
ANNEXE A

Programmes de traitement de données

Je détaille ici le principe de des programmes de traitement de donnée. Les logiciels que nous utilisons sont *MATLAB R2009a* et *Wolfram Mathematica 9*. Je ne donne ici que le principe de fonctionnement sous la forme de pseudo code, mais les programmes complets sont disponibles à l'adresse :

<http://www.quantumengineering-tlse.org>.

J'utilise un code couleur pour que la lecture soit plus simple :

- En **vert** : des commentaires permettant d'apporter certains détails.
- En **bleu** : le nom des fonctions utilisées.
- En **rouge** : les variables d'itération dans les boucles **for**
- Les mots en *italique* sont les variables utilisées dans le programme.

A.1 Programme 1 : mise en forme des données et traitement du fond thermique

Le premier programme présenté ici nous permet de mettre en forme les données brutes de l'expérience. Pour cela, nous créons un *film* de l'expérience en créant une image globale contenant chacune des images prises les unes après les autres. Nous étudions majoritairement la physique 1D dans nos expériences, nous avons donc décidé d'intégrer chacune des images selon la direction verticale pour ne garder que la dynamique selon l'axe du réseau optique. Nous obtenons alors l'évolution de la distribution en impulsion au cours de l'expérience, que ce soit en fonction de la durée du pulse du réseau (expérience de diffraction ??), d'un temps de maintien dans le réseau (expérience d'interférence entre paquets d'onde cohérents??) ou de l'angle de déphasage initial du réseau (étude de l'oscillation de paquets d'atomes en fonction du déphasage??). Ce programme est donc commun à toutes expériences et reste très général. Le traitement spécifique à chaque problème physique étudié est ensuite détaillé dans les autres programmes de traitement.

Programme 1 MATLAB. Mise en formes des données et traitement du fond thermique

%% Définition des paramètres %%

% Définition des paramètres propres à l'expérience traitée :

imDir : Chemin d'accès aux images d'absorption

imNom : Nom des images

Ntemps : nombre de temps total à traiter

Dtemps : pas de temps

% RQ : ce ne sont pas forcément des temps, il peut s'agir de n'importe quel paramètre incrémenté à chaque nouvelle image

→ *Tgrid* : vecteur allant de 0 à t_{max} par pas de *Dtemps*

% RQ : si le pas de temps n'est pas constant, il faut créer la base temporelle *Tgrid* à la main

Nmoy : nombre d'images par temps

% Nous faisons une image moyenne sur *Nmoy* images

Angle : Angle de rotation de chaque image

%% Création du cadre %%

Appel de la fonction [Calc_OD](#) pour la *Ncadre*-ème image choisie par l'utilisateur

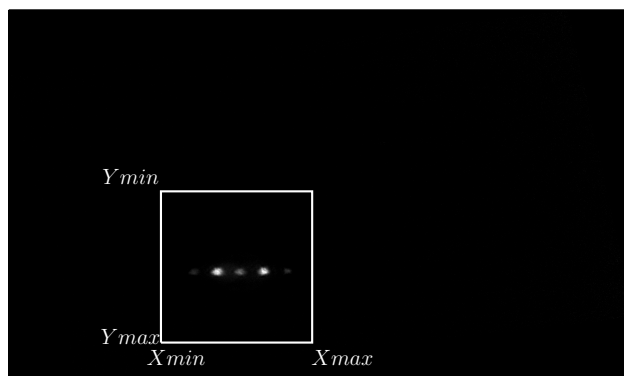
% La fonction *Calc_OD* calcule la densité optique tel qu'expliqué dans la section ??

Rotation de l'image de *Angle* ([imrotate](#))

Affichage de l'image ([imagesc](#))

__ PAUSE __

Définition à la main du cadre de l'image approprié (sur la console MATLAB)



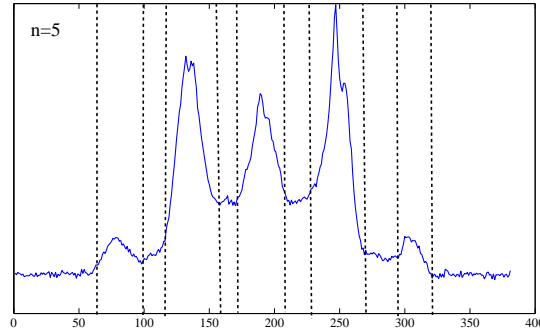
__ REPRISE __

Intégration verticale de la N_{cadre} ème image
Affichage de l'image

_ PAUSE _

Définition à la main du $Cadre$ autour de chaque paquet d'atome (sur la console MATLAB)

$\%Cadre$ servira par exemple à exclure le condensat de l'image pour déterminer le fond thermique



_ REPRISE _

$\%\%$ Boucle sur le temps $\%\%$

```

for  $j$  allant de 0 à  $t_{max}$  do
  for  $i$  allant de 1 à  $N_{moy}$  do
    Appel de l'image  $i$  du temps  $j$  via la fonction  $Calc\_OD$  → variable  $OptDens$ 
    if  $i = 0$  then
      Création de l'image :  $E0 = OptDens$ 
    else
      Moyennage :  $E0 = E0 + OptDens$ 
    end if
  end for
  Rotation de l'image moyenne de  $Angle$  ( $imrotate$ )
  Recadrage
  Concaténation verticale de l'image moyenne  $j$  avec les images précédentes
  ( $vertcat$ ) →  $globalplot$ 

```

$\%\%$ 1D : Intégration $\%\%$

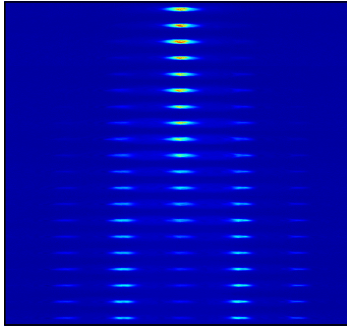
Intégration verticale de l'image moyenne j (sum) et normalisation à 1
 $\%$ cette étape ramène l'image à un vecteur et nous permet d'étudier la physique 1D du problème
 Concaténation verticale de l'image intégrée j avec les images précédentes
 ($vertcat$) → $globalplotinteg$

$\%\%$ 2D : Redimensionnement $\%\%$

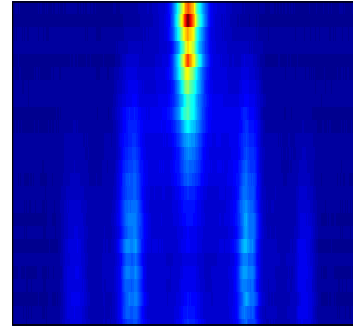
Redimensionnement de l'image moyennée j en un vecteur selon la méthode expliquée dans [1] pour être éventuellement traité plus tard par PCA
 end for

Résultats

Exemple de résultat : globalplot



Exemple de résultat : globalplotinteg



%% Détermination du fond thermique %%

for j allant de 0 à t_{max} **do**

Détermination du nombre de pics (`findpeaks`) pour la ligne j du film intégré `globalplotinteg`

L'initialisation de la largeur de la gaussienne pour l'ajustement du fond thermique est déterminée à partir du nombre de pics

if $j = 0$ **then**

Détermination de l'amplitude du pic central pour initialiser l'amplitude typique du fond thermique

% on initialise l'amplitude du fond thermique avec l'amplitude du pic central divisé par 10. On ne le fait que pour la première image car, au cours des expériences, le pic central peut s'annuler

end if

Détermination du centre de la gaussienne par la position du pic central

Définition des données correspondant au fond thermique en retirant tous les points compris dans `Cadre`

Définition des paramètres d'ajustement (`fittype`)

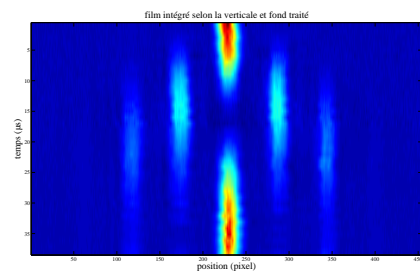
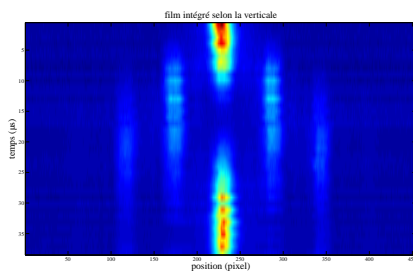
Ajustement du fond thermique par une gaussienne (`fit`)

% l'algorithme d'ajustement utilisé est l'algorithme Levenberg-Marquardt

Création des données nettoyées $\rightarrow DataClean(i) = globalplotinteg(i) - Fond$

end for

Affichage et sauvegarde des données



A.2 Programme 2 et 3 : traitement de l'expérience de diffraction

A.2.i Programme 2 : diffraction dans le régime de Raman-Nath

Ce programme est conçu pour traiter les données obtenues dans le cadre de l'expérience de diffraction d'un condensat de Bose-Einstein sur un réseau optique dans l'approximation Raman-Nath. Le protocole mis en place pour réaliser cette expérience est présenté dans la section ???. Dans cette approximation, la population dans l'ordre de diffraction n est donnée par l'expression :

$$P_n(t_{pulse}) \propto \left| J_{|n|}(V_0 t_{pulse}/2\hbar) \right|^2, \quad (\text{A.1})$$

où $J_n(x)$ est la fonction de Bessel d'ordre n , V_0 la profondeur du potentiel périodique et t_{pulse} le temps d'interaction entre le condensat et le réseau.

Le programme de traitement réalisé permet d'obtenir en sortie les profondeurs calculées à partir de chacun des ordres de diffraction et d'avoir ainsi la profondeur moyenne obtenue ainsi que l'incertitude sur cette profondeur. Nous utilisons pour cela les données déterminées dans le **programme 1** : *Cadre* et *Dataclean*

Programme 2 MATLAB. Traitement de l'expérience de diffraction dans l'approximation Raman-Nath

%%	Définition des paramètres	%%
----	---------------------------	----

Dataclean : données intégrées selon l'axe vertical et dont le fond thermique a été retiré à l'aide du **programme 1**

Cadre et n : Cadrage de chaque ordre de diffraction défini dans le **programme 1** et nombre total d'ordres de diffraction

$nmax = (n - 1)/2$: numéro de l'ordre de diffraction maximal visible

% Si $n = 5$, les ordres de diffraction visibles sont les ordres $-2, -1, 0, 1$ et 2 .
On a donc $nmax = 2$

estimU0 : estimation de la profondeur du potentiel périodique pour l'initialisation des fits

for m allant de 1 à n **do**

integordrenorm : matrice dont la colonne m représente l'évolution de la population dans l'ordre m . Ce vecteur est renormalisé de sorte que sa valeur maximale soit égale à 1

end for

%%	Traitement des données	%%
----	------------------------	----

for m allant de 0 à $n - 1$ **do**

ordre = $m - nmax$: numéro de l'ordre de diffraction considéré

fit de l'ordre de diffraction correspondant par la fonction de Bessel adaptée :

$AJ_{|ordre|}^2(Bx)$

% l'algorithme utilisé pour l'ajustement est l'algorithme Levenberg-Marquardt. Nous avons deux paramètres d'ajustement A et B . Le fond est fixé à 0 car il a été traité dans le **programme 1**

Chaque paramètre de l'ajustement ainsi que les intervalles de confiance associés sont enregistrés dans la variable *paramfit*

La profondeur obtenue grâce à l'ordre de diffraction m est enregistrée dans la variable *U0*

end for

Affichage des données expérimentales et des ajustements obtenus ainsi que des profondeurs déduites et sauvegarde des données

A.2.ii Programme 3 : Diffraction Kapitza-Dirac

La théorie développée dans la section ?? permet de traiter les mêmes données que précédemment sans l'approximation Raman-Nath. Pour cela nous avons 2 étapes :

- **Étape 1** : calcul de l'évolution théorique de la population dans chaque ordre de diffraction pour différentes profondeurs

$$P_n(q, t_{pulse}) = \left| \int_0^{2\pi} \frac{1}{\sqrt{2\pi}} \psi(\tilde{x}, t_{pulse}) e^{-2ni\tilde{x}} d\tilde{x} \right|^2, \quad (A.2)$$

$$\text{avec } \psi(\tilde{x}, t_{pulse}) = e^{-iq \frac{E_L t_{pulse}}{2\hbar}} \sum_{j=0}^{j_{max}} \frac{1}{\sqrt{\pi}} c_{2j} M(a_{2j}, q, \tilde{x}) e^{-ia_{2j} \frac{E_L t_{pulse}}{4\hbar}},$$

où $M(a, q, x)$ sont les fonctions de Mathieu et q est la profondeur renormalisée du réseau. Un programme Mathematica effectue ce calcul. Ce programme très simple n'est pas automatisé et chaque changement de profondeur et/ou d'ordre de diffraction nécessite de rentrer à la main les nouveaux paramètres. Les données que j'ai enregistré durant ma thèse (au format .txt) sont les suivantes :

- $q \in [0, 7]$ par pas de 0.2 : $n \in [0, 2]$ et $t \in [0, 70\mu s]$ par pas de $5\mu s$
- $q \in [7, 9]$ par pas de 0.2 : $n \in [0, 3]$ et $t \in [0, 70\mu s]$ par pas de $5\mu s$
- $q \in [9, 15]$ par pas de 0.2 : $n \in [0, 4]$ et $t \in [0, 20\mu s]$ par pas de $1\mu s$

Je ne détaille pas le fonctionnement de ce programme. Notons simplement que les fonctions Mathematica utilisées sont : [MathieuCharacteristicA\[j, q\]](#) pour évaluer le paramètre a_{2j} associé à l'indice j et à la profondeur q et [MathieuC\[a_{2j}, q, x\]](#) pour évaluer la valeur de la fonction de Mathieu symétrique correspondante.

- **Étape 2** : Détermination de la profondeur la plus en accord avec nos données expérimentales et donc de la profondeur V_0 associée. Cela est réalisé par le **programme 3** (Matlab). Il utilise notamment les valeurs expérimentales des populations dans chaque ordre de diffraction *integordrenorm* calculées dans le **programme 2**.

Programme 3 MATLAB. Détermination de la profondeur du réseau par diffraction kapitza-dirac

```
%%                               Définition des paramètres                               %%
```

integordrenorm : matrice regroupant l'évolution temporelle de chaque ordre de diffraction déterminée par le **programme 2** (données expérimentales)

% La structure de la matrice *integordrenorm* est la suivante :

ordre	$-n_{max}$	\dots	0	\dots	$+n_{max}$
$t_{pulse} = 0$	$P_{-n_{max}}(0)$	\dots	$P_0(0)$	\dots	$P_{n_{max}}(0)$
$t_{pulse} = D_{temps}$	$P_{-n_{max}}(D_{temps})$	\dots	$P_0(D_{temps})$	\dots	$P_{n_{max}}(D_{temps})$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$t_{pulse} = t_{max}$	$P_{-n_{max}}(t_{max})$	\dots	$P_0(t_{max})$	\dots	$P_{n_{max}}(t_{max})$

n et N_{temps} : Nombre d'ordres de diffraction total et nombre de temps de maintien déduits de la taille de la matrice *integordrenorm*

$n_{max} = (n - 1)/2$: numéro de l'ordre de diffraction maximal visible

s_{min} et s_{max} : borne de l'intervalle de profondeur que l'on souhaite explorer

N_{sm} : nombre de valeur de s entière différentes

Nsm ne prend pas en compte le pas de 0.2, ce ne sont que les valeurs entières

KapDir : Chemin d'accès aux données calculées par le programme Mathematica

```
%% Chargement des données calculées par le programme Mathematica %%
```

```
for j allant de 0 à nmax do
```

```
  for i allant de 1 à Nsm do
```

```
    for l allant de 1 à 6 do
```

```
      il y a 6 valeurs de profondeur par valeur de i : i.0, i.2, i.4, i.5, i.6 et i.8
```

```
      Chargement des données calculées par le programme Mathematica cor-  
      respondant à la profondeur i.l et à l'ordre de diffraction j (importdata)
```

```
      → création de la matrice datakap
```

```
      Renormalisation de ces données de telle sorte que la valeur maximale  
      soit égale à 1
```

```
      % Comme pour les données integordrenorm
```

```
    end for
```

```
  end for
```

```
end for
```

Sélection des données calculées avec le programme Mathématique et correspondant à un point expérimental → création de *datakapres*

% on sélectionne pour cela uniquement les temps t_{pulse} effectivement utilisés dans l'expérience. La structure de cette matrice est la suivante :

ordre	0	...	+ n_{max}
profondeur	s_{min}	...	s_{max}
$t_{pulse} = 0$	$P_0^{s_{min}}(0)$...	$P_0^{s_{max}}(0)$
⋮	⋮	⋮	⋮
$t_{pulse} = t_{max}$	$P_0^{s_{min}}(t_{max})$...	$P_0^{s_{max}}(t_{max})$

%% Calcul du résidu %%

Pour chaque ordre de diffraction, chaque temps, et chaque profondeur, on calcule le résidu des populations $r = (P^{exp} - P^{théo})^2$
 Pour chaque ordre de diffraction et chaque profondeur, on en déduit le *Mean Squared Error* ou $MSE = \sum_t r$

%% Comparaison des résidus %%

Pour chaque ordre de diffraction, on identifie quelle profondeur donne le MSE minimal ([min](#))

La moyenne des profondeurs obtenues sur l'ensemble des ordres de diffraction est sélectionnée comme la profondeur optimale et la dispersion de résultats fixe l'incertitude sur cette profondeur optimale ([mean](#) et [var](#)).

A.3 Programme 4 : Traitement de l'expérience d'oscillation

Ce programme permet d'extraire l'amplitude d'oscillation x_{max} dans le cadre de l'expérience d'oscillation présentée dans la section ???. Il permet aussi d'en déduire la vitesse maximale acquise par les atomes au cours de leur mouvement et la profondeur du réseau dans lequel ils oscillent (pour un déphasage initial de 45°). Pour cela, Le **programme 4** réutilise les résultats du **programme 1** suivants : `'imNom_DataIntegClean.txt'`, `'imNom_X.txt'`, `'imNom_T.txt'`.

Le programme se charge alors d'ajuster chaque image de l'expérience. Les ordres d'interférences sont approximés par des gaussiennes et l'enveloppe est elle approximée par 1 ou 2 gaussiennes. En effet, si nous n'utilisons qu'une seule gaussienne, cela ne nous permet pas de traiter les cas où il y a séparation des paquets d'atomes en deux par effet tunnel. De plus, il reste toujours quelques atomes dans les classes de vitesses opposées à celle qui décrit simplement l'oscillation, ce qui conduit systématiquement à sous-estimer l'amplitude de l'oscillation si nous nous restreignons à 1 gaussienne.

Il faut donc distinguer deux types d'images :

- Les images pour lesquelles les atomes ont une vitesse moyenne nulle. la distribution de vitesse est alors centrée sur l'ordre d'interférence 0 et il ne faut utiliser qu'une seule gaussienne.
- Les images pour lesquelles les atomes ont une vitesse non nulle (qu'il y ai eu séparation par effet tunnel ou non) pour lesquelles nous utilisons 2 gaussiennes : une pour les vitesses négatives et une pour les vitesses positives.

Afin d'avoir un critère simple et efficace pour distinguer les deux cas, nous calculons en amont le barycentre de la distribution de vitesses négatives et le barycentre de la distribution de vitesses positives. Lorsque l'image est centrée, ces deux barycentres sont proches. Tandis que dans le second cas, la différence des deux barycentres est plus grande. La quantité que nous avons retenue est la suivante :

$$varbar = \left(\frac{2(Bar_g - Bar_d)}{Bar_g + Bar_d} \right)^2, \quad (A.3)$$

où g et d désignent respectivement la gauche et la droite de l'image. Ce critère simple permet de trouver une valeur pertinente pour le critère de basculement entre la description à 1 gaussienne et celle à 2 gaussiennes. Le seuil de basculement est choisi manuellement.

Il faut néanmoins noter que cela reste un ajustement avec une dizaine de variables, sur des images parfois relativement bruitées. Il est donc nécessaire d'ajouter un critère de validation. Au final, nous ne retenons que les points pour lesquels l'ajustement a convergé avec un coefficient de détermination R^2 supérieur à 0.9 (R^2 est déterminé lors de l'ajustement par MATLAB).

Programme 4 MATLAB. Traitement de l'expérience d'oscillation d'un réseau de condensat. Détermination de l'amplitude d'oscillation

```
%%                               Définition des paramètres                               %%
```

ImNom : nom des images avant le '..._0000i_Back/NoAt/With', où '...' est le temps de maintien des atomes dans le réseau décalé

Le Programme charge alors les données *DataClean*, *Xgrid* et *Tgrid* qui sont respectivement le film de l'expérience intégré verticalement et dont le fond thermique a été retiré, la base spatiale horizontale et la base temporelle calculées par le **programme 1**

PixelSize : taille réelle d'un pixel

```
%%                               Définition du cadre                               %%
```

Affichage du film intégré de l'image

__ PAUSE __

Définition à la main de la limite gauche de l'image *G*, du centre de l'ordre d'interférence 0 *M* et de la limite droite *D* (sur la console MATLAB)

% Il faut choisir *G* et *D* de façon à ne pas sur-évaluer l'importance du fond. Il faut donc rester proche des ordres d'interférence non nuls

__ REPRISE __

```
%%  Détermination du critère de basculement entre 1 et 2 gaussiennes  %%
```

for *i* allant de 1 à *Ntemps* **do**

Barg(i) : barycentre coté gauche de l'image

Bard(i) : barycentre coté droit de l'image

bartot(i) : barycentre total

$varbar(i) = \left(\frac{2(Barg(i) - Bard(i))}{Barg(i) + Bard(i)} \right)^2$

 % *varbar* est ce qui va nous servir de critère de basculement. Si *varbar* est faible, l'image est centrée sur l'ordre 0 → une seule gaussienne sera nécessaire.

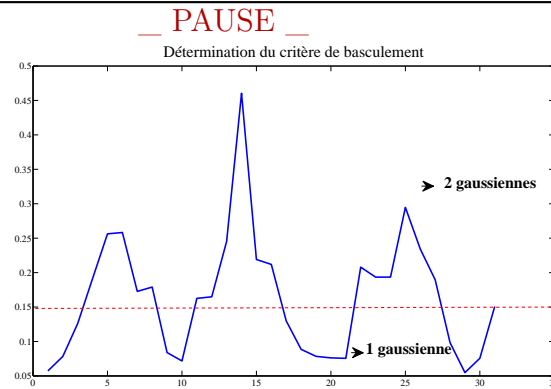
 Si *varbar* est grand, les atomes n'ont pas une vitesse nulle → il faudra 2 gaussiennes pour bien rendre compte de la dynamique

end for

Affichage de *varbar* en fonction du temps

Choix à la main du critère de basculement *Bascul* (sur la console MATLAB)

% La courbe affichée doit présenter la même périodicité que l'oscillation. Il faut choisir une valeur intermédiaire, comme représenté dans l'exemple ci-dessous



__ REPRISE __

%% Traitement des données %%

for *i* allant de 1 à *Ntemps* **do**

Détermination du nombre d'ordre d'interférence *n* à ajuster et de leur position ([findpeaks](#))

if *varbar(i)* > *Bascul* **then**

%% Il faut 2 gaussiennes %%

Construction de la fonction qui servira à ajuster l'image *i* :

$$\left(A_g e^{-(p-B_g)^2/C_g} + d e^{-(p-B_d)^2/C_d} \right) \times \left(e^{-(p-B_1)^2/D} + \dots + e^{-(p-B_n)^2/D} \right)$$

Initialisation des variables d'ajustement

% Les amplitudes sont initialisées empiriquement à 0.01, les centres B_1, \dots, B_n sont initialisés avec les positions des ordres déterminées précédemment. Le centre B_g (resp. B_d) est initialisé avec le barycentre de gauche (resp. de droite). Les largeurs des gaussiennes sont initialisées de façon empirique

Ajustement des données ([fit](#))

if L'ajustement a convergé et $R^2 > 0.9$ **then**

Enregistrement des résultats de l'ajustement

else

Le point n'est pas pris en compte

end if

else

%% Il faut 1 gaussienne %%

Construction de la fonction qui servira à ajuster l'image *i* :

$$A e^{-(p-B)^2/C} \times \left(e^{-(p-B_1)^2/D} + \dots + e^{-(p-B_n)^2/D} \right)$$

Initialisation des variables d'ajustement

Ajustement des données ([fit](#))

if L'ajustement a convergé et $R^2 > 0.9$ **then**

Enregistrement des résultats de l'ajustement

else

Le point n'est pas pris en compte

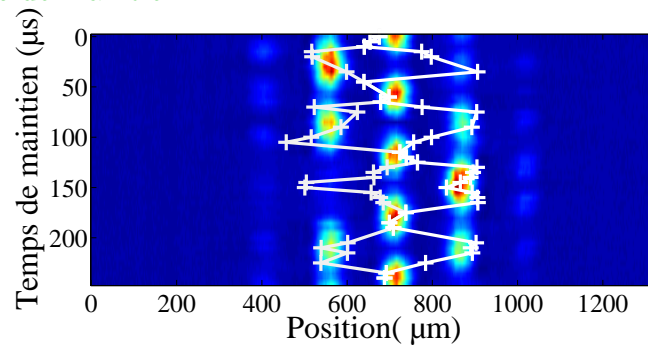
end if

end if

end for

Affichage des résultats des ajustements

% Affichage, sur l'image initiale, du centre des gaussiennes de l'enveloppe en fonction du temps de maintien



%%

Ajustement de l'oscillation

%%

Ajustement par une sinusoïde ([fit](#)) des trois variables suivantes :

posd : centre de la gaussienne d'enveloppe de droite

posg : centre de la gaussienne d'enveloppe de gauche

posmax : centre de la gaussienne associée à l'amplitude la plus grande

% Il faut initialiser chacun des ajustements de façon appropriée. Suivant les données, il faudra choisir l'ajustement le plus pertinent (cela dépend de la profondeur, d'une éventuelle séparation des paquets d'atomes en deux par effet tunnel etc ...)

Affichage des résultats et calcul de *vmax*, et *s* déterminés par les ajustements ainsi que les barres d'erreurs associées

Bibliographie

- [1] R. DUBESSY, C. DE ROSSI, T. BADR, L. LONGCHAMBON, ET H. PER-
RIN, “Imaging the collective excitations of an ultracold gas using statistical
correlations”, *New Journal of Physics* **16**, 122001 (2014).