

Mastère M1 Observatoire de Paris

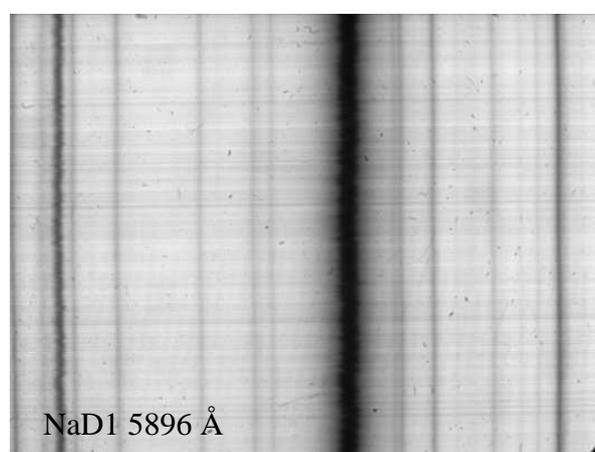
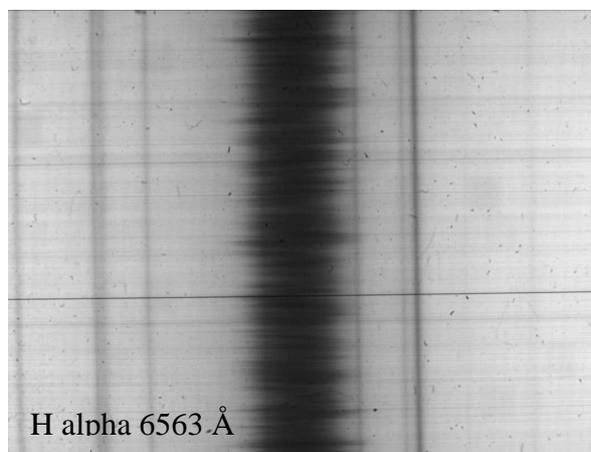
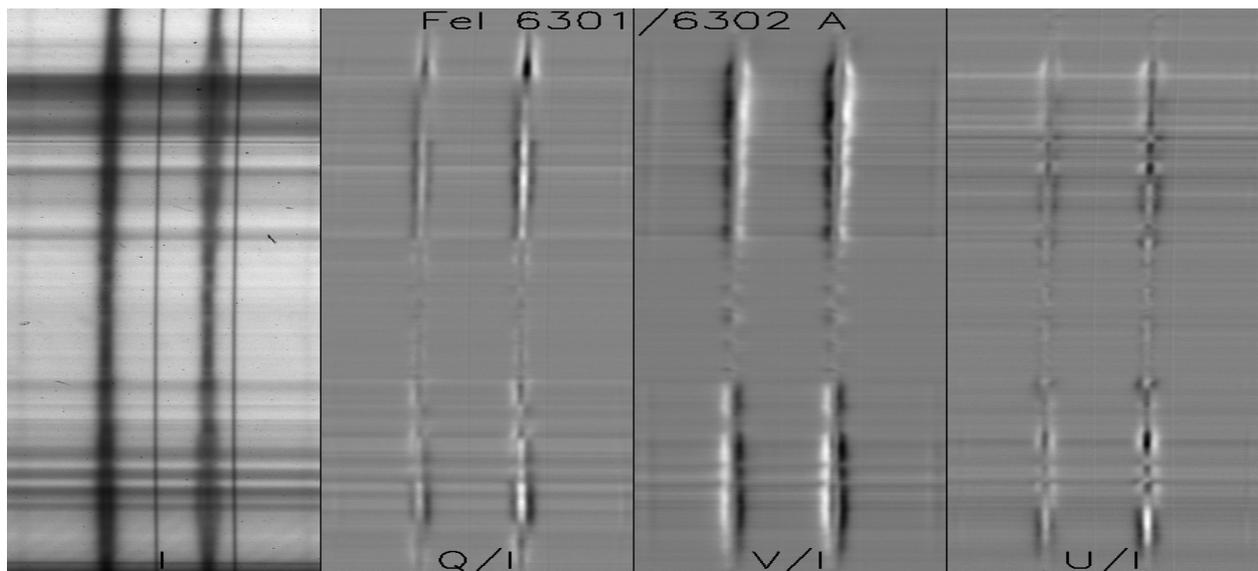
Module « optique solaire »

Travaux dirigés numériques

Travaux pratiques d'observation à la Tour Solaire de Meudon

Spectroscopie et spectro polarimétrie

Guillaume Molodij, Jean-Marie Malherbe, Jacques Moity
2008



I - Travaux dirigés numériques

Sommaire

Les TD sont présentés par ordre croissant de difficulté

TD 1 : effet Doppler sur les raies spectrales

TD 2 : « second spectre » : polarisation linéaire de résonance au limbe

TD 3 : « second spectre » : éléments rares, structure hyperfine, molécules

TD 4 : CaII K : une raie à Stokes V inversé dans les taches

TD 5 : Cartographie 2D de I, Q, U, V sur une tache solaire

TD 6 : effet Zeeman sur une tache solaire : détermination du champ magnétique longitudinal et transversal dans plusieurs raies

TD 7 : démodulation des paramètres de stokes I, Q, U, V

Atlas en ligne du spectre solaire :
http://solaire.obspm.fr/pages/spectres/spectre_3000.html



TD 1 : effet Doppler sur les raies spectrales (obligatoire !)

On fournit des profils de raies obtenus à la Tour Solaire de Meudon dans plusieurs raies : H α 6563Å, NaD1 5896 Å, NaD2 5890 Å, SrI 4607 Å sous forme d'images (λ , x), en abscisse la longueur d'onde et en ordonnée une direction spatiale solaire . Pour chaque observation :

- 1) identifier les raies présentes dans chaque champ à l'aide de l'atlas en ligne
- 2) identifier le sens de variation de la longueur d'onde : λ croît elle vers la gauche ou vers la droite ?
- 3) calculer la valeur du pixel spectral (combien de mÅ par pixel ?) à l'aide de l'atlas par comparaison avec les observations
- 4) calculer un profil **moyen** en moyennant 20 profils sur une zone « calme » juste au dessus de la zone étudiée plus 20 profils juste au dessous (de manière à compenser l'inclinaison des raies, que l'on ne demande pas de corriger, pouvant introduire un décalage systématique)
- 5) évaluer la vitesse radiale en **km/s** des spectres les plus décalés vers le bleu ou vers le rouge à partir du décalage Doppler entre le profil moyen et le profil étudié, **mesuré par la méthode du centre de gravité** (facile à programmer). On définira à cet effet la position d'une raie en longueur d'onde, à une fraction de pixel près, par la relation :

$$\lambda_r = [\int \lambda (1 - I(\lambda)/I_c) d\lambda] / [\int (1 - I(\lambda)/I_c) d\lambda]$$

où l'intégration porte sur un domaine spectral égal au double de la largeur à mi hauteur de la raie (de l'aile « bleue » à l'aile « rouge »). L'intégration peut être réalisée avec la fonction INT_TABULATED du langage IDL. $I(\lambda)$ est l'intensité du profil et I_c est l'intensité du continu (on pourra prendre pour I_c une moyenne des deux points extrêmes en longueur d'onde du profil $I(\lambda)$). La fonction $1 - I(\lambda)/I_c$ représente donc une fonction de poids qui est nulle dans les ailes de la raie et maximale au cœur de la raie.

Nota : cette méthode du centre de gravité servira dans d'autres exercices

TD 2 : « second spectre » : polarisation linéaire de résonance au limbe

Lorsqu'on observe au limbe ou à quelques secondes d'arc du limbe, on constate que certaines raies du spectre solaire sont polarisées linéairement avec un taux de polarisation linéaire Q/I qui peut atteindre 1 %. Cette polarisation est due à l'**éclaircissement anisotrope** (assombrissement centre bord) des atomes sur la ligne de visée par diffusion résonante de la lumière (excitation de l'atome vers un niveau d'énergie supérieur puis retour immédiat au niveau de départ). Les raies **SrI 4607 Å**, **BaII 4554 Å** et **CaI 4227 Å** fournies figurent parmi les plus polarisées du spectre du limbe solaire en polarisation linéaire, appelé aussi « second spectre ». Le taux de polarisation étant modéré, on a répété l'observation de nombreuses fois de manière à pouvoir améliorer la qualité du signal par sommation des spectres. Les fichiers nommés *0_b1.fts contiennent des spectres de type I+Q et ceux nommés *1_b1.fts contiennent des spectres de type I-Q. Il y a 3 sous répertoires correspondant à chacune des raies à étudier (observations faites au Pic du Midi). Pour chaque observation :

- 1) identifier les raies présentes dans chaque champ à l'aide de l'atlas en ligne, et repérer la position de SrI 4607 Å, BaII 4554 Å et CaI 4227 Å
- 2) identifier le sens de variation de la longueur d'onde : vers la gauche ou vers la droite ?
- 3) prendre le premier couple de fichiers I+Q et I-Q ; en extraire une fenêtre de travail (sens spectral et spatial) ; trouver le **facteur correctif de transmission** de telle sorte que les valeurs moyennes des deux images I+Q et I-Q soient égales, puis appliquer cette correction à l'image I-Q ; par somme et différence, calculer ensuite I puis Q et enfin l'image **Q/I**.
- 4) **moyenner** Q/I le long de la fente (direction spatiale) puis représenter graphiquement Q/I (λ)
- 5) évaluer le rapport signal/bruit obtenu après sommation le long de la fente sachant que chaque image possède au départ un rapport S/B = 100. De combien a t'on amélioré le rapport S/B en sommant les spectres le long de la fente ?
- 6) refaire le même traitement (3, 4, 5) en partant de la **somme** de tous les fichiers I+Q et de la **somme** de tous les fichiers I-Q. Evaluer le nouveau rapport S/B obtenu. De combien l'a t'on amélioré en travaillant sur la **somme** des fichiers de même nature ?

TD 3 : particularités et richesse du « second spectre solaire »: éléments rares, structure hyperfine, molécules

Le second spectre solaire, ou spectre en polarisation linéaire au limbe (Q/I) est extrêmement riche en phénomènes dont la complexité n'a commencé à se dévoiler qu'au cours des vingt dernières années. Il existe par exemple des raies **d'éléments rares** très peu visibles en intensité, mais qui apparaissent très clairement en polarisation de résonance. Tel est le cas de YI 4128 Å et de EuII 4130 Å, éléments peu présents dans l'atmosphère solaire.

- 1) identifier les raies présentes dans le champ à l'aide de l'atlas en ligne, et repérer la position de YI 4128 Å et de EuII 4130 Å
- 2) identifier le sens de variation de la longueur d'onde : vers la gauche ou vers la droite ?
- 3) effectuer la **somme** des fichiers I+Q (*a.fts) et la somme des fichiers I-Q (*b.fts); en extraire une fenêtre de travail (sens spectral et spatial) ; trouver le **facteur correctif de transmission** de telle sorte que les valeurs moyennes des deux images finales I+Q et I-Q soient égales, puis appliquer cette correction à l'image I-Q ; par somme et différence, calculer I puis Q et enfin l'image résultante **Q/I**. Afficher cette image.
- 4) **moyenner** ensuite Q/I le long de la fente (dans la direction spatiale) puis représenter graphiquement le résultat Q/I (λ) : que remarquez vous ?

Il existe aussi des raies présentant une **structure hyperfine étonnante**, comprenant plusieurs pics de polarisation autour du cœur de la raie. Tel est le cas de ScII 4247 Å.

- 5) identifier les raies présentes dans le champ à l'aide de l'atlas en ligne, et repérer la position de ScII 4247 Å
- 6) identifier le sens de variation de la longueur d'onde : vers la gauche ou vers la droite ?
- 7) effectuer la **somme** des fichiers I+Q (*a.fts) et la somme des fichiers I-Q (*b.fts); en extraire une fenêtre de travail (sens spectral et spatial) ; trouver le **facteur correctif de transmission** de telle sorte que les valeurs moyennes des deux images finales I+Q et I-Q soient égales, puis appliquer cette correction à l'image I-Q ; par somme et différence, calculer I puis Q et enfin l'image résultante **Q/I**. Afficher cette image.
- 8) **moyenner** ensuite Q/I le long de la fente (dans la direction spatiale) puis représenter graphiquement le résultat Q/I (λ) : que remarquez vous ? Les 3 pics de polarisation, ou structure **hyperfine**, sont liés aux **isotopes** de ScII, dont les **spins nucléaires** sont différents les uns des autres (l'isotope principal est Sc - A=45, Z=21 - avec un spin nucléaire 7/2)

Il existe enfin des **raies moléculaires** dans le spectre solaire, très peu visibles en intensité, car les molécules sont peu abondantes.

- 9) Examiner l'atlas en ligne autour de la longueur d'onde 5140 Å
- 10) effectuer la **somme** des fichiers I+Q (sa*.fts) et la somme des fichiers I-Q (sb*.fts); en extraire une fenêtre de travail (sens spectral et spatial) ; trouver le **facteur correctif de transmission** de telle sorte que les valeurs moyennes des deux images finales I+Q et I-Q soient égales, puis appliquer cette correction à l'image I-Q ; par somme et différence, calculer I puis Q et enfin l'image résultante **Q/I**. Afficher cette image.
- 11) **moyenner** ensuite Q/I le long de la fente (dans la direction spatiale) puis représenter graphiquement le résultat $Q/I(\lambda)$: **quelles sont les molécules qui ont une signature en polarisation ?**

TD 4 : CaII K : une raie à Stokes V inversé dans les taches

La raie CaII K à 3934 Å est une raie chromosphérique très profonde et aux ailes très étendues. Son profil est très sensible aux variations de température du milieu, en particulier le cœur de la raie (K3) ainsi que les ailes au voisinage immédiat du cœur (pics K2).

On dispose d'observations polarimétriques en polarisation circulaire effectuées au Pic du Midi autour d'une tache. Ces observations ont été répétées pour pouvoir améliorer lors du traitement le rapport S/B. Les fichiers contenant I+V et I-V sont nommés respectivement *0.fts et *1.fts.

- 1) identifier les raies présentes dans chaque champ à l'aide de l'atlas en ligne, et repérer la position de la raie CaII K
- 2) identifier le sens de variation de la longueur d'onde : λ croît elle vers la gauche ou vers la droite ?
- 3) calculer la valeur du pixel spectral (combien de mÅ par pixel ?) à l'aide de l'atlas (le pixel spatial vaut 0.2 arc sec)
- 4) **sommer** tous les fichiers I+V et **sommer** tous les fichiers I-V; en extraire une fenêtre de travail (sens spectral et spatial) ; par somme et différence, calculer I puis V et enfin **V/I**
- 5) tracer les profils I et V/I en quelques points d'une tache, d'une facule, et du soleil calme ; remarquer l'**inversion** de Stokes V en CaII K dans la tache par rapport aux nombreuses raies voisines, et la présence de **2 polarités opposées** le long de la fente du spectrographe

TD 5 : Cartographie 2D de I, Q, U, V sur une tache solaire

On a observé au télescope Thémis une tache solaire dans les raies FeI 6301/6302 Å et on souhaite effectuer une cartographie 2D des paramètres de Stokes. Le télescope a pour ce faire balayé la surface du soleil en se déplaçant par pas de 0.8 arc sec, en effectuant 60 pas. Le fichier scanquv.fts contient ainsi 60 positions de fente, décalées spatialement à chaque fois de 0.8'', avec pour chaque position dans l'ordre I+Q, I-Q, I+U, I-U, I+V, I-V (on a donc 60 sextuplets qui se suivent dans le fichier, un sextuplet pour chaque position spatiale de la fente sur le Soleil).

- 1) identifier les raies présentes dans le champ à l'aide de l'atlas en ligne ; choisir la raie FeI 6301.5 Å ou bien la raie FeI 6302.5 Å
- 2) identifier le sens de variation de la longueur d'onde : vers la gauche ou vers la droite ?
- 3) calculer la valeur du pixel spectral (combien de mÅ par pixel ?) à l'aide de l'atlas par intercomparaison (le pixel spatial vaut 0.4 arc sec)

- 4) pour chaque position de fente, à partir d'un sextuplet I+Q, I-Q, I+U, I-U, I+V, I-V, par somme et différence, calculer les images spectrales $I(\lambda, x)$, $Q(\lambda, x)$, $U(\lambda, x)$, $V(\lambda, x)$ puis :
 - $V/I(x)$ au pic V d'une des ailes de la raie
 - $I(x)$, $Q/I(x)$, $U/I(x)$, $(Q^2 + U^2)^{1/2}/I(x)$ et $U/Q(x)$ au cœur de raie
 - $I_c(x)$ intensité dans le continu loin de la raie
 (x est la coordonnée le long de la fente)
- 5) dresser, en répétant en boucle l'opération précédente sur les 60 positions de fente, une carte 2D pour $I(x, y)$, $I_c(x, y)$, $V/I(x, y)$, $(Q^2 + U^2)^{1/2}/I(x, y)$ et $\frac{1}{2} \arctan[U/Q(x, y)]$ (x est la coordonnée le long de la fente et y celle dans la direction du balayage). **Remarque** : le champ magnétique **longitudinal** est proportionnel à V/I ; le carré du champ magnétique **transversal** est proportionnel à $(Q^2 + U^2)^{1/2}/I$ et son azimuth donné par $\frac{1}{2} \arctan(U/Q)$
- 6) Effectuer un REBIN ou un CONGRID de toutes ces images avant affichage pour corriger l'**anamorphose** x/y qui provient du fait que la résolution spatiale le long de la fente dans le sens x est de $0.4''$ alors que le pas du balayage dans le sens y est de $0.8''$

TD 6 : effet Zeeman sur une tache solaire : détermination du champ magnétique longitudinal et transversal dans plusieurs raies

On a observé au Pic du Midi une tache solaire dans un grand nombre de raies spectrales différentes : CaI 6103 Å, CaI 6122 Å, FeI 5247 Å, FeI 5250 Å, FeI 5576 Å, FeI 6173 Å, FeI 6301/6302 Å, CaI 6103 Å, FeI 6337 Å, HI 6563 Å, MgI 5173 Å, MgI 5167 Å, MnI 4757 Å, NaI 5896 Å, en polarimétrie complète. Il y a plusieurs exemples (séquences) pour chaque raie. Pour chaque exemple, il y a 6 fichiers contenant I+Q, I-Q, I+U, I-U, I+V et I-V, les fichiers étant nommés respectivement *0.fts, *1.fts, *2.fts, *3.fts, *4.fts, *5.fts. Il est donc possible par somme et différence d'obtenir à partir de ces sextuplets d'observations des profils de Stokes $Q/I(\lambda)$, $U/I(\lambda)$ et $V/I(\lambda)$. Comment interpréter ces observations en termes de champs magnétiques ?

I - Mesure du champ magnétique longitudinal $B_{//}$ par la mesure des déplacements en longueur d'onde des composantes Zeeman

La méthode consiste à repérer l'écartement en longueur d'onde $2 \Delta\lambda_B = \lambda_1 - \lambda_2$ entre les deux profils I+V et I-V en mesurant l'écartement des deux composantes Zeeman, chacune pouvant être repérée en longueur d'onde par la méthode du centre de gravité **définie à l'exercice 1** :

$$\lambda_1 = \frac{\int \lambda (1 - (I+V(\lambda))/I_c) d\lambda}{\int (1 - (I+V(\lambda))/I_c) d\lambda} \text{ et}$$

$$\lambda_2 = \frac{\int \lambda (1 - (I-V(\lambda))/I_c) d\lambda}{\int (1 - (I-V(\lambda))/I_c) d\lambda}$$

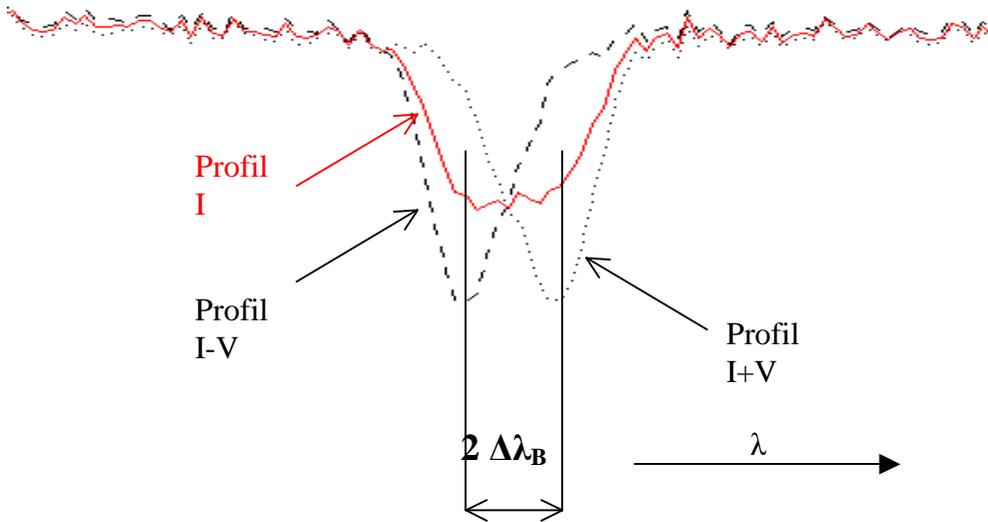
où l'intégration porte sur un domaine spectral égal au double de la largeur à mi hauteur de la raie (de l'aile bleue à l'aile rouge). L'intégration peut être réalisée avec la fonction INT_TABULATED du langage IDL.

La théorie de l'effet Zeeman nous donne $\Delta\lambda_B = (\lambda_1 - \lambda_2)/2 = 4.67 \cdot 10^{-13} B_{//} g^* \lambda^2$

D'où l'on en déduit $B_{//}$ (attention : B en Gauss et λ en Å dans cette formule).

- 1) identifier les raies présentes dans chaque champ à l'aide de l'atlas en ligne, et relever le **facteur de Landé** de la raie étudiée (valeur indiquée sur l'atlas)
- 2) identifier le sens de variation de la longueur d'onde : vers la gauche ou vers la droite ?
- 3) calculer la valeur du pixel spectral (le pixel spatial vaut 0.2 arc sec)
- 4) ré-échantillonner dans le sens spatial à 1 arc sec (REBIN par un facteur 5) pour améliorer le rapport S/B
- 5) tracer les profils **I+V** et **I-V** en un point d'une tache, d'une facule, ...
- 6) évaluer $\Delta\lambda_B = \lambda_1 - \lambda_2$ en utilisant la méthode du **centre de gravité** ci dessus
- 7) en déduire le champ magnétique en chaque point du champ sélectionné

Remarque : FeI 5576 Å possède un facteur de Landé g^* nul ; vérifiez qu'elle est insensible à l'effet Zeeman (on l'appelle « raie non magnétique »)



II - Mesure du champ magnétique longitudinal $B_{//}$ et du champ magnétique transverse B_{\perp} à partir des profils de Stokes

La méthode que nous proposons s'applique uniquement aux raies photosphériques fines et se base sur l'approximation théorique des champs faibles (voir cours). Dans cette approximation :

$$V(\lambda) = 4.67 \cdot 10^{-13} B_{//} g^* \lambda^2 \frac{dI}{d\lambda}$$

$$Q(\lambda_0) = -1/4 (4.67 \cdot 10^{-13} g^* \lambda^2)^2 B_{\perp}^2 \cos(2\varphi) \frac{d^2I}{d\lambda^2} \text{ au coeur de la raie}$$

$$U(\lambda_0) = -1/4 (4.67 \cdot 10^{-13} g^* \lambda^2)^2 B_{\perp}^2 \sin(2\varphi) \frac{d^2I}{d\lambda^2} \text{ au coeur de la raie}$$

On voit que $(Q^2 + U^2)^{1/2}(\lambda_0) = 1/4 (4.67 \cdot 10^{-13} g^* \lambda^2)^2 B_{\perp}^2 \frac{d^2I}{d\lambda^2}$ et $U/Q(\lambda_0) = \tan(2\varphi)$, formules valables au cœur de la raie seulement.

Attention : B est en Gauss et λ en Å dans ces formules . On suppose que le profil peut se mettre sous la forme d'une **gaussienne** $I(\lambda) = I_c (1 - r e^{-1/2 ((\lambda-\lambda_0)/\Delta\lambda)^2})$ avec r dépression centrale de la raie, I_c niveau du continu, λ_0 longueur d'onde centrale de la raie, et $2 \Delta\lambda$ largeur de la raie aux points d'inflexion du profil. Aux points où $\lambda = \lambda_0 \pm \Delta\lambda$, $|dI/d\lambda|$ est maximal, et on obtient :

$$V/I(\lambda_0 \pm \Delta\lambda) = \pm 4.67 \cdot 10^{-13} B_{//} g^* \lambda_0^2 [r e^{-1/2} / ((1 - r e^{-1/2}) \Delta\lambda)]$$

En mesurant approximativement r (dépression centrale) et $\Delta\lambda$ (demi largeur aux points d'inflexion) sur les profils en intensité $I(\lambda)$ ainsi que V/I aux deux pics V (situés au voisinage des points d'inflexion en $\lambda_0 \pm \Delta\lambda$), on peut en déduire une estimation du **champ longitudinal $B_{//}$** .

On peut employer la même méthode pour estimer le champ transverse B_{\perp} . Au point particulier où $\lambda = \lambda_0$ (centre de la raie), on obtient :

$$(Q^2 + U^2)^{1/2}/I(\lambda_0) = 1/4 (4.67 \cdot 10^{-13} g^* \lambda_0^2)^2 B_{\perp}^2 [r / (\Delta\lambda^2 (1 - r))]$$

En mesurant approximativement $(Q^2 + U^2)^{1/2}/I$ au centre de la raie, on peut en déduire une estimation du **champ transverse B_{\perp}** . Son **azimuth** sera donné par : $\varphi = 1/2 \arctan(U/Q)$ à 180° près.

- 1) identifier les raies présentes dans chaque champ à l'aide de l'atlas en ligne, et noter le facteur de Landé g^* de la raie sélectionnée (valeur indiquée sur l'atlas)
- 2) identifier le sens de variation de la longueur d'onde
- 3) calculer la valeur du pixel spectral (le pixel spatial vaut 0.2 arc sec)
- 4) ré-échantillonner dans le sens spatial à 1 arc sec (REBIN par un facteur 5) pour améliorer le rapport S/B
- 5) calculer puis tracer en fonction de la longueur d'onde les profils **I, Q/I, U/I et V/I** en un point d'une tache, d'une facule, ...
- 6) évaluer r et $\Delta\lambda$ sur le profil en intensité **I**
- 7) déduire des **pics V/I** le champ magnétique **longitudinal** en chaque point sélectionné
- 8) calculer le profil $(Q^2 + U^2)^{1/2}/I$ et déduire du pic central le champ magnétique **transversal** en chaque point sélectionné ; tirer du rapport **U/Q** au centre raie l'**azimuth** ϕ

TD 7 : démodulation des paramètres de stokes

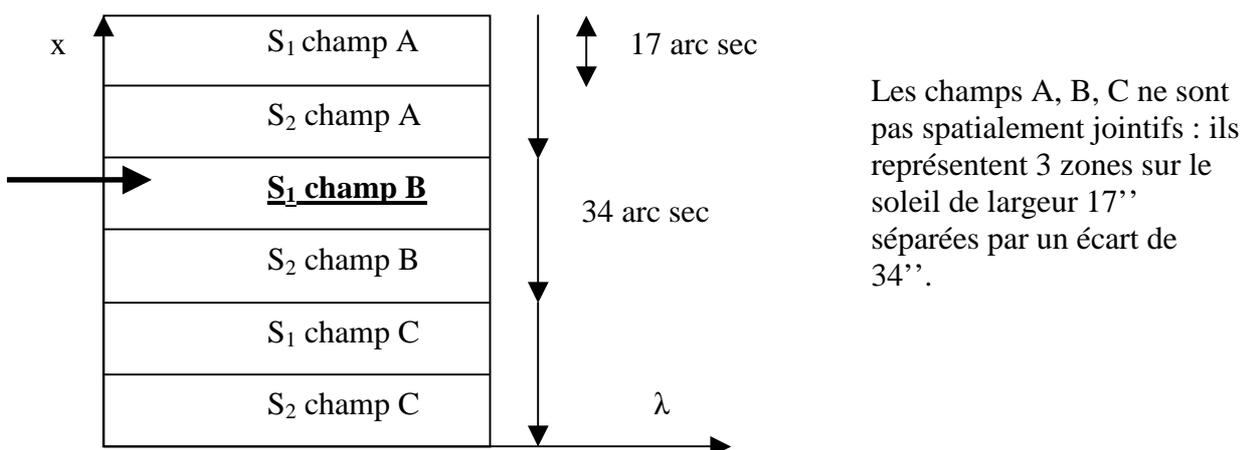
On a observé au télescope Thémis (Tenerife, Canaries) la raie SrI 4607 Å au voisinage du limbe solaire, et on s'intéresse à sa polarisation. Le polarimètre du télescope étant équipé de lames retardatrices qui présentent du chromatisme, la quantité mesurée S est une combinaison linéaire des paramètres de Stokes I, Q, U, V , que l'on écrit $S = \frac{1}{2} [I \pm (q Q + u U + v V)]$.

Il y a en fait 2 signaux mesurés simultanément : $S_1 = \frac{1}{2} [I_1 + (q Q_1 + u U_1 + v V_1)]$ (voie haute) et $S_2 = \frac{1}{2} [I_2 - (q Q_2 + u U_2 + v V_2)]$ (voie basse) car il y a un séparateur de faisceaux biréfringent donnant deux états de polarisation orthogonaux, comme l'indique la figure ci dessous.

De plus, il y a 3 champs d'observation différents : A, B, C, distants chacun sur le soleil de 17''.

On a effectué **6 observations séquentielles** correspondant à 6 combinaisons linéaires différentes à coefficients q, u, v différents, que l'on écrit pour les deux voies simultanées :

$$\mathbf{S}_{1i} = \frac{1}{2} [I_1 + q_i Q_1 + u_i U_1 + v_i V_1] \text{ et } \mathbf{S}_{2i} = \frac{1}{2} [I_2 - q_i Q_2 - u_i U_2 - v_i V_2] \quad 1 \leq i \leq 6$$



Considérons maintenant uniquement la voie haute \mathbf{S}_1 du champ **B** (indiquée sur le schéma par la flèche): les 6 réalisations séquentielles de \mathbf{S}_1 fournissent un ensemble de 6 équations à 4 inconnues I_1, Q_1, U_1, V_1 que l'on résoud aux moindres carrés (voir cours chapitre 5), ce qui donne un premier résultat I_1, Q_1, U_1, V_1 . Les coefficients q_i, u_i, v_i pour les 6 combinaisons $\mathbf{S}_{1i} = \frac{1}{2} [I_1 + q_i Q_1 + u_i U_1 + v_i V_1]$ sont donnés par le tableau ci dessous pour la longueur d'onde de 4607 Å :

	Coeff q_i	Coeff u_i	Coeff v_i	
1.0000000	0.9999601	0.0000000	0.0089313	mesure i=1
1.0000000	-0.9728179	0.0000000	0.2315715	mesure i=2
1.0000000	0.0135910	0.9864089	0.1637458	mesure i=3
1.0000000	0.0135910	-0.9864089	-0.1637458	mesure i=4
1.0000000	-0.1121443	0.0000000	0.9936919	mesure i=5
1.0000000	-0.1121443	0.0000000	-0.9936919	mesure i=6

On peut procéder ensuite de la même manière pour la voie basse S_2 (les coefficients q_i, u_i, v_i changent de signe) ce qui donne un second résultat I_2, Q_2, U_2, V_2 et on peut calculer finalement : $I = 1/2 (I_1 + I_2), Q = 1/2 (Q_1 - Q_2), U = 1/2 (U_1 - U_2), V = 1/2 (V_1 - V_2)$ pour chacun des champs A, B et C.

Données : sr4607.fts, dc.fts, demod4610.txt (coefficients ci dessus)

- 1) identifier les raies présentes dans le champ à l'aide de l'atlas en ligne
- 2) identifier le sens de variation de la longueur d'onde
- 3) calculer la valeur du pixel spectral à l'aide de l'atlas (le pixel spatial vaut 0.2 arc sec)
- 4) calculer le courant d'obscurité en effectuant la moyenne de toutes les images du fichier de courant d'obscurité *dc.fts*
- 5) prendre le premier sextuplet d'images (6 combinaisons) du fichier de données *sr4607.fts* et soustraire de chaque image le courant d'obscurité. En déduire de cette combinaison $I_1(\lambda), Q_1(\lambda), U_1(\lambda), V_1(\lambda)$ par la méthode de démodulation (cours chapitre 5), **en ne prenant que les signaux $S_{1i} = 1/2 [I_1 + q_i Q_1 + u_i U_1 + v_i V_1]$ du champ central B**, $1 \leq i \leq 6$ (voir figure).
- 6) **sommer le long de la fente** dans la direction **spatiale** sur ce champ B et représenter graphiquement les résultats sommés $I_1(\lambda)$, puis $Q_1/I_1(\lambda), U_1/I_1(\lambda)$, et $V_1/I_1(\lambda)$
- 7) évaluer le rapport signal/bruit obtenu après sommation le long de la fente sachant que chaque image possède au départ un rapport S/B = 300
- 8) refaire le même traitement (5, 6) sur les signaux S_1 du champ central B en partant de la **somme de tous les sextuplets** (dont on aura soustrait le courant d'obscurité) contenus dans le fichier (il y a 50 sextuplets qui se suivent au total, on sommera donc 50 fois les combinaisons linéaires identiques de chaque sextuplet). En déduire $I_1(\lambda), Q_1(\lambda), U_1(\lambda), V_1(\lambda)$ et évaluer le nouveau rapport S/B obtenu.
- 9) Pour terminer, faire le même traitement (8) sur les signaux S_2 du champ central B ; en déduire le second ensemble $I_2(\lambda), Q_2(\lambda), U_2(\lambda), V_2(\lambda)$
- 10) Calculer finalement $I(\lambda) = 1/2 (I_1 + I_2), Q(\lambda) = 1/2 (Q_1 - Q_2), U(\lambda) = 1/2 (U_1 - U_2), V(\lambda) = 1/2 (V_1 - V_2)$ et représenter graphiquement les rapports $Q/I(\lambda), U/I(\lambda)$ et $V/I(\lambda)$

Introduction au langage de traitement de données IDL/GDL

Introduction

IDL ou Interactive Data Language est un langage de programmation adapté au traitement interactif de données expérimentales, et en particulier les observations astronomiques.

L'origine d'IDL remonte au début des années 1980, avec l'apparition d'un langage interactif nommé ANA créé par les scientifiques de la NASA pour le traitement des données du satellite Solar Maximum Mission (SOLARMAX), au Goddard Space Flight Center. Depuis, le langage a rencontré un grand succès, s'est considérablement enrichi d'une large bibliothèque de fonctions mathématiques et d'algorithmes variés de résolution de problèmes numériques, basés notamment sur les NUMERICAL RECIPES. Il existe une librairie spécifique aux astronomes, appelée IDL ASTRO : consulter <http://idlastro.gsfc.nasa.gov/homepage.html>

La force d'IDL est de travailler avec des tableaux ou matrices 1D, 2D, 3D (ou plus) aussi simplement qu'avec des scalaires. Une panoplie d'outils graphiques, et surtout d'algorithmes, complète le tout. Mais attention : il ne faut pas manipuler avec IDL les indices de matrices comme en FORTRAN, sinon les performances s'écroulent ! On reviendra sur ce point plus loin.

En physique solaire, IDL est devenu le langage de programmation de référence et est universellement utilisé dans le monde. Le système de traitement SOLARSOFT des données spatiales des satellites SOHO (1996) , TRACE (1999), HINODE (2006)... est écrit en langage IDL, qui a également été adopté par la majorité des télescopes au sol (THEMIS par exemple). Depuis quelques années, IDL a commencé à s'étendre progressivement dans tous les domaines de l'astronomie.

IDL n'est pas un langage gratuit ! Mais il est désormais concurrencé par GDL, ou GNU Data Language, qui se veut compatible :

<http://gnudatalanguage.sourceforge.net/>

http://aramis.obspm.fr/~coulais/IDL_et_GDL/IDL_et_GDL.html

GDL est un logiciel libre, dont les fonctionnalités ne sont pas encore toutes au niveau d'IDL, mais sont en constant enrichissement. C'est donc avec GDL que vous travaillerez en M1.

Comment écrire une procédure IDL (fichier .pro)

Un programme IDL se rédige à l'intérieur d'un éditeur de texte indépendant selon cette forme :

```
Pro nom_programme  
  Suite d'Instructions  
end
```

Si ce fichier programme porte le nom *nom_programme.pro* on le compile en tapant sur le prompt IDL la commande suivante (ne pas oublier le point devant run):

```
IDL> .run nom_programme
```

puis on l'exécute en appelant le programme directement par son nom :

```
IDL> nom_programme
```

On peut passer des variables à une procédure en entrée sortie. Par exemple avec *var_in* en entrée et *var_out* en sortie:

```
Pro nom_programme, var_in, var_out  
  Suite d'Instructions  
End
```

puis on exécute la procédure en appelant le programme directement par son nom :
IDL> *nom_programme, var_in, var_out*

Les variables dans IDL

IDL travaille avec des variables de tout type :

- octets 8 bits de 0 à 255 (type BYTE)
- entiers courts 16 bits signés de -32768 à 32767 (type FIX ou INT)
- entiers courts 16 bits non signés de 0 à 65535 (type UINT)
- entiers longs 32 bits signés (type LONG)
- entiers longs 32 bits non signés (type ULONG)
- réelles 32 bits dites simple précision (type FLOAT), 7 chiffres significatifs entre $\pm 10^{38}$
- réelles 64 bits dites en double précision (type DOUBLE), 14 chiffres significatifs
- complexes 64 bits (type COMPLEX), 7 chiffres significatifs entre $\pm 10^{38}$
- complexes 128 bits (type DCOMPLEX), 14 chiffres significatifs entre $\pm 10^{308}$
- chaînes de caractères (type STRING)

exemples :

`a = 400` définit un entier court

`a = 100000` définit un entier long, et `a = long(400)` force à définir un entier long

`a = 3.4e-7` définit un réel simple précision (en maths $3.4 \cdot 10^{-7}$)

`a = -3.1d-7` définit un réel double précision (en maths $-3.1 \cdot 10^{-7}$)

`a = double(2.8)` force à définir un réel en double précision

`a = complex(1.5, 3.5)` définit un complexe (partie réelle, partie imaginaire) en simple précision, ou en double précision par `a = dcomplex(1.5, 3.5)`

`a = 'GDL en M1'` définit une chaîne de caractères

Conversions de types :

`a = byte(b)` conversion en octet 8 bits

`a = fix(b)` conversion en entier court 16 bits

`a = uint(b)` conversion en entier court non signé

`a = long(b)` conversion en entier long 32 bits

`a = ulong(b)` conversion en entier long non signé

`a = float(b)` conversion en réel 32 bits (flottant simple précision)

`a = double(b)` conversion en réel 64 bits (flottant double précision)

`a = complex(b,c)` conversion en complexe 64 bits (2 flottants simple précision)

`a = dcomplex(b,c)` conversion en complexe 128 bits (2 flottants double précision)

`a = string(b)` conversion en chaîne de caractères

Les vecteurs (1D), tableaux ou matrices (2D ou plus)

Si `dimx, dimy, ...` sont les dimensions du vecteur 1D ou du tableau (2D ou plus) :

`Tab=bytarr(dimx,dimy,...)` définit une matrice de type octet (type BYTE)

`Tab=intarr(dimx,dimy,...)` définit une matrice de type entier court (type FIX)

`Tab=lonarr(dimx,dimy,...)` définit une matrice de type entier long (type LONG)

`Tab=fltarr(dimx,dimy,...)` définit une matrice de type réel (type FLOAT)

`Tab=dblarr(dimx,dimy,...)` définit une matrice de type réel (type DOUBLE)

`Tab=complexarr(dimx,dimy,...)` définit une matrice de type complexe (type COMPLEX)

`Tab=dcomplexarr(dimx,dimy,...)` définit une matrice de type complexe (type DCOMPLEX)

Attention aux indices d'adressage d'un tableau qui partent toujours de 0:

`Tab(x,y)` est défini pour $0 \leq x \leq \text{dimx}-1$ et $0 \leq y \leq \text{dimy}-1$

Ces fonctions sont dynamiques, contrairement au FORTRAN. Les dimensions `dimx`, `dimy`,... peuvent être des variables résultant d'un calcul, et les tableaux peuvent être créés et dimensionnés en cours de programme. Les éléments sont initialisés à 0 lors de la création.

Informations sur une variable ou un tableau

`help`, `tab` fournit les dimensions du tableau et son type de données

Règles d'adressage des éléments d'un tableau et règles de redimensionnement

`Tab` ou `Tab(*,*)` représente la totalité du tableau de valeurs
`Tab(50:100,*)` est constitué des colonnes 50 à 100 et de toutes les lignes
`Tab(50:100,25 :*)` est constitué des colonnes 50 à 100 et des lignes à partir de la ligne 25
`Tab(50,*)` est un vecteur qui n'adresse que la colonne numéro 50
`Tab(x,y)` est l'élément situé à l'intersection de la colonne `x` et de la ligne `y`

Exemples :

`T = 2.* Tab` définit un nouveau tableau `T` égal au double du tableau `Tab`, de même dimension.
`T = Tab(30:55,150:200)` définit un nouveau tableau `T` contenant les colonnes 30 à 55 et les lignes 150 à 200 de `Tab`. Les dimensions de `T` sont donc 26 x 51. Les éléments de `T` sont donc numérotés de 0 à 25 (colonnes) et de 0 à 50 (lignes).
`T = Tab(*,10)` est un vecteur qui contient la ligne 10 de `Tab`.
`T = Tab(15,23)` est un scalaire

Remarque : contrairement à FORTRAN, IDL est concis. Il ne faut surtout pas utiliser de boucles sur les indices pour faire ces opérations.

Lorsqu'on extrait d'un tableau à `n` dimensions un sous tableau de dimensions `n-1` ou moins, la fonction **reform** permet de supprimer la (les) dimension(s) devenue(s) inutile(s). Exemple :

`Tab = fltarr(10,20)`

`T = Tab(*,2)` va créer un tableau `T` de dimensions (10,1) correspondant à la 2^{ème} ligne de `Tab`.

IDL n'a pas supprimé la seconde dimension devenue inutile.

`T = reform(T)` réorganise `T` en supprimant la dimension inutile; `T` devient donc un vecteur de dimension (10)

Créer un tableau en l'initialisant

Les fonctions :

`indgen (dimx, dimy,...)` pour entiers courts

`lindgen (dimx, dimy,...)` pour entiers longs

`findgen (dimx, dimy,...)` pour réels 32 bits

`dindgen (dimx, dimy,...)` pour réels 64 bits

génèrent des matrices de dimension (dimx, dimy,...) et les initialisent avec les valeurs de leurs indices. Exemple :

`Tab = findgen(100)`

Génère un vecteur de 100 valeurs réelles initialisées de 0. à 99. par pas de 1.

Connaître les dimensions d'un tableau lu dans un fichier : fonction size

`dim = size(Tab)` retourne un vecteur `dim`

`Ndim = dim(0)` donne le nombre de dimensions du tableau

`Dimx = dim(1)` donne la première dimension, etc...

Dimy = dim(2)
Dimz = dim(3) ...

Opérations arithmétiques courantes : +, -, *, /, ^ (élévation à la puissance)

Ces opérations courantes fonctionnent sur des scalaires ou sur des matrices, dans le cas des matrices les opérations sont réalisées **terme à terme**.

Si a et b sont deux matrices de dimensions et nombre d'éléments identiques, alors l'opération $c = a * b$ réalise un produit terme à terme, et non pas un produit matriciel. Il ne faut surtout pas utiliser de boucle comme en FORTRAN !

Multiplication matricielle :

$a = b ## c$ est la multiplication matricielle lignes par colonnes des matrices b et c, à condition que le nombre de colonnes de b soit égal au nombre de lignes de c

Fonctions diverses

`abs(a)` donne la valeur absolue ou le module de a (si a complexe)

`real_part(a)` donne la partie réelle de a si a complexe

`imaginary(a)` donne la partie imaginaire de a si a complexe

Ces opérations sont réalisées terme à terme si a est une matrice, et le résultat est une matrice

`max(a)` donne la valeur maximale de la matrice a

`min(a)` donne la valeur minimale de la matrice a

`mean(a)` donne la valeur moyenne de la matrice a

`moment(a)` donne les moments de la matrice a

Opérateurs logiques et de comparaison

AND, OR, EQ (=), NE (≠), LE (≤), LT (<), GE (≥), GT (>) *comme en FORTRAN*

Il faut y ajouter les opérateurs `<=`, `<`, `>`, `>=`, qui permettent d'affecter à une nouvelle variable la plus grande ou la plus petite de deux valeurs comparées. Exemple :

$a = b > c$ renvoie dans a la plus grande des deux valeurs b et c

$a = b < c$ renvoie dans a la plus petite des deux valeurs b et c

Les opérateurs peuvent s'appliquer aux matrices. Par exemple, si tab est un tableau de valeurs, écrire $tab = tab > 10$ signifie que l'on affecte terme à terme à tab « la plus grande valeur entre 10 et tab », donc toute valeur inférieure à 10 est mise à 10 dans tab.

Fonction where

Cette fonction évite la réalisation de tests bouclant sur les indices d'un tableau, qui sont extrêmement pénalisants sous IDL. La fonction where renvoie les indices de tableau dans un vecteur lorsque le test qui suit la fonction est vérifié :

$x = where(\text{condition logique sur un tableau}, \text{count})$

Renvoie dans le vecteur x les positions (indices) où le test est concluant et dans count (option facultative) le nombre d'occurrences concluantes

Exemple :

$x = where(tab LT 10.)$

Renvoie dans le vecteur x les positions où les éléments de tab sont inférieurs à 10, tab est ici un tableau de dimensions quelconques. On peut prendre ensuite une action comme $tab(x) = 50$.

Les chaînes de caractères

`a = 'Voici une chaîne'` définit une chaîne de caractères
`a = ''` est une chaîne de longueur nulle
`a = b + c` effectue la concaténation des deux chaînes a et b
`a = strarr(dim)` définit un vecteur de chaînes de caractères de dimension dim

Quelques fonctions indispensables :

`len = strlen(a)` renvoie la longueur de la chaîne de caractères a

`a = strupcase(a)` convertit la chaîne a en majuscules

`pos = strpos(a,b)` renvoie la position de la sous chaîne b dans la chaîne a ; si b n'est pas une sous chaîne de a, on obtient -1

`b = strmid(a, n, len)` extrait len caractères de la chaîne a à partir de la position n. Le premier caractère de a est en position n=0.

`Strput,a,b,pos` place la chaîne de caractères b à la position pos dans la chaîne a en écrasant ce qui peut exister en dessous. Le premier caractère de a est en position 0.

`a = strtrim(a)` élimine tous les caractères blancs inutiles en fin de chaîne

Les structures de contrôle

- boucle FOR

```
for i=deb,fin,step do begin
  instructions
endfor
```

- boucle WHILE

```
while (test logique) do begin
  instructions
endwhile
```

Attention! Le test logique (initialisé à TRUE sinon on n'entre pas dans la boucle) doit être modifié à l'intérieur pour devenir à un moment donné FALSE, sinon on ne peut pas sortir !

- boucle REPEAT

```
repeat begin
  instructions
endrep until (test logique)
```

Attention! Le test logique est exécuté au moins une fois. Il doit être modifié dans la boucle, et devenir TRUE à un moment donné pour pouvoir en sortir !

- test IF simple

```
if (test logique) then begin
  instructions
endif
```

- test IF/ELSEIF

```
if (test logique) then begin
  instructions
endif
else begin
  instructions
endelse
```

- structure CASE simple

```
case (expression) of
valeur1: instruction
valeur2: instruction
endcase
```

- structure CASE/ELSE

```
case (expression) of
valeur1: instruction
valeur2: begin
    instructions
    end
valeur3: instruction
else: begin
    instructions
    end
endcase
```

- GOTO, label

Cette instruction est à employer le moins possible car détruit la programmation structurée. Elle est néanmoins parfois utile pour sortir d'une boucle FOR.

```
Goto, label
instructions
label :
instructions
```

Entrées/sorties clavier/écran

- afficher une valeur var à l'écran : `print, var`
- saisir une valeur var au clavier : `read, var`

Entrées/sorties fichiers de données

En astronomie, les fichiers de données suivent un standard très stricte, le FITS (Flexible Image Transport System). Un fichier FITS se compose d'un en tête (ou HEADER) qui décrit les données sous forme de blocs de 2880 caractères, à l'aide de mots clef. Les mots clef de base sont :

BITPIX : le nombre de bits par pixel (exemple 16 si nombres entiers courts)

NAXIS : le nombre de dimensions

NAXIS1 : le nombre de pixels dans la première dimensions etc...

NAXIS2

NAXIS3...

L'en tête se termine toujours par le mot clef END.

Il est suivi par les données elles mêmes, sous forme de tableau de valeurs. Les fichiers FITS peuvent être constitués de nombres entiers courts, longs, réels flottants, sous forme de tableaux 1D, 2D, 3D, etc...

Les procédures de lecture/écriture de fichiers FITS ne font pas partie intégrante d'IDL : elles figurent dans IDLASTRO, en principe vous devez pouvoir les utiliser de façon transparente si votre !PATH comprend le chemin d'accès à IDLASTRO (voir plus loin pour le !PATH).

- lister un répertoire de fichiers FITS

```
fich = findfile('chemin/*.fts', count = nf)
```

*renvoie tous les noms de fichiers de type *.fts trouvés dans le répertoire vers le vecteur chaîne fich dont la dimension vaudra nf, nombre de fichiers présents dans le répertoire exploré. Fich est du type STRARR et a pour dimension nf.*

- lire un fichier FITS

```
tab = readfits('filename')
```

```
ou tab = readfits('filename', header)
```

range le contenu du fichier dans le tableau tab, et son en tête dans la variable header (chaîne de caractères), dont la lecture est optionnelle.

Les données sont souvent des entiers courts issus de caméras CCD (16 bits, -32767 à +32767). Il est recommandé avant d'envisager tout traitement de les convertir tout d'abord en entiers non signés (0 à 65535) par la fonction :

`tab=uint(tab)`

puis de les convertir en nombres flottants (nombres réels 32 bits) par la fonction :

`tab=float(tab)`

On obtient les dimensions du tableau par la fonction :

`dim = size(tab)`

`nd = dim(0)` donne le nombre de dimensions

`dimx = dim(1)` donne la première dimension, etc...

`dimy = dim(2)`...

Ou encore, `HELP, tab` fournira les mêmes informations sur `tab` (type, dimensions).

- écrire un nouveau fichier FITS

`writefits, 'filename', tab` écrit le tableau `tab` (2 ou 3 dimensions) dans le fichier 'filename'

- fichiers JPEG 8 bits

Ce type de fichier est restreint à 8 bits, de sorte que toute image devra être recadrée sur 8 bits en utilisant la commande `bytsc1` de deux manières possibles:

a) `Image=bytsc1(tab,min=valeur_min,max=valeur_max)`

Si `tab(x,y) < valeur_min` alors la fonction `bytsc1` retourne 0

Si `tab(x,y) > valeur_max` alors la fonction `bytsc1` retourne 255

Entre `valeur_min` et `valeur_max`, la fonction `bytsc1` a une réponse linéaire de 0 à 255

b) `Image=bytsc1(tab)`

Dans ce cas, la fonction `bytsc1` calcule elle-même automatiquement `valeur_min` et `valeur_max` à partir des valeurs de `tab`

Écriture JPEG :

`Write_jpeg, 'filename', image`

Inversement, on lit une image JPEG 8 bits par:

`Read_jpeg, 'filename', image`

- fichiers PNG 8 bits

Ce type de fichier est également restreint à 8 bits, de sorte que toute image devra être recadrée sur 8 bits en utilisant la commande `bytsc1` comme pour le JPEG

Écriture PNG :

`Write_png, 'filename', image`

Inversement, on lit une image PNG 8 bits par:

`Read_png, 'filename', image`

- fichiers binaires de sauvegarde réutilisables sous IDL uniquement (très utile)
création d'un nouveau fichier avec sauvegarde des variables de tous types `var1, var2`...qui peuvent être des tableaux de données :

`Save, var1, var2, filename='filename'`

Récupération des variables dans une session IDL ultérieure (elles porteront le même nom, auront le même type et les mêmes dimensions que lors de la sauvegarde):

`Restore, 'filename'`

Ré-échantillonnage des tableaux

- ré-échantillonnage d'un tableau de dimensions initiales `dimx, dimy` par sommation des pixels
`tab = rebin(tab,xdim,ydim)`

attention ! `xdim, ydim` sont des sous multiples de `dimx, dimy` (procéder à un redimensionnement si ce n'est pas le cas avant de ré-échantillonner). Les pixels étant regroupés, la fonction `rebin` travaille par sommation et améliore le rapport signal/bruit de l'image initiale.

- ré-échantillonnage d'un tableau de dimensions initiales `dimx, dimy` par interpolation
`tab = congrid(tab,xdim,ydim)`

*x*dim, *y*dim sont quelconques (plus grands ou plus petits que *dimx*, *dimy*), donc la fonction *congrid* travaille par interpolation, et entraîne une perte de qualité : utiliser *rebin* quand l'amélioration du rapport S/B est requise.

Intégration selon les lignes ou les colonnes d'un tableau : fonction *total*

tabx = *total*(*tab*,1) intègre dans le sens de la première dimension
taby = *total*(*tab*,2) intègre dans le sens de la seconde dimension, etc...
tabt = *total*(*tab*) somme tous les éléments du tableau

Rappel : valeur moyenne d'un tableau

tabmoy = *mean*(*tab*) est identique à *total*(*tab*)/(*dimx* * *dimy*) si *dimx*, *dimy* sont les dimensions de *tab*

Visualisation d'images

- créer une fenêtre de dimensions (*dimx*, *dimy*)

window, *n*, *xs=dimx*, *ys=dimy* créer la fenêtre numéro *n* de la dimension du tableau

n est le numéro de la fenêtre (à partir de 0)

sélectionner la fenêtre *m* : *wset*, *m*

détruire la fenêtre *m* : *wdelete*, *m*

- visualiser un tableau de dimensions (*dimx*,*dimy*) en niveaux de gris

window, *n*,*xs=dimx*, *ys=dimy*

*tv**scl*, *tab* ou *tv*,*bytscl*(*tab*) pour voir un tableau à 2 dimensions

*tv**scl*, *tab*(*,*,*i*) ou *tv*, *bytscl*(*tab*(*,*,*i*)) pour voir le *i*-ème plan d'un tableau à 3 dimensions

tv,*bytscl*(*tab*, *min=valeur_min*, *max=valeur_max*) pour contraindre l'affichage des valeurs entre les valeurs *min* et *max* spécifiées par le programmeur. Dans ce cas :

Si *tab*(*x*,*y*) < *valeur_min* alors la fonction *bytscl* retourne 0

Si *tab*(*x*,*y*) > *valeur_max* alors la fonction *bytscl* retourne 255

Entre *valeur_min* et *valeur_max*, la fonction *bytscl* a une réponse linéaire de 0 à 255

Quand on utilise *tv**scl*, *tab* (ou *tv*, *bytscl*(*tab*), ce qui est strictement équivalent), la fonction

bytscl calcule elle même automatiquement les *valeur_min* et *valeur_max* à partir des valeurs de *tab*.

- fausses couleurs en mode affichage 8 bits

Une palette de fausses couleurs peut être appliquée à la visualisation, pourvu qu'on soit en affichage 8 bits (256 couleurs), par :

Chargement de la palette IDL prédéfinie numéro *n* :

loadct, *n*

(*n* = 0 donne des niveaux de gris)

Choix interactif d'une palette prédéfinie par IDL:

xloadct

(fait apparaître la fenêtre interactive présentée ci contre) →

Lire une palette pour la modifier :

(rend 3 vecteurs rouge, vert, bleu de type BYTE):

*tv**lct*,*rouge*,*vert*,*bleu*,/*get*

Charger une nouvelle palette utilisateur :

(3 vecteurs BYTE rouge, vert, bleu de dimension 256):

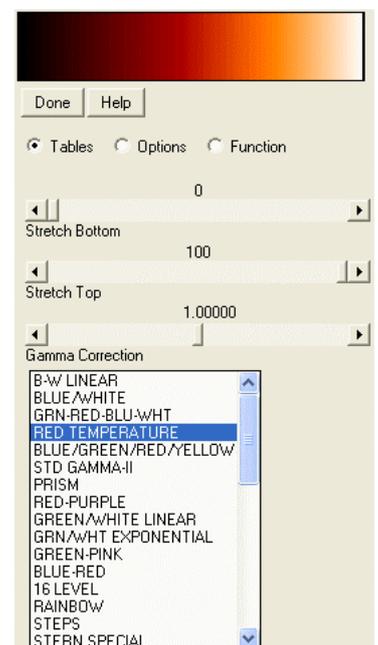
*tv**lct*,*rouge*,*vert*,*bleu*

- relever la position du curseur dans la fenêtre courante

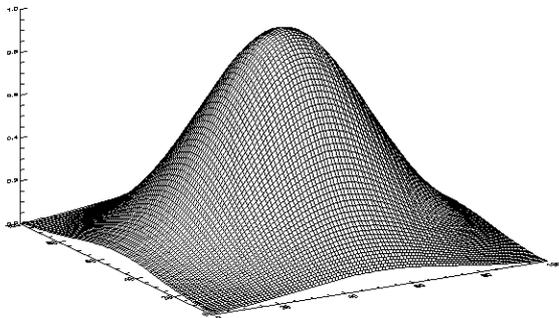
wset,*n* sélectionner la fenêtre numéro *n*

cursor, *x*, *y*, /*device* renvoie les coordonnées (*x*,*y*)

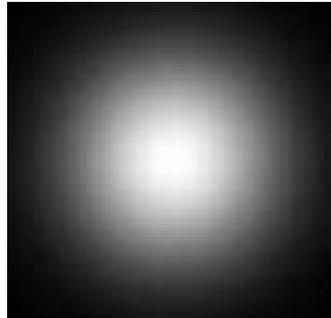
Fenêtre *xloadct* →



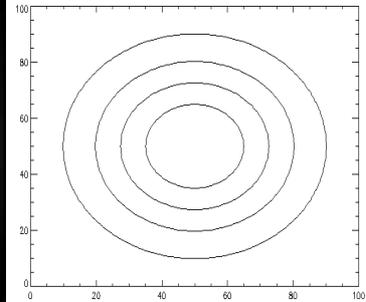
- Isocontours ou lignes de niveau d'un tableau
`Contour,tab` pour un tableau à 2 dimensions
`Contour,tab(*,*,i)` pour le *i*-ème plan d'un tableau à 3 dimensions
- Visualisation en 3D sous forme d'une surface composée d'un grillage
`Surface,tab` pour un tableau à 2 dimensions
`Surface,tab(*,*,i)` pour le *i*-ème plan d'un tableau à 3 dimensions



fonction surface



fonction tvscl



fonction contour

- Superposition d'isocontours sur une image en niveaux de gris
`window,0,xs=dimx,ys=dimy`
`Image_cont,tab`
- Visualisation en 3D sous forme d'une surface continue ombrée
`Shade_surf,tab` pour un tableau à 2 dimensions
`Shade_surf,tab(*,*,i)` pour le *i*-ème plan d'un tableau à 3 dimensions

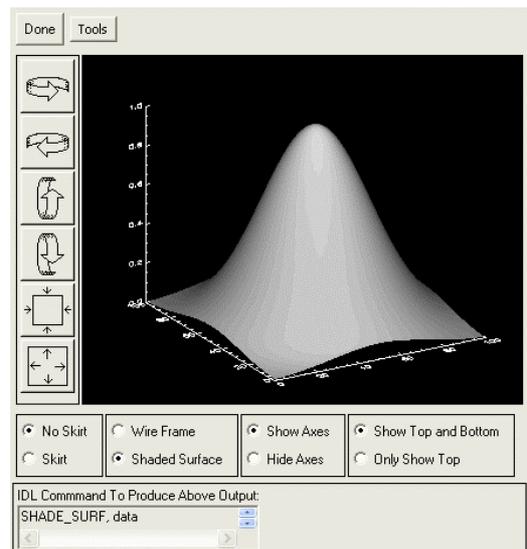
Commandes de visualisation évoluées

- Commande `xsurface, tab`

Visualisation en 3D d'une surface.
 Agit sur un tableau 2D avec action interactive sur la représentation dans l'espace de la fonction $z = \text{tab}(x,y)$

- Commande `xinteranimate`

Animation des plans d'un tableau 3D avec action interactive sur certains paramètres.
 On visualise successivement les plans du tableau $\text{tab}(x,y,z)$ pour z variant.



Fenêtre xsurface

Exemple :

On lit un fichiers FITS 3D représentant des spectres de la raie H alpha en fonction du temps et on passe en boucle les plans successifs du tableau 3D $\text{tab}(\text{lambda},x,t)$ représentant l'intensité, et dont les 3 dimensions sont longueur d'onde, espace et temps.

`tab=readfits('filename')`

On convertit `tab` en entiers non signés puis en réels flottants

`tab=uint(tab)`

`tab=float(tab)`

On détermine les dimensions du tableau `tab`

```
dim=size(tab)
dimx=dim(1)
dimy=dim(2)
dimz=dim(3)
```

Initialiser XINTERANIMATE:

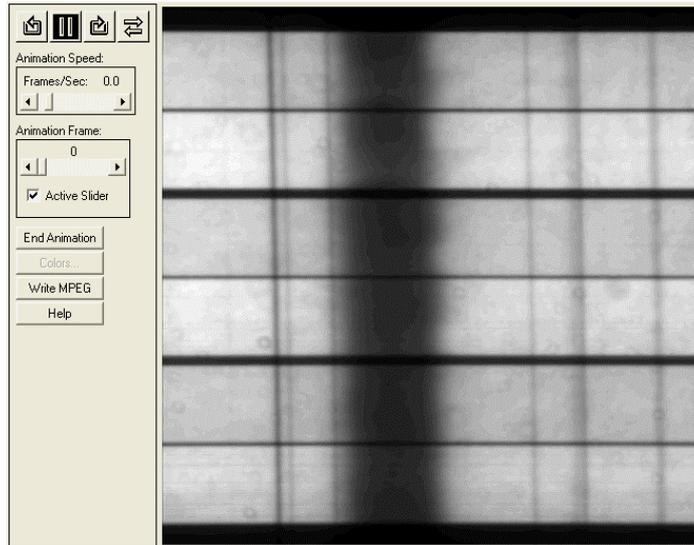
```
XINTERANIMATE, SET=[dimx, dimy, dimz], /SHOWLOAD, $ MPEG_QUALITY=100
```

Charger les images dans XINTERANIMATE:

```
FOR I=0,dimz-1 DO XINTERANIMATE, FRAME = I, IMAGE = bytscl(tab[*,* ,I])
```

Jouer l'animation:

```
XINTERANIMATE, 10, /KEEP_PIXMAPS
```



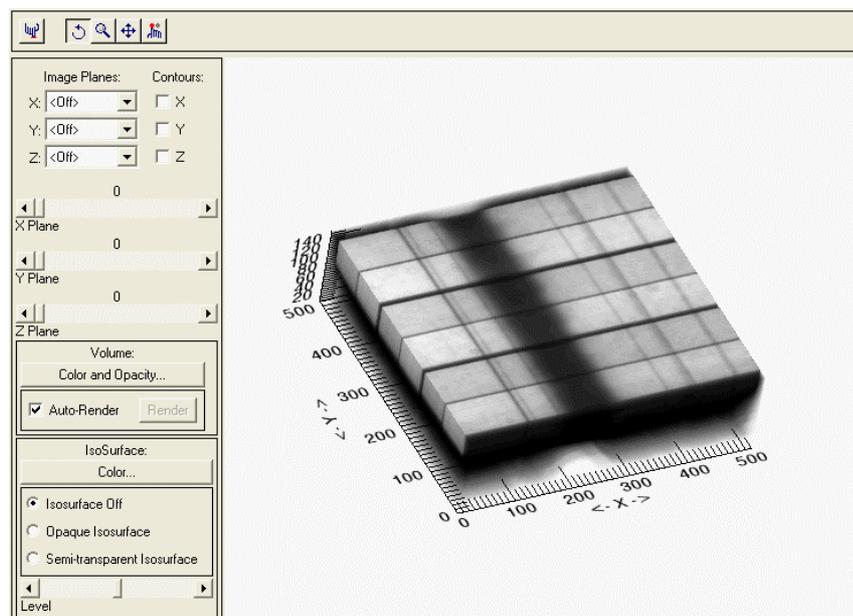
Fenêtre xinteranimate (ici tab contient des spectres de la raie H alpha en fonction du temps)

- Commande xvolume, tab

Visualisation en 3D d'un volume. Agit sur un tableau 3D avec action interactive sur la représentation dans l'espace du volume tab(x,y,z)

Fenêtre xvolume →

(ici tab contient des spectres de la raie H alpha en fonction du temps. Le tableau tab représente des intensités dont les 3 dimensions sont longueur d'onde, espace et temps)



Représentation graphique d'une fonction $y = f(x)$

Les abscisses et ordonnées sont contenues dans les vecteurs x et y .

`window, n, xs=640, ys=480` créer la fenêtre numéro n de la dimension souhaitée

`plot,x,y` où x = vecteur des abscisses, y = vecteur des ordonnées

`plot,tab(*,y)` pour faire une coupe horizontale d'ordonnée y dans un tableau tab

`plot,tab(x,*)` pour faire une coupe verticale d'abscisse x dans un tableau tab

Quelques options intéressantes :

`plot,x,y,linestyle=ls, thick=épaisseur,color=couleur,background=couleur, $`

`xrange=[xmin,xmax] ,yrange=[ymin,ymax], psym=symbol, $`

`xtitle='Label sous axe Ox',ytitle='Label sous axe Oy',title='Titre du graphique', $`

`charsize=taille,charthick=épaisseur`

avec :

`ls = type de trait : 0 — 1 2 ----- 3 -.-.-.- 4 -...-...- 5 - - -`

`symbol = 0 (aucun symbole) 1 (+) 2 (*) 3 (.) 4 (◇) 5(Δ) 6(□) 7(x)...`

épaisseur du trait ou des caractères: nombre décimal (1 par défaut)

couleur : ne fonctionne correctement qu'en affichage 8 bits, dans ce cas couleur prend une valeur entre 0 et 255 qui adresse la couleur d'une palette chargée par `loadct` ou par `xloadct`.

On peut tracer plusieurs courbes sur le même graphique. Dans ce cas, la première courbe sera tracée avec la fonction `PLOT`. Les suivantes le seront avec la fonction `OPLLOT` :

`plot, x, y, linestyle=0, xtitle='Label sous axe Ox', ytitle='Label sous axe Oy'`

`oplot, x, y1, linestyle=1`

`oplot, x, y2, linestyle=2`

On peut aussi tracer plusieurs graphiques dans une même fenêtre, en la divisant en N_{col} colonnes \times N_{lign} lignes. A chaque nouvelle instruction `PLOT`, on passe à la division suivante de la fenêtre.

`!P.multi = [0, Ncol, Nlign]`

Pour revenir au mode standard, faire `!P.multi = 0`

Sortie des graphiques en format POSTSCRIPT pour impression

Avant l'exécution des instructions graphiques, il faut spécifier le périphérique de sortie sous la forme d'un fichier postscript, car la sortie se fait sur l'écran par défaut :

`Set_plot,'PS'`

`Device,filename='nom_du_fichier.ps',bits=8,/color`

Instructions graphiques

`Device,/close`

L'impression du fichier se fait à l'aide d'une commande (hors IDL) du système d'exploitation.

On revient ensuite à l'affichage sur l'écran « X window » par :

`Set_plot,'X'`

Les fonctions utilisateur

Function `function_name (var1, var2...)`

Instructions

Return, var

End

La fonction reçoit en entrée les variables `var1, var2, ...` et fournit en sortie une unique variable

var (contrairement aux procédures PRO qui peuvent retourner plusieurs variables)
Appel: `y = function_name (var1, var2, ...)`

Exemple: `Function sinc (x)
 Var = sin(x)/x
 Return, var
End`

Les fonctions peuvent être écrites et compilées dans des fichiers indépendants (d'extension .pro), ou placées à la suite du programme principal, ou encore devant.

Il est possible de partager une grande quantité de variables entre les procédures et les fonctions sans les faire passer en argument en utilisant l'instruction COMMON :

`Common nom, var1, var2, var3,varN`

L'instruction `Common nom, variables` doit être placée identiquement dans chaque procédure ou fonction qui souhaite partager les variables. Chaque zone commune est identifiée par un nom.

Caractères spéciaux dans une procédure

`$` est le caractère de continuation d'une ligne à la ligne suivante. Un Exemple:

`Print, var1, var2, $
var3, var4`

`&` est le caractère qui sépare deux instructions tapées sur la même ligne. Exemple :

`x = 12.5 & y = -34.6 & z = 56.9`

`;` est le caractère qui précède un commentaire. Exemple :

`x = 10.5 ; on initialise la variable x`

Documentation en ligne : taper sur le prompt IDL le caractère **?**

Variable d'environnement !PATH

Cette variable permet de spécifier un chemin de recherche de procédures spécifiques à chaque utilisateur, le chemin par défaut étant le répertoire courant. Exemple :

`!PATH = '/usr/project/myprograms:' + !PATH`

Cette instruction ajoute au chemin de recherche standard l'exploration du répertoire utilisateur :
`/usr/project/myprograms`

Un exemple de procédure

On souhaite, à partir de deux observations polarimétriques des combinaisons $I+Q(\lambda, x)$ et $I-Q(\lambda, x)$ contenues dans deux fichiers FITS distincts f1.fits et f2.fits pouvoir visualiser le taux de polarisation $Q/I(\lambda, x)$ puis son intégration dans le sens spatial (x) sur le Soleil.

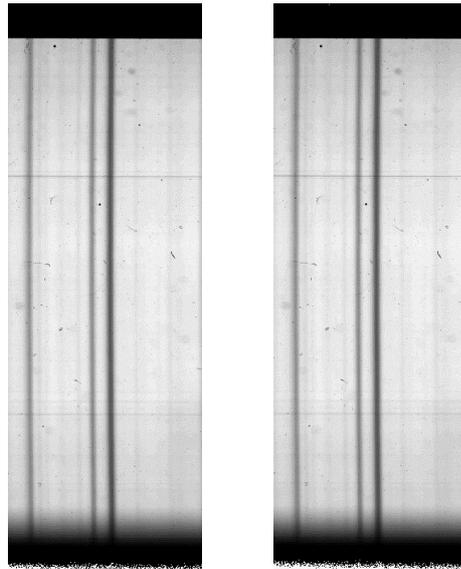
`Pro spectropol
;Auteur du programme :
;Date de création :
;Date de révision :
;Fonction réalisée par le programme :
;-----`

```

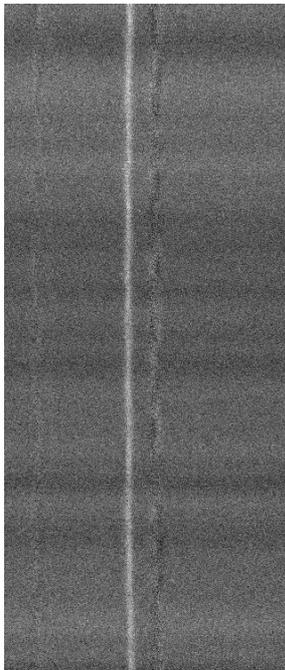
;répertoire de lecture
chemin='F:\COURS\2008\'
;lecture des deux fichiers d'entrée
t1=readfits(chemin+'f1.fits')
t2=readfits(chemin+'f2.fits')
;conversion en flottant 32 bits
t1=uint(t1) & t1=float(t1)
t2=uint(t2) & t2=float(t2)
;calcul de l'image Q/I(lambda,x)
i=t1+t2
q=t1-t2
qsuri=q/i
;taille de l'image Q/I(lambda,x)
dim=size(qsuri)
dimx=dim(1) & dimy=dim(2)
;couper 100 lignes en haut et en bas
qsuri=qsuri(*,100:dimy-100)
dimy=dimy-200
;visualisation de l'image Q/I(lambda,x)
window,0,xs=dimx,ys=dimy
tvsc1,qsuri
;intégration dans le sens spatial de Q/I(lambda,x)
qi=total(qsuri,2)
;moyenne
qi=qi/float(dimy)
;visualisation graphique de l'intégration Q/I(x)
window,1
plot,qi,xtitle='Wavelength (pixels)',ytitle='Q/I integrated',xstyle=1
end

```

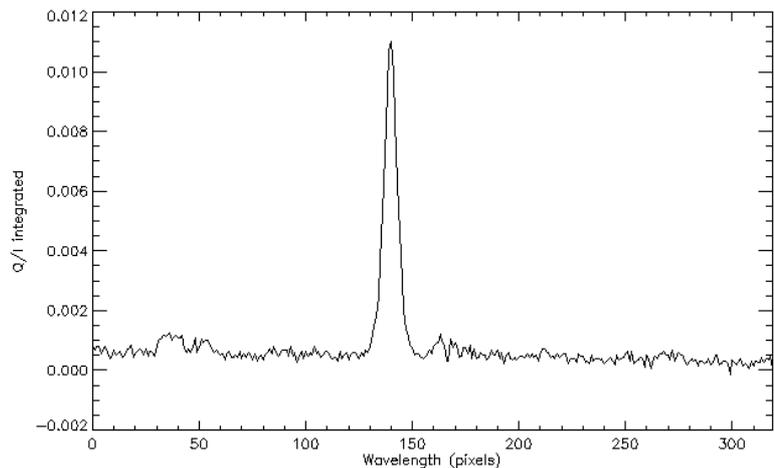
en abscisse : λ ; en ordonnée : x



spectres *f1.fits (I+Q)* et *f2.fits (I-Q)*



$\leftarrow Q/I(\lambda, x)$

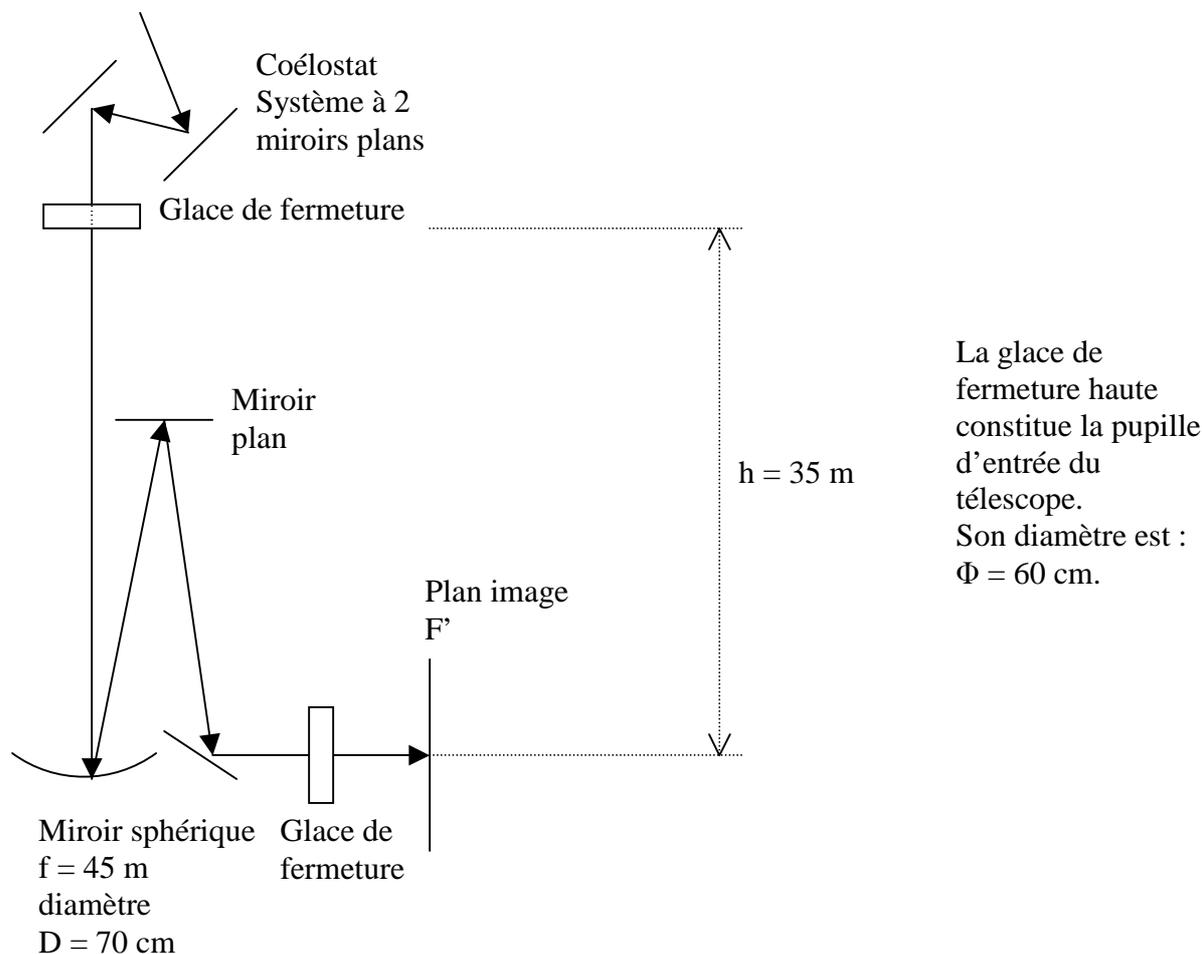


$Q/I(\lambda)$ intégré dans le sens spatial, en fonction de la longueur d'onde

**Travaux pratiques d'observation
à la Tour Solaire de Meudon**

TP d'optique solaire à la Tour Solaire de Meudon

I - Tour solaire de Meudon : compréhension du télescope



- 1) *Quel est le diamètre de l'image solaire au foyer F' ?*
- 2) *Que vaut 1 seconde d'arc en mm dans l'image au foyer F' ?*
- 3) *Quel est le pouvoir de résolution du télescope (on donne $\theta = 1.22 \lambda/D$ diamètre angulaire de la tache de diffraction d'une ouverture circulaire de diamètre D)*

II - Tour solaire de Meudon : compréhension du spectrographe

Il est constitué d'un miroir collimateur (f = 14 m) au foyer objet duquel se trouve la fente d'entrée du spectrographe et d'un objectif de chambre de distance focale identique.

Réseau : 300 traits/mm, angle de blaze $63^{\circ}26'$, dimension 13 x 25 cm environ

- 1) *Où se trouve l'image de la pupille d'entrée du télescope par le collimateur ?*
- 2) *Quelle est sa dimension sur le réseau ?*

Application de la formule des réseaux : $\sin(i) + \sin(i') = k \lambda / d$

k est appelé ordre d'interférence, et d se déduit du nombre de traits au mm (1/d) donné par le fabricant. En général, i et i' sont voisins de l'angle de blaze.

- 3) *On s'intéresse aux raies NaD1/D2 à 589.6/589.0 nm et H α 656.3 nm. Dans quel ordre d'interférence travaille t-on ?*
- 4) *Quelle est la dispersion du spectrographe en Angströms/cm ($\text{Å}/\text{cm}$ pour ces raies ?*

5) Quelle est la résolution spectrale du spectrographe $\Delta\lambda$ en mÅ pour une fente d'entrée de 1 arc sec = 220 μ de largeur pour ces raies ?

III - Suggestions d'observations

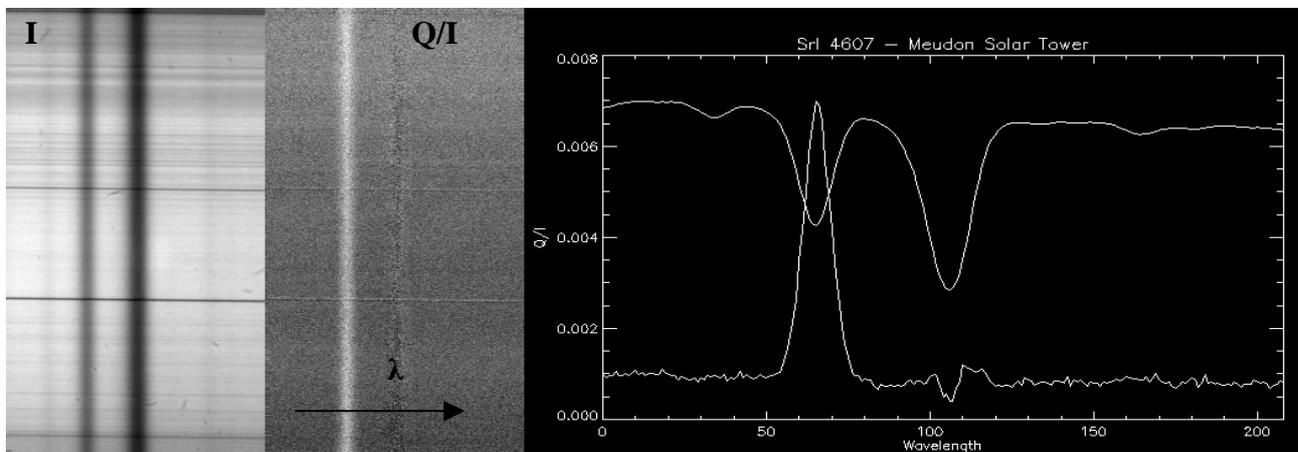
III - 1 - Effet Doppler (acquisition de données pour TD1)

On fera l'acquisition de quelques spectres de la raie H α et on mesurera l'effet Doppler en se basant sur le TD numérique n°1 (centre de gravité de la raie).

III - 2 - Polarisation linéaire de diffusion au limbe (acquisition de données pour TD2)

On fera l'acquisition, fente du spectrographe tangente au limbe, à environ 10 arc sec du bord, d'une cinquantaine de couples spectres I+Q et I-Q de la raie SrI 460.7 nm et on mesurera la polarisation linéaire de diffusion de cette raie en se basant sur le TD numérique n°2.

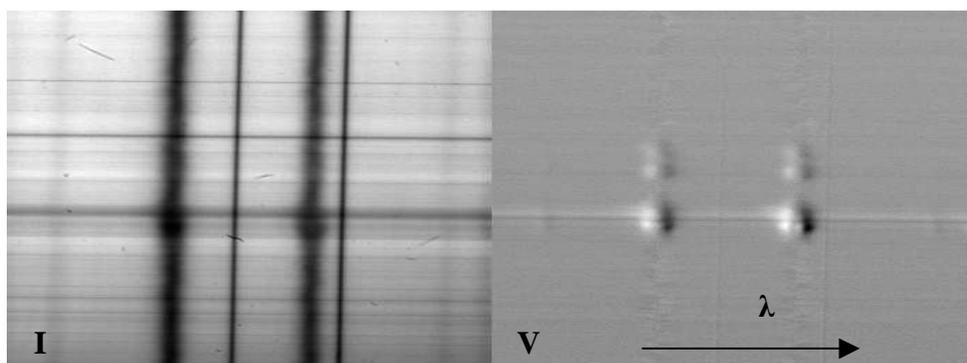
Les tensions à appliquer au polarimètre sont : **18469** pour I+Q (retard ONDE) et **24881** pour I-Q (retard 1/2 ONDE). On fera une **rafale de 50 couples**.



polarisation Q/I de la raie SrI 460.7 nm

III - 3 - Effet Zeeman sur une tache solaire : détermination du champ magnétique longitudinal (acquisition de données pour TD6, première partie)

On fera l'acquisition, sur une tache, d'une vingtaine de couples spectres I+V et I-V des raies FeI à 630.2 nm et on mesurera les champs magnétiques en se basant sur le TD numérique n°5, partie 1. Les tensions à appliquer au polarimètre sont : **16978** pour I+V (retard 3/4 ONDE) et **11940** pour I-V (retard 5/4 ONDE). On fera une rafale de **20 couples** pour sélectionner le meilleur couple en visualisant la qualité du signal V obtenu par **différence** des images I+V et I-V (exemple ci dessous).

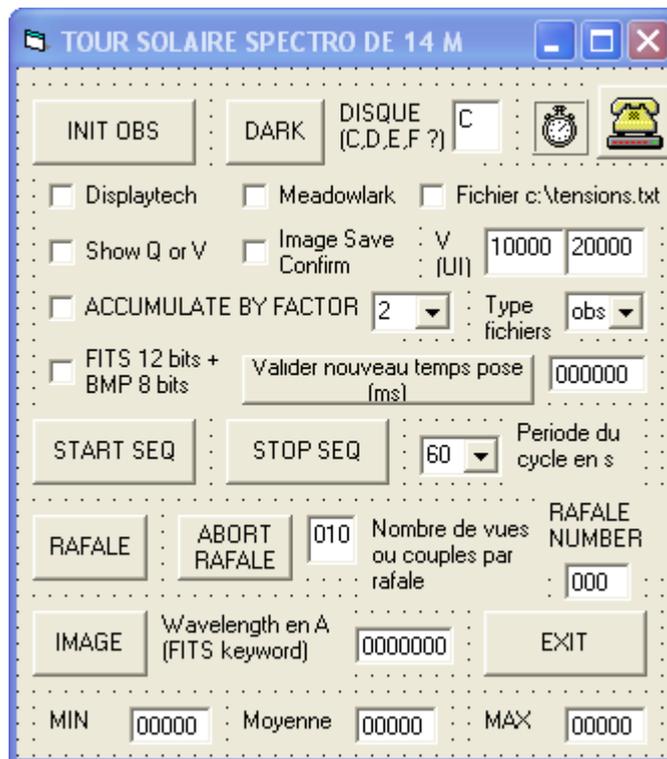


polarisation V/I des raies FeI 630.1 et 630.2 nm

ANNEXE 1

CCD refroidi Princeton Instruments

**acquisition de données en spectro polarimétrie au
spectrographe de la Tour Solaire de Meudon
à l'aide du polarimètre à cristal liquide**



Caractéristiques principales du CCD :

Pixels de 9 μ , format 1536 x 1024 fenêtrable, binning programmable, puits de potentiel de 85000 électrons, 1MHz (lecture de 10⁶ pixels/s), 12 bits (4094 niveaux)

Refroidissement double étage Peltier + circuit d'eau (< -15° C)

Obturbateur mécanique (temps de pose minimum de 30 ms)

Programme de pilotage constructeur: WinView

Le temps de pose, le fenêtrage du capteur, le « binning » se règlent sous WinView qu'il faut lancer en premier et conserver actif en permanence ; Winview assure l'affichage des images

Routine d'acquisition spécifique à la Tour Solaire de Meudon :

Paramètres d'acquisition à choisir :

DISQUE : les données seront enregistrées dans un sous répertoire créé dans ce volume et identifié par la date d'observation de la forme JJMAA (jour, mois, année) : disque C par défaut

Displaytec : sélectionner si on utilise ce polarimètre

Meadowlark : sélectionner si on utilise ce polarimètre (cas le plus fréquent)

Dans ce dernier cas (Meadowlark), les deux tensions seront lues dans le fichier c:\tensions.txt si cette option est sélectionnée ; dans le cas contraire, les tensions seront lues dans les deux cases du dessous. C'est la valeur des tensions qui détermine si l'on observe $\pm Q$ ou $\pm V$. Les tensions sont données en unités internes (UI). Un tableau pour chaque raie est disponible à la Tour. Les unités internes sont des milli Volts x 6.5536.

Show Q or V : actif uniquement en polarimétrie, déclenche le calcul de Q (ou de V) en temps réel par soustraction des deux polarisations suite à l'acquisition d'un couple $\pm Q$ (ou $\pm V$) ou d'une accumulation de plusieurs couples d'images $\pm Q$ (ou $\pm V$)

Dans ce dernier cas, image save confirm, si coché, pose une question avant toute écriture sur disque, de manière à ne conserver que les acquisitions satisfaisantes : comme les acquisitions des deux polarisations ne sont pas simultanées, une turbulence excessive peut en effet altérer considérablement le signal polarimétrique issu d'un seul couple.

Type fichiers : observations, ou flat field (change la syntaxe de nomination des fichiers selon le mode choisi pour mieux les reconnaître)

Accumulation : permet de sommer en temps réel une succession d'images pour augmenter le rapport S/B. Le minimum est de 2 images, le maximum est de 16 images. Avec 4 images, on obtient un résultat sur 14 bits et avec 16 images, on obtient un résultat sur 16 bits. En polarimétrie, une accumulation est obtenue par alternance des signaux de polarisation pour que deux polarisations contraires restent les plus proches dans le temps (si l'on sélectionne le facteur d'accumulation 16, la caméra acquiert 16 couples I±Q, ou I±V et effectue ensuite la sommation avant écriture sur disque).

Fits + BMP : les observations sont toujours écrites en FITS (12 à 16 bits) et, si l'on sélectionne cette option, en format BMP 8 bits pour « quick look »

Temps de pose : peut être modifié (valeur en ms) ; néanmoins, on recommande plutôt de le faire sous WinView

Initialisation de la caméra :

INIT : initialise la caméra (première commande à lancer après avoir choisi les paramètres d'acquisition ci dessus)

Observations :

Avant d'observer, on devra donner sous WinView une consigne de température (généralement – 15°C) et attendre qu'elle soit atteinte, puis sélectionner une fenêtre d'acquisition à l'intérieur de la zone 1536 x 1024 du capteur, puis le facteur de « binning » et le temps de pose. La longueur d'onde du domaine spectral observé peut être indiquée en Angströms dans la fenêtre du bas pour que le mot clef FITS « WAVELENGTH » soit correctement documenté (sinon on aura zéro). Les valeurs MIN, MOYENNE, MAX de l'observation en cours sont données en bas. Attention à la sous exposition et à la sur exposition. 12 bits représentent une saturation à 4096 niveaux, si on a choisi un facteur d'accumulation 4, on travaille en 14 bits (saturation à 16384 niveaux), un facteur d'accumulation 16 représentera 16 bits (saturation à 65536 niveaux). On recommande de se situer entre ¼ et ¾ de la saturation.

DARK : lance une séquence de courant d'obscurité avec obturateur fermé

IMAGE : lance l'acquisition d'une image, ou d'un couple d'images en polarimétrie, ou encore l'accumulation de N images (ou de N couples en polarimétrie) si l'on a sélectionné un facteur d'accumulation N. L'écriture se fait en FITS. Si l'on a demandé en polarimétrie l'affichage de la différence des deux polarisations observées, et si l'on a sélectionné l'option image save confirm, alors une question est posée avant l'écriture sur disque de l'observation, ce qui permet d'éviter d'écrire des résultats non satisfaisants (ils sont très dépendants du « seeing » en polarimétrie)

START SEQ/STOP SEQ : provoque la répétition à intervalle de temps régulier (temps de cycle choisi en s) de l'ensemble des actions faites par le bouton IMAGE décrit ci dessus

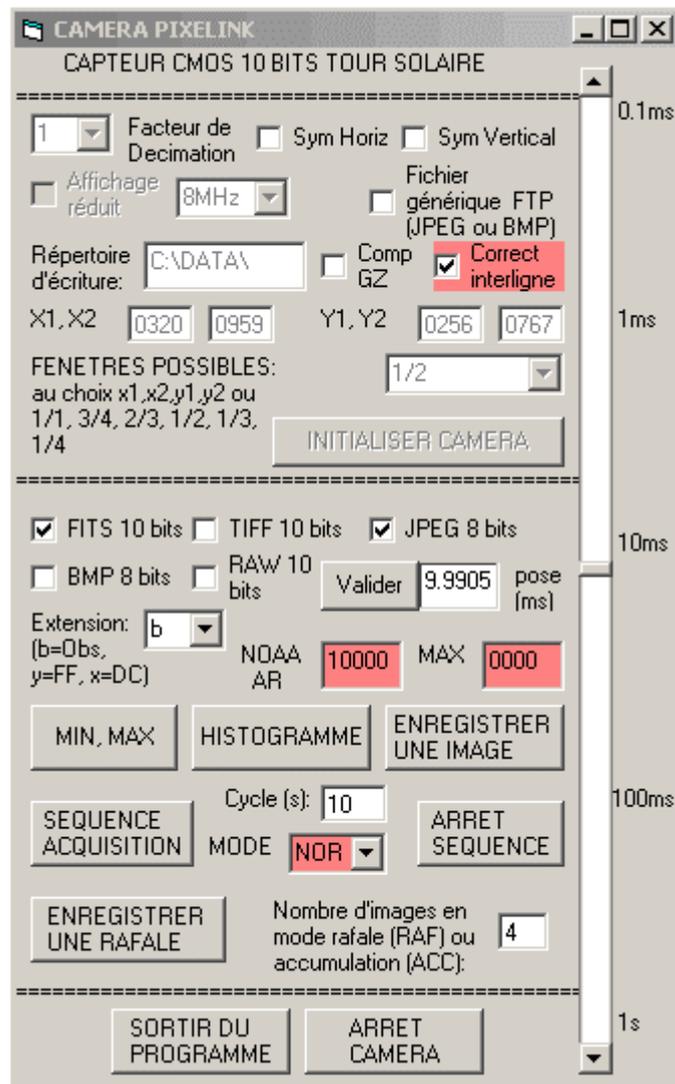
RAFALE/ABORT RAFALE : provoque la répétition le nombre de fois sélectionné de l'ensemble des actions faites par le bouton IMAGE décrit ci dessus (dès qu'une action est achevée, on enchaîne la suivante sans attendre). Une rafale peut être interrompue par le bouton ABORT en cas d'erreur. La commande ABORT n'est prise en compte qu'à la fin des actions en cours (patienter). En second spectre, on utilise systématiquement l'observation en mode RAFALE avec une centaine de couples de polarisation. Le Flat Field se fait également en mode RAFALE au centre du disque avec une ou quelques dizaines de couples d'images.

EXIT permet de sortir du programme d'acquisition.

ANNEXE 2

Caméra CMOS à imagerie Pixelink

**acquisition de données en imagerie large bande à la
Tour Solaire de Meudon**



Caractéristiques principales de la caméra :

Pixels de 6μ , format 1280 x 1024 fenêtrable, puits de potentiel de 40000 électrons, 8MHz (lecture de 810^6 pixels/s), 10 bits (1024 niveaux)

Pas de refroidissement

Obturateur électronique (temps de pose minimum de 0.1 ms)

La dimension du pixel est d'environ 0.1 arc sec sur ce ciel.

Programme d'acquisition spécifique à la Tour Solaire de Meudon

Paramètres d'acquisition à choisir:

REPERTOIRE : les données seront enregistrées dans ce répertoire

Facteur de décimation : permet de ne lire qu'un pixel sur n (1, 2, ou 4 possible) : la décimation 1 est sans effet. On gagne un facteur 4 en vitesse de lecture en décimation 2, 16 en décimation 4.

Symétrie verticale : symétrie sur l'image d'axe Oy

Symétrie horizontale : symétrie sur l'image d'axe Ox

Fréquence de lecture : ne pas changer (8 MHz optimum)

Affichage réduit : utile seulement si vous observez en plein format 1280 x 1024

Compression GZ : pour compresser en temps réel le FITS en FITS.GZ (attention ! Cette option ralentit fortement la cadence d'observation)

Correction interligne : le capteur CMOS possède une réponse légèrement différente entre les lignes paires et impaires. Cette option corrige cet effet en temps réel, mais n'est active que sur les fichiers enregistrés en FITS. Elle ralentit évidemment l'acquisition.

Fenêtrage : en mode X1,X2,Y1,Y2, les coordonnées de la fenêtre lues dans les 4 cases X1, X2, Y1, Y2 sont utilisées pour adresser le rectangle correspondant sur le capteur. Autrement, ces 4 cases sont ignorées, notamment si l'on choisit l'un des formats prédéfinis suivants : 1/1 (tout le capteur), $\frac{3}{4}$, $\frac{2}{3}$, $\frac{1}{2}$, $\frac{1}{3}$ ou $\frac{1}{4}$ du capteur. Dans chacun de ces choix prédéfinis, la fenêtre est centrée sur le capteur. Par exemple si l'on choisit le format 1/2, seule une fenêtre 640 x 512 est lue au centre. Mais, si l'on choisit le format 1/1 en décimation 2, c'est aussi une image de 640 x 512 qui sera obtenue par échantillonnage du capteur un point sur deux. Le choix de la fenêtre peut donc se combiner au facteur de décimation. Pour information, on a indiqué ci dessous les performances de lecture obtenues en format 1/1, $\frac{1}{2}$ et $\frac{1}{4}$ pour les décimations 1, 2 et 4. Il est ainsi possible d'enregistrer des RAFALES à cadence d'acquisition très élevée. La dimension et la résolution sont approximatives dans ce tableau.

Format	décimation	Taille image	Champ	Résolution	Cadence maxi
1/1	1	1280 x 1024	100''	0.1''	5 images/s
1/1	2	640 x 512	100''	0.2''	20
1/1	4	320 x 256	100''	0.4''	80
$\frac{1}{2}$	1	640 x 512	50''	0.1''	20
$\frac{1}{2}$	2	320 x 256	50''	0.2''	80
$\frac{1}{2}$	4	160 x 128	50''	0.4''	320
$\frac{1}{4}$	1	320 x 256	25''	0.1''	80
$\frac{1}{4}$	2	160 x 128	25''	0.2''	320

Initialisation :

INIT : initialise la caméra (première commande à lancer après avoir choisi les paramètres d'acquisition)

Observations :

Le numéro NOAA de la région observée peut être indiqué pour que le mot clef FITS correspondant soit correctement documenté (sinon on aura zéro). La valeur MAX de l'image en cours est donnée en permanence. Attention à la sous exposition et à la sur exposition. 10 bits représentent une saturation à 1024 niveaux. On recommande de se situer aux $\frac{3}{4}$ de la saturation vers 800 coups.

FITS : enregistre en format FITS sur 10 bits

TIFF: enregistre en format TIFF sur 10 bits

RAW: enregistre en format RAW sur 10 bits (le plus rapide à écrire sur disque)

BMP: enregistre en format BMP après dégradation à 8 bits

JPEG: enregistre en format JPEG après dégradation à 8 bits

Ces 5 formats sont combinables.

MAIS : lorsque l'on souhaite une cadence maximale, sélectionner le format RAW.

Temps de pose: il se règle avec le curseur de 0.1 ms à 1000 ms environ. On peut aussi entrer une consigne (nombre décimal en ms) que l'on validera par le bouton **VALIDER**.

MIN, MAX : fournit les valeurs MIN, MAX de l'image à la volée. La saturation du capteur est obtenue à 1023.

HISTOGRAMME : calcule un histogramme des valeurs de l'image.

ENREGISTRER IMAGE : lance l'acquisition d'une image et son écriture sur disque dans le(les) format(s) sélectionné(s).

SEQUENCE/ARRET SEQUENCE : provoque la répétition à intervalle de temps régulier (temps de cycle choisi en s) de l'acquisition d'une image (mode **NOR**mal), ou de l'accumulation de N images par sommation en temps réel (mode **ACC** basé sur le facteur d'accumulation N ou nombre d'images à sommer) ou encore d'une rafale de N images successives non sommées (mode **RAF**ale basé sur le nombre d'images par rafale N). L'écriture se fait dans le format sélectionné, utiliser le format RAW de préférence en mode **RAF**ale pour ne pas multiplier le nombre des fichiers (une rafale = un fichier 3D en mode RAW contre N fichiers dans les autres formats).

En mode **ACC**, si on a choisi un facteur d'accumulation 4, on travaille en 12 bits (saturation à 4094 niveaux) ; un facteur d'accumulation 16 représentera 14 bits (saturation à 16384 niveaux).

RAFALE: provoque l'acquisition d'une rafale de N images successives non sommées. L'écriture se fait dans le format sélectionné, utiliser le format RAW de préférence en mode **RAF** pour ne pas multiplier le nombre des fichiers (une rafale = un fichier 3D en mode RAW contre N fichiers dans les autres formats).

ARRET CAMERA : permet d'arrêter la caméra pour reconfigurer les paramètres d'acquisition, comme le facteur de décimation, le fenêtrage, etc...

SORTIE : permet de sortir du programme d'acquisition.

Lecture sous IDL du format RAW :

Comme il n'y a aucun en tête dans les fichiers RAW, il est indispensable de connaître :

- le nombre de colonnes de chaque image
- le nombre de lignes
- le nombre d'images successives enregistrées, si le fichier RAW contient une rafale.

Sous IDL, les lignes de code sont les suivantes pour un RAW 2D (une image) :

```
Dimx = nombre de colonnes  
Dimy = nombre de lignes  
Tab=intarr(dimx,dimy)  
Openr,10,'nom du fichier RAW'  
Readu,10,Tab  
Tab=Tab/64  
Close,10
```

pour un RAW 3D (N images enregistrées) :

```
Dimx = nombre de colonnes  
Dimy = nombre de lignes  
Dimz = nombre d'images  
Tab=intarr(dimx,dimy,dimz)  
Openr,10,'nom du fichier RAW'  
Readu,10,Tab  
Tab=Tab/64  
Close,10
```