

# Mémo rapide pour Linux, Ferret et NCO

Contact : [yann.ourmieres@univ-tln.fr](mailto:yann.ourmieres@univ-tln.fr)

## A) Linux

Linux est le système d'exploitation du PC (tout comme windows). Pour votre travail, la principale chose à comprendre est que ce système fonctionne moins par des boutons à cliquer que windows et plus avec des lignes de commande, que l'on tape dans une fenêtre préalablement ouverte que l'on appelle un terminal.

\* Ouvrir un terminal : dépendant de la version, soit on tape terminal dans la barre de recherche, soit on peut faire un clique-droit, soit dans les applications, soit dans outils système, etc...

\* une fois dans le terminal vous pouvez effectuer quelques opérations de base :

**pwd** (affiche le chemin absolu du répertoire courant)

**ls** (liste, affiche les répertoires et les fichiers du répertoire actif)

**ls** (affiche seulement les noms)

**ls toto\*** (affiche les fichiers commençant par toto)

**ls -l** (affiche le format long : types + droits + Nbre de liens + ....)

**cd** (change directory : changement de répertoire)

**cd /home/etudiants/Documents/repertoire1** (vers le répertoire dont le chemin absolu est donné)

**cd ..** (répertoire parent ou répertoire un niveau au-dessus)

**cd -** (répertoire visité précédemment)

**cd** (répertoire racine)

## B) Logiciel de visualisation Ferret

Ferret est un outil de visualisation. Il fonctionne par scripts ou en lignes de commandes (c'est à dire sans une interface graphique). Il fonctionne sous linux principalement. Les fichiers que vous allez manipuler dans ce TP sont des fichiers au format netcdf (extension .nc)

Avec Ferret, il est possible d'exploiter des fichiers NetCDF à travers des commandes, avec toujours la même syntaxe:

**COMMANDE /option1/option2/.../option\_n nom\_variable[d=numéro de fichier]**

\***Pour lancer Ferret**, aller dans un terminal et taper **pyferret** puis entrée.

(Attention : dépendant des machines, cette commande peut varier → voir avec l'enseignant)

\***Pour quitter Ferret**, il faut utiliser les commandes **exit** ou **quit**.

*A noter que Ferret ne fait pas la différence entre majuscules et minuscules.*

### \* Opérations de base:

– **Charger un fichier:** **use nom\_du\_fichier.nc**

Les différentes variables du fichier sont alors reconnues par Ferret et utilisables.

**Remarque** : si le fichier que vous souhaitez charger n'est pas dans le répertoire ou vous avez lancé ferret, il va falloir indiquer le chemin complet du fichier .

**Par exemple** : **use « /mnt/TLNDATA/OCEAN\_MODEL/nom\_du\_fichier.nc » pour un fichier qui serait dans /mnt/TLNDATA/OCEAN\_MODEL/**

– **Connaître les variables disponibles dans le/les fichiers chargés** : **show data** ( ou **sh d**)

la liste de toutes les variables et leurs dimensions respectives s'affichent alors.

- Pour utiliser plusieurs fichiers en même temps, il faut les appeler chacun avec la commande use, il seront ainsi dans la mémoire Ferret, puis préciser pour chaque variable à quel fichier elle appartient, par **[d=nombre]**. Par exemple si vous voulez visualiser la variable U du fichier numéro 3, il faudra mettre U[d=3].
- Pour **supprimer** des fichiers chargés en mémoire ferret: **Cancel nom\_du\_fichier** ou cancel data 2 si ce fichier était répertorié comme le jeux de données numéro 2
- pour **créer une nouvelle variable**, on la définit par la commande **LET nom\_variable= ..**
- **Pour ouvrir une nouvelle figure: set w/n** ; si on a 2 figures ouvertes (figure 1 et 2) et que l'on veut ouvrir une 3ème fenêtre: **set window 3**; dans la même logique pour visualiser quelque chose sur la figure numéro 3 par exemple, alors que l'on travaille sur une autre figure, on fait: **set window 3**.
- Ainsi, pour **fermer une figure** , la figure 2 par exemple, il suffit de faire **cancel window 2** ou plus court **cancel w 2**.

**ATTENTION : si on clique sur la croix pour fermer une image, 2 fois sur 3 cela fait planter FERRET !**

– Pour faire une **opération en shell sans quitter ferret: sp « commande shell»**  
exemple: on veut lister les fichiers disponibles dans le répertoire courant donc « ls » en shell. Il suffit de faire **sp ls**.

– **Pour sauvegarder une figure** : FRAME /FILE=picture.gif , par exemple ou .png  
Attention sur certaines versions de ferret, il faut faire en sorte que l'image soit au premier plan, sinon ça ne marche pas ! Il faut aussi éventuellement faire au préalable un **set window n**, pour sélectionner la fenêtre n si ce n'est pas la fenêtre actuellement valide.

## \* Visualisations :

### a) Visualisation 1D

Pour tracer en une dimension, il faut utiliser la commande PLOT. Après avoir fixé 1 ou 2 des composantes géographiques ( 2 parmi **i,j,k** ou **x,y,z** pour 1 fichier 3D, ou 1 pour un fichier 2D) et la composante temporelle si il y en a une (l ou t), PLOT trace une variable en fonction de la composante restante (du type  $y=f(x)$  donc) .

Pour tracer deux variables, les séparer par une virgule.

Exemple:

**PLOT /k=1:50/x=6/y=42 TEMP** -> Trace la variable TEMP en fonction de la profondeur, à la latitude (y=42) et longitude (x=6) fixées, pour les 50 premiers niveaux verticaux (k)

**PLOT /k=1:50/x=6/y=42 TEMP[d=1],TEMP[d=2]** -> Trace la variable TEMP en fonction de la profondeur, à la latitude (y=42) et longitude (x=6) fixées, pour le fichier 1 et 2.

### b) Visualisation 2D

Pour tracer en 2D, il existe trois commandes principales: **SHADE**, **FILL** et **CONTOUR**. Il suffit de fixer une des coordonnées géographiques, ainsi que la composante temporelle, et de choisir la variable désirée. SHADE trace alors cette variable en 2D, à l'aide d'un gradient de couleur. FILL fait la même chose mais en interpolant entre chaque valeur de couleur, donnant un résultat visuel plus «lisse». CONTOUR affiche les isolignes et leurs valeurs. Il est cependant possible de faire apparaître un gradient de couleur dans CONTOUR (CONTOUR/FILL) ou des isolignes dans SHADE (SHADE/LINE). Toutes les principales options nécessaires au tracé seront détaillées par la suite .

Exemple: **SHADE /k=1/l=1 TEMP** -> Trace la variable temp en surface (k=1) au pas de temps 1

(l=1)

### Tracer des vecteurs:

La commande **VECTOR** permet de tracer des vecteurs en 2D. Elle obéit aux mêmes lois que SHADE et CONTOUR. Il suffit de lui fixer une des coordonnées d'espace, puis de lui donner les deux composantes du vecteur.

Exemple:

VECTOR /i=120:140/j=100:120/k=1 U, V → trace le vecteur ayant pour composante U et V, pour la région i=120:140 et j=100:120, et le niveau vertical k=1

Il est possible de superposer des vecteurs à un SHADE ou FILL, en utilisant l'option /OV

### Une spécificité NEMO à retenir :

si vous faites shade/k=1/l=1 Temp la figure sera en point de grille (ordonnée / abscisse)

si vous faites shade/k=1/l=1 Temp,nav\_lon,nav\_lat la figure sera en degrés longitude, latitude (ordonnée / abscisse)

### c) Les options les plus utiles: (toute option doit être précédée de /)

-over ou ov : Utile pour superposer des vecteurs sur un shade, par exemple.

shade /k=1 module\_vitesse

- hlimits=min:max:pas ( pour les commandes plot, shade, contour, etc...) : Détermine les limites de l'axe horizontal. Pour l'axe vertical, on utilise vlimits.

Exemple:

plot/j=130/i=180/k=1:40/hlimits=10:30:0.5 temp → l'axe des y , représentant la variable temp sera borné entre 10 et 30 , par pas de 0.5.

*Option très importante quand on superpose des courbes qui n'ont pas les mêmes extremum !*

- xskip (pour la commande vector) : Donne le pas d'affichage entre deux vecteurs sur la coordonnée x(espace entre les flèches!). yskip pour la coordonnée y.

exemple:

vector/over/k=1/xskip=5/yskip=5 U[d=1],V[d=2] -> affiche un vecteur sur 5, en x et y

- length ( pour la commande vector) : Permet de spécifier la norme du vecteur étalon

exemple:

vector/over/k=1/length=0.4 U[d=1],V[d=2] -> Le vecteur de référence vaudra 0.4 m/s

- vs ou versus (pour la commande plot ) : Exprime une variable en fonction d'une autre.

Exemple:

plot/j=130/i=180/k=1:40/vs Temp, Sal → affiche la variable Sal en fonction de la variable temp sur les 40 premiers niveaux, au point de grille i=180, j=130.

- lev=(min,max,pas) (pour les commandes shade, contour) : Modifie la gamme et la précision de la barre de couleur.

Exemples:

shade/lev=(19,25,0.1)/k=1 temp → affiche la variable temps de 19 à 25 par pas de 0.1

fill/k=1/lev="(inf) (30,38.5,0.1) (39)" sal -> ajoute une borne inférieure à « moins l'infini » à la colorbar et une borne supérieure à 39

**Remarque:** pour faire afficher les côtes, il suffit de taper une commande (seule) :go coastline ou

## go medi\_coast

Cette commande va tracer les traits de côte sur la figure courante.

Les 2 commandes se valent, mais parfois, dépendant du répertoire dans lequel on est, une seule d'entre elle est opérationnelle

### d) les options de calcul

on peut faire une opération mathématique sur une ou plusieurs variables et coordonnées et visualiser avec les commandes classiques

- **@AVE pour la moyenne (Average)**
- **@MAX pour la maximum**
- **@MIN pour le minimum**
- **@INT pour l'intégrale**
- ...

#### \* Utilisation:

**shade/k=1 temp[d=1, l=@AVE]** → affiche la variable temp pour le niveau k=1, moyenné sur tous les pas de temps (tous les «l» pas de temps donc) disponibles dans le fichier 1.

**plot /k=1 sal[d=1, i=@MAX]** → affiche la valeur du maximum de la variable sal (pour k =1) pour tous les j, en regardant à chaque j, le maximum sur tous les i.

**shade/k=1 temp[d=1, l=@MAX]** → affiche le maximum de la variable temp pour le niveau k=1, calculé sur tous les pas de temps disponibles dans le fichier 1, pour tous les points i,j (si fichier 3D)

- pour calculer le **carré** d'une variable il faut mettre ^2 accolé à la variable

*exemple : shade/k=1/l=1 TEMP[d=1]^2*

- pour calculer la **racine carrée**, il faut mettre ^0.5.

- pour additionner/ soustraire, multiplier.. une valeur fixe à une variable ou plusieurs variables entre elles, il suffit d'utiliser le symbole mathématique classique que l'on souhaite

*exemple : Shade/k=1/l=1 TEMP[d=1]+12 permet de visualiser la variable TEMP augmentée de 12 (degrés).*

- on peut utiliser des parenthèses pour prioriser les calculs

*exemple : shade/k=1/l=1 (TEMP[d=1]+12)\*2*

\* **La commande stat** : elle donne l'écart-type, la moyenne , le min et le max d'une ou plusieurs variables.

on peut l'utiliser avec toutes les options identiques à celles utilisées pour les commandes de visualisation.

Exemple:

**stat/k=1/j=1:100/i=20:50 temp** → donne les valeurs énoncés ci-dessus pour la variable temp calculée pour le niveau k=1, et la zone i=20 à 50 et j= 1 à 100.

### **e) un fichier particulier : fichier.des**

les fichiers «fichier.des» sont des fichiers qui permettent virtuellement à ferret de considérer une concaténation de plusieurs fichiers pour avoir un fichier virtuel regroupant tous les pas de temps inclus dans tous les fichiers individuellement. Ce type de fichier est particulièrement pratique pour une concaténation temporelle de fichiers 3D , car une concaténation réelle donnerait un fichier trop lourd à manipuler

Il s'utilise de la même manière qu'un fichier normal:

**use fichier.des**

etc...

### **C) Logiciel nco**

Ce logiciel pourra être éventuellement utilisé pour une commande en particulier, qui permet de voir le header des fichiers « .nc ». Le header est un résumé des informations contenues dans le fichier: noms des variables, signification, dimensions des variables etc...

- Cette commande est:

**ncdump -h nom\_fichier.nc**

Cette commande permet notamment, pour le style de fichiers manipulés dans ce TP de vérifier la zone géographique correspondante au fichier (latitude, longitude)

- Si on veut afficher à l'écran la liste des valeurs d'une variable contenue dans le fichier on fait:

**ncdump -v nom\_variable nom\_fichier.nc**