CPE Lyon - 3ETI 2021-2022

Analyse (M-ANA) : Travaux pratiques Matlab

Exercice 0 : opération produit sous Matlab (avez-vous bien compris la différence entre « * » et « .* », entre « / » et « ./ », entre « ^ » et « .^ » ?)

On considère la fonction f définie par :

$$f(t) = \frac{e^{-2t^2}\cos\left(\frac{3\pi}{2}t\right)}{\pi(1+4t^2)}$$

Tapez le code suivant, il permet de calculer et d'afficher la fonction f sur l'intervalle $[-5, 5]$:

```
1  t=-5:0.01:5;
2  f=exp(-2.*t.^2).*cos(3.*pi.*t./2)./(pi.*(1+4.*t.^2));
3  plot(t,f);
```

Même si ce code est correct il contient des « . » inutiles, ré-écrire la ligne 2 en supprimant les « . » inutiles.

Exercice 1 : Série de Fourier

On considère la fonction f_0 définie par :

$$f_0(t) = e^{-t}, \qquad t \in [0\,;2[$$

En périodisant f_0 avec une période $T = 2$, on obtient le signal périodique f dont la représentation graphique est :

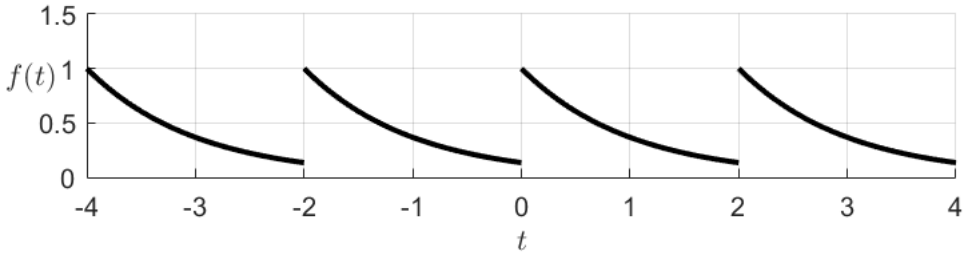


Fig. 1 : Représentation graphique de la périodisée de $f_0(t) = e^{-t}$ de période $T = 2$

Compléter le code de la page suivante afin que celui-ci permette de :

- afficher le signal f sur une période, par exemple entre 0 et 2
- calculer la pulsation du signal
- calculer le coefficient de Fourier a_0 (commande `trapz`)
- calculer les coefficients de Fourier a_n et b_n pour $n \in [\![1, 15]\!]$ (commande `trapz`)
- reconstruire le signal
- calculer l'énergie totale du signal (commande `trapz`)
- calculer l'énergie du signal reconstruit
- afficher le signal reconstruit en superposition du signal d'origine
- calculer l'erreur relative sur le calcul de l'énergie
- calculer et afficher le spectre de fréquence (commande `bar`)

On utilisera les notations suivantes :

- `t` : intervalle de temps sur une période [vecteur]
- `f` : signal $f(t)$ [vecteur]
- `T` : période du signal [scalaire]
- `omega` : pulsation du signal [scalaire]
- `a0` : coefficients de Fourier a_0 [scalaire]
- `energie_tot` : énergie totale du signal [scalaire]
- `N` : nombre d'harmoniques [scalaire]
- `f_rec` : signal reconstruit [vecteur]
- `energie_rec` : énergie du signal reconstruit [scalaire]
- `freqn_array` : fréquences des N harmoniques [vecteur]
- `An_array` : amplitudes des N harmoniques [vecteur]
- `an, bn` : coefficients de Fourier [scalaires]
- `erreur_rel` : précision relative du calcul de l'énergie [scalaire]

Code à compléter :

```
figure(1);hold on;

% intervalle de temps
t=_____________________________________;

% signal
f=_____________________________________;

% affichage du signal
subplot(121);hold on;
plot(____,____,'k','linewidth',2);
grid on;axis([0,2,0,1.2]);

% période et pulsation
T=2;
omega=_________________________________;

% coefficient a0
a0=____________________________________;

% enrgie totale du signal périodique (initialisation)
energie_tot=___________________________;

% nombre d'harmoniques prises en compte
N=15;

% signal reconstruit (initialisation)
f_rec=_________________________________;

% énergie du signal reconstruit (initialisation)
energie_rec=___________________________;

% fréquences des harmoniques (initialisation)
freqn_array=___________________________;

% amplitude des harmoniques (initialisation)
An_array=______________________________;

% calcul du signal reconstruit et de son énergie
for n=____:____
    an=________________________________;
    bn=________________________________;
```

```
    f_rec=_____________________________;
    energie_rec=_____________________;
    freqn_array(n)=_________________;
    An_array(n)=____________________;
end

% affichage du signal reconstruit (en superposition du signal d'origine)
subplot(121)
plot(____,____,'r');
xlabel('$t$','interpreter','latex','FontSize',12);
lg=legend('$f(t)$','S\''erie de Fourier');
set(lg,'interpreter','latex','FontSize',12);

% erreur relative sur le calcul d'énergie
erreur=_____________________________

% affichage du spectre
subplot(122);
bar(____,____);
xlabel('$\nu_n$','interpreter','latex','FontSize',12);
ylabel('$A_n$','interpreter','latex','FontSize',12);
```

L'exécution du programme doit donner le résultat suivant :

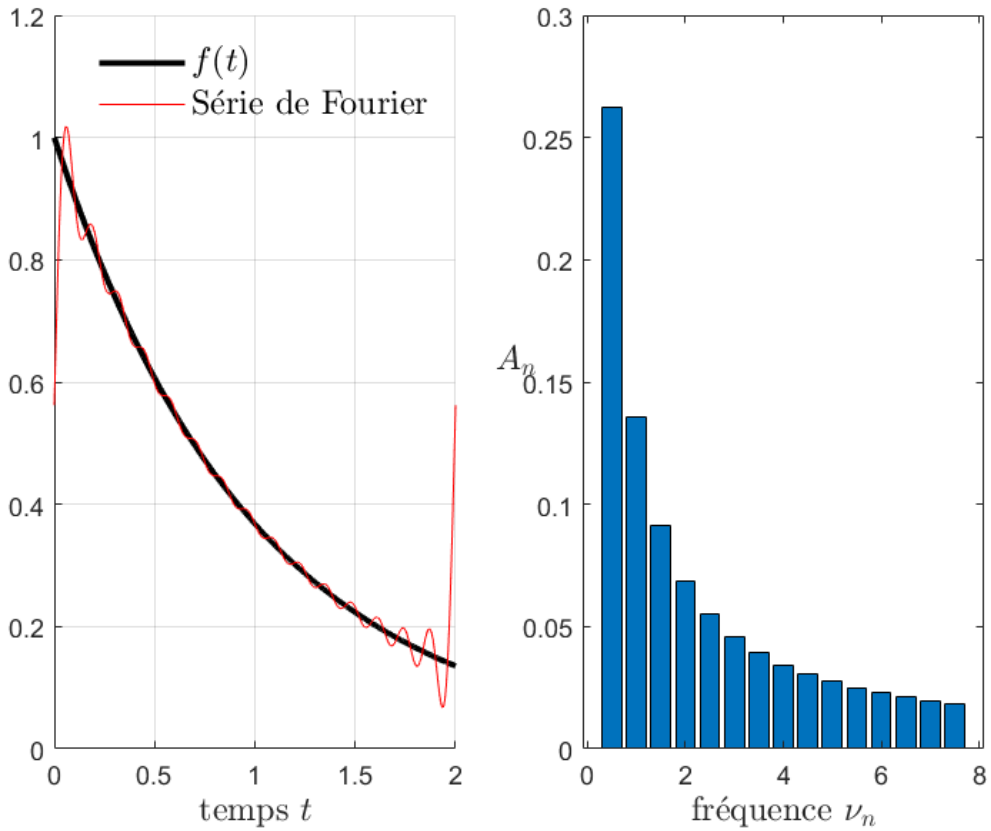


Fig. 2 : A gauche, signal d'origine $f(t)$ et le signal reconstruit à l'aide d'un développement en série de Fourier calculé avec 15 harmoniques. A droite, spectre de fréquences des 15 premières harmoniques.

Exercice 2 : transformée de Fourier

On considère la fonction f définie par :

$$\forall t \in \mathbb{R}, \qquad f(t) = e^{-\pi t^2} \cos(4\pi t) \quad (1)$$

La représentation graphique de f est la suivante :

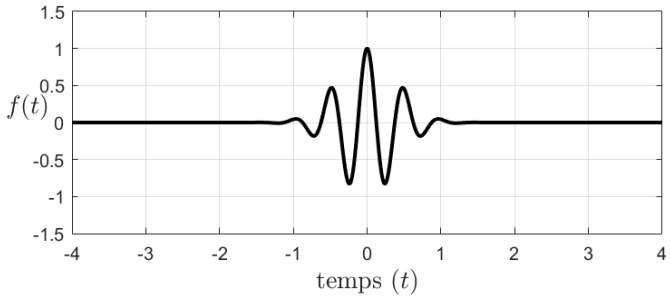


Fig. 4 : Représentation graphique de $f(t) = e^{-\pi t^2} \cos(4\pi t)$

1) On s'intéresse à la transformée de Fourier de f. Compléter le code suivant afin que celui-ci permette de calculer la transformée de Fourier de f (utiliser la commande `trapz` de Matlab), et d'afficher dans une même fenêtre, mais sur 3 graphiques différents, la partie réelle, la partie imaginaire et la phase de la transformée de Fourier (voir Fig. 5)

Code à compléter :

```
clear variables;close all;
i=complex(0,1);


% intervalle temporel
tmin=-5;tmax=5;Te=0.001;
t=tmin:Te:tmax;
N=length(t);


% signal f
f=_____________________________________________________________________________
plot(___________,__________,'k');  % affichage (en noir)


% intervalle fréquentiel
nu_e=1/Te;
nu=linspace(-nu_e/2,nu_e/2,N);

% transformée de Fourier (initialisation)
tf=_____________________________________________________________________________

% calcul de la transformée de Fourier (utiliser commande trapz)
for k=_____________________________ % pour chaque fréquence
    tf(k)=_____________________________________________________________________
end

% affichage
figure(1);
subplot(311);plot(nu,real(tf),'k');xlim([-10,10]);
subplot(312);plot(nu,imag(tf),'k');xlim([-10,10]);
subplot(313);plot(nu,angle(tf),'k');xlim([-10,10]);
```

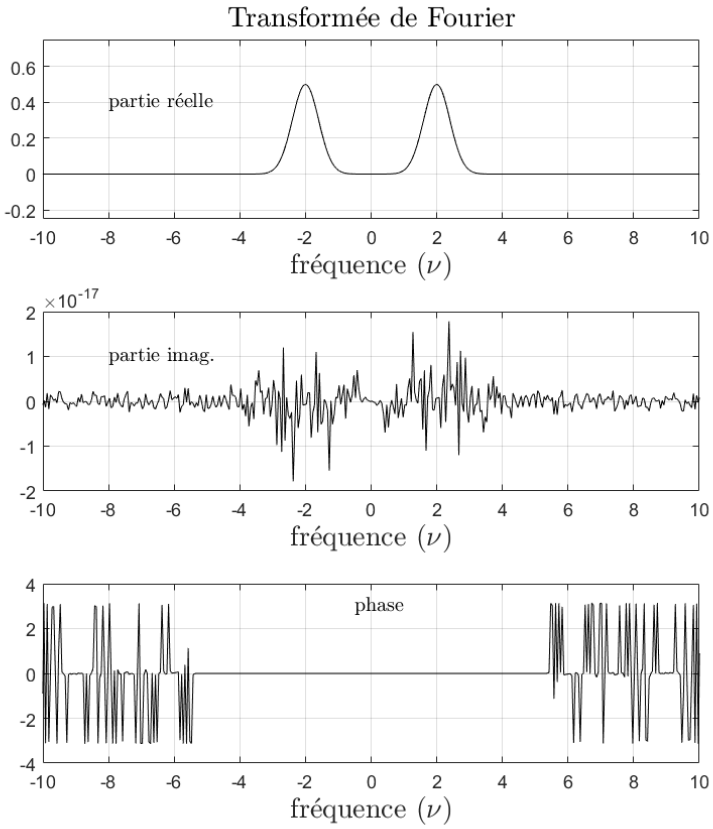


Fig. 5 : Partie réelle, partie imaginaire et phase de la transformée de Fourier de $f(t) = e^{-\pi t^2}\cos(4\pi t)$

2) Décrire et interpréter les résultats de la Fig. 5 en essayant de faire le lien avec le cours (ces résultats illustrent certaines propriétés du cours, lesquelles ?)

   Indication : on distinguera la partie réelle (pour laquelle on expliquera à l'aide d'un calcul pourquoi la partie réelle correspond à 2 gaussiennes centrées en -2 et 2), la partie imaginaire et la phase (l'interprétation de la phase est plus délicate, aidez-vous de la Fig. 6 ci-dessous) :

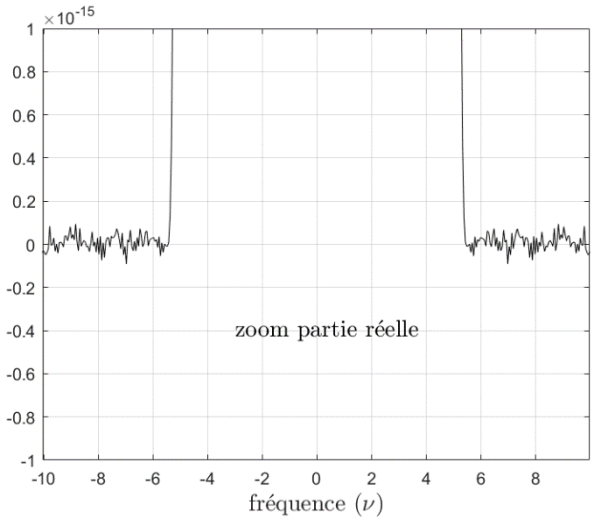
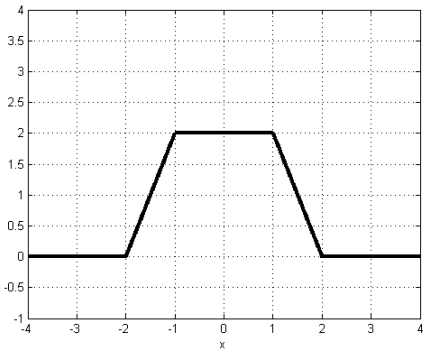


Fig. 6 : Zoom (selon l'axe des ordonnées) de la partie réelle de la transformée de Fourier de f

3) Tester à nouveau votre programme avec les fonctions suivantes :

$$\Pi(t), \qquad \Pi(t-2), \qquad \Pi\left(\frac{t}{3}\right), \qquad t\Pi(t), \qquad 3te^{-2\pi|t|}, \qquad e^{-\frac{t^2}{4}}, \qquad \frac{t}{(1+9t^2)^2}$$

ainsi que la fonction dont la courbe représentative est :



Indication : récupérer les fonctions `porte` et `triangle` utilisées dans l'exercice d'initiation Matlab.

Exercice 3 : transformée de Fourier et filtrage hautes fréquences

Taper le code suivant :

```
% intervalle temporel
tmin=-5;tmax=5;Te=0.01;
t=tmin:Te:tmax;
N=length(t);

% lorentzienne
f0=(1+(t+1).^2).^(-1);

% signal bruité (bruit gaussien)
f=f0+0.05*randn(1,N);

% affichage
figure(1);hold on;
plot(t,f);
xlabel('temps $(t)$','interpreter','latex');
```

Ce programme affiche une lorentzienne (`f0=(1+(t+1).^2).^(-1)`) à laquelle on a ajouté un bruit aléatoire gaussien (`0.05*randn(1,N)`), voir Fig. 7 ci-dessous :

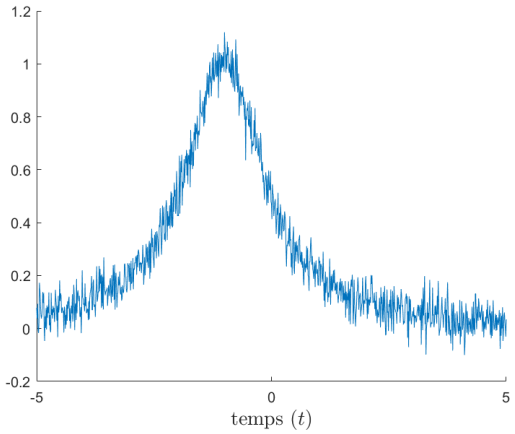


Fig. 7 : Signal bruité (lorentzienne + bruit aléatoire gaussien)

Ecrire, à la suite du code précédent, un programme Matlab permettant de :

- Calculer la transformée de Fourier du signal bruité (s'inspirer de l'exercice 3)
- Afficher la partie réelle et la partie imaginaire de la transformée de Fourier (voir Fig. 8)
- Supprimer les fréquences supérieures à 2.5 Hz en valeur absolue (on dit qu'on filtre les hautes fréquences), pour cela on pourra multiplier la transformée de Fourier par une porte de hauteur 1 et de largeur 5
- Afficher la partie réelle et la partie imaginaire de la transformée de Fourier après filtrage (voir Fig. 9)

- Calculer la transformée de Fourier inverse et afficher le signal ainsi obtenu (voir Fig. 10)
- Commenter les résultats. Que se passe-t-il lorsqu'on modifie la valeur de la fréquence de filtrage ?

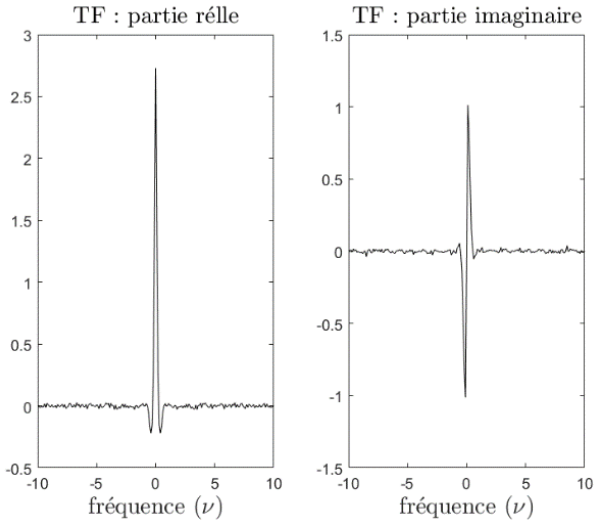


Fig. 8 : Transformée de Fourier du signal bruité

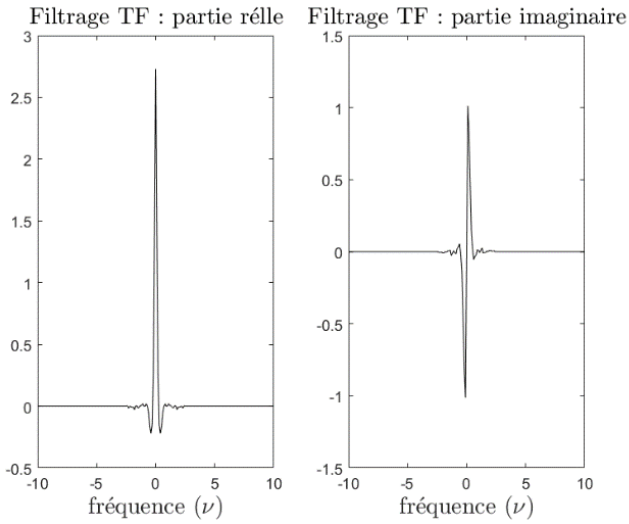


Fig. 9 : Filtrage des hautes fréquences de la transformée de Fourier

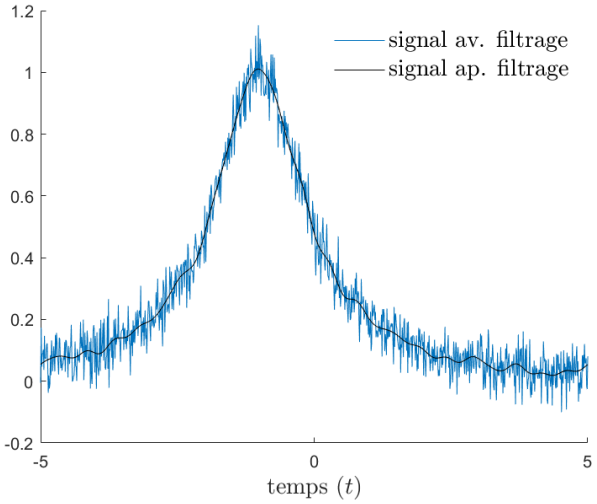


Fig. 10 : Signal bruité (en bleu) et signal obtenu par transformée de Fourier inverse après filtrage des hautes fréquences de la transformée de Fourier du signal bruité

Exercice 4 : produit de convolution et transformée de Fourier

On rappelle la définition du produit de convolution *continu* de deux fonctions f et g :

$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau \quad (1)$$

Dans la pratique, on ne calcule que très rarement le produit de convolution de deux signaux f et g à l'aide de la formule (1). En effet, la plupart du temps les signaux n'ont pas d'expression analytique ou alors, quand ils en ont une, le calcul exact de l'intégrale (1) n'est pas toujours possible. Pour remédier à cela, on échantillonne les signaux, cela signifie que les fonctions f et g ont été remplacées par des vecteurs contenant un nombre fini de composantes, puis on utilise une méthode approchée pour calculer l'intégrale de la formule (1) : on parle alors de produit de convolution *discret*.

De façon générale, le produit de convolution de deux vecteurs $f = (f_1, f_2, \dots, f_m) \in \mathbb{R}^m$ et $g = (g_1, g_2, \dots, g_n) \in \mathbb{R}^n$ donne un vecteur $h \in \mathbb{R}^{m+n+1}$ dont les composantes h_i sont données par la formule suivante :

$$\forall i \in [\![1, m+n-1]\!], \qquad h_i = \sum_{k=\max(1,i+1-n)}^{\min(i,m)} f_k g_{i-k+1} \quad (2)$$

Le produit de convolution discret défini par (2) est la version discrète de la formule (1), il est calculé à l'aide de la commande `conv` de Matlab.

Pour calculer le produit de convolution discret de deux signaux f et g on commence par les échantillonner. L'échantillonnage des signaux se fait de la façon suivante :

- l'intervalle de temps est découpé en $n-1$ intervalles d'amplitude T_e appelée *période d'échantillonnage*, on obtient alors un vecteur $t = (t_1, t_2, \dots, t_n) \in \mathbb{R}^n$ tel que :
$$t_k = t_{k-1} + T_e$$
- on définit les vecteurs $f = (f_1, f_2, \dots, f_n)$ et $g = (g_1, g_2, \dots, g_n)$ sur l'intervalle de temps t de la façon suivante :
$$f_k = f(t_k) \quad \text{et} \quad g_k = g(t_k)$$

Schématiquement on a :

t	f	g
t_1	$f_1 = f(t_1)$	$g_1 = g(t_1)$
$t_2 = t_1 + T_e$	$f_2 = f(t_2)$	$g_2 = g(t_2)$
⋮	⋮	⋮
$t_k = t_1 + (k-1)T_e$	$f_k = f(t_k)$	$g_k = g(t_k)$
⋮	⋮	⋮
$t_n = t_1 + (n-1)T_e$	$f_n = f(t_n)$	$g_n = g(t_n)$

On a alors l'approximation suivante :

$$\forall i \in [\![1, 2n-1]\!], \qquad h_i = (f * g)(t_i) = \int_{-\infty}^{+\infty} f(\tau)g(t_i-\tau)d\tau \approx T_e \sum_{k=\max(1,i+1-n)}^{\min(i,n)} f_k g_{i-k+1} \quad (3)$$

La multiplication par T_e dans la formule (3) s'explique par le fait que la commande `conv` ne prend pas en compte la période d'échantillonnage T_e.

Partie A : calcul direct du produit de convolution (à l'aide de la commande `conv` de Matlab)

1) A l'aide d'un programme Matlab, calculer et afficher le produit de convolution de deux portes standards ($f = g = \Pi$). L'exécution du programme doit donner la figure suivante :

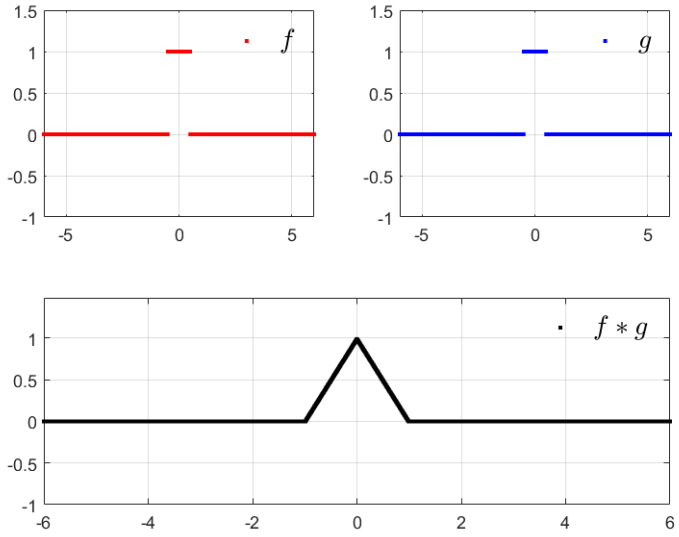


Fig. 11 : Produit de convolution de 2 portes standards ($f = g = \Pi$)

2) Compléter votre programme pour calculer et afficher les produits de convolution des fonctions f et g définies dans les exercices 1, 2 et 3 du chapitre II (cf. recueil d'exercices de travaux dirigés). Vérifier vos résultats en les comparant à ceux obtenues par calcul exact (c'est-à-dire « à la main ») lors de la séance de TD.

Partie B : calcul indirect du produit de convolution (à l'aide de la transformation de Fourier)

Refaire les calculs de la partie A sans calculer explicitement le produit de convolution mais en utilisant plutôt une propriété de la transformée de Fourier vue en cours.

Exercice 5 : transformée de Fourier et filtrage hautes fréquences

On considère un signal gaussien $f_0(t) = e^{-5\pi(t-1)^2}$ auquel on ajoute du « bruit » haute fréquence :

$$f_1(t) = 0.05\sin(2\pi\nu_1 t) + 0.02\sin(2\pi\nu_2 t) \quad \text{avec } \nu_1 = 20 \quad \text{et} \quad \nu_2 = 45$$

- Ecrire un programme Matlab permettant d'afficher le signal bruité $f = f_0 + f_1$ sur un l'intervalle $[-5, 5]$ avec une période d'échantillonnage $T_e = 0{,}01$.
- Calculer la transformée de Fourier de f (s'inspirer de l'exercice 3) sur l'intervalle fréquentiel $\left[-\frac{\nu_e}{2}; \frac{\nu_e}{2}\right]$ où $\nu_e = 1/T_e$ est la fréquence d'échantillonnage.
- Afficher, dans la même fenêtre mais sur deux autres graphiques (voir Fig. 12 ci-dessous), la partie réelle et la partie imaginaire de la transformée de Fourier de f. Commenter.
- Filtrer les fréquences supérieures à 40Hz (s'inspirer de l'exercice 3), puis afficher, la partie réelle et la partie imaginaire de la nouvelle transformée de Fourier ainsi modifiée, enfin, calculer la transformée de Fourier inverse (utiliser commande `trapz`) et l'afficher sa partie réelle (voir Fig. 13). Commenter.
- Refaire l'étape précédente mais cette fois-ci avec les fréquences supérieures à 15 Hz. Commenter.
- Ecrire votre programme de façon suffisamment « générique » afin de le tester pour d'autres signaux en modifiant simplement la ligne correspondant à f_0 ; par exemple, exécuter votre programme dans le cas où f_0 est la fonction triangle standard et où le bruit haute fréquence est :

  $$f_1(t) = 0.05\cos(2\pi\nu_1 t) + 0.02\cos(2\pi\nu_2 t) \quad \text{avec } \nu_1 = 20 \quad \text{et} \quad \nu_2 = 45$$

  Commenter.

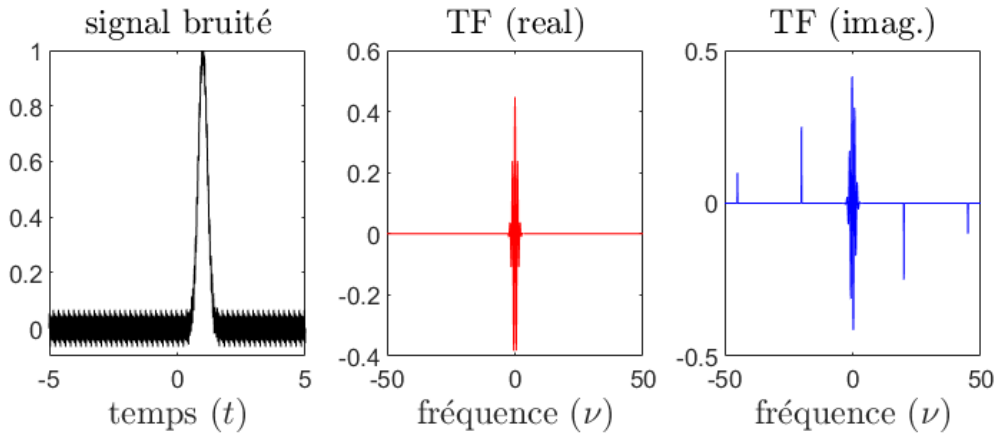


Fig. 12 : Signal bruité et sa transformée de Fourier

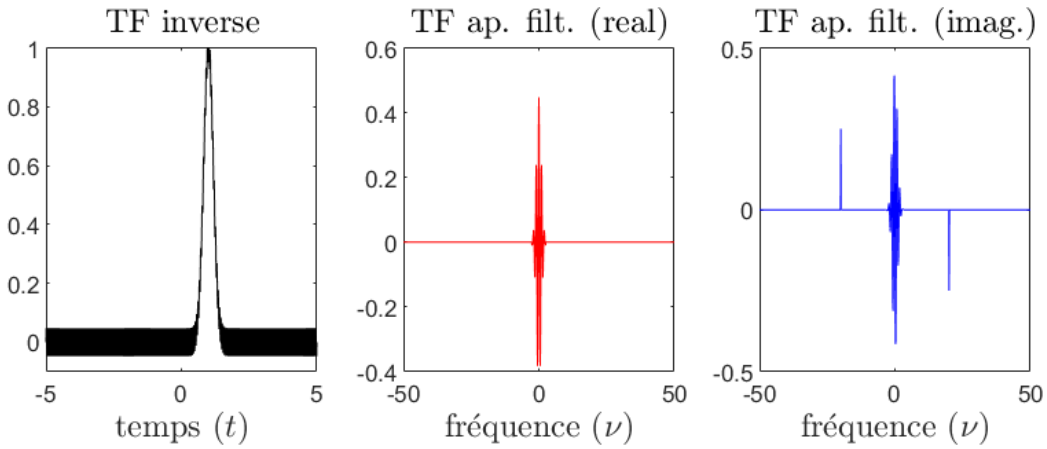


Fig. 13 : Filtrage des fréquences supérieures à 40 Hz

Exercice 6 : distribution de Dirac

Partie A : limite d'une série de fonctions continues

On considère la fonction suivante :

$$g_\sigma(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2} \qquad (1)$$

1) Tracer sur une même figure les courbes représentatives de g_σ, sur l'intervalle $[-4, 4]$ avec une période d'échantillonnage $T_e = 0{,}005$, pour les valeurs de σ égales à $1/2^k$ avec $k = 2, 4$ et 6 (utiliser commande `subplot`, voir Fig. 14 ci-dessous). Commenter.
2) Tracer sur la même figure les transformées de Fourier des fonctions g_σ de la question précédente sur l'intervalle fréquentiel $\left[-\frac{\nu_e}{2}; \frac{\nu_e}{2}\right]$ où $\nu_e = 1/T_e$ est la fréquence d'échantillonnage (utiliser commande `trapz`). Commenter.

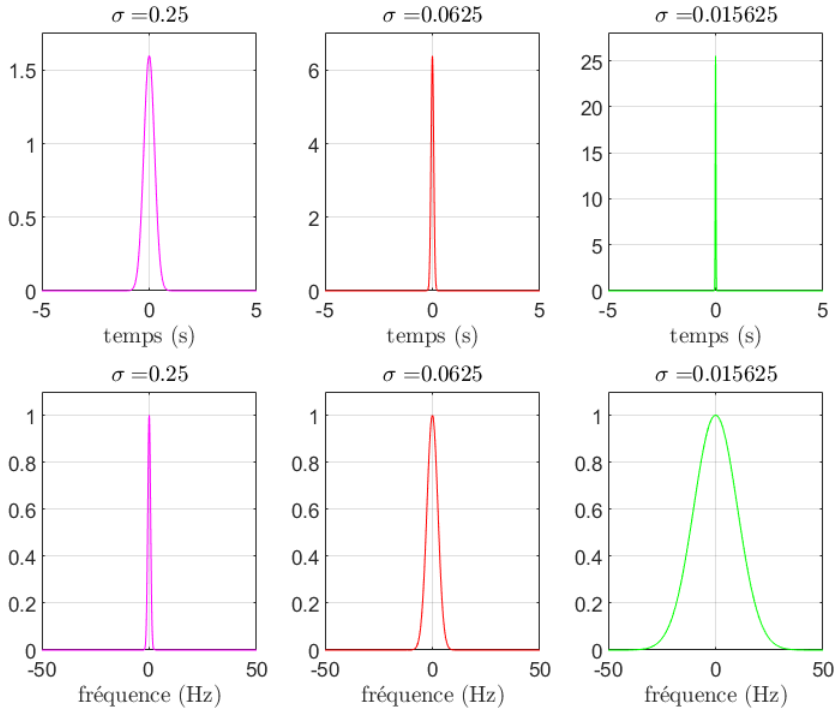


Fig. 14 : Trois gaussienne g_σ de largeur différentes (en haut) et leurs transformées de Fourier

3) Même question que la question précédente mais cette fois-ci avec 3 portes de largeurs σ égales à 0.5, 2 et 4. On doit obtenir le résultat suivant :

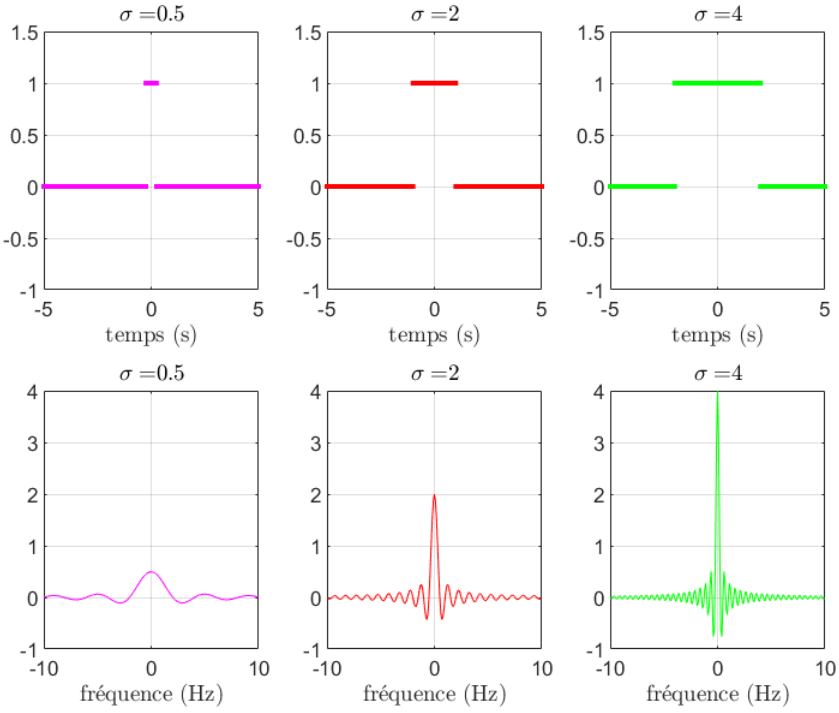


Fig. 15 : Trois portes de largeur différentes (en haut) et leurs transformées de Fourier

Partie B : convolution par un Dirac

On considère la fonction suivante :

$$s(t) = e^{-\pi t^2} \cos(2\pi \nu_0 t) \qquad (2)$$

1) Afficher dans une même figure mais dans 2 repères différents (cf. commande `subplot`) la courbe de la fonction $s(t)$ avec $\nu_0 = 5\ Hz$ sur l'intervalle $[-4, 4]$ avec une période d'échantillonnage $T_e = 0{,}005$, et la courbe de sa transformée de

Fourier $\hat{s}(\nu) = \mathcal{F}(s(t))$ sur l'intervalle fréquentiel $\left[-\frac{\nu_e}{2}; \frac{\nu_e}{2}\right]$ où $\nu_e = 1/T_e$ est la fréquence d'échantillonnage (utiliser commande `trapz`). On doit obtenir la figure suivante :

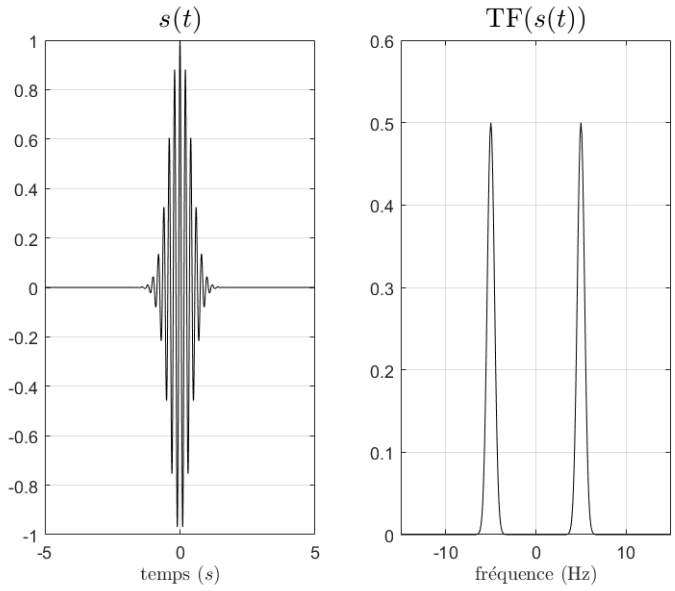


Fig. 16: Signal $s(t)$ et sa transformée de Fourier

Commenter.

2) Tracer dans une même figure mais dans 4 repères différents la fonction $s(t)$ et la fonction $y_\sigma(t) = (s * g_\sigma)(t)$ pour les valeurs de σ égales à $1/2^k$ avec $k = 2, 4$ et 6 (utiliser la commande `conv` de Matlab en choisissant l'option `'same'`). On doit obtenir la figure suivante :

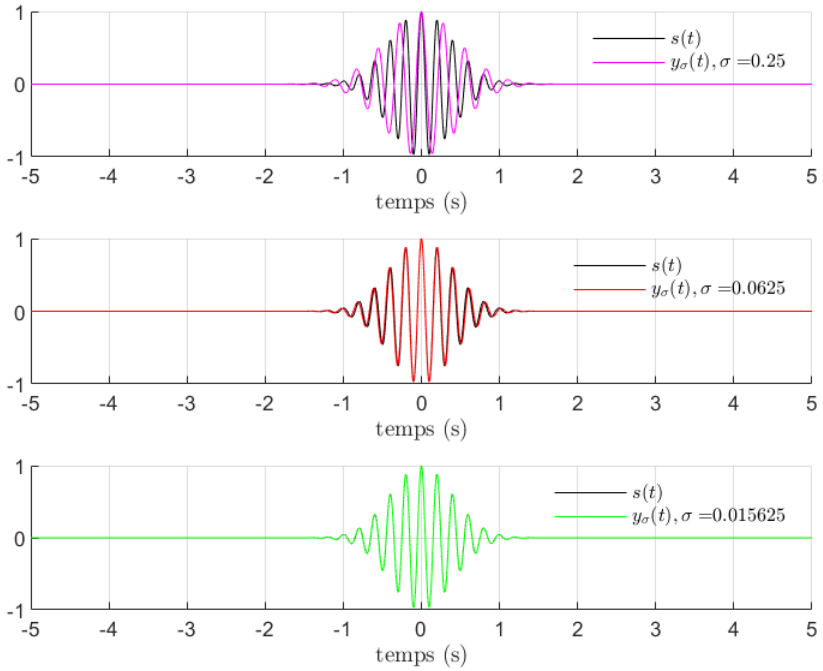


Fig. 17: Convolution du signal $s(t)$ par 3 gaussiennes g_σ de largeurs différentes

Commenter.

Exercice 7 : peigne de Dirac

1) Construire un vecteur t contenant $N = 2^{12}$ valeurs régulièrement espacées et comprises entre -1 (inclus) et $+1$ (exclus), puis afficher dans une même figure mais dans 4 repères différents (commande `subplot`) les peignes de Dirac de période $T_1 = 2^{-5}$ et $T_2 = 2^{-6}$ [utiliser la fonction `peigne` accessible sur CPe-campus] et leurs transformées de Fourier [utiliser la fonction `TransFourier` accessible sur CPe-campus]. Vous devriez obtenir la Fig. 18. Commenter.

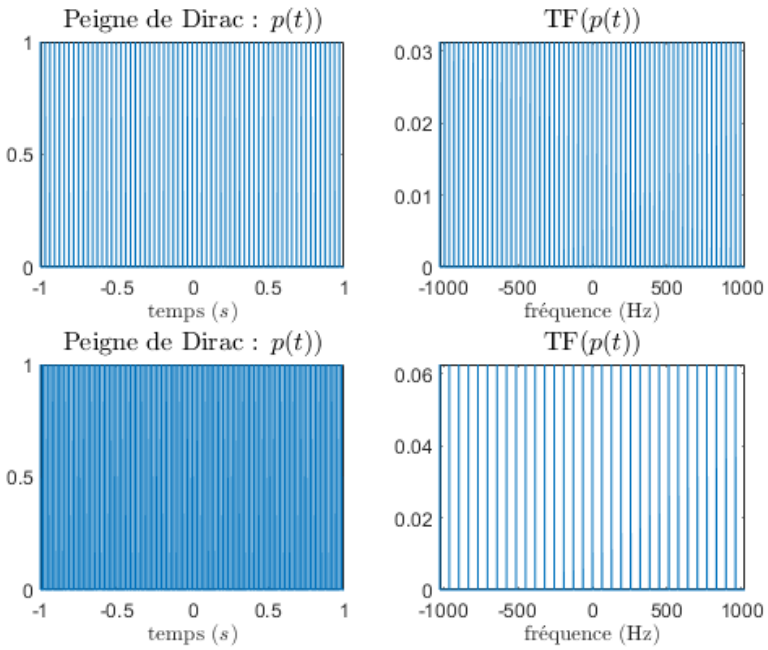


Fig. 18: (à gauche) Deux peignes de Dirac de périodes $T_1 = 2^{-5}$ (haut) et $T_2 = 2^{-6}$ (bas), et (à droite) leurs transformées de Fourier.

2) On considère le signal défini par :

$$\forall t \in [-1\,,1[\,, \qquad s(t) = e^{-\pi t^2}\cos(2\pi\nu_0 t) \quad \text{avec } \nu_0 = 5\ Hz$$

Ecrire un programme Matlab permettant de :

- calculer le produit $z(t) = s(t)\text{Ш}_{T_2}(t)$
- calculer $\hat{s}(\nu) = \mathcal{F}\big(s(t)\big)$ et $\hat{z}(\nu) = \mathcal{F}\big(z(t)\big)$ [utiliser la fonction `TransFourier` accessible sur CPe-campus].
- afficher dans une même fenêtre mais dans 3 repères différents $s(t), z(t), \hat{s}(\nu)$ et $\hat{z}(\nu)$ de sorte à obtenir le résultat suivant :

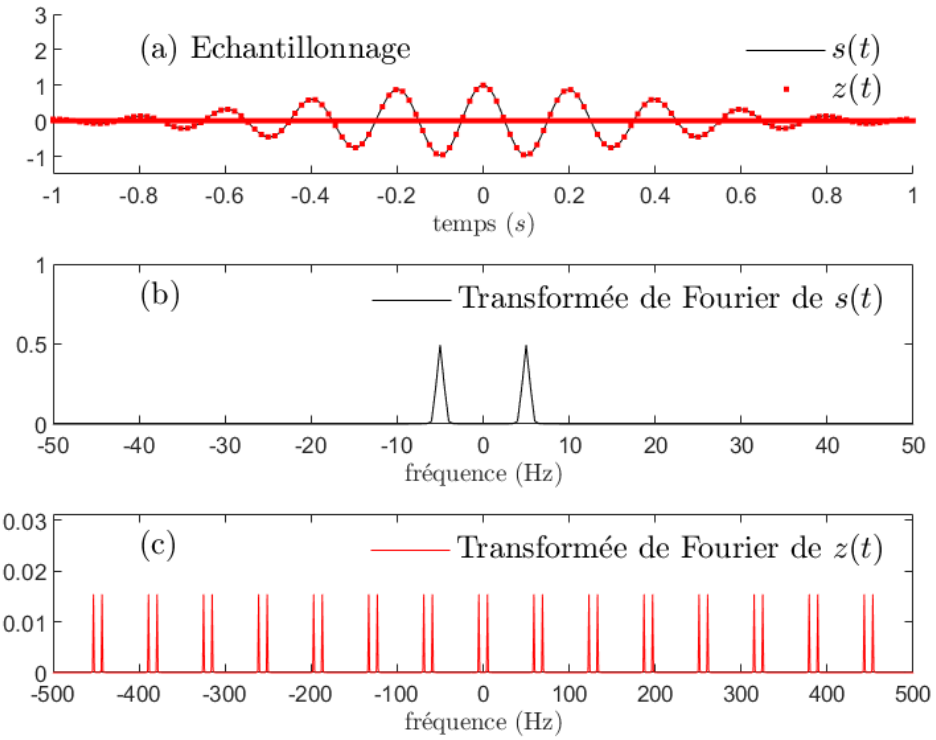


Fig. 19: (a) Signal $s(t)$ (en noir) et son échantillonnage $z(t)$ de période $T_e = 2^{-6}$ (points rouges). (b) Transformée de Fourier du signal $s(t)$. (c) Transformée de Fourier du signal échantillonné $z(t)$.

Commenter et expliquer ces résultats.

Exercice 8

Matlab permet de traiter la transformée de Fourier d'une image via la commande `fft2`, la transformée de Fourier inverse étant `ifft2`. La commande `fftshift` permet de recentrer l'image (comme dans le cas monodimensionnel). A toute matrice rectangulaire A de dimension $n \times m$, on peut associer une image de m pixels (largeur) sur n pixels (hauteur) et dont l'intensité de chaque pixel est égale au coefficient correspondant de la matrice. Cette image peut être affichée à l'aide de la commande Matlab `imshow(M)`.

1) On souhaite retrouver certains résultats vue en cours au chapitre IV concernant la transformée de Fourier bidimensionnelle. Pour cela nous avons besoin d'utiliser des portes de dimension 2 (fenêtres rectangulaires), le code Matlab suivant est celui permettant de définir une nouvelle fonction appelée `porte_2d` :

```
function P=porte_2d(n,m,lx,ly)
    x=-floor(n/2):floor(n/2);
    y=-floor(m/2):floor(m/2);
    p1=abs(x)<0.5*lx;
    p2=abs(y)<0.5*ly;
    P=p2'*p1;
```

Les paramètres d'entrée sont :

- `n` : dimension de l'image selon l'axe horizontal
- `m` : dimension de l'image selon l'axe vertical
- `lx` : largeur de la porte selon l'axe horizontal
- `ly` : largeur de la porte selon l'axe vertical

La sortie est une matrice binaire $P = (p_{i,j})$ de dimension $m \times n$ définie par :

$$p_{i,j} = \begin{cases} 1, & \text{si } -\frac{lx}{2} \le x_i \le \frac{lx}{2} \text{ et } -\frac{ly}{2} \le y_i \le \frac{ly}{2} \\ 0, & \text{sinon} \end{cases}$$

Travail à faire :

- Coder et ajouter la fonction `porte_2d` dans votre répertoire de travail.
- Taper et exécuter le programme suivant :

```
clear variables;close all;

n=300;
m=n;

figure(1);
x=-n/2:n/2;
y=x;
lx=45;
ly=20;
M1=porte_2d(m,n,lx,ly);
subplot(1,2,1);
imshow(M1);
```

- Compléter le programme par le calcul de la valeur absolue de la transformée de Fourier de M1 (utiliser commandes `abs`, `fft2` et `fftshift`). On notera `TF_M_real` la matrice correspondant à cette transformée de Fourier.
- Afficher l'image correspondante (commande `imshow`). Attention, `imshow` affiche une image en niveau de gris, ce qui signifie que la matrice donnée en paramètre de `imshow` doit avoir des coefficients compris entre 0 et 255 (utiliser la commande `uint8` pour convertir la matrice `TF_M_real` en une matrice, qu'on notera `TF_M_int`, ne contenant que des entiers compris entre 0 et 255). Commenter.
- Changer les dimensions de la fenêtre puis exécuter à nouveau le programme. Commenter.

2) On souhaite reprendre le programme précédent mais cette fois-ci avec une fenêtre circulaire centrée sur le centre de l'image. Pour cela on introduit une nouvelle fonction, appelée `disque`, dont on vous propose de compléter le code :

```
function D=disque(n,m,r)
    D=zeros(m,n);
    for i=-floor(n/2):floor(n/2)
        for j=-floor(m/2):floor(m/2)
            if ____________________
                D(____________,_____________)=1;
            end
        end
    end
```

Les paramètres d'entrée sont :

- `n` : dimension de l'image selon l'axe horizontal
- `m` : dimension de l'image selon l'axe vertical
- `r` : rayon du disque

La sortie est une matrice binaire $D = (d_{k,l})$ de dimension $m \times n$ définie par :

$$d_{k,l} = \begin{cases} 1, & \text{si ? (à compléter)} \\ 0, & \text{sinon} \end{cases}$$

Après avoir coder et ajouter la fonction disque à votre répertoire de travail, calculer et afficher les transformées de Fourier de plusieurs disques de rayon différents. Commenter.

2) Transformée de Fourier d'un motif périodique

- Taper et exécuter le code suivant :

```
clear all;close all;
[x,y]=meshgrid(1:n,1:m);
M=cos(0.5*x+0.5*y);
subplot(1,2,1);
imshow(M);
```

- Compléter le programme par le calcul de la valeur absolue de la transformée de Fourier de M puis afficher l'image correspondante et commenter.
- Modifier le programme de sorte à faire apparaitre deux pics supplémentaires dans le transformée de Fourier.
- Faire un programme permettant d'obtenir le résultat suivant (à peu de chose près) :

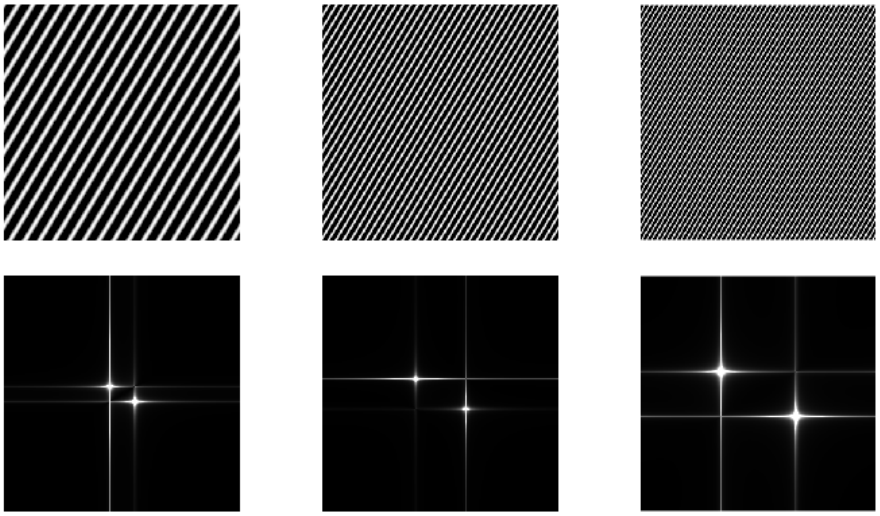


Fig. 20 : (haut) Motifs périodiques dont la fréquence augmente de gauche à droite.
(bas) Transformée de Fourier des motifs périodiques.

- Quel commentaire peut-on faire à la vue de ce dernier résultat ?

Exercice 9

On propose dans cet exercice de faire du traitement d'image à l'aide de la transformée de Fourier bidimensionnelle (commandes `fft2`, `fftshift` et `ifft2`).

- Après avoir récupéré le fichier `barbara.jpg` sur CPe-campus, écrire le programme suivant, l'enregistrer dans le même répertoire que `barbara.jpg` puis l'exécuter :

```
clear variables;close all;
A=imread('barbara.jpg');
[a,b]=size(A);
figure(1);
subplot(2,2,1);
imagesc(A);
colormap gray;
```

- Calculer la transformée de Fourier de A et afficher son module à l'aide de la commande `imagesc` (pour une lecture plus aisée, afficher plutôt le logarithme en base 10 de son module : commande `log10`).
- Filtrer les fréquences supérieures à 60 dans la transformée de Fourier (utiliser la fonction `disque` de l'exercice 8), afficher la transformée de Fourier ainsi modifiée, calculer sa transformée de Fourier inverser et afficher cette dernière (voir Fig. 21). Commenter.

- Réitérer l'étape précédente en modifiant la fréquence et interpréter les résultats.
- Proposer une méthode pour filtrer les fréquences inférieures à une fréquence donnée. Quel résultat devrait-on obtenir cette fois-ci ?

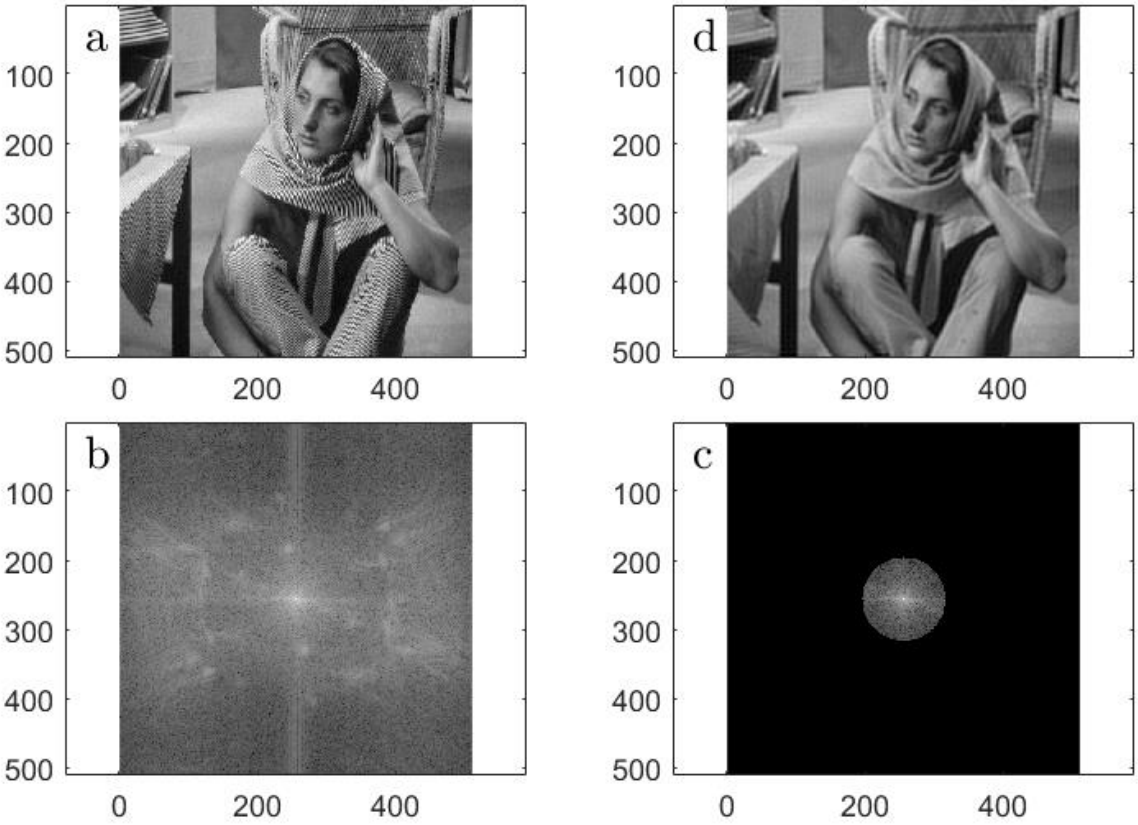


Fig. 21 : (a) Image d'origine. (b) Transformée de Fourier de l'image d'origine. (c) Filtrage des fréquences supérieures à 60. (d) Transformée de Fourier inverse après filtrage.