

Nihed BOUZOUTA

Institut National de la Météorologie – Tunisie

Mes premiers pas en
Phasage



RAPPORT DE STAGE

2004

Stage du 04 octobre au 16 novembre 2004 au
GMAP/SCM – Météo France – Toulouse (France)

P H A S A G E
DU PRE-CYCLE 29
DU MODELE ALADIN

Nihed BOUZOUITA

DSMB/PNT - Institut National de la Météorologie
(bouzouita@meteo.tn)

Sous la supervision de :

Ryad ELKHATIB & Yann SEITY
GMAP/SCM - Météo France - France

Stage du 04 octobre
au 16 novembre

2004

Au GMAP/SCM - Météo France - Toulouse (France)

INTRODUCTION

Le code d'ALADIN est en évolution permanente:

D'une part, le développement technique continu des machines du calcul, des plus simples au plus complexes, constitue une contrainte qui justifie cette évolution. D'autre part, le développement du code d'ARPEGE/ARPEGE-IFS doit être appréhendé dans le code d'ALADIN pour que ces deux modèles soient en «phase».

D'autre part, l'apport scientifique, engendré par l'introduction de nouvelles notions par exemple, amène naturellement à d'éventuels amendements et développements du code.

A suppléer que d'autres éléments, tels que l'évolution du nombre d'observations à travers le monde, la correction des bugs, les besoins d'allègement du code source et la nécessité de cohérence avec les nouvelles règles de programmation, induisent une telle évolution.

Le **PHASAGE** d'ALADIN est la validation de tous les nouveaux changements et développements de son code, qui découlent des paramètres sus-cités.

Cette procédure aboutit à une nouvelle version, nommée «cycle», qui sera une version portable et en date, donc, avec tous les nouveaux progrès techniques et scientifiques.

NOTIONS DE BASE DU PHASAGE

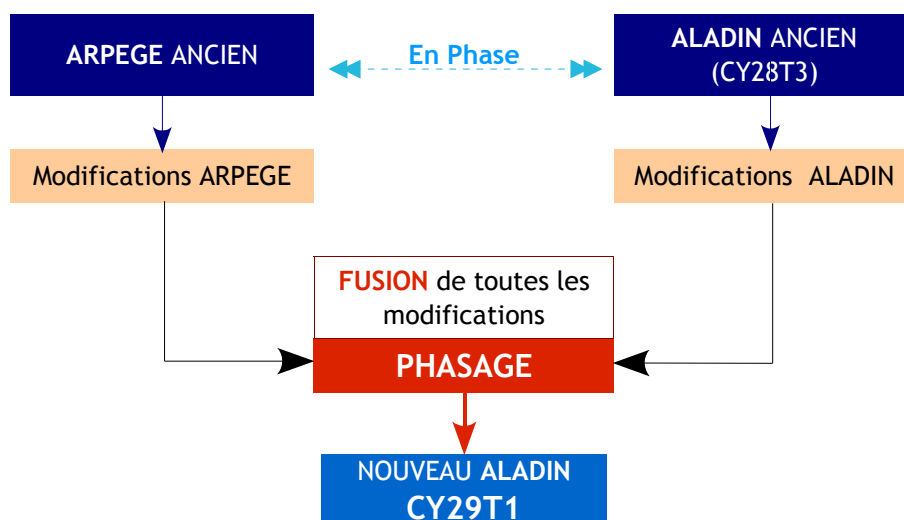
Il est à savoir que le code d'ALADIN est majoritairement celui d'ARPEGE.. Quelques spécificités lui sont propres, telles que sa géométrie, son couplage, ses transformées spectrales et sa version non-hydrostatique..

Cette propriété fait qu'ALADIN soit directement rattaché à l'évolution d'ARPEGE. Les développements sont alors phasés entre eux avant leur insertion au code des cycles officiels d'ALADIN ou d'ARPEGE.

Météo-France, Le Centre Européen (ECMWF) et les partenaires ALADIN contribuent au développement d'ALADIN. La fusion et l'intégration de tous ces développements assurent une optimisation du code, une maintenance continue, une durée de vie optimale et une portabilité faisant bénéficier tous les partenaires de l'évolution acquise..

Déjouer les incompatibilités locales et techniques, produire un cycle récent, optimisé et portable constituent «Le Phasage».

Ainsi, on part du dernier cycle opérationnel d'ALADIN, par exemple, le cycle 28T3. Les développements proposés depuis sa mise en opérationnel sont alors intégrés dans son code, après tests et vérifications, pour aboutir à un nouveau cycle, le 29T1:



A noter que les développements d'ALADIN induisent une actualisation, lors des phasages, de la partie du code le concernant dans ARPEGE..

PHASAGE DU PRE-CYCLE 29 D'ALADIN

Un cycle commun d'ARPEGE-IFS est généré périodiquement, en général chaque six mois.. Le code ainsi «actualisé» est envoyé à Météo-France -Toulouse- pour être évalué. Les modifications du code d'ALADIN sont alors fusionnées et intégrées.

Le code ainsi obtenu est alors compilé et soumis à une première évaluation.. Une fois validé, le Centre Européen annonce officiellement le nouveau cycle commun.

Le phasage constitue un rendez-vous spécial où sont réunis quelques développeurs (4 à 6) des pays membres. Ce rassemblement prend lieu deux fois par an en moyenne.

La première partie phasée concerne les configurations du modèle. La seconde celles des observations.

Le Phasage du pre-cycle 29 d'ALADIN a démarré le 02 octobre 2004. dans une première partie de mon stage, j'ai pris connaissance des diverses étapes du phasage, des points de vue théorique et pratique. La seconde partie a été consacrée à l'édition d'un document synoptique de l'utilisation des adaptations aux diverses plate-formes dans le code d'ALADIN (**xrd**).

PREMIERE PARTIE

SUPPRESSION DES INTERFACES DANS /INCLUDE

Mon premier exercice était d'examiner et évaluer les modifications concernant l'utilisation des interfaces dans les «includes» dans /ald/var.

Les étapes de cette manipulation sont:

- **Import du package** -code source- à l'aide de Clearcase:

```
cc_getpack -r 28t3 -b cy29 -v00 -u nihed
```

Sa toute première utilisation permet la création d'une nouvelle vue personnelle que j'ai nommée «nihed» (*arp_mrpe728_CY28T3_nihed*). Lorsqu'une nouvelle version est validé par GCO, on passe de -v00 à -v01, et ainsi de suite..

- **Validation des modifications** sur andante:

Deux sessions sur andante sont ouvertes, l'une servira pour l'examen des différences entre l'ancienne et la nouvelle version du code de la procédure en question, l'autre pour l'édition des procédures afin d'y intégrer les modifications validées pour figurer dans le code du nouveau cycle.

L'examen des différences se fait à l'aide de `cc-diff -h -f nom_fichier`. On examine ainsi les différences entre la version CY28T3 et les version CY29 (étiquetée CY28T3_CY29.00 pour la version 00). Ceci est réalisé sous /arp.

L'édition des procédures se fait à l'aide de `cc_edit -f nom_fichier -k`. Ceci est réalisé sous /ald.

- **Fusion des modifications** validées par GCO:

Une fois les toutes les modifications examinées, GCO procède à la fusion des diverses vues des phaseurs en une nouvelle version et un binaire est généré pour son évaluation.

MODIFICATIONS DANS /ald/var

Pour /ald/var, les procédures que j'ai examinées sont:

Nom procédure	Modifications/Commentaires	Intégration
ebalstat.F90	Aucune	Non
ebalstatad.F90	Nettoyage des includes	Oui
ebalvertad.F90	Nettoyage des includes	Oui
ebalverti.F90	Nettoyage des includes	Oui
ebalvertiad.F90	Nettoyage des includes	Oui
ecosjr.F90	Aucune	Non
ecvaru2i.F90	Nettoyage des includes	Oui
ecvaru2iad.f90	Nettoyage des includes	Oui
ejghcor.F90	Aucune	Non
ejghcori.F90	Aucune	Non
ejnrgg.F90	Nettoyage des includes Test LJB_NONLINEAR Ozone Tests	Oui Oui Oui
ejnrggad.F90	Nettoyage des includes Ozone variables change	Oui Oui
ejnrggi.F90	Nettoyage des includes Ozone variables change	Oui Oui
ejnrggiad.F90	Nettoyage des includes Ozone variables change	Oui Oui
ejgvor.F90	Aucune	Non
ejgvcor.F90	Aucune	Non
escaljgs.F90	Aucune	Non
suelges.F90	Wavelet JB Test	Oui
suellr.F90	Aucune	Non
suellt.F90	Memory allocation	Non
suellt7.F90	Nettoyage des includes	Non
suejbcosu.F90	Nettoyage des includes	Oui
suejbcov.F90	Nettoyage des includes	Oui
suescal.F90	Nettoyage des includes	Oui

DEUXIEME PARTIE

DOCUMENTATION SYNOPTIQUE DES ADAPTATIONS DANS xrd

Il existe dans le code d'ALADIN moultes adaptations pour le rendre portable sur les différentes plate-formes. Ces adaptations sont par exemple l'utilisation de clés qui attribue ou non une partie du code à la plate-forme sur laquelle la compilation est effectuée.

Pour alléger le code et le «nettoyer» pour éviter les redondances et les lourdeurs, optimisant ainsi et entre autres le temps de compilation, ces adaptations doivent être réduites au maximum.

Préliminairement, j'ai recensé les diverses adaptations sous xrd et rédigé un document synoptique qui constituerait un départ pour d'éventuels nettoyages:

DESCRIPTION SYNOTIQUE DES ADAPTATIONS

- **LOWRES**

DESCRIPTION: Macro pour le cas des champs à basse résolution

REFERENCES: [3]

/u/marp/marp001/dev/xrd/fa/facomp.h

/u/marp/marp001/dev/xrd/lfi/lficom0.h

/u/marp/marp001/dev/xrd/lfi/lfiini.F

USAGE:

/u/marp/marp001/dev/xrd/fa/facomp.h

```
#ifdef LOWRES
```

```
C Select the macro LOWRES in the case of low resolution fields
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
#ifdef HIGHRES
```

```
C Select the macro HIGHRES in the case of high resolution fields
```

```
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
```

```
#else
```

```
C If neither LOWRES nor HIGHRES is activated, the platform type is used
```

```
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined ( SUN )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
#if defined ( SGI ) || defined ( LX86P ) || defined ( IA32 ) || defined ( IA64 ) || defined ( AMD64 )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
C SX4, VPP and CRAY IS DEFAULT
```

```
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
```

```
#endif
```

```
#endif
```

```
#endif
```

```
#endif
```

JPXTRO : Troncature maximum gérable

JPXLAT : Nombre maximum de latitudes du pôle au pôle

JPXNIV : Nombre maximum des niveaux verticaux (champs d'altitude)

/u/marp/marp001/dev/xrd/lfi/lficom0.h

```
##ifdef LOWRES
```

```
    PARAMETER ( JPNXFI=50, JPFACX=20 )
```

```
#else
```

```
    PARAMETER ( JPNXFI=300, JPFACX=120 )
```

```
#endif
```

JPNXFI : Nombre maximum de fichiers indexés ouverts simultanément

JPFACX : Facteur multiplicateur maximum entre longueur d'article physique effective et élémentaire (de 1 à JPNXFI)

/u/marp/marp001/dev/xrd/lfi/lfiini.F

```
#ifdef LOWRES
```

```
    MFACTU(0)=1
```

```
#else
    MFACTU(0)=6
#endif
MFACTU(0) : facteur multiplicatif "par défaut" dans la table MFACTU,
table des facteurs multiplicatifs associés à une Unité Logique FORTRAN
destinée à être ouverte ultérieurement par le Logiciel de Fichiers Indexés
*LFI*.
```

- **HIGHRES**

DESCRIPTION: Macro pour le cas des champs à haute résolution

REFERENCES: [1]

/u/marp/marp001/dev/xrd/fa/facomp.h

USAGE:

/u/marp/marp001/dev/xrd/fa/facomp.h

```
#ifdef LOWRES
```

C Select the macro LOWRES in the case of low resolution fields

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
#ifdef HIGHRES
```

C Select the macro HIGHRES in the case of high resolution fields

```
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
```

```
#else
```

C If neither LOWRES nor HIGHRES is activated, the platform type is used

```
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
#if defined ( SGI ) || defined ( LX86P ) || defined ( IA32 ) || defined
( IA64 ) || defined ( AMD64 )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

C SX4, VPP and CRAY IS DEFAULT

```
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
```

```
#endif
```

```
#endif
```

```
#endif
```

```
#endif
```

- **ebug**

DESCRIPTION: Clé pour débogage (Impression messages "début" et "fin")

REFERENCES: [1]

/u/marp/marp001/dev/xrd/fa/decf10.F

USAGE:

/u/marp/marp001/dev/xrd/fa/decf10.F

```
#ifdef ebug
```

```
    print *, 'debut DECF10'
```

```
#endif
```

```
#ifdef ebug
```

```
    print *, 'fin DECF10, code-reponse=', IREPON
```

```
#endif
```

decf10.F : décale (en valeur) le descripteur "facteur d'échelle décimal" dans un message GRIB au niveau d'édition 1

- **SCALAR**

DESCRIPTION: Clé pour les champs scalaires

REFERENCES: [1]

/u/marp/marp001/dev/xrd/fa/fapula.F

USAGE:

/u/marp/marp001/dev/xrd/fa/fapula.F

```
CDP#ifdef SCALAR
```

```
    ALLOCATE ( ITAB2((ITRONC-1)*(IMTRONC-1)) )
```

```
CDP#else
```

```
CDP    ALLOCATE ( ITAB2(0:(ITRONC-1)*(IMTRONC-1)) )
```

```
CDP    ITAB2 = 0
```

```
CDP#endif
```

```
CDP#ifdef SCALAR
```

```
    IF (ITAB1(JK).GT.0) THEN
```

```
        IK = IK+1
```

```
        ITAB2(IK) = JK
```

```
        ITAB1(JK) = IK
```

```
    ENDIF
```

```
CDP#else
```

```
CDP    IK = IK + ITAB1(JK)
```

```
CDP    ITAB2(IK) = ITAB2(IK) + JK*ITAB1(JK)
```

```
CDP    ITAB1(JK) = IK
```

```
CDP#endif
```

fapula.F : Calcul de la puissance de laplacien qui aplatit le mieux le spectre des coefficients spectraux (hors sous-troncature) pour en réduire l'amplitude avant le codage GRIB.

- **DEC**

DESCRIPTION: Clé pour machines DEC

REFERENCES: [4]

/u/marp/marp001/dev/xrd/fa/facomp.h

/u/marp/marp001/dev/xrd/lfi/lficom0.h

/u/marp/marp001/dev/xrd/grib_mf/gabyte_mf.F

/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F

USAGE:

/u/marp/marp001/dev/xrd/fa/facomp.h

```
C If neither LOWRES nor HIGHRES is activated, the platform type is used
```

```
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined ( SUN )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
#if defined ( SGI ) || defined ( LX86P ) || defined ( IA32 ) || defined ( IA64 ) || defined ( AMD64 )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
C    SX4, VPP and CRAY IS DEFAULT
```

```
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
```

```
#endif
```

```
#endif
```

/u/marp/marp001/dev/xrd/lfi/lficom0.h

```
#ifdef ...
```

```
#elif DEC
```

```

PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )
PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
PARAMETER ( JPLARC=JPNCMO*JPLARD )
PARAMETER ( JPRECL=2*JPLARD )
#elif ...
JPNBIM : Nombre de bits par Mot Machine
JPNBIC : Nombre de bits par Caractère
LPCRAY : Clé de machine CRAY ("WHENEQ" utilisable)
JPNCMO : Nombre de caractères par Mot Machine
JPLARC : Longueur d'article "physique" exprimée en caractères
JPLARD : Longueur d'article "physique" élémentaire des fichiers (en mots)
JPRECL : Paramètre "RECL" de base pour l'ouverture des fichiers
/u/marp/marp001/dev/xrd/grib_mf/gsbyte_mf.F
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
    IF ( CDADIR.EQ.'D') THEN
        CALL GBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ELSE
        CALL SBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ENDIF
#else
C   SX4, VPP and CRAY IS DEFAULT
    IF (K.GT.JPSHOR.OR.K.LE.0) THEN
        CALL GSBITE_MF (KS,KD,KOFF,KSIZE,KSKBTW,K,KBPW,IMASKS,CDADIR)
    ELSEIF (CDADIR.EQ.'D') THEN
        CALL GBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ELSE
        CALL SBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ENDIF
#endif
gbytes_mf.F : Extrait des mots de longueur donnée et les aligne
successivement à partir d'une position donnée.
sbytes_mf.F : Insère des mots de longueur donnée à partir d'une position
donnée.
/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
    PARAMETER ( LPCRAY=.FALSE. )
#else
C   SX4, VPP and CRAY IS DEFAULT
    PARAMETER ( LPCRAY=.TRUE. )
#endif

```

- **RS6K**

DESCRIPTION: Clé pour machines IBM RS6000

REFERENCES: [4]

/u/marp/marp001/dev/xrd/fa/facomp.h

/u/marp/marp001/dev/xrd/lfi/lficom0.h

/u/marp/marp001/dev/xrd/grib_mf/gsbyte_mf.F

/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F

USAGE:

/u/marp/marp001/dev/xrd/fa/facomp.h

C If neither LOWRES nor HIGHRES is activated, the platform type is used

```
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
#else
#if defined ( SGI ) || defined ( LX86P ) || defined ( IA32 ) || defined
( IA64 ) || defined ( AMD64 )
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
#else
C   SX4, VPP and CRAY IS DEFAULT
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
#endif
#endif
/u/marp/marp001/dev/xrd/lfi/lficom0.h
#ifdef RS6K
    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )
    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
    PARAMETER ( JPLARC=JPNCMO*JPLARD )
    PARAMETER ( JPRECL=JPLARC )
#elif ...
/u/marp/marp001/dev/xrd/grib_mf/gabyte_mf.F
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
    IF (CDADIR.EQ.'D') THEN
        CALL GBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ELSE
        CALL SBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ENDIF
#else
C   SX4, VPP and CRAY IS DEFAULT
    IF (K.GT.JPSHOR.OR.K.LE.0) THEN
        CALL GSBITE_MF (KS,KD,KOFF,KSIZE,KSKBTW,K,KBPW,IMASKS,CDADIR)
    ELSEIF (CDADIR.EQ.'D') THEN
        CALL GBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ELSE
        CALL SBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)
    ENDIF
#endif
/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
    PARAMETER ( LPCRAY=.FALSE. )
#else
C   SX4, VPP and CRAY IS DEFAULT
    PARAMETER ( LPCRAY=.TRUE. )
#endif
AUTRES REFERENCES:
/u/marp/marp001/dev/xrd/include/drhook.h
/u/marp/marp001/dev/xrd/utilities/jsort.F
/u/marp/marp001/dev/xrd/support/sgemmx.F
/u/marp/marp001/dev/xrd/support/timef.F
/u/marp/marp001/dev/xrd/support/cptime.F
```

- **HPPA**

DESCRIPTION: Clé pour machines HPPA

REFERENCES: [4]

/u/marp/marp001/dev/xrd/fa/facomp.h

/u/marp/marp001/dev/xrd/lfi/lficom0.h

/u/marp/marp001/dev/xrd/grib_mf/gsbyte_mf.F

/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F

USAGE:

/u/marp/marp001/dev/xrd/fa/facomp.h

C If neither LOWRES nor HIGHRES is activated, the platform type is used

```
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
#if defined ( SGI ) || defined ( LX86P ) || defined ( IA32 ) || defined
( IA64 ) || defined ( AMD64 )
```

```
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
```

```
#else
```

```
C SX4, VPP and CRAY IS DEFAULT
```

```
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
```

```
#endif
```

```
#endif
```

/u/marp/marp001/dev/xrd/lfi/lficom0.h

```
#elif HPPA
```

```
    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )
```

```
    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
```

```
    PARAMETER ( JPLARC=JPNCMO*JPLARD )
```

```
    PARAMETER ( JPRECL=JPLARC )
```

```
#elif ...
```

/u/marp/marp001/dev/xrd/grib_mf/gsbyte_mf.F

```
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
```

```
    IF ( CDADIR.EQ.'D' ) THEN
```

```
        CALL GBYTES_MF ( KS,KD,KOFF,KSIZE,KSKBTW,K)
```

```
    ELSE
```

```
        CALL SBYTES_MF ( KS,KD,KOFF,KSIZE,KSKBTW,K)
```

```
    ENDIF
```

```
#else
```

```
C SX4, VPP and CRAY IS DEFAULT
```

```
    IF ( K.GT.JPSHOR.OR.K.LE.0 ) THEN
```

```
        CALL GSBITE_MF ( KS,KD,KOFF,KSIZE,KSKBTW,K,KBPW,IMASKS,CDADIR)
```

```
    ELSEIF ( CDADIR.EQ.'D' ) THEN
```

```
        CALL GBYTES_MF ( KS,KD,KOFF,KSIZE,KSKBTW,K)
```

```
    ELSE
```

```
        CALL SBYTES_MF ( KS,KD,KOFF,KSIZE,KSKBTW,K)
```

```
    ENDIF
```

```
#endif
```

/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F

```
#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )
```

```
    PARAMETER ( LPCRAY=.FALSE. )
```

```
#else
```

```
C SX4, VPP and CRAY IS DEFAULT
```

```

    PARAMETER ( LPCRAY=.TRUE. )
#endif

```

- **SUN**

DESCRIPTION: Clé pour machines SUN

REFERENCES: [4]

/u/marp/marp001/dev/xrd/fa/facomp.h

/u/marp/marp001/dev/xrd/lfi/lficom0.h

/u/marp/marp001/dev/xrd/grib_mf/gsbyte_mf.F

/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F

USAGE:

/u/marp/marp001/dev/xrd/fa/facomp.h

C If neither LOWRES nor HIGHRES is activated, the platform type is used

```

#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )

```

```

    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )

```

```

#else

```

```

#if defined ( SGI ) || defined ( LX86P ) || defined ( IA32 ) || defined
( IA64 ) || defined ( AMD64 )

```

```

    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )

```

```

#else

```

```

C SX4, VPP and CRAY IS DEFAULT

```

```

    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )

```

```

#endif

```

```

#endif

```

/u/marp/marp001/dev/xrd/lfi/lficom0.h

```

#elif SUN

```

```

    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )

```

```

    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )

```

```

    PARAMETER ( JPLARC=JPNCMO*JPLARD )

```

```

    PARAMETER ( JPRECL=JPLARC )

```

```

#elif

```

/u/marp/marp001/dev/xrd/grib_mf/gsbyte_mf.F

```

#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )

```

```

    IF (CDADIR.EQ.'D') THEN

```

```

        CALL GBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)

```

```

    ELSE

```

```

        CALL SBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)

```

```

    ENDIF

```

```

#else

```

```

C SX4, VPP and CRAY IS DEFAULT

```

```

    IF (K.GT.JPSHOR.OR.K.LE.0) THEN

```

```

        CALL GSBITE_MF (KS,KD,KOFF,KSIZE,KSKBTW,K,KBPW,IMASKS,CDADIR)

```

```

    ELSEIF (CDADIR.EQ.'D') THEN

```

```

        CALL GBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)

```

```

    ELSE

```

```

        CALL SBYTES_MF (KS,KD,KOFF,KSIZE,KSKBTW,K)

```

```

    ENDIF

```

```

#endif

```

/u/marp/marp001/dev/xrd/grib_mf/mxmn_mf.F

```

#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )

```

```

    PARAMETER ( LPCRAY=.FALSE. )
#else
C   SX4, VPP and CRAY IS DEFAULT
    PARAMETER ( LPCRAY=.TRUE. )
#endif

```

- **SGI / LX86P / IA32 / IA64 / AMD64**

DESCRIPTION: Clés pour machines SILICON GRAPHICS MIPS / LINUX X86 / Architecture INTEL 32-64 bits / AMD Architecture 64 bits

REFERENCES: [1]

/u/marp/marp001/dev/xrd/fa/facomp.h

USAGE:

/u/marp/marp001/dev/xrd/fa/facomp.h

C If neither LOWRES nor HIGHRES is activated, the platform type is used

```

#if defined ( DEC ) || defined ( RS6K ) || defined ( HPPA ) || defined
( SUN )

```

```

    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )

```

```

#else

```

```

#if defined ( SGI ) || defined ( LX86P ) || defined ( IA32 ) || defined
( IA64 ) || defined ( AMD64 )

```

```

    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )

```

```

#else

```

```

C   SX4, VPP and CRAY IS DEFAULT

```

```

    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )

```

```

#endif

```

```

#endif

```

JPXTRO : Troncature maximum gérable

JPXLAT : Nombre maximum de latitudes du pôle au pôle

JPXNIV : Nombre maximum des niveaux verticaux (champs d'altitude)

AUTRES REFERENCES:

SGI: /u/marp/marp001/dev/xrd/support/timef.F

- **VPP**

DESCRIPTION: Clé pour machines FUJITSU VPP5000

REFERENCES: [1]

/u/marp/marp001/dev/xrd/lfi/lficom0.h

USAGE:

/u/marp/marp001/dev/xrd/lfi/lficom0.h

```

#elif VPP

```

```

    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )

```

```

    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )

```

```

    PARAMETER ( JPLARC=JPNCMO*JPLARD )

```

```

    PARAMETER ( JPRECL=JPLARC )

```

```

#elif ...

```

AUTRES REFERENCES:

/u/marp/marp001/dev/xrd/include/drhook.h

- **HP**

DESCRIPTION: Clé pour machines HEWLETT PACKARD

REFERENCES: [2]

/u/marp/marp001/dev/xrd/lfi/lficom0.h


```

/u/marp/marp001/dev/xrd/lfi/lfidah.F
USAGE:
/u/marp/marp001/dev/xrd/lfi/lficom0.h
#elif HP
    PARAMETER ( JPNBIM=32, JPNBIC=8, LPCRAY=.FALSE. )
    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
    PARAMETER ( JPLARC=JPNCMO*JPLARD )
    PARAMETER ( JPRECL=JPLARC )
#elif
/u/marp/marp001/dev/xrd/lfi/lfidah.F
#ifdef HP
    CALL IDATE (IMOIS,IJOUR,IANNEE)
    CALL TIME (CLAUXIT)
#else
    CALL DATE (CLAUXI)
    READ (UNIT=CLAUXI,FMT='(I2,2(TR1,I2))') IMOIS,IJOUR,IANNEE
    CALL CLOCK (CLAUXI)
#endif

```

- **CRAY**

DESCRIPTION: Clé pour machines CRAY

REFERENCES: [1]

```
/u/marp/marp001/dev/xrd/lfi/lficom0.h
```

USAGE:

```

/u/marp/marp001/dev/xrd/lfi/lficom0.h
#else
C   CRAY IS DEFAULT
    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.TRUE. )
    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
    PARAMETER ( JPLARC=JPNCMO*JPLARD )
    PARAMETER ( JPRECL=JPLARC )
#endif

```

- **SX4**

DESCRIPTION: Clé pour machines SX4

REFERENCES: [1]

```
/u/marp/marp001/dev/xrd/lfi/lficom0.h
```

USAGE:

```

/u/marp/marp001/dev/xrd/lfi/lficom0.h
#elif SX4
    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )
    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
    PARAMETER ( JPLARC=JPNCMO*JPLARD )
    PARAMETER ( JPRECL=2*JPLARD )
#else ...

```

- **T3D**

DESCRIPTION: Clé pour machines T3D

REFERENCES: [1]

```
/u/marp/marp001/dev/xrd/lfi/lficom0.h
```

USAGE:

```
/u/marp/marp001/dev/xrd/lfi/lficom0.h
```

```
#elif T3D
```

```
    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )
```

```
    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
```

```
    PARAMETER ( JPLARC=JPNCMO*JPLARD )
```

```
    PARAMETER ( JPRECL=JPLARC )
```

```
#elif HP
```

• **f77**

DESCRIPTION: Clé pour compilation en Fortran 77

REFERENCES: [66]

```
/u/marp/marp001/dev/xrd/lfi/lficom1.h
```

```
/u/marp/marp001/dev/xrd/lfi/lfiafm.F
```

```
/u/marp/marp001/dev/xrd/lfi/lficap.F
```

```
/u/marp/marp001/dev/xrd/lfi/lficaq.F
```

```
/u/marp/marp001/dev/xrd/lfi/lficas.F
```

```
/u/marp/marp001/dev/xrd/lfi/lficax.F
```

```
/u/marp/marp001/dev/xrd/lfi/lficfg.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfidah.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfideb.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiecr.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiecx.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfief.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfieng.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfierf.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfifer.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfifmd.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfifmp.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfifra.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiini.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiist.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfilaf.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfilap.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfilas.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfilec.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiled.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfimst.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfinaf.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfineg.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfinfo.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfinim.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfinmg.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfinsg.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfinum.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfioef.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfioeg.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiofd.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiofm.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiomf.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiomg.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiosf.F
```

```
/u/marp/marp001/dev/xrd/lfi/lfiosg.F
```

```

/u/marp/marp001/dev/xrd/lfi/lfiouv.F
/u/marp/marp001/dev/xrd/lfi/lfipha.F
/u/marp/marp001/dev/xrd/lfi/lfipim.F
/u/marp/marp001/dev/xrd/lfi/lfipos.F
/u/marp/marp001/dev/xrd/lfi/lfirac.F
/u/marp/marp001/dev/xrd/lfi/lfiran.F
/u/marp/marp001/dev/xrd/lfi/lfirec.F
/u/marp/marp001/dev/xrd/lfi/lfiree.F
/u/marp/marp001/dev/xrd/lfi/lfiren.F
/u/marp/marp001/dev/xrd/lfi/lfisfm.F
/u/marp/marp001/dev/xrd/lfi/lfista.F
/u/marp/marp001/dev/xrd/lfi/lfisup.F
/u/marp/marp001/dev/xrd/lfi/lfitam.F
/u/marp/marp001/dev/xrd/lfi/lfiver.F
/u/marp/marp001/dev/xrd/lfi/lfivid.F
/u/marp/marp001/dev/xrd/lfi/tstlfi.F
/u/marp/marp001/dev/xrd/lfi/lfidst.F
/u/marp/marp001/dev/xrd/lfi/lfiedc.F
/u/marp/marp001/dev/xrd/lfi/lfiedo.F
/u/marp/marp001/dev/xrd/lfi/lfildo.F
/u/marp/marp001/dev/xrd/lfi/lfimod.F
/u/marp/marp001/dev/xrd/lfi/lfipxa.F
/u/marp/marp001/dev/xrd/lfi/lfipxf.F
/u/marp/marp001/dev/xrd/lfi/lfiintecr.F
/u/marp/marp001/dev/xrd/lfi/lfiintlec.F

```

USAGE:

Dans les routines sauf dans /u/marp/marp001/dev/xrd/lfi/lficom1.h

```
#ifndef f77
```

```
#include "precision.h"
```

```
#endif
```

precision.h : Définition des paramètres JPDBLE, JPDBLR et JP_SIMPLE_ENTIER INTEGER pour le codage des entiers «INTEGER (KIND=JPDBLE)» dans les fichiers indexés sur 32 ou 64 bits selon la plateforme.

```
/u/marp/marp001/dev/xrd/lfi/lficom1.h
```

```
#ifndef f77
```

```
    INTEGER (KIND=JPDBLE) MLGPOS (JPLARD*JPNXPI)
```

```
    INTEGER (KIND=JPDBLE) MTAMPD (JPLARD*JPNPDF*JPNXFI)
```

```
    INTEGER (KIND=JPDBLE) MDES1D (JPLARD*JPNXFI)
```

```
#else
```

```
    INTEGER MLGPOS (JPLARD*JPNXPI)
```

```
    INTEGER MDES1D (JPLARD*JPNXFI), MTAMPD (JPLARD*JPNPDF*JPNXFI)
```

```
#endif
```

```
/u/marp/marp001/dev/xrd/lfi/lfidah.F
```

```
#ifndef f77
```

```
#include "precision.h"
```

```
    INTEGER (KIND=JPDBLE) KDATE, KHEURE
```

```
    CHARACTER CLAUXIT*10
```

```
    CHARACTER CLDIFF*5
```

```
    INTEGER KTIME(8)
```

```
#else
```

```
    INTEGER KDATE, KHEURE
```

```
#endif
```

```
...
```

```
#ifndef f77
    CALL DATE_AND_TIME (CLAUXI,CLAUXIT,CLDIFF,KTIME)
    IANNEE=KTIME(1)
    IMOIS=KTIME(2)
    IJOUR=KTIME(3)
    IHEURE=KTIME(5)
    IMINUT=KTIME(6)
    ISECON=KTIME(7)
#else ...
    READ (UNIT=CLAUXI,FMT='(I2,2(TR1,I2))') IHEURE,IMINUT,ISECON
#endif
/u/marp/marp001/dev/xrd/lfi/lfiocr.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KTAB (KLONG)
#else
    INTEGER KTAB (KLONG)
#endif
/u/marp/marp001/dev/xrd/lfi/lfiexc.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KZONE (JPLARX)
#else
    INTEGER KZONE (JPLARX)
#endif
/u/marp/marp001/dev/xrd/lfi/lfifer.F
#ifndef f77
    INTEGER (KIND=JPDBLE) ITAMPO (JPLARX)
#else
    INTEGER ITAMPO (JPLARX)
#endif
/u/marp/marp001/dev/xrd/lfi/lfiist.F
#ifndef f77
    INTEGER (KIND=JPDBLE) IIDATE, IIHEUR
#else
    INTEGER IIDATE, IIHEUR
#endif
/u/marp/marp001/dev/xrd/lfi/lfilap.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KTAB (KLONG)
#else
    INTEGER KTAB (KLONG)
#endif
/u/marp/marp001/dev/xrd/lfi/lfilas.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KTAB (KLONG)
#else
    INTEGER KTAB (KLONG)
#endif
/u/marp/marp001/dev/xrd/lfi/lfilec.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KTAB (KLONG)
#else
    INTEGER KTAB (KLONG)
#endif
```

```
/u/marp/marp001/dev/xrd/lfi/lfiled.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KTAB (KLONG), IFOURT (JPLARX)
#else
    INTEGER KTAB (KLONG), IFOURT (JPLARX)
#endif
/u/marp/marp001/dev/xrd/lfi/lfiouv.F
#ifndef f77
    INTEGER (KIND=JPDBLE) IDATE, IHEURE
#else
    INTEGER IDATE, IHEURE
#endif
/u/marp/marp001/dev/xrd/lfi/lfiivid.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KTAMPO (JPLARX)
#else
    INTEGER KTAMPO (JPLARX)
#endif
/u/marp/marp001/dev/xrd/lfi/lfiiecd.F
#ifndef f77
    INTEGER (KIND=JPDBLE) KTAB (KLONG), IFOURT (JPLARX)
#else
    INTEGER KTAB (KLONG), IFOURT (JPLARX)
#endif
/u/marp/marp001/dev/xrd/lfi/lfimod.F
#ifndef f77
    INTEGER (KIND=JPDBLE) IDESCR (JPLARX)
#else
    INTEGER IDESCR (JPLARX)
#endif
/u/marp/marp001/dev/xrd/lfi/lfiintecr.F
#ifndef f77
    INTEGER (KIND=JPDBLE) ITAB (KLONG)
#else
    INTEGER ITAB (KLONG)
#endif
/u/marp/marp001/dev/xrd/lfi/lfiintlec.F
#ifndef f77
    INTEGER (KIND=JPDBLE) ITAB (KLONG)
#else
    INTEGER ITAB (KLONG)
#endif
NETTOYAGE PROPOSE
/u/marp/marp001/dev/xrd/fa/facomp.h
#ifdef LOWRES
C Select the macro LOWRES in the case of low resolution fields
    PARAMETER ( JPXTRO=215, JPXLAT=646, JPXNIV=200 )
#else
#ifdef defined ( HIGHRES ) || defined ( SX4 ) || defined ( VPP ) || defined
( CRAY )
C Select the macro HIGHRES in the case of high resolution fields
    PARAMETER ( JPXTRO=800, JPXLAT=1601, JPXNIV=999 )
#endif
#endif
```

```
/u/marp/marp001/dev/xrd/lfi/lficom0.h
#ifdef CRAY
    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.TRUE. )
#elif HP
    PARAMETER ( JPNBIM=32, JPNBIC=8, LPCRAY=.FALSE. )
#else
C   RS6K VPP T3D HPPA SUN DEC SX4
    PARAMETER ( JPNBIM=64, JPNBIC=8, LPCRAY=.FALSE. )
#endif
    PARAMETER ( JPNCMO=JPNBIM/JPNBIC )
    PARAMETER ( JPLARC=JPNCMO*JPLARD )
#if defined ( DEC ) || defined ( SX4 )
    PARAMETER ( JPRECL=2*JPLARD )
#else
    PARAMETER ( JPRECL=JPLARC )
#endif
```