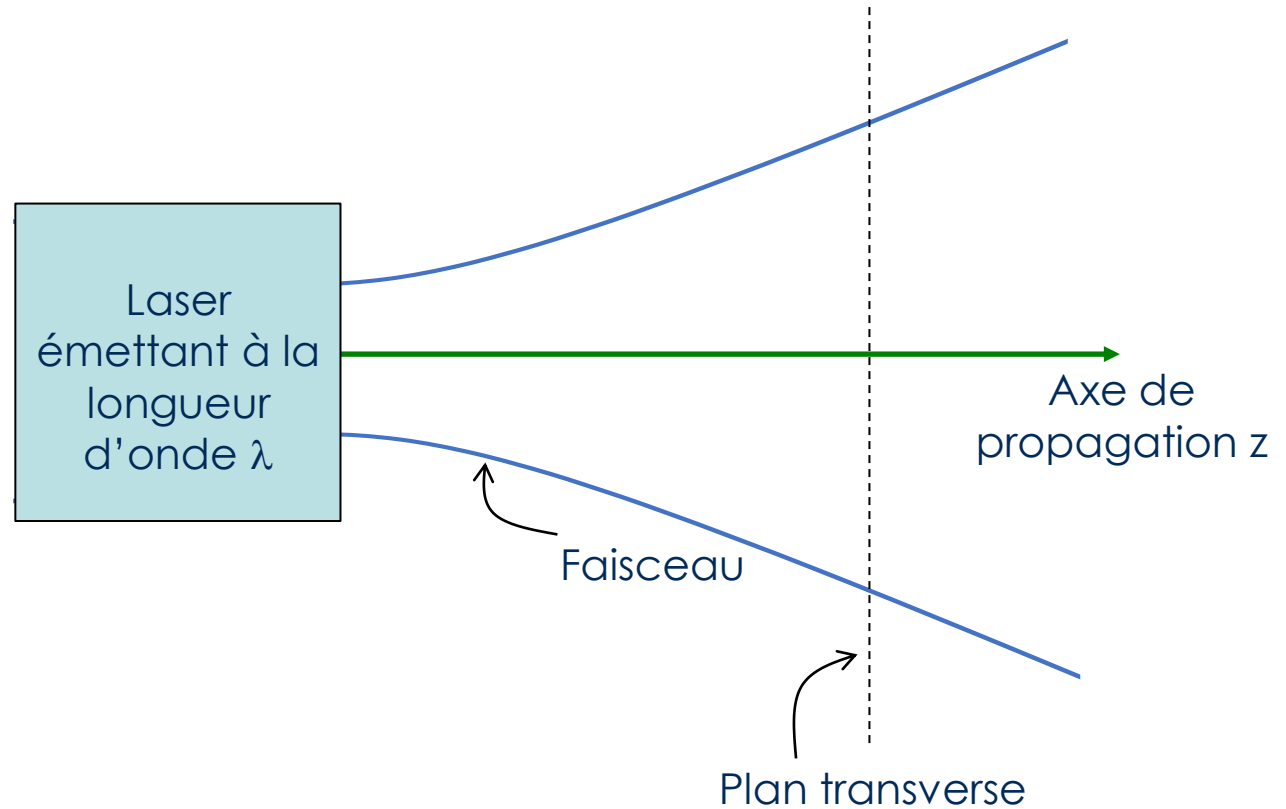


Outils Numériques pour Ingénieur.e en Physique

Bloc 2 Faisceaux laser

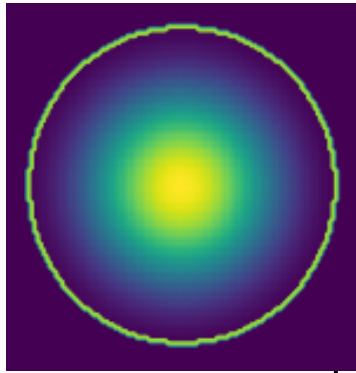
Introduction : les faisceaux laser gaussiens



Le faisceau de référence en physique des lasers est un faisceau dont la **distribution d'intensité** ou profil spatial dans le plan transverse est **gaussien** et dont l'évolution selon z est **limitée par la diffraction**.

Ce faisceau « parfait » n'existe pas ! inhomogénéités du milieu à gain, défauts des optiques ...

Introduction : les faisceaux laser gaussiens

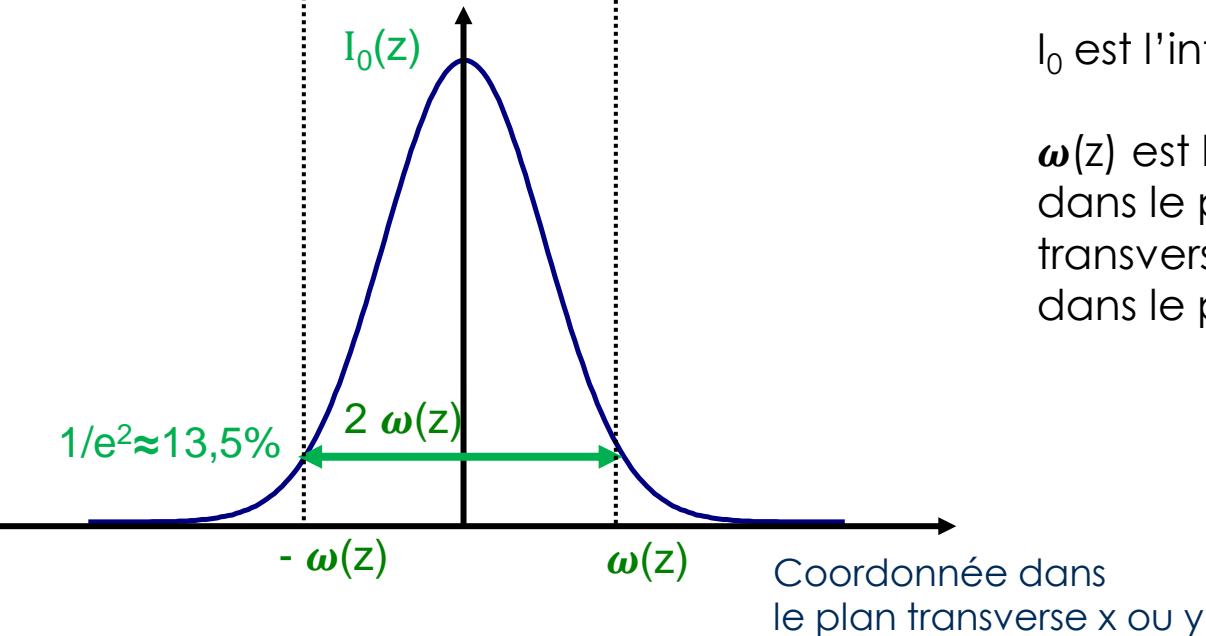


Une distribution transverse d'intensité gaussienne s'écrit :

$$I(x, y, z) = I_0(z) \exp\left(-2 \frac{x^2 + y^2}{\omega(z)^2}\right)$$

I_0 est l'intensité maximale

$\omega(z)$ est le rayon de l'onde gaussienne dans le plan transverse (ou encore "rayon transverse" ou encore "taille" du faisceau dans le plan transverse considéré) à $1/e^2$



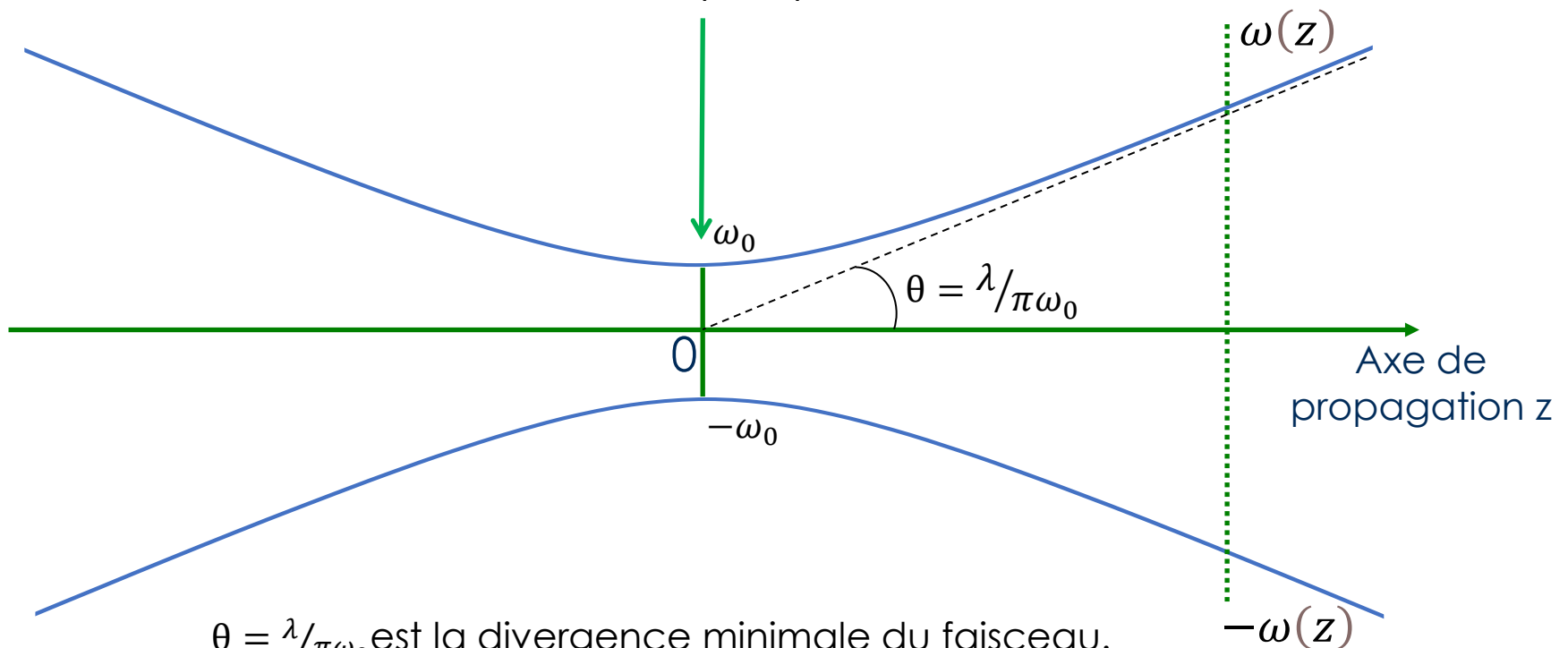
Introduction : les faisceaux laser gaussiens

L'évolution du rayon $\omega(z)$ est connue et s'écrit :

$$\omega(z) = \omega_0 \sqrt{1 + \left(\frac{z\lambda}{\pi\omega_0^2}\right)^2}$$

ω_0 est le rayon du waist (ou encore col) du faisceau à $1/e^2$

C'est l'endroit où le faisceau est le plus petit



$\theta = \lambda / \pi\omega_0$ est la divergence minimale du faisceau.

On dit ici que le faisceau est limité par la diffraction (cas idéal)

Problématique : caractérisation de faisceaux laser expérimentaux

En physique des lasers, il est souvent nécessaire de pouvoir qualifier la qualité spatiale d'un faisceau laser.

Pour cela on mesure l'écart du faisceau réel à un faisceau parfaitement gaussien.

Cet écart est caractérisé par le facteur M^2 défini par :

$$\omega(z) = \omega_0 \sqrt{1 + \left(\frac{zM^2\lambda}{\pi\omega_0^2} \right)^2}$$

NB : si le M^2 vaut 1, le faisceau est dit limité par la diffraction.
En pratique, le M^2 est supérieur à 1.

Méthode expérimentale

- Une méthode expérimentale pour mesurer le M^2 consiste à :
- prendre sur une caméra des images en coupe transverse à différentes abscisses z
 - calculer pour chaque image le rayon* $\omega(z)$
 - reconstituer le profil $\omega(z)$
 - estimer son écart à un faisceau parfaitement gaussien
 - en déduire le facteur M^2

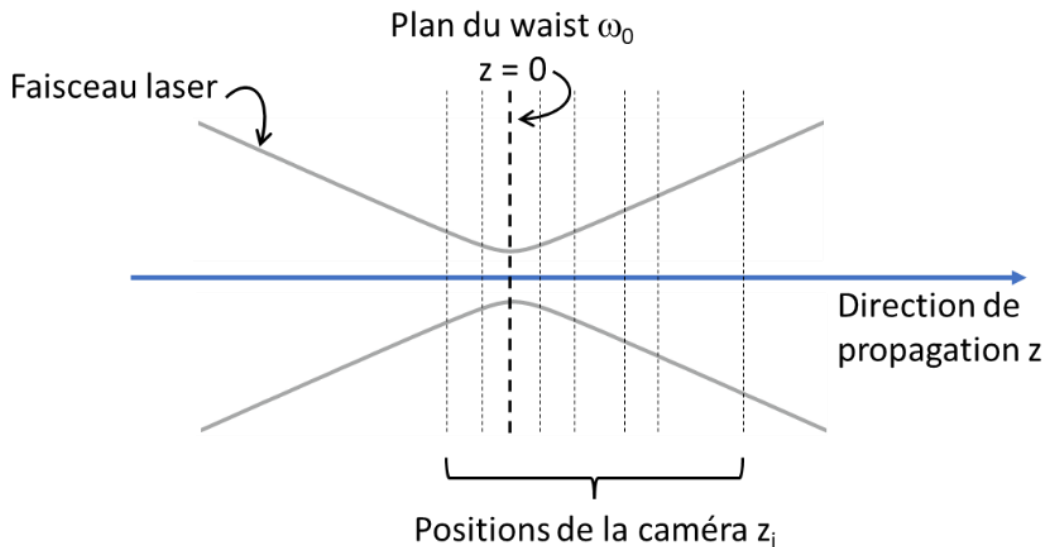
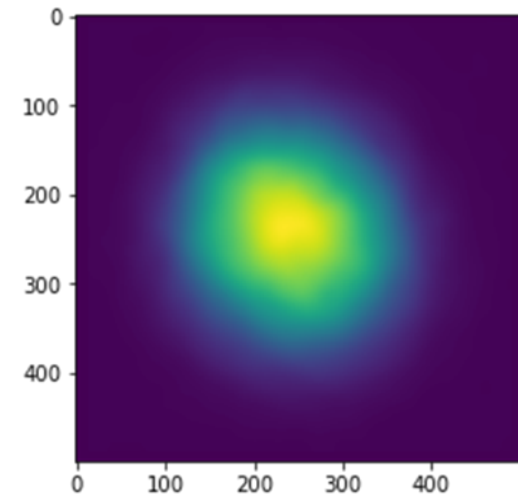


Schéma de principe de la méthode de mesure de M^2



Exemple d'une coupe transverse

*dans la suite, on s'intéresse à des faisceaux pas trop éloignés d'un faisceau gaussien. Dans ce cas le rayon $w(z)$ est pris à $1/e^2$. En cours, vous verrez une méthode plus générale.

Objectif du bloc 2

Nous vous fournissons 15 images de profils transverses d'un faisceau laser expérimental prises à 15 abscisses différentes.

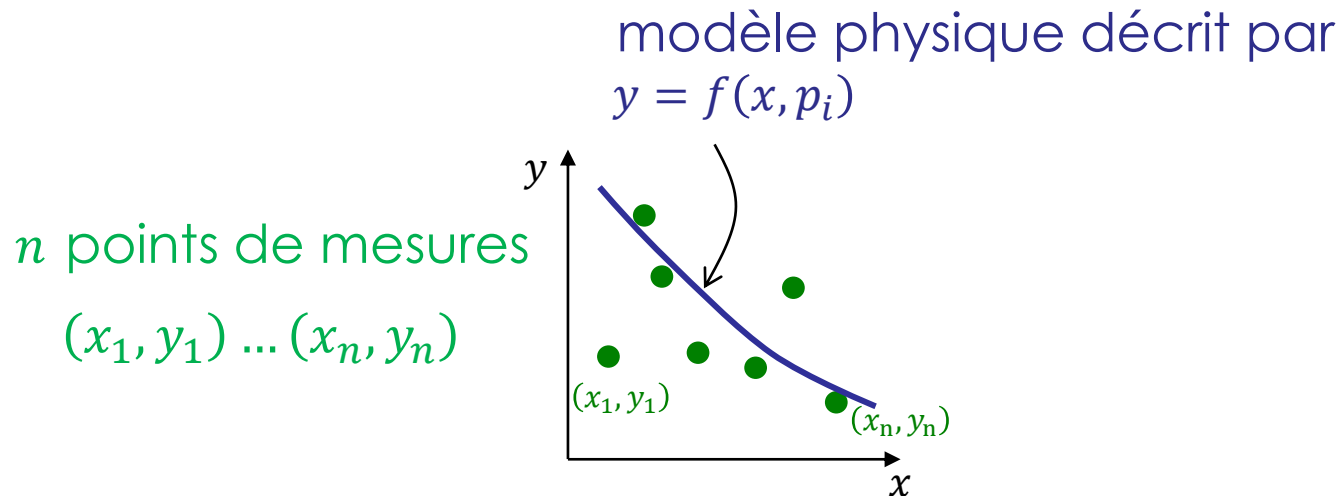
L'objectif principal est de calculer à partir de ces images le facteur M^2 de ce faisceau dans les deux directions transverses x et y.

Vous utiliserez pour cela une méthode numérique appelée **méthode des moindres carrés**.

Vous en connaissez déjà une forme simple avec la régression linéaire.

Ajustement de données

- Un problème très courant en physique est **d'ajuster** un modèle physique sur un nuage de points expérimentaux.
- Le modèle physique peut posséder **un ou plusieurs paramètres inconnus p_i** .
- « Ajuster » consiste à déterminer le jeu de paramètres qui permet de **reproduire au mieux les données expérimentales**.
- Si ces paramètres p_i ont un sens physique, on a alors **une estimation indirecte de leur valeur**.



But : trouver les paramètres p_i tels que la fonction f représente au mieux les points expérimentaux.

Méthode des moindres carrés – principe général

On introduit une fonction de ces paramètres p_i décrite par une somme quadratique des écarts des mesures au modèle* :

$$\chi^2(p_i) = \sum_{k=1}^n [f(x_k, p_i) - y_k]^2$$

On montre que cette fonction χ^2 est minimale lorsque les paramètres p_i sont les plus proches du modèle physique décrit par la fonction f .

Il s'agit donc d'une **recherche de minimum** de la fonction χ^2 .

C'est un problème non trivial.

Heureusement, il existe une fonction sous Python qui fait cela !

`scipy.optimize.curve_fit(f, xdata, ydata, p0= ..., ...)`

f : fonction qui traduit le phénomène physique

xdata : vecteur qui contient les abscisses des données expérimentales

ydata : vecteur qui contient les ordonnées correspondantes

p0 : vecteur qui contient les valeurs de départ des paramètres de l'ajustement

*si l'incertitude n'est pas identique en chaque point, il faut diviser chaque terme par σ_k^2

Brainstorming pour déterminer les outils et méthodes nécessaires

Vous disposez de 15 images en niveau de gris codées sur 8 bits.

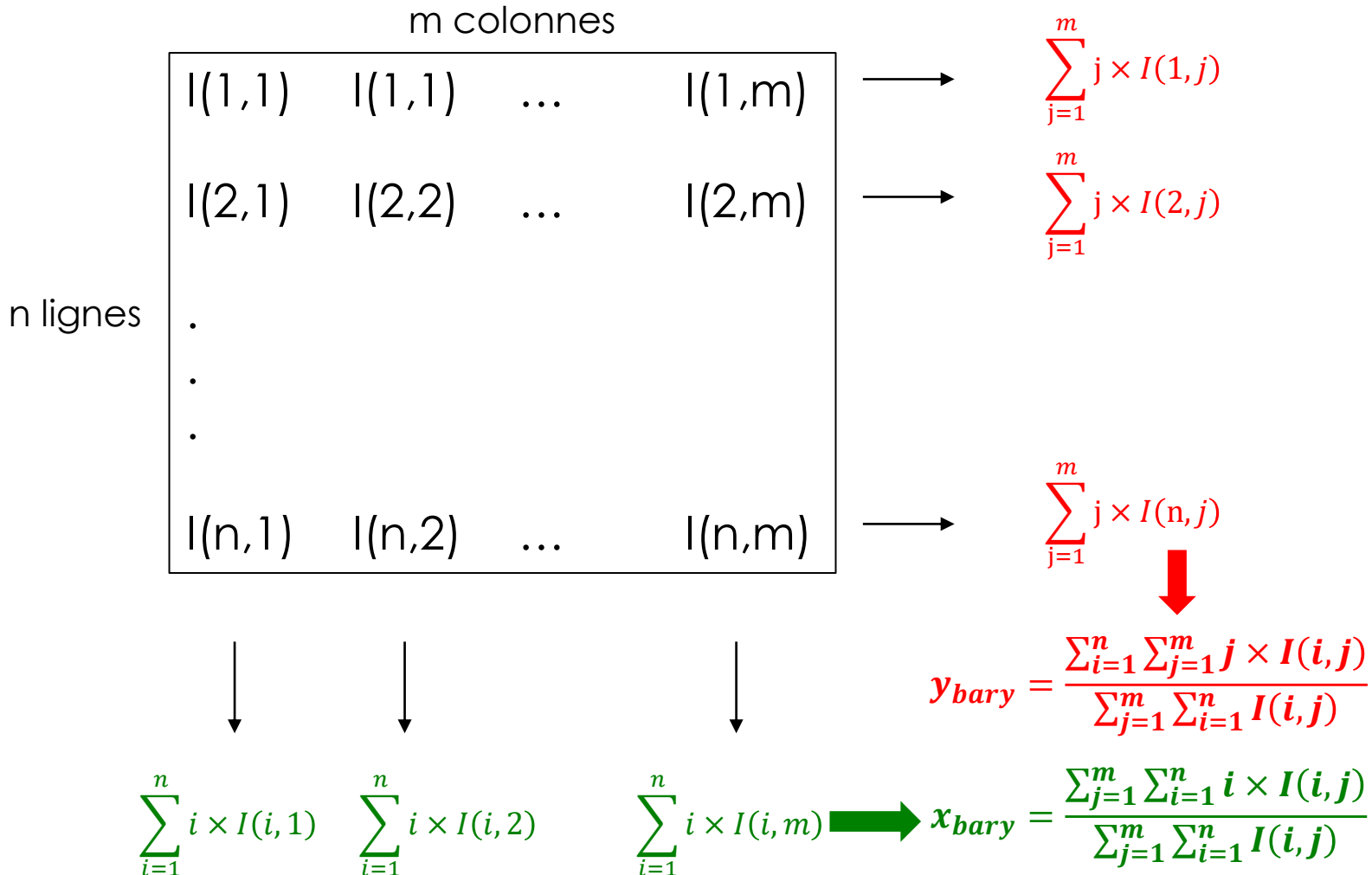
Les noms des images ainsi que l'abscisse z à laquelle elles ont été prises sont indiqués dans le fichier **data.csv**.

Et ensuite, de quels outils et méthodes a-t-on besoin :

- pour traiter une image ?
- pour traiter les 15 images ?

Calcul du barycentre d'une image

Pour calculer les coordonnées du **barycentre** d'une image de n lignes par m colonnes, on réalise une moyenne pondérée par l'intensité des pixels par ligne et par colonne :



Quelques fonctions utiles

- Dans la bibliothèque `matplotlib` :

`plt.imread(nom)` : lit une image depuis le fichier `nom` et renvoie un tableau sur cette image.

`plt.imshow(image)` : affiche les données contenues dans le tableau `image` sous forme d'une image.

`plt.subplot(nb lignes, nb colonnes, n° de la figure)` : affiche plusieurs graphiques sur une même figure.

`plt.axvline(x = ...)` ou `plt.axhline(y = ...)` : affiche une ligne verticale ou horizontale.

`plt.xlabel` ou `plt.ylabel` : affiche le titre d'un axe sur un graphique.

`plt.scatter` : affiche des points.

- Dans la bibliothèque **numpy** :

meshgrid

```
X = np.array([1,2,3]) #vecteur de 3 points
Y = np.array([4,5,6,7]) #vecteur de 4 points
XX,YY = np.meshgrid(X,Y)
```

XX est un tableau de vecteurs lignes représentant X autant de fois qu'il y a de points selon Y :

```
[1,2,3]
[1,2,3]
[1,2,3]
[1,2,3]
```

YY est un tableau de vecteurs colonnes représentant Y autant de fois qu'il y a de points selon X :

```
[4,4,4]
[5,5,5]
[6,6,6]
[7,7,7]
```

En compilant les deux, on dispose de **toutes les coordonnées possibles** et une seule fois. On peut ainsi utiliser une fonction de 2 variables très simplement.

Soit $g : x,y \rightarrow g(x,y) = 2x + y$

$g(XX,YY)$ s'écrit :

```
[g(1,4), g(2,4), g(3,4)]
[g(1,5), g(2,5), g(3,5)]
[g(1,6), g(2,6), g(3,6)]
[g(1,7), g(2,7), g(3,7)]
```

Soit

```
[6, 8, 10]
[7, 9, 11]
[8, 10, 12]
[9, 11, 13]
```

Ce projet comporte 14 questions.

Lors de la séance d'évaluation, vous nous présenterez les réponses aux questions, numériques ou graphiques, soit sous forme de Notebook Jupyter soit dans un fichier texte (word ou autre) à part.

Vous serez également évalué.e sur votre code selon les critères habituels (cf grille d'auto-évaluation).