Traitement Numérique du Signal

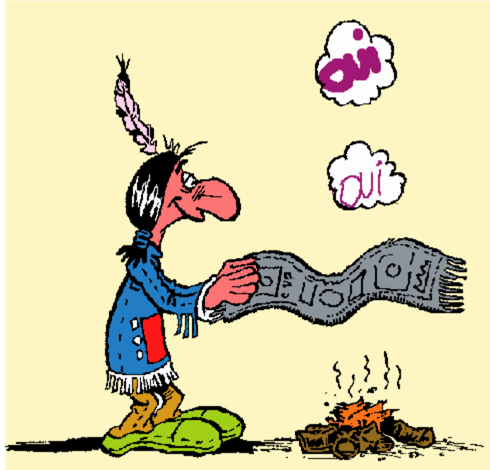
Nathalie Thomas

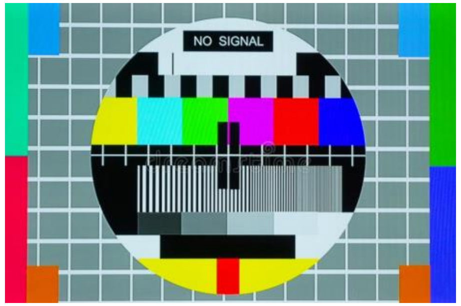
IRIT/ENSEEIHT
Nathalie.Thomas@enseeiht.fr

Traitement Numérique du Signal
INTRODUCTION

Qu’est-ce qu’un signal ?

→ Formes multiples et variées de signaux

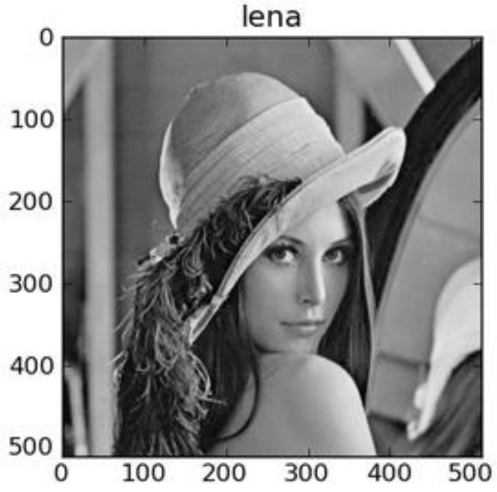




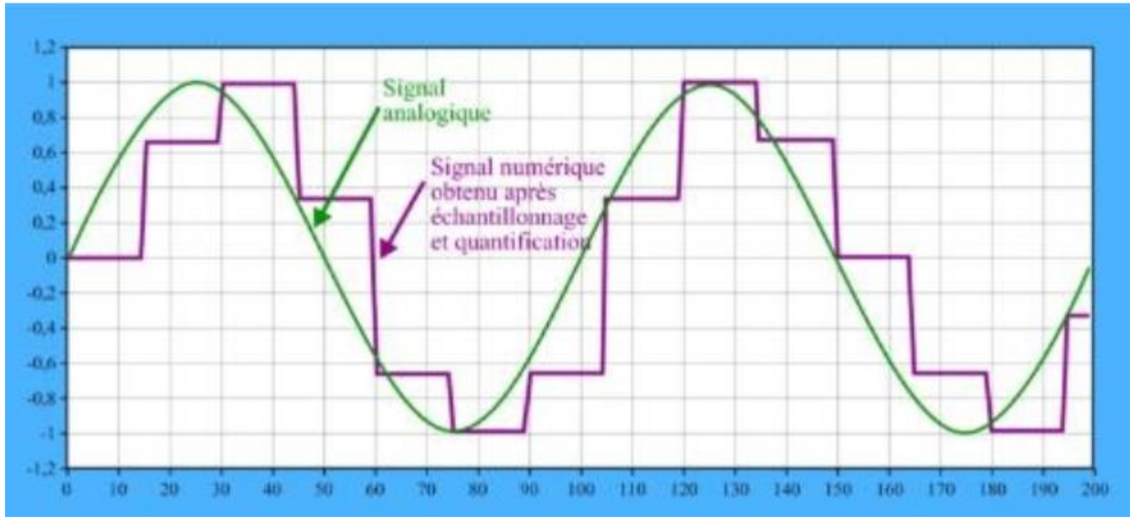
Point commun : représentent un message, contiennent une information.

Qu’est-ce qu’un signal ?

→ <u>Représentation théorique :</u> x(t), I(x,y)…



→ <u>Signaux analogiques, signaux numériques</u> (échantillonnage, quantification)

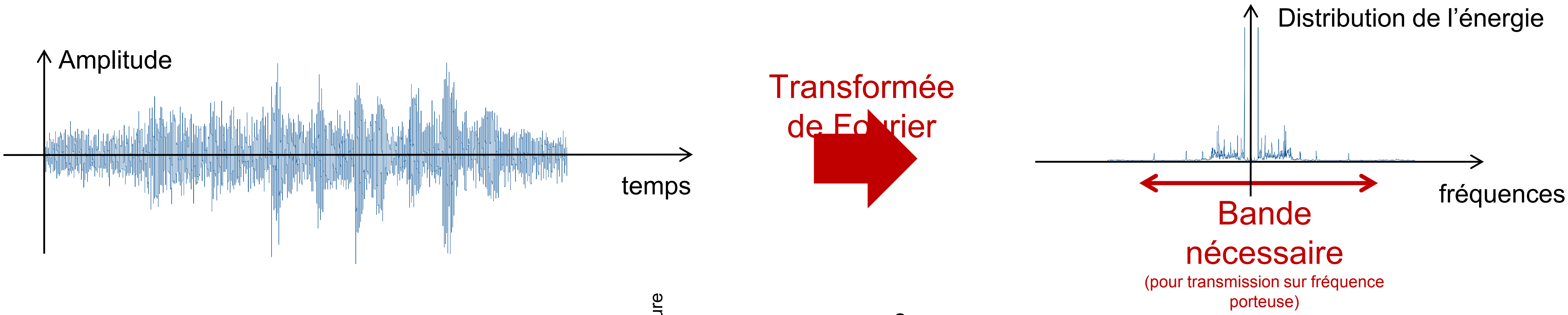


Le traitement du signal : pourquoi ?

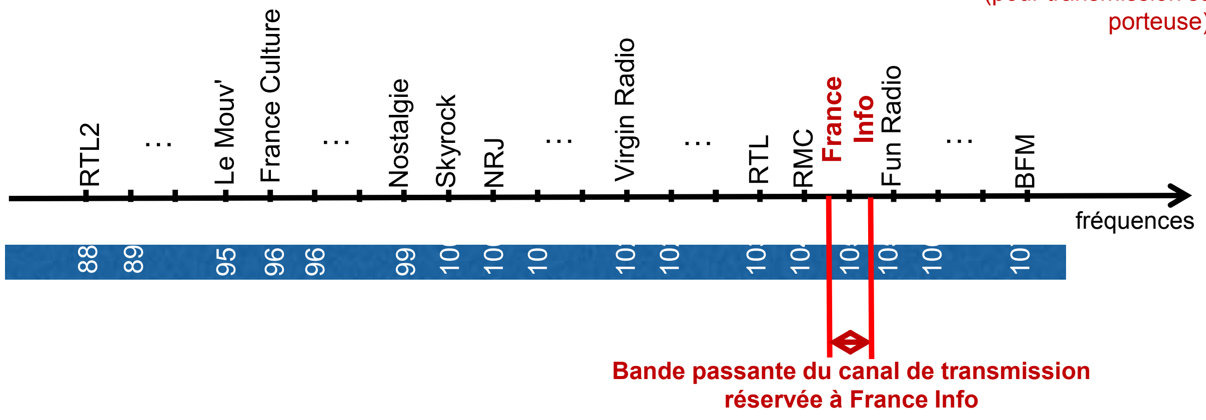
Le traitement du signal : pourquoi ?

Exemple 1

- Identifier la bande de fréquence nécessaire à la transmission d'un signal,



Exemple de partage de bande : bande FM à Toulouse



Le traitement du signal : pourquoi ?

Exemple 2

- Eliminer des composantes indésirables : le bruit, certaines fréquences…

SNR = 0 dB

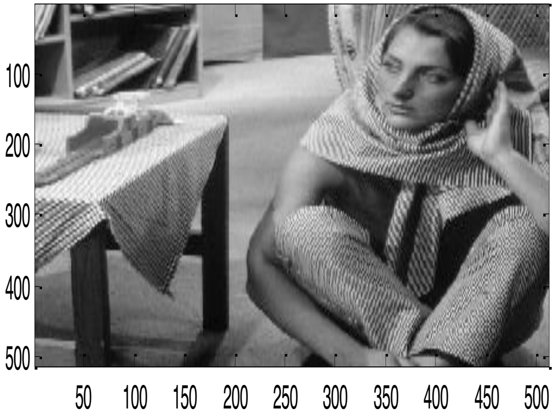


Image émise

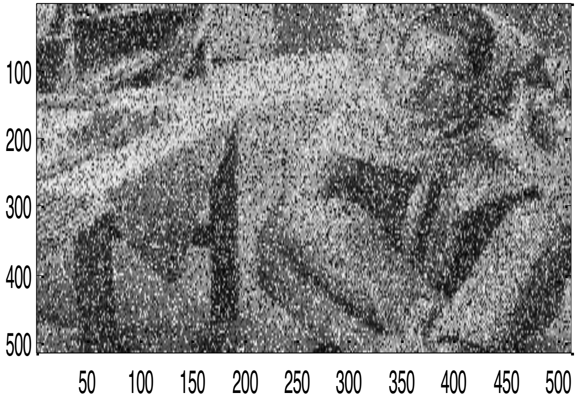


Image reçue
Avant filtrage de réception

TEB = 0.1581

Filtrage

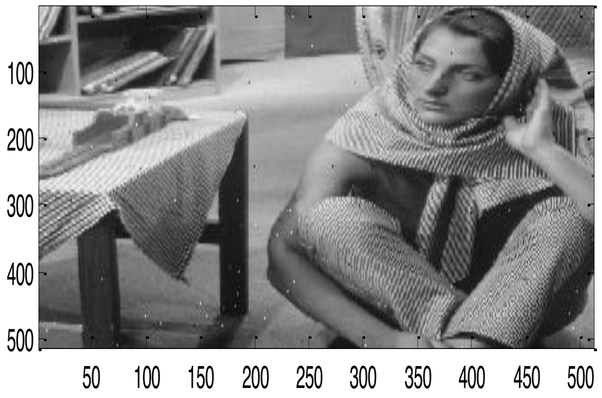


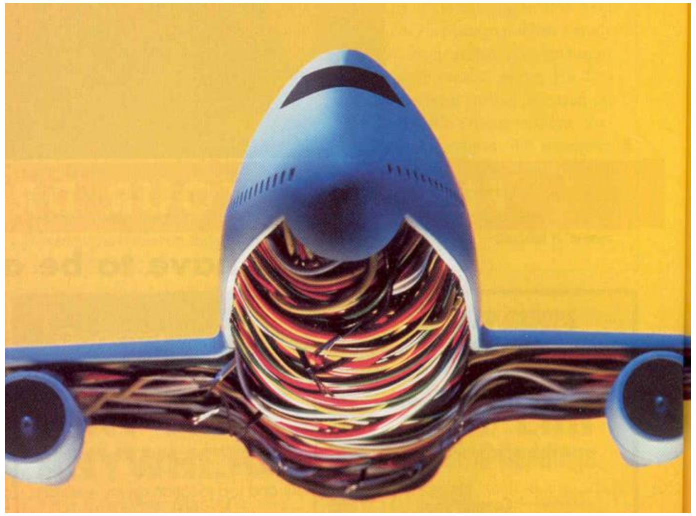
Image reçue
après filtrage de réception

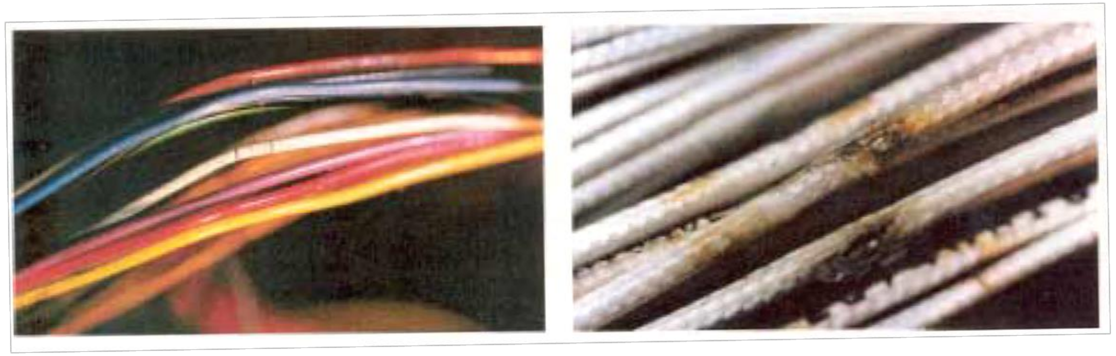
TEB = 7.5483e-04

Le traitement du signal : pourquoi ?

Exemple 3

Détecter des anomalies, des défauts (ECG, Arcs électriques sur les câbles d'alimentation d'un avion, dent cassée dans un engrenage…)
En éliminant des composantes indésirables : le bruit, certaines fréquences…





Usure des gaines d'isolation

$\Rightarrow$ Arcs électriques

$\Rightarrow$ Possible destruction d'une partie du réseau d'alimentation de l'avion.

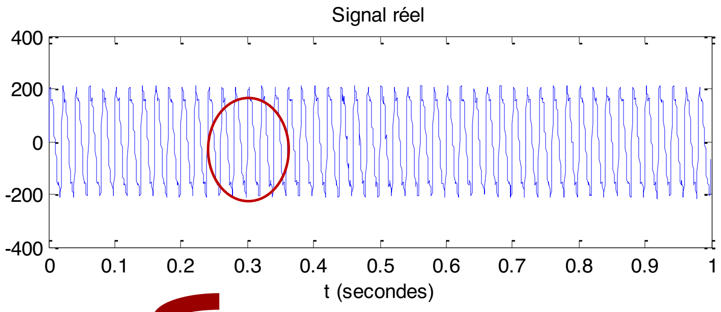
Le traitement du signal : pourquoi ?

Exemple 3

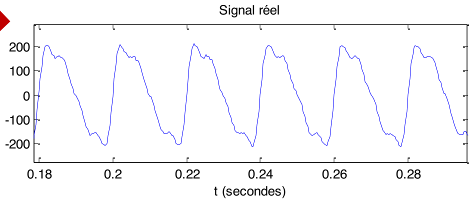
Détecter des anomalies, des défauts (ECG, Arcs électriques sur les câbles d'alimentation d'un avion, dent cassée dans un engrenage…)
En éliminant des composantes indésirables : le bruit, certaines fréquences…

Détection de perturbations « annonciatrices »
(de fréquences > 500 Hz noyées dans le 50 Hz)

Signal d'alimentation contenant la perturbation

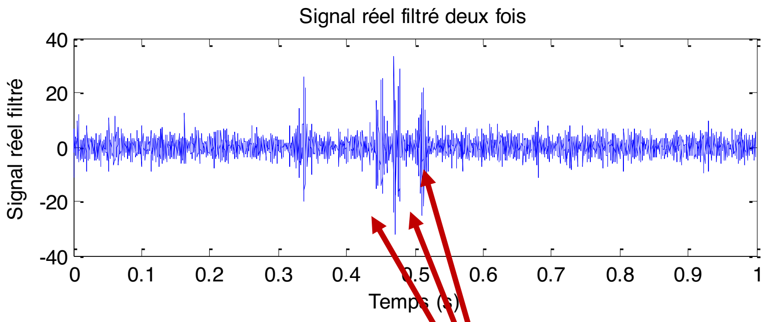


Zoom



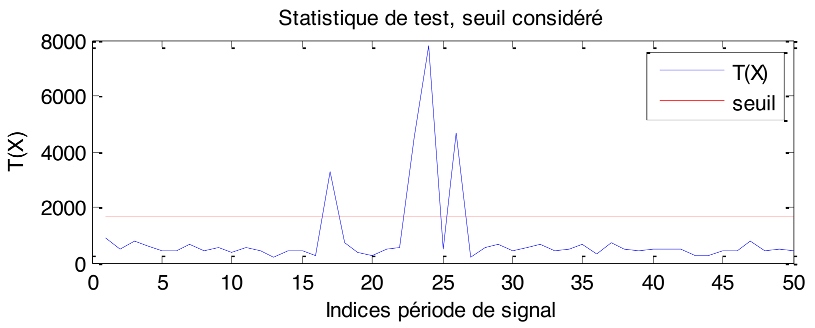
Filtrage

Signal après filtrage passe-haut
(élimination du 50 Hz)



Transitoires à détecter

Détecteur d'énergie



Le traitement du signal a besoin d'outils…

Partie 1 : Signaux et Systèmes à temps continu

Cours 1 à 4

- Modèles de signaux,
- Outils pour la représentation et l'analyse de signaux :
  + Représentation fréquentielle ou « spectre » (Transformée de Fourier, Densité Spectrale de Puissance : DSP),
  + Fonctions d'inter et d'autocorrélation.
  + Filtrage (linéaire, non linéaire) des signaux à temps continu.

TD1

Etude de différentes modélisations d'un signal, calcul de fonctions d'autocorrélation et de spectres (TF, DSP)

TD2

Exercices sur le filtrage linéaire et non linéaire.

…qui doivent être implantés en numérique

Partie 2 : Signaux et Systèmes à temps discret

Cours 5 à 7

- Numérisation des signaux : échantillonnage, quantification.
- Numérisation des outils pour la représentation et l'analyse de signaux (Transformée de Fourier Discrète, DSP et fonctions d'inter et d'autocorrélation numériques).
- Définition et implantation de filtres numériques.

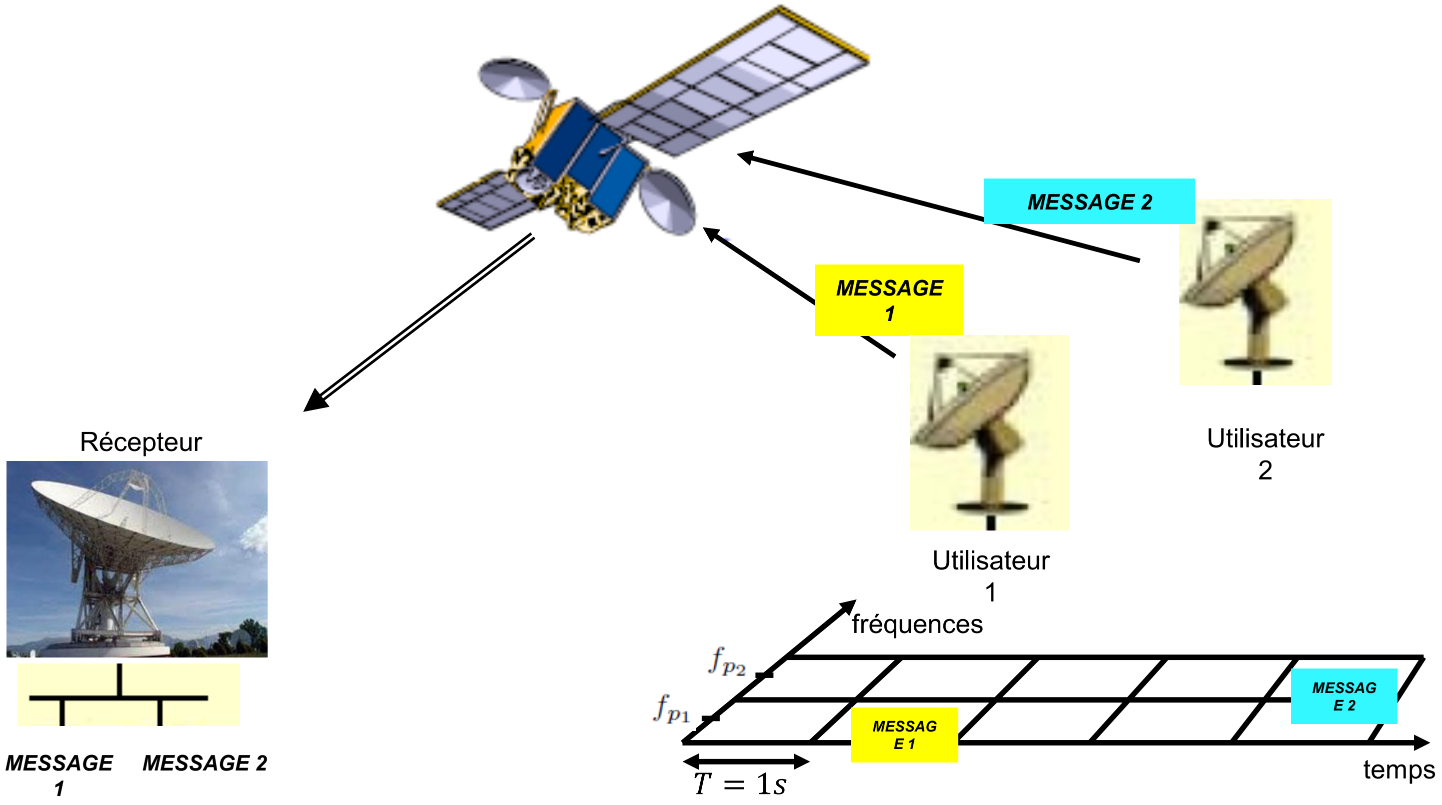
TD3

Etude de l'échantillonnage (impact, échantillonnage non idéal)

Mise en pratique (TPs et Projet signal/telecom)

TP1 : « Signaux et spectres »
TP2 : « Filtrage numérique »
Projet signal/télécom : « Simulation d'une transmission voie retour par satellite au format type DVB-RCS (Digital Video Broadcasting – Return Channel by Satellite, norme ETSI)

Simulation d’une transmission voie retour par satellite type DVB-RCS (norme ETSI)



MF-TDMA
(Multiple Frequency – Time Division Multiple Access)

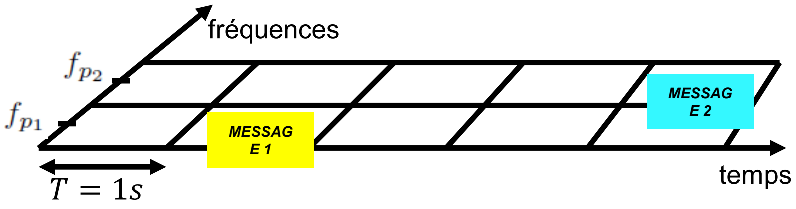
Projet Signal Telecom

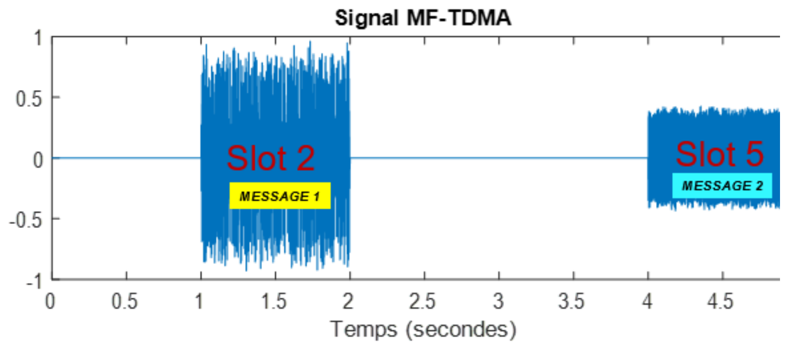
Simulation d'une transmission voie retour par satellite type
DVB-RCS (norme ETSI)

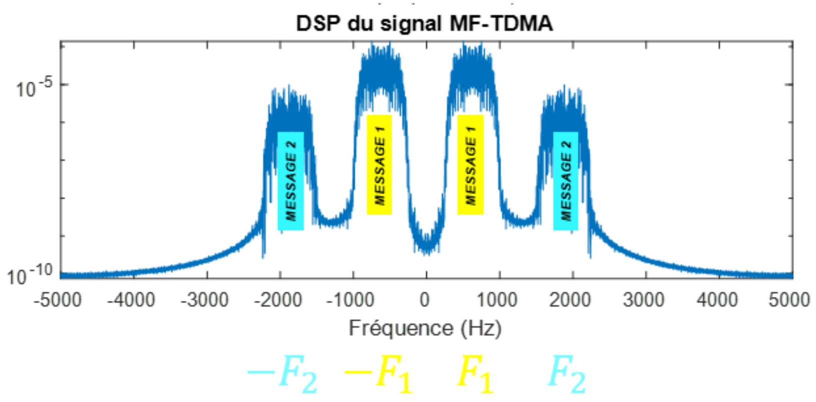
Modulation bande de base pour chaque utilisateur

= passage de l'information binaire à un signal « bande de base » pour chaque utilisateur

Construction de la trame MF-TDMA



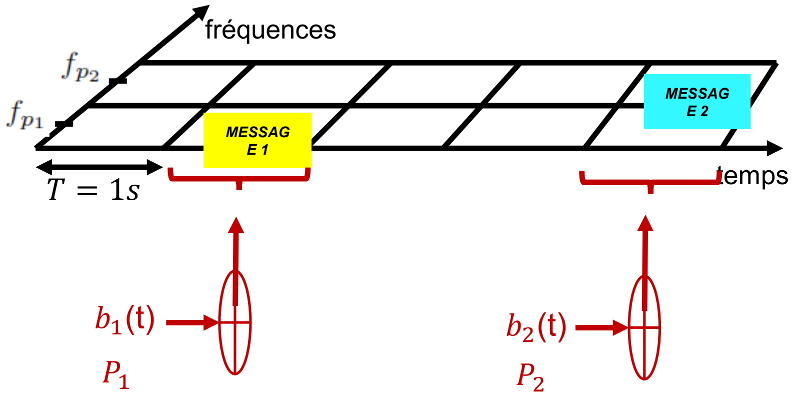




Projet Signal Telecom

Simulation d'une transmission voie retour par satellite au format DVB-RCS (norme ETSI)

Simulation du canal satellite (AWGN)

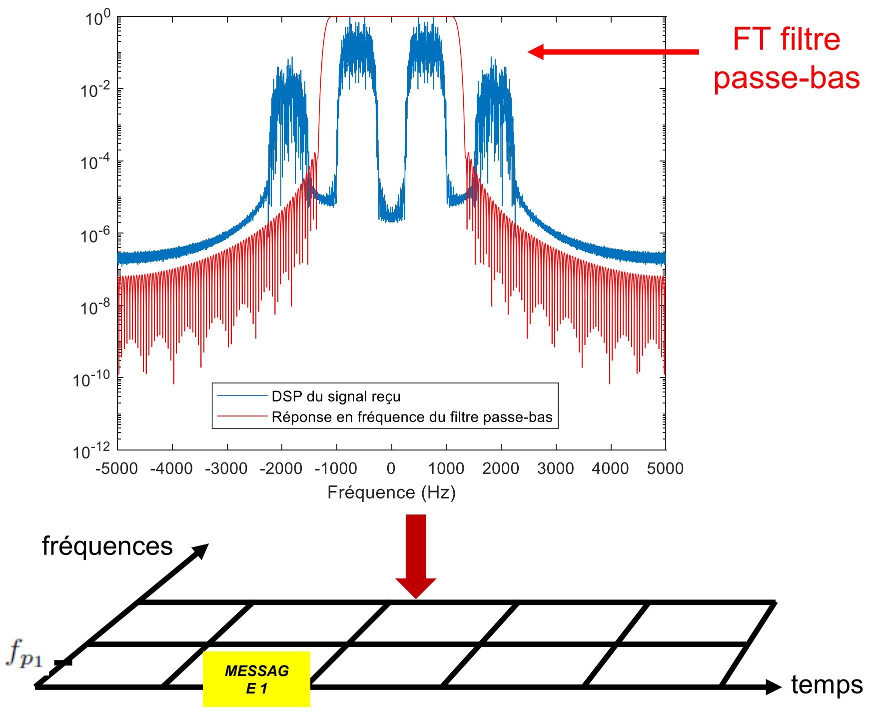


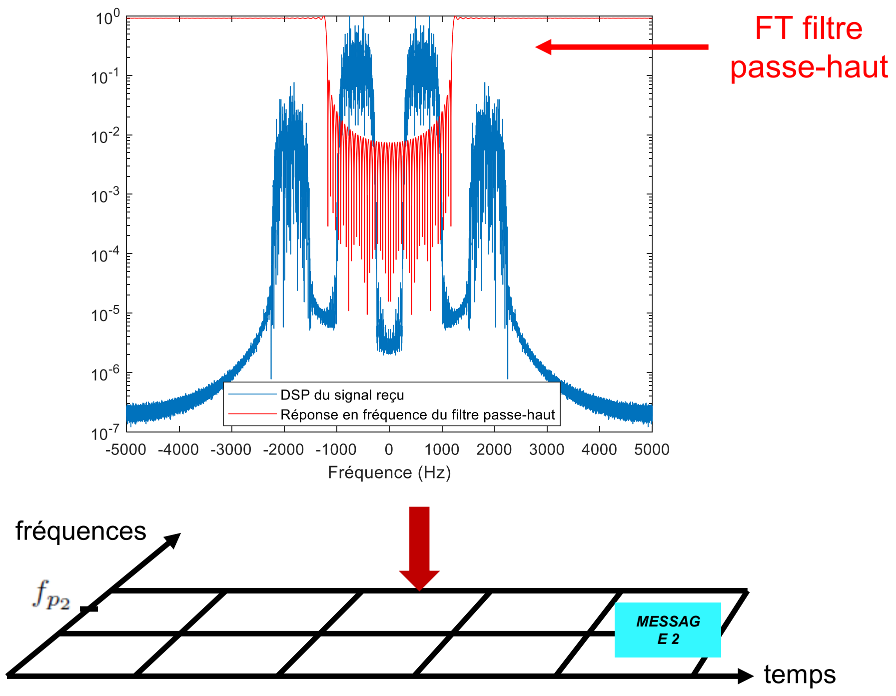
P_1 et P_2 choisies pour fonctionner au même rapport signal à bruit par bit à l'entrée du récepteur

Projet Signal Telecom

Simulation d'une transmission voie retour par satellite au format DVB-RCS (norme ETSI)

Demultiplexage des porteuses





Des traitements en temps réel ?

→ Temps de calcul en traitement numérique du signal :

Nombre d'opérations d'addition/multiplication

(MAC = Multiplication Accumulation)

→ Temps réel

y(n) est calculé avant que x(n+1) ne se présente

(T_e secondes entre deux x(n) et x(n+1))

$$x(n) \longrightarrow \boxed{\text{Traitement numérique}} \longrightarrow y(n)$$

Exemples :

→ Estimation biaisée de la fonction d'autocorrélation de x:

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)\times x^*(k-n), \;\; n = 0, ..., N-1$$

→ Transformée de Fourier Discrète (TFD) de x:

$$y(n) = \sum_{k=0}^{N-1} x(k)\times e^{-j2\pi\frac{kn}{N}}, \;\; n = 0, ..., N-1$$

→ Filtrage numérique à réponse impulsionnelle finie de x :

$$y(n) = \sum_{k=0}^{N-1} b_k\times x(n-k), \;\; n = 0, ..., N-1$$

Références

→"Traitement numérique du signal, théorie et pratique", M. Bellanger, Masson, collection CNET-ENST.

→"Traitement numérique des signaux", M. Kunt, Dunod, Traité d'électricité, d'électronique et d'électrotechnique.

→"Traitement numérique du signal, Une introduction", A.W.M. Van Den Enden et N.A.M. Verhoeckx, Masson

→"Introduction au traitement du signal", P. Duvaut, F. Michaut, M. Chuc, Hermes, Collection traitement du signal

→Documents sur la variable complexe, la transformée de Laplace et la transformée en z : http://dobigeon.perso.enseeiht.fr/teaching/complexe.html

→"Introduction to digital filters, with audio applications", J.O. Smith, BookSurge, 2007

→"Digital signal processing : fundamentals and applications", Tan Li, Jiang Jean, Elsevier, 2013.

→Alan V. Oppenheim, Ronald W. Schafer et J. R. Buck, Discrete-time signal processing, Upper Saddle River, N.J., Prentice Hall, 3^{ième} édition, 2009.

→Signal and Systems, by Simon Haykin and Barry Von Veen, Wiley, 2nde édition, 2002.

→John G. Proakis, Dimitri G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, Pearson Education, 4^{ième} édition, 2006.

Traitement Numérique du Signal

Nathalie Thomas

IRIT/ENSEEIHT
Nathalie.Thomas@enseeiht.fr

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d’inter et d’auto corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

Numérisation du signal

Signal analogique :

signal défini à tout instant par des valeurs réelles

Signal numérique :

signal défini à des instants discrets par un nombre fini de valeurs

Numérisation du signal

Signal analogique :
signal défini à tout instant par des
valeurs réelles

⬇

Numérisation :
Échantillonnage + quantification

⬇

Signal numérique :
signal défini à des instants discrets
par un nombre fini de valeurs

Numérisation du signal : Echantillonnage

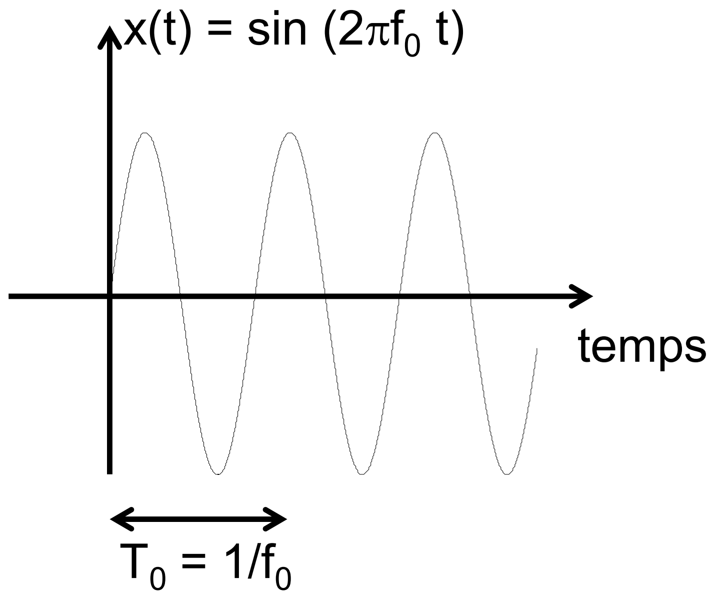
Signal échantillonné :

signal défini à des instants discrets par des valeurs réelles

Numérisation du signal : Echantillonnage

Signal échantillonné :
signal défini à des instants discrets par des valeurs réelles

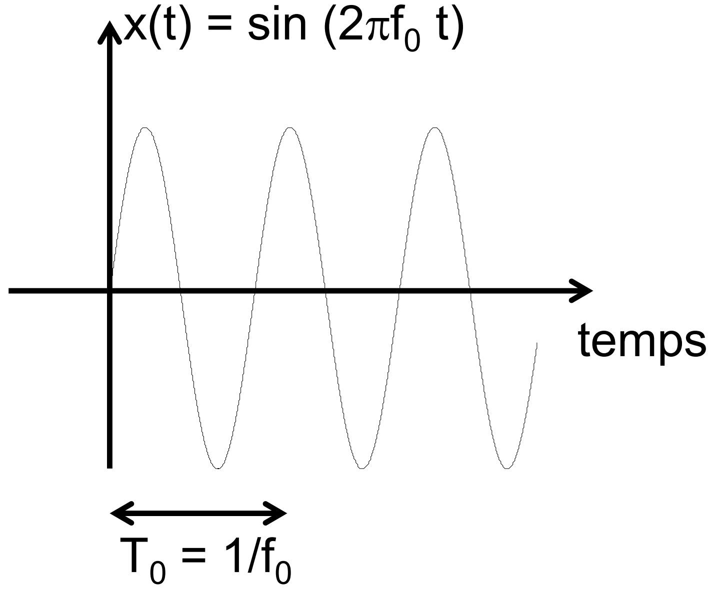
Exemple :



Numérisation du signal : Echantillonnage

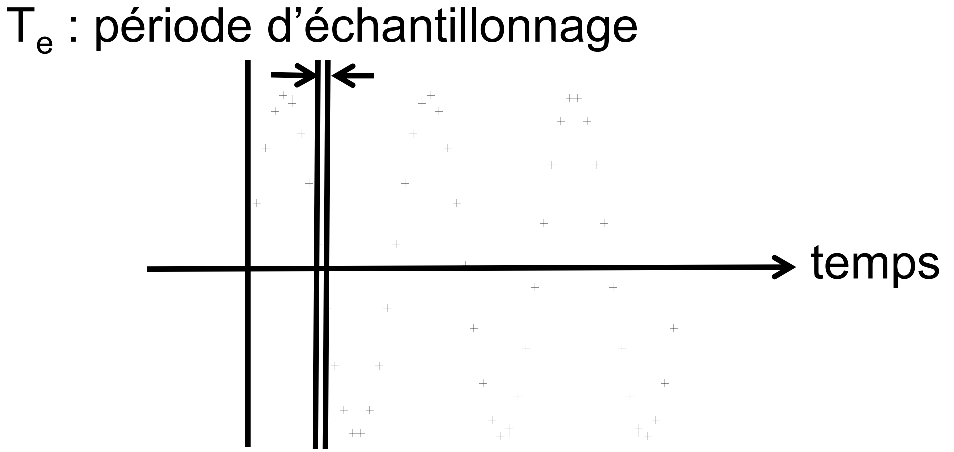
Signal échantillonné :
signal défini à des instants discrets par des valeurs réelles

Exemple :



k = indice de l'échantillon

$x(kT_e) = \sin(2\pi f_0\, kT_e)$



Numérisation du signal : Echantillonnage

Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

0. Slide de connexion

1 **Allez sur wooclap.com**

2 **Entrez le code d'événement dans le bandeau supérieur**

Code d'événement
SIGSEQ1

https://app.wooclap.com/SIGSEQ1

Copier le lien de participation

QUESTION 1

Allez sur **wooclap.com** et utilisez le code **SIGSEQ1**

Soit un signal x(t) échantillonné à Te. Est-il possible de garder tout l'information du signal de départ dans la suite d'échantillon ?

1. Oui
2. Non

Numérisation du signal : Echantillonnage

Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\,Ш_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * Ш_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Numérisation du signal : Echantillonnage

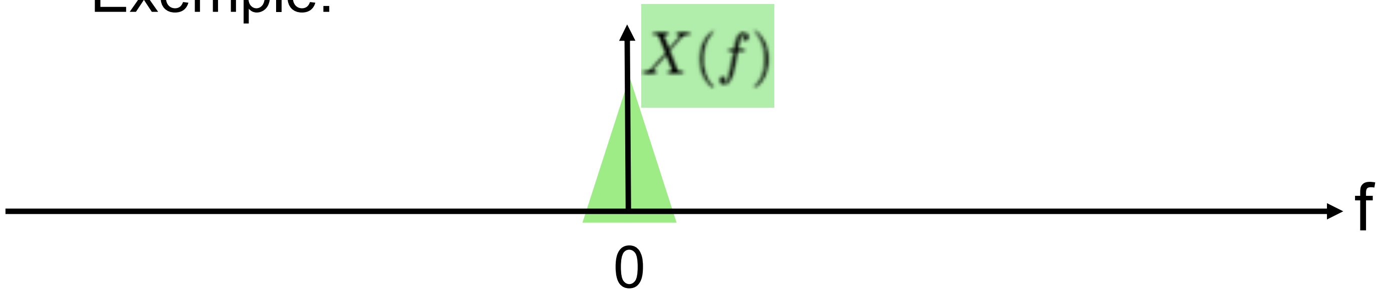
Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\,\text{Ш}_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * \text{Ш}_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Exemple:



Numérisation du signal : Echantillonnage

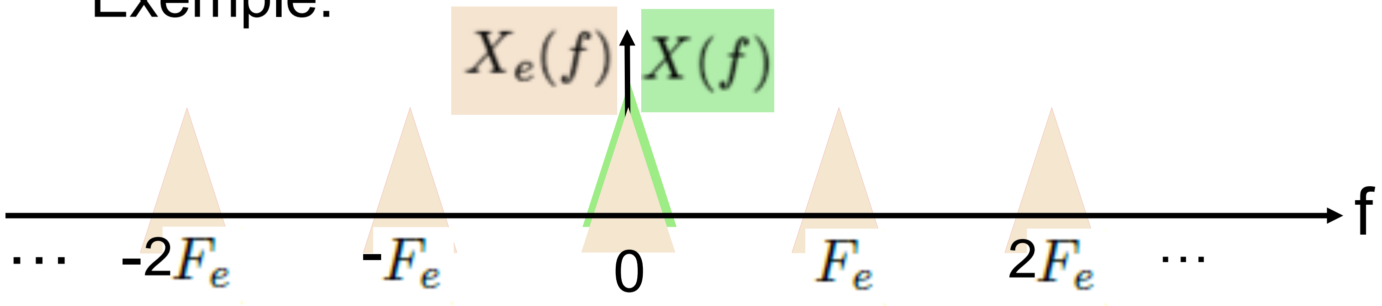
Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\, Ш_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * Ш_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Exemple:



Numérisation du signal : Echantillonnage

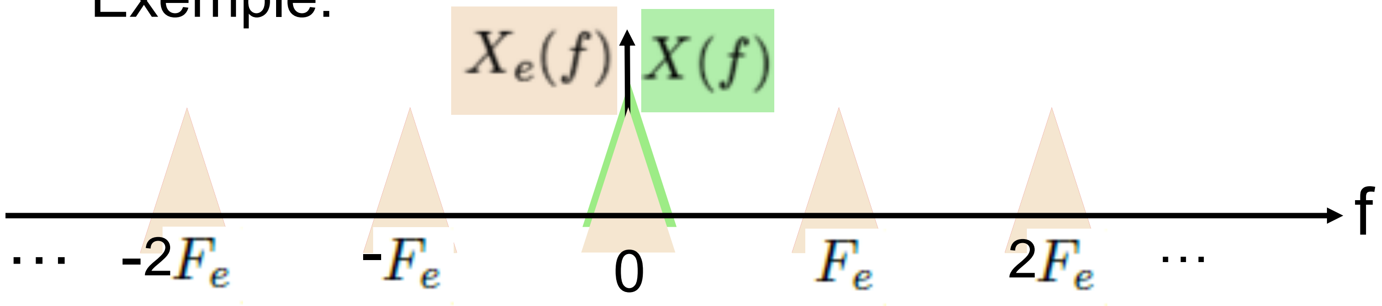
Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\, Ш_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * Ш_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Exemple:



QUESTION 2

Allez sur **wooclap.com** et utilisez le code **SIGSEQ1**

Soit un signal x(t) échantillonné à Te. Est-il possible de garder tout l'information du signal de départ dans la suite d'échantillon ?

1. Oui
2. Non

Numérisation du signal : Echantillonnage

Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\,Ш_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * Ш_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Oui ça l'est !

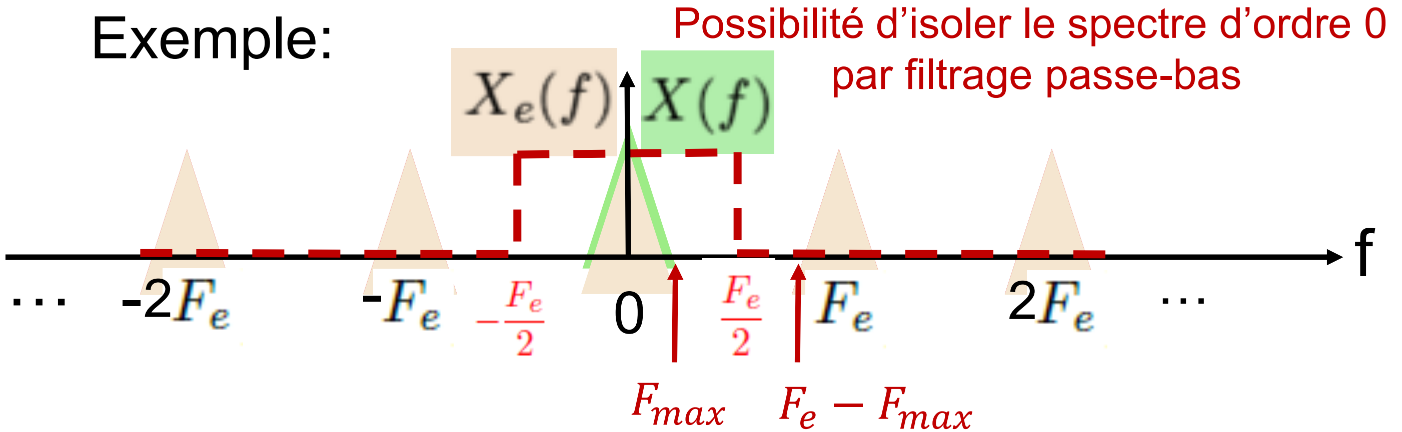
A une condition :

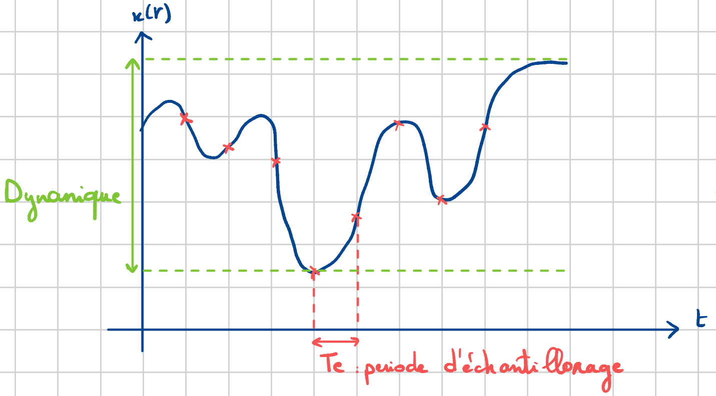
$$F_e = \frac{1}{T_e} > 2F_{max}$$

Condition de Shannon

Claude Elwood Shannon (1916 -2001)

Exemple:



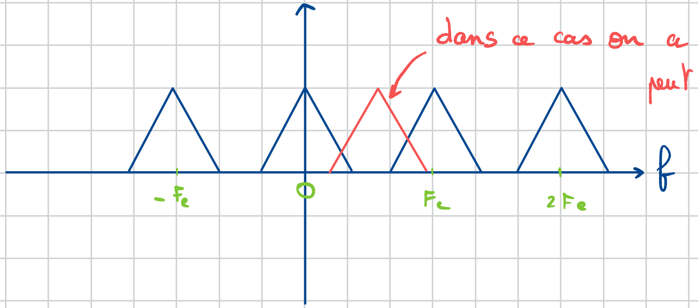


$x = [x(0) \;\; x(1) \; \ldots\ldots \; x(N-1)]$

$= [x(0) \;\; x(T_e) \; \ldots \; x((N-1)T_e)]$

$$x_e(t) = x(t) \; Ш_{T_e}(t) \xrightarrow{TF} X_e(f) = X(f) * \frac{1}{T_e} Ш_{\frac{1}{T_e}}(f) = F_e \, X(f) * \sum_k \delta(f - kF_e) = F_e \sum_k X(f - kF_e)$$

Avec : $F_e = \frac{1}{T_e}$ = fréquence d'échantillonage



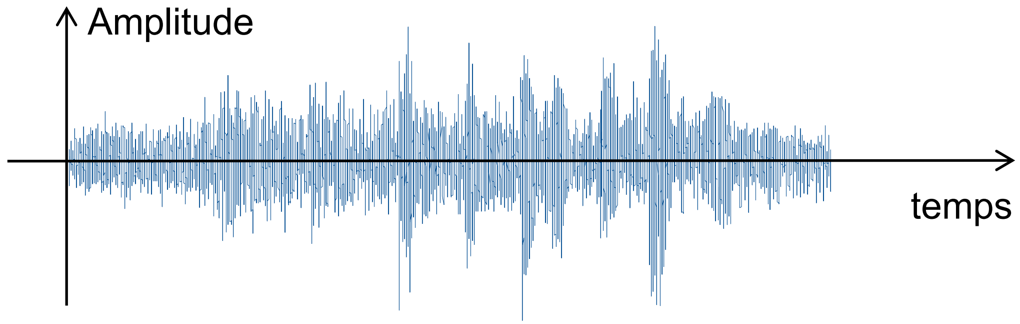
dans ce cas on a un recouvrement et on peut pas récuper le signal

Pour reconstituer mon signal :

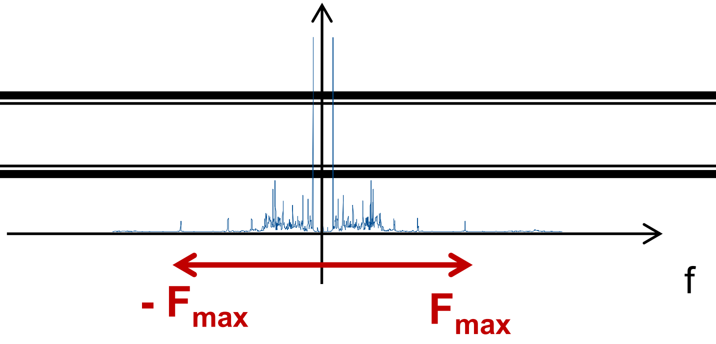
$$X_e(f) \cdot \Pi_{F_e}(f) \xrightarrow{TF^{-1}} x_e(t) * F_e \,\text{sinc}(\pi t F_e) = x(t) \sum_k \delta(t - T_e) * F_e \,\text{sinc}(\pi t F_e) = \sum_k x(kT_e)\,\delta(t - kT_e) * F_e \,\text{sinc}(\pi t F_e) = \overbrace{F_e \sum x(kT_e)\,\text{sinc}(\pi(t - kT_e)F_e)}^{\text{signal reconstruit}}$$

$\boxed{F_e > 2 F_{max}}$ Condition de Shanon

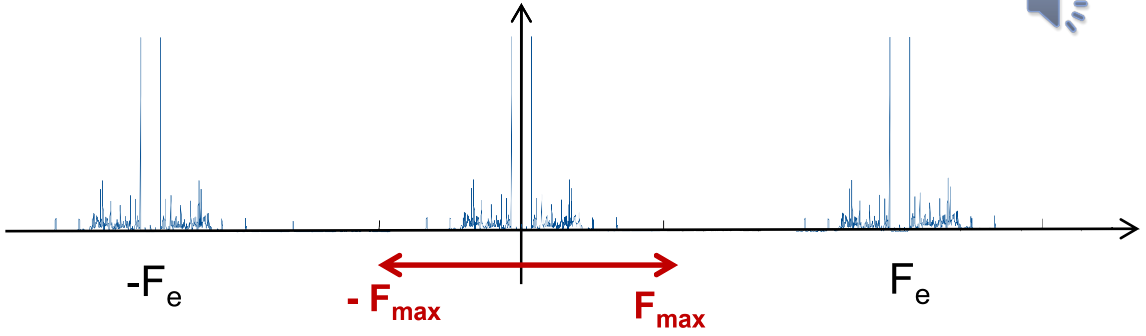
Exemple 1



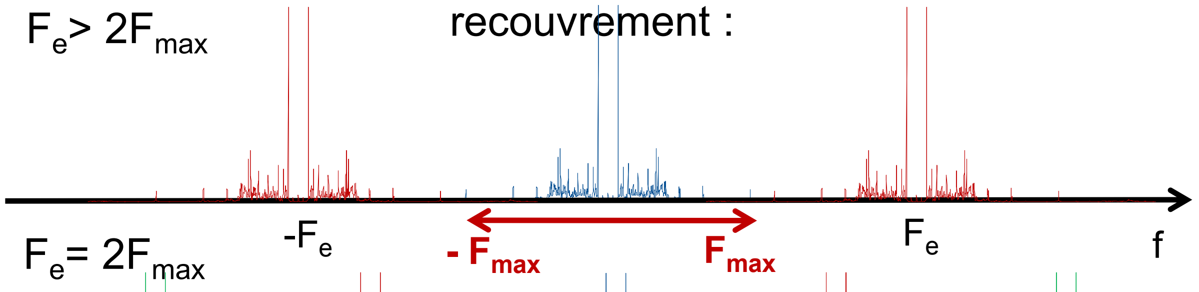
Transformée de Fourier :

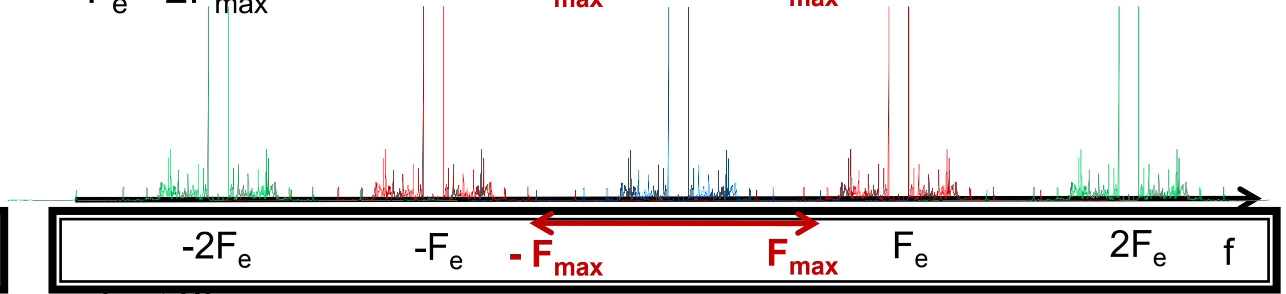


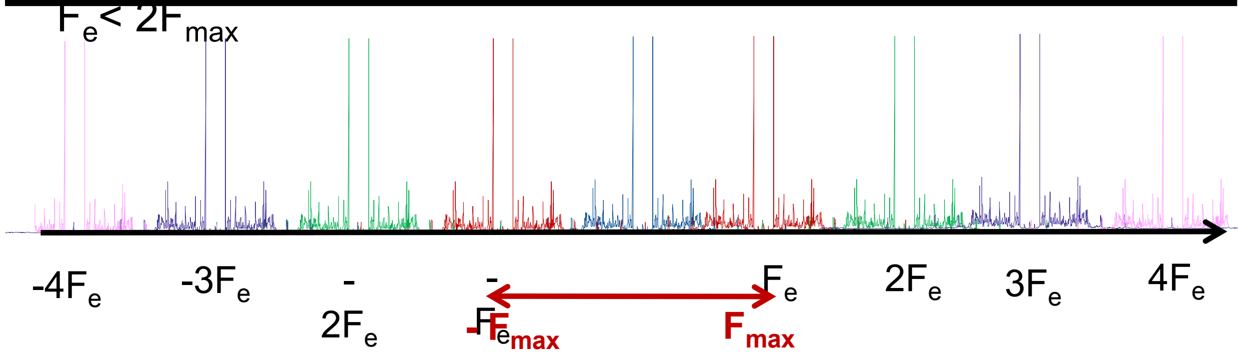
Transformée de Fourier du signal correctement échantillonné :



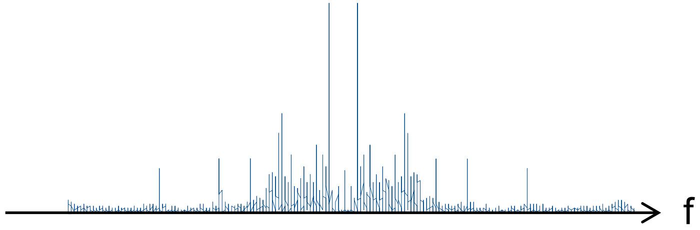
Quand on diminue la fréquence d'échantillonnage, les périodisations se rapprochent et finissent par apparaitre dans la bande occupée par le signal générant du recouvrement :







Transformée de Fourier du signal échantillonné à $F_e = F_{max} / 12$:



Exemple 2

Image d'origine :
512*512 pixels, quantifiée sur 8 bits

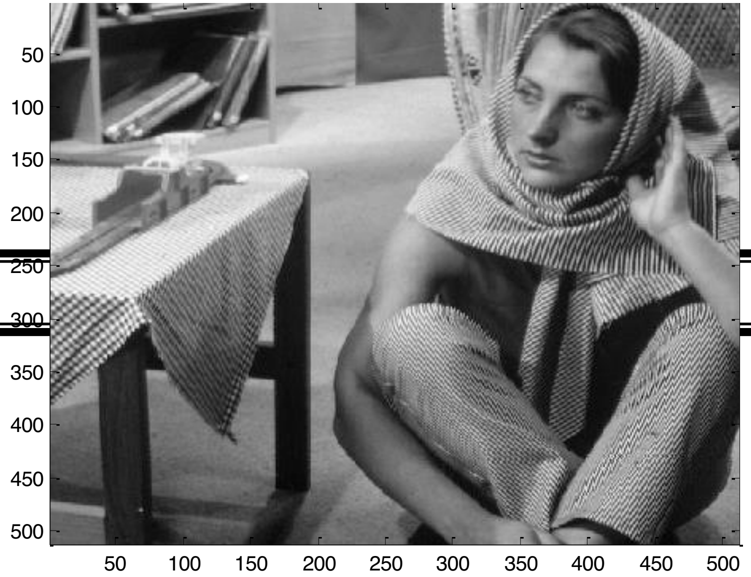


Image sous échantillonnée d'un facteur 2 :
256*256 pixels, quantifiée sur 8 bits

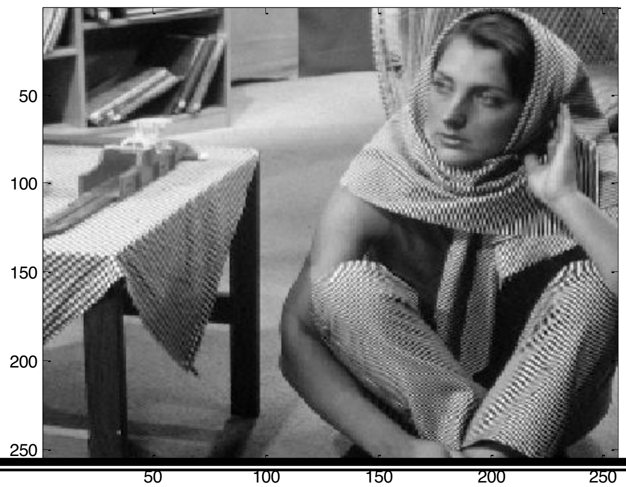
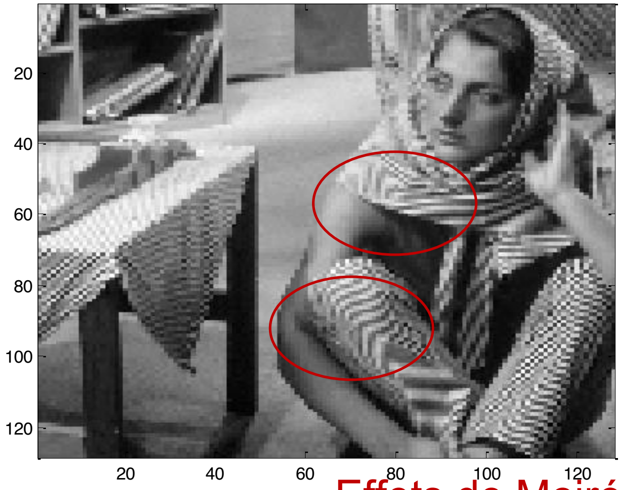


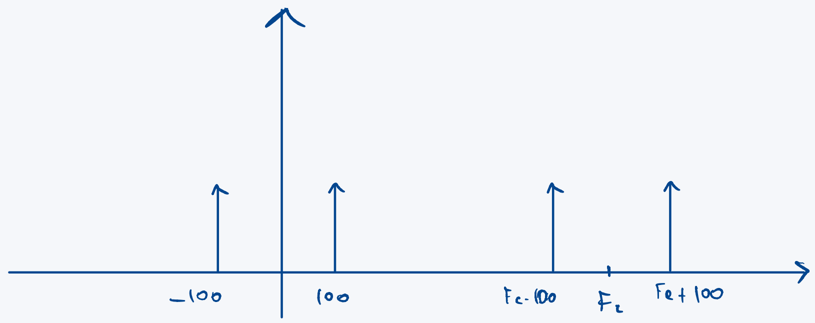
Image sous échantillonnée d'un facteur 4 :
128*128 pixels, quantifiée sur 8 bits



Effets de Moiré

QUESTION 3

Allez sur **wooclap.com** et utilisez le code **SIGSEQ1**



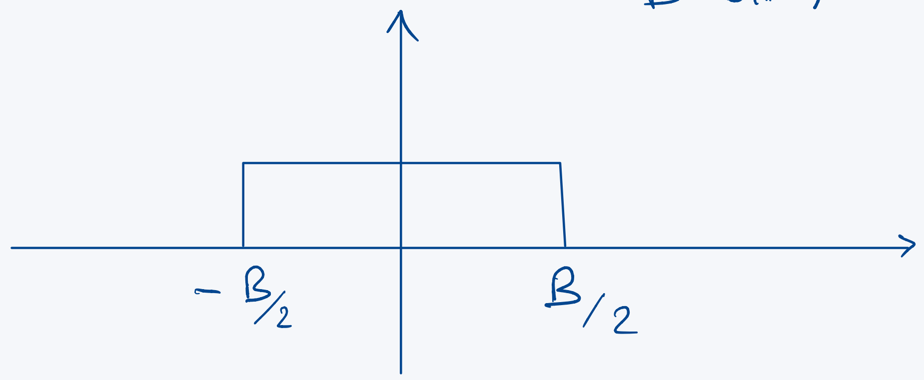
On échantillonne un cosinus de fréquence 100 Hz en utilisant une fréquence d'échantillonnage de 500 Hz. La suite d'échantillons obtenue contient-elle la même information que le signal de départ ?

1. Oui
2. Non

QUESTION 4

Allez sur **wooclap.com** et utilisez le code **SIGSEQ1**

$B\,\mathrm{sinc}(\pi B t) \xrightarrow{TF} \Pi_B(f)$

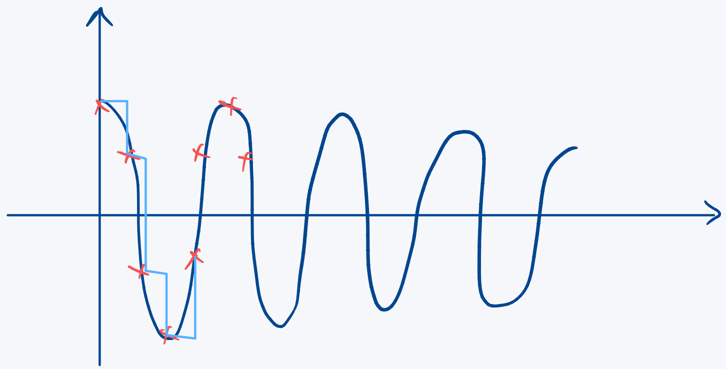


Un signal x(t)=Bsinc(πBt), B=200 Hz, est échantillonné avec une fréquence d'échantillonnage Fe= 300 Hz. La suite d'échantillons générée contient-elle la même information que le signal de départ ?

1. Oui
2. Non

QUESTION 5

Allez sur **wooclap.com** et utilisez le code **SIGSEQ1**



Soit un cosinus de fréquence 100 Hz échantillonné en utilisant une fréquence d'échantillonnage Fe=1000 Hz. Le cosinus de départ peut-il être parfaitement reconstruit, à partir de la suite d'échantillons, en bloquant la valeur de chaque échantillon pendant Te secondes (Te=1/Fe) ?

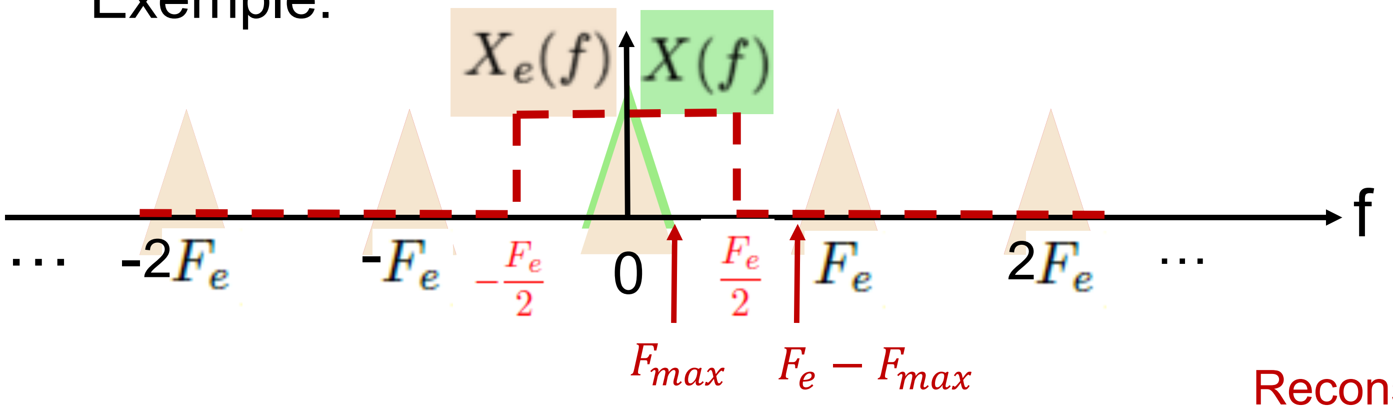
Pas parfaitement, l'interpolation idéale est celle qui utilise des sinus cardinaux.

1. Oui
2. Non

Numérisation du signal : Echantillonnage

Reconstruction du signal par filtrage à partir de la suite d'échantillons

Exemple:



Reconstruction idéale

$$\boxed{X_e(f) \times \prod_{F_e}(f) \xrightarrow{TF^{-1}} x_e(t) * F_e sinc(\pi t F_e) = F_e \sum_k x(kT_e) sinc(\pi(t - kT_e)F_e)}$$

$$X_e(f) \times H(f) \xrightarrow{TF^{-1}} x_e(t) * h(t) = F_e \sum_k x(kT_e) h(t - kT_e)$$

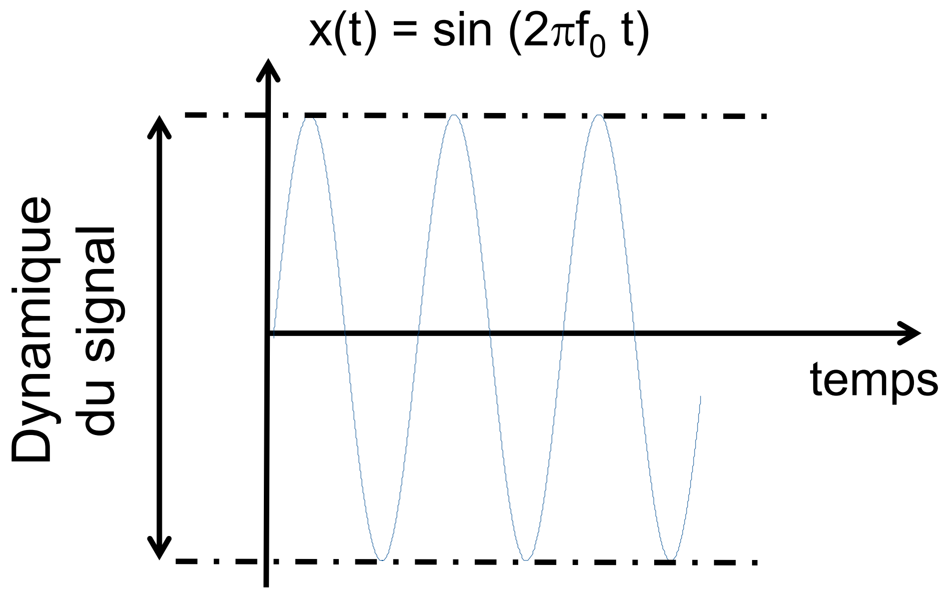
Numérisation du signal : Quantification

Signal quantifié :
signal défini à chaque instant par un nombre fini de valeurs

Numérisation du signal : Quantification

Signal quantifié :
signal défini à chaque instant par un nombre fini de valeurs

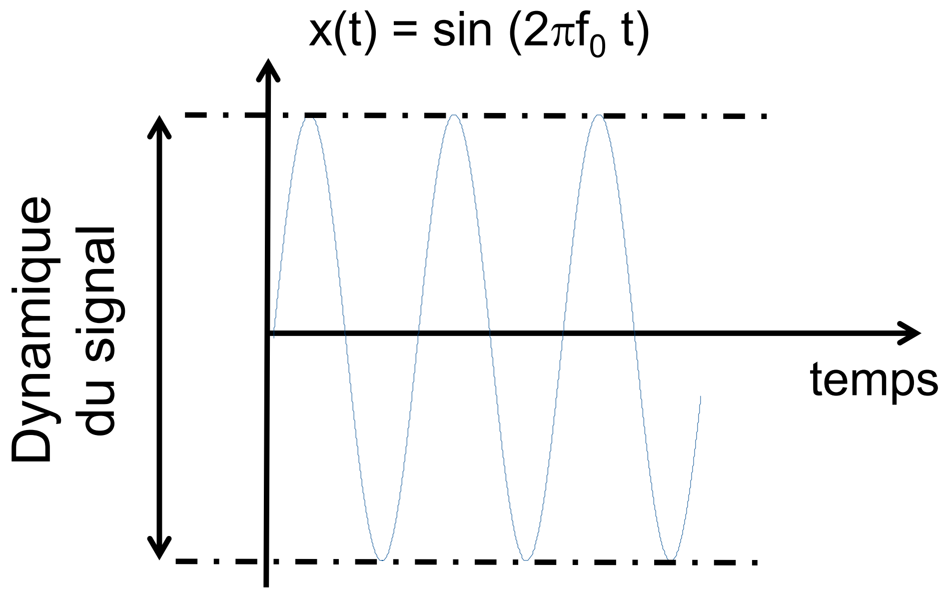
Exemple :



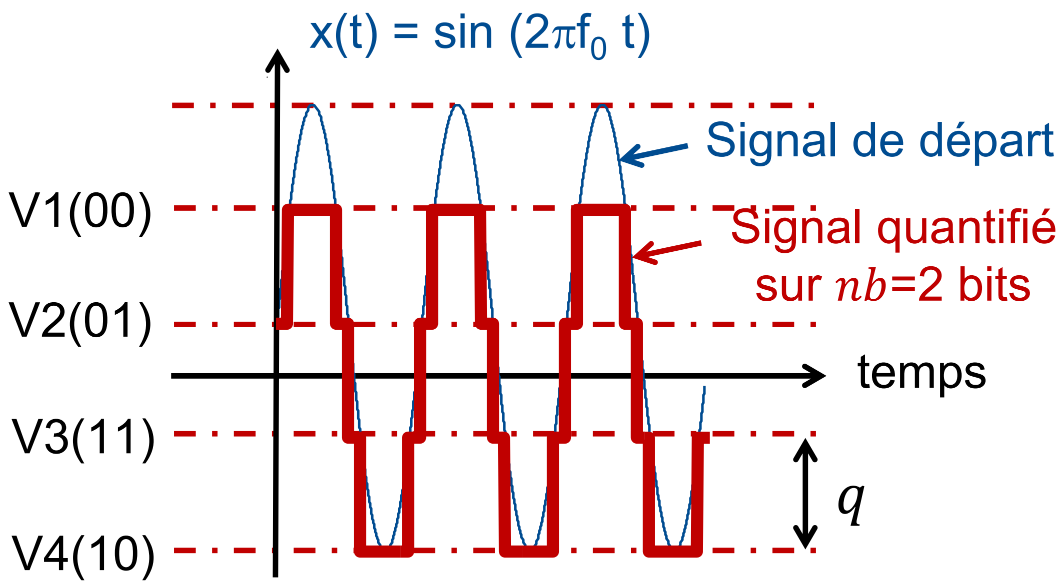
Numérisation du signal : Quantification

Signal quantifié :
signal défini à chaque instant par un nombre fini de valeurs

Exemple :



$x(t) = \sin(2\pi f_0\, t) \in \{V1, V2, V3, V4\}$



Pas de quantification : $q = \dfrac{Dynamique\ du\ signal}{2^{nb}}$

Numérisation du signal : Quantification

Peut-on conserver la même information dans le signal quantifié ?

QUESTION 6

Allez sur **wooclap.com** et utilisez le code **SIGSEQ1**

Après quantification, le signal obtenu (signal quantifié) contient-il la même information que le signal de départ ?

1. Oui

2. Non

Numérisation du signal : Quantification

Peut-on conserver la même information dans le signal quantifié ?

Non,
L'opération de quantification est une opération irréversible.

Numérisation du signal : Quantification = nb de niveaux possibles

Peut-on conserver la même information dans le signal quantifié ?

Mais le signal quantifié et le signal non quantifié peuvent cependant être très proches…

Non,
L’opération de quantification est une opération irréversible.

Une opération de quantification correctement réalisée va ajouter un bruit, dit *bruit de quantification*, au signal de départ (non quantifié).

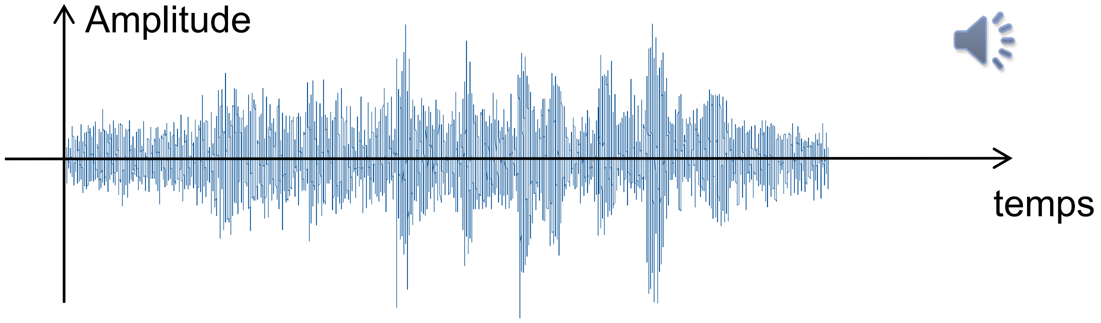
On montre que le rapport signal sur bruit (ou SNR) de quantification est donné par :

$$SNR_Q(dB) = 10\,log\,\frac{P_{signal\ non\ quantifié}}{P_{bruit\ de\ quantification}} = 6nb + constante$$

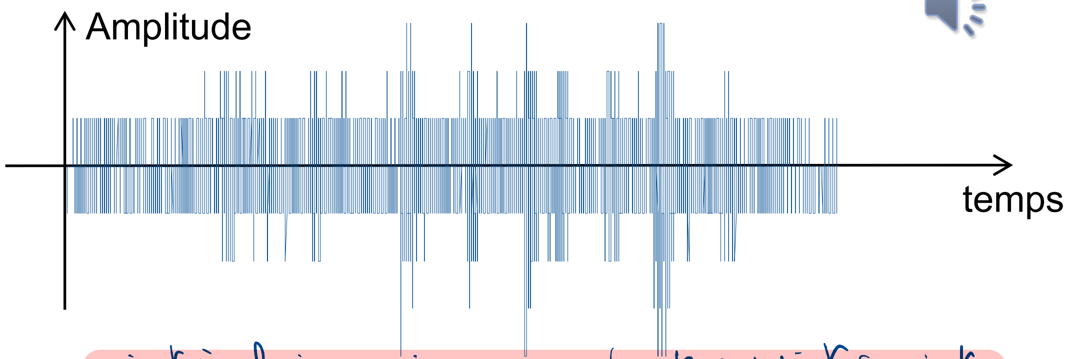
La constante dépendant du signal considéré

Exemple 1

→ Signal quantifié sur nb=16 bits => 2^{16}=65536 niveaux :



→ Signal quantifié sur nb = 4 bits => 2^4 = 16 niveaux :



Voix très claire mais avec un bruit qui vient s'ajouter

Exemple 2

Image d'origine
512*512,
Quantifiée sur 8 bits

Image quantifiée **sur 4 bits**

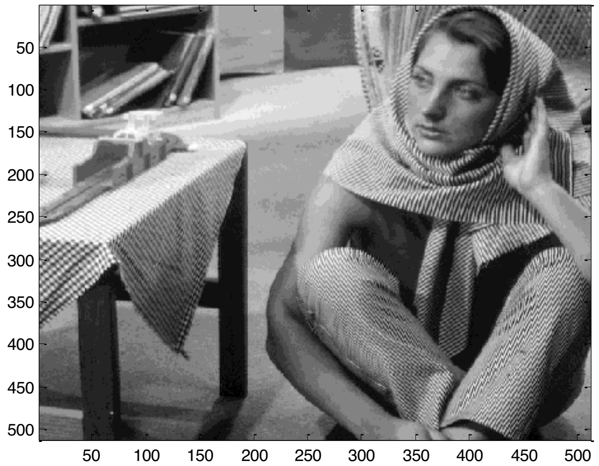
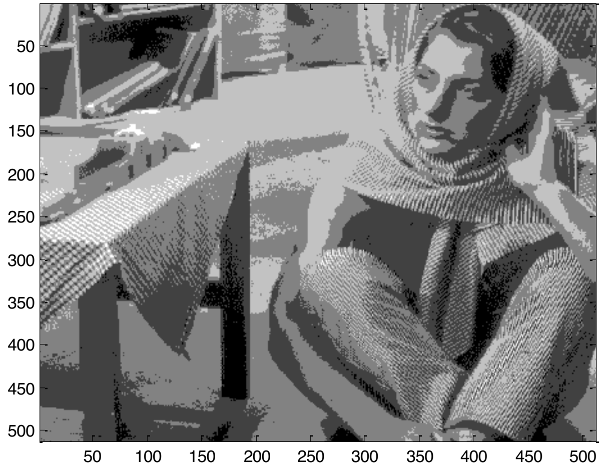


Image quantifiée **sur 2 bits**



Numérisation du signal : Echantillonnage + Quantification

Signal numérique :

signal défini à des instants discrets par un nombre fini de valeurs

= signal échantillonné et quantifié

Numérisation du signal : Echantillonnage + Quantification

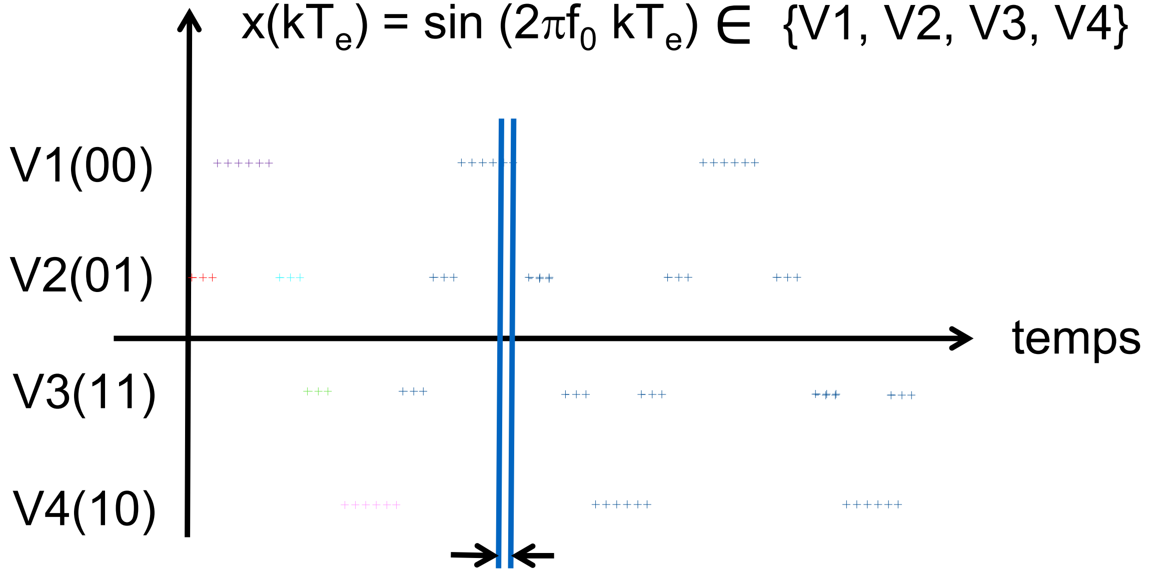
Signal numérique :

signal défini à des instants discrets par un nombre fini de valeurs

= signal échantillonné et quantifié

<u>Exemple 1 :</u>

$x(kT_e) = \sin(2\pi f_0\, kT_e) \in \{V1, V2, V3, V4\}$



T_e : période d'échantillonnage

Finalement l'information binaire associée à cette sinusoïde, ou sinusoïde numérisée, va être donnée par :

010101000000000000010101111111101010101010

Numérisation du signal : Echantillonnage + Quantification

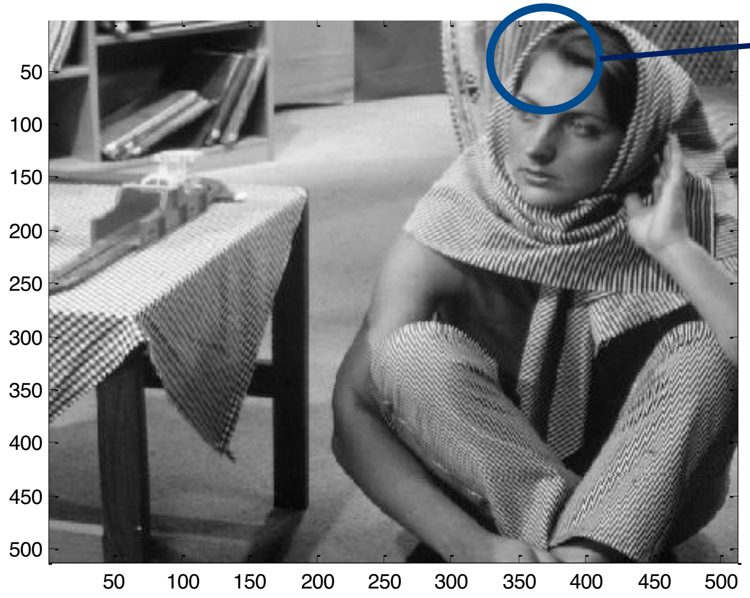
Signal numérique :

signal défini à des instants discrets par un nombre fini de valeurs
= signal échantillonné et quantifié

<u>Exemple 2 :</u>

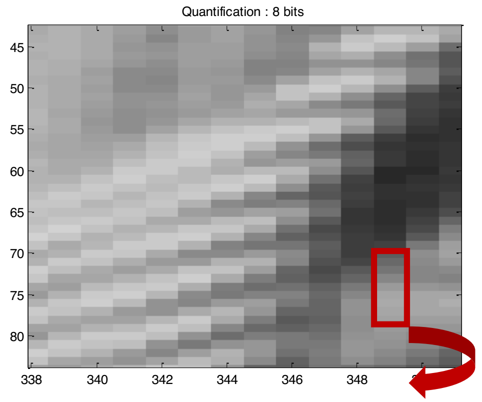
Original image:
512*512,
Quantized on 8 bits

512 pixels



512 pixels

Zoom



Un niveau de gris parmi 256 (2^8)

Génération d'un signal numérique

Exemple

$$x(t) = \cos(2\pi f_0 t)\,; \quad f_0 = 100\ Hz$$

Génération du signal :

$$t \rightarrow [0 : T_e : (N-1)\,T_e]$$

T_e ?

1- T_e = 100 Hz
2- T_e > 5 ms
3- T_e < 5 ms

QUESTION 7

Allez sur **wooclap.com** et utilisez le code **SIGSEQ1**

Je souhaite générer un cosinus de fréquence 100 Hz en numérique. Je dois utiliser comme période d'échantillonnage Te :

1. Te=200 Hz
2. Te<5 ms
3. Te>5 ms

Génération d'un signal numérique

Exemple

$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$

Génération du signal :

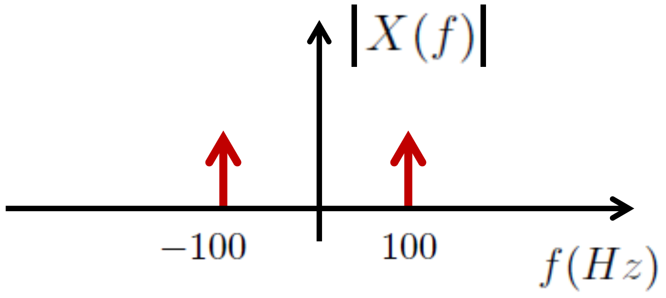
$t \rightarrow [0 : T_e : (N-1)\,T_e]$

T_e ?

1- T_e = 100 Hz
2- T_e > 5 ms
3- T_e < 5 ms

Transformée de Fourier :

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$



Génération d'un signal numérique

Exemple

$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$

Génération du signal :

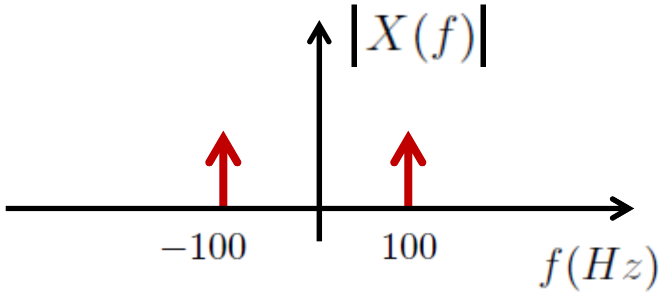
$t \rightarrow [0 : T_e : (N-1)\,T_e]$

T_e ?

1- T_e = 100 Hz
2- T_e > 5 ms
3- T_e < 5 ms

Transformée de Fourier :

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$



F_{max}=100 Hz

Génération d'un signal numérique

Exemple

$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$

Génération du signal :

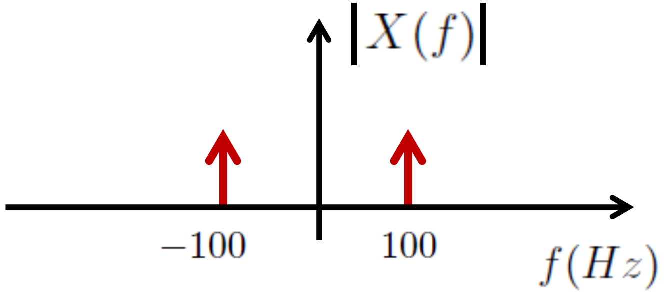
$t \rightarrow [0 : T_e : (N-1)\,T_e]$

T_e ?

1- T_e = 100 Hz
2- T_e > 5 ms
3- T_e < 5 ms

Transformée de Fourier :

$$X(f) = \frac{1}{2}\left(\delta\left(f + f_0\right) + \delta\left(f - f_0\right)\right)$$



F_{max}=100 Hz

$$F_e = \frac{1}{T_e} > 2F_{max} \Rightarrow T_e < 5\ ms$$

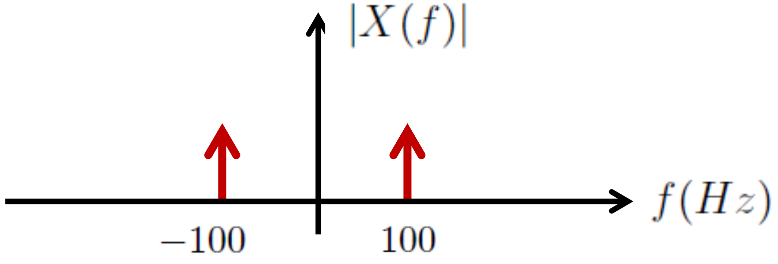
Génération d'un signal numérique

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

TF du signal :

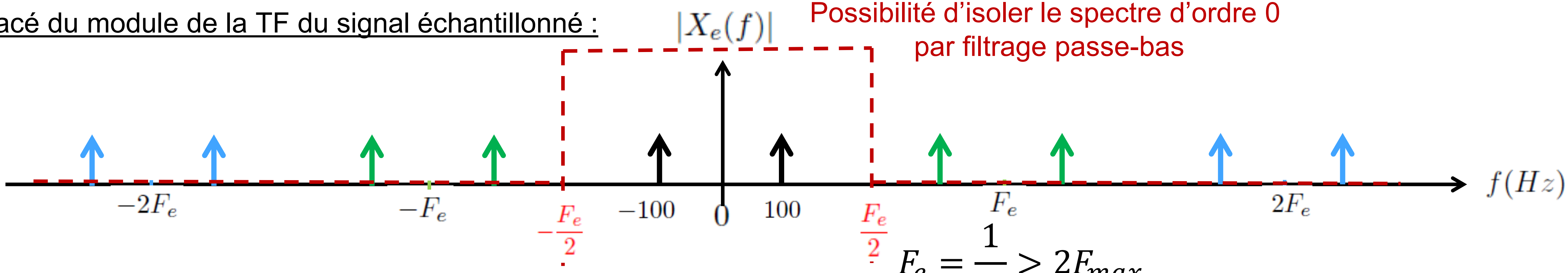
$$X(f) = \frac{1}{2}\left(\delta(f + f_0) + \delta(f - f_0)\right)$$

Tracé du module de la TF du signal :



$\Rightarrow F_{max}$=100 Hz

Tracé du module de la TF du signal échantillonné :



$$F_e = \frac{1}{T_e} > 2F_{max}$$

Génération d'un signal numérique

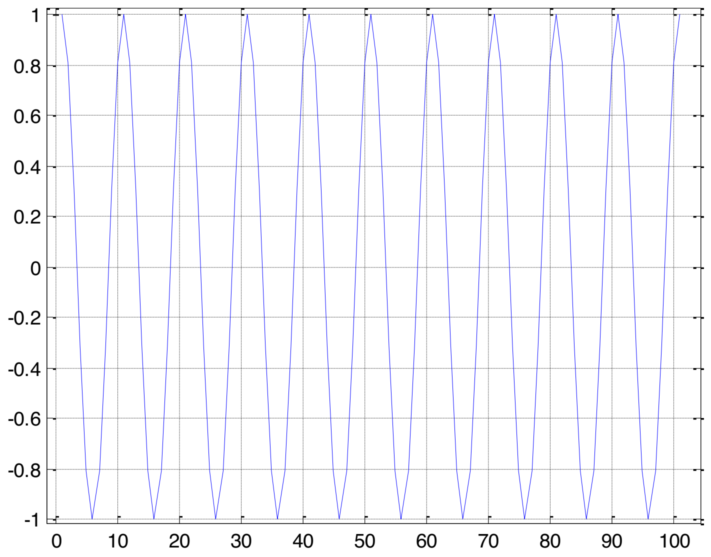
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab
(Suréchantillonnage d'un facteur 5)

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=101;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)
```

Signal obtenu :



Génération d'un signal numérique

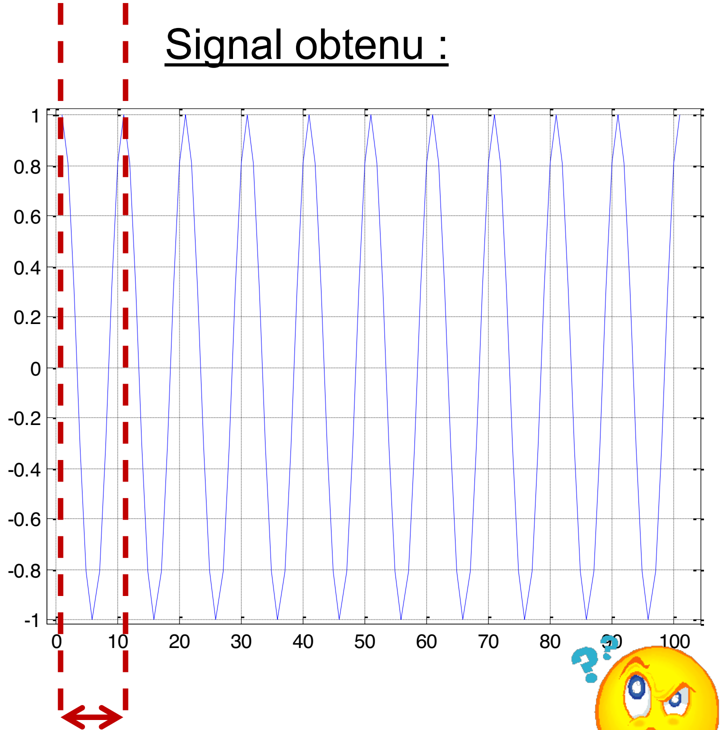
Exemple

$$x(t) = \cos\left(2\pi f_0 t\right)\,; \;\; f_0 = 100\ Hz$$

Code Matlab
(Suréchantillonnage d'un facteur 5)

```
%Paramètres
f0=100;       %fréquence du cosinus
Fe=1000;      %fréquence d'échantillonnage
Te=1/Fe;      %période d'échantillonnage
N=101;        %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)
```

Signal obtenu :



Une période = 10 secondes ??

Génération d’un signal numérique

Exemple

$$x(t) = \cos(2\pi f_0 t)\ ;\ \ f_0 = 100\ Hz$$

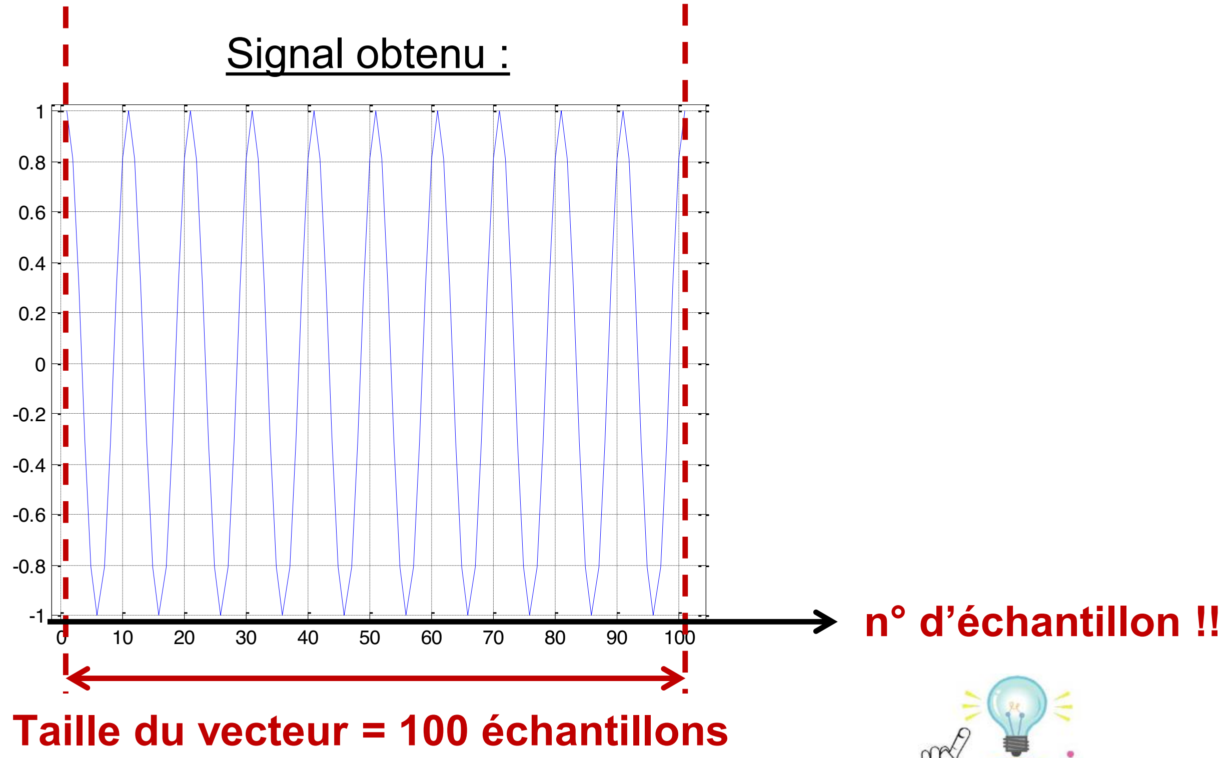
1 période = 10 échantillons de signal

Code Matlab

(Suréchantillonnage d’un facteur 5)

```
%Paramètres
f0=100;        %fréquence du cosinus
Fe=1000;       %fréquence d'échantillonnage
Te=1/Fe;       %période d'échantillonnage
N=100;         %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)
```

Signal obtenu :



Génération d’un signal numérique

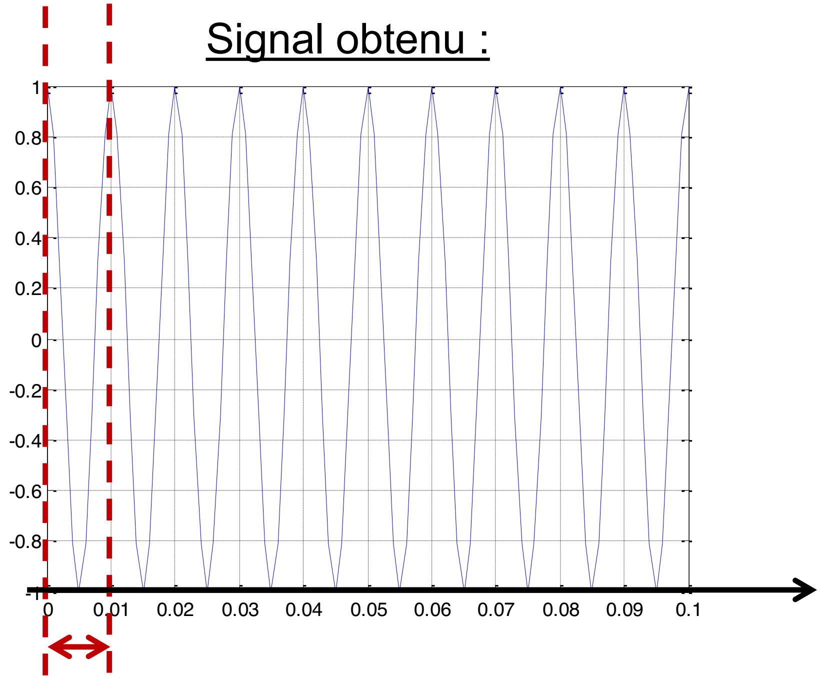
Exemple

$$x(t) = \cos\left(2\pi f_0 t\right); \;\; f_0 = 100\ Hz$$

Code Matlab
(Suréchantillonnage d’un facteur 5)

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot([0:Te:(N-1)*Te],x)
```

!! Echelle temporelle à donner !!



Temps (secondes)

Une période = 0,01 secondes

Traitement Numérique du Signal

Nathalie Thomas

IRIT/ENSEEIHT
Nathalie.Thomas@enseeiht.fr

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d'inter et d'auto corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

0. Slide de connexion

1 Allez sur wooclap.com

2 Entrez le code d'événement dans le bandeau supérieur

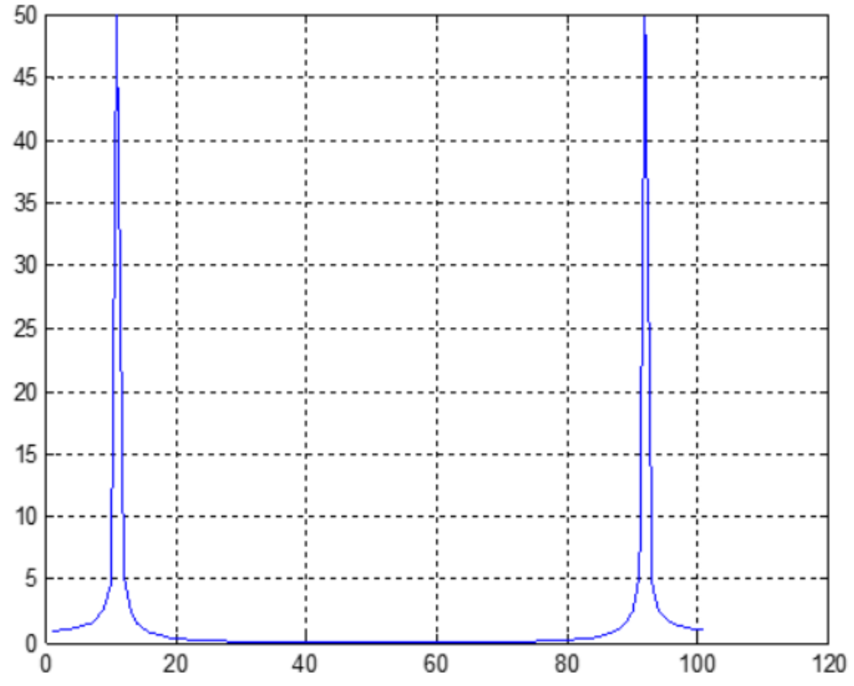
Code d'événement
SIGSEQ2

https://app.wooclap.com/SIGSEQ2

Copier le lien de participation

QUESTION 1

Allez sur **wooclap.com** et utilisez le code **SIGSEQ2**



Ce tracé représente le module de la transformée de Fourier estimée en numérique d'un signal correctement échantillonné (condition de Shannon respectée). Ce signal est-il :

1. Un cosinus de fréquence 10 Hz ?
2. Une somme de deux cosinus de fréquences 10 et 90 Hz ?
3. Pas assez d'éléments pour répondre à la question

Transformée de Fourier Discrète

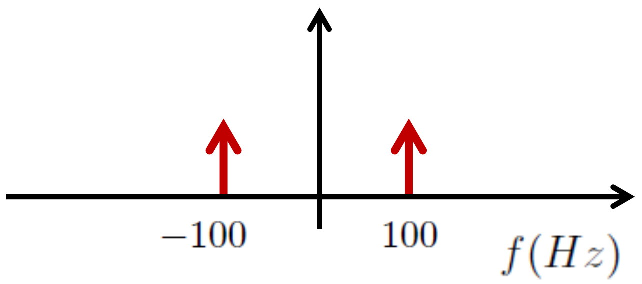
Exemple

$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$

<u>TF du signal :</u>

$$X(f) = \frac{1}{2}\left(\delta\left(f + f_0\right) + \delta\left(f - f_0\right)\right)$$

<u>Tracé du module de la TF du signal :</u>



Transformée de Fourier Discrète

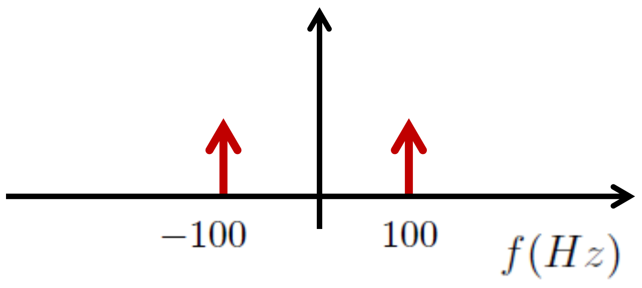
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

TF du signal :

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$

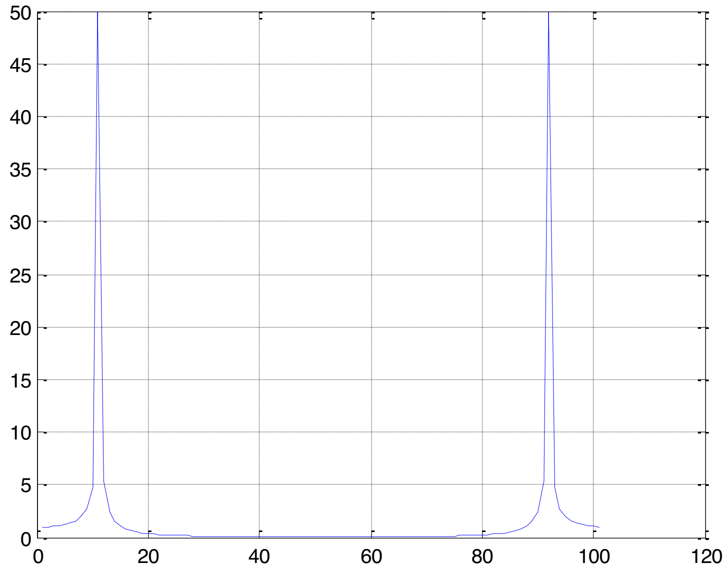
Tracé du module de la TF du signal :



Code Matlab :

```
%Paramètres
f0=100;        %fréquence du cosinus
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;      %période d'échantillonnage
N=100;         %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Transformée de Fourier Discrète

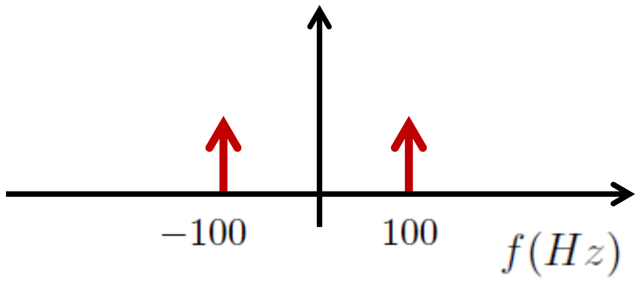
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

TF du signal :

$$X(f) = \frac{1}{2}\left(\delta\left(f + f_0\right) + \delta\left(f - f_0\right)\right)$$

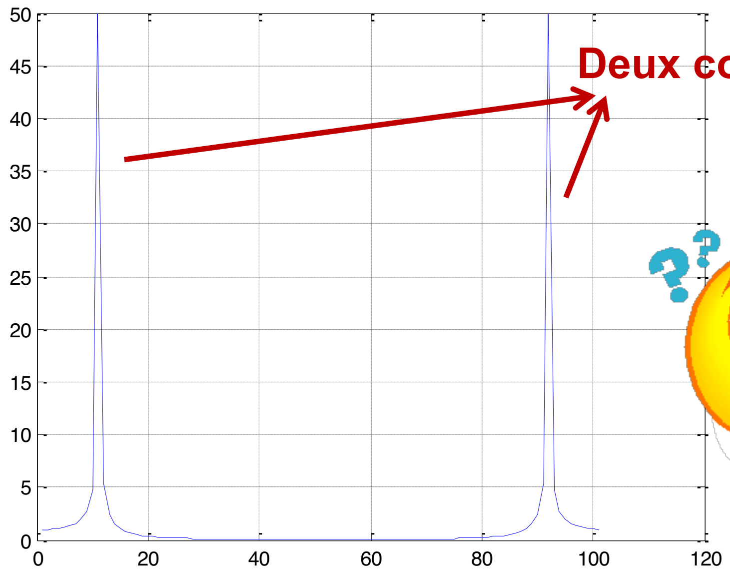
Tracé du module de la TF du signal :



Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



QUESTION 2

Allez sur **wooclap.com** et utilisez le code **SIGSEQ2**

L'échantillonnage d'un signal avec une période de Te :

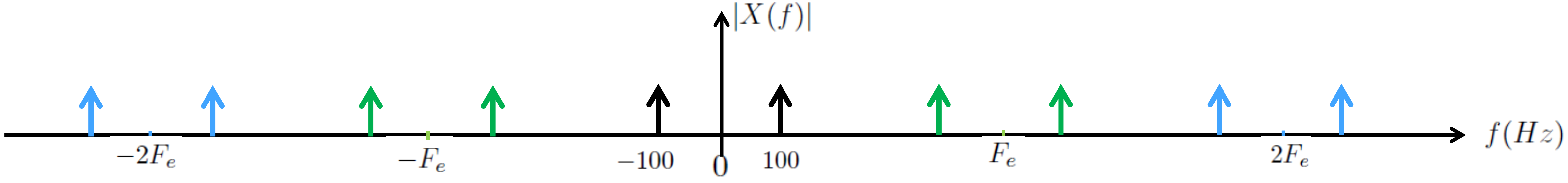
1. **Périodise sa transformée de Fourier tous les Fe=1/Te**
2. **Provoque l'échantillonnage de sa transformée de Fourier avec un pas de Fe=1/Te**
3. **Périodise sa transformée de Fourier tous les Te**

Transformée de Fourier Discrète

Echantillonnage temporel

Exemple

$$x(t) = \cos\left(2\pi f_0 t\right);\ \ f_0 = 100\ Hz$$



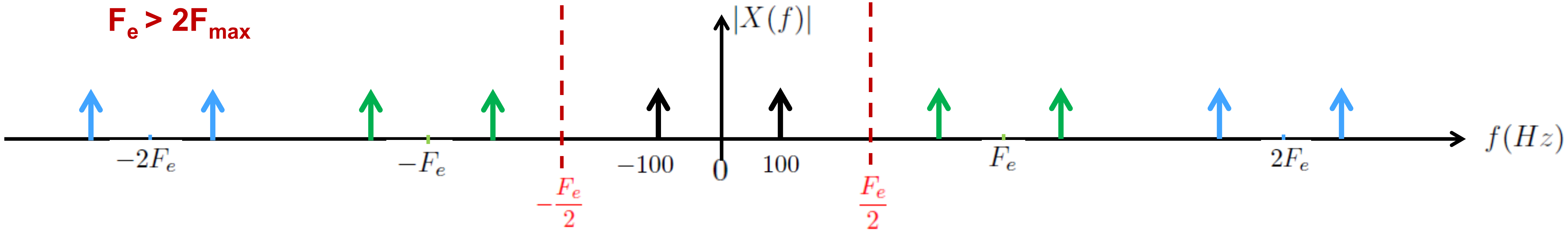
Transformée de Fourier Discrète

Echantillonnage temporel

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

$F_e > 2F_{max}$

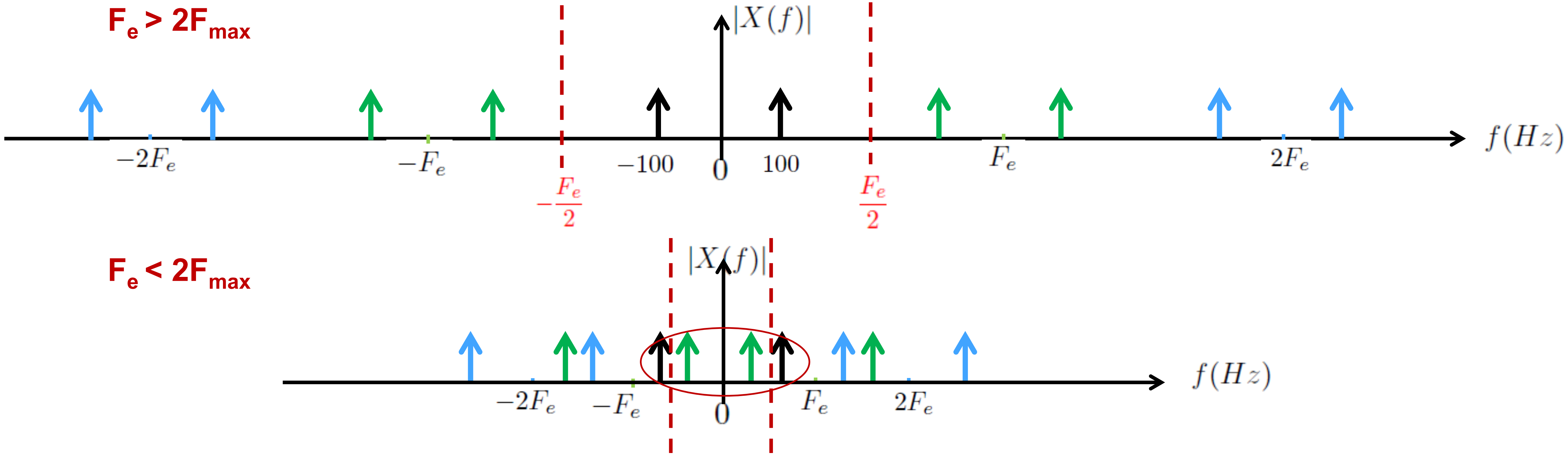


Transformée de Fourier Discrète

Echantillonnage temporel

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$



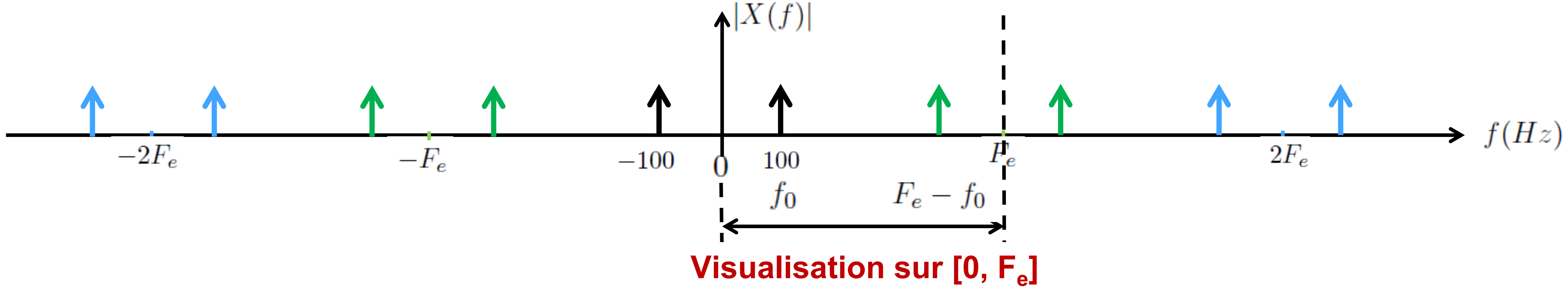
Recouvrement spectral (Aliasing)

Transformée de Fourier Discrète

Echantillonnage temporel

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

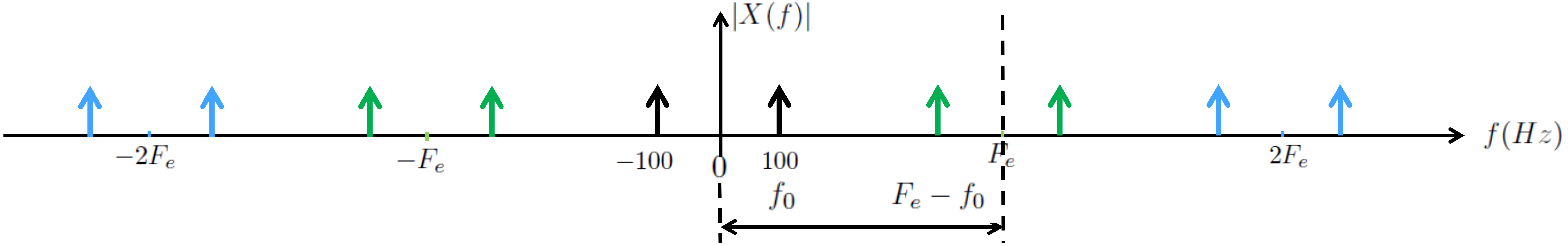


Transformée de Fourier Discrète

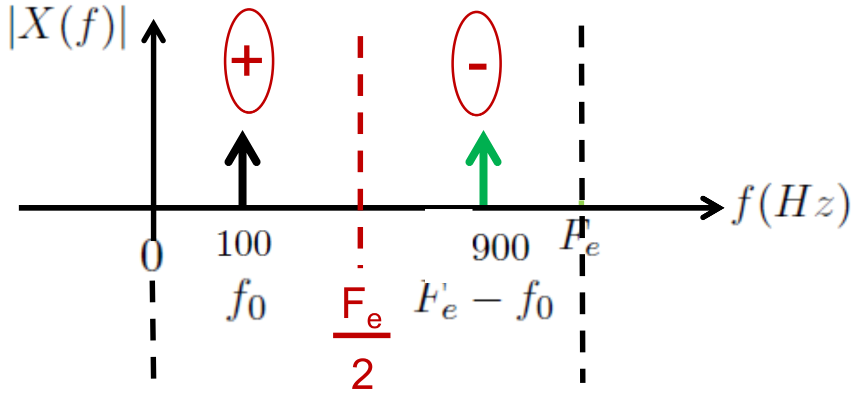
Echantillonnage temporel

Exemple

$$x(t) = \cos(2\pi f_0 t)\ ;\ f_0 = 100\ Hz$$



Visualisation sur [0, F_e] :



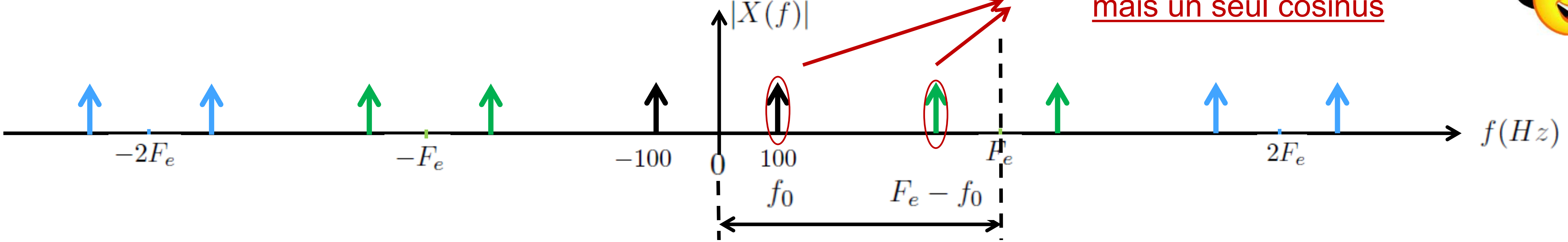
Transformée de Fourier Discrète

Echantillonnage temporel

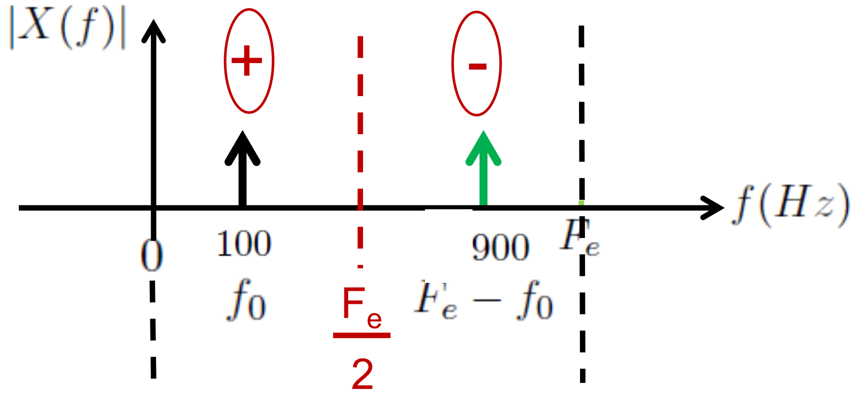
Exemple

$$x(t) = \cos(2\pi f_0 t)\ ;\ f_0 = 100\ Hz$$

Deux diracs côté fréquences positives
<u>mais un seul cosinus</u>



Visualisation sur [0, F_e] :



Transformée de Fourier Discrète

Echantillonnage temporel

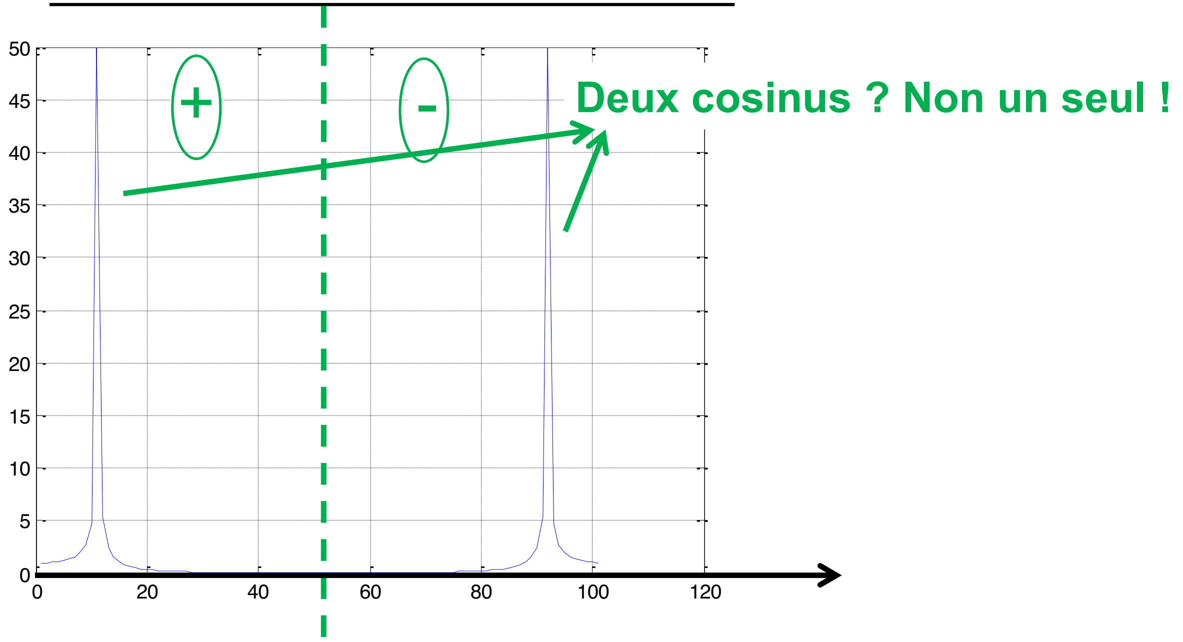
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;        %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Transformée de Fourier Discrète

!! Echelle fréquentielle !!

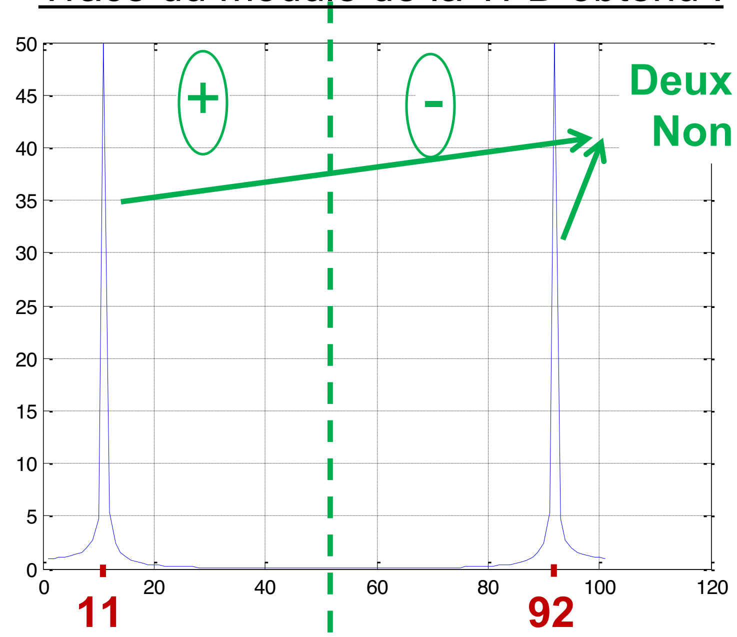
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Deux cosinus ?
Non un seul !

De fréquences f_0=11 Hz et F_e- f_0=92 Hz ??

Transformée de Fourier Discrète

!! Echelle fréquentielle !!

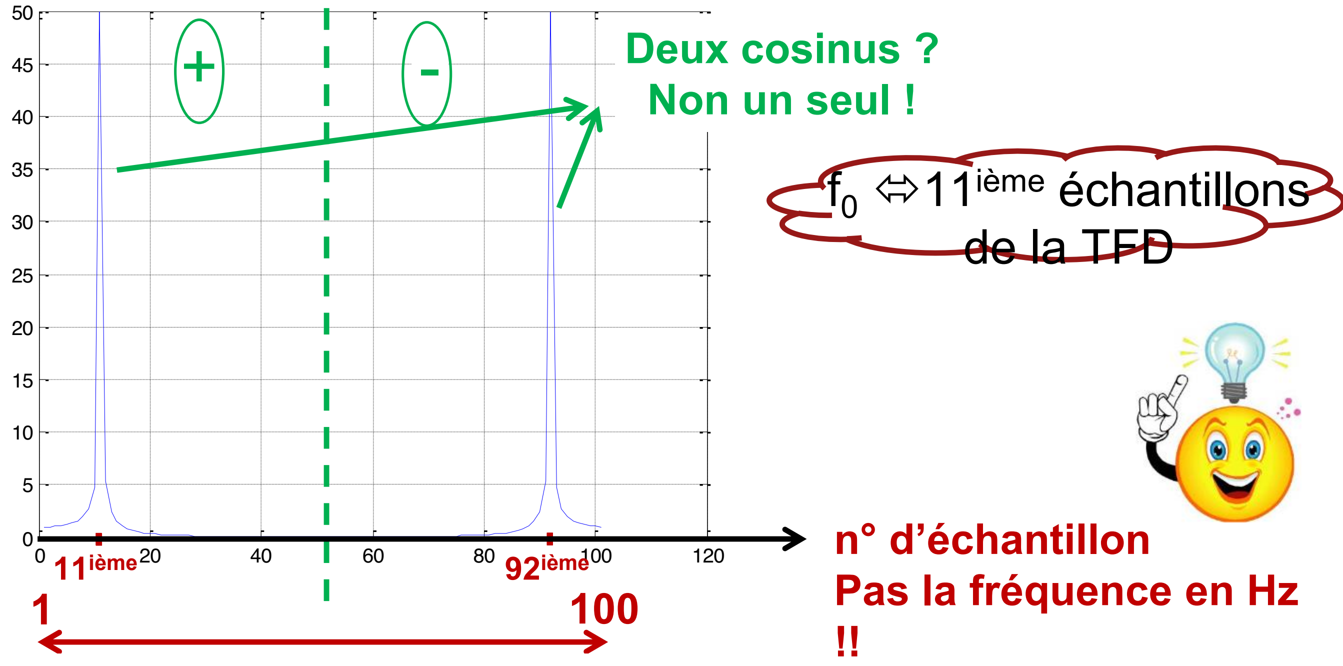
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Taille du vecteur = 100 échantillons

Transformée de Fourier Discrète

!! Echelle fréquentielle !!

Exemple

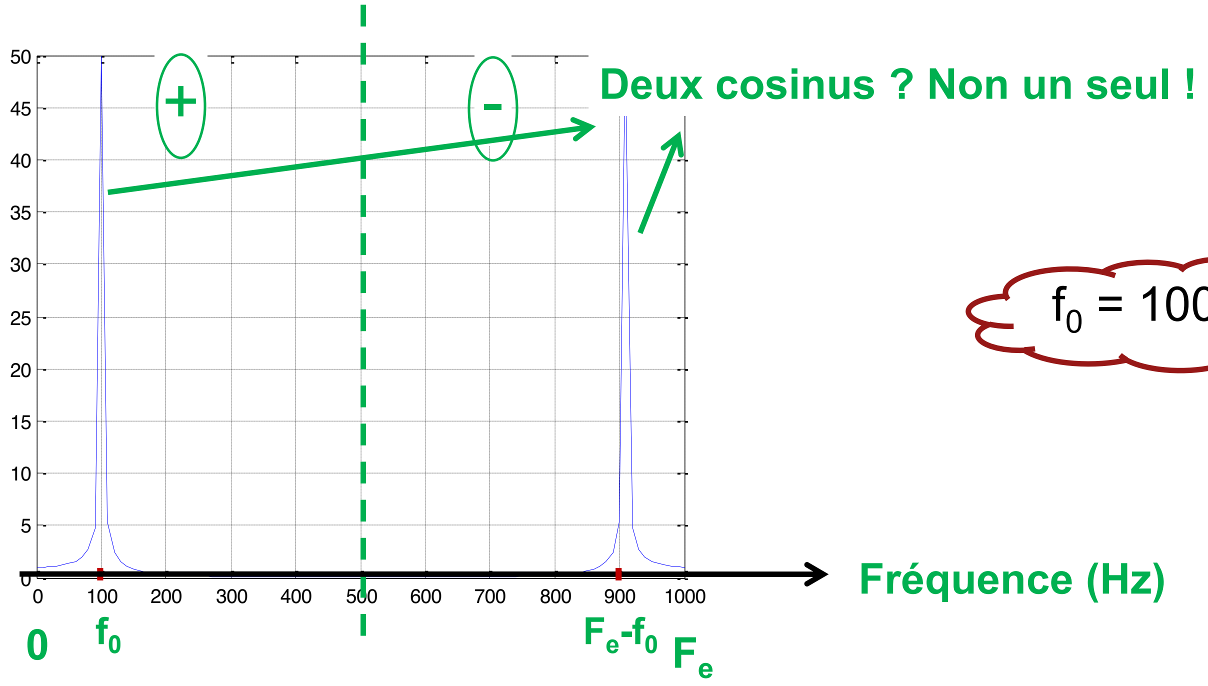
$$x(t) = \cos\left(2\pi f_0 t\right)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;       %fréquence du cosinus
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;      %période d'échantillonnage
N=100;         %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;
plot(linspace(0,Fe,length(X)),abs(X))
```

!! Echelle fréquentielle à donner !!

Tracé du module de la TFD obtenu :



f_0 = 100 Hz

Transformée de Fourier Discrète

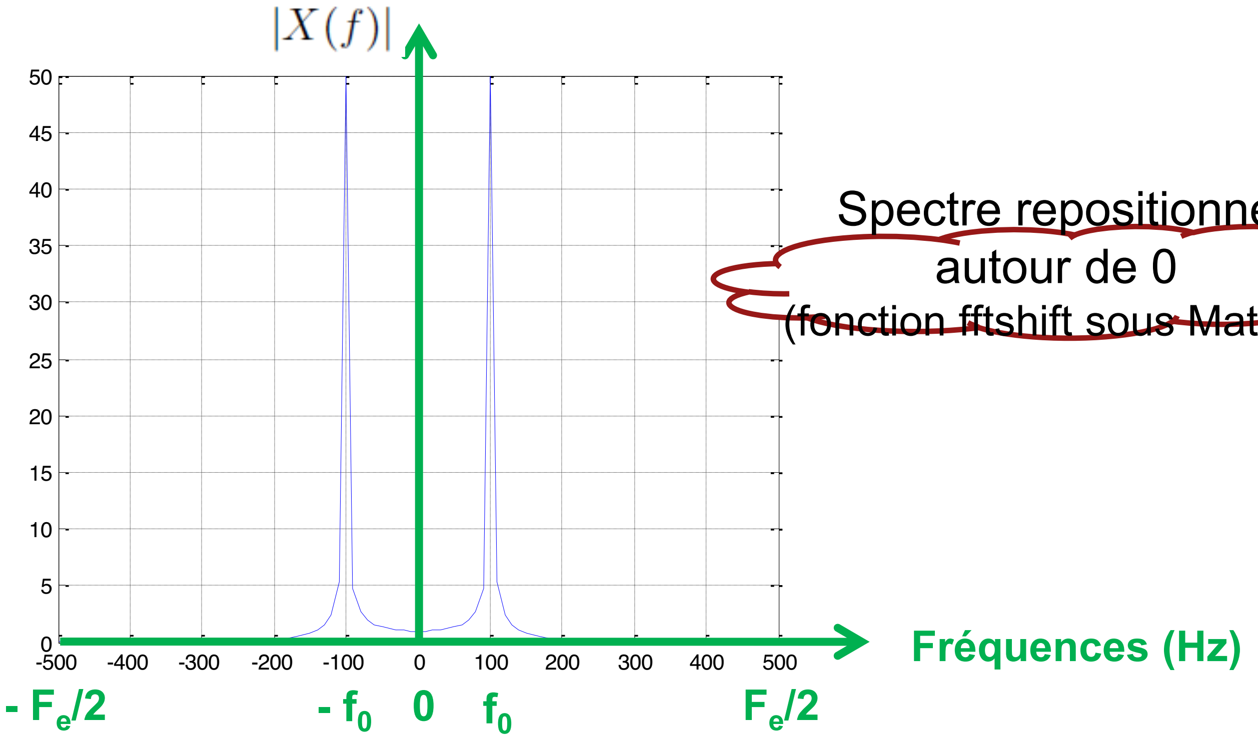
!! Echelle fréquentielle !!

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;     %fréquence du cosinus
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;      %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure; plot(linspace(-Fe/2,Fe/2,length(X)),fftshift(abs(X)))
```



Spectre repositionné autour de 0
(fonction fftshift sous Matlab)

Transformée de Fourier Discrète

Transformée de Fourier (TF) $\longrightarrow$ **Transformée de Fourier Discrète (TFD)**

<u>Echantillonnage temporel</u>

$$x(t) \;\rightarrow\; \{x(kT_e)\}_{k=-\infty,\dots,+\infty}$$

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{j2\pi ft}\,dt \;\rightarrow\; \sum_{k=-\infty}^{+\infty} x(kT_e)e^{j2\pi fkT_e}$$ (définie à T_e près)

Impact : Périodisation de la TFD

$\Rightarrow$!! Respecter la condition de Shannon !!

$\Rightarrow$!! Lecture des tracés et échelle !!

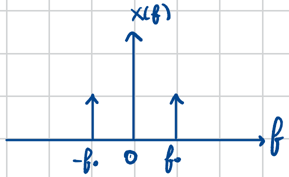
$$X(f + F_e) = X(f)$$

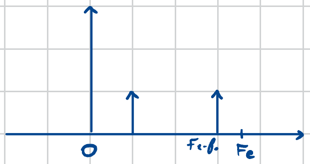
$$X(\tilde{f} + 1) = X(\tilde{f})$$

Définition de la fréquence normalisée : $\boxed{\tilde{f} = \frac{f}{F_e}}$

2ème approximation

$$x(t) = \cos(2\pi f_0 t) \longrightarrow \{x(kT_e)\}_{k\in\mathbb{Z}} \longrightarrow \{x(kT_e)\}_{k=0,\dots,N-1}$$





$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi f t}\, dt \longrightarrow X_e(f) = \sum_{k=-\infty}^{+\infty} T_e\, x(kT_e)\, e^{-j2\pi f kT_e} \xrightarrow{2} \sum_{k=0}^{N-1} x(kT_e)\, e^{-j2\pi f kT_e} = \sum_{-\infty}^{+\infty} x(kT_e) \times w(kT_e)\, e^{-j2\pi f kT_e}$$

avec $w(kT_e) = \begin{cases} 1 & \text{pour } k = 0 \text{ à } N-1 \\ 0 & \text{ailleurs} \end{cases}$

Méthode des rectangles

→ 1ère approximation

Si on prend :

$$\sum_{k=-\infty}^{+\infty} x(kT_e)\, w(kT_e)\, e^{-2j\pi f kT_e} \quad \text{au lieu de} \quad X_e(f) = \sum_{k=-\infty}^{+\infty} x(kT_e)\, e^{-2j\pi f kT_e}$$

Voir slide suivante

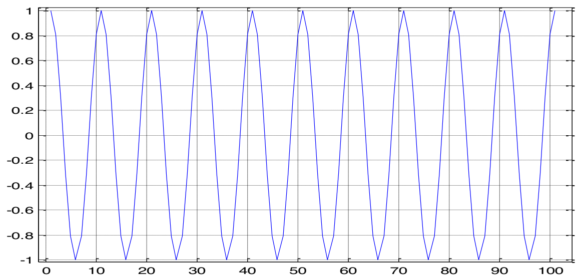
La Tf sera : $\underbrace{\underbrace{X_e(f)}_{\text{périodicité}} * W_e(f)}_{\text{distorsion}}$

Transformée de Fourier Discrète

Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$



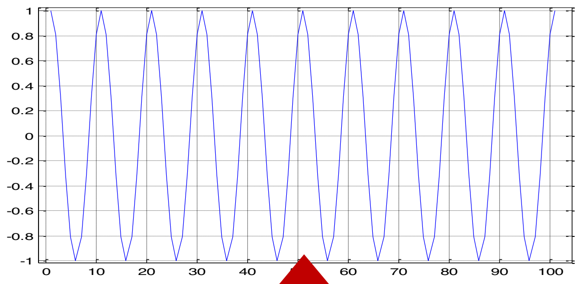
Signal de durée limitée

Transformée de Fourier Discrète

Signal de durée limitée

Exemple

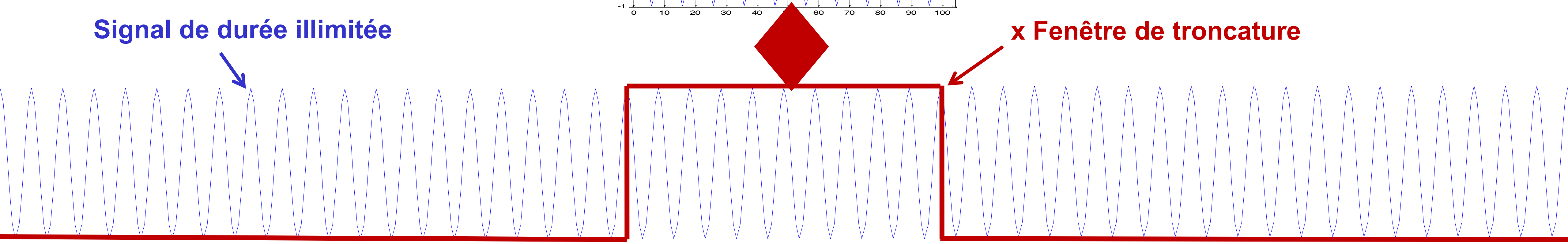
$$x(t) = \cos\left(2\pi f_0 t\right) \,;\;\; f_0 = 100\ Hz$$

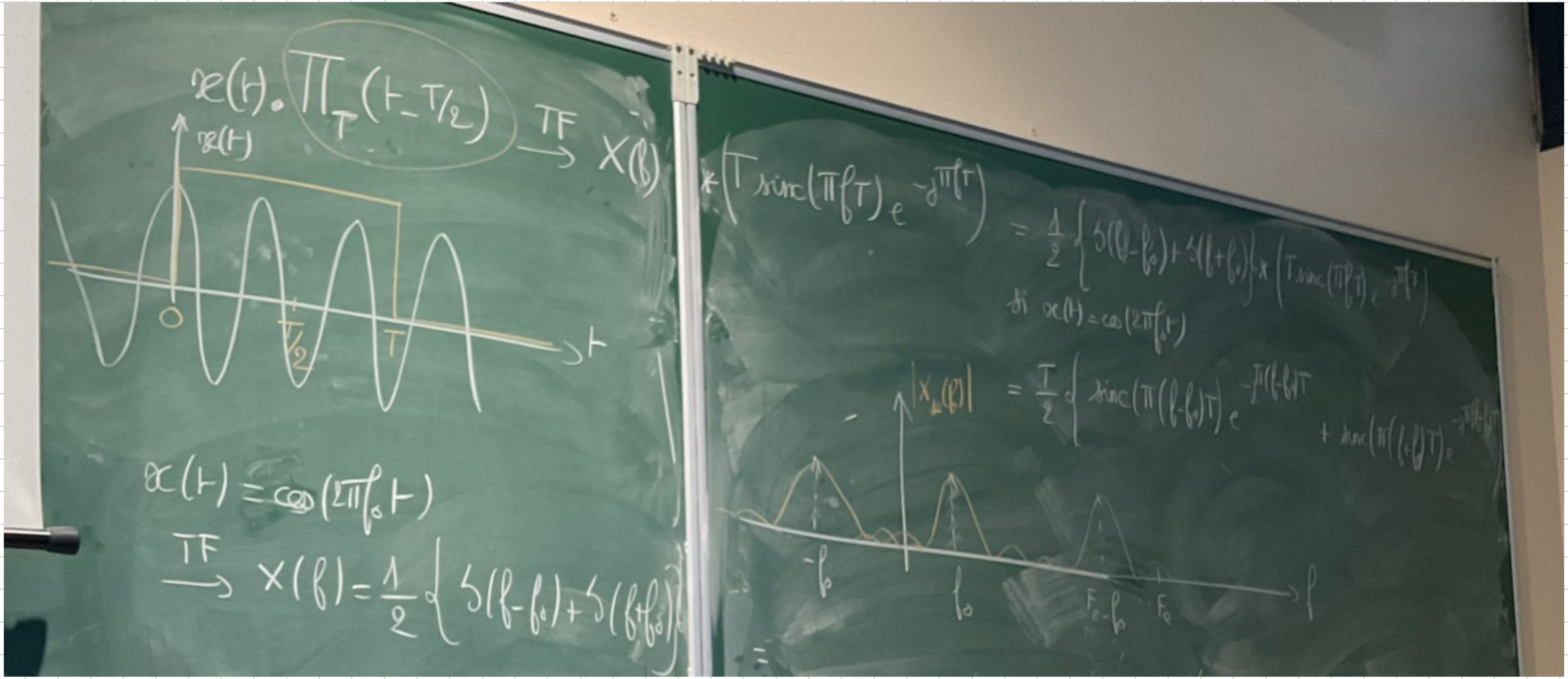


Signal de durée limitée

Signal de durée illimitée

x Fenêtre de troncature





x(t) . Π_T(t - T/2) TF → X(f)
x(t)
0
T/2
T
t
x(t) = cos(2πf₀t)
TF → X(f) = 1/2 {δ(f-f₀)+δ(f+f₀)}
* (T sinc(πfT) e^{-jπfT})
= 1/2 {δ(f-f₀)+δ(f+f₀)} * (T sinc(πfT) e^{-jπfT})
si x(t) = cos(2πf₀t)
|X_L(f)|
= T/2 {sinc(π(f-f₀)T) e^{-jπ(f-f₀)T}
+ sinc(π(f+f₀)T) e
-f₀
f₀
Fe-f₀
Fe
f

Transformée de Fourier Discrète

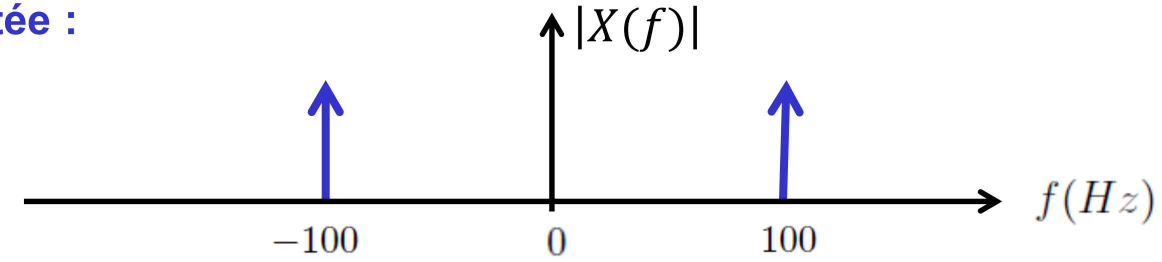
Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

TF du signal de durée illimitée :

$$x(t) \xrightarrow{TF} X(f)$$



Transformée de Fourier Discrète

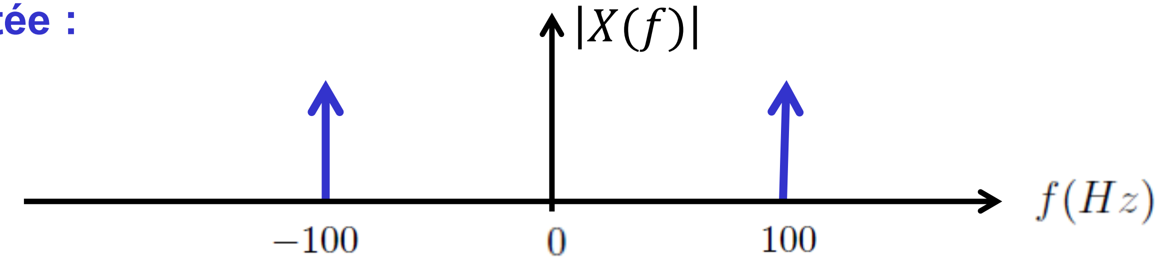
Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

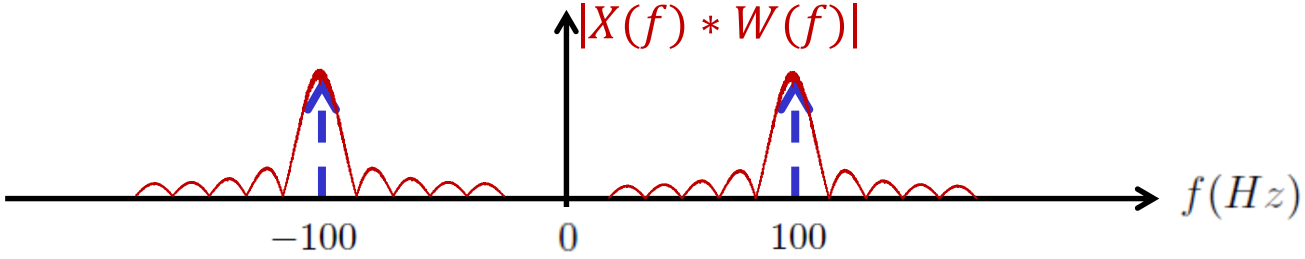
TF du signal de durée illimitée :

$$x(t) \xrightarrow{TF} X(f)$$



TF du signal à durée limitée = **TF du signal à durée illimitée** $*$ **TF de la fenêtre modélisant la troncature :**

$$x(t)w(t) \xrightarrow{TF} X(f) * W(f)$$



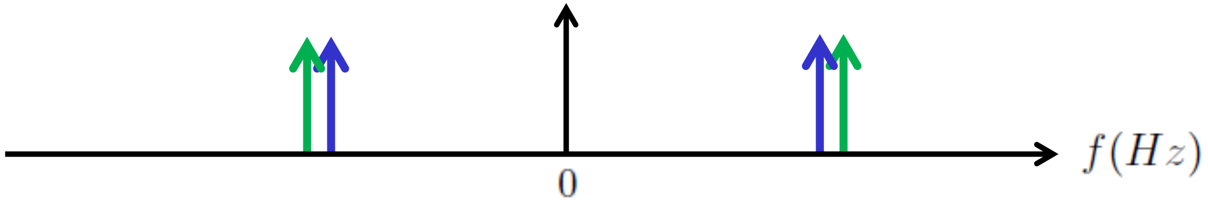
Transformée de Fourier Discrète

Signal de durée limitée

Problèmes posés :

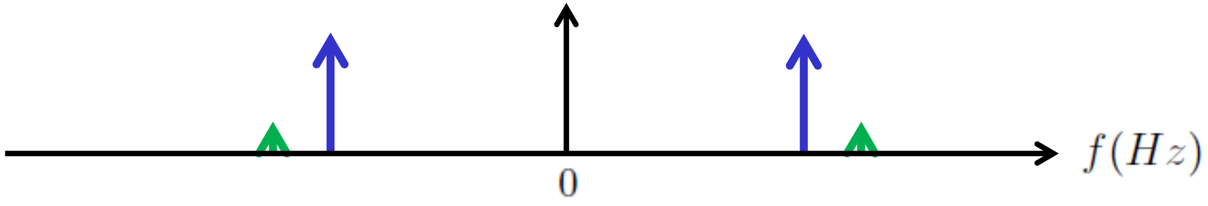
Exemple 1

Somme de deux cosinus proches en fréquences



Exemple 2

Somme de deux cosinus dont un de faible puissance



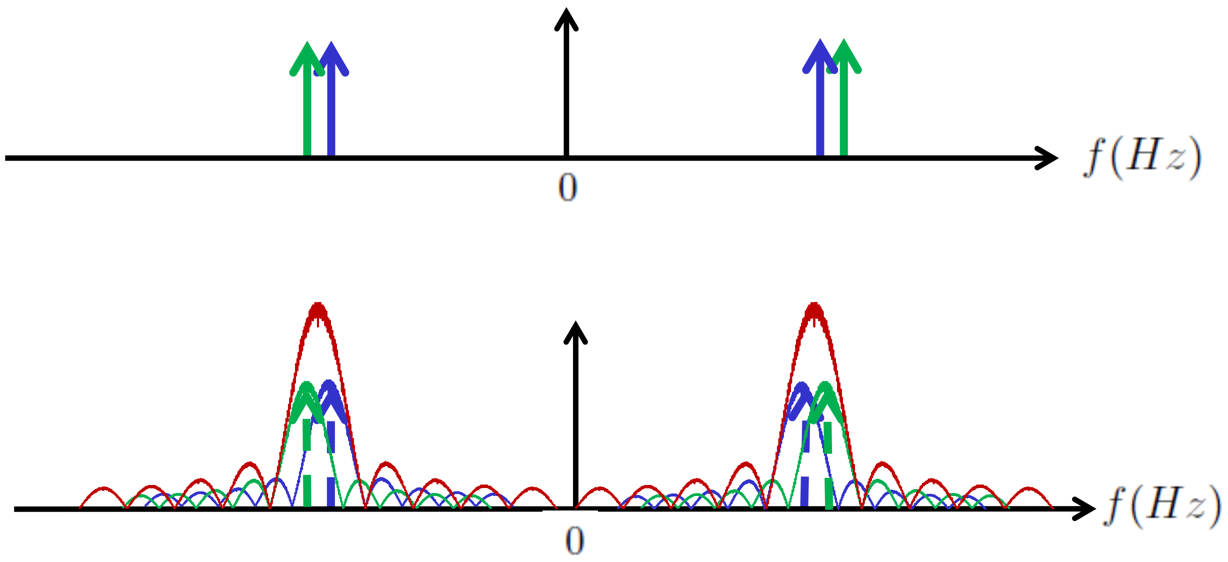
Transformée de Fourier Discrète

Signal de durée limitée

<u>Problèmes posés :</u>

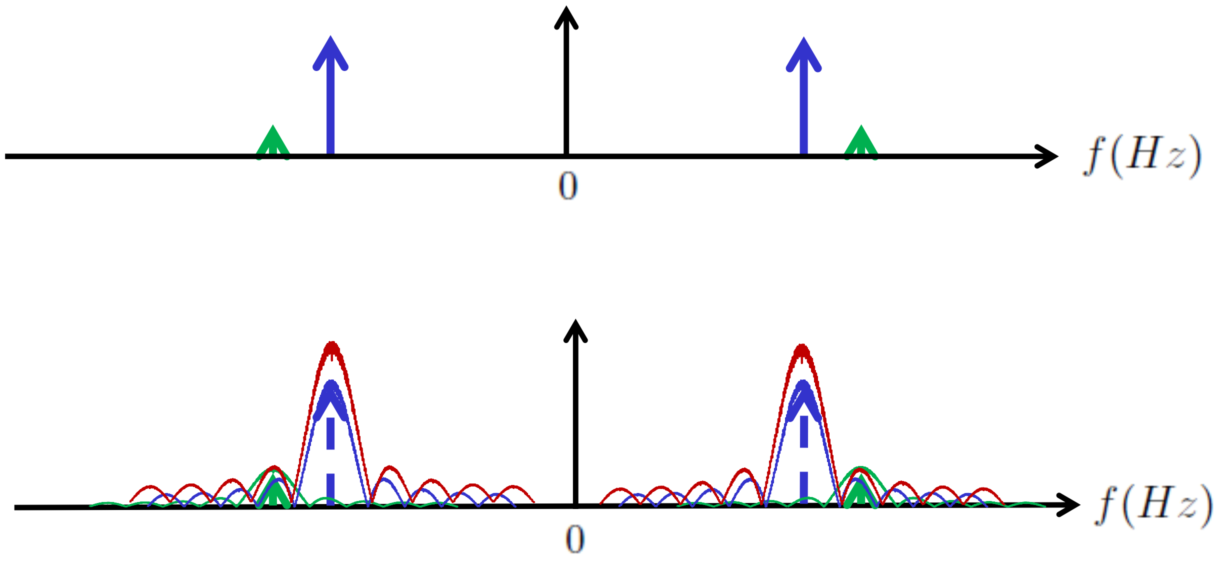
Exemple 1

Somme de deux cosinus proches en fréquences



Exemple 2

Somme de deux cosinus dont un de faible puissance



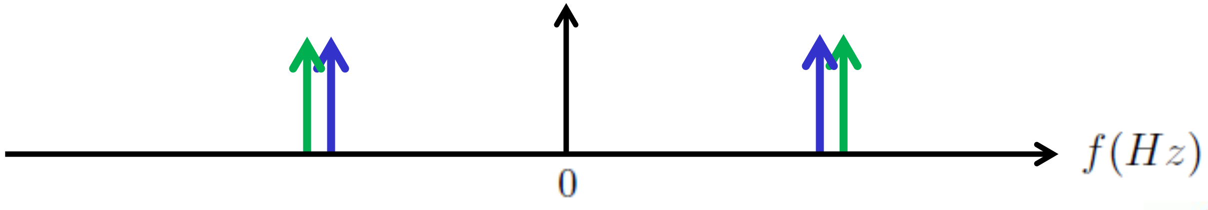
Transformée de Fourier Discrète

Signal de durée limitée

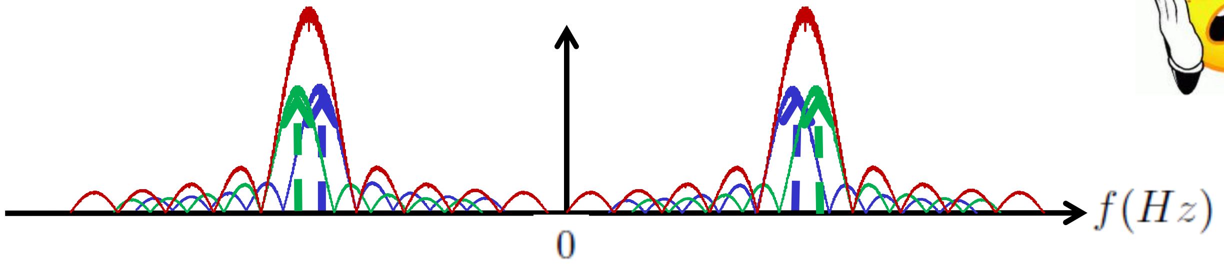
Problèmes posés :

Exemple 1

Somme de deux cosinus proches en fréquences

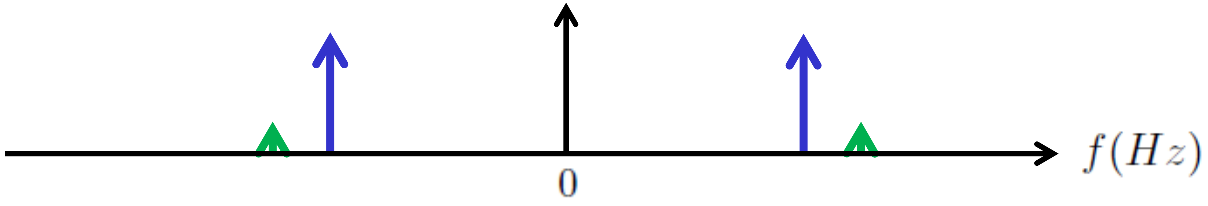


!! Un seul cosinus !!

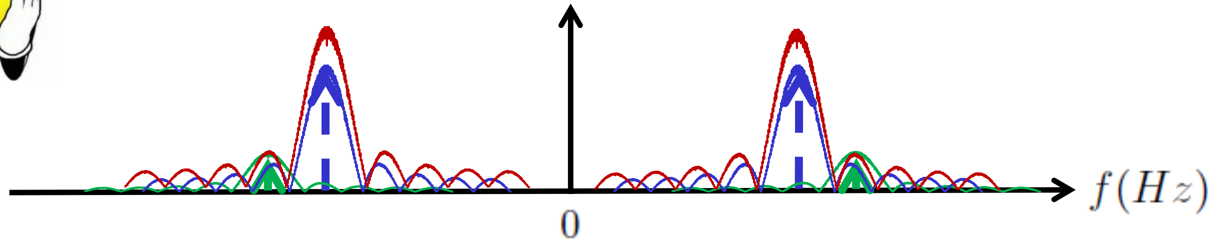


Exemple 2

Somme de deux cosinus dont un de faible puissance



!! Un seul cosinus !!



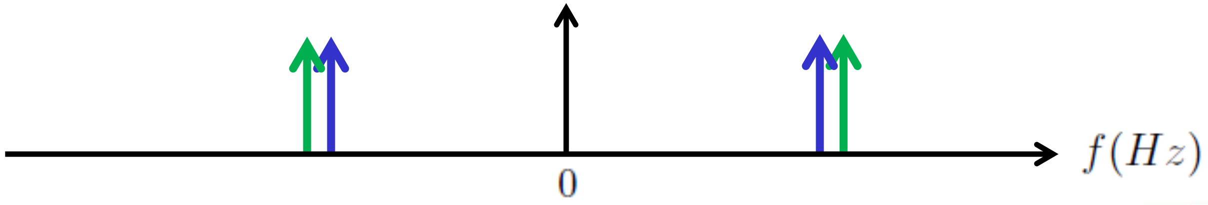
Transformée de Fourier Discrète

Signal de durée limitée

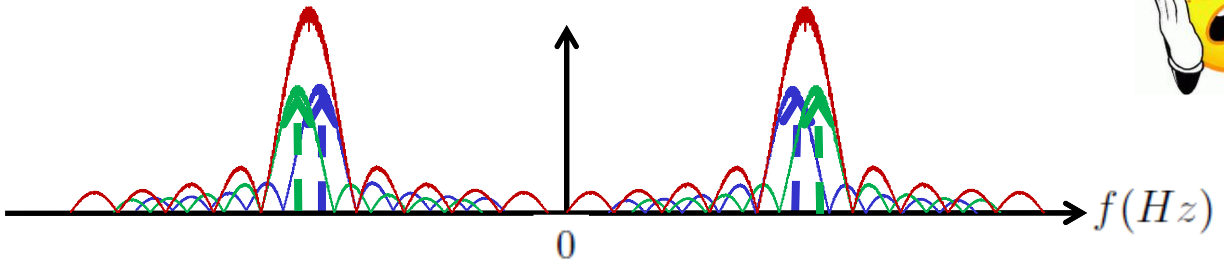
Problèmes posés :

Exemple 1

Somme de deux cosinus proches en fréquences

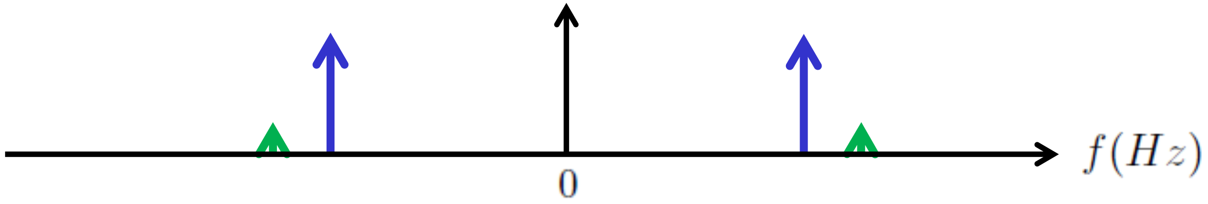


!! Un seul cosinus !!

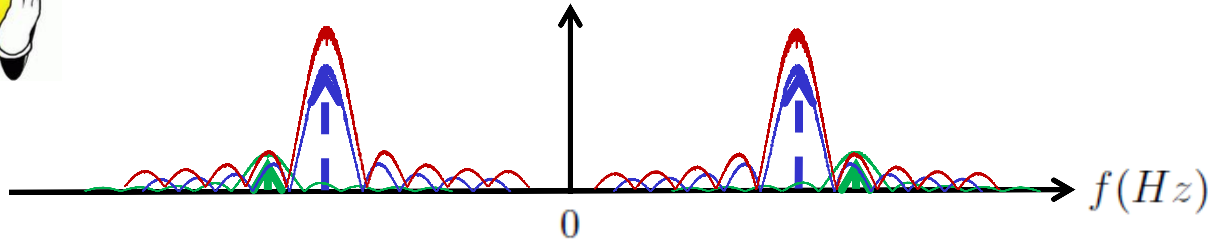


Exemple 2

Somme de deux cosinus dont un de faible puissance



!! Un seul cosinus !!



$\Rightarrow$ L'analyse spectrale NUMERIQUE aura :

un certain **pouvoir séparateur** et un certain **taux d'ondulation**

Capacité à séparer des motifs spectraux proches en fréquences

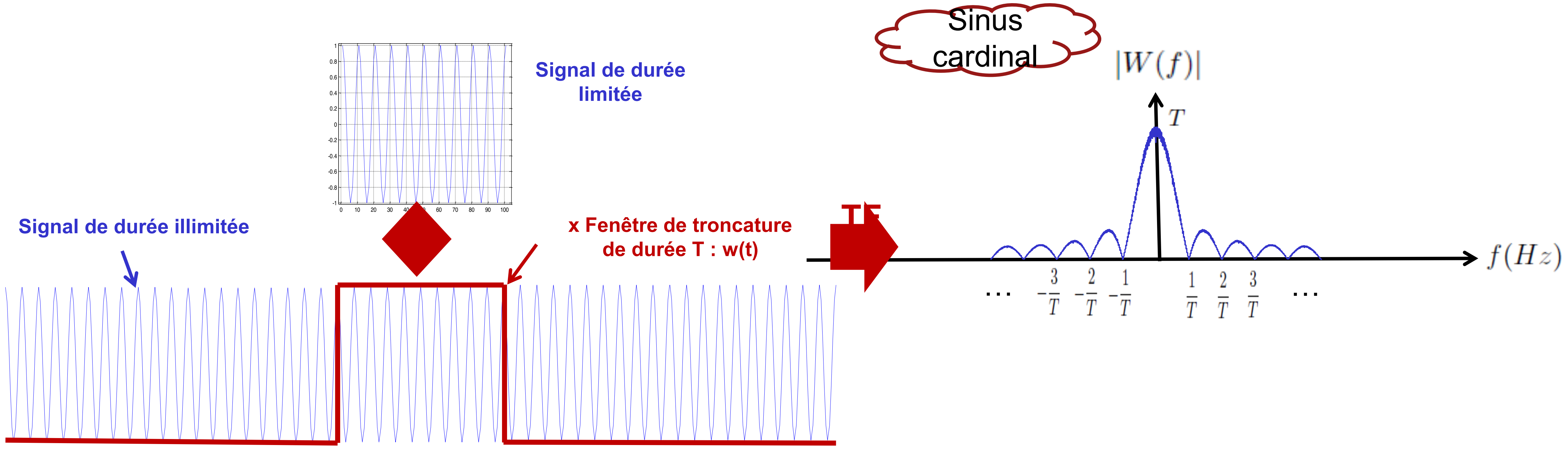
Masquage de motifs spectraux de faibles puissances

Transformée de Fourier Discrète

Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

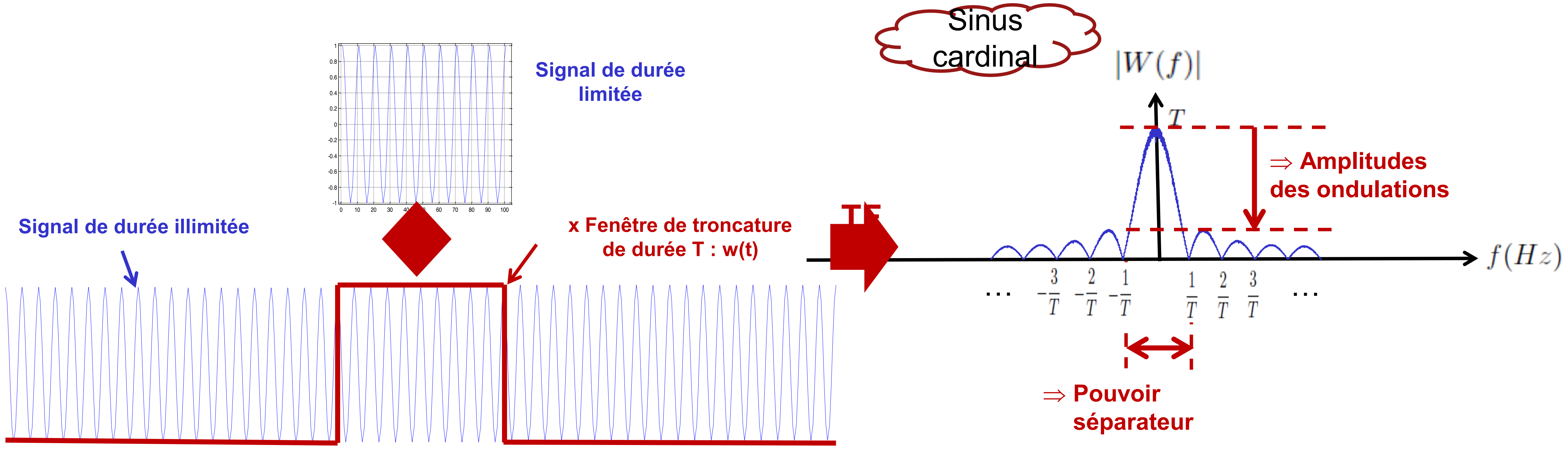


Transformée de Fourier Discrète

Signal de durée limitée

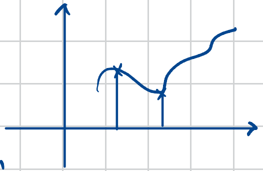
Exemple

$$x(t) = \cos(2\pi f_0 t)\;;\;\; f_0 = 100\ Hz$$

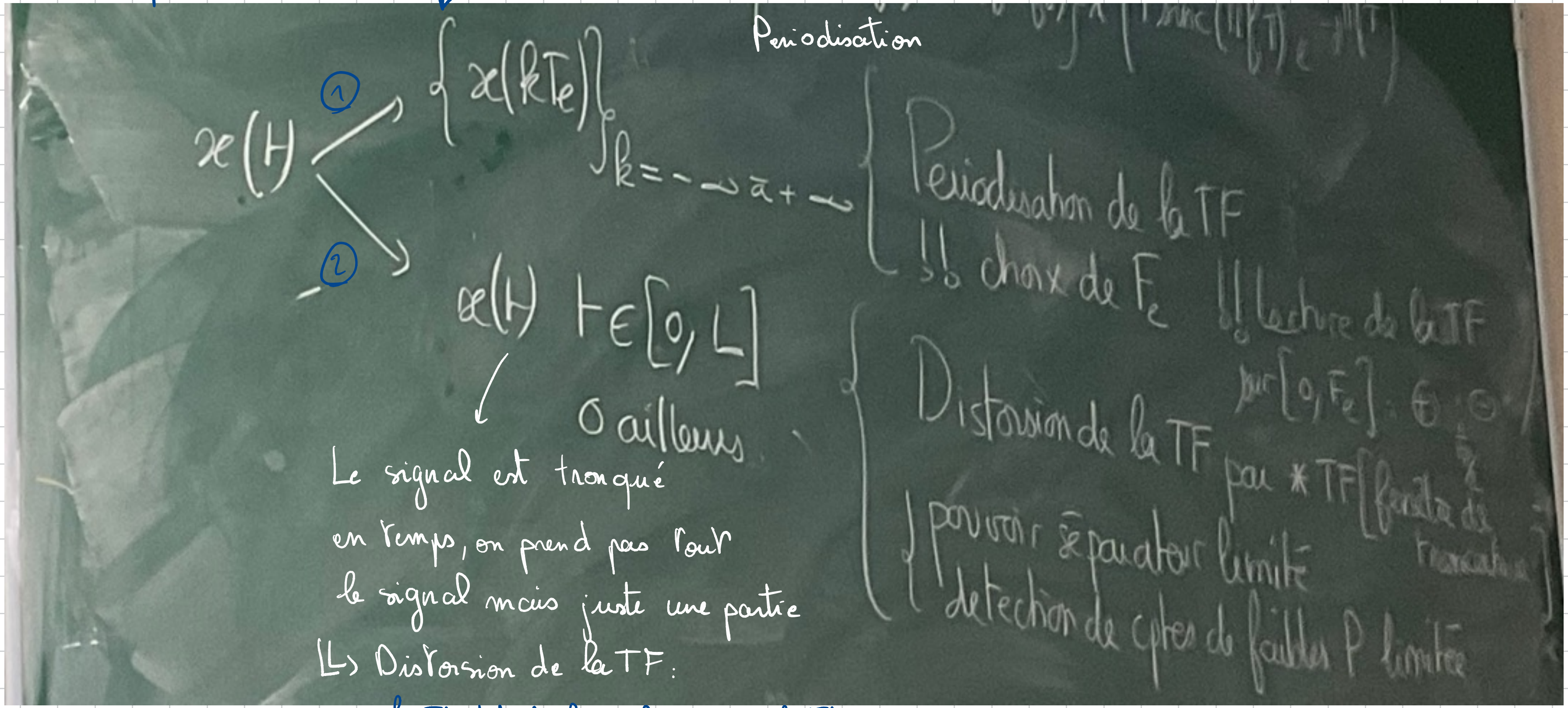


Recap :

on passe d'une intégrale à une somme
On prend pas tout les points mais des échantillons suivant le critère de Shanonn



Periodisation



① $x(t) \rightarrow \{x(kT_e)\}$

② $x(t) \rightarrow x(t)$, $t \in [0, L]$, 0 ailleurs

Le signal est tronqué en temps, on prend pas tout le signal mais juste une partie

↳ Distorsion de la TF :

la TF n'est plus la TF classique mais la TF $* \underbrace{\quad}_{\text{une fenêtre}}$

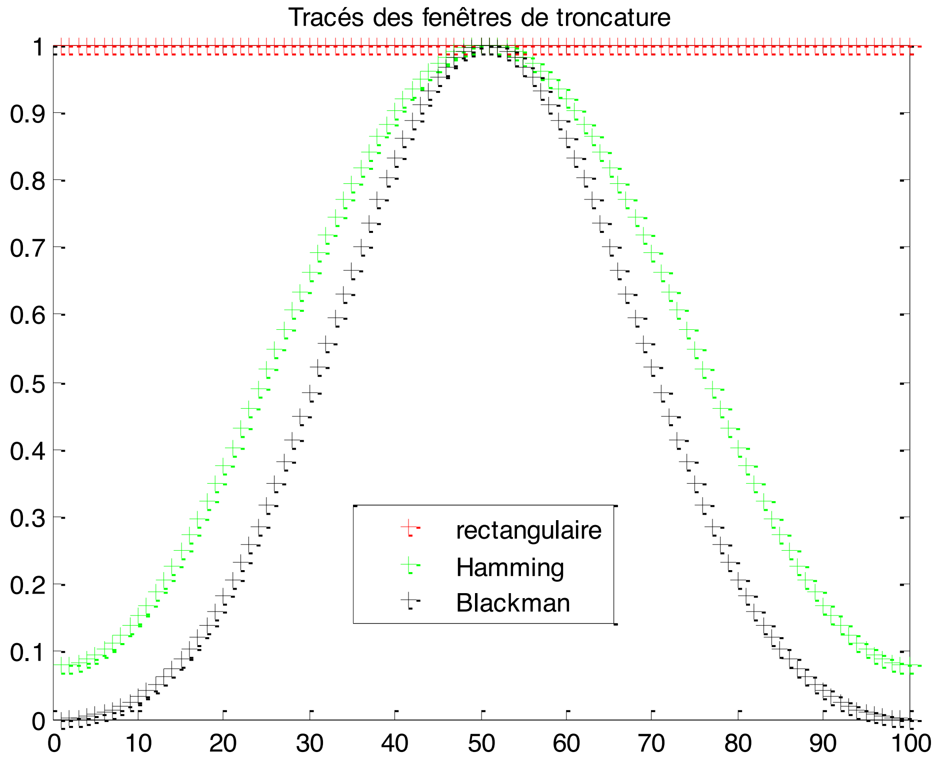
③ Limiter la marge de fréquence

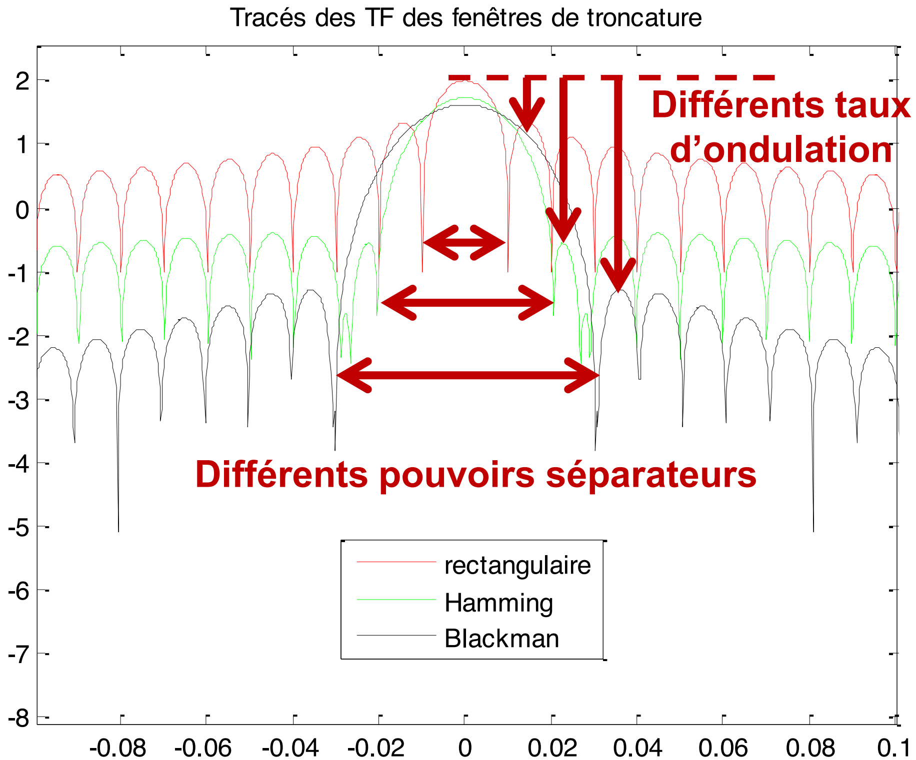
Transformée de Fourier Discrète

Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemples





Transformée de Fourier Discrète

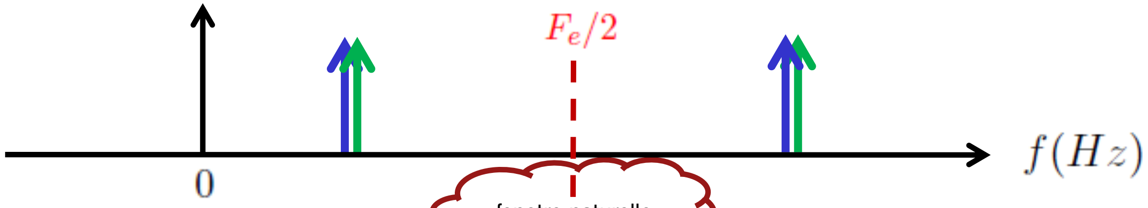
Signal de durée limitée

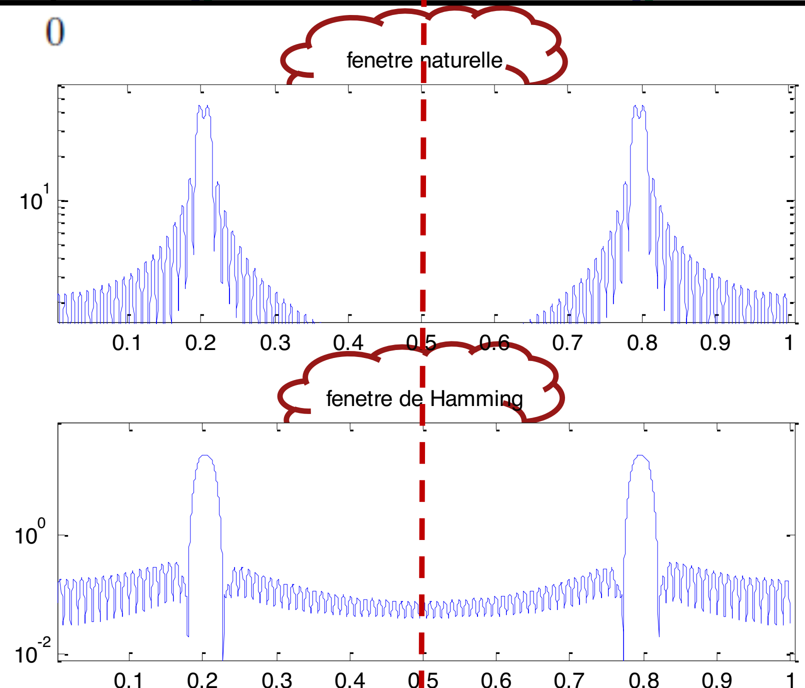
Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 1 : somme de deux cosinus proches en fréquences

```
%Exemple1
%Paramètres
f1=200;        %fréquence du cosinus 1
f2=207;        %fréquence du cosinus 2
Fe=1000;       %fréquence d'échantillonnage
Te=1/Fe;       %période d'échantillonnage
N=100;         %nombre d'échantillons

%Génération du signal
x1=cos(2*pi*f1*[0:Te:N*Te]);
x2=cos(2*pi*f2*[0:Te:N*Te]);
```

```
%fenêtre naturelle
x=x1+x2;
X_V1=fft(x,4096);
%fenêtre de hamming
w=window(@hamming,length(x));
x=(x1+x2).*w.';
X_V2=fft(x,4096);
%Tracés
figure
subplot(2,1,1)
plot(linspace(0,1,4096),log10(abs(X_V1)));
title('fenetre naturelle');
subplot(2,1,2)
plot(linspace(0,1,4096),log10(abs(X_V2)));
title('fenetre de Hamming');
```





Transformée de Fourier Discrète

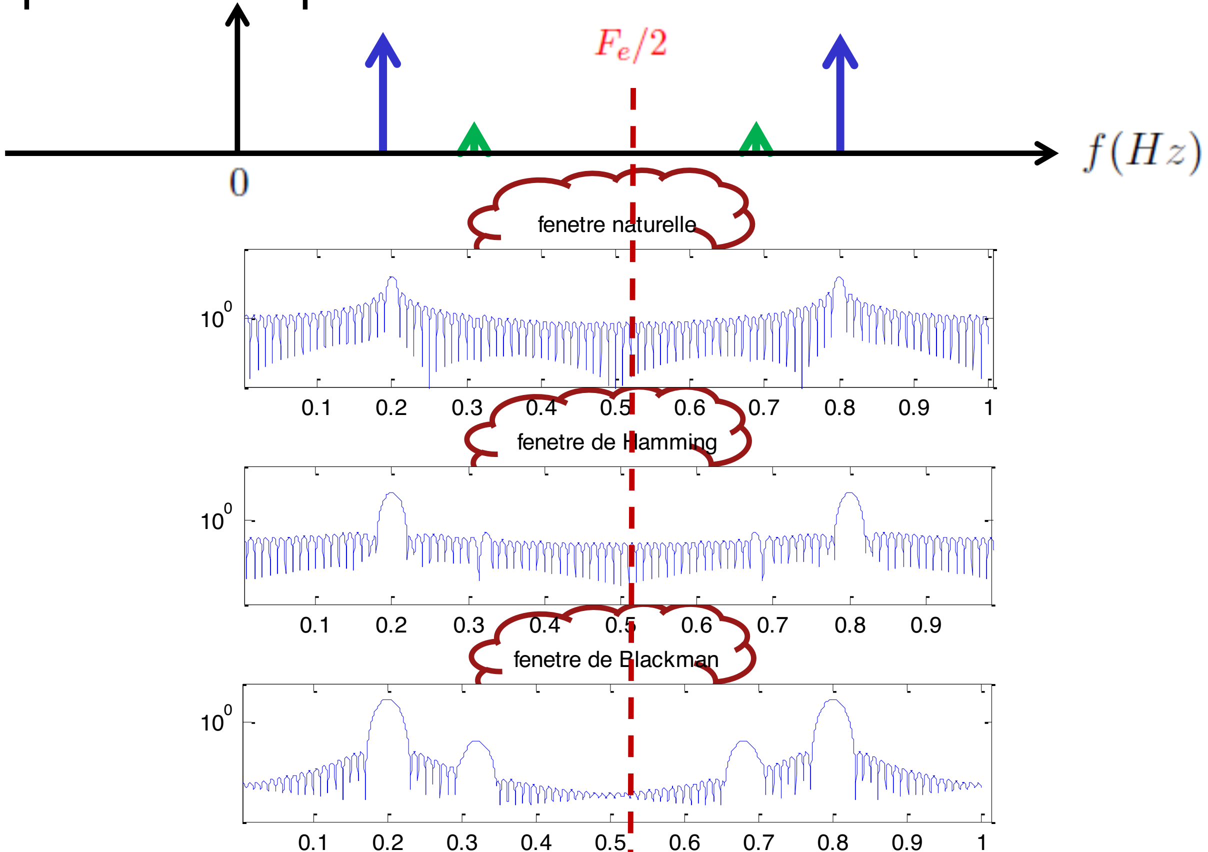
Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 2 : somme de deux cosinus proches de puissances différentes

```
%Exemple 2
%Paramètres
f1=200;     %fréquence du cosinus 1
f2=320;     %fréquence du cosinus 2
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;      %nombre d'échantillons

%Génération du signal
x1=cos(2*pi*f1*[0:Te:N*Te]);
x2=0.005*cos(2*pi*f2*[0:Te:N*Te]);
x=x1+x2;

%fenetre naturelle
X=fft(x,4096);
figure
subplot(3,1,1)
semilogy(linspace(0,1,4096),abs(X));
axis([0 1 10^-5 10^5])
title('fenetre naturelle');
```

```
%fenetre de hamming
w=window(@hamming,length(x));
x=(x1+x2).*w.';
X=fft(x,4096);
subplot(3,1,2)
semilogy(linspace(0,1,4096),abs(X));
title('fenetre de Hamming');

%fenetre de blackman
w=window(@blackman,length(x));
%w=blackman(N);
x=(x1+x2).*w.';
X=fft(x,4096);
subplot(3,1,3)
semilogy(linspace(0,1,4096),abs(X));
title('fenetre de Blackman');
```

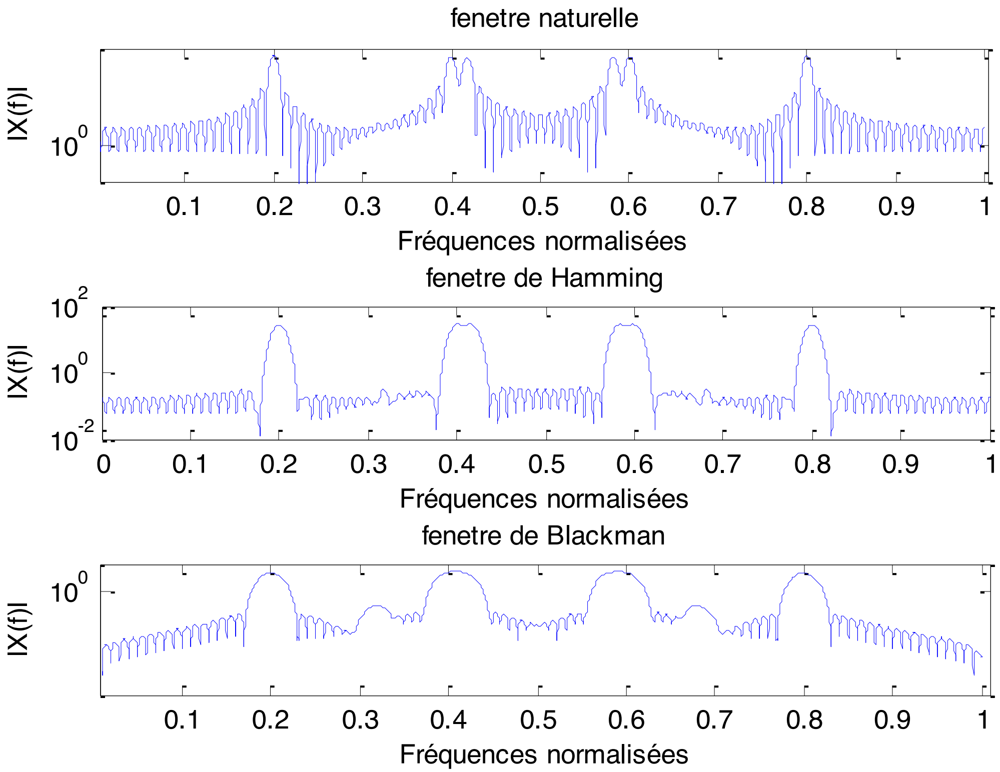


Transformée de Fourier Discrète

Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 3



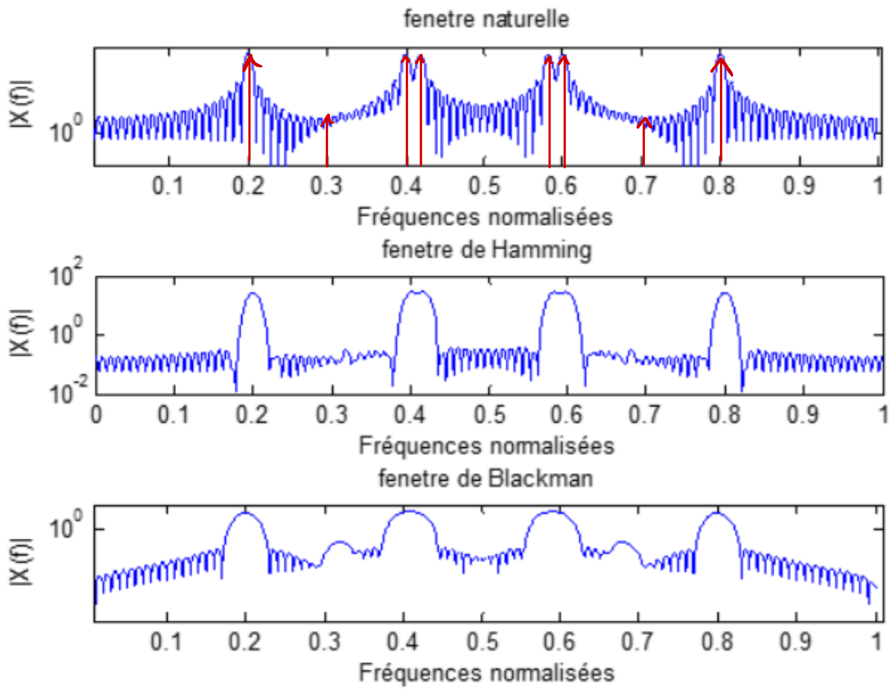
Signal ???

Ce signal est une somme de cosinus.
Il comprend :

1- Deux cosinus
2- Trois cosinus
3- Quatre cosinus
4- Je ne sais pas répondre

QUESTION 4

Allez sur **wooclap.com** et utilisez le code **SIGSEQ2**



Ce tracé représente le module de la transformée de Fourier estimée en numérique d'un signal correctement échantillonné (condition de Shannon respectée). Ce signal est une somme de cosinus. Il s'agit :

1. D'une somme de 2 cosinus
2. D'une somme de 3 cosinus
3. D'une somme de 4 cosinus

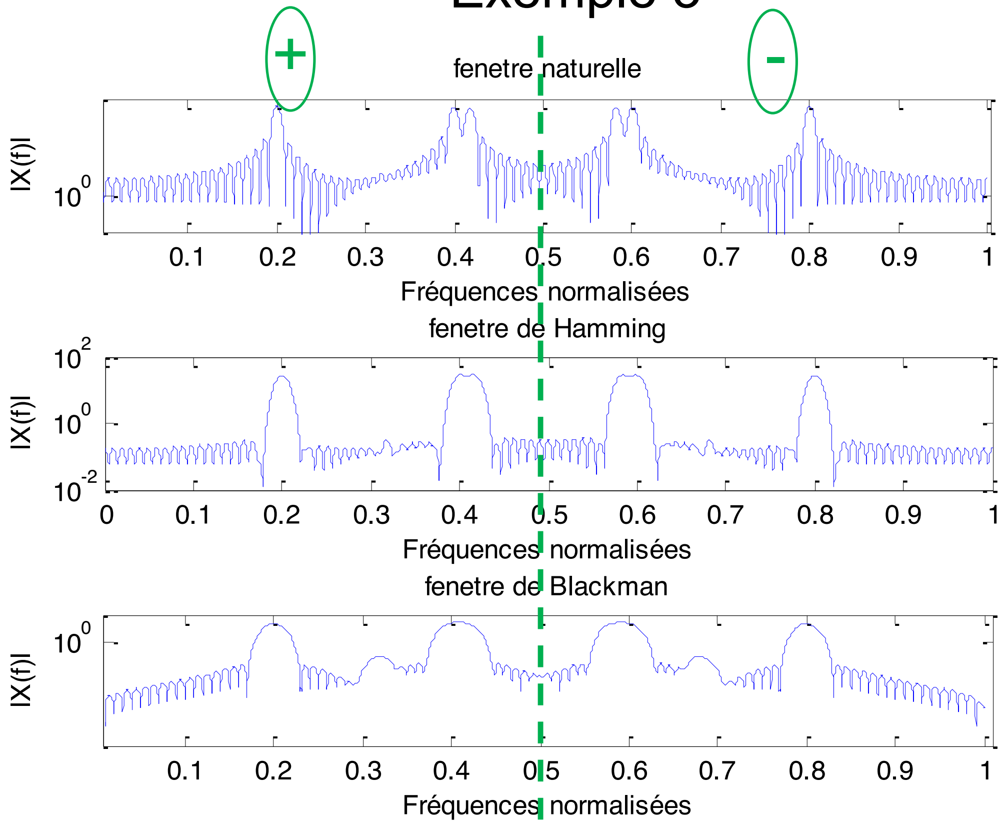
Transformée de Fourier Discrète

Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 3

4 cosinus :
3 de puissances identiques
1 de plus faible puissance



3 cosinus
de puissances identiques

2 cosinus
de puissances identiques
Mais quelque chose de bizarre…

3 cosinus
de puissances différentes

Transformée de Fourier Discrète

Transformée de Fourier (TF) $\longrightarrow$ **Transformée de Fourier Discrète (TFD)**

<u>Signal de durée limitée</u>

$$x(t) \to x_L(t) = \begin{cases} x(t) & \textit{pour } t \in [0, L] \\ 0 & \textit{ailleurs} \end{cases}$$

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{j2\pi f t}\, dt \;\to\; \int_{0}^{L} x(t) e^{j2\pi f t}\, dt = \int_{-\infty}^{+\infty} x(t) w(t) e^{j2\pi f t}\, dt = X(f) * W(f)$$

Impact : Distorsion de la TFD attendue

=> Analyse spectrale numérique avec pouvoir séparateur limité et apparition d’ondulations

=> Utilisation de plusieurs fenêtres de pondération

Transformée de Fourier Discrète

Transformée de Fourier (TF) → **Transformée de Fourier Discrète (TFD)**

Signal échantillonné et de durée limitée

$$x(t) \rightarrow \{x(kT_e)\}_{k=0,\ldots,N-1}$$

$$\boxed{X(f) = \int_{-\infty}^{+\infty} x(t)e^{j2\pi ft}\,dt \;\rightarrow\; \sum_{k=0}^{N-1} x(kT_e)e^{j2\pi fkT_e}} = \sum_{k=-\infty}^{+\infty} x(kT_e)w(kT_e)e^{j2\pi fkT_e}$$

$$= X_e(f) * W_e(f)$$

$$X_e(f) = \sum_{k=-\infty}^{+\infty} x(kT_e)e^{j2\pi fkT_e}$$

Noyau de Dirichlet

$$W_e(f) = \sum_{k=-\infty}^{+\infty} 1 \times e^{j2\pi fkT_e} = e^{-j\pi f(N-1)T_e}\frac{\sin(\pi fNT_e)}{\sin(\pi fT_e)}$$

Impact : Périodisation et distorsion de la TFD

$\Rightarrow$!! Respecter la condition de Shannon !!

$\Rightarrow$!! Lecture des tracés et échelle !!

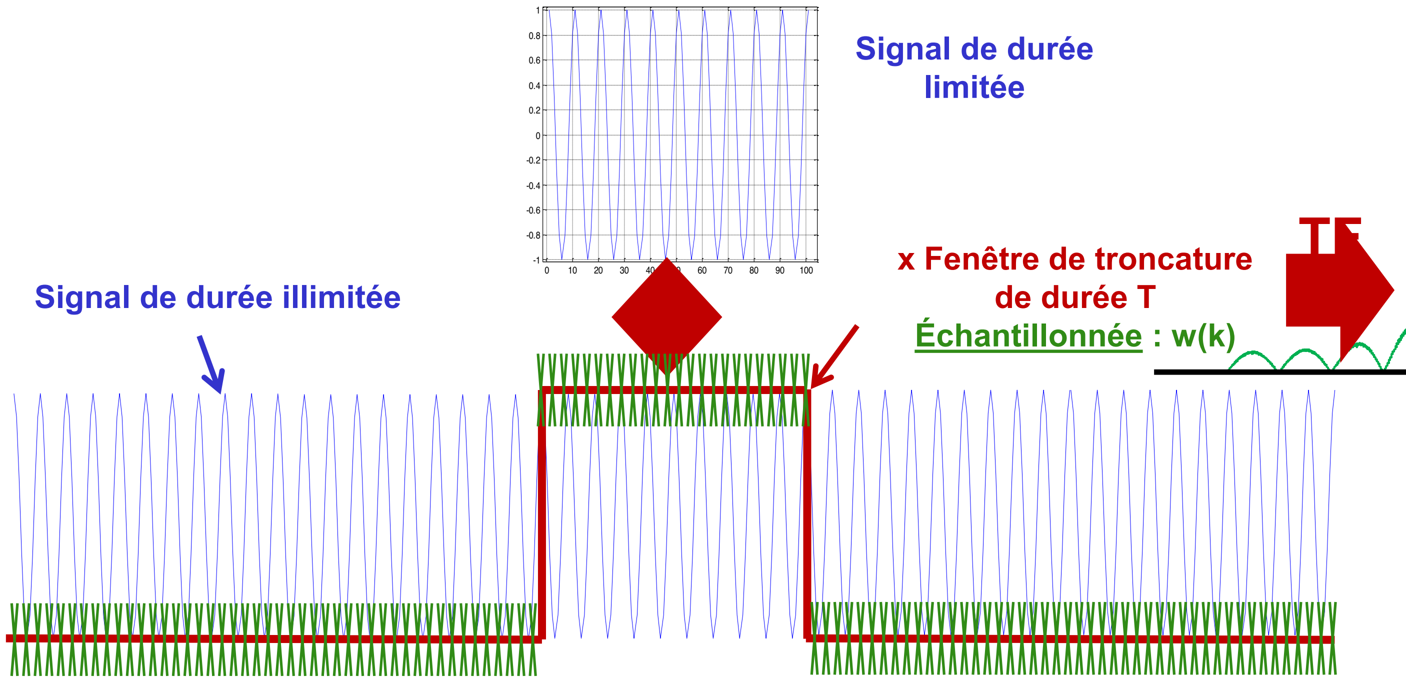
$\Rightarrow$ Utilisation de plusieurs fenêtres de troncature ou fenêtres de pondération

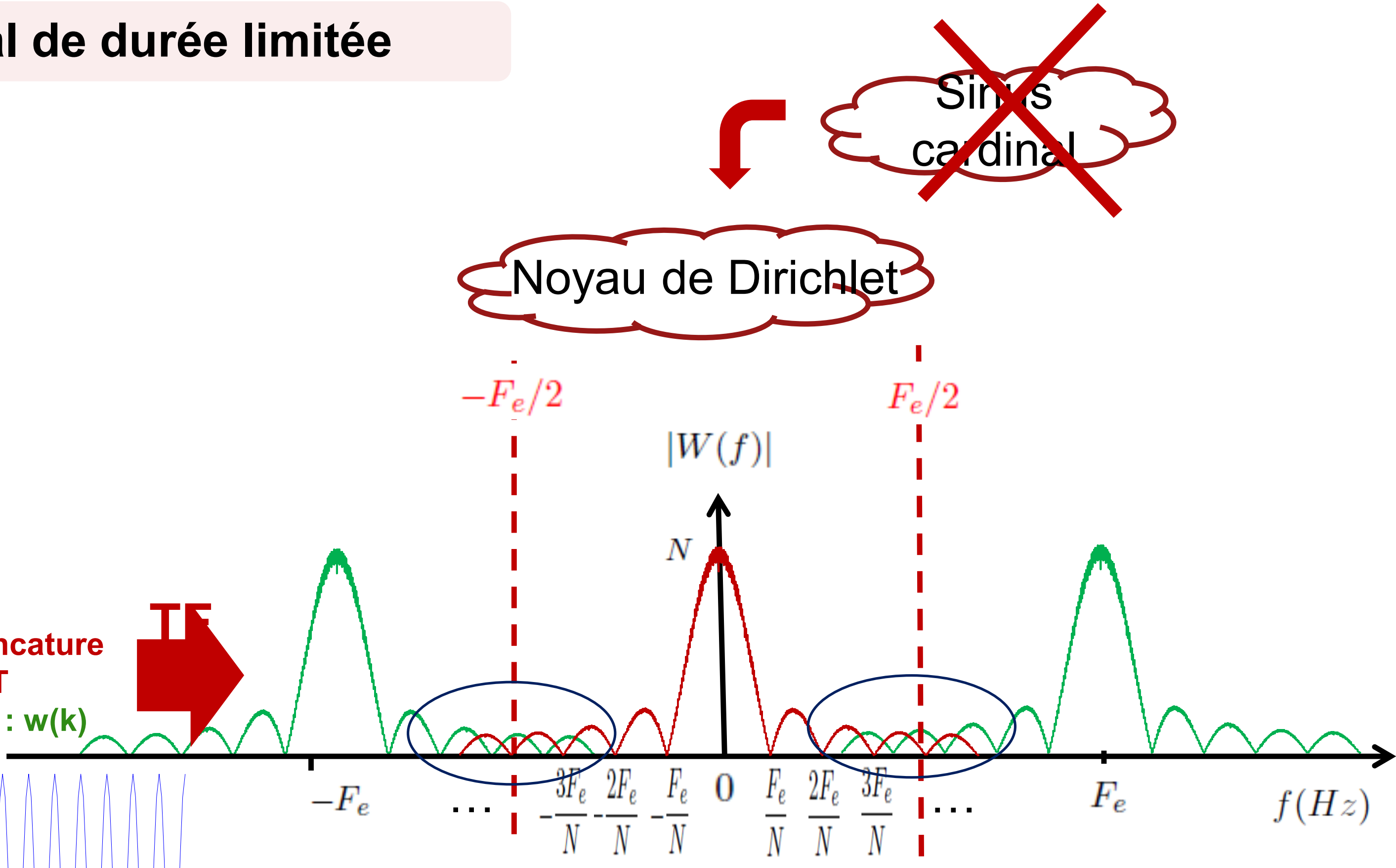
Transformée de Fourier Discrète

Signal de durée limitée

Exemple

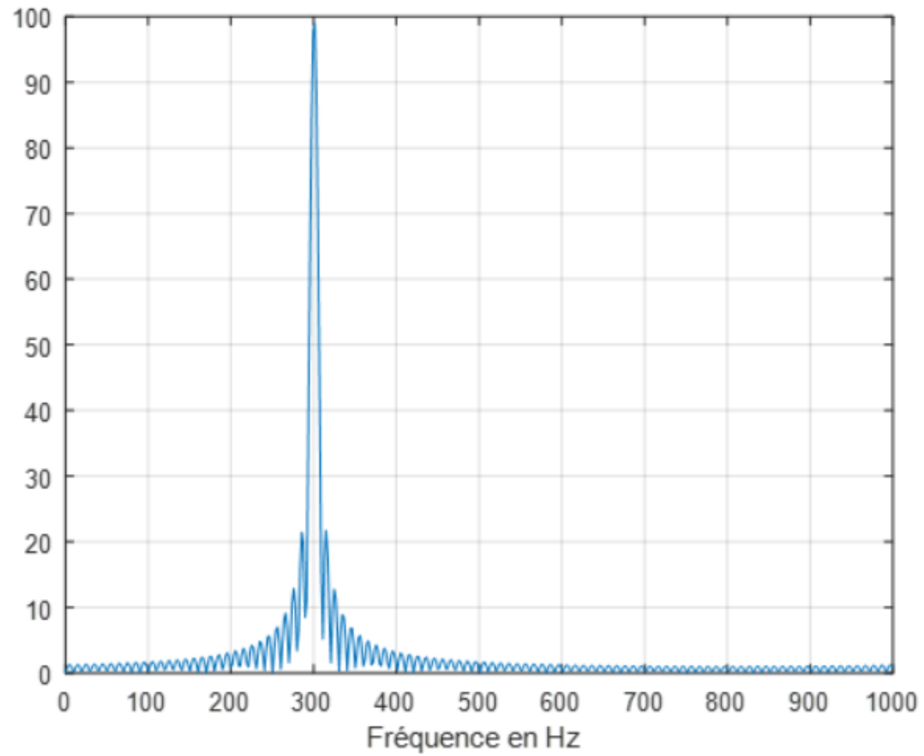
$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$





QUESTION 5

Allez sur **wooclap.com** et utilisez le code **SIGSEQ2**



Sachant que le signal a été correctement échantillonné avec une fréquence d'échantillonnage Fe=1000 Hz, ce tracé représente le module de la transformée de Fourier estimée en numérique :

1. **d'un cosinus de fréquence 300Hz**
2. **d'une exponentielle de fréquence 300Hz**
3. **d'une fonction porte de largeur 0.05 s**

Transformée de Fourier Discrète

Signal échantillonné et de durée limitée

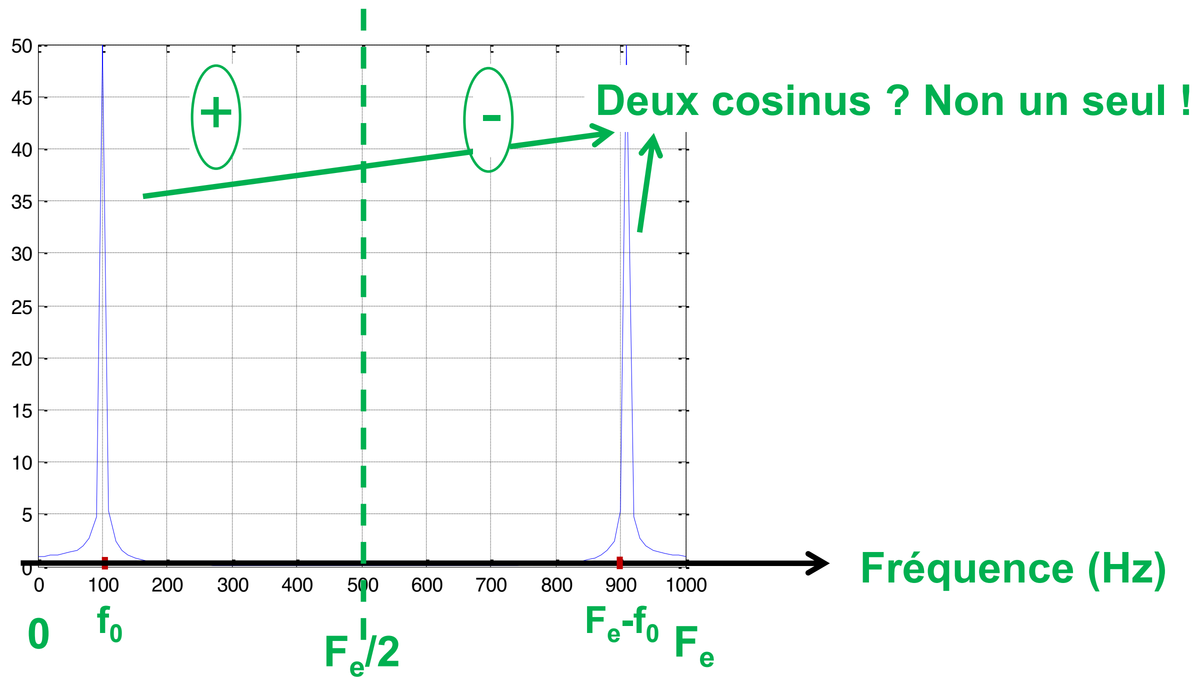
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=101;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure, plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure
plot(linspace(0,Fe,length(X)),abs(X))
```

Tracé du module de la TFD obtenu :



Transformée de Fourier Discrète

Signal échantillonné et de durée limitée

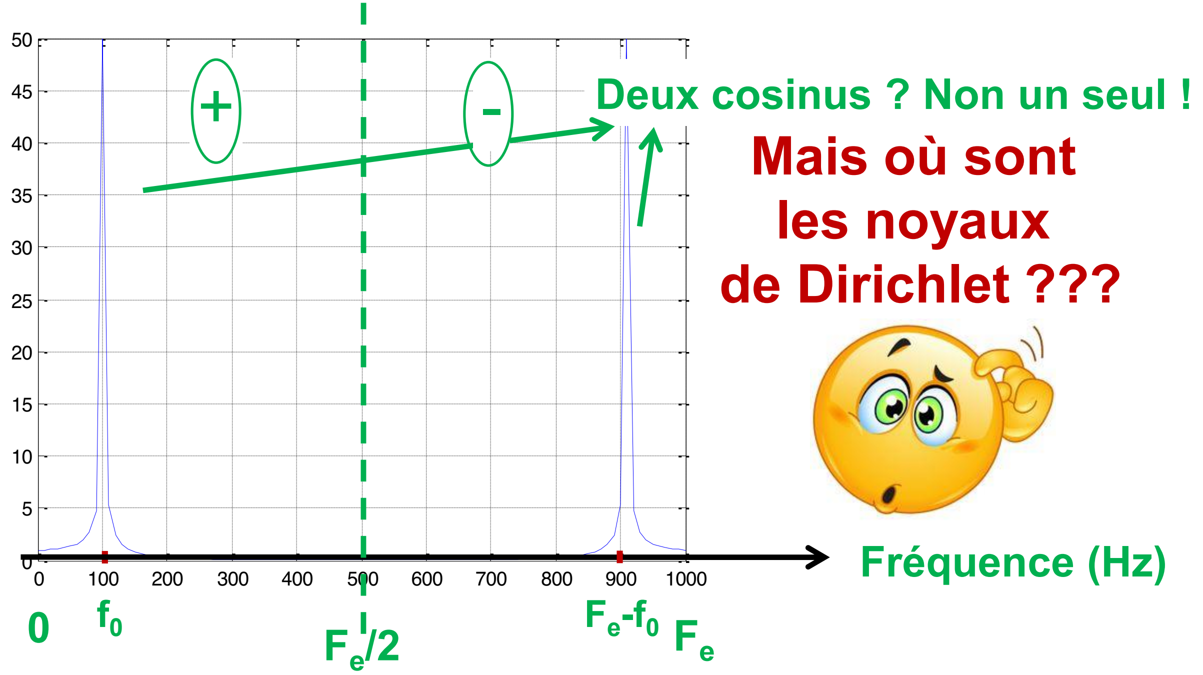
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=101;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure, plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure
plot(linspace(0,Fe,length(X)),abs(X))
```

Tracé du module de la TFD obtenu :

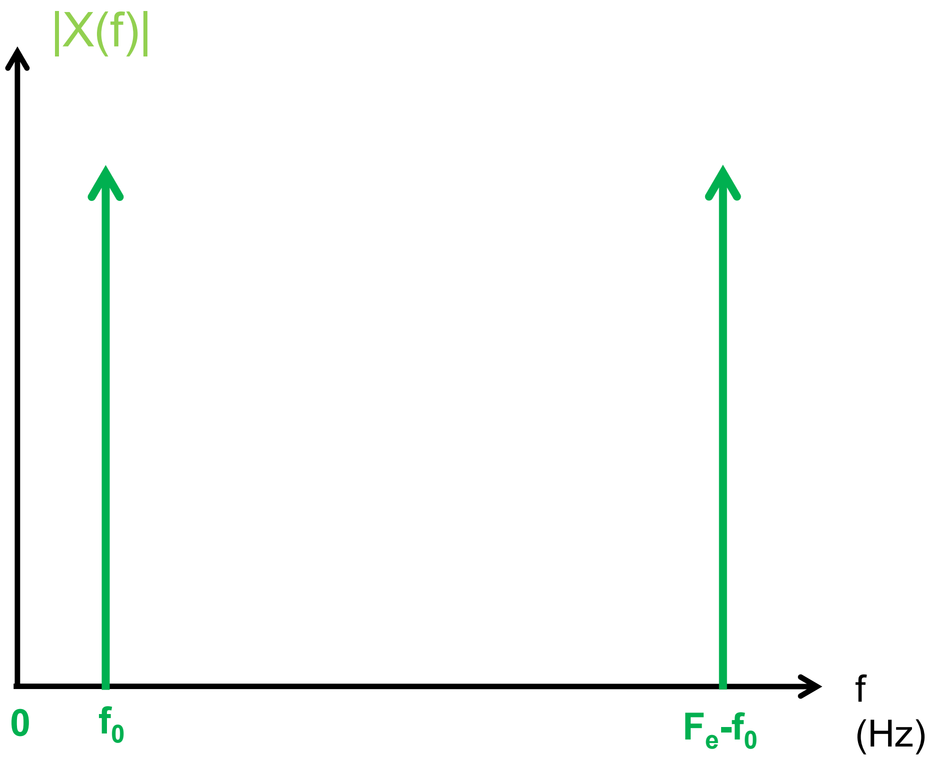


Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

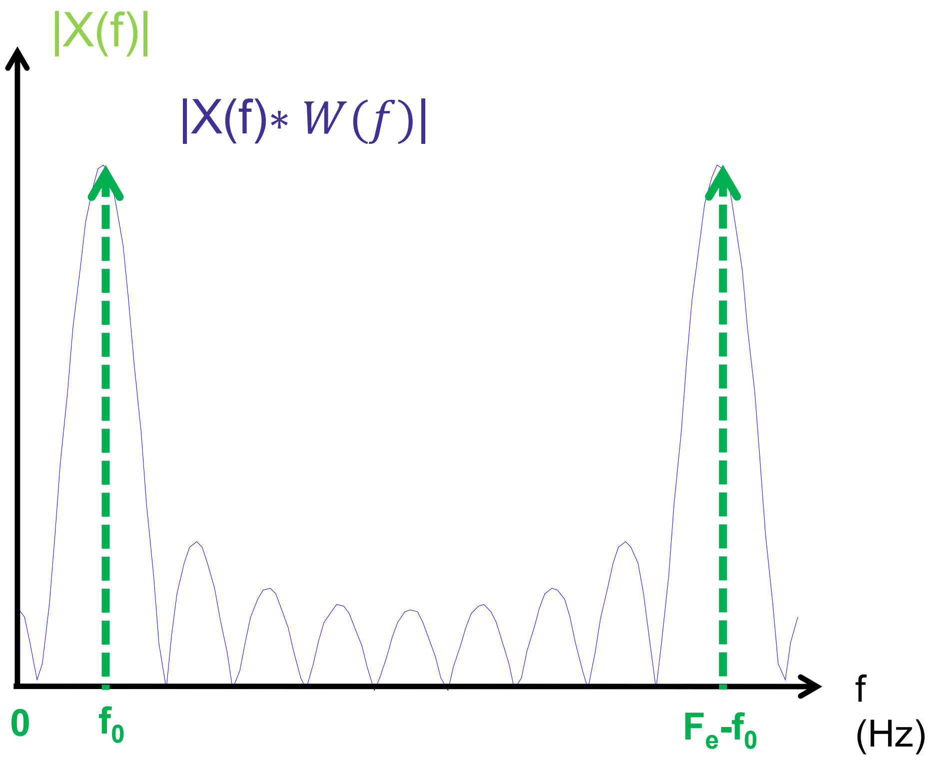
$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$



Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

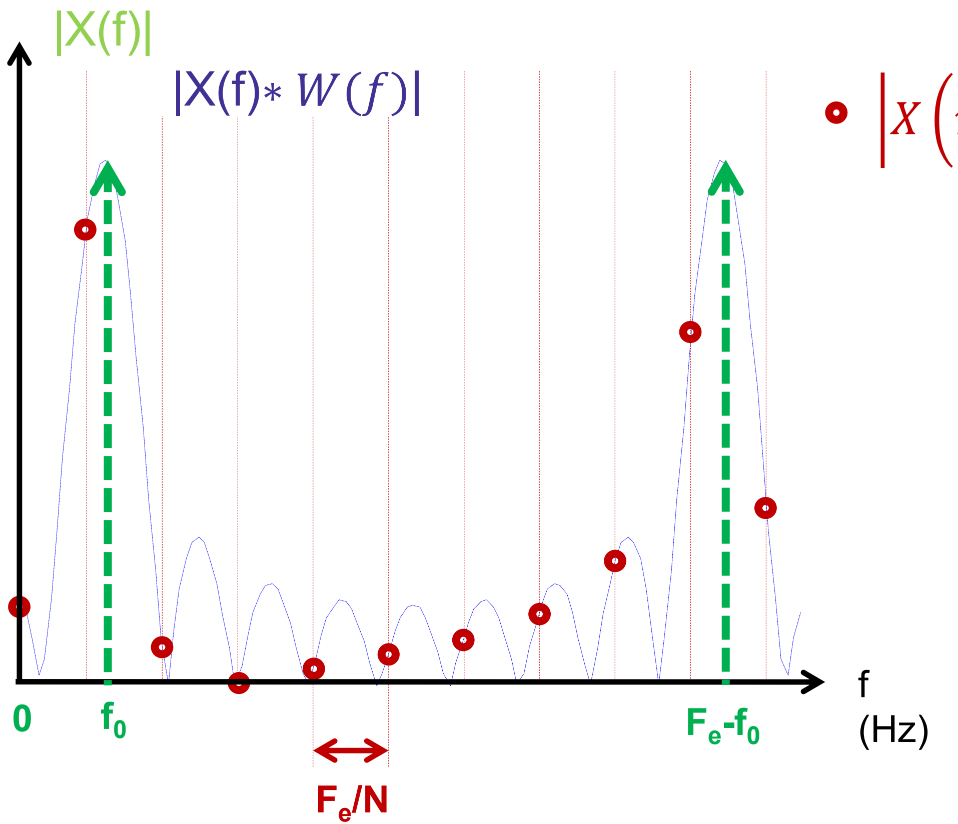


Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$



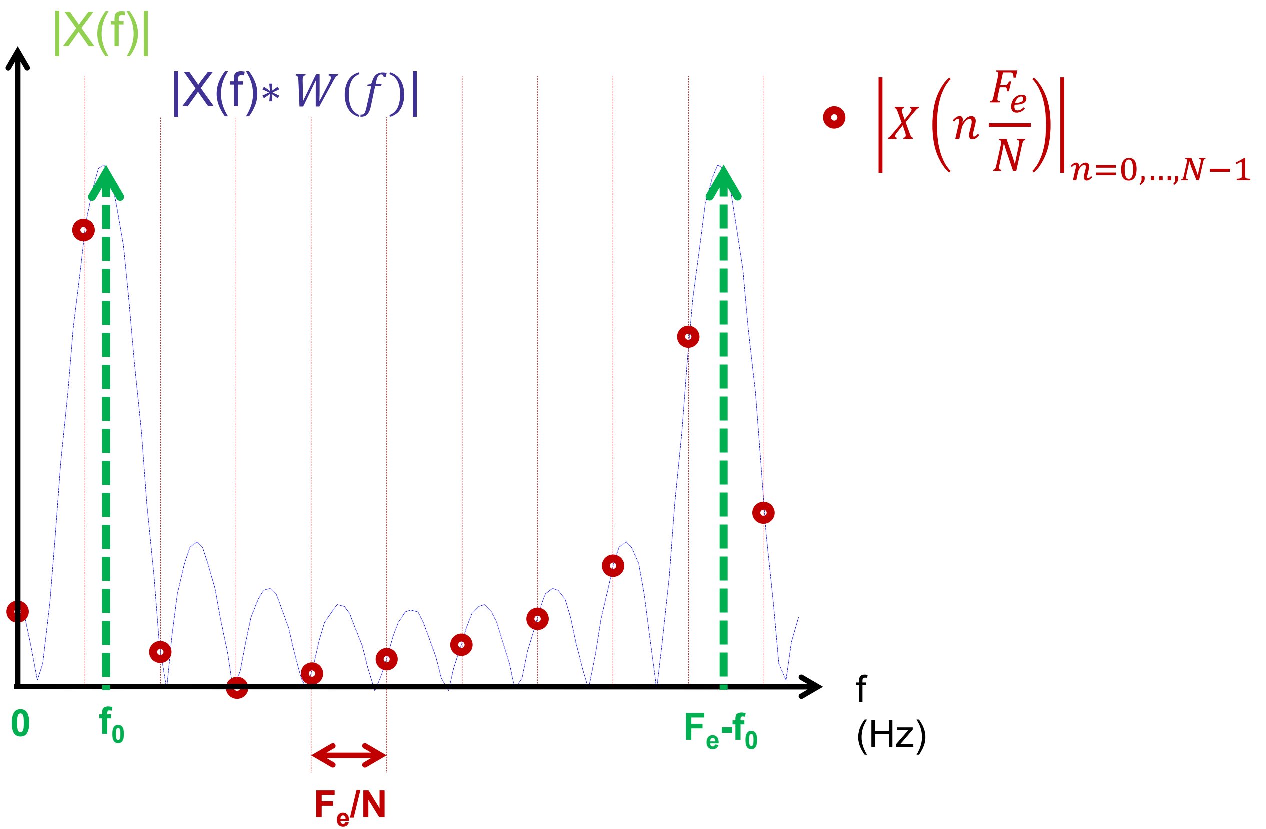
$$\left| X\left(n\frac{F_e}{N} \right) \right|_{n=0,\ldots,N-1}$$

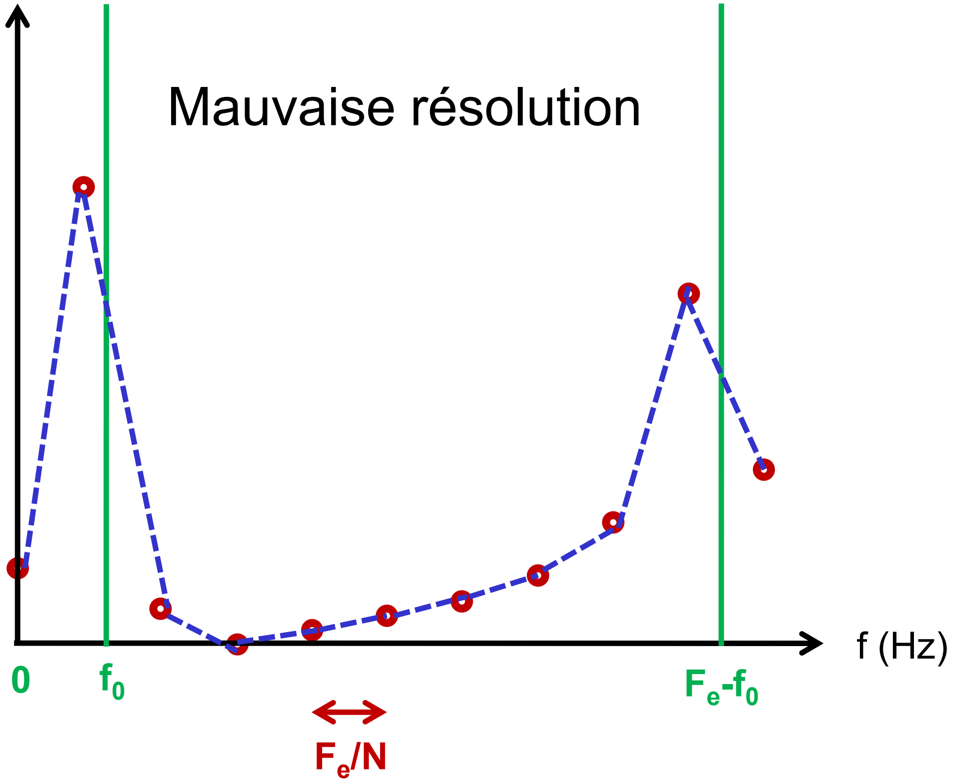
Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$





Transformée de Fourier Discrète

TF classique

$$Y(n) = \sum_{k=0}^{N-1} x(kT_e)e^{-j2\pi\frac{kn}{N}}$$

$F_e T_e = 1$

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

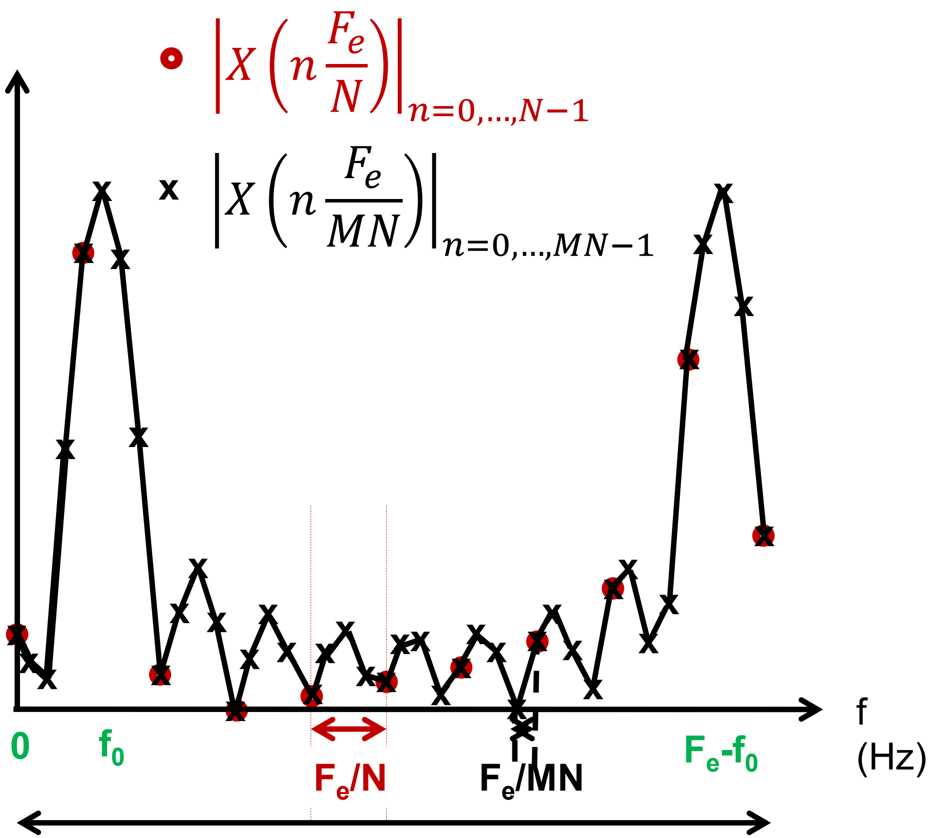
Interpolation par Zero padding :

$$y(k) \begin{cases} = x(k)\ pour\ k = 0, \ldots, N-1 \\ = 0\ pour\ k = N, \ldots, MN-1 \end{cases}$$

je rajoute juste des zéros

$$Y(n) = \sum_{k=0}^{N-1} x(kT_e)e^{-j2\pi\frac{kn}{MN}} \qquad pour\ k = 0, \ldots, MN-1$$

TFD calculée avec un pas de $\frac{F_e}{MN}$



MN points espacés de F_e/MN entre 0 et F_e

Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

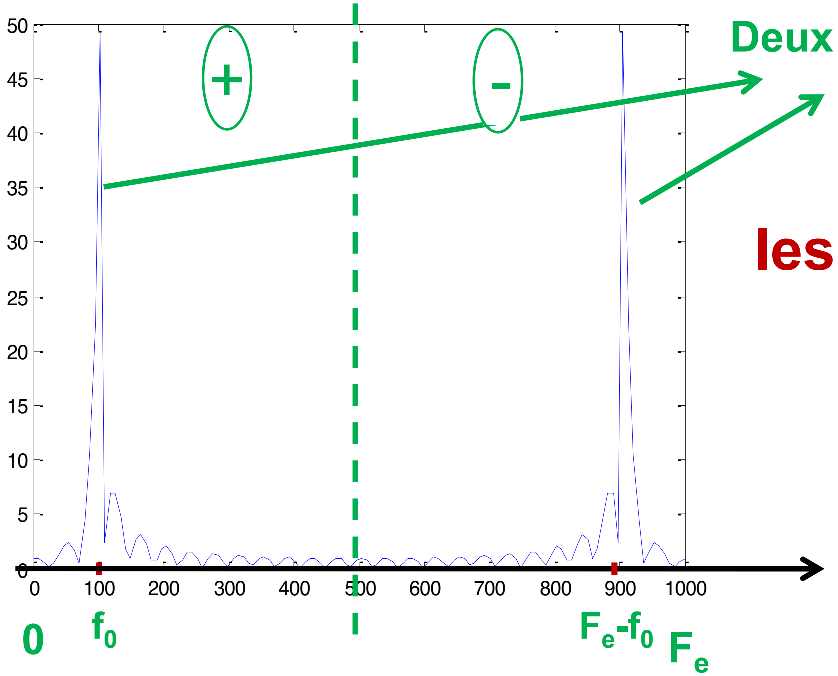
Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X=fft(x,128);
```

Utilisation de Zero Padding

```
%Tracé du module de la TFD du signal
figure;
plot(linspace(0,Fe,length(X)),abs(X))
```

Tracé du module de la TFD obtenu :



Deux cosinus ? Non un seul !

Voilà
les noyaux de Dirichlet !

QUESTION 6

Allez sur **wooclap.com** et utilisez le code **SIGSEQ2**

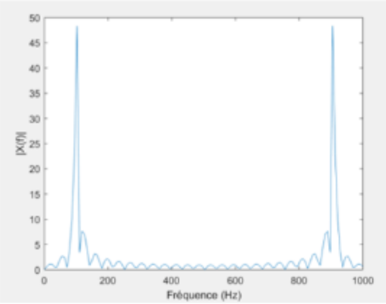


Figure 1 (ZP1)

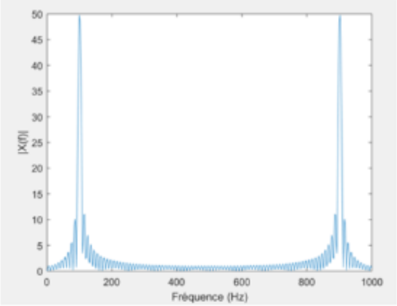


Figure 2 (ZP2)

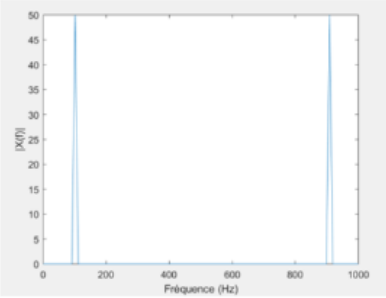


Figure 3 (ZP3)

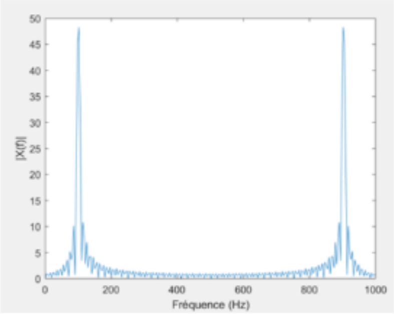


Figure 4 (ZP4)

Les quatre figures représentent le module de la TFD d'un cosinus de fréquence 100 Hz échantillonné à 1000 Hz. Elles utilisent différents paramètres de Zero Padding pour ce tracé (ZP1, ZP2, ZP3, ZP4). A t-on :

1. ZP1>ZP2>ZP3>ZP4
2. ZP2>ZP4>ZP1>ZP3
3. ZP4>ZP3>ZP2>ZP1
4. ZP3>ZP1>ZP4>ZP2

Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

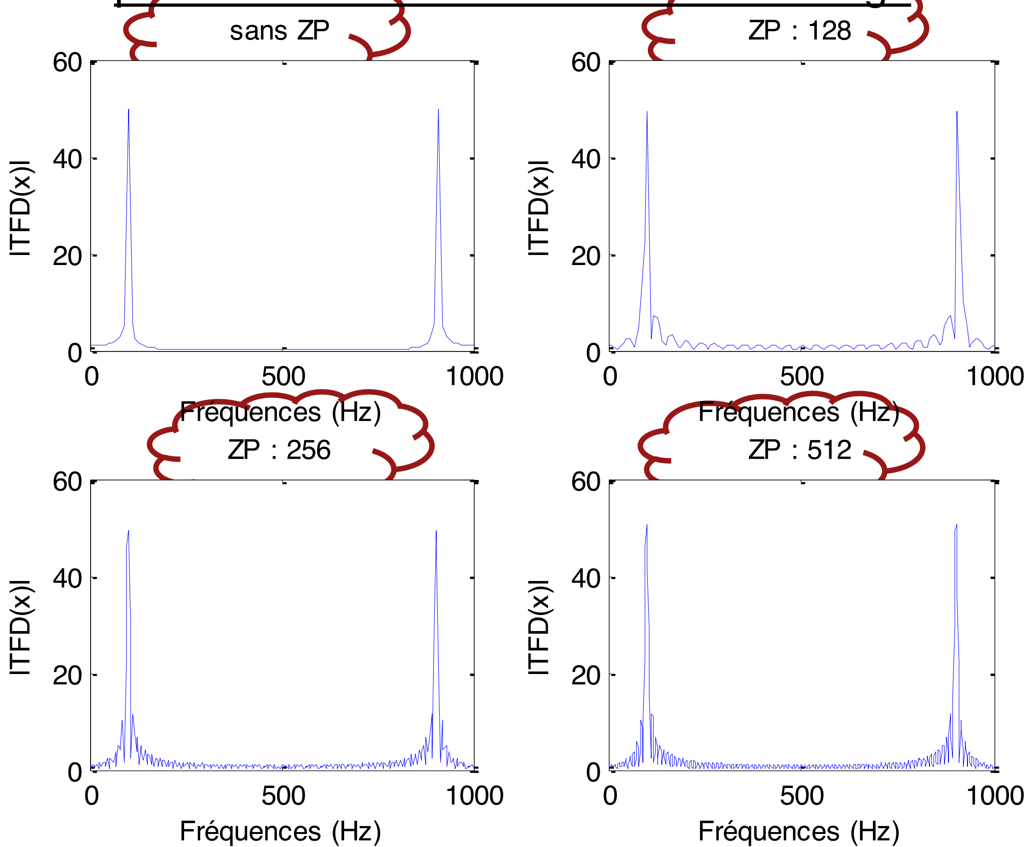
$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Simulation sous Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X1=fft(x);
X2=fft(x,128);
X3=fft(x,256);
X4=fft(x,512);
```

```
% Tracé du module de la TFD du signal
figure;
subplot(2,2,1)
plot(linspace(0,Fe,length(X1)),abs(X1))
xlabel('Fréquences (Hz)')
subplot(2,2,2)
plot(linspace(0,Fe,length(X2)),abs(X2))
xlabel('Fréquences (Hz)')
subplot(2,2,3)
plot(linspace(0,Fe,length(X3)),abs(X3))
xlabel('Fréquences (Hz)')
subplot(2,2,4)
plot(linspace(0,Fe,length(X4)),abs(X4))
xlabel('Fréquences (Hz)')
```

Tracé du module de la TFD du signal pour différentes valeurs de Zero Padding :



Transformée de Fourier Discrète

Transformée de Fourier (TF) $\longrightarrow$ **Transformée de Fourier Discrète (TFD)**

Echantillonnage fréquentiel

$$X(f) = \sum_{k=0}^{N-1} x(kT_e)e^{-j2\pi f kT_e} \rightarrow \left\{X\left(n\frac{F_e}{N}\right)\right\}_{n=0,\ldots,N-1}$$

$$\boxed{X(f) = \int_{-\infty}^{+\infty} x(t)e^{j2\pi f t}\,dt \rightarrow \sum_{k=0}^{N-1} x(kT_e)e^{j2\pi f kT_e} \rightarrow \sum_{k=0}^{N-1} x(kT_e)e^{j2\pi\frac{nk}{N}}\ , n = 0, \ldots, N-1}$$

Impact : Mauvaise résolution de la TFD

=> Nécessité d’interpoler dans le domaine des fréquences (Zero Padding)

Transformée de Fourier (TF)	Transformée de Fourier Discrète (TFD)
$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft}$	$X(n) = \sum_{n=0}^{N-1} x(k) e^{-j2\pi\frac{kn}{N}}, n = 0, \dots, N-1$
Transformée de Fourier inverse (TF^{-1})	**Transformée de Fourier Discrète inverse (TFD^{-1})**
$x(t) = \int_{-\infty}^{+\infty} X(f) e^{+j2\pi ft}\, df$	$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) e^{+j2\pi\frac{kn}{N}}, k = 0, \dots, N-1$

Echantillonnage temporel

$$x(t) \rightarrow \{x(kT_e)\}_{k=-\infty,\dots,+\infty}$$

⇒ Périodisation de la TFD

⇒ !! Respecter la condition de Shannon !!

⇒ !! Lecture des tracés !!

Signal de durée limitée

$$\{x(kT_e)\}_{k=-\infty,\dots,+\infty} \rightarrow \{x(kT_e)\}_{k=0,\dots,N-1}$$

⇒ Distorsion de la TFD attendue (analyse spectrale numérique avec pouvoir séparateur limité et apparition d'ondulations)

⇒ Utilisation de plusieurs fenêtres de pondération

Echantillonnage fréquentiel

$$X(f) = \sum_{k=0}^{N-1} x(kT_e) e^{-j2\pi fkT_e} \rightarrow \left\{X\left(n\frac{F_e}{N}\right)\right\}_{n=0,\dots,N-1}$$

⇒ Mauvaise résolution de la TFD obtenue

⇒ Interpolation fréquentielle par Zero Padding

Transformée de Fourier Discrète

Propriétés

→ Linéarité $TFD\left[x_1(k)+\lambda x_2(k)\right]=TFD\left[x_1(k)\right]+\lambda TFD\left[x_2(k)\right]$

→ Translation => rotation de phase $TFD\left[x(k-k_0)\right]=X(n)e^{-j2\pi\frac{k_0 n}{N}}$

→ Symétrie hermitienne $X(N-n)=X(-n)=X^*(n).$

!! La TFD et la TFD^{-1} transforment un produit en produit de convolution circulaire !!

→ Convolution circulaire $X_1(n)X_2(n)\xrightarrow{TFD^{-1}}x_1(k)\otimes x_2(k)=\sum_{p=0}^{N-1}x_1(p)x_2\left([k-p]_{modulo\ N}\right)$

→ Egalité de Parseval $\sum_{k=0}^{N-1}|x(k)|^2=\frac{1}{N}\sum_{n=0}^{N-1}|X(n)|^2$

→ Algorithme de calcul rapide (Fast Fourier Transform Algorithm : FFT) : $N\log_2(N)$ MAC << N^2 pour N points

Transformée de Fourier Discrète

Convolution circulaire

Convolution linéaire (« classique ») :

$$(x_1 * x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2(k-p)$$

Convolution circulaire :

$$(x_1 \otimes x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2\left((k-p)_{modulo\ N}\right)$$

Exemple

$x_1(p)$: . . . 0 0 0 1 2 3 0 0 0 . . .
$x_2(p)$: . . . 0 0 0 1 1 1 0 0 0 . . .

k=-3 . . 0 0 0 1 1 1 0 0 0 . . . → 0
k=-2 . . . 0 0 0 1 1 1 0 0 0 . . . → 1
k=-1 . . . 0 0 0 1 1 1 0 0 0 . . . → 3
k=0 . . . 0 0 0 1 1 1 0 0 0 . . . → 6
k=1 . . . 0 0 0 1 1 1 0 0 0 . . . → 5
k=2 . . . 0 0 0 1 1 1 0 0 0 . . . → 3
k=3 . . . 0 0 0 1 1 1 0 0 0 . . . → 0

$x_1(p_{modulo\ N})$: . . . 1 2 3 1 2 3 1 2 3 . . .
$x_2(p_{modulo\ N})$: . . . 1 1 1 1 1 1 1 1 1 . . .

k=-3 . . 1 1 1 1 1 1 1 1 1 . . . → 6
k=-2 . . . 1 1 1 1 1 1 1 1 1 . . . → 6
k=-1 . . . 1 1 1 1 1 1 1 1 1 . . . → 6
k=0 . . . 1 1 1 1 1 1 1 1 1 . . . → 6
k=1 . . . 1 1 1 1 1 1 1 1 1 . . . → 6
k=2 . . . 1 1 1 1 1 1 1 1 1 . . . → 6
k=3 . . . 1 1 1 1 1 1 1 1 1 . . . → 6

Transformée de Fourier Discrète

Convolution circulaire

Convolution lineaire (« classique ») :

$$(x_1 * x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2(k-p)$$

Convolution circulaire :

$$(x_1 \otimes x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2\left((k-p)_{modulo\ N}\right)$$

En prolongeant les signaux par N zéros

. . . 0 0 0 1 2 3 0 0 0 . . .
. . . 0 0 0 1 1 1 0 0 0 . . .

. . . 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0
. . . 0 0 0 1 1 **1** 0 0 0 . . . $\longrightarrow$ 1
. . . 0 0 0 1 **1 1** 0 0 0 . . . $\longrightarrow$ 3
. . . 0 0 0 **1 1 1** 0 0 0 . . . $\longrightarrow$ 6
. . . 0 0 0 **1 1** 1 0 0 0 . . . $\longrightarrow$ 5
. . . 0 0 0 **1** 1 1 0 0 0 . . . $\longrightarrow$ 3
. . . 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0

. . . 1 2 3 0 0 0 1 2 3 0 0 0 1 2 3 0 0 0. . .
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . .

. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 1
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 3
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 6
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 5
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 3
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $\boldsymbol{N = 2^p} \Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi\frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k) \times W_N^{-kn} \quad n = 0, \ldots, N-1$$

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $\boldsymbol{N = 2^p} \Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi \frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k) \times W_N^{-kn} \quad n = 0, \ldots, N-1$$

Première décomposition $\Rightarrow N + 2(N/2)^2 = N + N^2/2 < N^2$ opérations (+/x)

$$X(n) = X_1(n) + W_N^{-n} \times X_2(n), \quad n = 0, \ldots, N-1$$

N opérations (+/x)

Indices pairs
(TFD d'ordre N/2)

$$X_1(n) = \sum_{i=0}^{N/2-1} x(2i) W_{N/2}^{-in}, \quad i = 0, \ldots, N/2-1$$

$$X_2(n) = \sum_{i=0}^{N/2-1} x(2i+1) W_{N/2}^{-in}, \quad i = 0, \ldots, N/2-1$$

Indices impairs
TFD d'ordre N/2

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $\boldsymbol{N = 2^p} \Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi \frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k) \times W_N^{-kn} \quad n = 0, \ldots, N-1$$

Première décomposition $\Rightarrow N + 2(N/2)^2 = N + N^2/2 < N^2$ opérations (+/x)

$$X(n) = X_1(n) + W_N^{-n} \times X_2(n), \quad n = 0, \ldots, N-1$$

N opérations (+/x)

Indices pairs (TFD d'ordre N/2) $X_1(n) = \sum_{i=0}^{N/2-1} x(2i) W_{N/2}^{-in}, \ i = 0, \ldots, N/2-1$

$X_2(n) = \sum_{i=0}^{N/2-1} x(2i+1) W_{N/2}^{-in}, \ i = 0, \ldots, N/2-1$ Indices impairs TFD d'ordre N/2

Deuxième décomposition $\Rightarrow 2(N/2) + 4(N/4)^2 = N + N^2/4 < N^2$ opérations (+/x)

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n) \qquad X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n)$$

Indices pairs — Indices impairs — Indices pairs — Indices impairs

4 TFDs d'ordre N/4

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $\boldsymbol{N = 2^p}$ $\Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi\frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k) \times W_N^{-kn} \quad n = 0, \ldots, N-1$$

Première décomposition $\Rightarrow N + 2(N/2)^2 = N + N^2/2 < N^2$ opérations (+/x)

$$X(n) = X_1(n) + W_N^{-n} \times X_2(n), \quad n = 0, \ldots, N-1$$

N opérations (+/x)

Indices pairs (TFD d'ordre N/2) $X_1(n) = \sum_{i=0}^{N/2-1} x(2i) W_{N/2}^{-in}, \; i = 0, \ldots, N/2-1$

$X_2(n) = \sum_{i=0}^{N/2-1} x(2i+1) W_{N/2}^{-in}, \; i = 0, \ldots, N/2-1$ Indices impairs TFD d'ordre N/2

Deuxième décomposition $\Rightarrow 2(N/2) + 4(N/4)^2 = N + N^2/4 < N^2$ opérations (+/x)

$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n)$ — Indices pairs, Indices impairs

$X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n)$ — Indices pairs, Indices impairs

4 TFDs d'ordre N/4

⋮

$$p\frac{N}{2}\ TFD\ d'ordre\ 2 \Rightarrow p \times \frac{N}{2} \times 2 = Nlog_2(N)\ opérations\ (+/\times) \ll N^2$$

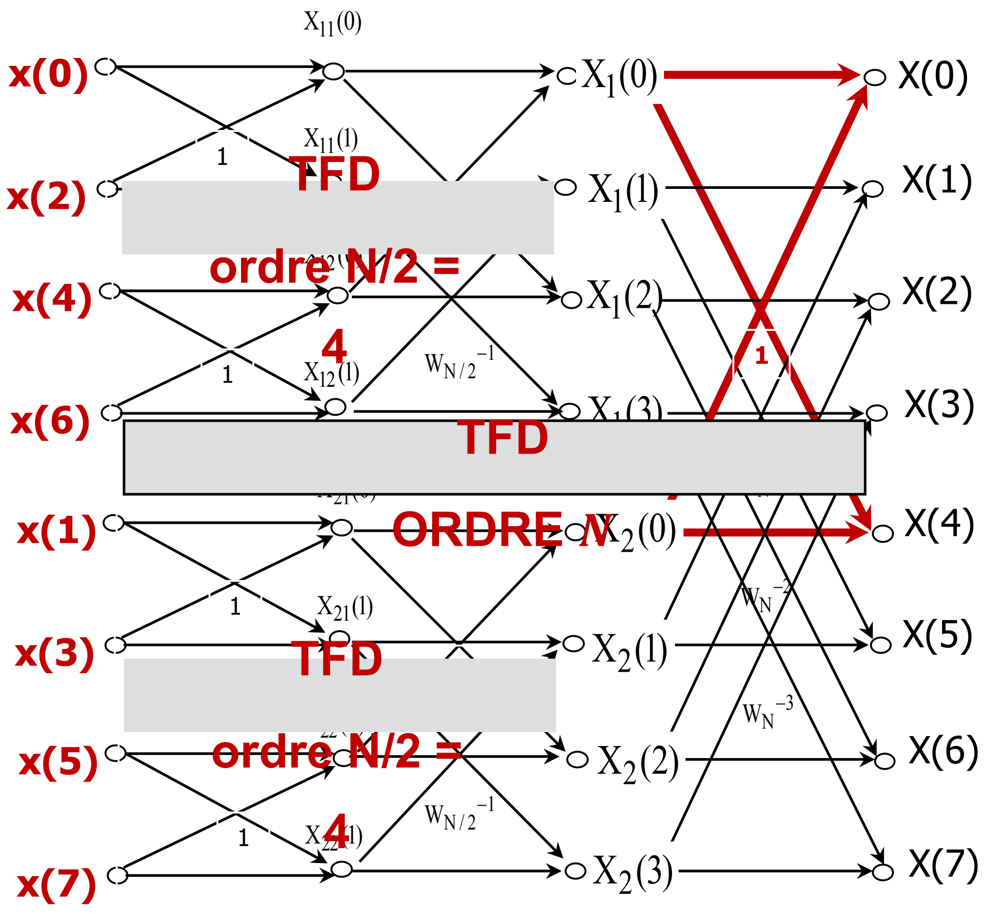
Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, \ldots, N-1$$

$N^2 = 64$ *opérations d'addition/multiplication* $(+/\times)$



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = X_1(n) + W_N^{-n} X_2(n),\ n = 0, \ldots, 7$$

Etape 1

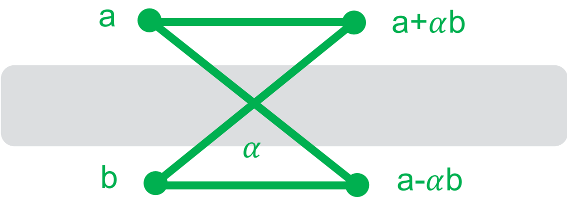
$sur\ x(0), x(2), x(4), x(6)$

Indices pairs
(TFD d’ordre N/2 = 4)

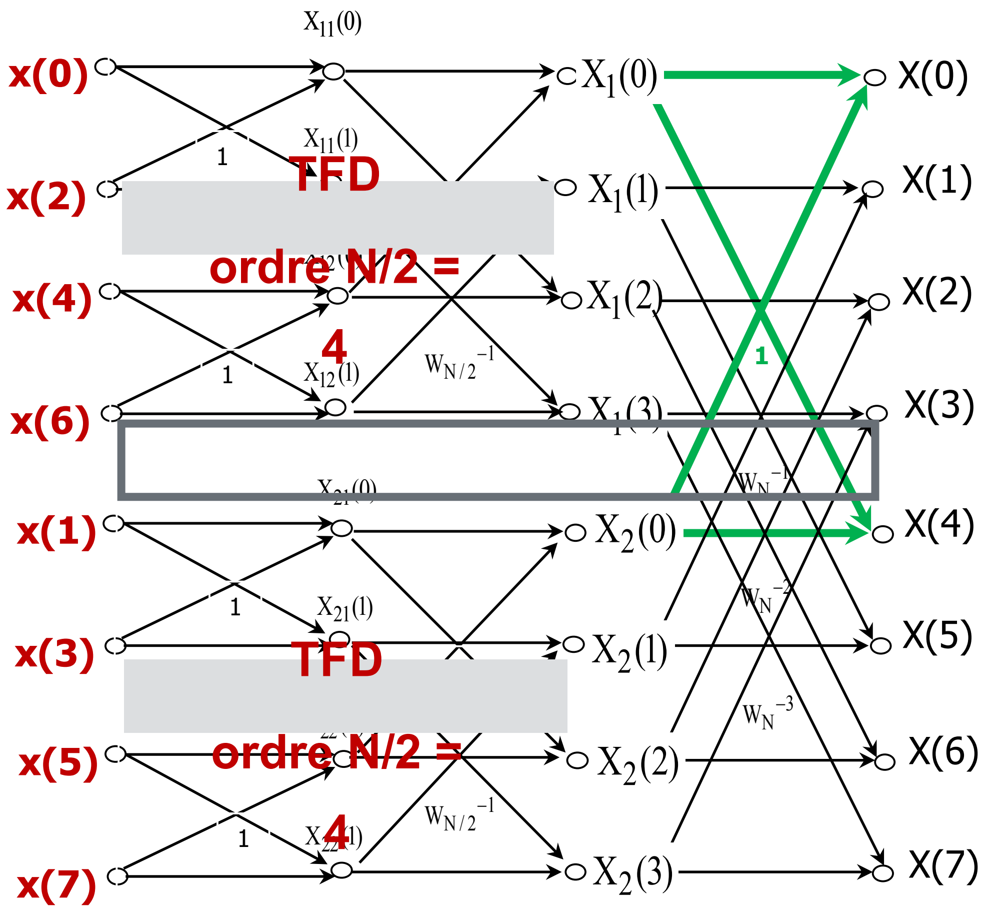
$sur\ x(1), x(3), x(5), x(7)$

Indices impairs
(TFD d’ordre N/2 = 4)

$$2 \times \left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N = 40 < N^2 = 64\ opérations\ (+/\times)$$



Papillon de la FFT
= 2 opérations +/x



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = X_1(n) + W_N^{-n} X_2(n), \;\; n = 0, \dots, 7$$

Etape 1

$sur\ x(0), x(2), x(4), x(6)$

Indices pairs
(TFD d'ordre N/2 = 4)

$sur\ x(1), x(3), x(5), x(7)$

Indices impairs
(TFD d'ordre N/2 = 4)

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n), \;\; n = 0, \dots, 3 \qquad X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n), \;\; n = 0, \dots, 3$$

Etape 2

$sur\ x(0), x(4)$

Indices pairs
(TFD d'ordre N/4=2)

$sur\ x(2), x(6)$

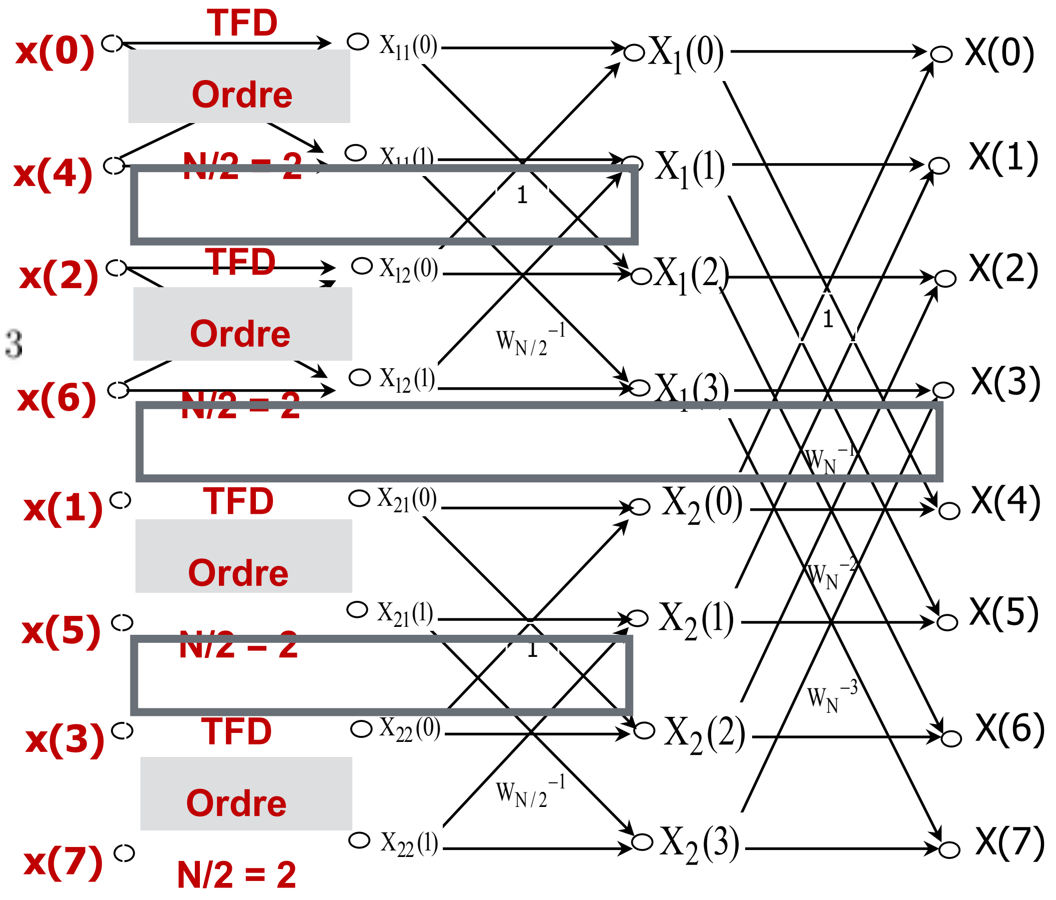
Indices impairs
(TFD d'ordre N/4=2)

$sur\ x(1), x(5)$

Indices pairs
(TFD d'ordre N/4=2)

$sur\ x(3), x(7)$

Indices impairs
(TFD d'ordre N/4=2)



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = X_1(n) + W_N^{-n} X_2(n),\ \ n = 0, \ldots, 7$$

Etape 1

$sur\ x(0), x(2), x(4), x(6)$

Indices pairs
(TFD d'ordre N/2 = 4)

$sur\ x(1), x(3), x(5), x(7)$

Indices impairs
(TFD d'ordre N/2 = 4)

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n),\ \ n = 0, \ldots, 3 \qquad X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n),\ \ n = 0, \ldots, 3$$

Etape 2

$sur\ x(0), x(4)$

Indices pairs
(TFD d'ordre N/4=2)

$sur\ x(2), x(6)$

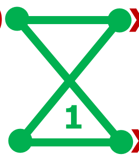
Indices impairs
(TFD d'ordre N/4=2)

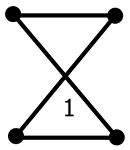
$sur\ x(1), x(5)$

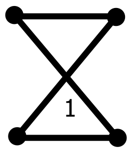
Indices pairs
(TFD d'ordre N/4=2)

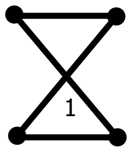
$sur\ x(3), x(7)$

Indices impairs
(TFD d'ordre N/4=2)

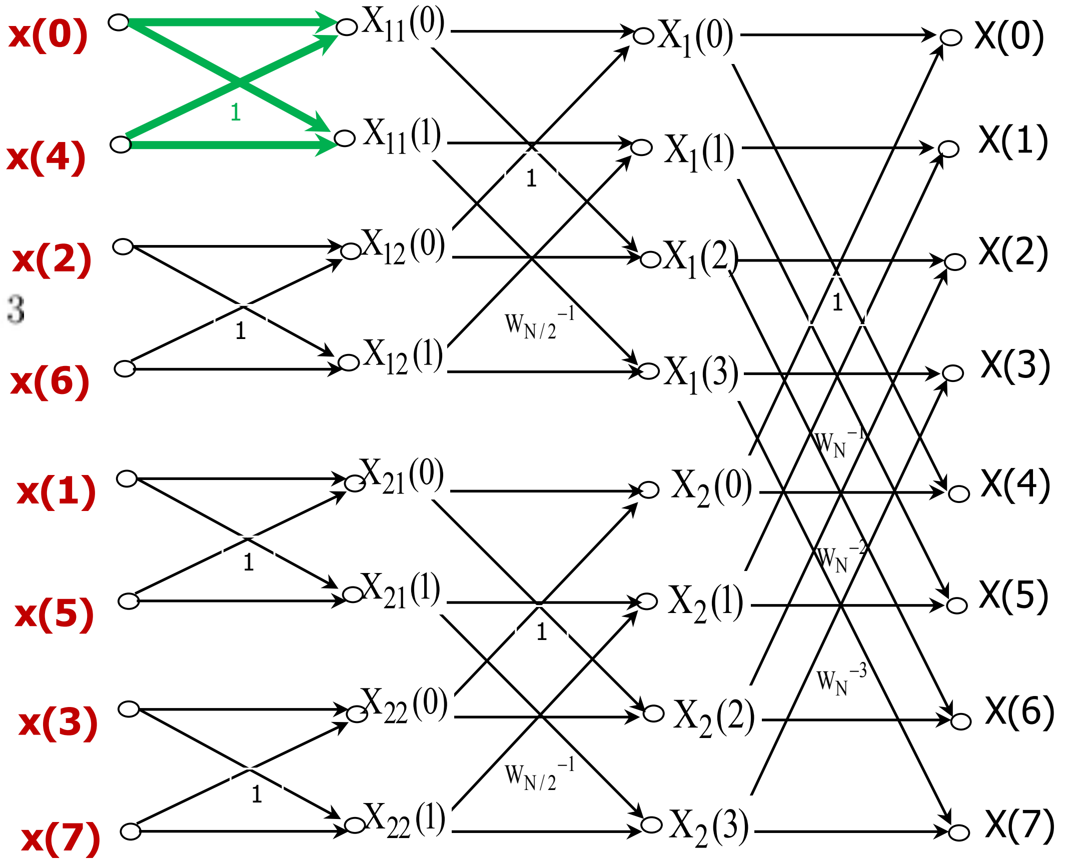








$$4 \times \left(\frac{N}{4}\right)^2 + 2 \times \left(\frac{N}{2}\right) = \frac{N^2}{4} + N = 24 < N^2 = 64\ \boldsymbol{opérations}\ (+/\times)$$



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

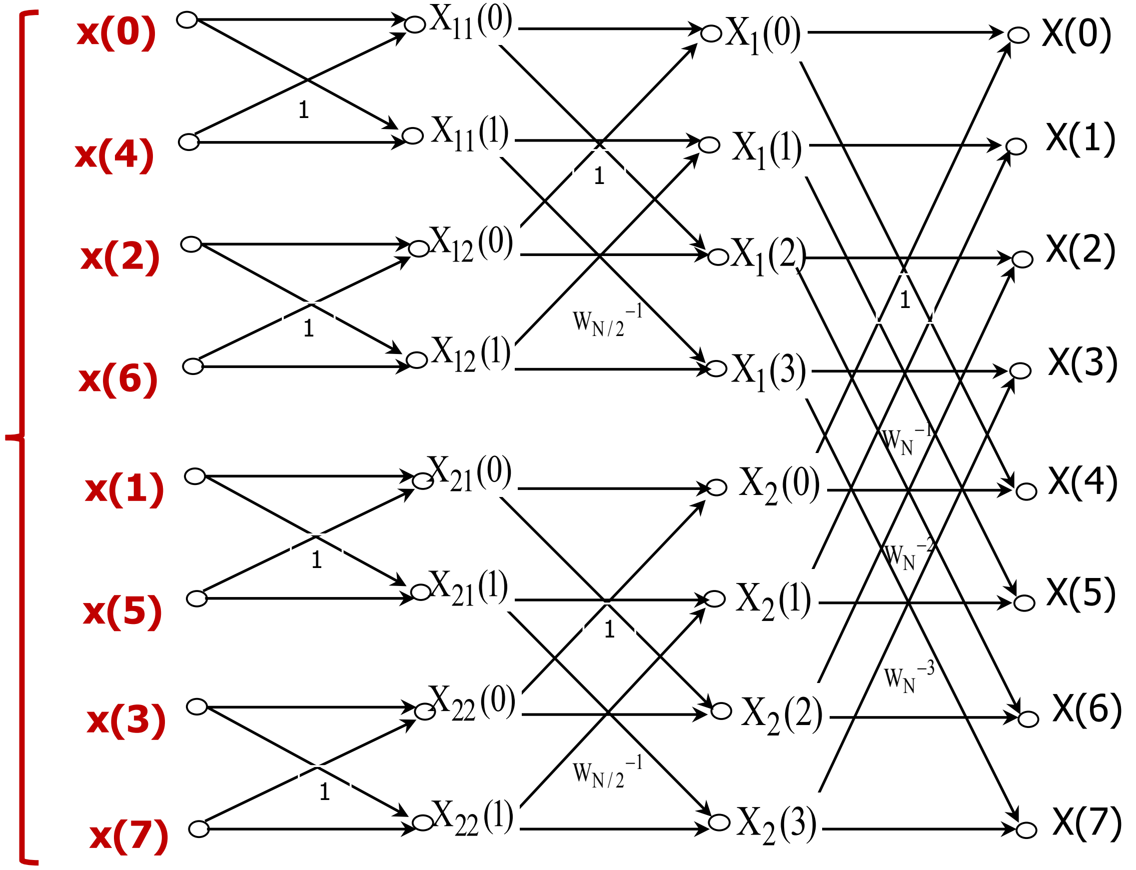
$$p \times \left(\frac{N}{2}\right) \times 2 = 24 < N^2 = 64 \; opérations \; (+/\times)$$

Présentation des données à l'algorithme :

Entrelacement temporel

k	rep. binaire	renv. Bits	nouvel ind.	echantillon
0	"000"	"000"	0	x(0)
1	"001"	"100"	4	x(4)
2	"010"	"010"	2	x(2)
3	"011"	"110"	6	x(6)
4	"100"	"001"	1	x(1)
5	"101"	"101"	5	x(5)
6	"110"	"011"	3	x(3)
7	"111"	"111"	7	x(7)

algorithme de renversement de l'adresse binaire
(« bit reversal »)



Transformée de Fourier Discrète

En résumé

→ Echantillonnage temporel => périodisation spectrale

- Respecter la condition de Shannon
- Attention à la lecture des tracés de la TFD, à la lecture de l'échelle fréquentielle

→ Signal de durée limitée => distorsion de la TFD attendue (analyse spectrale numérique avec un pouvoir séparateur limité, apparition d'ondulations)

- Utiliser plusieurs fenêtres de pondération du signal

  => différents pouvoirs séparateurs, différents taux d'ondulation pour l'analyse spectrale numérique

→ Echantillonnage spectral

- Attention à la mauvaise visualisation de la TFD (résolution insuffisante)

  => nécessité d'interpoler (méthode du zero padding)

- TFD et TFD^{-1} transforment un produit en produit de convolution circulaire

  => si besoin, convolution linéaire = convolution circulaire en prolongeant les signaux par des zéros

→ Algorithme de calcul rapide : FFT = Fast Fourier Transform

- Condition nombre de points de signal $N=2^p$ => décomposition en sous suites entrelacées