

## TP 7 : Mesure de la fréquence d'un pendule

Le but de ce TP est d'apprendre à gérer un film expérimental et d'en extraire des données. Ici on étudie le déplacement d'un pendule simple au cours du temps, on comparera les données expérimentales avec une théorie simple.

### Introduction : stockage d'un film

Vous avez vu durant le TP 5 que les images sont stockées sous la forme de matrices contenant la couleur (ou nuance de gris) de chaque pixel contenu dans l'image. Un film étant une succession d'images dans le temps, il peut être stocké au moyen d'une matrice  $M(i,j,k)$  à 3 indices. Les 2 premiers indices  $(i,j)$  font référence à la position du pixel, comme dans une image classique. La 3<sup>ème</sup> dimension  $(k)$  correspond à l'indice de temps (et non à sa valeur). La 3<sup>ème</sup> image du film (voir Fig 1) peut ainsi être appelée par la commande :

```
>> M(:, :, 3)
```

Pour le tableau  $M(i,j,k)$  on a donc

- $i$  l'indice de la ligne du pixel,
- $j$  l'indice de la colonne du pixel,
- $k$  l'indice du temps du pixel,

### Rappel : manipulation de tableau

On rappelle que dans une matrice  $M$  à 2 indices, on peut sélectionner l'ensemble de la  $i^{\text{ème}}$  ligne par la commande :

```
>> M(i, :);
```

1. A l'aide de la fonction `randi` créez une matrice  $M$  d'entrée aléatoire allant de 0 à 255 de taille  $10 \times 12$ . Puis, sans boucle, sélectionner tous les éléments de  $M$  situés à l'intersection des colonnes paires et des lignes paires.

Les fonctions `min` et `max` servent à trouver respectivement le minimum et la maximum d'un tableau.

2. Donner la valeur maximale du tableau  $M$

3. En vous aidant de la documentation de Matlab sur la fonction `max`, donner le maximum de chaque colonne de  $M$  ainsi que la position de ces maximums dans la colonne.

4. Remplacer les valeurs maximales trouvées précédemment par 999.

## Exercice 1 : Oscillation du pendule

### Objectif

À partir des deux films (pendule court et pendule long), nous allons retrouver la relation existant entre la fréquence d'oscillation  $T$  d'un pendule et sa longueur  $L$ , et la comparer avec la prédiction théorique pour de faibles oscillations

$$T = 2\pi\sqrt{L/g}$$

où  $g$  est l'accélération de la pesanteur. Ce qui suit doit donc être réalisé sur les deux films de pendule (court et long). La théorie veut que l'oscillation obtenue soit proche d'une oscillation sinusoïdale.

$$x(t) = \beta + \alpha \sin\left(\frac{2\pi t}{T} + \phi\right),$$

où  $\beta$  correspond à la valeur moyenne autour de laquelle le sinus oscille,  $\alpha$  correspond à l'amplitude de l'oscillation,  $T$  la période d'oscillation, et  $\phi$  est un déphasage. À partir du film, il faut donc obtenir l'évolution  $x(t)$  puis mesurer la période  $T$ .

### Visualisation du film

Vous disposez de deux films (que vous pourrez prévisualiser en dehors de Matlab ...) avec une fréquence d'échantillonnage de 100 images par seconde. Pour vous faciliter la tâche, une petite fonction est fournie pour convertir les deux films en tableaux, qui sont sauvegardés dans des fichiers `.mat`. Cette fonction s'utilise de la manière suivante

```
>> avi2mat('pendule_court');
```

La fonction ouvre le film, génère le tableau `M`, puis enregistre dans un `.mat`. Avec l'instruction

```
>> load('pendule_court.mat')
```

on charge le tableau `M`. Attention, son exécution demande un peu de temps.

1. Utiliser la fonction `avi2mat` puis charger le fichier `pendule_court.mat`.
2. Écrire avec une boucle, les instructions qui réalisent l'animation du film en niveau de gris, toutes les 10 images. On "pixélisera" l'image en n'affichant que tous les 4 pixels sur les lignes et les colonnes. On utilisera la commande `axis image` pour que l'image ait le bon rapport d'aspect et on règle la `colormap`. Vous afficherez comme titre le numéro de l'image. Pour l'animation : utiliser la command `pause` ou encore `drawnow`.
3. Repérer une image sur laquelle le fil est bien vertical. Noter le numéro de cette image.

### Mesure de la longueur $L$ du fil

4. Sur une figure, afficher uniquement une image (avec tous les pixels) pour laquelle le fil est bien vertical.
5. Déterminer avec `datacursor` la longueur du pendule en pixels puis convertir en mètres. Il faut bien prendre une longueur de pendule allant de l'axe de rotation au centre de masse.

## Mesure de la période $T$ du pendule

6. Copiez la ligne `M(150,150:400,:)` dans un tableau que vous appellerez `Tab`. Quelle est sa dimension ?

7. Afficher le film de `Tab` image par image en "binarisant" les niveaux de gris. Choisir un seuil pour ne voir que le fil en noir, le reste de l'image en blanc. Prendre par exemple `seuil=80`.

On souhaite à présent tracer l'évolution spatio-temporelle du déplacement de la corde, c'est-à-dire l'amplitude du déplacement en fonction du temps (voir Fig 2). Ainsi, on peut utiliser le tableau précédent correspondant à l'évolution des pixels dans une ligne.

8. Créer un tableau `Tab2=squeeze(Tab)`. Que s'est-il passé avec cette commande et expliquer les dimensions de `Tab2` ?

9. D'après la question précédente, représenter alors *l'image* de l'évolution spatio-temporelle avec `imagesc`. Les abscisses  $t$  doivent être en *secondes* et les ordonnées  $x$  en *mètres*<sup>1</sup>.

Le diagramme spatio-temporel que nous avons obtenu est encore une image (une matrice avec les tons de gris) et non la valeur de la position de la corde du pendule rangée dans vecteur.

10. Créer un tableau `osc1=Tab2(:,1)`. Tracer `osc1` en fonction de  $x$ . Que représente cette courbe ?

11. À l'aide de la commande `find` chercher les indices des pixels noirs de `osc1` et en déduire la coordonnée  $x$  du pixel noir du milieu.

12. Avec une boucle, répéter cette opération pour chaque image et construire un vecteur `osc` qui a autant d'éléments que d'images avec la coordonnée  $x$  du pixel central de la corde à chaque instant. Représenter alors sur un graphique l'évolution de `osc` en fonction du temps.

13. Estimer alors les valeurs optimales  $\alpha$ ,  $\beta$ ,  $T$  et  $\phi$  entre le modèle et les mesures.

## Comparaison avec la prédiction théorique

14. Une fois  $T$  et  $L$  déterminés pour les deux pendules, tracer le graphique de la variation théorique de  $T$  en fonction de  $L$  (avec  $g = 9,81 \text{ m} \cdot \text{s}^{-2}$ ). Superposer à ce graphique, les points  $(L_1, T_1)$  et  $(L_2, T_2)$  qui correspondent à nos deux mesures expérimentales.

15. La théorie marche-t-elle ? Combien de pourcents d'écart trouvez-vous sur la période ?

---

1. construire les bons vecteurs temps et position, lire l'aide de `imagesc`

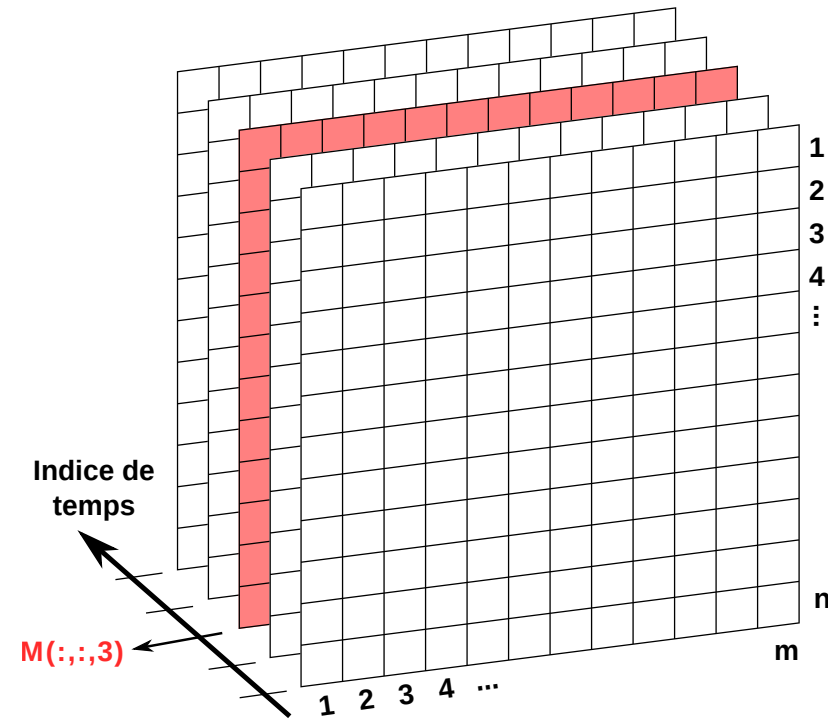


FIGURE 1 – Exemple de stockage d'un film sous Matlab, est stockée dans une matrice  $M$  de dimension  $m, n, k$  où  $m$  est le nombre de pixels dans une ligne d'une image,  $n$  est le nombre de pixels dans une colonne d'une image et  $k$  le nombre d'images totale du film.  $M(:, :, 3)$  permet donc d'appeler la 3<sup>ème</sup> image du film.

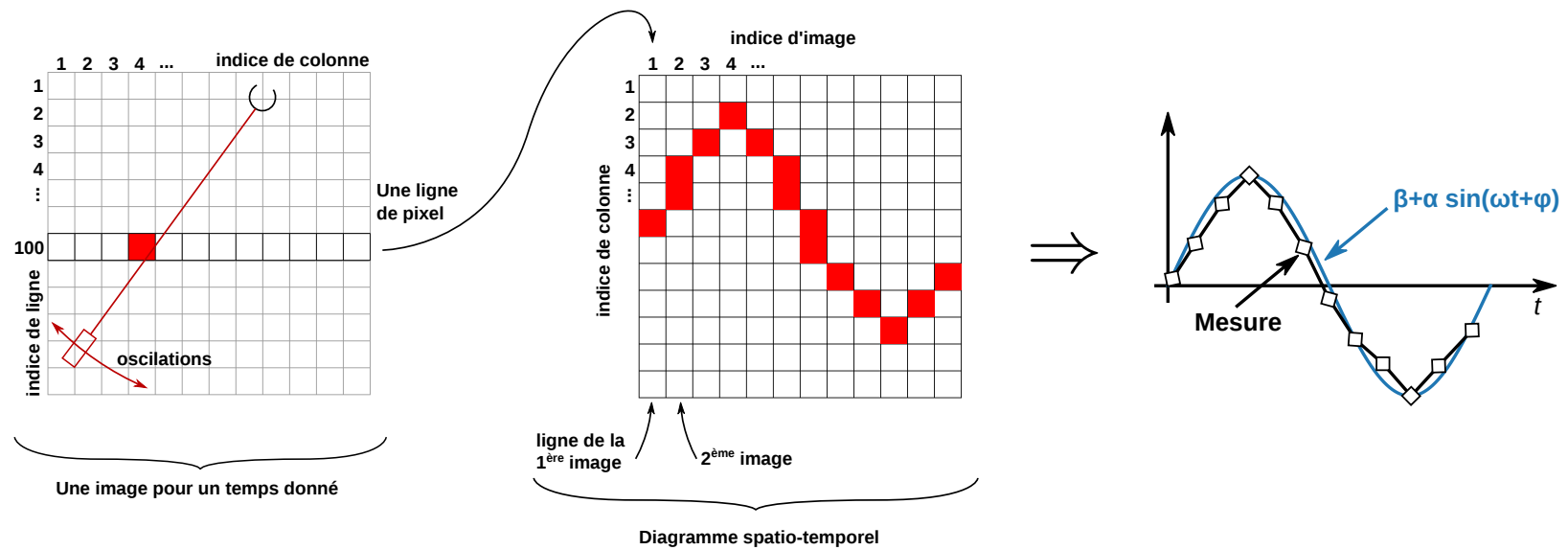


FIGURE 2 – Extraction du déplacement du pendule au cours du temps.

## Exercice 2 : Pour aller plus loin ...

Cette partie est facultative. Elle correspond aux idées nécessaires à la construction et à la compréhension de la fonction `avi2mat`

Dans ce TP, les deux fichiers `.mat` avec les tableaux contenant les images des deux films sont directement fournis au bon format pour Matlab. Dans la pratique, on dispose de films avec des formats classiques `avi`, `mpeg`, `mp4` .... Ce qui suit vous explique comment relire vous-même ces films et vous introduit quelques éléments sur les objets comme `videoReader`. Ce n'est pas au programme du 2A103.

Pour charger une vidéo dans matlab et créer un objet matlab de type vidéo<sup>2</sup> on utilisera la commande suivante :

```
>> vidObj = VideoReader('nom_de_la_video.avi');
```

1. En double cliquant sur `vidObj` dans votre workspace, que vaut le champ `CurrentTime`? Dans un script, on pourrait aussi accéder à cette variable qui s'appelle `vidObj.CurrentTime`.

Ensuite pour accéder à la première image et la "ranger" dans le tableau `img1` on écrit

```
>> img1 = readFrame(vidObj);
```

En faisant cela `vidObj` se prépare et il est passé à la seconde image.

2. En double cliquant à nouveau sur `vidObj` dans votre workspace, quelle est alors la nouvelle valeur du champ `CurrentTime`? Vous identifierez également le `FrameRate` (la fréquence d'échantillonnage) ainsi que le `Duration` (durée du film).
3. À l'aide de la commande `imagesc`, afficher la première image.
4. Quelle est la dimension de `img1` et pourquoi?
5. Copier le tableau `img1(:,:,1)` dans tableau à deux dimensions. De cette manière le rouge R du (RGB) devient le niveau de gris d'une image en noir et blanc.
6. En répétant les opérations précédentes, remplir un tableau `M` à trois dimensions contenant toutes les images en niveaux de gris (s'aider d'une boucle). Pour ne pas répéter la création, assez lente, de la matrice `M` chaque fois que l'on cherche à exploiter le film, on pourra enregistrer ce tableau dans un fichier `.mat`, et venir le charger une bonne fois pour toute (voir les commandes `save` et `load`). Notez que le fichier `.mat` est en revanche assez lourd en espace disque comparé au film initial.

---

2. pas de panique : c'est juste un autre objet, comme le sont les tableaux, les logiques, les chaînes de caractères ... sauf que l'objet `video` a plusieurs " tiroirs " où sont rangées d'autres informations! N'hésitez pas à explorer ce qu'il contient. C'est équivalent aux structures en C ou au type dérivé en fortran