque qu'il oscille entre certaines positions, cela correspond au franchissement d'une barrière de potentiel.

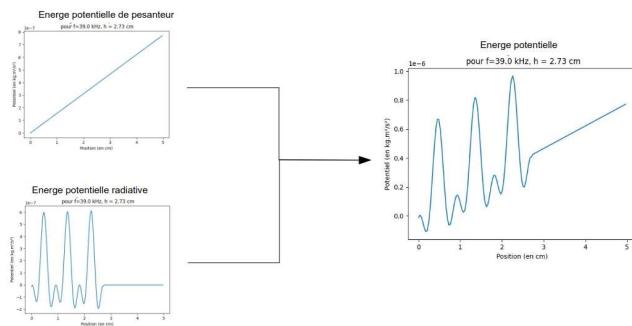


Figure 11: Explication du phénomène de lévitation acoustique.



Figure 12: Expérience avec un post-it pour confirmer la forme de l'énergie potentielle

## 4 Confrontation modèle théorique-expérimental

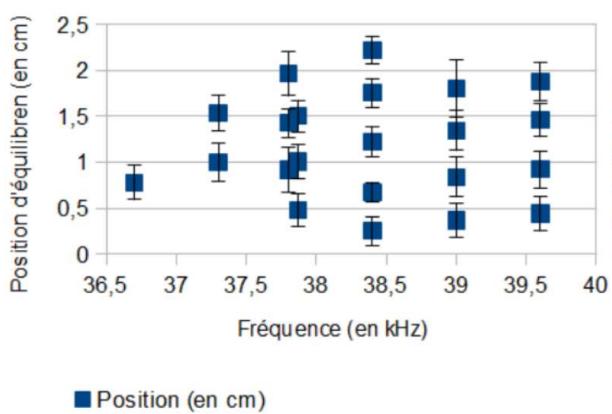


Figure 13: Mesures expérimentales.

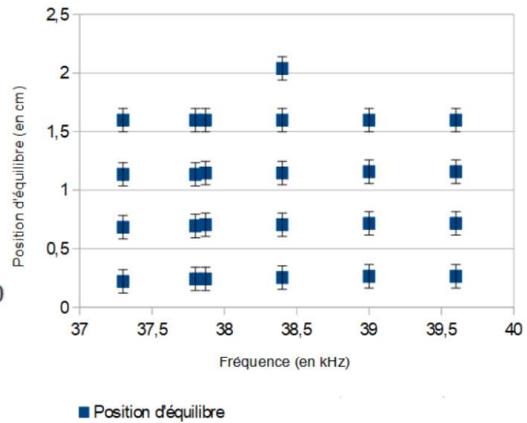


Figure 14: Mesures numériques

Ainsi notre modèle numérique nous permet de trouver des positions d'équilibre stables théoriques. Ces positions d'équilibre théorique d'environ 0,2 – 0,7 – 1,1 – 1,6 – 2,0 cm sont cohérentes avec les tendances des positions mesurées expérimentalement. Cependant il est difficile de comparer ces deux types de valeur. L'écart entre deux positions consécutives semble quand à lui être un paramètre pertinent, le modèle numérique donnant une moyenne de 0,45 cm (cf. figure 17) et le modèle expérimental de 0,50 cm (cf. figure 16). Le modèle théorique simplifié donne quand à lui  $\lambda/2 = 0,44$  cm. On obtient alors un écart relatif de 10%.

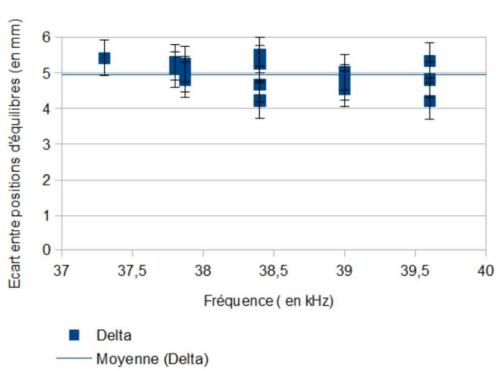


Figure 15: Graphique issu du modèle expérimental: Distance entre les positions d'équilibre en fonction de la fréquence.

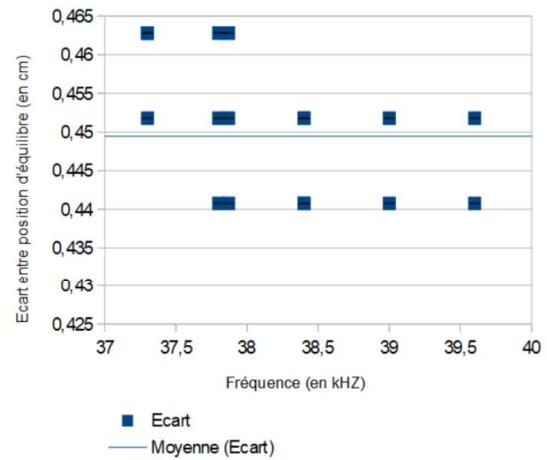


Figure 16: Graphique issu du modèle numérique: Distance entre les positions d'équilibre en fonction de la fréquence.

## 5 Conclusion et perspectives

### Conclusion:

Le résultat de la confrontation entre le modèle numérique et le modèle expérimental est satisfaisant.

Le modèle numérique semble être plus proche de l'expérience que le premier modèle théorique (ondes stationnaires).

### Perspectives:

Il reste possible de reprendre le prototype analogique afin d'amplifier le signal et d'avoir une expérience plus simplement manipulable. On pourrait aussi prendre en compte l'ordre 2 dans le modèle numérique du champs de vitesses.

On pourrait avoir recours à la Stroboscopie ou à la méthode shlieren pour permettre de visualiser les ondes acoustiques et ainsi avoir un approche plus pédagogique.

## 6 Annexes

### 6.1 Programme Python pour le modèle numérique

```
1 # Importation des bibliothèques utiles
2 from math import cos, sin, pi
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 # Fonction pour afficher un graphique
7 def graphique (X,Y, xtitre,ytitre,titre):
8     """
9         Afficher une graphique
10
11         :param X: axe des X
12         :param Y: axe des Y
13         :param xtitre: titre de l'axe des X
14         :param ytitre: titre de l'axe des Y
15         :param titre: titre du graphique
16         :return: show plot
17     """
18     plt.title(titre)
19     plt.plot(X,Y)
20     plt.xlabel(xtitre)
21     plt.ylabel(ytitre)
22     plt.show()
23
24 # Fonction pour le calcul théorique des variables
25 def Calculs_theoriques(h,f):
26     """
27         Calculs théorique des variables
28
29         :param h: distance entre les deux émetteurs
30         :param f: fréquence
31     """
32     ##Variables:
33     # déphasage entre les deux transducteurs (en rad)
34     phi = 2
35     # pulsation (en rad.s-1)
36     omega = 2 * pi * f
37
38     ## Données
39     # distance max entre les émetteurs (en m)
40     H = 5 * 10**-2
41
42     # vitesse du son dans l'air (en m.s-1)
43     c0 = 344
```

```

44 # kg.m-3, masse volumique air
45 rho0 = 1.292
46 # coefficient compressibilite air
47 kappa0 = 1/((c0**2) * rho0)
48
49 f0 = 40 *10**3
50 # vitesse des hauts parleurs de reference a f = 40 kHz
51 # (en m.s-1)
51 Am0 = 50*10**-2
52 # distance caracteristique de deplacement transducteurs
53 # (en m)
53 d0 = Am0/f0
54 # vitesse deplacement transducteurs (en m.s-1)
55 Am = d0 * f
56 Bm = Am
57
58 # rayon bille de polystyrene (en m)
59 a = 3*10**-3
60 # volume bille
61 Vp = (4/3) * pi * (a ** 3)
62 # vitesse du son dans le polystyrene (en m.s-1)
63 cp = 425
64 # masse volumique du polystyrene expanse (en kg.m-3)
65 rhoP = 14
66 kappaP = 1/((cp**2) * rhoP)
67
68 # Fonction donnant l'energie potentiel de radiation
69 def Urad(P_m, V_m):
70 """
71 Calcul de l'energie potentielle de radiation
72
73 :param P_m: Pression moyenne
74 :param V_m: Vitesse moyenne
75 :return: energie potentielle de radiation
76 """
77 f1 = 1 - (kappaP/kappa0)
78 f2 = 2*(rhoP/rho0 - 1) / (2*(rhoP/rho0) - 1)
79 return Vp * (f1 * (kappa0/2) * P_m - f2 * (3/4)*rho0*V_m)
80
81 # Fonction donnant l'energie potentielle de pesanteur
82 def Ep(z):
83 """
84 Calcul de l'energie potentielle de pesanteur
85
86 :param P_m: Altitude
87 :return: energie potentielle de pesanteur
88 """

```

```

89     M = Vp*ρp
90     # acceleration gravitationnelle (en m.s-1)
91     g = 9.81
92     return M*g*z
93
94 def Transducteur_ultrason(t,U,phi):
95 """
96 Calcul de l'energie potentielle de radiation
97
98 :param t: Variable temps
99 :param U: Variable energie potentielle de radiation
100 :param phi: Variable phase
101 :return: energie potentielle de radiation
102 """
103     return U*cos(ω*t + phi)
104
105 # Calcul champs pression + vitesse
106 # Nombre de subdivisions de H
107 Nz = 144
108 # Pas de la subdivision (en m)
109 δz = H/Nz
110 Nh = int(h/δz )
111
112 # Temps de l'experience (en s)
113 T = 0.0003
114 # Pas de la subdivision (en s)
115 δt= 10**-6
116 Nt = int(T/δt)
117
118 V = [[0 for m in range(Nz)] for n in range(Nt)]
119 P = [[0 for m in range(Nz)] for n in range(Nt)]
120 #Initialisation
121 V[0][0] = Transducteur_ultrason(0,Am,0)
122 V[0][Nh] = Transducteur_ultrason(0,Bm,ϕ)
123 V[1][0] = Transducteur_ultrason(δt,Am,0)
124 V[1][Nh] = Transducteur_ultrason(δt,Bm,ϕ)
125
126 #Calcul de V (champs vitesse) et de P (champs pression)
127 for n in range(2, Nt):
128     t = n*δt
129     V[n][0] = Transducteur_ultrason(t,Am,0)
130     V[n][Nh] = Transducteur_ultrason(t,Bm,ϕ)
131     for m in range(1,Nh):
132         V[n][m] = (((c0*δt)/(δz)) ** 2) * ( V[n-1][m+1]
133             - 2*V[n-1][m] + V[n-1][m-1]) + 2*V[n-1][m] -
134             V[n-2][m]

```

```

133     P[n][m] = P[n-1][m] + delta_t * (-1/kappa_0) * ((V[n][m] - V[n-1][m-1]) / delta_z)
134
135 #Initialisation
136 V_m, P_m = [0 for m in range(Nz)], [0 for m in range(Nz)]
137 U_rad_z = [0 for m in range(Nz)]
138 U = [0 for m in range(Nz)]
139 N = Nt - int(Nt/2)
140
141 #Calcul moyenne temporel champs vitesse/pression au
142 #carre
143 for m in range(Nz):
144     for n in range(int(Nt/2), Nt):
145         V_m[m] += (V[n][m]**2)/N
146         P_m[m] += (P[n][m]**2)/N
147         U_rad_z[m] = Urad(P_m[m], V_m[m])
148         U[m] = U_rad_z[m] + Ep(m*delta_z)
149
150 x=[i*delta_z*100 for i in range(Nz)]
151 graphique(x,U,"Position(en cm)","Potentiel(en kg.m / s )", "Energie Potentiel\npour f="+str(f/1000)+" kHz, h="+str(h*100)+" cm")
152 #graphique(x,P_m,"Position (en cm)", "Pression (en Pa)", "Graphique pression moyenne f="+str(f/1000)+" kHz, h = "+str(h*100)+" cm")
153 #graphique(x,V_m,"Position (en cm)", "Vitesse (en m.s-1)", "Graphique vitesse moyenne f="+str(f/1000)+" kHz, h = "+str(h*100)+" cm")
154 """
155 y=np.linspace(0,T*1000,Nt)
156 X,Y=np.meshgrid(x,y)
157 plt.figure(figsize=(25,15))
158
159 cf=plt.contourf(X,Y,V,100,cmap='jet')
160 graph=plt.contour(X,Y,V,10,colors='black')
161 plt.clabel(graph,inline=1,fontsize=10,fmt='%3.2f')
162 plt.colorbar(cf)
163 plt.xlabel("Position selon l'axe z (en cm)")
164 plt.ylabel("Temps (en ms)")
165 plt.title("Carte Vitesse par rapport au temps et au la position\na f="+str(f/1000)+" kHz, h = "+str(h*100)+" cm")
166 plt.show()
167 """
168

```

```
169 def main():
170     h = 2.73 * 10**-2 # distance entre les émetteurs en cm
171     f = 39 *10**3
172     Calculs_theoriques(h,f)
173     return None
174
175 main()
```

## 6.2 Programme C++ pour l'Arduino

Ce programme C++ est basé sur le code disponible dans le projet "Générateur de signaux carrés Arduino" [é121] [3]

```
1 // Masque binaire, permettant de viser la broche D11 de l'
2 // Arduino Uno
3 #define S11 B00100000
4 // Masque binaire, permettant de viser la broche D12 de l'
5 // Arduino Uno
6 #define S12 B01000000
7 // Masque binaire, permettant de viser la broche D10 de l'
8 // Arduino Uno
9 #define S10 B00010000
10 // Masque binaire, permettant de viser la broche D13 de l'
11 // Arduino UnoD10
12 #define S13 B11111111
13 float f = 40000;           // Frequence (en Hz)
14
15 #define periodeDuSignal (float)1/f*1000000 // Periode =
16     1/Frequence (en microsecondes)
17 #define tempo (float)periodeDuSignal/2        // Delai
18     correspondant a la frequence
19 float dephasage = 0;
20 #define delta (float)tempo*dephasage
21 void setup() {
22     DDRB = DDRB | S10;                      // Definit la broche D10
23     comme etant une "sortie"
24     DDRB = DDRB | S11;
25     DDRB = DDRB | S12;
26     DDRD = DDRD | S13;
27
28     analogWrite(2, 255);
29     analogWrite(3, 255);
30 }
31
32 void loop() {
33     PORTB = PORTB | S10;                    // Met la sortie D10 a l'
34     etat haut
35     PORTB = PORTB & ~S11;                  // Met la sortie D11 a l'
36     etat bas
37     _delay_us(delta);                   // Cree une pause de
38     temps delta
39     PORTB = PORTB & ~S12;
40     PORTD = PORTD | S13;
41
42     _delay_us(tempo-0.15-delta);
43     PORTB = PORTB & ~S10;
```

```

34 PORTB = PORTB | S11;
35 _delay_us(delta);
36 PORTB = PORTB | S12;
37 PORTD = PORTD & ~S13;
38 _delay_us(tempo - 0.4 - delta);
39 }
40 }
```

### 6.3 Données

Mesures expérimentales:

Fréq. <sup>1</sup>	Position 1 <sup>2</sup>	Position 2	Position 3
36,2			
36,7			
37,3			
37,8			0,915915915915916
37,87		0,480600750938673	1,00625782227785
38,4	0,252762430939227	0,675414364640884	1,22651933701657
39	0,3678963110668	0,840478564307079	1,34297108673978
39,6	0,441699604743083	0,921936758893281	1,45553359683794

<sup>1</sup> Fréquence

<sup>2</sup> Si cellule vide, pas de mesure.

Table 2: Relevé des mesures expérimentales.

Fréq.	Position 4	Position 5
36,2		
36,7		
37,3	0,996194100856327	1,53853472882969
37,8	1,42642642642643	1,96096096096096
37,87	1,50187734668335	
38,4	1,75276243093923	2,22099447513812
39	1,79760717846461	
39,6	1,87648221343874	

Table 3: Relevé des mesures expérimentales.

Mesures numériques:

Fréquence	Position d'équilibre
37,3	0,2203856749
37,3	0,6831955923
37,3	1,1349862259
37,3	1,5977961433
37,8	0,2424242424
37,8	0,694214876
37,8	1,1349862259
37,8	1,5977961433
37,87	0,2424242424
37,87	0,7052341598
37,87	1,1460055096
37,87	1,5977961433
38,4	0,2534435262
38,4	0,7052341598
38,4	1,1460055096
38,4	1,5977961433
38,4	2,0385674931
39	0,2644628099
39	0,7162534435
39	1,1570247934
39	1,5977961433
39,6	0,2644628099
39,6	0,7162534435
39,6	1,1570247934
39,6	1,5977961433

Table 4: Relevé des mesures numériques.

## References

- [Bar14] Diego Baresh. Pince acoustique : piégeage et manipulation d'un objet par pression de radiation d'une onde progressive. *HAL science*, page 208, 2014.
- [cMP20] Concours commun Mines-Ponts. La lévitation acoustique: Sujet mines physique. pages <https://www.concoursminesponts.fr/resources/Physique-PSI-2.pdf>, 2020.
- [MS12] Henrik Bruus Mikkel Settnes. Forces acting on a small particle in an acoustical field in a viscous fluid. *PHYSICAL REVIEW*, (12):<https://journals.aps.org/pre/pdf/10.1103/PhysRevE.85.016327>, 2012.
- [él21] Passion électronique. Projet : Générateur de signaux carrés arduino. pages <https://passionelectronique.fr/generateur-signal-carre-arduino-v1/>, 2021.