**Bibliothèque de fonctions**

**Manipulation d’images :**

* **bin2mat** (file) : charge l’image au format binaire en double dont les 32 premiers

bits correspondent à la taille selon x et les 32 suivants, selon y

* file : nom du fichier binaire à ouvrir
* **name\_Z** (fpath, Z) : retourne le filepath complet d’une image, avec le nombre correct de 0 devant la profondeur
* fpath : chemin relatif ou absolu jusqu’à ‘Z= ‘
* Z : la profondeur correspondant à l’image que l’on étudie
* **imdata** (dossier, numero, im) : renvoie l’image en position « numero » dans le dossier Mesures ou Calibration
* dossier : M ou 1 pour Mesures, Cou 0 pour Calibration
* num : valeur du focus
* im : pour le dossier Mesures, numéro de l’image
* **imshow2** (image) : identique à imshow mais affiche l’image sur une plage [min,max], en l’étirant
* **imshowf** (image,n) **:** affiche séparément l’amplitude et la phase d’une image dans Fourier

**Interface utilisateur :**

* **myui** (fpath, pas) : crée un slider pour l’affichage des images et des graphes. Le slider fait varier la profondeur Z et charge les images en conséquence affiche l'image avec des traits correspondant à l'inclinaison des franges
* fpath : chemin relatif ou absolu jusqu’à ‘Z= ‘
* pas : intervalle séparant les différentes profondeurs correspondant aux images
* **plotmoy**(dossier,pasx,pasy) : affiche les moyennes (image traitées avec masque\_rephase) pour différents focus

**Analyse d’image :**

* **moyhor** (img, n) : renvoie les moyennes horizontales d'une image, en partant du centre et en prenant un écart horizontal de n.
* img : image dont on cherche à obtenir la moyenne
* n : largeur de la bande que l’on prend en compte pour faire la moyenne
* **moyvert** (img, n) : renvoie les moyennes verticales d'une image, en partant du centre et en prenant un écart horizontal de n.
* img : image dont on cherche à obtenir la moyenne
* n : largeur de la bande que l’on prend en compte pour faire la moyenne
* **fit\_gauss** (x, y) : renvoie les coefficients de p correspondant à l’approximation d’une gaussienne avec offset de la forme y = A + B\*exp((x-D)2)/(2s2)
* x et y : données à approximer
* p : coefficients autour desquels on va chercher la bonne approximation : p(1) = offset, p(2) = amplitude, p(3) = moyenne, p(4) = écat-type
* **fit\_gauss2D** (img) : renvoie les coefficients de p correspondant à l’approximation d’une gaussienne 2D avec offset (voir fonction précédente)
* img : image de la gaussienne 2D dont on cherche les paramètres
* **minEQM** (h, s, r) : renvoie la valeur optimale du paramètre dans l’expression : .
* h : image de la perturbation
* s : signal à traiter
* r : image de référence
* **jaco\_der** (k, offset, A, mu1, mu2, s, x, y, e) : détermine la matrice jacobienne d’une fonction du type : f(x,y) =offset + A\*exp( -( ( x-mu1)2+(y-mu2) 2 ) /(2\*s2) ) (utilisé pour fit\_gauss2D)
* **Rgauss** (x, y, p0) : calcule une gaussienne 1D de centre x,y et de paramètres p0.
* **Rgauss2D** (data, p0, dim\_h, dim\_v) : calcule la gaussienne en deux dimensions en utilisant des vecteurs 1D
* data : données à partir desquelles on souhaite déterminer la gaussienne
* p0 : paramètres de la gaussienne
* dim\_h et dim\_v : dimensions horizontale et verticale des données
* **calc\_rapport** (interfrange) : calcule la valeur du rapport pas/lambda par approximation polynomiale (on a z=pas/lambda \* interfrange
* interfrange : vecteur contenant les valeurs des interfranges des images de calibration à profondeur croissante (tous les 25nm).

**Traitement d’image :**

* **passebas\_hg2D**(img,s,k) : renvoie l’image après application d’un filtre passe-bas classique : une hypergaussienne (de degré k) de révolution d’écart-type s

- img : l’image à traiter

- s : écart-type de l’hypergaussienne

- k : degré de l’hypergaussienne

* **traitement\_fft\_gaussienne**(image) : renvoie l’image avec la phase annulée (abs(img)) en Fourier

**-** image : image à traiter

* **masque\_hg**( img, c, a, s, offset, k ) : renvoie l’image multipliée par une hypergaussienne d’amplitude a. On sélectionne donc la partie centrale de l’image
* img : image à traiter
* c : centre souhaité de l’hypergaussienne
* a : amplitude de l’hypergaussienne
* s : écart-type de l’hypergaussienne. Si k est grand (~ >7), s ~ le rayon en px
* offset : offset éventuel de l’hypergaussienne
* k : degré de l’hypergaussienne, si k est grand, les bords sont plus durs
* **hgaussp** (dim, c, a, s, offset, k) : renvoie une hypergaussienne 2D
* dim : taille de l’image
* a : amplitude
* c : centre de la gaussienne 2D (x,y)
* s : écart-type
* offset : offset
* k : degré de la gaussienne, pair
* **frange\_fourier** : Script pour visualiser les franges, enlève une hypergaussienne en Fourier pour ne pas garder la tâche centrale.
* **masque\_rephase**(img, sigma) :   
  -applique un masque hypergaussien de variance sigma (sa saisie est optionnelle : par défaut il prendra 3 fois la variance de la gaussienne centrale)

-centre l’image en supprimant sa phase dans l’espace de Fourier

* **FiltreWiener**(h,img,lambda) : applique le filtre de Wiener associé a une image qui a été convoluée par h (entre autre une PSF). Lambda parametre du filtre peut être choisit avec minEQM.
* **Removefilter(img) :** Permet d’enlever l’interferogramme sur les images expérimentales, qui est dû au bruit de fond. On enlève donc environ 12 pics dans l’espace de Fourier.
* **Removegauss(img) :** enlève la gaussienne détectée sur une image