

Les fichiers de suivi d'EROS

| | |
|---------------|---------------------------|
| R. Ansari | ansari@lal.in2p3.fr |
| É. Aubourg | aubourg@hep.saclay.cea.fr |
| É. Lesquoy | lesquoy@hep.saclay.cea.fr |
| C. Magneville | cmv@hep.saclay.cea.fr |

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 3 |
| 2 | Structure des fichiers | 3 |
| 3 | Les fonctions de manipulations des suivis | 6 |
| 4 | Mémoire cache | 8 |
| 5 | Programmes utilitaires | 9 |
| | Index | 11 |

1 Introduction

Les données obtenues après la reconstruction photométrique des images d'EROS ¹ constituent une séquence de mesures photométriques pour un ensemble d'étoiles. A partir de la série d'images prises d'un même champ dans le ciel, on reconstruit donc les luminosités apparentes des étoiles observables sur ce champ en fonction du temps. Les courbes de lumière ainsi obtenues peuvent être ensuite analysées pour la recherche et l'étude des événements de type microlensing ou des étoiles variables. Le fichier de suivi contenant les courbes de lumière de N étoiles sur M images peut être symboliquement représenté par :

$$\left\{ \begin{array}{ll} (t_1, t_2, \dots, t_m, \dots) & m = 1, \dots, M \\ (Star)_s & (\phi_1, \phi_2, \dots, \phi_m, \dots)_s \quad s = 1, \dots, N \end{array} \right\}$$

Les t_m représentent les informations ou mesures se rapportant globalement à une prise de vue, comme la date (TStart), la durée d'exposition (Expose) ou le seeing. $(Star)_s$ désigne l'ensemble des informations indépendantes du temps, liées à une étoile. La position de l'étoile dans le ciel (α, δ) ou son flux moyenné sur l'ensemble des mesures sont des exemples de telles informations. Les mesures obtenues pour chacune des étoiles sur une image donnée, comme le flux mesuré de l'étoile sur l'image et l'erreur associée, sont représentées par $(\phi_m)_s$.

Cette note décrit la structure des fichiers de suivi, ainsi que l'ensemble des fonctions permettant de les manipuler (lire ou écrire).

2 Structure des fichiers

La structure logique des fichiers est en grande partie calquée sur le schéma symbolique décrit ci-dessus. Outre l'entête de fichier, contenant la description de la structure du fichier, cinq groupes d'information paramétrable par l'utilisateur ont été définis. L'entête n'est pas directement accessible et contient entre autre, le nombre d'étoiles contenues dans le fichier, ainsi que le nombre de mesures présentes. Chaque groupe d'information doit être défini à la création du fichier et peut contenir une succession de valeurs de type entier (I= 4 octets), short (S= 2 octets), flottant (F= 4 octets) ou double (D= 8 octets).

Les cinq groupes (ou Record) d'information constituant le fichier de suivi sont :

- **Comment (Rec-0)** est une zone de type texte (ascii). Sa longueur doit être définie par l'utilisateur. Cette zone ne peut être lue ou écrite qu'en un seul bloc.
- **GlobInfo (Rec-1)** peut contenir des informations globales concernant le fichier. On peut y stocker par exemple la description du champ d'étoiles ou le filtre utilisé. Il n'est évidemment pas utile de conserver le nombre d'étoiles ou de mesures dans GlobInfo. Ceux-ci sont conservés automatiquement dans l'entête.

¹Expérience de Recherche d' Objets Sombres

- **StarInfo (Rec-2)** correspond à $(Star)_s$ défini ci-dessus. Le fichier contiendra au total $N \times Rec2$ enregistrements de type StarInfo au total (N= Nombre d'étoiles).
- **TimeInfo (Rec-3)** correspond à l'information t_m définie ci-dessus. Le fichier contiendra au total $M \times Rec3$ enregistrements de type TimeInfo (M= Nombre de mesures).
- **Mesure (Rec-4)** correspond à $(\phi_m)_s$. Le fichier contiendra au total $N \times M \times Rec4$ enregistrements de type Mesure.

Les mesures sont en général ajoutées au fichier de suivi une à une, après traitement d'une image pour l'ensemble des étoiles. L'analyse de ces fichiers s'effectue le plus souvent étoile par étoile, par la lecture de toutes les mesures de l'étoile. Afin d'optimiser ces opérations, il est possible de regrouper les TimeInfo et les mesures par groupe de G (ou NItem) mesures. Ce paramètre est défini par l'utilisateur à la création du fichier. Un groupement par un nombre de l'ordre de 10% du nombre total des mesures est raisonnable. La figure 1 représente schématiquement la structure des fichiers de suivi.

Les fonctions de manipulation des suivis offrent une assez grande souplesse. Il y a cependant quelques limitations, en particulier :

Le format du fichier ne peut être modifié après la création de celui-ci (par Suivi-Create()).

On ne peut ajouter d'étoile dans le fichier une fois que des mesures ou timeinfo ont été écrites dans le fichier. (Après SuiviAddMesure()). Par contre, on peut modifier le contenu StarInfo d'une ou plusieurs étoiles à tout moment.

Il n'y a pas de procédure de suppression d'une étoile ou d'une mesure. Mais on peut modifier (surécrire) le contenu de n'importe quel record (Rec-i) déjà présent dans le fichier.

Les mesures (Mesure, TimeInfo) ne peuvent être qu'ajoutés successivement dans le fichier, alors qu'aucune contrainte n'existe pour les StarInfo. On peut en effet écrire (ajouter) l'étoile 100, alors que le fichier ne contient que deux étoiles. Dans ce cas, la place pour les étoiles 3 à 99 est réservée, sans que l'information correspondantes soit initialisée.

A partir de la version V_Fev98, la librairie offre la possibilité d'accès multi-plateforme aux fichiers de suivis : Entre des machines utilisant le codage des valeurs flottantes au format IEEE, mais ayant des conventions différentes pour l'ordre des octets en mémoire, les fonctions de lecture/écriture des suivis gèrent les échanges d'octets de manière transparente.

Dans le cas d'architecture matériel non conforme au norme IEEE, il est possible de traduire le fichier dans un format intermédiaire (MI : Machine Independant), transférer le fichier binaire résultant, pour reconstituer le fichier de suivi. Ces opérations s'effectuent à l'aide du programme `transmi`.

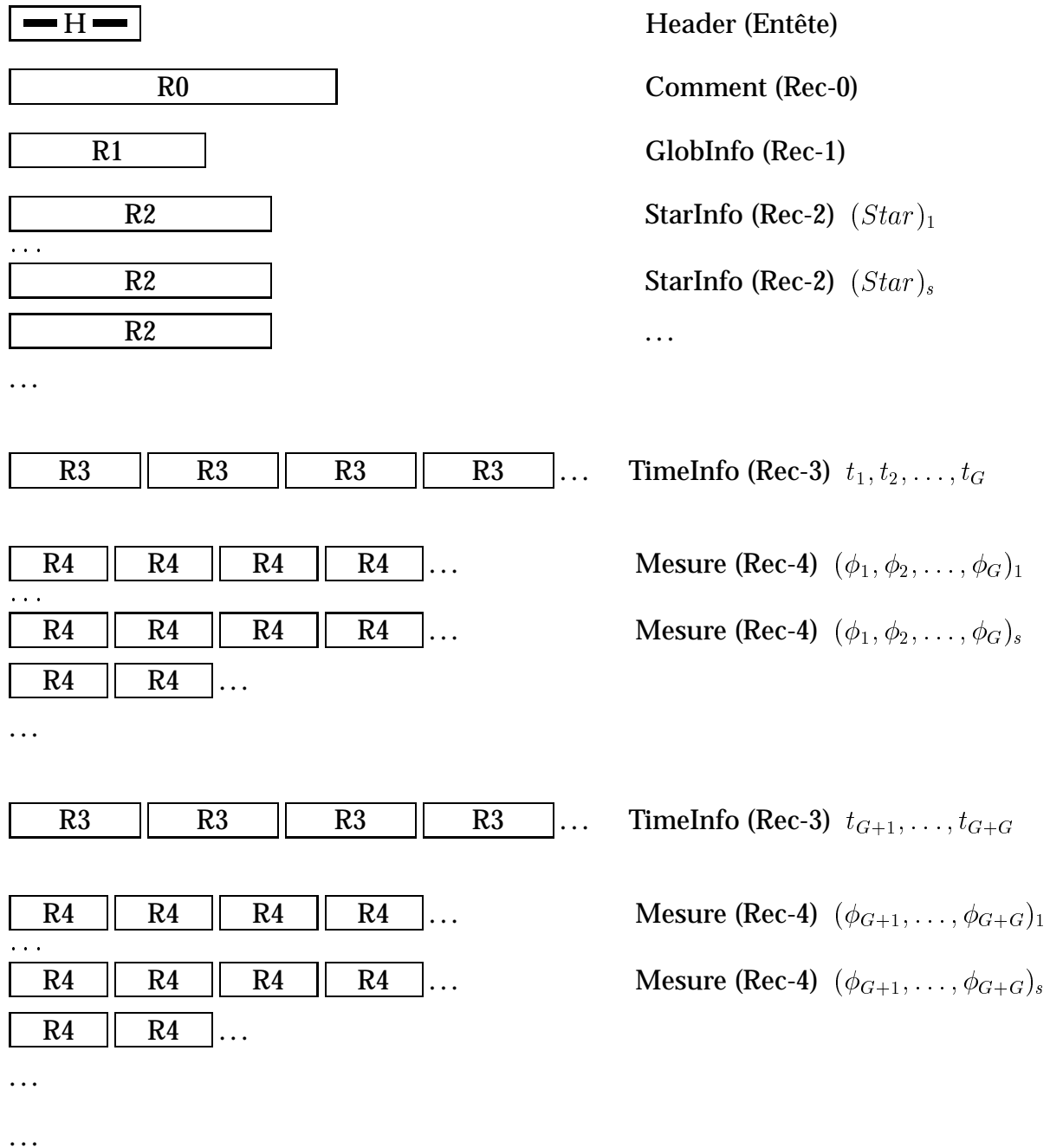


FIG. 1 – Structure du fichier de suivi

3 Les fonctions de manipulations des suivis

La déclaration de toutes les fonctions décrites ici se trouvent dans le fichier `fsv-cache.h`, associée à la librairie `libfsv.a`. La fonction de création de suivi crée la structure d'en-tête avec la définition des formats des records.

int SuiviCreate(char *filename, int typ, int rec0size, char *rec1fmt, char *rec2fmt, char *rec3fmt, char *rec4fmt, int nitem)

- filename : Nom de fichier (pointeur de chaîne de caractères)
- typ : Identificateur de type de fichier créé
- rec0size : Taille de la zone commentaire (Rec-0) en octets
- recifmt : (i=1 ..4) Format des records Rec-i (i=1 ..4)

Les formats sont spécifiés sous forme d'une chaîne de caractères décrivant succession de valeur constituant un record (Ex : "IIFFS"). Le codage suivant est utilisé :

I ou L : entier (4 octets)

S : entier (2 octets)

F : Réel (4 octets)

D : Double précision (8 octets)

- nitem : facteur de groupement des mesures (G) (défaut = 10)
- Rc : Le code de retour est nul (0) en cas de succès.

Une fois le fichier de suivi créé, toute manipulation ultérieure doit débuter par l'ouverture de fichier par `SuiviOpen()` et doit se terminer par `SuiviClose()`. Le pointeur (l'identificateur) de fichier renvoyé par `SuiviOpen()` est le premier argument de toutes les autres fonctions (`suivifip`).

SUIVIFIP *SuiviOpen(char *filename, int rw)

- filename : Nom de fichier (pointeur de chaîne de caractères)
- rw : Flag pour ouverture du fichier en mode lecture, ou lecture/écriture. L'activation de la mémoire cache se fait à travers ce flag (voir la section 4). rw = `SUOF_RW` ou `SUOF_RO` ou `SUOF_RW_MEMx` ou `SUOF_RO_MEMx`. x peut valoir rien ou de 2 à 4, permettant ainsi de sélectionner 4 tailles de mémoire cache.
- Rc : renvoie l'identificateur de fichier ou `NULL` en cas de problème.

Si le fichier est modifié, il est impératif de fermer le fichier à la fin des opérations. Dans le cas contraire, tout ou partie des modifications peut être perdue, et le fichier risque d'être inutilisable.

int SuiviClose(SUIVIFIP *suivifip)

- suivifip : Identificateur de fichier

Les fonctions suivantes permettent d'accéder à certaines informations sauvegardées dans l'en-tête (Le nombre d'étoiles, le nombre de mesures, le type du fichier et la lon-

gueur d'un enregistrement de type `Mesure`).

int_4 SuiviGetNbStars(SUIVIFIP *suivfip)

int_4 SuiviGetNbMesures(SUIVIFIP *suivfip)

int_4 SuiviGetType(SUIVIFIP *suivfip)

int_4 SuiviGetNbytStars(SUIVIFIP *suivfip)

La fonction **void SuiviSetPrtLevel**(int lev) permet de modifier le niveau d'impression (messages d'information/debug) des fonctions de manipulations de suivi.

Pour toutes les fonctions de lecture/écriture, `buff` désigne le pointeur d'une zone mémoire contenant l'information à écrire ou destinée à recevoir l'information lue dans le fichier de suivi. La variable `numet` contient le numéro d'étoile (1 ..NbStars) et `numes` le numéro de mesure (1 ..NbMes). Le code de retour est non nul en cas d'erreur.

Lecture et écriture de la zone commentaire (Rec-0) :

int SuiviWriteComment(SUIVIFIP *suivfip, char *buff)

int SuiviReadComment(SUIVIFIP *suivfip, char *buff)

Lecture et écriture de `GlobInfo` (Rec-1) :

int SuiviWriteGlobInfo(SUIVIFIP *suivfip, char *buff)

int SuiviReadGlobInfo(SUIVIFIP *suivfip, char *buff)

Lecture et écriture de `StarInfo` (Rec-2), pour étoile numéro `numet`

int SuiviWriteStarInfo(SUIVIFIP *suivfip, int_4 numet, char *buff)

int SuiviReadStarInfo(SUIVIFIP *suivfip, int_4 numet, char *buff)

Pour ajouter une nouvelle mesure dans le fichier, il est nécessaire d'appeler la fonction `SuiviAddMesure()` avant d'appeler `SuiviWriteTimeInfo()` ou `SuiviWriteMesure()`. `SuiviAddMesure()` étend donc le fichier pour une nouvelle mesure, et écrit le `TimeInfo` correspondant. Ce dernier peut bien sur être modifié (ou réécrite) par `SuiviWriteTimeInfo()`.

int SuiviAddMesure(SUIVIFIP *suivfip, char *buff)

int SuiviWriteTimeInfo(SUIVIFIP *suivfip, int_4 numes, char *buff)

int SuiviWriteMesure(SUIVIFIP *suivfip, int_4 numet, int_4 nummes, char *buff)

La lecture des mesures ou des TimeInfo peut s'effectuer par un seul appel pour une série de mesures (NumMes compris entre debut et fin). Dans ce cas, les informations lues sont placées dans buff de manière consécutive, chaque mesure occupant la longueur de record (Rec-3 et Rec-4) correspondant. Il faut donc faire attention lors du passage de tableau de structures ayant une taille plus grande que Rec-3 ou Rec-4.

Attention : La lecture d'un ensemble de structures Mesure ou TimeInfo dont la taille n'est pas identique à la longueur de record (Rec-3, Rec-4) peut entraîner des erreurs d'alignement. Il est donc déconseillé de lire les structures mesures ou TimeInfo par groupe.

```
int SuiviReadTimeInfo(SUIVIFIP *suivfip, int_4 debut, int_4 fin, char *buff)
```

```
int SuiviReadMesures(SUIVIFIP *suivfip, int_4 numet, int_4 debut, int_4 fin, char *buff)
```

4 Mémoire cache

Afin d'augmenter la vitesse de lecture-écriture des fichiers, un système d'accès aux fichiers disques à travers une mémoire tampon (mémoire cache) a été mis en place. L'activation de cette fonction se fait au moment de l'ouverture du fichier de suivi, par la valeur du flag `rw SUOF_RW_MEMx` ou `SUOF_RO_MEMx` ($x =$ " ou 2,3,4).

Le fonctionnement de ce système est très simple. Au moment de l'ouverture du fichier (par `zfopen()`), une liste chaînée de zones mémoire, (des "pages") est créée. Ces pages sont destinées à contenir une image mémoire des zones accédées du fichier au moment des opérations de lecture et d'écriture, réduisant ainsi le nombre des accès disques.

La fonction `zfopen()` permet l'ouverture d'un fichier avec activation possible de la mémoire cache. `zfclose()` permet de fermer le fichier, et libère l'espace mémoire occupé.

```
FILECACHE *zfopen(char *flnm, char *mode, int NPg, int PgSiz)
```

- flnm : Nom de fichier à ouvrir.
- mode : Chaîne de caractères passée à `fopen()` pour spécifier le mode d'ouverture du fichier.
- NPg : Nombre de "pages" allouées en mémoire. Si cet argument est nul (0), la fonction mémoire cache n'est pas activée. Une valeur négative indique l'utilisation des valeurs par défaut de NPg et de PgSiz.
- PgSiz : Taille de chaque page en mémoire.
- Rc : La fonction retourne un pointeur d'identificateur de fichier en cas de succès (NULL sinon).

```
void zfclose( FILECACHE *fcp )
```

les fonctions de positionnement, de lecture et d'écriture ont un syntaxe similaire aux fonctions standard `fseek()`, `fread()` et `fwrite()`.

int `zfseek`(FILECACHE *fcp, long int offset, int fg)

size_t `zfred`(void *buff, size_t size, size_t nitems, FILECACHE *fcp)

size_t `zfwrite`(void *buff, size_t size, size_t nitems, FILECACHE *fcp)

La déclaration de ces fonctions se trouvent dans le fichier `filecache.h` et la librairie associée est `libfsv.a`.

5 Programmes utilitaires

Les programmes suivants permettent de manipuler globalement les fichiers de suivi :

- **transmi** : Conversion des fichiers de suivi en ou depuis le format MI (Machine Indpendant). L'utilisation de ce programme n'est en principe plus nécessaire.

```
cmd> transmi -h
```

```
Transmi_Erreur: Arguments pas specifie !
```

```
Usage : transmi TOMI/FRMI FileIn FileOut [FgAsc]
```

```
- TOMI/FRMI : Suivi ->/<- Format de transfert
```

```
- FgAsc : 0 (X), 1(X->IBM-VM), 2(IBM-VM->X)
```

- **checkfsv** : Permet de vérifier l'intégrité du fichier de suivi. Le code de retour est non nul si le fichier est illisible ou corrompu. Ce programme permet aussi de tronquer le fichier de suivi (supprimer des mesures en fin de fichier).

```
cmd> checkfsv -h
```

```
Usage: CheckFSV NomSuivi
```

```
ou CheckFSV [-restore] NomSuivi
```

```
ou CheckFSV [-tronq nmes] NomSuivi
```

```
ou CheckFSV [-swapdefaut] NomSuivi
```

```
ou CheckFSV [-fgcorrnmnes fgcorr,nmes] NomSuivi
```

```
Rc=0 -> Fichier OK, RC > 0 = Nb de mesure OK
```

```
-restore : Tronquer le fichier au nombre de mesure OK
```

```
-tronq nmes : Tronquer le fichier au nombre de mesure nmes
```

```
-swapdef : Changer la valeur de Flg Swap a SWAPDEFAULT
```

```
-fgcorrnmnes fgcorr,nmes : Modifie les valeurs de FgCorrupt et NbMes ds
```

- **concatfsv** : Permet de concatener plusieurs fichiers de suivi en un seul. Tous les fichiers doivent avoir le même format. Seul le GlobInfo et le StarInfo du premier fichier sont copiés dans le fichier de suivi de sortie. Il est possible de supprimer certaines mesures à l'aide de l'option `-killmes`. `concatfsv` peut être utilisé

pour réécrire un fichier dans le format natif du byte-swap de la machine.

```
cmd> concatfsv -h
```

```
Usage: ConcatFSV [-killmes M1,M2,...] FSVOut FSVIn1 FSVIn2 [FSVIn3 ...
```

```
Specification des mesures a supprimer: Mi = NumFile:NumMes
```

```
1<=NumFile<=NFile (MaxNFile=100) 1<=NumMes<=NbMesFile
```

Index

checkfsv, 9

concatfsv, 9

SuiviAddMeasure, 7

SuiviClose, 6

SuiviCreate, 6

SuiviGetNbMesures, 7

SuiviGetNbStars, 7

SuiviGetNbytStars, 7

SuiviGetType, 7

SuiviOpen, 6

SuiviReadComment, 7

SuiviReadGlobInfo, 7

SuiviReadMesures, 8

SuiviReadStarInfo, 7

SuiviReadTimeInfo, 8

SuiviSetPrtLevel, 7

SuiviWriteComment, 7

SuiviWriteGlobInfo, 7

SuiviWriteMeasure, 7

SuiviWriteStarInfo, 7

SuiviWriteTimeInfo, 7

transmi, 9

zfclose, 8

zfdopen, 8

zfred, 9

zfseek, 9

zfwrite, 9