Description des arguments E/S d'un ensemble de paramétrisation physique – **phylmd**

Zhao-Xin LI

13 mars 1995

Laboratoire de Météorologie Dynamique du CNRS
24, rue Lhomond; 75231 PARIS cedex 05
tél: (1) 44 32 22 34; fax: (1) 43 36 83 92
e-mail: li@lmd.ens.fr

Résumé

phylmd est un packet complet de paramétrisation physique, destiné aux études numériques du climat terrestre. Les programmes sont écrits en FORTRAN 77. Le présent document fournit une description des arguments Entrée/Sortie des programmes.

Table des matières

1 Généralités

1.1 dimphy.h

C'est la dimension du modèle:

klon: nombre de points horizontaux

klev: nombre de couches verticales

1.2 YOMCST.h

Ce fichier contient les constantes utilisés dans les paramétrisations physiques (voir les documents du modèle Arpège).

```
C A1.0 Fundamental constants
      REAL RPI,RCLUM,RHPLA,RKBOL,RNAVO
C A1.1 Astronomical constants
      REAL RDAY,REA,REPSM,RSIYEA,RSIDAY,ROMEGA
c A1.1.bis Constantes concernant l'orbite de la Terre:
      real R_ecc, R_peri, R_incl
C A1.2 Geoide
      REAL RA,RG,R1SA
C A1.3 Radiation
      REAL RSIGMA,RI0
C A1.4 Thermodynamic gas phase
      REAL R,RMD,RMV,RD,RV,RCPD,RCPV,RCVD,RCVV
      REAL RKAPPA,RETV
C A1.5,6 Thermodynamic liquid,solid phases
      REAL RCW,RCS
C A1.7 Thermodynamic transition of phase
      REAL RLVTT,RLSTT,RLMLT,RTT,RATM
C A1.8 Curve of saturation
      REAL RESTT,RALPW,RBETW,RGAMW,RALPS,RBETS,RGAMS
      REAL RALPD,RBETD,RGAMD
C
      COMMON/YOMCST/RPI   ,RCLUM ,RHPLA ,RKBOL ,RNAVO
     S       ,RDAY  ,REA   ,REPSM ,RSIYEA,RSIDAY,ROMEGA
     s       ,R_ecc, R_peri, R_incl
     S       ,RA    ,RG    ,R1SA
     S       ,RSIGMA,RI0
     S       ,R     ,RMD   ,RMV   ,RD    ,RV    ,RCPD
     S       ,RCPV  ,RCVD  ,RCVV  ,RKAPPA,RETV
     S       ,RCW   ,RCS
     S       ,RLVTT ,RLSTT ,RLMLT ,RTT   ,RATM
     S       ,RESTT ,RALPW ,RBETW ,RGAMW ,RALPS ,RBETS ,RGAMS
     S       ,RALPD ,RBETD ,RGAMD
```

1.3 YOETHF.h

Ce fichier contient les constantes spécifiques de la thermodynamique.

```
      REAL R2ES, R3LES, R3IES, R4LES, R4IES, R5LES, R5IES,
     .     RVTMP2, RHOH2O
      COMMON /YOETHF/
     .      R2ES,   R3LES,   R3IES,   R4LES,   R4IES
     .  ,   R5LES,   R5IES,   RVTMP2,  RHOH2O
```

1.4 FCTTRE.h

Ce fichier contient les formules thermodynamiques sous forme de fonction en ligne (in line function).

```
      REAL ptarg, pdelarg, p5arg, pqsarg, pcoarg
      REAL FOEEW ! vapeur d'eau saturante
      REAL FOEDE ! deriv\'e par rapport \`a la temp\'erature
      REAL qsats ! vapeur d'eau saturante (equilibre solide)
      REAL qsatl ! vapeur d'eau saturante (equilibre liquide)
      REAL dqsats, dqsatl ! deriv\'e par rapport \`a la temp\'erature
      LOGICAL thermcep ! utiliser la formule du Centre Europeen
      PARAMETER (thermcep=.TRUE.)
C
      FOEEW (ptarg,pdelarg) = EXP (
     .   (r3les*(1.-pdelarg)+r3ies*pdelarg)*(ptarg-rtt)
     . / (ptarg-(r4les*(1.-pdelarg)+r4ies*pdelarg)) )
C
      FOEDE (ptarg,pdelarg,p5arg,pqsarg,pcoarg) = pqsarg*pcoarg*p5arg
     . / (ptarg-(r4les*(1.-pdelarg)+r4ies*pdelarg))**2
C
      qsats(ptarg) = 100.0 * 0.622 * 10.0
     .           ** (2.07023 - 0.00320991 * ptarg
     .           - 2484.896 / ptarg + 3.56654 * LOG10(ptarg))
      qsatl(ptarg) = 100.0 * 0.622 * 10.0
     .           ** (23.8319 - 2948.964 / ptarg
     .           - 5.028 * LOG10(ptarg)
     .           - 29810.16 * EXP( - 0.0699382 * ptarg)
     .           + 25.21935 * EXP( - 2999.924 / ptarg))

c
      dqsats(ptarg,pqsarg) = RLVTT/RCPD*pqsarg * (3.56654/ptarg
     .                     +2484.896*LOG(10.)/ptarg**2
     .                     -0.00320991*LOG(10.))
      dqsatl(ptarg,pqsarg) = RLVTT/RCPD*pqsarg*LOG(10.)*
     .                (2948.964/ptarg**2-5.028/LOG(10.)/ptarg
     .                +25.21935*2999.924/ptarg**2*EXP(-2999.924/ptarg)
     .                +29810.16*0.0699382*EXP(-0.0699382*ptarg))
```

1.5 suphec

suphec initialise les constantes du modèle.

1.6 physiq

C'est le moniteur général de la physique qui sert à gérer les différentes paramétrisations, à gérer la communication avec la dynamique, à gérer les variables propres de la physique (la température du sol, par exemple) et à gérer l'écriture des sorties physiques.

```
      SUBROUTINE physiq (nlon,nlev,nqmx,
     .                   debut,lafin,ijour,gmtime,pdtphys,
     .                   paprs,pplay,pphi,
     .                   u,v,t,qx,
     .                   dudyn, dvdyn,  dtdyn, dqxdyn,
     .                   omega,
     .                   du, dv, dt, dqx, dps)
```

nlon	E, I	nombre de points à passer de la dynamique à la physique
nlev	E, I	nombre de couches à passer de la dynamique à la physique
nqmx	E, I	nombre de traceurs (y compris la vapeur d'eau)
debut	E, L	variable logique indiquant le premier passage
lafin	E, L	variable logique indiquant le dernier passage
ijour	E, I	numéro du jour dans l'année (de 1 à 360)
gmtime	E, R	temps universel en secondes (de 1 à 86400)
pdtphys	E, R	pas d'intégration physique utilisé dans la dynamique
paprs	E, R (nlon,nlev+1)	pression (en Pascal) des inter-couches
pplay	E, R (nlon,nlev)	pression (en Pascal) au milieu de chaque couche
pphi	E, R (nlon,nlev)	géopotentiel (m^2/s^2) de chaque couche
u	E, R (klon,klev)	vitesse direction X (de O à E) ($m\ s^{-1}$)
v	E, R (klon,klev)	vitesse direction Y (de S à N) ($m\ s^{-1}$)
t	E, R (klon,klev)	température (Kelvin)
qx	E, R (klon,klev,nqmx)	traceurs, y compris l'humidité spécifique (en kg/kg)
dudyn	E, R (klon,klev)	tendance dynamique de u
dvdyn	E, R (klon,klev)	tendance dynamique de v
dtdyn	E, R (klon,klev)	tendance dynamique de t
dqxdyn	E, R (klon,klev)	tendance dynamique de qx ($kg\ kg^{-1}\ s^{-1}$ pour l'humidité)
omega	E, R (klon,klev)	vitesse verticale (en Pa/s)
dt	S, R (klon,klev)	tendance physique pour t
dq	S, R (klon,klev)	tendance physique pour qx (en $kg\ kg^{-1}\ s^{-1}$ pour l'humidité)
du	S, R (klon,klev)	tendance physique pour u (en $m/s/s$)
dv	S, R (klon,klev)	tendance physique pour v (en $m/s/s$)
dps	S, R (klon)	tendance physique pour pression au sol (Pascal)

1.7 phyetat0, phyredem

La philosophie générale acceptée dans l'organisation du présent modèle est de séparer entièrement la physique de la dynamique, ainsi il est possible de mettre au point plusieurs physiques en parallèle. C'est pour cette raison que deux informations (*dtime* et *rlat*), déjà présentes dans la dynamique qui fait l'intégration temporelle et l'interpolateur (*calfis*) qui fait la conversion entre la grille dynamique et la grille physique, sont de nouveau entrées au modèle. Il est donc nécessaire de s'assurer de la cohérence de ces deux informations dans la dynamique et la physique.

```
      SUBROUTINE phyetat0 (dtime,radpas,co2ppm,solaire,
     s                     iflagcon,rlat,tsol,deltat,qsol,snow,radsol)
      SUBROUTINE phyredem (dtime,radpas,co2_ppm,solaire,
     e                     iflagcon,rlat,tsol,deltat,qsol,snow,radsol)
```

dtime	R	pas de la physique (s)
radpas	I	fréquence d'appel au rayonnement (par rapport á *dtime*)
co2ppm	R	concentration du gaz carbonique (en ppm)
solaire	R	constante solaire (en W m^{-2})
iflagcon	I	indicateur du choix de la convection souhaitée
rlat	R (klon)	latitude (en degré) de chaque point de la grille physique
tsol	R (klon)	température du sol (en K)
deltat	R (klon)	écart de température à la SST climatologique (en K), utilisé seulement quand la couche mélangée océanique est activée
qsol	R (klon)	humidité du sol (de 0 à 150 mm)
snow	R (klon)	épaisseur de la neige (mm)
radsol	R (klon)	rayonnement net au sol (W m^{-2})

1.8 condsurf

Ce programme lit les conditions aux limites

```
      SUBROUTINE condsurf(jour,
     .                    nat, sst, alb, rug, bils)
```

jour	E, I	jour dans l'année (de 0 à 359)
nat	S, R (klon)	indicateur de la nature du sol (0:océan; 1:terre; 2:glace terrestre; 3:glace océanique)
sst	S, R (klon)	température SST (K)
alb	S, R (klon)	albédo du sol nu (entre 0 et 1)
rug	S, R (klon)	longueur de rugosité (m)
bils	S, R (klon)	bilan net d'énergie au sol (W m^{-2}), indispensable si le modèle de couche mélangée océanique est activé

2 Rayonnements

2.1 radmain

Ce programme est l'interface du rayonnement. Il risque de diverger quand le calcul s'effectue sur une station de travail en simple précision. Il est donc souhaitable de calculer le rayonnement en double précision tout en gardant les entrées et sorties en simple précision pour cause de cohérence avec le reste du modèle (on peut dire que le code est en précision 1.5). Je soupçonne que la divergence du code est dans la méthode δ-Eddington, et il suffirait de mettre seulement cette partie en double précision. Mais actuellement la totalité du code est en double précision.

```
      SUBROUTINE radmain(sig,dsig,s,
     .                 rjour, rlat, co2_ppm, solaire,
     .                 psolpa, pskapa,tsol,albedo, h,q,wo,
     .                 cldfra, cldemi, cldtau,
     .                 heat,cool,radsol,albpla,
     .                 topsw,toplw,solsw,sollw,
     .                 topsw0,toplw0,solsw0,sollw0)
```

sig	E, R (klev+1)	niveau σ
dsig	E, R (klev)	épaisseur des couches en σ
s	E, R (klev)	niveau s
rjour	E, I	jour dans l'année (de 1 à 360)
rlat	E, R (klon)	latitudes de chaque point en degré
co2ppm	E, R	concentration du gaz carbonique en ppm
solaire	E, R	constante solaire (W m^{-2})
psolpa	E, R (klon)	pression au sol (Pascal)
pskapa	E, R (klon)	exponentiel κ de la *psolpa*
tsol	E, R (klon)	température du sol (Kelvin)
albedo	E, R (klon)	albédo du sol (entre 0 et 1)
h	E, R (klon,klev)	enthalpie potentielle
q	E, R (klon,klev)	humidité spécifique (kg kg^{-1})
wo	E, R (klon,klev)	concentration de l'ozone (cm atm)
cldfra	E, R (klon,klev)	fraction nuageuse (entre 0 et 1)
cldemi	E, R (klon,klev)	emissivité des nuages dans l'IR (entre 0 et 1)
cldtau	E, R (klon,klev)	épaisseur optique des nuages dans le visible
heat	S, R (klon,klev)	échauffement atmosphérique (visible) (K $jour^{-1}$)
cool	S, R (klon,klev)	refroidissement atmosphérique (IR) (K $jour^{-1}$)
radsol	S, R (klon)	bilan radiatif net au sol (positif vers le bas) (W m^{-2})
albpla	S, R (klon)	albédo planétaire (entre 0 et 1)
topsw	S, R (klon)	rayonnement solaire net au sommet de l'atmosphère (W m^{-2})
toplw	S, R (klon)	rayonnement IR net au sommet de l'atmosphère (W m^{-2})
solsw	S, R (klon)	rayonnement solaire net au sol (W m^{-2})
sollw	S, R (klon)	rayonnement IR net au sol (W m^{-2})
topsw0	S, R (klon)	rayonnement solaire (ciel clair) net au sommet de l'atmosphère (W m^{-2})
toplw0	S, R (klon)	rayonnement IR (ciel cliar) net au sommet de l'atmosphère (W m^{-2})
solsw0	S, R (klon)	rayonnement solaire (ciel cliar) net au sol (W m^{-2})
sollw0	S, R (klon)	rayonnement IR (ciel cliar) net au sol (W m^{-2})

2.2 shortw

L'objet de ce module est de calculer le transfert du rayonnement solaire. Pour une description théorique du code, se reporter au document du Centre Européen. Ce code est une version ancienne, il appelle deux sous-programmes: *tttt* et *wflux*. Le premier calcule la transmission et le second la diffusion du rayonnement par la méthode δ-Eddington.

```
      SUBROUTINE shortw(sig,dsig,fcst,coeffect,rmu0,fract,
     e                  psol,albsur,ozcmatm,tave,wv,wco2,
     e                  cldsw,tau,cg,omega,
     s                  fsup,fsdn)
```

sig	E, R (klev+1)	niveau σ
dsig	E, R (klev)	épaisseur de chaque couche en σ
fcst	E, R	constante solaire (en W/m^2)
coeffect	E, R	coefficient qui corrige la distance terre-soleil $(1/r^2)$
rmu0	E, R (klon)	cosinus de 'angle zénithal du soleil
fract	E, R (klon)	quand rmu0 est une valeur moyenne sur une journée, alors fract indique la durée fractionnelle de l'ensoleillement (durée/24_heures); quand rmu0 indique une valeur instintanée, fract est soit 0, soit 1.
psol	E, R (klon)	pression au sol en Pa
albsur	E, R (klon,2)	albédo du sol pour les deux canaux du calcul (proche-infrarouge et visible)
ozcmatm	E, R (klon,klev)	concentration d'ozone en $cm.atm$
tave	E, R (klon,klev)	température de chaque couche en K.
wv	E, R (klon,klev)	vapeur d'eau pour chaque couche en (kg/kg)
wco2	E, R	concentration du CO_2 en kg/kg (si CO_2 est donné en ppm, il faut multiplier le facteur (44/29)1.0E-06)
cldsw	E, R (klon,klev)	fraction nuageuse pour chaque couche (valeur entre 0 et 1)
tau	E, R (klon,2,klev)	épaisseur optique des nuages (pour deux canaux)
cg	E, R (klon,2,klev)	facteur d'asymétrie de la diffusion (pour deux canaux)
omega	E, R (klon,2,klev)	albédo de diffusion simple (pour deux canaux)
fsup	S, R (klon,klev+1)	flux montant du rayonnement solaire sur chaque niveau (en W/m^2)
fsdn	S, R (klon,klev+1)	flux descendant du rayonnement solaire sur chaque niveau (en W/m^2)

2.3 longw0 et longw1

Ces deux sous-programmes calculent la transmission du rayonnement infrarouge dans l'atmosphère. Pour l'aspect théorique, se reporter au document du Centre Européen. *longw0* calcule les flux du ciel clair et quelques variables intermédiaires, celles-ci sont ensuite utilisées dans *longw1* pour calculer les flux du ciel nuageux. *longw0* appelle le sous-programme *hornel* pour calculer la transmission entre deux niveaux.

```
      SUBROUTINE longw0(sig,psol,dt0,em0,wco2,ozkgpkg,wv,tl,tave,
     .                  fupclr,fdnclr, cts,hfg,cntrb,bint,bsuin)
```

sig	E, R (klev+1)	valeur de σ pour chaque niveau
psol	E, R (klon)	pression au sol (en Pa)
tl	E, R (klon,klev+1)	température pour chaque niveau σ
tave	E, R (klon,klev)	température pour chaque couche
dt0	E, R (klon)	différence de température entre le sol et le premier niveau *tl(1)*
em0	E, R (klon)	émissivité du sol (entre 0 et 1)
wv	E, R (klon,klev)	vapeur d'eau en kg/kg
ozkgpkg	E, R (klon,klev)	ozone en kg/kg
wco2	E, R	concentration du CO_2 (en kg/kg). si CO_2 est donné en ppm, il faut multiplier le facteur (44/29)1.0E-06
fupclr	S, R (klon,klev+1)	flux montants du rayonnement infrarouge (en W/m^2)
fdnclr	S, R (klon,klev+1)	flux descendants du rayonnement infrarouge (en W/m^2)
cts	S, R (klon)	variable intermédiaire pour *longw1*
hfg	S, R (klon)	variable intermédiaire pour *longw1*
cntrb	S, R (klon,klev+1,llm+1)	variable intermédiaire pour *longw1*
bint	S, R (klon,klev+1)	variable intermédiaire pour *longw1*
bsuin	S, R (klon)	variable intermédiaire pour *longw1*

```
  SUBROUTINE longw1(fupclr,fdnclr,em0,cldlw,
 .                  fupcld, fdncld, cts,hfg,cntrb,bint,bsuin)
```

fupclr	E, R (klon,klev+1)	flux infrarouge montant du ciel clair en provenance de *longw0*
fdnclr	E, R (klon,klev+1)	flux infrarouge descendant du ciel clair en provenance de *longw0*
em0	E, R (klon)	émissivité du sol (entre 0 et 1)
cldlw	E, R (klon,klev)	fraction nuageuse effective (couverture nuageuse multipliée par l'émissivité du nuage)
cts	E, R (klon)	variable intermédiaire en provenence de *longw0*
hfg	E, R (klon)	variable intermédiaire en provenence de *longw0*
cntrb	E, R (klon,klev+1,llm+1)	variable intermédiaire en provenence de *longw0*
bint	E, R (klon,klev+1)	variable intermédiaire en provenence de *longw0*
bsuin	E, R (klon)	variable intermédiaire en provenence de *longw0*
fupcld	S, R (klon,klev+1)	flux infrarouge montant pour le ciel nuageux
fdncld	S, R (klon,klev+1)	flux infrarouge descendant pour le ciel nuageux

2.4 orbite

Ce programme est adapté de la formule fournie par A. Berger. Il est capable de déterminer la position de la terre sur son orbite autour du soleil pour un moment donné.

```
SUBROUTINE orbite(xjour,longi,dist)
```

xjour	E, R	jour de l'année à compter du 1er janvier
longi	S, R	la longitude vraie de la terre dans son orbite (en degré) par rapport au point vernal (21 mars)
dist	S, R	la distance terre-soleil en unité astronomique

2.5 zenith

Il s'agit d'un simple calcul astronomique, qui donne le cosinus de l'angle zénithal (h) du soleil en connaissant la déclinaison du soleil (δ), la latitude (ϕ) et la longitude (l) du point sur la terre, et le temps universel (s).

```
SUBROUTINE zenith (longi, gmtime, lat, long, pmu0)
```

longi	E, R	longitude vraie (en degré)
gmtime	E, R	temps universel en second qui varie entre 0 et 86400
lat	E, R	latitude en degré
long	E, R	longitude en degré
pmu0	S, R	cosinus de l'angle zénithal

2.6 angle

Ce programme calcule la durée d'ensoleillement pour une journée et la hauteur moyenne du soleil (cosinus de l'angle zénithal) correspondante.

```
SUBROUTINE angle(longi, lati, frac, muzero)
```

longi	E, R	longitude vraie (en degré)
lati	E, R	la latitude d'un point sur la terre (en degré)
frac	S, R	la durée d'ensoleillement dans la journée, divisée par 24 heures (unité: entre 0 et 1)
muzero	S, R	la moyenne du cosinus de l'angle zénithal sur la journée (0 a 1)

2.7 nuage

Ce programme calcule l'épaisseur optique et l'émissivité des nuages à partir de l'eau liquide nuageuse.

```
      SUBROUTINE nuage (sigs, dsig, s, psolpa, pskapa,
     .                  h, pqlwp, pclc, pcltau, pclemi,
     .                  pch, pcl, pcm, pct, pctlwp)
```

sigs	E, R (klev)	valeur σ des couches s
dsig	E, R (klev)	épaisseur de couches en σ
s	E, R (klev)	niveau s
psolpa	E, R (klon)	pression au sol (Pascal)
pskapa	E, R (klon)	exponentiel de *psolpa*
h	E, R (klon,klev)	enthalpie potentielle
pqlwp	E, R (klon,klev)	eau liquide nuageuse dans l'atmosphère (kg kg^{-1})
pclc	E, R (klon,klev)	fraction nuageuse (entre 0 et 1)
pcltau	S, R (klon,klev)	épaisseur optique dans le visible
pclemi	S, R (klon,klev)	émissivité des nuages dans l'IR

2.8 o3cm

Ce programme fournit, d'une manière simple, un profil de l'ozone dans l'atmosphère. La formule utilisée est celle de Green (1964):

$$w = \frac{w_p}{h} \exp \frac{z - z_p}{h} / (1 + \exp \frac{z - z_p}{h})^2 \tag{1}$$

où w_p=0.218; h=4.63; z_p=23.25; l'unité de la hauteur z est en km, l'ozone ainsi obtenu (w) est en $cm.atm/km$.

```
SUBROUTINE o3cm (amb, bmb, val, ntab)
```

amb, bmb	E, R	valeurs indiquant la couche à calculer (en mb)
val	S, R	l'ozone dans cette couche en $cm.atm$
ntab	E, I	sa valeur détermine la précision du calcul, il s'agit du nombre d'intervalles utilisés pour faire l'intégration (20 est un bon choix pour une couche de 100 mb)

3 Couche limite planétaire

3.1 clmain

C'est le programme interface de la couche limite

```
      SUBROUTINE clmain(sig,dsig,s,sigs,dtime,rnat,
     .                  h,q,u,v,ts,psolpa,pskapa,radsol,
     .                  capsol,beta,dif_grnd,
     .                  xlat, rugos,
     .                  d_h,d_q,d_u,d_v,d_ts,
     .                  flux_t,flux_q,flux_u,flux_v,itop,coefh,coefm)
```

sig	E, R (klev+1)	valeur σ pour chaque niveau
dsig	E, R (klev)	épaisseur de chaque couche en σ
s	E, R (klev)	valeur s pour chaque couche
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
dtime	E, R	intervalle du temps (en second)
rnat	E, R (klon)	indicateur de la nature du sol (0:ocean; 1:terre; 2:glace terrestre; 3:glace océanique)
h	E, R (klon,klev)	enthalpie potentielle
q	E, R (klon,klev)	humidité spécifique (kg kg^{-1})
u	E, R (klon,klev)	vitesse u (m s^{-1})
v	E, R (klon,klev)	vitesse v (m s^{-1})
ts	E, R (klon)	température du sol (Kelvin)
psolpa	E, R (klon)	pression au sol sol (Pascal)
pskapa	E, R (klon)	exponentiel κ de *psolpa*
radsol	E, R (klon)	bilan radiatif net au sol (positif vers le bas) (W m^{-2})
capsol	E, R (klon)	inverse de la capacité effective du sol (J m^{-2} K^{-1})
beta	E, R (klon)	coefficient de l'évaporation réelle (entre 0 et 1)
difgrnd	E, R (klon)	inverse du temps de relaxation de la température du sol vers la température profonde (utilisé seulement pour relaxer la glace de mer vers la température de congélation) (s^{-1})
xlat	E, R (klon)	latitudes en degrés
rugos	E, R (klon)	longueur de rugosité (m)
dh	S, R (klon,klev)	tendance de h
dq	S, R (klon,klev)	tendance de q
du	S, R (klon,klev)	tendance de u
dv	S, R (klon,klev)	tendance de v
dts	S, R (klon)	tendance de ts
fluxt	S, R (klon,klev)	flux de chaleur sensible (C_pT) orienté positivement vers le haut (J m^{-2} s^{-1} ou W m^{-2})
fluxq	S, R (klon,klev)	flux de vapeur d'eau orienté positivement vers le haut (kg m^{-2} s^{-1})
fluxu	S, R (klon,klev)	flux de la vitesse u (tension du vent X): Pascal ou (kg m/s)/(m**2 s)
fluxv	S, R (klon,klev)	flux de la vitesse v (tension du vent Y): Pascal ou (kg m/s)/(m**2 s)
itop	S, I (klon)	numéro de couche pour le sommet de la couche limite
coefh,m	S, R (klon,klev)	coefficient d'échange (m^2/s), la première valeur indique la valeur du frottement du sol (sans unité)

3.2 clqh

Ce programme calcule la diffusion de l'humidité q et de l'enthalpie potentielle h ($=C_pT/P^\kappa$ où C_p est la capacité calorifique de l'air à pression fixée, T est la température en kelvin, P est la pression en mb, κ est soit R/C_p, soit 0.286)

```
      SUBROUTINE clqh(sig,dsig,s,sigs,dtime,
     .                coef,h,q,ts,psolmb,pskapa,radsol,cal,beta,
     .                difgrnd, dh, dq, dts, fluxt, fluxq)
```

sig	E, R (klev+1)	valeur σ pour chaque niveau
dsig	E, R (klev)	épaisseur de chaque couche en σ
s	E, R (klev)	valeur s pour chaque couche
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
dtime	E, R	intervalle du temps (en second)
coef	E, R (klon,klevcl)	coefficient d'échange (en m^2/s); la première valeur indique C_{drag} (sans unité)
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg/kg)
ts	E, R (klon)	température du sol (en Kelvin)
psolmb	E, R (klon)	pression au sol (en mb)
pskapa	E, R (klon)	exponentiel κ de $psolmb$
radsol	E, R (klon)	rayonnement net au sol (solaire et infrarouge) (W/m^2)
cal	E, R (klon)	si C indique la capacité calorifique surfacique du sol (en $J/m^2/K$), alors $cal = C_p/C$
beta	E, R (klon)	le rapport entre l'évaporation Rle et l'évapotranspiration (capacité maximale)
difgrnd	E, R (klon)	inverse du temps de relaxation de la température du sol vers la température profonde (utilisé seulement pour relaxer la glace de mer vers la température de congélation) (s^{-1})
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
dts	S, R (klon)	incrémentation de la température du sol (ts)
fluxt	S, R (klon,klevcl)	variable diagnostique indiquant le flux de la chaleur sensible, c'est-à-dire, le flux de C_pT (unité en W/m^2) (orientation positive vers le bas)
fluxq	S, R (klon,klevcl)	variable diagnostique indiquant le flux de la vapeur d'eau ($kg/m^2/s$)

3.3 clvent

Ce programme calcule la diffusion de la vitesse

```
      SUBROUTINE clvent(klon,klev,llmcl,sig,dsig,s,sigs,dtime,
     .                  coef,h,ven,psolmb,pskapa,
     .                  dven,fluxv)
```

klon	E, I	nombre de points
klev	E, I	nombre de couches verticales
klevcl	E, I	nombre de couches dans la couche limite (< ou = klev)
sig	E, R (klev+1)	valeur σ pour chaque niveau
dsig	E, R (klev)	épaisseur de chaque couche en σ
s	E, R (klev)	valeur s pour chaque couche
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
dtime	E, R	intervalle du temps (en second)
coef	E, R (klon,klevcl)	coefficient d'échange (en m^2/s); la première valeur indique C_{drag} (sans unité)
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
ven	E, R (klon,klev)	une composante de la vitesse verticale (en m/s)
psolmb	E, R (klon)	pression au sol (en mb)
pskapa	E, R (klon)	exponentiel κ de $psolmb$
dven	S, R (klon,klev)	incrémentation de ven
fluxv	S, R (klon,klevcl)	variable diagnostique indiquant le flux vertical de la vitesse ven. l'unité est: $(kg)(m/s)/(m^2s)$

3.4 coefkz

Ce programme calcule le coefficient de frottement au sol et les coefficients d'échange dans l'atmosphère.

```
      SUBROUTINE coefkz(s, sig, sigs, psolpa, pskapa,
     .                  ts, rugos, rnat, xlat,
     .                  u,v,h,q,
     .                  itop, pcfm, pcfh)
```

s	E, R (klev)	valeur s de chaque couche
sig	E, R (klev+1)	valeur σ de chaque niveau
sigs	E, R (klev)	valeur σ de chaque couche
psolpa	E, R (klon)	pression au sol (en Pascal)
pskapa	E, R (klon)	exponentiel κ de *psolpa*
ts	E, R (klon)	température du sol en kelvin
rugos	E, R (klon)	rugosité du sol (en m)
xlat	E, R (klon)	latitudes en degré
u,v	E, R (klon,klev)	deux composantes du vent (en m/s)
h	E, R (klon,klev)	enthalpie potentielle
q	E, R (klon,klev)	humidité spécifique (en kg/kg)
pcfm	S, R (klon,klev)	coefficients d'échange (en m^2/s) pour la vitesse. Les valeurs (i,1) indiquent les coefficients de frottement au sol.
pcfh	S, R (klon,klev)	comme *pcfm*, mais pour la chaleur sensible et la chaleur latente

3.5 calbeta

Ce programme donne des coefficients de surface avant d'appliquer la couche limite

```
SUBROUTINE calbeta(dtime,rnat,snow,qsol,
.                  vbeta,vcal,vdif)
```

dtime	E, R	intervalle du temps (en secondes)
rnat	E, R (klon)	indicateur de la nature du sol: (0:ocean; 1:terre; 2:glace terrestre; 3:glace oceanique)
snow	E, R (klon)	épaisseur neigeuse au sol (mm)
qsol	E, R (klon)	humidité du sol (mm)
vbeta	S, R (klon)	coefficient d'évaporation réelle (entre 0 et 1)
vcal	S, R (klon)	inverse de la capacité effective du sol
vdif	S, R (klon)	inverse du temps de relaxation de la température du sol vers la température profonde (utilisé seulement pour relaxer la glace de mer vers la température de congélation) (s^{-1})

3.6 hydrol

Etant donné que la température du sol est calculée dans la couche limite, le traitement du sol ici est limité à l'hydrologie. Pour l'instant, aucune information de la végétation est prise en compte.

```
      SUBROUTINE hydrol(dtime,rnat,rain_fall,snow_fall,evap,
     .                  tsol,qsol,snow)
```

dtime	S, R	intervalle du temps (en secondes)
rnat	E, R (klon)	indicateur de la nature du sol: (0:ocean; 1:terre; 2:glace terrestre; 3:glace océanique)
rainfall	E, R (klon)	taux de pluie (mm s^{-1} ou kg m^{-2} s^{-1})
snowfall	E, R (klon)	taux d'eneigement (mm s^{-1} ou kg m^{-2} s^{-1})
evap	E, R (klon)	taux d'évaporation (mm s^{-1} ou kg m^{-2} s^{-1})
tsol	E/S, R (klon)	température du sol (Kelvin)
qsol	E/S, R (klon)	humidité du sol (mm)
snow	E/S, R (klon)	neige du sol (mm)

4 Instabilité verticale et convection dans l'atmosphère

4.1 ajsec

C'est l'ajustement sec qui sert à un complémentaire pour les schémas de Kuo et de Manabe

```
SUBROUTINE ajsec(s, dsig, h, dh)
```

s	E, R (klev)	valeur de s pour chaque couche
dsig	E, R (klev)	épaisseur de chaque couche en σ
h	E, R (klon,klev)	enthalpie potentielle
dh	E, R (klon,klev)	incrémentation de l'enthalpie potentielle

Remarque: ce programme a des problèmes sur les stations de travail quand l'on utilise certaines options d'optimisation.

4.2 conman ou fiajh

C'est le schéma de l'ajustement humide, initialement proposé par Manabe

```
      SUBROUTINE conman (dtime, s, sig, sigs, dsig
     e                 , h, q, psolpa, pskapa
     s                 , dh, dq, rain, snow, ibas, itop)
```

dtime	E, R	intervalle du temps (en secondes)
s	E, R (klev)	valeur s pour chaque couche
sig	E, R (klev+1)	valeur σ pour chaque niveau
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
dsig	E, R (klev)	épaisseur de chaque couche en σ
psolpa	E, R (klon)	pression au sol (en Pascal)
pskapa	E, R (klon)	exponentiel κ de $psolmb$
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg/kg)
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
rain	S, R (klon)	pluie engendrée pendant l'ajustement (en mm/s)
snow	S, R (klon)	neige engendrée pendant l'ajustement (en mm/s)
ibas	S, I (klon)	numéro de couche du bas de la convection
itop	S, I (klon)	numéro de couche du sommet de la convection

4.3 conkuo

Ce schéma de convection est basé sur l'idée initialement proposée par Kuo, mais fortement modifiée par Laval au LMD.

```
      SUBROUTINE conkuo(dtime, lbas, s, sigs, dsig
     e                , psolmb, pskapa, h, q, conv_q
     s                , dh, dq, rain, snow, ltop)
```

dtime	E, R	intervalle du temps (en secondes)
lbas	E, I	numéro de la couche qui représente la base de la convection
ltop	E, I	numéro de la couche qui représente le sommet de la convection
dsig	E, R (klev)	épaisseur de chaque couche en σ
s	E, R (klev)	valeur s pour chaque couche
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
psolmb	E, R (klon)	pression au sol (en mb)
pskapa	E, R (klon)	exponentiel κ de $psolmb$
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg/kg)
convq	E, R (klon,klev)	taux de convergence de l'humidité en $kg/kg/s$)
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
rain	S, R (klon)	pluie obtenue en $mm/jour$
snow	S, R (klon)	neige obtenue en $mm/jour$
ltop	S, I (klon)	numéro de couche indiquant le sommet de la convection

4.4 conema

C'est le schéma de convection d'Emanuel. Pour l'instant, le programme a été écrit pour une seule colonne (donc, pas de vectorisation possible et très coûteux).

```
      SUBROUTINE conema (dtime,s,sigs, sig, psolpa, pskapa,
     .                   h, q, work1, work2, dh, dq, rain, snow,
     .                   kbas,ktop, upwd,dnwd,dnwdbis)
```

dtime	E, R	intervalle du temps (en secondes)
sig	E, R (klev+1)	valeur σ pour chaque niveau
s	E, R (klev)	valeur s pour chaque couche
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
psolpa	E, R (klon)	pression au sol (en Pascal)
pskapa	E, R (klon)	exponentiel κ de $psolmb$
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg/kg)
work1,2	E/S, R (klon,klev)	deux tableaux de travail dont les valeurs doivent être mémorisées entre deux appels successifs (au début, on peut les mettre à zéro)
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
rain	S, R (klon)	pluie obtenue en $mm/jour$
snow	S, R (klon)	neige obtenue en $mm/jour$
kbas	S, I (klon)	numéro de couche du bas de la convection
ktop	S, I (klon)	numéro de couche du haut de la convection
upwd	S, R (klon,klev)	variable diagnostique indiquant le flux de masse montante dans la convection (unité: kg m^{-2}/s^{-1})
dnwd	S, R (klon,klev)	variable diagnostique indiquant le flux de masse descentante dans la convection (unité: kg m^{-2}/s^{-1})
dnwdbis	S, R (klon,klev)	variable diagnostique indiquant le flux de masse descentante en dehors de la convection, entraînée par la précipitation (unité: kg m^{-2}/s^{-1})

4.5 contie

C'est le schéma de convection de Tiedtke, utilisé dans le modèle du Centre Européen. Avant de l'appeler, il faut initialiser les constantes utilisées dans la paramétrisation (*sucumf*).

```
 SUBROUTINE contie (dtime,sigs,sig,s,psolpa,pskapa,rnat,
e                  h, q, u, v, con_h, con_q, omega, pqhfl, tsol,
s                  d_h, d_q, d_u,d_v,rain, snow, kbas, ktop)
```

dtime	E, R	intervalle du temps (en secondes)
sig	E, R (klev+1)	valeur σ pour chaque niveau
s	E, R (klev)	valeur s pour chaque couche
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
psolpa	E, R (klon)	pression au sol (en Pascal)
pskapa	E, R (klon)	exponentiel κ de $psolpa$
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg/kg)
convh	E, R (klon,klev)	taux de convergence de l'enthalpie potentielle h
convq	E, R (klon,klev)	taux de convergence de l'humidité en $kg/kg/s$)
omega	E, R (klon,klev)	vitesse verticale (en Pa s^{-1} ?)
pqhfl	E, R (klon)	flux d'humidité du sol vers l'atmosphère (évaporation), l'unité est de $kg/m^2/s$ (à verifier)
tsol	E, R (klon)	température du sol en kelvin
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
rain	S, R (klon)	pluie obtenue en $mm/jour$
snow	S, R (klon)	neige obtenue en $mm/jour$
kbas	S, I (klon)	numéro de couche du bas de la convection
ktop	S, I (klon)	numéro de couche du haut de la convection

4.6 conflx

C'est le schéma de convection proposé par Tiedtke, mais modifié légèrement dans l'adaptation du code. Le mot-clé "flx" désigne "schéma flux de masse". Ce programme est l'interface pour le programme "flxmain" qui appelle "flxini", "flxbase", "flxasc", "flxdlfs", "flxddraf", "flxflux" et "flxdtdq". Attention, dans tous les programmes "flx*", l'ordre des couches discretisées incrémente du sommet de l'atmosphère au sol. Dans le soucis de la stabilité numérique du code, le calcul se fait en double précision (mais les entrées et sorties en simple précision). L'expérience semble de dire que la double précision ne serait pas nécessaire. (Attention: les fonctions thermodynamiques sont déclarées en simple précision)

```
      SUBROUTINE conflx (dtime,sigs,sig,s,psolpa,pskapa,rnat,
     e                   h, q, con_q, pqhfl,
     s                   pmfu, pmfd, d_h, d_q, rain, snow, kbas, ktop)
```

dtime	E, R	intervalle du temps (en secondes)
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
sig	E, R (klev+1)	valeur σ pour chaque niveau
s	E, R (klev)	valeur s pour chaque couche
psolpa	E, R (klon)	pression au sol (en Pascal)
pskapa	E, R (klon)	exponentiel κ de *psolpa*
rnat	E, R (klon)	nature du sol (terre/océan)
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg/kg)
convq	E, R (klon,klev)	taux de convergence de l'humidité en $kg/kg/s$)
pqhfl	E, R (klon)	(à une signe $-$ près) flux d'humidité du sol vers l'atmosphère (évaporation), l'unité est de $kg/m^2/s$
pmfu	S, R (klon,klev)	masse de flux ascendant dans la tour convective ($kg/m^2/s$)
pmfd	S, R (klon,klev)	masse de flux descendant dans la tour convective ($kg/m^2/s$)
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
rain	S, R (klon)	pluie obtenue en $mm/jour$
snow	S, R (klon)	neige obtenue en $mm/jour$
kbas	S, I (klon)	numéro de couche du bas de la convection
ktop	S, I (klon)	numéro de couche du haut de la convection

4.7 flxini

Ce programme initialise les variables

```
      SUBROUTINE flxini(pten, pqen, pqsen, pgeo, paph, pgeoh, ptenh,
     .           pqenh, pqsenh, ptu, pqu, ptd, pqd, pmfd, pmfds, pmfdq,
     .           pdmfdp, pmfu, pmfus, pmfuq, pdmfup, pdpmel, plu, plude,
     .           klab,pen_u, pde_u, pen_d, pde_d)
```

pten	E, R (klon,klev)	température atmosphérique (environnement) (Kelvin)
pqen	E, R (klon,klev)	humidité spécifique (environnement) (kg/kg)
pqsen	E, R (klon,klev)	humidité spécifique saturante (environnement) (kg/kg)
pgeo	E, R (klon,klev)	géopotentiel (g * mètre)
paph	E, R (klon,klev+1)	pression aux demi-niveaux (Pascal)
pgeoh	S, R (klon,klev)	geopotentiel aux demi-niveaux
ptenh	S, R (klon,klev)	température (environnement) aux demi-niveaux
pqenh	S, R (klon,klev)	humidité spécifique (environnement) aux demi-niveaux
pqsenh	S, R (klon,klev)	humidité spécifique saturante (environnement) aux demi-niveaux
ptu	S, R (klon,klev)	température dans le panache ascendant
pqu	S, R (klon,klev)	humidité spécifique dans le panache ascendant
plu	S, R (klon,klev)	eau liquide nuageuse dans le panache ascendant
pmfu	S, R (klon,klev)	flux de masse du panache ascendant (kg m^{-2} s^{-1})
pmfus	S, R (klon,klev)	flux de l'énergie sèche dans le panache ascendant
pmfuq	S, R (klon,klev)	flux de l'humidité dans le panache ascendant
pdmfup	S, R (klon,klev)	quantité de l'eau précipitée dans le panache ascendant
plude	S, R (klon,klev)	quantité de l'eau liquide jetée du panache ascendant à l'environnement
pdpmel	S, R (klon,klev)	quantité de neige fondue
ptd	S, R (klon,klev)	température dans le panache descendant
pqd	S, R (klon,klev)	humidité spécifique dans le panache descendant
pmfd	S, R (klon,klev)	flux de masse du panache descendant (kg m^{-2} s^{-1})
pmfds	S, R (klon,klev)	flux de l'énergie sèche dans le panache descendant
pmfdq	S, R (klon,klev)	flux de l'humidité dans le panache descendant
pdmfdp	S, R (klon,klev)	quantité de l'eau précipitée dans le panache descendant

5 Condensation à grande échelle

5.1 lmd_satur

```
 SUBROUTINE lmd_satur (dtime,sigs,sig,s,psolpa,pskapa,
e                      h, q, pl, tsol,
s                      d_h, d_q, rain, snow, pfra, pl_new)
```

dtime	E, R	intervalle du temps (en secondes)
s	E, R (klev)	valeur s pour chaque couche
sig	E, R (klev+1)	valeur σ de chaque niveau
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg kg^{-1})
pl	E, R (klon,klev)	eau liquide (en kg kg^{-1})
psolpa	E, R (klon)	pression au sol (en Pa)
pskapa	E, R (klon)	exponentiel κ de *psolpa*
tsol	E, R (klon)	température du sol (K)
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
rain	S, R (klon)	pluie obtenue en $mm/jour$
snow	S, R (klon)	neige obtenue en $mm/jour$
pfra	S, R (klon,klev)	fraction nuageuse (de 0 à 1)
plnew	S, R (klon,klev)	eau liquide (en kg kg^{-1})

5.2 ec_satur

Ce programme est la procédure de condensation à grande échelle du Centre Européen. Se reporter aux documents concernés.

```
      SUBROUTINE ec_satur (dtime,sigs,sig,s,psolpa,pskapa,
     e                    h, q, tsol,
     s                    d_h, d_q, rain, snow)
```

dtime	E, R	intervalle du temps (en secondes)
sig	E, R (klev+1)	valeur σ de chaque niveau
s	E, R (klev)	valeur s pour chaque couche
sigs	E, R (klev)	la valeur σ correspondante de chaque niveau s
psolpa	E, R (klon)	pression au sol (en Pa)
pskapa	E, R (klon)	exponentiel κ de $psolpa$
h	E, R (klon,klev)	enthalpie potentielle (C_pT/P^κ)
q	E, R (klon,klev)	vapeur d'eau (en kg kg^{-1})
tsol	E, R (klon)	température du sol en kelvin
dh	S, R (klon,klev)	incrémentation de h
dq	S, R (klon,klev)	incrémentation de q
rain	S, R (klon)	pluie obtenue en mm $jour^{-1}$
snow	S, R (klon)	neige obtenue en mm $jour^{-1}$

5.3 q_sat

Ce programme calcule la vapeur d'eau saturante en connaissant la température et la pression

```
FUNCTION q_sat(kelvin, milibar)
```

q_sat	S, R	humidité spécifique en kg/kg
kelvin	E, R	température en kelvin
milibar	E, R	pression en mb