université de BORDEAUX

Licence 2ème année

UE Analyse Harmonique

TD Méthodes numériques

Printemps 2019

Enseignants : Rodolphe Boisgard, Baptiste Fabre, Lionel Truflandier

Table des matières

TD 1

Champs

A Champs scalaires

A.1 Présentation des outils

A.1.1 Exemple de représentation d'un champ de température

Soit un champ de température $T(x, y)$ défini dans le plan (x, y) sur le domaine espace $\mathcal{D} = [-5, 5] \times [-5, 5]$ par l'expression

$$T(x, y) = T_b + T_c \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{\sigma^2}\right) + T_f \exp\left(-\frac{(x - x_f)^2 + (y - y_f)^2}{\sigma^2}\right)$$

où $T_b = 20°\text{C}$ est la température de base, $T_c = 100°\text{C}$ est la température du point « chaud » et $T_c = -30°\text{C}$ est la température du point « froid ». Les points $(x_c, y_c) = (-2, 2)$ et $(x_f, y_f) = (2, -2)$ correspondent à la localisation des points « chaud » et « froid » respectivement et $\sigma = 2$ indique l'étalement de chacun de ces points sources.

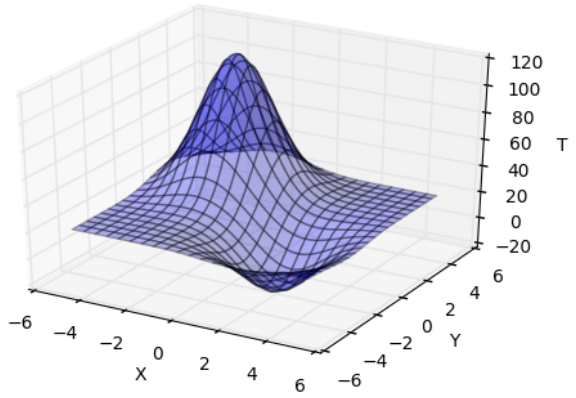


FIGURE 1.1 – *Champ de température $T(x, y)$: vue en 3D*

meshgrid : De même que pour représenter une fonction $f(x)$ d'une seule variable x, on discrétise l'intervalle d'étude $[a, b]$ avec la fonction `linspace(a,b,N)` où N est le nombre de points de la discrétisation, on doit de la même façon discrétiser le domaine d'espace $\mathcal{D}$ du plan (x, y) pour représenter la fonction $T(x, y)$.

Supposons que l'on veuille discrétiser le domaine $\mathcal{D} = [-5, 5] \times [-5, 5]$ avec 5 points suivant x et 3 points suivant y.

- Une première méthode « pédestre » consiste à définir 2 intervalles en x et y à l'aide de la fonction `linspace(a,b,N)`, puis à créer une matrice T contenant 5 colonnes et 3 lignes et enfin à la remplir avec la fonction $T(x, y)$ définie plus haut. Voici une portion code réalisant cette opération

```
def temp(x,y):
    return Tb + Tc*np.exp(-((x-xc)**2+(y-yc)**2)/sigma**2) \
            +Tf*np.exp(-((x-xf)**2+(y-yf)**2)/sigma**2)

xmin,xmax,ymin,ymax = -5,5,-5,5
xc,yc,xf,yf = -2,2,2,-2
sigma = 2
Tb,Tc,Tf = 20,100,-40
Nx,Ny = 5,3
x = np.linspace(xmin,xmax,Nx)
y = np.linspace(ymin,ymax,Ny)
Tp = np.zeros((Ny,Nx))  # crée une matrice de Ny lignes et Nx colonnes

for i in range(Nx):
    for j in range(Ny):
        Tp[j,i] = temp(x[i],y[j])
```

Ce n'est malheureusement pas une bonne idée car les boucles `for` sont particulièrement lente dans un langage de script comme python.

- Une méthode beaucoup efficace consiste à utiliser la fonction `meshgrid()` qui génère 2 matrices à partir de 2 tableaux 1D créés par exemple avec la fonction `linspace()`. On peut comprendre sont fonctionnement en testant les lignes de code suivantes :

```
x = np.linspace(-5,5,5)
print(x)
y = np.linspace(-5,5,3)
print(y)
```

qui donne

```
[-5.  -2.5  0.   2.5  5. ]
[-5.  0.  5.]
```

puis

```
X,Y = np.meshgrid(x,y)
print(X)
print(Y)
```

qui donne

```
[[-5.  -2.5  0.   2.5  5. ]
 [-5.  -2.5  0.   2.5  5. ]
 [-5.  -2.5  0.   2.5  5. ]]
[[-5. -5. -5. -5. -5.]
 [ 0.  0.  0.  0.  0.]
 [ 5.  5.  5.  5.  5.]]
```

On peut ainsi calculer directement le champ de température en exploitant la rapidité des calculs par tableaux

```
T = Tb + Tc*np.exp(-((X-xc)**2+(Y-yc)**2)/sigma**2) \
            +Tf*np.exp(-((X-xf)**2+(Y-yf)**2)/sigma**2)
```

Vérifiez que `Tp` et `T` ont bien les mêmes valeurs.

contour : Une fois le champ discrétisé dans le domaine choisi, on peut choisir de représenter ce champ par des courbes isovaleurs à la manière des courbes de niveau des cartes géographiques. Pour cela on utilisera la fonction `contour()` de la bibliothèque `pylab` ainsi que la fonction `clabel()` qui permet d'enrichir l'information sur les niveaux représentés.

Voici la portion de code réalisant la sortie de la figure **??**

```
CS = plt.contour(X,Y,T,16)
plt.clabel(CS, fontsize=10, fmt='%1.1f',)
plt.axis('equal')   # assure une même échelle en x et en y
plt.savefig("contours.png")   # si on veut sauver la figure
plt.show()
```

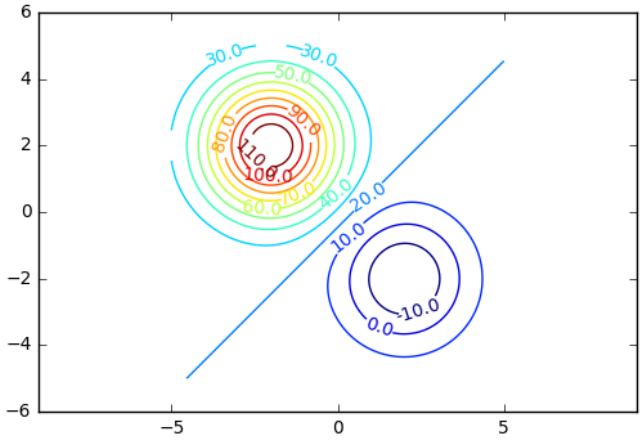


FIGURE 1.2 – *Représentation du champ de température en niveau d'isotherme*

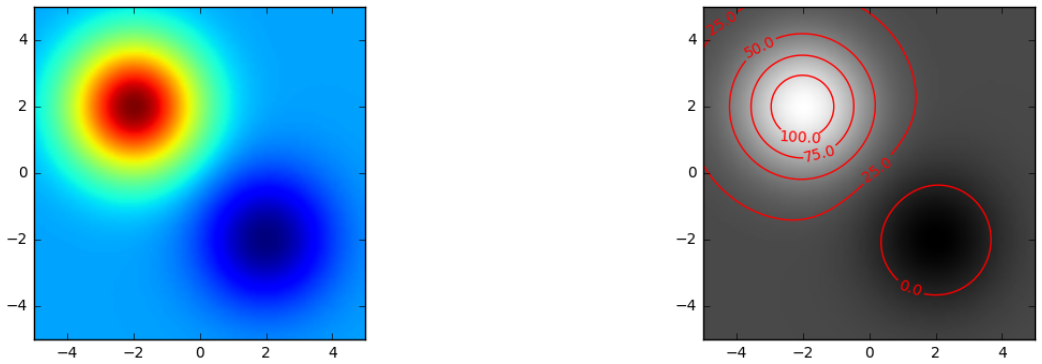
imshow : On peut aussi vouloir représenter le champ de température en niveau de couleur grâce à la fonction `imshow()` ce qui donne

```
im = plt.imshow(T, origin='lower', extent=(xmin,xmax,ymin,ymax))
plt.show()
```

Il faut rajouter l'option `origin='lower'` car sinon par défaut l'origine est placé en haut à gauche et l'option `extent=(xmin,xmax,ymin,ymax)` pour afficher les unités en x et y en unités physiques et non en nombre de pixels.

On peut aussi choisir une échelle de couleur en niveaux de gris et même combiner la représentation en niveau de couleur avec les courbes isovaleurs.

```
im = plt.imshow(T, interpolation='bilinear', origin='lower',
                cmap=plt.cm.gray, extent=(xmin,xmax,ymin,ymax))
CS = plt.contour(X,Y,T,5,colors='r')
plt.clabel(CS, fontsize=10, fmt='%1.1f')
plt.show()
```



(a) *Échelle par défaut* (b) *Niveaux de gris combiné avec les courbes isovaleurs*

FIGURE 1.3 – *Niveaux de couleurs*

A.1.2 Intégration numérique

Nous aurons besoin d'effectuer des intégrations numériques pour les calculs de potentiels et de champ en électromagnétisme. Pour cela nous allons nous familiariser ici avec l'utilisation de la fonction `quad()` de la bibliothèque `scipy.integrate` qui réalise cette opération, reportant à plus tard les détails de l'algorithme sous-jacent.

Soit la fonction $f(x) = x^3 + x$. Son intégrale évaluée analytiquement donne

$$\int_0^1 f(x)\, dx = \left[\frac{x^4}{4} + \frac{x^2}{2}\right]_0^1 = \frac{1}{4} + \frac{1}{2} = \frac{3}{4} = 0.75$$

Voici le code réalisant l'intégration numérique de cette fonction :

```
from scipy.integrate import quad
def f(x):
    return x**3+x

res, err = quad(f,0,1)
print(res)
```

ce qui donne bien le résultat attendu de 0.75.

Remarque : si nécessaire on peut aussi récupérer le paramètre `err` qui indique la précision sur le résultat retourné.

A.2 Orbitales atomiques

En physique microscopique on décrit une particule (comme un électron dans un atome) à l'aide d'une fonction d'onde $\psi(\mathbf{r}, t)$ à valeur complexe dont le carré du module représente la densité de probabilité de présence de cette particule au point $\mathbf{r}$ de l'espace à chaque instant t

$$p(\mathbf{r}, t) = |\psi(\mathbf{r}, t)|^2$$

On appelle état stationnaire un état dont la densité de probabilité est indépendante du temps.

Dans le cas de l'électron de l'atome d'hydrogène, on montre que la fonction d'onde des états stationnaires s'écrit en coordonnées sphériques (centrées sur le noyau) [1]

$$\psi(r, \theta, \varphi) = R_{n,l}(r)\, Y_l^m(\theta, \varphi)$$

où n, l et m sont 3 nombres entiers qui vérifient :

$$\begin{cases} n \in \mathbb{N}^* \\ 0 \leq l < n \\ -l \leq m \leq l \end{cases}$$

On s'intéresse ici à la représentation des premiers niveaux d'énergie stationnaires de l'atome d'hydrogène. Voici les expressions analytiques de ces premiers niveaux

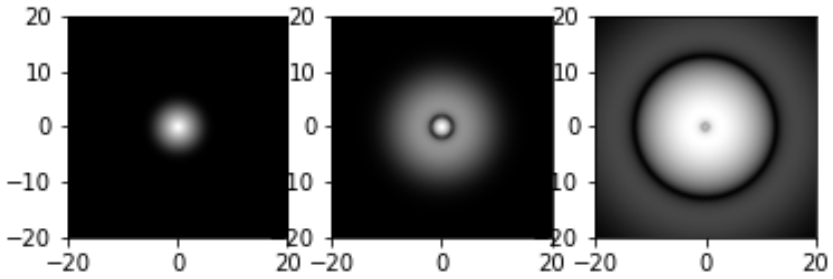
	partie angulaire	**partie radiale**
$1s$	$\frac{1}{2\sqrt{\pi}}$	$R_{10}(r) = \frac{2}{a_0^{3/2}} e^{-r/a_0}$
$2s$	$\frac{1}{2\sqrt{\pi}}$	$R_{20}(r) = \frac{1}{a_0^{3/2}} \frac{1}{\sqrt{2}} \left(1 - \frac{r}{2a_0}\right) e^{-r/2a_0}$
$3s$	$\frac{1}{2\sqrt{\pi}}$	$R_{30}(r) = \frac{1}{a_0^{3/2}} \frac{2}{\sqrt{3}} \left(1 - \frac{2r}{a_0} + \frac{4}{27}\left(\frac{r}{a_0}\right)^2\right) e^{-r/3a_0}$
$2p_x$	$\frac{1}{2}\left(\frac{3}{\pi}\right)^{1/2} \frac{x}{r}$	$R_{21}(r) = \frac{1}{a_0^{3/2}} \frac{1}{2\sqrt{6}} \frac{r}{a_0} e^{-r/2a_0}$
$2p_z$	$\frac{1}{2}\left(\frac{3}{\pi}\right)^{1/2} \frac{z}{r}$	$R_{21}(r)$
$3d_{xz}$	$\frac{1}{2}\left(\frac{15}{\pi}\right)^{1/2} \frac{xz}{r^2}$	$R_{32}(r) = \frac{1}{a_0^{3/2}} \frac{4}{81\sqrt{30}} \left(\frac{r}{a_0}\right)^2 e^{-r/3a_0}$
$3d_{z^2}$	$\frac{1}{4}\left(\frac{5}{\pi}\right)^{1/2} \frac{2z^2 - x^2 - y^2}{r^2}$	$R_{32}(r)$

TABLE 1.1 – *Expression analytiques des premiers niveaux d'énergie de l'atome d'hydrogène*

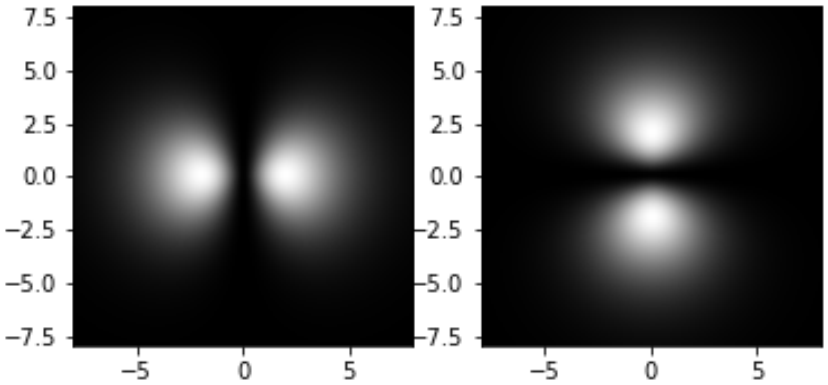
Exercice n° 1 *Orbitales s*

1. Tracer les représentations radiales des 3 premières orbitales s (R_{10}, R_{20} et R_{30}) sur un même graphe. On travaillera en unités atomiques pour lesquels $a_0 = 1$
2. Tracer les densités de probabilité de ces 3 orbitales en niveaux de gris. On choisira une représentation en échelle log de la densité de probabilité afin de permettre la visualisation des régions où la densité est presque nulle (nœuds de la partie radiale).

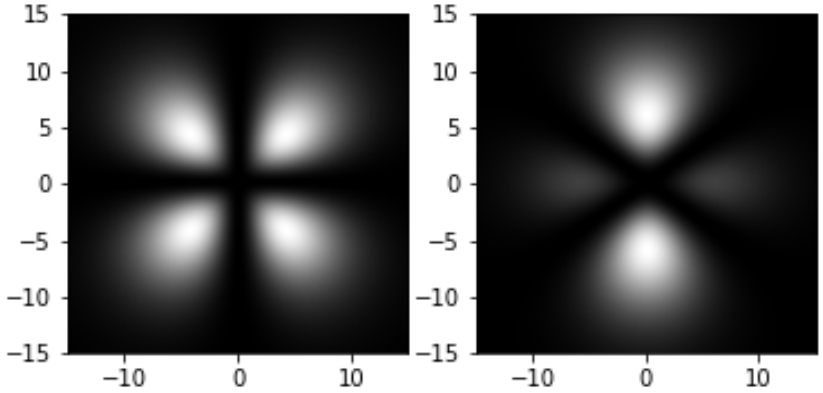
1. Il existe un 4ème nombre quantique associé au spin de l'électron, nous n'en parlerons pas ici...



(a) *Orbitales s*



(b) *Orbitales p*



(c) *Orbitales d*

Exercice n° 2 *Orbitales p*

Tracer les densités de probabilité des orbitales p_x et p_z dans le plan (x, z).

Exercice n° 3 *Orbitales d*

Tracer les densités de probabilité des orbitales $3d_{xz}$ et $3d_{z^2}$dans le plan (x, z).

A.3 Potentiels électrostatiques

On rappelle que le potentiel électrostatique ϕ en un point M de l'espace créé par un ensemble de N charges $\{q_i\}_{i=1..N}$ localisées aux points $\{A_i\}_{i=1..N}$ de l'espace a pour expression

$$\phi(M) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^{N} \frac{q_i}{A_i M}$$

Exercice n° 4 *Quadripôle*

On considère un ensemble de 4 charges placées aux sommets d'un carré de côté a, de charges identiques en valeur absolue et de signes opposées à leur deux plus proches voisins.

Dans le plan contenant ces 4 charges, tracer les courbes isopotentielles électriques créées par cette distribution. (cf ; figure **??**)

Remarque : on prendra un système d'unité tel que pour tout i

$$\left| \frac{q_i}{4\pi\epsilon_0} \right| = 1$$

et on limitera les valeurs des potentiels à l'intervalle $[-2, 2]$.

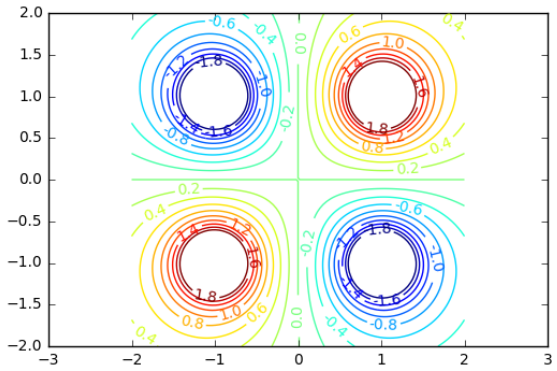


FIGURE 1.5 – *Courbes isopotentielles d'une distribution quadripolaire*

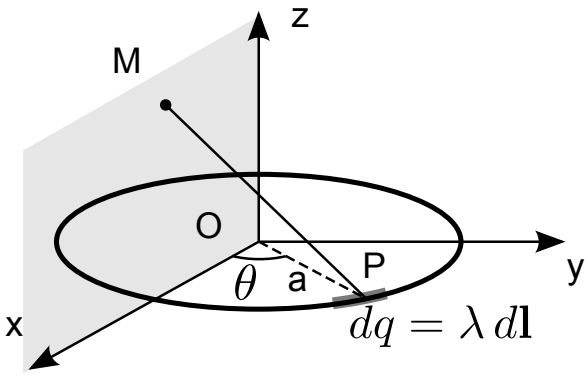
Exercice n° 5 *Anneau uniformément chargé*

On considère un anneau de rayon a sur lequel a été répartie uniformément une charge q. On notera λ la densité linéique de charge associée.

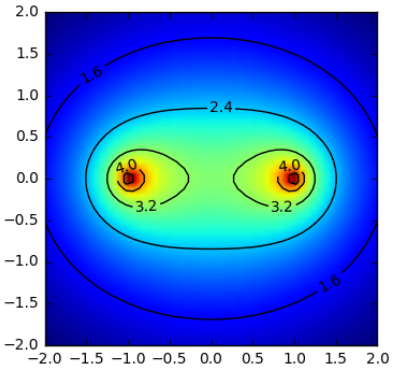
1. Montrer tout d'abord que le potentiel ϕ en un point M du plan méridien (x, z) s'écrit

$$\phi(x, z) = \frac{\lambda}{4\pi\epsilon_0} 2 \int_0^{\pi} \frac{a\, d\theta}{\left[x^2 + z^2 + a^2 - 2ax\cos\theta\right]^{1/2}}$$

2. En utilisant la fonction `quad`, définir une fonction python `phi(x,z)` permettant de calculer le potentiel électrostatique pour tout couple de valeur (x, z).
   On prendra là encore un système d'unité tel que $a = 1$ et $\frac{\lambda}{4\pi\epsilon_0} = 1$
3. Utiliser cette fonction `phi(x,z)` pour tracer les courbes isopotentielles dans le plan méridien (x, z).
   Indication : on « vectorisera » au préalable la fonction `phi` à l'aide de la fonction `vectorize` de la bibliothèque numpy.
4. Représenter enfin sur un même diagramme l'intensité du potentiel en code couleur et les courbes isopotentielles comme indiqué sur la figure **??**.



(a) *Géométrie*



(b) *Cartographie du potentiel électrostatique.*

FIGURE 1.6 – *Anneau uniformément chargé*

B Champs vectoriels

B.1 Présentation du problème

Pour un champ vectoriel $\mathbf{V}$ on rappelle la définition des lignes de champ : il s'agit des courbes de l'espace en tout point tangentes au champ $\mathbf{V}$. Ceci se traduit mathématiquement par la relation

$$d\mathbf{M} = \mathbf{V}\, du$$

où $d\mathbf{M}$ est le vecteur tangent à la courbe au point $M(x, y)$ et du est le coefficient de proportionnalité (infinitésimal)

Tracer une ligne de champ à un instant donné revient à résoudre (ici numériquement) le système d'équations différentielles

$$\begin{cases} \frac{dx}{du} = & \alpha V_x(x, y, t) \\ \frac{dy}{du} = & \alpha V_y(x, y, t) \end{cases}$$

avec les conditions initiales $x(u = 0) = x_0$ et $y(u = 0) = y_0$. On supposera dans cette partie que le champ est indépendant du temps.

B.2 Champ de vitesse

Exercice n° 6 *Cellule de Taylor*

En 1934, le physicien britannique G.I. Taylor [2] proposa un dispositif expérimental permettant d'observer la déformation d'une gouttelette soumise à des contraintes visqueuses au sein d'un fluide dans le but d'étudier les mécanismes de formation des émulsions [3]. Son dispositif représenté sur la figure **??**a est constitué de 4 rouleaux en rotation comme indiqué par les flèches. Taylor établit que le champ de vitesse $\mathbf{v}(\mathbf{r}) = v_x(x, y)\,\hat{\mathbf{e}}_x + v_y(x, y)\,\hat{\mathbf{e}}_y$ entre les rouleaux est de la forme

$$\begin{cases} v_x(x, y) & = +ax \\ v_y(x, y) & = -ay \end{cases} \tag{1.1}$$

où a est une constante (positive) déterminée expérimentalement.

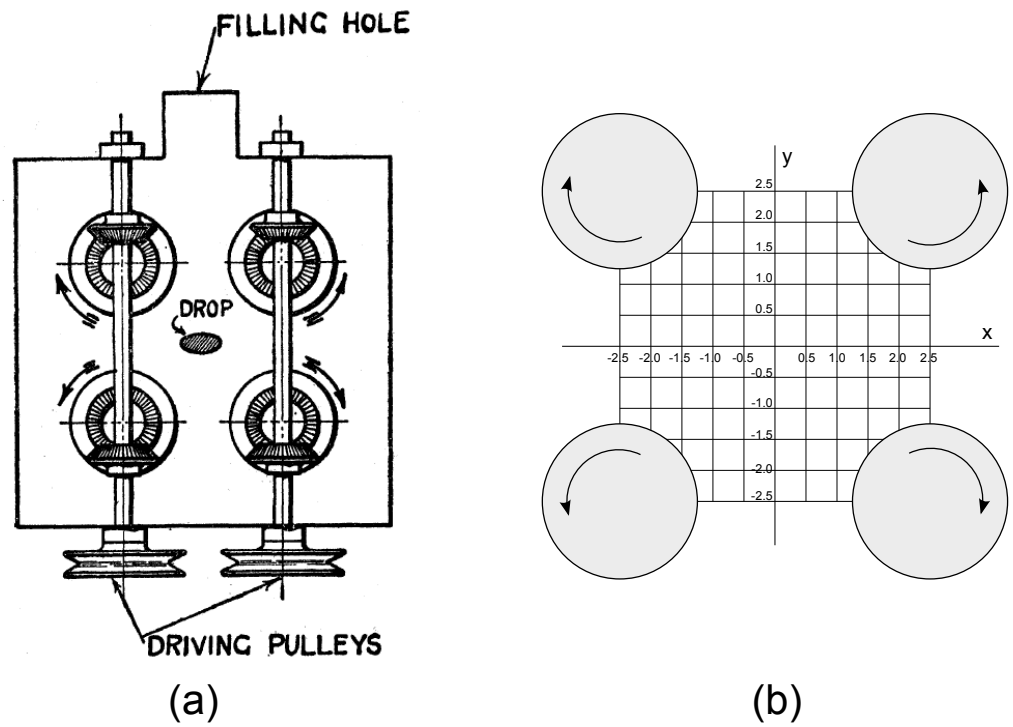
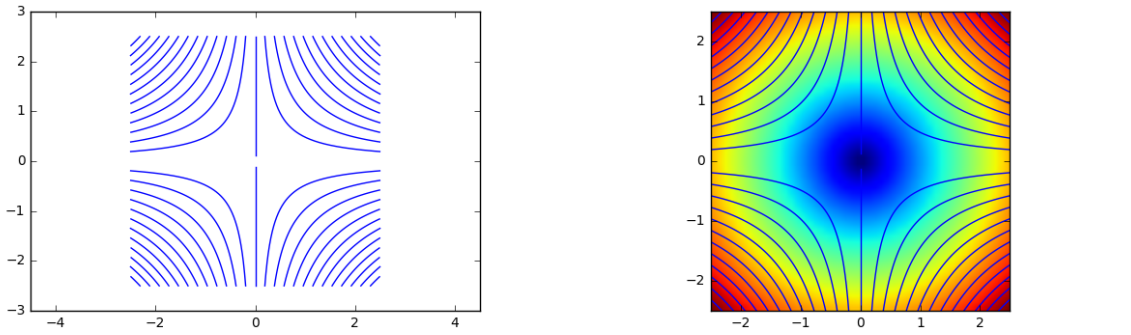


FIGURE 1.7 – *Cellule de Taylor à 4 rouleaux : (a) schéma original du dispositif expérimental, (b) schéma simplifié et représentation des lignes permettant le suivi des particules tests.*

2. *« The formation of emulsions in definable fields of flow »* Proc. R. Soc. Lond. A-1934-Taylor-501-23

3. Une émulsion est un mélange de 2 fluides non miscibles dans lequel un des constituants forme de très petites gouttes au sein de l'autre constituant. Exemple : la mayonnaise, où on réalise une émulsion d'huile dans l'eau.

1. Intégrer numériquement ce système d'équations en prenant $a = 1$ et tracer les lignes de champ de vitesse.
2. En calculant la norme du champ, représenter sur un même diagramme l'intensité du champ de vitesse en code couleur et les lignes de courant du champ de vitesse.



(a) *Lignes de courant* (b) *Lignes de courant et intensité du champ de vitesse*

FIGURE 1.8 – *Champ de vitesse*

Remarque : On peut aussi utiliser directement la fonction `streamplot()` de matplotlib qui permet de représenter des portions de ligne de champ après maillage du domaine d'espace choisi. Voici un exemple de code qui permet de représenter les lignes de courant dans la cellule de Taylor auquel on a adjoint une échelle de couleur associée à l'intensité du champ de vitesse.

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(xmin,xmax,50)
y = np.linspace(ymin,ymax,50)
X,Y = np.meshgrid(x,y)
Vx = X
Vy = -Y
Vnorme = np.sqrt(X**2+Y**2)

fig0, ax0 = plt.subplots()
strm = ax0.streamplot(X, Y, Vx, Vy, color=Vnorme, linewidth=2, cmap=plt.cm.Reds)
fig0.colorbar(strm.lines)
plt.axis('equal')
plt.show()
```

La sortie graphique de ce code est représentée sur la figure **??**

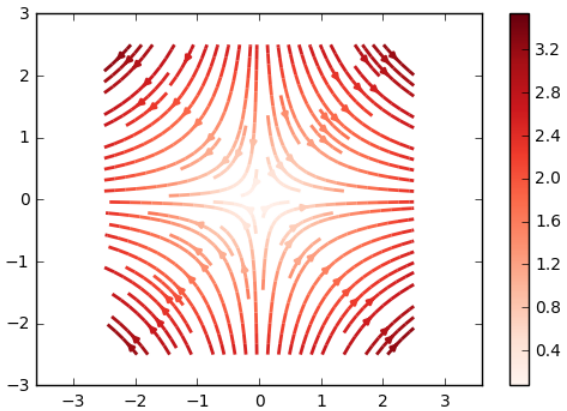


FIGURE 1.9 – *Utilisation de la fonction* `streamplot()`

B.3 Champs en électromagnétisme

B.3.1 Champs électriques

On rappelle que le champ électrique $\mathbf{E}$ en un point M de l'espace créé par une distribution de charges $\{q_i\}_{i=1..N}$ localisées aux points $\{A_i\}_{i=1..N}$ de l'espace s'écrit

$$\mathbf{E}(M) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^{N} \frac{q_i \, \mathbf{A_i M}}{\|A_i M\|^3}$$

où $\mathbf{A_i M}$ est le vecteur issu du point A_i et pointant en M.

Exercice n° 7 *Cas de deux charges de même signe :*

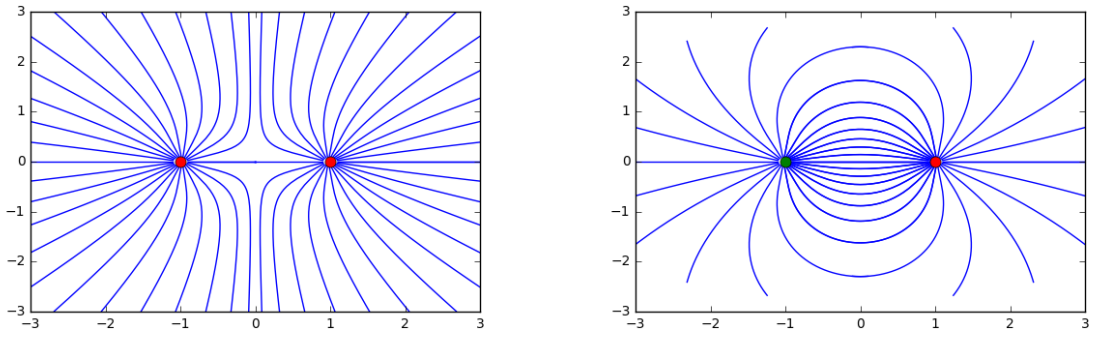
On considère 2 charges électriques q_A et q_B localisées aux points A et B de coordonnées (x_A, y_A) et (x_B, y_B). On prendra par exemple dans les calculs $(x_A, y_A) = (1, 0)$ et $(x_B, y_B) = (-1, 0)$.

1. Établir que les lignes de champs sont solutions du système d'équations différentielles couplées

$$\begin{aligned} \frac{dx}{du} &= \alpha \left[\frac{q_A (x - x_A)}{r_A^3} + \frac{q_B (x - x_B)}{r_B^3} \right] \\ \frac{dy}{du} &= \alpha \left[\frac{q_A (y - y_A)}{r_A^3} + \frac{q_B (y - y_B)}{r_B^3} \right] \end{aligned}$$

   où $r_A = \Big((x - x_A)^2 + (y - y_A)\Big)^{1/2}$ et $r_B = \Big((x - x_B)^2 + (y - y_B)\Big)^{1/2}$

2. Choisir un ensemble de conditions initiales autour de chacune des charges et tracer les lignes de champ correspondantes. On prendra $q_A = q_B = 1$ et $\alpha = 1$ et on prendra soin de rajouter un terme $r_0 = 0.1$ dans le calcul de r_A et r_B pour éviter les divergences dans le calcul des champs lorsqu'on s'approche des charges.



(a) *Charges identiques* (b) *Charges opposées*

FIGURE 1.10 – *Lignes de champs par intégration*

Exercice n° 8 *Cas de deux charges de signes opposés :*

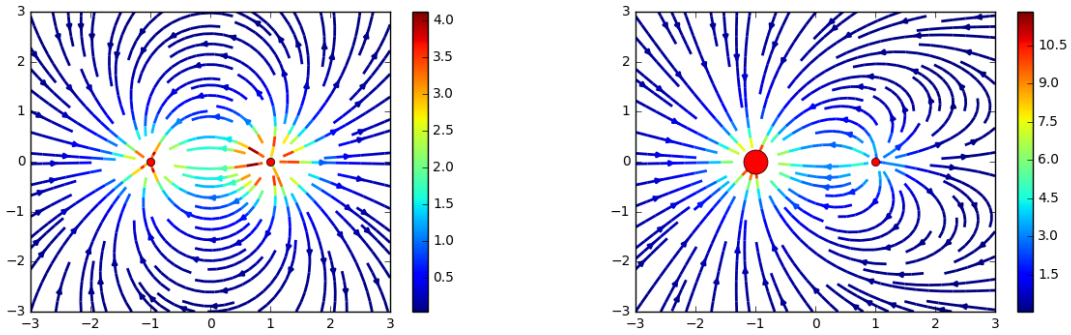
1. Reprendre l'exercice précédent en prenant cette fois-ci $q_A = 1$ et $q_B = -1$
2. Comparer votre résultat avec celui donné par l'utilisation de la fonction `streamplot()`.

B.3.2 Champ magnétiques

On rappelle l'expression du champ magnétique B en un point M de l'espace engendré par une distribution de courant d'intensité i donnée par la formule de Biot et Savart

$$\mathbf{B}(M) = \frac{\mu_0}{4\pi} \int \frac{i \, d\mathbf{l} \wedge \mathbf{AM}}{\|\mathbf{AM}\|}$$

où A est un point de la distribution parcouru par un courant de direction indiqué par le vecteur élémentaire $d\mathbf{l}$.



(a) q_A =1 *et* $q_B = -1$ (b) q_A =1 *et* $q_B = -3$

FIGURE 1.11 – *Utilisation de* `streamplot()`

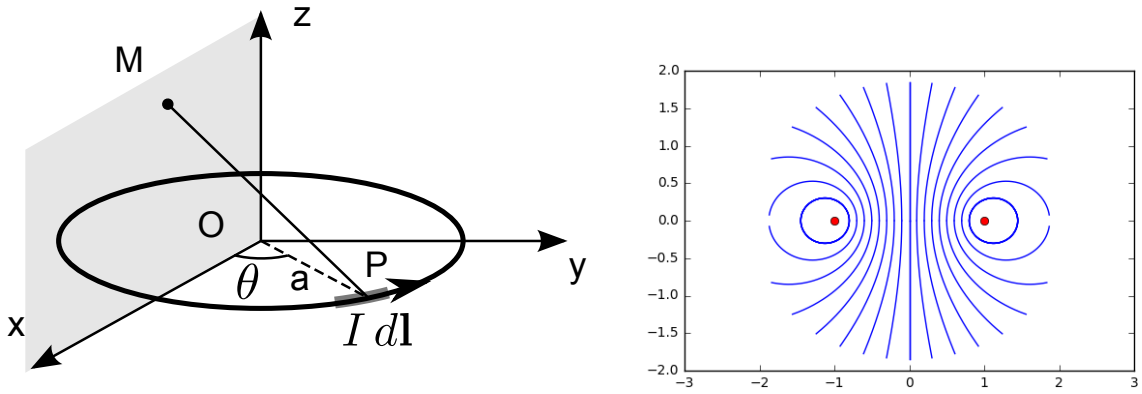
Exercice n° 9 *Champ magnétique engendré par une spire*

On considère une spire de rayon a parcourue par un courant d'intensité i. Compte tenu de la symétrique axiale du problème on peut se contenter de représenter les lignes de champ magnétique dans un plan méridien, ici le plan (x, z) (cf. figure **??**)

1. Montrer tout d'abord que le champ magnétique a pour composantes dans ce plan

$$B_x = \frac{\mu_0 i}{4\pi} 2a \int_0^\pi \frac{z \cos\theta \, d\theta}{\left[x^2 + z^2 + a^2 - 2ax\cos\theta\right]^{3/2}}$$
$$B_z = \frac{\mu_0 i}{4\pi} 2a \int_0^\pi \frac{a - x \cos\theta \, d\theta}{\left[x^2 + z^2 + a^2 - 2ax\cos\theta\right]^{3/2}}$$

2. En utilisant la fonction `quad`, définir deux fonctions python `Bx(x,z)` et `Bz(x,z)` permettant de calculer les composantes du champ magnétique en tout point (x, z) du plan méridien.
3. Exploiter alors ces deux fonctions pour calculer les lignes de champ magnétique au voisinage de la spire. Indications : on prendra comme points de départ des points situés dans le plan de la spire (donc en $z = 0$) et on exploitera la symétrie du problème pour ne calculer que les points des lignes de champ situés dans le premier quartier du plan (x, z).



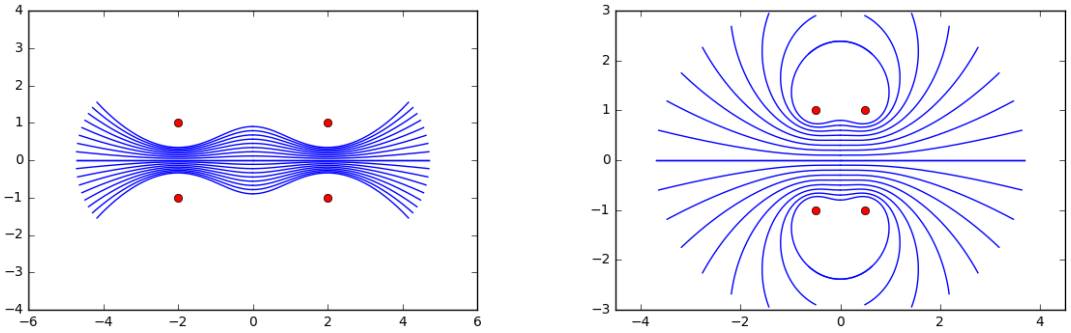
(a) *Géométrie* (b) *Lignes de champ au voisinage de la spire*

FIGURE 1.12 – *Champ magnétique créé par une spire*

Exercice n° 10 *Champ magnétique engendré par un solénoïde (mini-projet)*

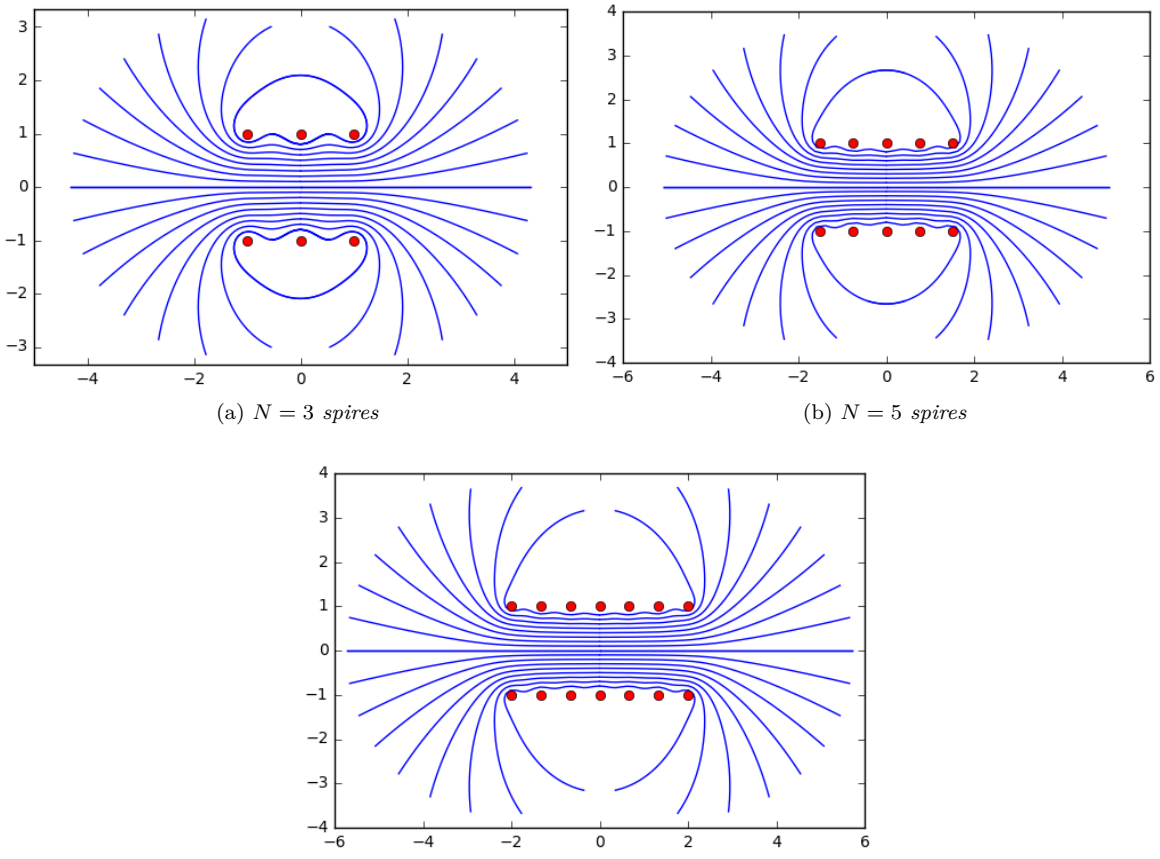
En reprenant les calculs du champ magnétique créé par une spire, calculer les lignes de champ d'un ensemble de N spires (solénoïde).

1. On étudiera tout d'abord le cas de deux spires ($N = 2$). On retrouvera en particulier le résultat (bien connu ?) du montage dit de « Helmholtz » où les deux spires sont séparées d'une distance égale au rayon des spire et où l'on montre que le champ est presque uniforme entre les deux spires.
2. Tracer enfin les lignes de champ magnétique pour $N = 3$, $N = 5$ et $N = 7$.



(a) *Pour une distance* $D = 4a$ (b) *Pour une distance* $D = a$

FIGURE 1.13 – *Montage d'Helmholtz*



(a) $N = 3$ *spires* (b) $N = 5$ *spires*

(c) $N = 7$ *spires*

FIGURE 1.14 – *Solénoïdes*

TD 2

Ondes

A Cinématique des ondes

On se propose de réaliser quelques animations qui illustrent les phénomènes de propagation d'ondes. Ce sera l'occasion d'illustrer le principe de superposition d'ondes monochromatiques pour réaliser des paquets d'ondes et ainsi de mettre en évidence la différence entre les notions de vitesse de phase et de vitesse de groupe.

A.1 Ondes monochromatiques

Exercice n° 11 *Outils pour animation*

On va illustrer la réalisation d'une animation sur l'exemple de la trajectoire parabolique d'une balle dans un champ de pesanteur uniforme. Le principe d'une animation étant l'affichage à intervalle de temps régulier d'une série d'images, nous créons tout d'abord les images au format `pdf`, puis en utilisant la possibilité de lancer en python des programmes, nous appellerons le programme `convert` (qui appartient au « couteau suisse » ImageMagick) pour confectionner un gif animé. Voici le listing du programme

```
"""
Modèle de création de gif animé
trajectoire d'un point matériel dans un champ de pesanteur
"""
import numpy as np
from numpy import pi
import pylab as plt
import os

#paramètres physiques
g = 10
v0 = 10
alpha = pi/4

#paramètres de plots
Nframes = 20   # nbre d'images
tini = 0
tfin = 1.4
tpas = (tfin-tini)/Nframes
Xmin, Xmax, Ymin, Ymax = 0, 12, 0, 3    # définition du cadre

# Construction d'une série d'images
for n in range(Nframes):
    t = tini + n*tpas
    y = -1/2*g*t**2 + v0*np.sin(alpha)*t
    x = v0*np.cos(alpha)*t
    plt.plot(x, y, 'o', color='b')
    if n == (Nframes-1):
        plt.text(6, 2, "Boum !", fontsize=20)
    plt.axis([Xmin, Xmax, Ymin, Ymax])
    filename = 'fichierTemp'+str('%02d' %n)+'.pdf'
```

```
    plt.savefig(filename)
    print('Nplot = ',  n)  # permet de suivre la progression
    plt.clf()   # nettoyage du plot

# assemblage des images dans une animation à l'aide de la fonction convert d'ImageMagick
cmd = 'convert -delay 50 -loop 0 fichierTemp*.pdf TrajectoireBoulet.gif'
os.system(cmd)
os.system('rm fichierTemp*.pdf')  # destruction des fichiers temporaires (utiliser del sous Windows)
print("C'est fini !")
```

Le programme `convert` reçoit les options : `-delay` intervalle de temps entre 2 images en millisecondes, `-loop` nombre de fois où la séquence est jouée (0 pour un nombre infini).

Recopier ce programme et testez-le.

Exercice n° 12 *Ondes progressives, ondes stationnaires*

1. Rappeler l'expression d'une onde plane monochromatique progressive $\psi(x,t)$. Qu'appelle-t-on période, longueur d'onde et vitesse de phase ? Exprimer $\psi(x,t)$ en fonction de la période T et de la longueur d'onde λ.
2. Tracer $\psi(x,t)$ en fonction de x pour $t = 0$. On choisira des unités adaptées. Faire varier la longueur d'onde et vérifier avec votre définition.
3. Tracer $\psi(x,t)$ en fonction de x pour différentes valeurs successives de t (et suffisamment proches de $t = 0$). On réalisera une animation sur le modèle de l'exercice précédent.
4. Reprendre le travail précédent dans le cas d'une onde régressive.
5. Représenter l'évolution temporelle d'une onde en réflexion parfaite. Localiser les noeuds et les ventres de l'onde stationnaire ainsi formée. Comparer avec l'onde progressive.

A.2 Paquet d'ondes

Exercice n° 13 *Paquet d'ondes (1) : superposition de 2 ondes*

On considère la superposition de deux ondes planes monochromatiques $\psi_1(x,t)$ et $\psi_2(x,t)$ de pulsations voisines. Soit $\psi(x,t) = \psi_1(x,t) + \psi_2(x,t)$.

1. Exprimer $\psi(x,t)$ en fonction des fréquences f_1 et f_2 des vitesses de phases v_1 et v_2 des ondes $\psi_1(x,t)$ et $\psi_2(x,t)$. Tracer tout d'abord $\psi(x,t)$ en fonction de x à des instants successifs et voisins puis réaliser une animation. Vérifier que si les vitesses de phase sont égales (on prendra $v_1 = v_2 = 1$), la vitesse de l'enveloppe est la même que la vitesse de phase (choisir $f_1 = 1$ et $f_2 = 1.1$). Comment la nomme-t-on ?
2. Supposez à présent que les vitesses de phase sont différentes ; on prendra $v_1 = 1$ et $v_2 = 0.95$ par exemple. Tracez de nouveau $\psi(x,t)$ à des instants successifs puis réaliser une animation. En déduire la vitesse de déplacement de l'enveloppe et comparer avec la formule théorique.

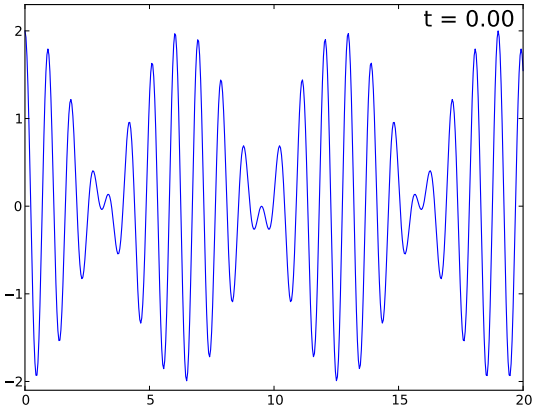


FIGURE 2.1 – *Battements*

Exercice n° 14 *Paquet d'ondes (2) : superposition de N ondes*

Pour représenter un paquet d'ondes réaliste il faut superposer un grand nombre d'ondes monochromatiques (en fait une infinité).

$$\psi(x,t) = \int_{f_{\min}}^{f_{\max}} g(f)\cos(2\pi f t - kx)\, d\nu$$

où k est donné par la relation de dispersion $k(\omega)$ ou $k(f)$ puisque $\omega = 2\pi f$. Nous supposerons ici que la répartition spectrale est de type lorentzienne, c'est-à-dire que le spectre en fréquence est centré sur ν_0 et de largeur Δf et suit la loi

$$g(f) = \frac{A}{1 + \left(\dfrac{f - f_0}{\Delta f}\right)^2}$$

où la constante A est un facteur de normalisation.

Milieu non dispersif On supposera tout d'abord que le milieu est non dispersif, c'est à dire que la vitesse de phase est constante (ne dépend pas de la fréquence).

1. Tracer l'allure du spectre $g(f)$ en prenant $f_0 = 1$ et $\Delta f = 0.2$ sur l'intervalle $[0, 2f_0]$.
2. Construire la superposition d'ondes monochromatiques $\psi(x,t)$ distribuées suivant la loi lorentzienne en approximant l'intégrale par une somme finie. Tracer $\psi(x,t)$ en fonction de x à différents instants.
3. Vérifier qualitativement que lorsque le nombre N d'ondes utilisées devient grand on tend vers un paquet d'ondes « idéal ».
4. Vérifier qualitativement que le produit de la largeur spatiale du paquet d'ondes δx par la largeur spectrale du paquet d'ondes $\delta\omega$ est une constante. Conséquences ?

Milieu dispersif On suppose maintenant que le milieu obéit à la relation de dispersion

$$\omega = \alpha k^2$$

c'est le cas par exemple des ondes capillaires à la surface de l'eau. On prendra pour les calculs qui suivent $\alpha = 1$.

1. Citer un exemple de situation physique où l'on rencontre une dépendance de la vitesse de phase avec la fréquence (ou la longueur d'onde).
2. Réaliser une animation et vérifier (visuellement) que les ondes à l'intérieur du paquet d'ondes se déplacent à une vitesse différente de l'enveloppe.
3. Dans les mêmes conditions que précédemment mais sur un intervalle de temps éventuellement plus grand, vérifier que la largeur du paquet d'ondes augmente, on parle alors d'étalement du paquet d'ondes (cf. figure **??**).

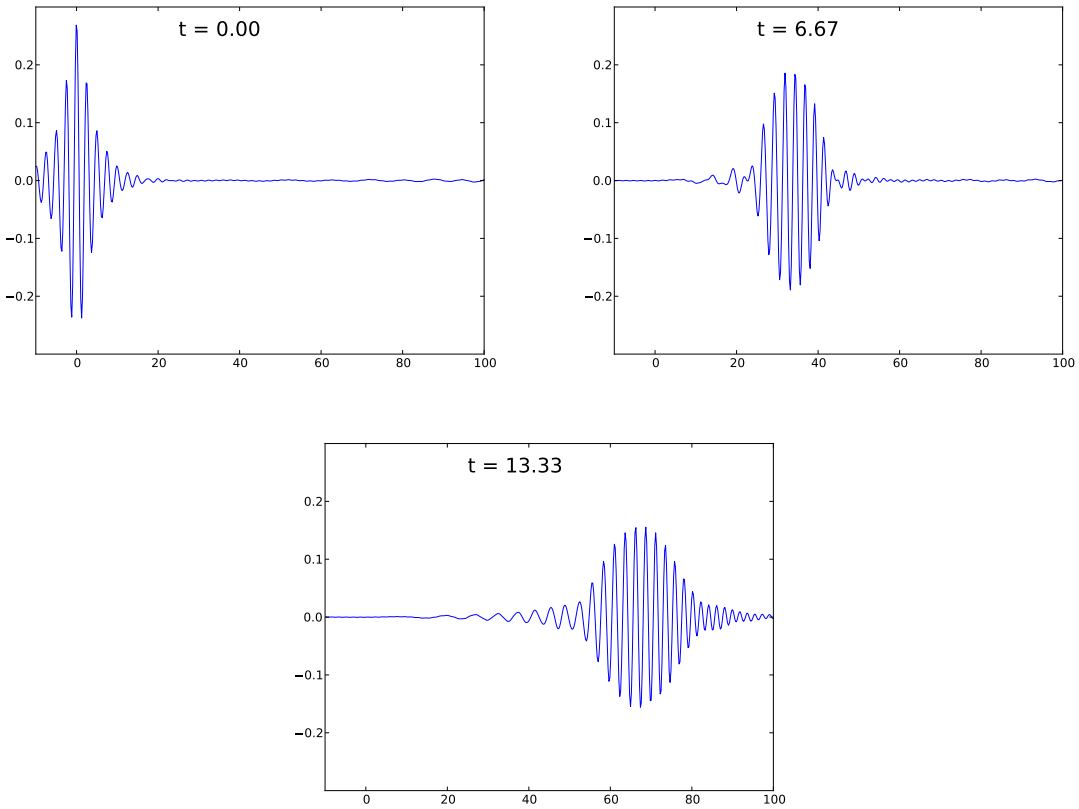


FIGURE 2.2 – *Étalement du paquet d'ondes*

TD 3

Traitement du signal

A Introduction et position du problème

1. Rappel sur la définition de la TF d'un signal $s(t)$

$$\tilde{s}(f) = \int_{-\infty}^{+\infty} s(t)\, e^{-j2\pi ft}\, dt$$

2. Considérations « numériques »
   (a) Notion de durée d'acquisition T_a d'un signal et conséquence sur la TF d'un signal enregistré en un temps fini

$$s_a(t) = s(t) \times \Pi_{T_a}(t)$$

   où $\Pi_{T_a}(t)$ est la fonction « porte » qui vaut 1 dans l'intervalle $[0, T_a]$ et 0 ailleurs.

$$\tilde{s}_a(f) = \int_{-\infty}^{+\infty} s_a(t)\, e^{-j2\pi ft}\, dt = \int_0^{T_a} s(t)\, e^{-j2\pi ft}\, dt$$

   (b) Notion d'échantillonnage d'un signal : approximation de la TF par la Transformée de Fourier Discrète (DFT Discrete Fourier Transform)

$$\tilde{s}_a(f) \simeq T_e \sum_{n=0}^{N_e-1} s(nT_e)\, e^{-j2\pi fnT_e}$$

3. Exemple d'une fonction sinusoïdale de fréquence f_0

B Transformée de Fourier discrète

B.1 Fabrication d'une DFT « à la main »

1. Fabriquer un signal sinusoïdal $s_0(t)$ de période T_0 sous-multiple de la durée d'acquisition T_a : $T_0 = \frac{T_a}{N_p}$. Prendre par exemple : $f_0 = 5\,\text{Hz}$, $T_a = 1\,\text{s}$et un nombre de points d'échantillonnage $N_e = 128$. En déduire la fréquence d'échantillonnage f_e et tracer le graphe de la fonction $A\cos(2\pi f_0 t)$.
2. Définir un tableau de fréquence `f_tab` entre $-f_e/2$ et $+f_e/2$ contenant par exemple 1024 points.
3. Définir une fonction `DFT_mano(f)` permettant de calculer la DFT de $s_0(t)$ pour une fréquence f donnée.
4. Tracer alors le spectre de $s_0(t)$ dans le domaine de fréquence défini par `f_tab`. On zoomera sur la partie intéressante du spectre.
5. Tracer le graphe de la TF exacte de ce signal sinusoïdal et comparer

B.2 Utilisation de la FFT (Fast Fourier Transform)

B.2.1 Cas simple où $T_a = N_p T_0$

1. Calculer le spectre de $s_0(t)$ en utilisant la fonction `fft` de la sous-bibliothèque de `numpy` : `numpy.fft`
2. Pour obtenir les fréquences correspondantes utilisez la fonction `fftfreq(Ne,Te)` et tracer le spectre. Comparer avec la solution exacte et la solution `DFT_mano` fabriquée précédemment.

B.2.2 Cas général (zéro-padding ou complétion de zéro)

1. Que se passe-t-il si à présent la période n'est plus un sous-multiple de la durée d'acquisition ? On prendra par exemple $T_a = 0,9\,\mathrm{s}$. Comparer là encore la solution exacte, la solution `DFT_mano` et la solution par FFT.
2. On remédier à cet artefact en rajoutant des zéros au signal. Le faire de façon à ce que le signal ainsi modifié contienne 1024 points (on aura donc ici $N_e = 128$, $N_z = 1024 - 128$). Comparer ce nouveau spectre avec le spectre calculé avec `DFT_mano`. Commentaire ?
3. Pour comprendre l'intérêt d'utiliser la fonction FFT plutôt que la fonction `DFT_mano` , calculer le temps réalisé pour calculer ces spectres. On utilisera la fonction time de la bibliothèque time qui donne l'instant au moment de l'appel. Il suffit alors de faire un appel à cette fonction avant le calcul puis après et par différence obtenir la durée du calcul. Commenter le résultat.

C Application – le filtrage

La transformée de Fourier est un outil très puissant pour filtrer des signaux bruités. La procédure est la suivante :

- Calcul de la transformée de Fourier du signal bruité.
- Suppression à l'aide d'un filtre des fréquences non voulues (exemple : hautes fréquences dans le cas d'un filtre passe-bas).
- Calcul de la transformée de Fourier inverse pour retrouver le signal de départ.

C.1 En dimension 1

1. Générer un signal temporel comme la somme de deux sinusoïdes :

$$f(t) = A_1 \sin(2\pi\nu_1 t + \phi_1) + A2 \sin(2\pi\nu_2 t + \phi_2)$$

   où : $A_1 = 1$, $A_2 = 0.5$, $\phi_1 = \pi/6$, $\phi_2 = 0$.
2. Calculer la transformée de Fourier rapide de ce signal et représenter sur le même graphe la partie réelle, la partie imaginaire et l'amplitude.
3. Ajouter un bruit aléatoire à ce signal à l'aide de la fonction `random.random_sample` de la bibliothèque `numpy.random` .
4. Effectuer la transformée de Fourier rapide de ce signal bruité. Tracer la partie réelle, la partie imaginaire et l'amplitude et les comparer aux résultats précédents.
5. Filtrer la transformée de Fourier à l'aide d'un filtre de type porte $\Pi\left(\frac{x-x_0}{b}\right)$.
   Attention : il faut filtrer l'ensemble de la transformée de Fourier (partie imaginaire et partie réelle).
6. Calculer la transformée de Fourier rapide inverse à l'aide de la fonction `ifft` et comparer le signal obtenu au signal de départ. Le filtrage est-il réussi ?

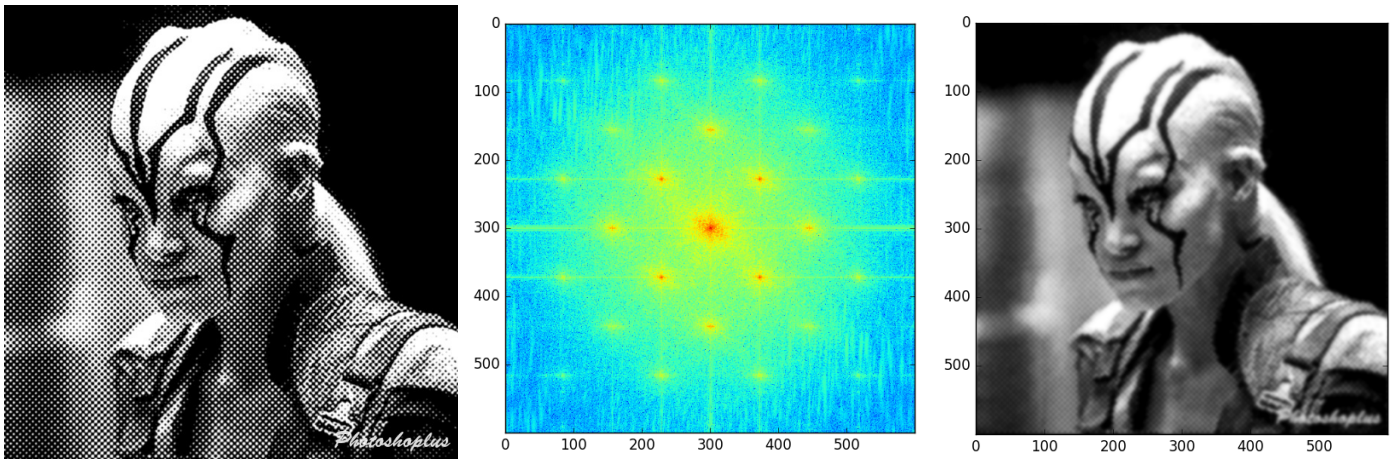
C.2 À 2 dimensions

On se propose dans cette dernière partie de détramée une photo. Le tramage correspond au réseau de points présent dans la figure suivante.

1. Charger l'image à l'aide du code suivant :

```
from skimage import io
img = io.imread('imagetramee.jpg', asgrey=True)
```

2. Effectuer la transformée de Fourier rapide à deux dimensions de cette image et la tracer en échelle logarithmique.
3. Définir un filtre passe-bas et l'appliquer à la transformée de Fourier.
4. Faire la transformée de Fourier inverse à 2 dimensions. Régler si nécessaire les paramètres du filtre pour retirer les structures. On doit obtenir une image comme sur la figure **??** :



(a) *Figure originale* (b) *Spectre de la figure* (c) *Image filtrée : disparition du tramage*

FIGURE 3.1 – *Figure originale, son spectre de Fourier à 2D et l'image détramée*