

Manipulation des fichiers sous Matlab

Ouverture et fermeture d'un fichier

- `fid=fopen('nomfich.ext','permission')` : ouvre un fichier `nomfich.ext` et retourne un entier identificateur de fichier dans `fid`
 - `'permission'` =
 - __ `'r'` ouvre en lecture seulement un fichier existant (par défaut)
 - __ `'w'` ouvre en écriture seulement un fichier existant ou crée un nouveau fichier; efface le contenu du fichier
 - __ `'a'` ouvre en écriture seulement un fichier existant ou crée un nouveau fichier; le contenu n'est pas effacé place les nouvelles données écrites à la fin du fichier
 - __ `'r+'` ouvre en lecture et en écriture un fichier existant place les nouvelles données écrites en début de fichier et écrase le contenu du fichier
 - __ `'w+'` ouvre en lecture et en écriture un fichier existant ou crée un nouveau fichier; efface le contenu du fichier
 - __ `'a+'` ouvre en lecture et en écriture un fichier existant ou crée un nouveau fichier; le contenu n'est pas effacé place les nouvelles données écrites à la fin du fichier
- `[fid,message]=fopen('nomfich.ext','permission')` : retourne un entier identificateur de fichier dans `fid` et un message d'erreur dans `message` si une erreur se produit à l'ouverture du fichier `nomfich.ext`
 - `fid` = un entier non négatif si aucune erreur à l'ouverture -1 si une erreur se produit
- `status=fclose(fid)` : ferme un fichier identifié à par `fid` lors de son ouverture par la commande `fopen`
 - retourne un entier non négatif = 0 si le fichier est correctement fermé, 1 si une erreur se produit lors de la fermeture
- `status = fclose('all')` : ferme tous les fichiers ouverts

Lire des données binaires

- `A=fread(fid, taille, 'précision')` : lit tout ou partie d'un fichier binaire identifié par `fid` lors de son ouverture et retourne les données dans une matrice `A`
 - `taille` : étendue des valeurs lues dans le fichier
 - `N` = lit les `N` premières valeurs du fichier
 - `inf` = lit jusqu'à la fin du fichier (par défaut)
 - `[n m]` = lit les valeurs du fichiers arrangées dans un tableau `n*m`
 - `'précision'` : contrôle le type des données pour chaque valeur
 - `'char'` et `'uchar'` = caractères signés et non signés (8 bits)
 - `'short'` et `'long'` = entiers courts et longs (16 et 32 bits)
 - `'float'` et `'double'` = valeurs à virgule flottante de précision simple et double (32 et 64 bits) (par défaut)
- `count=fwrite(fid, A, 'précision')` : écrit les éléments d'une matrice `A` dans un fichier identifié par `fid` et les enregistre dans un format numérique spécifié par `'précision'`
 - retourne le nombre d'éléments écrits dans le fichier

Lire des données texte

- **A = fscanf(fid, 'format' , taille)** : lit un fichier texte identifié par **fid** et retourne les données dans une matrice **A**

- **'format'** : interprétation des données

- '%s' = interprète une chaîne de caractères

- '%d' = interprète des entiers

- '%e', '%f' et '%g' = nombres à virgule flottante

- '%*d' = passe les valeurs entières dans les données

- **taille** : étendue des valeurs lues dans le fichier

- N = lit les N premières valeurs du fichier

- inf = lit jusqu'à la fin du fichier (par défaut)

- [n m] = lit n*m éléments du fichier arrangés dans un tableau n*m

- **tline = fgets(fid)** : lit une ligne d'un fichier texte identifié par **fid** et l'enregistre dans une chaîne de caractères **tline** sans le caractère de fin de ligne

- retourne -1 si **fgets** rencontre la fin du fichier

- fichiers texte seulement

- commence la lecture de la ligne à partir de l'indicateur de position (par défaut, le début du fichier à son ouverture)

(fgets / fgetl)

Ecrire des fichiers texte

- **count = fprintf(fid,'format' ,A)** : écrit les données de la matrice **A** dans un fichier identifié par **fid** et retourne le nombre de bytes écrit dans le fichier

- **'format'** : interprétation des données

- '%s' = interprète une chaîne de caractères

- '%d' = interprète des entiers

- '%e' = notation exponentielle

- '%f' = valeurs décimales à virgule fixe

- '\n' = retour à la ligne

- '\t' = espace tabulation