

ÉCOLE NATIONALE SUPÉRIEURE
DES SCIENCES APPLIQUÉES ET DE TECHNOLOGIE
SPÉCIALITÉ PHOTONIQUE

OPTIQUE DE FOURIER

TP MATLAB

Simulation de traitement optique d'une image
par filtrage spatial

– PHOT2 –

Texte proposé par : Margaux BARBIER^{*}

TP encadré en 2022-2023 par : Tristan GUEZENNEC[†]

2022-2023

^{*}. margaux.barbier@enssat.fr

[†]. tristan.guezennec@univ-rennes1.fr

Table des matières

Présentation du problème	1
1.1 Cadre de l'étude	1
1.2 Rappels : Modélisation du système physique étudié	1
Travail à réaliser	5
2.1 Définition des espaces de simulation numérique	5
2.2 Étude numérique préliminaire	7
2.3 Simulation numérique de filtrage optique d'une image	9
Annexe	11
Bibliographie	15

Présentation du problème

1.1 Cadre de l'étude

Nous vous proposons dans cette séance de TP de réaliser la **simulation numérique** d'une opération de traitement optique d'une image par filtrage spatial.

Une modélisation de ce système physique a été réalisée lors des séances de cours et de TD du module d'Optique de Fourier [1], et nous avons pu tester ce modèle (i.e. simuler le comportement du système prédit par ce modèle) dans des cas simples, c'est-à-dire pour lesquels une résolution analytique était possible. Nous avons ainsi étudié le cas d'objets de fonction de transmission simple (fentes, ouvertures binaires ou objets de phase rectangulaires ou circulaires, grilles, mires sinusoïdales ou en créneau...) éclairés par une onde plane monochromatique en incidence normale. Nous avons pu simuler *analytiquement*, sur de tels objets simples, des opérations de traitement optique par filtrage spatial telles que le détramage, l'épuration, la détection de contours, et les techniques d'imagerie d'objets de phase (strioscopie et contraste de phase).

Il s'agit ici de convertir ce modèle en un modèle *numérique* pour pouvoir simuler des cas d'objet dont la fonction de transmittance est plus complexe, difficilement modélisable sous la forme d'une équation mathématique. Nous continuerons en revanche à considérer ici que l'objet en question est éclairé par une onde plane monochromatique en incidence normale.

La section 1.2 ci-après rappelle le modèle que nous avons établi en cours et qui correspond à ce cadre d'étude. Le travail qui reste à réaliser est décrit dans la partie 2.

Remarque : Sur l'intérêt de l'outil qu'est la simulation numérique, la lecture de l'ouvrage (court et généraliste) [2] vous est chaudement recommandée.

1.2 Rappels : Modélisation du système physique étudié

Le système physique étudié est représenté schématiquement sur la figure 1.1.

Le modèle que nous avons établi en cours pour décrire mathématiquement le comportement d'un tel système repose sur la **théorie ondulatoire de la lumière**, et au sein même de cette théorie ondulatoire, sur la **théorie scalaire de la diffraction**, qui fut proposée en 1678 par Christiaan Huygens et complétée en 1818 par Augustin Fresnel, et formulée mathématiquement sous la forme de l'équation appelée *formule de Rayleigh-Sommerfeld* (1896) suite aux contributions successives (voire parallèles) de nombreux scientifiques – notamment Gustav Kirchhoff (1824-1887), Henri Poincaré (1854-1912) et Arnold Sommerfeld (1868-1951) [3].

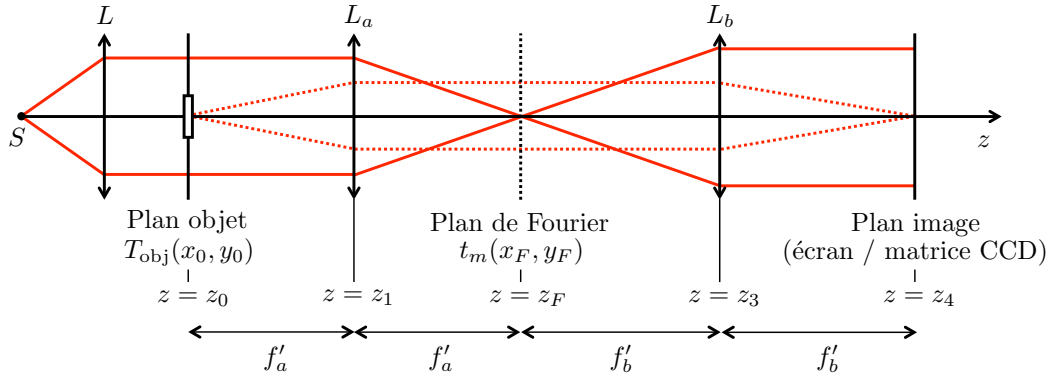


FIG. 1.1 – Montage 4f généralisé (les distances focales f'_a et f'_b des deux lentilles peuvent être différentes).

Au sein de cette théorie, nous nous sommes fixé un **cadre de modélisation** relativement simple, basé sur les **approximations et hypothèses** suivantes :

- approximation de Gauss-Fresnel pour la propagation de l'onde lumineuse ;
- objet plan, infiniment fin dans la direction longitudinale (= direction de propagation de l'onde lumineuse) et infiniment étendu dans le plan transverse ;
- lentilles convergentes minces et d'extension transverse non limitante ;
- onde lumineuse monochromatique et plane.

Au cours du présent TP, nous supposons de surcroît que :

- l'objet est placé dans le plan focal objet de la première lentille du montage 4f (lentille L_a sur la figure 1.1) ;
- l'onde lumineuse (monochromatique et plane) arrive en incidence normale sur l'objet ;
- les lentilles sont parfaitement transparentes.

Dans ce cadre, nous avons pu montrer que :

- ❶ la répartition de l'amplitude complexe normalisée¹ du champ électrique dans le plan focal image de la première lentille du montage 4f généralisé – appelé *plan de Fourier* (voir figure 1.1) – est donnée par :

$$U(x_F, y_F) = \kappa U_0 \text{TF} [T_{\text{obj}}(x_0, y_0)] \left(f_X = \frac{x_F}{\lambda f'_a}, f_Y = \frac{y_F}{\lambda f'_a} \right), \quad (1.1)$$

où U_0 est l'amplitude complexe normalisée du champ électrique de l'onde incidente dans le plan $z = z_0^-$ (i.e. juste avant l'objet), les couples (x_0, y_0) et (x_F, y_F) représentent les coordonnées transverses dans le plan de l'objet ($z = z_0$) et le plan de Fourier ($z = z_F$) respectivement, f'_a la distance focale image de la première lentille du montage 4f, T_{obj} la

1. La normalisation appliquée est telle que $|U|^2$ soit égal à l'intensité lumineuse I , en W/m^2 (cf. cours).

fonction de transmission (en amplitude) de l'objet, et (f_X, f_Y) le couple de fréquences spatiales (variables duales des coordonnées spatiales transverses par transformation de Fourier). Enfin, κ est un terme indépendant de x_F et y_F :

$$\kappa = \frac{e^{2ikf'_a}}{i\lambda f'_a}, \quad (1.2)$$

avec λ la longueur d'onde et $k = 2\pi/\lambda$ le module du vecteur d'onde de l'onde lumineuse qui se propage ;

- ② la répartition de l'amplitude complexe normalisée du champ électrique dans le plan focal image de la seconde lentille du montage 4f généralisé – appelé *plan image* – est donnée par :

$$U(x_4, y_4) = \kappa' \text{TF} [t_m(x_F, y_F) U^{(-)}(x_F, y_F)] \left(f_X = \frac{x_4}{\lambda f'_b}, f_Y = \frac{y_4}{\lambda f'_b} \right), \quad (1.3)$$

où le couple (x_4, y_4) représente les coordonnées transverses dans le plan image ($z = z_4$), f'_b la distance focale image de la seconde lentille du montage 4f, t_m la fonction de transmission (en amplitude) du masque² éventuellement ajouté dans le plan de Fourier, et $U^{(-)}(x_F, y_F)$ l'amplitude complexe normalisée du champ électrique en entrée du masque (plan $z = z_F^-$). Enfin, κ' est un terme indépendant de x_4 et y_4 :

$$\kappa' = \frac{e^{2ikf'_b}}{i\lambda f'_b}, \quad (1.4)$$

Remarque : Dans le cas particulier où aucun masque n'est placé dans le plan de Fourier, on montre facilement que l'image obtenue est une simple reproduction de l'objet, à un facteur de grandissement près (dépendant du rapport entre les distances focales f'_a et f'_b) :

$$U(x_4, y_4) = -U_0 \frac{f'_a}{f'_b} e^{2ik(f'_a + f'_b)} T_{\text{obj}} \left(x_0 = -\frac{f'_a}{f'_b} x_4, y_0 = -\frac{f'_a}{f'_b} y_4 \right). \quad (1.5)$$

Lorsque T_{obj} et t_m ont des formes simples, i.e. des formes décrites par des fonctions standard dont on est capable de calculer *analytiquement* la transformée de Fourier, alors les équations (1.1) et (1.3) peuvent être résolues analytiquement.

Pour résoudre les cas où T_{obj} a une forme plus complexe, il devient nécessaire de *discrétiser* l'espace de simulation (en l'occurrence, ici, le plan transverse) et d'utiliser des algorithmes de résolution numérique (en l'occurrence, ici, un algorithme de calcul de la transformée de Fourier discrète).

Le travail de discrétisation de l'espace de simulation est décrit dans la section 2.1. La prise en main de l'outil de calcul numérique de la transformée de Fourier discrète (algorithme appelé FFT, *fast Fourier transform*, ou transformée de Fourier rapide) via les fonctions MATLAB `fft2` et `fftshift` est traitée en section 2.2. Enfin, le travail de simulation numérique de l'opération de filtrage spatial en tant que telle est abordé en section 2.3.

2. Le masque est, lui aussi, supposé plan, infiniment fin dans la direction longitudinale et infiniment étendu dans le plan transverse.

Travail à réaliser

Le texte de cette partie est largement inspiré des énoncés de TP MATLAB suivants :

- C. Peucheret, “*Calcul numérique de figures de diffraction*”, Module d’Optique de Fourier, ENSSAT, PHOT2, 2016-2017.
- T. Chartier & C. Peucheret, “*Simulation de filtrage optique*”, Module d’Optique de Fourier, ENSSAT, PHOT2, 2016-2017.

2.1 Définition des espaces de simulation numérique

Nous cherchons à être capables de décrire la fonction de transmission T_{obj} d’un objet d’amplitude quelconque, au-delà des seuls cas simples où cette fonction peut être décrite de manière analytique. L’idée est donc de discrétiser l’espace de définition de cette fonction (i.e. le plan transverse) et de donner, élément de discrétisation par élément de discrétisation (= pixel par pixel), la valeur de T_{obj} (réel compris entre 0 et 1).

Nous devons donc indiquer, en début de code, les paramètres de discrétisation que nous souhaitons utiliser, à savoir **le nombre de pixels et la taille des pixels** – et ce à la fois dans la direction x et la direction y . Cela correspond donc à 4 paramètres, que nous noterons M , N , dx et dy .

Deux situations sont alors susceptibles de se présenter :

- ❶ **Dans une phase préliminaire d’apprentissage et de validation du code**, il sera judicieux de simuler numériquement le cas d’objets pour lesquels un résultat analytique est connu (par exemple, un objet dont la fonction de transmission est purement sinusoïdale dans la direction horizontale, et purement invariante dans la direction verticale). C’est la situation à laquelle vous serez confrontés dans la section 2.2.

Dans ce cas, vous pourrez librement choisir les valeurs des 4 paramètres M , N , dx et dy . Notez cependant que :

- plus le nombre total de pixels est élevé, plus le temps de calcul sera grand, bien sûr... ;
- la fonction MATLAB `fft2`, que vous allez utiliser pour passer du domaine direct au domaine de Fourier et inversement, calcule la transformée de Fourier discrète à partir d’un algorithme (FFT) dont les performances sont optimales lorsque le nombre d’échantillons des signaux discrets à traiter est **une puissance de 2**. Vous prendrez donc soin de respecter cette exigence dans votre code.

- ② Une fois que votre code sera validé et que vous en aurez la pleine maîtrise, il s'agira de l'utiliser pour simuler le cas d'un objet plus complexe qui vous sera fourni sous la forme d'un fichier image (en niveaux de gris) au format `.jpg`³. C'est la situation à laquelle vous serez confrontés dans la section 2.3.

Dans ce cas, il conviendra d'utiliser les informations propres au fichier image utilisé (nombres de pixels en x et en y) pour les paramètres M et N . Par ailleurs, n'ayant pas d'information supplémentaire quant aux dimensions physiques réelles de l'objet que le fichier `.jpg` en question représente, il faut choisir arbitrairement les valeurs des paramètres dx et dy (longueurs des côtés des pixels). Dans l'exemple ci-dessous, on a posé que chaque pixel est un carré de $100\ \mu\text{m}$ de côté :

```

1  %-----
2  % INITIALIZATION
3  %-----
4  clear all
5  close all
6
7  OBJECT = imread('name_of_the_file','jpg');
8  % Upload the file
9
10 OBJECT = double(OBJECT(:,:,1))/255;
11 % Convert the uploaded picture to a 2D matrix of 'doubles' with min/max ...
    values equal to 0/1 respectively
12
13
14 %-----
15 % SIMULATION PARAMETERS
16 %-----
17 M = size(OBJECT)(2);
18 % Number of simulation samples in the x-direction (given by the number of ...
    pixels of the uploaded picture file)
19 N = size(OBJECT)(1);
20 % Number of simulation samples in the y-direction (given by the number of ...
    pixels of the uploaded picture file)
21
22 dx = 1e-4;
23 % Separation between samples in the x-direction (in m)
24 dy = 1e-4;
25 % Separation between samples in the y-direction (in m)

```

Les 4 paramètres M , N , dx et dy étant fixés, il devient possible de définir la grille de simulation dans le domaine direct (= domaine spatial transverse), qui va notamment nous permettre de représenter graphiquement l'objet étudié sous la forme d'une figure MATLAB. On utilise pour cela la fonction `meshgrid` :

```

27 % Create the 2D spatial grid:
28 x = (-M/2:1:M/2-1)*dx;
29 y = (- N/2:1:N/2-1)*dy;
30 [X,Y] = meshgrid(x,y);

```

Enfin, nous allons aussi avoir besoin de représenter graphiquement, sous forme de figures MATLAB, le spectre de Fourier de l'objet étudié, avant et après filtrage. Il nous faut donc aussi

3. Pour chaque pixel, le niveau de gris code, sur 256 valeurs possibles (donc, là aussi, sous forme discrétisée), la valeur de la transmission de l'objet sur le pixel considéré – de 0 pour une transmission nulle (se traduisant par un pixel noir) à 255 pour une transmission égale à 1 (se traduisant par un pixel blanc).

définir la grille de simulation dans le domaine spectral. Le pas de discrétisation de cette grille dans les directions horizontale et verticale (paramètres dfx et dfy respectivement) est directement lié à la largeur totale de la grille du domaine spatial, comme vous l'avez appris lors de vos cours de Traitement du signal :

```
32 W = M*dx;
33 % Width of the object in the x-direction (in m)
34 H = N*dy;
35 % Height of the object in the y-direction (in m)
36
37 dfx = 1/W;
38 % Separation between spatial frequency samples relative to the x-direction ...
   (in m^-1)
39 dfy = 1/H;
40 % Separation between spatial frequency samples relative to the y-direction ...
   (in m^-1)
41
42 % Create the 2D spatial frequency grid:
43 fx = (-M/2:1:M/2-1)*dfx;
44 fy = (-N/2:1:N/2-1)*dfy;
45 [Fx,Fy] = meshgrid(fx,fy);
```

2.2 Étude numérique préliminaire

Cette étude préliminaire a pour objectif de vous permettre de prendre en main l'outil de calcul numérique de la transformée de Fourier discrète (algorithme de FFT) via les fonctions MATLAB `fft2` et `fftshift`. Vous pouvez vous référer à l'aide de MATLAB pour vérifier les bonnes conditions d'utilisation de ces fonctions. Les points essentiels à retenir pour le présent TP sont les suivants :

- ❶ Si T est une matrice de M colonnes et N lignes représentant les $M \times N$ échantillons d'un signal spatial transverse, alors $S = \text{fft2}(T) / (M \times N)$ sera une matrice de M colonnes et N lignes représentant le spectre de Fourier de T .
- ❷ **Cependant**, la fonction `fft2` retourne, dans la matrice S , les valeurs du spectre en commençant par celles correspondant aux fréquences spatiales positives (dans l'ordre croissant, de 0 à $(M/2-1) \times dfx$ pour les colonnes de S , et de 0 à $(N/2-1) \times dfy$ pour les lignes), puis celles correspondant aux fréquences spatiales négatives (là aussi en suivant leur ordre croissant, de $-(M/2) \times dfx$ à $-dfx$ pour les colonnes, et de $-(N/2) \times dfy$ à $-dfy$ pour les lignes) (voir annexe, figure A.1). Pour pouvoir représenter le spectre tel qu'il apparaîtrait sur un écran placé dans le plan de Fourier d'un montage $4f$, i.e. dans l'ordre des fréquences spatiales strictement croissantes à mesure que le numéro des lignes et des colonnes de la matrice S augmentent (de sorte que la valeur de spectre associée à la fréquence spatiale nulle soit positionnée au centre de la matrice), il faut donc ré-organiser les éléments de la matrice S (voir annexe, figure A.2). La ré-organisation adéquate est opérée à l'aide de la fonction `fftshift`, qu'il faut utiliser de la manière suivante : $S = \text{fftshift}(\text{fft2}(T) / (M \times N))$.

Remarque : Dans le présent TP, nous vous demanderons simplement de représenter graphiquement des grandeurs *normalisées* (que ce soit le spectre de Fourier de l'objet étudié, ou la répartition d'intensité lumineuse dans le plan image final). Vous pourrez donc omettre les facteurs $M \times N$ lors de l'utilisation de la fonction `fft2` :

```
1 % Calculate the normalized intensity spectrum:
2     amplitudeSpectrum = fftshift(fft2(OBJECT));
3     intensitySpectrum = abs(amplitudeSpectrum).^2;
4     intensitySpectrum = intensitySpectrum/max(max(intensitySpectrum));
```

À faire

- ❶ Créez un premier script MATLAB vous permettant de :
 - définir un objet d'amplitude carré de 20 mm de côté, discrétisé en 512×512 pixels, présentant une fonction de transmission sinusoïdale dans la direction horizontale (x), de période Δx (dont vous pourrez faire varier la valeur) ;
 - représenter graphiquement cet objet sur une figure MATLAB. Vous utiliserez pour cela la fonction `imagesc` et l'échelle de couleurs en niveaux de gris (`colormap(gray)`). Vous afficherez l'échelle de couleurs à l'aide de la commande `colorbar`. Les axes horizontal (x) et vertical (y) seront affichés en mm, et vous assurerez des échelles d'affichage identiques à l'aide de la commande `axis equal` ;
 - calculer le spectre de Fourier de cet objet (à l'aide des fonctions `fft2` et `fftshift`) ;
 - représenter graphiquement ce spectre sur une figure MATLAB. Vous représenterez le spectre en intensité, normalisé, en échelle logarithmique. Les axes horizontal (f_X) et vertical (f_Y), quant à eux, seront affichés en mm^{-1} . Vous pourrez aussi, pour visualiser les choses de manière plus confortable, représenter la coupe du résultat (en échelle linéaire et/ou logarithmique) le long d'une direction que vous choisirez judicieusement.
- ❷ Commentez le spectre obtenu.
- ❸ Faites varier la période Δx de la fonction de transmission. Commentez.
- ❹ Faites pivoter l'objet de 90° et représentez son spectre. Commentez.
- ❺ Remplacez l'évolution sinusoïdale de cette fonction de transmission par une évolution binaire rectangulaire avec un rapport cyclique de 50%. Vous pourrez par exemple combiner votre objet de transmission sinusoïdale avec une relation logique du type :

```
1     OBJECT_BINARY = OBJECT_SINUSOIDAL > 0.5;
```

- ❻ Représentez l'objet et son spectre sur deux figures MATLAB, et commentez.

2.3 Simulation numérique de filtrage optique d'une image

Vous allez maintenant simuler numériquement l'ensemble du système schématisé sur la figure 1.1 pour observer le résultat d'opérations de filtrage spatial sur un objet d'amplitude dont la fonction de transmission ne peut être décrite analytiquement.

L'objet à étudier est numériquement décrit par le fichier `ENSSAT_1750x1750.jpg` disponible sur la page Moodle du module d'Optique de Fourier [1]. Une reproduction en est donnée en figure 2.1.



FIG. 2.1 – *Objet à étudier.*

À faire

- ❶ Créez un second script MATLAB vous permettant de :
 - charger le fichier `ENSSAT_1750x1750.jpg` (fonction `imread`) et le convertir en une matrice de nombres compris entre 0 et 1 (fonction `double` et application d'une normalisation) ;
 - définir de manière appropriée les espaces de simulation numérique (domaine spatial et domaine spectral). Vous considérerez que chaque pixel du fichier `.jpg` correspond à un carré de $100\ \mu\text{m}$ de côté ;
 - représenter graphiquement l'objet étudié sur une figure MATLAB. Les axes horizontal (x) et vertical (y) seront affichés en mm ;
 - calculer le spectre de Fourier de cet objet ;
 - représenter graphiquement ce spectre sur une figure MATLAB. Vous représenterez le spectre en intensité, normalisé, en échelle logarithmique. Les axes horizontal (f_X) et vertical (f_Y), quant à eux, seront affichés en mm^{-1} .

- ② Décrivez le spectre obtenu. En vous appuyant sur les observations réalisées lors de l'étude préliminaire, discutez de l'origine des différents points de forte intensité apparaissant dans ce spectre.
- ③ Modifiez votre script de sorte à représenter graphiquement la distribution transverse d'intensité lumineuse (normalisée) que l'on obtiendrait si l'on plaçait un écran d'observation dans le plan de Fourier du montage 4f. Il s'agit d'appliquer la transformation adéquate aux coordonnées des axes horizontal et vertical pour qu'elles ne représentent plus les fréquences spatiales f_X et f_Y mais les dimensions transverses dans le plan focal image de la lentille L_a . Vous considérerez pour cela que la longueur d'onde d'éclairage est égale à 500 nm, et que la distance focale de la lentille L_a est égale à 1 m. Représentez le résultat en échelle linéaire (i.e. tel qu'on l'observerait, strictement, dans le montage réel) et en échelle logarithmique, et commentez les figures ainsi obtenues.
- ④ En l'absence de filtrage, calculez et représentez la distribution transverse d'intensité lumineuse (normalisée, et en échelle linéaire) que l'on obtiendrait si l'on plaçait l'écran d'observation dans le plan focal image de la lentille L_b . Vous supposerez pour cela que la distance focale de cette lentille est égale à 10 cm.
- ⑤ Commentez le résultat obtenu.
- ⑥ Créez trois fonctions MATLAB permettant, respectivement, de :
 - créer un filtre de type “fente horizontale” dont on peut faire varier la position et la largeur ;
 - créer un filtre de type “fente verticale” dont on peut faire varier la position et la largeur ;
 - créer un filtre de type “masque opaque carré” dont on peut faire varier la position et la largeur.
- ⑦ À l'aide de ces trois fonctions, modifiez votre script de sorte à ce que l'image finale ait, alternativement, les caractéristiques suivantes :
 - ① Toutes les rayures horizontales sont supprimées ; toutes les lettres restent visibles.
 - ② Toutes les rayures verticales sont supprimées ; toutes les lettres restent visibles.
 - ③ Toutes les rayures sont supprimées ; toutes les lettres restent visibles.
 - ④ Toutes les rayures de la ligne inférieure sont supprimées ; toutes les lettres restent visibles.
 - ⑤ La ligne inférieure a disparu.
 - ⑥ Les voyelles ont disparu.
 - ⑦ Les consonnes ont disparu.
 - ⑧ La ligne supérieure a disparu.

Annexe

Effet de la fonction MATLAB `fftshift`

Soit T un signal spatial transverse quelconque, décrit par de manière discrète par une matrice T .

On note $\mathcal{S}(f_X, f_Y)$ la valeur du spectre de Fourier de ce signal pour le couple de fréquences spatiales (f_X, f_Y) . Une version discrétisée de ce spectre peut être calculée numériquement à l'aide des fonctions `fft2` et `fftshift` de MATLAB. On notera δf_X et δf_Y le pas d'échantillonnage spectral dans chacune des deux directions transverses.

Les figures A.1 et A.2 indiquent l'arrangement des valeurs de $\mathcal{S}(f_X, f_Y)$ dans la matrice résultant du calcul numérique utilisant `fft2(T)` et dans la matrice résultant du calcul numérique utilisant `fftshift(fft2(T))`, respectivement.

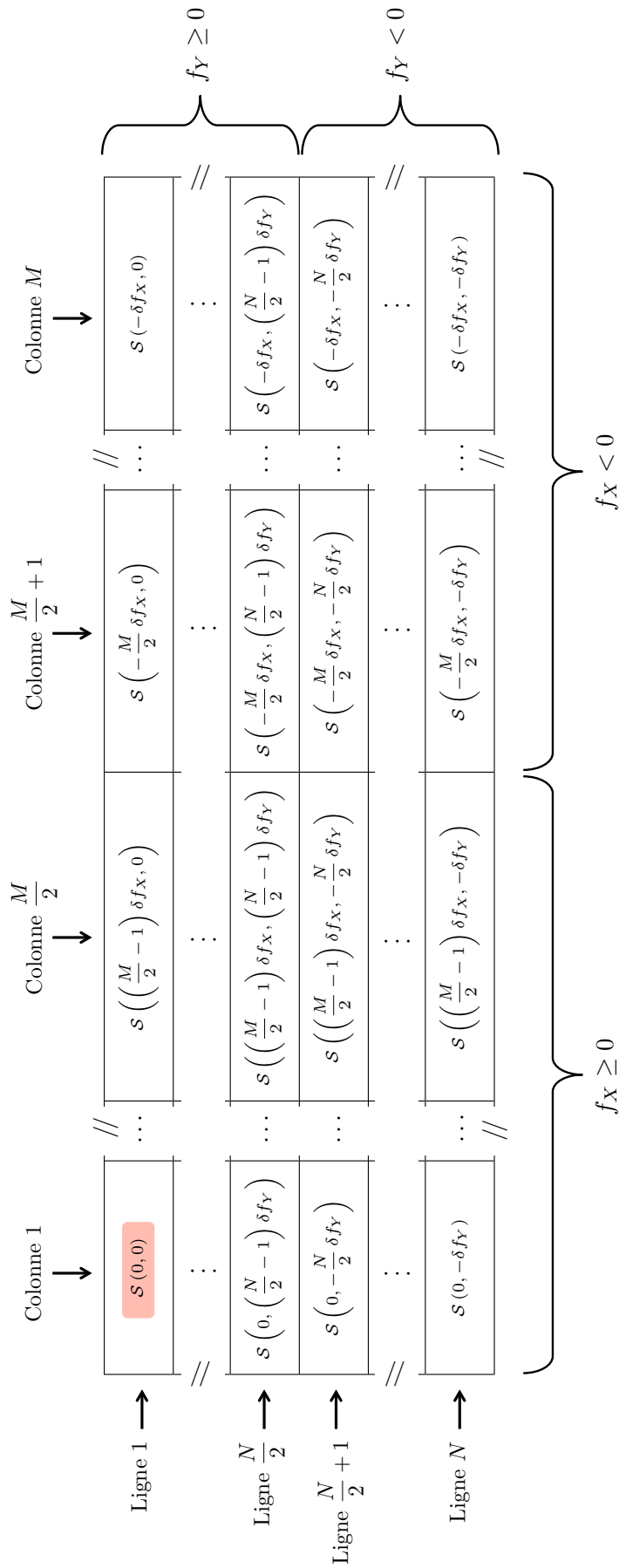


FIG. A.1 – Lorsque le spectre de T est calculé simplement à l'aide de fft2 (T), les $M/2$ premières colonnes de la matrice résultat contiennent les valeurs associées aux fréquences spatiales f_x positives (dans l'ordre croissant) et les $M/2$ colonnes suivantes contiennent les valeurs associées aux fréquences spatiales f_x négatives (dans l'ordre croissant également). Il y a donc une discontinuité dans l'incrément des valeurs de f_x entre la colonne $M/2$ et la colonne $(M/2) + 1$. Le comportement est similaire pour ce qui concerne les lignes (associées aux fréquences spatiales f_y). La valeur du spectre associée au couple de fréquences spatiales ($f_x = 0, f_y = 0$) est donc contenue dans le tout premier élément de la matrice (élément d'indices $(1, 1)$).

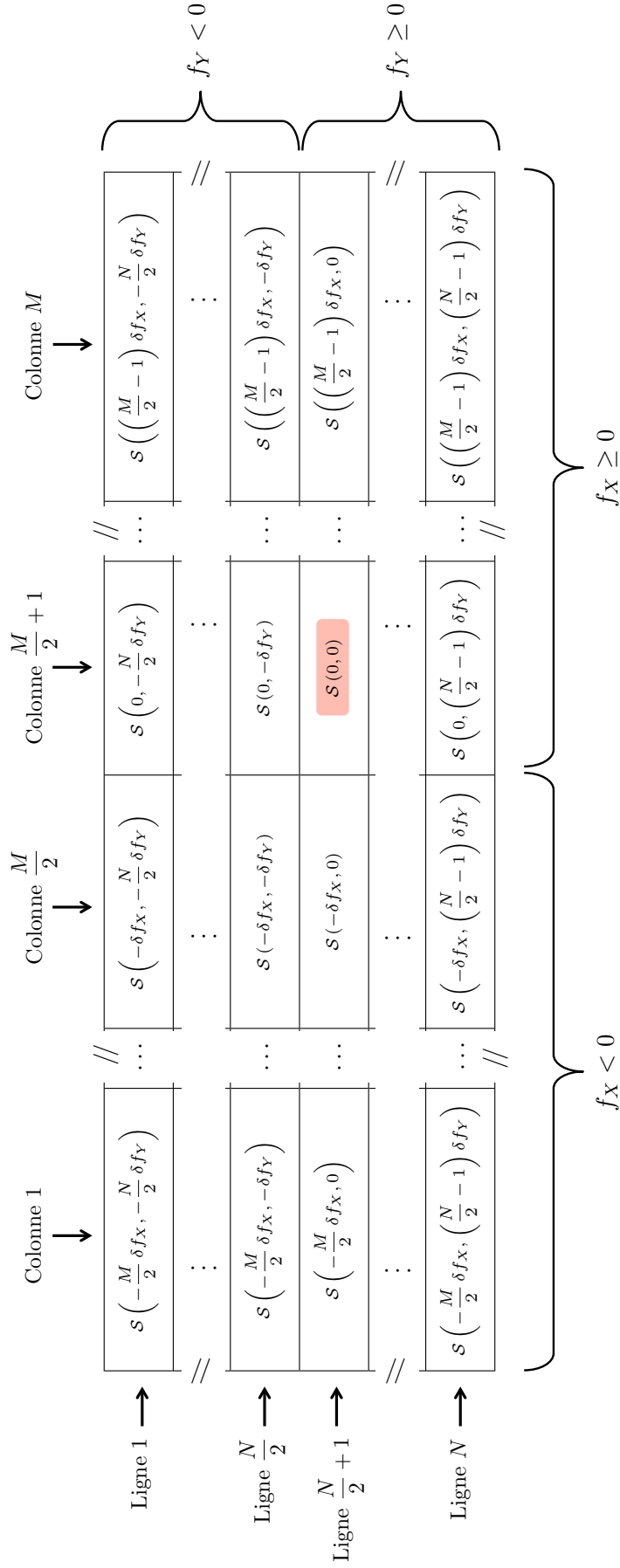


FIG. A.2 – Appliquée sur $\text{fft2}(T)$, la fonction fftshift a pour effet de ré-arranger les éléments de matrice de sorte à ce que les fréquences spatiales f_x soient strictement croissantes avec l'indice de colonne, et les fréquences spatiales f_y strictement croissantes avec l'indice de ligne. La valeur du spectre associée au couple de fréquences spatiales ($f_x = 0, f_y = 0$) est alors située au centre de la matrice (élément de matrice d'indices $(\frac{M}{2}+1, \frac{N}{2}+1)$).

Bibliographie

- [1] M. Barbier, *Optique de Fourier* (Moodle ENSSAT, PHOT2, 2022-2023). [online access]
- [2] G. Dubois, *La Simulation numérique – Enjeux et bonnes pratiques pour l'industrie* (Dunod, 2016). [online access]
- [3] J. W. Goodman, *Introduction to Fourier Optics* (McGraw-Hill, 1996, 2nd ed.).

