"FA" SUBROUTINES
or the
ARPEGE/ALADIN FILES PACKAGE

J. Clochard, R. El Khatib, D. Paradis

First english version translated from the french one which was updated in November, 2000, and corresponding to the cycle xr22 (winter 1999/2000), with references to the extensions for the limited area cases (ALADIN, Full-POS).

Last update : 04/09/02 by R. El Khatib for cycle 24

TABLE OF CONTENTS

1) GENERALITIES ABOUT THE RECORDED DATA

1.1) Arpege/Aladin files

Through **"ARPEGE/ALADIN files"** one points out the **main kind of files used to interface programms of the ARPEGE/ALADIN suite**, that is to say :

- guess and primary results from analysis ;
- initial conditions (to start only) and primary outputs from the forecasting model or its initialization ;
- input and output of the post-processing designed to feed a database ;
- input and output of the post-processing designed to make boundary files for the limited area model ALADIN ;
- result of preinitialization, or more generaly result of any program designed to use data that are external to the Arpege/Aladin software,or from a different model configuration.
- constant and climatology fields ;
- possibly : mean fields or fields cumulated in time.

1.2) Structure of the recorded data

The **structure** of these files is **indexed per name**, using the indexed files package "LFI" (Logiciel de Fichiers Indexés).

There is **one** logical record for each horizontal field, implicitly coded in a GRIB format. **Usually, the units used are those of the International System, though this is not necessary.** It is possible not to code the fields, or even to code them differently (later).

GRIB, OMM code(edition 0), is used here rather like a way to code "fields" data in a packed form, while a part of the GRIB autodocumentation is used for checkings. The GRIB part is not accessible itself ; only some descriptors are accessible.

An **horizontal field** is represented either in **consecutive** gridpoints values on the collocation model grid, or in consecutive spectral coefficients. **The way the spectral coefficients are stored is different from the one used in the model (in the "FA" package the zonal wave number is the index of the inner loop while the total wave number is the index of the outer loop).**

The current custom is to store upper air fields, surface pressure (pronostic variables on the model levels) and the surface geopotential as spectral coefficients, while all the other fields are stored as gridpoint fields.
It is highly recommended not to pack the surface geopotential, neither in spectral space nor in gridpoint space, so the field

values will not be affected while changing the kind of representation.

The "FA" package does not deal with any spectral/gridpoint transformations.

In the case of a file ARPEGE, furthermore, we follow the **hypothesis** below :

- the **grid** is the model collocation grid ; it is **symetric with respect to the equator** ; and the number of points or the maximum zonal wave number per latitude circle can change (increasing, from the "pole" to the "equator").

- The **storage** of the values of the gridpoint fields is **contigus**, from the (pseudo-) North pole to the (pseudo-) South pole, the points being stored from West to East, starting from the (pseudo-) longitude zero (the points which have the same latitude value are adjacents.
(Notice : for the time being, the program does not consider explicitely this rule, but just the fact that the values are contigus)

In the case of a file ALADIN, there is a similar hypothesis as follows :

- the grid is the model collocation grid, with a fixed number of points per "line".

- The **storage** of the values of the gridpoint fields is **contigus**, from the South-West corner to the North-East corner, with points stored from West to the East.

Concerning the <u>vectors</u> :
- in ARPEGE, the "zonal" and "meridional" components are relative to the transformed sphere, in other words the local referential is the one of the transformed sphere, and the module of the components is not the true one but the reduced one ;
- in ALADIN, the components "U" et "V" are relative to the local referential of the model, but they are not reduced (because the map factor is defined relatively to a scaling factor) : if we do not consider the referential, it means that the vectors are "seen on the real geographical grid".

1.3) Auto-documentation

Any file ARPEGE/ALADIN is **documented** thanks to what we call, as a convention, **a "framework" ("cadre" in french)**, and also thanks to **a "date"**.

A **"framework" ("cadre")** is, first of all, a group of variables defined thanks to an applicative program, and stored in the global variables of the "FA" package. One can consider the "framework" as an object that is not necessarily attached to a file. The "framework" is part of the "FA" files and and it is written inside as **records** described below :

NAME OF THE RECORD (VARIABLE)	Dimension in ARPEGE	Meaning in ARPEGE	Dimension in ALADIN	Meaning in ALADIN
CADRE-DIMENSIONS (MTRONC)	5 integers	1 : NSMAX = truncation	5 integers	1 : NSMAX=truncation, "y" axis direction
(NLATIT)		2 : NDGL = number of latitude rows from pole to pole		2 : NDGL = number of points along the "y" axis
(NXLOPA)		3 : NDLON = maximum number of longitude points per latitude row		3 : NDLON = number of points along the "x" axis
(NNIVER)		4 : NFLEVG = number of vertical levels		4 : NFLEVG = number of vertical levels
(NTYPTR)		5 : NSTTYP = kind of horizontal transformation		5 : (-1)*NMSMAX = (-1) * truncation, "x" axis direction
CADRE-FRANKSCHMI (SSLAPO)	4 reals	1 : Sine of the latitude of the pole of interest	4 reals	1 : EBETA = "Beta" angle of rotation for the subdomain
(SCLOPO)		2 : Cosine of the longitude of the pole of interest		2 : not used
(SSLOPO)		3 : Sine of the longitude of the pole of interest		3 : not used
(SCODIL)		4 : stretching factor		4 : PCODIL= binary indicator ; PCODIL = 0. for a real geographical domain (needs to define an extension zone) PCODIL = 1. for a virtual toric domain (to do academic studies)

CADRE-REDPOINPOL (NLOPAR)	At least half the number of latitude rows, and even, integers	number of longitude points per latitude row for the northern hemisphere (symetry with respect to the equator is expected)	(8 + 2*NSMAX + 4) integers	KNLOPA(1) : non-packed sub-truncation (do not use any more !) KNLOPA(2) : data and domain representation ; = 0 to tell that all the data is gridpoint on C+I only =-1 to tell that all the data is gridpoint on C+I+E =1 to tell that the dynamic data is spectral while the physical fields are gridpoints on C+I+E KNLOPA(3) =1 KNLOPA(4) = dimension of the geographical domain (C+I) along the "x" axis KNLOPA(5) = 1 KNLOPA(6) = dimension of the geographical domain (C+I) along the "y" axis KNLOPA(7) = number of points for the relaxation of coupling in the geographical domain (C+I) along the "x" axis KNLOPA(8) = number of points for the relaxation of coupling in the geographical domain (C+I) along the "y" axis
(NOZPAR)	At least half the number of latitude rows, and even, integers	maximum number of Fourier waves per latitude row for the northern hemisphere (symetry with respect to the equator is expected)		not used

CADRE-SINLATITUD (SINLAT)	At least half the number of latitude rows, and even, reals	Sine of the latitude rows for the Northern hemisphere (symetry is expected)	18 reals	1 : Rotation parameter ; = 0 : no rotation = 1 : the point (PLONR,PLATR) is rotated onto the equator 2 , 3 : longitude and latitude of the reference point for rotation (in radians) 4 , 5 : longitude and latitude of the South-West corner of the part (C+I) of the domain (in radians) 6 , 7 : longitude and latitude of the North-East corner of the part (C+I) of the domain (in radians) 8 , 9 : longitude and latitude of the reference point for the projection (in radians) 10 : parameter of projection 11 : parameter of isotropy for EGGX (useless here) 12 : choice of reference point for the projection 13 ,14 : dimension of the domain (C+I) in x and y directions (in meters) 15 ,16 : mesh size of the domain (C+I) in x and y directions (in meters) 17 , 18 : wave length of the domain in x and y directions (in meters)

CADRE-FOCOHYBRID (NPREFE) (SFOHYB)	1+2*(1+ number of levels), reals	1: reference pressure "Prefer"for the function *A* 2 -> end : functions *A/Prefer* and *B* of the vertical hybrid coordinate	1+2*(1+ number of levels), reals	1: reference pressure "Prefer"for the function *A* 2 -> end : functions *A/Prefer* and *B* of the vertical hybrid coordinate
"Framework" associated to a file	1 integer	Version number of the FA package library (=1)	1 integer	Version number of the FA package library (=1)

A **"date"** is an <u>array of 11 values according to the GRIB syntax</u>, that is to say, in the proper ordering :

- year (civil, 4 digits), month, day, hour, minute of the dissemination time ;

- indicator for the unit of forcast range (1 for hours, preferably), then 3 values characterising this range, which should be in general :

0, 0, 0 for a non-initialized analysis);

N, 0, 10 for a N-hours long forecast (not 0);

0, 0, 1 for an initialized analysis ("P0").

- 2 values which have a meaning only for the means or the temporal cumulatives fields and which should be equal to zero for the cases described above.

Since their respective cycles CY24T1 et AL15, the programs Arpege and Aladin are initializing the 10th element with the range of the previously created file ; so that it is possible, given a single file, to recover the time lengh of cumulation.

The content of the frame and of the date are written on the file.

For the files in creation mode, the frame should be defined before the file is opened.

For a file that is already existing, this is not necessary (the frame, as well as the date, are then read on file while it is opened, and this enables a dynamic treatment, **but if the frame is previously defined, there should be a strict accordance between it and the frame that is read on file.**

A frame is located in the program thanks to a symbolic name that is given by the user subroutine. This name can differ for a same file used in two different programs ; the name of the frame is an element in memory ; though one can find this name inside the file (cf. notion of file identifier) this is not a constraint when the file is re-used.

The same frame can be used for more than one file.

It will not be possible to redefine (or remove) a frame unless it is not attached to any opened file, such a redefinition would produce a warning message and it will be possible only while opening a file.

The redefinition of a date for an opened file already owning a date is possible, but a warning message will appear.

There is also, though this can be ignored, the notion of **file identifier** : one can give, get, or modify an (internal) name to a file. While the file is created, the symbolic name of the frame is used as initial identifier. This label can be used to define the kind of file (for instance : a guess).

1.4) Encoding options

A/ Description

A few options (needed to write the fields because of the GRIB encoding) **are implicit**, **tunable once a file is opened**, and the **implicit values** (used each time a file is opened) are **themselves tunable** :

- the kind of encoding (no encoding, standard GRIB ou Arpege-modified GRIB);

- the number of bits per gridpoint value, and the number of bits per spectral coefficient ;

- the dimension of the "unpacked sub-truncation" (**the truncation is** is supposed to be triangular in ARPEGE and elliptic in ALADIN);

- the "Dolby exposant" P of the fields in spectral coefficients, as well as its "degree of modulation" D : in order to treat this kind of fields, the packing is applied on a pre-treated field, that has been transformed by an operator (laplacian in ARPEGE, ad hoc in Aladin) to an exposant (integer) that is the best according to the criterion of the relative error multiplied by the absolute error due to the packing, starting from P and in the interval [P-D,P+D], and finishing as the first relative minimum is found.

B/ Implicit values

At this time, the implicit values *corresponding to* **the encoding options** are the following, **but** (except for the kind of GRIB encoding) they are not the ones that are used in operation :

- ARPEGE- modified GRIB (preservation of the exact values of the extrema and of the spectral coefficient of the unpacked sub-truncation, optimal use of the degrees of freedom of the packing);

- 24 bits for each elematary packed value (gridpoint as well as spectral);

- T10 as unpacked sub-truncation;

- 1 as implicit laplacian exposant, tunable in an interval from -5 to +5.

C/ Values in operations

The so-called "default" values, used in operation and which are generated by setting the value -1 to the arguments of the subroutines FAGOTE ou FAGIOT are :

- 16 bits for each elementary packed value in gridpoint (24 bits before 23 February 23rd, 2000);

- 16 (for a file ARPEGE) or 18 (for a file ALADIN) bits for each elementary packed value in spectral coefficient (24 bits before 23 February 23rd, 2000);

- variable sub-truncation depending on the truncations themselves and on the number of bits per spectral field, in a different manner in global case (ARPEGE) and limited area case (ALADIN) : we adopt the maximum of the three following

expressions :
- An increasing function of the truncation (affine linear, after it is rounded : (1+T)/10) (the biggest of both in the case ALADIN)
- a decreasing function of the number B of bits used to code the spectral coefficients (affine linear with respect to the inverse of the number of bits, after it is rounded) which depend also of the truncation in the case of ALADIN : ((1+T)*25)/(10*B), versus 480/B-10 for ARPEGE
- The previously tuned value (=10, that is to say : a fixed truncation for ARPEGE and Aladin up to february 23rd, 2000), with a diminution to (T-1) for the small resolutions.

(The formula above gives a sub-truncation equal to 20 for ARPEGE T199 or T149, equal to 13 for the associeted ALADIN-France, and equal to 10 for the other models ALADIN.

Remark : the application of this formulation for the spectral fields can lead, <u>with small truncation</u>, to a situation where a packed field is **longer** than the same non-packed field ... In the next release of the software, we shall adopt, in such cases, to switch off the spectral packing.

2) GENERALITIES ABOUT THE INTERFACES

2.1) Location of files

a file ARPEGE is located thanks to a **FORTRAN logical unit number**. It is possible to touch the file through its name while opening it.

2.2) Location of fields

The recorded data should be **consecutive data in memory** : this remark is important for the fields.

The **fields** are, as indicated above, **implicitely encoded in a GRIB format**. To insert other kind of data than fields, only interfaces to do plain reading and writing (without arrangement nor packing) are available. These last possibilities have been apparently useless except in distributed memory, together with coding-decoding subroutines, but the experience of changes of computers leads to the (warm) following recommandations : do make only **homogeneous** data records in kind, and prefix the associated data records explicitely: for instance, with '%R%' for reals, '%I%' for integers.

The **fields** are defined by a **prefix**, a **mandatory level number** , and a **suffix** ; these 3 entities are defining the name of a record linked to a field. the <u>prefix</u> is <u>either the vertical coordinate type</u>, and then the level number is effectively useful, <u>either conventional</u> ; in that case the level number is useless, and the distinction prefix/suffix is then rather formal. The prefix cannot exceed 8 characters. When the generated record name exceeds the 16 characters allowed by LFI, the suffix is truncated, which should not cause any problem but a tiny risk of homonymy.
The managed prefix in the package today (which does not prevent using other ones) are the following :
- "S" : hybrid levels
- "P" : isobaric levels
- "H" : height (above a reference orography)
- "V" : iso potential vorticity levels
- "T" : isentropic levels
- "SURF" : surface level
- "JET" : jet(ICAO) level
- "TROPO" : tropopause (ICAO) level
- "MER" : mean sea level

2.3) Environnement of the interfaces

The interfacing **subroutines names** are **prefixed with "FA"**, like " **F**ichiers **A**RPEGE".

These subroutines rely upon the software of indexed sequential files LFI, and are written in the same spirit. So one can find the same notations regarding the **informative messages**, the **treatement of errors**, the multitasking working mode or the debugging mode.

Besides, the internal working mode of the software leads to a need in **size limitations**, as well for **horizontal and vertical resolution**, and for the number of files that are simultaneously opened. The effective limits, configurables because they are FORTRAN "PARAMETER" defined by directives, are the following :

on "CRAY", "SX4" et "VPP" platforms :
- Truncation T511, 999 vertical levels

on "LOWRES", "DEC", "RS6K", "HPPA" et "SUN" platforms :
- Truncation T215, 100 vertical levels

For all :
- 512 latitudes ; 1024 longitudes rows
- by recompiling the interfaces, the vertical limit can be raised to 999 niveaux, without damage while re-reading files that have been previously created
- 20 simultaneously opened files
- 20 simultaneously pre-defined frameworks.

Important remark to avoid overflow of arrays:

While **getting the framework of a file that has been opened without pre-defined framework** , one gets, using the same subroutine (FACIES) **the plain variables and arrays.** Even if a user program is written so that it is able to read any FA file, there are anyhow limits in the sizes of arrays, and so in resolutions. If these limits are below those of the interfacing software, there can be overflow of arrays while getting the framework if the horizontal or vertical resolution of the file exceeds the capacities of the user program and not of the FA software.

To prevent from such situation, it is possible - and so strongly recommended - to specify the user limits en terms of resolution (FARFLU), or possibly to align onself before on the FA software limits (FALIMU, before any call to FARFLU).

3) DESCRIPTION OF THE USER INTERFACES

3.1) List and purpose

The user interfaces subroutines available at the beginning of the year 2000 ensure the following functions (one subroutine per function):

subroutine	category	Purpose
FACADE	Basic	Setup of a framework
FAITOU	Basic	To open a file
FANDAR	Basic	To write a date
FAIENC	Basic	To write a field on a file
FACILE	Basic	To read a field on a file
FACIES	Basic	To get the content of a framework
FADIES	Basic	To read a date
FANION	Basic	To get the characteristics of a record of the kind "horizontal field"
FAIRME	Basic	To close a file
FACOND	Distibuted memory	To encode a field before writing it in a file
FADECO	Distributed memory	To decode a field after having read it from a file
FAIRNO	Distributed memory	To close a pseudo-file (no actual I/O for non-writer processors)
FANOUV	Distributed memory	To open a pseudo-file (no actual I/O for non-writer processors)
FAGOTE	Tuning	To modify the GRIB encoding options for a given file
FAGIOT	Tuning	To modify the implicit GRIB encoding options
FAVEUR	Tuning	To get the GRIB encoding options for a given file
FAVORI	Tuning	To get the implicit GRIB encoding options
FACTUM	Manage-ment	Removal of a framework
FACAGE	Manage-ment	To modify the conservation option of a framework
FAISAN	Advanced	To write a "plain" record (ie : not equivalent to a horizontal field) on a file. (can be used with FACOND in distributed memory mode)
FALAIS	Advanced	To read a "plain" record (ie : not equivalent to a horizontal field) on a file. (can be used with FADECO in distributed memory mode)
FARINE	Advanced	Initialisation of FA software
FAMISO	Prefer-ences	Debugging mode or not
FANMSG	Prefer-ences	Modulation of the verboose level, globally

FANERG	Preferences	Modulation of the errors treatment, globally
FANIME	Preferences	Modulation of the verboose level, for a given file
FATALE	Preferences	Modulation of the errors treatment, for a given file
FARFLU	software	Tuning of the horizontal and vertical user limits
FALIMU	software	To get the horizontal and vertical user limits
FACOCH	others	To copy a record of the kind "horizontal field" from a file to another one, without decoding/coding
FAUTIF	others	To define the file identificator
FALSIF	others	To get a file identificator

3.2) Recommendations

In the framework of a user program creating a new file , first one must be able to write data in this file :

- First : define the file framework if it does not exist yet (the same framework can be used for more than one file) (FACADE);

- Open the file (FAITOU);

- Define the date (FANDAR);

... and **do not forget to close the file before ending the program**, in order to be able to re-use it later (FAIRME).

In the framework of a user program using an existing file , first one must be able to read the file :

- **If one expects to treat a specific configuration only, then define first the file framework** if it does not exist yet (FACADE);

- **On the contrary if one expects to treat any file, then specify first the horizontal and vertical limits** (FARFLU);

- Open the file, referencing possibly an existing framework (which implies a strict equality between this framework and the one written in the file) (FAITOU).

Then it is recommended to **get the date** (FADIES), and to **control it, if needed** (it is syntaxically controlled while it is defined). Besides, it is recommended to **close the file when it becomes useless** , at least before ending the user program.

Remark concerning the file framework:

When a user is defining explicitely a file framework (using FACADE), he specifies among the input argument -beside a symbolic name and the numerical characteristics- whether one should keep or not (in memory, inside the FA software) the file framework after closing the last file associated to this framework.

In the case of a **dynamically defined file framework**, that is to say **created by opening an already existing file**, **this file framework is implicitely removed while closing the last file associated to this framework**.

If one wish to avoid this behavior of the software, one must specify via FACAGE that this framework should be kept after closing the last related file.

Copy of horizontal fields:

To copy on a resulting file the **"constants" fields or those which do not change**, for instance **the land-sea mask, the orography**, it is highly recommended to **recopy these records with the subroutine "FACOCH" .**
That way one avoids the recoding of the field, and so its potential modification (via modified GRIB options, particularly if the originating field does not use the implicit options and/or in the case the computer is changing... in short **one avoids the variation of constants**.

About very old files:

The possible files ARPEGE created with the very first release of the interfacing software (cycle 2 of the auxilary library) are not readable with the following releases, because of the type of horizontal transformation added in the file framework, and because of the file identificator added as a new record at a precise location in the file.

Nevertheless if one wants to get records, even of the kind "horizontal field", it is still possible to handle such files using directly the LFI software interfaces (the list of records names can be obtained thanks to the subroutine LFILAF).

4) HEADERS OF THE USER INTERFACES

facade

```
      SUBROUTINE FACADE ( CDNOMC, KTYPTR, PSLAPO, PCLOPO, PSLOPO,
     S                          PCODIL, KTRONC, KNLATI, KNXLON, KNLOPA,
     S                          KNOZPA, PSINLA, KNIVER, PREFER, PAHYBR,
     S                          PBHYBR, LDGARD )
C****
C Sous-programme servant a DEfinir un CAdre, voire a le
C redefinir.
C**
C      Arguments : CDNOMC == Nom symbolique du cadre;
C (tous d'Entree)  KTYPTR == Type de transformation horizontale;
C                  PSLAPO == Sinus de la latitude du pole d'interet;
C                  PCLOPO == Cosinus " " longitude " " " ;
C                  PSLOPO == Sinus " " longitude " " " ;
C                  PCODIL == Coefficient de dilatation;
C                  KTRONC == Troncature;
C                  KNLATI == Nombre de latitudes (de pole a pole);
C                  KNXLON == Nombre maxi de longitudes par parallele;
C        (Tableau) KNLOPA == Nombre de longitudes par parallele;
C                            (du pole nord vers l'equateur seulement)
C        (Tableau) KNOZPA == Nombre d'onde zonal maxi par parallele;
C                            (du pole nord vers l'equateur seulement)
C        (Tableau) PSINLA == Sinus des latitudes de l'hemisphere nord
C                            (du pole nord vers l'equateur seulement)
C                  KNIVER == Nombre de niveaux verticaux;
C                  PREFER == Pression de reference (facteur multipli-
C                            catif de la premiere fonction de la
C                            coordonnee hybride)
C        (Tableau) PAHYBR == Valeurs de la fonction "A" de la coordo-
C                            nnee hybride AUX LIMITES DE COUCHES;
C        (Tableau) PBHYBR == Valeurs de la fonction "B" de la coordo-
C                            nnee hybride AUX LIMITES DE COUCHES;
C                  LDGARD == Vrai si le cadre doit etre conserve meme
C                            apres la fermeture du dernier fichier
C                            qui s'y rattache.
C*
C La "redefinition" d'un cadre est possible a l'une de ces
C conditions:
C
C - le cadre a ete defini, mais n'a aucun fichier qui s'y rattache;
C - le cadre defini a au moins un fichier qui s'y rattache, et les
C nouveaux parametres de definition sont identiques a ceux deja
C definis.
C
C Toute "redefinition" de cadre donne lieu a une messagerie
C de niveau 1, donc non masquee par defaut.
C
```

facage

```
      SUBROUTINE FACAGE ( CDNOMC, LDGARD )
C****
C Sous-programme servant a redefinir l'option de conservation
C d'un cadre preexistant ( CAdre a Garder Eventuellement... )
C**
C Arguments :     CDNOMC == Nom symbolique du cadre;
C (tous d'Entree) LDGARD == Vrai si le cadre doit etre conserve meme
C                           apres la fermeture du dernier fichier
C                           qui s'y rattache.
C
```

facies

```
      SUBROUTINE FACIES ( CDNOMC, KTYPTR, PSLAPO, PCLOPO, PSLOPO,
     S                          PCODIL, KTRONC, KNLATI, KNXLON, KNLOPA,
     S                          KNOZPA, PSINLA, KNIVER, PREFER, PAHYBR,
     S                          PBHYBR, LDGARD )
C****
C Sous-programme servant a obtenir le contenu d'un Cadre.
C ( FACIES... par analogie avec FADIES, avec "C" pour Cadre... )
C**
C Arguments :       CDNOMC == Nom symbolique du cadre;
C (tous de Sortie, KTYPTR == Type de transformation horizontale
C sauf CDNOMC)      PSLAPO == Sinus de la latitude du pole d'interet;
C                   PCLOPO == Cosinus " " longitude " " " ;
C                   PSLOPO == Sinus " " longitude " " " ;
C                   PCODIL == Coefficient de dilatation;
C                   KTRONC == Troncature;
C                   KNLATI == Nombre de latitudes (de pole a pole);
C                   KNXLON == Nombre maxi de longitudes par parallele;
C         (Tableau) KNLOPA == Nombre de longitudes par parallele;
C                             (du pole nord vers l'equateur seulement)
C         (Tableau) KNOZPA == Nombre d'onde zonal maxi par parallele;
C                             (du pole nord vers l'equateur seulement)
C         (Tableau) PSINLA == Sinus des latitudes de l'hemisphere nord
C                             (du pole nord vers l'equateur seulement)
C                   KNIVER == Nombre de niveaux verticaux;
C                   PREFER == Pression de reference (facteur multipli-
C                             catif de la premiere fonction de la
C                             coordonnee hybride)
C         (Tableau) PAHYBR == Valeurs de la fonction "A" de la coordo-
C                             nnee hybride AUX LIMITES DE COUCHES;
C         (Tableau) PBHYBR == Valeurs de la fonction "B" de la coordo-
C                             nnee hybride AUX LIMITES DE COUCHES;
C                   LDGARD == Vrai si le cadre doit etre conserve meme
C                             apres la fermeture du dernier fichier
C                             qui s'y rattache.
C
```

facile

```
       SUBROUTINE FACILE ( KREP, KNUMER, CDPREF, KNIVAU, CDSUFF, KCHAMP,
     S                          LDCOSP )
C****
C Sous-programme de LECTURE d'un CHAMP HORIZONTAL sur un fichier
C ARPEGE.
C ( Champ d'Interet en LEcture )
C**
C Arguments :   KREP (Sortie) == Code-reponse du sous-programme;
C             KNUMER (Entree) == Numero de l'unite logique;
C             CDPREF (Entree) == Prefixe eventuel du nom d'article;
C             KNIVAU (Entree) == Niveau vertical eventuel;
C             CDSUFF (Entree) == Suffixe eventuel du nom d'article;
C ( Tableau ) KCHAMP (Sortie) == Valeurs REELLES du champ lu;
C             LDCOSP (Entree) == Vrai si le champ est represente
C                                par des coefficients spectraux.
```

facoch

```
       SUBROUTINE FACOCH ( KREP, KNUME1, KNUME2, CDPREF, KNIVAU, CDSUFF )
C****
C Sous-programme de reCOpie d'un Champ Horizontal d'un fichier
C ARPEGE sur un autre .
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUME1 (Entree) == Numero d'unite logique en entree;
C           KNUME2 (Entree) == Numero d'unite logique en sortie;
C           CDPREF (Entree) == Prefixe eventuel du nom d'article;
C           KNIVAU (Entree) == Niveau vertical eventuel;
C           CDSUFF (Entree) == Suffixe eventuel du nom d'article.
C
```

facond

```
      SUBROUTINE FACOND ( KREP, KNUMER, CDPREF, KNIVAU, CDSUFF,
     S                         PCHAMP, LDCOSP, CDNOMA, KLNOMA, PVALCO,
     S                         KLONGD )
#include "facom0.h"
C****
C Sous-programme de CODAGE d'un CHAMP HORIZONTAL destine a etre
C sur un fichier ARPEGE/ALADIN.
C ( COdage de (Nouvelles ?) Donnees )
C**
C Arguments :   KREP (Sortie) == Code-reponse du sous-programme;
C             KNUMER (Entree) == Numero de l'unite logique;
C             CDPREF (Entree) == Prefixe eventuel du nom d'article;
C             KNIVAU (Entree) == Niveau vertical eventuel;
C             CDSUFF (Entree) == Suffixe eventuel du nom d'article;
C ( Tableau ) PCHAMP (Entree) == Valeurs REELLES du champ a ecrire;
C             LDCOSP (Entree) == Vrai si le champ est represente
C                                 par des coefficients spectraux;
C             CDNOMA (Sortie) == Nom de l'article-champ a ecrire;
C             KLNOMA (Sortie) == Nombre de caracteres utiles dans
C                                 CDNOMA;
C ( Tableau ) PVALCO (Sortie) == Donnees destinees a l'ecriture;
C             KLONGD (Sortie) == Nombre de valeurs (mots de 64 bits
C                                 en principe) a ecrire.
C
C Remarques:
C
C - PVALCO est type reel par commodite, et doit avoir une longueur
C suffisante pour stocker les donnees codees. Le dimensionnement
C "tous terrains" est (2+JPXCHA), qui permet le cas echeant de
C stocker un champ a pleine resolution sans codage effectif.
C
C - CDNOMA doit avoir au moins JPXNOM caracteres.
```

factum

```
      SUBROUTINE FACTUM ( CDNOMC )
C****
C Sous-programme servant a supprimer un cadre.
C ( Cadre a TUer Methodiquement ? )
C**
C Argument : CDNOMC (Entree) == Nom symbolique du cadre.
C
```

fadeco

```
      SUBROUTINE FADECO ( KREP, KNUMER, CDPREF, KNIVAU, CDSUFF,
     S                          LDCOSP, CDNOMA, KLNOMA, PVALCO, KLONGD,
     S                          PCHAMP )
C Sous-programme de controle et de DECODAGE d'un CHAMP HORIZONTAL
C venant d'etre lu sur un fichier ARPEGE/ALADIN.
C ( DECOdage de donnees )
C**
C Arguments :   KREP (Sortie) == Code-reponse du sous-programme;
C             KNUMER (Entree) == Numero de l'unite logique;
C             CDPREF (Entree) == Prefixe eventuel du nom d'article;
C             KNIVAU (Entree) == Niveau vertical eventuel;
C             CDSUFF (Entree) == Suffixe eventuel du nom d'article;
C             LDCOSP (Entree) == Vrai si le champ est represente
C                                 par des coefficients spectraux;
C             CDNOMA (Sortie) == Nom de l'article-champ lu;
C             KLNOMA (Sortie) == Nombre de caracteres utiles dans
C                                 CDNOMA;
C ( Tableau ) PVALCO (Entree) == Donnees issues de la lecture;
C             KLONGD (Entree) == Nombre de valeurs (mots de 64 bits
C                                 en principe) lues;
C ( Tableau ) PCHAMP (Sortie) == Valeurs REELLES du champ lu.
```

fadies

```
      SUBROUTINE FADIES ( KREP, KNUMER, KDATEF )
C****
C Sous-programme permettant d' obtenir la date d'un fichier ouvert
C pour le logiciel de Fichiers ARPEGE, et deja muni d'une date.
C ( "DIES" = jour en latin... )
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero de l'unite logique;
C (Tableau) KDATEF (Sortie) == Date elle-meme (JPLDAT mots).
C
```

fagiot

```
      SUBROUTINE FAGIOT ( KNGRIB, KNBPDG, KNBCSP, KSTRON, KPUILA,
     S                          KDMOPL )
C****
C Ce sous-programme permet de modifier les options implicites
C liees au codage GRIB des champs .
C CES OPTIONS NE SONT UTILISEES QUE POUR (RE)ECRIRE DES CHAMPS
C codes en GRIB, et les nouvelles valeurs implicites ne serviront
C que LORS d'une OUVERTURE de FICHIER ULTERIEURE.
C ( Grib, Implicites Options Techniques )
C**
C Arguments : KNGRIB (Entree) == Niveau de codage GRIB (0,1,2);
C             KNBPDG (Entree) == Nombre de bits par valeur point-
C                                 de-grille;
C             KNBCSP (Entree) == Nombre de bits par partie reelle/
C                                 imaginaire de coeff. spectral;
C             KSTRON (Entree) == Sous-troncature non compactee;
C             KPUILA (Entree) == Puissance de laplacien;
C             KDMOPL (Entree) == Degre de modulation de KPUILA.
C
C N.B.: Il doit y avoir coherence vis-a-vis des cadres deja definis
C et vis-a-vis des limites usagers.
C ( ce qui en pratique, ne concerne que KSTRON )
C
```

fagote

```
      SUBROUTINE FAGOTE ( KREP, KNUMER, KNGRIB, KNBPDG, KNBCSP, KSTRON,
     S                          KPUILA, KDMOPL )
C****
C Ce sous-programme permet d'ajuster, pour un fichier ARPEGE
C ouvert, les options liees au codage GRIB des champs.
C CES OPTIONS NE SONT UTILISEES QUE POUR (RE)ECRIRE DES CHAMPS
C codes en GRIB.
C ( Grib, Options Techniques Effectives )
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C             KNUMER (Entree) == Numero d'Unite Logique concernee;
C             KNGRIB (Entree) == Niveau de codage GRIB (0,1,2);
C             KNBPDG (Entree) == Nombre de bits par valeur point-
C                                 de-grille;
C             KNBCSP (Entree) == Nombre de bits par partie reelle/
C                                 imaginaire de coeff. spectral;
C             KSTRON (Entree) == Sous-troncature non compactee;
C             KPUILA (Entree) == Puissance de laplacien;
C             KDMOPL (Entree) == Degre de modulation de KPUILA.
C
```

faienc

```
      SUBROUTINE FAIENC ( KREP, KNUMER, CDPREF, KNIVAU, CDSUFF, KCHAMP,
     S                           LDCOSP )
C****
C Sous-programme d'ECRITURE d'un CHAMP HORIZONTAL sur un fichier
C ARPEGE.
C ( Integration par Ecriture d'un (Nouveau ?) Champ )
C**
C Arguments :   KREP (Sortie) == Code-reponse du sous-programme;
C             KNUMER (Entree) == Numero de l'unite logique;
C             CDPREF (Entree) == Prefixe eventuel du nom d'article;
C             KNIVAU (Entree) == Niveau vertical eventuel;
C             CDSUFF (Entree) == Suffixe eventuel du nom d'article;
C ( Tableau ) KCHAMP (Entree) == Valeurs REELLES du champ a ecrire;
C             LDCOSP (Entree) == Vrai si le champ est represente
C                                 par des coefficients spectraux.
C
```

fairme

```
      SUBROUTINE FAIRME ( KREP, KNUMER, CDSTTU )
C****
C Sous-programme de FERMETURE d'une unite logique "Fichier ARPEGE"
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero de l'unite logique;
C           CDSTTU (Entree) == "STATUS" eventuel pour "CLOSE".
```

fairno

```
      SUBROUTINE FAIRNO ( KREP, KNUMER, CDSTTU )
#include "facom0.h"
C****
C Sous-programme de FERMETURE d'une unite logique "Fichier ARPEGE"
C (utilise par des processeurs qui ne font pas d'ecriture effective
C sur disque)
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero de l'unite logique;
C           CDSTTU (Entree) == "STATUS" eventuel pour "CLOSE".
C
```

faisan

```
      SUBROUTINE FAISAN ( KREP, KNUMER, CDNOMA, PDONNE, KLONGD )
C****
C Sous-programme d' ecriture d'un article de donnees non assimila-
C bles a un champ horizontal sur un fichier ARPEGE.
C ( Integration Simple d'un Article Non code )
C**
C Arguments :   KREP (Sortie) == Code-reponse du sous-programme;
C             KNUMER (Entree) == Numero de l'unite logique;
C             CDNOMA (Entree) == Nom de l'article;
C ( Tableau ) PDONNE (Entree) == Donnees a ecrire;
C             KLONGD (Entree) == Nombre de mots a ecrire.
C
```

faitou

```
      SUBROUTINE FAITOU ( KREP, KNUMER, LDNOMM, CDNOMF, CDSTTU, LDERFA,
     S                          LDIMST, KNIMES, KNBARP, KNBARI, CDNOMC )
C****
C Sous-programme d' OUVERTURE d'une unite logique "Fichier ARPEGE"
C Il s'agit d'un fichier indexe, traite par le logiciel LFI.
C
C**
C ARGUMENTS : Ce sont les memes que pour "LFIOUV", avec CDNOMC comme
C             argument supplementaire.
C
C         KREP (Sortie) == Code-reponse du sous-programme;
C       KNUMER (Entree) == Numero de l'unite logique;
C       LDNOMM (Entree) == Vrai si l'unite logique doit etre
C                           associee a un NOM de Fichier EXP-
C                            LICITE lors de l'"OPEN" FORTRAN;
C       CDNOMF (Entree) == Nom de fichier explicite, si
C                           *LDNOMM* est VRAI - Meme si ce
C                           n'est pas le cas, ce *DOIT* ETRE
C                           UN OBJET DE TYPE "CHARACTER" .
C       CDSTTU (Entree) == "STATUS" pour l'"OPEN" FORTRAN
C                           ('OLD','NEW','UNKNOWN','SCRATCH')
C                           par defaut, mettre 'UNKNOWN';
C       LDERFA (Entree) == Option d'erreur fatale;
C       LDIMST (Entree) == Option impression de Statistiques
C                           au moment de la fermeture;
C       KNIMES (Entree) == Niveau de la Messagerie (0,1 ou 2)
C                           ( 0==Rien, 2==Tout )
C       KNBARP (Entree) == Nombre d'articles logiques prevus,
C                           ce qui n'est utilise que lors de
C                           la Creation du fichier,
C                           et qui n'empeche quand meme pas
C                           d'avoir plus d'articles logiques;
C       KNBARI (Sortie) == Nombre d'articles logiques de don-
C                           nees sur le fichier, initialement.
C                           (zero si creation)
C       CDNOMC (Entree) == Nom du CADRE associe au fichier.
C*
C N.B. : Pour un fichier en mode creation, ce cadre doit avoir ete
C defini au prealable (via le sous-programme FACADE, ou par
C l'ouverture d'un fichier preexistant).
C Pour un fichier ARPEGE preexistant, le cadre est lu sur le
C fichier; s'il etait deja defini auparavant, il y a controle
C de coherence entre les deux versions du cadre.
C
```

falais

```
      SUBROUTINE FALAIS ( KREP, KNUMER, CDNOMA, PDONNE, KLONGD)
C****
C Sous-programme de lecture d'un article de donnees non assimila-
C bles a un champ horizontal sur un fichier ARPEGE.
C ( Lecture d'un Article Integre Simplement, non code )
C**
C Arguments :   KREP (Sortie) == Code-reponse du sous-programme;
C             KNUMER (Entree) == Numero de l'unite logique;
C             CDNOMA (Entree) == Nom de l'article;
C ( Tableau ) PDONNE (Entree) == Donnees a ecrire;
C             KLONGD (Entree) == Nombre de mots a ecrire.
C
```

falimu

```
      SUBROUTINE FALIMU ( KXNIVV, KXTRON, KXLATI, KXLONG )
C****
C Sous-programme servant a obtenir les valeurs courantes
C des LIMites Utilisateur en termes de Resolutions horizontale
C et verticale, valides globalement pour tous les Fichiers et Cadres
C ARPEGE manipules par le programme utilisateur.
C**
C Arguments :       KXNIVV == Nombre maximum de niveaux verticaux;
C  (tous de Sortie) KXTRON == Troncature maximum;
C                   KXLATI == Nombre maximum de latitudes pole a pole;
C                   KXLONG == Nombre maxi de longitudes par parallele.
C
```

falsif

```
      SUBROUTINE FALSIF ( KREP, KNUMER, CDIDEN )
C****
C Sous-programme renvoyant le NOM de l'Identificateur
C d'un fichier ARPEGE .
C ( Lecture Specifique de l'Identificateur de Fichier )
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero de l'unite logique;
C           CDIDEN (Sortie) == Nom de l'identificateur.
C
```

famiso

```
      SUBROUTINE FAMISO ( LDEBUG )
C****
C Ce sous-programme permet d'activer ou de desactiver le mode
C "Mise au point du logiciel" . ( par defaut, inactif )
C A noter que le mode "mise au point" du logiciel LFI n'est pas
C modifie.
C**
C Argument : LDEBUG (Entree) == Vrai si on doit activer ce mode.
C
```

fandar

```
      SUBROUTINE FANDAR ( KREP, KNUMER, KDATEF )
C****
C Sous-programme de definition d'une (Nouvelle) Date sur un fichier
C ARpege.
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero de l'unite logique;
C (Tableau) KDATEF (Entree) == Date elle-meme (JPLDAT mots).
C*
C En cas de modification effective (si le fichier etait deja muni
C d'une date), il y a messagerie de niveau 1.
C
```

fanerg

```
      SUBROUTINE FANERG ( KNIVAU )
C****
C Ce sous-programme se charge de mettre le Niveau Global d'Erreur
C Fatale du logiciel de Fichiers ARPEGE (*NRFAGA*) a la valeur
C KNIVAU, de meme que la variable correspondante du logiciel LFI.
C Par defaut, NRFAGA vaut 1.
C**
C Argument : KNIVAU (Entree) == Niveau global d'erreur fatale.
C                               Valeurs possibles:
C
C 0 : Rendre fatale toute erreur detectee, meme si elle correspond
C     a un fichier ouvert avec l'option "pas d'erreur fatale".
C 1 : Ne rend fatales que certaines erreurs "globales", c'est-a-dire
C     non reliables a un fichier ouvert, et les erreurs sur les fi-
C     chiers ouverts avec l'option "erreur fatale" (Mode par defaut)
C 2 : Passer outre toute erreur detectee, meme si elle correspond
C     a un fichier ouvert avec l'option "erreur fatale".
C     Neanmoins, le code-reponse eventuel ne sera pas zero.
C
```

fanime

```
    SUBROUTINE FANIME ( KREP, KNUMER, KNIMES )
C****
C Ce sous-programme permet d'ajuster le Niveau de Messagerie
C propre aux actions faites sur un fichier particulier, ouvert pour
C le Logiciel de Fichiers ARPEGE, de meme que le Niveau correspon-
C dant du logiciel LFI .
C Cependant, tant que le Niveau de Messagerie Global *NIMSGA*
C vaut 0 ou 2, le niveau propre au fichier est inoperant.
C *NIMSGA* vaut par defaut 1, et est reglable via le s/p "FANMSG".
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero d'Unite Logique concernee;
C           KNIMES (Entree) == Niveau de Messagerie souhaite.
C
```

fanion

```
      SUBROUTINE FANION ( KREP, KNUMER, CDPREF, KNIVAU, CDSUFF, LDEXIS,
     S                         LDCOSP, KNGRIB, KNBITS, KSTRON, KPUILA )
C****
C Sous-programme renseignant sur l'EXISTENCE et les CARACTERISTI-
C QUES eventuelles d'un Article de type CHAMP dans un Fichier ARPEGE
C ( LDEXIS est le "fanion" leve si l'article existe )
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero de l'unite logique;
C           CDPREF (Entree) == Prefixe eventuel du nom d'article;
C           KNIVAU (Entree) == Niveau vertical eventuel;
C           CDSUFF (Entree) == Suffixe eventuel du nom d'article;
C           LDEXIS (Sortie) == Vrai si l'article de type CHAMP
C                               existe bien dans le Fichier;
C           LDCOSP (Sortie) == Vrai si le champ est represente
C                               par des coefficients spectraux;
C           KNGRIB (Sortie) == Niveau de codage GRIB;
C           KNBITS (Sortie) == Nombre de bits de codage eventuel;
C           KSTRON (Sortie) == Sous-troncature non codee " -le;
C           KPUILA (Sortie) == Puissance de laplacien eventuelle.
C
C KNBITS n'a de sens que si l'article existe et a ete code;
C de meme pour KSTRON et KPUILA, qui ne sont applicables qu'a un
C champ represente en coefficients spectraux.
C Les arguments de sortie n'ayant pas de sens sont mis a
C 0 pour les entiers, .FALSE. pour les logiques.
```

fanmsg

```
      SUBROUTINE FANMSG ( KNIVAU )
C****
C Ce sous-programme se charge de mettre le Niveau Global d'
C impression des Messages du logiciel de Fichiers ARPEGE (*NIMSGA*)
C a la valeur KNIVAU , de meme que la variable correspondante du
C du logiciel LFI . Par defaut, NIMSGA vaut 1.
C**
C Argument : KNIVAU (Entree) == Niveau Global d'Impression
C                                des Messages.
C Valeurs possibles:
C
C 0 : N'emettre que les messages d'erreurs reellement importants .
C 1 : N'emettre qu'un minimum de messages "globaux", et les messages
C     lies a un fichier ouvert qui sont de niveau au plus egal au
C     niveau de la messagerie pour ce fichier (Mode par defaut) .
```

fanouv

```
      SUBROUTINE FANOUV ( KREP, KNUMER, LDNOMM, CDNOMF, CDSTTU, LDERFA,
     S                         LDIMST, KNIMES, KNBARP, KNBARI, CDNOMC )
#include "facom0.h"
C****
C Sous-programme d'initialisation SANS OUVERTURE d'une unite logique
C "Fichier ARPEGE". Il s'agit d'un fichier indexe,
C traite par le logiciel LFI.
C
C FANOUV est derive de FAITOU, mais n'appelle pas la couche LFI
C pour l'ouverture reelle.
C
C utilise pour la seule compression des donnees par des processeurs
C qui ne font pas d'ecriture effective sur disque
C
C
C**
C ARGUMENTS : Ce sont les memes que pour "LFIOUV", avec CDNOMC comme
C argument supplementaire.
C
C       KREP (Sortie) == Code-reponse du sous-programme;
C     KNUMER (Entree) == Numero de l'unite logique;
C     LDNOMM (Entree) == Vrai si l'unite logique doit etre
C                        associee a un NOM de Fichier EXP-
C                        LICITE lors de l'"OPEN" FORTRAN;
C     CDNOMF (Entree) == Nom de fichier explicite, si
C                        *LDNOMM* est VRAI - Meme si ce
C                        n'est pas le cas, ce *DOIT* ETRE
C                        UN OBJET DE TYPE "CHARACTER" .
C     CDSTTU (Entree) == "STATUS" pour l'"OPEN" FORTRAN
C                        ('OLD','NEW','UNKNOWN','SCRATCH')
C                        par defaut, mettre 'UNKNOWN';
C     LDERFA (Entree) == Option d'erreur fatale;
C     LDIMST (Entree) == Option impression de Statistiques
C                        au moment de la fermeture;
C     KNIMES (Entree) == Niveau de la Messagerie (0,1 ou 2)
C                        ( 0==Rien, 2==Tout )
C     KNBARP (Entree) == Nombre d'articles logiques prevus,
C                        ce qui n'est utilise que lors de
C                        la Creation du fichier,
C                        et qui n'empeche quand meme pas
C                        d'avoir plus d'articles logiques;
C     KNBARI (Sortie) == Nombre d'articles logiques de don-
C                        nees sur le fichier, initialement.
C                        (zero si creation)
C     CDNOMC (Entree) == Nom du CADRE associe au fichier.
C*
C N.B. : Pour un fichier en mode creation, ce cadre doit avoir ete
C defini au prealable (via le sous-programme FACADE, ou par
C l'ouverture d'un fichier preexistant).
C Pour un fichier ARPEGE preexistant, le cadre est lu sur le
C fichier; s'il etait deja defini auparavant, il y a controle

C de coherence entre les deux versions du cadre.
```

farine

```
      SUBROUTINE FARINE ( KOPTIO )
C****
C Ce sous-programme est charge des INITIALISATIONS du logiciel
C de Fichiers ARPEGE FA ( Routine d'INitialisation )
C**
C Argument :     KOPTIO == OPTION concernant le mode d'utilisation.
C (Entree)                (MULTI-TACHE ou NON)
C VALEURS POSSIBLES : 0 == Mode MONO-Tache prescrit;
C                     1 == Mode MULTI-Taches prescrit;
C                     2 == Utilisation du mode par defaut si c'est
C                           le premier appel; sinon on garde le mode
C                           prescrit anterieurement .
C
```

farflu

```
      SUBROUTINE FARFLU ( KXNIVV, KXTRON, KXLATI, KXLONG )
C****
C Sous-programme servant a specifier des Limites Utilisateur
C en termes de Resolutions horizontale et verticale, valides
C globalement pour tous les Fichiers et Cadres ARPEGE.
C ( Resolution Fichiers - Limites Utilisateur )
C**
C Arguments :        KXNIVV == Nombre maximum de niveaux verticaux;
C (tous d'Entree)    KXTRON == Troncature maximum;
C                    KXLATI == Nombre maximum de latitudes pole a pole;
C                    KXLONG == Nombre maxi de longitudes par parallele.
C*
C S'il y a des cadres deja definis (dynamiquement ou non) avec
C des valeurs correspondantes plus grandes, cela provoque une erreur
```

fatale

```
      SUBROUTINE FATALE ( KREP, KNUMER, LDERFA )
C****
C Ce sous-programme permet d'activer ou de desactiver l'option
C rendant fatale toute erreur detectee sur un fichier particulier,
C ouvert pour le Logiciel de Fichiers ARPEGE, de meme pour l'option
C correspondante du logiciel LFI.
C Cependant, tant que le niveau global d'erreur fatale *NRFAGA*
C vaut 0 ou 2, l'option propre au fichier est inoperante.
C *NRFAGA* vaut par defaut 1, et est reglable via le s/p "FANERG".
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero d'Unite Logique concernee;
C           LDERFA (Entree) == Option d'Erreur Fatale (Vrai=oui).
C

C 2 : Emettre tous les messages possibles, meme s'ils ne correspon-
C     dent pas a un fichier ouvert avec le niveau de Messagerie 2 .
C
```

fautif

```
      SUBROUTINE FAUTIF ( KREP, KNUMER, CDIDEN )
C****
C Sous-programme permettant de donner un NOM a l'Identificateur
C d'un fichier ARPEGE .
C ( l'Utilisateur Traite son Identificateur de Fichier )
C**
C Arguments : KREP (Sortie) == Code-reponse du sous-programme;
C           KNUMER (Entree) == Numero de l'unite logique;
C           CDIDEN (Entree) == Nom de l'identificateur.
C
C Une messagerie de niveau 1 est emise dans les cas "normaux"
C
```

faveur

```
      SUBROUTINE FAVEUR ( KREP, KNUMER, KNGRIB, KNBPDG, KNBCSP, KSTRON,
     S                          KPUILA, KDMOPL )
C****
C Ce sous-programme permet d' obtenir, pour un fichier ARPEGE
C ouvert, les options courantes liees au codage GRIB des champs.
C CES OPTIONS NE SONT UTILISEES QUE POUR (RE)ECRIRE DES CHAMPS
C codes en GRIB.
C ( Verification des options Effectives pour l'UtilisateuR )
C**
C Arguments :   KREP == Code-reponse du sous-programme;
C  (tous de   KNUMER == Numero d'Unite Logique concernee;
C  SORTIE)    KNGRIB == Niveau de codage GRIB (0,1,2);
C             KNBPDG == Nombre de bits par valeur point-de-grille;
C             KNBCSP == Nombre de bits par partie reelle/imaginaire
C                       de coefficient spectral;
C             KSTRON == Sous-troncature non compactee;
C             KPUILA == Puissance de laplacien;
C             KDMOPL == Degre de modulation de KPUILA.
C
```

favori

```
      SUBROUTINE FAVORI ( KNGRIB, KNBPDG, KNBCSP, KSTRON, KPUILA,
     S                          KDMOPL )
C****
C Ce sous-programme permet de connaitre les options implicites
C courantes liees au codage GRIB des champs.
C CES OPTIONS NE SONT UTILISEES QUE POUR (RE)ECRIRE DES CHAMPS
C codes en GRIB, et les valeurs implicites ne servent
C que LORS d'une OUVERTURE de FICHIER.
C ( Verification des Options Reputees Implicites )
C**
C Arguments :   KNGRIB == Niveau de codage GRIB (0,1,2);
C (tous de      KNBPDG == Nombre de bits par valeur point-de-grille;
C  SORTIE)      KNBCSP == Nombre de bits par partie reelle/imaginaire
C                         de coeff. spectral;
C               KSTRON == Sous-troncature non compactee;
C               KPUILA == Puissance de laplacien;
C               KDMOPL == Degre de modulation de KPUILA.
C

C globale.
C Si les valeurs donnees en argument depassent les limites cor-
C repondantes du logiciel, elles sont ecretees, avec un message.
C Une messagerie de niveau 1 est emise dans le cas normal ou
C ci-dessus.
C
```

<u>5) CATALOG OF THE FILES ARPEGE/ALADIN</u>

5.1) Historical (or "ICMSH") files

Through "HISTORICAL" files or **"ICMSH"** (standing for **I**n-**C**ore **M**odel **S**pectral **H**ybrid) one points out the **<u>main file of raw data used as interfaces between programs of the suite ARPEGE/ALADIN as input to run the forecasting model</u>**.

Conventionally, all the letters of the fields names are in **CAPITAL**.

The following sections describe the fields written in this kind of file :

<u>FIELDS IN SPECTRAL COEFFICIENTS</u> :

The proposed list of "dynamical" fields needed for an adiabatic moist model is the following :

- Surface geopotential **(in spectral coefficients)**: prefix 'SPECSURF', suffix 'GEOPOTENTIEL'; arbitrary level.

- Surface pressure: prefix 'SURF', suffix 'PRESSION'; arbitrary level.

- Other upper air fields : prefix 'S', the level is counted from 1 at the "top" of the atmosphere, and the suffixes are self-speaking for french-speaking people :

'TEMPERATURE'	Temperature
'HUMI.SPECIFIQUE'	Specific humidity
'POT.VITESSE'	Velocity potential (file ARPEGE)
'FONC.COURANT'	Stream function (file ARPEGE)
'WIND.U.PHYS'	"U" momentum of wind (file ALADIN)
'WIND.V.PHYS'	"V" momentum of wind (file ALADIN).

To which one should add, in the case of the Aladin non-hydrostatic model, the 2 following upper air fields :

'VERTIC.DIVER'	Vertical divergence
'PRESS.DEPART'	Pressure departure

<u>GRIDPOINRT FIELDS</u> :

Regarding gridpoints, the historical files contain climatology fields, pseudo-historical fields, historical fields and diagnostic fields as instantaneous or cumulated fluxes (the former ones, by the way, are not always actual "fluxes" but sometimes just instantaneous fields).

Climatology fields :

prefix	*suffix*	*trun-cated part of the suffix*	*Remarks*
SURF	ET.GEOPOTENT	IEL	Standard deviation ...
SURF	IND.TERREMER		Land-sea mask 1. ou 0. (one bit per value is recommended)
SURF	PROP.VEGETAT	ION	PROPortion of vegetation (de 0. à 1.)
SURF	EMISSIVITE		Emissivity
SURF	VAR.GEOP.ANI		Anisotropy of sub-scale orography
SURF	VAR.GEOP.DIR		Angle between the direction of orography and the x-axis
SURF	IND.VEG.DOMI		(ISBA) Index of vegetation (from 1. to 5.)
SURF	RESI.STO.MIN		(ISBA) stomatal minimal resistance
SURF	PROP.ARGILE		(ISBA) proportion of clay
SURF	PROP.SABLE		(ISBA) proportion of sand
SURF	EPAIS.SOL		(ISBA) soil depth
SURF	IND.FOLIAIRE		(ISBA) leaf area index
SURF	GZ0.THERM		(ISBA) thermal roughness length times g.

Pseudo-historical fields :

prefix	*suffix*	*trun-cated part of the suffixe*	*Remarks*
SURF	Z0.FOIS.G		dynamical roughness length times g.
SURF	ALBEDO		
SURF	RES.EVAPOTRA		(ISBA) Resistance to evapotranspiration

Historical fields :

prefix	*suffix*	*trun-cated part of the suffix*	*Remarks*
SURF	TEMPERATURE		
PROF	TEMPERATURE		
SURF	RESERV.EAU		Surface water content
PROF	RESERV.EAU		Deep soil water content
SURF	RESERV.GLACE		(ISBA) Surface ice content
PROF	RESERV.GLACE		(ISBA) Deep soil ice content
SURF	RESERV.NEIGE		surface snow amount
SURF	RESERV.INTER		(ISBA) Interception content

Cumulated fluxes :

prefix	*suffix*	*trun-cated part of the suffix*	*remarks*
SURF	PREC.EAU.CON	VECTIVE	Convective rain
SURF	PREC.EAU.GEC	HELLE	Stratiform rain
SURF	PREC.NEI.CON	VECTIVE	Convective snow
SURF	PREC.NEI.GEC	HELLE	Stratiform snow
SURF	FLU.RAY.THER	MIQUE	
SOMM	FLU.RAY.THER	MIQUE	
SURF	FLU.RAY.SOLA	IRE	
SOMM	FLU.RAY.SOLA	IRE	
SURF	FLU.MEVAP.EA	U	
SURF	FLU.MSUBL.NE	IGE	
SURF	FLU.CHA.SENS	IBLE	
SURF	FLU.LAT.MEVA	PORATION	
SURF	FLU.LAT.MSUB	LIMATION	
SURF	TENS.TURB.ZO	NAL	
SURF	TENS.TURB.ME	RIDIEN	
SURF	TENS.DMOG.ZO	NAL	Zonal gravity wave stress

SURF	TENS.DMOG.ME	RIDIEN	Meridian gravity wave stress

<u>Instantaneous fluxes</u> :

prefix	*suffix*	*trun-cated part of the suffix*	*Remarks*
SURF	NEBUL.TOTALE		from 0. to 1.
SURF	NEBUL.CONVEC	TIVE	from 0. to 1.
SURF	NEBUL.HAUTE		from 0. to 1.
SURF	NEBUL.MOYENN	E	from 0. to 1.
SURF	NEBUL.BASSE		from 0. to 1.
CLS	VENT.ZONAL		
CLS	VENT.MERIDIEN		
CLS	TEMPERATURE		
CLS	HUMI.SPECIFIQ	UE	
CLS	HUMI.RELATIVE		
CLS	MINI.TEMPERAT	URE	
CLS	MAXI.TEMPERAT	URE	

<u>NOTICE</u> :

All the **fields of the kind "flux" are calculated positively from top to bottom...** which explains that in front of "Evaporation" and "Sublimation", we had added an "M" like "Minus" to remind that this field is mainly negative or zero.

The two fields CLSMINI.TEMPERATURE et CLSMAXI.TEMPERATURE are "reset" after writing a new historical file ; so that they represent extrema only between two consecutive historical files.

5.2) Coupling files ALADIN

Through "coupling files" or "LBC" files (standing for Lateral Boundary Conditions) one points out the raw data file containing the limits conditions to execute the limited area model ALADIN.

These files have the same format and contain the same fields as the historical files (except the fluxes), though the physical surface fields inside are useless since the coupling apply only on upper air fields.

5.3) Climatology files

This kind of file contains climatology fields **as monthly mean values**, among which some are necessary to build historical files. it is generated by the so-called configuration "923" of ARPEGE/ALADIN software.

The **needed fields for ARPEGE/ALADIN** which are present in such files are the followings :

prefix	*suffix*	*trun-cated part of the suffix*	*Remark*
SPECSURF	GEOPOTEN	TIEL	spectral coefficients
SURF	GEOPOTENTIEL		gridpoint
SURF	TEMPERATURE		
PROF	TEMPERATURE		
SURF	PROP.RMAX.EA	U	Proportion of the maximum surface water content
SURF	PROP.RMAX.EA	U	Proportion of the maximum deep soil water content
SURF	Z0.FOIS.G		
SURF	ALBEDO		
SURF	RESERV.NEIGE		
SURF	ET.GEOPOTENT	IEL	standard deviation ...
SURF	IND.TERREMER		1. ou 0. (one bit per value)
SURF	PROP.VEGETAT	ION	PROPortion ... (from 0. to 1.)
SURF	EMISSIVITE		
SURF	VAR.GEOP.ANI		Anisotropy of sub-scale orography
SURF	VAR.GEOP.DIR		Angle between the direction of orography and the x-axis

SURF	IND.VEG.DOMI		(ISBA) Index of vegetation (from 1. to 5.)
SURF	RESI.STO.MIN		(ISBA) stomatal minimal resistance
SURF	PROP.ARGILE		(ISBA)
SURF	PROP.SABLE		(ISBA)
SURF	EPAIS.SOL		(ISBA) d2
SURF	IND.FOLIAIRE		(ISBA)
SURF	GZ0.THERM		(ISBA)

5.4) Post-processing files ("Fullpos")

Through "post-processing files" or "Fullpos" files one points out a file containing raw or elaborated data, as a result of the internal post-processing software of ARPEGE or ALADIN.

There is no rules concerning the fields inside such files ; though it is recommended, for an easy usage, to write out data in gridpoints rather than in spectral.
Concerning the file format, it can be either :
- an ARPEGE geometry (gaussian grid, possibly stretched and/or rotated, together with a triangular truncation),
- an ALADIN geometry(limited area projected onto a plan, together with a spectral elliptic truncation),
- a latitude x longitude regular grid in each direction (this format is actually a possible ALADIN geometry, but not used for the forecasting model).
In the common language, one often mix "Fullpos" files with such "latitude x longitude" files.